

CARACTERIZACIÓN Y ANÁLISIS COMPARATIVO DE TRES MODELOS BASADOS EN LAS TÉCNICAS PCA PARA EL RECONOCIMIENTO DE ROSTROS



Monografía presentada como requisito
parcial para optar por el título de
Ingeniero en Sistemas

**Presenta:
JESUS ALFONSO CERON**

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Ingeniería de Sistemas
Popayán, Junio de 2006**

ABSTRACT

This project be oriented from a approach research with the purpose of studying three models of face recognition selected having as it bases specific criteria that must fulfill these models, between which we have: the model must be for face recognition, use in the recognition process the technique of Principal Components Analysis and must be use a classification system based on Artificial Neuronal Networks. The theoretical subjects related to these models are: Principal Components Analysis (PCA), Digital Processing Images (PDI), Patterns Recognitions and Artificial Neuronal Networks (ANN); these subjects studied thoroughly with the purpose of being able to implement one of the three models that are model FR_SOM, which this based on technique PCA for reduction of dimensionality and extraction of features, and uses the Artificial Neuronal Networks of Self Organized Maps.

The result of the investigation we conclude that the model that uses the self organized maps by faces classification is the one that better behavior in front to presents similar models that use another type of neuronal network. This doesn't guarantee that it is the best model of recognition of faces, but that given the case which certain specifications considered that had to fulfill the models for their study; this allowed concluding that model FR_SOM in front of presented a better result of classification the other two selected models.

Key words: Face Recognition, PCA, Artificial Neuronal Networks.

AGRADECIMIENTOS

En primer lugar quiero agradecer a mi padre EMANUEL CERON GIL que ha sido el mejor ejemplo de vida, creatividad e ingenio que un hijo pudo tener. A mi madre ISAURA LOMBANA BOLAÑOS, que a pesar de no estar más a nuestro lado fue para mi un gran ejemplo de ternura, tolerancia y comprensión, y sé que desde donde esté, me acompaña en el camino para que las cosas salgan como las he deseado.

A mi hermana CARMEN ELENA CERON LOMBANA, que siempre ha estado apoyándome en todo lo que he emprendido. A mi sobrina CAREL DAYANA CERON, que desde que esta a nuestro lado se ha convertido en motivo alegría y que es la prueba viviente de que alguien tan pequeño puede ser mas grande que el propio universo.

A la mujer que amo, ALEXANDRA ÑAÑEZ LOPEZ, que ha sido la persona que desde el momento en que la conocí se convirtió en mi aliciente, mi apoyo, la persona que ha estado a mi lado en todos los momentos difíciles de mi vida, procurando hacerlos mas llevaderos.

A todos mis maestros de quienes aprendí mucho más de lo que me impartieron en clase, gracias por ayudarme a ampliar mi conocimiento y en especial al Ingeniero CARLOS ALBERTO COBOS que al exigir lo mejor de sus alumnos hace que uno logre superar sus propias metas.

A todos mis compañeros que me sirvieron de ejemplo para saber que se debía y que no se debía hacer para poder llegar al punto en el que me encuentro en este momento. En especial a JOSE ANDRES ORTIZ, con quien pude compartir una gran amistad, experiencias y conocimiento.

A mi familia tíos, tías, primos, primas y sobrinos.

ÍNDICE

1. INTRODUCCION	12
1.1. Caracterización y Análisis Comparativo de Tres Modelos Basados en la Técnica PCA para el Reconocimiento de Rostros	12
1.2. Planteamiento	12
1.3. Objetivos	13
1.4. Justificación	13
1.5. Alcance	13
1.6. Organización Del Documento	14
2. MARCO TEORICO	15
2.1. Biometría	15
2.1.1. Introducción	15
2.1.2. Sistemas Biométricos	15
2.1.3. Tipos de sistemas biométricos existentes	16
2.1.4. Funcionamiento Básico de los Sistemas Biométricos	17
2.1.4.1. Verificación	18
2.1.4.2. Identificación	18
2.2. Reconocimiento de Patrones	18
2.2.1. Introducción	18
2.2.2 Reconocimiento Patrones (RP)	20
2.2.3.1. Metodologías para el reconocimiento de patrones	21
2.2.3.1.1 Heurísticas	21
2.2.3.1.2. Matemáticas	21
2.2.3.1.3. Lingüísticas (sintácticas)	21
2.2.3.2. Etapas para el reconocimiento de patrones	21
2.2.2.4. Reconocimiento de Rostros	22
2.2.4.1 Preprocesamiento de la Imagen	23
2.2.4.2. Extracción de características	23
2.2.4.3. Entrenamiento	23
2.2.4.4. Reconocimiento del Rostro	23
2.2.3.5. Técnicas de Reconocimiento de Rostros	24
2.3. Procesamiento Digital de Imágenes	25
2.3.1. Introducción	25
2.3.2. Imágenes Digitales	25
2.3.2.1. Dispositivos de adquisición de Imágenes	26
2.3.2.2. Digitalización	26
2.3.2.3. Almacenamiento de imágenes digitales	28
2.3.3. Procesamiento Digital de Imágenes	30
2.4. Análisis de Principales Componentes (PCA)	30
2.4.1. Introducción	30
2.4.2. Principios del Análisis de Principales Componentes	31

2.4.2.1. Obtención de la matriz con los datos originales	32
2.4.2.2. Ajuste de los datos originales	32
2.4.2.3. Calculo de la matriz de covarianza	33
2.4.2.4. Cálculo de los <i>eigenvalores</i> y los <i>eigenvectores</i> de la matriz de covarianza	34
2.4.2.4.1. <i>Eigenvectores</i>	34
2.4.2.4.2. <i>Eigenvalores</i>	35
2.4.2.5. Búsqueda de los componentes principales y elaboración de la matriz de vectores característicos	36
2.4.2.6. Derivar el nuevo conjunto de datos	36
2.5. Redes Neuronales Artificiales RNA	37
2.5.1. Introducción	37
2.5.2. Fundamentación Biológica	37
2.5.3. Red de Neuronas Artificiales	38
2.5.3.1. Neurona Artificial	38
2.5.3.1.1. Entradas y salidas	39
2.5.3.1.2. La Regla de Propagación	39
2.5.3.1.3. El peso sináptico de la neurona	40
2.5.3.2. Características de las RNA	42
2.5.3.2.1 Topología de las Redes Neuronales	42
2.5.3.2.2 Mecanismo de Aprendizaje	42
2.5.3.3. Estructura Básica de la Red	43
2.5.3.4. Ventajas de las RNA	46
2.5.3.4.1. Aprendizaje Adaptativo	46
2.5.2.4.2. Autoorganización	47
2.5.3.4.3. Tolerancia a Fallos	47
2.5.3.4.4. Operación en Tiempo Real	47
2.5.3.4.5. Fácil inserción dentro de la tecnología existente	47
2.5.3.5 Implementación de RNA	47
2.6. Perceptron Multilayer	48
2.6.1. Introducción	48
2.6.2. Arquitectura del Perceptrón Multicapa	48
2.6.2.1. Diseño de la arquitectura del Perceptrón Multicapa	49
2.6.2.2. Proceso de Aprendizaje del Perceptrón Multicapa	50
2.6.2.3. Actualización de los pesos que unen la capa intermedia con la de salida	52
2.7. Redes Neuronales de Funciones de Base Radial RBF	54
2.7.1. Introducción	54
2.7.2 Topología de la Red de funciones de base radial RBF	54
2.7.3. Método de aprendizaje de las RBF	55
2.7.3.1. Selección de centros	56
2.8. Mapas Auto-Organizados De Kohonen – SOM	58
2.8.1. Introducción	58
2.8.2. Aprendizaje competitivo	59

2.8.3. Aprendizaje no Supervisado	60
2.8.4. Topología de las Redes SOM	60
2.8.5. Método de aprendizaje de las redes SOM	61
3. SELECCIÓN DEL MODELO DE RECONOCIMIENTO DE ROSTROS	63
3.1. Exploración y búsqueda de información	63
3.2. Criterio de selección de los modelos	65
3.3. Modelos Seleccionados	65
3.3.1. <i>Face recognition: a convolutional neural-network approach</i>	65
3.3.2. <i>Face Recognition with Multi-Layer Perceptrons</i>	66
3.3.3. <i>N-Feature Neural Network Human Face Recognition</i>	66
3.4. Caracterización de los modelos seleccionados	66
3.5. Análisis comparativo de los modelos seleccionados	69
3.5.1. Modelo FR_PML	70
3.5.1.1. Descripción del Sistema	70
3.5.1.2. Ventajas y desventajas del Modelo	72
3.5.2. Modelo FR_RBF	72
3.5.2.1. Descripción del Sistema.	72
3.5.2.2. Ventajas y desventajas del Modelo	73
3.5.3. Modelo FR_SOM	74
3.5.3.1. Descripción del Sistema	74
3.5.3.2. Detalles de la simulación	74
3.5.3.3. Ventajas y desventajas del Modelo	75
3.6. Elección del modelo para implementación del Prototipo	75
4. ANÁLISIS DEL SISTEMA	78
4.1. Requerimientos	78
4.1.1. Adquisición de la Imagen	78
4.1.1.1. Entrenamiento General de la Red Neuronal Artificial	78
4.1.1.2. Clasificación Individual de los Rostros	78
4.1.1.3. Imágenes desde Base de Datos	79
4.1.1.4. Imágenes desde Cámara Digital	79
4.1.2. Preprocesamiento de la Imagen	79
4.1.2.1. Filtro de extracción de rostro de la imagen	79
4.1.2.2. Paso de la Imagen a Escala de Grises	79
4.1.2.3. Filtro de Reducción del tamaño de la imagen	79
4.1.2.4. Filtro de Binarización de la imagen	80
4.1.3. Análisis de los Principales Componentes	80
4.1.3.1. Función PCA	80
4.1.3.2. Función de obtención de la matriz de covarianza	80
4.1.3.3. Función del método Jacobi	80
4.1.3.3.1. Función de producto entre matrices	80

4.1.3.3.2. Función de transpuesta de una matriz	80
4.1.4. Entrenamiento de la Red Neuronal Artificial	80
4.1.5. Clasificación de los rostros	80
4.2. Actores	80
4.3. Descripción Funcional del Sistema	82
4.3.1. Análisis de Principales Componentes (<i>PCA</i>)	84
4.3.2. Captura de Imágenes	75
4.3.3. Identificación de Rostros	86
4.3.4. Filtros	87
5. DISEÑO DEL SISTEMA	88
5.1. Arquitectura	88
5.2. Módulos	89
5.2.1. Módulo de Captura	89
5.2.2. Módulo de Almacenamiento de la Imágenes	89
5.2.3. Módulo de Filtros	89
5.2.4. Módulo de Despliegue gráfico	90
5.2.5. Módulo de Análisis de Principales Componentes	90
5.2.6. Módulo de clasificación de los rostros	90
5.3. Diagramas de Clases	90
5.4. Diagrama de Secuencias	91
5.5. Algoritmos	91
5.5.1 Algoritmos de Preprocesamiento	92
5.5.1.1 Algoritmo de Extracción del Rostro	92
5.5.1.2. Algoritmos de Normalización	93
5.5.2. Algoritmos de Análisis de Principales Componentes	95
5.5.3. Algoritmo de Clasificación	100
6. IMPLEMENTACIÓN DEL SISTEMA	102
6.1. Herramientas	102
6.1.1. Hardware	102
6.1.2. Software	103
6.2. Implementación del Prototipo Funcional	103
6.2.1. Filtros de Preprocesamiento	103
6.2.2. Procesamiento de la Imagen	104
6.2.3. Entrenamiento y Clasificación con RNA	106
6.3. Implementación de las Redes Neuronales Artificiales	107
6.3.1. Red Neuronal Artificial SOM	108
7. PRUEBAS Y RESULTADOS	109
7.1. Velocidad de Procesamiento	109
7.1.1. Procesamiento de la Imagen	109
7.1.2. Clasificación del Rostro	109
7.2. Porcentaje de Clasificación Errónea del Rostro	110

7.3. Porcentaje de Acierto en la construcción del Rostro	111
8. ANÁLISIS DE RESULTADOS	112
<hr/>	
8.1. Metas propuestas y Logros alcanzados	112
9. CONCLUSIONES	115
<hr/>	
9.1. Conclusiones Particulares Del Sistema	115
9.2. Líneas Futuras	
9.3. Limitaciones	116
9.4. Conclusiones Generales	117
	117
BIBIOGRAFIA	
	118

ÍNDICE DE IMÁGENES

- Imagen 2.1.1 Porcentajes de Error en aceptación y rechazo
- Imagen 2.2.1 Hemisferios del cerebro
- Imagen 2.2.2 Etapas en un sistema de Reconocimiento de patrones
- Imagen 2.3.1 Imágenes que muestran la calidad visual al variar los bits
- Imagen 2.3.2 Convención de Ejes
- Imagen 2.3.3 Digitalización del primer renglón de una imagen
- Imagen 2.3.4 Sistema de registro por reflexión
- Imagen 2.3.5 Paso de imagen por *Frame Grabber*
- Imagen 2.4.1. Representación de datos en forma de nube
- Imagen 2.5.2 Modelo de Neurona Artificial estándar
- Imagen 2.5.3 Esquema de una Red de tres capas totalmente interconectadas
- Imagen 2.5.4 Equivalencia entre redes al utilizar funciones
- Imagen 2.5.5 Red de dos capas
- Imagen 2.6.1. Arquitectura del Perceptrón Multilayer
- Imagen 2.6.2. Topología de un Perceptrón N-L-M
- Imagen 2.7.1: Topología de las redes de funciones de base radial
- Imagen 2.8.1 Arquitectura del Mapa de Kohonen
- Imagen 2.8.2 Topología de una Red SOM.
- Imagen 2.8.3 Función $g(\cdot)$ gaussiana de la distancia entre las neuronas

ÍNDICE DE TABLAS

- Tabla 2.1. Funciones de Activación habituales
- Tabla 3.1 Caracterización de modelos seleccionados
- Tabla 3.2 Diferentes topologías de red para los mismos datos de entrada
- Tabla 3.3 Porcentaje de Clasificación correcta en el conjunto de testeo
- Tabla 3.4 Análisis comparativo de los modelos
- Tabla 3.5. Factores e Indicadores de referencia para el análisis del documento “*Face recognition: a convolutional neural-network approach*”
- Tabla 4.1 Caso de Uso PCA
- Tabla 4.2 Caso de Uso Captura de Imágenes
- Tabla 4.3 Caso de Uso Identificación de Rostros
- Tabla 4.4 Caso de Uso Filtros
- Tabla 7.1 Mediciones de Velocidad de procesamiento
- Tabla 7.2 Resultados prueba de clasificación
- Tabla 8.1 Relación de Metas y Logros del proyecto

CAPÍTULO 1: INTRODUCCION

1. INTRODUCCION

En este primer capítulo se presenta una introducción al trabajo realizado en la tesis, se comienza presentando un resumen de lo que comprende el proyecto, después se expone el planteamiento, los objetivos que se fijaron, la justificación, el alcance que delimitó el proyecto y finalmente la organización del documento.

1.1. CARACTERIZACIÓN Y ANÁLISIS COMPARATIVO DE TRES MODELOS BASADOS EN LA TÉCNICA PCA PARA EL RECONOCIMIENTO DE ROSTROS

Este proyecto consiste en la exploración del campo del reconocimiento de rostros en busca de tres modelos diseñados para la implementación de software de clasificación de rostros humanos. Los modelos seleccionados deberán utilizar como base de su implementación el uso de la técnica¹ del Análisis de Principales componentes para la reducción de dimensionalidad de las imágenes así como su caracterización para un posterior entrenamiento de un sistema de clasificación, deben también los modelos usar algún tipo de Red Neuronal Artificial para la clasificación de los rostros.

La exploración y selección de los modelos se realizará teniendo en cuenta los criterios del equipo de trabajo, para determinar cuales son los modelos que se adecuan al proyecto, y la disponibilidad de información de cada uno de los modelos obtenidos.

1.2. PLANTEAMIENTO

La seguridad es hoy en día un campo de interés que atrae muchos sectores de la investigación, y es un tema que se esta volviendo obligado en la mayoría de las empresas tanto privadas como estatales.

La biometría se presenta como una opción bastante viable para la solución de problemas relacionados con la seguridad, como lo es la identificación de las personas, y dentro de la identificación y reconocimiento de las personas encontramos un área bastante interesante como lo es el reconocimiento de rostros.

Dentro del reconocimiento de rostros existen muchos modelos y metodologías utilizadas para la implementación de aplicaciones para este fin, sin embargo son escasos los estudios realizados para determinar cual es el modelo que más se ajusta a unas determinadas condiciones de uso. De ahí, surge la idea de explorar los modelos existentes para el reconocimiento de rostros, de seleccionar un

¹ Palabra de origen griego que significa teknikos, de tekné, arte.

conjunto de tres de ellos y hacer un estudio comparativo de estos modelos con el fin de determinar cual es el modelo que mejor se desempeño presenta en diferentes ambientes específicos.

1.3. OBJETIVOS

Los objetivos fijados para el trabajo realizado en esta tesis son:

- Identificar y estudiar tres modelos de reconocimiento de rostros basados en la técnica PCA, y seleccionar de acuerdo a los resultados uno de los tres modelos para su implementación.
- Desarrollar un prototipo software para el reconocimiento de rostros en dos dimensiones basado en el modelo que utiliza la técnica PCA.
- Establecer el grado acierto y tiempo de respuesta del prototipo implementado.

1.4. JUSTIFICACION

Esta tesis es el resultado del desarrollo del proyecto propuesto dentro del área de investigación de Inteligencia Artificial, y que da una pauta para la incursión de la investigación en áreas diferentes como lo son el procesamiento digital de imágenes, la estadística y las RNA, para hacer una integración de ellas y poder obtener resultados satisfactorios dentro de los modelos para reconocimiento de rostros que involucran estas tres áreas.

1.5. ALCANCE

Se deben analizar los modelos existentes de reconocimiento de rostros que utilicen la técnica de Análisis de Principales componentes, que trabajen con imágenes bidimensionales, y que involucren en su implementación RNA, de esta exploración y análisis resultaran seleccionados tres modelos que cumplan con las condiciones básicas establecidas por el equipo de trabajo. Una vez filtrados y seleccionados los tres modelos se procederá a su estudio, caracterización y análisis comparativo con el fin de seleccionar uno de los tres modelos que mejor desempeño tenga frente a los otros dos modelos, en cuanto a tiempo de respuesta y acierto en la clasificación de los rostros.

Una vez seleccionado el modelo se realizara la implementación de un prototipo de prueba basado en éste modelo, con el fin de recrear todos los procesos necesarios para el reconocimiento de rostros, se tomará datos de los resultados de cada una de las etapas del prototipo, se analizarán los registros de las mediciones obtenidas y se confrontarán con los resultados obtenidos en fase de exploración teórica.

El prototipo debe generar la caracterización de las imágenes de los rostros que se encuentran en la base de datos, para el posteriormente entrenar con esta

información las redes neuronales artificiales. Una vez entrenada la red neuronal el prototipo debe permitir clasificar los rostros y hacer las mediciones de tiempo de respuesta y de acierto en la clasificación de los rostros.

1.6. ORGANIZACIÓN DEL DOCUMENTO

En el primer capítulo se explica el contexto y el planteamiento del problema sobre el cual se desarrolla el trabajo de tesis, los objetivos planteados para este proyecto, y el alcance del mismo.

En el segundo capítulo se presenta el marco teórico necesario para estructurar los conocimientos básicos que se requieren para comprender los conceptos de biometría, del procesamiento digital de imágenes, del análisis de principales componentes, del reconocimiento de patrones y de las RNA.

En el tercer capítulo se relaciona toda la información obtenida en el proceso de selección de los tres modelos que se analizarán con el fin de compararlos y determinar cual de ellos es le que mejor desempeño tenga en cuanto a tiempo de respuesta y acierto en la clasificación de los rostros.

En el cuarto capítulo se describen los requerimientos fundamentales del sistema, basándose en las especificaciones técnicas de la arquitectura del modelo seleccionado.

En el quinto capítulo se presenta el diseño del sistema del prototipo con los módulos que lo componen, la estructura del reconocimiento de rostros y la arquitectura correspondiente.

En el sexto capítulo se muestra la implementación del prototipo funcional y las herramientas de software y hardware utilizadas para el desarrollo del prototipo.

En el séptimo capítulo se presentan las pruebas y los resultados obtenidos del prototipo de reconocimiento de rostros desarrollado en estas tesis.

En el capítulo octavo se relacionan los metas propuestas y los alcances obtenidos durante el desarrollo del proyecto.

En el noveno capítulo se detallan las conclusiones obtenidas a lo largo del proyecto y se incluyen reflexiones y lineamientos para trabajos futuros.

CAPÍTULO 2: MARCO TEORICO

2. Marco Teórico

A continuación se presentan el marco teórico que relaciona los temas involucrados en el proyecto, para mayor profundización de los temas se puede acudir al capítulo 1 de los anexos.

2.1. Biometría

La Información relacionada con el área de la biometría es basada principalmente en los documentos [1], [2], [3], [4] y [131] para profundizar más sobre el tema ver [5], [6], [7].

2.1.1. Introducción

La Identificación y Autenticación biométrica (I&A) explota el hecho de que ciertas características biológicas son singulares e inalterables y son además, imposibles de perder, transferir u olvidar. Esto las hace más confiables, amigables y seguras que los *passwords*.

En el pasado el procesamiento de I&A biométrico era hecho manualmente por gente que física y mentalmente comparaba huellas dactilares contra tarjetas, rostros contra fotos de pasaportes y voces contra cintas grabadas. Hoy en día, dispositivos tales como escáner, videocámaras, y micrófonos pueden, electrónicamente, capturar y entregar estas mismas características biométricas para automatizar procesos y comparaciones. Cada tecnología biométrica (huella dactilar, rostro, voz, etc.) tiene sus propias características, variedades y certezas.

Con la evolución de las tecnologías asociadas a la información, las tareas que tradicionalmente las hacían los seres humanos son, gracias a las mejoras tecnológicas, realizadas mediante sistemas automatizados. Dentro de la amplia gama de posibles actividades que pueden automatizarse, la relacionada con la capacidad para establecer la identidad de los individuos, es de las que ha tomado más importancia y, como consecuencia directa, la *biometría* se ha transformado en un área emergente [1].

2.1.2. Sistemas Biométricos

Una característica anatómica tiene la cualidad de ser relativamente estable al paso del tiempo, como por ejemplo una huella dactilar, la silueta de la mano o patrones del rostro, de la retina o el iris ocular. En cambio, una característica de comportamiento es menos estable pues depende de la disposición psicológica de la persona. La firma podría ser una característica de este tipo [2].

No cualquier característica anatómica puede ser utilizada con éxito por un sistema biométrico [3], [131]. En general, para que una característica pueda ser considerada como válida debe considerar los siguientes aspectos:

- *Universalidad*: Que cualquier persona del mundo posea esta característica.
- *Unicidad*: Que la probabilidad que dos personas tengan esta característica idéntica sea muy pequeña.
- *Permanencia*: Que la característica perdure en el tiempo.
- *Cuantificación*: Que la característica pueda ser medida de forma cuantitativa.

Estos requerimientos sirven como criterio para descartar o aprobar una característica como *indicador biométrico*. Una vez concretada la característica debe existir un *sistema biométrico* capaz de reconocer esta característica con los siguientes requisitos:

- Que el sistema posea una exactitud y rapidez correctas y a su vez los recursos usados por éste sean aceptables.
- Que el sistema sea aceptado y tenga la confianza de los usuarios.
- Que el sistema sea fiable y seguro contra fraudes.

2.1.3. Tipos de sistemas biométricos existentes

En la actualidad existen sistemas biométricos que basan su acción en el reconocimiento de diferentes características como pueden ser el rostro, las huellas dactilares, la geometría o las venas de la mano, los patrones de iris o de retina, la voz, la firma, etc. Vamos a comentar rápidamente las características de los más comunes para poder inferir posteriormente el impacto de éstas contra la privacidad de la persona.

- Verificación de Rostros
- Verificación de escritura
- Verificación de huellas dactilares
- Verificación de patrones en la mano
- Verificación de la voz
- Verificación de patrones oculares

El sistema biométrico basado en la verificación de rostros es el de mayor interés para el proyecto, por tal razón se profundizará teóricamente en este tipo de sistema biométrico.

2.1.3.1. Verificación de Rostros

Los sistemas de reconocimiento de rostro son tal vez los más fáciles de comprender ya que para nosotros la cara es la manera más directa de identificar a

los familiares, amigos, conocidos o celebridades. Como la manera en que funcionan es similar a tomarse una foto, estos sistemas gozan de mucha aceptación entre sus usuarios.

Los métodos utilizados en el reconocimiento de rostros van desde la correlación estadística de la geometría y forma de la cara, hasta el uso de tecnología de redes neuronales que buscan imitar la manera en que funciona el cerebro humano.

Muchos de estos sistemas pueden reconocer a una persona aun cuando esta se haya dejado crecer la barba o el bigote, se pinte o se cambie el estilo del cabello, tenga maquillaje o use anteojos.

2.1.4. Funcionamiento Básico de los Sistemas Biométricos

La mayoría de los sistemas biométricos funcionan de maneras muy similares y se puede resumir en dos pasos:

El primer paso consiste en que la persona debe *registrarse* (“*enroll*”) en el sistema. Durante el proceso de registro, el sistema captura el rasgo característico de la persona, como por ejemplo la huella digital, y lo procesa para crear una representación electrónica llamada *modelo de referencia* (“*reference template*”). El modelo de referencia debe ser guardado en una base de datos, una tarjeta inteligente (“*smart card*”), o en algún otro lugar del cual será extraído en cualquier ocasión futura para el segundo paso. El segundo paso consiste en comprobar los datos reales con los almacenados en los modelos de referencia con el fin de determinar si el individuo es o no quien dice ser [4].

A pesar de que es poco probable obtener dos tomas iguales aún del mismo individuo, a causa de diferencias ambientales y otras condiciones en el momento de la captura, el sistema aún debe poder funcionar correctamente. La mayoría de los algoritmos de comparación generan un ámbito para cada ensayo de comparación el cual es cotejado dentro de determinados **umbrales** antes de ser aceptados o rechazados. Cada proveedor de tecnología biométrica configura la falsa aceptación/rechazo de forma diferente. La Imagen 2.1.1 muestra esta relación de compromiso.

Si el umbral es demasiado bajo, se vuelve demasiado fácil para una persona no autorizada ser aceptada por el sistema, en cambio si el umbral está demasiado alto, personas autorizadas pueden llegar a ser rechazadas.

Las tasas de errores son medidas de dos maneras, una por la cantidad de personas con permiso que son rechazadas (tasa de falso rechazo) y otro por la cantidad de personas sin permiso que son aceptadas (tasa de aceptación indebida). En este caso, es claro, que la mayor preocupación se centra con el segundo tipo, pero en implementaciones prácticas el primer problema genera mucha molestia.

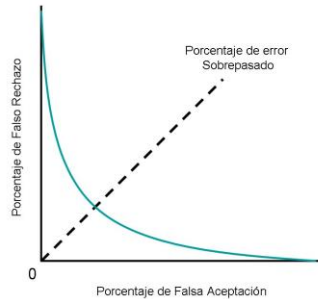


Imagen 2.1.1 Porcentajes de Error en aceptación y rechazo (Tomada de [4])

De acuerdo a la teoría tradicional en biometría, el segundo paso depende de si la función del sistema biométrico consiste en verificar la identidad de la persona o identificar a la persona.

2.1.4.1. Verificación

La persona le informa al sistema cual es su identidad ya sea presentando una tarjeta de identificación o entrando alguna clave especial. El sistema captura el rasgo característico de la persona (la huella digital en nuestro ejemplo) y lo procesa para crear una representación electrónica llamada *modelo en vivo* (“*live template*”). Por último, el sistema compara el modelo en vivo con el modelo de referencia de la persona. Si ambos modelos parecen la verificación es exitosa. De no serlo, la verificación es fallida.

2.1.4.2. Identificación

La persona no le informa al sistema biométrico cual es su identidad. El sistema tan solo captura el rasgo característico de la persona y lo procesa para crear el modelo en vivo. Luego el sistema procede a comparar el modelo en vivo con un conjunto de modelos de referencia para determinar la identidad de la persona [5], [6], [7].

2.2. Reconocimiento de Patrones

La Información relacionada con el área del Reconocimiento de Patrones es basada principalmente en los documentos [11], [12], [13], [14], [15], [56], [57], [58] para profundizar más sobre el tema ver [120 - 130].

2.2.1. Introducción

El objetivo del procesamiento e interpretación de datos sensoriales es lograr una descripción concisa y representativa del universo observado. La información de interés incluye nombres, características detalladas, relacionamientos, modos de comportamiento, etc., que involucran a los elementos del universo (objetos,

fenómenos, conceptos). Estos elementos se perciben como patrones y los procesos que llevan a su comprensión son llamados *procesos preceptuales*.

El etiquetado (clasificación, asignación de nombres) de esos elementos es lo que se conoce como *reconocimiento de patrones*. Por lo tanto, el reconocimiento de patrones es una herramienta esencial para la interpretación automática de datos sensoriales.

El hemisferio derecho (que está conectado con el lado izquierdo del cuerpo) es responsable, principalmente, de nuestra orientación en el espacio, de las iniciativas artísticas, de los oficios, la imagen del cuerpo y el reconocimiento de rostros. Procesa información de una manera más difusa que el hemisferio izquierdo y sus responsabilidades exigen una integración inmediata de muchas cosas a la vez [8], [9], [10], [11].

El sistema nervioso humano recibe aproximadamente 10^9 bits de datos sensoriales por segundo y la mayoría de esta información es adquirida y procesada por el sistema visual. Análogamente, la mayoría de los datos a ser procesados automáticamente aparecen en forma de imágenes.

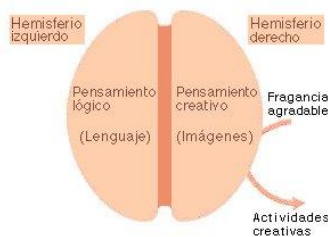


Imagen 2.2.1 Hemisferios del cerebro (Tomada de [11])

El rostro humano es un patrón muy complejo. Los humanos detectamos e identificamos rostros en una imagen con muy poco esfuerzo. El reconocimiento automático de tales patrones es, a la vez, un problema difícil e interesante. En años recientes este problema ha atraído la atención de muchos científicos.

Un sistema para localizar e identificar correctamente rostros humanos sería útil en muchas aplicaciones tales como la identificación de criminales y en sistemas de seguridad y autenticación. Otras aplicaciones donde el reconocimiento de patrones puede ser usado incluyen la asistencia en el reconocimiento del habla y la comunicación visual a través del teléfono y otros medios.

Un sistema completo para el reconocimiento de rostros humanos debería permitir realizar las siguientes tareas:

- Determinar si una imagen contiene un rostro. Si es así, determinar su número, posición y tamaño.
- Determinar la identidad de cada rostro.

- Proporcionar una descripción acerca del rostro (felicidad, enojo, tristeza, etcétera), Entregar información extra, por ejemplo, edad, sexo, etcétera.

Obviamente esta tarea es bastante compleja, siendo aún un problema difícil de resolver. Muchos aspectos acerca del reconocimiento de rostros quedan por ser resueltos.

Varios métodos para el reconocimiento de rostros humanos han sido propuestos, por ejemplo, los métodos basados en redes neuronales, los métodos que usan mosaicos de imágenes, y los métodos que usan *eigen*²-rostros.

2.2.2. Reconocimiento Patrones (RP)

El RP es una disciplina relativamente madura hasta el punto de que existen ya en el mercado un cierto número de sistemas comerciales de reconocimiento de patrones que emplean esta técnica. En RP, un patrón se representa por un vector numérico de dimensión n . De esta forma, un patrón es un punto en un espacio n -dimensional (*de características*). Un RP funciona en dos modos diferentes: entrenamiento y reconocimiento. En *modo de entrenamiento*, se diseña el extractor de características para representar los patrones de entrada y se entrena al clasificador con un conjunto de datos de entrenamiento de forma que el número de patrones mal identificados se minimice. En el *modo de reconocimiento*, el clasificador ya entrenado toma como entrada el vector de características de un patrón desconocido y lo asigna a una de las clases o categorías [12].

El proceso de toma de decisiones en un RP se puede resumir como sigue. Dado un patrón representado por un vector de características

$$X = (x_1, x_2, \dots, x_n)^T \quad (2.2.1)$$

Asignarlo a una de las C clases o categorías, C_1, C_2, \dots, C_c . Dependiendo del tipo de información disponible sobre las densidades condicionales de las clases, se pueden diseñar varias estrategias de clasificación. Si todas las densidades condicionales $P(x|C_i)$, $i=1,2,\dots,c$ se conocen, la regla de decisión es la de Bayes que establece los límites entre las diferentes clases. Sin embargo, en la práctica las densidades condicionales no se conocen y deben ser estimadas (*aprendidas*) partiendo de los patrones de entrada. Si se conoce la forma funcional de estas densidades pero no sus parámetros, el problema se llama de *toma de decisión paramétrico*. En caso contrario, estamos ante un problema de toma de decisión no paramétrico.

