

JULIÁN RENÉ MUÑOZ BURBANO



**Universidad
del Cauca**

**ESTIMACIÓN DE LA ALTURA DE UNA FORMA DENTRO
DE UNA IMAGEN 2D A PARTIR DE SU SOMBRA POR
MEDIO DE REDES NEURONALES**

Universidad del Cauca

**Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones**

Grupo I+D en Nuevas Tecnologías en Telecomunicaciones - GNTT
Línea de investigación: Señales y Sistemas de Telecomunicaciones
Popayán, noviembre de 2023

ESTIMACIÓN DE LA ALTURA DE UNA FORMA DENTRO DE UNA IMAGEN 2D A PARTIR DE SU SOMBRA POR MEDIO DE REDES NEURONALES



Universidad
del Cauca

JULIÁN RENÉ MUÑOZ BURBANO

Director:

Dr. Pablo Emilio Jojoa Gómez

Codirector:

Dr. Fausto Miguel Castro Caicedo

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Telecomunicaciones

Grupo I+D en Nuevas Tecnologías en Telecomunicaciones - GNTT

Línea de investigación: Señales y Sistemas de Telecomunicaciones

Popayán, noviembre de 2023

JULIÁN RENÉ MUÑOZ BURBANO

**ESTIMACIÓN DE LA ALTURA DE UNA FORMA DENTRO DE UNA
IMAGEN 2D A PARTIR DE SU SOMBRA POR MEDIO DE REDES
NEURONALES**

**Trabajo de grado presentado a la Facultad de
Ingeniería y Electrónica y Telecomunicaciones de
la Universidad del Cauca para la obtención del
Título de Magister**

Magister en:

ELECTRÓNICA Y TELECOMUNICACIONES

Director:

Dr. Pablo Emilio Jojoa Gómez

Codirector:

Dr. Fausto Miguel Castro Caicedo

**Popayán
2024**

Página de Aprobación

*A mis hermosos padres Efrén y Eunice
por todo su amor y fe en mí.
A mi hermano Ariel por su confianza.
A mi amor, amiga y esposa Diana
por todo su amor, cuidados y respaldo
quien ha transformado mi vida para bien.*

Agradecimientos

En primer lugar, mi agradecimiento infinito a mis tutores y profesores, cuya sabiduría y dedicación han sido guía en este trabajo de investigación. Su paciencia, rigor académico y pasión por la enseñanza han sido cruciales para mi formación.

En particular quiero agradecer especialmente al PhD. Pablo Emilio Jojoa Gómez, docente investigador de la Universidad del Cauca, por su fe y confianza en mí para llevar a feliz término este trabajo de investigación. Su orientación y experiencia han sido fundamentales para mi crecimiento personal y profesional. Al PhD. Fausto Miguel Castro Caicedo, docente investigador de la Universidad del Cauca, por su invaluable aporte y orientación. Al PhD. José Giovanni López Perafán por disposición y orientación académica. A la Mg. Virginia Solarte Muñoz por sus consejos y orientación. Agradecimiento a la Ingeniera Victoria Eugenia Patiño Arenas por apoyo académico en este proceso. Al Mg. Neil Andrés Jiménez Pinzón por su apoyo y amistad.

Quiero dedicar un reconocimiento especial a la memoria de mis padres, Eunice y Efrén, cuyo amor, enseñanzas y valores continúan siendo una fuente de inspiración y fortaleza en mi vida. A mi hermano Ariel, por su apoyo incondicional y por estar siempre a mi lado. A mi esposa Diana, compañera de vida, cuyo amor, comprensión y apoyo constante han sido un pilar fundamental en los momentos más desafiantes de esta etapa.

Mi gratitud también a mis compañeros de estudios, quienes me han brindaron su apoyo y amistad.

Finalmente, este logro no es solo un reflejo esfuerzo personal, sino también la suma de muchas contribuciones valiosas. A todos ustedes, mi más profundo agradecimiento.

Resumen

En esta investigación se desarrollaron e implementaron dos modelos de redes neuronales, un modelo 1, de una sola entrada, la sombra y un modelo 2, con dos entradas la figura y sombra. Estos se evaluaron en 600, 800, 1,000, épocas respectivamente, enfocados en estimar la altura de objetos dentro de una imagen *2D* a partir de su sombra. En su desarrollo se utilizaron técnicas de aumento de datos y regularización, para prevenir el sobreajuste y mantener la complejidad del modelo. Estos se diseñaron para predecir alturas dentro de un rango específico (1 a 10 cm) donde ambos modelos tienen un equilibrio entre precisión y capacidad de generalización y tolerancia al error (0,02) en las predicciones. Para ello, se construyó un conjunto de datos con imágenes fotográficas de formas u objetos, así como sus sombras proyectadas desde diferentes ángulos y ubicaciones. La estructura del conjunto de datos está conformada por el nombre de la forma, el nombre de la sombra, la longitud de la sombra, la altura de la forma y el ángulo de la fuente de luz. En este sentido, la investigación se centra en el análisis de la luz y sombras, en diferentes formas geométricas u objetos dentro de una imagen *2D*, donde la sombra proyectada es la información a utilizar y con la que se busca determinar la altura de las formas u objetos. El motivo principal de esta investigación está orientado a que, en un futuro, personas que presentan discapacidad visual ya sea total o parcial, puedan mediante un mecanismo táctil, definir o diferenciar un objeto, que ha sido recreado de forma tridimensional a partir de una imagen *2D* (fotografía o pintura). Por ello, es fundamental que a partir de una imagen *2D* se resalten los objetos a representar en una maqueta *2.5D*, donde la altura de los objetos seleccionados será la clave para crear la maqueta. En este sentido, en este documento se presenta un modelo que permite determinar la altura de objetos predefinidos en una imagen bidimensional a partir de su sombra. Se obtuvo un *dataset* que incluye imágenes fotográficas de formas u objetos con sus sombras proyectadas, un *dataframe* con la información del conjunto de datos. En los dos modelos (1 y 2) la combinación de las características visuales (figura y sombra) con el *dataframe*, mejoró notablemente las predicciones en los dos modelos, lo que potencialmente corrige ambigüedades al momento de interpretar la forma de un objeto. A partir de los resultados se realizó una evaluación integral de los dos mismos, que sobresalen por su equilibrio entre precisión y generalización. Estos se analizaron con diferentes métricas de evaluación como *MAE*, *MSE*, R^2 , y correlación de *Pearson* (r) entre otras.

Palabras clave: sombras, formas, red neuronal convolucional, *2D*, altura, *DataFrame*.

Abstract

In this research, two neural network models were developed and implemented, a model 1, with a single input, the shadow, and a model 2, with two inputs, the figure and shadow. These were evaluated at 600, 800, 1,000, epochs respectively, focused on estimating the height of objects within a 2D image from their shadow. Data augmentation and regularization techniques were used in their development to prevent over-fitting and maintain model complexity. These were designed to predict heights within a specific range (1 to 10 cm) where both models have a balance between accuracy and generalization capacity and tolerance to error (0.02) in the predictions. For this purpose, a dataset was constructed with photographic images of shapes or objects, as well as their shadows projected from different angles and locations. The structure of the dataset consists of the name of the shape, the name of the shadow, the length of the shadow, the height of the shape and the angle of the light source. In this sense, the research focuses on the analysis of light and shadows, in different geometric shapes or objects within a 2D image, where the projected shadow is the information to be used and with which we seek to determine the height of the shapes or objects. The main reason for this research is oriented to the possibility that, in the future, people with visual impairment, either total or partial, may use a tactile mechanism to define or differentiate an object, which has been recreated in three-dimensional form from a 2D image (photograph or painting). Therefore, it is essential that from a 2D image the objects to be represented in a 2.5D model are highlighted, where the height of the selected objects will be the key to create the model. In this sense, this paper presents a model that allows to determine the height of predefined objects in a two-dimensional image from their shadow. A dataset was obtained that includes photographic images of shapes or objects with their projected shadows, a DataFrame with the information of the dataset. In the two models (1 and 2) the combination of the visual characteristics (figure and shadow) with the DataFrame, significantly improved the predictions in the two models, which potentially corrects ambiguities when interpreting the shape of an object. Based on the results, a comprehensive evaluation of the two models, which stand out for their balance between accuracy and generalization, was performed. These were analyzed with different evaluation metrics such as MAE, MSE, R^2 , and Pearson's correlation (r) among others.

Keywords: shadows, shapes, convolutional neural network, 2D, height, DataFrame.

Contenido

	Pág.
Lista de Figuras	XVII
Lista de Tablas	XXI
Lista de Símbolos	XXII
Lista de Abreviaturas	XXIII
Capítulo 1. Introducción	23
1.1. Motivación de la Investigación	25
1.2. Organización del documento	27
Capítulo 2. Fundamentos Teóricos	29
2.1. La Red Neuronal Artificial	29
2.2. La sombra (<i>The Shadow</i>).....	31
2.3. Métodos y Algoritmos para el procesamiento de sombras.....	34
2.3.1. Algoritmos para detección de sombras.....	38
2.3.2. Algoritmos para Análisis de sombras.....	39
2.4. Tipos de Redes Neuronales.....	41
2.5. Redes Neuronales Convolucionales (<i>Convolutional Neural Networks</i>).....	42
2.5.1. ¿Qué es una Red Neuronal Convolutiva? (<i>CNN</i>).....	42
2.5.2. Parámetros e hiperparámetros	48
2.5.3. El Conjunto de Datos (<i>Dataset</i>)	51
2.5.4. El <i>DataFrame</i>	54
2.5.5. Regresión	55
2.5.6. Normalización	58
2.5.7. Generalización	58
2.5.8. <i>Ensembles</i> : combinación de modelos.....	60
2.6. Métricas de desempeño en modelos de regresión.....	61
2.6.1. Error Absoluto Medio (<i>Mean Absolute Error, MAE</i>)	62
2.6.2. Error Cuadrático Medio (<i>Mean Square error, MSE</i>).....	62

2.6.3. Raíz del error cuadrático medio (<i>RMSE - Root Mean Squared Error</i>)	63
2.6.4. Coeficiente de determinación (<i>R2</i>).....	63
2.6.5. Coeficiente de Pearson o de correlación de Pearson (<i>r</i>)	64
2.6.6. Suma de los Cuadrados Residuos (<i>Residual Sum of Squares, RSS</i>)	66
2.6.7. Método de validación cruzada (<i>Cross Validación Method</i>)	67
2.7. Metodología de implementación	70
Capítulo 3. Modelo Propuesto	73
3.1. Descripción general del modelo	73
3.2. Implementación de la metodología	73
3.3. Construcción del <i>dataset</i>	82
3.3.1. Descripción del conjunto de Datos (<i>Dataset</i>).....	91
3.3.2. Estructura del conjunto de datos (<i>Dataset</i>).....	93
3.4. Arquitectura.....	98
3.4.1. Arquitectura del modelo No.2 (dos entradas):	99
3.4.2. Arquitectura del modelo No.1 (una entrada):	102
3.5. Diseño del modelo	107
3.6. Descripción general del algoritmo	107
Capítulo IV. Implementación y experiencias realizadas	111
4.1. Validación cruzada (<i>k-Fold Cross Validation</i>)	111
4.1.1. Validación cruzada modelo No.2. $k = 5$	112
4.1.2. Validación cruzada modelo No.2. $k = 10$	113
4.1.3. Validación cruzada modelo No.2. $k = 20$	114
4.1.4. Análisis validación cruzada del modelo No.2.....	116
4.1.5. Validación cruzada modelo No.1. $k = 5$	118
4.1.6. Validación cruzada modelo No.1. $k = 10$	119
4.1.7. Validación cruzada modelo No.1. $k = 20$	120
4.1.8. Análisis validación cruzada del modelo No.1.....	122
4.2. Análisis integral del modelo No.2. y No.1.....	123
4.2.1. Resultados gráficos y cuantitativos: modelo No.2. con 1.250 épocas.....	124
4.2.2. Resultados gráficos y cuantitativos: modelo No.1. con 1.250 épocas.....	132
4.3. Resultados y tablas comparativas de los modelos No.2. y No.1.....	143
Capítulo V. Conclusiones y Trabajos Futuros	166

5.1. Conclusiones.....	166
5.2. Trabajos futuros	167
Referencias bibliográficas	169
ANEXO A. Elementos principales de la maqueta	181
ANEXO B. Estructura general del algoritmo	182
ANEXO C. Explicación del modelo No. 1. por capas detallado.....	183
ANEXO D. Explicación del modelo No. 2. por capas detallado.....	185
ANEXO E. Entorno y herramientas de desarrollo.	187
E.1. TensorFlow	187
E.2. Keras	187
E.3. Matlab.....	188
E.4. TensorBoard.....	190
E.5. Netron.....	191

Lista de Figuras

	Pág.	
1.1	Altura de una forma geométrica	15
2.1	Esquema de una Red Neuronal Artificial	28
2.2	Arquitecturas de una red neuronal	31
2.3	Luz y sombras de una esfera	32
2.4	Mapa de características: cada filtro crea un mapa de características diferente que detecta distintos patrones en la imagen (Goldberg, M. and Gougeon, F. 1980)	43
2.5	Estructura de una red neuronal convolucional. (Adaptado de López Pinaya, W.H. et al., (2020)	44
2.6	Arquitectura de una CNN (Adaptado de Loy J., (2019)	45
2.7	La arquitectura multimodal propuesta por (Khvostikov <i>et al.</i> , 2018).	46
2.8	Filtro en una <i>CNN</i> (Ikeuchi, K. <i>et al.</i> , 2020)	46
2.9	Operación de convolución (López Pinaya, W.H. <i>et al.</i> , 2020)	47
2.10	(Izquierda) operación producto punto. (Derecha) La operación de convolución (López Pinaya, W.H. <i>et al.</i> , 2020)	48
2.11	Importancia de la tasa de aprendizaje	50
2.12	Relación conceptual entre la dimensionalidad de los datos y la complejidad del modelo	53
2.13	Ejemplo de un DataFrame (creación propia)	54
2.14	Línea de mejor ajuste en regresión lineal (recta de regresión)	56
2.15	Ejemplo de la predicción del modelo: relación entre valores reales vs. valores predichos	58
2.16	Generalización. Izquierda: conjunto de entrenamiento y sobreajustado (overfitting). Derecha: conjunto de validación y modelo generalizado	59
2.17	Gráfico del coeficiente de determinación (R^2) en una CNN con 50 épocas.	64
2.18	Gráfico de dispersión y coeficiente de correlación de Pearson (r). (creación propia)	65
2.19	Ejemplo de la suma de los cuadrados de los residuos (RSS). (creación propia)	66
2.20	Caracterización de la validación cruzada. (Berzal, 2018).	67
2.21.	Visualización validación cruzada (<i>Cross-Validación</i>) con $k = 5$ (<i>folds</i>). (creación propia)	68
2.22.	Resultados de la validación cruzada	69
2.23.	Etapas de desarrollo del modelo	70

3.1.	Fotografías de tomadas de dos superficies, una de madera y otra superficie de madera cubierta con vinilo de color blanco mate	79
3.2.	Margen de error en al área segmentación de la imagen de la sombra en el preprocesamiento	81
3.3.	Imagen de la figura, sombra y el dataframe	83
3.4.	Modelo 1 (una entrada): sombra y dataframe	84
3.5.	Medida de lúmenes en tres niveles de intensidad de la lámpara led	85
3.6.	Sombras solapadas (creación propia)	85
3.7.	Imagen de un cilindro con sombras diferentes	86
3.8.	Guía general de formación de sombras de una figura al rotar. (creación propia)	87
3.9.	Altura de todos los cilindros (creación propia)	88
3.10.	Fotografías de sombras de figuras obtenidas al rotar la fuente de iluminación. (creación propia)	88
3.11.	Imágenes fotografías con un nivel de iluminación Imágenes fotografías con niveles de iluminación e imágenes	89
3.12.	Altura de un cilindro, esfera y cubo y sus respectivas sombras (dataset). (creación propia)	90
3.13.	Longitud de la sombra de un cilindro (creación propia)	90
3.14.	Vista general de figuras usadas para la generación del dataset. (creación propia)	92
3.15.	Izquierda: Fotografías del cubo con sombras respectivas. Derecha: fotografías del cilindro con sombras respectivas	94
3.16.	(a), (b) cilindro y esfera con la primera ubicación de la fuente de iluminación (lámpara).	95
3.17.	Rotación especificada en grados de la lámpara alrededor de la superficie circular	95
3.18.	Nombres de las imágenes de las figuras y de las sombras. (creación propia)	96
3.19.	Estructura del dataset de entrenamiento, validación y prueba. (creación propia)	97
3.20.	Descripción arquitectura del modelo No.2.	99
3.21.	Resumen de la arquitectura del Modelo No.2. (creación propia)	100
3.22.	Resumen de la arquitectura del modelo No.1.	103
3.23.	Arquitectura del Modelo No.1. (creación propia)	104
	Esquema General del modelo propuesto. (creación propia)	107
4.1.	Entrenamiento y validación MSE (Error Cuadrático Medio) para altura modelo No.2. (creación propia)	124
4.2.	Métricas MSE cambia a lo largo de las épocas. (creación propia)	125
4.3.	Entrenamiento y validación MAE (Error Absoluto Medio) para altura Modelo No.2. (creación propia)	125
4.4.	Pérdida en entrenamiento y validación. Modelo No.2. (creación propia)	126

4.5.	Coeficiente de determinación R^2 en función de las épocas modelo No.2. (creación propia)	126
4.6.	Curva de aprendizaje del modelo No.2. (creación propia)	128
4.7.	Gráficos de Puntos (Scatter Plots) para MAE y MSE. modelo No.2. (creación propia)	128
4.8.	Gráficos de densidad y validación MAE modelo No.2. (creación propia)	129
4.9.	Gráfico de dispersión del modelo No.2. (creación propia)	129
4.10.	Histograma de las diferencias entre las alturas reales y las alturas predichas modelo No.2. (creación propia)	130
4.11.	Gráfico de caja del modelo No.2. Errores de predicción por Tipo de figura (creación propia)	131
4.12.	Gráfico de residuos del modelo No.2. (creación propia)	131
4.13.	Error cuadrático medio (MSE) en entrenamiento y validación en función de las épocas del modelo No.1. (creación propia)	133
4.14.	Entrenamiento y validación MAE vs épocas del modelo No.1. con 1.250 épocas. (creación propia)	134
4.15.	Pérdida del modelo en el entrenamiento y la validación, modelo No.1. (creación propia)	135
4.16.	Gráfica de la curva de aprendizaje modelo No.1. (creación propia)	137
4.17.	Gráfica de puntos del MAE (entrenamiento y validación) y MSE (entrenamiento y validación) modelo No.1. (creación propia)	138
4.18.	Gráficos de densidad para entrenamiento y validación de las métricas MAE y MSE modelo No.1. (creación propia)	139
4.19.	Gráfica de dispersión: comparación entre alturas reales y alturas predichas modelo No.1. (creación propia)	139
4.20.	Histograma de diferencias entre las alturas reales y las alturas predichas modelo No.1. (creación propia)	141
4.21.	Errores de predicción por tipo de figura modelo No.1 con 1.250 épocas (creación propia)	142
4.22.	Gráfica de Residuos modelo No.1. (creación propia)	143
4.23.	MAE, MSE y Loss del modelo 1 con regularización L1 y aumento de datos con 1.000 épocas	149
4.24.	Gráfica de densidad del MAE y del MSE y de residuos modelo 1 con regularización L1 y aumento de datos con 1.000 épocas	149
4.25.	Gráfica de dispersión y gráfico de caja del modelo 1 con regularización L1 y aumento de datos con 1.000 épocas	150
4.26.	Modelo 1 con regularización L2 y aumento de datos con 1.000 épocas	151
4.27.	Gráfica de dispersión, gráfico de caja del modelo 1 con regularización L2 y aumento de datos con 1.000 épocas	151
4.28.	Gráfica de densidad del MAE, MSE y de residuos del modelo 1 con regularización L2 y aumento de datos con 1.000 épocas	152
4.29.	Modelo 1 con regularización L1 y L2 aumento de datos con 1.000 épocas	152

4.30.	Gráfica de dispersión, gráfico de caja del modelo 1 con regularización L1 y L2 y aumento de datos con 1.000 épocas	153
4.31.	Gráfica de densidad del MAE, MSE y de residuos del modelo 1 con regularización L1 y L2 y aumento de datos con 1.000 épocas	153
4.32.	Modelo 2 con regularización L1 y aumento de datos con 1.000 épocas	
4.33.	Gráfica de dispersión, gráfico de caja del modelo 2 con regularización L1 y aumento de datos con 1.000 épocas	154
4.34.	Gráfica de densidad del MAE, MSE y de residuos del modelo 2 con regularización L1 y aumento de datos con 1.000 épocas	155
4.35.	Modelo 2 con regularización L2 y aumento de datos con 1.000 épocas	155
4.36.	Gráfica de dispersión, gráfico de caja del modelo 2 con regularización L2 y aumento de datos con 1.000 épocas	156
4.37.	Gráfica de densidad del MAE, MSE y de residuos del modelo 2 con regularización L2 y aumento de datos con 1.000 épocas	157
4.38.	Modelo 2 con regularización L2 y aumento de datos con 1.000 épocas	157
4.39.	Gráfica de dispersión, gráfico de caja del modelo 2 con regularización L1 y L2 y aumento de datos con 1.000 épocas	158
4.40.	Gráfica de densidad del MAE, MSE y de residuos del modelo 2 con regularización L1 y L2 y aumento de datos con 1.000 épocas	158
4.41.	Distribución de las diferencias Absolutas del Modelo 1 (M1_L2C_AUM) con 1.000 épocas (Aum. datos y Reg. L2)	162
4.42.	Distribución de las diferencias Absolutas del Modelo 2 (M2_L2C_AUM) con 1.000 épocas (Aum. datos y Reg. L2)	163
4.43.	Distribución de las diferencias Absolutas del Modelo 2 (M2_L2C_AUM) con 1.400 épocas (Aum. datos y Reg. L2)	163
5.1	Imágenes no uniformes con sombras proyectadas	168

Lista de Tablas

		Pág.
3.1	Variaciones en las imágenes, disociación objeto cámara, iluminación, etc.....	81
3.2	Alturas de las figuras y medidas en centímetros (creación propia).....	86
3.3	Distribución de alturas de las formas geométricas del dataset.(creación propia).....	92
3.3	Distribución de <i>dataset</i> 1. (creación propia).....	95
3.4	Estructura del <i>DataFrame</i> . (creación propia).....	97
4.1	Validaciones cruzadas del modelo No.2. con $k = 5$. (creación propia).....	113
4.2	Resultados Generales de la validación cruzada aplicada al modelo No.2. (creación propia).....	113
4.3	Validaciones cruzadas del modelo No.2. con $k = 10$. (creación propia).....	113,114
4.4	Resultados finales de la validación cruzada del modelo No.2. con $k=10$. (creación propia)	114
4.5	Validación cruzada del modelo No.2. con $k=20$. (creación propia)	115
4.6	Resultados finales de la validación cruzada del modelo No.2. con $k=20$. (creación propia)	116
4.7	Resumen de la validación cruzada del modelo No.1. (creación propia)	116
4.8	Resumen de la validación cruzada del modelo No.1. (creación propia)	116
4.9	Resumen de la validación cruzada del modelo No.1. (creación propia)	117
4.10	Validación cruzada del modelo No.1. con $k=5$. (creación propia)	118
4.11	Resultados generales de la validación cruzada aplicada al modelo No.1. (creación propia)	118
4.12	Validación cruzada del modelo No.1. con $k=10$. (creación propia)	119
4.13	Resultados finales de la validación cruzada del modelo No.1. con $k=10$. (creación propia)	120,121
4.14	Validación cruzada del modelo No.1. con $k=20$. (creación propia)	121
4.15	Resultados finales de la validación cruzada del modelo No.1. con $k=20$. (creación propia)	122
4.16	Resumen de validaciones cruzadas del modelo No.1. con $k=5$. (creación propia)	122
4.17	Resumen de validaciones cruzadas del modelo No.1. con $k=10$. (creación propia)	122
4.18	Resumen de validaciones cruzadas del modelo No.1. con $k=20$. (creación propia)	122
4.19	Valores de R^2 para la visualización rápida de cuál modelo. (creación propia)	127
4.20	Coefficiente de determinación R^2 , MAE y MSE del modelo No.1. en varias épocas. (creación propia)	136

4.21	Coeficiente de correlación (r), MAE y MSE del modelo No.1. (creación propia)	136
4.22	Proceso de experimentación del modelo 1 y del modelo 2 con ajustes: aumento de datos y regularización L1 y L2	144
4.23	Tabla comparativa modelo 1 una entrada (arriba) y modelo 2 de dos entradas (abajo)	146
4.24	Validación cruzada modelo 1 (izquierda) y modelo 2 (derecha), sin aumento de datos y sin regularización	147
4.25	Modelo 2. (dos entradas) con aumento de datos y regularización L1	159
4.26	Modelo 2. (dos entradas) con aumento de datos y regularización L2	159
4.27	Modelo 2. (dos entradas) con aumento de datos y regularización L1 y L2	159
4.28	Gráficos del modelo 1 (una entrada) con 1.000 épocas, aumento de datos y regularización L1	160
4.29	Gráficos del modelo 1 (una entrada) con 1.000 épocas, aumento de datos y regularización L2	160
4.30	Gráficos del modelo 1 (una entrada) con 1.000 épocas, aumento de datos y regularización L1 y L2	160

Lista de Abreviaturas

CNN: Convolutional Neural Network, Red Neuronal Convolutacional

ANS: Artificial Neural Systems

MSE: Mean Square Error

ML: Machine Learning

DL: El Deep Learning

LMS: Least Mean Square

RNA: Red Neuronal Artificial

RL: Reinforcement Learning

FC: Fully Connected

ANN: Artificial Neural Networks

CNN: Convolutional Neural Networks

MLP: Multilayer Neural Networks

MAE: Mean Absolute Error

MSE: Mean Square Error

GPU: Graphic Processing Units

RL: Reinforcement Learning

GAN`s: Generative Adversarial Neural Networks

Capítulo 1. Introducción

Para los investigadores que utilizan las fotografías como las de superficie de planetas como fuente de información, muchas de estas imágenes tienen ruido y/o ambigüedades e inconsistencias, lo que dificulta el adecuado reconocimiento y extracción de la información. Una de las posibilidades es observar si los objetos o formas contenidos en la imagen tienen sombras proyectadas, para así utilizar estas combinadas con la dirección de la fuente de iluminación, para encontrar atributos y/o características tales que den un indicio de cómo es su aspecto tridimensional y donde sea posible, utilizarla para obtener estimaciones de la altura correspondiente.

Por ejemplo, Chen, Z., *et al.* (2021) en su investigación utilizaron la fotogrametría, la cual requiere imágenes estereoscópicas (*High-Resolution Imaging Science Experiment HiRISE*), las cuales deben ser imágenes de alta resolución de Marte, estas imágenes son escasas y tienen ambigüedades, entre ellas la variabilidad de iluminación solar, polvo y partículas atmosféricas, algunas de ellas debido a los efectos atmosféricos del planeta que atenúa la radiancia media. Para esto se desarrolló una red neuronal convolucional (CNN), que han mostrado un rendimiento prometedor en la reconstrucción 3D de escenas a corta distancia: Este grupo de científicos desarrollaron un algoritmo basado en CNN que es capaz de generar un modelo digital de elevación DEM (*digital elevation models*) a partir de imágenes individuales; una de las características es que los DEM tienen las mismas resoluciones que las imágenes de entrada. En el proceso de entrenamiento propuesto se utilizaron dos conjuntos de datos. El de datos sintéticos para inicializar los parámetros de la red, y un conjunto de datos reales para ajustar esos parámetros. Esta estrategia se basa en el hecho de que la calidad de las imágenes capturadas por la Context Camera (CTX) varía. Los resultados experimentales de esta investigación, muestran que el método que proponen basado en CNNs tiene un gran potencial para la reconstrucción de superficies en 3D en aplicaciones planetarias. Según Chen, Z. *et al.* (2021) el principio de la entrenamiento de las CNNs, según el cual las relaciones entre las entradas y las salidas se estudian directamente, garantiza que las ambigüedades (como el ruido y las incertidumbres) se aprendan y eliminen implícitamente. Este método consideró elementos que causan ambigüedades en el proceso de reconstrucción 3D, como efectos atmosféricos, variaciones de albedo y ruido de la imagen para mejorar la precisión en los cálculos. En este punto los métodos de reconstrucción de superficies en 3D están basados en imágenes individuales, como (*Shape from shading, SfS*) y (*Shape and albedo from shading, SAfS*), donde las formas a partir de las sombras se utilizaron para analizar la intensidad de los píxeles de la imagen y de cómo la superficie planetaria refleja la irradiancia entrante, como la luz solar, afectando la calidad de las imágenes.

Ahora bien, a partir de lo anterior es posible construir modelos con diferentes objetos 3D a escala, donde por medio de una fuente de iluminación se proyecte la sombra de la figura u objeto, variando la ubicación e intensidad de la fuente de iluminación, observaciones que se capturan en fotografías, para que en una red neuronal convolucional se procesen, se obtenga información que permita determinar la apariencia y la altura a partir de su sombra.

Por otra parte, las personas con discapacidad visual, ya sea total o parcial (*Please Touch the Art: 3-D Printing Helps Visually Impaired Appreciate Paintings*)(tomado de <https://www.smithsonianmag.com/>), tienen dificultades para que a partir del tacto definan o diferencien un objeto, volviéndose imposible que identifiquen qué objetos hay en una imagen 2D (fotografía o pintura, por ejemplo) sin embargo una de las formas para que estas personas puedan tener una idea de lo que una imagen contiene es recreando la imagen a partir de objetos 3D (Memeo, M. *et al.*, 2021). Estos objetos pueden ser maquetas ya realizadas y definidas para un objeto específico o para una pintura en un museo por ejemplo o el uso de papel en relieve (*Art Museums Are Creating 3D Versions of Paintings for Visually Impaired People To Touch*) (tomado de mymodernmet.com).

Es importante mencionar que Tenthoff, *et al.* (2020) en su investigación perfeccionaron el algoritmo de la forma a partir de la sombra (*Shape from Shading, SfS*) el cual permite la reconstrucción 3D a partir de mosaicos de imágenes, que se ha utilizado anteriormente para crear modelos digitales del terreno (*digital terrain models MDT*) de las superficies lunar y marciana, con el propósito de generar MDTs de alta resolución de Mercurio a partir de imágenes. Bajo este contexto es viable la posibilidad de combinar varios modelos, incluido el que se propone en esta investigación, para obtener mejores resultados para una investigación futura.

Por tanto, con las ventajas tecnológicas actuales está abierta la posibilidad de utilizar tabletas con *tacxels* o las impresoras 3D, por lo que a partir de una imagen 2D es posible destacar los objetos que se van a representar en un modelo 2.5D, donde la altura de los objetos seleccionados serán la clave para crear el modelo, de ahí la importancia de determinar la altura de objetos predefinidos en una imagen bidimensional a partir de su sombra. En este sentido, este trabajo se enfocó en el análisis de luz y de las sombras de diferentes formas u objetos geométricos dentro de una imagen 2D, los cuales se proyectan a partir de una fuente de iluminación desde una ubicación definida. La sombra proyectada por la forma u objeto y la apariencia de la imagen es la información con la que se cuenta inicialmente, y con la que se busca llegar a obtener la altura de la forma u objeto.

Para obtener la altura del objeto o de la forma se usaron técnicas de *Machine Learning (ML)*, entre las que se destacan las redes neuronales convolucionales (*CNN*), que permiten la detección de patrones, clasificación de imágenes, información de las sombras, la extracción de características entre otras cosas, y que se consideraron importantes para el proceso de aprendizaje y de la experimentación de la red. Además, para obtener mejores resultados se desarrolló un conjunto de datos (*dataset*) de imágenes de formas u objetos y también de las sombras que proyectan, como la

observada en la figura 1.1. Estas son las imágenes que se utilizaron para entrenar, validar y probar la red neuronal convolucional.

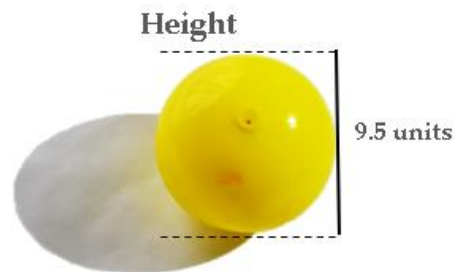


Figura 1.1. Altura de una forma geométrica

Con el objetivo de centrarse en los aspectos fundamentales de la investigación y evitar desviar el proceso investigativo hacia asuntos dispendiosos relacionados con la fuente de iluminación y de la forma del objeto, se definió que las sombras proyectadas de las formas u objetos sean generadas a partir de una sola fuente de iluminación y que las formas geométricas se limitaran únicamente al cubo, al cilindro y a la esfera con sus sombras respectivas.

Como se mencionó anteriormente, un aspecto relevante es que las sombras proyectadas de las formas u objetos son generadas a partir de una sola fuente de iluminación, pero esto no impide que sea posible aumentar la cantidad de imágenes del conjunto de datos con más fotografías desde otros ángulos, de otras formas y con más fuentes de iluminación, pero sería una tarea muy dispendiosa y que va más allá de lo planteado para este trabajo.

1.1. Motivación de la Investigación

Las tecnologías utilizadas en la implementación y procesamiento digital de imágenes desempeñan un papel importante en la extracción y análisis de la información contenida en las imágenes, permitiendo la identificación y la caracterización automática de objetos y formas mediante el uso de algoritmos. Su aplicación principal está orientada a que al procesarlas de forma automática mediante algoritmos pueden proporcionar detalles más precisos y detallados sobre dichos objetos. Se han investigado a los algoritmos que son utilizados para la detección y el procesamiento de las sombras, las cuales están asociadas a objetos o formas y bordes, áreas o tamaño, etc. que permiten inferir entre muchos otros aspectos, el tipo de iluminación, la textura, la ubicación, la forma, etc. Es así, que se utilizaron las redes neuronales artificiales y el procesamiento digital de imágenes para establecer una ruta de trabajo con el propósito de abordar la pregunta de investigación.

De este modo para esta investigación se desarrolló una red neuronal convolucional (CNN) con regresión, con un *dataset* o conjunto de datos adecuado para el desarrollo del trabajo. Este último conformado por las imágenes que en este caso son las fotografías

que mediante la elaboración de una maqueta especialmente acondicionada se tomaron, posteriormente se etiquetaron y organizaron en carpetas bajo determinados criterios. Estas fotografías fueron obtenidas bajo diferentes condiciones de iluminación, lo cual se presentará en un capítulo posterior con más detalle. En cuanto a la preparación de las imágenes o las formas, se preprocesaron de tal manera que, con la información obtenida, sea posible determinar las características geométricas tales como la altura, tipo de figura o forma, ancho, sombra, etc. Siendo más puntual, se entrenó una red neuronal convolucional (*CNN*) con el conjunto de datos (*dataset*) anteriormente indicado con el objetivo de predecir la altura de una forma o figura *2D* contenida en una imagen a partir de su sombra.

El conjunto de datos se creó totalmente para esta investigación, ya que contiene formas geométricas definidas y etiquetadas para el proceso de aprendizaje de la red neuronal. Por tanto, a la red neuronal se le proporcionó una entrada la cual tiene la sombra de un objeto o forma, y se espera que después del entrenamiento, ésta cuente con el conocimiento / aprendizaje necesario y sea capaz de identificar la forma y la altura del objeto a la cual corresponde.

Muchos trabajos e investigaciones han buscado soluciones de como a partir de una imagen *2D* se pueda reconstruir un modelo tridimensional, ya sea para fines lúdicos, o para dar solución a problemas de ingeniería o para ayudar a personas con discapacidad visual a “observar” con el tacto una imagen. Entre los diferentes problemas que se presentan están el determinar los objetos que se encuentran en una imagen, la ubicación espacial y su tamaño relativo. Una forma de afrontar estos problemas es determinar mediante la sombra el formato de los objetos y su tamaño relativo, en el proyecto actual, se ha trabajado en determinar la altura de objetos regulares (cubos, cilindros y esferas) mediante la sombra proyectada.

Una de las motivaciones al proponer esta investigación fue la posibilidad de que con la información que se recopile, procese y analice, mediante el uso de las redes neuronales se obtenga un modelo de aprendizaje profundo y un conjunto de datos, que pueden ser utilizados para encontrar la combinación adecuada de parámetros que permitan obtener un modelo para abordar el problema planteado. De ahí la importancia de este tema, porque se comprobó que es posible combinar el aprendizaje profundo, un conjunto de datos con imágenes, las sombras y el *DataFrame* en el desarrollo del algoritmo, que en la experimentación permitió obtener resultados óptimos y que son un punto de partida para realizar ajustes o mejoras que conduzcan a una mayor precisión, y eficiencia a problemas más complejos. Siendo este un referente teórico-práctico y un punto de partida a temas relacionados con las sombras, entendiendo que en estas hay más información de la que se puede percibir a simple vista.

1.2. Organización del documento

Este documento se dividió en cinco capítulos, en donde su estructura está diseñada para ofrecer un recorrido inicial y coherente a través del problema de investigación, el soporte teórico, la propuesta de un modelo, su implementación y los resultados obtenidos a partir de la experimentación, así como conclusiones y trabajos futuros.

Capítulo I: Introducción

Este capítulo introduce al lector en el contexto general de la investigación. Inicia con la presentación y el planteamiento del problema, motivando la importancia y relevancia del estudio. Se proporcionan antecedentes claves que establecen el estado actual del área de estudio y la justificación que respalda la necesidad de la investigación. El capítulo concluye delineando la metodología adoptada, así como los alcances y limitaciones del estudio.

Capítulo II: Fundamentos Teóricos

Aquí se encuentra el soporte teórico o marco teórico del trabajo relacionados con las redes neuronales artificiales, sus características y estructura enfatizando más en las redes neuronales artificiales convolucionales, el aprendizaje profundo, el aprendizaje automático. De igual manera se indican los algoritmos más importantes para la detección y el análisis de sombras.

Capítulo III: Modelo Propuesto

El núcleo del trabajo de investigación. En este capítulo se introduce y describe detalladamente la propuesta de solución al problema planteado. Se expone el diseño del modelo, los sistemas y algoritmos que lo componen, las herramientas software utilizados para su desarrollo e implementación, destacando el uso de *Keras* y *TensorFlow*. Se da especial atención al conjunto de datos o *dataset*, cómo fue elaborado, y las características de las fotografías incluidas en el.

Capítulo IV: Implementaciones y experiencias realizadas

Este capítulo, se enfoca en modelo propuesto. Se indica desde los procesos de experimentación, preparación del *DataFrame* con las pruebas realizadas. Se indica el número de experimentos realizados, y los resultados obtenidos en cada uno de ellos. El lector puede entender el método de investigación adoptado, principalmente enfocado en la experimentación y prueba y error.

Capítulo V: Conclusiones y Trabajos Futuros

Finalmente, se recapitula el estudio, resaltando los hallazgos más significativos y cómo estos contribuyen al campo de estudio. Se presentan conclusiones derivadas de

la investigación y se sugieren líneas de trabajo o enfoques para futuras investigaciones en el área.

El presente trabajo de grado de Maestría busca ser una contribución en el campo de de investigación de las redes neuronales y del procesamiento de imágenes.

Capítulo 2. Fundamentos Teóricos

Es en este capítulo proporciona una revisión exhaustiva y estructurada de las principales teorías, conceptos, y estudios que se relacionan directamente con el problema de investigación propuesto, mediante un recorrido lógico y coherente por la literatura existente, el lector podrá comprender los conceptos básicos de la investigación, las definiciones que guían nuestro estudio y las interrelaciones entre estos.

2.1. La Red Neuronal Artificial

Una red neuronal artificial (*Artificial Neural Network, ANN*) es un modelo matemático diseñado para aprender a partir de datos, y realizar tareas específicas de manera autónoma. Consta de una serie red de interconexiones, que permiten procesar la información que ingresa y genera una salida. Las neuronas tienen una o varias entradas y realiza operaciones matemáticas con estas para generar una salida. Después, esta salida puede ser enviada a otras neuronas interconectadas dando como resultado una red de procesamiento de información (ver figura 2.1).

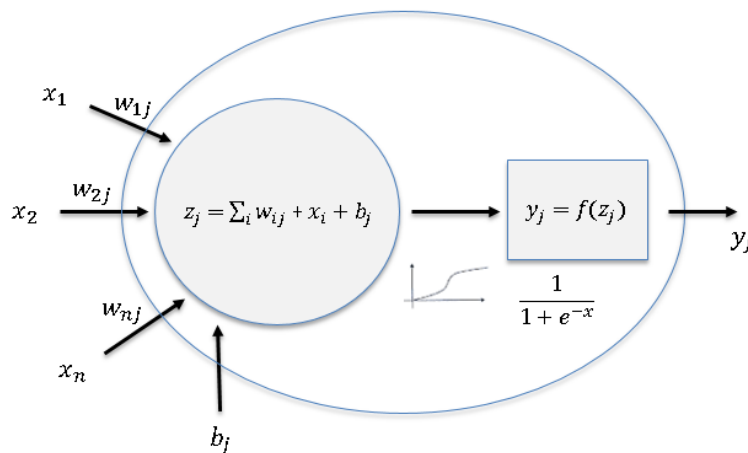


Figura 2.1. Esquema de una Red Neuronal Artificial (Torres I Viñals, J. 2020)

Las ANN tienen la capacidad de aprender por medio de la detección de patrones entre los datos por medio del proceso de entrenamiento, lo cual permite que las conexiones entre las diferentes neuronas de la red se ajusten, y esto consecuentemente minimiza el error en la salida del modelo. Berzal (2019) menciona que el enfoque común de todas las técnicas que están basadas en las redes neuronales artificiales es expresar una solución a un problema complejo, no solo como un algoritmo secuencial, que resuelve un problema aplicando una serie de pasos definidos, sino como el resultado de combinar en paralelo información de un gran número de elementos de procesamiento (neuronas) que se encuentran interconectados entre sí. Por el diseño de la arquitectura, las redes neuronales artificiales tienen características como la capacidad de aprender a partir de

la experiencia, estas pueden abstraer características esenciales a partir de datos de entrada reconociendo patrones.

Las siguientes son algunas de las características de las redes neuronales artificiales:

- *El aprendizaje automático:* las ANN tienen la capacidad de aprender a partir de datos de entrada, ajustando los parámetros con el objetivo de mejorar la precisión en la tarea designada.
- *El aprendizaje Adaptativo:* este aspecto se refiere al proceso mediante el cual las ANN se entrenan y, en consecuencia, aprenden.
- *La autoorganización:* una ANN estructura su organización basada en el entrenamiento y la experiencia derivada de cada dato que es procesado. Así, diferentes secciones de la red se especializan en reconocer determinados patrones específicos.
- *La operación en tiempo real:* una vez entrenadas, las ANN tienen la capacidad de reconocer y clasificar datos de manera instantánea.
- *La integración tecnológica:* por la característica matricial de las operaciones requeridas para el entrenamiento de las ANN, es factible la utilización de componentes, como las GPUs, que están diseñadas para acelerar los procesos al realizar operaciones matriciales paralelas (procesamiento en paralelo).
- *La adaptabilidad:* las ANN pueden ajustarse a nuevos conjuntos de datos de entrada, modificando su comportamiento de acuerdo con la información proporcionada.
- *El paralelismo:* las ANN tienen la virtud de ser intrínsecamente paralelas, lo que les permite procesar extensos volúmenes de datos simultáneamente.
- *La generalización:* esta característica permite a las RNA aplicar el conocimiento adquirido de un conjunto específico de datos (*dataset*) de entrenamiento a nuevos conjuntos de datos nuevos previamente no conocidos por el modelo.
- *La no linealidad:* las ANN tienen la habilidad de modelar relaciones no lineales entre los datos de entrada y de salida.
- *La escalabilidad:* estas redes pueden adaptarse para gestionar conjuntos de datos más amplios y tareas de mayor complejidad.
- *La optimización:* utilizan técnicas avanzadas de optimización para calibrar los parámetros y, de este modo, mejorar su rendimiento en la tarea asignada.

Arquitectura de una red neuronal

Una red neuronal artificial (*Neural Network Architecture, ANN*), tiene una estructura y diseño definido, conformada por múltiples neuronas que están interconectadas entre sí entre varias capas de la red, donde cada capa la conforman un determinado número de neuronas.

Una red neuronal está conformada por tres tipos de capas, las capas de entrada, las capas ocultas y las capas de salida. Las capas de entrada son las primeras capas de la red, y están encargadas de recibir los datos de entrada.

En la figura 2.2. se presentan diferentes arquitecturas de las redes monocapa las cuales tienen una capa de entrada, una capa oculta de procesamiento y la capa de salida con una o más neuronas. La figura 2.2.a, y la figura 2.2.d, muestran dos redes que tienen la capacidad de clasificar patrones de entrada más complejos o realizar predicciones. La cantidad de neuronas de la capa oculta depende de la capacidad de procesamiento y clasificación. De igual forma el número de neuronas de la capa de salida dependerá del problema que se esté abordando.

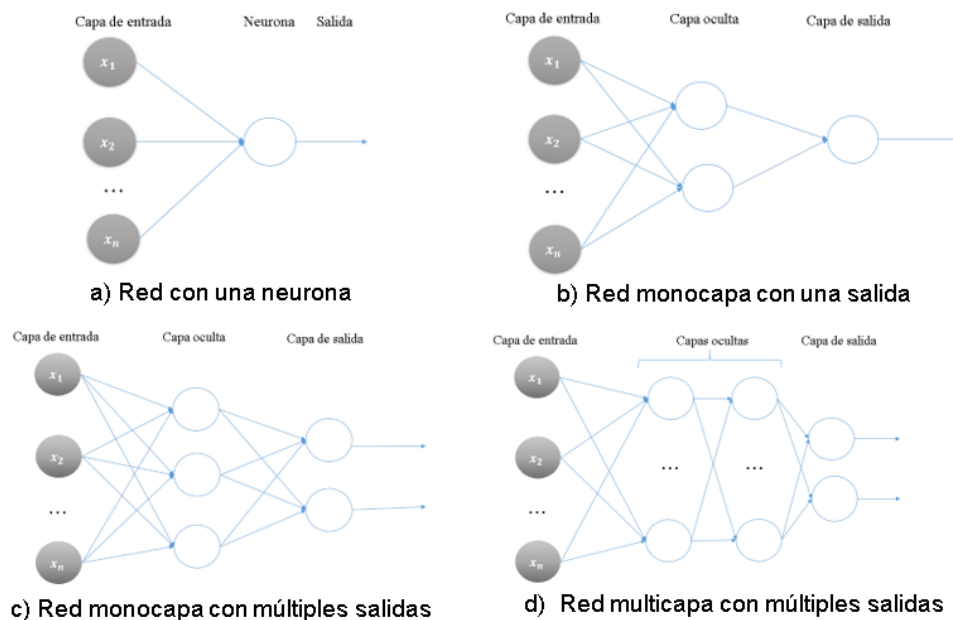


Figura No 2.2. Arquitecturas de una red neuronal

2.2. La sombra (*The Shadow*)

Las sombras se pueden definir como las partes de la escena que no están directamente iluminadas por una fuente de iluminación, debido a un objeto u objetos que la obstruyen. En la detección de sombras se hace uso de una definición similar e

igualmente se plantea una sombra típica que podría dividirse en dos tipos diferentes. Un tipo se denota como sombra propia donde la región de la sombra está en el objeto mismo; y el otro tipo es la sombra proyectada para la cual la región de la sombra está en el fondo o en otros objetos. Aquí la sombra proyectada suele dividirse en dos partes, umbra y penumbra como se puede observar en la figura 2.3. Dare (2005) definió la umbra como la representación de la región de sombra donde la fuente de luz primaria está completamente oscurecida; mientras que la penumbra es la región alrededor del borde de una sombra donde la fuente de luz solo está parcialmente oscurecida. Jiang y Ward (1988) en su investigación mencionan que una fuente de luz puntual generará solo umbra, mientras que una fuente de luz de área generará también penumbra y argumentan que es posible que no aparezca una penumbra delgada en la imagen debido a los efectos de digitalización (J. Yao, Z. Zhang, 2009).

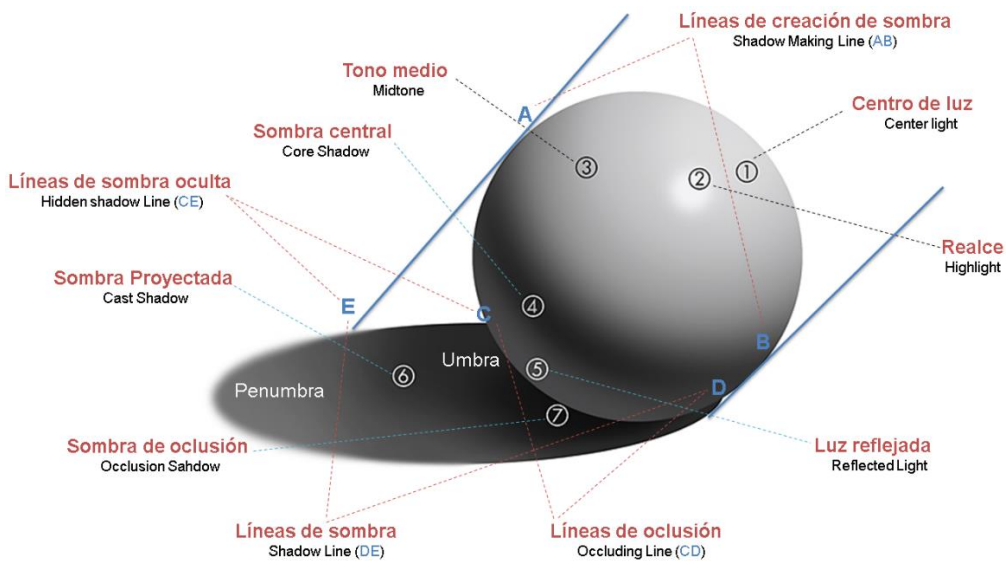


Figura 2.3. Luz y sombras de una esfera

En las sombras, los límites se pueden clasificar en cuatro clases: líneas de creación de sombras, líneas de sombras, líneas de oclusión y líneas de sombras ocultas. Las líneas de creación de sombras, *AB* (*Shadow making line*), separan la superficie iluminada y la superficie no iluminada de un objeto. Son las proyecciones de las líneas de creación de sombras en la dirección de los rayos de luz se denominan líneas de sombra, *DE* (*Shadow Line*). Las líneas de oclusión, *CD* (*Occluding line*), separan un objeto de su sombra proyectada. Una línea de sombra oculta, *CE* (*Hidden shadow line*), es una línea de sombra que corresponde a una línea de creación de sombras no visible.

De igual forma, para Sato *et al.* (2003), las sombras en una escena son causadas por la oclusión de la luz entrante y, por lo tanto, contienen varios elementos de información sobre la iluminación de una escena a partir del brillo de la imagen. De tal manera que el brillo de la imagen en el interior de las sombras se utiliza para proporcionar pistas claras que permitan estimar una distribución de iluminación. Más concretamente, el método

recupera la distribución de la iluminación de la escena a partir de del brillo de la imagen dentro de las sombras proyectadas por un objeto de forma conocida en la escena.

Para Salvador, Cavallaro y Ebrahimi (2004), las sombras en una imagen 2D tienen información relevante sobre la escena, permiten obtener la localización de la forma u objeto, las características de la superficie y la fuente de luz. Mencionan técnicas que se basan en modelos que representan el conocimiento de la geometría de la escena, los objetos o formas, la fuente de luz, y técnicas basadas en propiedades que identifican las sombras utilizando características como la geometría, el brillo y el color de las sombras. Esto proporciona varios elementos importantes donde la sombra es una de las partes esenciales de la investigación, ya que permite inferir formas de objetos, y cuando está presente en una imagen proporciona información que puede ser utilizada para determinar las formas y orientaciones de los objetos.

Tipos de Sombras

En una escena la percepción visual y la interpretación están ligadas a la interacción de la luz y las sombras. Estas no solo definen la tridimensionalidad y la textura de los objetos, sino que también definen las características propias de las formas.

A continuación, se describen distintas categorías de sombras y su importancia en varios ámbitos. (Martínez, A.D. *et al.*, 2021), (Shilpa, M., Gopalakrishna, M.T. and Naveena, C. 2020):

- *Las sombras propias:* son las que se generan cuando un objeto obstruye la luz, arrojando una sombra sobre sí mismo o sobre una superficie inmediata. Son más definidas cuando la fuente luminosa se halla en proximidad al objeto.
- *Las sombras proyectadas:* corresponden a las sombras que un objeto proyecta sobre superficies más alejadas. Su morfología y extensión están influenciadas por la disposición geométrica de la escena y la posición de la fuente luminosa (Datta, S. *et al.*, 2022).
- *Las sombras ambientales:* estas sombras se caracterizan por ser más tenues y desdibujadas, originadas por la iluminación indirecta y la dispersión lumínica en la escena. Contribuyen a dotar de realismo y atmósfera a una imagen.
- *La influencia de la iluminación:* la orientación de la fuente luminosa respecto a los objetos y la cámara determina la apariencia y dirección de las sombras. Una iluminación frontal alarga las sombras detrás de los objetos, mientras que una lateral resalta las texturas y relieves (Luo, S., Li, H. and Shen 2020).
- *El suavizado de sombras:* en contextos reales, las sombras presentan bordes difuminados y zonas de transición con variaciones lumínicas. Se recurre a

algoritmos especializados para replicar estos matices en la renderización digital, potenciando la autenticidad visual (Kim, D.S. *et al.*, 2018).

- *La eliminación de sombras:* en ámbitos como la teledetección o la visión artificial, se precisan técnicas de supresión de sombras para optimizar la identificación de objetos y el análisis escénico, mediante segmentación y ajuste de píxeles (Inoue, N. and Yamasaki 2021).
- *La estimación de profundidad:* la correlación entre objetos, las sombras y la fuente luminosa puede emplearse para inferir profundidades y ubicaciones relativas en una escena, esencial en la visión por computadora y la robótica (Asaf Karnieli, Ohad Fried, Yacov Hel-Or 2022).
- *La renderización y gráficos 3D:* en la creación de gráficos tridimensionales, la interacción de la luz con los objetos es fundamental para generar imágenes convincentes, en donde las sombras realzan la autenticidad visual (Chen, Z., Wu, B. and Liu 2021).
- *El aprendizaje profundo y la segmentación:* las sombras pueden influir en la segmentación y reconocimiento en imágenes. Las estrategias de aprendizaje profundo pueden capitalizar la información sombría para incrementar la precisión en tareas de análisis visual (Tiwary, K., Klinghoffer, T. and Raskar 2022).

El estudio y manipulación de las sombras en imágenes digitales representa un ámbito esencial en la comprensión visual. Las sombras en imágenes digitales son un aspecto fascinante y fundamental, que afecta la percepción visual y la calidad de las imágenes. Por tanto, comprender cómo se forman, cómo interactúan con la luz y cómo se pueden manipular ofrece posibilidades en diversos campos.

2.3. Métodos y Algoritmos para el procesamiento de sombras

En las imágenes digitales las sombras permiten la interpretación y percepción de una escena en particular, dado que contienen una fuente adicional de información en aspectos como la iluminación, la forma, el tamaño, la ubicación, y la orientación de objetos. Entonces el adecuado tratamiento, procesamiento y análisis de esta información contribuye a mejorar la percepción de profundidad y tridimensionalidad en una imagen. Para estas tareas existen algoritmos y métodos que tienen la capacidad de utilizar dicha información con un objetivo en específico. Para Zhang, J. and Liu, J. (2022) las sombras pueden conservar información importante sobre la dinámica de escenas y objetos. Por otro lado, mencionan que las sombras son una fuente de incertidumbre, las cuales pueden generar complicaciones en la interpretación y en el análisis de imágenes, afectando la precisión en el reconocimiento de objetos. De tal forma que proponen una

de aprendizaje no supervisado y la combinación de un módulo de atención de bloque convolutivo (*convolutional block attention module and Unsupervised learning network*, CBAM) para adaptarse a diferentes conjuntos de datos, mejorando la capacidad de generalización del modelo sin necesidad de etiquetado.

Esto plantea sus propios retos, como la necesidad de un modelo preciso que estime la geometría del objeto y la forma en que la luz incide sobre él. Así, la altura de un objeto puede proporcionar información relevante sobre su forma, tamaño y posición. Sin embargo, estimar la altura de un objeto o forma en una imagen *2D* a partir de su sombra es un problema de más complejidad, debido a la falta de información tridimensional del mismo. Es así como, el uso de las de redes neuronales es una herramienta adecuada para abordar este tipo de problema. En este contexto, se presenta una solución basada en redes neuronales que permite estimar o calcular la altura de una imagen *2D* a partir de la sombra que proyecta. La arquitectura seleccionada es de una red neuronal convolucional (*CNN*) con regresión, mediante la cual se pueden extraer características relevantes de la imagen y posteriormente, después de desarrollar el modelo, es posible predecir el valor de la altura correspondiente a una figura determinada.

El enfoque aquí propuesto se evaluó en un conjunto de datos de prueba y se obtuvo la precisión respectiva, en la estimación de las alturas de los objetos en comparación con otros métodos existentes. Con las redes neuronales se pueden procesar y analizar grandes cantidades de datos e información, lo que permite en este caso en particular, una mayor precisión en la estimación de la altura de una forma a partir de su sombra; además de posibilidad de integrar nuevas formas geométricas y sombras, dentro del conjunto de imágenes del *dataset*, que son utilizadas en el modelo de aprendizaje, hace que esta propuesta sea adaptable y escalable para futuras investigaciones en las que se implemente con formas irregulares. Es decir, que integrar la información que proporcionan las sombras proyectadas, en un modelo de aprendizaje puede ser un enfoque prometedor, al resolver el problema de estimar la altura de una forma a partir de su sombra en una imagen *2D*.

Del mismo modo, S. Mohajerani, P. Saeedi (2018) abordaron su investigación sobre la detección automática en una imagen de sombras mediante un método de segmentación y el uso de *Deep Learning*, donde lo que hacen es identificar las regiones de sombras a nivel de píxel en una imagen *RGB*, con lo que consiguen extraer las características de las sombras, dotando a una red neuronal convolucional (*CNN*) de conocimiento para detectar patrones de las sombras local y globalmente, e información previa de la fuente de iluminación y la dinámica de los objetos en la imagen. Por otro lado, Tao, M. W., Srinivasan, P. P., Hadap, *et al.* (2017) proponen un algoritmo basado en la estimación de profundidad densa, que combina métricas de desenfoque y coherencia. Además, definen un marco de optimización que integra la foto coherencia, la coherencia de profundidad y la coherencia de sombreado, para la estimación de profundidad de campo de luz en diferentes escenarios. Este algoritmo incorpora las señales de desenfoque, correspondencia y sombreado superando a otros algoritmos más avanzados. Por otra parte, Vicente y Samaras (2014) indican que en las imágenes digitales *2D* se pueden encontrar diferentes escenas que contienen elementos como

texturas, bordes, formas, colores, sombras, etc., y sombras, de las cuales se puede extraer información relevante y utilizarla en investigaciones mediante la implementación de algoritmos de detección de sombras. Esta información se utiliza para establecer la relación entre la geometría del objeto, la fuente de luz y la zona de sombra.

