

ESCUELA POLITÉCNICA DEL EJÉRCITO

SEDE LATACUNGA



CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**“RECONOCIMIENTO DIGITAL DE IMÁGENES APLICADO A
ROSTROS HUMANOS BASADO EN PCA UTILIZANDO REDES
NEURONALES”**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA DE
SISTEMAS E INFORMÁTICA**

EDWIN OSWALDO CHASIQUIZA MOLINA

Latacunga, Enero 2008

ESCUELA POLITÉCNICA DEL EJÉRCITO

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICADO

ING. JOSÉ LUÍS CARRILLO (DIRECTOR)
ING. ARMANDO ÁLVAREZ (CODIRECTOR)

El trabajo titulado "RECONOCIMIENTO DIGITAL DE IMÁGENES APLICADO A ROSTROS HUMANOS BASADO EN PCA UTILIZANDO REDES NEURONALES" realizado por el señor EDWIN OSWALDO CHASIQUIZA MOLINA ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de la Escuela Politécnica del Ejército.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional **SI** recomiendan su publicación.

El mencionado trabajo consta de UN empastado y UN disco compacto el cual contiene los archivos en formato digital. Autorizan al señor CHASIQUIZA MOLINA EDWIN OSWALDO que lo entregue al ING. EDISON ESPINOSA, en su calidad de Coordinador de Carrera.

Latacunga 28 de Enero del 2008

Ing. José Luís Carrillo
DIRECTOR

Ing. Armando Álvarez
CODIRECTOR

ESCUELA POLITÉCNICA DEL EJÉRCITO

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, CHASIQUIZA MOLINA EDWIN OSWALDO

Autorizo a la ESCUELA POLITÉCNICA DEL EJÉRCITO la publicación en la biblioteca virtual de la institución del trabajo de Grado titulado "RECONOCIMIENTO DIGITAL DE IMÁGENES APLICADO A ROSTROS HUMANOS BASADO EN PCA UTILIZANDO REDES NEURONALES" cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Latacunga 28 de Enero del 2008

Edwin Chasiquiza
C.I. N° 050276725-4

AGRADECIMIENTOS

MI MAS SINCERO AGRADECIMIENTO A DIOS, POR PERMITIRME ALCANZAR UN LOGRO MAS EN MI VIDA, A MIS PADRES, HERMANOS, ABUELOS, FAMILIARES POR COMPARTIR SU APOYO, AFECTO, EN ESPECIAL SIEMPRE SU AMOR QUE TUVIERON PRESENTE EN TODO MOMENTO.

A LA ESCUELA POLITÉCNICA DEL EJERCITO SEDE LATACUNGA QUE ME FACILITO LAS INSTALACIONES DE SUS LABORATORIOS Y RECURSOS, LOS PROFESORES POR SUS VALIOSAS SUGERENCIAS Y CRITERIOS EN EL DESARROLLO DE LA PRESENTE INVESTIGACIÓN, A MIS AMIGOS QUE COMPARTIERON MOMENTOS DE ALEGRÍA LOS CUALES GUARDARE SIEMPRE EN MI CORAZÓN.

EDWIN

PRESENTACIÓN

EL PRESENTE PROYECTO ESTA ORIENTADO A LAS TECNOLOGIAS DE RECONOCIMIENTO DE PATRONES POR MEDIO DE TRATAMIENTO DE IMÁGENES DIGITALES, EL ANALISIS DE COMPONENTES PRINCIPALES, EL ESTUDIO DE REDES NEURONALES QUE SON NECESARIAS EN EL CAMPO PROFESIONAL, Y DE GRAN AYUDA AL ESTUDIANTE COMO FUENTE DE CONSULTA.

ASI COMO DEMOSTRAR TANTO LAS VENTAJAS DE ESTAS TECNOLOGIAS PARA FUTURAS INVESTIGACIONES.

ÍNDICE

CAPITULO I.....	1
RECONOCIMIENTO DE ROSTROS	1
1.1. INTRODUCCIÓN	1
1.1.1. DETECCIÓN DEL ROSTRO	2
1.1.2. REPRESENTACIÓN DEL ROSTRO	2
1.1.3. VERIFICACIÓN	3
1.2. EXTRACCIÓN DE ATRIBUTOS	4
1.2.1. EXTRACCIÓN DE CARACTERÍSTICAS DE UNA IMAGEN	4
1.2.2. PUNTOS CARACTERÍSTICOS A PARTIR DE ARISTAS	6
1.2.3. PUNTOS CARACTERÍSTICOS A PARTIR DE LA PROPIA IMAGEN	6
1.2.4. VERSIÓN ESCALADA DE LA IMAGEN	7
1.2.5. HISTOGRAMA DE INTENSIDAD Y DE COLOR	8
1.2.6. PARTICIÓN BINARIA LOCAL (LBP)	9
1.2.7. HISTOGRAMA SOBEL	10
1.2.8. CARACTERÍSTICAS DE TAMURA	10
1.2.9. OTRAS FORMAS DE EXTRACCIÓN DE ATRIBUTOS	13
1.3. PONDERACIÓN DE ATRIBUTOS	13
1.4. SELECCIÓN DE ATRIBUTOS	14
1.4.1. MÉTODOS DE SELECCIÓN DE ATRIBUTOS	15
CAPITULO II	18
MÉTODOS PARA EL RECONOCIMIENTO DE ROSTROS.....	18
2.1. INTRODUCCIÓN	18
2.1.1. MÉTODOS BASADOS EN RASGOS FACIALES	19
2.1.2. ANÁLISIS DE BAJO NIVEL	19
2.1.3. ANÁLISIS DE RASGOS FACIALES	20
2.1.4. REDES NEURONALES	21
2.1.5. ANÁLISIS PROBABILÍSTICOS	21
2.2. REDES NEURONALES ARTIFICIALES (ANNs).....	21
2.2.1. ESTRUCTURA DE LAS REDES NEURONALES ARTIFICIALES	22
2.2.1.1. ESTRUCTURA DE CONEXIÓN	23
2.2.1.2. FASE DE PRUEBA	25
2.2.1.2. TAMAÑO DE LAS REDES NEURONALES ARTIFICIALES	23
2.2.2. TAXONOMIA DE LAS REDES NEURONALES ARTIFICIALES	24
2.2.2.1. FASE DE APRENDIZAJE	24
2.2.3. CLASIFICACIÓN DE LAS REDES NEURONALES ARTIFICIALES	25
2.2.3.1. REDES NEURONALES ARTIFICIALES DE ENTRENAMIENTO SUPERVISADO	25
2.2.3.1.1. RED PERCEPTRON	26
2.2.3.1.2. RED ADALINE	27
2.2.3.1.3. LAS REDES MULTILAYER:	27
2.2.3.2. REDES NEURONALES ARTIFICIALES DE ENTRENAMIENTO NO SUPERVISADO	29
2.2.3.2.1. REDES DE MEMORIA PROASOCIATIVA	29
2.2.3.2.1.1. MEMORIA ASOCIATIVA LINEAL (LAM)	29
2.2.3.2.1.2. MEMORIA ASOCIATIVA NO LINEAL PARA LA RECUPERACIÓN HOLOGRÁFICA	31
2.2.3.2.1.3. REDES DE MEMORIA RETROASOCIATIVA	32
2.2.3.2.1.4. RED DE KOHONEN	35

2.3. ANALISIS DE COMPONENTES PRINCIPALES (ACP)	36
2.3.1. PRINCIPIOS DEL ACP	37
2.4. MÉTODOS BAYESIANOS	44
2.4.1. TEOREMA DE BAYES	44
2.4.2. METODOLOGÍA BAYESIANA	46
2.4.3. CÓMO CUANTIFICAR LA INFORMACIÓN A PRIORI	48
2.5. EIGENFACES	49
2.6. FISHERFACES	50
2.7. METODOS HIBRIDOS	51
2.7.1. MÉTODOS BASADOS EN KERNELS	51
2.7.2. EVOLUTIONARY PURSUIT (EP)	52
2.7.3. SUPPORT VECTOR MACHINE (SVM)	53
2.7.4. ACTIVE APPEARANCE MODEL (AAM)	53
2.7.5. MODELADO 3D	54
2.7.6. ELASTIC BUNCH GRAPH MATCHING (EBGM)	55
2.7.7. MODELOS DE MARKOV ESCONDIDOS (HMM)	56
2.7.8. LOCAL BINARY PATTERNS (LBP)	57
CAPITULO III	59
ALGORITMOS GENÉTICOS	59
3.1. INTRODUCCIÓN	59
3.1.1. MÉTODOS DE REPRESENTACIÓN	60
3.1.2. MÉTODOS DE SELECCIÓN	61
3.1.2.1. SELECCIÓN ELITISTA	62
3.1.2.2. SELECCIÓN PROPORCIONAL A LA APTITUD	62
3.1.2.3. SELECCIÓN ESCALADA	62
3.1.2.4. SELECCIÓN POR TORNEO	62
3.1.2.5. SELECCIÓN POR RANGO	62
3.1.2.6. SELECCIÓN GENERACIONAL	63
3.1.2.7. SELECCIÓN POR ESTADO ESTACIONARIO	63
3.1.2.8. SELECCIÓN JERÁRQUICA	63
3.1.3. MÉTODOS DE CAMBIO	63
3.1.4. OTRAS TÉCNICAS DE RESOLUCIÓN DE PROBLEMAS	65
3.1.4. VENTAJAS DE LOS ALGORITMOS GENÉTICOS	66
3.1.4.1. ASCENSO A COLINA (HILL CLIMBING)	65
3.1.4.2. RECOCIDO SIMULADO (SIMULATED ANNEALING)	65
3.1.5. LIMITACIONES DE LOS ALGORITMOS GENÉTICOS	67
3.1.6. EJEMPLOS ESPECÍFICOS DE LOS ALGORITMOS GENÉTICOS	69
3.2. FUNCIONES DE APRENDIZAJE	75
3.2.1. APRENDIZAJE AUTOMÁTICO	75
3.2.2. TIPOS DE ALGORITMOS	80
CAPITULO IV	83
DESARROLLO E IMPLEMENTACION DEL SISTEMA	83
4.1. ESPECIFICACION DE REQUISITOS DE SOFTWARE	83
4.1.1. INTRODUCCION	83
4.1.2. PROPOSITO	83
4.1.3. AMBITO DEL SISTEMA	83
4.1.4. DEFINICION DE ACRONIMOS Y ABREVIATURAS	84
4.1.4.1. DEFINICIONES	84
4.1.4.2. ACRONIMOS	84
4.1.4.3. ABREVIATURAS	84
4.1.5. REFERENCIAS	85
4.1.6. DESCRIPCION GENERAL	85