Los procesos de percepción del ser humano pueden ser modelados como un sistema de tres estados:

² La palabra *eigen* significa "propio" o "apropiado" en alemán.

- Adquisición de datos sensoriales
- Extracción de características
- Toma de decisiones

Por lo tanto es conveniente dividir el problema del reconocimiento automático de una manera similar.

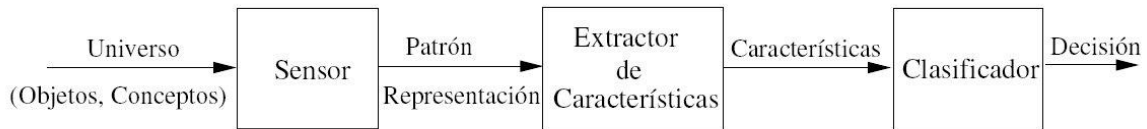


Imagen 2.2.2 Etapas en un sistema de RP (tomada de [13])

2.2.2.1. Metodologías para el reconocimiento de patrones

De manera general, se puede decir que existen tres metodologías básicas para el reconocimiento de patrones [16]: las heurísticas, las matemáticas y las lingüísticas.

2.2.2.1.1 Heurísticas

Esta metodología es la que hace uso de las experiencias y la intuición humana. Por lo general, los sistemas que son desarrollados bajo estos métodos, están hechos a la medida del problema que se desea resolver.

2.2.2.1.2. Matemáticas

Este tipo de metodologías hace uso de las propiedades comunes de los patrones y se basan en las reglas de clasificación formuladas en un marco matemático. Además, este enfoque se divide a su vez en otras dos categorías: las determinísticas y las estadísticas. Entre los métodos determinísticos se puede encontrar, por ejemplo, los algoritmos de aprendizaje iterativo tales como los algoritmos utilizados para el entrenamiento de redes neuronales. Por otra parte, dentro de los métodos estadísticos están los clasificadores basados en la regla de clasificación de Bayes [15].

2.2.2.1.3. Lingüísticas (sintácticas)

Por ultimo se encuentran las metodologías lingüísticas en las cuales se hace uso de los elementos primitivos que componen a los patrones (sub-patrones) y la relación que existe entre ellos.

2.2.2.2. Etapas para el reconocimiento de patrones

Cuando se intenta resolver algún problema que implique el reconocimiento de patrones se deben tener en cuenta las siguientes etapas:

- Adquisición de datos
- Preprocesamiento
- Extracción de características [17]
- Reconocimiento

2.2.3. Reconocimiento de Rostros

De todos los rasgos anatómicos, el *rostro* es el elemento que con más frecuencia utilizamos los seres humanos para identificar a otro individuo. Para ello, el cerebro comienza por establecer los aspectos físicos de una cara, a continuación determina si estas facciones son conocidas o no y, por último, procede a otorgar un nombre a lo que ve.

Este proceso tan aparentemente sencillo para nosotros, puede resultar muy difícil para una máquina. Por eso, antes de desarrollar un sistema biométrico preciso, los científicos se han dedicado a analizar los procesos mentales de reconocimiento facial. De este modo han averiguado, por ejemplo, que existe una región en la base posterior del cerebro que responde preferentemente cuando se ven caras en contraste con la visión de otras partes de la anatomía o de objetos. También hay evidencias de que los procesos de interpretación de los gestos del rostro son independientes del proceso de identificación de caras, por lo que un buen sistema de reconocimiento facial debe ignorar la expresión facial.

Como se mencionó previamente, el reconocimiento de rostros es un área dentro del reconocimiento de patrones que ha sido investigada desde hace varios años. Desde entonces se han propuesto varias técnicas para resolver esta tarea, entre ellas se pueden mencionar las siguientes: las técnicas que utilizan igualamiento de plantillas [18], cálculo de *eigenfaces* [19], características geométricas [18], y los métodos que utilizan redes neuronales [19], entre otros.

Para la implementación de este tipo de sistemas, independientemente de la técnica o metodología que sea implementada, se utilizan generalmente dos conjuntos de datos. El primer conjunto de datos es el que es utilizado durante la etapa de aprendizaje, el cual es comúnmente llamado *conjunto de entrenamiento*. Se debe de tratar que los rostros que integran este conjunto, sean lo más diferentes posibles entre sí, y que además, representen al problema, para poder obtener un buen porcentaje de generalización. El segundo conjunto de rostros, es el que se utiliza durante la etapa de reconocimiento y es llamado el *conjunto de prueba* [20], [21], [22].

Al igual que en el aspecto general del reconocimiento de patrones, en la mayoría de las técnicas de reconocimiento de rostros se involucran las siguientes etapas:

- Preprocesamiento de la imagen.
- Extracción de características.
- Alguna forma de entrenamiento.
- La etapa de reconocimiento del rostro.

2.2.3.1 Preprocesamiento de la Imagen

Dado que las imágenes de los rostros regularmente son tomadas en momentos diferentes, esto trae como consecuencia que las imágenes de una misma tengan variaciones en cuanto a la iluminación, la orientación y el tamaño del rostro. Por tal motivo, es necesario que la imagen sea preprocesada antes de que puede ser utilizada [24], [25], [26]. Entre las tareas de preprocesamiento más comunes podemos encontrar las siguientes:

- Extraer la imagen del rostro de una imagen más grande que contenga información irrelevante para el reconocimiento.
- Normalización en cuanto al tamaño, es decir, que todas las imágenes de los rostros tengan un tamaño igual.
- Y la aplicación de algún método de filtrado para mejorar la calidad de la imagen.

2.2.3.2. Extracción de características

La extracción de características es una de las etapas de las cuales depende en gran medida el buen desempeño del sistema de reconocimiento de rostros. El objetivo principal de esta etapa es extraer la información más discriminatoria de un rostro, eliminando la información que resulte irrelevante para el reconocimiento. En esta etapa se puede encontrar desde las técnicas que obtienen las características de manera manual, hasta técnicas más sofisticadas que extraen las características de forma automática. En varias técnicas, las principales características que son extraídas de los rostros son aquellas que describen a cada uno de los componentes básicos del rostro, tales como los ojos, la nariz, la boca, la barba, las cejas, y por supuesto, la relación existente entre ellos, es decir, la distancia entre cada uno de estos componentes [27], [28], [29], [30].

2.2.3.3. Entrenamiento

El entrenamiento básicamente consiste en utilizar alguna forma de aprendizaje que le permita al sistema “*aprender*” los rostros que constituyen el conjunto de entrenamiento. El tipo de entrenamiento que sea utilizado para el aprendizaje, dependerá en gran medida, de la metodología que se esté utilizando para el reconocimiento. Por ejemplo, al utilizar una metodología como la de redes neuronales, el entrenamiento consiste en obtener los valores correspondientes a

cada una de las conexiones (pesos) que constituyen a la red neuronal. Para este tipo de metodologías existen diferentes algoritmos para realizar el entrenamiento, entre los cuales el algoritmo de retro-propagación es uno de los más populares.

2.2.3.4. Reconocimiento del Rostro

La última etapa que comúnmente es realizada en un sistema de reconocimiento de rostros, es la etapa de reconocimiento. Esta etapa básicamente consiste en alimentar al sistema con imágenes diferentes de las personas utilizadas durante los entrenamientos, esperando obtener como resultado, alguna forma de codificación que nos permita identificar de qué persona se trata, o que el sistema devuelva datos que permita construir el rostro de la persona.

2.2.3.5. Técnicas de Reconocimiento de Rostros

Diferentes técnicas y métodos son propuestos los cuales pueden ser divididos en dos grupos: estáticas (imágenes 2D) y dinámicas (video) [121]. Las técnicas estáticas son las aplicaciones más conocidas, que están basadas en fotografías de rostros, donde aspectos como la iluminación, orientación, rotación, tamaño, etc., son puntos tomados en consideración y deben ser controlados [122]. Las técnicas dinámicas, toman otros aspectos adicionales ya que en una escena de video, aparecen otros objetos que no corresponden a un rostro además las imágenes de una cámara de video tienden a ser de baja calidad [123].

Otra forma de dividir las técnicas de reconocimiento de rostros es: técnicas basadas en componentes constituyentes y basadas en el rostro [124].

La técnica basada en constituyentes se apoya en la interrelación de las diferentes características faciales, tales como nariz, boca, ojos, forma del rostro, etc., [125], [126], la técnica de *Template deformable* es una de las más representativas de este tipo, por ejemplo Yuille et al. [127] lo usó para la detección de los ojos en una imagen de un rostro, lo cual nos ayuda a resolver algunos problemas de invariancia a transformaciones geométricas. Esta técnica proporciona alta flexibilidad en la manipulación de características faciales, tales como ojos, nariz, boca etc. Sin embargo, la garantía de este método depende de la correcta detección de las características faciales, y por tal motivo la resolución de la imagen debería ser relativamente alta, para asegurar una correcta detección de las características [122].

Las técnicas basadas en el rostro, utilizan la imagen bidimensional con variaciones de intensidad, y el reconocimiento es realizado por medio de la búsqueda de regularidades subyacentes estadísticas. El método de análisis de componentes principales (*PCA principal component analysis*) es un ejemplo típico que utiliza esta técnica, la cual tiene ventajas como considerar el rostro como una característica global y la resolución de la imagen puede ser relativamente baja [128], [129].

Otra técnica utilizada para el reconocimiento de rostros es la de usar secuencias de histogramas [130].

2.2.3.6. Tipos de Reconocimiento de Rostros

El reconocimiento de rostros es una de las áreas de investigación de mayor interés en computación, debido a las muchas aplicaciones que se tienen y dependiendo de cada aplicación se pueden identificar varios tipos de sistemas de reconocimiento de rostros, tales como los sistemas para la elaboración de tarjetas para la clasificación de individuos, los sistemas para la reconstrucción rostros a partir imágenes parciales, los sistemas de clasificación de rostros y los sistemas de seguridad para control de acceso[130].

2.3. Procesamiento Digital de Imágenes

2.3.1. Introducción

El termino procesamiento digital de imágenes versa sobre la manipulación y análisis de imágenes por computadora.

El manejo de las imágenes digitales se ha convertido en las últimas décadas en un tema de interés extendido en diferentes áreas de las ciencias naturales, las ciencias médicas y las aplicaciones tecnológicas entre otras. El crecimiento en el poder de computo, las capacidades de almacenamiento y los nuevos sistemas de desplegado, captura e impresión de bajo costo han facilitado el desarrollo de esta disciplina.

2.3.2. Imágenes Digitales

El termino imagen se refiere a una función bidimensional de intensidad de luz $f(x,y)$, donde x e y denotan las coordenadas espaciales y el valor de f en cualquier punto (x,y) es proporcional al brillo (o nivel de gris) de la imagen en ese punto. La Imagen 2.3.1 muestra la convención de coordenadas usada.

Una imagen digital es una imagen $f(x,y)$ que ha sido discretizada en coordenadas espaciales y en brillo. Una imagen digital puede considerarse como una matriz cuyos índices del renglón y columna identifican un punto en la imagen y el correspondiente valor del elemento de la matriz que identifica el nivel de intensidad de luz en ese punto. Los elementos de tal arreglo digital son llamados *elementos de imagen*, *elementos de pintura*, *píxel*³ o *pels*.

Para trabajar con números en la computadora, el nivel de brillo, o valor de cada píxel, es cuantizado a códigos binarios enteros positivos (el brillo no puede ser negativo). El número de niveles de cuantización está determinado por la relación.

³abreviaturas del inglés *picture elements*

$$L = 2^B \quad (2.3.1)$$

donde B representa el número de bits necesarios para cada muestra. Esto es, con 5 bits de cuantización en la intensidad se pueden representar 32 niveles de gris $2^5 = 32$. Para tener una imagen de buena calidad para el ojo humano es necesario tener como mínimo 64 niveles de gris [55]. Para una apreciación fina se usa una cuantización estándar a 8 bits, esto es, 256 niveles de gris. A una imagen cuantizada de esta manera se le conoce como imagen en tonos de gris.

La cuantización a 8 bits (1 *byte*) se dice estándar porque es el mínimo número direccionable directamente por la mayoría de los microprocesadores. Una imagen binaria es una imagen monocromática cuantizada a 1 *bit* por *píxel*, esto es, dos niveles, blanco ó negro.

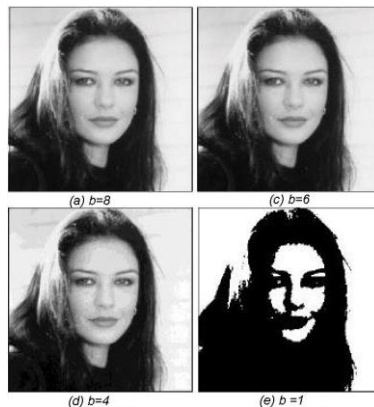


Imagen 2.3.1 Imágenes que muestran la calidad visual al variar el número de bits. *Bits=8, (b)-bits=6, (c) bits=4, (d) bits=1* (Tomada de [34]).

2.3.2.1. Dispositivos de adquisición de Imágenes

Existen diferentes medios para la obtención de una imagen digital, los más comunes son: los *scanners* de cama plana y las cámaras digitales. Ambos se basan en un dispositivo llamado *CCD (Coupled Charge Device)*, éste recibe la luz de la imagen, ya sea por reflexión o por transmisión e integra en un tiempo definido la cantidad de luz que llega a él. Formando un arreglo de *CCD's* es posible realizar la digitalización de la imagen por renglones o bien entera.

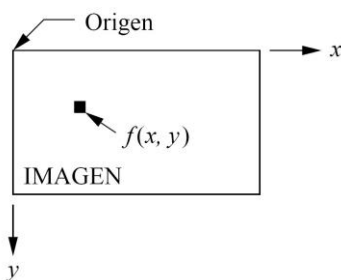


Imagen 2.3.2 Convención de Ejes (Tomada de [35])

2.3.2.2. Digitalización

La digitalización consiste en la descomposición de la imagen en una matriz de $n \times m$ puntos, donde cada punto tiene un valor proporcional a su nivel de gris. Dado que este valor puede ser cualquiera dentro de un rango continuo, es preciso dividir dicho rango en una serie de k intervalos, de forma que el nivel de gris de cada punto sea asignado a uno de los valores que representa dicho intervalo.

Los modernos sistemas de proceso digital de imágenes suelen ser capaces de discriminar 256 niveles de gris.

Cada elemento en que se divide la imagen recibe el nombre de *píxel*, como ya se mencionó antes. El número de niveles de gris y las dimensiones de la matriz (numero de filas por número de columnas) nos condicionan la capacidad de resolución de la imagen digital.

La resolución espacial de la imagen viene dada por el número de píxel que tiene la imagen. Cuanto mayor sea este número mayor va a ser la resolución.

La resolución cromática depende del número de bits que utilicemos para almacenar el valor de un *píxel*. Si utilizamos un *bit* podemos tener únicamente dos valores (0,1) (blanco y negro), si utilizamos 4 bits el número posible de niveles de gris será de 16 (2^4) y si utilizamos 8 bits (1 *byte*) el número de niveles de gris posibles es de 256 (2^8) [56], [57], [58].

Se dice que la imagen ha sido *digitalizada* cuando con cada región de ella se genera un número que representa la cantidad de luz que fue registrada. El siguiente diagrama (Imagen 2.3.3) se muestra un arreglo de 3×3 de digitalización. Para cada celda de registro se genera un número que corresponde a la cantidad

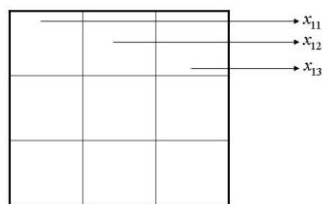


Imagen 2.3.3 Digitalización del primer renglón de una imagen (Tomada del Autor)

de luz que se registro en la zona, en la imagen los valores x_{11} , x_{12} , x_{13} son éstas cantidades. De igual manera se hace con todos los renglones. En general para un arreglo de $n \times m$ celdas se generará una matriz de al forma,

$$I = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix}$$

Cada elemento de la matriz representa una paridad de la imagen. El sistema más simple de digitalización corresponde al llamado *tono de gris*, éste indica la cantidad o intensidad de luz registrada. Por ejemplo para un sistema de transmisión, como es el negatoscopio⁴ utilizado en radiología, se coloca una fuente de luz a continuación del negativo o lámina con la imagen y al final del arreglo de CCD's.

El otro modelo corresponde a los sistemas basados en *reflexión*, este es el caso de los *scanners* de cama plana. Estos dispositivos en vez de tener un arreglo bidimensional de CCD's, sólo tienen un arreglo lineal de éstos. Mediante un sistema óptico se envía luz desde una lámpara a la imagen y es recibida en un arreglo lineal de detectores y un subsistema de desplazamiento basado en un motor. La imagen 2.3.4 muestra el principio de operación del dispositivo.

El principio de las cámaras digitales es también la reflexión de la luz, solo que la fuente es externa al dispositivo (la cámara). En general puede haber varias fuentes de luz, pero solo hay un sistema de registro.

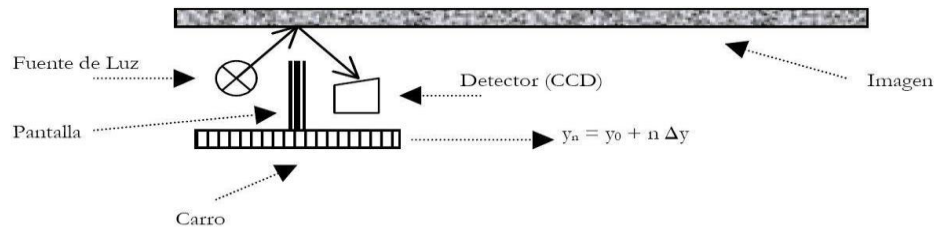


Imagen 2.3.4 Sistema de registro por reflexión (Tomada de [36])

2.3.2.3. Almacenamiento de imágenes digitales

Una vez capturada la imagen necesitamos pasarla a un ordenador donde se almacenará y procesará. Si el dispositivo de captación genera una imagen digital (escáner o cámara digital) la imagen la podremos almacenar y procesar directamente. Si lo que genera es una imagen analógica deberemos de contar con una tarjeta digitalizadora o "*frame grabber*" que nos la convierta a formato digital [59].

⁴ Material constituido por una placa translúcida colocada delante de una fuente luminosa, utilizado para examinar las radiografías

Un “*frame grabber*” está formado por una serie de convertidores analógico/digitales, un área de memoria de imagen, una paleta de color o lut (*look-up table*) y una serie de convertidores digital analógico. La cámara genera una señal de vídeo que es digitalizada por medio de los convertidores analógico/digitales (A/D). Si cámara es en blanco y negro se utiliza un convertidor y si se trata de una cámara en color se utilizan tres, uno para cada banda. La imagen digitalizada es almacenada en el área de memoria. Para visualizarla se transforma la imagen digital en una señal analógica (señal de vídeo), mediante los convertidores digital/ analógico (D/A) y es visualizada en un monitor. La paleta de colores es una tabla donde a cada valor se asigna un color o un nivel de gris que será el que se utilice para representar la imagen en el monitor.

Para almacenar una imagen en disco a memoria se han desarrollado una diversidad de formatos, los campos básicos a considerarse son los descriptores principales de la imagen (que se relaciona con una imagen en tonos de gris).

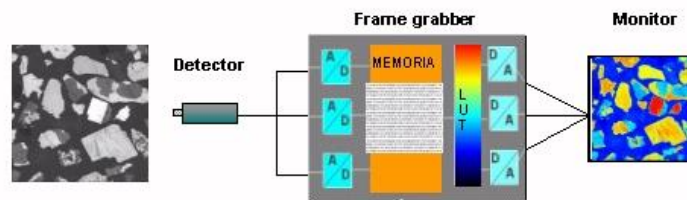


Imagen 2.3.5 Paso de imagen por *Frame Grabber* (Tomada de [36])

Se almacena mínimamente: el ancho, el alto, la profundidad en bits y la matriz de píxeles. A esta representación se le llama un “Mapa de Bits” o simplemente *BitMap*. Por ejemplo para guarda una imagen en tonos de gris adquirida de una fotografía de 2” por 3” con una resolución de 300 *dpi* (*dots per inch*) y una profundidad en bits de 8, se requiere un espacio de almacenamiento de:

$$N_1 = (2 \times 300)(3 \times 300) \text{ bytes} = 5.4 \times 10^5 \text{ bytes}$$

Que corresponde aproximadamente a ½ Mega Byte. Si tomamos ahora una postal de (5” x 8”) y la digitalizamos a 600 *dpi* en color a 24 bits, el espacio de almacenamiento requerido para el mapa de bits será:

$$N_1 = (5 \times 600)(8 \times 600) \text{ bytes} = 1.44 \times 10^7 \text{ bytes}$$

Que ahora corresponde a 13.73 MB. Puede notarse que el aumento de resolución al doble (300 → 600) y el aumento en el tamaño de la imagen en $(\frac{5}{2}, \frac{8}{3}) \approx (2.5, 2.66)$ refleja el carácter no lineal de la transformación, donde $N_2 = 2 \times 2 \times 2.5 \times (8/3) N_1 = 26.66 N_1$, donde el crecimiento es aproximadamente 26.7 veces. Los formatos de imágenes más usuales son: TIFF, PCX, BMP, PICT, GIF, JPG.

2.3.3. Procesos Digitales sobre Imágenes

Frecuentemente es deseable transformar una imagen digitalizada a través de una o más operaciones sobre ésta. Estas operaciones son transformaciones geométricas (como la rotación), realce de orillas o detección de orillas y bordes, segmentación, restauración, reconocimiento de patrones, correlaciones, extracción de partes y suavizamiento selectivo, entre otras. Por otra parte, las operaciones locales de imágenes, permiten transformaciones limitadas en áreas pequeñas de la imagen, las cuales están definidas como un conjunto de vecinos de un punto dado en una imagen digital. Los puntos vecinos incluidos en un conjunto particular, se especifican en un rango de valores de x e y , alrededor del punto (x, y) ; por ello muchas de las operaciones en una imagen incorporan tanto operaciones locales como de sus repeticiones, con objeto de lograr un efecto de transformación global [37], [38], [42], [43], [44], [61].

Una operación conveniente para cierto tipo de imágenes la constituye la representación vectorial o conjunto de segmentos de línea, en sustitución de un arreglo de *píxel*; debido a que los resultados obtenidos favorecen su manipulación como objetos gráficos de forma más simple. Estos formatos también incluyen escalas de grises al asociar códigos de niveles de grises en cada segmento de línea. Los formatos vectoriales también pueden utilizar representaciones polinomiales de órdenes mayores [45], [46], [63], [64].

2.4. Análisis de Principales Componentes (PCA)

2.4.1. Introducción

La Información relacionada con el área del Análisis de Principales Componentes es basada principalmente en los documentos [69] y [72] para profundizar más sobre el tema ver [65 - 71].

El análisis de principales componentes *PCA* (*Principal Component Analysis*), es una técnica estadística, quizás la más vieja, ya que fue propuesta a principios del siglo pasado por Karl Pearson [65] como parte del análisis de factores en el área de la biología y fue desarrollada por Hotelling [66] en el área de la psicometría. Sin embargo la complejidad de los cálculos retrasó su desarrollo hasta la aparición de los computadores y su utilización en la segunda mitad del siglo XX. El relativamente reciente florecimiento de los métodos basados en componentes principales hace que ellos sean poco utilizados por una gran cantidad de investigadores no especialistas en estadística [67], [68], [71].

Podría decirse que el objetivo principal que persigue el PCA es la representación de las medidas numéricas de varias variables en un espacio de pocas dimensiones donde nuestros sentidos puedan percibir relaciones que de otra manera permanecerían ocultas en dimensiones superiores.

2.4.2. Principios del Análisis de Principales Componentes

El objetivo de este método es determinar un conjunto de m vectores ortogonales en el espacio de datos, que contenga la mayor cantidad de información posible, de la varianza de los mismos. Con la proyección de los datos del espacio n -dimensional original ($n \gg m$), sobre el sub-espacio m -dimensional generado por estos vectores, se forman agrupamientos que son más viables a ser clasificados mediante la proyección en una dirección de gran varianza [69], [72].

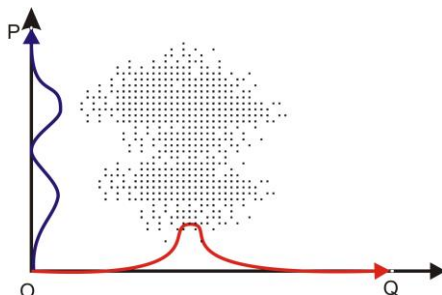


Imagen 2.4.1. Representación de datos en forma de nube (Tomada del Autor)

En la Imagen 2.4.1 **OP** es la dirección del mayor componente principal de la distribución generada por la agrupación de puntos, y **OQ** es la dirección del segundo mayor componente principal. La proyección sobre **OP** muestra más estructura que la proyección sobre **OQ**, por lo que los agrupamientos son más viables a ser identificados por su proyección en direcciones de gran varianza. Por consiguiente la reducción de la dimensión matricial en *PCA* puede mantener la mayor cantidad de información intrínseca de los datos.

La técnica *PCA* transforma un conjunto de variables correlacionadas en un conjunto de variables no correlacionadas, simplifica la transformación encontrando los componentes más cercanos a las variables originales pero ordenados en forma decreciente al orden de su varianza. Esta transformación es una rotación ortogonal en el espacio de las variables originales.

La técnica *PCA* consta de las siguientes partes para su realización:

- Obtención de la matriz con los datos originales.
- Ajuste de los datos originales.
- Cálculo de la matriz de covarianza
- Cálculo de los *eigenvalores*⁵ y los *eigenvectores* de la matriz de covarianza.
- Búsqueda de los componentes principales y elaboración de la matriz de vectores característicos.
- Derivar el nuevo conjunto de datos.

⁵ Los valores propios se llaman también valores característicos o autovalores, y los vectores propios, vectores característicos o autovectores.

Por medio de un ejemplo se seguirán los pasos que comprende el proceso del Análisis de Principales Componentes.

2.4.2.1. Obtención de la matriz con los datos originales

La matriz que contiene los datos originales con los cuales se trabajará será una matriz cuadrada, pero igual el proceso se puede realizar con cualquier tipo de matrices. Los datos que contendrán las matrices trabajadas en el proyecto serán los valores de los píxeles de la imagen que se procesará. Esta matriz contiene una gran cantidad de datos debido al tamaño de las imágenes que se procesan, y una de las características fundamentales del *PCA* es la reducción de la dimensión de estos datos sin perder la información relevante e ellos asociada.

$$A = \begin{pmatrix} X & Y \\ 2,5 & 2,4 \\ 0,5 & 0,7 \\ 2,2 & 2,9 \\ 1,9 & 2,2 \\ 3,1 & 3 \\ 2,3 & 2,7 \\ 2 & 1,6 \\ 1 & 1,1 \\ 1,5 & 1,6 \\ 1,1 & 0,9 \end{pmatrix}$$

Imagen 2.4.2. Matriz con datos Originales (Tomada de [69])

En el ejemplo se presentará una matriz pequeña, teniendo en cuenta que el proceso es el mismo sin importar el tamaño de la matriz.

En la matriz de la imagen 2.4.2 podemos observar una serie de datos con los cuales se ilustrará el proceso de la técnica *PCA*. Cada uno de los vectores verticales de la matriz representará una dimensión, lo que significa que esta matriz tendrá tres dimensiones (x, y), esto se utilizará en el proceso del cálculo de la matriz de covarianzas.

2.4.2.2. Ajuste de los datos originales

Este proceso consiste en tomar los datos originales por vector vertical y hallar su media, para posteriormente restársela a cada uno de los elementos de cada dimensión. La media sustraída es el promedio de cada dimensión, así todos los valores X tienen sustraída la media \bar{X} . Al igual que los valores Y tienen sustraída la media \bar{Y} .

Esta matriz con datos ajustado se utiliza en el final para la obtención de los datos del componente principal.

2.4.2.3. Calculo de la matriz de covarianza

La matriz de covarianza es una matriz con características singulares que benefician el proceso del cálculo de sus valores y sus vectores propios. Estas características son: es una matriz cuadrada, simétrica y ortogonal.

$$A = \begin{pmatrix} 0,69 & 0,49 \\ -1,31 & -1,21 \\ 0,39 & 0,99 \\ 0,09 & 0,29 \\ 1,29 & 1,09 \\ 2,30 & 2,70 \\ 0,19 & -0,31 \\ -0,81 & -0,81 \\ -0,31 & -0,31 \\ -0,71 & -1,01 \end{pmatrix}$$

Imagen 2.4.3. Datos Ajustados (Tomada de [69])

El proceso del cálculo de la matriz de covarianza se realiza teniendo en cuenta las siguientes formulas:

$$Cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \quad (2.4.1)$$

$$C^{n \times n} = \begin{pmatrix} Cov(x, x) & Cov(x, y) & Cov(x, z) \\ Cov(y, x) & Cov(y, y) & Cov(y, z) \\ Cov(z, x) & Cov(z, y) & Cov(z, z) \end{pmatrix} \quad (2.4.2)$$

Como se observa en la formula 2.4.2 la matriz de covarianza tiene en su diagonal principal la covarianza de cada una de las dimensiones tomadas entre si, y la simetría de la matriz es debido a que $Cov(x, y) = Cov(y, x)$ [70].

Siguiendo con el ejemplo se obtiene la siguiente matriz:

$$Cov^{n \times n} = \begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{pmatrix}$$

Así se observa que los elementos que no están en la diagonal principal son positivos, debe ser así, excepto cuando las variables X e Y se incrementa juntas.

2.4.2.4. Cálculo de los *eigenvalores* y los *eigenvectores* de la matriz de covarianza

2.4.2.4.1. *Eigenvectores*

Como se sabe, se pueden multiplicar dos matrices entre si, siempre y cuando sus tamaños sean compatibles. Los *eigenvectores* son un caso especial de esto, como se observa en la imagen 2.4.4^a, considerar las dos multiplicaciones entre una matriz y un vector

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 11 \\ 5 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Imagen 2.4.4^a. Ejemplo de un *Eigenvector* y un no *eigenvector* (Tomada de [69]).

$$2 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 6 \\ 4 \end{pmatrix} = \begin{pmatrix} 24 \\ 16 \end{pmatrix} = 4 \times \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

Imagen 2.4.4b. Ejemplo de cómo un *eigenvector* escalado es también un *eigenvector* (Tomada de [69]).

En el primer ejemplo el vector resultante no es un múltiplo de un entero y el vector original, sin embargo en el segundo ejemplo, el vector resultante es cuatro veces el vector original, esto significa que el vector resultante es un vector espacial en dos dimensiones.

El vector $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ (del segundo ejemplo de multiplicación) representa un vector dirigido

desde el origen (0,0), hasta el punto (3,2). La otra matriz, la cuadrada, puede ser la representación de una matriz transformada. Si se multiplica esta matriz a la izquierda de un vector, el resultado es otro vector trasladado de su posición original. Esta es la naturaleza de la transformación que los *eigenvectores* generan. Supóngase una matriz transformación que, cuando se multiplicó a la izquierda, refleja vectores en la línea $x = y$. Entonces se observa que si había un vector ubicado en la línea $y = x$, es la reflexión de si mismo. Este vector (y todas los múltiplos de éste, porque no interesa el tamaño que el vector posea), podría ser un *eigenvector* de la matriz de transformación.

Entre las propiedades de los *eigenvectores* están, lo primero que se debe tener en cuenta es que los *eigenvectores* solo pueden encontrarse en matrices cuadradas. Y no todas las matrices cuadradas tienen *eigenvectores*. Y dada una matriz de $n \times n$

n que tiene *eigenvectores*, tiene n de ellos. Dada una matriz de 3×3 , hay 3 *eigenvectores*. Otra propiedad de los *eigenvectores* es que aun cuando se escale el vector por alguna cantidad antes de multiplicarlo, se obtiene el mismo múltiplo como resultado, como se observa en la imagen 2.4.4b. Esto es gracias a que si se escala un vector por alguna cantidad, lo único que se está haciendo es alargándolo, mas no cambiando su dirección. Por ultimo, todos los *eigenvectores* de una matriz son perpendiculares, en otras palabras son Ortogonales. Esto es importante porque significa que se pueden expresar los datos en términos de estos *eigenvectores* perpendiculares, en vez de expresarlos en términos de los ejes x , y .

Otra cosa importante que se debe saber, es que cuando se buscan los *eigenvectores* deben hacerse tratando de que su longitud sea exactamente uno. Esto es porque, como se debe saber, la longitud de un vector no afecta si es o no un *eigenvector*, por cuanto lo hace su dirección. Así, para obtener un *eigenvector* estándar, siempre que encontremos un *eigenvector* usualmente se escala para hacer que éste tenga la longitud de 1. Se aprecia mejor con un ejemplo.