Asimismo, Kriegman, Belhumeur (1998) y D. C. Knill, *et al.* (1997) se refieren a que las sombras son una fuente de información relevante a nivel de formas de superficies u objetos, permitiendo localizar áreas de interés en una imagen, dirección de la fuente de iluminación, geometría de la forma, entre otras características. Del mismo modo, las investigaciones de Knill, Mamassian y Kersten (1997) y S. A. Shafer, T. Kanade. (1983), indican que estas propiedades geométricas de las sombras son de especial interés, porque permiten establecer relaciones perceptibles entre formas, sombras e iluminación, según la estructura y altura de la superficie. Por otra parte, Hintze y Morse (2019), en su investigación sostienen que la información que se extrae de la sombra, mediante técnicas, métodos y algoritmos se enfoca en determinar la localización de la forma u objeto, el área de interés, la dirección de la fuente de iluminación entre otros, suele estar orientada a su posterior eliminación.

De igual forma, Salvador, Cavallaro y Ebrahimi (2004), abordan aspectos generales relacionados con el procesamiento digital de imágenes, empleando redes neuronales convolucionales (*CNN's*) con diferentes conjuntos de datos (*datasets*), donde estos últimos tienen como característica principal, imágenes de primitivas geométricas y las sombras que proyectan de acuerdo con una fuente de iluminación en un lugar específico. También Panagopoulos, Hadap y Samaras (2013) emplean en su investigación imágenes sintéticas divididas en pequeñas regiones rectangulares, y, por lo tanto, para cada una de estas es posible capturar lo correspondiente a la distribución de intensidades y la capacidad de manejar superficies. En este punto se combinan las secciones de geometría, la distribución de intensidades, integradas en un diccionario con el cual todas las secciones se pueda generar una hipótesis sobre esa forma.

En la actualidad y como nunca antes, tanto la utilización como el desarrollo de algoritmos se ha incrementado de forma exponencial, siendo estos algoritmos cada vez más eficientes y más precisos. Por lo que en este documento se mencionan los algoritmos más importantes utilizados en el análisis y el procesamiento de las sombras. Existen diferentes métodos y algoritmos para el procesamiento de sombras, algunos de ellos los más comunes se encuentran:

- El *Thresholding* es un método que consiste en convertir una imagen en escala de grises en una imagen binaria, utilizando un valor de umbral. Este método se utiliza para segmentar objetos o regiones en áreas de interés en una imagen.
- La segmentación es el proceso que se realiza para fragmentar una imagen en partes o regiones con el propósito de detectar o identificar objetos que para facilitar su análisis. En el caso de este trabajo, la segmentación se usa para separar la sombra del objeto y extraer características relevantes (Al-Najdawi,

N. *et al.* 2012).

- El aprendizaje profundo que hace referencia a las diferentes arquitecturas de redes neuronales que se han utilizado para múltiples propósitos, así como también para el análisis como para el procesamiento de sombras.

El procesamiento de sombras puede utilizar transformaciones geométricas para alinear sombras y objetos en una imagen. Estos son sólo algunos de los métodos y algoritmos utilizados en el procesamiento de sombras donde la elección del método apropiado depende del problema específico que se esté abordando. Ya en el procesamiento digital de imágenes, los algoritmos que trabajan con las sombras realizan tareas desde la identificación hasta análisis de las mismas. Para esta labor hay algoritmos diseñados para la detección de sombras y otros centrados en el análisis de estas los cuales serán tratados a continuación.

Los algoritmos de detección de sombras se dedican a extraer información más detallada de las mismas, abarcando aspectos como su forma, tamaño, grado de intensidad, entre otras características inherentes (Al-Najdawi, N. *et al.*, 2012). La salida más común de los algoritmos de detección de sombras es una máscara binaria o una imagen segmentada, en la cual las regiones en sombra están claramente demarcadas, mientras que los algoritmos de análisis de sombras ofrecen un conjunto más detallado de atributos o características, asociados a cada sombra detectada (Shilpa, M., Gopalakrishna, M.T. and Naveena, C. 2020). Estos atributos pueden incluir propiedades geométricas, distribución espacial o patrones de intensidad (Singh, A.K. and Pandit, A. 2019), (Sanin, A., Sanderson, C. and Lovell 2012). También son valiosos en la corrección de imágenes, donde la intención puede ser la eliminación o atenuación de las sombras, (Chen, Z., Wu, B. and Liu, W.C. 2021), (Tiwarly, K., Klinghoffer, T. and Raskar 2022), (Martínez, A.D. *et al.*, 2021).

Algunas investigaciones como la de Sanin, A., Sanderson, C. y Lovell, B.C. (2012), presentan varios métodos y algoritmos para la detección, eliminación de sombras en movimiento basada en características que se compone de cuatro categorías: cromaticidad, física, geometría y texturas. Proponen el uso del rendimiento de seguimiento como enfoque para determinar los métodos y algoritmos de detección de sombras. De los métodos que ellos destacan se encuentran: el método basado en cromaticidad que es el más rápido de implementar y ejecutar, pero sensible al ruido y falla cuando las propiedades espectrales de los objetos cuando estas son similares a las del fondo y el método basado en texturas de grandes y pequeñas regiones donde los resultados son más precisos, aunque su coste computacional es elevado.

Así mismo, S. H. Khan, *et al.* (2014) utilizan múltiples redes neuronales convolucionales profundas (*ConvNets*), donde su arquitectura consiste en alternar capas de convolución y submuestreo. Su investigación se centra en la detección de sombras, teniendo como entrada una base de datos que contiene en varias condiciones de iluminación con ambientes soleados, nublados y oscuros. Para ello utilizaron un modelo

de *Campo Aleatorio Condicional (CRF)*, que se traduce en predicciones para cada píxel de cada imagen de prueba, y que posteriormente se comparan con las máscaras de sombra reales. Esto refuerza la coherencia del etiquetado en todos los nodos de un gráfico cuadrículado definido en la imagen, eliminando los resultados de etiquetado aislados.

2.3.1. Algoritmos para detección de sombras

Las sombras para el procesamiento digital de imágenes representan un insumo con el cual los algoritmos las utilizan con diferentes propósitos, por la información relevante que estas tienen. Las sombras son percibidas en algunos casos sin mucha relevancia, pero en realidad tienen una complejidad única, dado que, a partir de estas, se puede interpretar y analizar imágenes en una escena, las cuales combinadas con redes neuronales pueden aprender patrones, características o detalles más complejos para aumentar la capacidad de generalización de los modelos. En este método, se calculan los gradientes de la imagen y se crean histogramas de gradientes orientados para detectar las regiones de sombra (N. Dalal and B. Triggs, 2005).

En la detección de sombras en imágenes, la transformada *wavelet* se utiliza para descomponer la imagen en diferentes frecuencias utilizando un umbral o que permite detectar regiones de sombra. La transformada *wavelet* descompone una imagen en sub-bandas (frecuencias), en diferentes niveles de detalle, es decir se está descomponiendo una señal en componentes de diferentes frecuencias. En otra investigación se utiliza la segmentación basada en umbrales, que es uno de los métodos más simple para la detección de sombras, donde el umbral se utiliza para detectar las regiones de sombra en la imagen. Este algoritmo de detección de sombras basada en la cromacidad (*chromacity-based shadow detection*). Por otro lado, Huang, X., Gao, J., *et al.* (2007) en su investigación abordan la estimación de la forma a partir del sombreado (*SFS*), cuyo objetivo es resolver un problema con pocas restricciones para estimar el mapa de profundidad, a partir de una única imagen, desarrollando un método basado en ejemplos para mejorar la precisión del (*SFS*), mejorando la calidad de la reconstrucción a partir de imágenes reales de diferentes formas, obteniendo un conocimiento previo de su apariencia para el reconocimiento de formas tridimensionales. De forma similar está la propuesta de E. Prados y O. Faugeras (2006) con el *shape from shading (SFS)* en la cual con una sola imagen estiman la forma *3D* de una superficie, donde en una superficie difusa no se conoce la posición de la fuente de iluminación y donde no se conoce el mapa de reflectancia.

De otra parte para M. Daum, G. Dudek (1998) la reconstrucción de superficies *3D* a partir de las sombras que se proyectan, se realiza para obtener información sobre la estructura de la escena (estructura tridimensional), basándose en la recolección de sombras desde un punto fijo y haciendo rotar la fuente bajo diferentes condiciones de iluminación, lo que da como resultado, un algoritmo de reconstrucción de escenas *3D* que toma detalles sobre la forma, la ubicación espacial de los objetos a partir de las

sombras y de la fuente de iluminación.

S. Savarese, *et al.* (2007), S. Savarese, *et al.* (2002) y R. Gouiaa, J. Meunier (2014) proponen un nuevo método para recuperar una forma a partir de sombras (SFS). Estas sombras proyectadas son información relevante sobre la forma de los objetos, descubriendo cavidades que no están disponibles a partir de pistas como los límites ocluidos. Este método se denomina límites ocluyentes (*occluding boundaries*). En función del volumen ocupado por un objeto es posible identificar y esculpir las regiones del volumen, que presentan inconsistencias con el patrón observado en las sombras. En resumen, su propuesta es un sistema de reconstrucción para recuperar la forma a partir de siluetas y esculpido de sombras donde las concavidades y las siluetas se utilizan para reconstruir la estimación inicial conservadora de la forma del objeto. Por otro lado D. Forsyth y, A. Zisserman (1991) muestran cómo la iluminación mutua y la irradiancia de la imagen dentro de una imagen dan lugar a estructuras más complejas (Al-Najdawi, N. *et al.*, 2012). Igualmente, R. Liu, *et al.* (2022) en su investigación abordan la reconstrucción 3D cuando la estructura del objeto está parcial o totalmente ocluida. Introducen en el método sombras proyectadas de un objeto no observado, para realizar la inferencia del posible volumen 3D al que corresponde. El modelo de imagen "heredable" permite inferir conjuntamente la forma 3D del objeto, su ubicación y la de la fuente de iluminación. La ventaja de este enfoque es la estimación de múltiples escenas 3D a partir de sombras. Estas investigaciones se utilizaron como referencia al plantear la solución al modelo propuesto en este trabajo, tales como selección del tipo de red neuronal, fuente de iluminación, análisis de la luz y de las sombras, procesamiento computacional, etc., determinando los requerimientos iniciales, tales como el *Dataset*, el *Dataframe* al modelar la solución.

2.3.2. Algoritmos para Análisis de sombras

El algoritmo de análisis de sombras está basado en la segmentación de objetos para separar las áreas de la sombra e iluminación, que analizan la forma y el tamaño de las sombras para detectar objetos y su posición espacial. Por otro parte está el algoritmo de separación de sombras de *chroma key*, el cual se utiliza para separar la sombra del objeto, el cual está basado en la eliminación de los componentes de color y la manipulación de los canales *RGB*.

En Hosseinzadeh, S., *et al.* (2005) se muestra que el concepto de secciones corresponde a un *collage*, de secciones con diferente reflectancia y que están iluminadas con una fuente de iluminación, que varía lentamente en intensidad, y con formas que tienen un factor de escala constante. Este modelo de aprendizaje con el conjunto de datos a generar se utiliza para determinar la altura de la superficie, y la generación 3D del objeto o forma, tomando inicialmente la información de la sombra que proyecta sobre una imagen 2D. Salvador, Cavallaro, Ebrahimi y Touradj (2004), G. Liasis y S. Stavrou. (2016) indican que la iluminación, las sombras y la reflectancia son parte integral de las diferentes escenas, proporcionando información relevante sobre la forma y apariencia

del objeto.

Panagopoulos, Hadap, y Samaras (2013) proponen el uso de un diccionario (*Dictionary*) que está conformado por diferentes parches (*Patches*) de geometría asociados a la imagen, a una distribución de intensidades de píxeles que pueden generar esa geometría, y con el uso de regiones de imagen más grandes que permiten capturar datos de apariencia. Este indica un punto de partida donde se evalúa si es posible con un conjunto de datos y apoyado en redes neuronales, diseñar un modelo que se ajuste a la investigación propuesta. De otra parte, de Varol, Shaji, Salzmann y Fua (2011) se tiene un enfoque basado en la recuperación de formas deformables, que puede trabajar bajo iluminación compleja y en superficies parcialmente texturizadas. Para esto utilizaron un algoritmo que realiza un mapeo aprendido de la distribución de la intensidad y la forma de los parches locales de la superficie, centrándose en la estimación en la forma de la superficie 3D.

D. S. Kim, M. Arsalan y K. R. Park (2018) emplean una red neuronal convolucional (CNN) para la detección de sombras mediante cámaras, utilizando la base de datos abierta CAVIAR (*Context-Aware Vision Using Image-Based Active Recognition*). Indican que en su investigación el modelo que plantean no es computacionalmente costoso, tanto en el entrenamiento como en la validación, y este se puede adaptar a tareas como la segmentación de imágenes.

De forma similar, en Atkinson, *et al.* (2007), se presenta un método novedoso para la reconstrucción de superficies 3D con información de polarización y sombreado a partir de dos vistas. Utilizan la teoría de *Fresnel* en el procesamiento de imágenes para obtener estimaciones de las normales de la superficie. Aquí enfatizan cómo los brillos de los píxeles medidos dependen de la orientación de la superficie combinada con las estimaciones refinadas, para determinar la correspondencia entre dos vistas de un objeto. R. Liu, *et al.* (2022) abordan la reconstrucción 3D cuando la estructura del objeto está parcial o totalmente ocluida e introducen en el método sombras proyectadas de un objeto no observado, para realizar la inferencia del posible volumen 3D al que corresponde. El modelo de imagen "heredable" permite inferir conjuntamente la forma 3D del objeto, su localización y la de la fuente de iluminación.

Asimismo, D. Samaras y D. Metaxas (2003) presentan un método para la integración de restricciones *holonómicas* no lineales en modelos deformables y su aplicación a los problemas de estimación de forma y dirección de la fuente de iluminación a partir de sombreros. Su propuesta indica que funciona cuando no se conoce la dirección de la fuente de iluminación. Así que, acoplaron el método de estimación de la forma con un método de estimación de la luz, en el que una mejor estimación de la forma resulta en una mejor estimación de la luz y viceversa.

Con la investigación de Yuniarti, A. and Suciati, N. (2019), se estableció un concepto más claro sobre la capacidad de inferencia de los modelos de redes neuronales, que en su investigación lo plantea con un algoritmo diseñado para inferir formas 3D, con la utilización de sombras y el concepto de redes de ocupación (*occupancy network*). Este

tipo de modelo desarrollado orientado a reconstruir un objeto, el cual se encuentra parcial o totalmente ocluido muestra el potencial de las arquitecturas de redes neuronales generativas. La investigación de Yuniarti, A. and Suciati, N. (2018) se centra en el modelo diferenciable de formación de imágenes, que permite inferir conjuntamente la forma 3D de un objeto y la posición de una fuente de iluminación. Es así que, en su investigación se muestra como los modelos generativos que se entrenan con datos 3D a gran escala, son capaces de aprender e inferir a partir de datos empíricos la estructura de los objetos. El modelo es compatible con muchas opciones de representación 3D, modelando los volúmenes 3D con un campo de ocupación. En Mescheder, L. et al. (2019) se amplía y se utiliza el concepto de red de ocupación (*occupancy network*) que es definida como la red neuronal la cual estima la probabilidad de un volumen asociado (masa), por medio del modelo generativo que se entrena para producir los parámetros de la red de ocupación el cual es utilizado en Yuniarti, A. y Suciati, N. (2019). Lo anterior se utilizó como punto de partida y un hallazgo importante para plantear la utilización de las redes neuronales como la mejor opción en la solución al problema en cuestión del presente trabajo.

2.4. Tipos de Redes Neuronales

Las redes neuronales artificiales se caracterizan por tener diferentes estructuras y modelos, además de considerar es nivel de complejidad, porque muchas de las arquitecturas de las redes neuronales profundas, son especiales para problemas más complejos o con grandes volúmenes de información, ya que pueden aprender características mucho más abstractas y representativas (Sarker, I.H., 2021).

Algunos tipos de redes neuronales:

- a. Las redes neuronales multicapa (*Multilayer Neural Networks, MLP*) son las redes neuronales que está conformadas por una serie de capas en su estructura: una capa de entrada, una o más capas ocultas y una capa de salida. (Vemuri, V.K., 2020).
- b. Las redes neuronales recurrentes (*Recurrent Neuronal Networks, RNN*) son redes que por su tipología carecen de memoria, donde las conexiones de una red recurrente pueden ser dirigidas, aunque también pueden ser bidireccionales, que en este caso se obtienen una red que tiene la capacidad de completar patrones con ruido o incompletos (John J. Hopfield, 1982).
- c. Las redes neuronales convolucionales (*Convolutional Neural Networks, CNN*) se utilizan principalmente para el procesamiento de imágenes, también en la detección de un objeto en una escena, donde la redes convolutivas aprovechan su estructura, para diseñar arquitecturas especializadas, para ampliar información

(Kunihiko Fukushima, 1975), (Kunihiko Fukushima, 1980), (Bye, S.J. & Adams, 1993).

- d. Las redes neuronales adversarias generativas (*Generative Adversarial Neural Networks, GAN's*): diseñadas específicamente para la generación de datos. Las *GAN's* consisten en dos redes, una red generadora (el generador) y una red discriminadora (discriminador) y que generan datos sintéticos de alta calidad.

2.5. Redes Neuronales Convolucionales (*Convolutional Neural Networks*)

2.5.1. ¿Qué es una Red Neuronal Convolutiva? (*CNN*)

Las redes neuronales convolucionales o *convolutional Neural Networks* o *ConvNets*, que están diseñadas para el procesamiento de datos en este caso de imágenes, que tienen la capacidad de aprender patrones complejos. Entre las características de una red neuronal convolutiva se encuentra la operación de convolución y en el modelo es una capa convolutiva la cual aprende patrones locales dentro de la imagen en pequeñas ventanas de dos dimensiones. Por tanto, para una capa convolutiva su objetivo es detectar características o rasgos visuales en las imágenes (líneas, bordes, sombras, color, textura, etc.). Entonces una vez aprendida la característica esta se puede reconocer después en cualquier parte de la imagen, además estas capas convolucionales pueden aprender jerarquías espaciales de patrones, de tal forma que es así que la capa convolutiva puede aprender a reconocer bordes y la segunda capa convolutiva puede aprender patrones compuestos de elementos básicos aprendidos (reconocer sombras) en la capa anterior y así sucesivamente hasta aprender patrones más complejos y abstractos (ver figura 2.4) LeCun, Y., Bottou, L., Bengio, *et al.*, (1998).

En las redes neuronales las capas convolucionales funcionan por medio de tensores 3D, llamados mapas de características (*features maps*), los cuales tienen ejes espaciales con la altura (*height*) y el ancho (*width*). También cuentan con los canales (*channels*) que son llamados profundidad (*depth*), que para una imagen de color *RGB* son (*red, blue, green*) con una dimensión de profundidad de 3 canales el rojo, azul y verde, que a diferencia de una imagen a blanco y negro la dimensión del eje de profundidad (*depth*) sería 1.

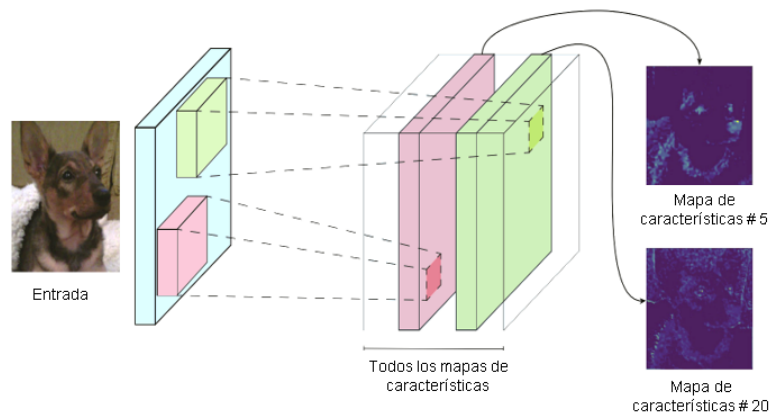


Figura 2.4. Mapa de características: cada filtro crea un mapa de características diferente que detecta distintos patrones en la imagen *Goldberg, M. and Gougeon (1980)*

La diferencia entre las redes neuronales convencionales y las convolucionales se encuentra en la capacidad de extraer características, donde estas redes incluyen conceptos como submuestreo (*pooling*) e hiperparámetros: desplazamiento o paso (*stride*), relleno (*padding*), dimensión y cantidad de núcleos o filtros de profundidad (*kernels*). En las redes neuronales convolucionales su estructura de capas está organizada en tres dimensiones: ancho, largo y profundidad. Poseen tres tipos de capas: capa convolucional, capa de agrupamiento y capa completamente conectada (*FC: Fully Connected*).

En las redes neuronales convolucionales también está la agrupación (*pooling*) la cual reduce la dimensionalidad del mapa de características al condensar la salida de pequeñas regiones de neuronas en una sola salida, lo que simplifica las siguientes capas y reduce la cantidad de parámetros que el modelo necesita aprender (López Pinaya, W.H. *et al.*, 2020). De igual forma es importante tener en cuenta que esta operación reduce la dimensionalidad espacial (ancho y alto, no de profundidad) de la entrada con el propósito de disminuir la cantidad de parámetros y cómputo en el modelo (Sill, J. *et al.*, 2009). Es así como para esta investigación se pueden considerar tres formas de utilizar una *CNN* para el análisis imágenes que consisten en lo siguiente: el primero es entrenarla desde cero, donde este método es altamente preciso, aunque también es el más desafiante si necesitan cientos de miles de imágenes etiquetadas y su concepto de basa en la idea de que puede utilizar el aprendizaje de un tipo de problema para resolver uno similar. El segundo utilizar un modelo (transferencia de aprendizaje), utilizar un modelo pre-entrenado, donde este requiere menos datos y recursos computacionales que el anterior. Y el tercero puede utilizar un modelo pre-entrenado para extraer funciones para entrenar a un modelo de aprendizaje automático.

Estructura de una Red Neuronal Convolucional (CNN)

La estructura de una *CNN* se compone de las siguientes capas una de ellas es la capa de entrada (*Input Layer*) que recibe la entrada que este caso es una imagen (ver

figura 2.5) La entrada se procesa como un *tensor* tridimensional de tamaño que tiene ancho, altura, y canales (López Pinaya, W.H. *et al.*, 2020).

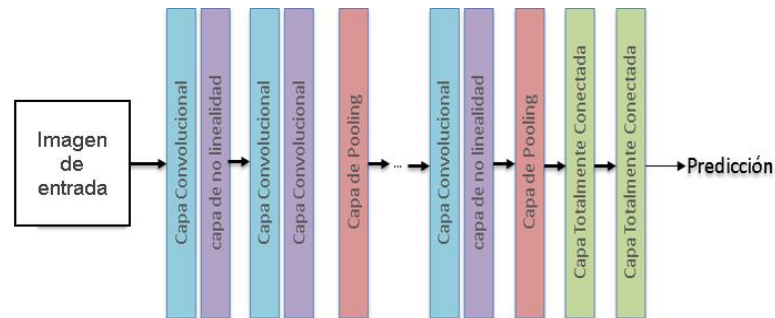


Figura 2.5. Estructura de una red neuronal convolucional. (Adaptado de López Pinaya, W.H. *et al.*, (2020)).

Estos filtros se aplican sobre la imagen de entrada recorriendo en incrementos llamados *stride*, que generan los mapas de características (*features maps*) que representan las partes importantes de la imagen. La capa de agrupación (*Pooling Layer*) es la encargada de reducir la resolución espacial de la entrada, disminuyendo la cantidad de parámetros de la red condensando la información y dando robustez a la red frente a pequeñas variaciones y traslaciones en la imagen. Las operaciones de *pooling* más comunes (*MaxPooling* y *Average pooling*). La siguiente es la capa completamente conectada (*Fully Connected Layer*) que toma los mapas de características y los convierte en un vector lineal unidimensional. Por último, está la capa de salida (*Output Layer*) que produce la salida final de la red dependiendo del problema específico que se esté abordando (clasificación, regresión, etc.). La salida para el modelo desarrollado en este documento la salida será la de una predicción de un valor escalar de la altura de una figura, la cual será dada en centímetros (cm).

En la investigación propuesta el modelo predice un valor de altura de la figura, ya que es una red neuronal convolucional con regresión. Es así como, el número de neuronas de esta capa depende del problema que se está resolviendo, que en esta propuesta es la predicción de un valor de altura. Por otra parte, ya para problemas de clasificación binaria, se utiliza una única neurona de salida con una función de activación sigmoidea. Para problemas de clasificación multiclase, el número de neuronas de salida es igual al número de clases, y se utiliza una función de activación *softmax* (Sharma, N. *et al.*, 2018).

Arquitectura de una red neuronal convolucional (CNN)

Las Redes Neuronales Convolucionales (*CNN*) están diseñadas especialmente para el procesamiento de datos con estructura espacial, es decir imágenes y datos. En la figura 2.6 se observa que la *CNN* es una combinación de una serie de capas de convolución, activación y agrupación que le permiten a la red aprender patrones y características útiles.

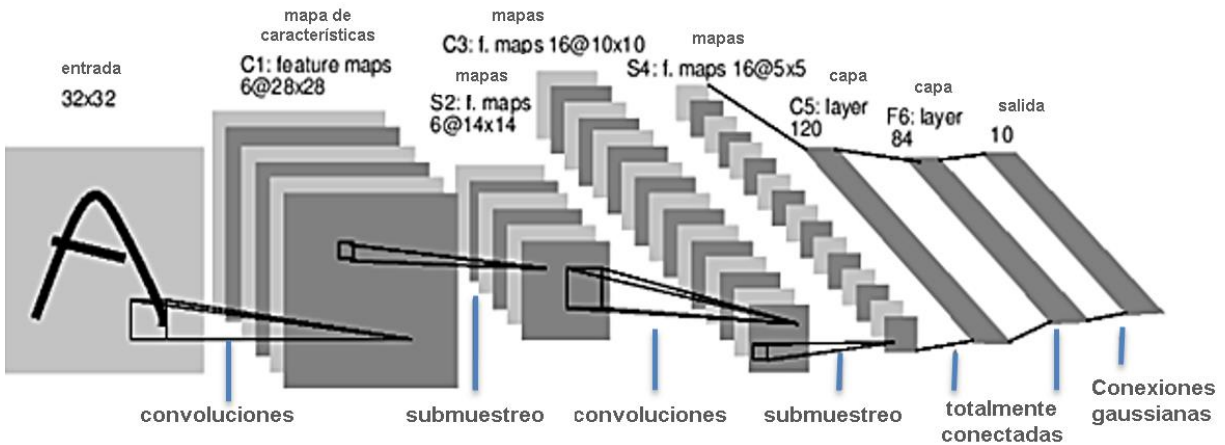


Figura 2.6. Arquitectura de una CNN (Adaptado de Loy J., (2019)).

En las redes CNN se encuentran los núcleos (*kernels*) que son instrumentos ideales para la visión por computadora debido a su habilidad para procesar estructuras de datos con coherencia espacial. Es así como en esta arquitectura el núcleo, también conocido como filtro, es una pequeña matriz utilizada para deslizarse a través de la entrada de datos (por ejemplo, una imagen) para realizar una operación de convolución. Esta operación matemática resulta en la extracción de características de la entrada, tales como bordes, texturas o patrones más complejos. Al aplicar el *kernel* sobre la imagen, se produce un mapa de características que resalta las propiedades de la imagen que el *kernel* está diseñado para detectar. Por ejemplo, un *kernel* puede estar configurado para resaltar bordes verticales, mientras que otro podría detectar esquinas. A medida que la red se entrena, estos *kernels* se ajustan automáticamente para capturar las características más relevantes para la tarea en la que se esté trabajando.

Yadav, R., Nascetti, A. y Ban (2023) en su investigación proponen un enfoque supervisado con un conjunto de datos que incluye en las entradas, tanto imágenes y valores de alturas, lo cual reafirma la posibilidad de que utilizar las redes neuronales para el desarrollo de esta propuesta de investigación fue viable, porque se puede construir por una parte una base de datos con las alturas y las características de diferentes formas o figuras, que finalmente se le pasaron al modelo en el proceso de entrenamiento. De tal forma que Yadav, R., Nascetti, A. and Ban (2023) con su investigación propusieron una red de regresión multimodal de altura de Edificios (*MBHR-Net*) supervisada, y que se utilizaron para estimar sus alturas con una resolución espacial de 10 m utilizando series temporales de satélites *Sentinel-1* (S1) y *Sentinel-2* (S2) (ver figura 2.7).

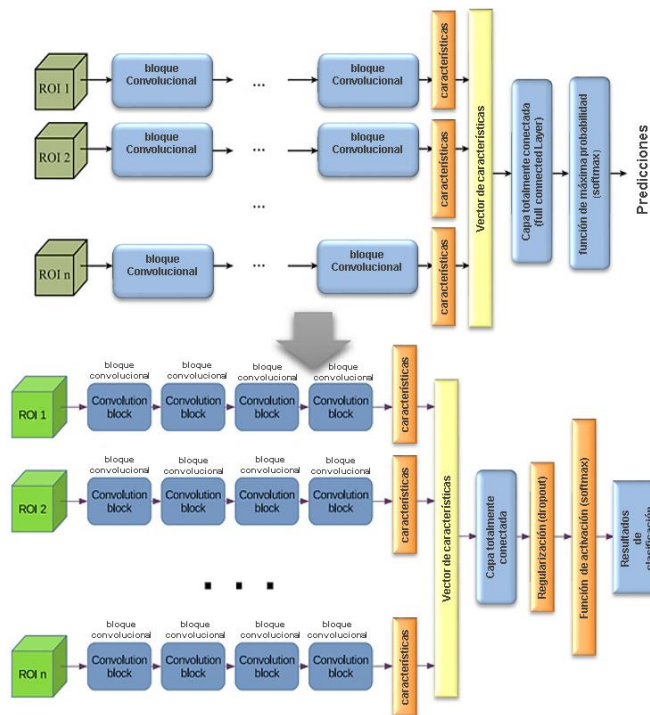


Figura 2.7. La arquitectura multimodal propuesta por Khvostikov *et al.*, (2018).

Es una arquitectura basada en bloques que está diseñada para ser multimodal. Cada bloque convolutivo está seguido de una capa de agrupación. Los bloques extraen características de los datos de entrada, mientras que las capas de agrupación se encargan de reducir la dimensionalidad de los mismos.

Cuando un filtro se aplica a una imagen a través de la operación de convolución, produce un mapa de características (ver figura 2.8). Este mapa resalta las regiones de la imagen donde la característica representada por el filtro es más prominente. Durante el proceso de entrenamiento, los pesos del filtro se ajustan automáticamente para capturar las características más relevantes para el problema específico (Khvostikov *et al.*, 2018).

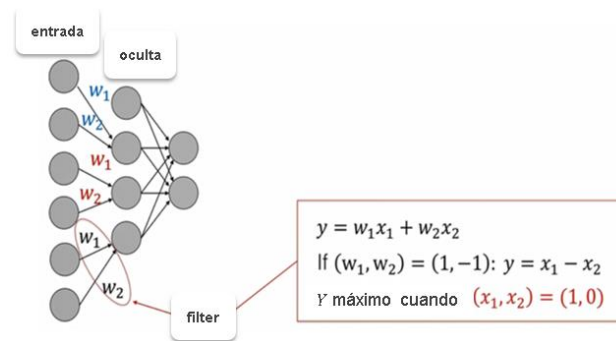


Figura 2.8. Filtro en una CNN (Adaptado de Ikeuchi, K. *et al.*, 2020)

Los filtros en las *CNN* son esenciales para la detección y representación jerárquica de características en datos visuales. Estos componentes, al adaptarse y refinar sus pesos durante el entrenamiento, permiten que las *CNN* se especialicen en identificar patrones relevantes. Finalmente es importante tener en cuenta que un núcleo se refiere generalmente a la matriz utilizada en la operación de convolución para un canal específico, mientras que un filtro se compone de uno o más núcleos que trabajan en todos los canales de la imagen de entrada para producir un mapa de características.

Convolución (Convolution)

Las imágenes a color tienen tres canales (*RGB*). Las *CNNs* manejan esto utilizando un filtro separado para cada canal. La profundidad del mapa de características resultante es igual al número de filtros aplicados. Después de la operación de convolución, se aplica una función de activación (*ReLU*) a cada píxel del mapa de características. Esto introduce no linealidades que permiten a la red aprender relaciones más complejas. A medida que se avanza en la profundidad de una *CNN*, las capas extraen características más abstractas y complejas (ver figura 2.9).

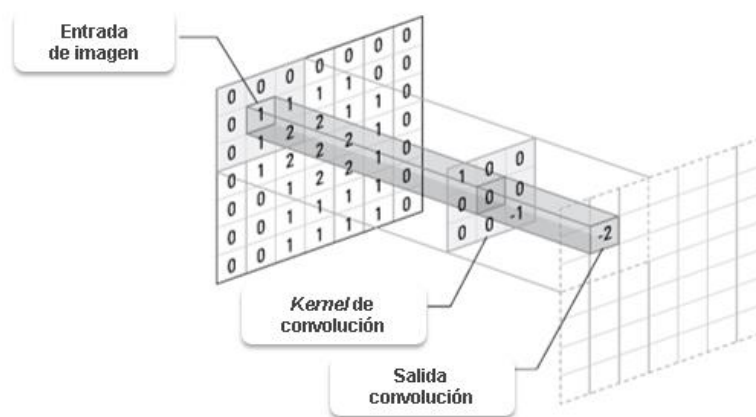


Figura 2.9. Operación de convolución (Adaptado de López Pinaya, W.H. *et al.* (2020))

Por ejemplo, con los filtros adecuados, es posible identificar bordes, texturas y otros detalles en una imagen. Una vez que un filtro ha aprendido una característica específica (como un borde vertical), puede reconocerla en cualquier parte de la imagen (ver figura 2.10). Esto le da a las *CNNs* la posibilidad de que la red puede reconocer características independientemente de su posición en la entrada (Khvostikov *et al.*, 2018).

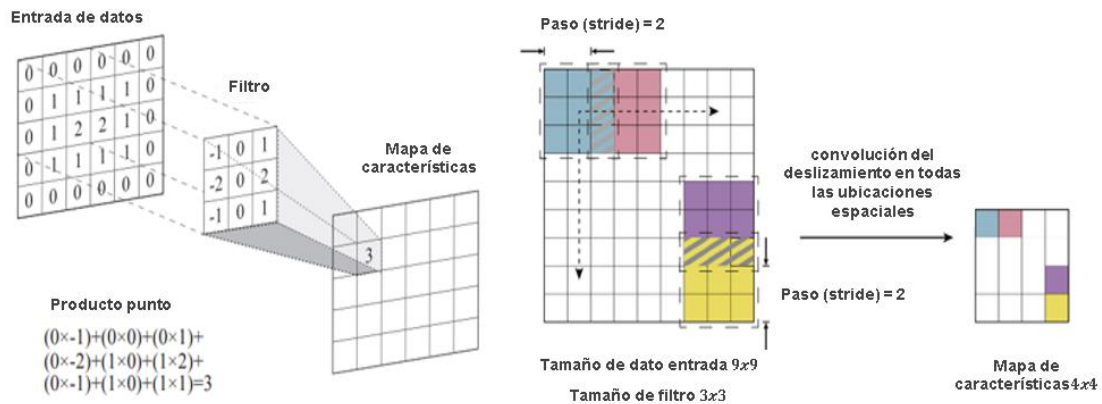


Figura 2.10. (Izquierda) operación producto punto. (Derecha) La operación de convolución (Adaptado de López Pinaya, W.H. *et al.*, (2020))

Como se ha indicado la convolución aprovecha la estructura espacial de los datos, donde los píxeles cercanos en una imagen están más relacionados entre sí que los que están distantes. Al operar sobre regiones locales, la convolución respeta y explora estas relaciones espaciales, y a esto es a lo que se le llama conexiones espaciales. Por otra parte, está la jerarquía de características, donde las múltiples capas de convolución en una *CNN* permiten la construcción de una jerarquía de características (Miceli, P.A., *et al.*, 2018). Las primeras capas pueden aprender características como bordes, mientras que las capas más profundas pueden combinar estas características básicas para reconocer patrones más complejos. Una de las principales razones es la eficiencia computacional donde el proceso de convolución es altamente paralelizable, lo que lo permite utilizar la aceleración con hardware específico, como unidades de procesamiento gráfico (*Graphic Processing Units, GPUs*), lo que se traduce en entrenamientos e inferencias más rápidas.

2.5.2. Parámetros e hiperparámetros

Sobre los parámetros

Los parámetros en una *CNN* se dividen principalmente en pesos (*weights*) y sesgos (*biases*), y su correcto ajuste es primordial para la precisión del modelo. A continuación, se detallan los tipos de parámetros y su función:

- *Los pesos (weights)* son los valores que se multiplican por la entrada en cada capa de la red para producir la salida.
- *Los sesgos (biases)* son los valores que se suman a la salida de cada capa de la red. Por ejemplo, en una capa convolucional, los sesgos se suman a los resultados de la convolución antes de aplicar la función de activación. En una capa totalmente conectada, los sesgos se suman a la salida de la multiplicación de los pesos y las entradas.

Sobre los hiperparámetros (*hyperparameters*)

Los hiperparámetros, son valores fijos que se eligen antes de entrenar el modelo y que surgen a la forma en que se realiza el entrenamiento. Por lo cual los hiperparámetros en el aprendizaje automático en general, son un parámetro cuyo valor se establece antes de que comience el proceso de entrenamiento del modelo. A diferencia de los parámetros de un modelo, que son aprendidos directamente durante el entrenamiento (como los pesos y sesgos en una red neuronal mencionados anteriormente), los hiperparámetros se configuran externamente y definen aspectos estructurales y operativos del proceso de entrenamiento.

Los hiperparámetros más conocidos y utilizados son los siguientes:

- a. *El tamaño de los filtros (kernel size)*: corresponde al tamaño de las matrices que se aplican a la imagen de entrada, con el objetivo de extraer características en una capa convolucional, donde el tamaño del filtro determinará la escala de las características que la *CNN* podrá captar. Así, mientras los filtros sean más grandes se pueden capturar patrones más amplios y los filtros pequeños pueden detectar detalles más finos.
- b. *La profundidad de la red*: que es el número de capas que tendrá la red. Una red más profunda puede aprender características más complejas, pero también puede sobreajustar (*overfitting*) los datos de entrenamiento. Con este parámetro se debe tener presente que es crítico, ya que una mayor profundidad permite a al modelo aprender jerarquías de características más complejas. Las capas adicionales aumentan la complejidad y costo computacional puede generar sobreajuste, lo que indica que modelo se ajusta demasiado a los datos de entrenamiento y no pueda generalizar.
- c. *La tasa de aprendizaje (Learning rate, η)*: este es un hiperparámetro que permite controlar el tamaño de las actualizaciones que se realizan sobre los parámetros del modelo que se está entrenando, ya que se utiliza para graduar la velocidad con la que se realizan cambios incrementales de los parámetros del modelo, en la figura 2.11 se muestran diferentes comportamientos de la tasa de aprendizaje.

Entonces para determinar la tasa de aprendizaje (*Learning rate*) adecuada, se realiza ajustando el valor y evaluando el desempeño de la red en un conjunto de datos de validación. Lo que permite la experimentación es encontrar una tasa de aprendizaje donde el modelo pueda converger rápidamente con un buen desempeño.

Es importante indicar el impacto de la tasa de aprendizaje en el modelo, teniendo en cuenta su comportamiento con el propósito de determinar cuán grandes son los ajustes realizados a los pesos del modelo en cada época del entrenamiento. Se inicia con una tasa de aprendizaje relativamente alta y se va reduciendo durante el

entrenamiento (1, 0.1, 0.01, 0.001).

En la figura 2.11 se observa el comportamiento de diferentes tasas de aprendizaje en el proceso de entrenamiento en relación con la función de pérdida. De esta forma la gráfica con una baja tasa de aprendizaje (*low learning rate*) indica que el modelo está aprendiendo muy lentamente y que posiblemente requiere más épocas de entrenamiento, lo que no sería eficiente (costo computacional). Si es una alta tasa de aprendizaje (*high learning rate*) la pérdida fluctúa, indica que el modelo está realizando actualizaciones demasiado grandes en sus pesos, y que tenga un aprendizaje inestable y conlleve a una divergencia en el entrenamiento. Una muy alta tasa de aprendizaje (*very high learning rate*) indica que la pérdida aumenta rápidamente, y que el modelo está divergiendo (función de pérdida o error no disminuye en el entrenamiento, aumenta con el tiempo).

De otro lado una buena tasa de aprendizaje (*good learning rate*), en este caso la pérdida disminuye rápida y de forma estable, que indica que la tasa de aprendizaje está bien ajustada y que el modelo está convergiendo de forma eficiente. Y la tasa de aprendizaje adaptativa (*adaptive learning rate*), donde la pérdida disminuye rápidamente al principio y luego se estabiliza, lo que indica que la tasa de aprendizaje se ajusta a lo largo del tiempo, disminuyendo mientras el modelo se acerca a la convergencia.

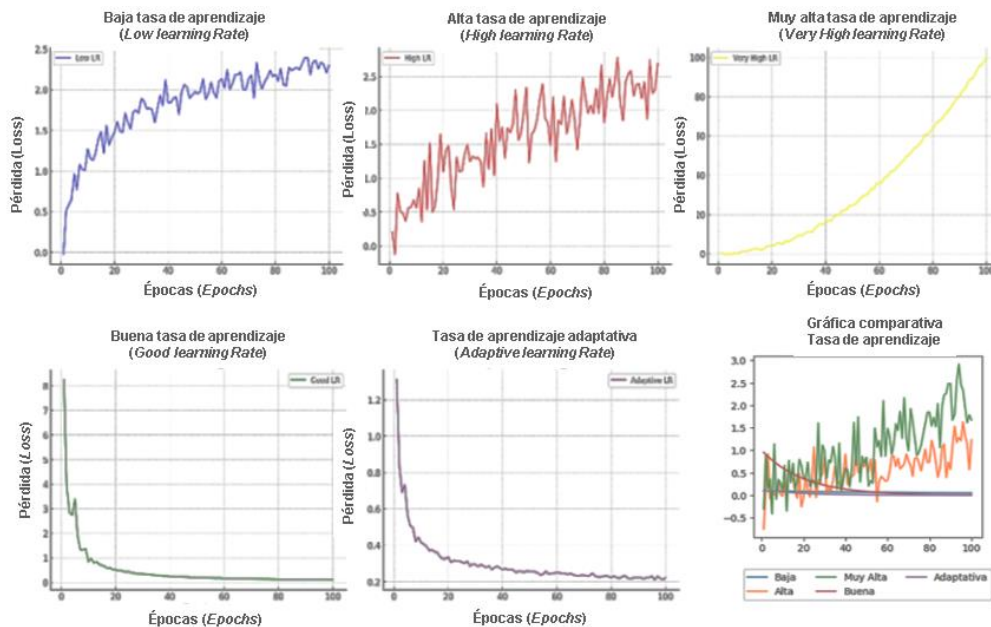


Figura 2.11. Importancia de la tasa de aprendizaje (adaptado de <http://cs231n.stanford.edu>)

- d. El tamaño del lote (*Batch size*): los parámetros de una *CNN* son los valores que se aprenden durante el entrenamiento del modelo. Son los pesos de las conexiones entre las diferentes capas de la red y son los que determinan cómo se transforma la

entrada en la salida. En una *CNN*, estos parámetros corresponden a los filtros convolucionales, las capas de *pooling* y las capas totalmente conectadas.

e. Funciones de activación: tales como *ReLU*, *Adam* entre otros.

En general, el proceso de experimentación es el que puede permitir ajustar los hiperparámetros en el entrenamiento de una *CNN*, orientados a mejorar la precisión del modelo.

2.5.3. El Conjunto de Datos (*Dataset*)

Para evaluar un modelo se debe tener un conjunto de datos independiente y representativo de los datos asociados al problema, donde estos están representados proporcionalmente. Se puede reservar $2/3$ para el conjunto de entrenamiento y $1/3$ para el para el conjunto de prueba, aunque es importante aclarar que esto depende del tipo de problema que se esté abordando y que con el proceso de experimentación con el modelo. Por tanto, siempre el conjunto de datos que se tenga disponible se divide en conjunto de entrenamiento para construir modelos y el conjunto de validación para probar diferentes parámetros del algoritmo de aprendizaje. Razonablemente podría ser, 60% para el conjunto de entrenamiento, 20% para el conjunto de validación y 20% para el conjunto de prueba. Así, en el aprendizaje supervisado, se dividen los datos en tres conjuntos principales:

- *Conjunto de entrenamiento (Entrenamiento Set)* Por ejemplo, podría contener el 70% ó 80% de las muestras.
- *Conjunto de validación (Validación Set)* que es el utilizado para validar los hiperparámetros y las métricas del modelo durante el entrenamiento. Es también un subconjunto del dataset utilizado para ajustar los hiperparámetros y validar el rendimiento del modelo durante el entrenamiento. Generalmente, podría ser del 15% ó 10% del *dataset* total.
- *Conjunto de prueba (Test Set)*: este conjunto no se utiliza en ninguna etapa del entrenamiento y podría contener, por ejemplo, el 10% ó 15% restante de las muestras.

Selección y extracción de características

La selección de características implica o consiste en escoger aquellos atributos que son realmente relevantes y que contribuyen a resolver el problema. Existen métodos automatizados para la seleccione y extracción de atributos donde ambos métodos pueden realizarse de forma manual, si este es el caso se requiere experiencia para seleccionar los que se consideren más relevantes para abordar el problema que se desee resolver.

A continuación, se expone de forma muy general los métodos de selección de características o atributos (*features selection*) que son los que permiten identificar los atributos que proporcionan información relevante orientado a elegir el mejor subconjunto de atributos. Hay dos métodos que son los algoritmos filtro (*filter*) y los algoritmos empotrados (*wrappers*) (Guyon, Isabelle *et al.*, 2003). El algoritmo filtro (*filter*) es donde los atributos o subconjuntos de atributos son evaluados de forma independiente del método de clasificación que se utilizará posteriormente. Y algoritmos empotrados (*wrappers*) donde el método de selección de características utiliza el clasificador que se usará posteriormente para evaluar qué características o subconjunto de características es el más adecuado.

Un algoritmo que selecciona las características de las variables más interesantes del conjunto de datos, lo hace previamente a la construcción del modelo de aprendizaje. Se realiza como una etapa de preprocesamiento de los datos, de forma independiente a la construcción del modelo. Hay un algoritmo filtro (*filter*) y un algoritmo empotrado (*wrapper*). El primero por filtro (*filter*) la selección de características se hace mediante el uso de filtros el cual es eficiente computacionalmente, y suele ser robusto frente al sobreaprendizaje, aunque puede tener una tendencia a seleccionar los conjuntos de las variables redundantes. El segundo la empotrado (*wrapper*) que en este caso se van seleccionando los conjuntos de variables y se van construyendo modelos alternativos. La evaluación de esos modelos sirve para decidir que variables, deberían incluirse en el modelo final (Sill Joseph, 2009). Ya en el modelo normalmente la selección de estas características en una *CNN* se realiza típicamente en dos etapas, la primera corresponde a la extracción de características, donde se utiliza una capa de convolución para extraer características importantes de las imágenes de entrada cual tiene varios filtros que se aplican a la imagen, y que pueden detectar patrones específicos en la imagen, tales como bordes, texturas o formas. Ya el segundo corresponde a la selección de características donde se eligen las características más relevantes y se descartan las menos importantes.

Este es un proceso importante con el cual se puede reducir la complejidad del modelo y el tiempo del proceso de entrenamiento, relacionado también con el costo computacional, al mismo tiempo que mejora la precisión del modelo al tomar las características más relevantes, aunque puede llevar a una pérdida de información, para lo que es indispensable un balance entre la cantidad de características seleccionadas y la precisión del modelo. Finalmente es importante indicar que el objetivo de la extracción de características es la determinación de un espacio de dimensionalidad que mantenga o preserve las características útiles y elimine la información que sea redundante.

La dimensionalidad

Y. Wu, G, *et al.*, (2022) proponen una estrategia de reducción de dimensionalidad por medio de un enfoque de red neuronal artificial, para simplificar el modelado inverso de filtros. Esta estrategia divide todas las variables en dos categorías para su procesamiento (ver figura 2.12).

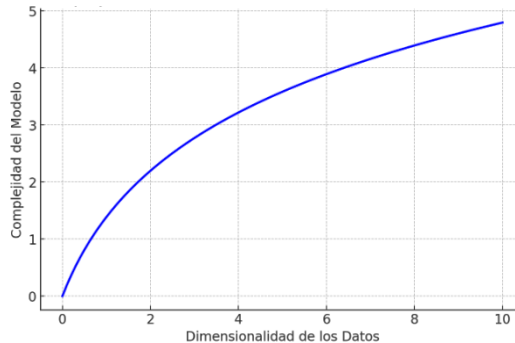


Figura 2.12. Relación conceptual entre la dimensionalidad de los datos y la complejidad del modelo (creación propia).

La dimensionalidad juega un papel muy importante en este proceso porque afecta la forma en que la red puede capturar patrones y características importantes en los datos. De igual forma W. W. Y. NG *et al.* (2002) indican que la dimensionalidad en las redes neuronales es el número de variables o características, que se utilizan para describir una instancia del conjunto de datos. S. Wojtowycsch *et al.* (2020), indican que la dimensionalidad es una de las problemáticas actuales y compara las principales estrategias, para reducir la dimensionalidad, la correlación entre características, la transformación y análisis de sensibilidad de las redes neuronales.

Respecto a la complejidad del modelo se refiere a la capacidad que tiene el modelo para adaptarse a la estructura de los datos, dado que a medida que aumenta la dimensionalidad el volumen del espacio de las características (atributos) aumenta exponencialmente. Se entiende que la complejidad del modelo debe estar determinada por la cantidad de datos disponibles, las características de estos y el problema que se está resolviendo (Hu, X. *et al.*, 2021).

- *El PCA (Análisis de Componentes Principales)*: en el caso de imágenes, se puede aplanar cada imagen en un vector y luego aplicar *PCA* para reducir la dimensión de ese vector. Sin embargo, en imágenes, el *PCA* puede no ser la técnica más efectiva debido a la estructura específica de los datos.
- *Las capas de agrupación (Pooling)*: en las capas de agrupación (como la capa de *MaxPooling*), se toma una ventana (por ejemplo, 2×2) y se selecciona el valor máximo (o promedio) dentro de esa ventana. Esto reduce el tamaño de la imagen y, por lo tanto, la cantidad de características, manteniendo características relevantes. El *pooling* reduce la dimensionalidad espacial de las características manteniendo las más destacadas, lo que puede ayudar a reducir la cantidad de parámetros en el modelo.
- *La selección de características manuales*: aquí se debe indicar cuales son las características específicas más importantes para la estimación de la altura, se

puede considerar seleccionar un subconjunto de características para utilizar en el modelo.

En última instancia, la elección de la técnica de reducción de dimensionalidad dependerá del problema específico, la cantidad de datos disponibles y el rendimiento que se pretenda alcanzar y evidentemente del problema a solucionar.

2.5.4. El *DataFrame*

El *DataFrame* es una estructura de datos semejante a una tabla de base de datos, consta de filas y de columnas con datos de un tipo numérico, categórico, texto, etc. (ver figura 2.13) utilizados para la manipulación, transformación y análisis de grandes volúmenes de datos.

filename_image	name_figure	height_figure	filename_image_shadow	name_image_shadow	shadow_length	illumination_source_angle
cilindros/cilindro1_00001G170.jpg	cilindro	1.0	cilindros/cilindro1_00001G170s.jpg	sombra_cilindro	1.0	45
cilindros/cilindro1_00001G180.jpg	cilindro	1.0	cilindros/cilindro1_00001G180s.jpg	sombra_cilindro	1.0	45
cilindros/cilindro1_00001G190.jpg	cilindro	1.0	cilindros/cilindro1_00001G190s.jpg	sombra_cilindro	1.0	45
cilindros/cilindro1_00001G215.jpg	cilindro	1.0	cilindros/cilindro1_00001G215s.jpg	sombra_cilindro	1.0	45
cubos/cubo2_00001G325.jpg	cubo	2.0	cubos/cubo2_00001G325s.jpg	cubo	2.0	45
cubos/cubo2_00001G335.jpg	cubo	2.0	cubos/cubo2_00001G335s.jpg	cubo	2.0	45
cubos/cubo2_00001G355.jpg	cubo	2.0	cubos/cubo2_00001G355s.jpg	cubo	2.0	45
cubos/cubo3_00001G0.jpg	cubo	3.0	cubos/cubo3_00001G0s.jpg	cubo	3.0	45
cubos/cubo3_00001G70.jpg	cubo	3.0	cubos/cubo3_00001G70s.jpg	cubo	3.0	45
cubos/cubo3_00001G80.jpg	cubo	3.0	cubos/cubo3_00001G80s.jpg	cubo	3.0	45
cubos/cubo3_00001G90.jpg	cubo	3.0	cubos/cubo3_00001G90s.jpg	cubo	3.0	45

Figura 2.13. Ejemplo de un *DataFrame* (creación propia)

Características y descripción del *DataFrame*

- La manipulación y limpieza:* esta fase es importante porque antes de entrenar modelos de *ML* o *DL*, los datos a menudo necesitan ser limpiados y transformados para mejorar la calidad del análisis. Los *DataFrames* permiten realizar operaciones como filtrado, reemplazo de valores faltantes y transformación de variables entre otras.
- La integración:* aquí los *DataFrames* tienen la capacidad para leer y escribir datos desde y hacia una variedad de formatos, incluyendo extensiones de tipo (*.csv), en *Microsoft Excel*, bases de datos, entre otros. Esta es una muy importante característica cuando se utiliza la integración de datos en proyectos de *ML* y *DL*.
- La exploración de datos:* aquí la estructura tabular de un *DataFrame* facilita la exploración y el análisis estadístico de los datos, ayudando a identificar correlaciones, tendencias y posibles valores atípicos.
- Facilita el *Features Engineering:* se refiere a que, en *ML*, la creación y transformación de características (*features*) es un paso esencial, con el cual los *DataFrames* permiten realizar operaciones vectorizadas para generar

rápidamente nuevas características a partir de las existentes.

- e. *La estructura organizada*: está relacionada con el *DataFrame* cuando este presenta una estructura organizada y tabular que facilita el almacenamiento y la manipulación de los datos, donde cada fila representa una instancia única del experimento, lo que hace que sea más sencillo rastrear y gestionar los datos.
- f. *Las etiquetas de sombra y figura*: en este caso como ejemplo están los campos *name_image_shadow* y *name_Figura* que proporcionan etiquetas claras y distintas para las sombras y las figuras en las imágenes. Esto es fundamental para la tarea de estimación de altura, ya que permite al modelo identificar y relacionar las partes relevantes de la imagen.
- g. *La información geométrica*: otro ejemplo son los campos como *shadow_length* e *illumination_source_angle* aportan información geométrica crucial sobre la formación de sombras en las imágenes. Esta información puede utilizarse para capturar los efectos de la iluminación y la perspectiva, lo que enriquece el conjunto de datos y mejora la capacidad del modelo para realizar estimaciones precisas.
- h. *La diversidad de escenarios*: el *DataFrame* abarca una variedad de escenarios de sombras al incluir imágenes con diferentes ángulos de iluminación, longitudes de sombra y alturas de figuras. Esta diversidad permite que el modelo se entrene en una amplia gama de condiciones, lo que lo hace más robusto y generalizado.
- i. *La evaluación rigurosa*: aquí al dividir el *DataFrame* en conjuntos de entrenamiento, validación y prueba, utilizando buenas prácticas para determinar la efectividad y el rendimiento del modelo, se realiza una evaluación rigurosa de cómo el modelo generaliza los datos no vistos.
- j. *El potencial para experimentación*: se determina si el *DataFrame* es lo suficientemente completo como para poder realizar experimentos y análisis detallados.
- k. Los datos multimodales: son los que incluyen imágenes, valores numéricos, así como información adicional como la altura real de la figura y la longitud de la sombra que es una fuente valiosa de datos multimodales.

2.5.5. Regresión

R. Setiono y A. Azcarraga (2001) proponen un método con el cual se genera un conjunto de ecuaciones de regresión lineal múltiple utilizando redes neuronales. El número de estas ecuaciones de regresión está determinado al agrupar las variables de entrada ponderadas, y las predicciones para las muestras del mismo grupo se calculan mediante la misma ecuación de regresión. Los resultados experimentales de datos

mostraron que el nuevo método genera relativamente pocas ecuaciones de regresión a partir de las muestras de datos de entrenamiento. Entonces se modelan las complejidades de los datos que, en lugar de utilizar directamente la salida de la red, generan un grupo de ecuaciones de regresión lineal múltiple, que proporcionan una estrategia para combinar la capacidad de modelado de redes neuronales con la regresión lineal múltiple, aprovechando el agrupamiento para manejar las complejidades de datos multivariados. Los modelos de regresión se utilizan para predecir las variables destino de forma continua, ya que en una regresión lineal el objetivo es abstraer en un modelo, la relación entre una o varias características y un variable destino continua.

Regresión lineal simple

En la figura 2.14 se puede observar la línea de ajuste o recta de regresión, las líneas verticales que van desde la recta de regresión, a los puntos de muestra son los llamados residuos (*offsets*) que son los errores en la predicción. Su objetivo es modelar una relación entre una característica simple (variable explicativa x) y una respuesta de valor continuo (variable destino y).