4.1.6.1. PERSPECTIVA DEL PRODUCTO.....	85
4.1.6.2. REQUISITOS DE LA APLICACIÓN.....	85
4.1.6.3. CARACTERÍSTICAS DE LOS USUARIOS.....	88
4.1.6.4. RESTRICCIONES.....	88
4.1.6.5. SUPOSICIONES Y DEPENDENCIAS.....	88
4.1.6.5.1. SUPOSICIONES.....	88
4.1.6.5.2. DEPENDENCIAS.....	89
4.1.7. REQUISITOS ESPECÍFICOS.....	89
4.1.7.1. REQUISITOS FUNCIONALES.....	89
4.1.7.1.1. INGRESO DE PERSONA.....	89
4.1.7.1.2. CONSULTAR PERSONA.....	89
4.1.7.1.3. IDENTIFICACIÓN PERSONA.....	90
4.1.7.1.4. IMPRESIÓN DE REPORTE.....	90
4.1.7.2. REQUISITOS DE INTERFACES EXTERNAS.....	90
4.1.7.2.1. INTERFACES DEL ADMINISTRADOR.....	90
4.1.7.2.2. INTERFACES DE SOFTWARE.....	90
4.1.7.3. REQUISITOS DE DESARROLLO.....	90
4.1.7.4. REQUISITOS TECNOLÓGICOS.....	91
4.1.7.4.1. REQUERIMIENTOS DE HARDWARE.....	91
4.1.7.4.2. REQUERIMIENTOS DE SOFTWARE.....	91
4.1.7.4.3. REQUERIMIENTOS DE AMBIENTE.....	91
4.2. CONSTRUCCION DEL SISTEMA.....	92
4.2.1. EL PROCESO UNIFICADO RATIONAL (RUP).....	92
4.2.1.1. ARQUITECTURA DEL SISTEMA.....	94
4.2.2. ANALISIS DEL SISTEMA.....	94
4.2.2.1. DIAGRAMA DE CASOS DE USO.....	95
4.2.2.2. DIAGRAMAS DE CASOS DE USO EN FORMATO EXPANDIDO.....	97
4.2.3. DISEÑO DEL SISTEMA.....	102
4.2.3.1. DIAGRAMA DE SECUENCIA: INGRESAR PERSONA.....	103
4.2.3.2. DIAGRAMA DE SECUENCIA: CONSULTAR PERSONA.....	104
4.2.3.3. DIAGRAMA DE SECUENCIA: IDENTIFICAR PERSONA.....	105
4.2.3.4. DIAGRAMA DE SECUENCIA: IMPRIMIR REPORTE.....	106
4.2.3.5. DIAGRAMA DE COLABORACIÓN: INGRESAR PERSONA.....	107
4.2.3.6. DIAGRAMA DE COLABORACIÓN: CONSULTAR PERSONA.....	107
4.2.3.7. DIAGRAMA DE COLABORACIÓN: IDENTIFICAR PERSONA.....	108
4.2.3.8. DIAGRAMA DE COLABORACIÓN: IMPRIMIR REPORTE.....	108
4.2.4. DIAGRAMA DE CLASES.....	109
4.3. IMPLEMENTACIÓN.....	109
4.3.1. EJEMPLO DE EXTRACTO DE CÓDIGO PARA EL ANÁLISIS DE COMPONENTES PRINCIPALES DE UNA IMAGEN.....	109
4.3.2. EJEMPLO EXTRACTO DE CÓDIGO PARA CREAR LA UNA RED NEURONAL.....	110
4.4. PRUEBAS.....	113
4.4.1. PRUEBA DE LAS HIPÓTESIS.....	113
4.4.1.1. ESTABLECIMIENTO DEL NIVEL DE SIGNIFICANCIA.....	114
4.4.1.2. SELECCIÓN DE LA MUESTRA.....	114
4.4.1.3. PASOS PARA LA RECOLECCIÓN DE DATOS.....	114
4.4.1.4. TABULACIÓN DE DATOS.....	115
4.4.1.5. CÁLCULO DEL ESTADÍSTICO.....	116
4.4.1.6. GRADOS DE LIBERTAD.....	118
4.4.1.7. DECISIÓN.....	118
4.5 IMPLANTACIÓN.....	119
4.5.1. PRERREQUISITOS PARA LA INSTALACIÓN.....	119

CAPITULO V	120
CONCLUSIONES Y RECOMENDACIONES	120
5.1. CONCLUSIONES	120
5.2. RECOMENDACIONES	122

ÍNDICE DE FIGURAS

Figura 1.2.1. Ejemplo de filtro de Canny	5
Figura 1.4.1 Clasificador del método Wrapper	16
Figura 2.1. Jerarquización de métodos para el reconocimiento de Rostros	18
Figura 2.2. Red Neuronal Artificial	22
Figura 2.2.1.1. Estructura de Conexión.....	23
Figura 2.2.3.1. Diagrama esquemático del sistema de entrenamiento	25
Figura 2.2.3.1.1. Red Perceptron.....	26
Figura 2.2.3.1.2. Red multilayer feed-forward.....	27
Figura 2.2.3.1.3. Red <i>back-propagation</i>	28
Figura 2.2.3.2. Diagrama esquemático del sistema de entrenamiento	29
Figura 2.2.3.2.1.2. Memoria asociativa no lineal.....	31
Figura 2.2.3.2.1.3. Memoria Retroasociativa.....	33
Figura 2.2.3.2.1.4. Red Konohen	36
Figura 2.3.1.1. Graficación de los datos.....	38
Figura 2.3.1.2. Graficación de Rectas y Gravedad.....	39
Figura 2.4.1 Teorema de Bayes.....	45
Figura 2.4.2 Metodología Bayesiana	47
Figura 2.7.8. Operador LBP.....	57
Figura 3.1.1. Árboles utilizados en la programación genética	61
Figura 3.1.3.1. Método de Cruzamiento	64
Figura 3.1.3.2. Método de Mutación.....	64
Figura 4.2.1.1. Representación de la Arquitectura del Sistema	94
Figura 4.2.1: Interacción entre las dimensiones Tiempo y Componentes de Proceso	93
Figura 4.2.2.1. Diagrama de casos de uso	95
Figura 4.2.3.1. Diagrama de Secuencia Ingresar Persona	103
Figura 4.2.3.2. Diagrama de Secuencia Consultar Persona	104
Figura 4.2.3.3. Diagrama de Secuencia Identificar Persona	105
Figura 4.2.3.4. Diagrama de Secuencia Imprimir Reporte.....	106
Figura 4.2.3.5. Diagrama de Colaboración Ingresar Persona.....	107
Figura 4.2.3.6. Diagrama de Colaboración de Consultar Persona	107
Figura 4.2.3.7. Diagrama de Colaboración de Identificar Persona	108
Figura 4.2.3.8. Diagrama de Colaboración de Imprimir Reporte.....	108
Figura 4.2.4. Diagrama de Clases del Sistema SIFARE.	109
Figura 4.3.2.1. Arquitectura de la red neuronal	111
Figura 4.3.2.2. Representación de la imagen con vectores únicos.....	112
Figura 4.4.1.1. Curva de distribución del chi cuadrado	114
Figura 4.4.1.2. Zona de Aceptación de la hipótesis	118

ÍNDICE DE TABLAS

Tabla 2.3.1. Datos para el Análisis de Componentes Principales	37
Tabla 4.1.4.1. Definiciones	84
Tabla 4.1.4.2. Acrónimos	84
Tabla 4.1.4.3. Abreviaturas	84
Tabla 4.4.1.4. Contingencia de los resultados de las pruebas	116
Tabla 4.4.1.5.1. Contingencia ponderada	116
Tabla 4.4.1.5.2. Frecuencias observadas	117
Tabla 4.4.1.5.3. Frecuencias teóricas esperadas	117

CAPITULO I

RECONOCIMIENTO DE ROSTROS

Este capítulo muestra las etapas de preprocesamiento de imágenes dejando las imágenes digitales listas para una extracción de características visuales e información importante para hacer posible el reconocimiento de rostros. Los procesos que se aplican, dependen de las características que se desean obtener.

1.1. INTRODUCCIÓN

El reconocimiento de rostros humanos se puede sintetizar como la identificación de un rostro utilizando únicamente los rasgos faciales, sin utilizar otros elementos (voz, huellas dactilares, firma, ADN, etc.). Es interesante notar que los trabajos pioneros en el reconocimiento automático de rostros se pueden remontar al s. XIX y concretamente, a los trabajos precursores de Francis Galton¹.

En el siguiente capítulo se mencionan diferentes planteamientos para el reconocimiento de rostros, especialmente los surgidos en los últimos años. Sin embargo la mayoría utiliza una serie de supuestos, más o menos explícitos, como son: Imagen frontal o perfil, por lo tanto va a ser posible encontrar todos los rasgos en la imagen, la imagen está derecha, no tiene ninguna inclinación o giro, el número de casos es relativamente pequeño.

Dado que se busca identificar el rostro, se supone que la localización ha sido ya hecha y las características ya extraídas. Se puede agrupar los puntos anteriores en tres problemas que son: Detección del rostro, representación del rostro y clasificación del rostro, las cuales se describirán a continuación.

¹ G. W. Cottrell, M. K. Fleming. Face recognition using unsupervised feature extraction

1.1.1. DETECCIÓN DEL ROSTRO

La detección del rostro comprende en localizar un rostro en la imagen digital, de una manera sencilla, para ello debe cumplir con ciertas condiciones de control sobre las imágenes obtenidas, por ejemplo las fotografías obtenidas por la policía, registro civil, inmigración, etc. Deben obtenerse de una manera muy clara para una detección eficaz, en otros casos la localización del rostro en la imagen no es conocida inmediatamente.

Durante el proceso de detección del rostro existen factores que vuelven a este problema muy complejo. Uno de ellos es el problema del vello facial (bigote, barba), maquillaje, etc. que enmascaran las características faciales; otro es la variación en la escala y orientación de la cara en la imagen. Por último, existen otros dos factores que van a dificultar la detección: la iluminación que tenga la escena y la calidad de las imágenes.