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

Este es un *eigenvector* y su longitud es:

$$\sqrt{(3^2 + 2^2)} = \sqrt{13}$$

Se divide el vector original por esta cantidad para hacer que su longitud sea 1

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix} \div \sqrt{13} = \begin{pmatrix} 3/\sqrt{13} \\ 2/\sqrt{13} \end{pmatrix}$$

Desafortunadamente esto no es tan fácil en matrices grandes. Después de todo el camino más usual para el cálculo de *eigenvectores* es por medio de métodos iterativos complicados.

2.4.2.4.2. Eigenvalores:

Están ligados íntimamente con los *eigenvectores*, de hecho, vimos *eigenvalores* en la imagen 4.4^a. En este ejemplo 4 es el *eigenvalor* asociado al *eigenvector*. Se debe observar que los *eigenvectores* y los *eigenvalores* siempre vienen en pares. Para el ejemplo que se trae de la técnica PCA, se encontraran sus *eigenvectores* y sus *eigenvalores* de la matriz de covarianza:

$$\text{Eigenvalues: } \begin{pmatrix} 0.490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{Eigenvectores:} \begin{pmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{pmatrix}$$

Es importante notar que estos *eigenvectores* son ambos *eigenvectores* unitarios, sus longitudes para ambos es 1. Esto es muy importante para *PCA*.

Se puede observar que los datos tienen una gran relación, como se esperaba de la matriz de covarianza, las dos variables se incrementan juntas.

Así pues, mediante este proceso de obtener los *eigenvectores* de la matriz de covarianza, se han podido extraer líneas que caracterizan los datos. El resto de los pasos involucra la transformación de los datos para expresarlos en términos de estas líneas.

2.4.2.5. Búsqueda de los componentes principales y elaboración de la matriz de vectores característicos

En este punto lo que resta es ubicar el mayor *eigenvalor*, que está ligado directamente con el vector característico principal, y así sucesivamente en orden decreciente de los *eigenvalores*.

Para el caso del ejemplo seguido, el mayor *eigenvalor* es 1.28402771 que está relacionado con el *eigenvector*

$$\begin{pmatrix} -0.677873399 \\ -0.735178656 \end{pmatrix}.$$

Así, teniendo ya ubicados los *eigenvectores* en su orden de prioridad, se procede a elaborar la matriz con estos *eigenvectores*:

$$\text{VectorCaracteristico} = (\text{eigen}_1, \text{eigen}_2, \text{eigen}_3, \dots, \text{eigen}_n)$$

2.4.2.6. Derivar el nuevo conjunto de datos

Este es el paso final en *PCA*, y es también el más fácil. Una vez escogidos los componentes característicos (los *eigenvectores*), que se consideren representativos de los datos, y formado el vector característico, simplemente tomamos la transpuesta del vector y lo multiplicamos a la izquierda del conjunto de datos original, transpuesto.

$$\text{DatosFinales} = \text{FilaVectorCaracteristico} \times \text{FilaDatosAjustados}$$

Para el ejemplo los datos finales serían:

$$\begin{array}{ll} -0.827970186 & -0.175115307 \\ 1.77758033 & 0.142857227 \\ -0.992197494 & 0.384374989 \\ -0.274210416 & 0.130417207 \end{array}$$

-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

2.5. Redes Neuronales Artificiales RNA

2.5.1. Introducción

La Información relacionada con el área de Redes Neuronales Artificiales esta basada principalmente en los documentos [73], [74] y [75] para profundizar más sobre el tema ver [76], [].

La preocupación del ser humano por entender su propia naturaleza lo ha llevado a indagar sobre su ser, sobre las características que nos diferencian de los demás animales y que nos hacen una especie única entre los seres vivos, estamos hablando de la inteligencia, característica que discrimina absolutamente nuestra especie, y es por esto que alrededor de la inteligencia que giran gran cantidad de estudios científicos y filosóficos [73], [74].

La construcción de sistemas inteligentes es uno de los retos a los que se enfrenta el ser humano hoy en día, y es ahí donde aparecen las RNA utilizadas para diseñar máquinas con elementos neuronales de procesamiento paralelo, con el fin de emular lo más fiel posible los sistemas neuronales animales [117].

2.5.2. Fundamentación Biológica

El tejido nervioso es el más diferenciado del organismo y está constituido por células nerviosas, fibras nerviosas y la neuroglia⁶, que está formada por varias clases de células. La célula nerviosa se denomina neurona, que es la unidad funcional del sistema nervioso. Hay neuronas bipolares, con dos prolongaciones de fibras y multipolares, con numerosas prolongaciones. Pueden ser neuronas sensoriales, motoras y de asociación.

El sistema de neuronas biológico esta compuesto por neuronas de entrada (censores) conectados a una compleja red de neuronas (neuronas ocultas), las cuales, a su vez, están conectadas a las neuronas de salidas que controlan, por ejemplo, los músculos. Los sensores pueden ser señales de los oídos, ojos, etc. las respuestas de las neuronas de salida activan los músculos correspondientes.

⁶ Células no excitables que sostienen las neuronas del sistema nervioso centra (neuro = nervio; glia = pegamento)

El tamaño y la forma de las neuronas es variable, pero con las mismas subdivisiones que muestra la Imagen 2.5.1. El cuerpo de la neurona o Soma contiene el núcleo. Se encarga de todas las actividades metabólicas de la neurona y recibe la información de otras neuronas vecinas a través de las conexiones sinápticas [75], [76].

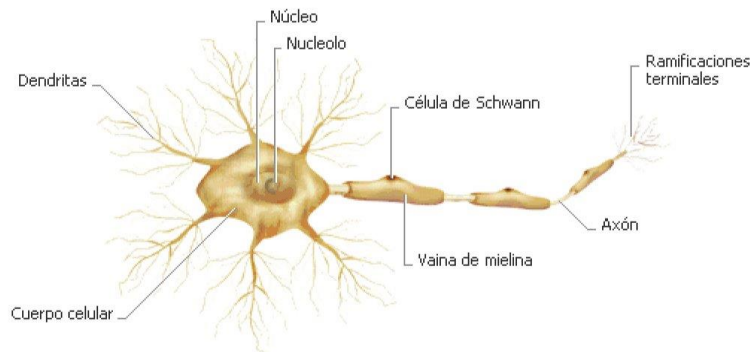


Imagen 2.5.1 Estructura Neurona biológica (Tomada de [75])

2.5.3. Red de Neuronas Artificiales

Las RNA son modelos que intentan reproducir el comportamiento del cerebro humano, imitan la estructura hardware del sistema nervioso, con intención de construir sistemas de procesamiento de la información paralelos, distribuidos y adaptativos, que puedan presentar un cierto comportamiento “Inteligente”. Tal modelo, realiza una simplificación, averiguando cuáles son los elementos relevantes del sistema, bien porque la cantidad de información de que se dispone es excesiva o bien porque es redundante. Una elección adecuada de sus características, más una estructura conveniente, es el procedimiento convencional utilizado para construir redes capaces de realizar determinada tarea.

Pese al extraordinario desarrollo de la electrónica y las ciencias de la computación, ni el ordenador más potente puede llevar a cabo tareas tales como reconocer una mosca y atraparla al vuelo, que un sistema tan simple como el cerebro de la rana es capaz de llevar a cabo con eficiencia.

Existen modelos muy diversos de redes de neuronas en los cuales se siguen filosofías de diseño, reglas de aprendizaje y funciones de construcción de las respuestas muy distintas. Una primera clasificación se hace en función del recorrido que sigue la información dentro de la red, y así se distinguen redes alimentadas hacia delante y redes con retroalimentación.

2.5.3.1. Neurona Artificial

La neurona artificial, célula, autómatas o procesador elemental, es un dispositivo simple de cálculo que posee un estado interno, llamado nivel de activación, y

recibe un valor de entrada procedente de exterior o de otras neuronas que le permiten un cambio de estado, en busca de proporcionar una única respuesta o salida.

Los elementos que constituyen la neurona son los siguientes:

2.5.3.1.1. Entradas y salidas

Conjunto de entradas $x_j(t)$. Las variables de entrada pueden ser binarias (digitales) o continuas (analógicas), dependiendo del modelo y aplicación. Por ejemplo, un perceptrón multicapa o *MLP (Multilayer Perceptron)*, admite ambos tipos de señales [77], [79].

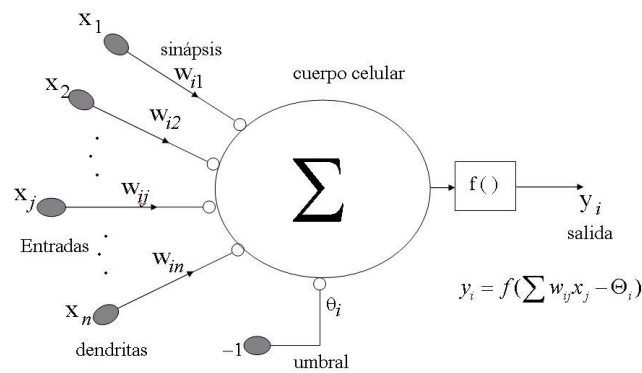


Imagen 2.5.2 Modelo de Neurona Artificial estándar (Tomada de [77])

2.5.3.1.2. La Regla de Propagación

La Regla de Propagación permite obtener, a partir de las entradas y los pesos, el valor del potencial postsináptico h_i de la neurona

$$h_i(t) = \sigma(w_{ij}, x_j(t))$$

La función más habitual es de tipo lineal, y se basa en la suma ponderada de las entradas con los pesos sinápticos

$$h_i(t) = \sum_j w_{ij} x_j$$

que formalmente también puede interpretarse como el producto escalar de los vectores de entrada y pesos

$$h_i(t) = \sum_j w_{ij} x_j = w_i^T \cdot x$$

2.5.3.1.3. El peso sináptico de la neurona w_{ij}

Define la intensidad de interacción entre la neurona presináptica j y la postsináptica i . Dada una entrada positiva, si el peso es positivo tenderá a excitar a la neurona postsináptica, si el peso es negativo tenderá a inhibirla.

Una regla de tipo no lineal, de uso más limitado, es el siguiente:

$$h_i(t) = \sum_j w_{ij1j2\dots jp} x_{j1} x_{j2} \dots x_{jp}$$

que implica una interacción de tipo multiplicativo entre las entradas de la neurona (como se observa realmente en determinadas sinapsis biológicas). El uso de esta última regla de propagación determina que una neurona se denomine de *orden superior* o neurona *sigma-pi* [78] (por emplear sumas y productos), e implica una mayor complejidad, tanto en el estudio de la dinámica de la red neuronal como en su realización hardware.

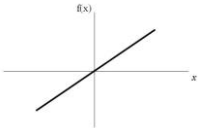
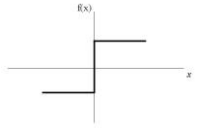
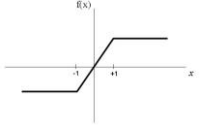
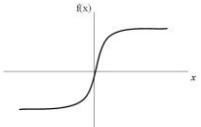
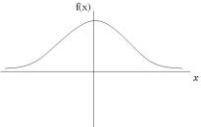
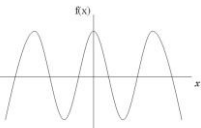
	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{signo}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal o Tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[1-, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = A.e^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A.\text{sen}(wx + \varphi)$	$[-1, +1]$	

Tabla 2.1. Funciones de Activación habituales (Se han omitido algunas constantes)

Otra regla de propagación habitual, especialmente en los modelos de Redes Neuronales basadas en el cálculo de distancias entre vectores (como RBF, mapas de Kohonen o LVQ), es la *distancia euclídea*

$$h^2_i(t) = \sum_j (x_j - w_{ij})^2$$

que representa la distancia (al cuadrado) existente entre el vector de entradas y el de pesos. Cuando ambos vectores son muy similares, la distancia es muy pequeña; cuando son muy diferentes, la distancia crece. Por tanto, este tipo de regla opera de manera diferente a las anteriores ya citadas.

En la imagen 2.5.2 se muestra un modelo que representa un grupo de entradas x_1, x_2, \dots, x_n son introducidas en una neurona artificial. Estas estradas definidas por un vector X , corresponden a las señales de la sinapsis de una neurona biológica. Cada señal se multiplica por un peso asociado w_1, w_2, \dots, w_n antes de ser aplicado la sumatoria etiquetada por Σ . Cada peso corresponde a la fuerza de una conexión sináptica, es decir el nivel de concertación iónica de la sinapsis, y se representa por W .

La sumatoria, que corresponde al cuerpo de la neurona, suma todas las entradas ponderadas algebraicamente, produciendo una salida que se denomina E, así:

$$E = x_1w_1 + x_2w_2 + \dots + x_nw_n$$

En forma vectorial queda representado como:

$$E = X^T W$$

Las señales E son procesadas además por una función llamada función de activación o de salida F , que produce la señal de salida de la neurona S . Dependiendo de la función F , habrá distintos modelos de autómatas, por ejemplo:

- Lineal: $S = KE$ con K constante.
- Umbral: $S = 1$ si $E \geq \Theta$, $S = 0$ si $E < \Theta$; siendo Θ el umbral constante.
- Cualquier función: $S = F(I)$; siendo F una función cualquiera.

Existen tres tipos de neuronas en cualquier sistema: entradas, salidas y ocultas. Las células de entrada reciben señales desde el entorno; las de salida envían la señal fuera de la red, y las unidades ocultas son aquellas cuyas entradas y salidas se encuentran dentro del sistema. Se conoce como capa o nivel a un conjunto de neuronas cuyas entradas provienen de la misma fuente y cuyas salidas se dirigen al mismo destino.

2.5.3.2. Características de las RNA

Existen cuatro aspectos que caracterizan una red neuronal: su topología, el mecanismo de aprendizaje, tipo de asociación realizada entre la información de entrada y salida, y la forma de representación de estas informaciones.

2.5.3.2.1 Topología de las Redes Neuronales

La arquitectura de las redes neuronales consiste en la organización y disposición de las neuronas formando capas más o menos alejadas de la entrada y salida de la red. En este sentido, los parámetros fundamentales de la red son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas.

Redes Monocapa: Se establecen conexiones laterales, cruzadas o autorrecurrentes entre las neuronas que pertenecen a la única capa que constituye la red. Se utilizan en tareas relacionadas con lo que se conoce como autoasociación; por ejemplo, para generar informaciones de entrada que se presentan a la red, incompletas o distorsionadas.

Redes Multicapa: Son aquellas que disponen de conjuntos de neuronas agrupadas en varios niveles o capas. Una forma de distinguir la capa a la que pertenece la neurona, consiste en fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. Según el tipo de conexión se distinguen las redes feedforward (hacia delante), y las redes feedback/feedforward (hacia delante y hacia atrás).

2.5.3.2.2 Mecanismo de Aprendizaje

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre las neuronas, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero, una conexión se destruye cuando su peso pasa a ser cero. Se puede afirmar que el proceso de aprendizaje ha finalizado (la red ha aprendido) cuando los valores de los pesos permanecen estables ($dw_{ij}/dt = 0$).

Un criterio para diferenciar las reglas de aprendizaje se basa en considerar si la red puede aprender durante su funcionamiento habitual, o si el aprendizaje supone la desconexión de la red.

Otro criterio suele considerar dos tipos de reglas de aprendizaje: las de aprendizaje supervisado y las correspondientes a un aprendizaje no supervisado, estas reglas dan pie a una de las clasificaciones que se realizan de las RNA: Redes neuronales con aprendizaje supervisado y redes neuronales con aprendizaje no supervisado. La diferencia fundamental entre ambos tipos estriba

en la existencia o no de un agente externo (supervisor) que controle el aprendizaje de la red.

2.5.3.3. Estructura Básica de la Red

En la imagen 2.5.3 se muestra un ejemplo de una neurona estándar que compone una red de neuronas artificiales. A la izquierda se ve una serie de entradas a la neurona; cada una llega de la salida de otra neurona de la red o del exterior. Una vez calculada la salida de una neurona, ésta se propaga, vía conexiones de salida, a las células de destino. Todas las conexiones de salida reciben el mismo valor de salida.

A la manera que las células se conectan entre sí se le denomina patrón de conectividad o arquitectura de la red. La estructura básica de interconexión entre células es la de la red multicapa, mostrada en la imagen 2.5.3. Se trata de una estructura típica de implementación del paradigma conocido como Retro-Propagación. El primer nivel lo constituyen las células de entrada; estas unidades reciben los valores de unos patrones representados como vectores que sirven de entrada a la red. A continuación hay una serie de capas intermedias, llamadas capas ocultas, cuyas unidades responden a rasgos particulares que pueden aparecer en los patrones de entrada. Puede haber uno o varios niveles ocultos. El último nivel es el de salida. La salida de estas unidades sirve como salida de toda la red.

Cada interconexión entre unidades de proceso actúa como una ruta de comunicación: a través de estas interconexiones viajan los valores numéricos de una célula a otra. Estos valores son evaluados por los pesos de las conexiones. Los pesos de las conexiones se ajustan durante la fase de aprendizaje para producir una Red de Neuronas Artificial final.

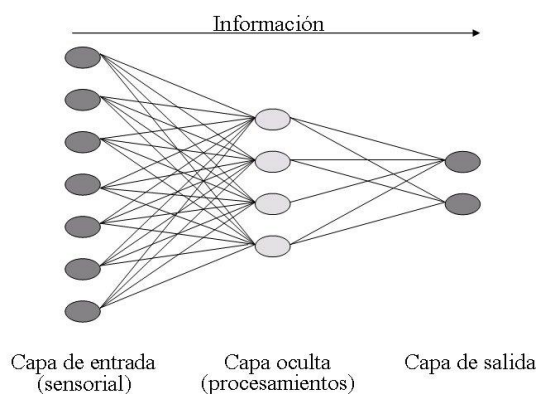


Imagen 2.5.3. Esquema de una Red de tres capas totalmente interconectadas (Tomada de [74])

Así pues, una red de neuronas artificial podría definirse como un grafo cuyos nodos están constituidos por unidades de proceso idénticas, y que propagan

información a través de los arcos. En este grafo se distinguen tres tipos de nodos: los de entrada, los de salida y los intermedios.

El funcionamiento de la red es simple. Para cada vector de entrada, éste es introducido en la red copiando cada valor de dicho vector en la célula de entrada correspondiente. Cada célula de la red, una vez recibida la totalidad de sus entradas, la procesa y genera una salida que es propagada a través de las conexiones entre células, llegando como entrada a la célula destino. Una vez que la entrada ha sido completamente propagada por toda la red, se producirá un vector de salida, cuyos componentes son cada uno de los valores de salida de las células de salida.

El esquema de funcionamiento de una Red de Neuronas por capas como la de la imagen 2.5.5 puede describirse mediante la ecuación:

$$\bar{S} = F(F(\bar{X}.W_1)W_2)$$

donde W_1 y W_2 son los pesos de la primera y la segunda capa, respectivamente; F es la función de activación idéntica en todas las neuronas; \bar{X} es el vector de entrada de la red, y \bar{S} es el vector de salida que la red produce. W_1 y W_2 son matrices de conexiones entre las capas de la red, y por lo tanto se trata de multiplicaciones de matrices. La función de activación F desempeña un papel importante en un esquema de Red de Neuronas. Supóngase que se utiliza una función lineal del tipo $F(x) = k.x$. En este caso, si se sustituye dicha función en la ecuación anterior quedará de la siguiente manera:

$$\bar{S} = k.((K.\bar{X}.W_1).W_2)$$

Lo cual es equivalente a una red con una sola capa de conexiones W_t , donde:

$$W_t = k^2.W_1.W_2$$

Se va a ilustrar con un ejemplo. Sea la red de la imagen 2.5.6, donde:

$$W_1 = \begin{pmatrix} 3 & 2 & 3 \\ 4 & 1 & 2 \\ 5 & 2 & 1 \\ 6 & 1 & 1 \end{pmatrix} \quad W_2 = \begin{pmatrix} 2 & 3 \\ 1 & 2 \\ 1 & 1 \end{pmatrix}$$

Si la función de activación es $F(x) = 2x$ y se introduce el vector $\bar{x} = \{-3, 1, 4, 2\}$, según la ecuación $\bar{S} = F(F(\bar{X}.W_1)W_2)$ la salida será:

$$\bar{A} = X.W_1 = (-3 \ 1 \ 4 \ -2) \cdot \begin{pmatrix} 3 & 2 & 3 \\ 4 & 1 & 2 \\ 5 & 2 & 1 \\ 6 & 1 & 1 \end{pmatrix} = (3 \ 1 \ -5)$$

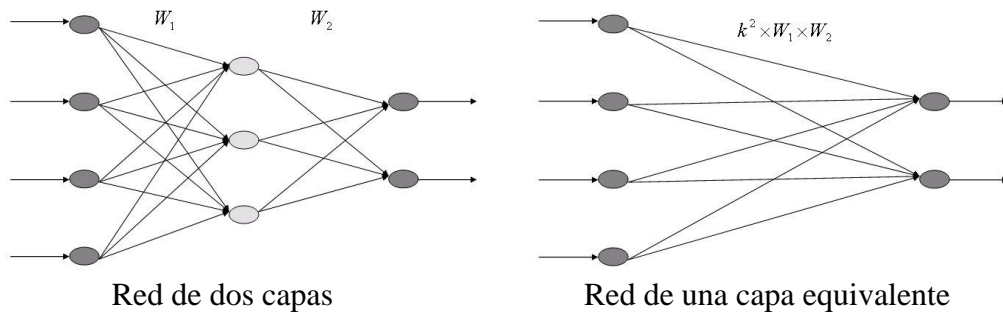


Imagen 2.5.4. Equivalencia entre redes al utilizar funciones de activación lineales (Tomada de [74])

$$\bar{S} = F(F(\bar{A}).W_2) = (6 \ 2 \ -10) \cdot \begin{pmatrix} 2 & 3 \\ 1 & 2 \\ 1 & 1 \end{pmatrix} = F((4 \ 12)) = (8 \ 24)$$

La misma salida se obtendría con una sola capa con una matriz de conexiones calculada de la siguiente manera:

$$W_t = k^2.W_1.W_2 = 4.W_1 = \begin{pmatrix} 3 & 2 & 3 \\ 4 & 1 & 2 \\ 5 & 2 & 1 \\ 6 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 3 \\ 1 & 2 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 44 & 64 \\ 44 & 64 \\ 52 & 80 \\ 56 & 84 \end{pmatrix}$$

Con una sola capa la salida, para el vector de entrada anterior, se calcula simplemente como:

$$\bar{S} = X.W_t = (-3 \ 1 \ 4 \ -2) \cdot \begin{pmatrix} 44 & 64 \\ 44 & 64 \\ 52 & 80 \\ 56 & 84 \end{pmatrix} = (8 \ 24)$$

Se puede observar que se obtiene la misma salida que anteriormente. Esto quiere decir que si la función de activación es lineal, el introducir más capas en la Red es irrelevante; existirá siempre una red con una sola capa equivalente a cualquier

otra con un número de capas arbitrario, y su cálculo es trivial, por lo tanto el incluir funciones de activación lineales hará que la red pierda gran parte de sus capacidades. Todas las redes cuentan con funciones de activación no lineales, que hacen que el potencial de la red sea elevado para dar solución a problemas en forma genérica.

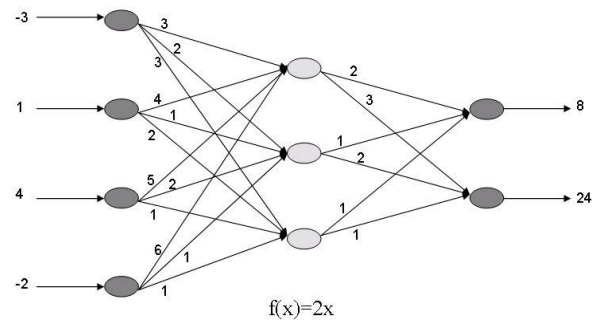


Imagen 2.5.5. Red de dos capas (Tomada de [74])

2.5.3.4. Ventajas de las RNA

Debido a su constitución y a sus fundamentos, las RNA presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas.

2.5.3.4.1. Aprendizaje Adaptativo

Es una de las características más atractivas de las redes neuronales, es la capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial.

En el proceso de aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtengan unos resultados específicos. Una RNA no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de los pesos de los enlaces mediante el aprendizaje. También existen redes que continúan aprendiendo a lo largo de su vida, después de completado el periodo inicial de entrenamiento.

La función del diseñador es únicamente la obtención de la arquitectura apropiada. No es problema del diseñador el cómo la red aprenderá a discriminar, sin embargo, si es necesario que desarrolle un buen algoritmo de aprendizaje que proporcione la capacidad de discriminar de la red mediante un entrenamiento con patrones.

2.5.3.4.2. Autoorganización

Las redes neuronales usan su capacidad de aprendizaje adaptativo para organizar la información que reciben durante el aprendizaje y/o la operación. Una Red Neuronal Artificial puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje. Esta autoorganización provoca la facultad de las redes neuronales de responder apropiadamente cuando se les presentan datos o situaciones a los que no habían sido expuestas anteriormente.

2.5.3.4.3. Tolerancia a Fallos

Comparados con los sistemas computacionales tradicionales, los cuales pierden su funcionalidad en cuanto sufren un pequeño error de memoria, en las redes neuronales, si se produce un fallo en un pequeño número de neuronas, aunque el comportamiento del sistema se ve influenciado, sin embargo no sufre una caída repentina.

Hay dos aspectos distintos respecto a la tolerancia a fallos: primero, las redes pueden aprender a reconocer patrones incompletos, con ruido, o distorsionados. Segundo pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red.

La razón por la que las redes neuronales son tolerantes a fallos es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en ese tipo de almacenamiento, a diferencia de la mayoría de los ordenadores algorítmicos y sistemas de recuperación de datos que almacenan cada pieza de información en un estado único, localizado y direccionable.

2.5.3.4.4. Operación en Tiempo Real

Los computadores neuronales pueden ser realizados en paralelo, y se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.

2.5.3.4.5. Fácil inserción dentro de la tecnología existente

Debido a que una red puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación hardware de bajo costo, es fácil insertar RNA para aplicaciones específicas dentro de sistemas existentes (chips, por ejemplo). De esta manera, las redes neuronales se pueden utilizar para mejorar sistemas de forma incremental, y cada paso puede ser evaluado antes de acometer un desarrollo más amplio.

2.5.3.5 Implementación de RNA

En la búsqueda de sistemas inteligentes en general, se ha llegado a un importante desarrollo del software, dentro de esta línea se encuentran algunos de los neuro-

computadores más conocidos. Un neuro-computador es básicamente un conjunto de procesadores conectados con cierta regularidad que operan concurrentemente. En la actualidad ya existen una serie de neuro-computadores comerciales destinados a la realización de redes neuronales. Por otro lado la realización de RNA puede llevarse a cabo por medio de uno o varios circuitos integrados específicos, para así poder obtener una estructura que se comporte lo más similar posible a como lo haría una red neuronal. Otra tecnología que podría ser apropiada en la implementación de las redes neuronales es la tecnología electro óptica, con la ventaja de utilizar la luz como medio de transporte de la información, permitiendo la transmisión, masiva de datos.

2.6. Perceptrón *Multilayer*

2.6.1. Introducción

El Perceptrón *Multilayer* (multinivel, multicapa), es una generalización del Perceptrón Simple, y surgió como consecuencia de las limitaciones de dicha arquitectura en lo referente al problema de la separabilidad no lineal. Minsky y Papert [80], mostraron el 1969 que la combinación de varios Perceptrones simples podían resultar una solución adecuada para tratar ciertos problemas no lineales. Sin embargo, los autores no presentaron una solución al problema de cómo adaptar los pesos de la capa de entrada a la capa oculta, pues la regla de aprendizaje del perceptrón simple no puede aplicarse en este escenario [81], [82], [85].

2.6.2. Arquitectura del Perceptrón Multicapa

La Red Neuronal Perceptrón Multicapa es una red del tipo *feedforward* y su arquitectura se caracteriza por que tiene sus neuronas agrupadas en capas de diferentes niveles. Cada una de las capas esta formada por un conjunto de neuronas y se distinguen tres tipos de capas diferentes: la capa de entradas, las capas ocultas y la capa de salida, como se observa en la imagen 6.1 [86], [87], [88], [89], [90], [91], [92].

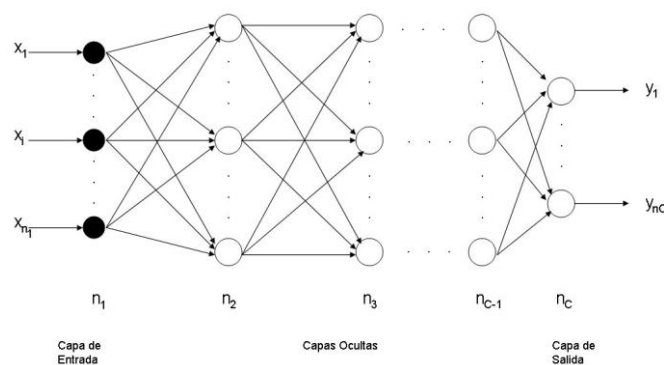


Imagen 2.6.1. Arquitectura del Perceptrón Multilayer Tomada de [86]

Las neuronas de la capa de entrada no actúan como neuronas propiamente dichas, sino que se encargan únicamente de recibir las señales o patrones que procedan del exterior y propagar dichas señales a todas las neuronas de la siguiente capa. La última capa actúa como salida de la red, proporcionando al exterior la respuesta de la red para cada uno de los patrones de entrada. Las neuronas de las capas ocultas realizan un procesamiento no lineal de los patrones recibidos.

El Perceptrón multicapa permite establecer regiones de decisión mucho más complejas que las de dos semiplanos, como el Perceptrón de un solo nivel, que solo puede establecer dos regiones separadas por una frontera lineal en el espacio de patrones de entrada.

Un Perceptrón con tres niveles puede formar cualquier región convexa (o cerrada) en ese espacio. Las regiones convexas se forman mediante la intersección entre las regiones formadas por cada neurona de la segunda capa. Cada elemento se comporta como un perceptrón simple.

Un Perceptrón con cuatro capas puede formar regiones de decisión arbitrariamente complejas. El proceso de separación de clases que se lleva a cabo consiste en la partición de la región deseada en pequeños hipercubos (cuadrados para dos entradas de la red).

Las conexiones entre las neuronas llevan asociado un número real, llamado peso de la conexión. Todas las neuronas llevan también asociado un umbral, que en el caso del Perceptrón multicapa puede tratarse como una conexión más a la neurona, cuya entrada es constante e igual a 1 [83], [84], [94].

2.6.2.1. Diseño de la arquitectura del Perceptrón Multicapa

Cuando se aborda un problema utilizando Perceptrón multicapa, uno de los primeros pasos a realizar es el diseño de la arquitectura de la red. Este diseño implica la determinación de la función de activación a emplear, el número de neuronas y el número de capas en la red.

La elección de la función de activación se suele hacer basándose en el recorrido deseado, y el hecho de elegir una u otra, generalmente, no influye en la capacidad de la red para resolver el problema.

En cuanto al número de neuronas y capas, algunos de estos parámetros vienen dados por el problema y otros deben ser elegidos por el diseñador. Así por ejemplo, tanto el número de neuronas en la capa de entrada como en la de salida, vienen dados por la cantidad de variables que definen el problema.

El número de capas ocultas y el número de neuronas en estas capas deben ser elegidos por el diseñador. No existe un método o regla que determine el número óptimo de neuronas ocultas para resolver un problema dado. En la mayor parte de las aplicaciones prácticas, estos parámetros se determinan por prueba y error.

2.6.2.2. Proceso de Aprendizaje del Perceptrón Multicapa

La Red Neuronal Perceptrón multicapa también se denomina red de retropropagación, ya que como se ve a continuación, durante el entrenamiento se modifican los pesos asociados a las neuronas utilizando el error obtenido en la capa de salida. El aprendizaje se realiza en dos etapas:

- En la primera etapa, los vectores de entrada se presentan a la capa de entrada de la red, esta información se utiliza para realizar varias operaciones a lo largo de las distintas capas hasta la capa de salida, en la cual se genera un resultado.
- En la segunda etapa, el resultado que proporciona la red se compara con el resultado esperado para cada uno de los patrones de entrenamiento, si no coinciden, el error se utiliza para modificar los pesos de las capas intermedias. El hecho de que el error se propague desde la capa de salida hacia la de entrada, es lo que da nombre a este tipo de redes: retropropagación (*backpropagation*).

Para explicar cual es el método de aprendizaje del perceptrón multicapa se utiliza una red $N-L-M$ (ver Imagen 6.2), es decir, una red con una sola capa intermedia, sin que ello suponga pérdida de generalidad, ya que el teorema de Aproximación Universal establece que todos los perceptrones tienen igual capacidad representativa independientemente del número de capas intermedias presentes.