La ecuación de un modelo lineal se define así:

$$y = w_0 + w_1x$$

donde,

w_0 : eje interceptado

w_1 : coeficiente ponderado de la variable explicativa

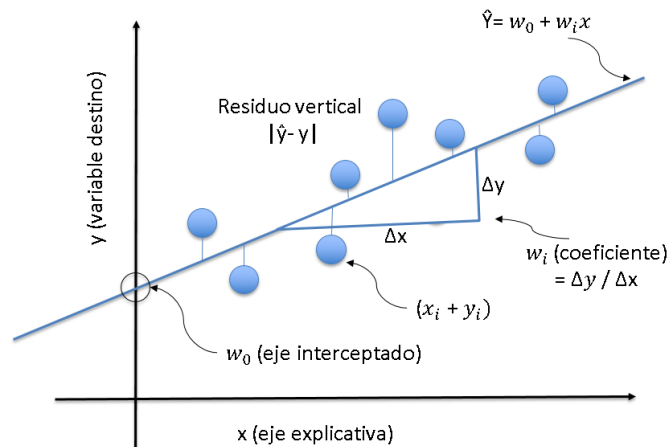


Figura 2.14. Línea de mejor ajuste en regresión lineal (recta de regresión) (Adaptado de Berzal, F., 2018)).

Regresión lineal múltiple

$$y = w_0x_0 + w_1x_1 + \dots + w_mx_m = \sum_{i=0}^m w_ix_i = w^T x \quad (2.1)$$

donde,

- w_0 : es el eje que intercepta con $x_0 = 1$,
- w_1 : coeficiente ponderado de la variable explicativa.

Predicción del modelo de regresión

En modelos de regresión, la precisión no se utiliza comúnmente como métrica de evaluación, dado que no se trata de clasificación sino de predicciones de valores numéricos continuos. Por tanto, en lugar de la precisión se utilizan las métricas específicas para modelos de regresión en este caso para evaluar la calidad de las predicciones en términos de error, lo que significa que es un proceso que estima valores numéricos continuos a partir de los datos proporcionados en la entrada del modelo.

También está la validación cruzada la cual evalúa la generalización del modelo a través de varias particiones (pliegues), de los datos en los conjuntos de entrenamiento y de prueba. Este procedimiento ayuda a reducir el sobreajuste y mejora la predicción del modelo en datos no vistos el cual se tratará en un capítulo posterior. La utilización de *MSE*, *RMSE* y *MAE*, junto con la validación cruzada y del análisis de residuales, son determinantes para la calidad de las predicciones del modelo más robusto.

Algunas de estas métricas son las más utilizadas para los modelos de regresión que se explican a detalle más adelante:

- El error absoluto medio (*Mean Absolute Error, MAE*)
- El Error cuadrático medio (*Mean Squared Error, MSE*)
- La raíz del error cuadrático medio (*Root Mean Squared Error, RMSE*)
- El coeficiente de determinación (*R-Squared, R²*)
- El coeficiente de correlación de *Pearson* (r)
- La suma de los cuadrados de los residuos (*Residual Sum of Squares, RSS*)
- El análisis de residuales (*residual analysis*)
- Las técnicas de regularización
- La validación cruzada (*cross-validación*)

La tarea de predicción mediante modelos de regresión de redes neuronales es compleja. La evaluación de dichos modelos va más allá de la simple aplicación de métricas de error. Implica una comprensión profunda del problema subyacente, la selección cuidadosa de los métodos de evaluación y pruebas, con la aplicación de técnicas de regularización y del análisis de robustez. En la figura 2.15 el eje y muestra los valores reales (puntos de color azul), y el eje x el valor predicho por el modelo (puntos de color rojo) la cual se utiliza para evaluar gráficamente las predicciones del modelo.

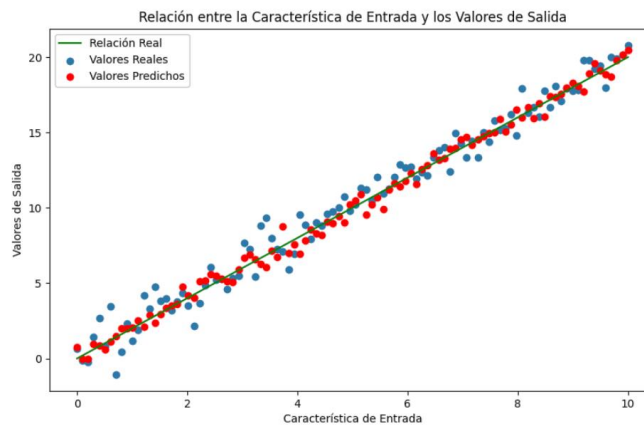


Figura 2.15. Ejemplo de la predicción del modelo: relación entre valores reales vs. valores predichos (creación propia)

La experimentación continua es fundamental para perfeccionar la predicción en modelos de regresión en redes neuronales. La experimentación no solo debe centrarse en la selección y optimización de la arquitectura de la red, sino también en la ingeniería de características y en la inclusión de diferentes tipos de datos que podrían enriquecer las capacidades predictivas del modelo.

2.5.6. Normalización

La normalización puede lograrse dividiendo cada valor de píxel por el valor máximo (255 en este caso). La normalización permite estandarizar la entrada de la red y puede mejorar la convergencia durante el entrenamiento. Para realizar el proceso de entrenamiento de las redes neuronales, los datos deben estar normalizados en todas las etapas del modelo, dado que estabiliza y acelera el entrenamiento por medio del descenso del gradiente.

Sobre la importancia de la normalización, está la convergencia con la cual los modelos de aprendizaje profundo suelen converger más rápido cuando las características están normalizadas. Por otro lado, mejora *el rendimiento*, lo que puede llevar a un rendimiento mejor y más consistente del modelo.

2.5.7. Generalización

La capacidad de generalización indica que al construir un modelo de *Machine Learning* este sea lo suficientemente robusto, se desempeñe bien no solamente con los conjuntos de validación y prueba, sino con datos que nunca haya visto, teniendo en cuenta que los datos de entrenamiento, validación y prueba sean diferentes.

Los siguientes son varios de los aspectos importantes sobre la generalización en redes neuronales:

- *El sobreajuste (Overfitting) y subajuste (Underfitting):* el sobreajuste se presenta en un modelo, cuando este se ajusta demasiado a los datos de entrenamiento, por lo tanto, cuanto mayor sea la complejidad del modelo aprendido, más tenderá a ajustarse al conjunto de datos de entrenamiento.

Por otro lado, el subajuste sucede cuando un modelo no ha capturado adecuadamente los patrones en los datos de entrenamiento ni en los nuevos datos (ver figura 2.16)

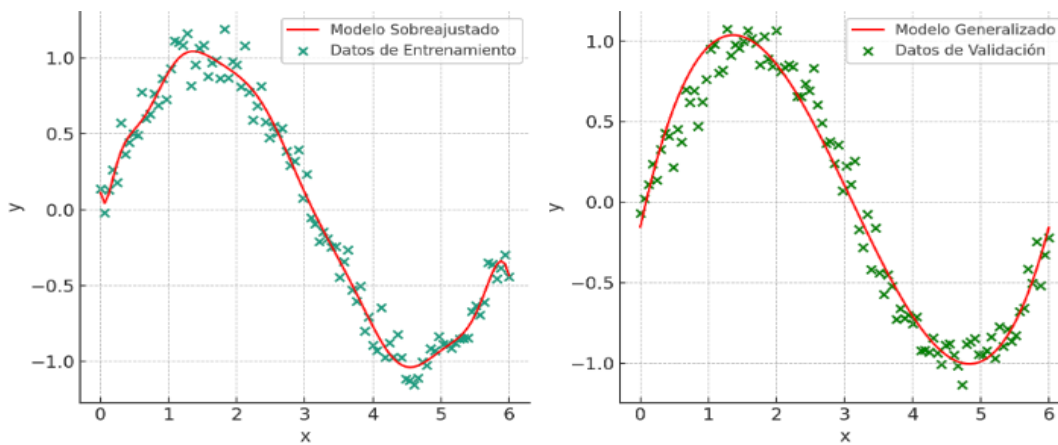


Figura 2.16. Generalización. Izquierda: conjunto de entrenamiento modelo sobreajustado (overfitting). Derecha: conjunto de validación modelo generalizado (creación propia)

- *La regulación y regularización:* se refiere a las técnicas de regularización, como la $L1$ y $L2$, que se utilizan para controlar el sobreajuste.
- *La validación cruzada (Cross-Validación):* para realizarla inicialmente se divide el conjunto de entrenamiento en varios subconjuntos para entrenar y evaluar el modelo varias veces. Esto proporciona una idea más precisa y clara sobre el rendimiento del modelo en los datos nuevos, lo que permite estimar detectar posibles problemas de generalización.
- *Conjunto de prueba y validación:* es importante dividir los datos en conjuntos de entrenamiento, validación y prueba. El conjunto de prueba se utiliza para evaluar el rendimiento final después de que el modelo está completamente entrenado.
- *Regularización y tamaño del conjunto de entrenamiento:* tener más datos de entrenamiento puede ayudar a mejorar la generalización al proporcionar al modelo una visión más completa de los patrones en los datos. Para el tamaño de los datos

existen las técnicas de regularización pueden ayudar a controlar el sobreajuste.

- *Balance entre complejidad y generalización*: hay un equilibrio delicado entre la capacidad del modelo para capturar patrones complejos en los datos y su capacidad para generalizar a nuevos datos. A medida que se aumenta la complejidad del modelo, puede ser más propenso al sobreajuste.
- *Dropout*: es una técnica de regularización que implica "apagar" aleatoriamente un porcentaje de neuronas durante cada paso de entrenamiento.
- *Regularización L1 y L2*: estas son técnicas que se aplican a la función de pérdida durante el entrenamiento para penalizar los pesos de las neuronas.
- *Selección del modelo y hiperparámetros*: se debe realizar una búsqueda sistemática para encontrar la configuración que genere un buen equilibrio entre rendimiento en entrenamiento y generalización.

2.5.8. *Ensembles*: combinación de modelos

A partir de la investigación desarrollada en ese documento, es posible mejorar el rendimiento de varios modelos por separado estando orientados en un mismo propósito. Hay investigaciones de que los *Ensembles* pueden combinar múltiples modelos predictivos mejorando el rendimiento y la generalización del modelo final ya que son más robustos y precisos que un modelo individual. Estos realizan una predicción a partir de las predicciones efectuadas por los modelos independientes. El *Ensemble* no construye un modelo, por el contrario, aplica un algoritmo para gestiona la información proporcionada por los modelos individuales. Ahora bien, la construcción de un *Ensemble* es un modelo de aprendizaje supervisado.

Según *Breiman Leo*. (1996), estos son los llamados meta-aprendizajes: aprender a partir de modelos de aprendizaje, donde el modelo derivado de este meta-aprendizaje puede ser de cualquier tipo. De forma empírica los *Ensembles* obtienen los mejores resultados cuando hay cierta diversidad en los modelos que los conforman (Breiman L., 2001). De igual forma Menze *et al.*, (2015), en su investigación propuso para la segmentación de tumores cerebrales (*Brain tumor segmentation*), mediante la utilización de ensamblaje de modelos multimodales (*Multimodal Brain Tumor Image Segmentation Benchmark*), donde los resultados indicaron que mediante el uso de redes neuronales profundas en modelos 3D, y la fusión de los mismos producen predicciones con una alta precisión. Esto lo probaron en cerca de 20 algoritmos avanzados de segmentación de tumores en 65 resonancias magnéticas. Con los resultados determinaron que ninguno de estos algoritmos fue el mejor, pero la fusión de modelos les permitió obtener resultados superiores a cualquier algoritmo individual.

Igualmente Kamnitsas *et al.* (2017), desarrollaron un sistema más fiable, en el cual

utilizaron siete modelos de *CNN*'s diferentes, donde los modelos se entrenaron de manera completamente independiente. Es así como, al combinar una colección heterogénea de modelos, construyeron un modelo insensible a fallos independientes de los componentes de *CNN*'s. Posteriormente las salidas de todas las arquitecturas definidas se combinaron (fusionaron) mejorando la capacidad de generalización. Este método propuesto reafirma la flexibilidad de las *CNN*s que, con un conjunto de técnicas, puede ser un enfoque prometedor para aplicar procesos de segmentación convolucionales.

De lo anterior es posible llegar a considerar que con el diseño y entrenamiento de varios modelos de *CNN*'s con imágenes que incluyan objetos y sombras, éstos posiblemente podrían aprender a correlacionar las dimensiones de la sombra con la altura del objeto, donde al combinar dichas salidas de estos modelos, se mejore la estimación de la altura, dado que cada modelo puede detectar características o errores que otro no. Es así, como el ensamblaje de modelos puede llegar a ser considerarse como una opción para esta investigación en un trabajo posterior.

2.6. Métricas de desempeño en modelos de regresión

Evaluación de modelos de regresión

Las métricas de desempeño proporcionan un punto de referencia que en este caso es cuantitativo, de cómo se comporta del modelo, donde la evaluación de modelos de regresión son un proceso holístico y multidimensional. Es decir, aplicar estas métricas entre muchas otras existentes a los modelos, no solo cuantifica el desempeño, sino que busca entender, de una manera más profunda y crítica en cuanto a la calidad, limitaciones y posibilidades que tiene el modelo, teniendo siempre presente que deben considerarse como un proceso evaluativo global.

De tal forma que estas métricas pueden aplicarse cuando se está utilizando una red neuronal convolucional para problemas de regresión, como la estimación de valores numéricos continuos (por ejemplo, la predicción de precios, puntuaciones, medidas, etc.). Estas son solo algunas de las métricas más comunes y eficientes para evaluar modelos de regresión. La elección de la métrica dependerá del contexto y los objetivos específicos del análisis. Es importante seleccionar la métrica que mejor se adapte a las necesidades y comprender su interpretación del rendimiento del modelo.

Aquí se indican algunas de las métricas más eficientes y utilizadas para evaluar modelos de regresión y que se utilizaron en esta investigación:

- La suma de cuadrados residuales (*Residual Sum of Squares, RSS*)
- El coeficiente de determinación (Coefficient of determination, R^2)
- El error absoluto medio (*Mean Absolute Error, MAE*)
- El error cuadrático medio (*Mean Square Error, MSE*)

- La raíz del error cuadrático medio (*Root Mean Squared Error, RMSE*)
- El coeficiente de correlación de *Pearson* (*Pearson Correlation Coefficient, r*)

Estas métricas se utilizan para cuantificar el rendimiento del modelo, mientras que las técnicas son métodos más generales para evaluar y mejorar ese rendimiento.

2.6.1. Error Absoluto Medio (*Mean Absolute Error, MAE*)

El *MAE* es la media de las diferencias absolutas entre los valores predichos y los valores reales. A diferencia del *RMSE*, no se elevan al cuadrado, lo que lo hace menos sensible a valores atípicos y errores grandes. El *MAE* es la métrica más simple y directa para la evaluación para la evaluación del grado de divergencia entre dos conjuntos de valores, representado por la ecuación, que calcula el promedio de las diferencias absolutas entre las predicciones y los valores reales. Se expresa como:

$$MAE = \frac{1}{|D|} \sum_{d \in D} |f(d) - h(d)| \quad (2.2)$$

donde D es el número de instancias, $h: X \rightarrow \mathbb{R}$ es la función del modelo y $f: X \rightarrow \mathbb{R}$ es la función objetivo con las etiquetas de las instancias. Para este caso todos los residuos tienen la misma contribución al absoluto final.

En este caso sería:

$$MAE = \frac{1}{n} \sum_{d \in D} |predicción - valor\ real| \quad (2.3)$$

donde n es el número de ejemplos en el conjunto de datos. El *MAE* proporciona una medida directa del error promedio en unidades de la variable objetivo (en este caso, altura).

O también puede expresarse como la media de las diferencias, en valor absoluto, entre las predicciones $f(x)$ y las observaciones y :

$$MAE = \frac{1}{N} \sum_{i=1}^N |f(x) - y| \quad (2.4)$$

2.6.2. Error Cuadrático Medio (*Mean Square error, MSE*)

El *MSE* se utiliza para realizar comparaciones entre modelos de regresión o en su defecto para la modificación de los parámetros. Es la métrica más empleada para la evaluación de los modelos de regresión y se calcula de la siguiente forma, y que se aplica sobre las N muestras del conjunto de datos, que debe ser con el que se esté evaluando la calidad del modelo cuantitativo:

$$MSE = \frac{1}{N} \sum_{i=1}^N (f(x) - y)^2 \quad (2.5)$$

$$MSE = \frac{1}{|D|} \sum_{d \in D} (f(d) - h(d))^2 \quad (2.6)$$

donde D es el número de instancias, $h: X \rightarrow \mathbb{R}$ es la función del modelo y $f: X \rightarrow \mathbb{R}$ es la función objetivo con las etiquetas de las instancias. Con la utilización del MSE se penaliza los residuos grandes. En ese sentido, si el modelo se aproxima correctamente a gran parte de las instancias del conjunto de datos, pero comete importantes errores en unos pocos, la penalización en esta métrica será muy superior a la indicada si se emplea la métrica (MAE).

2.6.3. Raíz del error cuadrático medio ($RMSE$ - *Root Mean Squared Error*)

El $RMSE$ mide la desviación de las diferencias entre los valores de las predicciones $f(x)$ y los valores observados y . Estas diferencias se les denomina residuos cuando se estiman sobre el conjunto de datos con el cual se construye el modelo. Se expresa como:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (f(x) - y)^2} \quad (2.7)$$

donde:

- n : es el número total de observaciones.
- y_i : es el valor observado o real de la i –ésima observación.
- \hat{y}_i : es el valor predicho para la i –ésima observación

Esto produce una métrica absoluta, cuanto más pequeño el valor mejor será el modelo, que representa el número promedio de las alturas en los que las predicciones son erróneas.

Es importante tener en cuenta que al momento de utilizar los errores al cuadrado puede ocasionar que los cálculos realizados sean sensibles a la presencia de anomalías (*outliers*), lo que se sugiere es utilizar el error absoluto medio (*Mean Absolute Error, MAE*) en lugar del $RMSE$. Sin embargo, también es prudente utilizar varios criterios para evaluar la calidad del modelo al igual que diferentes métricas para el problema en particular que esté tratando.

2.6.4. Coeficiente de determinación (R^2)

R^2 , permite estimar el nivel de cambio en la variable dependiente, donde los valores oscilan entre 0 y 1: un R^2 cercano a 1 indica que el modelo capta adecuadamente la variabilidad inherente en los datos, mientras que un valor cercano a 0 indica que el modelo no explica la variabilidad de la variable de respuesta. En términos prácticos, R^2

ofrecen una medida relativa de qué tan bien las predicciones del modelo, se alinean con los valores reales observados, o en otras palabras R^2 proporciona una idea de que tan bien el modelo de regresión lineal se ajusta a los datos, lo que hace que sea uno de los métrica de evaluación para poder realizar ajustes a un modelo de regresión. En la figura 2.17 se observa que el R^2 mejora a medida que aumentan el número de épocas, que sería en este ejemplo que el modelo está aprendiendo y ajustándose mejor a los datos a través del tiempo.

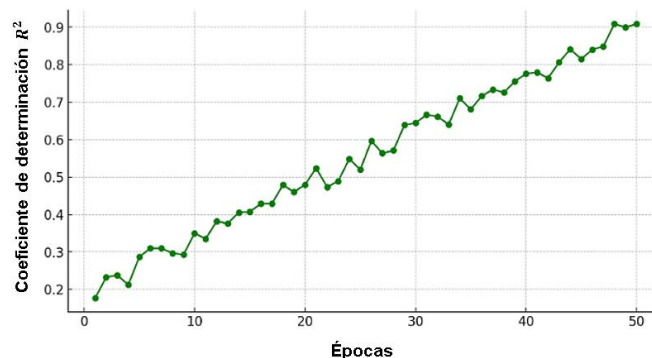


Figura 2.17. Gráfico del coeficiente de determinación (R^2) en una CNN con 50 épocas. (creación propia)

La Interpretación numérica del coeficiente de determinación (R^2) es la siguiente:

- Un R^2 de 0 indica que el modelo no explica ninguna de la variabilidad de la variable dependiente alrededor de su media.
- Un R^2 de 1 indica que el modelo explica toda la variabilidad de la variable dependiente alrededor de su media.
- Por ejemplo, un R^2 de 0,75 sugiere que el modelo explica el 75% de la variabilidad de la variable de respuesta.

2.6.5. Coeficiente de Pearson o de correlación de Pearson (r)

El coeficiente de correlación de *Pearson* (r), llamado también coeficiente de *Pearson* o correlación de *Pearson* r se define como:

$$r = \frac{\sum_{i=1}^n (x_i - x_{media})(y_i - y_{media})}{\sqrt{\sum_{i=1}^n (x_i - x_{media})^2 \sum_{i=1}^n (y_i - y_{media})^2}} \quad (2.8)$$

Donde:

x_i y y_i : son los valores individuales de las dos variables que se están correlacionando.
 x_i y y_i : son las medias de las variables x e y respectivamente

n : es el número de observaciones.

El valor de r siempre está entre -1 y 1 , inclusive. La magnitud del valor indica la fuerza de la correlación, mientras que el signo indica la dirección:

- $r = 1$: correlación lineal perfecta positiva. Cuando la correlación es cercana a 1 , sugiere una fuerte relación lineal positiva. Esto significa que a medida que una variable aumenta, la otra también tiende a aumentar.
- $r = -1$: correlación lineal perfecta negativa. Cuando la correlación es cercana a -1 , indica una fuerte relación lineal negativa. Esto quiere decir que cuando una variable aumenta, la otra tiende a disminuir.
- $r = 0$: no hay correlación lineal. Ahora bien si la correlaciones cerca a 1 sugiere que no hay una relación lineal evidente entre las dos variables.

En la figura 2.18 es visible la relación de un modelo de ejemplo de regresión entre los valores predichos y los valores reales, al igual que el r , que es aproximadamente en este ejemplo de $0,94$, el cual indica una fuerte correlación positiva entre los valores reales y los valores predichos. De tal manera que a medida que los valores reales aumentan los valores predichos también aumentan y viceversa. Entonces los valores del coeficiente de correlación de *Pearson* varían de 1 a -1 , donde un valor de 1 es una correlación perfecta positiva, -1 indica una correlación negativa perfecta y 0 significa que no hay correlación lineal.

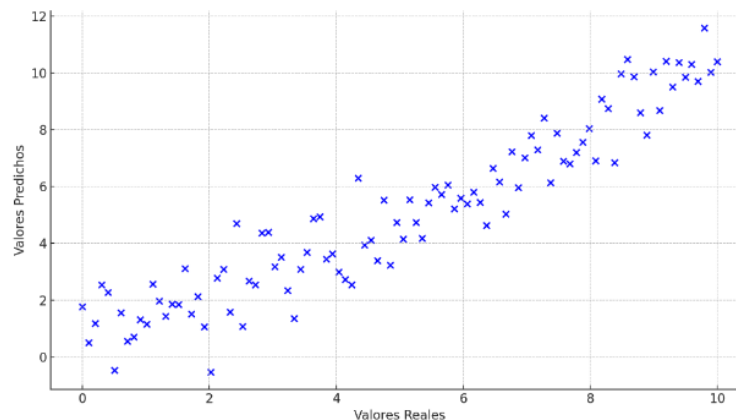


Figura 2.18. Gráfico de dispersión y coeficiente de correlación de *Pearson* (r). (creación propia)

Es importante indicar que aunque están relacionados, R^2 y el coeficiente de correlación de *Pearson* tienen usos e interpretaciones distintas. R^2 se utiliza más en el contexto de la evaluación de modelos, mientras que el coeficiente de correlación de *Pearson* se utiliza más para evaluar la relación lineal entre dos variables.

2.6.6. Suma de los Cuadrados Residuos (*Residual Sum of Squares, RSS*)

Para un conjunto de n observaciones, el *RSS* se expresa como:

$$RSS = \sum_{i=1}^n (y_i - \hat{y})^2 \quad (2.9)$$

Donde \hat{y} es el valor real de la i –ésima observación y \hat{y} es el valor predicho por el modelo para la i –ésima observación. La suma se extiende a todas las n observaciones en el conjunto de datos. Según lo anterior el objetivo al entrenar un modelo de regresión es generalmente minimizar el *RSS*. Al hacerlo, estamos buscando los parámetros del modelo que hacen que la predicción se acerque lo más posible a los valores reales, al menos para los datos de entrenamiento.

En la figura 2.19 el eje horizontal el número de valores considerados y el eje vertical el valor acumulativo de *RSS*, donde cuanto menor sea el *RSS* es mejor y se estaría entendiendo que el modelo se está justando a los datos.

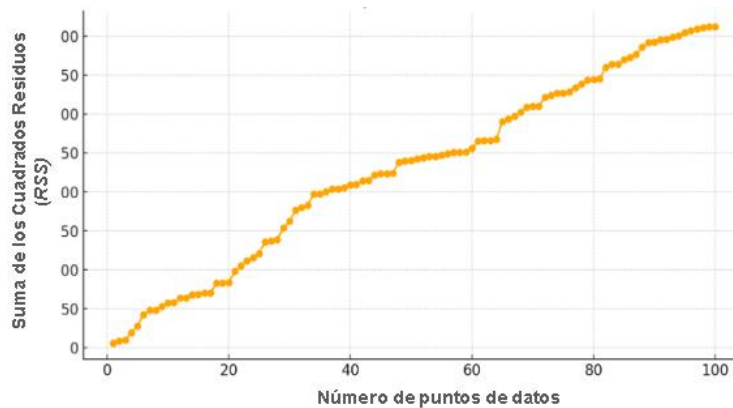


Figura 2.19. Ejemplo de la suma de los cuadrados de los residuos (*RSS*).
(creación propia)

Riesgos del *RSS* Como Única Métrica

Aunque minimizar el *RSS* es un objetivo válido, hay que tener en cuenta varios aspectos, tales como el sobreajuste el cual cuando se presenta un valor muy bajo de *RSS* en el conjunto de entrenamiento puede indicar que el modelo se ha ajustado demasiado a esos datos, capturando incluso el ruido, y podría no generalizar bien a nuevos datos. Y por otra parte está el subajuste que es cuando un valor alto de *RSS* podría indicar que el modelo es demasiado simple para capturar la complejidad subyacente en los datos.

Relación con otras métricas

RSS está relacionado también con métricas como el error cuadrático medio (MSE) y el coeficiente de determinación (R^2). Con el (MSE) es simplemente el RSS dividido por el número de observaciones n así:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.10)$$

Y con el R^2 proporciona un indicio de cuánta variabilidad en y es explicada por el modelo. Esta se calcula a partir del RSS y la suma total de cuadrados (TSS), que es la variabilidad total alrededor de la media, así:

$$R^2 = 1 - \frac{RSS}{TSS} \quad (2.11)$$

2.6.7. Método de validación cruzada (*Cross Validación Method*)

En la validación cruzada (*K-Fold Cross-Validación*) de k iteraciones, se realiza la división de forma aleatoria del conjunto de datos de entrenamiento en k iteraciones, de las cuales $k - 1$ iteraciones se utilizan para el entrenamiento del modelo y una iteración para evaluación del rendimiento del modelo. Por tanto, este proceso se repite k veces para obtener k modelos y estimación es de rendimiento. Cuando la validación termina en la última iteración k definida, ya se puede estimar el rendimiento medio del modelo que esta soportado, en cada una de las iteraciones (que son independientes) para obtener la dicha estimación de rendimiento, donde lo que se pretende es encontrar los valores óptimos de la configuración de los hiperparámetros esperando que produzcan un rendimiento de la generalización óptima (Markatou, H. *et al.*, 2005).

La Figura 2.20 muestra el proceso completo de la realización del proceso de una validación cruzada de forma general aplicado sobre un conjunto de entrenamiento (Dataset). Es importante indicar que en la validación cruzada se evalúa los resultados de las iteraciones por medio de un análisis estadístico, donde el conjunto de datos está dividido en pliegues de prueba y pliegues de entrenamiento, pero teniendo en cuenta que en cada iteración un pliegue diferente se utiliza como un conjunto de prueba y el porcentaje restante como conjunto de entrenamiento.

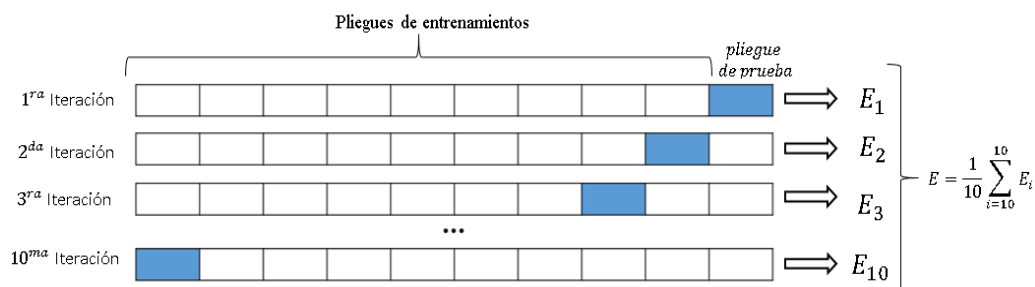


Figura 2.20. Caracterización de la validación cruzada (Berzal, 2018).

Un valor utilizado para el valor de k en la validación cruzada es $k = 10$ en k iteraciones según Kohavi Ron (1995) es de 10 el cual ofrece la mejor compensación entre el sesgo y la varianza. De igual forma se sugiere también que si es esta utilizando un conjunto de datos pequeño puede ser de utilidad para aumentar el número de las iteraciones, que por consiguiente si aumento el valor de k se usaran mas datos de entrenamiento en cada iteración lo cual da como resultado un sesgo más bajo para estimar el rendimiento de generalización promediando las estimaciones individuales del modelo (B. Efron *et al.*, 1995). Por tanto, si el valor de k es alto también hará que aumente el tiempo de ejecución del algoritmo de validación cruzada lo que generara valores de varianza más altos, a razón de que las iteraciones de entrenamiento serán más parecidas entre sí (German Stuart *et al.*, 1995).

En resumen, los resultados de aplicar la validación cruzada (sin remplazo) indica que tan bien o mal funciona el modelo final, se obtiene información acerca de que tan variable puede ser la varianza, y de ahí o relevante de diseñar o plantear las pruebas para evaluar un modelo. Entonces se puede decir que la validación cruzada evalúa el rendimiento y la generalización de un modelo, como la red neuronal convolucional (CNN) estima la altura de figuras a partir de sombras, en este caso.

En la figura 2.21 el conjunto de datos se observa que está dividido en $k = 5$ particiones, cada una de estas los datos de color azul corresponden a los datos utilizados para el entrenamiento en cada iteración del modelo, mientras que los datos de color rojo son los datos de prueba utilizados para evaluar el modelo, teniendo en cuenta su capacidad para generalizar nuevos datos y la eficiencia del modelo.

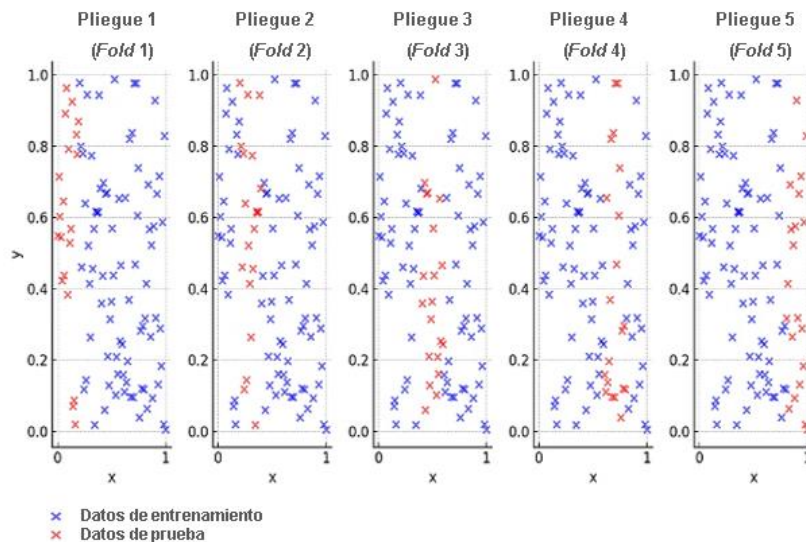


Figura 2.21. Visualización validación cruzada (Cross-Validación) con $k = 5$ (folds).
(creación propia)

Aquí hay una descripción paso a paso de cómo se aplica la validación cruzada al

modelo:

- *División del conjunto de datos:* se divide el conjunto de datos en K subconjuntos (pliegues) aproximadamente del mismo tamaño. K es un número que determina cuántos pliegues se usan en la validación cruzada (por ejemplo, $K = 5$ ó $K = 10$).
- *Ciclo de Validación Cruzada:* para cada pliegue i de 1 a K :
 - Divide el conjunto de datos en un conjunto de entrenamiento y conjunto de prueba. El conjunto de prueba es el pliegue i , y el conjunto de entrenamiento es el resto de los pliegues.
 - Entrenamiento del modelo utilizando la *CNN*.
- *Promedio de resultados:* este valor se obtiene al completar el ciclo de validación cruzada para todos los pliegues, entonces se calcula el promedio de las métricas de rendimiento, (*LOSS*, *MAE*, *MSE*) con el propósito de evaluar del rendimiento del modelo.
- *Interpretación de resultados:* se debe analizar los resultados promediados para comprender el rendimiento general del modelo en diferentes conjuntos de prueba. Esto proporcionará una idea de cómo se comporta el modelo en diferentes configuraciones de datos. En *TensorFlow* o *Keras*, se puede utilizar la función *fit* con argumentos como *validación data* para realizar la validación cruzada.

En la figura 2.22 se observan las barras azules que representa la precisión estimada por el modelo en un *fold* específico, y la línea roja indica o representa la precisión media obtenida a lo largo del proceso de la validación en cruzada en cada *fold*, que es una estimación general del rendimiento del modelo.

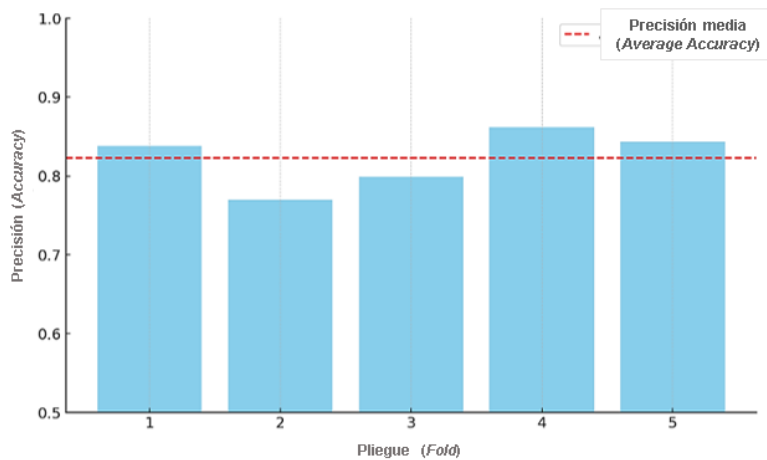


Figura 2.22. Resultados de la validación cruzada. (creación propia)

2.7. Metodología de implementación

La metodología incremental es una estrategia de desarrollo de software que se construye en etapas sucesivas (denominada iteración en este trabajo), permitiendo que cada etapa añada funcionalidad al producto final. Esta estrategia es particularmente útil en proyectos en los que se requiere retroalimentación y adaptarlos a las necesidades cambiantes (León Yacelga, R.A., *et al.*, 2021). En la figura 2.23 el diagrama representa un flujo del proceso iterativo de software. Este enfoque iterativo puede adaptarse a los cambios y mejorar con los ciclos de retroalimentación y mejora. En este tipo de modelos se observa que la metodología incremental es una estrategia para desarrollar software de forma gradual, que se divide en capas sucesivas denominadas iteraciones, donde en cada iteración se agregan nuevas funcionalidades al producto final, lo que permite adaptarlo a las necesidades cambiantes y realizar retroalimentación continua.



Figura 2.23. Etapas de desarrollo del modelo. (Martin, R.C., 2014)

La utilización de esta metodología en este trabajo obedece a que tiene varias ventajas una de ellas es su flexibilidad, ya que permite adaptarse a cambios en los requisitos o prioridades del proyecto. También está la reducción de riesgos ya que al dividirse el proyecto se minimizan los riesgos asociados al desarrollo completo en una sola fase. Otra muy importante es la validación temprana la cual permite obtener retroalimentación a lo largo del proceso, lo que es muy útil porque permite corregir errores y mejora el producto final. También es importante indicar que con esta metodología se tiene mayor control del proyecto ya que facilita planificación y seguimiento al progreso del proyecto.

De igual forma es importante mencionar que la utilización de esta metodología también tiene algunas desventajas, como por ejemplo la mayor complejidad de la gestión ya que requiere planificación y gestión más cuidadosa de las iteraciones. También la posible sobrecarga de documentación para cada iteración, dado que si no se gestiona correctamente puede ser difícil mantener la coherencia del producto final a lo largo de las iteraciones.

El desarrollo de modelos de aprendizaje profundo requiere de un enfoque estructurado y sistemático para garantizar su eficacia y precisión. A continuación, se detallan como se

utilizó esta metodología en los modelos de redes neuronales convolucionales, de tal forma que en esta metodología cada iteración debe cumplir con los siguientes requisitos:

a. Definición de Requerimientos:

- Establecimiento de objetivos claros para la red neuronal.
- Definición de las características y funcionalidades que se desean implementar en cada incremento.
- Priorización de los requerimientos según importancia y riesgo.

b. Análisis y diseño:

- Diseño de la arquitectura de la red neuronal basado en los requerimientos.
- Selección de parámetros, capas y estructura general que se adaptarán en los siguientes incrementos.

c. Codificación:

- Implementación de la arquitectura diseñada.
- Uso de herramientas para probar la arquitectura
- Asegurar que el código sea modular para facilitar la incorporación de futuros incrementos.

d. Pruebas:

- Validación y verificación del modelo implementado en este incremento.
- Uso de conjuntos de datos de validación para evaluar el rendimiento del modelo.
- Iteración en el diseño y la codificación basado en los resultados de las pruebas.

e. Entrega:

- Despliegue del incremento del modelo de red neuronal.
- Presentación de los resultados y logros al cliente o al equipo de proyecto.
- Recopilación de retroalimentación para informar los siguientes incrementos.

f. Retroalimentación y revisión:

- Evaluación de la funcionalidad entregada en el entorno real.
- Recopilación de comentarios y sugerencias para mejoras.
- Planificación de los siguientes incrementos basándose en la retroalimentación recibida.
- Este proceso se repite hasta que el sistema completo esté implementado y cumpla con todos los requerimientos.

Capítulo 3. Modelo Propuesto

3.1. Descripción general del modelo

Para el desarrollo de los dos modelos se indica a continuación de forma general los siguientes pasos:

Inicialmente para la *preparación de datos* se desarrolló totalmente una base de datos con fotografías de las figuras, de las sombras, un *DataFrame* con los datos correspondiente a las figuras geométricas. Estos campos son el nombre del archivo, nombre de la figura, altura de la figura, nombre de la sombra, entre otros. La base de datos contiene figuras geométricas de diferentes volúmenes, diferentes alturas indicadas en centímetros (unidades), texturas y diferentes niveles de iluminación con el objetivo de mejorar la precisión de la red neuronal.

Posteriormente en el *preprocesamiento de los datos*, la información que en este caso son las imágenes fotográficas y la información del *dataframe*. Los primeros se normalizaron, se redimensionaron, se extrajo la información de la sombra y de la figura por separado para ser utilizada por los dos modelos. La segunda corresponde a información relacionada con las imágenes.

En el *diseño de la red neuronal*, se utilizó el enfoque de las redes neuronales convolucionales para el trabajo con las imágenes y los datos. Su arquitectura es el de una red *CNN* con regresión, ya que esta como salida indica un valor correspondiente a la estimación o predicción de la altura.

En el *entrenamiento de la red neuronal convolucional* se distribuye las imágenes en los conjuntos de entrenamiento, de validación y de prueba.

La *prueba y evaluación de la red neuronal* se realizaron después del entrenamiento de la red neuronal, con imágenes de sombras de formas nunca vistas por la red, y se evalúa para determinar si nivel de ajuste.

La *Implementación* de la red neuronal convolucional se realizó sobre ambientes de desarrolló en Anaconda, *TensorFlow*, entre otros, utilizando como lenguaje de programación a Python.

3.2. Implementación de la metodología

Considerando la metodología propuesta del capítulo anterior, en el desarrollo del trabajo se realizaron 17 iteraciones, donde cada una de estas iteraciones corresponde a un mismo ciclo, es decir a la definición de requerimientos, diseño, codificación, pruebas,

entrega, y retroalimentación:

Las 17 iteraciones del modelo son las siguientes:

- Iteración 1: Construcción y preprocesamiento inicial del *Dataset*
- Iteración 2: Proceso de captura de imágenes y preprocesamiento
- Iteración 3: Diseño y prototipo inicial de la red neuronal
- Iteración 4: Entrenamiento y validación inicial
- Iteración 5: Implementación y pruebas de campo
- Iteración 6: Integración de técnicas de aprendizaje profundo adicionales
- Iteración 7: Ampliación del *Dataset* y pruebas de generalización
- Iteración 8: Optimización y ajustes
- Iteración 9: Aumento de la robustez del modelo frente a variaciones de iluminación
- Iteración 10: Validación y pruebas de generalización
- Iteración 11: Mejora de la precisión y la robustez
- Iteración 12: Integración y sincronización de datos de imágenes y sombras
- Iteración 13: Mejora del modelo utilizando características adicionales
- Iteración 14: Análisis detallado de la relación entre sombras y alturas
- Iteración 15: Evaluación rigurosa
- Iteración 16: Preparación del modelo para diferentes entornos de iluminación

A continuación, se detalla cada una de las iteraciones anteriores:

Iteración 1: construcción y preprocesamiento inicial del *Dataset*

- Definición de requerimientos: crear un conjunto de datos con imágenes de sombras de figuras geométricas y sus alturas correspondientes.
- Diseño: seleccionar figuras geométricas (esferas, cubos, cilindros) y capturar imágenes desde varios ángulos e iluminaciones.
- Codificación: generar un *dataframe* con datos de las figuras como nombre, altura y sombra correspondiente.
- Pruebas: verificar la diversidad y representatividad del *dataset*.
- Entrega: *dataset* inicial de 2.244 imágenes.
- Retroalimentación y revisión: evaluar la adecuación del *dataset* para entrenamiento inicial.

Iteración 2: refinamiento del proceso de captura de imágenes y preprocesamiento

- Definición de requerimientos: asegurar la calidad y consistencia en la captura de imágenes.
- Diseño: establecer un protocolo riguroso para la toma de fotografías, teniendo en cuenta la iluminación, la superficie y la distancia de la cámara.
- Codificación: automatizar el proceso de ajuste y organización de imágenes según su tipo y altura.
- Pruebas: verificar la uniformidad y calidad de las nuevas imágenes capturadas.

- Entrega: un conjunto de datos mejorado y más consistente.
- Retroalimentación y revisión: evaluar la eficacia del nuevo protocolo de captura de imágenes.

Iteración 3: diseño y prototipo inicial de la red neuronal

- Definición de requerimientos: diseñar una *CNN* básica para la regresión que para estimar la altura de un objeto dentro de una imagen 2D a partir de la sombra.
- Diseño: crear una arquitectura con capas convolucionales, de agrupamiento y completamente conectadas.
- Codificación: implementar la arquitectura propuesta en un entorno de desarrollo.
- Pruebas: probar el modelo con un subconjunto del *dataset*.
- Entrega: un prototipo inicial de la red neuronal.
- Retroalimentación y revisión: recoger *feedback* sobre la arquitectura inicial y su rendimiento.

Iteración 4: entrenamiento y validación inicial

- Definición de requerimientos: entrenar la *CNN* con el conjunto de datos.
- Diseño: ajustar hiperparámetros como tasa de aprendizaje, tamaño del lote y número de épocas.
- Codificación: realizar el entrenamiento utilizando técnicas como la validación cruzada.
- Pruebas: evaluar el modelo en el conjunto de datos de validación.
- Entrega: un modelo entrenado y validado inicialmente.
- Retroalimentación y revisión: ajustar la arquitectura o hiperparámetros según sea necesario.

Iteración 5: implementación y pruebas de campo

- Definición de requerimientos: implementar el modelo en una aplicación o sistema para uso práctico.
- Diseño: desarrollar una interfaz para la entrada de imágenes de sombra y la salida de alturas estimadas.
- Codificación: integrar la *CNN* en una aplicación con funcionalidades de predicción.
- Pruebas: realizar pruebas de campo para evaluar el rendimiento en condiciones reales.
- Entrega: una aplicación o sistema completamente funcional con la *CNN* integrada.
- Retroalimentación y revisión: recopilar *feedback* de los usuarios y analizar el rendimiento en el mundo real.

Iteración 6: integración de técnicas de aprendizaje profundo adicionales

- Definición de requerimientos: explorar y aplicar técnicas avanzadas de aprendizaje profundo para mejorar el rendimiento del modelo.
- Diseño: investigar enfoques como redes neuronales profundas, transferencia de

aprendizaje, o redes generativas adversarias (GANs) que podrían mejorar la precisión del modelo.

- Codificación: integrar y entrenar el modelo con las nuevas técnicas.
- Pruebas: evaluar el impacto de estas técnicas en la precisión y generalización del modelo.
- Entrega: un modelo enriquecido con técnicas avanzadas de aprendizaje profundo.
- Retroalimentación y revisión: analizar la efectividad y la aplicabilidad de las técnicas incorporadas.

Iteración 7: ampliación del dataset y pruebas de generalización

- Definición de requerimientos: ampliar el *dataset* para incluir mayor diversidad en alturas, iluminaciones y ángulos de las figuras geométricas.
- Diseño: planificar la captura de imágenes adicionales considerando las variaciones requeridas.
- Codificación: incorporar las nuevas imágenes al *dataset* existente y actualizar el *dataframe* asociado.
- Pruebas: evaluar la capacidad del modelo para generalizar a partir del dataset ampliado.
- Entrega: *dataset* ampliado con mayor diversidad en figuras, sombras y alturas.
- Retroalimentación y revisión: analizar el impacto de las nuevas imágenes en la precisión y generalización del modelo.

Iteración 8: optimización y ajuste fino

- Definición de requerimientos: refinar y optimizar el modelo para mejorar su rendimiento.
- Diseño: realizar ajustes en la arquitectura y hiperparámetros basándose en las pruebas de validación.
- Codificación: implementar los ajustes y continuar el entrenamiento.
- Pruebas: validar nuevamente con el conjunto de datos de validación.
- Entrega: un modelo más afinado y optimizado.
- Retroalimentación y revisión: evaluar la mejora en el rendimiento y hacer ajustes adicionales si es necesario.

Iteración 9: aumento de la robustez del modelo frente a variaciones de iluminación

- Definición de requerimientos: mejorar la robustez del modelo para que funcione de manera eficaz bajo diferentes escenarios de iluminación.
- Diseño: adaptar y optimizar la arquitectura del modelo para manejar variaciones en la iluminación.
- Codificación: aplicar ajustes y mejoras para aumentar la tolerancia del modelo a cambios en la iluminación.
- Pruebas: probar el modelo con un conjunto de datos que incluya variaciones significativas en la iluminación.
- Entrega: modelo optimizado para manejar diferentes condiciones de iluminación.

- Retroalimentación y revisión: evaluar la capacidad del modelo para generalizar en diferentes escenarios de iluminación.

Iteración 10: validación y pruebas de generalización

- Definición de requerimientos: asegurar que el modelo generalice bien a nuevas figuras y condiciones no vistas en la fase de entrenamiento.
- Diseño: establecer un conjunto de pruebas de validación riguroso que incluya figuras y condiciones de iluminación no presentes en el entrenamiento.
- Codificación: implementar pruebas de validación y ajustar el modelo según sea necesario.
- Pruebas: evaluar el modelo con el conjunto de datos de validación y realizar ajustes basados en los resultados.
- Entrega: un modelo validado y probado para generalización.
- Retroalimentación y revisión: analizar los resultados de las pruebas de generalización y hacer ajustes para mejorar la precisión y robustez del modelo

Iteración 11: mejora de la precisión y la robustez

- Definición de requerimientos: mejorar la precisión del modelo y su robustez frente a variaciones en las imágenes.
- Diseño: ajustar la arquitectura de la red neuronal para manejar mejor la diversidad y las complejidades del *dataset* ampliado.
- Codificación: implementar mejoras en la red y reentrenar con el *dataset* actualizado.
- Pruebas: realizar pruebas exhaustivas para evaluar la precisión y robustez mejoradas.
- Entrega: modelo mejorado con mayor precisión y robustez.
- Retroalimentación y revisión: recoger *feedback* sobre la eficacia de los ajustes realizados.

Iteración 12: integración y sincronización de datos de imágenes y sombras

- Definición de requerimientos: asegurar una correcta correlación entre imágenes de figuras y sus respectivas sombras.
- Diseño: desarrollar un sistema para vincular automáticamente cada imagen de figura con su sombra correspondiente basándose en la nomenclatura de los archivos.
- Codificación: implementar un algoritmo para asociar cada par de imágenes (figura y sombra) y extraer los datos relevantes del *dataframe*.
- Pruebas: verificar la exactitud de la vinculación y la consistencia de los datos extraídos.
- Entrega: un conjunto de datos perfectamente sincronizado y listo para su uso en el entrenamiento del modelo.
- Retroalimentación y revisión: evaluar la efectividad del sistema de integración y realizar ajustes si es necesario.

Iteración 13: mejora del modelo utilizando características adicionales

- Definición de requerimientos: incorporar características adicionales como el ángulo de iluminación y la longitud de la sombra en el modelo.
- Diseño: modificar la arquitectura del modelo para integrar estas nuevas características.
- Codificación: implementar los cambios en el modelo y ajustar el proceso de entrenamiento.
- Pruebas: evaluar el impacto de las nuevas características en la precisión del modelo.
- Entrega: un modelo mejorado que utiliza un conjunto más rico de características.
- Retroalimentación y revisión: analizar la efectividad de las características adicionales y hacer ajustes según sea necesario.

Iteración 14: análisis detallado de la relación entre sombras y alturas

- Definición de requerimientos: examinar cómo las diferentes longitudes y ángulos de sombras se correlacionan con las alturas de las figuras.
- Diseño: desarrollar un método de análisis para estudiar la relación entre las características de las sombras y las alturas de las figuras.
- Codificación: implementar análisis estadísticos y técnicas de aprendizaje automático para descubrir patrones y correlaciones.
- Pruebas: evaluar la fuerza y la significancia de las correlaciones encontradas.
- Entrega: un análisis detallado de la relación entre sombras y alturas.
- Retroalimentación y revisión: usar los hallazgos para mejorar la precisión del modelo en la estimación de alturas.

Iteración 15: evaluación rigurosa y preparación para el despliegue

- Definición de requerimientos: realizar una evaluación exhaustiva del modelo.
- Diseño: preparar criterios y métodos de evaluación, como métricas de rendimiento.
- Codificación: ejecutar pruebas con el conjunto de prueba para evaluar el rendimiento del modelo.
- Pruebas: comparar las predicciones del modelo con las alturas reales de las figuras.
- Entrega: un informe detallado de la evaluación del modelo.
- Retroalimentación y revisión: identificar áreas para mejoras futuras o ajustes.

Iteración 16: preparación del modelo para diferentes entornos de iluminación

- Definición de requerimientos: asegurar que el modelo funcione eficientemente bajo diferentes condiciones de iluminación.
- Diseño: desarrollar estrategias para adaptar el modelo a variaciones en la iluminación.

- Codificación: aplicar y probar ajustes que permitan al modelo manejar diversas condiciones de luz.
- Pruebas: probar el modelo en escenarios con variadas condiciones de iluminación.
- Entrega: un modelo robusto y adaptable a diferentes entornos de luz.
- Retroalimentación y revisión: evaluar la adaptabilidad del modelo a distintas condiciones de iluminación.

Los resultados finales de la aplicación de esta metodología se describen en las siguientes secciones.

En la figura 3.1 se observan tres figuras, la de una esfera, un cilindro y un cubo las cuales corresponde a fotografías que se tomaron inicialmente. En la parte superior de la imagen las tres figuras están sobre una superficie de madera, y en la parte inferior las tres figuras están sobre una superficie que es de color blanco mate. En las dos situaciones se proyectan las sombras de las figuras, solamente que, en una de estas superficies, la de color blanca mate la sombra está más definida.

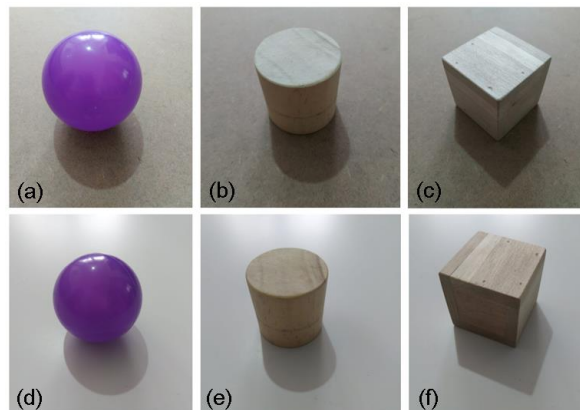


Figura No.3.1. Fotografías de dos superficies, una de madera y otra superficie de madera cubierta con vinilo de color blanco mate. (creación propia)

Otro aspecto muy importante que se analizó en este proceso es que la lámpara debe estar ubicada a diferentes alturas, lo cual permite variar el largo de las sombras, dado que estas se contraen o se elongan. En lo que respecta a la cámara fotográfica, la ubicación es fija con una inclinación de 45° . Posteriormente se tomaron fotografías con variaciones en las intensidades de la fuente de iluminación. Con estas variaciones la cantidad de imágenes en el *dataset*, se aumentó de forma progresiva, por tanto, desde el inicio de su creación, las imágenes se etiquetaron, se ubicaron en las carpetas respectivas a cada figura, para su posterior utilización en los dos modelos. Es muy importante señalar que en el momento de realizar las predicciones utilizando el modelo, estas pueden ser diferentes de la medida real, es decir, cada una de las figuras proyecta una sombra sobre la superficie, y ésta tiene un borde determinado, el cual es suave y no tan definido (es tenue), que divide el área iluminada del área oscura (sombra). Por tanto,

se sugiere, diversificar el conjunto de datos.

De igual forma con esta propuesta se obtuvo un gran avance respecto a la estructura del método planteado, como alternativa de solución. Resultado importante en el cual se encontró un método que puede realizar predicciones o estimaciones de la altura que a futuro puede ser escalable y optimizarse. Esta es una de las razones principales de que, al modelo en el conjunto de entrenamiento, se le enseñaran diferentes variaciones en el nivel de intensidad de la fuente de iluminación y alturas de la lámpara. De igual forma, las fotografías que componen el *dataset*, se les puede aplicar filtros para atenuar su aspecto e incluirlas dentro del conjunto de datos, con el objetivo de que la red aprenda que puede haber variaciones en la intensidad de la luz, en la altura de la lámpara y que estos afectan a la sombra proyectada. Esto es algo que las redes neuronales con el proceso de entrenamiento aprenden a inferir, por tanto, se debe considerar que, si se pretende mejorar el nivel de las predicciones de modelo, se debe proporcionar variedad en los elementos (imágenes) que conforman el *dataset*, para que el modelo posteriormente sea capaz de generalizar correctamente imágenes que nunca ha visto.

En el método propuesto en este documento, se tomaron fotografías con variaciones en las intensidades de la fuente iluminación y variando la altura de la lámpara, con el objetivo de mejorar las predicciones del modelo. En la tabla 3.1 se relacionan las diferentes variaciones para la toma de fotografías. Entre ellas están: la ruta de la figura, el nombre de la figura, la altura de la figura, la ruta de la sombra, el nombre de la sombra, el largo de la sombra, el ángulo fuente de iluminación, el nivel de intensidad fuente de iluminación lúmenes (*lumens*), la altura de la cámara, la altura de fuente de iluminación, el diámetro de la superficie circular (cms), la ubicación del objeto o forma (radio de la superficie en centímetros, el color del objeto (*RGB*), la rotación alrededor de la superficie circular (en grados), la textura de la superficie (madera lisa con vinilo de color blanco mate).

3.3. Construcción del *dataset*

Para realizar la construcción del *dataset* se realizó una serie de variaciones con el propósito de diversificarlo. Para ello, al momento de tomar todas las fotografías a cada una de las figuras de forma secuencial se siguieron los siguientes pasos, que son parte de la *iteración 1: construcción y preprocesamiento inicial del Dataset* y de la *iteración 2: proceso de captura de imágenes y preprocesamiento*:

- Diseño de la maqueta
- Selección de formas: cilindro, esfera y cubo
- Selección del tipo de lámpara (*led*)
- Construcción física de las formas (madera y plástico)
- Selección de alturas de las formas
- Selección de sombras de las formas (cilindro, cubo y esfera)
- Selección, altura y ángulo de la cámara fotográfica
- Toma de las fotografías
- Preprocesado fotografías:
 - Extracción de las figuras y sombras en archivos independientes
 - Reducción de la dimensión de las fotografías (800 x 600)
- Etiquetado de las fotografías
- Organización en carpetas de cada una de las imágenes

En la figura 3.3 se observa la imagen original de un cilindro con una altura de 7 centímetros, y su sombra proyectada que está ubicada a los 160° sobre la superficie. Ahí, también se observa en la misma figura en la sección (d), la ubicación tanto de la base de la cámara fotográfica, que siempre permanece fija a los 0° , como de la lámpara que rota o gira alrededor de la superficie circular. Estos giros por parte de la lámpara se hicieron para cada tipo de figura o forma, en diferentes tipos de intensidades de iluminación, en este caso con 55.0, 57.2, 64.0 lúmenes respectivamente, y de las alturas de la fuente iluminación, respecto de la superficie a 45 y 50 centímetros respectivamente.

La construcción del *dataset* se elaboró manualmente, ajustándolo a las necesidades que en su momento requirió la investigación en el diseño de la maqueta y del modelo. Es importante indicar que la elección de las figuras utilizadas (cilindro, el cubo y la esfera) para la conformación del *dataset*, se seleccionaron principalmente porque, éstas tienen las características ideales y básicas, que combinadas pueden facilitar la representación de otros objetos tridimensionales más complejos, como un rostro (*ArtRocket*, 2021).

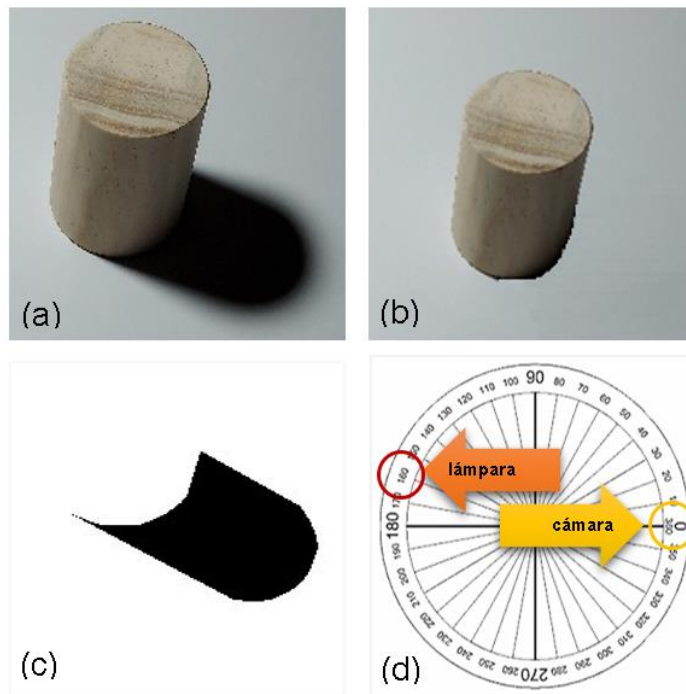


Figura No.3.3. Margen de error en al área segmentación de la imagen de la sombra en el preprocesamiento (creación propia).