Existen métodos para la detección de rostros que se basan en: plantillas flexibles, análisis de componentes principales, redes neuronales, tratamiento de imágenes, etc.

1.1.2. REPRESENTACIÓN DEL ROSTRO

La representación del rostro en ecuaciones matemáticas se realiza según las características que se quieran extraer o identificar de la imagen digital.

Las representaciones más utilizadas son:

- Imágenes como matrices bidimensionales de niveles de gris.
- Vectores característicos.

En las matrices bidimensionales se requiere compactar la información cuando el número de imágenes es elevado. Se ha puesto en evidencia que imágenes de 32 x 32 píxeles con 4 bits por píxel son suficientes para identificar a un sujeto. Ejemplos de esta representación son los métodos basados en compactación usando análisis de componentes principales, fisherfaces y redes neuronales.

En el caso de vectores característicos se utilizan características bidimensionales o tridimensionales, que se basan en modelos del rostro. Se engloban en esta categoría los trabajos centrados en emparejamiento de plantillas flexibles, Jets de Gabor, y otras técnicas que han aparecido en los últimos años.

1.1.3. VERIFICACIÓN

La verificación se define como la tarea de comprobar si un nombre coincide con un rostro dentro del reconocimiento de rostros. El primer paso es seleccionar cómo se va a realizar este proceso, para ello se pueden utilizar varios métodos; los más utilizados son: redes neuronales, correspondencia de plantillas y autocaras. Existen otros métodos, los cuales se estudiarán en el capítulo siguiente.

Todos estos métodos toman como hipótesis de partida el hecho de que para un rostro, los valores de las características que lo definen no varían mucho en diferentes imágenes. Es más, si un conjunto de características es muy diferente en dos imágenes los rostros correspondientes serán también diferentes.

Otros hechos de la verificación que realizan los seres humanos que se intentan simular son:

- La importancia de los rasgos faciales decrece de arriba hacia abajo en el siguiente orden: pelo, ojo, nariz, boca, barbilla. Pero caras con rasgos no habituales son identificadas rápidamente.
- Las caras familiares son identificadas de una forma más rápida que otras menos familiares.
- Es complicado describir un rostro humano, ya que tiene muchas características.
- El reconocimiento de un rostro es el resultado de un análisis de rasgos globales junto con otros locales.

Se debe tener en cuenta que no existe una forma de calcular el número de características necesarias. Las mismas deben reunir una serie de condiciones las cuales son:

- Deben ser fáciles de estimar.

- Deben ser independientes de la iluminación.
- Deben ser independientes de cambios en la expresión facial.
- Deben tener rasgos altamente discriminantes.

Se toma en cuenta que las medidas han de ser independientes de la posición, escala y orientación de la cara en escena para obtener un resultado satisfactorio.

1.2. EXTRACCIÓN DE ATRIBUTOS

La extracción de atributos es la utilización de métodos y técnicas que permitan reunir información a partir de imágenes, los atributos que se van a extraer son puntos. Se podría haber elegido como atributos las líneas o bordes. Las líneas son más fáciles de detectar lo cual es muy importante a la hora de mostrar eficiencia. Los puntos son importantes a la hora de correlacionar atributos. En vez de extraer puntos, se extrae líneas o bordes, dependiendo de la posición en la que la cámara se encuentre, dará una perspectiva de dicha línea o borde, a la hora de utilizar el algoritmo se presentará más dificultades con los puntos para estimar que características son similares.

1.2.1. EXTRACCIÓN DE CARACTERÍSTICAS DE UNA IMAGEN

Una imagen contiene una gran cantidad de datos, la mayoría de los cuales proporciona muy poca información para reconocer un rostro dentro de esta imagen. Cualquier sistema que pretenda trabajar con ella debe, en un primer paso, extraer de la forma más eficaz y robusta las características que proporcione la máxima información posible. Estas características deben cumplir, entre otras, las siguientes condiciones:

- Su extracción a partir de la imagen no debe suponer un coste excesivo al sistema en el cual está integrado. El tiempo total de extracción debe ser lo más pequeño posible.
- La forma en que se va a extraer características de la imagen debe ser muy precisa. El error cometido al extraer las características también debe ser lo más pequeño posible.

- Estas características deben ser robustas y estables, estas deben contener una gran cantidad de información para el reconocimiento del rostro, es decir, debe ser capaz de extraer toda la información que necesaria a partir de ellas.

Las aristas, esquinas y uniones satisfacen estas condiciones. Aparecen de forma natural en la mayoría de las imágenes. Sin embargo, la obtención de las aristas suele ser fuente de muchos errores debido a que señalan lugares innecesarios.

Por ejemplo el algoritmo de detección de aristas más comúnmente utilizado, el algoritmo de Canny. Este algoritmo combina un operador diferencial con un filtro gaussiano. Variando el parámetro de varianza del filtro que utiliza este algoritmo.

Cuando se disminuye la varianza aparecen un gran número de aristas, la mayoría de las cuales no aportan mucha información. La localización de las aristas en este caso es muy buena ya que señala las zonas más importantes del rostro dentro de la imagen digital. Al aumentar la varianza se quedan las aristas en una zona con gran contraste. Sin embargo, la localización en este caso no es exacta y el tiempo de procesamiento aumenta. Este comportamiento sucede también en la mayoría de los filtros de detección de aristas. También el uso de derivadas de la imagen para el cálculo de aristas supone un error adicional por el ruido que se produce en el proceso de captación de la imagen. Para ver el comportamiento del filtro de Canny ver la figura 1.2.1

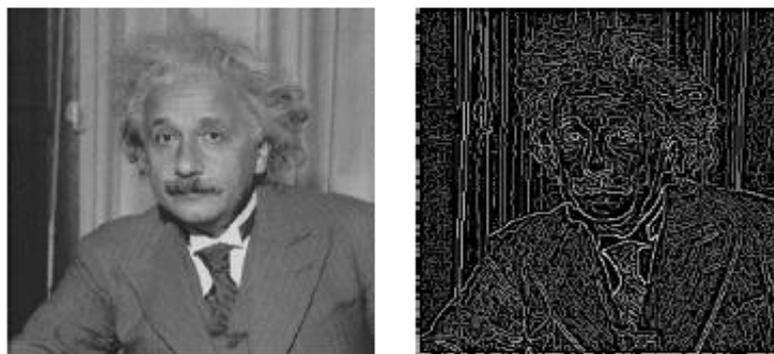


Figura 1.2.1. Ejemplo de filtro de Canny

Desde el punto de vista computacional, se han propuesto una serie de enfoques para la detección de este tipo de características bidimensionales. Se puede clasificar estos enfoques en dos grupos principales:

1. Métodos que obtienen las aristas de la imagen mediante algún método de detección de aristas como el de Canny, para, a continuación, detectar puntos de cruce entre aristas. Estos puntos se clasifican como puntos unión.

2. El resto de métodos trabajan directamente sobre imágenes, ya sea en escala de grises o en imágenes a color, es decir, no se infieren a partir de la extracción de aristas.

1.2.2. PUNTOS CARACTERÍSTICOS A PARTIR DE ARISTAS

Es la extracción de puntos a partir de aristas dentro de una imagen digital. Una vez obtenida las aristas estas se representan mediante un código de cadena. El cual permite expresar el cambio de dirección de un píxel al siguiente. Dentro del código de cadena se deben encontrar los puntos de máxima curvatura.

1.2.3. PUNTOS CARACTERÍSTICOS A PARTIR DE LA PROPIA IMAGEN

Hay otros métodos que obtienen los puntos de máxima curvatura dentro de una imagen, por ejemplo utilizar directamente los valores de la imagen, sin realizar el paso previo de obtención de aristas. Estos métodos definen una medida que suelen denominar de “esquinidad” la cual se calcula para todos los puntos de la imagen. Cuando esta medida supera un cierto umbral se considera que el punto es una esquina. La mayoría de estos métodos utilizan operadores diferenciales.

Uno de los primeros métodos para la obtención de puntos característicos es el de Moravec².

² H. P. Moravec. “Towards automatic visual obstacle avoidance. In *Proc. of the International Joint Conference on Artificial Intelligence*,” 1977.

Dada una ventana rectangular, este operador calcula el valor de cambio de la intensidad de la imagen cuando se desplaza ligeramente esta ventana. Una esquina viene caracterizada por elevados cambios de intensidad. Es altamente sensible a ruido.

Beaudet³ propone un operador que calcula el determinante de una matriz y queda definido como:

$$DET = I_{xx}I_{yy} - I_{xy}^2,$$

Donde I_{xy} es la derivada parcial de la imagen con respecto al eje x y después con respecto al eje y , y I_{xx} y I_{yy} es la segunda derivada con respecto a x e y , respectivamente. Este operador presenta ciertas características de curvatura interesantes. Las esquinas se obtienen aplicando este operador y realizando una umbralización. R. Deriche y G. Giraudon⁴ amplía este método incorporando conocimiento de arista y multiresolución y se realiza un estudio analítico del modelo de esquina. Posteriormente, Harris y Stephens⁵ continúan investigando sobre las propiedades del determinante antes mencionado. En este último trabajo se presenta un nuevo método de detección de puntos característicos que presenta robustez frente al ruido. La definición de este método se realiza de manera similar al operador de Moravec, pero calculando las derivadas de primer orden de la imagen.

1.2.4. VERSIÓN ESCALADA DE LA IMAGEN

Es una técnica que utiliza la comparación de las imágenes, que consiste en evaluar directamente la matriz de píxeles cuando ambas imágenes tienen la misma dimensión.

Mientras las dimensiones de una imagen crecen linealmente, el número de píxeles crece cuadráticamente en función del área.

Uno de los propósitos de extraer características de una imagen para las aplicaciones de recuperación por contenido, es reducir la complejidad de la matriz de píxeles de la imagen original, razón por la cual la versión escalada de las imágenes o *down scaled image* ha sido propuesta y utilizada.

³ P.R. Beaudet. "Rotational invariant image operators. In *Proc. of the 4th. International Conference on Pattern Recognition*", 1978.

⁴ R. Deriche and G. Giraudon. "A computational approach for corner and vertex detection. *International Journal of Computer Vision*," 1993.

⁵ C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Fourth Alvey Vision Conference*,(1988).