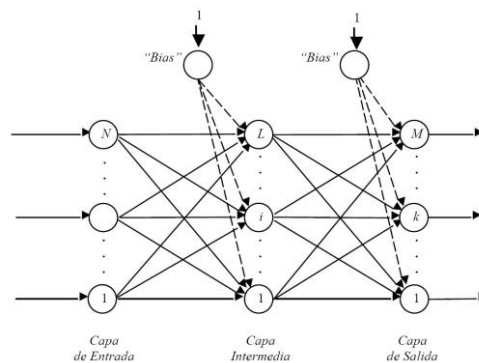


Imagen 2.6.2. Topología de un Perceptrón N-L-M (Tomada de [86])

No existen reglas que indiquen en qué orden se deben presentar los pares de vectores de entrenamiento a la red. En general, es conveniente no entrenar la red de forma consecutiva con grupos de vectores que proporcionen la misma salida.

En términos de notación,

- Se denotará por $x^p = (x_1^p, x_2^p, \dots, x_n^p)^T$ al p -ésimo vector de entrada del conjunto de entrenamiento.
- Se denotará por $y^p = (y_1^p, y_2^p, \dots, y_n^p)^T$ al p -ésimo vector de salida esperado.

- Se utilizará el subíndice i para designar a las neuronas de la capa intermedia y el subíndice k para designar a las neuronas de la capa de salida. Así, por ejemplo, se denotará por s_i , b_i , y d_i al nivel de excitación, “bias” y estado de activación de la neurona i de la capa intermedia, respectivamente. Análogamente para las neuronas de la capa de salida.

En primer lugar, el vector de entrada se propaga hacia la capa intermedia como se muestra en (2.6.2), por tanto para la neurona oculta i , el nivel de excitación s_i , ante el vector de entrada x^p , se determina mediante la siguiente regla de propagación:

$$s_i^p = \sum_{j=1}^n w_{ji} x_j^p + b_i \quad (2.6.1)$$

donde n es el número de entradas de la red, w_{ji} es el valor del peso de la conexión entre la neurona j de la capa de entrada y la neurona i de la capa intermedia y b_i , es el valor de las “bias” asociadas a la neurona i . Si se asume que el estado de activación de la neurona i , es función del vector de entrada a la red, entonces la salida de las mismas viene dada por:

$$y_i^p = F_i(s_i^p) \quad (2.6.2)$$

Análogamente, para una neurona cualquiera k de la capa de salida, las ecuaciones que determinan su estado de activación son:

$$s_i^p = \sum_{i=1}^L w_{ik} y_j^p + b_k \quad (2.6.3)$$

$$y_k^p = F_k(s_k^p) \quad (2.6.4)$$

donde L es el número de neuronas de la capa intermedia, w_{ik} es el valor del peso de la conexión entre la neurona i de la capa intermedia y la neurona k de la capa de salida y b_k es el valor de las “bias” asociada a la neurona k . El valor inicial de los pesos se genera de forma aleatoria, por ejemplo con valores en el intervalo $[-0.1, 0.1]$.

2.6.2.3. Actualización de los pesos que unen la capa intermedia con la de salida

La actualización de los pesos sinápticos se basa en la minimización del error cuadrático medio *LMS* (*Least Mean Square*). Para ello, se considera la siguiente función de error a minimizar:

$$E^p = \frac{1}{2} \sum_{k=1}^M (d_p - y_k^p)^2 \quad (2.6.5)$$

donde el término $(d_p - y_k^p)$ representa el error de la salida presente en la neurona k para el p -ésimo patrón de entrenamiento. El factor $\frac{1}{2}$ se utiliza por conveniencia para calcular la derivada en operaciones posteriores.

Para minimizar la función de error, se debe elegir la dirección hacia la cual deben modificarse los pesos de la capa de salida. La dirección elegida, es la opuesta a la dirección del gradiente (o dirección de máxima variación) del error E^p con respecto a los pesos w_{ik} , y denotado por ∇E^p . Así pues, el cambio en el vector de pesos será proporcional en una constante o tasa de aprendizaje, a la dirección opuesta del gradiente. A este método de minimización se le conoce como método de descenso del gradiente.

Considerando de forma separada cada uno de los componentes de ∇E^p (derivadas parciales de E^p respecto a w_{ik} , $\forall i$), puede escribirse que:

$$\frac{\partial E^p}{\partial w_{ik}} = \frac{\partial E^p}{\partial s_k^p} \frac{\partial s_k^p}{\partial w_{ik}} \quad (2.6.6)$$

Si se define la sensibilidad de la neurona de salida k -ésima ante un vector de entrada p como:

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p} \quad (2.6.7)$$

Considerando (2.6.5) y que las funciones de activación sean funciones diferenciables (lo cual permite aplicar la regla de la cadena), se tiene que la sensibilidad δ_k^p puede escribirse como:

$$\delta_k^p = (d_k^p - y_k^p) \frac{\partial F_k}{\partial s_k^p} \quad (2.6.8)$$

Si se representa la derivada parcial $\frac{\partial F_k}{\partial s_k^p}$ por ∂F_k , y teniendo en cuenta que considerando (2.6.3) se tiene que:

$$\frac{\partial s_k^p}{\partial w_{ik}} = \frac{\partial}{\partial w_{ik}} \left(\sum_{j=1}^L w_{ik} y_j^p + b_k \right) = y_i^p \quad (2.6.9)$$

entonces,

$$\frac{\partial E^p}{\partial w_{ik}} = (d_k^p - y_k^p) F_k'(s_k^p) y_i^p = -\delta_k^p y_i^p \quad (2.6.10)$$

Por tanto, si se considera que la modificación del vector de pesos es proporcional a la dirección contraria del gradiente, y además se tiene en cuenta el término de momento, la regla para actualizar los pesos de la capa de salida viene dada por:

$$w_{ik}(t+1) = w_{ik}(t) + \nabla_p w_{ik}(t) + \mu(w_{ik}(t) - w_{ik}(t-1)) \quad (2.6.11)$$

donde

$$\nabla_p w_{ik} = -\eta \frac{\partial E^p}{\partial w_{ik}} = \eta(d_k^p - y_k^p)F'_k(s_k^p)y_i^p = \eta\delta_k^p y_i^p \quad (6.12)$$

siendo η la tasa de aprendizaje y μ el momento.

A continuación se consideran dos funciones de activación concretas.

En primer lugar, si se considera un comportamiento lineal de la neurona de salida, se tiene que la función de activación de la misma está definida por:

$$F_k(s_k^p) = s_k^p \quad (2.6.13)$$

Teniendo en cuenta que la derivada de la función de activación es 1, se tiene, sin más que sustituir en la expresión (2.6.11) que:

$$w_{ik}(t+1) = w_{ik}(t) + \eta(s_k^p - y_k^p)y_i^p + \mu(w_{ik}(t) - w_{ik}(t-1)) \quad (2.6.14)$$

Esta expresión, sin considerar el momento, coincide con la regla Delta utilizada por Widrow y Hoff [120] en el modelo ADALINE.

En segundo lugar, si se considera que la neurona de salida presenta un comportamiento no lineal, caracterizado por ejemplo, por tener como función de activación la función logística, es decir,

$$F_k(s_k^p) = \frac{1}{1 + e^{-s_k^p}} \quad (2.6.15)$$

Considerando que la derivada de la función de activación viene dada por:

$$F'_k = F(1 - F_k) = y_k^p(1 - y_k^p) \quad (2.6.16)$$

se tiene que:

$$\delta_k^p = (d_k^p - y_k^p)y_k^p(1 - y_k^p) \quad (2.6.17)$$

y por lo tanto

$$w_{ik}(t+1) = w_{ik}(t) + \eta(d_k^p - y_k^p)y_k^p(1 - y_k^p)y_i^p + \mu(w_{ik}(t) - w_{ik}(t-1)) \quad (2.6.18)$$

De forma análoga, el valor de las “bias” se puede actualizar utilizando el mismo procedimiento,

$$b_k(t+1) = b_k(t) + \eta\delta_k^p y_i^p + \mu(b_k(t) - b_k(t-1)) \quad (2.6.19)$$

2.7. Redes Neuronales de Funciones de Base Radial RBF (*Radial Basis Function*)

2.7.1. Introducción

Las redes de neuronas de base radial son redes multicapa con conexiones hacia delante (*feedforward*), al igual que el Perceptrón Multicapa antes visto. Las redes de base radial se caracterizan porque están formadas por una única capa oculta y cada neurona de esta capa posee un carácter local, en el sentido de que cada neurona oculta de la red se activa en una región diferente del espacio de patrones de entrada. Este carácter local viene dado por el uso de las llamadas funciones de base radial, generalmente la función Gaussiana, como funciones de activación. Las neuronas de la capa de salida de las redes de base radial simplemente realizan una combinación lineal de las activaciones de las neuronas ocultas [99], [100], [101], [102].

2.7.2 Topología de la Red de funciones de base radial RBF

La topología de las redes RBF representada en la Figura 3.1 se caracteriza por:

- La capa de entrada de la red de funciones de base radial, es un receptor para los datos de entrada.
- La capa intermedia, realiza una transformación no lineal del espacio de entrada al espacio del nivel intermedio. Las neuronas del nivel intermedio, son las funciones base para los vectores de entrada.
- La capa de salida, calcula la combinación lineal ponderada de las salidas de las neuronas del nivel intermedio.

Las bases son normalmente funciones Gaussianas, cuyas medias y desviaciones estándar pueden calcularse teniendo en cuenta los vectores del espacio de entrada. Por tanto, si \mathbf{x} es el vector de entrada y L el número de neuronas del nivel intermedio, se denota por $h(\mathbf{x}) = (h_1(\mathbf{x}), \dots, h_i(\mathbf{x}), \dots, h_L(\mathbf{x}))^T$, al vector de salidas de las neuronas del nivel intermedio, donde

$$h_i(\mathbf{x}) = e^{-\lambda \|\mathbf{x} - \mathbf{g}_i\|^2} \quad (2.7.1)$$

siendo, g_i los centros de las funciones Gaussianas definidos por los vectores del espacio de entrada, y $\|x - g_i\|$ la distancia euclídea entre un vector de entrada x y el i -ésimo centro g_i .

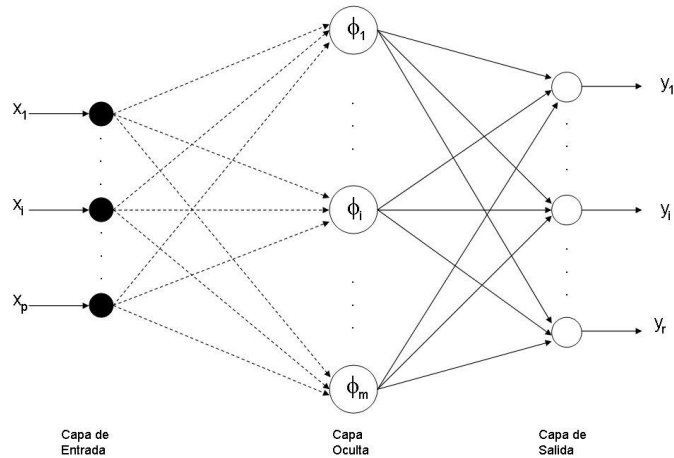


Imagen 2.7.1: Topología de las redes de funciones de base radial (Tomada de [98])

El estado de activación de la k -ésima neurona de la capa de salida y_k , se calcula de acuerdo con la siguiente expresión:

$$y_k = \sum_{i=1}^L w_{ik} h_i(x) + b_k, \quad k = 1, 2, \dots, M \quad (2.7.2)$$

donde w_{ik} , ($i = 1, 2, 3, \dots, L$) representa los pesos de las conexiones entre la capa intermedia y la neurona k -ésima de la capa salida, y b_k son las “bias” de la unidad de salida.

2.7.3. Método de aprendizaje de las RBF

Las RBF son excelentes aproximadores universales, siendo los parámetros que definen el proceso de aproximación los que se citan a continuación:

- Los pesos entre los centros y las neuronas del nivel de salida.
- La posición de los centros.
- Las funciones Gaussianas de los centros.

Al igual que en el MLP, se presentan los pares de vectores de entrenamiento sucesivamente a la red. El valor de los pesos, los centros y las funciones gaussianas se ajusta en cada uno de estos pasos.

El método de aprendizaje, es decir, la forma en que se adaptan los pesos de la red, puede realizarse minimizando el error cuadrático medio, o mediante la utilización de otros algoritmos como el método de Seudo-Inversa [95], [97], [98], [103], [104], [105], [109].

El método de la minimización del error cuadrático medio, se basa en minimizar el error existente entre la salida de la red (y_p) y la salida esperada (d_p), para un vector de entrada (x_p). Así, para el p -ésimo patrón de entrenamiento el error de la red viene dado por:

$$E^p = \frac{1}{2} \sum_{k=1}^M (d_k^p - y_k^p)^2 \quad (2.7.3)$$

Considerando que las funciones base no dependen de los pesos sinápticos entre la capa intermedia y la capa de salida, se tiene que:

$$\begin{aligned} \frac{\partial E^p}{\partial w_{ik}} &= \frac{1}{2} \frac{\partial}{\partial w_{ik}} \left\{ \sum_{j=1}^M (d_k^p - y_k^p)^2 \right\} = \frac{1}{2} \frac{\partial}{\partial w_{ik}} \{(d_k^p - y_k^p)^2\} = \\ &-(d_k^p - y_k^p)^2 \frac{\partial}{\partial w_{ik}} \left\{ \sum_{j=1}^L w_{jk} h_j(x^p) + b_k \right\} = -(d_k^p - y_k^p)^2 h_i(x^p) \end{aligned} \quad (2.7.4)$$

siendo $h_i(x^p)$ la salida del i -ésimo centro para el p -ésimo patrón de entrenamiento.

Una vez determinado el gradiente del error, los pesos pueden actualizarse en la dirección opuesta al mismo, es decir,

$$\Delta_p w_{ik} = -\eta \frac{\partial E^p}{\partial w_{ik}} = \eta (d_k^p - y_k^p)^2 h_i(x^p) \quad (2.7.5)$$

por lo que una vez inicializados los pesos en el instante de tiempo $t = 0$, estos pueden ir actualizándose según se vayan presentando patrones de entrenamiento en instantes sucesivos, de acuerdo con la siguiente regla:

$$w_{ik}(t) = w_{ik}(t) + \Delta_p w_{ik}(t) = w_{ik}(t) + \eta (d_k^p - y_k^p)^2 h_i(x^p) \quad (2.7.6)$$

2.7.3.1. Selección de centros

La selección de los centros es uno de los puntos cruciales en el proceso de entrenamiento de la red. El número y valor de los mismos se puede determinar de muy diversas formas. A continuación se presentan 3 formas de seleccionar un número y valor para los centros [96], [106], [107], [108].

7.3.1.1. Selección aleatoria de centros

Esta primera forma implica la selección de algunos vectores del espacio de entrada. Si es posible, deben ser seleccionados de tal forma que estén repartidos de manera regular por todo el espacio de entrada. En este caso, la desviación estándar de la función Gaussiana utilizada por los centros, se elige teniendo en consideración la separación entre ellos. Así para el centro g_i se tiene que:

$$h_i(x) = G(\|x - g_i\|) = e^{-\frac{L}{d^2}\|x - g_i\|^2} \quad (7.7)$$

donde L es el número de centros, y d es la distancia entre los dos centros más distantes. En este caso, la desviación estándar de la función Gaussiana es:

$$\sigma = \frac{d}{\sqrt{2L}} \quad (7.8)$$

7.3.1.2. Auto selección de centros

Los centros se pueden mover durante el entrenamiento de la red. Una posibilidad es la utilización de métodos de entrenamiento que no requieran supervisión (a la vez que se utiliza entrenamiento o aprendizaje supervisado para ajustar los pesos, por ejemplo utilizando el algoritmo LMS) [96]. Los centros se sitúan en aquellas zonas del espacio de entrada donde haya un número significativo de datos. La norma k vecinos próximos (KNN, *k-nearest-neighbour*), se puede utilizar para ajustar el valor de los centros. Con esta regla, por ejemplo, se puede asociar a cada vector de entrada los centro más próximo al mismo, y así mover dichos centros en la dirección del vector de entrada [106].

7.3.1.3 Selección supervisada de centros

Los centros se pueden ajustar utilizando algoritmos supervisados. En este caso, los cambios en cada una de las componentes del centro g_i , se realizan minimizando el error cuadrático medio. Así, utilizando el método de descenso del gradiente para llevar a cabo la minimización, se tiene que:

$$\Delta g_{ij} = -\eta_2 \frac{\partial E}{\partial g_{ij}} \quad (7.9)$$

donde

- g_{ij} es la componente j -ésima del centro g_i .
- E es el error cuadrático medio total, es decir,

$$E = \frac{1}{2} \sum_{p=1}^P (d^p - y^p)^2 \quad (7.10)$$

donde P es el número total de patrones de entrenamiento, y d^p e y^p son la salida real y deseada para el p -ésimo patrón de entrenamiento, respectivamente.

Dichos algoritmos no garantizan la convergencia, ya que E no está linealmente relacionada con la ubicación de todos los centros y, por tanto, se pueden alcanzar mínimos locales. La tasa de aprendizaje utilizada para ajustar los centros, no necesita ser la misma que la tasa para ajustar los pesos. De forma similar, es posible adaptar la varianza de las funciones Gaussianas para maximizar los efectos de los cambios producidos en los centros. Las redes RBF en las que todos sus parámetros se ajustan mediante la utilización de algoritmos de aprendizaje supervisado, gozan de mayor capacidad de generalización [107].

2.8. Mapas Auto-Organizados De Kohonen – Som (*Self-Organizing Maps*)

2.8.1. Introducción

Los mapas autoorganizados están más ligados a las redes biológicas que otras redes, como por ejemplo las BAM (*Bidirectional Associative Memory*), Hopfield o retropropagación. Esta afirmación se basa en un número de observaciones sobre el estudio del cerebro. Estudios neurobiológicos sugieren que diferentes partes del cerebro, están especializadas o dedicadas a tareas específicas. Los centros correspondientes a actividades tan diversas como el pensamiento, visión, oído y las funciones motoras, se encuentran en zonas concretas del cerebro que están situadas de una cierta manera unas con respecto a otras.

Aunque en gran medida esta organización neuronal está predeterminada genéticamente, es probable que parte de ella se origine mediante el aprendizaje, esto sugiere que el cerebro podría poseer la capacidad inherente de formar mapas topológicos de la información recibida del exterior, de hecho esta teoría podría explicar su poder de operar con elementos semánticos: algunas áreas simplemente podrían crear y ordenar neuronas especializadas o grupos con características de alto nivel y sus combinaciones, en definitiva se construirían mapas especiales para atributos y características. Lo mismo sucede en el entrenamiento de los mapas autoorganizados, donde las neuronas de salida se organizan en grupos de neuronas, y las neuronas de cada grupo responden ante una señal de entrada de un modo similar.

Kohonen fue uno de los pioneros en el campo del aprendizaje no supervisado y el creador de las redes que llevan su nombre, también conocidas como SOM (*Self-Organizing Maps*) [110], consistente en una Red Neuronal de dos capas, una primera capa de entrada y una segunda de competición (Imagen 2.8.1).

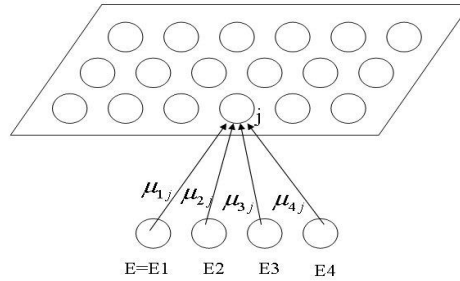


Imagen 2.8.1 Arquitectura del Mapa de Kohonen (Tomada de [110])

La capa de entrada recibe la señal de entrada a la red. Como siempre la dimensión de la entrada depende de los atributos que tengan los datos de entrada, sea n dicho número de atributos, cada entrada viene definida por un vector $\varepsilon = \{\varepsilon_1, \dots, \varepsilon_n\}$. Cada neurona de la entrada está conectada a todas las células de la capa de competición, si hay m células en la capa de competición, entonces los pesos de las conexiones entre las dos capas se puede definir mediante la siguiente matriz:

$$\begin{pmatrix} \mu_{11}, & \mu_{12}, & \dots, & \mu_{1m} \\ \mu_{21}, & \mu_{22}, & \dots, & \mu_{2m} \\ \dots, & \dots, & \dots, & \dots \\ \mu_{n1}, & \mu_{n2}, & \dots, & \mu_{nm} \end{pmatrix}$$

donde μ_{ij} es el peso de la conexión entre la célula i -ésima de la capa de entrada y la j -ésima de la capa de competición. Estas conexiones son inicializadas aleatoriamente al principio de la fase de aprendizaje de la red, y son modificadas a lo largo de dicha fase.

2.8.2. Aprendizaje competitivo

Las redes SOM utilizan el aprendizaje competitivo. Este método de aprendizaje agrupa vectores de entrada próximos entre sí (de acuerdo a alguna métrica), de forma que cada grupo represente una clase o categoría (conjuntos de patrones relacionados entre sí y que se diferencian del resto) identificada en el espacio de entrada. Un patrón de entrada \mathbf{x} , es un punto en el espacio real o binario de vectores N -dimensional. Habitualmente, las neuronas de la capa de salida toman valores binarios (0 ó 1), de forma que si se activa una neurona de salida, se identifica el patrón de entrada con la categoría correspondiente.

En este sentido, la red actúa como un clasificador, el cual toma un vector de entrada y devuelve un índice correspondiente al nodo o neurona que mejor encaja en una determinada clase. La red se organiza a sí misma de forma que las neuronas que corresponden a una determinada clase, tienen respuestas similares, todas las neuronas representan a alguna de estas clases, es decir, las neuronas

de salida se organizan en grupos de neuronas, y las neuronas de cada grupo responden de una manera similar ante vectores de entrada específicos.

2.8.3. Aprendizaje no Supervisado

Los algoritmos de clasificación no supervisados son aquellos que no requieren un etiquetado de cada uno de los vectores de entrada; se suelen llamar también algoritmos auto-asociativos, porque asocian entradas a ellas mismas.

El tipo más común de algoritmos de aprendizaje o clasificación no supervisada son los algoritmos de análisis de grupos o clustering; estos algoritmos tratan de dividir las muestras del conjunto de entrada en una serie de grupos con características comunes. Un algoritmo debe descubrir cuáles son estos clusters, pero también cuáles son las características que define ese cluster y cuántos clusters hay; pero éste último es un problema que no tiene solución fácil.

Los procedimientos de clasificación no supervisados, se basan a menudo en algunas técnicas de agrupamiento de patrones similares. Para llevar a cabo el proceso de clasificación, es necesario definir una distancia o medida de similitud para evaluar el grado de semejanza de los patrones, como por ejemplo: distancia euclídea, absoluta, Manhattan⁷, Voronoi, etc. Normalmente, la más utilizada es la distancia euclídea, que dados dos puntos se define como:

$$d(x_i, x_j) = \|x_i - x_j\|_2 = \sqrt{\sum_k [x_i(k) - x_j(k)]^2}$$

2.8.4. Topología de las Redes SOM

Las redes SOM constan de dos capas: una capa de entrada y una capa de salida. Las neuronas en la capa de entrada, únicamente distribuyen los valores de la señal de entrada a las neuronas de la siguiente capa, dispuestas en un arreglo unidimensional o bidimensional de forma circular, triangular, rectangular, etc., con un cierto orden topológico que depende de la aplicación. Las señales de entrada $x = (x_1, x_2, \dots, x_n)^T$, se aplican a todas las neuronas de salida en paralelo. Una neurona de salida cualquiera k , recibe su entrada a través de un conjunto de conexiones ponderadas. Los pesos de estas conexiones se representan por el correspondiente vector de pesos $w_k = (w_{1k}, w_{2k}, \dots, w_{nk})^T$.

La Imagen 2.8.2 muestra un ejemplo de red de Kohonen. Todas las neuronas de la capa de salida, están conectadas a todas las neuronas de la capa de entrada con pesos w_{jk} . Cuando se presenta un vector de entrada \mathbf{x} , sólo se activa una neurona de la capa de salida (“neurona ganadora”). En un entrenamiento correcto

⁷ suma de los valores absolutos de la diferencia entre observaciones para cada variable.

de la red, todos los vectores de entrada de una misma clase, tendrán como ganadora a la misma neurona de salida.

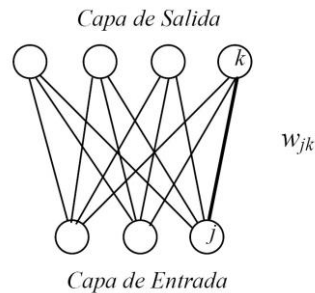


Imagen 2.8.2 Topología de una Red SOM. (Tomada de [110])

2.8.5. Método de aprendizaje de las redes SOM

La idea básica es que las redes SOM incorporan a la regla de aprendizaje competitivo, un cierto grado de sensibilidad con respecto al conjunto de las neuronas vecinas a la ganadora. Esto hace que el proceso de aprendizaje no sea global, sino local (la modificación de los pesos se realiza en una mayor o menor grado dependiendo de la proximidad a la neurona ganadora), y esto ayuda a que se destaquen propiedades topológicas que aparecen en la proyección de características.

Supongamos que un vector de entrada tiene N características, representado por un vector \mathbf{x} de un espacio N -dimensional. La red trata de proyectar el espacio de entrada sobre el de salida. El entrenamiento debe realizarse de forma que la proyección preserve el orden topológico. Kohonen propuso que las neuronas de salida interactuaran lateralmente, llegando así a los mapas de características autoorganizados. La propiedad más importante del modelo, es el concepto de aprendizaje en un vecindario próximo a la neurona ganadora. El tamaño del vecindario decrece en cada iteración.

A continuación, se describe el método de aprendizaje seguido en las redes SOM cuando se presenta un vector de entrada $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$. Para cada neurona de salida k , caracterizada por su vector de pesos $\mathbf{w}_k = (w_{1k}, w_{2k}, \dots, w_{nk})^T$, se determina su estado de activación y_k definido por la distancia entre \mathbf{x} y \mathbf{w}_k .

A continuación, se selecciona como ganadora a aquella neurona de salida, que minimice la distancia entre el vector de entrada actual y el vector de pesos asociado a la misma. En el caso particular de considerar como distancia la euclídea, y que los vectores estén normalizados, se cumple que minimizar la distancia entre dos vectores, es equivalente a maximizar el producto escalar de ambos. De esta forma, se considera que la salida de la neurona es:

$$y_k = \sum_{j=1}^N w_{jk} x_j = w_k^T x \quad (2.8.1)$$

Una vez determinados los estados de todas las neuronas de salida para el vector de entrada \mathbf{x} , se selecciona la neurona i con la máxima salida, es decir $i / y_k \leq y_i, \forall k \neq i$. Los estados se reajustan a continuación de forma que $y_i = 1$ e $y_{k \neq i} = 0$. En este paso se manifiesta el aspecto competitivo de la red.

Una vez que se ha seleccionado la neurona ganadora i , los pesos de la red se actualizan de acuerdo con la siguiente regla de aprendizaje:

$$w_k(t-1) = w_k(t) + \eta g(k, i)(x(t) - w_k(t)) \quad (2.8.2)$$

donde:

- η es la tasa de aprendizaje.
- $g(k, i)$ es una función decreciente con la distancia entre las neuronas k e i , tal que $g(i, i) = 1$.

Por ejemplo, se puede utilizar una función Gaussiana para $g(\cdot)$ de forma que:

$$g(k, i) = e^{-|k-i|^2} \quad (2.8.3)$$

Cada vector de peso que participa en el proceso de actualización, gira libremente hacia el vector de entrada \mathbf{x} . A medida que progresa el proceso de entrenamiento, el vector de pesos de cada neurona va convergiendo a un valor, que es representativo de las coordenadas de puntos que se encuentren cerca de la ubicación física de la neurona ganadora.

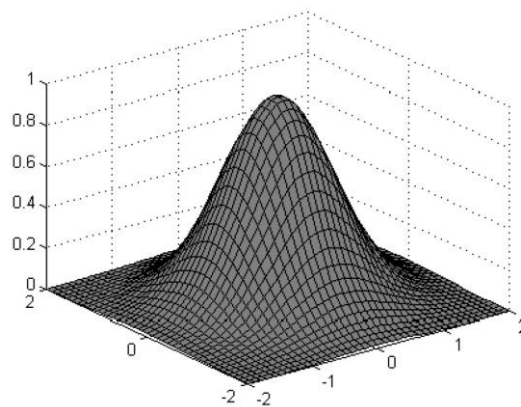


Imagen 2.8.3 Función $g(\cdot)$ gaussiana de la distancia entre las neuronas [111]

CAPÍTULO 3: SELECCIÓN DEL MODELO DE RECONOCIMIENTO DE ROSTROS

3. SELECCIÓN DEL MODELO DE RECONOCIMIENTO DE ROSTROS

En este capítulo se muestran las fases llevadas a cabo durante el proceso de exploración, filtrado y selección de los modelos de reconocimiento de rostros que son la base de este proyecto.

Para la selección de los tres modelos a los cuales se les realizó su análisis y caracterización, se hizo una exploración general de todos los modelos existentes para el desarrollo de proyectos de implementación de software para el Reconocimiento de Rostros.

De esta exploración fueron filtrados los documentos que tienen las características básicas necesarias para el desarrollo del proyecto, como son: que el modelo trabaje con imágenes bidimensionales, que utilice como método de extracción de características la técnica del análisis de los principales componentes y por último que utilice para el reconocimiento o clasificación de los rostros algún tipo de red neuronal artificial.

3.1. Exploración y búsqueda de información

En esta fase se recopilaron los documentos relacionados con el tema de reconocimiento de rostros con el fin de filtrarlos y ubicar los documentos que cumplieran con las condiciones mínimas requeridas para ser involucrados en el proyecto. Las fuentes de información consultadas para este fin fueron:

Publicaciones de Revistas de Investigación

- IEEE (*Institute of Electrical and Electronics Engineers, Inc.*).
- ACM (*Association for Computing Machinery*).
- FERET (*Face Recognition Technology*).
- CiteSeer.IST (*Scientific Literature Digital Library*)
- CIDETEL (*Computing and Information Technology Interactive Digital Educational Library*).
- NSDL (*The National Science Digital Library*).
- DBLP (*Computer Science Bibliography*).
- *The MIT Press. (Massachusetts Institute of Technology)*

Páginas Web de Publicaciones de Tesis y Grupos de Investigación de Universidades

- Universidad del Cauca
- Universidad de Santiago de Chile⁸
- Pontificia Universidad Católica de Chile⁹
- Universidad de Antioquia¹⁰
- Instituto Tecnológico de Hermosillo¹¹
- Universidad de Castilla- La Mancha¹²
- Universidad Politécnica de Valencia¹³
- *University of Illinois at Urbana-Champaign*¹⁴
- *McGill University*¹⁵
- *Massachusetts Institute of Technology*¹⁶
- *University of Ljubljana*¹⁷
- *University Duisburg*¹⁸
- *MIT Media Laboratory Vision and Modeling Group*¹⁹
- *Nacional University of Singapore*²⁰
- *Georgia Institute of Technology*²¹
- *University of Brasilia*²²
- *Carnegie Mellon University*²³
- *Mitsubishi Electric Research Laboratories*²⁴
- *University of Maryland*²⁵
- *Gjovik Callege*²⁶
- *University of Windsor*²⁷
- *Amirkabir University of Technology*²⁸

Como resultado de la exploración general de los modelos existentes para el reconocimiento de Rostros con imágenes bidimensionales fueron obtenidos, de fuentes confiables, 70 documentos relacionados con el tema, estos documentos

⁸ http://cabierta.uchile.cl/revista/17/articulos/paper4/#_ftn1

⁹ http://www2.ing.puc.cl/~iing/ed432/anales/reconocimiento_de_caras.htm

¹⁰ <http://neurologia.rediris.es/congreso-1/conferencias/neuropsicologia-2-2.html>

¹¹ <http://www.depi.itchihuahua.edu.mx/electro/archivo/electro2001/mem2001/papers.html>

¹² <http://www.uclm.es/>

¹³ <http://www.iti.upv.es/>

¹⁴ <http://vision.ucsd.edu/~leekc/papers/cvpr01b.pdf>

¹⁵ <http://www.cim.mcgill.ca/~mrl/>

¹⁶ <http://www.csail.mit.edu/index.php>

¹⁷ <http://vicos.fri.uni-lj.si/>

¹⁸ <http://www.uni-duisburg-essen.de/index.shtml.en>

¹⁹ <http://vismod.media.mit.edu/vismod/demos/facerec/>

²⁰ <http://www.iss.nus.edu.sg/iss/index.jsp>

²¹ <http://www.cc.gatech.edu/>

²² <http://www.unb.br/portal/>

²³ <http://www.csd.cs.cmu.edu/>

²⁴ <http://www.merl.com>

²⁵ <http://www.umiacs.umd.edu/>

²⁶ <http://www.hig.no/eway/default0.asp?>

²⁷ <http://www.uwindsor.ca/units/eng-electric/electric2.nsf/tovr/F7F0FDECEB796F7C85256FA3006D38DA>

²⁸ <http://ele.aut.ac.ir/pajou/home.htm>

fueron sometidos a un análisis más profundo con el fin de seleccionar los tres modelos que cumplen con los requisitos necesarios para ser considerados como los modelos que fundamentan los estudios realizados en este proyecto.