También es importante indicar que al modelo se le proporcionó, un entendimiento gradual de la complejidad espacial, es decir, el cilindro introduce información del radio y de la altura, la esfera la noción de radio y centro, y el cubo de aristas y vértices. Un enfoque que permitió al modelo aprender gradualmente características espaciales esenciales de los objetos. Por otro parte, la detección de posibles errores del modelo, con la interpretación de resultados, permitió realizar los ajustes o configuraciones necesarias para mejorar la calidad de la predicción y ajuste.

Otra razón muy importante es la escalabilidad del modelo, ya que, al iniciar el entrenamiento con formas básicas o simples, se puede escalar gradualmente la complejidad del modelo a medida que se introducen formas irregulares. Esto es muy importante porque evita problemas de convergencia, y mejora la capacidad del modelo para adaptarse a formas o figuras que tienen diferentes niveles de complejidad.

En resumen, la elección del cilindro, la esfera y el cubo como figuras principales del conjunto de datos, se fundamentó en que sus características proporcionaron un marco de aprendizaje sólido y eficiente para el modelo, que a su vez son un cambio gradual y evolutivo en la comprensión de formas y objetos más complejos.

La figura 3.4 muestra la sombra y el *dataframe* con el cual se relaciona la información de cada una de las imágenes que componen el *dataset*.

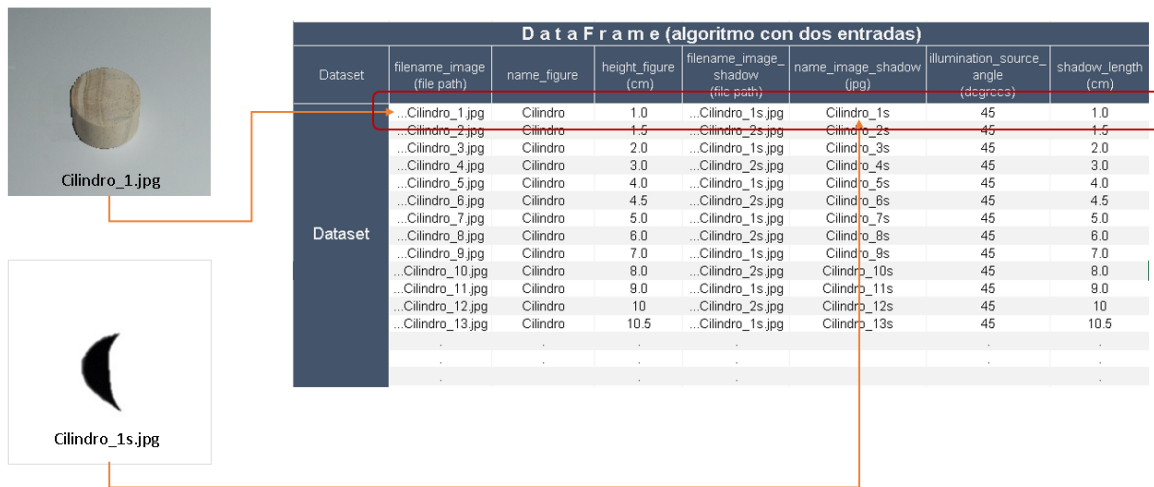


Figura No.3.4. Imagen de la figura, sombra y el *dataframe* (creación propia).

Por otra parte, el ambiente se controló con el propósito de adaptar las imágenes a los requerimientos iniciales del modelo, se utilizó una lámpara led dimerizable que ofreció la posibilidad de regular la intensidad de luz en las fotografías, una cámara fotográfica para capturar las imágenes correspondientes, una superficie de madera cubierta con un vinilo blanco de color mate que tiene una forma circular, que permitió ubicar y rotar una lámpara por medio de un trípode alrededor de la superficie circular.

Dado que la finalidad del modelo fue tomar las fotografías de cada forma, desde diferentes ubicaciones y en condiciones variables de iluminación, se tomaron las mismas a cada una de las formas (esfera, cilindro, esfera), entendiéndose así, de que cada fotografía tuvo solamente una figura, razón por la cual el modelo fue más eficiente al detectar errores y posteriores ajustes.

Ahora bien, se tomaron cada una de las fotografías a cada figura, teniendo en cuenta que solamente se ubicó una de éstas en la maqueta, y se fue rotando la lámpara alrededor de la superficie circular de madera, teniendo presente que ésta se ubicó inicialmente a 4 cms y con un ángulo de 45°. En cada giro de la lámpara se capturaron las fotografías de las figuras, y posteriormente se realizó otro dando un giro con la lámpara ubicada a 50° respecto de la superficie e igualmente variando la intensidad de la lámpara. En la figura 3.5 se indican los datos obtenidos por la aplicación móvil Fotómetro pro, para medir, algunos de ellos son: la iluminancia y brillo, en la cual se encuentra una unidad de medida que es el Lux es una medida de iluminancia que indica la cantidad de luz incidente en una superficie. Cuanto mayor sea el valor de lux, más brillante será la luz en el área medida. La potencia de la bombilla de (10 w). La temperatura de Color (6500K), la cual indica que la luz tiene un tono frío y se asemeja a la luz el día. Este tipo de luz es comúnmente utilizado en entornos de trabajo y tareas que requieren una buena visibilidad. Y es dimerizable, significa que se puede ajustar la intensidad de la luz (ver figura 3.5).



Figura No 3.5. Medida de lúmenes en tres niveles de intensidad de la lámpara led

Otro aspecto muy importante es que el modelo desarrollado en esta investigación se está proyectando para realizar una tarea más compleja, dado que se van a utilizar formas irregulares y con más variables, como, por ejemplo, sombras solapadas sobre otra figura o sobre una superficie vertical, teniendo en cuenta otros criterios al momento que el modelo evalúa las formas, como, por ejemplo, sombras solapadas, figuras superpuestas, varias formas en una sola imagen entre otras. En la figura 3.6 Se muestran dos fotografías en las cuales las sombras se encuentran solapadas, que son una de las muchas variaciones que puede tener el modelo, pero que en esta propuesta no se incluye. Por tanto, en esta investigación se desarrolló un modelo No.1. con una entrada (la sombra), y un modelo No.2. con dos entradas (la figura y la sombra), como ya se ha indicado anteriormente con relación al problema planteado, y que mediante la experimentación constante con ajustes posteriores se obtuvo una predicción de la altura de una figura dentro de una imagen 2D a partir de su sombra.

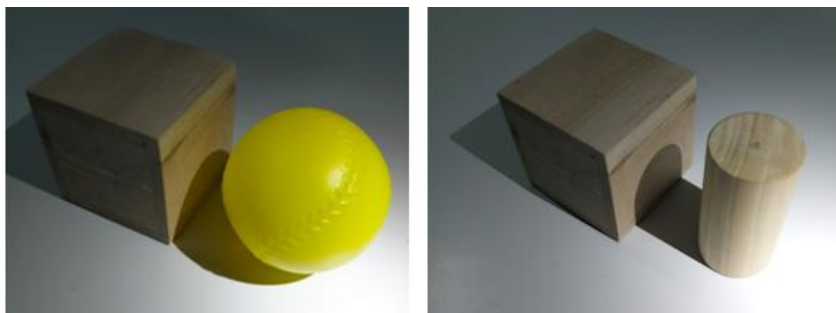


Figura No.3.6. Sombras solapadas (creación propia).

En la figura 3.7. se observan varias fotografías las cuales se tomaron con una cámara ubicada y la fuente de iluminación ubicada a 45 centímetros de la superficie con una inclinación de 45°, lo cual genera una longitud determinada de la sombra de cuerdo a cada figura, lo que permitió establecerle al modelo que en determinadas condiciones de iluminación la imagen de la sombra varia de longitud, dependiendo de la ubicación de la fuente de iluminación.

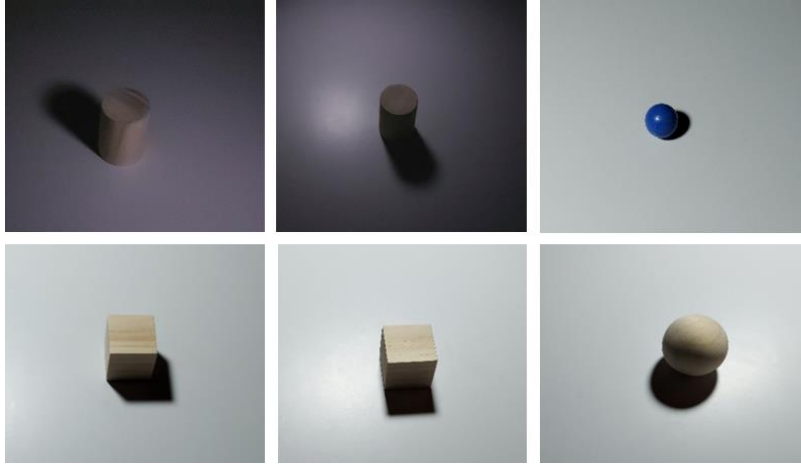


Figura No.3.7. Imagen de un cilindro con sombras diferentes (creación propia).

De esta forma se construyó el conjunto de datos con imágenes tomadas desde varios ángulos (45° y 50°), se sometió a pruebas al modelo, se obtuvieron resultados muy interesantes los cuales se detallarán más adelante en este capítulo. En el modelo las fotografías que se utilizaron para el *dataset* incluyen solamente están desde los 45 centímetros inicialmente, posteriormente se utilizarán las que se tomaron desde 50 centímetros de la superficie. El conjunto de datos se estructuró, para facilitar el proceso de aprendizaje del modelo (en el entrenamiento). Para esto, fue fundamental que las entradas en este caso imágenes de figuras, de sombras, de las salidas (valores continuos) estuvieran organizadas y hubieran sido preprocesadas adecuadamente. En este caso, se utilizó un archivo (*DataFrame.csv*) para almacenar las etiquetas (valores continuos) con los valores correspondientes a cada imagen y cada sombra respectivamente. Finalmente, los datos obtenidos en la salida son valores numéricos, ya que se trata de un problema de regresión, donde la salida para cada imagen es un valor numérico, o un vector de valores numéricos (alturas) ver tabla 3.2.

Figura / unidades	Medida en unidades. 1 unidad = 1 centímetro																	
	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	7	8	9	9,5	10	11	Total
Cubo	X	X	X		X		X	X	X	X	X	X	X	X		X		1.500
Cilindro	X	X	X		X		X	X	X		X	X	X	X		X	X	1.500
Esfera	X	X	X	X	X	X	X	X	X	X			X		X			1.500

Tabla No. 3.2. Alturas de las figuras y medidas en centímetros (*creación propia*)

Para entender el ambiente en el cual se tomaron las fotografías se puede observar la figura 3.8. en la cual se detallan la ubicación de la cámara fotográfica, la cual es graduable en ángulo y altura respecto de la superficie, la fuente de iluminación (lámpara led) dimerizable. Una superficie circular de color blanco donde en su centro se ubica cada figura, una lámpara la cual se rota, y en cada ubicación de esta se tomó una fotografía,

lo que permitió obtener figuras y sombras de cada una de estas figuras. Es importante indicar que la cámara permaneció siempre fija a 45° respecto del centro de la superficie, donde se encuentra la figura. Esta superficie tiene un diámetro de 90 centímetros y su centro (radio) está ubicado a 45 centímetros. En lo relacionado a la fuente de iluminación (de 840 lúmenes), está siempre dirige su luz hacia al centro de la superficie circular. En este punto es importante indicar que, en el modelo propuesto que se observa en la figura 3.8 se utilizó la geometría de las formas tridimensionales de los objetos o formas uniformes, combinado con la sombra correspondiente desde distintas ubicaciones en el plano y generados por una única fuente de iluminación, para ofrecer diversidad de datos. Aquí la fuente de iluminación se ubicó a 90 centímetros, es decir a un extremo de la superficie pero apuntando al centro de ésta, donde las alturas de la lámpara son de 45 centímetros y 50 centímetros respecto de la superficie, es decir su variación es de 5 centímetros inicialmente pero es posible aumentar el margen de diferencia, ya que esto contrae o elonga la sombra, y esta variabilidad el conocimiento que se le proporciona al modelo, y de que las diferentes sombras sean observables en la superficie bajo variadas intensidades de iluminación.



Figura 3.8. Guía general de formación de sombras de una figura al rotar. (creación propia).

La lámpara rota o gira alrededor de la superficie circular y de la figura (cilindro, esfera y cubo) para obtener sombras desde diferentes ángulos. Algunos de los aspectos que se tienen en cuenta al momento de elaborar la estructura, son las propiedades de reflectancia de la superficie de la sombra, e igualmente la longitud de la sombra dependiendo de la altura de la lámpara respecto de la superficie de madera. Una de las principales razones para la selección de la bombilla led, es que ésta dirige y concentra su capacidad lumínica en un área de interés específica dentro de la escena. Las características de esta bombilla la hacen ideal para esta tarea, donde los bordes de las

sombras proyectadas deben estar bien definidos en la superficie de la maqueta. Esta bombilla se probó en la maqueta en todas las figuras, proyectando sus sombras, con el objetivo de identificar que efectivamente los bordes estuviesen definidos, lo que se traduce en que, el límite o frontera entre el área iluminada y el área oscura (sombra) fueran propicios para el proceso de experimentación inicial, ver figura 3.10.

La figura 3.9 muestra las fotografías del cilindro con todas las alturas, que se utilizaron en el entrenamiento de los dos modelos.

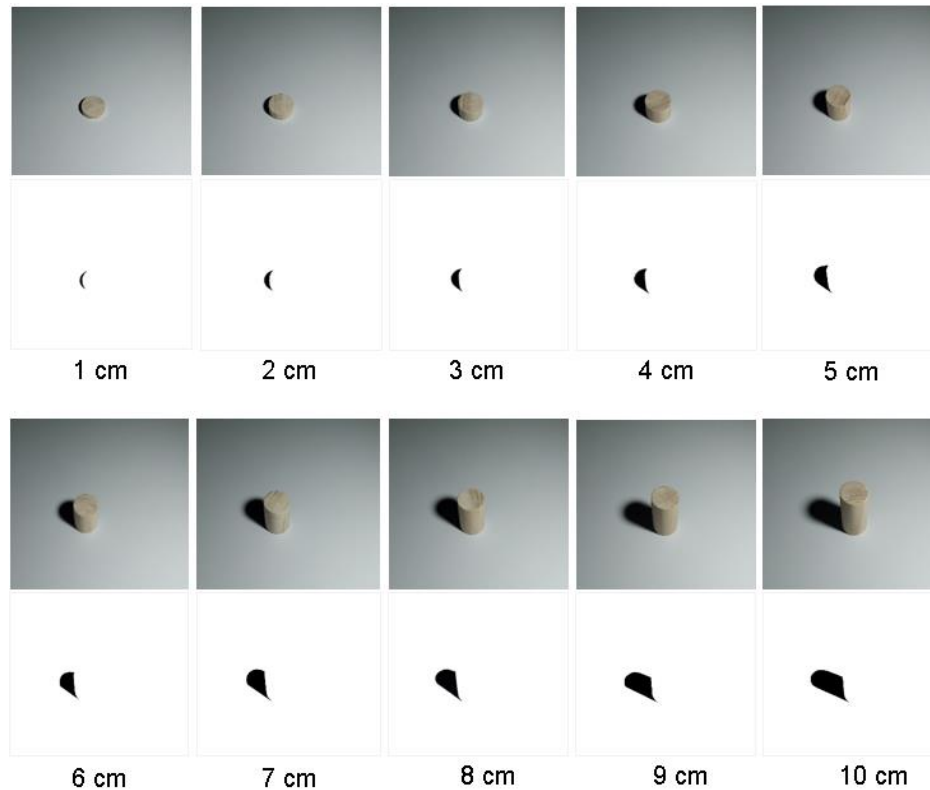


Figura No.3.9. Altura de todos los cilindros (creación propia)

Con el ambiente listo se tomaron las primeras fotografías, las cuales se utilizaron en los dos modelos respectivamente, después de ser preprocesadas de forma manual (separar en la fotografía de la figura y de la sombra), con el objetivo de realizar las pruebas iniciales y los ajustes que se consideraron pertinentes, se determinó que, para proporcionarle un mejor conocimiento a la red en el proceso de entrenamiento, validación y prueba, en relación a la calidad de las imágenes tanto de las formas y como de su sombra respectiva, la superficie circular debería tener una textura lisa y uniforme posible, para que al capturar las fotografías, estas tuviesen la calidad ideal, por tanto la superficie que inicialmente era de madera se cubrió con un vinilo adhesivo de color blanco mate.

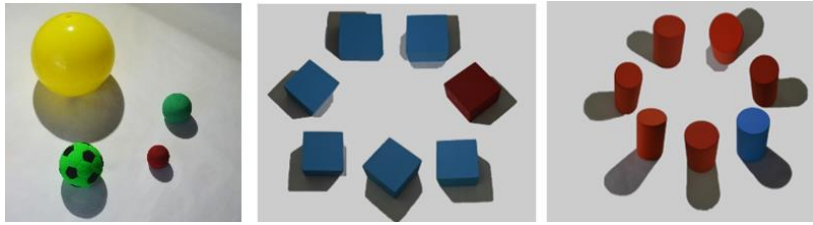


Figura 3.10. Fotografías de sombras de figuras obtenidas al rotar la fuente de iluminación.
(creación propia)

El proceso de experimentación se realiza en condiciones controladas tanto en la iluminación, como en las formas y en la superficie. Tales condiciones se aplicaron sobre el modelo permitiendo de forma inicial analizar y probar el modelo, razón por la cual se optó por utilizar las fotografías de tres figuras geométricas con sus sombras correspondientes, ubicadas sobre una superficie uniforme (plana). En la figura 3.11 se observan los diferentes niveles de iluminación que se utilizaron en la construcción del con el propósito de indicarle al modelo la información etiquetada y parámetros asociados a cada una de las imágenes, es así que, las fotografías corresponden a las figuras anteriormente indicadas, teniendo en cuenta que para cada una de éstas formas, se giró alrededor de la superficie circular y se tomaron las fotografías correspondientes a un determinado nivel de intensidad de la fuente de iluminación.

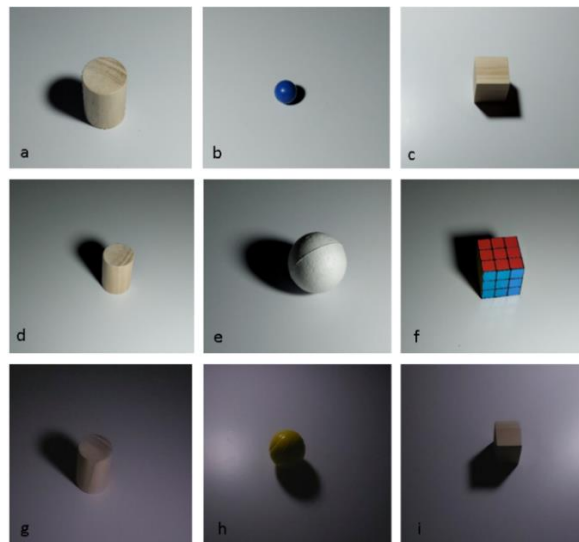


Figura No. 3.11. Imágenes fotografías con un nivel de iluminación Imágenes fotografías con niveles de iluminación e imágenes

Aquí la fuente de iluminación es una lámpara led dimerizable se realizó la variación en los Lúmenes, la cual se rota alrededor de la figura permaneciendo a una altura y a un ángulo definidos, con el objetivo de capturar las fotografías bajo diferentes niveles de iluminación (*lumens*), diferentes tipos de figuras y de sombras. Los lúmenes en los tres niveles de iluminación de las fotografías se calcularon con una aplicación móvil: Fotómetro Pro y Lux con al cual se obtuvieron los siguientes valores aproximados para

ver más detalles ver el Anexo F.

A continuación, se muestran algunas fotografías de las formas que contiene el *dataset* y que están diferenciadas por su altura y forma, donde la altura está dada en centímetros (unidades). En la figura 3.12 se observa la altura de un cilindro, de una esfera y de un cubo de 14, 9.5 y 6.5 unidades (centímetros) correspondientemente.

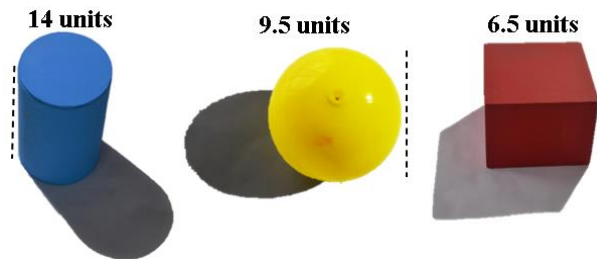


Figura 3.12. Altura de un cilindro, esfera y cubo y sus respectivas sombras (*dataset*).
(creación propia)

Los diferentes tipos de sombras son las que se generan a partir de la ubicación de la lámpara en cada figura, y se pueden observar en la figura 3.13 que estas tienen diferentes longitudes: En este caso de un cilindro cada figura o forma, la cual se elonga o se contrae dependiendo de la altura de lámpara o de la altura de la figura.



Figura 3.13. Longitud de la sombra de un cilindro (creación propia)

Proyección de la sombra (Shadow)

Cuando un objeto bloquea la fuente de iluminación, proyecta la sombra en la dirección opuesta a dicha fuente de luz. Esta sombra puede ser representada en una imagen *2D*, lo que permite relacionar la fuente de iluminación, el objeto y la superficie.

Algunos de los conceptos y características de las sombras se indican a continuación:

- La *formación de la sombra* se presenta cuando la luz recae sobre un objeto o forma, y es aquí donde este puede bloquear parcial o completamente la fuente de iluminación, creando una región en la dirección opuesta de la fuente de iluminación.
- *La longitud y forma de la sombra*: se refiere sus áreas pueden variar (deformarse) *Por ejemplo* si la fuente de luz está cerca del objeto, la sombra será más larga y estirada. Si la fuente de luz está lejos, la sombra será más corta y compacta.

3.3.1. Descripción del conjunto de Datos (*Dataset*)

Para estos modelos, es crucial contar con un conjunto de datos adecuadamente estructurado y caracterizado. A continuación, se presentan las características que debe tener dicho conjunto de datos:

a. **La diversidad:**

Hace referencia a que el conjunto debe ser representativo de todas las posibles variaciones y escenarios, en este caso de las figuras, sombras y alturas. El conjunto de datos se realizó representativo del espacio completo de las condiciones bajo las cuales el modelo podría ser eventualmente desplegado. Esto involucra diferentes valores de alturas, variaciones en la iluminación, orientación de las formas geométricas, y posiblemente textura, entre otros, para asegurar que el modelo tenga la capacidad de generalizar bien a datos no vistos previamente.

En esta investigación se desarrolló un *Dataset* conformado por un conjunto de formas geométricas, sombras y alturas que tiene tres componentes asociados: una imagen de una figura, una imagen de la sombra proyectada por la figura, y la altura real de la figura. Las figuras, son imágenes o representaciones visuales de objetos o formas geométricas tridimensionales con formaciones regulares. Las sombras son igualmente imágenes de las sombras proyectadas por estas figuras a partir de una fuente de iluminación. Por tanto, cada figura proyectará una sombra única dependiendo de su forma, tamaño y orientación, así como de la dirección y calidad de la luz en el entorno. De otro lado está la altura, la cual es un valor numérico que representa la altura real de la figura en la imagen, donde este sería el valor objetivo que el modelo intentaría predecir.

La fotografía con las imágenes de todas las figuras geométricas utilizadas sonde tres tipos: el cilindro, el cubo y la esfera con las figuras, sombras y las alturas definidas (ver figura 3.14)

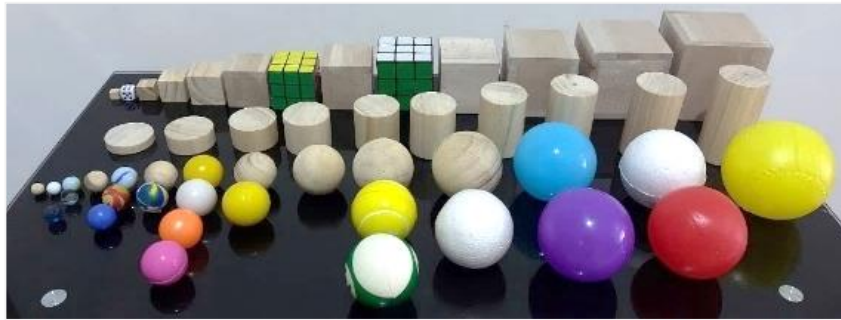


Figura 3.14. Vista general de figuras usadas para la generación del *dataset*. (creación propia)

La diversidad en las alturas de las formas geométricas seleccionadas implicó la utilización de diferentes escenarios, tamaños de formas, diferentes tipos de iluminación, teniendo en cuenta las medidas de cada una de las figuras geométricas, las cuales se describen en la tabla 3.3 (Distribución de altura de las formas geométricas del *dataset*).

Figura	Color	Altura (cm)	Cantidad
Cubo	Marrón	1	1
Cubo	Marrón	1,5	1
Cubo	Marrón	2	1
Cubo	Marrón	3	1
Cubo	Marrón	4	1
Cubo	Marrón	5	1
Cubo	Rojo, blanco, amarillo, verde, azul, morado	5,5	1
Cubo	Marrón	6	1
Cubo	Rojo, blanco, amarillo, verde, azul, morado	6,5	1
Cubo	Marrón	7	1
Cubo	Marrón	8	1
Cubo	Marrón	9	1
Cubo	Marrón	10	1
Total Cubos			13

Figura	Color	Altura (cm)	Cantidad
Esfera	Marrón	1	1
Esfera	Blanco	1,5	1
Esfera	Blanco	2	3
Esfera	Marrón	2,5	1
Esfera	Blanco, Amarillo, Azul	3	3
Esfera	Marrón, Azul con amarillo	3,5	2
Esfera	Amarillo, Blanco, Naranja, Rosado	4	4
Esfera	Marrón, Amarillo	4,5	2
Esfera	Marrón	5	1
Esfera	Marrón, Amarillo, Verde, Blanco	6	3
Esfera	Marrón, Blanco	7	2
Esfera	Azul, Violeta	8	2
Esfera	Blanco, Rojo	8,5	2
Esfera	Amarillo	9	1
Total Esfera			28

Figura	Color	Altura (cm)	Cantidad
Cilindro	Marrón	1	1
Cilindro	Marrón	2	1
Cilindro	Marrón	3	1
Cilindro	Marrón	4	1
Cilindro	Marrón	5	1
Cilindro	Marrón	6	1
Cilindro	Marrón	7	1
Cilindro	Marrón	8	1
Cilindro	Marrón	9	1
Cilindro	Marrón	10	1
Cilindro	Marrón	10,5	1
Total Cilindro			11

Tabla No.3.3. Distribución de alturas de las formas geométricas del *dataset*. (creación propia)

El desbalanceo inicial de las tres figuras (esfera, cilindro y cubo) seleccionadas para el proceso de entrenamiento, se manejó de forma prudente y transitoria teniendo en cuenta que en su momento se requería construir un modelo base para la predicción de la altura de dichas formas y que este fuera funcional, las que posteriormente se balancearon con el aumento de datos (*ImageDataGenerator*) y la regularización $L1(Lasso)$ y $L2(Ridge)$.

b. Calidad y Resolución

El conjunto de datos incluye fotografías de 800×600 a color (*RGB*) de varias figuras geométricas de varios colores, tomadas desde una distancia de 90 cms del borde la superficie. La superficie donde está ubicada la forma u objeto es blanca, lisa la cual es de color blanco mate. La superficie tiene un tamaño de 90 cms diseñada de forma circular para facilitar el desplazamiento de la lámpara led, en madera *MDF (Medium Density Board)* de 9 mm . Las características de cada una de las imágenes (fotografías) son:

El formato es *JPG (RGB: Joint Photographic Experts Group)*.

- Dimensiones de la imagen: 800 (Ancho) x 600 (Alto)
- Profundidad en pixeles 24 bits, cada canal es una imagen de 8 bits
- El número total de colores es de 24 bits es de $(2^8)^3 = 16,777,216$

Todas las imágenes se ajustaron al tamaño definido y se organizaron según su tipo de forma y medida en altura. En la figura 3.15 se observan las fotografías de dos figuras y su correspondiente sombra que hace parte del *dataset*, las cuales se preprocesaron antes de realizar el proceso de entrenamiento con el modelo. De esta forma en los dos modelos, el modelo 1 (una entrada: la sombra) y el modelo 2 (dos entradas: figura y sombra) se utilizaron en el proceso de experimentación.

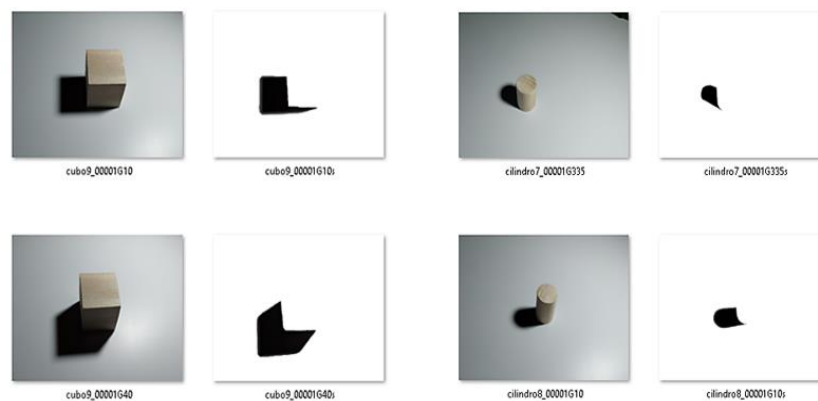


Figura No. 3.15. Izquierda: Fotografías del cubo con sombras respectivas. Derecha: fotografías del cilindro con sombras respectivas

3.3.2. Estructura del conjunto de datos (*Dataset*)

Se indica a continuación el proceso para la construcción del conjunto de datos, conformado por diferentes fotografías con las sombras y las alturas. Teniendo en cuenta que, en las fotos solo habrá una única figura (ver figura 3.15).

Procedimiento para la toma de las fotografías:

La toma de las fotografías para cada una de las figuras geométricas se realiza ubicando la cámara fotográfica en un lugar fijo, estando ésta ubicada a 45° grados respecto de la imagen y para la captura de las imágenes se rotó la fuente de iluminación para obtener sombras desde diferentes ubicaciones en el sentido de las manecillas del reloj y por último se graduó la intensidad de la iluminación en cada una de las figuras, dado que la fuente de iluminación es dimerizable, (ver figura 3.16) Posteriormente se modificó la altura de la lámpara led, quedando está a 50 cms de la superficie, obteniendo así fotografías de sombras con características diferentes (el área y la longitud de la sombra). Las rotaciones se realizaron a medida que cambio algo dentro de la maqueta, ya estas se pueden realizar cada vez que se modifica algo, lo que permite que el modelo aprende de estos cambios para generalizar mejores datos nunca vistos. Esto hace también que el modelo sea escalable y de forma proporcional mejore su nivel de predicción, en este caso de la predicción de la altura.

En la figura 3.16 y figura 3.17 se observa la forma en que se realizó la rotación de la lámpara, y de todas las ubicaciones que se utilizaron, especificadas en grados, para la toma de las fotografías a cada una de las figuras.

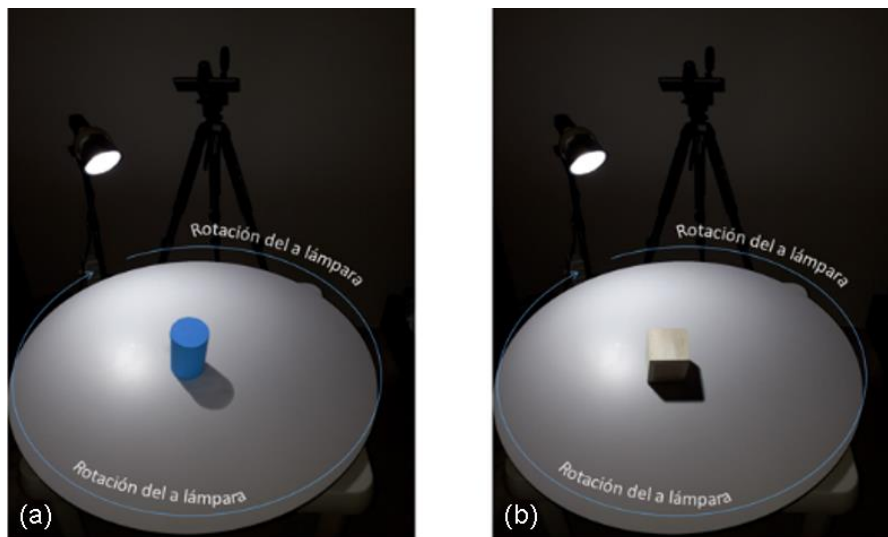


Figura 3.16. (a), (b) cilindro y esfera con la primera ubicación de la fuente de iluminación (lámpara).

Para la organización del *dataset* estas se agruparon según las medidas de las alturas y su forma geométrica. Las medidas se tomaron en centímetros que posteriormente se manejan en unidades, con el propósito de facilitar la reutilización del conjunto de datos en cualquier otro sistema de medición: 1 unidad igual a un 1 centímetro, 2 unidades igual a 2 centímetros, 3 unidades igual a 3 centímetros, 4 unidades igual a 4 centímetros, 5 unidades igual a 5 centímetros, etc.

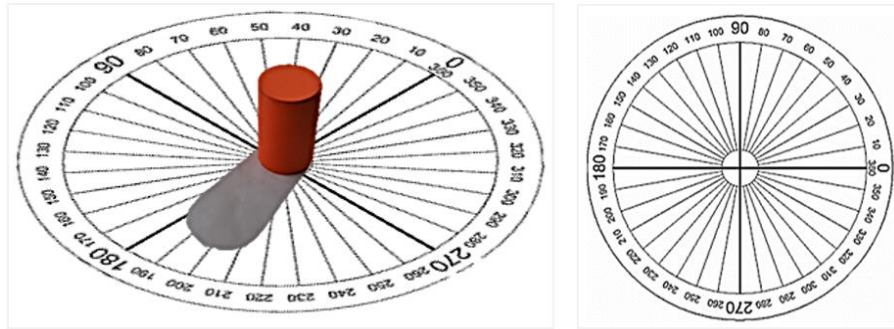


Figura	Altura (cm)	Ubicación en grados de la fuente de iluminación																				
cilindro 1	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°	
cilindro 2	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°	
cilindro 3	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°	
cilindro 4	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°	
cilindro 5	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°	
cilindro 6	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°	
cilindro 7	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°	
cilindro 8	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°	
cilindro 9	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°	
cilindro 10	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°	
cilindro	10.5	20°	40°	50°	90°	100°	110°	120°	130°	140°	150°	170°	180°	190°	215°	225°	235°	250°	300°	325°	335°	355°

Figura No.3.17. Rotación especificada en grados de la lámpara alrededor de la superficie circular.

En la tabla 3.3 La cantidad de figuras de formas geométricas y de sombras del *DataFrame* es de: 2.244, las cuales se utilizaron en los dos modelos desarrollados, el modelo No.1. de uno entrada y el modelo No.2. de dos entradas, desarrollado en esta investigación.

Conjunto de datos	Figuras	Entrenamiento	Validación	Prueba	Subtotal imágenes
		80%	10%	10%	
Dataset No.1	Esferas	Esferas y sombras		664	1.328
		Sombras		664	
	Cilindros	Cilindros y sombras		190	380
		Sombras		190	
	Cubos	Cubos y sombras		268	536
		Sombras		268	
Total imágenes (100%)			2.244		2.244

Tabla No.3.3. Distribución de *dataset* 1. (creación propia)

Estas posteriormente se distribuyen en el conjunto de entrenamiento el cual tiene 673 imágenes y 673 sombras, el conjunto de validación que tiene 225 imágenes y 225 sombras y finalmente el conjunto de prueba tiene 225 imágenes y 225 sombras. De otra parte, es importante indicar que se realizó una prueba en los modelos utilizando un *dataset* de un total de 1.140 imágenes entre figuras geométricas y sombras, correspondiente a cada una de ellas, el cual está balanceado en las cantidades de imágenes respectivas

para los conjuntos de entrenamiento, de validación y de prueba.

Preparación de los datos

Los datos se organizaron en carpetas separadas para el conjunto de entrenamiento, validación y prueba. Contienen la información de esferas, cubos y cilindros. En detalle la estructura de las carpetas se puede apreciar en la figura 3.18, donde la carpeta principal del *dataset* tiene tres carpetas: una para los cilindros, otra para los cubos y el otro para las esferas. Dentro de cada subcarpeta de las figuras, hay dos subcarpetas: una para las imágenes de las figuras y otra para las imágenes de sus sombras. Los nombres de la figura y de su sombra respectiva se lo determinó así, para dejar alguna similitud entre ellos, y poder realizar gestiones no solo a nivel de archivos sino también del *dataframe* (ver figura 3.18).

Nombre de la figura	Nombre de la sombra
cilindro10_00001G170.jpg	cilindro10_00001G170s.jpg

	
--	---

Figura 3.18. Nombres de las imágenes de las figuras y de las sombras. (creación propia)

La tabla 3.19 es un ejemplo de algunas de las imágenes del *DataFrame* muestra cómo se estructuró el dataset y sus carpetas, ver figura 3.19, de tal forma que además de contener las imágenes correspondientes a cada carpeta, cuenta con el ángulo de fuente de iluminación y el largo de la sombra. La estructura del *DataFrame* tiene características las siguientes características: nombre de la imagen (*filename_image*), nombre de la figura (*name_Figura*), altura de la figura (*height_Figura*) nombre de la sombra(*filename_image_shadow*), nombre de la imagen de sombra (*name_image_shadow*), longitud de la sombra (*shadow_length*), ángulo de iluminación (*illumination_source_angle*).



Figura 3.19. Estructura del *dataset* de entrenamiento, validación y prueba. (creación propia)

En la tabla 3.4 cada fila del archivo representa una imagen del dataset nombre de la imagen (*filename_image*), nombre de la figura (*name_Figure*), altura de la figura (*height_Figure*) nombre de la sombra (*filename_image_shadow*), nombre de la imagen de sombra (*name_image_shadow*), longitud de la sombra (*shadow_length*), ángulo de iluminación (*illumination_source_angle*).

Nombre del archivo imagen(jpg)	Nombre de la imagen	Altura de la figura	Nombre del archivo sombra (jpg)	Nombre de la sombra	Longitud de la sombra	Ángulo de la fuente de iluminación
cilindro1_00001G170	cilindro	1.0	cilindro1_00001G170s	sombra_cilindro	1.0	45
cilindro1_00001G180	cilindro	1.0	cilindro1_00001G180s	sombra_cilindro	1.0	45
cilindro1_00001G335	cilindro	1.0	cilindro1_00001G335s	sombra_cilindro	1.0	45
cilindro3_00001G180	cilindro	3.0	cilindro3_00001G180s	sombra_cilindro	3.0	45
cubo7_00002G335	cubo	7.0	cubo7_00002G335s	sombra_cubo	7.0	45
cubo8_00001G80	cubo	8.0	cubo8_00001G80s	sombra_cubo	8.0	45
cubo9_00001G335	cubo	9.0	cubo9_00001G335s	sombra_cubo	9.0	45
esfera1.5_00001G100	esfera	1.5	esfera1.5_00001G100s	sombra_esfera	1.5	45
esfera1_00001G190	esfera	1.0	esfera1_00001G190s	sombra_esfera	1.0	45
esfera2.5_00002G10	esfera	2.5	esfera2.5_00002G10s	sombra_esfera	2.5	45
.
.
.

Tabla No.3.4. Estructura del *DataFrame*. (creación propia)

El *DataFrame* se define con el propósito de optimizar la precisión del modelo, quedando este estructurado de la siguiente forma:

- *Nombre del archivo imagen (jpg)*: es el nombre del archivo de la imagen original de la figura 2D. Este campo almacena la ruta o nombre de archivo de la imagen que representa la figura de la que se está estimando la altura.

- *Nombre de la imagen*: representa el nombre o etiqueta de la figura 2D en cuestión. Es una descripción del tipo de figura que se encuentra en la imagen, como "cilindro", "cuadrado", "triángulo", etc.
- *Altura de la figura*: es la altura real de la figura 2D en unidades específicas (por ejemplo, centímetros o metros). Este valor es el dato objetivo que se intenta estimar utilizando la red neuronal a partir de la sombra de la figura.
- *Nombre del archivo sombra (jpg)*: corresponde al nombre del archivo de la imagen de la sombra de la figura. Similar a `filename_image`, este campo almacena la ruta o nombre de archivo de la imagen de la sombra de la figura.
- *Nombre de la sombra*: es el nombre o etiqueta de la imagen de la sombra. al igual que `name_figura`, esta columna proporciona información sobre el tipo de sombra que se está utilizando en el análisis.
- *Longitud de la sombra*: representa la longitud de la sombra proyectada por la figura en la imagen de la sombra. Este valor está definido en unidades, como centímetros o metros.
- *Angulo de la fuente de iluminación*: es el ángulo de la fuente de iluminación en relación con la figura y la sombra, ya que la dirección y el ángulo de la luz pueden afectar la forma y la longitud de la sombra, lo que a su vez influye en la estimación de la altura de la figura.

Con la construcción de este *dataset* se da cumplimiento a las iteraciones 1, 2 y 8.

3.4. Arquitectura

Se desarrollaron dos modelos: el modelo No.1. con una entrada, la sombra y un modelo No.2. con dos entradas, la figura y la sombra. Para este modelo No.1. se utilizó inicialmente una arquitectura de una *CNN*, con las capas convolucionales que se encargaron de extraer características relevantes de las sombras, y de las imágenes. Estas características pasaron por las capas de agrupación para reducir la dimensionalidad y, finalmente, se aplanaron para producir una estimación escalar de la altura de la figura 2D correspondiente a la sombra.

Con un total de 76,993 parámetros, este modelo es relativamente ligero y eficiente para la tarea de regresión propuesta. Por tanto, el modelo presentado ofrece una estructura inicial para la estimación de la altura de figuras 2D a partir de sombras en imágenes. El diseño convolucional lo hace apto para aprender patrones complejos en los datos visuales, mientras que la función de activación lineal en la capa de salida permite realizar tareas de regresión.

El modelo No.2. por su parte procesa dos imágenes relacionadas de forma independiente y posteriormente concatena sus características para realizar una

predicción. También utiliza una arquitectura de una *CNN* con regresión. Este modelo al combinar las dos fuentes de información corrige ambigüedades que pueden presentarse al analizar imágenes por separado.

A continuación, se expone una descripción más detallada de los dos modelos:

3.4.1. Arquitectura del modelo No.2 (dos entradas):

En la figura 3.20 se observa la arquitectura de este modelo No.2., la cual se diseñó para la extracción de características tanto de las imágenes de las figuras, como de sus sombras respectivas. Estas se combinaron para realizar una predicción de la altura de las figuras. Se diseñó teniendo en cuenta la extracción de características significativas de dos fuentes distintas de entrada, procesaron de manera paralela y luego se combinaron esta información para generar la predicción o estimación de la altura.

```
Model: "model"
```

Layer (type)	Output Shape	Param #	Connected to
input_figure (InputLayer)	[(None, 128, 128, 3)]	0	
input_shadow (InputLayer)	[(None, 128, 128, 3)]	0	
conv2d (Conv2D)	(None, 126, 126, 32)	896	input_figure[0][0]
conv2d_2 (Conv2D)	(None, 126, 126, 32)	896	input_shadow[0][0]
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0	conv2d[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 63, 63, 32)	0	conv2d_2[0][0]
conv2d_1 (Conv2D)	(None, 61, 61, 64)	18496	max_pooling2d[0][0]
conv2d_3 (Conv2D)	(None, 61, 61, 64)	18496	max_pooling2d_2[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 64)	0	conv2d_1[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 30, 30, 64)	0	conv2d_3[0][0]
flatten (Flatten)	(None, 57600)	0	max_pooling2d_1[0][0]
flatten_1 (Flatten)	(None, 57600)	0	max_pooling2d_3[0][0]
concatenate (Concatenate)	(None, 115200)	0	flatten[0][0] flatten_1[0][0]
output_height (Dense)	(None, 1)	115201	concatenate[0][0]

Total params: 153,985
 Trainable params: 153,985
 Non-trainable params: 0

Figura 3.20. Descripción arquitectura del modelo No.2.

- **Entrada:** este modelo es un poco más complejo ya que utilizó dos imágenes como entrada: la de la figura y su sombra correspondiente. Las columnas *filename_image* y *filename_image_shadow* del *DataFrame* proporcionaron las rutas para cargar estas imágenes.
- **Salida:** al igual que el modelo No.1. la salida es la altura de la figura, tomada de *height_Figura*.

Como el modelo No.2. presenta una arquitectura robusta y diseñada con dos entradas, esta le permitió al modelo tener un conjunto más completo de información

visual para tareas de regresión, e igualmente utilizó la información de un *dataframe* asociado a cada imagen del conjunto de datos (ver figura 3.21).

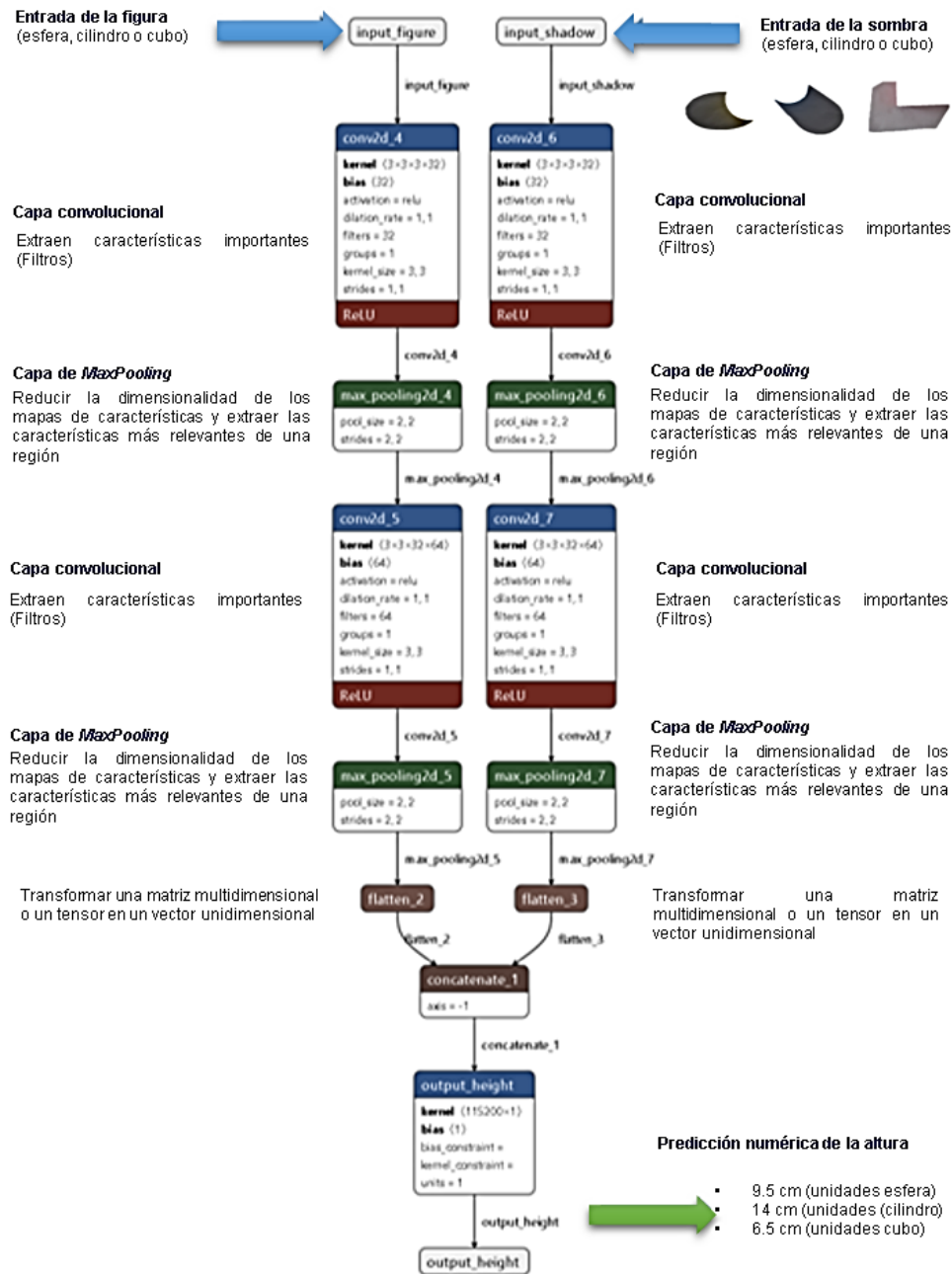


Figura 3.21. Resumen de la arquitectura del Modelo No.2. (creación propia)

La secuencia de capas en el modelo, donde se observan las capas iniciales que extrajeron las características visuales de las imágenes de la figura y la sombra más un *Dataframe*. De tal forma que a través de operaciones de convolución y *MaxPooling* detectaron patrones visuales como bordes y formas en las imágenes. Luego las capas *Flatten* que transforman las activaciones resultantes en vectores *1D* para cada imagen.

Posteriormente las capas de concatenación combinaron los vectores aplanados de la figura y de la sombra, permitiendo que el modelo considere tanto la figura como su sombra al hacer predicciones. Finalmente, la capa de salida densa con función de activación lineal produce una única salida numérica que representa la predicción de la altura del objeto 2D, basada en la información visual extraída de las imágenes. Por tanto, la topología y la estructura de las capas de este modelo se diseñaron para capturar las relaciones entre la figura, la sombra y la altura correspondiente. Esta topología de red neuronal convolucional para regresión aprovechó la información visual en las imágenes de figura, sombra y del *DataFrame* para realizar predicciones precisas de la altura del objeto 2D a partir de su sombra.

Arquitectura del modelo No.2. Modelo con dos entradas de imágenes: sombra e imagen de la figura

El modelo descrito tiene un total de catorce capas (14), así:

- a. Capa de entrada para la figura: *input_Figura (InputLayer)*
- b. Primera capa convolucional para la figura: *conv2d (Conv2D)*
- c. Primera capa de pooling para la figura: *max_pooling2d (MaxPooling2D)*
- d. Segunda capa convolucional para la figura: *conv2d_1 (Conv2D)*
- e. Segunda capa de pooling para la figura: *max_pooling2d_1 (MaxPooling2D)*
- f. Capa de aplanamiento para la figura: *flatten (Flatten)*
- g. Capa de entrada para la sombra: *input_shadow (InputLayer)*
- h. Primera capa convolucional para la sombra: *conv2d_2 (Conv2D)*
- i. Primera capa de pooling para la sombra: *max_pooling2d_2 (MaxPooling2D)*
- j. Segunda capa convolucional para la sombra: *conv2d_3 (Conv2D)*
- k. Segunda capa de pooling para la sombra: *max_pooling2d_3 (MaxPooling2D)*
- l. Capa de aplanamiento para la sombra: *flatten_1 (Flatten)*
- m. Capa de concatenación: *concatenate (Concatenate)*
- n. Capa densa de salida: *output_height (Dense)*

En total, el Modelo No.2. tiene 14 capas.

Entradas del modelo:

Las entradas la sombra (*input_shadow InputLayer*) y la imagen (*input_Figura, InputLayer*).

En este proceso se tomaron fotografías de forma manual a objetos que ya se tenían predefinidos, a partir de los cuales se tomaron fotografías y a partir de estas se preprocesaron para extraer en archivos separados la imagen de la sombra y la imagen de la figura. Estos archivos se almacenaron y organizaron para ser utilizados por el modelo. Pero es importante indicar que también en un trabajo posterior se puede automatizar esta tarea, pero en el desarrollo de esta investigación no se incluye.

- *Capas de Entrada:*

- *input_Figura (InputLayer)*: capa de entrada para la imagen de la figura.
- *input_shadow (InputLayer)*: capa de entrada para la imagen de la sombra.
- Descripción: ambas capas de entrada se diseñaron para recibir imágenes de tamaño 128×128 píxeles con tres canales de color (*RGB*).

Procesamiento de la imagen de la figura:

- *Capas convolucionales para figuras:*
 - *conv2d (Conv2D)*: primera capa convolucional con 32 filtros de $3 \times 3 \times 3$ y activación ReLU.
 - *conv2d_1 (Conv2D)*: segunda capa convolucional con 64 filtros de $3 \times 3 \times 3$ y activación ReLU.
- *Capas de Agrupación para Figuras:*
 - *max_pooling2d (MaxPooling2D)*: capa de agrupación máxima post-primer convolución.
 - *max_pooling2d_1 (MaxPooling2D)*: capa de agrupación máxima post-segunda convolución.
- *Capa de Aplanado para figuras:*
 - *flatten (Flatten)*: transforma las matrices *2D* en vectores *1D*.

Procesamiento de la imagen de la sombra:

- *Capas convolucionales para sombras:*
 - *conv2d_2 (Conv2D)*: primera capa convolucional con 32 filtros de $3 \times 3 \times 3$ y activación ReLU.
 - *conv2d_3 (Conv2D)*: segunda capa convolucional con 64 filtros de $3 \times 3 \times 3$ y activación ReLU.
- *Capas de agrupación para sombras:*
 - *max_pooling2d_2 (MaxPooling2D)*: capa de agrupación máxima post-primer convolución.
 - *max_pooling2d_3 (MaxPooling2D)*: capa de agrupación máxima post-segunda convolución.
- *Capa de aplanado para sombras:*
 - *flatten_1 (Flatten)*: transforma las matrices *2D* en vectores *1D*.

Concatenación y salida:

- *Capa de concatenación:*
 - *concatenate (Concatenate)*: combina los vectores aplanados de las figuras y las sombras.
- *Capa densa de salida:*
- *output_height (Dense)*: capa densa con una neurona de salida y activación lineal para la regresión de la altura.

3.4.2. Arquitectura del modelo No.1 (una entrada):

- *Entrada:* este modelo utilizó solo la imagen de la sombra como entrada. La columna `filename_image_shadow` del `DataFrame` proporcionó la ruta al modelo para cargar estas imágenes, en este caso la imagen de la sombra.
- *Salida:* la salida o etiqueta para el modelo es la altura de la figura, que se tomó de la columna `height_Figura`.
- *Propósito:* el modelo No.1. se diseñó para establecer una relación directa entre la apariencia de una sombra y la altura de la figura que la proyecta. Dado que utilizó solo la sombra como entrada, el modelo aprendió a interpretar las sutilezas y características de las sombras para hacer predicciones más precisas sobre la altura de las figuras.

El modelo entrenado puede, entonces, recibir una imagen de sombra como entrada y estimar la altura de la figura correspondiente mediante un proceso de inferencia que involucra todas las capas mencionadas anteriormente (ver figura 3.22).

```
Model: "model"
-----
Layer (type)                 Output Shape              Param #
-----
input_shadow (InputLayer)    [(None, 128, 128, 3)]    0
-----
conv2d (Conv2D)              (None, 126, 126, 32)     896
-----
max_pooling2d (MaxPooling2D) (None, 63, 63, 32)       0
-----
conv2d_1 (Conv2D)            (None, 61, 61, 64)       18496
-----
max_pooling2d_1 (MaxPooling2 (None, 30, 30, 64)       0
-----
flatten (Flatten)           (None, 57600)            0
-----
output_height (Dense)        (None, 1)                57601
-----
Total params: 76,993
Trainable params: 76,993
Non-trainable params: 0
```

Figura 3.22. Resumen de la arquitectura del modelo No.1.

El Modelo No. 1. se propone una estructura simplificada y eficiente para realizar tareas de regresión sobre las imágenes, siendo más específico en este caso para estimar la altura de figuras *2D* a partir de imágenes de su sombra. Utiliza como entrada la sombra para estimar la altura de la figura a partir de la sombra que proyecta y la salida es un valor dado en centímetros (ver figura 3.23).