El propósito de escalar las imágenes es contar con una muestra representativa de los píxeles originales. La versión escalada de la imagen puede ser entendida como un arreglo de intensidades que representa la señal original pasando a un problema de detección de diferencias entre las señales. Algunas extensiones para esta representación son aplicar filtros sobre la imagen antes o después de escalarla para resaltar detalles de interés.

Uno de los problemas en esta representación es resolver cuáles píxeles de la imagen A corresponden con los de la imagen B, aunque las versiones escaladas tienen las mismas dimensiones, los valores de intensidad pueden estar desfasados cuando la imagen tiene una apariencia equivalente, convirtiendo el problema en uno de registro de imágenes.

1.2.5. HISTOGRAMA DE INTENSIDAD Y DE COLOR

Esta técnica es utilizada para aproximar distribuciones de probabilidad. Los histogramas pueden ser representados de forma paramétrica cuando se tiene algún conocimiento acerca de la distribución de los datos, o utilizando bins en el caso contrario. En la distribución queda representada por un número controlado de parámetros, mientras si se tiene un gran número de bins se representan en la distribución.

Los métodos convencionales para la construcción de histogramas, adoptó el enfoque no paramétrico para la estimación de la función de densidad, el cual va de la siguiente manera: El espacio x de características es dividido en M regiones. Estas regiones forman una grilla espaciada regularmente, en forma de hipercubos del mismo tamaño llamados $x^{(m)}$, que cumplen:

$$\chi^{(m)} \subset \chi$$

Teniendo

$$\bigcup_{m=0}^{M-1} \chi^{(m)} = \chi$$

Y también

$$\chi^{(m)} \cap \chi^{m'} = \phi, \forall m \neq m'$$

Para calcular la densidad en cada sub región de probabilidad se cuentan los puntos que caen dentro del hipercubo, en particular los histogramas de color son una partición del espacio tridimensional de color.

Cuando se tiene una imagen en escalas de grises, cada uno de los píxeles tiene un valor entre 0 y 256. En este caso el espacio de intensidades está claramente definido y la construcción del histograma se reduce a contabilizar el número de píxeles que tienen un determinado valor en escala de gris, obteniendo un resultado con 256 posiciones. El histograma de Lenna tiene una distribución más centrada mientras que la imagen de histología tiende a los valores más brillantes por la gran cantidad de regiones blancas.

El espacio RGB tiene más de 16 millones de colores en total, un histograma de estas dimensiones es bastante complejo de manejar y proporciona poca información, porque en muchos casos esta magnitud es varias veces mayor que el número de píxeles en la imagen. Por esta razón se prefiere definir una partición del espacio diferente cambiando el tamaño del *bin* o hipercubo. Como el espacio RGB es tridimensional, cada una de sus dimensiones se divide en k partes de igual tamaño, obteniendo un total de k^3 sub cubos. El histograma de color presenta problemas de discontinuidad, porque un pequeño cambio en la imagen genera un gran cambio en el histograma.

1.2.6. PARTICIÓN BINARIA LOCAL (LBP)

La partición binaria local es un descriptor de textura en imágenes propuesto por Wang⁶. Para cada píxel P en la imagen se examina su 8-vecindad, es decir los 8 píxeles que se encuentran a su alrededor. Por cada píxel en esa vecindad debe determinarse si el valor de intensidad es mayor o menor que el valor de intensidad del píxel P , en caso de ser mayor se asigna 1 a esa posición de vecindad y de lo contrario se asigna un 0. Con esos valores asignados se codifica una cadena binaria por píxel, conteniendo en total 8 bits que indican la distribución de las intensidades a su alrededor.

⁶L.Wang and D. C. He, Texture classification using texture spectrum, Pattern Recognit. Lett. 13, 1990, 905 910.

Esta partición constituye un espacio con $2^8 = 256$ posibles valores, de manera que se puede calcular una distribución de textura de forma similar a los histogramas en escalas de grises.

1.2.7. HISTOGRAMA SOBEL

Para la detección de bordes en imágenes el histograma de Sobel es uno de los más famosos. Su principio fundamental es calcular la diferencia de intensidad en la vecindad de cada píxel en las direcciones vertical y horizontal, lo cual se interpreta como la derivada de la función en el punto. Con el operador de Sobel puede obtenerse información de la magnitud y la dirección del cambio en cada punto utilizando:

$$M(x, y) = \sqrt{M_x(x, y)^2 + M_y(x, y)^2}$$

Y

$$\theta(x, y) = \tan^{-1} \left(\frac{M_y(x, y)}{M_x(x, y)} \right)$$

En donde M es la magnitud del vector en la dirección del eje y es la dirección. La función diferencial para x y para y puede definirse de tamaño arbitrario utilizando los coeficientes en el triángulo de pascal

Para calcular las diferencias entre las vecindades, primero se multiplica cada vecino por el coeficiente de la matriz, luego se efectúa la suma algebraica entre filas y columnas y después se aplica la definición de magnitud dada. En este caso la imagen resultante contiene en cada píxel el valor de la magnitud de cambio, de manera que construir un histograma con base en estos datos, proporciona información sobre la distribución de los cambios en la imagen.

1.2.8. CARACTERÍSTICAS DE TAMURA

Las características de Tamura es la correlación de la percepción humana, son en total 6, la tres primeras son: coarseness, contraste y direccionalidad. Tamura presenta la forma de calcular 6 características para obtener un valor escalar por cada imagen procesada, realiza algunas adaptaciones a la formulación para tener la posibilidad de calcular las característica por píxel.

Coarseness de Tamura

El coarseness⁷ proporciona información acerca del tamaño de las texturas en la imagen. Cuando las texturas tienen un micro-patrón y un macro-patrón el procedimiento de Tamura considera el patrón más grande, y la forma de hacerlo es aplicando operadores de distinto tamaño de la siguiente manera:

Sea I una imagen; $I(x, y)$ el valor del píxel en la posición (x, y) .

1. Para cada píxel (x, y) se calcula el promedio sobre una vecindad definida que debe ser de un tamaño potencia de dos.

$$A(x, y) = \frac{1}{2^{2k}} \sum_{i=1}^{2^k} \sum_{j=1}^{2^k} I(x - 2^{k-1} + i, y - 2^{k-1} + j)$$

2. Para cada punto (x, y) se calcula la diferencia entre las vecindades que no se excedan los lados opuestos del punto en las direcciones horizontal y vertical.

$$D_k^h(x, y) = |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)|$$

Y

$$D_k^v(x, y) = |A_k(x, y + 2^{k-1}) - A_k(x, y - 2^{k-1})|$$

3. Para cada punto (x, y) seleccionar el valor con la diferencia más grande:

$$S(x, y) = \text{Max}_{d=h,v} \{E_k^d(x, y)\}, k = \{1, \dots, 5\}$$

Este valor se asigna a cada píxel obteniendo el coarseness local de la imagen, lo cual permite calcular una distribución de coarseness en la imagen, es decir, un histograma coarseness que permita medir los tamaños de las texturas.

Contraste de Tamura

El contraste está influenciado por los niveles de grises, la polarización en los niveles de blanco y negro, definición de los bordes y período de repetición de los patrones. Este se

⁷ Se define como rugosidad

calcula utilizando la media y la varianza de los valores de intensidad en una vecindad dada de la siguiente manera:

$$F_{con} = \frac{\sigma}{\alpha_4^z} \quad \text{con} \quad \alpha_4^z = \frac{\mu_4}{\sigma_4}$$

En donde

$$\mu_4 = \frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y (I(x, y) - \mu)^4$$

Es el cuarto momento alrededor de la media μ , σ^2 es la varianza de los valores de grises en la vecindad, y z ha sido configurado experimentalmente como 1/4.

Direccionalidad de Tamura

No se trata de la orientación sino la presencia de dirección en la imagen es interesante. Para calcular la direccionalidad, las derivadas horizontal y vertical son calculadas al convolucionar la imagen utilizando operadores de 3×3 :

$$\Delta_H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Y

$$\Delta_V = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Luego, para cada posición (x, y) se calcula el ángulo de orientación utilizando:

$$\theta = \frac{\pi}{2} + \tan^{-1} \frac{\Delta_V(x, y)}{\Delta_H(x, y)}$$

Con la definición de las características de Tamura presentadas anteriormente, se obtienen tres medidas de textura por píxel en la imagen, con los cuales puede construirse una

imagen RGB que no contiene información de los colores sino de las texturas, esto permite construir un histograma de Tamura.

1.2.9. OTRAS FORMAS DE EXTRACCIÓN DE ATRIBUTOS

Existen muchas otras características desde sencillos estadísticos para regiones de interés hasta técnicas de segmentación, transformaciones y filtros.

- *Coefficientes Wavelet*: La imagen se reescala a 32x32. Luego se realiza una descomposición utilizando la wavelet haar obteniendo un conjunto de coeficientes que son utilizados para representar la imagen.
- *Flesh*: Utilizan un operador de *detección de piel* para detectar gente desnuda.
- *Versiones Grid*: para algunas una de las descritas en secciones anteriores, se propone dividir la imagen en 4 partes iguales, y calcular las características en la baldosa inferior izquierda. Otra alternativa es dividir la imagen original en 3 piezas iguales horizontalmente, obteniendo características para cada pieza.
- *Características Gabor*: El filtro Gabor extrae información acerca de las texturas calculando información espacial y frecuencial simultáneamente utilizando la Transformada de Fourier por Ventanas. Al aplicar esta transformada, la señal original (imagen) es multiplicada por una función de ventaneo, siendo necesario encontrar un tamaño adecuado de la ventana para obtener información espectral útil, y al utilizar como función de ventaneo el filtro Gabor, se obtiene un tamaño de ventana óptimo.

1.3. PONDERACIÓN DE ATRIBUTOS

Un atributo es ponderado⁸ dependiendo del número de objetos que contiene; esto es, si un atributo presenta únicamente un objeto, este atributo discrimina muy bien a ese objeto y por lo tanto tendrá un voto mayor.

Por lo contrario si todos los objetos presentan un mismo atributo, esto indica que el atributo no discrimina lo suficiente entre objetos y por lo tanto tendrá un voto menor.

⁸ Computación y sistemas Vol.3 pp. 214-219 © 2000 , CIC-IPN ISSN 1405-5546

1.4. SELECCIÓN DE ATRIBUTOS

El proceso de selección de atributos escoge un subconjunto de atributos del conjunto original, este proceso pretende elegir atributos que sean relevantes para una aplicación y lograr el máximo rendimiento con el mínimo esfuerzo.