3.2. Criterio de selección de los modelos

Para la selección de los tres modelos, necesarios para el desarrollo del proyecto, se tuvieron en cuenta las siguientes consideraciones:

- El modelo debe hacer el reconocimiento de rostros sobre imágenes de dos dimensiones, en escala de grises.
- Usar la técnica estadística de Análisis de Principales Componentes.
- Usar dentro de su sistema algún tipo de red neuronal.
- Los tres modelos deben presentar alguna similitud en los procesos anteriores al reconocimiento de la imagen o de clasificación de las imágenes.
- Los tres modelos deben usar la misma base de rostros.
- Se debe contar con la suficiente información sobre el modelo, como para tener criterios de comparación con los demás seleccionados.
- Los modelos deben ser reconocidos a nivel internacional.
- Cada modelo seleccionado debe usar un tipo de red neuronal diferente al anterior, debido a que en este punto es donde se realizaran las comparaciones de eficiencia y eficacia del modelo.

3.3. Modelos Seleccionados

Una vez recopilada la documentación mas relevante relacionada con los modelos utilizados para el reconocimiento de rostros, y establecidos las condiciones que deben cumplir los modelos que se necesitan para el desarrollo del proyecto, se procedió a hacer la selección de tres modelos que servirán de base de estudio. De dicha selección fueron obtenidos los siguientes documentos:

3.3.1. Face recognition: a convolutional neural-network approach

Para efectos de facilidad en la comprensión del documento este modelo se identificará con el acrónimo de modelo FR_SOM.

Documento obtenido de la IEEE²⁹.

Autores: Steve Lawrence, C. Lee Giles, Ah Chung Tsoi y Adrew D. Back.

Fecha de Publicación: enero de 1997

Identificación Digita: 10.1109/72.554195

²⁹ http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?isnumber=12062&arnumber=554195&type=ref

Puesto en línea: 2002-08-06 21:24:45.0

3.3.2. Face Recognition with Multi-Layer Perceptrons

Para efectos de facilidad en la comprensión del documento este modelo se identificará con el acrónimo de modelo FR_PML.

Documento obtenido de CiteSeer³⁰.
Autores: Erik Hjelmás y Jorn Wroldsen.

Fecha de Publicación: Jul 1999

3.3.3. N-Feature Neural Network Human Face Recognition

Para efectos de facilidad en la comprensión del documento este modelo se identificará con el acrónimo de modelo FR_RBF.

Documento obtenido de IEEE³¹
Autores: Javad Haddadnia, KarimFaez y Majid Ahmadi.
Fecha de Publicación: 2004
Volumen: 22
Páginas: 1071-1082

3.4. Caracterización de los modelos seleccionados

Los modelos seleccionados presentan las características relacionadas en la tabla 3.1.

Nombre del Modelo	Componentes del sistema	Base de Datos	Red Neuronal	Formato Imagen	Imágenes (Cant.)	Img. Test	Elementos PCA	Mejor Resultado
FR_PML	<ul style="list-style-type: none">•Input Face•PCA (Dimensionality reduction)•Scaling Procedure•Classifier: Single MLP or Committee MLP	ORL Database	Perceptron MultiLayer	PGM (256 gray levels)	400	200	40	90,5%
FR_RBF	<ul style="list-style-type: none">•Input Image•Face Localization•Feature Extraction•Neural Network Classifier•Output person	ORL Database	Radial Bases Function	PGM (256 gray levels)	400	200	30	99,7%
FR_SOM	<ul style="list-style-type: none">•Images•Image Sampling•Dimensionality reduction•Features Estraction•Classifier•Clasification	ORL Database	Self Organization Map	PGM (256 gray levels)	400	200	46	100,0%

Tabla 3.1 Caracterización de modelos seleccionados

³⁰ <http://citeseer.ist.psu.edu/66390.html>

³¹ <http://iee-biometrics.dti.unimi.it/knowledge.htm> - <http://citeseer.ist.psu.edu/708567.html>

3.4.1. Descripción de las características de los modelos

A continuación se presenta una descripción de cada una de las características encontradas en cada uno de los modelos seleccionados las cuales se encuentran relacionadas en la tabla 3.1.

3.4.1.1. Descripción de las características del Modelo FR_PML

- **Componentes del sistemas:** El modelo FR_PML presenta como principales componentes del sistema los siguientes módulos, un modulo para el ingreso de las imágenes. El siguiente modulo es el encargado de recibir la imagen convertida en una matriz para realizarle varios procesos con el fin de disminuir su dimensionalidad y de analizar sus principales componentes. Debido a que los coeficientes obtenidos en el proceso del análisis de los principales componentes son de magnitudes de diferente orden, se los compensa por medio de un escalamiento lineal antes de ser usados como entradas de entrenamiento del clasificador. El siguiente modulo está compuesto por el clasificador, que es una red neuronal basa en la estructura del Perceptron MultiCapa.
- **Base de Datos:** la base de datos utilizada por este modelo, al igual que los otros dos modelos, como base para el entrenamiento de las redes neuronales, así como para las respectivas pruebas de clasificación es: *The ORL database of Faces*, que contiene un conjunto de imágenes tomadas entre abril de 1992 y abril de 1994 en *AT&T Laboratorios Cambridge*, la base de datos contiene diez (10) imágenes diferentes de cada uno de los 40 sujetos distintos³².
- **Red Neuronal Artificial:** la red neuronal utilizada en este modelo es la Perceptron multicapa.
- **Formato de la Imagen:** el formato de las imágenes utilizadas el es *PGM (Portable Gray Map)*.
- **Cantidad de Imágenes:** la cantidad de imágenes utilizadas para la implementación y pruebas de este modelo es de 400 imágenes, 200 para el entrenamiento, que correspondes a 5 imágenes por individuo para el entrenamiento de la red neurona, y las doscientas restantes para el *test* de prueba.
- **Elementos PCA para el entrenamiento de la RNA:** en este modelo se hace uso de los treinta (30) principales componentes de cada una de las imágenes.

³² <http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html>

- **Mejor Resultado:** el mejor resultado obtenido en el proceso de pruebas para este modelo es del 99.7%.

3.4.1.2. Descripción de las características del Modelo FR_RBF

- **Componentes del sistemas:** El modelo FR_RBF presenta como principales componentes del sistema los siguientes módulos, un modulo para el ingreso de las imágenes. El siguiente modulo es el encargado de ubicar el rostro dentro de la imagen. El siguiente modulo recibe la imagen del rostro y la procesa con el fin de extraer sus principales características, las cuales son utilizadas para el entrenamiento de la red Neuronal de clasificación, que es lo que componen el siguiente modulo. De este modulo se obtiene una salida que es la respuesta de la red neuronal que es la imagen de la persona ya clasificada.
- **Base de Datos:** la base de datos utilizada por este modelo es la misma del modelo anterior *The ORL database of Faces*.
- **Red Neuronal Artificial:** la red neuronal utilizada en este modelo es la basada en una estructura de Función de Base Radia.
- **Formato de la Imagen:** el formato de las imágenes utilizadas el es *PGM (Portable Gray Map)*.
- **Cantidad de Imágenes:** la cantidad de imágenes utilizadas en este modelo para la implementación y las pruebas, al igual que en los otros dos modelos, es de 400 imágenes, 200 para el entrenamiento, que corresponden a 5 imágenes por individuo para el entrenamiento de la red neurona, y las doscientas restantes para el *test* de prueba.
- **Elementos PCA para el entrenamiento de la RNA:** en este modelo se hace uso de los cuarenta (40) principales componentes de cada una de las imágenes.
- **Mejor Resultado:** el mejor resultado obtenido en el proceso de pruebas para este modelo es del 90.5%.

3.4.1.3. Descripción de las características del Modelo FR_SOM

- **Componentes del sistemas:** El modelo FR_SOM presenta como principales componentes del sistema los siguientes módulos, un modulo para el ingreso de las imágenes. El siguiente modulo es el encargado de recibir la imagen y le realiza un Image Sampling³³ para posteriormente pasarla al modulo

³³ Maneras de obtener valores antes de dibujar una imagen gráfica cuando se ha agrandado (ampliando los píxeles) o se ha contraído (comprimiendo los píxeles). Los algoritmos hacen compensaciones entre la calidad y la velocidad de dibujo.

que le deduce la dimensión a la imagen. De este modulo pasa al modulo de procesamiento para la extracción de características principales, las cuales pasan al modulo de clasificación, en un principio para hacer el entrenamiento de la red neuronal utilizada en este modelo que es la *Self Organization Map* y finalmente para la clasificación de los rostros con las misma RNA.

- **Base de Datos:** la base de datos utilizada por este modelo, al igual que los otros dos modelos, como base para el entrenamiento de las redes neuronales, así como para las respectivas pruebas de clasificación es: *The ORL database of Faces*.
- **Red Neuronal Artificial:** la red neuronal utilizada en este modelo es la basada en una arquitectura de Mapas Auto Organizado.
- **Formato de la Imagen:** el formato de las imágenes utilizadas el es *PGM (Portable Gray Map)*.
- **Cantidad de Imágenes:** la cantidad de imágenes utilizadas para la implementación y pruebas de este modelo es de 400 imágenes, 200 para el entrenamiento, que correspondes a 5 imágenes por individuo para el entrenamiento de la red neurona, y las doscientas restantes para el *test* de prueba.
- **Elementos PCA para el entrenamiento de la RNA:** en este modelo se hace uso de los cuarenta y seis (46) principales componentes de cada una de las imágenes.
- **Mejor Resultado:** el mejor resultado obtenido en el proceso de pruebas para este modelo es del 100%.

3.5. Análisis Comparativo de Modelos Seleccionados

Para el análisis comparativo de los modelos se deben tener en cuenta algunos aspectos fundamentales que permitan que permitan determinar cual de los modelos es el que presenta un mejor comportamiento frente a los demás en el proceso de clasificación de rostros.

Los aspectos que se deben tener en cuenta son: ubicar todos los puntos de similitud que puedan tener los modelos seleccionados, determinar cuales son las características que hacen que un modelo presente algunas ventajas frente a los demás, y viceversa, o sea cuales pueden ser sus desventajas.

Los tres modelos seleccionados tienen una serie de similitudes que nos permiten llegar hasta un nivel específico como lo es el del análisis de los principales componentes, pasando por la captura de las imágenes y el preprocesamiento de

las mismas. Una vez se llega a este punto se analiza cada uno de los modelos independientemente en busca de las características que nos permiten identificar las ventajas y desventajas que presenta un modelos con relación a los demás.

Como se puede observar la diferencia fundamental en los tres modelos se encuentra en el tipo de red neuronal que utiliza para hacer la identificación o clasificación de los rostros, por tal motivo, es aquí donde se centran los esfuerzos para determinar teóricamente cual modelo se puede considerar como el más eficiente con respecto a los otros dos.

Dicho lo anterior, el paso siguiente es la caracterización de los modelos y el análisis de los tipos de red que cada uno de ellos implementa.

3.5.1. Modelo FR_PML [112]

Modelo para un sistema de reconocimiento de rostros con extracción de características usando Análisis de Principales componentes, y una red Neuronal Perceptrón Multicapa (MLP) para la clasificación.

Los datos involucrados en el modelo al igual que en los demás modelos escogidos es la base de datos ORL, que consiste en 10 imágenes de rostros de 40 diferentes personas, para un total de 400 imágenes. La base de datos tiene algunas variaciones en posición, escala, expresión facial y detalles. De la base de datos se extraen 200 imágenes para el entrenamiento de la red neuronal (cinco imágenes de cada sujeto), y las 200 restantes para el testeo.

Se utiliza el Análisis de Principales Componentes (*PCA*) como preprocesamiento (reducción de la dimensión y extracción de características) de las imágenes de entrada. El conjunto de entrenamiento se pasa por un *PCA* con el fin de encontrar los 200 eigenvectores correspondientes (principales componentes). En la literatura se exige que para una buena reconstrucción se deben usar como mínimo los 40 primero componentes principales.

3.5.1.1. Descripción del Sistema

La imagen 1.1 ilustra la descripción del sistema. Las imágenes de entrada son en formato PGM, con 256 niveles de grises por píxel. Después de ser cargada la matriz, esta es convertida en un vector mediante la concatenación de las columnas de la matriz.

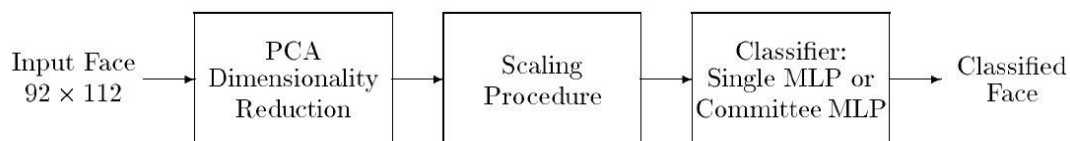


Imagen 3.1 Descripción del sistema de reconocimiento de rostros

Se constituye el conjunto de entrenamiento extrayendo los *eigenvectores*, se buscan los coeficientes de expansión de los *eigenvectores* logrando una reducción de la dimensión de las imágenes (1034 (92x112) a 200 o menor).

Topologías diferentes de Redes Neuronales MLP se entrenaron para realizar la tarea de clasificación de los rostros. Se repitió la misma topología cinco veces con una inicialización aleatoria de todos los parámetros en las MLP, así cada resultado relacionado en la tabla 3.1 es el promedio de los cinco intentos.

El algoritmo de entrenamiento utilizado fue el de Retropropagación, con un porcentaje de entrenamiento bajo (*normally* < 0.01). Cada MLP alcanzó la meta de error cuando la suma promedio de error era < 1. (Sumada sobre todos los patrones y las unidades de salida).

Principal Components	Net Topology	Best	Average
1-40	40:40:40	89.5	88.9
1-40	40:20:40	86.5	83.8
1-40	40:80:40	90.5	89.1

Tabla 3.2 Diferentes topologías de red para los mismos datos de entrada

Según los autores, lo más importante de este modelo sería la cantidad de información disponible en los coeficientes de los *eigenrostros*. Si es significativa la disminución de datos irrelevantes de los rostros en la reducción, y quizá alrededor del 92% es el mejor resultado de clasificación que se puede obtener.

Principal Components	Net Topology	Best	Average
1-25	25:40:40	89	87.7
1-30	30:80:40	88.5	87.9
1-35	35:80:40	90	89.7
1-40	40:40:40	89.5	88.9
1-50	50:40:40	89.5	86.6
1-100	100:60:40	86	84.1
1-200	200:80:40	84	82.2

Tabla 3.3 Porcentaje de Clasificación correcta en el conjunto de testeo (200 imágenes)

Los coeficientes de los *egienrostros* pueden ser considerados como macro características de la imagen, lo contrario a las características locales como los ojos, nariz y la boca. Estas características locales causan ciertos problemas en el reconocimiento de rostros con alto grado de distorsión en las imágenes con relación al conjunto de imágenes de entrenamiento de la red neuronal.

3.5.1.2. Ventajas y desventajas del modelo

Ventajas

Las redes neuronales *MLP* son clasificadores de patrones de gran alcance. Con una o dos capas ocultas pueden aproximar virtualmente cualquier mapa de entrada-salida. Se ha demostrado que pueden aproximarse al funcionamiento de clasificadores estadísticos óptimos en problemas difíciles. Utilizan eficientemente la información contenida en los datos de entrada.

Desventajas

Las redes neuronales *MLP* son clasificadores estáticos; es decir, el mapa de la entrada-salida depende solamente de la entrada actual. Si deseamos procesar datos temporales, cada vez la muestra tiene que ser alimentada de diversas entradas, requiriendo redes muy grandes. Se necesitan porciones de datos de entrada. El entrenamiento puede ser lento, y el ajuste de los parámetros puede ser difícil para problemas difíciles.

3.5.2. Modelo FR_RBF [113]

Este modelo introduce un nuevo método para el reconocimiento de rostros humanos que emplea un conjunto de diferentes clases de características de las imágenes de los rostros y una red neuronal *Radial Basis Function (RBF)* llamado *Hybrid N-Feature Neural Network (HNFNN)* sistema de reconocimiento de rostros humanos.

3.5.2.1. Descripción del Sistema



Imagen 3.2 Descripción del sistema de reconocimiento de rostros

El modelo toma una imagen y extrae de ella el rostro objeto de análisis como se muestra en el modelo de descripción de la imagen 3.2. Una vez localizado el rostro dentro de la imagen se forma una sub-imagen que contiene únicamente la información del rostro que se desea analizar. Una vez obtenida la imagen con la que se desea trabajar, se le extraen las características necesarias por medio del método *PCA*, las cuales se utilizarán tanto para el entrenamiento de la red neuronal como para el testeo de la misma. Una vez obtenidos los vectores característicos de las imágenes se procede al entrenamiento y clasificación por medio de una red neuronal *RBF*.

Para comprobar la efectividad del modelo propuesto, se realizaron estudios experimentales sobre la base de datos de imágenes de rostros *ORL* de la

Universidad de *Cambridge*, 400 imágenes de rostros de 40 individuos diferentes, han sido utilizadas para evaluar las características del modelo propuesto. Ninguno de los 10 ejemplares es idéntico al otro. Las imágenes varían en posición, rotación, escala y expresión. En esta base de datos cada persona ha cambiado su expresión facial en cada uno de los 10 ejemplos (abrir/cerrar los ojos, sonreír, etc.) Para algunos individuos, las imágenes fueron tomadas en momentos diferentes, variando detalles faciales. Ejemplos de las imágenes de la base de datos usada se presentan en la imagen 3.3.



Imagen 3.3 Ejemplo de rostros de la Base de datos *ORL*

El vector característico extraído con el *PCA* ha sido creado con los 30 principales valores característicos.

Un total de 200 imágenes han sido usadas para el entrenamiento de la red, y las 200 restantes son usadas para el Testeo. Cada conjunto de entrenamiento consta de 5 imágenes del mismo individuo seleccionadas aleatoriamente.

El resultado de la experimentación dio como índice de clasificación acertada un 99.7% de efectividad.

3.5.2.2. Ventajas y desventajas del modelo

Ventajas

La ventaja de las redes neuronales de función de base radial es que buscan las entradas para dar un mapa como salida usando aproximadores locales. Cada una de estas piezas es cargada linealmente en la salida de la red. Debido a que tienen pocos pesos, estas redes se entrenan a extremadamente rápido y requieren de pocas muestras para el entrenamiento.

Desventajas

Un problema podría ser que se requiere de muchas funciones de base radial para cubrir un espacio de dimensionalidad grande.

3.5.3. Modelo FR_SOM [114]

El sistema usado para el reconocimiento de rostros en este modelo es una combinación de una reducción de dimensionalidad, extracción de *eigenfaces* (PCA) y redes neuronales *Self Organizing Map* (SOM).

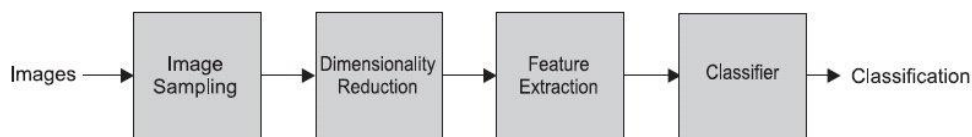


Imagen 3.4 Caracterización del sistema de reconocimiento de rostros

Las imágenes de rostros utilizadas para la experimentación es la *ORL*, que como ya se ha mencionado antes consiste en 400 imágenes de rostros de 40 individuos diferentes.

Se toman al igual que los otros dos modelos anteriores 200 imágenes para el entrenamiento y las 200 restantes para el testeo.

3.5.3.1. Descripción del Sistema

Para el conjunto de imágenes de entrenamiento, una ventana de tamaño fijo (por ejemplo 5x5) se mueve sobre toda la imagen y muestras locales son extraídas en cada paso. Cada paso la ventana es movida cada cuatro píxeles.

Una *SOM* (por ejemplo de tres dimensiones y cinco nodos por dimensión $5^3 = 125$ total de nodos) es entrenada con los vectores obtenidos de la etapa anterior. La *SOM* cuantifica los vectores de la entrada 25-dimensionales en 125 valores topológicamente solicitados. Las tres dimensiones de las *SOM* se pueden tomar como tres características.

La misma ventana antes relacionada, es pasada tanto por el conjunto de imágenes de entrenamientos como las de prueba. Las muestras locales de la imagen son pasadas por el *SOM*, de tal modo que se crean nuevos conjuntos de entrenamiento y de prueba en el espacio de la salida creado por el mapa de auto-organizado. (Cada imagen es ahora representada por 3 mapas, cada uno de los cuales corresponde a una dimensión en el *SOM*. El tamaño de estos mapas es igual al tamaño de la imagen de entrada (92x112) dividida por el tamaño de paso (para un tamaño de paso de 4, los mapas son 23x28).)

3.5.3.1. Detalles de la simulación

Para la *SOM*, el entrenamiento es dividido en dos partes, una fase de ordenamiento y una fase de ajuste. 100.000 actualizaciones son realizadas en la primera fase y 50.000 en la segunda fase. En la primera fase el radio de la vecindad se inicia en dos tercios del tamaño del mapa y se reduce linealmente hasta 1. El porcentaje de

aprendizaje en esta fase es: $0.7 * (1 - n/N)$ donde n es el número inicial de las actualizaciones, y N es el número total de actualizaciones. En la segunda fase el radio de la vecindad arranca en 2 y se reduce hasta 1. El porcentaje de aprendizaje durante esta fase es: $0.02 * (1 - n/N)$

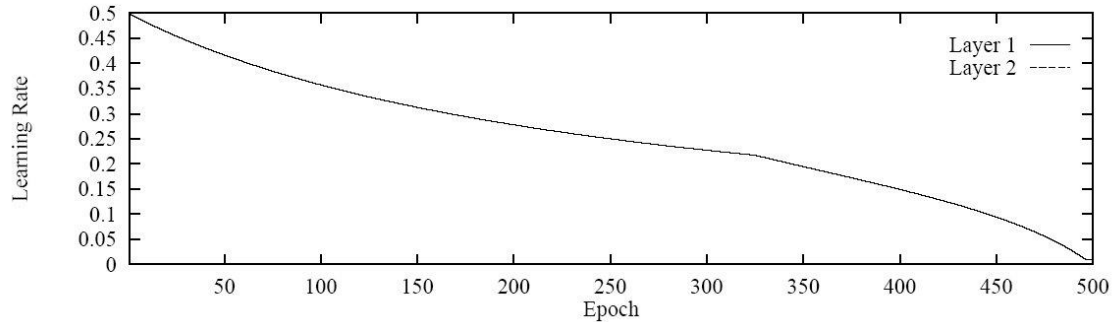


Imagen 3.5 Porcentaje de aprendizaje en función del número de ciclos

3.5.3.2. Ventajas y desventajas de las redes neuronales SOM

Ventajas

La ventaja dominante de esta red es el *clustering* producido por el SOM que reduce el espacio de entrada en características representativas usando un proceso de auto-organización. Por lo tanto la estructura subyacente del espacio de entrada se guarda, mientras que la dimensionalidad del espacio se reduce.

Desventajas

Es difícil medir el funcionamiento del SOM, dando como resultado la ausencia de un criterio de la parada. Esto puede afectar la estabilidad del mapa.

3.6. Elección del modelo para implementación del Prototipo

Al realizar el análisis comparativo de los modelos y del tipo de red neuronal que utilizan para la clasificación de los rostros, y observando los resultados expuestos por los autores de los modelos en sus respectivas pruebas, que nos indican que el modelo FR_SOM presenta una ventaja con relación al acierto en la clasificación del de 9.5% frente al modelo FR_PML y un ventaja del 0.3% frente al modelo FR_RBF, se puede afirmar que el modelo con el cual se obtuvieron mejores resultados fue con el modelo FR_SOM, por tanto, la conclusión a la cual se ha llegado es que teóricamente el modelo FR_SOM presenta un mejor desempeño frente a los otros dos modelos, por tal razón ha sido elegido para el desarrollo del prototipo funcional de este proyecto. En la tabla 3.4 se presenta una relación mas detallada de las ventajas y desventajas encontradas en los modelos.

El prototipo que se desarrollará será con el fin de determinar si los resultados teóricos concuerdan con los resultados que se obtengan al realizar las respectivas pruebas con el prototipo de Reconocimiento de Rostros que se implemente.

Modelo	Tipo de RNA	Ventajas	Desventajas	Resultado de Prueba
FR_SOM	Mapas Auto Organizados	Formación de clusters. Reducción de Dimensionalidad.	Dificultad en la medición del rendimiento. Inestabilidad del Mapa.	100% de exactitud en la clasificación
FR_PML	Perceptrón Multicapa	Clasificadores de gran alcance. Uso eficiente de la información de entrada.	Clasificadores estáticos. Entrenamiento lento. Difícil ajuste de parámetros.	92% de exactitud en la clasificación
FR_RBF	Funciones de Base Radial	Rapidez en el entrenamiento. Pocas muestras para el entrenamiento.	Requiere de muchas funciones de base Radial.	99.7% de exactitud en la clasificación

Tabla 3.4 Análisis comparativo de los modelos

A continuación se presenta la tabla 3.5 en la cual se relacionan los factores e indicadores de referencia tenidos en cuenta para el análisis del documento seleccionado que contiene el modelo de referencia.

FACTORES	INDICADORES
1. Aspectos Formales (Características del autor y del documento)	1.1. Autores: Steve Lawrence, C. Lee Giles, Ah Chung Tsoi y Adrew D. Back. 1.2. Título del Documento: <i>Face recognition: a convolucional neural-network approach.</i> 1.3. Tipo de Material: reporte Técnico.
2. Asunto Investigativo (Objeto, fenómeno o proceso de estudio)	2.1. Tema Central: Reconocimiento de Rostros. 2.2. Núcleo Temático: Análisis de Principales componentes, Redes Neuronales, Reconocimiento de Rostros. 2.3. Problema: Presentar una solución con redes neuronales comparando la favorabilidad con otros métodos.

<p>3. Delimitación contextual (Parámetros relacionados con el contexto de la investigación)</p>	<p>3.1. Delimitación Espacial: Internacional 3.2. Delimitación Temporal: 2002-08-06 21:24:45.0 3.3. Sujetos Investigados: Convolutional Network, SOM.</p>
<p>4. Propósito (Fin buscado por el autor con los resultado de su investigación)</p>	<p>4.1. Objetivo General: Realizar un Sistema de Reconocimiento de Rostros en tiempo Real. 4.2. Objetivos Específicos: Buscar una imagen dentro de una base de datos que corresponde a una imagen de referencia. Identificar personas en tiempo real, permitir el ingreso de un grupo de personas y negar el ingreso a otro grupo de ellas.</p>
<p>5. Enfoque (Referente disciplinar y conceptual desde el cual se analiza el objeto de estudio)</p>	<p>5.1. Disciplina: Biometría. 5.2. Referentes Teóricos: Características Geométricas. Eigenfaces. Análisis de Principales Componentes. Plantilla de concordancia. Redes Neuronales. Mapas Auto Organizados. Transformada de Karhunen Loeve. Convolución de Redes. 5.3. Conceptos Principales: Biometría, Reconocimiento de Rostros, Análisis de Principales Componentes, Redes Neuronales SOM. 5.4. Tesis: El uso de las redes SOM permiten implementar un sistema de clasificación confiable y ágil. 5.5. Tipo de Investigación: Exploratoria.</p>
<p>6. Resultados. (Los señalados por el autor en el documento como producto de su investigación)</p>	<p>6.1. Conclusiones: Se ha presentado un sistema rápido, automático y confiable para el reconocimiento de rostros. El cual es la combinación de imágenes locales en una red neuronal de mapas auto organizados y convolución de redes.</p>

Tabla 3.5. Factores e Indicadores de referencia para el análisis del documento *Face recognition: a convolutional neural-network approach* [116].

4. Análisis del Sistema

En este capítulo se describen los requerimientos funcionales del sistema, basándose en las especificaciones técnicas de de la arquitectura de los modelos seleccionados. Además se presentan los actores y casos del uso del sistema utilizando notación *UML (Unified Modeling Lenguaje)* [118].

4.1. Requerimientos

Los requerimientos para el Prototipo de Reconocimiento de Rostros se pueden dividir en las siguientes categorías:

- Adquisición de la Imagen
- Preprocesamiento de la Imagen
- Análisis de Principales componentes
- Entrenamiento de la Red Neuronal Artificial
- Clasificación de los Rostros

4.1.1. Adquisición de la Imagen

Las imágenes son incorporadas en el sistema en dos ocasiones diferentes y de dos maneras diferentes:

4.1.1.1. Entrenamiento General de la Red Neuronal Artificial

Para el entrenamiento general de la red neuronal artificial se deben tomar todas las imágenes con que se cuenta en la base de datos y pasarlas por el proceso de preprocesamiento con el fin de normalizar las imágenes.

Este proceso debe hacerse de manera automática buscando la base de datos y recorriendo todas las imágenes existentes con el fin de extraerle sus principales componentes para posteriormente realizar el entrenamiento de la red neuronal artificial mediante el simulador.

4.1.1.2. Clasificación Individual de los Rostros

Las imágenes en esta ocasión se debe seleccionar manualmente ya sea desde la base de datos o adquirirla des de una cámara con el fin de extraerle sus principales componente y pasarlos por la red neuronal con el fin de su clasificación.

4.1.1.3. Imágenes desde Base de Datos

Para levantar las imágenes individualmente se debe poder abrir la base de datos y seleccionar una de las imágenes de los rostros con que se cuenta en la base de datos y pasarla por el sistema de preprocesamiento para poder normalizar las imágenes y posteriormente extraerle sus principales características las cuales se pasan por el red neuronal con el fin de clasificar la imagen.

4.1.1.4. Imágenes desde Cámara Digital

Para la clasificación de los rostros se debe poder tomar la imagen desde una cámara digital, y una vez se cuenta con la imagen se debe pasar por los filtros necesarios para poder obtener el rostro de la imagen tomada, y posteriormente normalizarla y extraer sus características al igual que el paso anterior con el fin de estos valores pasarlos a la red neuronal para clasificar la imagen.

4.1.2. Preprocesamiento de la Imagen

En la fase de preprocesamiento de la imagen se deben tener en cuenta dos cosas: la primera es que si la imagen es tomada desde la cámara digital, esta imagen se debe filtrar con el fin de extraer el rostro que contiene la imagen. La segunda es que la imagen debe ser preprocesada con el fin de normalizar todas las imágenes una vez se haya extraído el rostro de la imagen, y este proceso se le debe realizar también a las imágenes levantadas de la base de datos.

4.1.2.1. Filtro de extracción de rostro de la imagen

La imagen tomada desde la cámara debe ser pasada por un filtro que se encargue de extraer el rostro de la imagen y que se deseche el resto de información que no es relevante dentro de la imagen para los propósitos del proyecto.

El filtro se le debe aplicar a las imágenes que van a formar parte de la base de datos con la que se entrenará la Red neuronal artificial, así como a las imágenes individuales que se presentaran a la red neuronal para su clasificación.

4.1.2.2. Paso de la Imagen a Escala de Grises

El primer filtro que se debe aplicar a la imagen obtenida es el de convertir la imagen de un formato *RGB* (*Red, Green, Blue*) a un formato de Escala de grises.

4.1.2.3. Filtro de Reducción del tamaño de la imagen

Dentro de los filtro que se debe aplicar a las imágenes con el fin de normalizarlas es el de reducir su tamaño, esto nos garantizará que todas las imágenes, tanto las que se utilizaran para el entrenamiento, como las que se utilizarán individualmente para la clasificación tendrán las mismas dimensiones, tanto de ancho como de alto.

4.1.2.4. Filtro de Binarización de la imagen

Al igual que el filtro anterior, este hace parte de los filtros encargados de la normalización de las imágenes, y este filtro debe encargarse de umbralizar los valores de los píxeles de las imágenes, con el fin de trabajar con una matriz de de dos tipos de valores.

4.1.3. Análisis de los Principales Componentes

Para el análisis de los principales componentes se debe contar con las siguientes funciones:

- PCA
 - Matriz de Covarianza
 - Covarianza
 - Media
 - Método Jacobi
 - Producto entre matrices
 - Transpuesta de una matriz

4.1.3.1. Función PCA

La función PCA debe encargarse de obtener los principales componentes de la matriz de los valores que componen las imágenes que se le presentan para su procesamiento.