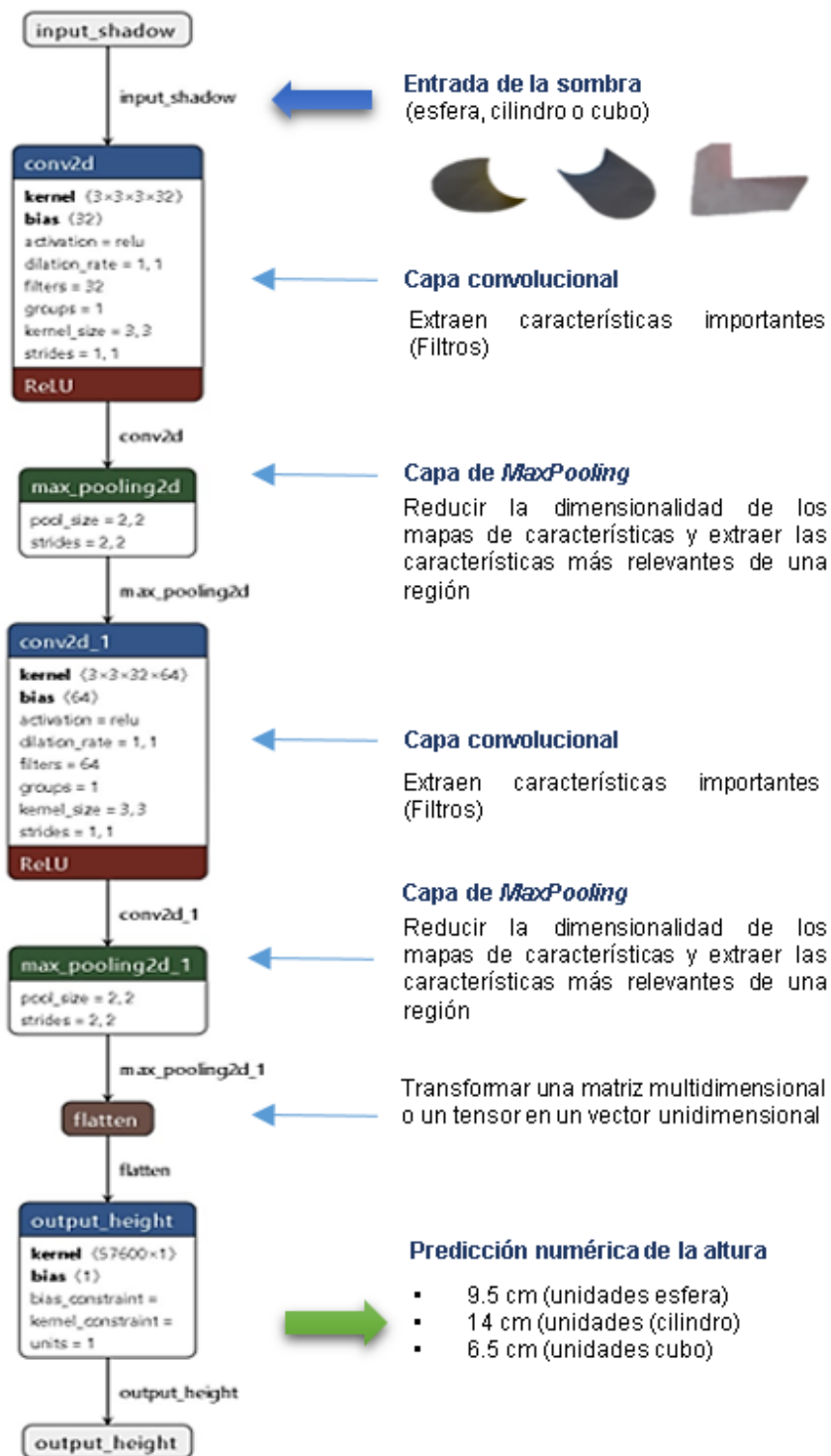


Figura 3.23. Arquitectura del Modelo No.1. (creación propia)

A continuación, se describe la estructura, diseño, arquitectura e implementación y los resultados del modelo No.1.:

Arquitectura Modelo No.1. Modelo con una entrada de imágenes: sombra de la figura

Esta corresponde a la arquitectura del modelo desarrollado con una entrada de imagen que es de la sombra de la figura:

El modelo descrito tiene un total de siete capas, así:

- a. Capa de Entrada (*InputLayer*): *input_shadow*.
- b. Primera Capa Convolutiva (*Conv2D*): *conv2d*.
- c. Primera Capa de *Pooling* (*MaxPooling2D*): *max_pooling2d*.
- d. Segunda Capa Convolutiva (*Conv2D*): *conv2d_1*.
- e. Segunda Capa de *Pooling* (*MaxPooling2D*): *max_pooling2d_1*.
- f. Capa de Aplanado (*Flatten*): *flatten*.
- g. Capa de Salida (*Dense*): *output_height*.

A continuación, se presenta a detalle cada una de las siete capas anteriores del modelo No.1.:

a. Capa de Entrada (*input_shadow – Input Layer*):

- Tipo: *InputLayer*
- Forma de salida: (*None*, 128, 128, 3)
- Descripción: Esta capa sirve como punto de entrada para las imágenes de sombras. Acepta tensores de forma (128, 128, 3), lo cual se traduce en imágenes de 128x128 píxeles con tres canales de color (*RGB*). El término 'None' refiere al tamaño del lote, que puede variar.

b. Primera Capa Convolutiva (*conv2d*):

- Tipo: convolutiva - *Conv2D*
- Forma de Salida: (*None*, 126, 126, 32)
- Parámetros: 896
- Descripción: Consta de 32 filtros de tamaño 3x3 y función de activación *ReLU*. Esta capa es crucial para la extracción inicial de características y patrones de las imágenes de entrada.

c. Primera Capa de *Pooling* (*max_pooling2d – MaxPooling2D*):

- Tipo: *MaxPooling2D*
- Forma de salida: (*None*, 63, 63, 32)
- Descripción: capa de aplanado (*Flatten*):
- Tipo: aplanamiento.
- Salida: (*None*, 57.600).
- Parámetros: 0.
- Descripción: Convierte el volumen de entrada en un vector unidimensional

de 57.600 elementos, necesario para la conexión con la capa densa siguiente. Utiliza un tamaño de ventana de 2×2 para reducir la dimensionalidad espacial de la entrada, preservando las características más significativas y reduciendo el riesgo de sobreajuste.

d. Segunda Capa Convolutiva (*conv2d_1*):

- Tipo: *Conv2D*
- Forma de salida: (*None*, 61, 61, 64)
- Parámetros: 18,496
- Descripción: Equipada con 64 filtros de tamaño 3×3 y función de activación *ReLU*, esta capa profundiza en la extracción de características más complejas y detalladas de las imágenes procesadas.

e. Segunda Capa de Pooling (*max_pooling2d_1*):

- Tipo: *MaxPooling2D*
- Forma de salida: (*None*, 30, 30, 64)
- Descripción: similar a la primera capa de *pooling*, esta capa reduce aún más la dimensionalidad de la entrada mientras conserva las características cruciales.

f. Capa de Aplanamiento (*flatten*):

- Tipo: *Flatten*
- Forma de salida: (*None*, 57600)
- Descripción: convierte los tensores $2D$ en vectores $1D$, preparando los datos para la capa densa siguiente. Convierte el volumen de entrada en un vector unidimensional de 57,600 elementos necesarios para la conexión con la capa densa siguiente.

g. Capa de Salida (*output_height*):

- Tipo: *Dense*
- Forma de salida: (*None*, 1)
- Parámetros: 57,601
- Descripción: esta es la capa de salida con una única unidad y activación lineal, diseñada para predecir un valor continuo, en este caso, la altura de la figura. Compuesta por una neurona con activación lineal, que emite una salida escalar correspondiente a la altura estimada de la figura de la imagen.

3.5. Diseño del modelo

El modelo de *CNN* optimizado representa un proceso automatizado que aprende a mapear patrones en las imágenes de sombra a las alturas correspondientes. Este proceso de aprendizaje se logra mediante la optimización iterativa de los pesos y sesgos en las capas de la red durante el entrenamiento. En el momento de realizar el proceso de aprendizaje del modelo de red neuronal propuesto, se tiene en cuenta que éste se distribuye o divide en un conjunto de entrenamiento, un conjunto de validación y un conjunto de prueba (para evaluar la calidad del modelo). Se realizó una distribución de los datos, el 80% para el conjunto de entrenamiento, el 10% para el conjunto de validación y el 10% para el conjunto de prueba. La figura 3.24 es una forma de caracterizar el proceso que se realizó.

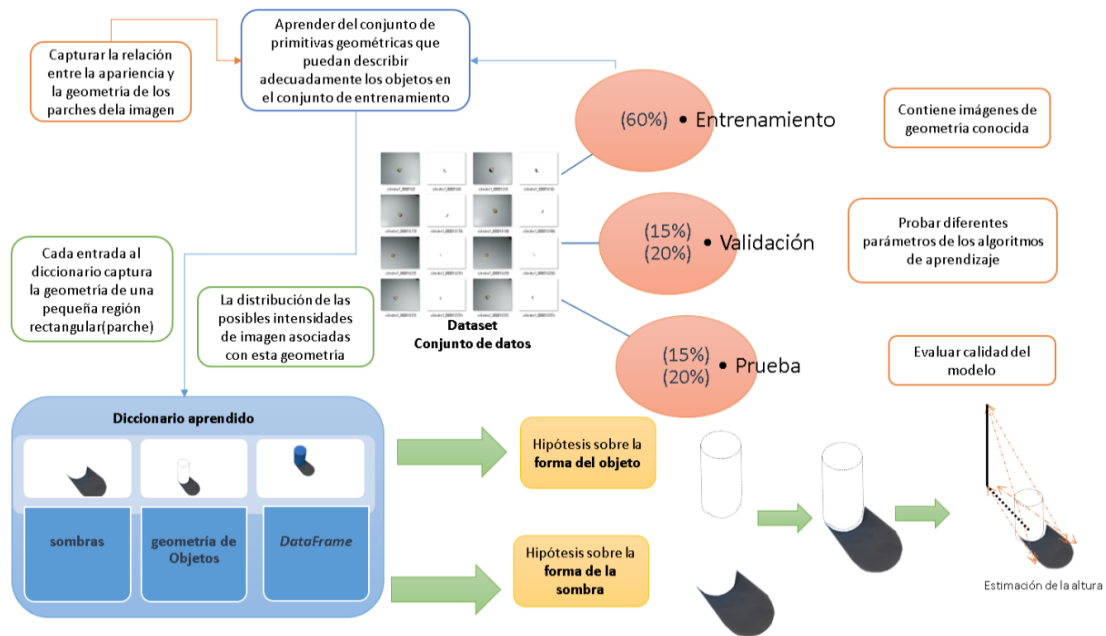


Figura 3.24. Esquema General del modelo propuesto. (creación propia)

3.6. Descripción general del algoritmo

A continuación, se relacionan los hiperparámetros del modelo No.1. propuesto en esta investigación, los cuales están indicados en cada una de las capas definidas en el modelo:

- *Epochs* = 600, 800, 1250, 1300, 1600
- *Batch size* = 32
- *Learning rate* = 0,001
- Tamaño de las imágenes = (128, 128, 3)

- **Capa de convolución 1 (para la imagen de la sombra):**
 - Filtros: 32
 - Tamaño del *kernel*: (3, 3)
 - Función de activación: *ReLU*
 - Capa de *max-pooling* 1 (para la imagen de la sombra):
 - Tamaño del *pool*: (2, 2)
- **Capa de convolución 2 (para la imagen de la sombra):**
 - Filtros: 64
 - Tamaño del *kernel*: (3, 3)
 - Función de activación: *ReLU*
- **Capa de max-pooling 2 (para la imagen de la sombra):**
 - Tamaño del *pool*: (2, 2)
- **Capa Flatten (para la imagen de la sombra):**
 - No tiene hiperparámetros específicos, ya que simplemente aplanar la salida.
- **Capa Dense (para la regresión de la altura):**
 - Neuronas de salida: 1 (indicando regresión)
 - Función de activación: *Linear* (lineal)
- **Compilación del modelo:**
 - Optimizador: *Adam*
 - Función de pérdida: *Mean Squared Error (MSE)*
 - Métricas: *Mean Absolute Error (MAE)* y *Mean Squared Error (MSE)*

A continuación, se relacionan algunos de los hiperparámetros del modelo No.2. propuesto en esta investigación, los cuales están indicados en cada una de las capas del modelo:

- *Epochs* = 600, 800, 1250, 1300, 1600
- *Batch size* = 32
- *Learning rate* = 0,001
- Tamaño de las imágenes (128, 128, 3)
- **Capa de convolución 1 (para la imagen de la figura):**
 - Filtros: 32
 - Tamaño del *kernel*: (3, 3)
 - Función de activación: *ReLU*
- **Capa de max-pooling 1 (para la imagen de la figura):**
 - Tamaño del *pool*: (2, 2)
- **Capa de convolución 2 (para la imagen de la figura):**

- Filtros: 64
- Tamaño del *kernel*: (3, 3)
- Función de activación: *ReLU*
- **Capa de max-pooling 2 (para la imagen de la figura):**
 - Tamaño del *pool*: (2, 2)
- **Capa Flatten (para la imagen de la figura):**
 - No tiene hiperparámetros específicos, ya que simplemente aplanar la salida.
- **Capa de convolución 1 (para la imagen de la sombra):**
 - Filtros: 32
 - Tamaño del *kernel*: (3, 3)
 - Función de activación: *ReLU*
- **Capa de max-pooling 1 (para la imagen de la sombra):**
 - Tamaño del *pool*: (2, 2)
- **Capa de convolución 2 (para la imagen de la sombra):**
 - Filtros: 64
 - Tamaño del *kernel*: (3, 3)
 - Función de activación: *ReLU*
- **Capa de max-pooling 2 (para la imagen de la sombra):**
 - Tamaño del *pool*: (2, 2)
- **Capa Flatten (para la imagen de la sombra):**
 - No tiene hiperparámetros específicos, ya que simplemente aplanar la salida.
- **Capa Concatenate (combinar las salidas de las capas de convolución de la figura y la sombra):**
 - No tiene hiperparámetros específicos.
- **Capa Dense (para la regresión de la altura):**
 - Neuronas de salida: 1 (indicando regresión)
 - Función de activación: *Linear* (lineal)
- **Compilación del modelo:**
 - Optimizador: Adam
 - Función de pérdida: *Mean Squared Error (MSE)*
 - Métricas: *Mean Absolute Error (MAE)* y *Mean Squared Error (MSE)*

El *preprocesamiento* de las imágenes *RGB* de la sombra se preprocesaron a un tamaño de tamaño (128x128x3). Con las capas convolucionales se extrajeron características de las imágenes de sombra. Estas capas ayudaron a la red neuronal convolucional a identificar patrones y características relevantes en las imágenes. La

extracción de información se realizó mediante una serie de filtros definidos y aplicados a la imagen de entrada para extraer características o patrones. *La función de activación* está presente al final de cada capa convolucional, en este caso se aplica una función de activación, *ReLU (Rectified Linear Unit)*, la cual es una función que se utilizó porque introduce no linealidad en el modelo, permitiendo que la red aprenda relaciones más complejas entre los datos.

Las *capas de agrupamiento (Pooling)* se encuentran en los modelos No.1, No.2, después de las capas convolucionales, se utilizaron en las capas de agrupamiento para reducir la dimensionalidad de las características extraídas, lo que hace que la red neuronal sea más eficiente en términos computacionales. Estas capas reducen la dimensionalidad de los datos conservando las características más importantes. En el modelo, se usa el *MaxPooling*, porque este selecciona el valor máximo de un grupo de *píxeles*. Además, se definió porque estas capas también permitieron que los modelos fueran más eficientes y permitió prevenir el sobreajuste. El *aplanado (Flatten)* es donde las características resultantes se aplanaron para transformar la estructura de datos de una matriz *2D* a un vector *1D*. *En la capa densa de salida* las características aplanadas se alimentaron en una capa densa con una función de activación lineal. Esta capa densa generó un único valor continuo que representó la altura predicha. En los modelos, la capa densa realizó la predicción final de la altura, produciendo un valor continuo. *La función de pérdida y optimizador:* en este caso, se utiliza el *MSE*. El optimizador *Adam* se utiliza para ajustar los pesos de la red basándose en esta pérdida, buscando minimizarla. Este *DataFrame* es esencial para entrenar y validar los modelos No.1 y No.2, ya que proporciona tanto las entradas (imágenes de figuras y sombras) como las salidas/etiquetas (alturas) necesarias para los procesos de aprendizaje supervisado. Además, al tener el ángulo de iluminación, se ofrece un contexto adicional que podría ser crucial para entender y mejorar el rendimiento de los modelos. Ambos modelos, No.1. y No.2., hacen uso de este *DataFrame*, aunque de maneras ligeramente diferentes:

Importancia de los Metadatos (DataFrame):

Las columnas adicionales, como *name_Figura* e *illumination_source_angle*, proporcionan metadatos valiosos. Estos metadatos pueden ser útiles para el análisis exploratorio de datos (EDA) ya que antes de modelar, es crucial comprender los datos. Saber cómo varían las sombras con diferentes figuras o cómo diferentes ángulos de iluminación afectan la longitud de la sombra puede proporcionar intuiciones valiosas. En futuras iteraciones del modelo, estos metadatos pueden usarse para crear características adicionales que podrían mejorar el rendimiento del modelo. También se utilizan para el diagnóstico del modelo ya que, si un modelo comete errores en sus predicciones, estos metadatos pueden ayudar a identificar patrones. Por ejemplo, si el modelo tiende a cometer errores con sombras de cilindros bajo ciertos ángulos de iluminación, estos metadatos pueden ayudar a identificar y corregir ese patrón.

Capítulo IV. Implementación y experiencias realizadas

En el capítulo anterior se desarrolló la arquitectura de los dos modelos, la estructura y construcción del *dataframe* y del *dataset* por medio de la metodología Incremental. Por lo tanto, en este capítulo se realiza la implementación de los dos modelos de redes neuronales convolucionales con regresión, para predecir la altura, el primero de estos modelos está conformado por dos entradas, que son la imagen de la figura y la imagen de la sombra, y el segundo modelo con una sola entrada, la sombra de la figura. Posteriormente se utiliza el *dataframe* y el *dataset* que son utilizados para el proceso de entrenamiento, validación y prueba de los dos modelos.

A continuación, se expone el proceso de validación cruzada aplicado a los dos modelos

4.1. Validación cruzada (*k-Fold Cross Validation*)

El propósito de evaluar un modelo en diferentes épocas, es determinar cómo la cantidad de iteraciones durante el entrenamiento afecta el rendimiento del modelo. Se le aumentan las épocas para que pueda aprender patrones de los datos, evitando el riesgo de sobreajuste. La validación cruzada (*K-Fold Cross-Validación*) se utiliza para evaluar la capacidad de generalización de un modelo al entrenarlo, e igualmente evaluarlo en diferentes subconjuntos del conjunto de datos original. En este caso, se ha utilizado un *K-fold* con $k = 5$, lo que significa que los datos se han dividido en 5 subconjuntos (*folds*). En cada iteración de validación cruzada, 4 de estos *folds* se utilizan para entrenar el modelo y 1 *fold* se utiliza para evaluarlo Markatou, *et al.* (2005). Este proceso se repite 5 veces, cada vez con un *fold* diferente como conjunto de entrenamiento y de validación.

- **Primer Iteración:**
 - *Folds* 1, 2, 3, 4: **entrenamiento**
 - *Fold* 5: **validación**
- **Segunda Iteración:**
 - *Folds* 1, 2, 3, 4: **entrenamiento**
 - *Fold* 5: **validación**
- **Tercera Iteración:**
 - *Folds* 1, 2, 3, 4: **Entrenamiento**
 - *Fold* 5: **Validación**

- **Cuarta Iteración:**
 - *Folds* 1, 2, 3, 4: **entrenamiento**
 - *Fold* 5: **validación**

- **Quinta Iteración:**
 - *Folds* 1, 2, 3, 4: **entrenamiento**
 - *Fold* 5: **validación**

En general, es una buena práctica realizar la validación cruzada para obtener una estimación más estable del rendimiento del modelo. Esta se realizó a los dos modelos tanto al modelo No.1. que se le ha llamado así, porque tiene una entrada (la sombra) y al modelo No.2. que se le ha llamado así porque tiene dos entradas (la imagen y la sombra), como se ha indicado anteriormente.

A continuación, se muestran las validaciones cruzadas para los dos modelos de forma secuencial con sus gráficas respectivas con el objetivo de permitir una evaluación más completa respectivamente, para más detalles de este proceso (Hecht, E. *et al.*, 2016).

4.1.1. Validación cruzada modelo No.2. $k = 5$

A continuación, en la tabla No 4.1, se muestra el proceso y los resultados realizados en la validación cruzada aplicada, en este caso con $k = 5$, los cuales corresponden al modelo 2.

Modelos	Épocas	<i>Fold</i>	<i>Loss</i>	<i>MAE</i>	<i>MSE</i>	Modelos	Épocas	<i>Fold</i>	<i>Loss</i>	<i>MAE</i>	<i>MSE</i>
Modelo 2A	600	1	0,63	0,51	0,63	Modelo 2B	800	1	0,61	0,50	0,61
Modelo 2A	600	2	1,04	0,59	1,04	Modelo 2B	800	2	1,18	0,57	1,18
Modelo 2A	600	3	0,46	0,47	0,46	Modelo 2B	800	3	0,47	0,49	0,47
Modelo 2A	600	4	0,59	0,51	0,59	Modelo 2B	800	4	0,59	0,53	0,59
Modelo 2A	600	5	0,46	0,47	0,46	Modelo 2B	800	5	0,35	0,42	0,35
Rendimiento promedio			0,64	0,51	0,64	Rendimiento promedio			0,64	0,50	0,64
Modelos	Épocas	<i>Fold</i>	<i>Loss</i>	<i>MAE</i>	<i>MSE</i>	Modelos	Épocas	<i>Fold</i>	<i>Loss</i>	<i>MAE</i>	<i>MSE</i>
Modelo 2C	1250	1	0,60	0,52	0,60	Modelo 2D	1600	1	0,52	0,48	0,52
Modelo 2C	1250	2	0,80	0,53	0,80	Modelo 2D	1600	2	0,86	0,52	0,86
Modelo 2C	1250	3	0,43	0,45	0,43	Modelo 2D	1600	3	0,49	0,48	0,49
Modelo 2C	1250	4	0,54	0,50	0,54	Modelo 2D	1600	4	0,55	0,50	0,55
Modelo 2C	1250	5	0,34	0,42	0,34	Modelo 2D	1600	5	0,44	0,46	0,44

Rendimiento promedio	0,54	0,48	0,54	Rendimiento promedio	0,57	0,49	0,57
----------------------	------	------	------	----------------------	------	------	------

Tabla No.4.1. Validaciones cruzadas del modelo No.2. con $k = 5$. (creación propia)

Después de realizar la validación cruzada se obtuvieron los siguientes resultados del modelo No.2. en diferentes épocas, posterior a los resultados de la tabla 4.2., se analiza que en este caso se sugiere seleccionar el modelo 2C, que tiene los siguientes resultados:

Modelos	Épocas	Folds=1,2,3,4,5	Pérdida del Modelo (Loss)	Error Absoluto Medio (MAE)	Error Cuadrático Medio (MSE)
Modelo 2A	600	1,2,3,4,5	0,64	0,51	0,64
Modelo 2B	800	1,2,3,4,5	0,64	0,50	0,64
Modelo 2C	1.250	1,2,3,4,5	0,54	0,48	0,54
Modelo 2D	1.600	1,2,3,4,5	0,57	0,49	0,57

Tabla No.4.2. Resultados Generales de la validación cruzada aplicada al modelo No.2. (creación propia)

4.1.2. Validación cruzada modelo No.2. $k = 10$

A continuación, en la tabla 4.3, se muestra el proceso y los resultados realizados en la validación cruzada aplicado, en este caso con $k = 10$, los cuales corresponden al modelo 2.

Modelos	Épocas	Fold	Loss	MAE	MSE	Modelos	Épocas	Fold	Loss	MAE	MSE
Modelo 2A	600	1	0,44	0,50	0,44	Modelo 2B	800	0,34	0,40	0,34	0,34
Modelo 2A	600	2	0,76	0,58	0,76	Modelo 2B	800	0,73	0,58	0,73	0,73
Modelo 2A	600	3	0,43	0,46	0,43	Modelo 2B	800	0,51	0,56	0,51	0,51
Modelo 2A	600	4	0,63	0,52	0,63	Modelo 2B	800	0,70	0,55	0,70	0,70
Modelo 2A	600	5	0,56	0,54	0,56	Modelo 2B	800	0,43	0,46	0,43	0,43
Modelo 2A	600	6	0,44	0,45	0,44	Modelo 2B	800	0,44	0,44	0,44	0,44
Modelo 2A	600	7	0,71	0,54	0,71	Modelo 2B	800	0,64	0,47	0,64	0,64
Modelo 2A	600	8	0,69	0,61	0,69	Modelo 2B	800	0,51	0,49	0,51	0,51
Modelo 2A	600	9	0,49	0,49	0,49	Modelo 2B	800	0,29	0,39	0,29	0,29
Modelo 2A	600	10	0,66	0,53	0,66	Modelo 2B	800	0,54	0,48	0,54	0,54
Rendimiento promedio			0,52	0,48	0,52	Rendimiento promedio			0,51	0,48	0,51
Modelos	Épocas	Fold	Loss	MAE	MSE	Modelos	Épocas	Fold	Loss	MAE	MSE
Modelo 2C	1.250	1	0,30	0,38	0,30	Modelo 2D	1.600	0,30	0,39	0,30	0,30
Modelo 2C	1.250	2	0,82	0,57	0,82	Modelo 2D	1.600	0,77	0,51	0,77	0,77
Modelo 2C	1.250	3	0,36	0,44	0,36	Modelo 2D	1.600	0,37	0,44	0,37	0,37

Modelo 2C	1.250	4	0,73	0,54	0,73	Modelo 2D	1.600	0,62	0,50	0,62	0,62
Modelo 2C	1.250	5	0,41	0,45	0,41	Modelo 2D	1.600	0,40	0,42	0,40	0,40
Modelo 2C	1.250	6	0,42	0,41	0,42	Modelo 2D	1.600	0,35	0,38	0,35	0,35
Modelo 2C	1.250	7	0,61	0,47	0,61	Modelo 2D	1.600	0,56	0,44	0,56	0,56
Modelo 2C	1.250	8	0,49	0,50	0,49	Modelo 2D	1.600	0,48	0,50	0,48	0,48
Modelo 2C	1.250	9	0,35	0,43	0,35	Modelo 2D	1.600	0,36	0,38	0,36	0,36
Modelo 2C	1.250	10	0,36	0,41	0,36	Modelo 2D	1.600	0,50	0,45	0,50	0,50
Rendimiento promedio			0,49	0,46	0,49	Rendimiento promedio			0,47	0,44	0,47

Tabla No 4.3. Validación cruzada del modelo No.2. con $k = 10$. (creación propia)

Después de realizar la validación cruzada se obtuvieron los siguientes resultados del modelo No.2. en diferentes épocas con $k = 10$, se sugiere en este caso utilizar el modelo 2D (ver tabla 4.4).

Modelos	Fold=1,2,...,10	Épocas	Pérdida del Modelo (Loss)	Error Absoluto Medio (MAE)	Error Cuadrático Medio (MSE)
Modelo 2A	1...10	600	0,52	0,48	0,52
Modelo 2B	1...10	800	0,51	0,48	0,51
Modelo 2C	1...10	1.250	0,49	0,46	0,49
Modelo 2D	1...10	1.600	0,47	0,44	0,47

Tabla No.4.4. Resultados finales de la validación cruzada del modelo No.2. con $k = 10$. (creación propia)

4.1.3. Validación cruzada modelo No.2. $k = 20$

A continuación, en la tabla 4.5, se muestra el proceso y los resultados realizados en la validación cruzada aplicado, en este caso con $k = 20$, los cuales corresponden al modelo 2.

Modelos	Épocas	Fold	Loss	MAE	MSE	Modelos	Épocas	Fold	Loss	MAE	MSE
Modelo 2A	600	1	0,26	0,38	0,26	Modelo 2B	800	1	0,48	0,50	0,48
Modelo 2A	600	2	0,41	0,43	0,41	Modelo 2B	800	2	0,45	0,43	0,45
Modelo 2A	600	3	0,49	0,51	0,49	Modelo 2B	800	3	0,51	0,51	0,51
Modelo 2A	600	4	1,18	0,56	1,18	Modelo 2B	800	4	1,05	0,61	1,05
Modelo 2A	600	5	0,43	0,50	0,43	Modelo 2B	800	5	0,68	0,59	0,68
Modelo 2A	600	6	0,22	0,36	0,22	Modelo 2B	800	6	0,27	0,36	0,27
Modelo 2A	600	7	0,65	0,50	0,65	Modelo 2B	800	7	0,58	0,51	0,58

Modelo 2A	600	8	0,72	0,61	0,72	Modelo 2B	800	8	1,04	0,70	1,04
Modelo 2A	600	9	0,27	0,38	0,27	Modelo 2B	800	9	0,50	0,51	0,50
Modelo 2A	600	10	0,48	0,50	0,48	Modelo 2B	800	10	0,71	0,57	0,71
Modelo 2A	600	11	0,29	0,41	0,29	Modelo 2B	800	11	0,31	0,41	0,31
Modelo 2A	600	12	0,44	0,46	0,44	Modelo 2B	800	12	0,61	0,56	0,61
Modelo 2A	600	13	1,03	0,59	1,03	Modelo 2B	800	13	1,16	0,65	1,16
Modelo 2A	600	14	0,32	0,40	0,32	Modelo 2B	800	14	0,39	0,43	0,39
Modelo 2A	600	15	0,57	0,55	0,57	Modelo 2B	800	15	0,77	0,62	0,77
Modelo 2A	600	16	0,48	0,48	0,48	Modelo 2B	800	16	0,51	0,51	0,51
Modelo 2A	600	17	0,30	0,39	0,30	Modelo 2B	800	17	0,31	0,43	0,31
Modelo 2A	600	18	0,55	0,46	0,55	Modelo 2B	800	18	0,49	0,51	0,49
Modelo 2A	600	19	0,33	0,45	0,33	Modelo 2B	800	19	0,39	0,46	0,39
Modelo 2A	600	20	0,52	0,45	0,52	Modelo 2B	800	20	0,58	0,50	0,58
Rendimiento promedio			0,50	0,47	0,50	Rendimiento promedio			0,57	0,52	0,57
Modelos	Épocas	Fold	Loss	MAE	MSE	Modelos	Épocas	Fold	Loss	MAE	MSE
Modelo 2C	1250	1	0,48	0,50	0,48	Modelo 2D	1600	1	0,30	0,44	0,30
Modelo 2C	1250	2	0,45	0,43	0,45	Modelo 2D	1600	2	0,48	0,48	0,48
Modelo 2C	1250	3	0,51	0,51	0,51	Modelo 2D	1600	3	0,76	0,61	0,76
Modelo 2C	1250	4	1,05	0,61	1,05	Modelo 2D	1600	4	0,11	0,59	0,11
Modelo 2C	1250	5	0,68	0,59	0,68	Modelo 2D	1600	5	0,71	0,61	0,71
Modelo 2C	1250	6	0,27	0,36	0,27	Modelo 2D	1600	6	0,25	0,38	0,25
Modelo 2C	1250	7	0,58	0,51	0,58	Modelo 2D	1600	7	0,62	0,47	0,62
Modelo 2C	1250	8	1,04	0,70	1,04	Modelo 2D	1600	8	0,87	0,64	0,87
Modelo 2C	1250	9	0,50	0,51	0,50	Modelo 2D	1600	9	0,32	0,42	0,32
Modelo 2C	1250	10	0,71	0,57	0,71	Modelo 2D	1600	10	0,57	0,56	0,57
Modelo 2C	1250	11	0,31	0,41	0,31	Modelo 2D	1600	11	0,32	0,40	0,32
Modelo 2C	1250	12	0,61	0,56	0,61	Modelo 2D	1600	12	0,58	0,56	0,58
Modelo 2C	1250	13	1,16	0,65	1,16	Modelo 2D	1600	13	0,12	0,69	0,12
Modelo 2C	1250	14	0,39	0,43	0,39	Modelo 2D	1600	14	0,40	0,43	0,40
Modelo 2C	1250	15	0,77	0,62	0,77	Modelo 2D	1600	15	0,56	0,55	0,56
Modelo 2C	1250	16	0,51	0,51	0,51	Modelo 2D	1600	16	0,41	0,47	0,41
Modelo 2C	1250	17	0,31	0,43	0,31	Modelo 2D	1600	17	0,34	0,46	0,34
Modelo 2C	1250	18	0,49	0,51	0,49	Modelo 2D	1600	18	0,11	0,58	0,11
Modelo 2C	1250	19	0,39	0,46	0,39	Modelo 2D	1600	19	0,38	0,48	0,38
Modelo 2C	1250	20	0,58	0,50	0,58	Modelo 2D	1600	20	0,70	0,52	0,70
Rendimiento promedio			0,57	0,52	0,57	Rendimiento promedio			0,47	0,55	0,47

Tabla No.4.5. Validación cruzada del modelo No.2. con $k = 20$. (creación propia)

Después de realizar la validación cruzada se obtuvieron los siguientes resultados en la tabla 4.6, con los valores promedio correspondientes al *MAE*, *MSE* y *LOSS*.

Modelos	Fold	Épocas	Loss Pérdida	MAE (Error Absoluto Medio)	MSE (Error Cuadrático Medio)
Modelo 2A	1...20	600	0,58	0,54	0,58
Modelo 2B	1...20	800	0,57	0,52	0,57
Modelo 2C	1...20	1.250	0,57	0,52	0,57
Modelo 2D	1...20	1.600	0,47	0,55	0,47

Tabla No.4.6. Resultados finales de la validación cruzada del modelo No.2. con $k = 20$.
(creación propia)

4.1.4. Análisis validación cruzada del modelo No.2.

Este análisis solo comprende la validación cruzada del modelo No.2. con dos entradas: la imagen y la sombra, ver figura 4.7, 4.8 y 4.9.

Resumen de resultados de las validaciones cruzadas con fold = 5, 10 y 20:

Estos valores corresponden a las pruebas de validación del modelo No.2. que es la red neuronal convolucional para regresión con un dato de entrada: la sombra.

Con fold = 5

Modelo No.2	Épocas	Fold	Loss Pérdida	MAE (Error Absoluto Medio)	MSE (Error Cuadrático Medio)
Modelo 1A	600	1,2,3,4,5	0,64	0,52	0,64
Modelo 1B	800	1,2,3,4,5	0,64	0,50	0,64
Modelo 1C	1.250	1,2,3,4,5	0,54	0,48	0,54
Modelo 1D	1.600	1,2,3,4,5	0,57	0,49	0,57

Tabla No.4.7. Resumen de la validación cruzada del modelo No.1. (creación propia)

Con fold = 10

Modelo No.2	Épocas	Fold	Loss Pérdida	MAE (Error Absoluto Medio)	MSE (Error Cuadrático Medio)
Modelo 1A	600	1,...,10	0,58	0,52	0,58
Modelo 1B	800	1,...,10	0,57	0,52	0,57
Modelo 1C	1.250	1,...,10	0,61	0,53	0,61
Modelo 1D	1.600	1,...,10	0,58	0,52	0,58

Tabla No.4.8. Resumen de la validación cruzada del modelo No.1. (creación propia)

Con *fold* = 20

Modelo No.1	Épocas	Fold	Loss Pérdida	MAE (Error Absoluto Medio)	MSE (Error Cuadrático Medio)
Modelo 1A	600	1,...,20	0,58	0,52	0,58
Modelo 1B	800	1,...,20	0,58	0,52	0,58
Modelo 1C	1.250	1,...,20	0,58	0,52	0,58
Modelo 1D	1.600	1,...,20	0,60	0,52	0,60

Tabla No.4.9. Resumen de la validación cruzada del modelo No.1. (*creación propia*)

Análisis

Según lo anterior de las validaciones realizadas se selecciona el modelo con 1.250 épocas. Analizando los *folds* y los valores de los resultados indicado en cada una de las tablas, se puede decir lo siguiente:

En el *fold* = 5, todos los modelos muestran una pérdida de 0,54 a 0,64, los valores de *MAE* varían de 0,48 a 0,52 siendo el modelo 1C el mejor en esta métrica y los valores del *MSE* son consistentes con los del *MAE*, también indicando que el modelo 1C es el mejor. En el *fold* = 10, los valores de pérdida están entre 0,57 y 0,61, siendo el 1B el más bajo. El *MAE* de 0,52 es más bajo para los modelos 1A y 1D y el *MSE* tiene resultados similares al *MAE*, con los modelos 1A y 1D teniendo el valor más bajo de 0,58. En el *fold* = 20, los valores de pérdida varían de 0,58 a 0,60, el *MAE* es uniforme para todos los modelos en 0,52. El *MSE* oscila entre 0,58 y 0,60, con 0,58 es el mismo en la mayoría de los modelos excepto el modelo 1D. Al analizar los resultados, se puede observar que no hay un modelo que sea más consistente, que supere a los demás en todas las métricas y configuraciones de *fold*. El modelo 1C tiene un buen desempeño especialmente en el *fold* de 5. De igual forma es notorio que el aumento del número de épocas no necesariamente mejora el rendimiento del modelo, ya que no hay una tendencia clara de mejora o podría empeorar con un mayor número de épocas.

Basado en estos datos, se selecciona el modelo 1C, ya que tiene el mejor *MAE* con un *fold* de 5 y su desempeño es el mejor respecto de otras configuraciones. Aunque el modelo 1D tiene un buen *MAE* y *MSE*, con un *fold* de 10, pérdida aumenta ligeramente con un *fold* de 20, lo puede indicar una menor estabilidad. En cuanto al rendimiento el modelo 1C tiene una variación relativamente baja en las métricas entre los *folds*, lo que sugiere que es menos sensible a la variación en los datos de entrenamiento. Se opta por este modelo por los valores bajos del *MAE* y del *MSE*, ya que esto indica que el modelo tiene la capacidad de predecir con precisión la altura de la figura. Según lo anterior y basándose únicamente en estos resultados, los de la validación cruzada, el modelo 1C sea la elección más robusta y equilibrada para la estimación de altura.

4.1.5. Validación cruzada modelo No.1. $k = 5$

A continuación en la tabla 4.10 y 4.11, se muestra el proceso y los resultados realizados en la validación cruzada aplicado, en este caso con $k = 5$, los cuales corresponden al modelo 1.

Modelo	Épocas	Fold	Loss	MAE	MSE	Modelo	Épocas	Fold	Loss	MAE	MSE
Modelo 1A	600	1	0,62	0,55	0,62	Modelo 1B	800	1	0,62	0,54	0,62
Modelo 1A	600	2	0,66	0,55	0,66	Modelo 1B	800	2	1,30	0,62	1,30
Modelo 1A	600	3	1,35	0,63	1,35	Modelo 1B	800	3	0,65	0,56	0,65
Modelo 1A	600	4	0,60	0,54	0,60	Modelo 1B	800	4	0,86	0,65	0,86
Modelo 1A	600	5	0,43	0,49	0,43	Modelo 1B	800	5	0,56	0,50	0,56
Rendimiento promedio			0,73	0,55	0,73	Rendimiento promedio			0,80	0,57	0,80
Modelo	Épocas	Fold	Loss	MAE	MSE	Modelo	Épocas	Fold	Loss	MAE	MSE
Modelo 1C	1250	1	0,60	0,54	0,60	Modelo 1D	1600	1	0,73	0,55	0,73
Modelo 1C	1250	2	1,40	0,62	1,40	Modelo 1D	1600	2	0,62	0,55	0,66
Modelo 1C	1250	3	0,55	0,52	0,55	Modelo 1D	1600	3	1,35	0,63	1,35
Modelo 1C	1250	4	0,67	0,56	0,67	Modelo 1D	1600	4	0,60	0,54	0,60
Modelo 1C	1250	5	0,52	0,49	0,52	Modelo 1D	1600	5	0,43	0,49	0,43
Rendimiento promedio			0,75	0,54	0,75	Rendimiento promedio			0,75	0,55	0,76

Tabla No.4.10. Validación cruzada del modelo No.1. con $k = 5$. (creación propia)

Después de realizar la validación cruzada se obtuvieron los siguientes resultados del modelo No.1. en diferentes épocas, se sugiere en este caso utilizar el modelo 1C. (ver tabla 4.11).

Modelos	Épocas	Fold=1,2,3,4,5	Pérdida del Modelo (Loss)	Error Absoluto Medio (MAE)	Error Cuadrático Medio (MSE)
Modelo 1A	600	1,2,3,4,5	0,73	0,55	0,73
Modelo 1B	800	1,2,3,4,5	0,80	0,57	0,80
Modelo 1C	1.250	1,2,3,4,5	0,75	0,54	0,75
Modelo 1D	1.600	1,2,3,4,5	0,75	0,55	0,76

Tabla No 4.11. Resultados generales de la validación cruzada aplicada al modelo No.1. (creación propia)

A continuación, se detalla parte del proceso y de los resultados realizados en la validación cruzada aplicada a los diferentes modelos.

4.1.6. Validación cruzada modelo No.1. $k = 10$

A continuación en la tabla 4.12 y 4.13, se muestra el proceso y los resultados realizados en la validación cruzada aplicado, en este caso con $k = 10$, los cuales corresponden al modelo 1.

Modelo	Épocas	Fold	Loss	MAE	MSE	Modelo	Épocas	Fold	Loss	MAE	MSE
Modelo 1A	600	1	0,44	0,50	0,44	Modelo 1B	800	1	0,47	0,50	0,47
Modelo 1A	600	2	0,76	0,58	0,76	Modelo 1B	800	2	0,76	0,61	0,76
Modelo 1A	600	3	0,43	0,46	0,43	Modelo 1B	800	3	0,51	0,51	0,51
Modelo 1A	600	4	0,63	0,52	0,63	Modelo 1B	800	4	0,72	0,55	0,72
Modelo 1A	600	5	0,56	0,54	0,56	Modelo 1B	800	5	0,47	0,50	0,47
Modelo 1A	600	6	0,44	0,45	0,44	Modelo 1B	800	6	0,51	0,48	0,51
Modelo 1A	600	7	0,71	0,54	0,71	Modelo 1B	800	7	0,73	0,51	0,73
Modelo 1A	600	8	0,69	0,61	0,69	Modelo 1B	800	8	0,72	0,62	0,72
Modelo 1A	600	9	0,49	0,49	0,49	Modelo 1B	800	9	0,41	0,47	0,41
Modelo 1A	600	10	0,66	0,53	0,66	Modelo 1B	800	10	0,46	0,46	0,46
Rendimiento promedio			0,58	0,52	0,58	Rendimiento promedio			0,57	0,52	0,57
Modelo	Épocas	Fold	Loss	MAE	MSE	Modelo	Épocas	Fold	Loss	MAE	MSE
Modelo 1C	1250	1	0,45	0,50	0,45	Modelo 1D	1600	1	0,35	0,45	0,35
Modelo 1C	1250	2	1,09	0,70	1,09	Modelo 1D	1600	2	0,96	0,67	0,96
Modelo 1C	1250	3	0,43	0,45	0,43	Modelo 1D	1600	3	0,44	0,48	0,44
Modelo 1C	1250	4	0,74	0,54	0,74	Modelo 1D	1600	4	0,86	0,59	0,86
Modelo 1C	1250	5	0,47	0,49	0,47	Modelo 1D	1600	5	0,40	0,48	0,40
Modelo 1C	1250	6	0,52	0,48	0,52	Modelo 1D	1600	6	0,56	0,51	0,56
Modelo 1C	1250	7	0,78	0,54	0,78	Modelo 1D	1600	7	0,68	0,52	0,68
Modelo 1C	1250	8	0,58	0,58	0,58	Modelo 1D	1600	8	0,64	0,58	0,64
Modelo 1C	1250	9	0,48	0,47	0,48	Modelo 1D	1600	9	0,52	0,49	0,52
Modelo 1C	1250	10	0,61	0,50	0,61	Modelo 1D	1600	10	0,49	0,48	0,49
Rendimiento promedio			0,61	0,53	0,61	Rendimiento promedio			0,58	0,52	0,58

Tabla No.4.12. Validación cruzada del modelo No.1. con $k = 10$. (creación propia)

Después de realizar la validación cruzada se obtuvieron los siguientes resultados del modelo No.1. en diferentes épocas con $k = 10$, se sugiere en este caso utilizar el modelo 1B (ver tabla 4.13).

Modelos	Fold=1,2,...,10	Epochs	Pérdida del Modelo (Loss)	Error Absoluto Medio (MAE)	Error Cuadrático Medio (MSE)
Modelo 1A	Promedio	600	0,58	0,52	0,58
Modelo 1B	Promedio	800	0,57	0,52	0,57
Modelo 1C	Promedio	1.250	0,61	0,53	0,61
Modelo 1D	Promedio	1.600	0,58	0,52	0,58

Tabla No 4.13. Resultados finales de la validación cruzada del modelo No.1. con $k = 10$. (creación propia)

4.1.7. Validación cruzada modelo No.1. $k = 20$

A continuación, en la tabla 4.14, se muestra el proceso y los resultados realizados en la validación cruzada aplicado, en este caso con $k = 20$, los cuales corresponden al modelo 1.

Modelo	Épocas	Fold	Loss	MAE	MSE	Modelo	Épocas	Fold	Loss	MAE	MSE
Modelo 1A	600	1	0,40	0,49	0,40	Modelo 1B	800	1	0,37	0,44	0,37
Modelo 1A	600	2	0,56	0,50	0,56	Modelo 1B	800	2	0,49	0,48	0,49
Modelo 1A	600	3	0,53	0,55	0,53	Modelo 1B	800	3	0,61	0,59	0,61
Modelo 1A	600	4	0,86	0,64	0,86	Modelo 1B	800	4	0,87	0,56	0,87
Modelo 1A	600	5	0,55	0,52	0,55	Modelo 1B	800	5	0,49	0,53	0,49
Modelo 1A	600	6	0,20	0,31	0,20	Modelo 1B	800	6	0,37	0,45	0,37
Modelo 1A	600	7	0,43	0,43	0,43	Modelo 1B	800	7	0,86	0,57	0,86
Modelo 1A	600	8	0,78	0,60	0,78	Modelo 1B	800	8	0,85	0,61	0,85
Modelo 1A	600	9	0,29	0,39	0,29	Modelo 1B	800	9	0,43	0,50	0,43
Modelo 1A	600	10	0,78	0,61	0,78	Modelo 1B	800	10	0,75	0,62	0,75
Modelo 1A	600	11	0,36	0,45	0,36	Modelo 1B	800	11	0,36	0,43	0,36
Modelo 1A	600	12	0,69	0,57	0,69	Modelo 1B	800	12	0,60	0,56	0,60
Modelo 1A	600	13	0,13	0,70	0,13	Modelo 1B	800	13	0,12	0,61	0,12
Modelo 1A	600	14	0,32	0,40	0,32	Modelo 1B	800	14	0,39	0,47	0,39
Modelo 1A	600	15	0,67	0,60	0,67	Modelo 1B	800	15	0,64	0,56	0,64
Modelo 1A	600	16	0,53	0,51	0,53	Modelo 1B	800	16	0,45	0,47	0,45
Modelo 1A	600	17	0,50	0,53	0,50	Modelo 1B	800	17	0,33	0,45	0,33
Modelo 1A	600	18	0,65	0,51	0,65	Modelo 1B	800	18	0,57	0,54	0,57
Modelo 1A	600	19	0,53	0,56	0,53	Modelo 1B	800	19	0,39	0,47	0,39
Modelo 1A	600	20	0,72	0,55	0,72	Modelo 1B	800	20	0,65	0,47	0,65
Rendimiento promedio			0,58	0,54	0,58	Rendimiento promedio			0,57	0,52	0,57

Modelo	Épocas	Fold	Loss	MAE	MSE	Modelo	Épocas	Fold	Loss	MAE	MSE
Modelo 1C	1250	1	0,37	0,44	0,37	Modelo 1D	1600	1	0,30	0,44	0,30
Modelo 1C	1250	2	0,49	0,48	0,49	Modelo 1D	1600	2	0,48	0,48	0,48
Modelo 1C	1250	3	0,61	0,59	0,61	Modelo 1D	1600	3	0,76	0,61	0,76
Modelo 1C	1250	4	0,87	0,56	0,87	Modelo 1D	1600	4	0,11	0,59	0,11
Modelo 1C	1250	5	0,49	0,53	0,49	Modelo 1D	1600	5	0,71	0,61	0,71
Modelo 1C	1250	6	0,37	0,45	0,37	Modelo 1D	1600	6	0,25	0,38	0,25
Modelo 1C	1250	7	0,86	0,57	0,86	Modelo 1D	1600	7	0,62	0,47	0,62
Modelo 1C	1250	8	0,85	0,61	0,85	Modelo 1D	1600	8	0,87	0,64	0,87
Modelo 1C	1250	9	0,43	0,50	0,43	Modelo 1D	1600	9	0,32	0,42	0,32
Modelo 1C	1250	10	0,75	0,62	0,75	Modelo 1D	1600	10	0,57	0,56	0,57
Modelo 1C	1250	11	0,36	0,43	0,36	Modelo 1D	1600	11	0,32	0,40	0,32
Modelo 1C	1250	12	0,60	0,56	0,60	Modelo 1D	1600	12	0,58	0,56	0,58
Modelo 1C	1250	13	1,18	0,61	1,18	Modelo 1D	1600	13	0,12	0,69	0,12
Modelo 1C	1250	14	0,39	0,47	0,39	Modelo 1D	1600	14	0,40	0,43	0,40
Modelo 1C	1250	15	0,64	0,56	0,64	Modelo 1D	1600	15	0,56	0,55	0,56
Modelo 1C	1250	16	0,45	0,47	0,45	Modelo 1D	1600	16	0,41	0,47	0,41
Modelo 1C	1250	17	0,33	0,45	0,33	Modelo 1D	1600	17	0,34	0,46	0,34
Modelo 1C	1250	18	0,57	0,54	0,57	Modelo 1D	1600	18	0,11	0,58	0,11
Modelo 1C	1250	19	0,39	0,47	0,39	Modelo 1D	1600	19	0,38	0,48	0,38
Modelo 1C	1250	20	0,65	0,47	0,65	Modelo 1D	1600	20	0,70	0,52	0,70
Rendimiento promedio			0,57	0,52	0,57	Rendimiento promedio			0,47	0,55	0,47

Tabla No.4.14. Validación cruzada del modelo No.1. con $k = 20$. (creación propia)

Después de realizar la validación cruzada se obtuvieron los siguientes resultados del modelo No.1. en diferentes épocas con $k = 20$, se sugiere en este caso utilizar el modelo 2C (ver tabla 4.15).

Modelos	Fold=1,2,...,20	Épocas	Loss	MAE	MSE
Modelo 1A	Promedio	600	0,58	0,54	0,58
Modelo 1B	Promedio	800	0,57	0,52	0,57
Modelo 1C	Promedio	1.250	0,57	0,52	0,57
Modelo 1D	Promedio	1.600	0,47	0,55	0,47

Tabla No.4.15. Resultados finales de la validación cruzada del modelo No.1. con $k = 20$. (creación propia)

4.1.8. Análisis validación cruzada del modelo No.1.

Este análisis solo comprende la validación cruzada de modelo No.1. con dos entradas: la imagen y la sombra.

Resumen de resultados de la validación cruzada con $fold = 5, 10$ y 20

Estos valores corresponden a las pruebas de validación del modelo No.1. que es la red neuronal convolucional para regresión con un dato de entrada: la sombra. En las tablas 4.16, 4.17, 4.18 resume el proceso de validación cruzada.

Con $fold = 5$

Modelo	Épocas	Fold	Pérdida del Modelo (Loss)	MAE (Error Absoluto Medio)	MSE (Error Cuadrático Medio)
Modelo 1A	600	1,...,5	0,73	0,55	0,73
Modelo 1B	800	1,...,5	0,80	0,57	0,80
Modelo 1C	1.250	1,...,5	0,75	0,54	0,75
Modelo 1D	1.600	1,...,5	0,75	0,55	0,76

Tabla No.4.16. Resumen de validaciones cruzadas del modelo No.1. con $k = 5$. (creación propia)

Con $fold = 10$

Modelo	Épocas	Fold	Pérdida del Modelo (Loss)	MAE (Error Absoluto Medio)	MSE (Error Cuadrático Medio)
Modelo 1A	600	1,...,10	0,58	0,52	0,58
Modelo 1B	800	1,...,10	0,58	0,52	0,58
Modelo 1C	1.250	1,...,10	0,60	0,52	0,60
Modelo 1D	1.600	1,...,10	0,59	0,52	0,59

Tabla No.4.17. Resumen de validaciones cruzadas del modelo No.1. con $k = 10$. (creación propia)

Con $fold = 20$

Modelo	Épocas	Fold	Pérdida del Modelo (Loss)	MAE (Error Absoluto Medio)	MSE (Error Cuadrático Medio)
Modelo 1A	600	1,...,20	0,58	0,54	0,58
Modelo 1B	800	1,...,20	0,57	0,52	0,57
Modelo 1C	1.250	1,...,20	0,57	0,52	0,57
Modelo 1D	1.600	1,...,20	0,47	0,55	0,47

Tabla No.4.18. Resumen de validaciones cruzadas del modelo No.1. con $k = 20$. (creación propia)

Análisis:

Con el $fold = 5$, los valores de pérdida varían de 0,73 a 0,80, siendo el modelo 1^a el que la menor pérdida. El MAE va de 0,54 a 0,57 con el modelo 1C teniendo el valor más bajo de 0,54. El MSE y la pérdida tienen un valor de 0,73 en el modelo 1A. Con el $fold = 10$, la pérdida es bastante uniforme variando de 0,58 a 0,60. El MAE es constante en 0,52 para todos los modelos lo que indica un rendimiento similar El MSE varía de 0,58 a 0,60, siguiendo de cerca a la pérdida Con un $fold = 20$, la pérdida del modelo es consistente con valores de 0,47 y 0,58. Se analizó que el Modelo 1A es el seleccionado por el rendimiento en la validación cruzada, y también por el rendimiento a lo largo de las tres configuraciones de la validación cruzada, con un $fold$ de 5, 10 y 20 respectivamente. Esta consistencia es del Modelo 1A, que es probablemente la mejor elección para generalizar bien nuevos datos. Su rendimiento en todos los escenarios de la validación cruzada realizados lo hace el mejor de los cuatro modelos.

4.2. Análisis integral del modelo No.2. y No.1.

En esta investigación mediante la experimentación las redes neuronales convolucionales (CNN) con regresión se utilizaron para el diseño la extracción automática de características en imágenes, lo que las convierte en candidatas ideales para tareas de regresión relacionadas con la interpretación de imágenes. En el caso de los dos modelos propuestos, la metodología incremental adoptada permitió un desarrollo iterativo, donde cada etapa se centró en refinar y mejorar la *performance* del modelo, adaptándose a la retroalimentación y los resultados obtenidos.

El primer modelo (Modelo No.1.), con una única entrada, es una solución inicial al problema, mientras que el segundo, al incorporar dos entradas (la figura y su sombra), añadió una dimensión adicional al análisis, potencialmente capturando relaciones más complejas en los datos. A través de las iteraciones, se evidenció la importancia de un diseño cuidadoso, la optimización y el ajuste de la arquitectura para lograr predicciones precisas. En resumen, las $CNNs$, cuando se aplican adecuadamente en tareas de regresión, tienen el potencial de transformar imágenes en valiosa información cuantitativa, como es la estimación de la altura a partir de una sombra en este contexto.

La distribución de las imágenes del *dataset* como se ha indicado anteriormente es el mismo para los dos modelos al igual que el *DataFrame*. Las pruebas de validación y métricas de rendimiento son relevantes para evaluar los modelos No.1. y No.2. Con estos se los entrenó en 600, 800, 1.250 y 1.600 épocas respectivamente. Es entonces que se aplicaron diferentes métricas y métodos para observar el comportamiento de los modelos a lo largo del entrenamiento, el rendimiento, permitiendo una comprensión más profunda de los mismos. En ellos se identificaron valores atípicos, patrones de error, entre otros que son fundamentales para la optimización y mejora continua de los modelos.

Los siguientes son los resultados que obtuvieron al aplicar estas diferentes técnicas,

métricas y gráficos que corresponden a los dos modelos con 1.250 épocas:

4.2.1. Resultados gráficos y cuantitativos: modelo No.2. con 1.250 épocas

A continuación, se relacionan diferentes técnicas, métricas y gráficos orientados a evaluar los resultados del modelo No.2. de una forma más completa y exhaustiva.

- a. Error cuadrático medio (*Mean Square Error, MSE*):
- b. Error absoluto medio (*Mean Absolute Error, MAE*)
- c. Pérdida (*Loss*)
- d. Coeficiente de determinación (*Coefficient of determination, R²*)
- e. Coeficiente de correlación de Pearson (*Pearson Correlation Coefficient, r*):
- f. Gráfica de la curva de aprendizaje
- g. Gráficos de puntos (*Scatter Plots*) para el *MAE* y el *MSE*
- h. Gráficos de densidad de las métricas *MAE* y *MSE*
- i. Gráfico de dispersión
- j. Histograma
- k. Gráfico de caja (*BoxPlot*)
- l. Gráfica de residuos (*Residual Sum of Squares, RSS*)

a. Error cuadrático medio (*MSE*):

Es una métrica comúnmente utilizada en problemas de regresión para cuantificar el error entre los valores reales y los valores predichos. Para calcularlo, toma la diferencia entre cada valor predicho y el valor real correspondiente, lo cuadra, y luego toma el promedio de todos estos valores cuadrados. En general, cuanto más pequeño es el *MSE*, mejor es el rendimiento del modelo. Sin embargo, debido a que los errores se elevan al cuadrado antes de que se promedien, el *MSE* da más peso a los errores grandes (ver figura 4.1).

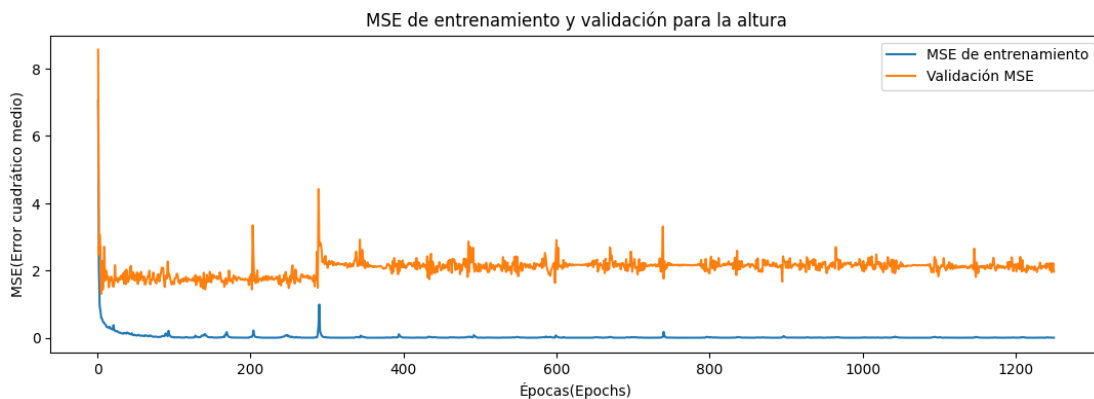


Figura 4.1. Entrenamiento y validación *MSE* (Error Cuadrático Medio) para altura modelo No.2. (creación propia)

El valor obtenido para el MSE es del cuadrado de los errores entre las predicciones del modelo y de la altura real de las formas es 0.46. Es así como el MSE da más peso a los errores más grandes debido a su naturaleza cuadrática. En este caso, un valor de MSE de 0.46 sugiere que existe una variabilidad en los errores, incluidos algunos errores que son más grandes y afectan el promedio. El MSE de 0.46 indica la magnitud promedio de los errores cuadráticos en las predicciones (ver figura 4.2).