Se debe tener en cuenta que los atributos irrelevantes y redundantes pueden tener un efecto negativo en los algoritmos de clasificación:

1) Al tener más atributos, normalmente implica la necesidad de tener más instancias para garantizar la fiabilidad de los patrones obtenidos, variabilidad estadística entre patrones de diferente clase. Por consiguiente, el algoritmo de clasificación tardará más tiempo.

2) Los atributos irrelevantes y los redundantes, pueden confundir a los algoritmos de aprendizaje. Por lo que en general, el clasificador obtenido es menos exacto que otro que aprenda sobre datos relevantes.

3) Además, con la presencia de atributos redundantes o de irrelevantes, el clasificador obtenido será más complejo, dificultando el entendimiento de los resultados.

4) Además, la reducción de características se podría tener en cuenta en futuras capturas de datos, reduciendo el coste de almacenamiento y tal vez el económico.

En resumen, el resultado de la selección de atributos sería:

Menos datos → los algoritmos pueden aprender más rápidos.

Mayor exactitud → el clasificador generaliza mejor.

Resultados más simples → más fácil de entender.

Menos atributos → evitar obtenerlos posteriormente.

La selección es efectiva al eliminar atributos irrelevantes y redundantes, incrementando la eficiencia en las tareas, mejorando el rendimiento y la comprensión de los resultados.

Existen dos aproximaciones para realizar una reducción de dimensionalidad, selección y transformación de características, ambas son técnicas de preprocesado que se usan frecuentemente. La transformación de parámetros es el proceso a través del cual se crea un nuevo conjunto de parámetros.

Existiendo dos variantes:

Construcción de parámetros: es el proceso de descubrir información oculta sobre relaciones entre parámetros, aumentando el espacio de los atributos. Después de la construcción de parámetros, se obtienen p atributos adicionales:

$$X_{n+1}; X_{n+2}; \dots; X_{n+p}.$$

Extracción de parámetros: es el proceso de extraer un conjunto de nuevos atributos de los originales, a través de algunas funciones. Después se obtiene:

$$B_1; B_2; \dots; B_s$$

$$(s < n), B_i = F_i(X_1; X_2; \dots; X_n), \text{ y } F_i \text{ es una función.}$$

1.4.1. MÉTODOS DE SELECCIÓN DE ATRIBUTOS

Existen dos métodos para seleccionar atributos dentro de una imagen cuyos datos permitan un mejor resultado en los algoritmos de clasificación y estos son:

Métodos basados en filtros.- Se filtran atributos antes de procesarlos, se basan en medidas de calidad de atributos.

El método de filtro se basa en la idea de:

- Dar una calidad de los atributos.
- Suelen aplicar haciendo un ranking de los atributos
- Utilizan las Métricas

- Ganancia de Información: Es una heurística utilizada para la elección del mejor atributo en una imagen.
- Chi Cuadrado: Proporciona medias de discrepancia que existe entre datos observados y datos obtenidos.
- Ratio de ganancia: Es medir información de atributos que tienen muchos valores.
- Relief: Es la asignación de pesos a variables para determinar la significancia de cada variable en el contexto de la tarea de clasificación supervisada.
- Otros

Métodos basados en Wrappers.- Se seleccionan los atributos en función de la calidad del modelo asociado a los atributos utilizados.

El método Wrapper realiza un estudio de la calidad de los atributos en función de la precisión obtenida por el clasificador, inicia con la búsqueda de características, las evalúa y clasifica obteniendo como resultado un conjunto final de características, este es el motor para el entrenamiento de datos y evaluación de datos durante las pruebas determinando la precisión como se observa en la figura:

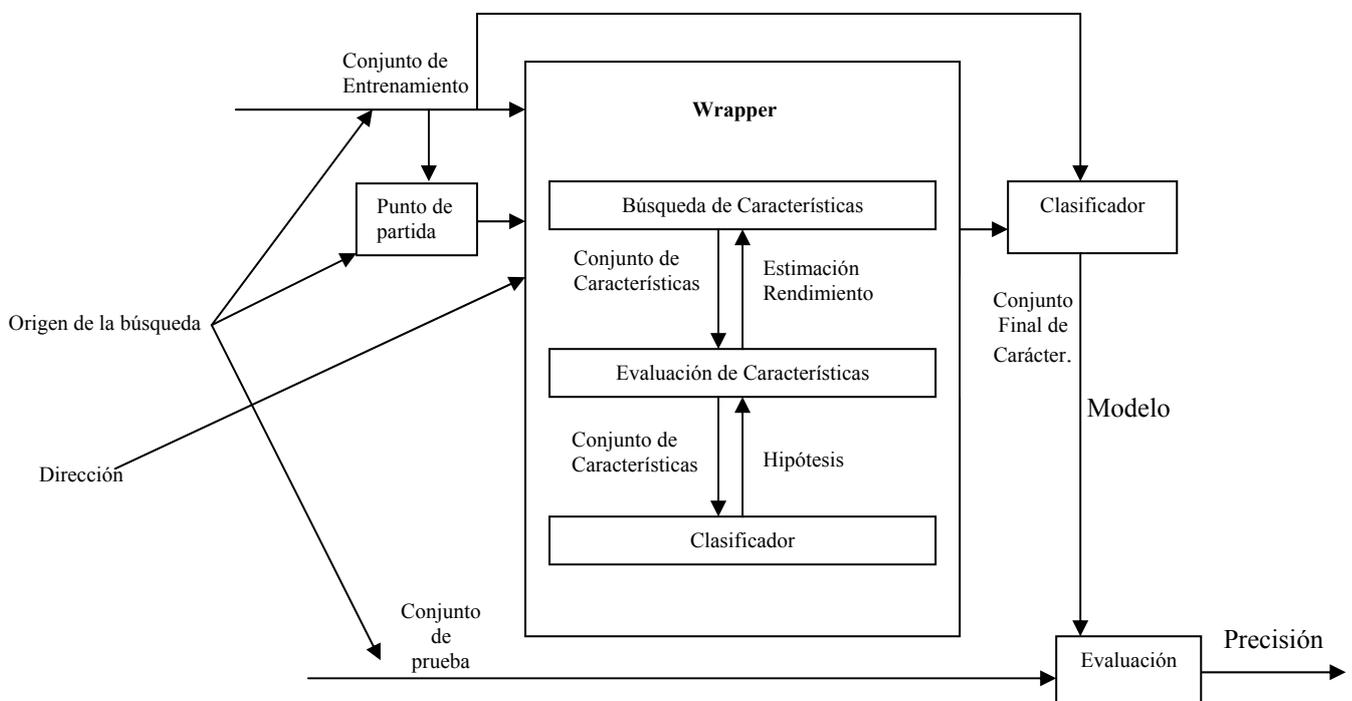


Figura 1.4.1 Clasificador del método Wrapper

Para seleccionar atributos se requiere búsquedas, al momento de usar wrappers o filtros.

Las búsquedas cumplen con las siguientes acciones:

1. Origen de la búsqueda: Se lo puede realizar con todos los atributos, ninguno o de forma aleatoria.
2. Dirección: Es el rumbo que toma la búsqueda esta puede ser hacia adelante (forward) o atrás (backward)
3. Estrategia Control: Es determinar las pautas para un correcto funcionamiento del método wrapper.

CAPITULO II

MÉTODOS PARA EL RECONOCIMIENTO DE ROSTROS

En este capítulo se describirá los métodos existentes para el reconocimiento de rostros, Se mencionan diferentes planteamientos que han aparecido en los últimos años, se realizará un estudio del análisis de componentes principales (PCA) y redes neuronales artificiales.

2.1. INTRODUCCIÓN

En la actualidad existe una gran cantidad de algoritmos de detección de rostros con desempeño variado y dependiente de los escenarios a considerar.

Por este motivo se realiza una jerarquización de los métodos de detección de rostros como se muestra en la siguiente figura:

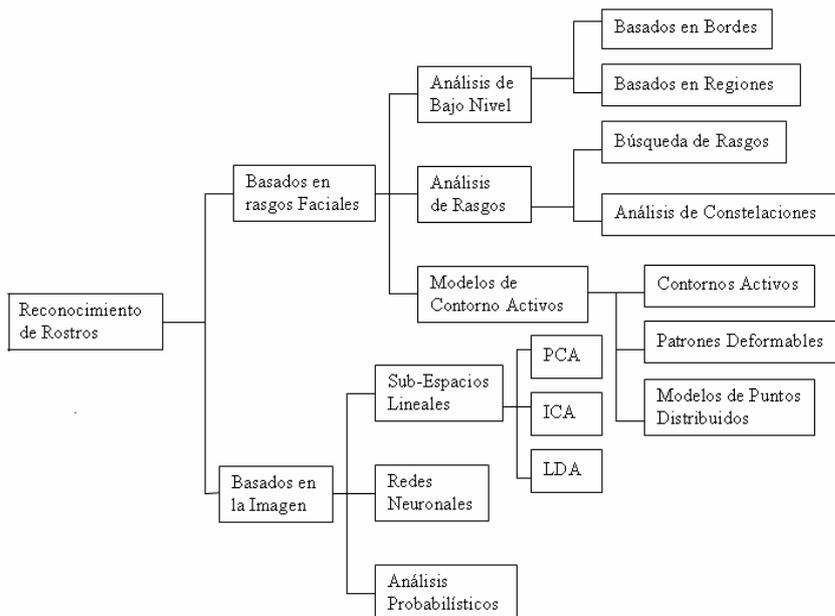


Figura 2.1. Jerarquización de métodos para el reconocimiento de Rostros

Métodos Basados en Rasgos Faciales: buscan encontrar aquellas características presentes en cualquier rostro: ojos, cejas, labios, boca, mentón, líneas de contorno, etc.

Métodos Basados en la Imagen: Aplican herramientas generales de reconocimiento de patrones para sintetizar un modelo a partir de un conjunto de imágenes de entrenamiento. Trabajan con la imagen completa o una región de esta sin buscar rasgos faciales de forma localizada.