Esta función se apoya en dos funciones que son la de obtener la matriz de covarianza y la función que implementa el método Jacobi para la obtención de los valores y vectores propios de la matriz de covarianza.

La función PCA debe recibir de las funciones en las que se apoya los valores y vectores propios correspondientes a cada imagen, y con estos valores y vectores debe armar el vector característico de la imagen, el cual esta compuesto por los dos vectores principales de la matriz obtenida en la función del método Jacobi. Estos dos vectores principales son los correspondientes a las dos columnas de los vectores característicos relacionadas con los dos mayores valores de la matriz de valores característicos.

4.1.3.2. Función de obtención de la matriz de covarianza

Esta función se debe encargar de procesar la matriz formada por los valores de los píxeles que representan la imagen del rostro que se desea analizar, con el fin de encontrar la matriz de covarianza correspondiente a ésta matriz.

Esta función se debe apoyar en la funciones de obtención de la covarianza que se encargara de recibir un par de vectores de la matriz original de la imagen y los

convertirá en un solo valor que será la covarianza correspondiente a estos dos vectores. La función de covarianza se apoyara en la función de obtención de la media, que se encargara de obtener el promedio de los valores correspondientes a cada vector que se estará procesando en la función de obtención de la covarianza.

4.1.3.3. Función del método Jacobi

Esta función se debe encargar de implementar el método Jacobi, que es un método iterativo para el cálculo de valores y vectores característicos. Esta función se apoyará en la función de producto de matrices y la función de transpuesta de una matriz.

4.1.3.3.1. Función de producto entre matrices

Esta función debe encargarse de recibir dos matrices y realizar el proceso del producto entre dos matrices y retornar la matriz obtenida de este producto.

4.1.3.3.2. Función de transpuesta de una matriz

Esta función debe recibir una matriz y encargarse de convertir los vectores columna de esta matriz en los vectores fila de la matriz que será el resultado de esta función.

4.1.4. Entrenamiento de la Red Neuronal Artificial

Una vez obtenidos los valores característicos de cada una de las imágenes se debe proceder a realizar el entrenamiento de la red neuronal artificial, que debe recibir como parámetros de entrada los valores característicos de las imágenes y como parámetros de salida recibirá la matriz de la imagen representativa de cada individuo.

4.1.5. Clasificación de los rostros

Una vez entrenada la red neuronal artificial el prototipo debe estar en capacidad de procesar individualmente cada una de las imágenes con el fin de extraer sus valores característicos y presentarlos como valores de entrada a la red neuronal artificial para su clasificación.

4.2. Actores

Los actores del sistema sirven para identificar las interacciones que tiene el sistema con el exterior, en concreto, un actor representa [118]:

- Un rol que un usuario puede jugar con respecto al sistema.

- Una entidad, como otro sistema o base de datos que resida fuera del sistema.

Los actores que intervienen en el sistema son:

4.2.1. Usuario

Es la persona encargada de la administración y configuración del sistema. Sus responsabilidades incluyen:

- Capturar las imágenes desde una cámara digital con el fin de formar la base de datos de las imágenes de los rostros.
- Captura las imágenes desde una cámara digital con el fin de hacer el reconocimiento del rostro que se encuentra en dicha imagen.
- Levanta una imagen de un rostro desde un archivo, con el fin de hacer el reconocimiento individual de esta imagen.
- Clasificar los rostros.
- Hallar los principales componentes de las imágenes de los rostros con el fin de entrenar la red neuronal por medio del simulador.

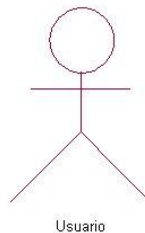


Imagen 3.1 Actor: usuario

4.2.2. Dispositivo de Captura de Imágenes

Envía las imágenes al prototipo y el usuario se encarga de seleccionar el cuadro de imagen que se ajusta a sus necesidades.

4.2.3. Simulador de Redes Neuronales

Se encarga de recibir los parámetros de entrada y de salida, con el fin de hacer un entrenamiento de la red neuronal artificial, una vez entrenada, el simulador provee la biblioteca de clases necesaria para realizar la clasificación de los rostros desde el prototipo de la aplicación.

4.3. Descripción Funcional del Sistema

Para la descripción funcional del sistema se emplean casos de uso con sus correspondientes flujos de eventos. Un Caso de Uso es una secuencia de acciones que un actor realiza dentro del sistema para alcanzar una meta particular [119]. Un flujo de eventos es la descripción de los sucesos y eventos que son necesarios para obtener el comportamiento deseado del sistema.

El sistema del prototipo de Reconocimiento de Rostros consta de los tres actores mencionados. La manera general en que interactúan con el sistema se muestra en la imagen 3.2 En esta imagen se pueden observar los siete casos de uso y las interacciones entre ellos y los actores del sistema.

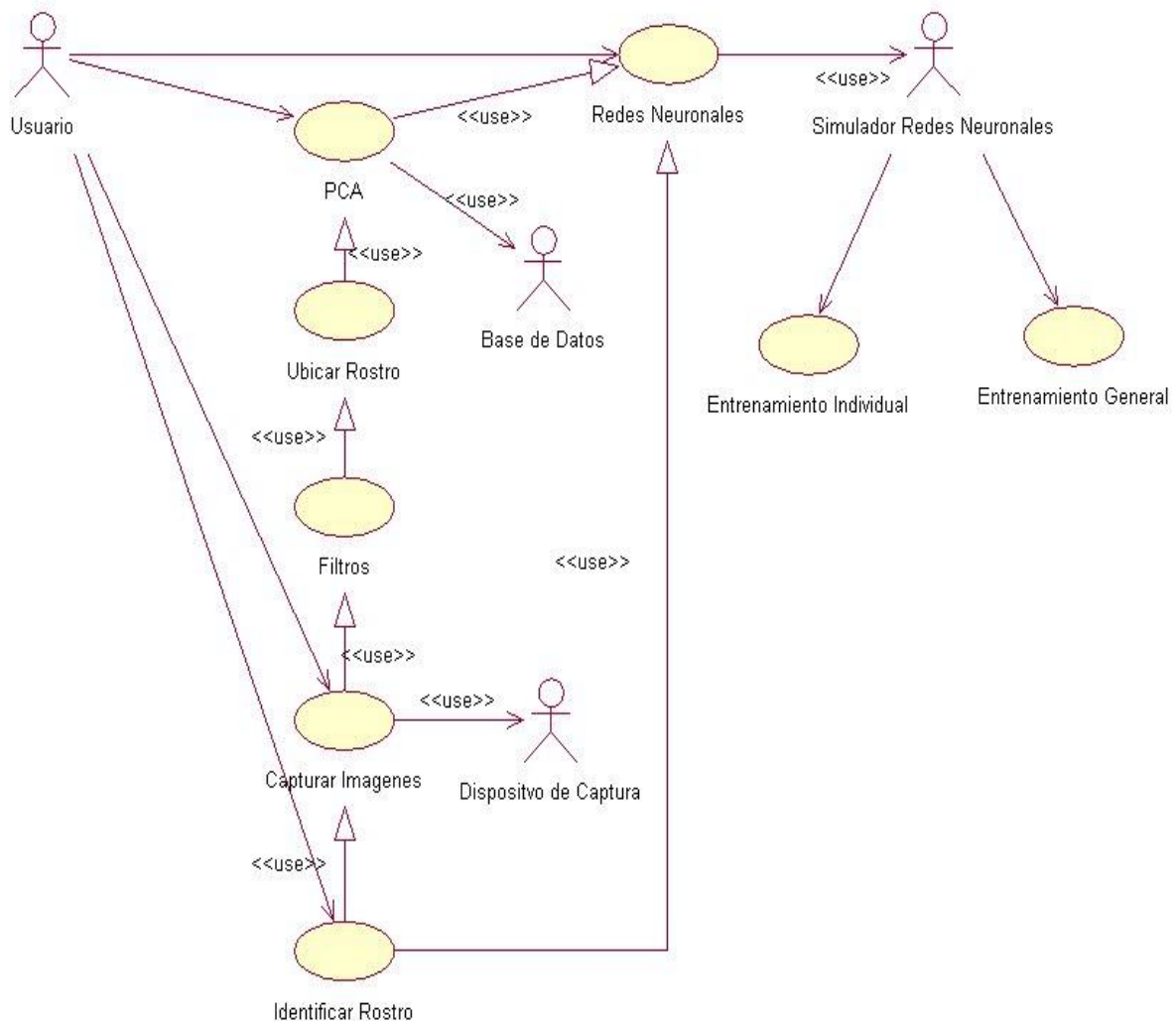


Imagen 3.2 Diagrama de Casos de uso en su vista principal

Los casos de uso son: PCA, Captura de imágenes, Identificación del Rostro, Filtros. A continuación se presenta una descripción de los casos de uso.

4.3.1. Análisis de Principales Componentes (PCA)

Este caso de uso es iniciado por el usuario. Ofrece al usuario la capacidad de seleccionar que tipo proceso de análisis de principales componentes desea realizar, análisis para un entrenamiento individual o para un entrenamiento general.

Caso de Uso	Análisis de Principales Componentes (<i>PCA</i>)
Precondiciones	Antes de que este caso de uso comience su ejecución es necesario contar con la base de datos que contenga las imágenes necesarias para hacer el entrenamiento inicial de la red neuronal artificial.
Flujo Principal	<p>Este caso de uso comienza una vez el usuario haya seleccionado que tipo de proceso de análisis de principales componentes desea realizar. El usuario puede seleccionar dos tipos de procesos, generar los valores característicos de las imágenes para un entrenamiento general o para un entrenamiento particular.</p> <p>Si la actividad seleccionada es la de procesar las imágenes para un ENTRENAMIENTO GENERAL, es llevado a cabo el subflujo Sub1.</p> <p>Si la actividad seleccionada es la de procesar las imágenes para un ENTRENAMIENTO PARTICULAR, es llevado a cabo el subflujo Sub2.</p> <p>Si la actividad seleccionada es la de procesar una imagen individual para su CLASIFICACION dentro de la red neuronal artificial, es llevado a cabo el Sub3.</p> <p>Si se ha realizado la actividad solicitada por el usuario, el caso de uso termina.</p>
Subflujos	<p>Sub1: Entrenamiento General</p> <p>Este flujo debe presentarse al iniciar por primera vez el sistema, y también se presenta posteriormente en el transcurso del uso del prototipo.</p> <p>El usuario debe seleccionar que el entrenamiento se realizara desde el simulador.</p> <p>Sub2: Entrenamiento Particular</p> <p>Este flujo puede presentarse cuando el usuario desee</p>

	<p>entrenar la red con un individuo en particular, y para esto debe seleccionar el individuo al que desea procesar con el fin de obtener sus valores característicos.</p> <p>Sub3: Clasificación</p> <p>Este flujo puede presentarse cuando el usuario ha seleccionado la captura de una imagen y lo que desea hacer con ella es clasificarla con la red neuronal artificial.</p>
--	---

Tabla 4.1 Caso de Uso PCA

4.3.2. Captura de Imágenes

Este caso de uso es iniciado por el usuario. Ofrece al usuario la capacidad de obtener imágenes de rostros con dos propósitos, el primero de ellos es el de ampliar la base de datos con la que se entrenará la red neuronal, y la otra es la de obtener un imagen de un rostro con el fin de clasificarlo con la red neuronal artificial ya entrenada.

Caso de Uso	Captura de Imágenes
Precondiciones	Antes de iniciar este caso de uso, es necesario haber conectado la cámara digital, y que se haya iniciado la captura de video desde este dispositivo.
Flujo Principal	<p>Este caso de uso inicia inmediatamente después de la ejecución del sistema, el usuario será capaz de efectuar diferentes acciones referentes a la captura de la imagen, como son: Obtener imágenes con el fin de ampliar la base de datos de los rostros y obtener la imagen de un rostro con el fin de su clasificación.</p> <p>Si la actividad seleccionada es la de Obtener imágenes para ampliar la base de datos es llevado a cabo el subflujo Sub4.</p> <p>Si la actividad seleccionada es la de obtener la imagen de un rostro para su clasificación es llevado a cabo el subflujo Sub5.</p>
Subflujos	<p>Sub4: Obtener Imágenes para ampliar la base de datos, este flujo a su vez hace uso de otros flujos, como son: Creación de un nuevo directorio (SubAd1), Filtro de ubicación del rostro dentro de la imagen (SubAd2), Guarda la nueva imagen (SubAd3).</p> <p>Sub5: El usuario puede presentar ante la cámara un individuo al que se desea verificar si se encuentra dentro del entrenamiento de la red neuronal y una vez tomada la imagen, el sistema hace uso de los siguiente flujos: Filtro de Ubicación</p>

	del Rostro (SubAd2), Análisis de principales componentes para su clasificación Sub3.
Flujos Adicionales	<p>SubAd1: Creación de un nuevo directorio</p> <p>Este flujo se inicia cuando el usuario desea ampliar la base de datos de los rostros con los cuales desea hacer el entrenamiento de la red neuronal, entonces se debe crear una nueva carpeta en donde se ubicaran las nuevas imágenes de los rostros del nuevo individuo.</p> <p>SubAd2: Filtro de ubicación del rostro dentro de la Imagen</p> <p>Este flujo se inicia automáticamente cuando el usuario ha seleccionado tomar una imagen de un individuo por medio de la cámara digital con el fin de realizar una identificación o clasificación de su rostro, también se inicia cuando el usuario desea ampliar la base de datos de los rostros.</p> <p>SubAd3: Guardar la nueva imagen</p> <p>Este flujo se dispara automáticamente cuando el usuario captura la imagen desde la cámara digital y pasa por el (SubAd2) y su objetivo es guardar la imagen ya filtrada en el directorio respectivo.</p>

Tabla 4.2 Caso de Uso Captura de Imágenes

4.3.3. Identificación de Rostros

Este caso de uso es iniciado por el usuario. Otorga la capacidad de clasificar un rostro de un individuo determinado haciendo uso de la red neuronal artificial ya entrenada.

Caso de Uso	Identificación de Rostros
Precondiciones	Antes de iniciar este caso de uso se debe tener la imagen del rostro que se desea clasificar, y la red neuronal artificial debe haberse entrenado previamente.
Flujo Principal	<p>Este caso de uso es iniciado por el usuario cuando decide tomar una imagen, ya sea desde la cámara digital o desde un archivo, con el fin de clasificarla con la red neuronal artificial.</p> <p>Una vez iniciado este flujo se inicia el caso de uso filtros para normalizar la imagen y el de PCA para obtener los datos característicos de la imagen que se desea clasificar.</p>

Tabla 4.3 Caso de Uso Identificación de Rostros

4.3.4. Filtros

Este caso de uso es iniciado automáticamente por el sistema cuando el usuario desea clasificar un rostro, ampliar la base de datos con nuevas imágenes de rostros y hacer un análisis de los principales componentes de las imágenes.

Caso de Uso	Filtros
Precondiciones	Antes de iniciar este caso de uso se debe tener la imagen a la cual se van a aplicar los filtros, ya sea desde un archivo o desde la cámara digital.
Flujo Principal	Este caso de uso comienza inmediatamente después de haberse obtenido una imagen desde la cámara digital, o desde un archivo. Si la imagen es tomada desde la cámara digital los flujos que se activaran son los siguientes: FILTRO DE UBICACIÓN DEL ROSTRO (Sub6), FILTRO DE NORMALIZACIÓN DE INTENSIDAD DE LUZ (Sub7), FILTRO DE NORMALIZACIÓN DE TAMAÑO (Sub8).
Subflujos	Sub6: Filtro de ubicación del rostro dentro de la imagen Una vez obtenida la imagen desde la cámara digital, el flujo que se inicia es el de ubicar dentro de ésta imagen el rostro del individuo. Sub7: Filtro de Normalización de intensidad de luz de la imagen Este flujo se iniciará en el caso de que la imagen necesite ser aclarado u oscurecida con el fin de obtener una mejor imagen y que todas las imágenes de la base de datos estén dentro de un mismo rango de intensidad luminica. Sub8: Filtro de Normalización de Tamaño de la imagen del rostro Este flujo se inicia en el momento en el que ya se tiene el rostro de la imagen obtenida desde la cámara digital, aquí se estandarizan todas las imágenes de los rostros a un mismo tamaño.

Tabla 4.4 Caso de Uso Filtros

CAPÍTULO 5: DISEÑO DEL SISTEMA

5. Diseño del Sistema

En este capítulo se presenta el diseño del sistema. Empieza describiendo la arquitectura del sistema para el reconocimiento de rostros, los módulos que lo componen, el diagrama de clases con una breve descripción de cada una de las clases, y los diagramas de secuencias que exponen las interacciones entre los diferentes componentes. Finalmente se presentan los algoritmos empleados en el procesamiento de las imágenes.

5.1. Arquitectura

La arquitectura del sistema es fundamental en la organización del sistema visto como un todo. Incluye elementos estáticos y dinámicos, la manera cómo esos elementos trabajan en conjunto y el estilo general del sistema [118].

Es también importante porque da una idea clara de los componentes lógicos que se persiguieron al diseñar este sistema. La imagen 5.1 muestra el modelo arquitectónico de prototipo de Reconocimiento de Rostros desarrollado en esta tesis.

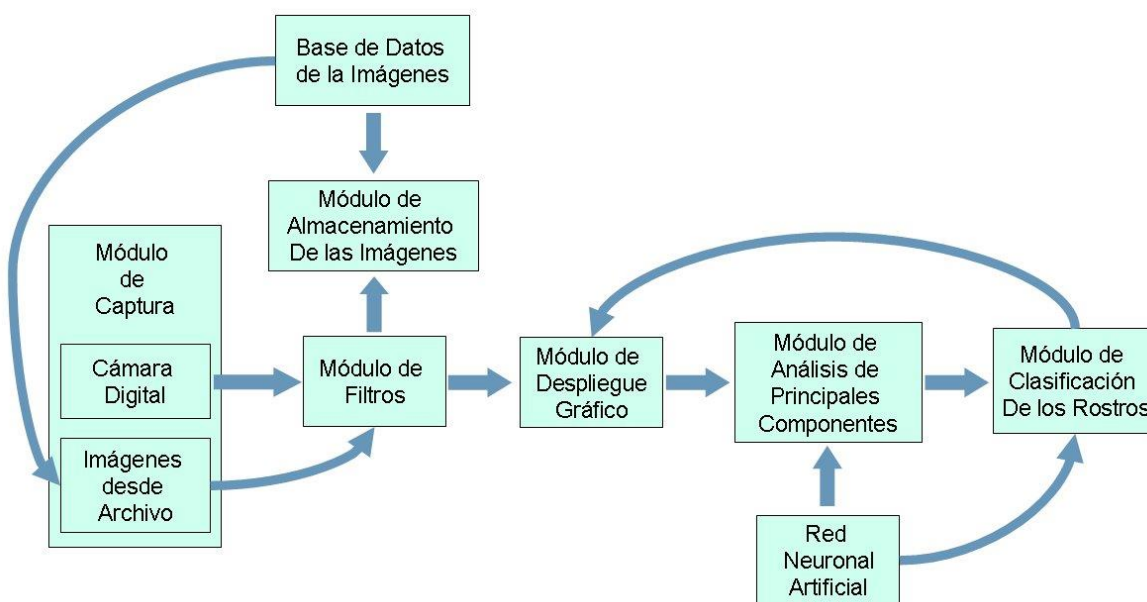


Imagen 5.1 Arquitectura del Sistema

Es interesante observar que esta arquitectura puede ser utilizada para otros sistemas de Reconocimiento de rostros; cada uno de los módulos lógicos que la conforman pueden ser adaptados a los requerimientos particulares de diferentes aplicaciones de reconocimiento de rostros, incluso es posible agregar o quitar

módulos sin modificar los demás. A continuación se describen las responsabilidades de cada uno de los módulos que componen la arquitectura.

5.2. Módulos

Los módulos que componen la arquitectura del sistema de reconocimiento de rostros son:

5.2.1. Módulo de Captura

El módulo de captura tiene la responsabilidad de buscar un dispositivo físico de captura de imágenes digitales (instalado en el computador), y la de ubicar la base de datos que contiene las imágenes de los rostros que se desean procesar. Además provee funcionalidades para establecer:

1. Una guía para la ubicación del rostro en la imagen.
2. La resolución de las imágenes capturadas desde la cámara digital.
3. El tamaño de las imágenes capturadas desde la cámara digital.
4. Un sistema de búsqueda de imágenes de rostros dentro de la base de datos.

La salida de este flujo es una imagen que contiene un rostro en representación digital, es decir, los píxeles de la imagen son ordenados en matrices, la posición del píxel en la matriz corresponde a la posición del píxel en la imagen. El valor del píxel se interpreta de acuerdo al espacio de color seleccionado.

5.2.2. Módulo de Almacenamiento de la Imágenes

Este módulo se encarga de crear los nuevos directorios dentro de la base de datos de imágenes ya existentes, con el fin de ubicar cada individuo en un directorio diferente. Al mismo tiempo el módulo guarda la imagen digital en formato *jpg* dentro del directorio correspondiente al individuo al cual se le están tomando las imágenes.

5.2.3. Módulo de Filtros

Este módulo es el encargado de preprocesar la imagen digital obtenida desde la cámara. El preprocesamiento aplicado a las imágenes es con el fin de mejorar la calidad de la imagen, reducir su tamaño y de eliminar la información contenida en la imagen que no es relevante o que no pertenece al rostro. Los filtros contenidos en este módulo son:

1. Filtro de Ubicación del rostro.
2. Filtro de Oscurecer la imagen.
3. Filtro de Aclarar.

4. Filtro de Ampliación.
5. Filtro de Reducción.
6. Filtro de Binarización.
7. Filtro de Ubicación de Contornos.

La salida de este módulo es una imagen preprocesada y lista para ser utilizada en los módulos siguientes con los cuales tiene conexión.

5.2.4. Módulo de Despliegue gráfico

Este módulo es el encargado de presentarle al usuario toda la interfaz grafica necesaria para pueda interactuar con el sistema.

5.2.5. Módulo de Análisis de Principales Componentes

Este modulo es el encargado de procesar la imagen digital del rostro preprocesada con el fin de extraer de ella sus principales características.

Este modulo contiene las siguientes funciones que son necesarias para el proceso de extracción de los valores característicos de las imágenes de los rostros:

1. Función para hallar la matriz de covarianza.
2. Función para hallar los valores y vectores propios de la matriz de covarianza.
3. Función para hallar el vector de los valores característicos.

La salida de este modulo son los vectores con los valores característicos de las imágenes de los rostros, tanto para la clasificación de los rostros como para el entrenamiento de la red neuronal artificial.

5.2.6. Módulo de clasificación de los rostros

Este modulo es el encargado de recibir los valores característicos de la imagen de un rostro y procesarlo con la red neuronal artificial en busca de su clasificación.

La salida de este modulo es la imagen devuelta por la red neuronal artificial.

5.3. Diagramas de Clases

El diagrama de clases muestra las clases y sus relaciones de asociación. Una clase es una colección de objetos que comparten una misma característica. El estereotipo de las clases puede ser entidad, control o interface [118].

Las clases entidad representan objetos en el mundo real. Contienen datos y operaciones que describen al objeto. Las clases de interface son utilizadas para

manejar la comunicación entre el sistema y entidades externas como los usuarios, operadores y otros sistemas. Las clases de control son creadas para describir comportamientos que no encajen fácilmente en clases de entidad o de interface. La imagen 5.1 muestra el diagrama de clases del prototipo de reconocimiento de rostros.

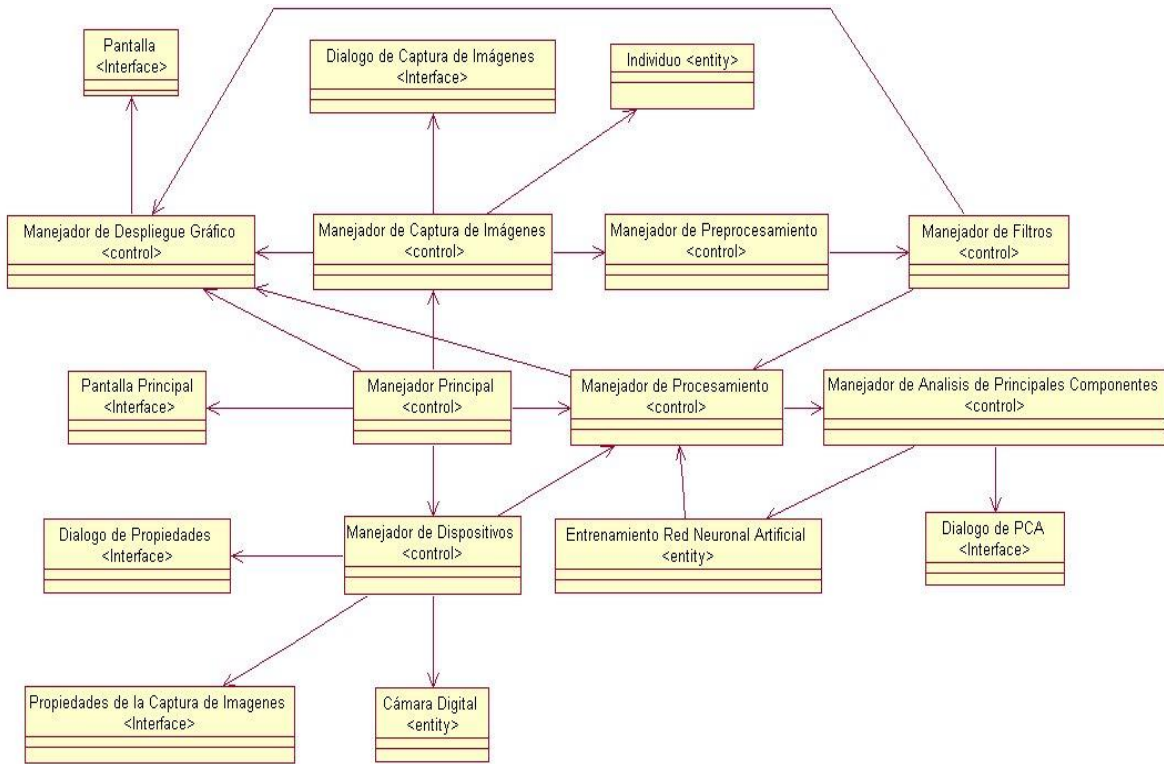


Imagen 5.1 Diagrama de Clases

Como se puede observar en el diagrama de clases, los manejadores son los encargados de realizar tareas que involucran a varias clases y constituyen la espina dorsal del sistema. A continuación se describen las responsabilidades de cada clase.

5.4. Diagrama de Secuencias

La documentación relacionada con los diagramas de secuencia y la explicación de los mismos se encuentra relacionada en el capítulo dos (2) de los anexos.

5.5. Algoritmos

En esta sección se describe en detalle los algoritmos de cada una de las etapas del reconocimiento de rostros. Como se ha descrito anteriormente las etapas son: captura, preprocesamiento, análisis de Principales componentes, entrenamiento y clasificación.

Todas las etapas, salvo la captura, el entrenamiento presentan algoritmos que transforman la imagen en información útil. A continuación se explican de manera detallada los algoritmos empleados. Esta explicación servirá para aterrizar muchos de los conceptos desarrollados en el capítulo 2.

5.5.1 Algoritmos de Preprocesamiento

Dentro de la etapa de preprocesamiento nos encontramos con algunas fases que son necesarias para que pueda ser llevada a cabo y estas fases son:

- Extracción de la Imagen del rostro.
- Normalización

5.5.1.1 Algoritmo de Extracción del Rostro

El objetivo de la ubicación y extracción de un rostro contenido dentro de una imagen más grande, es el de desechar información que no es relevante. Para el desarrollo de esta tarea existen varios métodos que pueden ser utilizados, entre ellos la clasificación mayor que se les puede hacer es en métodos dinámicos y métodos estáticos, los métodos dinámicos de extracción de rostros son los que descomponen la imagen en busca de patrones específicos, en este caso el de un rostro, y lo hace recorriendo la imagen por zonas en busca de estos patrones; los métodos estáticos son lo que hacen uso de plantillas en las cuales se conoce con anticipación la ubicación del rostro dentro de la imagen, lo que exige un ambiente controlado de captura de imágenes. La ventaja de un método con relación al otro es que el método estático es más económico en cuanto a recursos y tiempo de procesamiento. Y la desventaja del mismo es que es necesario tener un ambiente controlado para determinar con exactitud la ubicación del rostro dentro de la imagen.

En el prototipo se han implementado los dos algoritmos con el fin de hacer uso de cada uno de ellos en el momento en que las condiciones son favorables para cada método.

En el algoritmo del método estático deben establecerse unos límites dentro de los cuales se enmarca la imagen del rostro dentro de la imagen mayor, y esto se puede hacer únicamente porque tenemos una guía ubicada en la cámara digital en el momento de la captura de la imagen.

Una vez establecidos los límites que enmarcan el rostro, se debe proceder a recorrer la imagen para extraer el rostro y ubicarlo en la imagen definitiva que contendrá el rostro contenido en la imagen.

```

for(int x=limSuperior; x <limInferior; x++ )
{
    for(int y = limDerecho; y<limIzquierdo; y++)
    {
        x1 = x-limSuperior;
        y1 = y-limDerecho;
        pixel = imgOrigen.GetPixel(y,x);
        B.SetPixel(y1,x1,pixel);
    }
}

```

Imagen 5.11 Algoritmo de extracción del Rostro

En el método Dinámico lo primero que se realiza es preprocesar la imagen con dos filtros que son indispensables para poder realizar la busque del rostro dentro de la imagen, que son: el filtro de Binarización de la imagen y el filtro de ubicación de contornos dentro de la imagen. Una vez se tiene la imagen de los contornos de la imagen original se procede al calculo de los límites, los cuales son establecidos automáticamente haciendo un recorrido de la imagen en busca de formas generales similares a las de un rostro, una vez establecidos los limites necesarios para enmarcar el rostro contenido en la imagen original, se procede a la extracción del rostro y a su ubicación en la imagen definitiva que lo contendrá.

5.5.1.2. Algoritmos de Normalización

Los métodos de normalización utilizados en el prototipo son los relacionados con el filtro de modificación de intensidad lumínica, tamaño y el de Binarización de la imagen.

El filtro de modificación de la intensidad lumínica se encarga de oscurecer o aclarar la imagen dependiendo de cual sea la necesidad de la imagen. Lo primero que se hace dentro del algoritmo de este filtro es establecer el ancho y el alto de la imagen que se va a procesar, posteriormente se procede a declarar dos bucles anidados con el fin de recorrer toda la matriz correspondiente a la imagen, dentro de estos bucles anidados el primer paso es el de capturar el píxel correspondiente a las coordenadas x e y en las cuales se encuentra cada uno de los bucles, posteriormente se calcula el nuevo valor del píxel de la siguiente manera: lo primero que se debe obtener es el valor en escala de grises del píxel, lo cual se obtiene extrayendo el valor *RGB* del píxel y dividiéndolo entre tres, este valor lo dividimos entre 255 que es el valor máximo de la escala de grises, una vez obtenido este valor se procede a elevarlo al cuadrado y el valor obtenido de esta operación lo multiplicamos por 255, así obtenemos un valor de píxel con el incremento necesario para obscurecer la imagen.

En el filtro de Modificación de tamaño, procesamos la imagen con el fin de ampliar o disminuir la cantidad de píxeles posee la imagen tanto del ancho como del alto. Para llevar a cabo este proceso nos apoyamos en una función del lenguaje de programación que nos permite hacer la modificación del tamaño recibiendo como parámetros el alto y el ancho que se desea darle a la imagen original.

```

for(int y=0;y<alto;y++)
{
    for(int x=0;x<ancho;x++)
    {
        pixel=imgOrigen.GetPixel(x,y); //Guardo el color del pixel
        g = (int) (255.0 * Math.Pow (((int) (pixel.R + pixel.B + pixel.G)/3)/ 255.0, 2));
        p=Color.FromArgb(g,g,g);

        imgDestino.SetPixel(x,y,p);
    }
}

```

Imagen 5.12 Algoritmo Filtro para Obscurecer la Imagen

También recibe una variable en la cual almacena el método de devolución de llamada por cancelación anticipada de ejecución y otra que es el tipo específico de plataforma que se utiliza para representar un puntero o un identificador.

```

{
    //metodo de devolución de llamada por cancelación anticipada de ejecución
    Image.GetThumbnailImageAbort callback = new Image.GetThumbnailImageAbort(ThumbnailCallback);
    Image imgReducida = imgOriginal.GetThumbnailImage(alto, ancho, callback, IntPtr.Zero);
    return imgReducida;
}

```

Imagen 5.13 Algoritmo Filtro de Ampliación y Reducción de la Imagen

En el algoritmo de filtro de Binarización lo primero que hacemos es establecer los valores del alto y el ancho de la imagen que se va a procesar, una vez obtenidos estos valores, se establece un valor umbral para realizar la Binarización.

```

/*en estos ciclos binarizamos la imagen original reducida
 * y alimentamos la matriz que se usara para el proceso
 * de obtencion de eigenvectores y eigenvalores.*/
for(int y=0;y<alto;y++)
{
    for(int x=0;x<ancho;x++)
    {
        pixel = imgOrigen.GetPixel(x,y);
        g = (int) (pixel.R+pixel.G+pixel.B)/3;
        if(g>=Umbral)
        {
            pixelBinarizado = Color.FromArgb(255,255,255);
            matriz[x,y]= 1;
            imgDestino.SetPixel(x,y,pixelBinarizado);
        }
        else
        {
            pixelBinarizado = Color.FromArgb(0,0,0);
            matriz[x,y]= 0;
            imgDestino.SetPixel(x,y,pixelBinarizado);
        }
    }
}
}

```

Imagen 5.14 Algoritmo Filtro de Binarización

Establecidos estos valores se procedes a elaborar dos bucles anidados los cuales contendrán en su interior las siguientes operaciones: lo primero que se hace es obtener el valor del píxel correspondiente al punto en el cual se encuentra el bucle en ese momento (x, y), obtenido el píxel se extrae su valor en escala de grises de la misma forma que en la sección anterior, y este valor entero lo analizamos con el fin de determinar si esta por encima o por debajo del umbral establecido. Una vez se tome esta decisión, se le asigna a una variable el valor del píxel que corresponderá a los dos valores extremos de la escala de grises, o sea, 0 para un píxel en negro y 255 para un píxel en blanco; al mismo tiempo se va elaborando una matriz de valores enteros la cual contendrá solo dos tipos de valores 1 ó 0, que será la matriz con la cual se trabajará en los procesos subsiguientes.