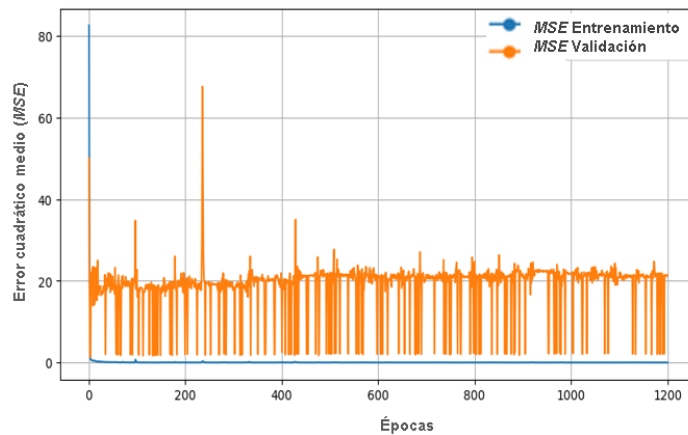


Figura 4.2. Métricas MSE cambia a lo largo de las épocas. (creación propia)

El MSE de 0.46 también proporciona información sobre la variabilidad y distribución de los errores

b. Error absoluto medio (MAE):

El MAE es otra métrica utilizada en problemas de regresión. Para calcularlo, toma la diferencia entre cada valor predicho y el valor real correspondiente, toma el valor absoluto de esta diferencia, y luego toma el promedio de todos estos valores absolutos. A diferencia del MSE y del $RMSE$, el MAE da igual peso a todos los errores, sin importar su tamaño (ver figura 4.3).

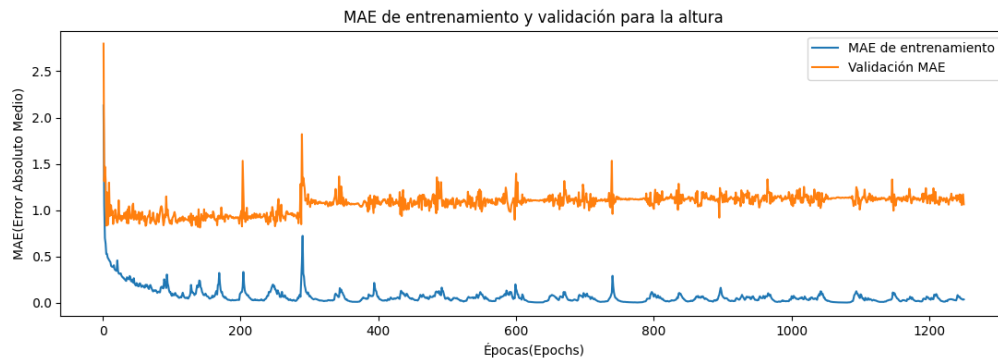


Figura 4.3. Entrenamiento y validación MAE (Error Absoluto Medio) para altura Modelo No.2. (creación propia)

El valor obtenido del *MAE* es de 0.24. Esto significa que, en promedio, las predicciones del modelo tienen un error absoluto de 0.24 unidades en la predicción de la altura. En otras palabras, en promedio, las predicciones del modelo se desvían en 0.24 unidades de la altura real de los objetos. El *MAE* de 0.24 indica el tamaño promedio de los errores absolutos en las predicciones. Un *MAE* de 0.24 es relativamente bajo y sugiere que las predicciones tienden a estar cercanas a la altura real de los objetos (ver figura 4.4).

c. Pérdida (Loss)

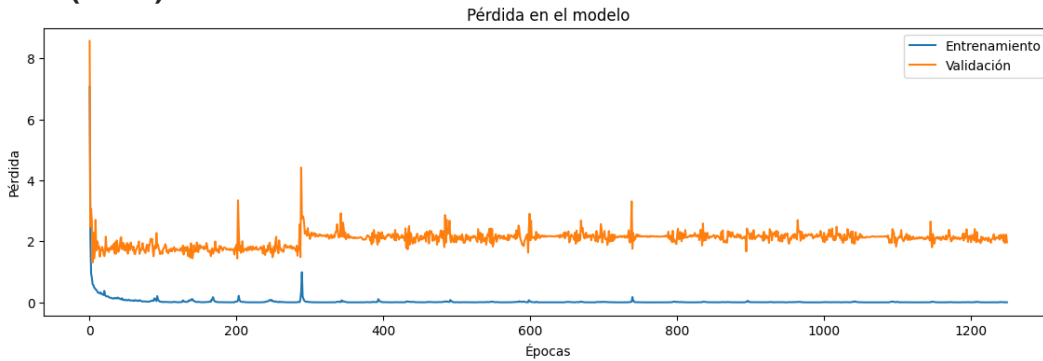


Figura 4.4. Pérdida en entrenamiento y validación. Modelo No.2. (creación propia)

d. Coeficiente de determinación R^2 (Coefficient of determination, R^2)

El coeficiente de determinación, también conocido como R cuadrado, es una medida de cuánto de la variabilidad en la variable objetivo puede ser explicado por el modelo. Un valor de R^2 de 1 significa que el modelo puede explicar toda la variabilidad, mientras que un valor de R^2 de 0 significa que el modelo no puede explicar ninguna de la variabilidad. En general, cuanto más alto sea el valor de R^2 , mejor será el modelo. Sin embargo, es posible que un modelo con un alto valor de R^2 todavía haga predicciones pobres si hay errores sistemáticos en las predicciones. El valor del coeficiente de determinación R^2 en este caso es de 0.92 del modelo No.2 (ver figura 4.5).

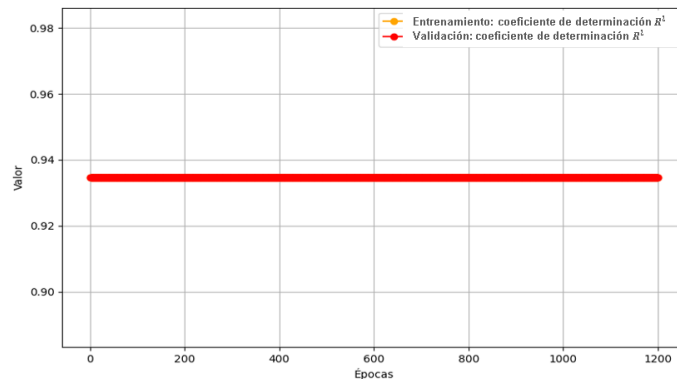


Figura. 4.5. Coeficiente de determinación R^2 en función de las épocas modelo No.2. (creación propia)

El coeficiente de determinación R^2 varía entre 0 y 1, donde 1 indica que el modelo puede explicar toda la variabilidad de los datos y 0 indica que el modelo no puede explicar ninguna variabilidad. Un valor de R^2 de 0.93 es bastante alto, lo que sugiere que el modelo tiene una buena capacidad para explicar la variabilidad en los datos (ver tabla 4.19).

El valor del coeficiente de determinación R^2 es:

Modelos	Valores (R^2)
150	0,94
250	0,93
300	0,94
400	0,93
500	0,93
600	0,93
1150	0,94
1200	0,93
1250	0,93

Tabla No.4.19. Valores de R^2 para la visualización rápida de cuál modelo. (creación propia)

e. Coeficiente de correlación de *Pearson* (*Pearson Correlation Coefficient*, r)

El coeficiente de correlación de *Pearson* se puede utilizar para evaluar que tanto las predicciones del modelo se ajustan con los valores reales.

- Si el coeficiente es cercano a 1, significa que las predicciones del modelo tienden a aumentar cuando los valores reales aumentan (y viceversa).
- Si el coeficiente es cercano a -1 , significa que las predicciones del modelo tienden a disminuir cuando los valores reales aumentan (y viceversa).
- Si el coeficiente es cercano a 0, significa que las predicciones del modelo no tienen una tendencia clara a aumentar o disminuir en relación con los valores reales.

Un coeficiente de correlación de *Pearson* de 0.86 es alto y sugiere que hay una fuerte relación lineal positiva entre las predicciones y los valores reales.

f. Gráfica de la Curva de Aprendizaje

Después de la caída inicial, la pérdida de entrenamiento parece estabilizarse y fluctúa ligeramente, lo cual es un comportamiento normal ya que el modelo busca minimizar la pérdida en el espacio de parámetros (ver figura 4.6).

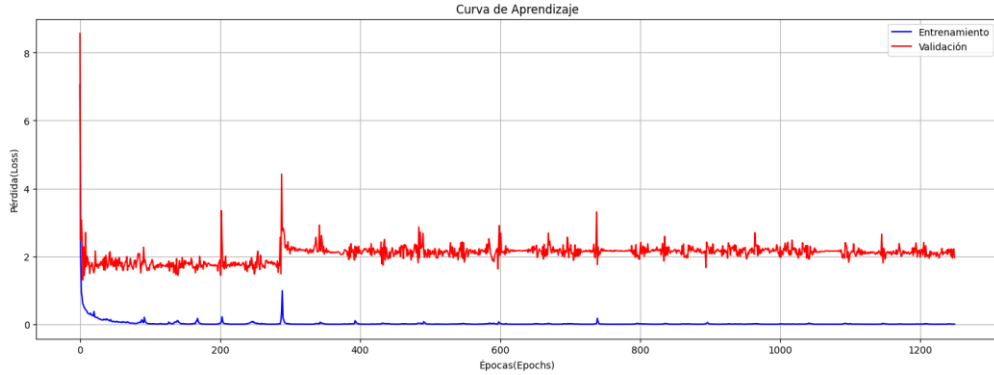


Figura 4.6. Curva de aprendizaje del modelo No.2. (creación propia)

g. Gráficos de Puntos (*Scatter Plots*) para el *MAE* y el *MSE*

Existe una correlación positiva visible entre *MAE* y *MSE*. Esto es esperado ya que ambas métricas tienden a aumentar o disminuir juntas; sin embargo, el *MSE* tiende a ser más alto que el *MAE* porque penaliza más los errores grandes al elevar al cuadrado las diferencias antes de promediarlas. La gráfica sugiere que el modelo puede estar experimentando sobreajuste después de un cierto número de épocas, y que podría ser beneficioso detener el entrenamiento más temprano. Es importante indicar que mientras el *MAE* proporciona una idea de la magnitud promedio del error, el *MSE* puede dar más información sobre la variabilidad de esos errores (ver figura 4.7).

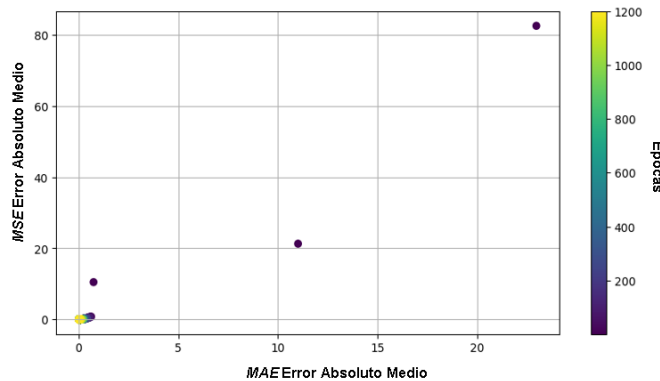


Figura 4.7. Gráficos de Puntos (*Scatter Plots*) para *MAE* y *MSE*. modelo No.2. (creación propia)

h. Gráficos de densidad de las métricas *MAE* y *MSE*.

La distribución del *MAE* del entrenamiento es muy puntiaguda y centrada cerca de cero, lo que sugiere que el modelo tiene un error muy bajo en el conjunto de entrenamiento. La curva puntiaguda indica que la mayoría de las predicciones del modelo

son muy precisas, con errores cercanos a cero. Se observa que tiene un error significativamente más bajo en el conjunto de entrenamiento en comparación con el conjunto de validación, lo que sugiere que el modelo puede estar sobreajustado y que podría beneficiarse de una estrategia para mejorar su capacidad de generalización (ver figura 4.8).

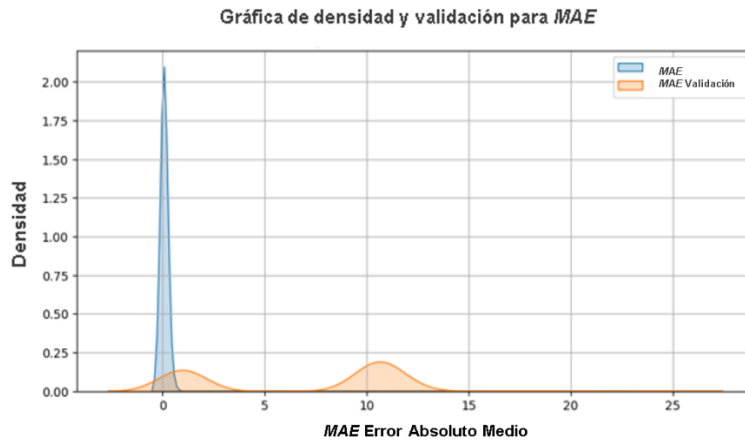


Figura 4.8. Gráficos de densidad y validación MAE modelo No.2. (creación propia)

i. Gráfico de dispersión

Para visualizar las diferencias entre las alturas reales y las alturas predichas, se creó un gráfico de dispersión donde el eje x representa las alturas reales y el eje y representa las alturas predichas por el modelo. Un modelo perfecto se reflejaría como una línea diagonal en este gráfico (ver figura 4.9).

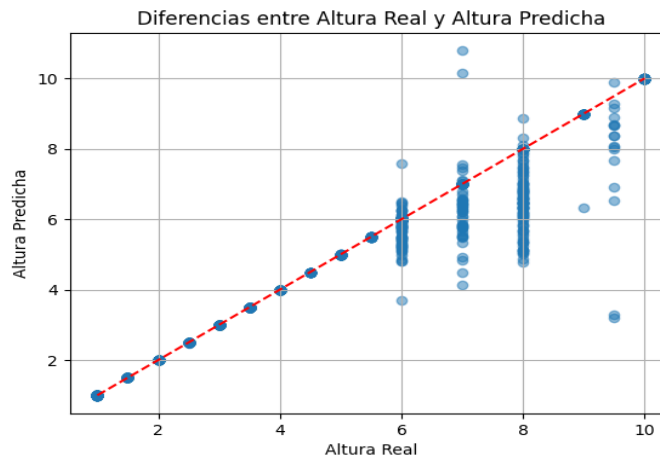


Figura 4.9. Gráfico de dispersión del modelo No.2. (creación propia)

El modelo parece predecir con una precisión razonablemente buena para un rango de alturas, pero su rendimiento disminuye a medida que las alturas reales aumentan. Hay una variabilidad considerable en las predicciones para alturas reales más allá del valor

6. Los puntos se dispersan más verticalmente a medida que aumenta la altura real, lo que indica que el modelo tiene más dificultad para hacer predicciones precisas en este rango.

j. Histograma

Esta gráfica muestra en el eje horizontal la diferencia en unidades de altura (real menos predicha), y el eje vertical muestra la frecuencia de esas diferencias entre los datos analizados. La mayoría de las diferencias se concentran alrededor del cero, lo que indica que, para la mayoría de las predicciones, la altura predicha por el modelo es muy cercana a la altura real. Esto es un buen indicador de precisión del modelo (ver figura 4.10).

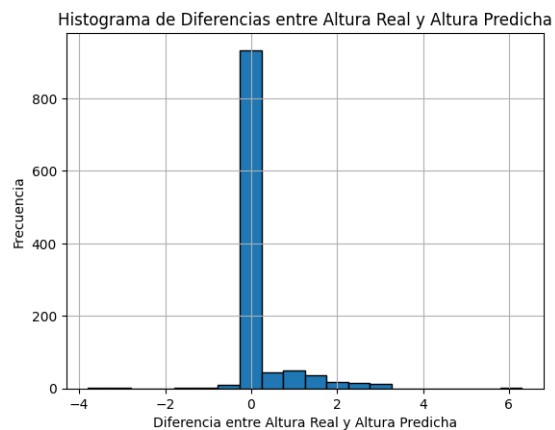


Figura 4.10. Histograma de las diferencias entre las alturas reales y las alturas predichas modelo No.2. (creación propia)

La variabilidad de las diferencias, basada en el ancho de la distribución, parece ser bastante baja, lo que sugiere que el modelo tiene una buena precisión general. El histograma sugiere que el modelo es bastante confiable y consistente en sus predicciones, con la mayoría de las predicciones teniendo un pequeño error respecto a los valores reales.

k. Gráfico de caja

Crear un gráfico de caja para visualizar los errores de predicción en función del tipo de figura: un este gráfico de caja, cada caja representa un grupo de figuras de un tipo específico (cilindros, cubos o esferas), y la altura de la caja muestra la distribución de los errores de predicción dentro de ese grupo. Este tipo de visualización permitirá comparar cómo se comportan los errores de predicción entre los diferentes tipos de figuras (ver figura 4.11).

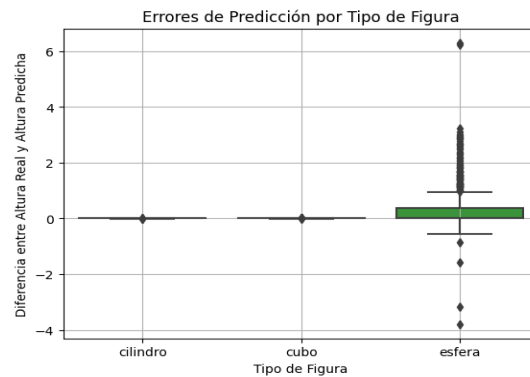


Figura 4.11. Gráfico de caja del modelo No.2. Errores de predicción por Tipo de figura (creación propia)

El modelo predice con alta precisión para cilindros, con buena precisión para cubos, y con precisión razonable para esferas, aunque con algunos errores significativos. En cuanto al cilindro la caja para el este es solo una línea, lo que indica que la mayoría de las predicciones son exactas o muy cercanas al valor real (la mediana está en cero y no hay variabilidad significativa). No hay bigotes, lo que sugiere que no hay valores atípicos y que todas las predicciones caen dentro de un rango muy estrecho. No hay bigotes, lo que sugiere que no hay valores atípicos y que todas las predicciones caen dentro de un rango muy estrecho.

I. Gráfica de Residuos (*Residual Sum of Squares, RSS*)

La gráfica de residuos ayudará a identificar patrones en los errores del modelo. Idealmente, los residuos deben estar distribuidos aleatoriamente alrededor de la línea horizontal en $y = 0$, sin patrones discernibles. Si hay patrones o estructuras en la gráfica de residuos, podría indicar problemas con el modelo, como heteroscedasticidad o no linealidad. En la figura 4.12. se observa una concentración de residuos en la parte superior derecha del gráfico, lo que indica un patrón en los errores de predicción. Si hay algún patrón en los residuos (por ejemplo, una tendencia clara en lugar de un patrón aleatorio), esto podría sugerir que el modelo no está capturando alguna característica de los datos.

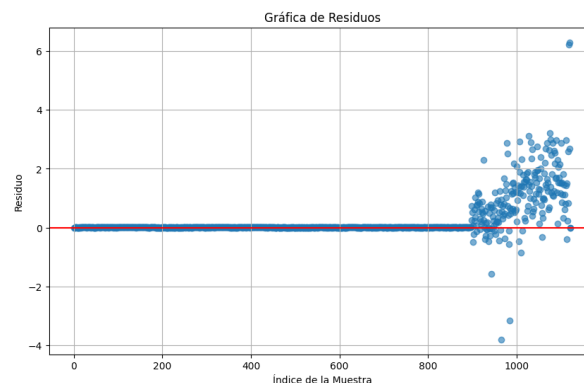


Figura 4.12. Gráfico de residuos del modelo No.2. (creación propia)

Los residuos están distribuidos simétricamente alrededor de la línea horizontal en $y = 0$, esto es una buena señal. Pero el modelo tiene un sesgo sistemático donde varios puntos individuales que se desvían significativamente del resto. La dispersión de los residuos aumenta, lo cual podría indicar heteroscedasticidad (la varianza de los errores no es constante). La agrupación de los residuos y el aumento de la dispersión pueden ser signos de sobreajuste o subajuste, si el modelo está sobreajustado a una parte específica del conjunto de datos y no generaliza bien a todo el conjunto. Los residuos no se distribuyen al azar alrededor de la línea horizontal en cero (en rojo), lo que sugeriría que el modelo está bien ajustado. En cambio, se observa una concentración de residuos en la parte superior derecha del gráfico, lo que indica un patrón en los errores de predicción.

4.2.2. Resultados gráficos y cuantitativos: modelo No.1. con 1.250 épocas

A continuación, se indican las técnicas, métricas y gráficos que se utilizaron para evaluar el modelo No.1. que se utilizaron para evaluar los resultados generados de una forma más completa y exhaustiva.

- a. Error cuadrático medio (*Mean Square Error, MSE*):
- b. Error absoluto medio (*Mean Absolute Error, MAE*)
- c. Pérdida (*Loss*)
- d. Coeficiente de determinación (*Coefficient of determination, R^2*)
- e. Coeficiente de correlación de *Pearson* (*Pearson Correlation Coefficient, r*)
- f. Gráfica de la curva de Aprendizaje
- g. Gráficos de puntos (*Scatter Plots*) para el *MAE* y el *MSE*
- h. Gráficos de densidad de las métricas *MAE* y *MSE*
- i. Gráfico de dispersión
- j. Histograma
- k. Gráfico de caja (*BoxPlot*)
- l. Gráfica de residuos (*Residual Sum of Squares, RSS*)

a. Error cuadrático medio (MSE):

Como el *MSE* es una métrica con la cual se puede medir el promedio de las diferencias al cuadrado entre las predicciones del modelo y los valores reales. Similar al *MAE*, el *MSE* cuantifica la calidad de las predicciones del modelo. En este caso la predicción de la altura a partir de sombras, con el *MSE* dice que tan cerca están las predicciones del modelo de las alturas reales, pero penalizando más las predicciones que están lejos del valor real. Al igual que con el *MAE*, un valor de *MSE* más bajo indica una mejor calidad de las predicciones. En la figura 4.13 el *MSE* es muy alto, lo que indica que el modelo no estaba realizando buenas predicciones, luego hay una disminución en el entrenamiento del *MSE* lo que es una señal, de que el modelo está aprendiendo y posteriormente tanto el *MSE* en el entrenamiento, como el *MSE* en la validación. Aquí el entrenamiento como

la validación del *MSE* se mantienen similares, lo que indica que el modelo no está sobreajustado. De igual forma el bajo nivel de la validación del *MSE* a través de las épocas indica que el modelo es estable y generaliza bien nuevos datos que no ha visto en el entrenamiento.

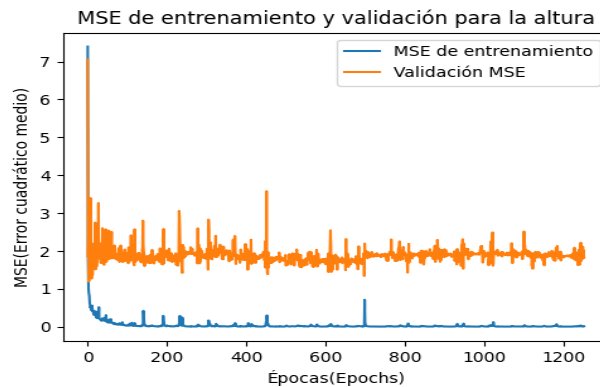


Figura 4.13. Error cuadrático medio (*MSE*) en entrenamiento y validación en función de las épocas del modelo No.1. (creación propia)

Según la gráfica se puede inferir que la *CNN* es capaz de estimar la altura de las figuras geométricas a partir de su sombra, con un error razonablemente bajo y constante. De igual forma el *MSE* de validación se mantiene bajo y cerca del *MSE* de entrenamiento, con esto se puede decir que el modelo no solo está aprendiendo particularidades del conjunto de entrenamiento, sino que también generaliza bien nuevos datos. El modelo está indicando buena capacidad predictiva, e igualmente el hecho de que el error se estabilice a lo largo de las épocas, sugiere que el modelo probablemente no mejorará significativamente con más entrenamiento.

b. Error absoluto medio (MAE)

Es importante indicar que esta métrica estima el promedio absoluto de las diferencias entre las predicciones del modelo y los valores reales. Esto se refiere para el modelo de regresión que predice la altura de una forma u objeto a partir de la sombra de una imagen 2D, cuánto o que tanto se desvían en promedio, las predicciones que estima el modelo de las alturas reales de los objetos, en este caso están en función de sus sombras. Lo ideal es que cuanto menor sea el valor de *MAE*, mejor, lo que indica que las predicciones del modelo son más cercanas a los valores reales.

En la figura 4.14 el *MAE* de *entrenamiento* y el *MAE* de *validación*, se comparan las métricas de *Error Absoluto Medio (MAE)* para el conjunto de entrenamiento y el conjunto de validación. Se observa que ambos gráficos disminuyen a lo largo de las épocas y están cerca uno del otro, lo cual indica que el modelo está generalizando bien y no está sobreajustado. En la gráfica se observa que en las primeras épocas el *MAE* de entrenamiento y el de validación disminuyen, lo que es un indicador de que el modelo está aprendiendo, y de que su precisión en la predicción de los datos mejora.

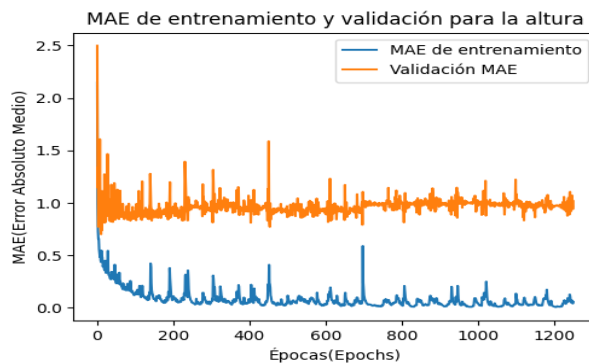


Figura 4.14. Entrenamiento y validación *MAE* vs épocas del modelo No.1. con 1.250 épocas. (creación propia)

En unas épocas más adelante el *MAE* se estabiliza, lo cual es un indicador de que el modelo está alcanzando un punto de convergencia, y que no habrá mejoras significativas. La gráfica en este caso muestra que el *MAE* del conjunto de validación es mayor que el *MAE* del conjunto de entrenamiento, que es común, dado que los modelos tienden a tener mejor desempeño con datos que ya han visto o conocen (entrenamiento), en comparación con los datos que aún son nuevos para él. En la gráfica las dos líneas se mantienen paralelas, lo que puede indicar que el modelo no está sobreajustado. Esta información permite entender si el modelo está teniendo problemas con el manejo de los errores pequeños o con errores grandes, y si el tipo de error (absoluto o cuadrático) está teniendo un impacto más significativo en la predicción. Con este gráfico se evalúa la relación entre el *MAE* y *MSE* en términos de magnitud. Si la barra del *MSE* es más alta que la barra de *MAE*, eso indica que el modelo está penalizado más por los errores más grandes (el cuadrado en el cálculo del *MSE*), lo que en este caso indica que hay algunas predicciones muy inexactas en el conjunto de datos. Ahora bien, si la diferencia entre las dos barras es pequeña, esto podría indicar que el modelo está teniendo un rendimiento similar en términos de errores pequeños y grandes, lo que significa que el modelo está equilibrado y que no es el caso aquí.

c. Pérdida

La pérdida es una medida que indica que tan bien un modelo de aprendizaje realiza predicciones comparados con los valores reales en el conjunto de datos de entrenamiento. Estos pueden variar según la función de pérdida utilizada (por ejemplo, *MSE*, *MAE*, etc.). En un gráfico de pérdida a lo largo de las épocas, se busca una disminución constante de la pérdida. Si la pérdida disminuye en las primeras épocas y luego se estabiliza o aumenta, puede indicar que el modelo está sobreajustando los datos de entrenamiento. En la figura 4.15 se observa un pico muy alto de la pérdida de entrenamiento, posteriormente la pérdida de entrenamiento disminuye rápidamente lo que indica que el modelo está aprendiendo y mejorando su precisión en la predicción de alturas. De igual forma a lo largo del entrenamiento la pérdida de validación tiene variaciones, pero permanece en un rango bajo, lo que indica que generaliza bien nuevos datos. En general el modelo tiene un buen rendimiento con una pérdida reducida y

estable.

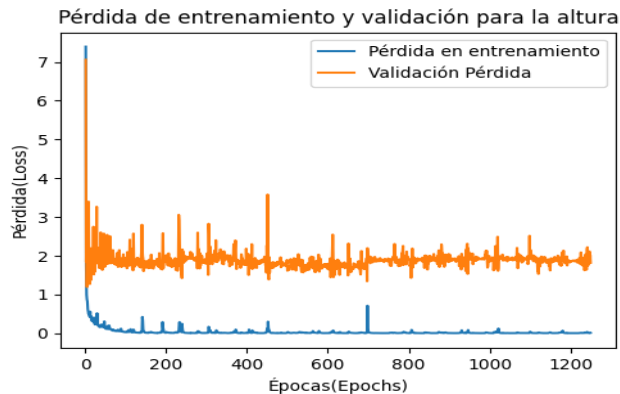


Figura 4.15. Pérdida del modelo en el entrenamiento y la validación, modelo No.1. (creación propia)

Estas métricas se refieren al *MAE* (*Error Absoluto Medio*), la pérdida (*Loss*) y el *MSE* (*Error Cuadrático Medio*) para la altura en un modelo de regresión. Estas métricas y su evolución a lo largo de las épocas proporcionan información valiosa sobre la calidad de las predicciones del modelo, su capacidad para generalizar y el posible sobreajuste.

d. Coeficiente de determinación (R^2)

Para comprender la interpretación de este indicador respecto del modelo propuesto en esta investigación, se puede afirmar que el R^2 es un indicador de la calidad de un modelo estadístico, que refleja la proporción de la varianza para la variable dependiente. En la experimentación se obtuvo un valor de R^2 de 0.95 o 95% que es alto e indica que el modelo puede explicar (capacidad predictiva) el 95% de la variabilidad observada en los datos. Sin embargo, es necesario mencionar que importante se utilizaron otras métricas adicionales para determinar que el modelo sea óptimo. En general, si el valor de R^2 esta cerca de 1 mejor será el modelo, ya que un R^2 de 1 indica que el modelo puede explicar toda la variabilidad de los datos (ver tabla 4.16). La variabilidad se refiere a que el modelo es capaz de explicar el 95% de la variabilidad de la altura de la figura en este caso. Consecuentemente esto indica que el modelo No.1. tiene una buena precisión en las predicciones, sin embargo, aunque el R^2 sea alto lo cual es bueno, no fue el único criterio que se utilizó para evaluar el rendimiento del modelo. Por esta razón se optó por es realizar validaciones adicionales, como la validación cruzada (*cross-validación*), que permite obtener métricas de rendimiento como el error absoluto medio (*MAE*), el error cuadrático medio (*MSE*), entre otros. También se consideró que el R^2 muy alto también podría indicar una señal de que el modelo tiene sobreajuste. Por esto también se recomienda utilizar los resultados gráficos, que son especialmente útiles para comprender cómo el rendimiento del modelo evoluciona con el tiempo y determinar si está mejorando o estabilizándose. Para este caso, el valor del coeficiente de determinación en 1.250 épocas R^2 es igual a 0,93, (ver tabla 4.20).

Épocas	Error Absoluto Medio (MAE)	Error Cuadrático Medio MSE	Coefficiente de determinación (R^2)
500	0,41	0,54	0,94
600	0,41	0,54	0,93
1150	0,27	0,43	0,94
1200	0,27	0,43	0,93
1250	0,20	0,34	0,93

Tabla No. 4.20. Coeficiente de determinación R^2 , MAE y MSE del modelo No.1. en varias épocas. (creación propia)

Analizando los resultados para las diferentes épocas junto con MAE y el MSE y R^2 , se seleccionó el óptimo, uno por el valor de R^2 más alto como por los valores más bajos del MAE y del MSE, que indican menor error en las predicciones del modelo. En las 1.250 épocas el modelo alcanza un $R^2 = 0,93$ con un MAE = 0,20 y un MSE = 0,34 es un punto óptimo porque ofrece mejor capacidad predictiva con el menor error. Teniendo estos valores se continua con la valoración y análisis del modelo No.1. con más detalle.

e. Coeficiente de correlación de Pearson (r):

Este coeficiente se utiliza para evaluar que tan bien las predicciones del modelo se aproximan a los valores reales. Por lo tanto, si el coeficiente es cercano a 1, significa que las predicciones del modelo tienden a aumentar cuando los valores reales aumentan (y viceversa). De otro lado si el coeficiente es cercano a -1 , significa que las predicciones del modelo tienden a disminuir, cuando los valores reales aumentan (y viceversa). Si el coeficiente es cercano a 0, significa que las predicciones del modelo no tienen una tendencia clara, al aumentar o disminuir en relación con los valores reales (Kumar, A., Yin, B., Shaikh, A.M. *et al.*, 2022). Con en el proceso de experimentación con el modelo No.1., el resultado obtenido para el coeficiente de correlación de Pearson fue de 0,97 en las 1.250 épocas en el conjunto de prueba, e indica una fuerte relación lineal positiva entre las predicciones y los valores reales. Se observó también que a medida que el modelo se entrenó con más épocas, se obtuvo la mejora en r y los valores de MAE y MSE se mantuvieron bajos. (ver tabla 4.21).

Épocas	Error Absoluto Medio (MAE)	Error Cuadrático Medio MSE	Coefficiente de Correlación de Pearson (r)
500	0,41	0,54	0,96
800	0,41	0,54	0,96
1000	0,41	0,54	0,97
1250	0,27	0,43	0,97
1300	0,27	0,43	0,97
1600	0,20	0,34	0,97

Tabla No. 4.21. Coeficiente de correlación (r), MAE y MSE del modelo No.1. (creación propia)

Es así que, el coeficiente de correlación de *Pearson* puede ayudar a entender cómo se comportan las predicciones en relación con los valores reales desde una perspectiva lineal.

f. Gráfica de la Curva de Aprendizaje

En la figura 4.16 la gráfica representa la pérdida durante el entrenamiento y la validación en varias épocas, en este caso en 1.250 épocas. Se debe indicar que con estas dos líneas o curvas se puede detectar si el modelo este sobreajustado (*overfitting*) o subajustado (*underfitting*). Aquí, el sobreajuste ocurre cuando el modelo aprende muy bien los datos, pero no generaliza datos nuevos, y el subajuste cuando el modelo no aprende bien ni con los datos de entrenamiento Viering, T. and Loog, M. (2023). La la pérdida del error de predicción del modelo, donde los valores comenzaron muy altos y decrecieron rápidamente, lo cual indicó que el modelo aprendió los datos. Más adelante la estabilidad de la pérdida en las últimas épocas, indicó que el modelo alcanzó su punto de convergencia y que el modelo generaliza bien.

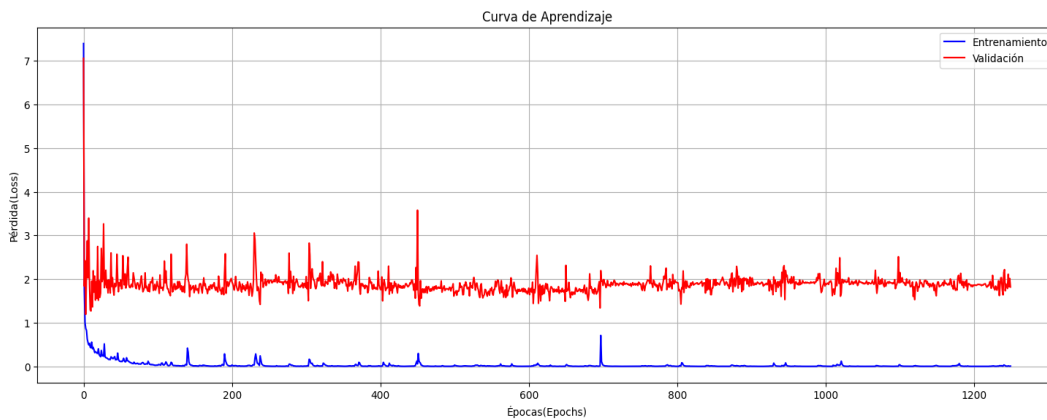


Figura 4.16. Gráfica de la curva de aprendizaje modelo No.1. (creación propia)

De igual forma esta curva decrece rápidamente y tiende a estabilizarse, lo cual también indicó que el modelo no solo está memorizando los datos de entrenamiento, sino que es capaz de generalizar bien datos nuevos. Adicionalmente se observó la convergencia de las dos curvas, que después de decrecer rápidamente, tienden a estabilizarse y continuar paralelas entre sí. La experimentación indicó que la diferencia del rendimiento del modelo entre los datos de entrenamiento y los de validación se mantuvo constante, lo que puede interpretarse que el modelo alcanzó un equilibrio entre el aprendizaje de los datos y la generalización de datos nuevos.

g. Gráficos de Puntos (Dot Plots) para el MAE y el MSE

En la figura 4.17 el *MAE* y el *MSE del entrenamiento* disminuyeron a medida que el número de épocas aumentó, lo que fue un indicador de que el modelo está mejorando en la precisión a medida que continúa aprendiendo de los datos de entrenamiento. Los

puntos se desplazaron hacia abajo y hacia la izquierda, lo que indicó una disminución de ambos errores, que es una señal positiva de aprendizaje. También mostró un valor atípico que es particularmente difícil para el modelo.

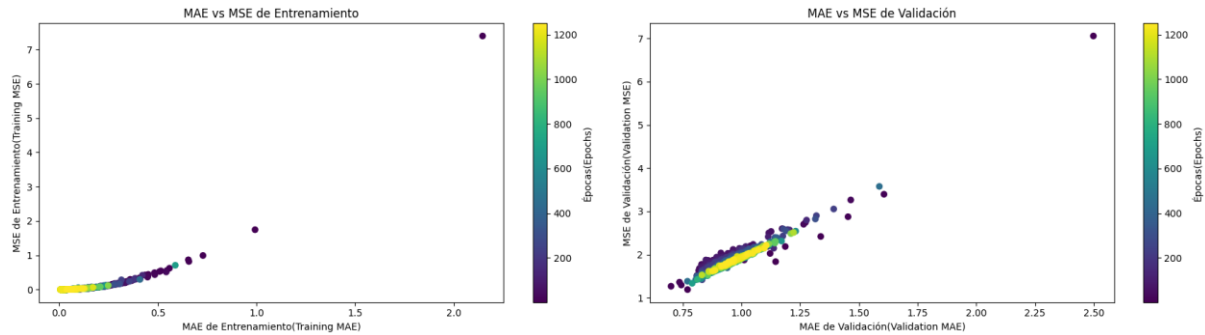


Figura 4.17. Gráfica de puntos del *MAE* (entrenamiento y validación) y *MSE* (entrenamiento y validación) modelo No.1. (creación propia)

En la gráfica del lado derecho los valores del *MAE* y *MSE* disminuyeron con más épocas, pero con una mayor dispersión de puntos. Esto indicó que, aunque el modelo este generalizando bien nuevos datos, hay algunas instancias donde el rendimiento varía más en relación con el conjunto de entrenamiento. De acuerdo a la gráfica el modelo mantiene una generalización consistente, según los puntos agrupados en el origen que reflejan errores más bajos (Schulz, A., Hinder, F. and Hammer, 2020). En general el modelo presentó una buena adaptación y aprendizaje progresivo, con tendencia la reducción de errores a medida que se incrementó por medio del entrenamiento. La validación indica que hay que investigar los valores atípicos para que el modelo pueda ser consistente en diferentes escenarios.

h. Gráficos de densidad de las métricas MAE y MSE

Se utilizó el gráfico de densidad con el objetivo de analizar la representación visual de la distribución de los errores calculados, por el error absoluto medio (*MAE*) y el error cuadrático medio (*MSE*) para un conjunto de entrenamiento y validación. En el gráfico de la izquierda (ver figura 4.18). la curva azul representa el *MAE* de los datos de entrenamiento y la gráfica naranja representa el *MAE* de los datos de validación. La curva azul (entrenamiento) y la curva naranja (validación) son muy puntiagudas y estrechas lo que indica que la mayoría de las predicciones tienen un error similar y posiblemente bajo. De otro lado la curva azul (entrenamiento) está ligeramente más alta y a la izquierda que la de la curva naranja (validación), lo que indica que el modelo tiene un menor error en el conjunto de entrenamiento. En el gráfico de la derecha, del mismo modo la línea azul representa el *MSE* de los datos de entrenamiento y la línea naranja representa el *MSE* de los datos de validación. Aquí, las gráficas pueden ser más anchas o tener colas más largas lo cual es un indicador de valores atípicos en los errores. En general mientras menores sean los valores del *MAE* y del *MSE*, mejor es la precisión en la predicción del modelo.

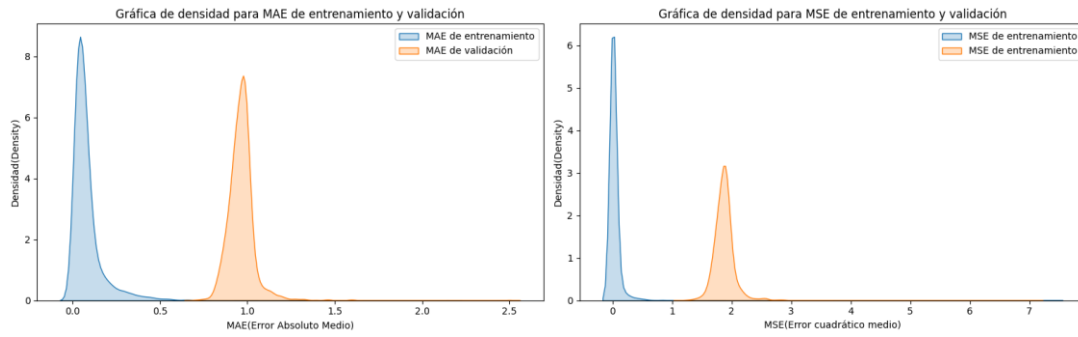


Figura 4.18. Gráficos de densidad para entrenamiento y validación de las métricas MAE y MSE modelo No.1. (creación propia)

En general los resultados de los gráficos indican que el modelo está bien ajustado, con una buena generalización y sin señales claras de sobreajuste. Igualmente tiene un buen rendimiento, ya que las métricas de error son bajas y las distribuciones de errores son concentradas.

i. Gráfico de dispersión (Scatter Plot)

La figura 4.19 corresponde a un gráfico de dispersión que permite visualizar las diferencias entre las alturas reales y las alturas predichas por el modelo de regresión, por tanto, se consideró importante la utilización del gráfico de dispersión para visualizar y analizar los resultados del modelo (Weissgerber, T.L. *et al.*, 2015). Entonces para iniciar con el proceso se agrega una línea diagonal en rojo con estilo de línea punteada, donde esta línea representa la línea ideal (línea de mejor ajuste) donde las alturas reales y las alturas predichas serían iguales. Si un punto cae sobre esta línea, significa que la predicción fue perfectamente precisa.

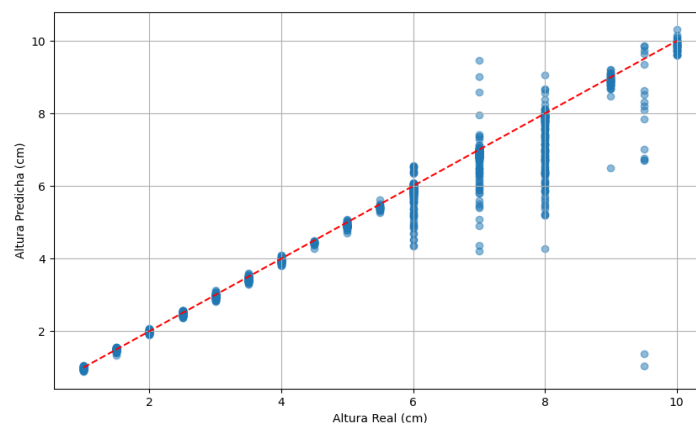


Figura 4.19. Gráfica de dispersión: comparación entre alturas reales y alturas predichas modelo No.1. (creación propia)

En esta gráfica se muestra cada punto como una relación entre la altura real y la altura

predicha, donde cada punto representa un ejemplo de los datos. Por tanto, los valores obtenidos (puntos) que están cerca de esta línea roja son las predicciones que están siendo precisas. También se obtuvieron puntos que están dispersos alrededor de la línea roja, lo que significa que existen diferencias entre las alturas reales y las alturas predichas. De igual forma algunos valores obtenidos se desviaron aún más de la línea roja, lo que indica errores en la estimación de la altura; estos son los que se alejaron significativamente de la línea diagonal en una dirección específica (arriba o abajo). En este caso el modelo propuesto en la investigación en algunos casos sobreestimó la medida de la altura, y con ello predijo valores que fueron consistente más altos que los valores reales, y en otros casos subestimó la medida de la altura y predijo valores que fueron consistente más bajos que los valores reales. En la gráfica también son visibles varios valores atípicos (anomalías) en las predicciones, estos están ubicados en la parte inferior derecha y fueron identificados como puntos que se desviaron significativamente de la línea diagonal. De otra parte, es notorio que se presentaron errores en las predicciones en los valores 6 al 10 (el eje de altura real) y en los valores 6 al 10 (eje altura predicha). Es así que las dispersiones de los puntos alrededor de la línea dan una idea de la calidad general de las predicciones del modelo.

j. Histograma

En este caso se seleccionó el histograma para obtener una comprensión más detallada del rendimiento del modelo, también se puede visualizar un histograma de las diferencias entre las alturas reales y las alturas predichas. Este permite ver cómo se distribuyen los errores de predicción. En la figura 4.21 sobre el eje x en el histograma, se representa la diferencia entre altura real y altura predicha, es decir los valores en el eje x , muestran cuánto se desvían las predicciones del modelo en comparación con los valores reales. De igual forma, en el histograma en el eje y , se muestra la frecuencia con la que ocurren diferentes rangos de diferencias, entre las alturas reales y las alturas predichas. Así, cuanto mayor sea la barra en un rango particular en el eje x , mayor será la frecuencia con la que se producen diferencias en ese rango. Se obtuvieron valores positivos, los cuales indicaron que las predicciones son más altas que las alturas reales, mientras que los valores negativos indicaron que las predicciones fueron más bajas que las alturas reales. También se observa la forma del histograma, donde la mayoría de las diferencias están concentradas alrededor de cero, lo que sugiere que las predicciones del modelo son generalmente precisas. también en la figura 4.20 hay una barra que es más alta que las demás, la cual se encuentra justo en el centro, lo que indica que la diferencia que más se presenta es muy pequeña, cercana a cero, siendo este, un buen indicador del rendimiento del modelo, ya que significa que, en la mayoría de los casos, la altura predicha por el modelo es muy similar a la altura real.

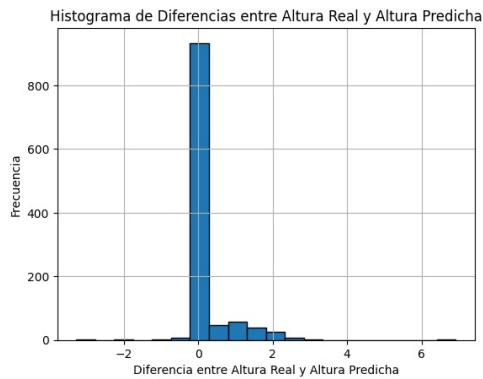


Figura 4.20. Histograma de diferencias entre las alturas reales y las alturas predichas modelo No.1. (creación propia)

Otro aspecto relevante en este caso es la forma del histograma, la cual está indicando como se distribuyen los errores de la predicción, en términos de sus diferencias con respecto a las alturas reales. En este caso el histograma está más sesgado hacia el lado derecho (valores positivos), lo que puede estar indicando que el modelo tiende a sobreestimar las alturas.

k. gráfico de caja (Box Plot)

Este tipo de gráfico se seleccionó para la visualización de los errores de predicción en función de la figura en este caso, de los cilindros, los cubos o las esferas, lo que permitió comparar cómo se comportan los errores de predicción entre los diferentes tipos de figuras. En este gráfico de caja, cada caja representa un grupo de figuras de un tipo específico (cilindros, cubos o esferas), y la altura de la caja muestra la distribución de los errores de predicción dentro de ese grupo. En la figura 4.21 en ambos gráficos el eje *y* etiquetado con “diferencia entre altura real y altura predicha”, representa la magnitud del error de predicción donde los valores positivos indican sobreestimación, es decir la altura predicha es mayor que la real y los valores negativos indican subestimación, es decir la altura predicha es menor que la real. Las cajas representan el rango intercuartílico (*IQR*), es decir, el 50% central de los datos de errores de predicción para cada figura. La línea dentro de la caja indica la mediana de los errores, que parece ser cero para todas las figuras, lo cual indica que la red no tiene sesgo sistemático hacia la sobreestimación o subestimación. La gráfica corresponde al modelo No.1. donde el cilindro en la distribución de los errores es bastante ajustada (rango intercuartílico estrecho *IQR*), donde esto sugiere que la red neuronal es consistentemente precisa en la estimación de la altura de los cilindros, con la mayoría de las predicciones concentradas alrededor de la mediana. Igualmente, la gráfica para el cilindro no muestra errores grandes en la sobreestimación para esta figura. Hay presencia de pocos valores atípicos que indica que hay algunas instancias donde la red comete errores significativos, pero estos son relativamente pequeños. Por otro lado, los bigotes no son muy largos y los valores atípicos están cerca indica que incluso los errores más extremos no están muy alejados de la mayoría de las predicciones.

De tal forma la red es bastante precisa, como se indica por la mediana igual a cero y IQR estrecho. Para el cubo la mediana está en cero, lo que sugiere que no hay sesgo sistemático hacia la sobreestimación en las predicciones de la altura.

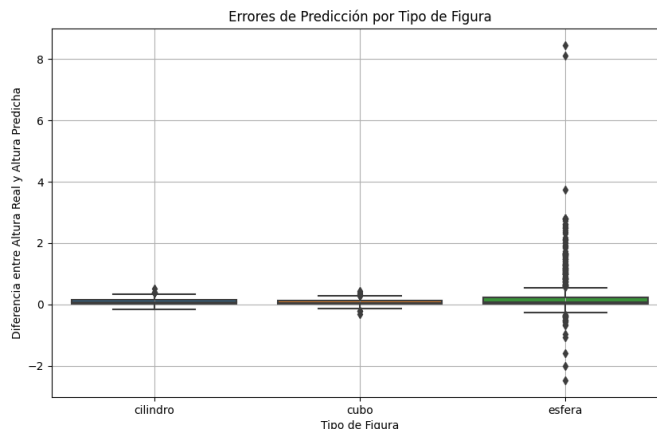


Figura 4.21. Errores de predicción por tipo de figura modelo No.1 con 1.250 épocas (creación propia)

El *IQR* más ancho que el del cilindro sugiere mayor o variabilidad en los errores, lo que es un indicador de que la forma del cubo presenta más desafíos para el modelo. En esta figura la gráfica muestra la mediana de errores del cubo exactamente en cero. Para la esfera la mediana en cero y un *IQR* amplio implica que, aunque en la mediana las predicciones son precisas, hay una variabilidad considerable en los errores de las predicciones para las esferas.

Los bigotes largos y la cantidad de valores atípicos sugieren que la red neuronal tiene casos en los que se desvían significativamente de la altura real. Esto debido a que las sombras de las esferas varían de forma más compleja con la altura, dada su forma simétrica en todas las direcciones. De igual forma los errores en la predicción de la altura de las esferas podría estar relacionado con las limitaciones del modelo para interpretar las sombras que proyectan, que podrían ser menos distintivas en términos de cambios de altura en comparación con formas más angulares como los cilindros o cubos.

En general, según la gráfica la red es bastante precisa para todas las figuras en términos de mediana de errores (mediana en cero para cilindro, cubo y esfera). Para cada figura se realizó el siguiente análisis:

- El cilindro tiene menos variabilidad en los errores y pocos valores atípicos, lo que indica un alto grado de precisión.
- El cubo tiene una mayor variabilidad en los errores, reflejada por el IQR más ancho, pero su mediana de errores está también en cero.
- La esfera muestra una variabilidad aún mayor en los errores del rango

intercuartílico (IQR más amplio) y una cantidad significativa de valores atípicos, lo que indica que, aunque la mediana de los errores es cero, hay casos en los que la red predice de forma inexacta la altura de las esferas.

I. Gráfica de Residuos

La gráfica en el eje x representa el índice de la muestra (número de observación o imagen en el conjunto de datos) y el eje vertical los residuos (diferencia entre los valores estimado por el modelo y lo valores reales). Los puntos representan los residuos individuales para cada muestra y la línea roja es la referencia para identificar la desviación de los residuos de un ajuste perfecto. Ahora bien, la distribución de los residuos en la figura 4.22 están concentrados cerca de la línea roja, con una dispersión que aumenta a medida que el índice de muestra aumenta. Esto indica que para algunas muestras la modelo está realizando predicciones con mayor error. También hay unos puntos que están lejos de la línea de residuo 0, especialmente uno que está muy alejado hacia el lado superior derecho de la gráfica (*outliers*).

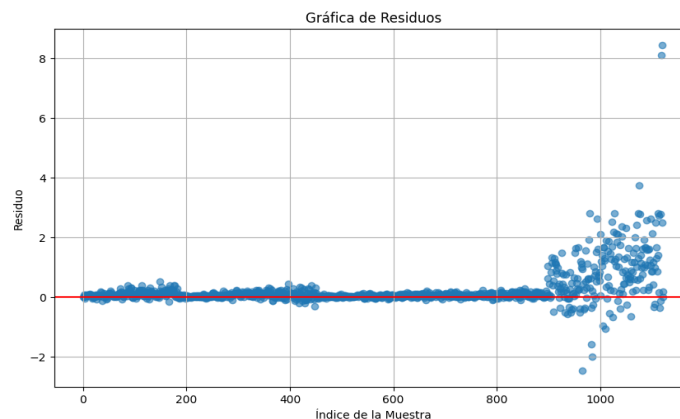


Figura 4.22. Gráfica de Residuos modelo No.1. (creación propia)

Como todos los residuos del modelo están alrededor de 0, indicaría que el modelo en general tiene un buen ajuste. Sin embargo, la precisión del modelo parece disminuir a medida que el índice de muestra aumenta, o que podría ser una señal de sobreajuste, subajuste o influencia de características no modeladas adecuadamente para estas muestras.

4.3. Resultados y tablas comparativas de los modelos No.2. y No.1.

A continuación, se muestra los resultados cuantitativos obtenidos a partir de los dos modelos desarrollados e implementados del modelo No.1. y del modelo No.2. con 600, 800, 1.000, 1.250 y 1.600 épocas respectivamente.

En las primeras pruebas a los modelos desarrollados en este trabajo que son el modelo No.1 y el modelo 2, y propuestos como una posible solución al problema planteado, se concluyó que ambos tienen la posibilidad de ser escalables, pueden ser implementados, optimizados y probados en otros modelos y arquitecturas de redes neuronales. El desarrollo de estos dos modelos es uno de los resultados iniciales muy importantes, y que para etapas posteriores van evolucionando y son los que a continuación se detallan. En estos se detallan aspectos como métricas de rendimiento, implementación de regularización $L1$ y $L2$ y el aumento de datos.

Para comprender mejor la experimentación y el proceso de análisis y evaluación de los modelos se elaboró la tabla 4.22 en la cual se describen las siguientes etapas:

Ítem	Tipo de modelo	Nombre	Descripción	Épocas	Aumento de datos	Regularización	
					<i>ImageDataGenerator</i>	$L1=0,01$	$L2=0,01$
a.	Modelo 1 (modelo base)	Modelo 1A	Modelo 1 sin regularización y sin aumento de datos	600	NA	NA	NA
		Modelo 1B	Modelo 1 sin regularización y sin aumento de datos	800	NA	NA	NA
		Modelo 1C	Modelo 1 sin regularización y sin aumento de datos	1.000	NA	NA	NA
	Modelo 2 (modelo base)	Modelo 2A	Modelo 2 sin regularización y sin aumento de datos	600	NA	NA	NA
		Modelo 2B	Modelo 2 sin regularización y sin aumento de datos	800	NA	NA	NA
		Modelo 2C	Modelo 2 sin regularización y sin aumento de datos	1.000	NA	NA	NA
b.	Modelo 1	M1_L1A_AUM	Modelo 1: una entrada con aumento de datos y regularización $L1$	600	Si	$L1=0,01$	NA
		M1_L1B_AUM	Modelo 1: una entrada con aumento de datos y regularización $L1$	800	Si	$L1=0,01$	NA
		M1_L1C_AUM	Modelo 1: una entrada con aumento de datos y regularización $L1$	1.000	Si	$L1=0,01$	NA
c.	Modelo 1	M1_L2A_AUM	Modelo 1: una entrada con aumento de datos y regularización $L2$	600	Si	NA	$L2=0,01$
		M1_L2B_AUM	Modelo 1: una entrada con aumento de datos y regularización $L2$	800	Si	NA	$L2=0,01$
		M1_L2C_AUM	Modelo 1: una entrada con aumento de datos y regularización $L2$	1.000	Si	NA	$L2=0,01$
d.	Modelo 1	M1_L1_L2A_AUM	Modelo 1: una entrada con aumento de datos y regularización $L1, L2$	600	Si	$L1=0,01$	$L2=0,01$
		M1_L1_L2B_AUM	Modelo 1: una entrada con aumento de datos y regularización $L1, L2$	800	Si	$L1=0,01$	$L2=0,01$
		M1_L1_L2C_AUM	Modelo 1: una entrada con aumento de datos y regularización $L1, L2$	1.000	Si	$L1=0,01$	$L2=0,01$
e.	Modelo 2	M2_L1A_AUM	Modelo 2: dos entradas con aumento de datos y regularización $L1$	600	Si	$L1=0,01$	NA
		M2_L1B_AUM	Modelo 2: dos entradas con aumento de datos y regularización $L1$	800	Si	$L1=0,01$	NA
		M2_L1C_AUM	Modelo 2: dos entradas con aumento de datos y regularización $L1$	1.000	Si	$L1=0,01$	NA
f.	Modelo 2	M2_L2A_AUM	Modelo 2: dos entradas con aumento de datos y regularización $L2$	600	Si	NA	$L2=0,01$
		M2_L2B_AUM	Modelo 2: dos entradas con aumento de datos y regularización $L2$	800	Si	NA	$L2=0,01$
		M2_L2C_AUM	Modelo 2: dos entradas con aumento de datos y regularización $L2$	1.000	Si	NA	$L2=0,01$
g.	Modelo 2	M2_L1_L2A_AUM	Modelo 1: dos entradas con aumento de datos y regularización $L1, L2$	600	Si	$L1=0,01$	$L2=0,01$
		M2_L1_L2B_AUM	Modelo 1: dos entradas con aumento de datos y regularización $L1, L2$	800	Si	$L1=0,01$	$L2=0,01$
		M2_L1_L2C_AUM	Modelo 1: dos entradas con aumento de datos y regularización $L1, L2$	1.000	Si	$L1=0,01$	$L2=0,01$

Tabla No.4.22 Proceso de experimentación del modelo 1 y del modelo 2 con ajustes: aumento de datos y regularización $L1$ y $L2$

Para resumir lo anterior en todo el proceso se utilizó el aumento de datos, desde (b) hasta (g), teniendo en cuenta que lo que cambió fueron los tipos de regularización, los cuales se aplicaron por separado en los modelos, a excepción de los últimos con (regularización elástica) que son la combinación de $L1$ y $L2$ (d) y (g). Solamente (a), (b) se los trabajó sin aumento de datos y regularización.