2.1.1. MÉTODOS BASADOS EN RASGOS FACIALES

Existen varios métodos para el reconocimiento de rostros, motivados por ideas intuitivas de la representación de un rostro basado en rasgos faciales. Los primeros métodos buscaban resolver el problema de encontrar los rasgos de un rostro en una situación totalmente controlada como fondo blanco, rostro de frente, expresión neutra. La mayoría de ellos se basaban en buscar relaciones geométricas entre puntos característicos hallados mediante heurística⁹. A lo largo de los años, y en particular en la última década creció de manera sustancial el interés por resolver este problema, logrando así un desarrollo muy importante del área. Se pueden definir tres ramas dentro del conjunto de métodos basados en rasgos faciales:

2.1.2. ANÁLISIS DE BAJO NIVEL

Son técnicas que trabajan directamente con los píxeles, principalmente hay dos:

Basados en bordes: buscan bordes, los afinan, etiquetan y finalmente buscan estructuras similares a las de un rostro.

Basados en regiones: aprovechan el hecho de que hay zonas más oscuras que el resto del rostro (cejas, pupilas, etc). Umbralizan y separan la imagen en regiones. Localizan el rostro comparando la distribución de las regiones presentes con la distribución de regiones dentro de un rostro.

⁹ De la Real Academia, heurística: 4. f. En algunas ciencias, manera de buscar la solución de un problema mediante métodos no rigurosos, como por tanteo, reglas empíricas, etc.

2.1.3. ANÁLISIS DE RASGOS FACIALES

Se basan fuertemente en las relaciones geométricas que cumplen los diferentes rasgos representativos en un rostro. Existen dos grandes aproximaciones al respecto:

Búsqueda de rasgos: intentan realizar una búsqueda ordenada de los rasgos característicos de un rostro. Se basan en hipótesis sobre la pose y orientación del rostro y utilizan heurística.

Análisis de Constelaciones: buscan levantar algunas de las hipótesis de los métodos anteriores sobre la pose y orientación del rostro. Se basan en un análisis probabilístico de la ubicación de ciertos puntos característicos de un rostro.

Análisis mediante modelos de contornos activos: buscan adaptar un modelo genérico de un rasgo (ojo, boca, contorno de la cara) a la imagen o porción de imagen en cuestión. Para esto, buscan iterar deformando el modelo hasta adaptarlo al rasgo buscado. Se basan fuertemente en la información local de la imagen (bordes, nivel de gris). Existen tres grandes técnicas:

- Snakes o contornos activos: son comúnmente utilizados para encontrar el contorno del rostro. Se basan en la minimización de una función de energía para adaptar el modelo.
- Patrones deformables: buscan adaptar modelos paramétricos de cada rasgo. Al igual que los Snakes se basan en minimizar una función de energía para la adaptación.
- Modelos de puntos distribuidos (PDM): son una manera compacta de representar de manera paramétrica las formas buscadas. El ajuste de estos modelos se basa en discretizar el contorno PDM en diferentes puntos etiquetados. Las variaciones posibles de estos puntos se guardan en un modelo estadístico realizado cuidadosamente a partir de un conjunto de entrenamiento.

2.1.4. REDES NEURONALES

Se basan en conexiones entre pequeñas y sencillas estructuras (neuronas)¹⁰. En los últimos años ha habido una fuerte tendencia a aplicar este tipo de estructuras a problemas de reconocimiento de patrones. Las redes neuronales necesitan de una etapa inicial de aprendizaje con conjuntos representativos de entrenamiento, para que luego el funcionamiento sea el deseado. Existen modelos de red de distinta complejidad (cantidad de neuronas, tipo de neuronas, cantidad de capas de neuronas, conexiones entre neuronas) con las que se obtienen distintas performances. También existen aproximaciones locales de redes neuronales (no holísticas), cuyo objetivo es encontrar diferentes estructuras locales y luego en base a ellas reconocer el objeto en cuestión.

2.1.5. ANÁLISIS PROBABILÍSTICOS

Se basan en principios fundacionales del reconocimiento de rostros como pueden ser el principio de máxima verosimilitud o la distancia de Kullback-Leibler.

Esta clase de métodos busca estimar distribuciones de probabilidad mediante histogramas y luego comparar los mismos frente a histogramas medios aprendidos estadísticamente de imágenes con caras.

2.2. REDES NEURONALES ARTIFICIALES (ANNs)

Las Redes Neuronales Artificiales (Artificial Neural Networks - ANNs) se constituyeron inicialmente como una simulación abstracta de los sistemas nerviosos biológicos formados por un conjunto de unidades llamadas neuronas que están conectados unos con otros.

Las redes neuronales artificiales¹¹ son modelos estadísticos de sistemas del mundo real los cuales son construidos ajustando un conjunto de parámetros. Estos parámetros, conocidos como pesos, describen formas de trazar un conjunto de valores dados conocidos como entradas hacia su conjunto de valores: las salidas.

¹⁰ www.monografias.com

¹¹ <http://www.fortunecity.com/skyscraper/chaos/279/articulos/RedesNeuronales.txt>

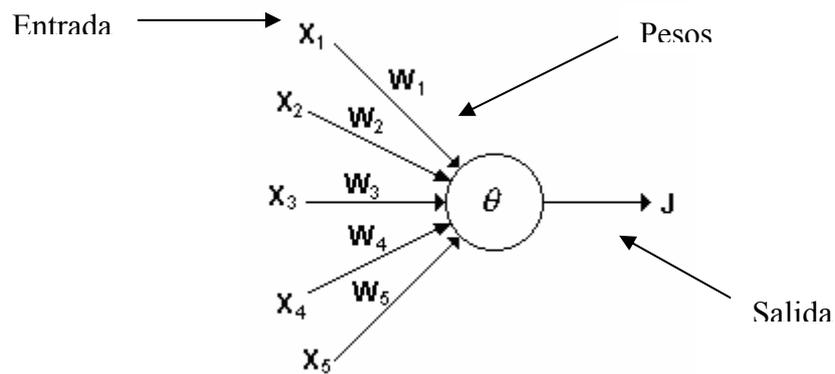


Figura 2.2. Red Neuronal Artificial

Las principales características de las Redes Neuronales Artificiales son las siguientes:

Auto Organización y Adaptabilidad

Estas utilizan algoritmos de aprendizaje en constante adaptación y auto organización lo que permite que los procesos crezcan y se adapten.

Procesado No Lineal y Paralelo

Estas aumentan la capacidad de la neurona para poder aproximar y clasificar información haciéndose más inmune a los datos desordenados.

En este sentido, la red constituida para una aplicación determinada poseerá una arquitectura concreta de elementos de procesos adaptables, masivos y paralelos que se combinan en estructuras de interconexión de red jerárquica.

2.2.1. ESTRUCTURA DE LAS REDES NEURONALES ARTIFICIALES

La estructura de las redes neuronales artificiales se subdivide en dos puntos relevantes los cuales se describirán a continuación:

2.2.1.1. ESTRUCTURA DE CONEXIÓN

Una red neuronal está determinada por la neurona y la matriz de pesos. El comportamiento de la red depende en gran medida del comportamiento de la matriz de pesos. Hay tres tipos de capas de neuronas: la de entrada, las ocultas y la de salida. Entre dos capas de neuronas existe una red de pesos de conexión, que puede ser de los siguientes tipos: Hacia delante, hacia atrás, lateral y de retardo.

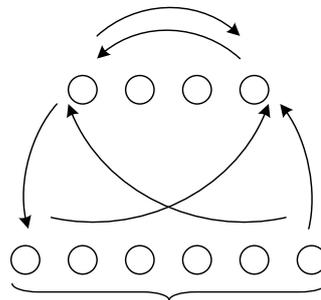


Figura 2.2.1.1. Estructura de Conexión

Conexiones hacia delante: Los datos de las neuronas de una capa inferior son propagados hacia las neuronas de la capa superior por medio de las redes de conexiones hacia adelante.

Conexiones hacia atrás: Los datos de las neuronas de una capa superior son propagados hacia las neuronas de la capa inferior por medio de las redes de conexiones hacia adelante.

Conexiones con retardo: los elementos de retardo se incorporan en las conexiones para implementar modelos dinámicos y temporales, es decir, modelos que precisan de memoria.

2.2.1.2. TAMAÑO DE LAS REDES NEURONALES ARTIFICIALES

FeedBack

El Tamaño de las Redes depende del Número de Capas y del Número de Neuronas Ocultas por Capa.

Número de capas: En una Red Multicapa, hay una o más capas de neuronas ocultas entre la entrada y la salida. El número de capas se cuenta a menudo a partir del número de capas de pesos en vez de las capas de neuronas.

Número de unidades ocultas: El Número de Unidades Ocultas está directamente relacionado con las capacidades de la red. Para que el comportamiento de la red sea correcto se tiene que determinar apropiadamente el número de neuronas de la capa oculta

2.2.2. TAXONOMIA DE LAS REDES NEURONALES ARTIFICIALES¹²

La red neuronal artificial consta de dos fases: la fase de aprendizaje o entrenamiento y la fase de prueba o funcionamiento directo. En la fase de entrenamiento se usa un conjunto de datos o patrones de entrenamiento para determinar los pesos que definen el modelo neuronal. Este modelo una vez entrenado, se usará en la fase de funcionamiento directo en la que se procesarán patrones de prueba que constituyen la entrada habitual de la red con el objetivo de analizar las prestaciones definitivas de la misma.

2.2.2.1. FASE DE APRENDIZAJE

Una de las características más resaltantes de las redes neuronales artificiales es su capacidad de aprender. Éstas aprenden por la actualización o variación de los pesos sinápticos que caracterizan a las conexiones. Los pesos son adaptados de acuerdo a la información extraída de los patrones de entrenamiento nuevos que se van presentando. Normalmente, los pesos óptimos se obtienen de la minimización o maximización de alguna función.

El entrenamiento de las redes neuronales artificiales puede realizarse en tiempo real. La elección de resultados implica velocidades de proceso diferentes que afecta a los algoritmos y hardware utilizados.

¹² Universidad Nacional de San Agustín **Jorge Antonio Delgado Palomino** - j-delgado-p@hotmail.com

2.2.1.2. FASE DE PRUEBA

Una vez obtenidos los pesos de la red neuronal artificial a partir de patrones representativos de entradas que se denominan patrones de entrenamiento. Los pesos pueden ser calculados según el tipo de red neuronal y en función de las ecuaciones dinámicas de prueba. luego se procede a comparar las salidas deseadas con los valores de las neuronas de la última capa para determinar si el diseño de la red es valido o no.

2.2.3. CLASIFICACIÓN DE LAS REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales se clasifican comúnmente en términos de sus correspondientes algoritmos o métodos de entrenamiento: redes no supervisadas, y redes de entrenamiento supervisado.