5.5.2. Algoritmos de Análisis de Principales Componentes

Una vez se tiene la matriz correspondiente a los valores de la imagen binarizada, las tareas siguientes son las relacionadas con la extracción de los valores característicos correspondientes a la imagen que la matriz representa.

Para realizar el proceso del análisis de los principales componentes de las imágenes se han desarrollado algunas funciones en las cuales se apoyará el modulo principal *PCA* para el cálculo de los valores característicos de las imágenes, estas funciones son: función para el cálculo de la matriz de covarianza, función para el cálculo de la covarianza, función para el calculo de la media, función para el calculo de los valores y vectores propios (método Jacobi), función para hallar el producto de dos matrices, función para hallar la transpuesta de una matriz, función para hallar los valores característicos.

El proceso de análisis de principales componentes esta desarrollado dentro de la función *PCA*, en esta función lo primero que se hace es realizar un recorrido por la base de datos con el fin de levantar las imágenes correspondientes a cada individuo que la componen, esto se hace desarrollando dos bucles anidados para hacer el recorrido tanto de los directorios como de las imágenes contenidas en ellos. Una vez levantada la imagen se procede a normalizar la imagen se procede a hacer el llamado a las funciones que procesaran la imagen; la primera función que es llamada al interior de *PCA* es la encargada de calcular la matriz de covarianza.

La función encargada del cálculo de la matriz de covarianza tiene como objetivo principal hallar la covarianza de cada par de vectores correspondientes a los valores de la matriz original, para tal fin se recorre la matriz original obtenida de la Binarización de la imagen en dos bucles anidados y verificando que se cumplan las condiciones de una matriz de covarianza que son las mismas de una matriz simétrica.

Si se está ubicado en la diagonal principal de la matriz de covarianza, los dos vectores que se envían a la función de cálculo de la covarianza es uno solo

repetido, el cual corresponde al vector columna de la posición de la diagonal principal en la cual se encuentre el bucle (x).

Si la posición está ubicada por fuera de la diagonal principal de la matriz de covarianza, se recorre la matriz original de la imagen con el fin de obtener los dos vectores que se le envían a la función de covarianza. Estos dos vectores corresponden, el primero al vector columna de la posición actual de bucle más externo y el otro vector corresponde al vector vecino de la derecha del primer vector. Debido a que la matriz de covarianza, como ya se comentó, es una matriz simétrica, el valor de la posición en la matriz (x, y) es igual al valor de la posición (y, x), siempre y cuando no sea la diagonal principal.

```
for(int x = 0; x < alto; x++)
{
    for(int y = 0; y < alto; y++ )
    {
        if(x==y)
        {
            for(int val=0; val < alto; val++)
            {
                vector1[val] = matriz[x,val];
            }
            vector2 = vector1;
            matrizCovarianza[x,y]=Covarianza(vector1,vector2);
        }
        if(x<y)
        {
            for(int val=0; val < alto; val++)
            {
                if(x+1 >= alto)
                    break;
                vector1[val] = matriz[x,val];
                vector2[val] = matriz[x+1,val];
            }
            cov = Covarianza(vector1,vector2);
            matrizCovarianza[x,y]=cov;
            matrizCovarianza[y,x]=cov;
        }
    }
}
```

Imagen 5.15 Algoritmo Calculo de la Matriz de Covarianza

La función para el cálculo de la covarianza, utilizada en la función del cálculo de la matriz de covarianza, obtiene el valor de la media de cada uno de los dos vectores recibidos haciendo uso de la función para el cálculo de la media, y posteriormente calcula la covarianza dentro de un bucle realizando las siguientes operaciones: asigna a una variable su valor anterior adicionándole el producto del valor del primer y segundo vector en la posición x menos sus correspondientes medias dividido entre el valor de la longitud de los vectores.

```

double media1 = Media(vector1);
double media2 = Media(vector2);
covarianza = 0;
for(int x = 0; x < vector1.Length; x++)
{
    covarianza += ((vector1[x]-media1) * (vector2[x]-media2)) / (vector1.Length-1);
}

```

Imagen 5.16 Algoritmo Calculo de la covarianza

La función para el calculo de la media usada en la función anterior, lo que hace es recorrer dentro de un bucle el vector que recibe con el fin de sumar los valores de cada uno sus campos y dividirlos entre la longitud del vector.

```

//se recorre el vector y se suman sus valores
for(int x = 0; x < vector.Length; x++)
{
    media = media + vector[x];
}
//se divide el resultado anterior entre la longitud del vector
media = media/vector.Length;

```

Imagen 5.17 Algoritmo calculo de la media

La función para el cálculo de los valores y vectores característicos (método Jacobi) recibe como parámetro por valor el valor máximo de las iteraciones que se desean realizar para la aproximación en el cálculo de los valores y vectores característicos.

El método Jacobi es un método iterativo utilizado para aproximar sistemas de la forma $Ax=b$.

Dentro de la función del calculo de los valores y vectores característicos el primer paso es el de declarar las variables que se utilizaran para el calculo de estos datos. Se declaran como variables locales el valor de la tolerancia, que sirve como valor de parada dentro del bucle principal, se declaran también tres matrices, una de ellas corresponde a la matriz de covarianza, otra a una matriz de giro, y la ultima que almacenará los vectores característicos de la matriz de covarianza; y por ultimo se declara una variable llamada vectores propios la cual almacenara los valores propios de la matriz de covarianza.

Una vez declaradas las variables necesarias, el primer paso es el de recorrer la matriz de los vectores propios con un bucle simple para inicializarla ubicando el valor 1 en su diagonal principal.

En un bucle mayor se procede a realizar los cálculos de aproximación de los valores y vectores propios de la matriz de covarianza. Los pasos para este proceso son: el primer paso es el de obtener el mayor valor contenido dentro de la matriz de covarianza, y como la matriz de covarianza es simétrica, se deben analizar los valores de las casillas de la matriz que se encuentran en y por debajo de la diagonal principal.


```

//obtenemos el valor máximo contenido dentro de la matriz de covarianza
for(i=0; i<n-1; i++)
{
    for(j=i+1; j<n; j++)
    {
        if(Math.Abs(matrizOr[i,j])>maximo)
        {
            k=i;
            l=j;
            maximo=Math.Abs(matrizOr[i,j]);
        }
    }
}

```

Imagen 5.18 Algoritmo mayor valor de la matriz de covarianza

El segundo paso es el de calcular la tolerancia máxima permitida que se hace dentro de un bucle asignando a una variable el valor anterior de dicha variable mas el valor al cuadrado del valor contenido en casilla de la matriz de covarianza en la posición actual del bucle de su diagonal principal; una vez obtenido el valor de esta variable, se le asigna a la variable de tolerancia el resultado de dividir raíz cuadrada del valor de la variable obtenida anteriormente entre el producto del número de filas por 10000.

```

//Calculo de la tolerancia
sumsq=0.0;
for(i=0; i<n; i++)
{
    sumsq+=matrizOr[i,i]*matrizOr[i,i];
}
//se asigna la tolerancia
tolerancia=0.0001*Math.Sqrt(sumsq)/n;

```

Imagen 5.19 Algoritmo calculo de la Tolerancia

El tercer paso es el de verificar si el mayor valor de la matriz de covarianza obtenido en el primer paso es menor que la tolerancia, si es así se sale del bucle principal, de lo contrario se sigue con las iteraciones.

El cuarto paso es el de calcular la matriz ortogonal de la matriz de giro, que inicialmente es la matriz unidad, para inicializar la matriz de giro como una matriz unidad se anidan dos bucles con el fin de recorrer la matriz de giro y ubicar en las posiciones correspondientes a la diagonal principal un 1 (uno) y en las posiciones diferentes a la diagonal principal un 0 (cero).

Para continuar con la obtención de la matriz ortogonal de la matriz de giro, asignamos a una variable (var1) la diferencia entre los valores de las posiciones (k, l) de la matriz de covarianza, donde k y l son las coordenadas del mayor valor de la matriz de covarianza, una vez obtenida esta variable, se procede a verificar si el valor absoluto de var1 es menor que cero, si es así, se asigna a dos variables (var2, var3) el resultado del seno de $\pi/4$, de lo contrario se asigna a una variable (var4) el doble del mayor valor de la matriz de covarianza.

```

//calcula la matriz ortogonal de la matriz de giro
//inicialmente es la matriz unidad
for(i=0; i<n; i++)
{
    for(j=0; j<n; j++)
    {
        if(i == j)
        {
            matrizGiro[i,i]=1.0;
        }
        else
        {
            matrizGiro[i,j]=0.0;
        }
    }
}

```

Imagen 5.20 Algoritmo Inicialización matriz de giro

A una variable (var5) se le asigna la raíz cuadrada del cuadrado de var1 más el cuadrado de var4, a var2 se le asigna la raíz cuadrada de var5 más var1 dividida entre el doble de var5 y por ultimo se le asigna a var3 el producto de la función signo por la raíz cuadrada de var5 menos var1 dividida entre el doble de var5 .

La función signo se encarga de devolver un 1 si el valor de la división var1 y var4 es mayor que cero y -1 si el resultado de la división es menor que cero. Una vez se han obtenido todas estas variables se procede a asignar a la matriz de giro en la casilla de la posición (k, k) y (l, l) el valor de var2, en la posición (k, l) el valor de var3 y en la posición (l, k) el valor de -var3.

```

//Asignacion de los angulos de giro
y=matrizOr[k,k]-matrizOr[l,l];
if(Math.Abs(y)<CERO)
{
    c=s=Math.Sin(Math.PI/4);
}
else
{
    x=2*matrizOr[k,l];
    z=Math.Sqrt(x*x+y*y);
    c=Math.Sqrt((z+y)/(2*z));
    s=Signo(x/y)*Math.Sqrt((z-y)/(2*z));
}
matrizGiro[k,k]=c;
matrizGiro[l,l]=c;
matrizGiro[k,l]=s;
matrizGiro[l,k]=-s;

```

Imagen 5.21 Algoritmo Asignación de ángulos de giro

Hasta este punto se ha obtenido la matriz de giro con todos sus componentes necesarios para iniciar las operaciones con las matrices en busca de los valores y vectores característicos.

Estas operaciones son: a la matriz original se le asigna el producto de la matriz de giro por el producto de la matriz original por la transpuesta de la matriz de giro. Y a

la matriz de los valores propios se le asigna el resultado del producto de la matriz de los valores propios por la transpuesta de la matriz de giro.

```
matrizOr = ProductoMatrices(matrizGiro, ProductoMatrices(matrizOr, MatrizTranspuesta(matrizGiro)));  
matrizVP = ProductoMatrices(matrizVP, MatrizTranspuesta(matrizGiro));
```

Imagen 5.22 Algoritmo operaciones con matrices

Todas las operaciones anteriormente descritas se hacen dentro del bucle principal que tiene como valores de parada la tolerancia y el valor máximo de iteraciones. Si la salida del bucle es provocada por la tolerancia, se garantiza que los valores y vectores propios de la matriz de covarianza fueron aproximados satisfactoriamente, pero si la salida del bucle fue provocada por la llegada al número máximo de iteraciones, esto significa que no fue posible el cálculo de estos valores.

5.5.3. Algoritmo de Clasificación

Para realizar el proceso de clasificación se debe procesar la imagen apoyándose en la función de Análisis de los Principales Componentes (*PCA*), y una vez obtenidos los valores característicos de la imagen se debe acudir a una función que provee la librería de clases encargada del manejo de la red neuronal artificial. El resultado es almacenado en un objeto que posteriormente será recorrido con el fin de construir la imagen devuelta por la Red Neuronal Artificial.

```
//extraccion de los valores caracteristicos de la imagen  
long time = PCA(contador, contador1, carpeta, llamado);  
  
//Llamado a una funcion de clasificacion de la RNA.  
responseData = aNN.GetNetworkResponse(path, best_Weights, inputData);  
  
//se muestra la imagen resultado de la clasificacion con la RNA.  
this.rostroResultado.Image = this.AmpliarImagen(CargarImagen(), 270, 280);
```

Imagen 5.23 Algoritmo de Clasificación

En la función para recorrer el objeto y construir la imagen a partir de los valores devueltos por la Red Neuronal Artificial lo primero que se hace es convertir el objeto en una matriz que contendrá los valores devueltos por la red neurona artificial, una vez obtenida esta matriz se procede a recorrer sus casillas con el uso de dos bucles anidados, se verifica si no se ha llegado al máximo de píxeles de la imagen, se verifica también que el valor que se le asignará a los píxeles este dentro del rango de la escala de grises (0, 255) y por ultimo se asigna el valor actual de la casilla de la matriz en la misma posición del píxel de la imagen.

```

double[,] data=(double[,])this.responseData;
//se recorre el objeto con el fin de obtener los valores devueltos por la RNA
//y asignarselos a una imagen como los valor de sus pixeles
for (int x= 0; x < 25; x++)
{
    for (int y= 0; y < 25; y++)
    {
        if(cont == 624) break ;
        temp = p;
        p = (int)data[0,cont];
        if(p > 255) p=temp;
        if(p < 0) p=temp;
        imagen.SetPixel(x,y,Color.FromArgb(p,p,p));
        cont ++;
    }
}
}

```

Imagen 5.24 Algoritmo construcción de imagen a partir de un objeto.

CAPITULO 6: IMPLEMENTACIÓN DEL SISTEMA

6. IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se expone el prototipo de Reconocimiento de Rostros desde una perspectiva operativa. Se presentan las herramientas de software y hardware utilizadas en su desarrollo, la implementación del prototipo funcional y los resultados que se obtuvieron.

6.1. Herramientas

En esta sección se describen las herramientas tanto de software como de hardware sobre las cuales se construyó el prototipo de reconocimiento de rostros.

Se comienza exponiendo las herramientas de hardware empleadas y se continúa con una explicación detallada de las herramientas de software.

6.1.1. Hardware

El hardware permite situar el entorno físico que se utilizó para crear el sistema, y al mismo tiempo permite entender algunas limitaciones que se presentaron en el transcurso del desarrollo del proyecto.

Una de las características de este proyecto es que no tiene un costo muy elevado, y algo que contribuyó a esto fue el uso de hardware no especializado, es decir se utilizaron equipos que se pueden conseguir prácticamente en cualquier tienda de cómputo.

La plataforma sobre la que corre el prototipo de reconocimiento de rostros es un PC estándar:

Procesador	AMD Athlon(tm) XP 2800+ 2.09 Ghz.
Memoria	768 MB
Tarjeta de Video	NVIDIA RIVA TNT2 Model 64

Tabla 6.1 Características de la CPU

La cámara digital utilizada es la Creative PC Cam 350 que es una cámara genérica muy económica cuyo costo es de \$150.000 pesos colombianos que es relativamente bajo comparado con otros tipos de cámaras digitales existentes en el mercado, este tipo de dispositivo se puede utilizar gracias a que el análisis de los principales componentes se puede realizar en imágenes que son relativamente pequeñas y con un tipo de resolución baja.

6.1.2. Software

El sistema operativo sobre el cual se desarrollo el prototipo de reconocimiento de rostros fue Windows XP Profesional versión 2002 con *Service Pack 2*. El sistema fue programado utilizando Microsoft Visual C# .Net y las librerías de clases para el manejo de las redes neuronales fue programado en Microsoft Visual C++ y Microsoft Visual Basic de .Net. Las RNA necesarias para la clasificación fueron desarrolladas utilizando el simulador de *NeuroSolutions* Versión 5.00.

6.2. Implementación del Prototipo Funcional

La implementación del prototipo de reconocimiento de rostros fue dividida en tres partes. La primera fue el desarrollo de los filtros para el preprocesamiento de las imágenes. La segunda es la parte de procesamiento de la imagen, donde se realiza el proceso de análisis de los principales componentes para la extracción de los valores característicos de las imágenes. La tercera parte es la encargada del proceso de entrenamiento y clasificación con el uso de RNA.

6.2.1. Filtros de Preprocesamiento

En los filtros de preprocesamiento es donde se realizan prácticamente todas las operaciones para la extracción del rostro de una imagen y las de normalización de las imágenes de los rostros.

El filtro tiene dos líneas de entradas dependiendo del tipo de captura de la imagen y una línea de salida. En la imagen 6.1 se muestra el filtro de preprocesamiento.



Imagen 6.1 Filtros de Preprocesamiento

Cada función de filtrado recibe como parámetro una imagen y su salida es otra imagen ya procesada.

El filtro de extracción de rostros recibe como parámetro una imagen desde la cámara digital, esta imagen es asignada a un variable de tipo *Image* proporcionada por el lenguaje de programación *C#*, que es una clase base abstracta que proporciona funcionalidad para las clases descendientes *System.Drawing.Bitmap* y *System.Drawing.Image.Metafile*.

La variable del tipo *Image* es convertida en una variable de la clase *Bitmap*, la cual encapsula un mapa de bits de *GDI+*, formado por los datos de píxel de una imagen de gráficos y sus atributos. Un objeto *System.Drawing.Bitmap* es un objeto utilizado para trabajar con imágenes definidas por datos de píxel.

La salida de este filtro es una imagen del rostro que se ha extraído de la imagen que recibió como parámetro.

Los filtros de normalización, al igual que el filtro de extracción de rostros, reciben una imagen como parámetro de entrada, y su objetivo es el modificar las imágenes con el fin mejorar su calidad, generar un estándar entre ellas y dejar lista la imagen para ser procesada en las siguientes etapas del prototipo.

El principio general de todos los filtros de normalización es el de asignar la imagen que recibe como parámetro a una variable del tipo *Bitmap* para posteriormente hacer un recorrido dentro sus valores matriciales con el fin de hacer las modificaciones asignadas en cada caso del filtro utilizado.

La salida de los filtros de normalización es una imagen con las características requeridas por los procesos subsiguientes al filtro utilizado en su momento.

6.2.2. Procesamiento de la Imagen

El procesamiento de la imagen de los rostros consiste en realizar una serie de procedimientos matemáticos y estadísticos con el fin de extraer los valores característicos de esta imagen.

Esta etapa presenta al usuario una interfaz gráfica por medio de la cual puede interactuar con el proceso de extracción de los valores característicos de las imágenes. El usuario en esta interfaz puede seleccionar el análisis de un individuo en particular o el análisis de todas las imágenes de la base de datos.

Esta etapa es la responsable de conectar en el orden adecuado los filtros de preprocesamiento y el despliegue de las imágenes en la pantalla de interfaz con el usuario.

La entrada a la fase principal del procesamiento es una imagen de un rostro en formato binario cuyos valores de cada píxel son asignados a su correspondiente casilla dentro de una matriz de valores enteros. En este punto es donde se aprovecha la gran ventaja con que cuenta el uso del Análisis de los Principales Componentes como método para extraer valores característicos de una imagen, que es el poder trabajar con imágenes de baja resolución y de tamaño muy pequeño. La imagen con que se trabaja en este punto es una imagen de 25x25 píxeles, que garantiza una disminución considerable de iteraciones en su procesamiento pero conserva los suficientes datos como para representar ampliamente la imagen con sus respectivos valores característicos.

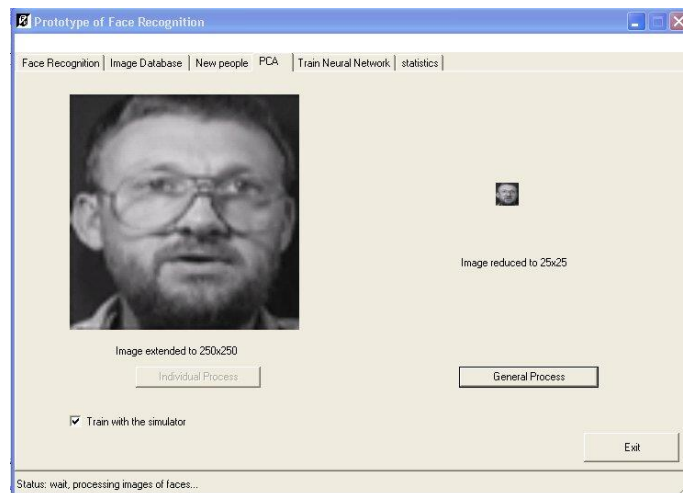


Imagen 6.2 Interfaz de Análisis de Principales Características

El primero proceso que se lleva a cabo con la matriz de la imagen del rostro es el de hallar la matriz de covarianza, esta función toma la matriz correspondiente a los valores de los píxeles de la imagen y calcula la matriz de covarianza correspondiente a esta matriz. Se declara una variable global llamada *MatrizCovarianza* que es una matriz de tipo doble del tamaño del ancho y el alto de la imagen, y se declaran dos variables de tipo doble para almacenar el par de vectores que se envían a la función encargada de calcular la covarianza.

La salida de este primer proceso es una matriz que contiene la covarianza correspondiente a la matriz de los valores de la imagen del rostro.

El segundo proceso es el encargado de analizar la matriz de covarianza y extraer de ella los valores y vectores característicos con el método Jacobi, este proceso recibe como entrada la matriz de covarianza obtenida en el proceso anterior.

La salida de este proceso es un vector con los valores propios y una matriz con los vectores característicos de la matriz de covarianza.

El proceso final es el de identificar dentro de la matriz de vectores propios los que corresponden a los vectores característicos de dicha matriz, los valores característicos están relacionados directamente con el vector de valores propios, las posiciones de los valores más altos corresponde a los primeros vectores característicos de la matriz.

La salida de este proceso es una matriz con los valores característicos correspondientes a cada una de las imágenes de rostros procesados.

La salida general de la fase del procesamiento de la imagen es un conjunto de vectores que contienen los valores característicos de los rostros, estos vectores

están agrupados por individuo y tienen indexado a ellos una matriz con los valores de los píxeles de la imagen del rostro representativo de cada individuo.

6.2.3. Entrenamiento y Clasificación con RNA

El proceso de entrenamiento de las RNA se lleva a cabo en el simulador de *NeuroSolution*, y para tal propósito se deben generar los archivos que contienen tanto las entradas como las salidas de las redes, las entradas son los valores característicos de la imagen del rostro, y las salidas son las imágenes de referencias que se cargan por cada individuo. Estos datos son entregados por la fase de procesamiento de la imagen vista en la sección anterior.

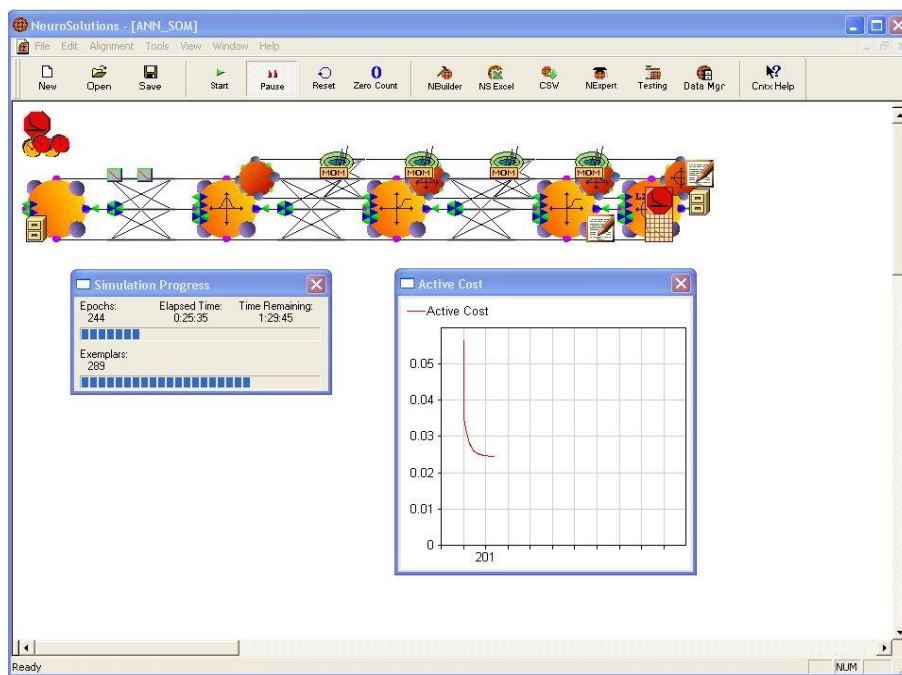


Imagen 6.3 Fase de Entrenamiento de la RNA con *NeuroSolution*

El resultado de este proceso es una librería de clases llamada *ManagerANN* que es utilizada para el manejo y control de las RNA, así como las RNA entrenadas y listas para realizar la clasificación de los rostros.

Para iniciar al proceso de clasificación el usuario cuenta con una interfaz por medio de la cual escoge de donde capturará la imagen que desea clasificar y cual será la red neuronal que utilizará para la clasificación. En la imagen 6.4 se puede observar la interfaz con la cuenta el usuario para iniciar el proceso de clasificación.

El proceso de clasificación se hace utilizando el *ManagerANN* para controlar que red neuronal se va a utilizar para la clasificación, el primer paso para el proceso de clasificación es la declaración de las variables que contienen las rutas de almacenamiento de los datos básicos que necesita el manejador de la RNA como

son: la ruta del archivo que contiene la librería de la red neuronal que se utilizará para la clasificación (*i.e. C:/Prototype_FR/bin/NeuralNetwork/SOM/ANN_SOM.dll*) y la ruta del archivo que contiene los pesos de los enlaces de la RNA (*i.e. C:/Prototype_FR/bin/NeuralNetwork/SOM/BestWeights.nsw*).

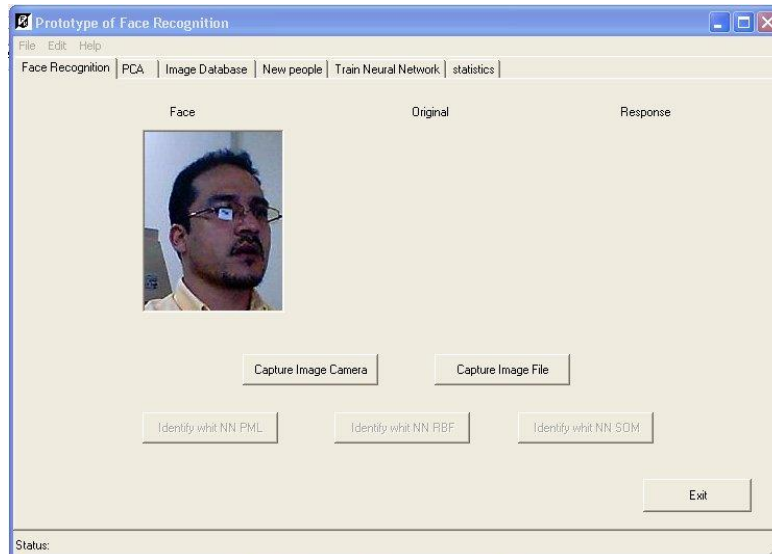


Imagen 6.4 Fase de clasificación

El segundo paso el proceso de clasificación es el de analizar la imagen con el fin de hacer la extracción de los valores característicos y estos valores característicos son enviados a la RNA con el fin de realizar la clasificación del rostro, la red neuronal en su método de clasificación recibe como parámetros las rutas de los pesos y de la librería de la red neuronal, así como los datos de entrada que son los valores característicos de la imagen. Una vez realizado el proceso de clasificación, la red neuronal retorna un objeto con los datos que representan los píxeles correspondientes a la imagen que se obtuvo de la clasificación del rostro.

El paso final del proceso de clasificación, es el de desplegar la imagen en la interfaz de usuario, esto se hace en la función de cargar imagen, la que se encarga de tomar el objeto devuelto por la red neuronal y organizar los píxeles dentro de la imagen que se mostrará al usuario.

La salida de este proceso es una imagen que contiene el rostro clasificado con la red neuronal escogida por el mismo usuario como se muestra en la imagen 6.5.

6.3. Implementación de las Redes Neuronales Artificiales

Para la implementación de las redes neuronales artificiales se utilizaron los parámetros específicos suministrados por los modelos seleccionados para el estudio. Se utilizaron 75 entradas que corresponden a los tres primeros vectores característicos de cada una de las imágenes y 625 salidas que corresponden a

una imagen de 25*25. Se describe la implementación de la Red Neuronal SOM, puesto que en este tipo de red se fundamenta el prototipo de implementación, pero las mismas características se desarrollan en la implementación de los otros dos tipos de redes neuronales de los modelos restantes, que se desarrollaron para poder hacer las pruebas pertinentes para la comprobación de las conclusiones teóricas. Los detalles de la implementación de los tres tipos de redes neuronales se encuentra en el capítulo 3 de los anexos.

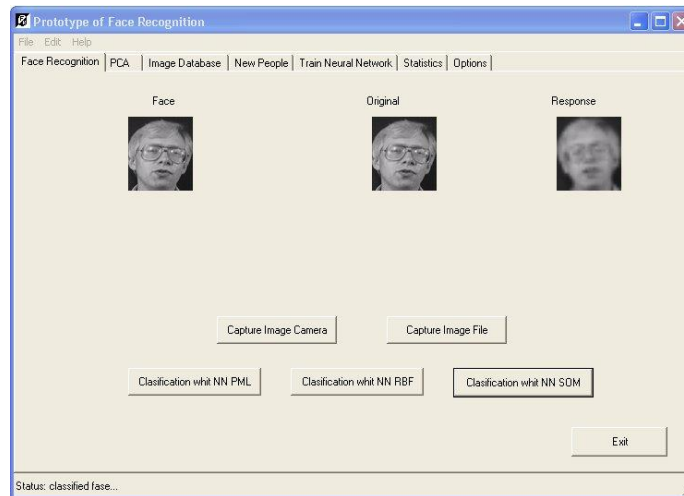


Imagen 6.5 Resultado del proceso de clasificación.

6.3.1. Red Neuronal Artificial SOM

Esta red neuronal tiene una capa de entrada, una de salida y dos capas intermedias. La función de transferencia utilizada es la Sigmoidea, la regla de aprendizaje es la de *momentum*, tamaño de paso de 0.1, máximo de iteraciones no supervisadas 1000, porcentaje de aprendizaje inicio 0.01 con una caída de 0.001. Número máximo de iteraciones para el entrenamiento supervisado 30.000, umbral de error mínimo 0.01.

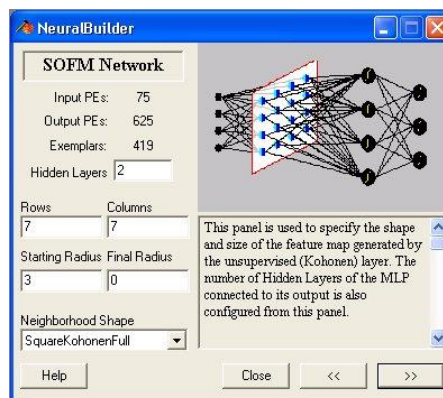


Imagen 6.6 Características RNA SOM

CAPÍTULO 7: PRUEBAS Y RESULTADOS

7. Pruebas y Resultados

En este capítulo se presentan las pruebas realizadas al prototipo de Reconocimiento de Rostros y los resultados obtenidos. Las pruebas estuvieron orientadas a medir el grado de exactitud de la clasificación de las RNA's y los tiempos de respuesta de cada una de ellas.

Se realizaron tres pruebas de desempeño:

Velocidad de procesamiento.
Porcentaje de clasificación errónea del rostro.