Los resultados iniciales que se muestran en la tabla 4.22, y la tabla 4.23, que corresponden a los modelos 1 y modelo 2 respectivamente, los cuales no tienen implementado el aumento de datos, ningún tipo de regularización y los datos están desbalanceados. Por tanto, esta es una información que permitió determinar la arquitectura y el diseño del modelo de red neuronal que se utilizó, en este caso una red neuronal convolucional con regresión (CNN). Es así como, estos resultados se muestran porque son un punto de referencia para los procesos de experimentación posteriores,

que son modelos a los cuales se les realizaron diferentes ajustes. Es importante indicar que el desbalanceo inicial de las tres figuras (esfera, cilindro y cubo) seleccionadas para esta tarea, se manejó de forma prudente teniendo en cuenta que en su momento se requería construir un modelo base para la predicción de la altura de dichas formas y que este fuera funcional, las que posteriormente se balancearon con el aumento de datos (*ImageDataGenerator*) y la regularización *L1* (*Lasso*) y *L2* (*Ridge*). Por tanto, solo se permitió trabajar así de forma transitoria con las cantidades desbalanceadas de las imágenes de las figuras, que ya más adelante cuando el modelo base estuvo desarrollado por completo se balancearon para evitar el sobreajuste y controlar la complejidad del modelo.

A continuación, se relacionan los aspectos y características las más relevantes del modelo 1 y del modelo 2:

- **Hiperparámetros:**
 - Tasa de aprendizaje (*Learning rate*): 0,001
 - Número de épocas (*Epochs*): 600, 800, 1.000
 - Tamaño de lote (*Batch size*): 32
 - Número de capas: para el modelo 2 son 14 capas y el modelo 1 tiene 7 capas
- **Arquitectura de red:** es una red neuronal convolucional (*CNN*) para regresión
- **Función de pérdida:** error cuadrático medio (*MSE, Mean Squared Error*)
- **Función de activación:** *ReLU, Adam*
- **Aumento de datos** (*ImageDataGenerator*), para balanceo de datos. Los parámetros utilizados fueron:
 - Rotación aleatoria (*rotation_range*) = 20, rango de rotación aleatorio de imágenes entre 0° y 180°
 - Traslaciones horizontales (*width_shift_range* = 0.2), trasladar aleatoriamente las imágenes horizontalmente.
 - Traslaciones verticales (*height_shift_range*= 0.2), trasladar aleatoriamente las imágenes verticalmente.
 - Distorsión (*shear_range*= 0.2), cizallamiento aleatorio a las imágenes: distorsión de una imagen a lo largo de un eje)
 - Aumento (*zoom_range*= 0.2), aumento o reducción (ancho x alto) aleatorio de imágenes
 - Rotación (*horizontal_flip*=*True*), imágenes volteadas horizontalmente de forma aleatoria
 - Rellenar (*fill_mode*='nearest'), rellenar del pixel vacío o que quedaron expuestos con un valor del pixel más cercano
- **Regularización:**
 - Regularización *L1*: con valor de 0,01
 - Regularización *L2*: con valor de 0,01
 - Combinación de la regularización *L1* y *L2* (*ElasticNet*) que es la combinación de *L1* y *L2* con el valor de 0,01.

a. Análisis del modelo 1 sin regularización y sin aumento de datos

Los resultados en la tabla 4.23 muestran para el *MAE* como para el *MSE* en el entrenamiento como para la validación una disminución a medida que aumentan las épocas, lo que indica que el modelo está comenzando a aprender, sin embargo, puede ser propenso a sobreajustarse. Adicionalmente se observa que el coeficiente de determinación (R^2) su valor es alto lo que indica que el modelo puede explicar gran parte de la variabilidad de los datos, esto se refuerza con el resultado del coeficiente de correlación de *Pearson* (r) es alto tanto lo que indica una fuerte relación lineal entre las variables predichas y las variables reales en general. El *P-value* es bajo tanto en entrenamiento como en la validación, lo cual indica que la relación entre las variables predichas y las variables reales es estadísticamente significativa. De igual forma los valores muestran una disminución en la pérdida a medida que aumentan el número de épocas que es un indicio de que el modelo está convergiendo.

Épocas	Error absoluto medio MAE		Error cuadrático medio MSE		Coeficiente de determinación (R^2)	Coeficiente de correlación de Pearson (r)			Pérdida (Loss)		Promedio historial de entrenamiento	
	Entre	Valid	Entre	Valid		r	P -Value	Entre	Valid	MAE	MSE	
600	0,13	1,01	0,08	2,05	0,94	0,97	0,000	0,08	2,05	0,13	0,08	
800	0,11	0,95	0,06	1,81	0,94	0,97	0,000	0,06	1,81	0,11	0,06	
1000	0,09	0,88	0,04	1,80	0,95	0,98	0,000	0,04	1,60	0,09	0,04	
1250	0,08	0,96	0,04	1,88	0,94	0,97	0,000	0,04	1,88	0,08	0,04	

Épocas	Error absoluto medio MAE		Error cuadrático medio MSE		Coeficiente de determinación (R^2)	Coeficiente de correlación de Pearson (r)			Pérdida (Loss)		Promedio historial de entrenamiento	
	Entre	Valid	Entre	Valid		r	P -Value	Entre	Valid	MAE	MSE	
600	0,11	1,09	0,07	2,16	0,93	0,97	0,000	0,07	2,16	0,11	0,07	
800	0,09	1,05	0,04	2,07	0,92	0,96	0,000	0,09	1,05	0,09	0,04	
1000	0,08	1,08	0,04	1,94	0,93	0,97	0,000	0,04	1,94	0,11	0,06	
1250	0,07	0,98	0,03	1,81	0,94	0,97	0,001	0,03	1,81	0,07	0,03	

Tabla No.4.23 Tabla comparativa modelo 1 una entrada (arriba) y modelo 2 de dos entradas (abajo)

De otra parte, ya con la validación cruzada se encontró que, todos los modelos 1A, 1B, 1C y 1D (validación cruzada para el modelo 1) tienen un buen rendimiento en la validación cruzada, además de los valores del *MAE*, *MSE* y de la pérdida, e igualmente valores altos de (R^2) y del coeficiente de correlación que son un análisis de los modelos de la validación cruzada del modelo 1, y la tabla 4.24 (izquierda). En este caso se observa que los modelos tienen un rendimiento similar, analizando que se pueden tener varias opciones, tales como, si en este caso la precisión es la prioridad se puede seleccionar el modelo 1A por el *MAE* más bajo, la otra es que, si se refiere a eficiencia computacional es el modelo 1D que tiene menos épocas, y si lo que se busca es un buen balance entre precisión y eficiencia sería el modelo 1B o el modelo 1C.

De tal forma que, estos resultados son iniciales, ya que más adelante se muestran otros resultados, pero con la implementación del aumento de datos y de la regularización tanto *L1* como *L2*. Por tanto, estos resultados son un punto de referencia para la evaluación en este caso del modelo 1 que es de una entrada respecto a otros modelos posteriores que tienen más ajustes y actualizaciones.

En resumen, el modelo 1 (una entrada), que es de una sola entrada tiene un bajo error y una alta precisión, y aprende a predecir con mayor precisión a medida que aumentan las épocas, explicando gran parte de la variabilidad de los datos, tiene fuerte relación lineal entre variables predichas y reales (en este caso la altura) la cual les estadísticamente significativa.

Modelos	Épocas	Folds = 1,2,3,4,5	Pérdida del Modelo	Error Absoluto Medio (MAE)	Error Cuadrático Medio (MSE)
Modelo 1A	600	1,2,3,4,5	0,73	0,55	0,73
Modelo 1B	800	1,2,3,4,5	0,8	0,57	0,8
Modelo 1C	1.250	1,2,3,4,5	0,75	0,54	0,75
Modelo 1D	1.600	1,2,3,4,5	0,75	0,55	0,76

Validación cruzada modelo No.2. k=5

Modelos	Épocas	Folds=1,2,3,4,5	Pérdida del Modelo (Loss)	Error Absoluto Medio (MAE)	Error Cuadrático Medio (MSE)
Modelo 2A	600	1,2,3,4,5	0,64	0,51	0,64
Modelo 2B	800	1,2,3,4,5	0,64	0,5	0,64
Modelo 2C	1.250	1,2,3,4,5	0,54	0,48	0,54
Modelo 2D	1.600	1,2,3,4,5	0,57	0,49	0,57

Validación cruzada modelo No.2. k=5

Modelos	Épocas	Folds = 1,2,...,10	Pérdida del Modelo (Loss)	Error Absoluto Medio (MAE)	Error Cuadrático Medio (MSE)
Modelo 1A	600	1,...,10	0,58	0,52	0,58
Modelo 1B	800	1,...,10	0,57	0,52	0,57
Modelo 1C	1.250	1,...,10	0,61	0,53	0,61
Modelo 1D	1.600	1,...,10	0,58	0,52	0,58

Validación cruzada modelo No.2. k=10

Modelos	Épocas	Fold=1,2,...,10	Pérdida del Modelo (Loss)	Error Absoluto Medio (MAE)	Error Cuadrático Medio (MSE)
Modelo 2A	600	1... 10	0,52	0,48	0,52
Modelo 2B	800	1... 10	0,51	0,48	0,51
Modelo 2C	1.250	1... 10	0,49	0,46	0,49
Modelo 2D	1.600	1... 10	0,47	0,44	0,47

Validación cruzada modelo No.2. k=10

Modelos	Épocas	Folds = 1,2,...,20	Pérdida del Modelo (Loss)	Error Absoluto Medio (MAE)	Error Cuadrático Medio (MSE)
Modelo 1A	600	1,...,20	0,58	0,54	0,58
Modelo 1B	800	1,...,20	0,57	0,52	0,57
Modelo 1C	1.250	1,...,20	0,57	0,52	0,57
Modelo 1D	1.600	1,...,20	0,47	0,55	0,47

Validación cruzada modelo No.2. k=20

Modelos	Épocas	Fold=1,2,...,20	Pérdida del Modelo (Loss)	Error Absoluto Medio (MAE)	Error Cuadrático Medio (MSE)
Modelo 2A	600	1... 20	0,58	0,54	0,58
Modelo 2B	800	1... 20	0,57	0,52	0,57
Modelo 2C	1.250	1... 20	0,57	0,52	0,57
Modelo 2D	1.600	1... 20	0,47	0,55	0,47

Validación cruzada modelo No.2. k=20

Tabla No. 4.24. Validación cruzada modelo 1 (izquierda) y modelo 2 (derecha), sin aumento de datos y sin regularización

Análisis del modelo 2 sin regularización y sin aumento de datos

Con los resultados se observa que el general el modelo 2, muestra un rendimiento que puede mejorar a lo largo de las épocas, lo que sugiere que está aprendiendo de forma efectiva y generalizando bien datos no vistos, sin embargo, se somete más adelante a otros ajustes y actualizaciones.

Los resultados que a continuación se detallan fueron obtenidos a partir de los modelos 1 y modelo 2, los cuales se presentan en varias tablas comparativas y se da una descripción a más detalle de los ajustes realizados a los modelos. Es importante indicar que, el conjunto de datos a partir de aquí estuvo siempre balanceado por medio del aumento de datos (*ImageDataGenerator*) e igualmente se aplicó la regularización *L1* y *L2* con cada uno de los modelos en diferentes épocas. Posteriormente con el balanceo del conjunto de datos para cada una de las figuras, se implementaron la regularización *L1, L2* en los modelos 1 y 2 de forma independiente en diferentes épocas, dando como

resultado una mejora considerable en el rendimiento de estos tanto en la disminución del error como en las predicciones.

Posteriormente se aplicó la regularización a los modelos 1 y 2, independientemente teniendo en cuenta que el conjunto de datos esta balanceado. En las tablas 4.24, 4.25 y 4.26 se observa la tendencia del modelo a través de las épocas a mejorar en rendimiento, lo que indica que generaliza mejor los datos nuevos. Para contextualizar mejor los resultados es importante tener en cuenta que al aplicar la regularización, es bueno saber que esta tiene un coeficiente de regularización conocido como el parámetro *lambda* (λ), el cual es un parámetro que controlan la magnitud de la penalización, y que se aplica a los pesos del modelo durante el entrenamiento para evitar el sobreajuste. Este coeficiente es utilizado en técnicas de regularización *L1* y *L2* como en este caso que serían (λ_1, λ_2) respectivamente. Y es así como, cuanto mayor sea el valor de *lambda* (λ), mayor será la penalización, forzando a los pesos a ser más pequeños. En general el modelo mostró un rendimiento que mejoró a lo largo de las épocas. Las métricas de rendimiento y la pérdida indicaron que el modelo está aprendiendo de manera efectiva y realizó predicciones precisas. La validación cruzada aplicada al modelo 2 permitió observar una mejoría en las métricas en general a medida que aumentaron.

b. Análisis del modelo 1 con regularización *L1* y aumento de datos (*ImageDataGenerator*)

Este modelo tiende a mejorar ligeramente la precisión y el coeficiente de determinación y correlación de Pearson se mantienen relativamente constantes y no hay sobreajuste significativo. El modelo en general tiene un buen desempeño al estimar la atura, buenas señales de aprendizaje y generalización. De igual forma para el modelo podría ser una buena opción ampliar el *dataset*. El modelo M1_L1C_AUM tiene el mejor rendimiento general en términos del *MAE* y de la pérdida en la validación.

De los modelos en los cuales se utilizó la regularización *L1* con aumento de datos, estos tienden a mejorar el *MAE* y el *MSE* a medida que aumentan las épocas y se mantienen bastante cercanos lo que es una buena señal de generalización. El modelo M1_L1C_AUM con 1.000 épocas tiene un *MAE* y un *MSE* más bajos en el proceso de validación y una mejor generalización, en comparación al modelo de 600 y 800 épocas. En la figura 4.23 se observa que el *MAE* disminuye rápidamente en el entrenamiento y la validación llegando a estabilizarse y mantenerse consistentes lo cual es una señal de convergencia del modelo y generalización.

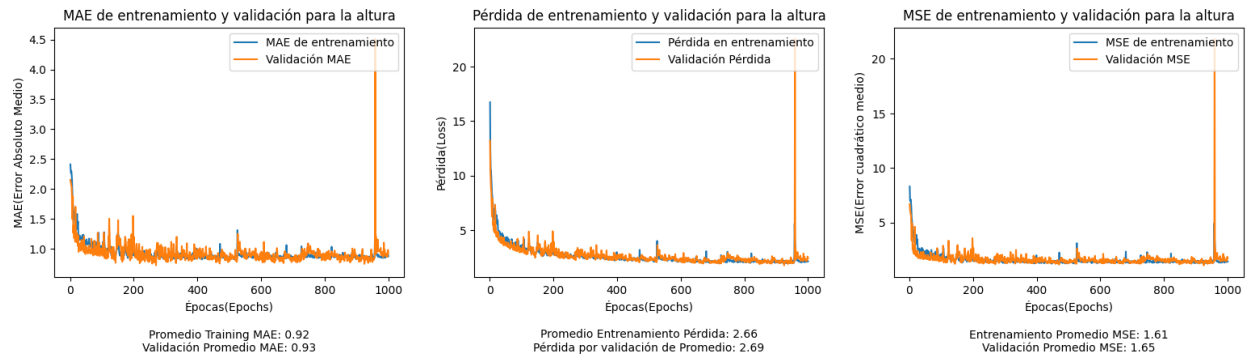


Figura No.4.23. MAE, MSE y Loss del modelo 1 con regularización $L1$ y aumento de datos con 1.000 épocas

Ambas gráficas en la figura 4.24 demuestran que el modelo tiene un desempeño consistente con una tendencia a predecir con más consistencia en el conjunto de entrenamiento que en el de validación.

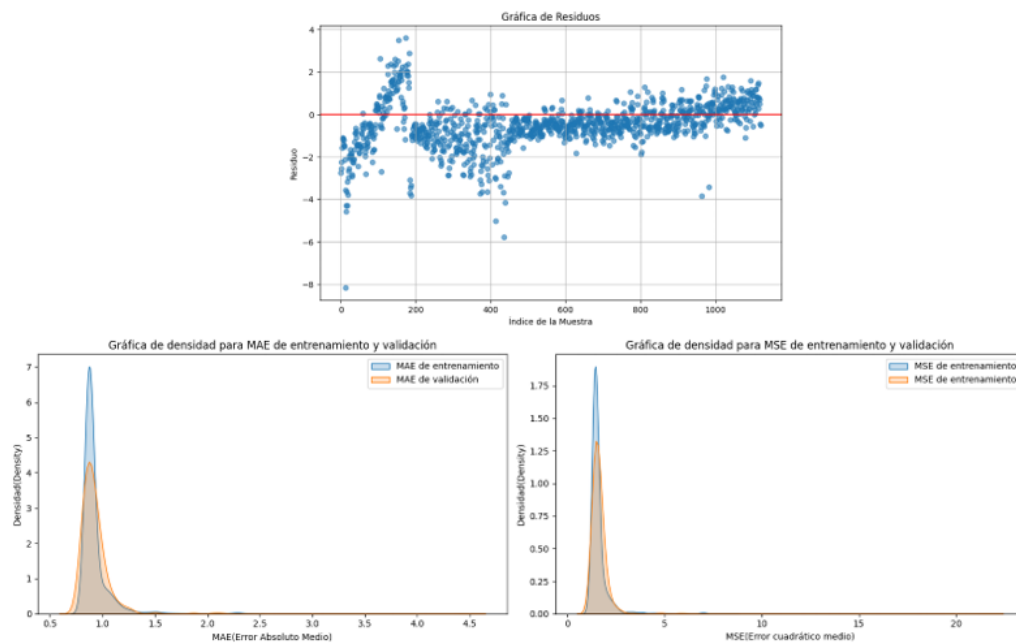


Figura No. 4.24. Gráfica de densidad del MAE y del MSE y de residuos modelo 1 con regularización $L1$ y aumento de datos con 1.000 épocas

La figura 4.25 muestra una correlación positiva ya que a medida que la altura real aumenta también lo hace la altura predicha. La mediana de errores está cerca a cero, lo cual es bueno, ya que indica que no hay un sesgo sistemático en la predicción de la altura. Se presentó una variabilidad en las predicciones lo cual es una señal de que hay casos en los cuales la predicción en la altura es relativamente diferente a la esperada. De tal manera que, para el cilindro (caja azul) existe una cantidad notable de predicciones que se desvían de la línea de ajuste. De igual el rango intercuartílico (IQR) muestra que

la mayoría de los errores se concentran aproximadamente entre los valores 2 cm y -2 cm, sin embargo, los bigotes de la caja indican que se extiende a valores más extremos de errores, y que los valores atípicos (*outliers*) muestran que hay algunas predicciones con errores significativamente mayores. Para el cubo las predicciones fueron en su mayoría (caja naranja) aunque hay valores atípicos estos no son tan extremos como los

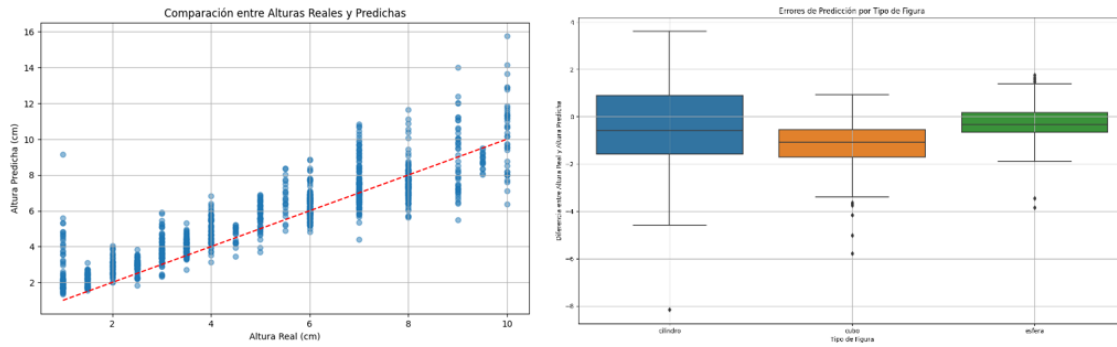


Figura No. 4.25. Gráfica de dispersión y gráfico de caja del modelo 1 con regularización $L1$ y aumento de datos con 1.000 épocas

del cilindro. En este caso el cubo este tiene la mediana cerca a cero, pero su rango intercuartílico (*IRQ*) es más estrecho que el del cilindro. De otro lado la esfera (caja verde) la mediana este cerca de cero y el rango intercuartílico (*IRQ*) es bastante estrecho, en este caso indica que las predicciones son más precisas y consistentes, teniendo errores menos significativos lo que indica que los valores atípicos son menos extremos.

En general su precisión fue razonablemente aceptable para las tres figuras con un pequeño número de predicciones atípicas, debido a características que no son bien capturadas por el modelo. Es así como en este caso esta combinación de ajustes es más consistente y precisa para predecir las alturas de los cubos y de las esferas, y es menos consistente para los cilindros. Por tanto, el modelo puede requiere aumentar dentro del conjunto de datos imágenes con características discriminativas para los cilindros.

c. Análisis del modelo 1 con regularización $L2$ y aumento de datos (*ImageDataGenerator*)

De los modelos con regularización $L2$ muestran una tendencia a mejorar en el proceso de validación cuando aumentan el número de épocas. El modelo M1_L2C_AUM con 1.000 épocas tiene el *MAE* y el *MSE* más bajo en la validación y un R^2 más alto, lo que es una señal de este modelo es más preciso y confiable. En la figura 4.26 se muestra un buen rendimiento con el aumento de las épocas, donde todas las métricas de validación mejoran, no hay señales de sobreajuste, dado que el *MAE* y el *MSE* de validación son significativamente mayores que los de entrenamiento.

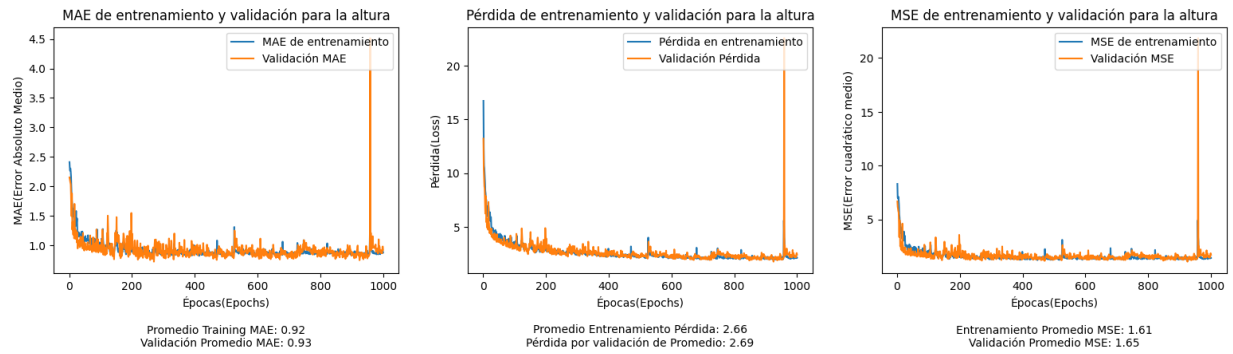


Figura No.4.26. Modelo 1 con regularización L_2 y aumento de datos con 1.000 épocas

La gráfica de dispersión permitió observar que en el modelo hay una concentración más alta de puntos azules en las alturas más bajas, mientras que en alturas más altas estado puntos se dispersan más, que en este caso indica que el modelo tiende a ser más preciso en medidas de alturas más bajas ver figura 4.27. El modelo en sí tiene un buen ajuste general sin sesgos claros

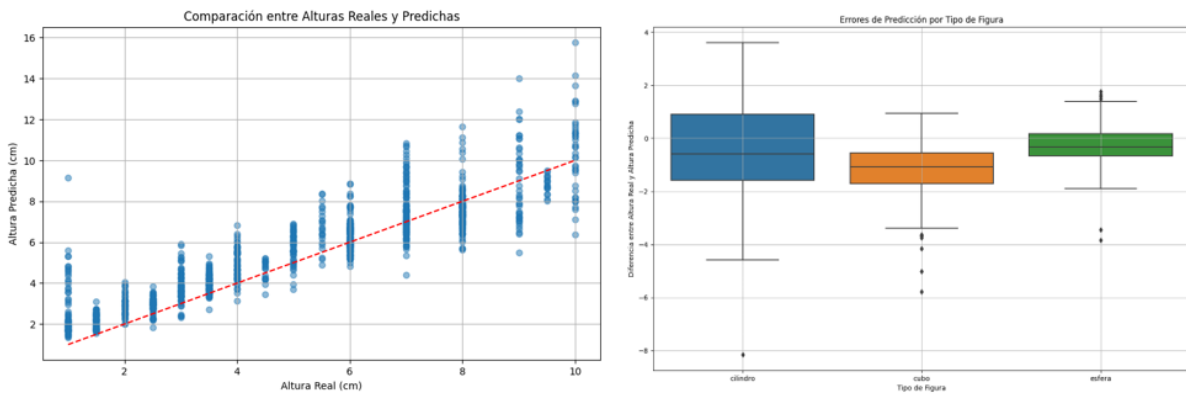


Figura No.4.27 Gráfica de dispersión, gráfico de caja del modelo 1 con regularización L_2 y aumento de datos con 1.000 épocas

El modelo tiene una tendencia positiva y predice mejor un rango de inferior de alturas, pero presentó una variabilidad en las predicciones a medida que el valor de las alturas reales aumentó, lo que podría indicar la necesidad de aumentar el conjunto de datos, pero analizando el comportamiento de los puntos más extremos en las predicciones. En la figura 4.28 la similitud y rango de las distribuciones indicaron que no hay sobreajuste. Donde hay que revisar es las colas en las gráficas es una señal de mejoras por realizar al modelo que le permitan manejar errores más grandes.

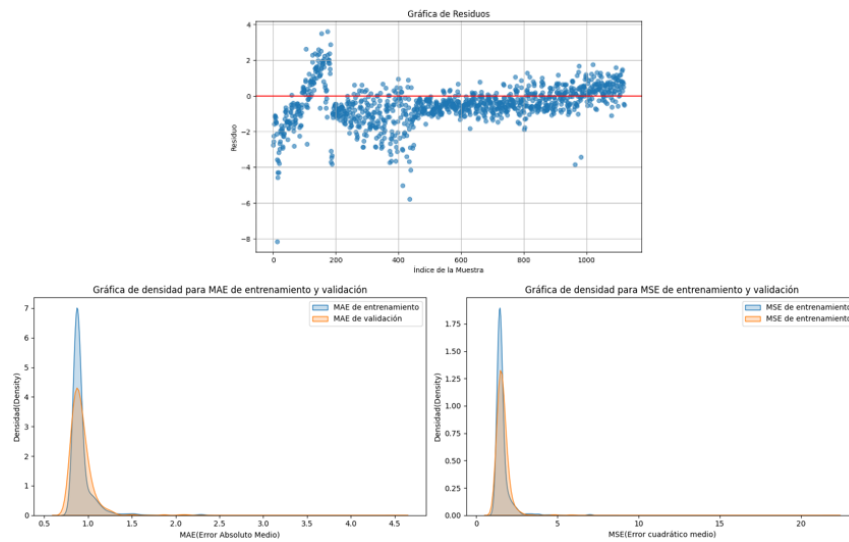


Figura No. 4.28 Gráfica de densidad del MAE, MSE y de residuos del modelo 1 con regularización L_2 y aumento de datos con 1.000 épocas

d. Análisis del modelo 1 con regularización L_1 , L_2 (regularización elástica) y aumento de datos (*ImageDataGenerator*).

En este caso los modelos no mejoraron con el aumento de las épocas a pesar de que el MAE y el MSE son relativamente bajos, no son mejores que los modelos de los dos grupos de tablas anteriores ver figura 4.29. Sin embargo, en este caso el modelo M1_L2C_AUM es el que tienen la mejor predicción y capacidad de generalización, así como el valor más alto del coeficiente de correlación de *Pearson* (r) y de R^2 .

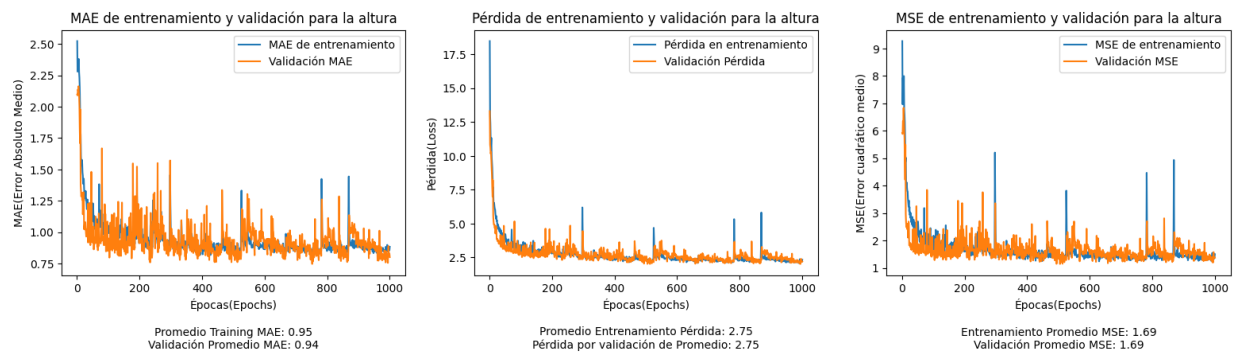


Figura No. 4.29. Modelo 1 con regularización L_1 y L_2 aumento de datos con 1.000 épocas

El sistema presenta más variabilidad para los valores más grandes de altura. Hay una tendencia a positiva, sin embargo, el modelo tiende a subestimar las alturas más grandes ya que muchos puntos se encuentran por debajo de la línea roja. El rango intercuartílico (IQR) es el más estrecho y la mediana está cerca de cero, por tanto, el modelo es más preciso para predecir alturas de cilindros. El cubo tiene un IQR más amplio y la mediana

está ligeramente hacia arriba lo que sugiere un error sistemático en las predicciones. De otra parte, la esfera tiene un IQR más amplio lo que sugiere que es la figura menos precisa y la mediana más cercana a cero ver figura 4.30.

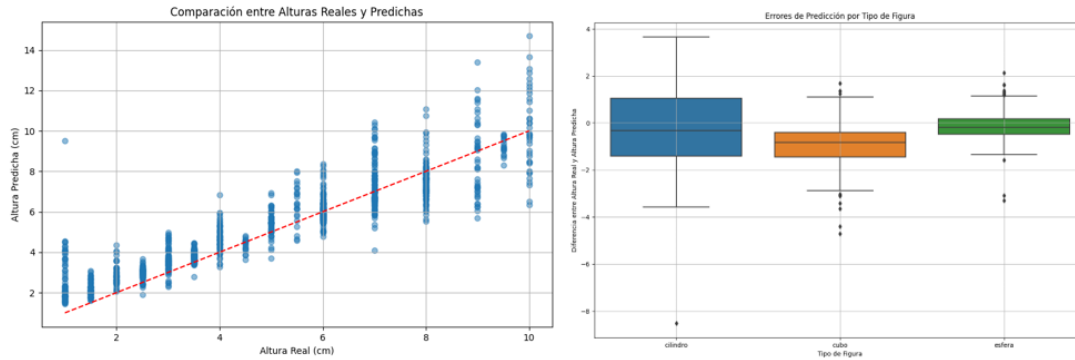


Figura No.4.30 Gráfica de dispersión, gráfico de caja del modelo 1 con regularización $L1$ y $L2$ y aumento de datos con 1.000 épocas

No se observa un sesgo sistemático o algún patrón o tendencia en la distribución los residuos. Las curvas de densidad del MAE como del MSE tanto de entrenamiento como de validación con similares y cercanas entre sí lo cual indica que el modelo generaliza bien ver figura 4.31.

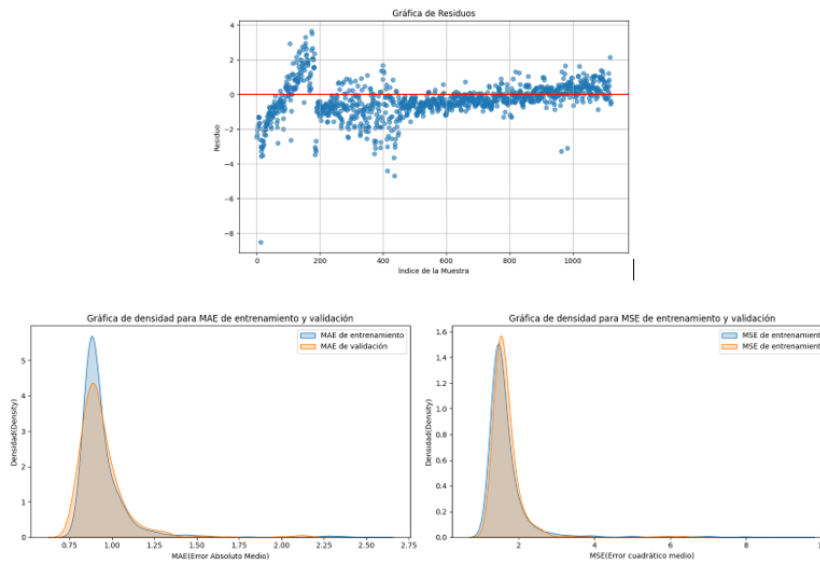


Figura No. 4.31 Gráfica de densidad del MAE , MSE y de residuos del modelo 1 con regularización $L1$ y $L2$ y aumento de datos con 1.000 épocas

Basado en los resultados el modelo está bien regularizado y generaliza adecuadamente. Sin embargo, hay que revisar varios valores atípicos que presenta.

e. Análisis modelo 2 con aumento de datos y regularización $L1=0,01$ con 1.000 épocas con 1.000 épocas

En estos los valores de los resultados del *MAE* y el *MSE* son consistentes (ver figura 32), lo que indica que el modelo tiene un rendimiento similar en ambos conjuntos de datos. En resumen, la consistencia en los datos de entrenamiento y validación en todas las métricas sugiere que el modelo es generalizable y no está sobreajustado. La cercanía del *MAE* de entrenamiento y de validación indica que el modelo no está sobreajustado, sin embargo, se presentaron algunos picos alrededor de las épocas 200 y 800 lo que podría indicar problemas temporales de aprendizaje. Igualmente, el *MSE* se estabiliza después de las primeras épocas de aprendizaje ver tabla 4.25.

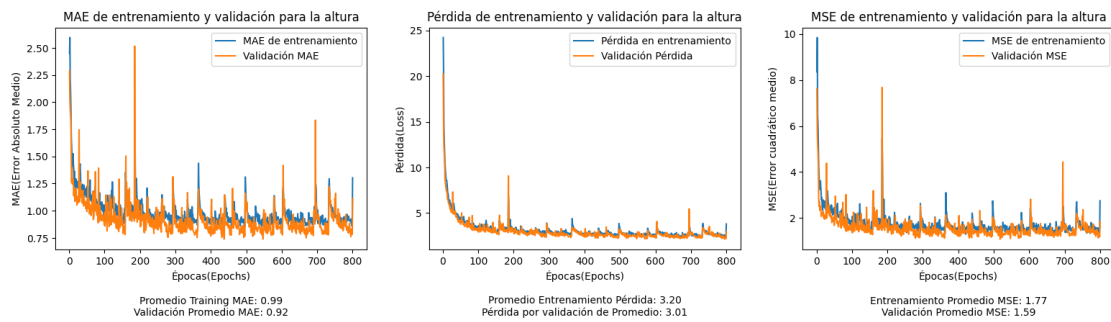


Figura No. 4.32. Modelo 2 con regularización $L1$ y aumento de datos con 1.000 épocas

Igualmente, la gráfica de dispersión indica una variabilidad considerable en las predicciones, especialmente con los valores de la altura real que son más altos. Lo anterior indica que el modelo es menos preciso para valores más altos de alturas reales ver figura 4.33. También predominaron algunos valores atípicos que están lejos de la línea de ajuste. Existe la posibilidad de que el modelo tenga heteroscedasticidad, para lo cual se realizó una transformación de los datos para estabilizar la varianza de errores. El modelo en la gráfica de caja muestra que para los cilindros es más variable en las predicciones, mientras que para los cubos y las esferas tiende a ser más consistente. Este modelo se le realizaron ajustes posteriores como la utilización de la regularización $L2$ que es la regularización elástica, que podría ofrecer un equilibrio entre la selección de características y la penalización de los pesos ver figura 4.34.

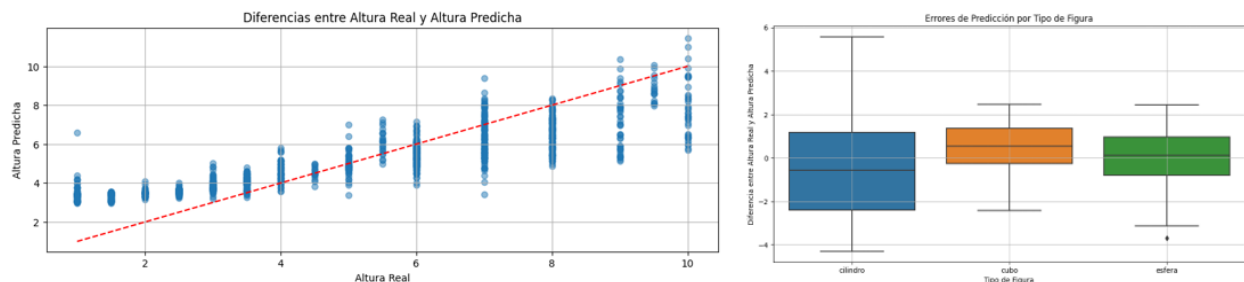


Figura No. 4.33 Gráfica de dispersión, gráfico de caja del modelo 2 con regularización $L1$ y aumento de datos con 1.000 épocas

En este caso lo esperado del modelo fue que la diferencia entre los valores reales y los valores predichos por el modelo (residuos) presentaran una dispersión o varianza constante a lo largo todas las predicciones, pero sucede lo contrario no es constante y esto tiende a aumentar o disminuir con las variables independientes.

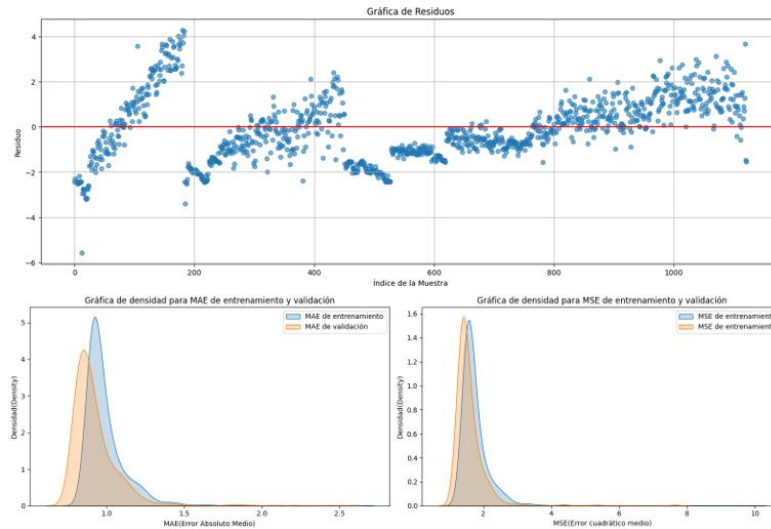


Figura No. 4.34 Gráfica de densidad del MAE, MSE y de residuos del modelo 2 con regularización $L1$ y aumento de datos con 1.000 épocas

f. Análisis modelo 2 con aumento de datos y regularización $L2 = 0,01$ con 1.000 épocas

Los resultados del modelo No.2 indican que presentó un buen rendimiento, tanto en las métricas de entrenamiento como de validación, con una pequeña brecha entre ellas, lo que indica una buena generalización. De igual forma la regularización $L2$ funcionó adecuadamente para prevenir el sobreajuste, lo cual fue soportado por la pérdida relativamente baja. De igual forma el aumento de datos con *ImageDataGenerator* ayudó a mejorar la generalización, lo que se observó reflejado en el buen *performance* del modelo en el conjunto de validación ver figura 4.35.

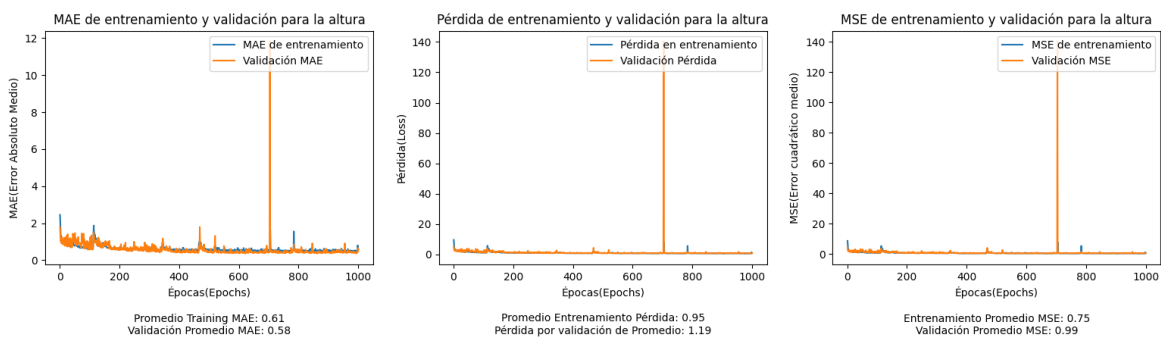


Figura No. 4.35. Modelo 2 con regularización $L2$ y aumento de datos con 1.000 épocas

El pico en la época 200 que es menos pronunciado y 800 aproximadamente puede estar indicando un dato anómalo o atípico o mal etiquetado en esos puntos específicos del entrenamiento, o posiblemente un problema de convergencia. Después de los picos el modelo mostró un comportamiento de aprendizaje muy estable con una convergencia de pérdida cercana a cero, lo que indica que el modelo ha aprendido bien las características de los datos y generalizó de manera bien efectiva. Las gráficas de densidad permitieron entender cómo se distribuyeron los errores en el modelo de tal forma que, en este caso, ambos tipos de errores están cercanos a cero, lo que indica el buen rendimiento tanto en los datos de entrenamiento como en los de validación ver figura 4.36, 4.37.

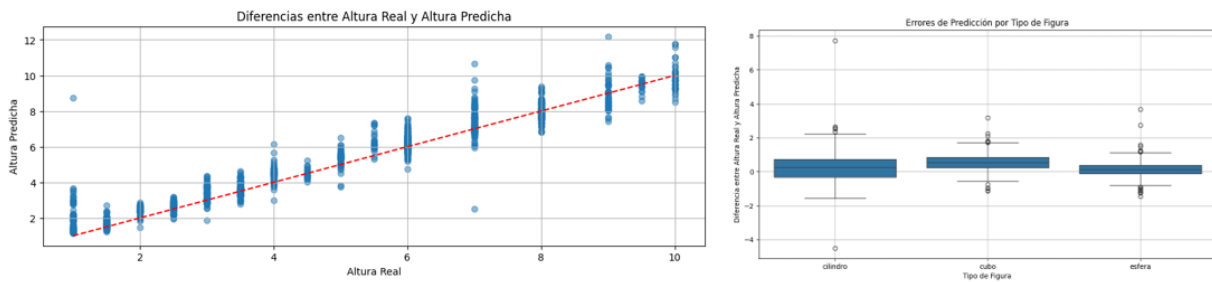


Figura No.4.36 Gráfica de dispersión, gráfico de caja del modelo 2 con regularización L_2 y aumento de datos con 1.000 épocas

En resumen, el modelo presentó un aprendizaje robusto a lo largo del tiempo, lo que mostró un modelo bien regularizado y que puede generalizar los datos adecuadamente nuevos datos. En la figura 4.37 se muestra la regularización L_2 y el aumento de datos contribuyeron positivamente en la estabilidad y desempeño del modelo. Estaría pendiente por profundizar en la razón de los picos en las gráficas para determinar su causa y mejorar la confiabilidad del modelo. En la gráfica de densidad también se puede observar que el modelo es bastante consistente con una variabilidad de errores relativamente baja, lo cual puede deberse a la regularización L_2 , que ayuda a prevenir el sobreajuste (pesos pequeños del modelo), y del aumento de datos que mejora la generalización al exponerlo a una mayor variedad de situaciones durante el entrenamiento. La gráfica de residuos mostró que no existe un patrón o tendencia de dispersión a medida que aumentan los datos, lo cual muestra un buen ajuste a lo largo de las épocas.

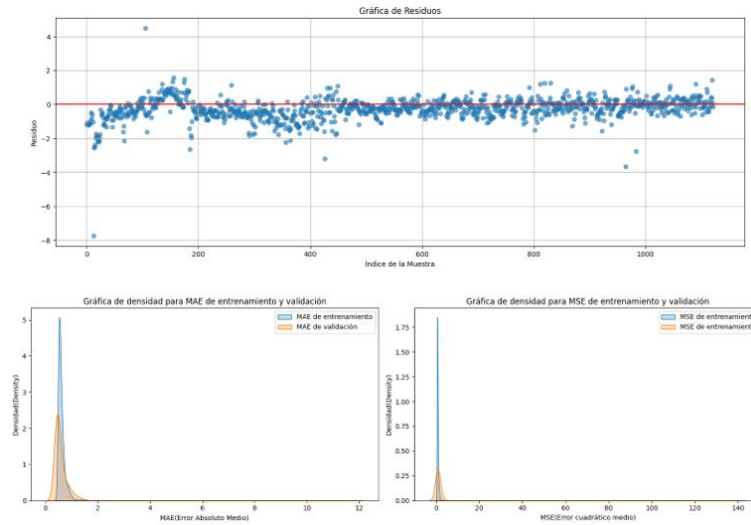


Figura No. 4.37 Gráfica de densidad del MAE, MSE y de residuos del modelo 2 con regularización L_2 y aumento de datos con 1.000 épocas

g. Análisis del modelo 2 con aumento de datos y regularización $L_1 = 0,01$ y $L_2 = 0,01$ con 1.000 épocas

En el modelo dos se realizó la combinación de la regularización L_1 y L_2 con el aumento de datos. Esta combinación es llamada regularización elástica (*ElasticNet*) se utilizó para evitar el sobreajuste, el cual se presenta en este caso cuando el modelo es demasiado complejo y aprende ruido de los datos de entrenamiento en lugar de las relaciones reales.

En el modelo No.2 con la utilización de la regularización L_1 y L_2 este muestra una mejoría en las predicciones a medida que aumentan las épocas, lo que indica de que el modelo está aprendiendo y contribuye al buen rendimiento y generalización del mismo ver figura 4.38.

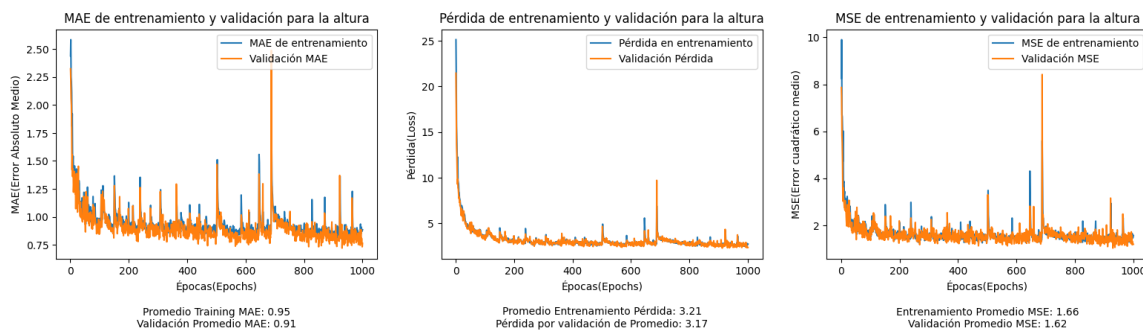


Figura No. 4.38. Modelo 2 con regularización L_2 y aumento de datos con 1.000 épocas

En general con los ajustes realizados el modelo este comienza a estabilizarse teniendo un buen rendimiento, con el MAE y el MSE que disminuyen rápidamente

manteniéndose estables en el tiempo ver figura 4.39.

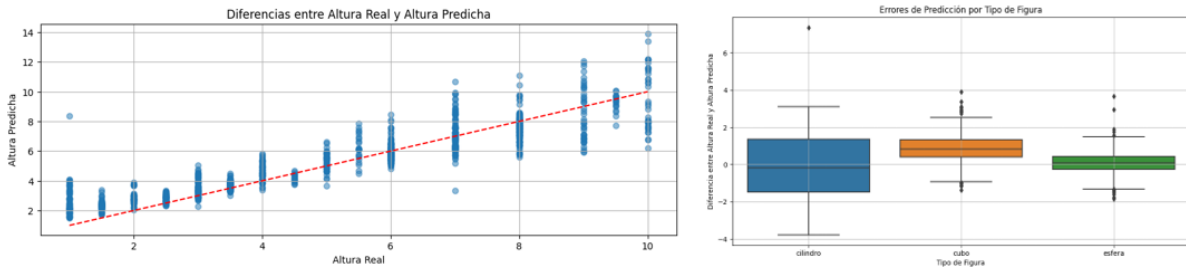


Figura No.4.39 Gráfica de dispersión, gráfico de caja del modelo 2 con regularización $L1$ y $L2$ y aumento de datos con 1.000 épocas

El modelo tiene un rendimiento variable, siendo más preciso en el rango inferior de alturas según la gráfica de dispersión. Si bien el modelo de 1.000 épocas parece ser el óptimo por tener unos valores más bajos en el MAE tiene un MSE más alto que significa errores grandes en algunas predicciones y teniendo en cuenta también que un MSE podría ser un indicio de sobreajuste. De otra parte, el modelo 800 épocas y teniendo en cuenta que lo que se busca es la estabilidad en la predicción, es la elección en este caso ver tabla 4.25. Es una muestra bastante uniforme lo que quiere decir que no tiene un sesgo sistemático y además de la dispersión uniforme alrededor de la línea cero es una señal de que el modelo está realizando predicciones razonables. Sin embargo, hay presencia de algunos valores atípicos lo cual es una parte por investigar, para determinar si es una anomalía o un patrón subyacente que el modelo no ha detectado ver figura 4.40.

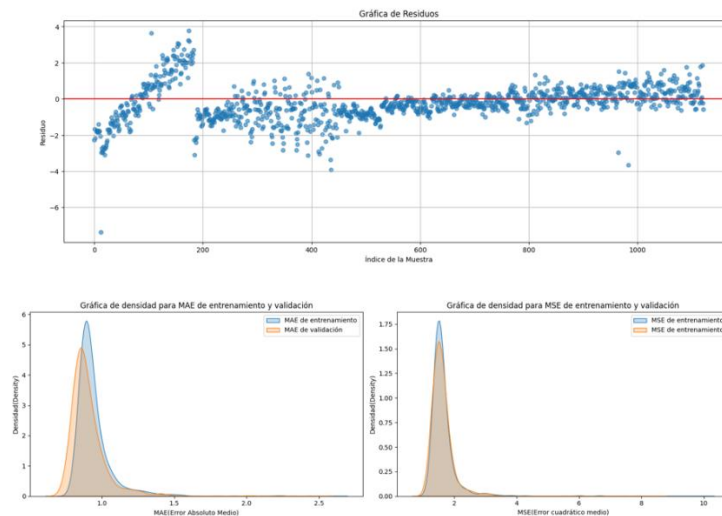


Figura No. 4.40 Gráfica de densidad del MAE , MSE y de residuos del modelo 2 con regularización $L1$ y $L2$ y aumento de datos con 1.000 épocas

Las siguientes tablas son el resumen de los resultados del modelo 2 con la implementación del aumento de datos y de la regularización $L1$ y $L2$.

Tablas resumen del modelo 2 (dos entradas) con aumento de datos y regularización L1 y L2 con 1.000 épocas

Modelos	Número de épocas	Error absoluto medio MAE		Error cuadrático medio MSE		Coeficiente de determinación R ²	Coeficiente de correlación de Pearson			Pérdida (Loss)		Promedio historial de Entrenamiento		Regularización		Hiperparámetros				Aumento de datos	Margen de error
		Entrenam	Validación	Entrenam	Validación		(r)	P-Value	Entrenam	Validación	MAE	MSE	L1=0.01	L2=0.01	Tasa de aprendizaje	Tamaño de lote	# Capas	Funcion de activación	ImageDataGenerator		
M2_L1A_AUM	600	0,94	0,93	1,68	1,68	0,829	0,917	0,000	3,19	3,18	0,94	1,68	SI	NO	0,001	32	14	ReLu	SI	X	
M2_L1B_AUM	800	0,99	0,92	1,77	1,59	0,696	0,878	0,000	3,20	3,01	0,99	1,77	SI	NO	0,001	32	14	ReLu	SI	X	
M2_L1C_AUM	1000	0,96	0,92	1,75	1,63	0,820	0,919	0,000	3,13	3,00	0,96	1,75	SI	NO	0,001	32	14	ReLu	SI	X	

Tabla No.4.25. Modelo 2. (dos entradas) con aumento de datos y regularización L1

Modelos	Número de épocas	Error absoluto medio MAE		Error cuadrático medio MSE		Coeficiente de determinación R ²	Coeficiente de correlación de Pearson			Pérdida (Loss)		Promedio historial de Entrenamiento		Regularización		Hiperparámetros				Aumento de datos	Margen de error
		Entrenam	Validación	Entrenam	Validación		(r)	P-Value	Entrenam	Validación	MAE	MSE	L1=0.01	L2=0.01	Tasa de aprendizaje	Tamaño de lote	# Capas	Funcion de activación	ImageDataGenerator		
M2_L2A_AUM	600	0,67	0,75	0,91	1,51	0,93	0,97	0,000	1,12	1,72	0,67	0,91	NO	SI	0,001	32	14	ReLu	SI	X	
M2_L2B_AUM	800	0,62	0,68	0,76	1,06	0,92	0,97	0,000	0,95	1,25	0,62	0,76	NO	SI	0,001	32	14	ReLu	SI	X	
M2_L2C_AUM	1000	0,61	0,58	0,75	0,99	0,93	0,97	0,000	0,95	1,19	0,61	0,75	NO	SI	0,001	32	14	ReLu	SI	X	

Tabla No.4.26. Modelo 2. (dos entradas) con aumento de datos y regularización L2

Modelos	Número de épocas	Error absoluto medio MAE		Error cuadrático medio MSE		Coeficiente de determinación R ²	Coeficiente de correlación de Pearson			Pérdida (Loss)		Promedio historial de Entrenamiento		Regularización		Hiperparámetros				Aumento de datos	Margen de error
		Entrenam	Validación	Entrenam	Validación		(r)	P-Value	Entrenam	Validación	MAE	MSE	L1=0.01	L2=0.01	Tasa de aprendizaje	Tamaño de lote	# Capas	Funcion de activación	ImageDataGenerator		
M2_L1_L2A_AUM	600	1,22	1,19	2,51	2,55	0,69	0,83	0,00	3,52	3,54	1,22	2,51	SI	SI	0,001	32	14	ReLu	SI	X	
M2_L1_L2B_AUM	800	0,98	0,97	1,81	1,94	0,81	0,91	0,00	3,10	3,21	0,98	1,81	SI	SI	0,001	32	14	ReLu	SI	X	
M2_L1_L2C_AUM	1000	0,95	0,91	1,66	3,17	0,84	0,91	0,00	3,21	3,17	0,95	1,66	SI	SI	0,001	32	14	ReLu	SI	X	

Tabla No.4.27. Modelo 2. (dos entradas) con aumento de datos y regularización L1 y L2

Tablas resumen del modelo 1 (una entrada) con aumento de datos y regularización L1 y L2 con 1.000 épocas

Modelos	Número de épocas	Error absoluto medio MAE		Error cuadrático medio MSE		Coeficiente de determinación R ²	Coeficiente de correlación de Pearson			Pérdida (Loss)		Promedio historial de Entrenamiento		Regularización		Hiperparámetros				Aumento de datos ImageDataGenerator	Margen de error 0,02
		Entrenam	Validación	Entrenam	Validación		(r)	P-Value	Entrenam	Validación	MAE	MSE	L1=0.01	L2=0.01	Tasa de aprendizaje	Tamaño de lote	# Capas	Funcion de activación			
M1_L1A_AUM	600	0,98	0,93	1,83	1,69	0,80	0,90	0,000	2,95	2,8	1,04	1,99	SI	NO	0,001	32	7	ReLu	SI	X	
M1_L1B_AUM	800	1,00	0,91	1,86	1,61	0,78	0,91	0,000	3,08	2,82	1,00	1,86	SI	NO	0,001	32	7	ReLu	SI	X	
M1_L1C_AUM	1000	0,92	0,93	1,61	1,65	0,78	0,91	0,000	2,66	2,69	0,92	1,61	SI	NO	0,001	32	7	ReLu	SI	X	

Tabla No. 4.28. Gráficos del modelo 1 (una entrada) con 1.000 épocas, aumento de datos y regularización L1

Modelos	Número de épocas	Error absoluto medio MAE		Error cuadrático medio MSE		Coeficiente de determinación R ²	Coeficiente de correlación de Pearson			Pérdida (Loss)		Promedio historial de Entrenam		Regularización		Hiperparámetros				Aumento de datos ImageDataGenerator	Margen de error 0,02
		Entrenam	Validación	Entrenam	Validación		(r)	P-Value	Entrenam	Validación	MAE	MSE	L2=0.01	L1=0.01	Tasa de aprendizaje	Tamaño de lote	# Capas	Funcion de activación			
M1_L2A_AUM	600	0,83	0,92	1,31	1,64	0,84	0,90	0,00	1,70	2,03	0,83	1,31	NO	SI	0,001	32	7	ReLu	SI	X	
M1_L2B_AUM	800	0,83	0,93	1,25	1,71	0,87	0,93	0,00	1,60	2,06	0,80	1,25	NO	SI	0,001	32	7	ReLu	SI	X	
M1_L2C_AUM	1000	0,79	0,87	1,22	1,48	0,85	0,94	0,00	1,52	1,79	0,79	1,22	NO	SI	0,001	32	7	ReLu	SI	X	

Tabla No. 4.29. Gráficos del modelo 1 (una entrada) con 1.000 épocas, aumento de datos y regularización L2

Modelos	Número de épocas	Error absoluto medio MAE		Error cuadrático medio MSE		Coeficiente de determinación R ²	Coeficiente de correlación de Pearson			Pérdida (Loss)		Promedio historial de Entrenamiento		Regularización		Hiperparámetros				Aumento de datos ImageDataGenerator	Margen de error 0,02
		Entrenam	Validación	Entrenam	Validación		(r)	P-Value	Entrenam	Validación	MAE	MSE	L1=0.01	L2=0.01	Tasa de aprendizaje	Tamaño de lote	# Capas	Funcion de activación			
M1_L1_L2A_AUM	600	0,96	0,96	1,73	1,73	0,81	0,90	0,000	2,9	2,89	0,96	1,73	SI	SI	0,001	32	7	ReLu	SI	X	
M1_L1_L2B_AUM	800	0,96	0,96	1,73	1,73	0,81	0,90	0,000	2,9	2,89	0,96	1,73	SI	SI	0,001	32	7	ReLu	SI	X	
M1_L1_L2C_AUM	1000	0,95	0,94	1,69	1,69	0,82	0,91	0,000	2,75	2,75	0,95	1,69	SI	SI	0,001	32	7	ReLu	SI	X	

Tabla No. 4.30. Gráficos del modelo 1 (una entrada) con 1.000 épocas, aumento de datos y regularización L1 y L2

Del proceso anterior en el cual se obtuvieron diferentes modelos donde se les implementó la regularización tanto $L1= 0,01$ como $L2 = 0,01$ y se utilizó el aumento de datos, con el propósito de balancear el conjunto de datos, se puede afirmar que mejoraron de forma notoria los resultados con relación a la primera experimentación con modelos, sin aumento de datos y sin regularización, que igualmente sirvieron de referencia o de comparación y evolución para uno de los modelos.