2.2.3.1. REDES NEURONALES ARTIFICIALES DE ENTRENAMIENTO SUPERVISADO

Las redes de entrenamiento supervisado han sido los modelos de redes más desarrolladas. Los datos para el entrenamiento están constituidos por varios pares de patrones de entrenamiento de entrada y de salida. El hecho de conocer la salida implica que el entrenamiento se beneficia de la supervisión de un maestro. Dado un nuevo patrón de entrenamiento (m+1), los pesos serán adaptados de la siguiente forma:

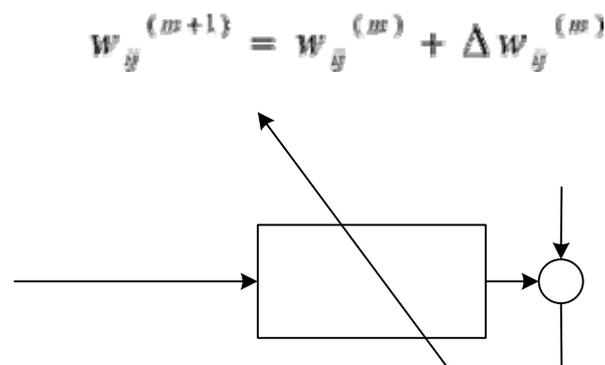


Figura 2.2.3.1. Diagrama esquemático del sistema de entrenamiento

Las redes neuronales artificiales de entrenamiento supervisado constituyen la línea fundamental de desarrollo en este campo. Algunos ejemplos bien conocidos de las primeras redes son red perceptrón, ADALINE/MADALINE, y varias redes multicapa. En el entrenamiento supervisado hay dos fases a realizar: fase de prueba y fase de entrenamiento.

2.2.3.1.1. RED PERCEPTRON

La red conocida como *perceptron simple* es una red neuronal tipo *feed-forward* supervisada, sin capa oculta, constituida por un vector de “*p*” *inputs*, $X=(x_1, x_2, \dots, x_n)$, un vector de “*n*” *outputs* deseados, $X=(y_1, y_2, \dots, y_n)$. La relación entre ambos vectores, (*inputs*; *outputs*) se obtiene mediante la regla de aprendizaje, *perceptron learning rule*. Se demuestra que converge de forma correcta en un número finito de iteraciones (*perceptron convergence theorem*). Si adicionalmente las clases son linealmente separables, permite su utilización en problemas de clasificación con más de una categoría.

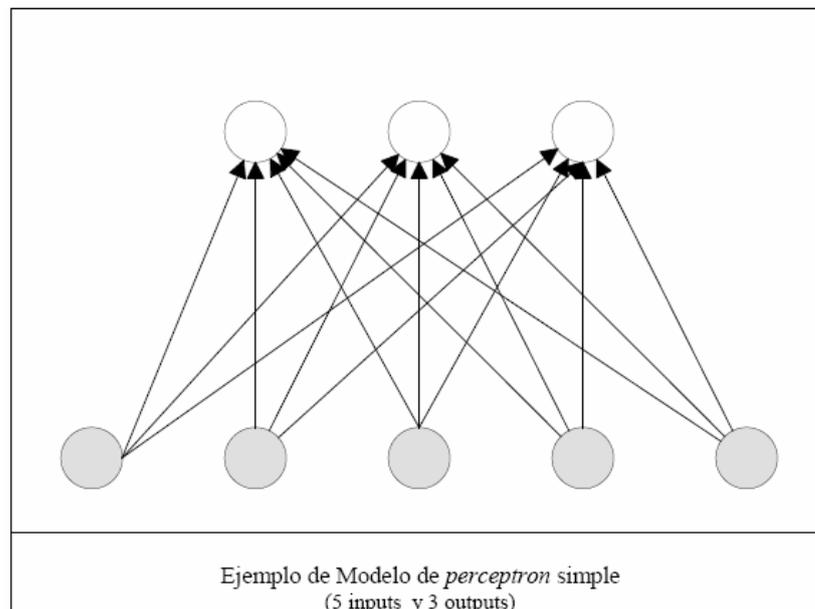


Figura 2.2.3.1.1. Red Perceptron

2.2.3.1.2. RED ADALINE

Un modelo neuronal que representa la relación lineal entre *input* y *output* es la red *Adaline* (*adaptive linear element*). Este modelo utiliza una neurona similar a la del *Perceptron* simple pero de respuesta lineal. Su utilización es posible siempre que los *inputs* sean linealmente independientes, lo cual implica, de forma unidireccional, la condición de separabilidad entre los mismos. El mecanismo que posee para su aprendizaje es la regla de *Windrow-Hoff* o *least mean square* (LMS), que puede considerarse un caso particular de la regla de aprendizaje delta, *delta learning rule*. Ésta última considera como función de activación no lineal la función sigmoidea. La versión multicapa de la red Adaline se denomina *Madaline* y su homólogo en términos de perceptron simple son las redes *multilayer feed-forward*.

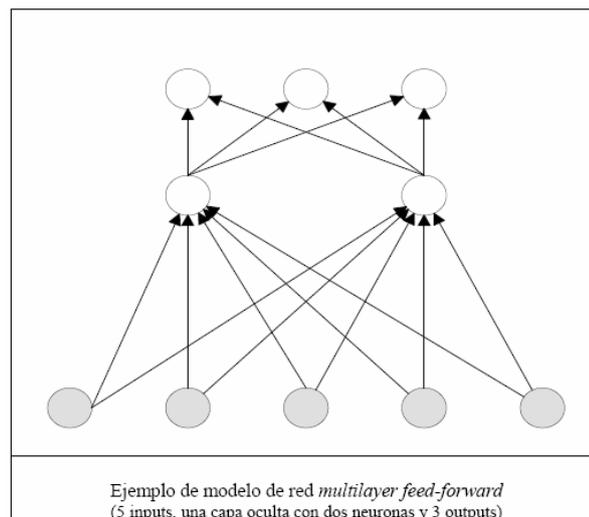


Figura 2.2.3.1.2. Red multilayer feed-forward

2.2.3.1.3. LAS REDES MULTILAYER:

Las redes *Multilayer feed-forward* (MLP) pueden considerarse unos aproximadotes funcionales universales, es decir, una red con una única capa oculta, puede aproximar hasta el nivel deseado dentro de un conjunto compacto cualquier función continua. Las redes multilayer feed-forward son entrenadas habitualmente con el algoritmo de aprendizaje denominado *Backpropagation* o BP, uno de los algoritmos con más importancia histórica

en el desarrollo de las redes neuronales. Las redes neuronales asociadas al algoritmo Backpropagation se definen como *redes back-propagation*.

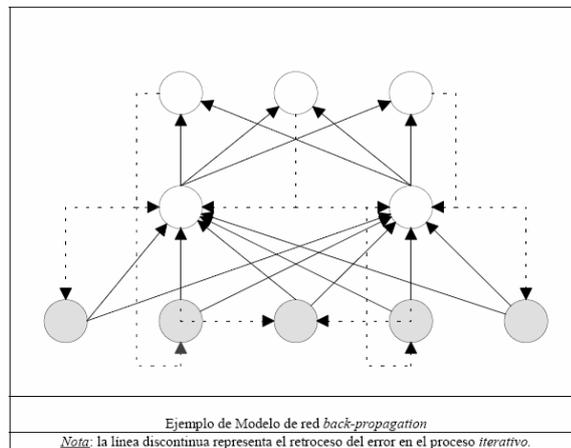


Figura 2.2.3.1.3. Red *back-propagation*.

Los factores que influyen en el proceso de aprendizaje del algoritmo backpropagation (BP) son, entre otros.

1 Los pesos iniciales que son normalmente inicializados de forma aleatoria, pero existen otras posibilidades.

- La constante de aprendizaje, factor de gran importancia en el proceso de convergencia, tanto en lo referente a cómo afecta su valor, como cual es el mejor valor a utilizar en las aplicaciones.
- Las funciones de coste, usualmente se utiliza la función cuadrática. El momentum, filtrado de paso bajo del gradiente del error. Técnicas de optimización utilizadas, métodos de gradiente descendente, método de Newton, método de quasi-Newton, método de dirección conjugada, Aprendizaje y generalización, Número de neuronas en las capas ocultas.
- Muchas aplicaciones requieren que las redes neuronales diseñadas respondan a secuencias en el tiempo de patrones, es decir, series temporales.

2.2.3.2. REDES NEURONALES ARTIFICIALES DE ENTRENAMIENTO NO SUPERVISADO

Las Redes de Entrenamiento no Supervisado utilizan datos de entrenamiento consistentes en sólo patrones de entrada. Por lo tanto, la red es entrenada sin el beneficio de un maestro. La red aprende a adaptarse basada en las experiencias recogidas de los patrones de entrenamiento anteriores.

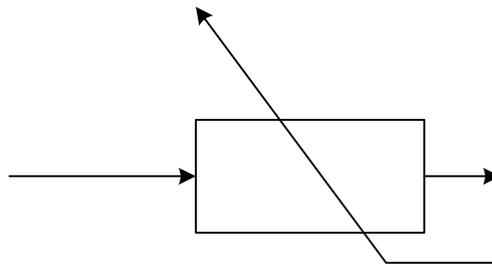


Figura 2.2.3.2. Diagrama esquemático del sistema de entrenamiento

La Regla de Aprendizaje de Hebb: Refuerza el peso que conecta dos nodos que se estimulan simultáneamente.

Regla de Aprendizaje Competitiva: Determina la factibilidad de pertenencia de un patrón a una clase reconocida previamente. En caso de darse esta pertenencia, la inclusión de este nuevo patrón a la clase reconocida cambiará al representante de la misma.

2.2.3.2.1. REDES DE MEMORIA PROASOCIATIVA

2.2.3.2.1.1. MEMORIA ASOCIATIVA LINEAL (LAM)

Una red de memoria asociativa es matemáticamente un mapeado de un espacio de entrada sobre uno de salida. Las redes de memoria asociativa se pueden usar tanto para las aplicaciones autoasociativas como para las heteroasociativas. En las aplicaciones autoasociativas la dimensión del espacio de entrada es igual al de salida. En las

aplicaciones heteroasociativas la dimensión del espacio de entrada y del espacio de salida son en general diferentes. Los valores de entrada y de salida pueden ser reales o binarios.