7.1. Velocidad de Procesamiento

La primera prueba realizada es la de obtener la velocidad de procesamiento del prototipo de Reconocimiento de Rostros. Se calcula el tiempo que tarde en capturar, procesar y desplegar la imagen de un rostro en el proceso de clasificación. Para tomar los tiempos se utilizó el reloj interno del computador. La medición se realizó en dos tiempos distintos:

7.1.1. Tiempo de Procesamiento de la Imagen

El tiempo del procesamiento de la imagen se mide desde el momento en que la imagen del rostro es enviada a la función *PCA*. Al enviar la solicitud de la función se inicializa una variable t_0 con el tiempo actual del reloj del sistema, se espera a que se realice todo el proceso de análisis de los principales componentes y en el momento en que ya se ha calculado y se retorna al punto de llamado de la función *PCA* se inicializa otra variable t_1 que contiene el valor actual del reloj del sistema después de haber realizado las operaciones de calculo de los principales componentes de la imagen. El tiempo que tomo procesar la imagen de uno o varios rostros es la diferencia de $(t_1 - t_0)$. Los tiempos de procesamiento por cada uno de los modelos se relacionan en la tabla 7.1.

7.1.2. Tiempo de Clasificación del Rostro

Este proceso recibe la información enviada por la función *PCA*. Calcula el tiempo que tarda en hacer la clasificación del rostro la red neuronal. El tiempo t_0 se toma cuando llegan los datos de la función *PCA*, al llegar el resultado de la clasificación del rostro se inicializa la variable t_1 con la información del tiempo actual del sistema. Una vez obtenidas esta dos variable (t_0, t_1) se realiza la diferencia

$(t_1 - t_0)$ y se obtiene el tiempo que ha sido requerido por la red neuronal para la clasificación de un rostro.

Para calcular el tiempo total del proceso de clasificación del rostro, se suman los tiempos de las dos fases involucradas en el proceso $T_t = T_p + T_c$ (tiempo de procesamiento más el tiempo de clasificación).

Punto de Medición	TC RNA PML (Milisegundos)	TC RNA RBF (Milisegundos)	TC RNA SOM (Milisegundos)
Procesamiento de la Imagen	785.5	740.1	744.8
Clasificación del Rostro	126.4	190.4	129.6
Tiempos Promedios	911.9	930.5	874.4

Tabla 7.1 Mediciones de Velocidad de procesamiento

Los valores relacionados en la tabla 7.1 representan el promedio de los tiempos empleados en las fases de procesamiento y clasificación. El promedio fue calculado tomando el tiempo empleado en el procesamiento de 10 imágenes de rostros diferentes.

7.2. Porcentaje de Clasificación Errónea del Rostro

La siguiente prueba realizada estuvo orientada a obtener el porcentaje de clasificación errónea de los rostros. El proceso consistió en enviar imágenes de rostros a cada una de las redes neuronales con el fin de realizar su clasificación.

El procedimiento fue el siguiente:

1. Selección aleatoria de cada una de las imágenes a procesar.
2. Procesar la imagen.
3. Seleccionar la red neuronal para la clasificación.
4. Contar los rostros clasificados erróneamente. Un rostro clasificado erróneamente es aquel que no concuerda con el rostro de referencia asignado al rostro procesado.
5. Contar el total de imágenes procesadas por cada una de las RNA.

Los resultados son mostrados en la tabla 7.2.

RNA	Imágenes Procesada	Imágenes mal clasificadas	Porcentaje de error
PML	200	7	3.5 %
RBF	200	9	4.5 %
SOM	200	5	2.2 %

Tabla 7.2 Resultados prueba de clasificación

7.3. Porcentaje de Acierto en la Construcción del rostro

Esta prueba consiste en evaluar los datos retornados por la red neuronal en el proceso de clasificación. Una vez se construye la imagen con los datos retornados por la RNA y que se haya determina que la clasificación es acertada, se analiza que porcentaje de similitud tiene la imagen del rostro que se construyó con la imagen de referencia con la cual se entrenó la red neuronal. En otras palabras, que tan parecida es la imagen retornada por el red neuronal a la imagen del rostro que se tomó como referencia para el entrenamiento.





Imagen de Referencia	SOM	PML	RBF	Porcentaje de Acierto
				80%

Tabla 7.2 Porcentaje de acierto el construcción de imágenes

En la tabla 7.2 se observan las imágenes de prueba obtenidas con cada uno de los modelos, y en todos los casos el resultado fue el mismo, lo que indica que en este aspecto los tres modelos tienen el mismo grado de acierto en la construcción de las imágenes con los datos devueltos por la red neuronal.

El porcentaje es asignado con relación a la nitidez de la imagen tomando como una base del 100% a la imagen original que sirvió de referencia para el entrenamiento de la RNA. El 80% obtenido por estas imágenes es un porcentaje aceptable de clasificación considerando el tamaño de la imagen con la cual se esta trabajando, que es de 25x25 píxeles.

CAPÍTULO 8: ANÁLISIS DE RESULTADOS

8. Análisis de Resultados

En este capítulo se recapitulan los objetivos que pretendía abarcar en el presente proyecto de Tesis y se muestran los resultados obtenidos durante todo el proceso

8.1. Metas propuestas y Logros alcanzados

Las metas propuestas y los logros alcanzados se encuentran relacionados en detalle en la tabla 8.1.

Metas Propuestas	Logros Alcanzados	Comentarios
1. Identificar y estudiar tres modelos de reconocimiento de rostros basados en la técnica PCA, y seleccionar de acuerdo a los resultados uno de los tres modelos para su implementación.	<p>Durante el proceso de búsqueda de información se pudo identificar, documentar y conocer las técnicas existentes para el reconocimiento de rostros.</p> <p>Se pudo conocer el amplio campo de acción que tiene la biometría a nivel nacional e internacional, así como la profundidad de la investigación relacionada con el campo del reconocimiento de rostros.</p> <p>Se adquirió conocimiento en el área de procesamiento digital de Imágenes, reconocimiento de patrones, arquitectura de redes neuronales artificiales y el método estadístico de extracción de características por medio del análisis de principales componentes.</p> <p>Se identificaron los tres modelos que cumplían con los requisitos fundamentales para ser involucrados en el proyecto, se documentaron y caracterizaron estos modelos con el fin de determinar cual de ellos presentaba mejor rendimiento frente a otros dos modelos, y de estos se obtuvo el</p>	<p>Fue la etapa mas interesante del proyecto debido a que durante el proceso de su realización se adquirió abundante conocimiento en las áreas relacionadas con los modelos que se escogieron para el estudio y desarrollo de este proyecto.</p>

	<p>modelo que sería el derrotero para la realización del prototipo de reconocimiento de rostros.</p>	
<p>2. Desarrollar un prototipo software para el reconocimiento de rostros en dos dimensiones basado en el modelo que utiliza la técnica PCA.</p>	<p>En la etapa de la construcción del prototipo software de reconocimiento de rostros se desarrollaron habilidades para el tratamiento de imágenes digitales, se profundizó en el conocimiento del método Jacobi para el cálculo de los valores y vectores propio de una matriz simétrica. Se aprendió el manejo del simulador de redes neuronales NeuroSolution. Con NeuroSolution se puso en práctica el conocimiento adquirido en la etapa de documentación del proyecto, sobre especificaciones y características que deben presentar las redes neuronales artificiales para un mejor desempeño.</p> <p>Se desarrollaron habilidades en el desarrollo de librerías de clases para el manejo de Redes Neuronales Artificiales en C++.</p> <p>Se profundizo en el conocimiento de patrones de diseño, Arquitecturas de software para ser aplicadas al prototipo funcional resultado de esta etapa.</p> <p>Se obtuvo el prototipo software de reconocimiento de rostros con las especificaciones necesarias para ser puesto a prueba.</p>	<p>Fue la etapa mas complicada de todo el proyecto, debido a la complejidad de los algoritmos que se debían manejar para el desarrollo de algunas etapas del prototipo, como por ejemplo la parte estadística implementada en el método Jacobi.</p>
<p>3. Establecer el grado acierto y tiempo de respuesta del</p>	<p>En esta etapa se pudo comprobar todas y cada una de las conclusiones teóricas acerca del modelo elegido, concluyendo que efectivamente el modelo que utiliza Redes Neuronales SOM es el modelo mas eficiente para la</p>	<p>Fue la etapa más motivadora del proyecto, debido a que en ella se pudo cosechar el fruto del trabajo de todo el proceso</p>

prototipo implementado.	clasificación de rostros. Se obtuvo conocimiento como para plantear una alternativa diferente para la clasificación de los rostros ampliando el modelo seleccionado para el estudio.	desarrollo.
--------------------------------	---	-------------

Tabla 8.1 Relación de Metas y Logros del proyecto

CAPÍTULO 9: CONCLUSIONES

8. Conclusiones

En este capítulo se presentan las conclusiones del trabajo desarrollado en esta tesis. En la primera sección se exponen las conclusiones particulares del sistema. La segunda sección esta dedicada a las líneas futuras. Se continúa con las limitaciones del sistema y por último, se concluye con reflexiones generales sobre el proyecto del prototipo de Reconocimiento de Rostros.

La realización de este proyecto motivada por un interés personal en los sistemas de seguridad y de inteligencia artificial, debido a que se pretende desarrollar una aplicación de control de acceso que integra modelos de seguridad biométricos, permitió adquirir conocimiento en distintas áreas y la puesta en práctica del saber obtenido en las materias vistas durante la carrera.

Con el análisis comparativo de los modelos se logró determinar cual de los modelos existentes que utilizan la técnica de Análisis de principales componentes para el reconocimiento de rostros y que usa redes neuronales para la clasificación de los rostros es el mas eficiente, así como el ampliar nuestro conocimiento en el área de inteligencia artificial y redes neuronales para plantear un forma de clasificación derivada del modelo seleccionado.

Con el desarrollo de este proyecto se adquirió conocimiento relacionado con los lineamientos que se deben seguir para realizar un proceso de comparación de modelos, sobre procesamiento digital de imágenes, sobre visión artificial, sobre redes neuronales y sobre métodos estadísticos para el análisis de componentes de matrices de datos. Se desarrollaron habilidades para la interacción entre aplicaciones y redes neuronales artificiales.

Con relación a los aportes del proyecto a la comunidad científica, el documento de monografía servirá como marco de referencia para futuros trabajos que tengan como objetivo la comparación entre modelos o técnicas. Otro aporte es que con los resultados obtenidos en el desarrollo de este proyecto se ponen a disposición criterios que permiten seleccionar un modelo con características específicas para el reconocimiento de rostros, así mismo, la arquitectura utilizada en el desarrollo del prototipo servirá de base para futuros desarrollo en el área.

8.1. Conclusiones Particulares del Sistema

El sistema del prototipo de Reconocimiento de Rostros cumplió exitosamente con los objetivos y requerimientos planteados. Se pudo determinar cual de los tres modelos seleccionados para su estudio fue el de mejor desempeño frente los otros dos, y de esto se pudo concluir que el modelo que utiliza las RNA SOM es el mas apropiado para desarrollar un sistemas de reconocimiento de rostros para clasificación. El prototipo del sistema está en la capacidad de capturar la imagen

de un rostro para preprocesarla, procesarla en busca de sus principales componentes y clasificar el rostro con una RNA. Esta clasificación se ha realizado con un porcentaje aceptable de reconocimiento.

La aplicación le permite al usuario configurar de manera sencilla el dispositivo de captura empleado, los valores básicos para el análisis de los principales componentes y los valores de las características de las imágenes de los rostros que se desean procesar.

Las herramientas hardware empleadas se consiguen con facilidad en el mercado, sin embargo, uno de los factores que pueden ser modificados para mejorar el desempeño y la eficiencia del Prototipo de Reconocimiento de Rostros es precisamente este. La utilización de dispositivos que trabajen en paralelo brindan la ventaja de aprovechar mejor las propiedades de las Redes Neuronales Artificiales.

Las herramientas software se han amoldado perfectamente a las necesidades del proyecto. El desempeño, la modularidad y su disponibilidad han permitido el desarrollo de este proyecto de manera exitosa. Además permite a futuros desarrolladores del sistema utilizar un ambiente al que seguramente estarán acostumbrados (dada la popularidad de Windows y .Net) y facilitará el realizar mantenimientos y actualizaciones que puedan ser requeridas por el sistema.

8.2. Líneas Futuras

El reconocimiento de rostros en nuestro país es una línea que se encuentra en vía de desarrollo, es por esto que este proyecto representa un escalón importante en la investigación de este tema en nuestra región. La meta a futuro es implementar una aplicación de reconocimiento de rostros en tiempo real basada en modelos de tres dimensiones, para ampliar los alcances de los estudios realizados en el área, y apoyados en las investigaciones desarrolladas en el departamento de Ingeniería Física quienes han desarrollado sistemas para la digitalización de rostros en tres dimensiones.

Por otro lado la visión por computador o visión artificial, áreas pertenecientes a la inteligencia artificial, es otro campo de aplicación del reconocimiento de patrones, que no solo se utiliza para la seguridad, sino en áreas como la robótica y la biomedicina por mencionar algunas.

Al haber recopilado la información disponible relacionada con el reconocimiento de rostros y al haber realizado un estudio mas profundo sobre los modelos existentes, se deja una base teórica que sirve de fundamento para la toma de decisiones en el caso de ser necesario la implementación de una sistema de clasificación de rostros.

Una vez realizado el estudio del modelo seleccionado se plasmó toda la parte teórico-matemática en el prototipo, lo cual permite dar una base de los procesos a seguir con el fin de transformar la teoría en prototipos productivos. Por otro lado, se plantea una alternativa de mejoramiento de procesos dentro de la implementación de software inyectado agilidad y perfección a las actividades con la inclusión de métodos estadísticos y matemáticos.

8.1. Limitaciones

El prototipo de Reconocimiento de Rostros puede ser mejorado en muchos aspectos en donde las pruebas indican porcentajes de fallas bastante considerables. Esto consistirá en un desafío para aquellos que continúen el trabajo de esta tesis. Sin embargo, esta sección pretende indicar los aspectos donde se debe poner especial atención y que constituyen las limitaciones del sistema:

1. El rendimiento del sistema de reconocimiento de rostros se puede ampliar considerablemente si en vez de utilizar imágenes bidimensionales en escala de grises se hace uso de las imágenes en sus tres canales de colores (RGB).
2. Otra limitante es el control de la iluminación de la imágenes, debido a que se trabajó con una base de datos de imágenes utilizadas por otros autores, no se pudo hacer el control requerido para disminuir el error inyectado por las alteraciones de iluminación, aspecto que se puede tener en cuenta y controlarlo en el momento en el que se está iniciando la creación de la base de datos.

8.4. Conclusiones Generales

El prototipo del sistema de reconocimiento de rostros desarrollado en esta tesis ha contribuido a ampliar el interés en el área de inteligencia artificial y más exactamente en el área de reconocimiento de patrones entre las personas con las cuales se ha compartido el proyecto. En este sentido este trabajo de tesis es de los primeros de muchos por venir, todos ellos motivados por el amplio campo de aplicación que tienen los sistemas relacionados con la inteligencia artificial.

BIBLIOGRAFIA

- [1]. Vernet, D. Canaleta, X. La biometría y su legalidad. Universidad Ramón Llull. 2004.
- [2]. Urraza, J. Sistemas Biométricos. Universidad Católica de Nuestra Señora de Asunción. 2004.
- [3]. VARGAS, Enrique J., Biometría. La Nueva Clave Secreta. JYE Studio. Artículo Introductorio. 2001.
- [4]. Payo Laborda, M. y Medina Quero, J. Biométrica. Universidad de Granada. 2003.
- [5]. Sandström, M. *Liveness detection in fingerprint recognition systems*, Linköping University Electronic Press, Thesis. 2004.
- [6]. J. Ashbourn, *A biometric white paper*. 1999.
- [7]. Bromba, M. *Biometrics FAQ*, Last Change: 2006-02-09
- [8]. Wertheimer, M. *Significant Contributions to the History of Psychology. Series A. Orientations*, vol. ii. [English translation in D.N. Robinson, ed 1977]. 1912.
- [9]. Waltz, D., *Understanding line drawings of scenes with shadows*, Ed. *The Psychology of computer vision*, Mac- Graw Hill. 1975.
- [10]. Horn, B.K.P, *Robot Vision*. The MIT Press, Copyright 1986 by the Massachusetts Institute of Technology, Ninth printing 1993.
- [11]. Ornstein, Robert E. *The Psychology of Consciousness*. Harcourt Brace Joavonovich, Inc. New York. 1977.
- [12]. Romero, L. A., Calonge Cano, T. Redes Neuronales y Reconocimiento de Patrones. Universidad de Salamanca, Universidad de Valladolid. España.
- [13]. Kittler, J. Reconocimiento de Patrones. GTI-IIE. 2002.
- [14]. Watanabe, S. *Pattern Recognition: Human and Mechanical*. Wiley, New York. 1985.
- [15]. Ullman, J. *High Level Vision. Object Recognition and Visual Cognition*. The MIT Press. 1996.
- [16]. Tou, J. T., Gonzalez, R. C. *Pattern Recognition Principles*. Addison-Wesley Publishing Company. 1974.
- [17]. Hrycej T., Modular Learning in Neural Networks: A Modularized. Approach to Neural Network Classification. John Wiley & Soncs, Inc. 1992.
- [18]. Brunelli R., Poggio T., *Face Recognition: Features versus Templates*. IEEE Trans. Patt. Anal. And Mach. Intell., vol. 15, No. 10, pp. 1042-1052, 1993.
- [19]. Jun Zhang, Young Yan, And Martin Lades. "Face Recognition: Eigenface, Elastic Matching, And Neural Nets." Proceedings of The. IEEE. vol. 85. No. 9, pp.1422-1435. 1997.
- [20]. Gutiérrez González, L. Reconocimiento de Rostros usando Redes Neuronales. Tesis Maestría. Universidad de las Américas. 2000.
- [21]. Lopera, F. Procesamiento de Caras: bases Neurológicas, trastornos y evaluación. Universidad de Antioquia. 2000.
- [22]. Daugman, J. *Entropy reduction and decorrelation in visual coding by oriented neural receptive fields*, Transactions on Biomedical Engineering 36 no. 1, 107-114. 1989.
- [23]. González R. and Wood R., *Digital Image Processing*. Addison-Wesley. 1992.

- [24]. Banham, M. and Katsaggelos, A., *Digital image restoration*, IEEE Signal Processing Magazine, vol. 14, 24-41, 1997.
- [25]. Branch, J. and Olague, G. La visión por computador. Una aproximación al estado del arte. 2001.
- [26]. Pratt W., *Digital Image Processing*. Wiley-Interscience, 1991.
- [27]. Rosenfeld A., Kak A., *Digital Picture Processing (V.I y II)*. Academic Press, 1982.
- [28]. Rock, I., *Perception*, Scientific American Library. 1984.
- [29]. Cai, L.D., *Scale-based surface understanding using diffusion smoothing*, PhD Thesis, Department of Artificial Intelligence, University of Edinburgh. 1990.
- [30]. Wieckert, J., *Anisotropic diffusion in image processing*, B.G. Teubner Stuttgart, 1998
- [31]. Witkin, A.P., *Scale-space filtering*, Proc. 8th Int. Conf. on Artificial Intelligence IJCAI-83, Karlsruhe, 1019-1022. 1983.
- [32]. Lindeberg, T., *Scale-space for discrete signal*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 12, no 3, 234-254. 1990.
- [33]. Mallat, S., *A theory of multi-resolution image processing: the wavelet representation*. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 11, no6, 674-693. 1989.
- [34]. Castleman K., *Digital Image Processing*. Prentice-Hall, Inc. 1996.
- [35]. Gauch, J., *Noise removal and contrast enhancement*, en The Colour Image Processing Handbook, Sangwine S.J. and Horne, R.E.N. eds., Chapman and Hall. 1998.
- [36]. Wiener, N., *Extrapolation, interpolation and smoothing of stationary time series*, MIT Press, Cambridge, MA, 1942.
- [37]. Lagendijk, R. and Biemond, J., *Iterative identification and restoration of images*, Kluwer Academic Publishers. 1991.
- [38]. Katsaggelos, A., *Digital image restoration*, Springer-Verlag. 1991.
- [42]. Besag, J., *On the statistical analysis of dirty pictures*, J. Roy. Statist. Soc. B, vol. 43, pp 259-302. 1986.
- [43]. Geman D. and Geman S., *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images* IEEE Trans. on PAMI. vol 6, pp 721-741. 1984.
- [44]. Ripley, B., *Statistical Inference for Spatial Processes*, Cambridge University Press. 1988.
- [45]. Umbaugh, S.E., *Computer vision and image processing. A practical approach using CVIptools*, Prentice Hall, 1998.
- [46]. Jain, A., *Fundamentals of digital image processing*, Prentice-Hall. 1989.
- [54]. Sonka, M., Hlavac, V. and Boyle, R., *Image processing, analysis and machine vision*, Chapman and Hall. 1998.
- [55]. Platero, C., Asencio, G., Moreno, J. Combinación de la Difusión Isotrópica y anisotrópica en el procesamiento de imágenes. UPM. 2004.
- [56]. Gonzalez RC, Woods RE, "*Digital Image Processing*", 2nd Ed., Prentice Hall, 2002.
- [57]. Villareal Benítez, J., Montante López, R., Heras Lara, L. Procesamiento digital de imágenes y de texto. Guías y textos de cómputo. UNAM. 2000.
- [58]. Martínez N., A. Proceso digital de Imágenes. Universidad de Oviedo. 2001.

- [59]. Azpiroz Leehan, J. y Charleston Villalobos, S. Estudio comparativo de cuatro métodos de compresión para imágenes de resonancia magnética nuclear. Universidad Autónoma Metropolitana-Iztapalapa s/n, 09340.
- [60]. Ortiz, M. Filtros Digitales Espaciales. Universidad Autónoma de Puebla. 2000.
- [61]. Yuras, G. Introducción a la Percepción Remota del Océano. PROFCA. Universidad de Concepción.
- [62]. De la Fraga, L. Procesamiento Digital de Imágenes. CINVESTAV. 2001.
- [63]. De Freitas, Joylon M. *Digital Filter Design Solutions*. Norwood, MA, USA: Artech House, Incorporated. 2005.
- [64]. Ortiz, M. Procesamiento Digital de Imágenes. CS. 2005.
- [65]. Pearson, K. *On lines and planes of closest fit to systems of points in space*. Philosophical Magazine. 2: 559-572. 1991.
- [66]. Hotelling, H. "Analysis of a complex of statistical variable into principal componets" J. Educ. Psych., vol. 24, pp. 417-441, 1933.
- [67]. Clavijo M., J. Análisis de Componentes Principales ACP. Universidad del Tolima. 2004.
- [68]. Colmenares, G. Análisis Multivariante y Aplicaciones. Análisis de Principales Componentes. Universidad de Los Andes. Mérida- Venezuela. 2004.
- [69]. SMITH, Lindsay I. *A tutorial on Principal Components Analysis*. 2002.
- [70]. Stanley I. Grossman. Algebra Lineal. Segunda Edición. Grupo Editorial Iberoamérica. 1987.
- [71]. Vazquez, Ernesto, Chacón, Oscar L., Altuve, Hector J. Aplicación del Análisis de pincipales componentes y redes neuronales en la identificación de oscilaciones de potencia. Vol. II No. 5 .1999.
- [72]. Rao, C. R. *The Use and Interpretation of Principal Component Analysis in Applied Research*. Sankhya. 26:329 -358.1964.
- [73]. Isasi Viñuela, Pedro, Galván León, Inés M. Redes Neuronales Artificiales Un Enfoque Práctico. PEARSON EDUCATION S.A. Madrid, 2004.
- [74]. Martín de Brío, Bonifacio, Sanz Molina, Alfredo. *Redes Neuronales y Sistemas Difusos*. 2da. Edición. AlfaOmega Grupo Editor S.A. de C.V. México D.F. 2002.
- [75]. Luís Pavía, G. *Informe sobre redes neuronales*. Perú: Ilustrados.com. 2005.
- [76]. Acosta B., M. I., Zuluaga M., C. A., Salazar I., H. Tutorial de Redes Neuronales. Universidad Tecnológica de Pereira. UTP. 2000.
- [77]. Müller, B., Reinhardt, J. *Neural Networks. An Introduction*, Springer-Verlag, 1990.
- [78]. Rumelhart, D. E., McClelland, J.L. (eds.). *Parallel Distributed Processing*. Vol1: *Foundations*. MIT Press, 1986.
- [79]. Hinton, G. E, Sejnowski, T. J. *Learning and relearning in Boltzmann Machines*. En pp. 282-317, 1986.
- [80]. Minsky, M. and Papera, S. *Perceptrons: An introduction to computacional geometry*. The MIT Press. (1989).
- [81]. Rumelhart, D., Hinton, G., and Willians, R. *Parallel Distributed Processing, chapter Learning representation by back-propagating errors*. MIT Press. (1986).
- [82]. Werbos, P. J. *Beyond Regresión: New Tools for Prediction an Analysis in Behavioral Sciences*. Doctoral Dissertation, Appyied Mathematics, Harvard University, noviembre 1974.

- [83]. Cybenko, G. *Approximation by superposition of a sigmoidal function. Mathematics of Control, Signal and Systems*, 2:3003-314. 1989.
- [84]. Hornik, K., Stinchcombe, M., and White, H. *Multilayer Feedforward networks are universal approximators. Neural Networks*, 2:359-366. 1989.
- [85]. Cohen, M., Franco, H., Morgan, N., Rumerhart, D, and Abrash, V. *Advances in Neural Information Processing System*, capítulo *Context dependent multiple distribution phonetic modeling with MLP's*, págs: 649-657. Morgan Kaufmann. 1993.
- [86]. Sackinger, E., Boser, B., Bromley, J., Lecun, L., and Jackel, L. *Application of an neural network chip to high-speed character recognition. IEEE Transactions on Neural Networks*, 3:498-505. 1992.
- [87]. Guyon, I. *Applications of neural networks to character recognition. International Journal of pattern recognition and Artificial Intelligence*, 5:353-382. 1991.
- [88]. Werbos, P. *Backpropagation and neurocontrol: a review and prospectus. In international Joint Conference on Neural Networks*, volumen 1, págs: 209-216. 1989.
- [89]. Narendra, K. and Parthasarathy, K. *Identificación and control of dynamical systems. IEEE Transactions on Neural Networks*, 1:4-24. 1990.
- [90]. Pomerleau, D. *Neural Network perception for mobile robot guidance. PhD Thesis*, Computer Science, Carnegie Mellon University. 1992.
- [91]. Baxt, W. *The application of the artificial neural network to clinical decision making. In conference on neural information processing systems- Natural and Synthetic*. 1992.
- [92]. Weigend, A., Huberman, B., and Rumelhart, D. *Predicting the future: a connection approach. Technical report*, PARC. 1990.
- [94]. Hartman, E. J., Keeler, J. D., and Kowalski, J. M. *Layered neural networks with Gaussian hidden units as universal approximations. Neural Computation*, 2:210--215. 1990.
- [95]. Geman, S., Bienenstock, E., Doursat, R. *Neural networks and the bias/variance dilemma Neural Computation*, 4, 1-58. 1992.
- [96]. Moody, J. and Darken, C. *Fast learning in networks of locally-tuned processing units. Neural Computation*, 1:281-294. 1989.
- [97]. Renals, S. *Radial basis function network for speech pattern classification. Electronic Letters*, 25:437-439. 1989.
- [98]. Poggio, T. and Girosi, F. *Networks for approximation and learning. In Proceeding of the IEEE*, volumen 78, págs: 1481-1497. 1990.
- [99]. Park, J. and Sandberg, I. *Universal approximation using radial basis function networks. Neural Computation*. 3:246-257. 1991.
- [100]. Kandirkamanathan, V., Niranjan, M., and Fallside, F. *Sequential adaptation of radial basis function neural network. In Advances in Neural Information Processing System*. Págs: 721-727. Morgan Kaufmann. 1991.
- [101]. Saha, A., Christian, J., Tang, D., and Wu, C. *Oriented non-radial basis function for image coding and analysis. In Advances in Neural Information Processing Systems*, págs: 728-734. Morgan Kaufmann. 1991.
- [102]. Lowe, D. and Webb, A. *Exploiting prior knowledge in network optimization: an illustration from medical prognosis. Network*, 1:299-323. 1990.

- [103]. Niranjan, M. and Fallside, F. *Neural networks and radial basis functions in classifying static speech patterns*. *Computer Speech and Language*, 4:275-289. 1990.
- [104]. Broomhead, D. and Lowe, D. *Multivariable functional interpolation and adaptive networks*. *Complex Systems*, 2:321-355. 1988.
- [105]. Duda, R. O. and Hart, P. E. *Pattern Classification and Scene Analysis*. Wiley. (1973).
- [106]. Haykin, S. *Neural Networks. A Comprehensive Foundation*. 2ª edición. Prentice-Hall, 1994-1999.
- [107]. Fritzke, B. *Growing cell structures - a self-organizing network for unsupervised and supervised learning*. *Neural Networks*. 1994.
- [108]. Fyfe, L. *A Neural Implementation of Canonical Correlation Analysis*. 1999
- [109]. Colmenares, G. *Función de Base Radial*. *Redes Neuronales*. Universidad de Los Andes. Mérida- Venezuela. 2002.
- [110]. Kohonen T. *Self-Organized formation of topologically correct feature maps*. *Biological Cybernetics*. 43:59-56. 1982.
- [111]. Kröse, B., Van der Smagt, P. *An introduction to neural network*. Universidad de Amsterdam. 1996.
- [112]. Erik Hjelmas y Jorn Wroldsen, *Face Recognition with Multi_Layer Perceptrons*. <http://citeseer.ist.psu.edu/66390.html>. 1999.
- [113]. Javad Haddadnia, KarimFaez y Majid Ahmadi. *N-Feature Neural Network Human Face Recognition*. <http://ieee-biometrics.dti.unimi.it/knowledge.htm>. 2004.
- [114]. Steve Lawrence, C. Lee Giles, Ah Chung Tsoi y Andrew D. Back. *Face Recognition: A Hybrid Neural Network Approach*. 1997.
- [116]. Carlos Enrique Serrano, *Modelo Integral para el Profesional en Ingeniería*, Editorial: Universidad del Cauca, Pág. 38. 2005.
- [117]. Acosta B., M. I., Zuluaga M., C. A., Salazar I., H. *Tutorial de Redes Neuronales*. Universidad Tecnológica de Pereira. UTP. 2000.
- [118]. Scott, Kendall. *UML Explained*, Addison-Wesley. 2001.
- [119]. Brown, David William. *An Introduction to OBJECT-ORIENTED ANALYSIS Objects and UML in Plain English* John Wiley & Sons, Inc. 2002.
- [120]. Windrow, B., Hoff, M., *“Adaptive Switching Circuits”*, WESCON Convention Record, New York: Part 4, pp. 96-104, 1960.
- [121]. Chellappa, R., Wilson, C., and Sirohey, S. *Human and machine recognition of faces: A survey*. *Proceedings of the IEEE*, 83(5):703–740. 1995.
- [122]. Lai, J. H., Yuen, P. C., and Feng, G. C. *Face recognition using holistic fourier invariant features*. *Pattern Recognition*, 34:95–109. (2001).
- [123]. Campos, R., de Campos, T., Y Cesar, R. *Detection and tracking of facial features in video sequences*. *Lectura en Artificial Intelligence*, 1793:197–206. 2000.
- [124]. Samal, A. and Lyengar, P. *Automatic recognition and analysis of human faces and facial expressions: a survey*. *Pattern Recognition*, 25(1):65–77. 1992.
- [125]. Lam, K. y Yan, H. *Location and extracting the eye in human face images*. *Pattern Recognition*, 29(5):771–779. 1996.
- [126]. Jeng, S., Liao, H., Han, C., Chern, M., and Liu, Y. *Facial feature detection using geometrical face model: an efficient approach*. *Pattern Recognition*, 31(3):273–282. 1998.

- [127]. Yuille, A., Hallinan, P., and Cohen, D. *Feature extraction from faces using deformable templates. International J. Comput. Vision*, 8(2):99–111. 1996.
- [128]. Sirovich, L. and Kirby, M. *Lowdimensional procedure for the characterization of human faces. J. Opt. Soc. Am. A.*, 4(3):519–524. 1987.
- [129]. Gutiérrez, J. C., Beltrán, C., y Liang, Z. Reconocimiento de Rostros Utilizando la Transformada de Wavelets y la Red Neuronal Hopfield. Congreso Latinoamericano de Sistemas, Informática, Telemática, Electrónica y Computación, 2002, Arequipa. Anales del CISAISI2002, 2002.
- [130]. Moreno, J., Gómez, F. J., Fernández M. A., Fernández Caballero A. Reconocimiento de Rostros Utilizando Secuencias de Histogramas como Tramas Espacio-Temporales. Departamento de Informática, Escuela Politécnica Superior, Universidad de Castilla-La Mancha, 02071 – Albacete, España.
- [131]. Morales L., D., Ruiz del Solar, J. Sistemas Biométricos: Matching de Huellas Dactilares Mediante Transformada de Hough Generalizada. Revista Chilena de Ingeniería. Santiago de Chile. 2002.