Evaluación general de todos los modelos 1 y modelo 2 desarrollados

Todos los resultados de los modelos se detallaron en las tablas anteriores con el propósito de facilitar la comparación en rendimiento. Estos son un punto de discusión para trabajos e investigaciones posteriores. De los 18 modelos probados 9 del modelo 1 y 9 del modelo 2, se seleccionaron dos modelos, los cuales son el *M1_L2C_AUM* y el *M2_L2C_AUM* a los cuales se les evaluó utilizando las siguientes métricas, consideradas las más importantes para realizar esta tarea: están el error absoluto medio (*MAE*), el error cuadrático medio (*MSE*), el coeficiente de determinación (R^2), el coeficiente de correlación de *Pearson* (r), la pérdida de entrenamiento y validación. Además, es importante indicar que el conjunto de datos se balanceó precisamente para realizar este proceso de experimentación.

Los modelos seleccionados tienen las siguientes características:

- ***M1_L2C_AUM***: modelo 1 (una entrada), con aumento de datos y regularización $L2 = 0,01$ con 1.000 épocas
- ***M2_L2C_AUM***: modelo 2 (dos entradas), con aumento de datos y regularización $L2 = 0,01$ con 1.000 épocas

La precisión de los modelos evolucionó positivamente, sin embargo, se presentaron valores atípicos, los cuales requieren de un trabajo de experimentación más extenso y más profundo. Sobre la base de las métricas generales y teniendo en cuenta el equilibrio entre el rendimiento de los datos de entrenamiento y validación, ambos modelos seleccionados mostraron buen equilibrio entre las métricas de error y los coeficientes de determinación, lo cual indica que han aprendido el patrón subyacente sin sobrejustarse al ruido de los datos de entrenamiento.

Valor de tolerancia al error

En las pruebas realizadas a cada uno de los modelos, estos realizaron predicciones donde se estableció un margen de tolerancia de error de 0,02, en relación con las diferencias de las alturas reales y las alturas predichas de las figuras. De igual forma con el aumento en las épocas en el proceso de entrenamiento (1.400 épocas) en el modelo 2 (*M2_L2C_AUM*) en *MAE* continúa disminuyendo aún más la diferencia entre alturas reales y las alturas predichas de las figuras.

En este proceso se obtuvieron dos gráficas, la figura 4.41 y la figura 4.42 muestran los

resultados de las predicciones de los modelos 1 (*M1_L2C_AUM*) y modelo 2 (*M2_L2C_AUM*) con 1.000 épocas respectivamente, con el margen de tolerancia indicado anteriormente (0,02), lo que permite comparar la distribución de las predicciones y detectar en que rangos de medidas se están presentando los valores atípicos.

En el modelo 1 (*M1_L2C_AUM*) y en el modelo 2 (*M2_L2C_AUM*) se observa como los valores tienden acercarse hacia cero, lo cual es un indicador de que el modelo está realizando predicciones muy cerca de los valores reales y por tanto son bastante precisas. En la distribución de las barras en el histograma para el modelo 1 (*M1_L2C_AUM*) muestra que la mayoría de las diferencias absolutas se concentran cerca de cero (0), lo que sugeriría que este tiene una precisión relativamente buena, ya que la mayoría de las predicciones están cerca de los valores reales. Sin embargo, también presenta una cola larga hacia la derecha con un número significativo de predicciones, que se están alejando más de los valores reales, lo cual podría requerir más experimentación y ajustes al modelo ver figura 4.41.

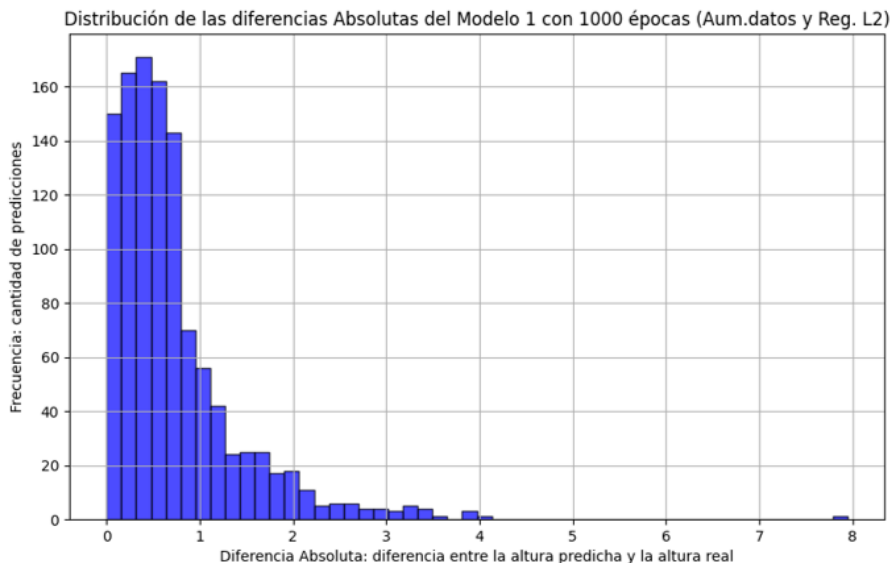


Figura No. 4.41 Distribución de las diferencias Absolutas del Modelo 1 (*M1_L2C_AUM*) con 1.000 épocas (Aum. datos y Reg. L2)

En la figura 4.42 y la 4.43 la figura la mayoría de las diferencias entre las predicciones y los valores reales se concentran cerca de cero, esto significa que muchas predicciones son bastante precisas o están muy cerca del valor real. También se observa una cola larga que se extiende hacia la derecha del gráfico, pero que tiene frecuencias significativamente menores en comparación con las barras cercanas a cero. Esta es una señal de que hay predicciones que se alejan bastante de los valores reales, pero que son menos frecuentes.

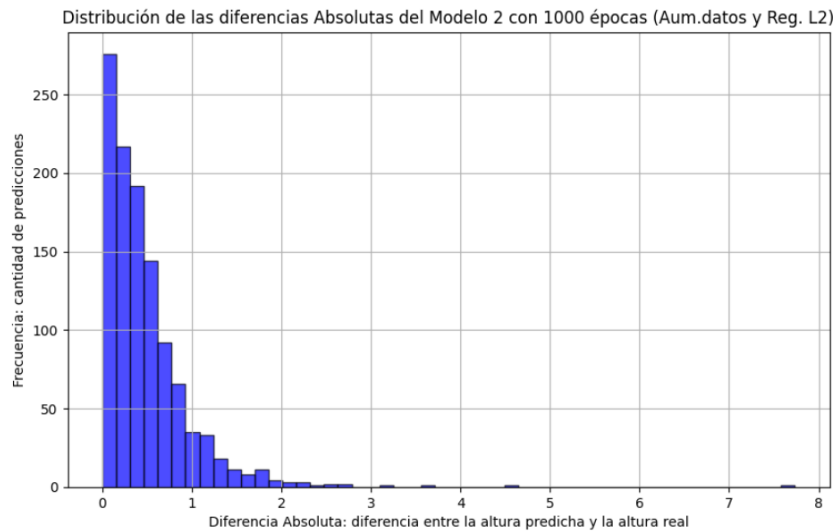


Figura No. 4.42 Distribución de las diferencias Absolutas del Modelo 2 ($M2_L2C_AUM$) con 1.000 épocas (Aum. datos y Reg. $L2$)

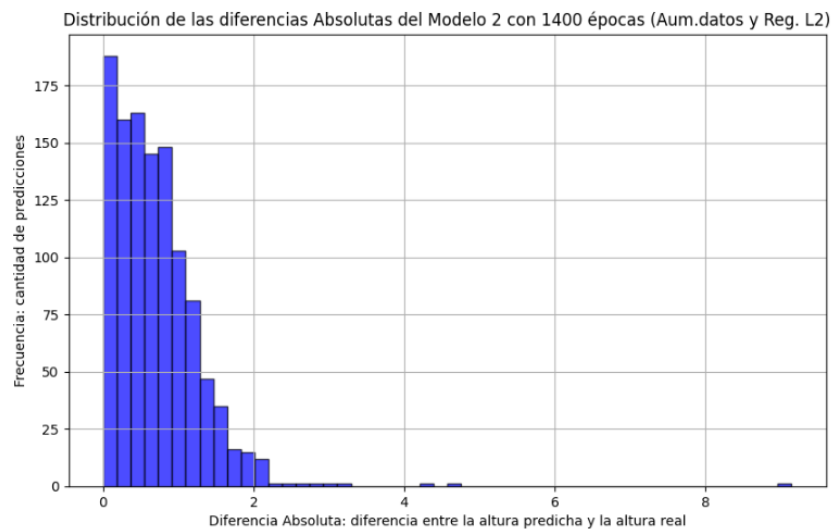


Figura No. 4.43 Distribución de las diferencias Absolutas del Modelo 2 ($M2_L2C_AUM$) con 1.400 épocas (Aum. datos y Reg. $L2$)

El modelo 2 (una entrada) $M1_L2C_AUM$ tiene un coeficiente de determinación y coeficiente de correlación de *Pearson* altos, errores de validación bajos, que es muy bueno porque indica que está generalizando bien. De otra parte, el modelo 2 (dos entradas) $M2_L2C_AUM$ muestra un patrón similar de alto rendimiento con bajos errores de validación, un coeficiente correlación y coeficiente de correlación de *Pearson* altos, que según la experimentación la regularización $L2$ con la entrada dual, contribuyen positivamente a la capacidad del modelo para realizar predicciones más precisas.

De estos dos modelos con mejor desempeño el *M2_L2C_AUM* puede llegar a considerarse el mejor modelo en general, dado que el coeficiente de determinación R^2 y el coeficiente de correlación de *Pearson* son ligeramente más altos, y también por que presenta un rendimiento más consistente tanto en entrenamiento como en la validación, a pesar de ser un modelo más complejo con dos entradas. Por lo tanto, este modelo es la mejor opción para estimar la altura de un objeto contenido en una imagen *2D* a partir de su sombra, por la capacidad de generalización y de precisión predictiva obtenidas. De igual forma el número de *outliers* en cada uno de éstos modelos pueden llevar a un análisis más profundo con pruebas en otras arquitecturas de redes neuronales.

Capítulo V. Conclusiones y Trabajos Futuros

5.1. Conclusiones

Con este estudio se logró el desarrollo exitoso de un conjunto de datos, compuesto por las fotografías etiquetadas de formas u objetos uniformes, junto con la implementación de un modelo de red neuronal convolucional (CNN) para regresión. Esto fue posible desarrollarlo mediante la creación de un ambiente controlado para el diseño experimental, pruebas exhaustivas y retroalimentación continua entre otros. Por tanto, al emprender este tipo de investigaciones es esencial la selección apropiada de herramientas y técnicas, así como la realización de un proceso iterativo de prueba y error, como un componente crucial en la mejora del sistema.

A partir del análisis comparativo de los modelos de redes neuronales convolucionales presentados, se pueden derivar las siguientes observaciones: los modelos desarrollados en esta investigación muestran diversos grados de generalización. Es decir, este es un indicador de la calidad y de la selección del modelo, porque un modelo útil no solo debe ajustarse bien a los datos con los que se entrenó, sino también funcionar eficientemente con datos no vistos previamente sin una pérdida significativa de precisión. El rendimiento predictivo del modelo M2_L2C_AUM evaluado por el coeficiente de determinación ($R^2 = 0.93$) y el coeficiente de correlación de *Pearson* ($r = 0.97$), destaca la relación entre las variables independientes y dependientes del conjunto de datos, que es un indicador de que el modelo tiene buena precisión en la predicción. Igualmente, en $MAE = 0.58$ y el $MSE = 0.99$ permitieron mantener una buena capacidad de generalización con datos nuevos.

Un criterio de relevancia es el margen de error en los modelos, dado que en ciertas aplicaciones en la cuales se requiera de una gran precisión, pequeños errores de predicción pueden llegar a tener consecuencias significativas. Por tanto, los modelos utilizados para las predicciones de las alturas de las figuras a partir de su sombra fueron diseñados específicamente para predecir alturas de figuras en un rango limitado, en este caso de 1 a 10 *cm*, donde la capacidad de estos para predecir correctamente alturas es influenciada por la características intrínseca de los datos, la arquitectura, la complejidad del modelo y de los métodos aplicados durante su entrenamiento. Esto significa que los modelos son buenos para capturar y aprender las relaciones entre sombras y alturas dentro de este rango de medidas.

Para esta investigación se utilizó un rango de medidas de 1 a 10 *cm*. Pero es importante indicar que es necesario incluir dentro del conjunto de datos más variedad en los valores de la altura de las figuras, con el propósito de mejorar la capacidad de

extrapolación, que en este caso es la capacidad del modelo para predecir valores fuera del rango observado en el entrenamiento. El modelo M2_L2C_AUM es la opción más destacada, ya que combina un sólido conjunto de características, valores altos en el coeficiente de determinación R^2 y el coeficiente de correlación de *Pearson* (r), con métricas de error bajas y coherencia entre el entrenamiento y la validación, que hace que el modelo tenga el mejor equilibrio entre precisión y complejidad. Cabe destacar que la selección del modelo se toma basado en los resultados obtenidos y con las métricas establecidas, pero debe considerarse y tenerse en cuenta los recursos computacionales y el tiempo de entrenamiento. No obstante, este resultado, y el hecho de que el modelo seleccionado implique una combinación de sombra con la figura, hace que su costo computacional sea más alto que el del modelo 1 que solo utiliza la sombra.

Esta investigación ha demostrado que la integración de un *dataframe* detallado, que contiene las características relevantes, asociadas con cada imagen que conforma el *dataset* (las imágenes de las sombras y de las figuras), han permitido obtener mejores inferencias por parte de los modelos 1 y 2 propuestos, ya que la combinación de información visual con datos estructurados ha permitido que los modelos por medio de las redes neuronales convolucionales, puedan inferir de forma precisa las relaciones entre las sombras y las alturas de las figuras, dado que al tener acceso a información contextual adicional sobre cada imagen, los modelos pueden ajustar sus predicciones de forma más efectiva, lo que resulta en una generalización mejorada. Por tanto, esta investigación resalta el potencial que tienen los datos multimodales en las redes neuronales convolucionales, que permiten a los modelos alcanzar una buena capacidad de discernimiento, permitiéndoles realizar tareas específicas, como la estimación de la altura a partir de objetos contenidos en una imagen *2D* a partir de sus sombras, sino que también establece un precedente para el desarrollo de sistemas con la capacidad de interpretar complejidades en una gran variedad más de contextos y de aplicaciones.

La investigación abre la posibilidad de futuras mejoras en los modelos, ya sea a través de la expansión del conjunto de datos, el ajuste de la arquitectura del modelo o la implementación de otras técnicas más avanzadas de aprendizaje automático, que mejoren la generalización sin comprometer la precisión dentro del rango de entrenamiento. Estas mejoras podrían permitir a los modelos manejar un espectro más amplio de alturas con mayor precisión.

5.2. Trabajos futuros

Para un trabajo posterior o trabajo futuro es importante desarrollar un método que, a través de las sombras proyectadas de un objeto o *forma irregular*, en una imagen *2D* se pueda predecir el valor estimado de la altura de la superficie del mismo. Esto implicaría que el *dataset* deberá contener tanto formas uniformes, como de formas irregulares con sus sombras e intensidades respectivas.

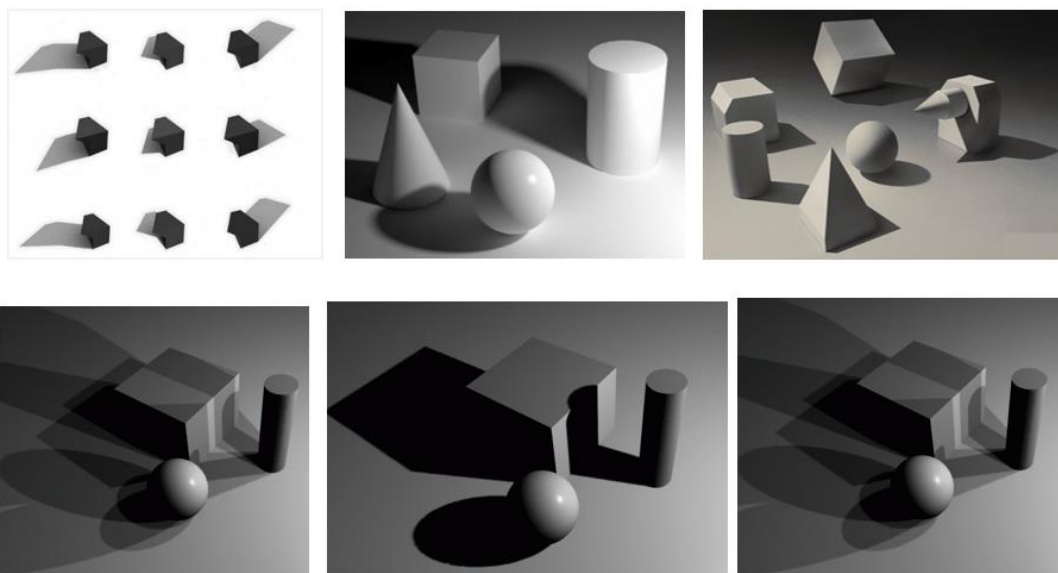


Figura 5.1. Imágenes no uniformes con sombras proyectadas

Referencias bibliográficas

Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* (2021). *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*. J Big Data. Vol 8, Issue 53.

Al-Najdawi, N. *et al.* (2012). *A survey of cast shadow detection algorithms*, *Pattern Recognition Letters*, vol 33, Issue 6, pp. 752–764.

ArtRocket (2021). Guía para dibujar sombras. Disponible en: (Accesado en febrero 2021).

Alzubaidi, L. *et al.* (2021). *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, *Journal of Big Data*.

Alzubaidi, L. *et al.* (2021). *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, *Journal of Big Data*. Springer International Publishing.

Alzubaidi, L. *et al.* (2021). *Review of Deep Learning: concepts, CNN architectures, challenges, applications, future directions*, *Journal of Big Data*. Vol 8, Issue 53.

Asaf Karnieli, Ohad Fried, and Yacov Hel-Or. (2022). *DeepShadow: Neural Shape from Shadow*. In *Computer Vision – ECCV 2022: 17th European Conference*, Tel Aviv, Israel, Proceedings, Part II. Springer-Verlag, Berlin, Heidelberg, pp. 415–430.

Atkinson, G.A. & Hancock, E.R. (2007). *Shape estimation using polarization and shading from two views*. *IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol 29, Issue 11, pp. 2001-2017.

Berzal, F. (2018). *Redes Neuronales & Deep Learning*. Edición en español.

Blancolam S, Selin V, Park SW, Tamblyn I. (2021). *Correspondence between neuroevolution and gradient descent*. *Nat Commun.*, Vol 2, Issue 12(1). pp. 6317.

Box, G.E.P., (1976). *Science and statistics*. *Journal of the American Statistical Association*, Vol 71, Issue 356, pp.791-799.

Box, G.E.P., (1979). *Robustness in the strategy of scientific model building*. En: *R.L. Launer & G.N. Wilkinson, eds. Robustness in Statistics*. Academic Press, pp. 201-236.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, Vol 24, Issue 2, pp. 123-140.

Breiman, L. (2001). Random forests. *Machine Learning*, Vol 45, Issue 1, pp. 5-32.

Bye, S.J. and Adams, A. (1993), *Neural network paradigm for visual pattern recognition*, *IEE Conference Publication*, Issue 372, pp. 11–15.

Cavanaugh, T. W., & Eastham, N. P. (2019). *Creating Tactile Graphs for Students With Visual Impairments: 3D Printing as Assistive Technology*. In Santos, I. M., Ali, N., & Areepattamannil, S. (Ed.), *Interdisciplinary and International Perspectives on 3D Printing in Education*, pp. 223-240.

Chandra, P. & Singh, Y., (2003). *Regularization and feedforward artificial neural network Entrenamiento with noise*. En: *Proceedings of the 2003 International Joint Conference on Neural Networks*, vol3. IEEE, pp. 2366-2371.

Chapelle, O., Schölkopf, B. y Zien, A., eds., (2006). *Semi-Supervised Learning*. Cambridge, MA: The MIT Press.

Chen, Z., Wu, B. and Liu, W.C. (2021). *Mars3DNet: CNN-based resolution 3D reconstruction of the Martian surface from single imágenes*, *Remote Sensing*, 13(5), vol 13, Issue 5, pp. 1–22.

Cui, N. (2018). *Applying Gradient Descent in Convolutional Neural Networks*. *Applying Gradient Descent in Convolutional Neural Networks*.

Dalal, N. y Triggs, B., (2005). *Histograms of oriented gradients for human detection*. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, EE. UU, vol. 1, pp. 886-893.

Dare, P.M. (2005). *Shadow Analysis in High-Resolution Satellite Imagery of Urban Areas*. **Photogrammetric Engineering and Remote Sensing**, Vol 71, pp. 169-177.

Datta, S. et al. (2022). *Neural Shadow Mapping*, *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '22 Conference Proceedings)*, Vancouver, BC, Canada. Association for Computing Machinery. Vol 1, Issue 1, pp. 1-9.

Daum, M. & Dudek, G. (1998). *On 3-D surface reconstruction using shape from sombras*. In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)**, pp. 461-468.

Du, K.-L.; Swamy, M.N.S. (2019). *Fundamentals of Machine Learning*. Neural Net. Stat. Learn, pp. 21–63.

- Efron, B. & Tibshirani, R. (1997). *Improvements on Cross-validation: The .632+ Bootstrap Method*. *Journal of the American Statistical Association*, Vol 92 Issue 438, pp. 548-560.
- Forsyth, D. y Zisserman, A. (1991). *Reflections on Shading*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, Issue 7, pp. 671-679.
- Friedman, J.H. (1997). On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, Vol 1, Issue 1, pp. 55-77.
- Gardner, M.W. and Dorling, S.R. (1998). *Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences*, *Atmospheric Environment*, vol 32, Issue 14–15, pp. 2627–2636.
- Geman, S., Bienenstock, E. y Doursat, R. (1992). *Neural networks and the bias/variance dilemma*, *Neural Computation*, Vol 4, Issue1, pp. 1-58.
- Gouiaa, R. & Meunier, J. (2014). *3D Reconstruction by Fusing Shadow and Silhouette Information* In *Proceedings of the IEEE Conference on Computer and Robot Vision (CRV'14)*, pp. 378-384.
- Guyon, I. & Elisseeff, A. (2003). *An Introduction to Variable and Feature Selection*. *Journal of Machine Learning Research*, Vol 3, Issue1157-1182.
- Guyon, I. & Elisseeff, A., (2003). *An Introduction to Variable and Feature Selection*. *Journal of Machine Learning Research*, Vol 3, pp.1157-1182.
- Haoran Wu, Zhiyong Xu, Jianlin Zhang, Wei Yan, and Xiao Ma. (2017). *Face recognition based on convolution siamese networks*. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5.
- Hintze, R. & Morse, B. (2019). *Shadow patching: guided image completion for shadow removal*. In **Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)**, pp. 1999–2008.
- Hirahara D, Takaya E, Takahara T, Ueda T. (2020). *Effects of data count and image scaling on deep learning Entrenamiento*. *PeerJ Comput Sci*. Vol 6. pp. 312.
- Hopfield, J.J. (1982). *Neural networks and physical systems with emergent collective computational abilities.*, *Proceedings of the National Academy of Sciences of the United States of America*, Vol 79, Issue 8, pp. 2554–2558.

Hosseinzadeh, S., Shakeri, M. y Zhang, H. (2018). *Fast Shadow Detection from a Single Image Using a Patched Convolutional Neural Network*. *Natural Science and Engineering Research Council (NSERC) through the NSERC Canadian Field Robotics Network (NCFRN) and by Alberta Innovates Technology Future (AITF)*, pp. 3124-3129.

Hosseinzadeh, S., Shakeri, M., & Zhang, H. (2018). *Fast shadow detection from a single image using a patched convolutional neural network*. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'18)*, pp. 3124-3129.

Hu, X. et al. (2021). *Model complexity of deep learning: a survey*, *Knowledge and Information Systems*, vol 63, issue 10, pp. 2585–2619.

Huang, X., Gao, J., Wang, L., & Yang, R. (2007). *Exemplar-based shape from shading*. In *Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling (DIM 2007)**, pp. 349-356.

Indrakumari, R.; Poongodi, T.; Singh, K. (2021). *Introduction to Deep Learning*. *EAI/Springer Innov. Commun. Comput.* pp. 1–22.

Inoue, N. and Yamasaki, T. (2021). *Learning from Synthetic Sombras for Shadow Detection and Removal*, *IEEE Transactions on Circuits and Systems for Video Technology*, vol 31, Issue 11, pp. 4187–4197.

Ioffe, S. and Szegedy, C. (2015). *Batch normalization: Accelerating deep network Entrenamiento by reducing internal covariate shift*, *32nd International Conference on Machine Learning, ICML, Vol 1*, pp. 448–456.

Kamnitsas, K., Bai, W., Ferrante, E., McDonagh, S., Sinclair, M., Pawlowski, N., et al. (2017). *Ensembles of multiple models and architectures for robust brain tumour segmentation*. In *International MICCAI Brain lesion workshop*, pp. 450e462.

Khan, A., Sohail, A., Zahoora, U. et al. (2020), *A survey of the recent architectures of deep convolutional neural networks*, *Artificial Intelligence Review*, Vol 53, pp. 5455–5516.

Khan, S.H., Bennamoun, M., Sohel, F., & Togneri, R. (2014). *Automatic feature learning for robust shadow detection*. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'14)*, pp. 1939-1946.

Khvostikov, A. et al. (2018), *3D CNN-based classification usings MRI and MD-DTI imágenes for Alzheimer disease studies*.

Khvostikov, A., Aderghal, K., Benois-Pineau, J., Krylov, A., & Catheline, G. (2018). *3D CNN-based classification using sMRI and MD-DTI imágenes for Alzheimer disease studies*.

Kim, D.S., Arsalan, M., & Park, K.R. (2018). *Convolutional neural network-based shadow detection in imágenes using visible light*. *Sensors*, Vol 18, Issue 4, pp. 960.

Knill, D.C., Mamassian, P., & Kersten, D. (1997). *Geometry of sombras*. *Journal of the Optical Society of America A*, Vol 14, Issue 12, pp. 3216.

Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection*. *En Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, Vol 14, Issue 12, pp.1137-43.

Kriegman, D.J. et al. (1999). *3D Photography Using Sombras in Dual-Space Geometry*. *Lecture Notes in Computer Scienc*, Vol 35, Issue 2, pp. 129-149.

Kumar, A., Yin, B., Shaikh, A.M. et al. (2022). *CorrNet: pearson correlation based pruning for efficient convolutional neural networks*. *Int. J. Mach. Learn. & Cyber*. Vol. 13, pp.3773–3783.

Kunihiko Fukushima. (1975). *Cognitron: A self-organizing multilayered neural network*. *Biological Cybernetics*, Vol 20, Issue 3, pp. 121-136.

Kunihiko Fukushima. (1980) *Neocognitron: A self-organizing neural network model for a mechanism of pattern re-cognition unaffected by shift in position*. *Biological Cybernetics*, Vol 36, issue 4, pp. 193-202.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). *Gradient-based learning applied to document recognition*. *Proceedings of the IEEE*, Vol 86, Issue 11, 2278e2324.

León Yacelga, R.A., Acosta Espinoza, J.L. and Díaz Vásquez, R.A. (2021). *Aplicación de la metodología incremental en el desarrollo de sistemas de información*, *Universidad y Sociedad*, vol. 13, pp. 175–182.

Leung, C.-S., Woner, K.-W., Sum, P.-F. & Chan, L.-W., (2001). *A pruning method for the recursive least squared algorithm*. *Neural Networks*, Vol 14, Issue 2, pp.147-174.

Liasis, G. & Stavrou, S. (2016). *Satellite Imágenes Analysis for Shadow Detection and Building Height Estimation*. *ISPRS Journal of Photogramme*, Vol 119, 437-450.

Liu, H., Hussain, F., Tan, C.L. y Dash, M., (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, Vol 6, Issue 4, pp.393-423.

Liu, R., Menon, S., Mao, C., Park, D., Stent, S., & Vondrick, C. (2022). *Sombras shed light on 3D objects*.

Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully convolutional networks for semantic segmentation*. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431-3440.

Lopez Pinaya, W.H. et al. (2020). *Convolutional neural networks, Machine Learning: Methods and Applications to Brain Disorders*. pp. 173-191.

Luo, S., Li, H. and Shen, H. (2020). *Deeply supervised convolutional neural network for shadow detection based on a novel aerial shadow imagery dataset*, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol 167, pp. 443–457.

Markatou, M., Tian, H., Biswas, S., & Hripcsak, G.M. (2005). Analysis of Variance of Cross-validation Estimators of the Generalization Error. *Journal of Machine Learning Research*, Vol 6, pp. 1127-1168.

Martin, R.C. (2014) *A Programming Episode Design, Agile Software Development: Principles, Patterns, and Practices*.

Martínez, A.D. et al. (2021). *Lights y sombras in Evolutionary Deep Learning: Taxonomy, critical methodological analysis, cases of study, learned lessons, recommendations and challenges*, *Information Fusion*, vol 67, pp. 161–194.

Memeo, M. et al. (2021). *Enabling visually impaired people to learn three-dimensional tactile graphics with a 3DOF haptic mouse*, *Journal of NeuroEngineering and Rehabilitation*, vol 18, Issue 1, pp. 1–21.

Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., et al. (2015). *The multimodal brain tumor image segmentation benchmark (BRATS)*. *IEEE Transactions on Medical Imaging*, Vol 34, Issue 10, pp.

Mescheder, L. et al. (2019). *Occupancy networks: Learning 3D reconstruction in function space*, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4455–4465.

Miceli, P.A., Blair, W.D. and Marrón, M.M. (2018) *Isolating Random and Bias Covariances in Tracks*, *21st International Conference on Information Fusion*.

Mnih, V., Kavukcuoglu, M., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M.A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. y Hassabis, D., (2015). *Human-level control through deep reinforcement learning*. Nature, Vol 518, Issue 7540, pp.529-533.

Mohajerani, S. & Saeedi, P. (2018). *CPNet: A Context Preserver Convolutional Neural Network for Detecting Sombras in Single RGB Imágenes*. In **Proceedings of the IEEE 20th International Workshop on Multimedia Signal P*, pp. 1-5.

Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M. y Bowling, M., (2017). *Deepstack: Expert-level artificial intelligence in heads-up no-limit poker*. Science, 356(6337), pp.508-513.

Ng, W.W.Y. & Yeung, D.S. (2002). *Input dimensionality reduction for radial basis neural network classification problems using sensitivity measure*. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol.4, pp. 2214-2219.

Ng, W.W.Y. & Yeung, D.S., (2002). *Input dimensionality reduction for radial basis neural network classification problems using sensitivity measure*. En: *Proceedings. International Conference on Machine Learning and Cybernetics*, Beijing, China. IEEE, vol.4, pp. 2214-2219.

Panagopoulos, A., Hadap, S. y Samaras, D. (2013). *Reconstructing shape from dictionaries of shading primitives*, *Lecture Notes in Computer Science*,

Paulus, R., Xiong, C. y Socher, R., (2017). *A deep reinforced model for abstractive summarization*.

Pentland, A. (1988). *Shape Information from Shading: A theory about Human Perception*. In *Proceedings of the International Conference on Computer Vision*, pp. 404-413.

Pérez-D'Arpino, C. y Shah, J.A., (2017). *C-LEARN: Learning Geometric Constraints from Demonstrations for Multi-Step Manipulation in Shan red Autonomy*. En *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA'2017)*, pp. 4058-4065.

Prados, E., & Faugeras, O. (2006). *Shape from shading*. In N. Paragios, Y. Chen, & O. Faugeras (Eds.), *Handbook of Mathematical Models in Computer Vision*, pp. 375-388. Springer.

Prince, S. (2012). **Computer vision: models, learning, and inference**. Cambridge University Press.

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-net: Convolutional networks for biomedical image segmentation*. In *International Conference on Medical image computing and computer- assisted intervention*, pp. 234e241).

Saleh AM, Hamoud T. (2021). *Analysis and best parameters selection for person recognition based on gait model using CNN algorithm and image augmentation*. *J Big Data*. Vol 8. Issue 1. pp.1–20.

Salvador, E., Cavallaro, A., & Ebrahimi, T. (2001). *Shadow Identification And Classification Using Invariant Color Models*. In *ICASSP*.

Salvador, E., Cavallaro, A., & Ebrahimi, T. (2004). *Cast shadow segmentation using invariant color features*. *Computer Vision and Image Understanding*, Vol 95, Issue 2, pp. 238-259.

Salvador, E., Cavallaro, A., & Ebrahimi, T. (2004). *Cast Shadow Segmentation Using Invariant Color Features*. *Computer Vision and Image Understanding*, Vol 95, Issue 2, pp. 238-259.

Samaras, D. & Metaxas, D. (2003). *Incorporating illumination constraints in deformable models for shape from shading and light direction estimation*. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol 25, Issue 2, pp. 247-264.

Sanin, A., Sanderson, C. & Lovell, B.C. (2012). *Shadow Detection: A Survey and Comparative Evaluation of Recent Methods*. **Pattern**, Vol 45, Issue 4, pp. 1684-1695.

Sanin, A., Sanderson, C. and Lovell, B.C. (2012). *Shadow detection: A survey and comparative evaluation of recent methods*, *Pattern Recognition*, vol 45, Issue 4, pp. 1684–1695.

Sarker, I.H. (2021). *Machine Learning: Algorithms, Real-World Applications, and Research Directions*. *SN Comput. Sci*, Vol 2, pp. 1–21.

Sato, I., Sato, Y. and Ikeuchi, K. (1999) *Illumination distribución from sombras*, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pp. 306-312.

Savarese, S., & Rushmeier, H.E. (2002). *Implementation of a shadow carving system for shape capture*. In *Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission*, pp. 12-23.

Savarese, S., et al. (2007). *3D reconstruction by shadow carving: Theory and practical evaluation*. *International Journal of Computer Vision*, Vol 71, Issue 3, pp. 305-306.

Scanlan, J.M., Chabries, D.M., & Christiansen, R.W. (1990). *A Shadow Detection and Removal Algorithm for 2D Images*. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Schulz, A., Hinder, F. and Hammer, B. (2021) *DeepView: Visualizing classification boundaries of deep neural networks as scatter plots using discriminative dimensionality reduction*, IJCAI International Joint Conference on Artificial Intelligence, January, pp. 2305–2311.

Setiono, R. & Azcarraga, A., (2001). *An effective method for generating multiple linear regression rules from artificial neural networks*. En: *Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence*. ICTAI, Dallas, TX, USA. IEEE, Vol 4, pp. 171-178.

Shafer, S.A., & Kanade, T. (1983). *Using sombras in finding surface orientations*. *Computer Vision, Graphics, and Image Processing*, Vol 22, Issue 1, 145-176.

Sharma, N. y Sharma, N. (2018). *An analysis of convolutional neural networks for image classification*, *Procedia Computer Science*, 132(Iccids), pp. 377–384

Shilpa, M., Gopalakrishna, M.T. and Naveena, C. (2020). *Approach for shadow detection and removal using Machine Learning techniques*, *IET Image Processing*, vol 14, Issue 13, pp. 3161–3168.

Shorten C, Khoshgoftaar TM. (2019). *A survey on image data augmentation for deep learning*. *J Big Data*. Vol 6, Issue 1, pp. 60.

Sill, J. et al. (2009). *Feature-Weighted Linear Stacking*, pp. 1–17.

Sill, J., Takacs, G., Mackey, L.W., & Lin, D. (2009). *Feature-weighted linear stacking*. *arXiv e-prints*.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. y Hassabis, D., (2016). *Mastering the game of Go with deep neural networks and tree search*. *Nature*, Vol 529, Issue 7587, pp.484-489.

Singh, A.K. and Pandit, A. (2019). *Shadow Detection and Removal Techniques: A Perspective View*, *International Journal of Trend in Scientific Research and Development (IJTSRD)*, vol 3, Issue 4, pp. 1596–1599.

Solomon, C. y Breckon, T. (2011). *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. John Wiley & Sons.

Sutton, R.S. y Barto, A.G., (1998). *Introduction to Reinforcement Learning*. 1st ed. Cambridge, MA: MIT Press.

Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C. (2018). *A survey on deep transfer learning*. In: *International conference on artificial neural networks*. pp. 270–9.

Tao, M.W., Srinivasan, P.P., Hadap, S., Rusinkiewicz, S., Malik, J., & Ramamoorthi, R. (2017). *Shape Estimation from shading, defocus, and correspondence using light-field angular coherence*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 39, issue 3, pp. 546–560.

Tenthoff, Moritz, Kay Wohlfarth, and Christian Wöhler. (2020). *High Resolution Digital Terrain Models of Mercury Remote Sensing*, Vol 12, Issue 23: pp. 1-37.

Tian, Y. and Zhang, Y. (2022) 'A comprehensive survey on regularization strategies in machine learning', *Information Fusion*, Vol 80, pp. 146–166.

Tiwary, K., Klinghoffer, T. and Raskar, R. (2022). *Towards Learning Neural Representations from Sombras, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13693 LNCS, pp. 300–316.

Torres I Viñals, J. (2020). *Python Deep Learning: Introducción práctica con Keras y TensorFlow 2*. 1ª ed. Barcelona: Marcombo. ISBN 9788426728289.

Varol, A., Shaji, A., Salzmann, M., & Fua, P. (2012). *Monocular 3D reconstruction of locally textured surfaces*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 34, Issue 6, pp. 1118-1130.

Vemuri, V.K. (2020). *The Hundred-Page Machine Learning Book*, *Journal of Information Technology Case and Application Research*, Vol 2, Issue 2, pp. 136–138.

Vicente, T.F.Y., Yu, C.-P. y Samaras, D. (2013). *Single image shadow detection using multiple cues in a supermodular MRF*, in *Proceedings of the British Machine Vision Conference*, pp. 126.1-126.11.

Viering, T. and Loog, M. (2023). *The Shape of Learning Curves: A Review*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, Issue 6, pp. 7799–7819.

Weber, M., Perona, P. & Bouguet, J., (1999). *What Do Planar Sombras Tell About Scene Geometry? En: 2013 IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado. IEEE, Vol 1, pp. 514-520.

Weiss K, Khoshgoftaar TM, Wang D. (2016). *A survey of transfer learning*. J Big Data. Vol 3, Issue 1, pp. 9.

Weissgerber, T.L. et al. (2015). *Beyond Bar and Line Graphs: Time for a New Data Presentation Paradigm*, PLoS Biology, vol 13, Issue 4, pp. 1–11.

Wojtowysch, S. & E, W. (2020). *Can Shallow Neural Networks Beat the Curse of Dimensionality? A Mean Field Entrenamiento Perspective*. *IEEE Transactions on Artificial Intelligence*, Vol 1, Issue 2, pp. 121-129.

Wojtowysch, S. & E, W., (2020). *Can Shallow Neural Networks Beat the Curse of Dimensionality? A Mean Field Entrenamiento Perspective*. *IEEE Transactions on Artificial Intelligence*, Vol 1, Issue 2, pp.121-129.

Wu, Y., Pan, G., Lu, D. & Yu, M. (2022). *Artificial Neural Network for Dimensionality Reduction and Its Application to Microwave Filters Inverse Modeling*. *IEEE Transactions on Microwave Theory and Techniques*, Vol 70, Issue 11, pp. 4683-4693.

Wu, Y., Pan, G., Lu, D. & Yu, M., (2022). *Artificial Neural Network for Dimensionality Reduction and Its Application to Microwave Filters Inverse Modeling*. *IEEE Transactions on Microwave Theory and Techniques*, 70(11), pp. 4683-4693.

Y. LeCun, Y. Bengio, and G. Hinton. (2015). *Deep learning*, *Nature*, vol. 521, Issue 7553, pp. 436–444.

Yadav, R., Nascetti, A. & Ban, Y., (2023). *A CNN regression model to estimate buildings height maps using Sentinel-1 SAR and Sentinel-2 MSI time series*.

Yang, Q., Zhang, Y., Dai, W. y Pan, S. J. (2020). *Learning transfer*. Cambridge University Press.

Yao, J. & Zhang, Z. (2004). *Systematic Static Shadow Detection*. In **17th International Conference on Pattern Recognition (ICPR)**.

Yuniarti, A. and Suciati, N. (2019) *Reconstruction of 2D Images*, *2019 12th International Conference on Information & Communication Technology and System (ICTS)*.

Zhang, D. *et al.* (2012). *Sparse patch-based label fusion for multi-atlas segmentation*, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7509 LNCS, pp. 94–102.

Zhang, J. and Liu, J. (2022). *A novel single robot image shadow detection method based on convolutional block attention module and unsupervised learning network*, *Frontiers in Neurorobotics*, vol 16.

Zhang, T. *et al.* (2018). *Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation*, *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5628–5635.

Zhu, X., Goldberg, A.B., (2009). *Introduction to Semi-Supervised Learning*. *Morgan and Claypool Publishers*.

Hecht, E. *et al.* (2016) *Global Edition, The Physics Teacher*. Available at: www.pearsonglobaleditions.com.

ANEXO A. Elementos principales de la maqueta

- Cámara de 64 Megapíxeles
- Base plana circular: 90 centímetros de diámetro, superficie cubierta con vinilo color blanco mate
- Bombillo led dimerizable: se utilizó un bombillo led 840 lúmenes dimerizable de luz blanca de 10 watts
- Trípode para cámara fotográfica profesional
- Trípode para lámpara led profesional



Figura A.1. Trípode, Lámpara y bombilla Led dimerizable



Figura A.2. Dimer

ANEXO B. Estructura general del algoritmo

```
# Carga el DataFrame
data_frame = cargar_data_frame ()

# Divide el DataFrame
conjunto_entrenamiento, conjunto_validacion, conjunto_prueba =
dividir_data_frame(data_frame)

# Crea y configura la red neuronal
red_neuronal = crear_red_neuronal ()
red_neuronal = definir_hiperparametros(red_neuronal)

# Entrena la red neuronal
entrenar_red_neuronal (red_neuronal, conjunto_entrenamiento)

# Evalúa la red neuronal en el conjunto de validación
metricas_validacion = evaluar_red_neuronal (red_neuronal, conjunto_validacion)

# Realiza predicciones en el conjunto de prueba
predicciones = predecir_alturas (red_neuronal, conjunto_prueba)

# Compara predicciones con datos reales y calcular métricas
metricas_prueba = comparar_con_datos_reales (predicciones, conjunto_prueba)

# Imprime métricas y analiza resultados
imprimir_metricas (metricas_validacion, metricas_prueba)
analizar_resultados (metricas_validacion, metricas_prueba)
```

ANEXO C. Explicación del modelo No. 1. por capas detallado

Entrada para la imagen de la sombra:

Se define la capa de entrada para procesar la imagen de la sombra.

- Entrada: $input_shadow \in \mathbb{R}^{128 \times 128 \times 3}$
Esta capa recibe imágenes de tamaño 128×128 con 3 canales *RGB*

Primera capa convolucional para la imagen de la sombra:

Convolución y activación:

$$\begin{aligned} Z_{conv1_shadow} &= input_shadow * W_{conv1_shadow} + b_{conv1_shadow} \\ A_{conv1_shadow} &= RELU(Z_{conv1_shadow}) \end{aligned}$$

$input_shadow$ se convoluciona con el filtro W_{conv1_shadow} y luego se le añade el sesgo b_{conv1_shadow}

Se aplica la función de activación *RELU* para introducir no linealidad

Primer MaxPooling

$$A_{pool1_shadow} = MP(A_{conv1_shadow})$$

La operación de *MaxPooling* (*MP*) reduce las dimensiones espaciales (ancho y alto) de la entrada, seleccionando el valor máximo en ventanas de un tamaño definido.

Segunda capa convolucional para la imagen de la sombra:

$$\begin{aligned} Z_{conv2_shadow} &= A_{pool1_shadow} * W_{conv2_shadow} + b_{conv2_shadow} \\ A_{conv2_shadow} &= RELU(Z_{conv2_shadow}) \end{aligned}$$

Se aplica otra capa de convolución a la salida de la operación de *pooling* anterior, y nuevamente se utiliza la función de activación *ReLU*.

Segundo Max Pooling para la imagen de la sombra:

$$A_{pool2_shadow} = MP(A_{conv2_shadow})$$

Una segunda operación de *MaxPooling* se realiza sobre la segunda capa convolucional.

Flatten:

$$A_{flatten_shadow} = FLATTEN(A_{pool2_shadow})$$

Flatten convierte las matrices 3D en vectores 1D para preparar la salida para la capa densa.

Capa densa para regresión de la altura:

$$y^{\wedge} = A_{flatten_shadow} * W_{dense_height} + b_{dense_height}$$

donde, W_{dense_height} son los pesos de la capa densa y b_{dense_height} es el sesgo de la capa densa y y^{\wedge} es la salida final del modelo, que representa una regresión de la altura basada en la imagen de entrada. El modelo procesa una imagen de sombra a través de capas convolucionales y de *pooling*, aplanando la salida y finalmente pasándola a través de una capa densa para obtener una regresión de la altura. Los parámetros de la red W y b se aprenden durante el entrenamiento.

ANEXO D. Explicación del modelo No. 2. por capas detallado

Es la arquitectura del modelo de red neuronal convolucional en una notación matemática, con explicación por las capas que la componen:

Se definen las entradas de la red:

Es la imagen de la figura. Tiene un tamaño de $128 \times 128 \times 128$ píxeles y 3 canales (correspondientes a los colores *RGB*).

$$X_{figure} \in \mathbb{R}^{128 \times 128 \times 3}$$

Es la sombra de la figura. Tiene un tamaño de 128×128 píxeles y 3 canales

$$X_{shadow} \in \mathbb{R}^{128 \times 128 \times 3}$$

Capas convolucionales para la figura(Figura):

En este bloque se procesa la imagen

$$X_{figure}$$

Primera Convolución y Activación:

$$\begin{aligned} Z_{conv1_figure} &= X_{figure} * W_{conv1_figure} + b_{conv1_figure} \\ A_{conv1_figure} &= RELU(Z_{conv1_figure}) \end{aligned}$$

Aquí, X_{figure} se convoluciona con el filtro W_{conv1_figure} y luego se le añade el sesgo

$$b_{conv1_figure}$$

La función de activación $RELU(x) = \max(0, x)$ se aplica luego de la convolución. Introduce no-linearidad en el modelo y transforma valores negativos a 0

Primer Max Pooling

$$A_{pool_figure} = MP(A_{conv1_figure})$$

La operación de max pooling (MP) reduce las dimensiones espaciales (ancho y alto) de la entrada, seleccionando el valor máximo en ventanas de un tamaño definido.

Segunda Convolución y Activación:

$$\begin{aligned} Z_{conv2_figure} &= A_{pool_figure} * W_{conv2_figure} + b_{conv2_figure} \\ A_{conv2_figure} &= RELU(Z_{conv2_figure}) \end{aligned}$$

Una segunda capa de convolución se aplica a la salida de la operación de pooling anterior. También se aplica la función de activación *ReLU*.

Segundo Max Pooling

$$A_{pool2_figure} = MP(A_{conv2_figure})$$

Otra operación de max pooling se realiza sobre la segunda capa convolucional

Capas convolucionales para la sombra (shadow):

Este bloque procesa la imagen $X_{shadows}$ de manera similar al bloque anterior. Las mismas operaciones de convolución, ReLU y max pooling se aplican, pero con diferentes pesos y sesgos.

Flatten y Concatenación:

Después de procesar ambas imágenes con capas convolucionales y de pooling, se aplanan las salidas y se concatenan

$$\begin{aligned} A_{flatten_figure} &= FLATTEN(A_{pool2_figure}) \\ A_{flatten_shadow} &= FLATTEN(A_{pool2_shadow}) \\ A_{combined} &= [A_{flatten_figure}, A_{flatten_shadow}] \end{aligned}$$

Flatten convierte las matrices 2D o 3D en vectores 1D. Los vectores aplanados de ambas imágenes se concatenan en un vector más grande,

$$A_{combined}$$

Capa Densa para Regresión:

Finalmente, la representación combinada se pasa a través de una capa densa (completamente conectada) para obtener la salida:

$$\hat{y} = A_{combined} * W_{dense} + b_{dense}$$

donde, W_{dense} son los pesos de la capa densa y b_{dense} es el sesgo de la capa densa \hat{y} es la salida final del modelo, que puede representar, por ejemplo, una regresión de algún valor basado en las entradas

ANEXO E. Entorno y herramientas de desarrollo.

E.1. TensorFlow

Desarrollado por *Google*, es un software de computación numérica creado por *Google*, orientado a problemas de *Deep Learning*. *TensorFlow* es una librería desarrollada a partir de la combinación de *C++* y *CUDA*. La principal estructura de datos que se maneja en esta librería son los tensores. Con un tensor se refiere a un conjunto de valores primitivos, números enteros o flotantes, organizados por un *array* de 1 o *N* dimensiones, donde el rango del tensor sería el número de dimensiones¹.

Fundamentos de *TensorFlow*:

- *Tensores y gráficos Computacionales*: En el corazón de *TensorFlow* se encuentran los tensores, que son estructuras multidimensionales que representan los datos en el flujo de trabajo. Los cálculos se organizan en un "gráfico computacional", donde los nodos representan operaciones y los bordes representan los flujos de datos (tensores). Esta representación permite la optimización y paralelización automáticas de los cálculos.
- *API flexible*: *TensorFlow* proporciona una *API* flexible en *Python* que permite a los desarrolladores definir y ejecutar operaciones matemáticas de manera eficiente. Además, presenta una interfaz de alto nivel (*Keras*) que facilita la construcción y entrenamiento de redes neuronales de manera más intuitiva.
- *Entrenamiento y optimización*: *TensorFlow* ofrece optimizadores listos para usar que implementan algoritmos de optimización como el descenso de gradiente estocástico (*SGD*) y sus variantes. Esto permite a los usuarios ajustar los parámetros del modelo y minimizar funciones de pérdida de manera eficiente².
- *GPU y TPU Acceleration*: *TensorFlow* aprovecha la potencia de las unidades de procesamiento gráfico (*GPU*) y las unidades de procesamiento tensorial (*TPU*) para acelerar los cálculos. Esto permite entrenar modelos más grandes y complejos en un tiempo mucho menor.

E.2. Keras

Keras es una biblioteca de aprendizaje profundo de código abierto que opera como una interfaz de alto nivel para la creación y entrenamiento de redes neuronales. Diseñada

¹ <https://www.tensorflow.org/?hl=es-419>

² https://www.tensorflow.org/model_optimization?hl=es-419

para ser intuitiva, eficiente y flexible, *Keras* ha ganado popularidad en la comunidad de aprendizaje automático al proporcionar una manera elegante de implementar arquitecturas de redes neuronales de manera rápida y eficaz.

Características principales de *Keras*:

- *Abstracción de bajo nivel simplificada*: *Keras* elimina gran parte de la complejidad de la implementación de redes neuronales al brindar una abstracción de bajo nivel sobre marcos de trabajo más potentes como *TensorFlow*, *Theano* o *Microsoft Cognitive Toolkit*. Esto permite a los desarrolladores concentrarse en diseñar arquitecturas y definir capas de manera más intuitiva.
- *Modelado de redes neuronales*: *Keras* proporciona una variedad de capas predefinidas, desde capas de convolución hasta capas recurrentes y de agrupación. Estas capas se pueden combinar para construir arquitecturas de redes neuronales complejas. Además, los modelos se pueden construir en modo secuencial o funcional, lo que ofrece flexibilidad en la disposición de las capas.
- *Personalización y experimentación*: A pesar de su facilidad de uso, *Keras* permite una amplia personalización de los modelos. Los usuarios pueden definir sus propias funciones de activación, inicializadores de pesos y funciones de pérdida. Esto facilita la experimentación con diferentes configuraciones para encontrar la arquitectura óptima.
- *Transferencia de aprendizaje y modelos Preentrenados*: se pueden ajustar los modelos a tareas específicas.
- *Compatibilidad multiplataforma*: *Keras* es compatible con varias plataformas de hardware, como *CPU*, *GPU* y *TPU*. Esto permite un rendimiento escalable y optimizado para diversos tipos de cómputo.
- *Integración con TensorFlow y otros Backends*: *Keras* funciona con varios Backends, siendo *TensorFlow* el más popular.

E.3. Matlab

Matlab es un entorno utilizado en áreas como el procesamiento de señales, el análisis numérico y la ingeniería. Dentro del campo del aprendizaje automático y las redes neuronales, *Matlab* ofrece una variedad de funciones y herramientas que facilitan la creación, implementación y evaluación de modelos de redes neuronales, lo que lo convierte en una elección popular para investigadores y profesionales en esta área.

Características principales de *Matlab* aplicadas a redes neuronales:

- Funciones de Redes Neuronales: *Matlab* proporciona un conjunto completo de funciones para crear y entrenar redes neuronales artificiales.
- Caja de Herramientas de aprendizaje profundo: *Matlab* incluye una caja de herramientas de aprendizaje profundo que ofrece capas predefinidas para construir arquitecturas de redes neuronales convolucionales (*CNN*), recurrentes (*RNN*) y autoencoders, entre otros.
- Visualización y depuración: *Matlab* ofrece herramientas de visualización para trazar gráficos de pérdida, gráficos de precisión y otros indicadores clave durante el entrenamiento de la red.
- Optimización y regularización: *Matlab* permite ajustar los hiperparámetros de la red neuronal mediante algoritmos de optimización como el descenso de gradiente estocástico (*SGD*) y optimizadores más avanzados como *Adam*. También ofrece técnicas de regularización para evitar el sobreajuste y mejorar la generalización.
- Funciones de transferencia y métricas de evaluación: los usuarios pueden personalizar las funciones de transferencia en las neuronas y definir métricas de evaluación como precisión, error medio cuadrático y otras para evaluar el rendimiento del modelo.
- Interfaz gráfica de usuario (*GUI*): *Matlab* proporciona una *GUI* para la creación y entrenamiento de redes neuronales sin necesidad de escribir código
- Integración con herramientas de procesamiento de señales: *Matlab* es ampliamente conocido por sus capacidades de procesamiento de señales.

Aplicaciones en Redes Neuronales:

Matlab encuentra aplicación en una variedad de áreas relacionadas con redes neuronales:

- Investigación y desarrollo: investigadores y científicos utilizan *Matlab* para desarrollar y validar prototipos de modelos de redes neuronales en diversas áreas, desde el procesamiento de imágenes hasta el análisis de datos secuenciales.
- Procesamiento de imágenes y visión por computadora: *Matlab* es efectivo para implementar redes neuronales convolucionales (*CNN*) en tareas de

procesamiento de imágenes, como reconocimiento de objetos, segmentación semántica y detección de rostros.

- **Análisis de datos secuenciales:** Matlab se utiliza para construir y evaluar modelos de redes neuronales recurrentes (*RNN*) y *Long Short-Term Memory (LSTM)* en aplicaciones como series temporales, análisis de texto y procesamiento de voz.
- **Procesamiento de señales biomédicas:** en el campo biomédico, *Matlab* es útil para el procesamiento de señales de imágenes médicas y datos biométricos, donde las redes neuronales pueden utilizarse para diagnóstico y detección de anomalías.
- **Aprendizaje por refuerzo:** Matlab puede aplicarse a problemas de aprendizaje por refuerzo, como la formación de agentes inteligentes en entornos simulados.

E.4. TensorBoard

TensorBoard, desarrollado como parte del ecosistema *TensorFlow*, sirve como una herramienta integral de visualización. Sus características incluyen:

- *Gráficos de la arquitectura del modelo:* *TensorBoard* despliega gráficos computacionales, revelando la conexión entre operaciones y tensores..
- *Visualización de métricas en tiempo real:* a medida que el modelo se entrena, *TensorBoard* proporciona visualizaciones en tiempo real de métricas clave como la pérdida y la precisión, facilitando la detección temprana de problemas potenciales, incluido el sobreajuste.
- *Histogramas:* esta funcionalidad proporciona una vista de la distribución de valores tensoriales a lo largo del tiempo, esencial para observar la evolución de los pesos y sesgos.
- *Proyecciones de Embeddings:* para modelos que utilizan embeddings, *TensorBoard* ofrece visualizaciones tridimensionales, ilustrando agrupaciones y relaciones inherentes.
- *Integración con TensorFlow:* la cohesión entre *TensorFlow* y *TensorBoard* permite una integración sin fricciones, posibilitando la recopilación y visualización de datos con mínimas modificaciones en el código.

E.5. Netron

Netron emerge como una herramienta de visualización independiente, adecuada para una variedad de frameworks de aprendizaje automático. Sus características distintivas incluyen:

- *Vista esquemática detallada*: cada capa del modelo, junto con sus conexiones y parámetros, se representa claramente, permitiendo inspecciones detalladas al hacer clic en componentes individuales.
- *Amplia compatibilidad*: más allá de *TensorFlow*, Netron es compatible con una serie de *frameworks*, incluyendo *PyTorch*, *ONNX*, *Keras*, entre otros.
- Usabilidad directa: a diferencia de *TensorBoard*, *Netron* no requiere etapas previas de entrenamiento o configuraciones de servidor. Los usuarios simplemente cargan el modelo y acceden a sus visualizaciones.
- Diversas plataformas: disponible como una aplicación de escritorio y *web*, *Netron* asegura la accesibilidad en múltiples plataformas, incluyendo *Windows*, *macOS* y *Linux*.