Una LAM es una red de una capa de propagación. La LAM se deriva de un conjunto de pares de patrones de entrada/salida $\{b^{(m)}, a^{(m)}\}$.

Aquí la entrada es $b^{(m)} = [b_1^{(m)}, b_2^{(m)}, \dots, b_k^{(m)}]^T$

Y la salida es $a^{(m)} = [a_1^{(m)}, a_2^{(m)}, \dots, a_k^{(m)}]^T$

Para $m=1, 2, \dots, M$, donde $[-]^T$ denota la transpuesta del Vector o Matriz. El objetivo de LAM es recuperar el patrón de salida basado en la información total o parcial del patrón de entrada.

Patrones de entrada Continuos

La Matriz de Pesos Sinápticos W en la Memoria Asociativa Lineal se obtiene de la correlación de los pares de patrones originales: $W = \sum a^{(m)} b^{(m)T}$ donde a y b son Vectores Reales Continuos: $a \in \mathbb{R}^N$ y $b \in \mathbb{R}^N$. Las entradas de la matriz se denotan por $W_{ij} = \sum a_i^{(m)} b_j^{(m)}$.

Patrones de Entrada Binarios

Si las entradas son binarias (1/0), entonces los elementos de la matriz de pesos W , de $N \times K$ dimensiones, que se calculan como:

$$W_{ij} = \sum_{m=1}^M (2a_i^{(m)} - 1) (2b_j^{(m)} - 1)$$

Donde $a, b \in \{0, 1\}^k$ (Vectores Binarios). Esta fórmula permite que la condición de ortogonalidad sea impuesta más apropiadamente. Para producir una salida binaria, los elementos del vector Wt se ajustan primero por sus umbrales respectivos:

$$\theta_i = -\frac{1}{2} \sum W_{ij}$$

j=1

Luego son procesados por alguna unidad no lineal en los nodos de salida. Si el valor ajustado es positivo entonces la salida será 1; de cualquier otro modo, la salida será 0.

2.2.3.2.1.2. MEMORIA ASOCIATIVA NO LINEAL PARA LA RECUPERACIÓN HOLOGRÁFICA

La memoria asociativa no lineal es esencial para eliminar las perturbaciones indeseadas. Dado un Patrón de Prueba t , definiendo el Vector Resultado s como el Producto Interno entre $b^{(m)}$ y el Patrón de Prueba t y se escribe como $(b^{(m)}, t)$.

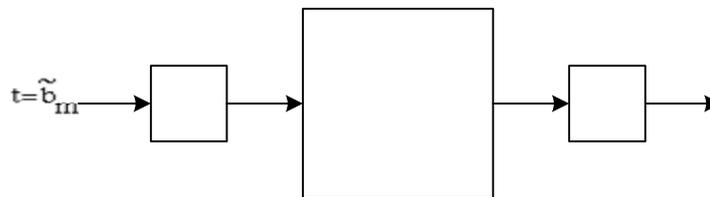


Figura 2.2.3.2.1.2. Memoria asociativa no lineal

$S = [(b^{(1)}, t), (b^{(2)}, t), \dots, (b^{(M)}, t)]$ donde la operación del Producto Interno para una entrada de valor real se define como:

$$(b^{(m)}, t) \equiv b^{(m)T} t \equiv \sum_{i=1}^k b_i^{(m)} t_i$$

Al vector resultado s se le hace un Procesado no Lineal llegando a un Vector de Decisión Binario $V = N \{s\}$ que se espera tenga solo un elemento distinto de cero. Si este elemento se posiciona correctamente, entonces se puede realizar la Recuperación Holográfica. El patrón a recuperar es el Valor de Salida Av. constituida por la matriz formada por los vectores columna $a^{(k)}$.

El propósito del operador NOLINEAL $N \{-\}$ es seleccionar sólo un nodo ganador y simultáneamente descartar todos los otros nodos. El propósito es suprimir el ruido llegando a la Recuperación Holográfica. Los operadores no lineales se pueden manifestar como un elemento de umbral.

Redes de Hamming

Las redes de Hamming son comúnmente utilizadas cuando las entradas son de tipo binario. La red de Hamming selecciona un ganador de entre los patrones almacenados $\{b^{(m)}, m=1, \dots, M\}$, que tienen la menor distancia de Hamming al vector de entrada. Para los vectores bipolares $(-1/1)$ se puede adoptar la misma definición de producto interno introducida anteriormente. Para los valores binarios $(1/0)$ de entrada, el producto interno se tiene que redefinir como:

$$(b^{(m)}, t) = \sum_{i=1}^k (2b_i^{(m)} - 1)(2t_i - 1)$$

Por lo tanto se tiene que:

$$(b^{(m)}, t) = \sum_{i=1}^k (2b_i^{(m)} - 1)(2t_i - 1)$$

En donde la distancia de Hamming es el número de inconsistencias entre los bits de los dos vectores. Esto prueba que en este caso, tanto el valor del producto interno como la distancia de Hamming darán el mismo efecto.

2.2.3.2.1.3. REDES DE MEMORIA RETROASOCIATIVA

Una Red con Realimentación necesita de muchas iteraciones hasta que conseguir la recuperación del patrón final. La Red de Retroasociación más popular es el Modelo de Hopfield el cual que tiene las siguientes características:

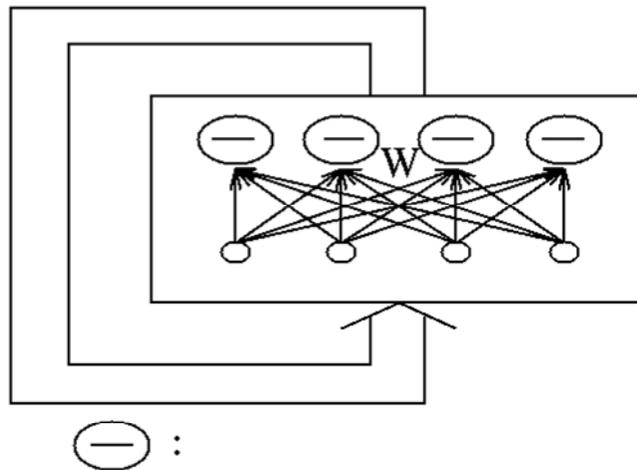


Figura 2.2.3.2.1.3. Memoria Retroasociativa

Los Pesos sinápticos son prealmacenados.

Se usan operaciones no lineales de escalonamiento en cada etapa para producir valores binarios.

La retroalimentación tiene la función de propiciar que los estados se puedan actualizar iterativamente.

Las iteraciones convergen hacia una solución que minimiza una función de energía de la red.

Modelo De Hopfield Secuencial (Asíncrono)

Obtención de los Pesos Sinápticos

Dados M patrones binarios (i.e., $\{a_i^{(m)}\}$ tiene valores binarios 0 o 1), los pesos en el modelo de Hopfield se obtienen de la siguiente forma:

$$W_{ij} = \sum_{m=1}^M (2^{a_i^{(m)}} - 1) (2^{a_j^{(m)}} - 1) \quad i \neq j$$

El umbral de la red se da de la siguiente forma:

$$\theta_i = - \frac{1}{2} \sum_{j=1}^k W_{ij}$$

Funciones de Energía y Convergencia

Unidad

Utilizando la Función de Liapunov como concepto de Función de Energía:

$$E = - \frac{1}{2} \sum_i \sum_j W_{ij} a_i a_j - \sum_i \theta_i a_i$$

Bajo la situación ideal de que los vectores almacenados son perfectamente ortogonales, entonces cada patrón original representa un mínimo local (o global) de la función de energía. Esto motiva que se diseñe la red para que iterativamente se pueda buscar el estado de mínimo local. La técnica del gradiente lleva al modelo secuencial de Hopfield. La diferencia de la función de energía antes y después de la actualización de un estado es:

$$\Delta_k E = E(k+1) - E(k)$$

En caso de una actualización secuencial (asíncrona), hay solo una adaptación de un bit al mismo tiempo. Sin pérdida de generalidad, se asume que sea en:

$\Delta a_i(k+1)$ en el bit i -ésimo:

$$\Delta_k E = - u_i(k+1) \Delta a_i(k+1) - \frac{1}{2} W_{ij}$$

Puesto que $W_{ii} = 0$:

$$\Delta_k E = - U_i(k+1) \Delta a_i(k+1)$$

Se Ingresa una versión discreta del gradiente como:

$$\frac{\Delta_k E}{\Delta a_i(k+1)} = U_i(k+1)$$

Para garantizar el descenso de la Función de Energía $\Delta a_i(k+1)$ se debería actualizar en la dirección de descenso del gradiente:

$$\Delta a_i(k+1) \propto u_i(k+1)$$

Modelo De Hopfield, Algoritmo Secuencial

Suponiendo que la entrada a la red de retroalimentación es a , que se usa como el vector de estado inicial, esto es, se fija $a = a(0) = (a_1(0), a_2(0), \dots, a_N(0))^T$ y las iteraciones inician en

$k=1$ hasta la convergencia. Durante la iteración k -ésima, la red realiza la actualización en orden secuencial desde $i=1, i=2, \dots$, hasta $i=N$ se tiene que:

Cálculo del valor de red

$$U_i(K + 1) = \sum W_{ij} a_j (k) + \theta_i$$

Actualización de los estados

$$a_i (k + 1) = 1 \quad U_i (k + 1) > 0$$

$$0 \quad U_i (k + 1) < 0$$

$$a_i (k) \quad U_i (k + 1) = 0$$

Se repite el mismo proceso para la siguiente iteración hasta la convergencia, lo que ocurre cuando ninguno de los elementos cambia de estado durante alguna iteración.

$$\Delta_k E = - U_i(k+1) \Delta a_i (k+1)$$

2.2.3.2.1.4. RED DE KOHONEN

La idea básica que yace en las SOFM es la incorporación a la regla de aprendizaje competitivo un cierto grado de sensibilidad con respecto al vecindario o la historia. Esto hace que el número de neuronas que no aprenden desaparezca y ayuda a que se destaquen propiedades topológicas que aparezcan en el "mapeado" de características.

Suponiendo que un vector de entrada tiene N características y se representa por un vector x en un espacio de patrones N -dimensional. La red mapea el patrón de entrada hacia un espacio de salida. Kohonen propuso que las neuronas de salida interactuaran lateralmente, llegando así a los *mapas de características autorganizativos*.

La característica más importante del modelo es el concepto de aprendizaje en un vecindario próximo a la neurona ganadora.

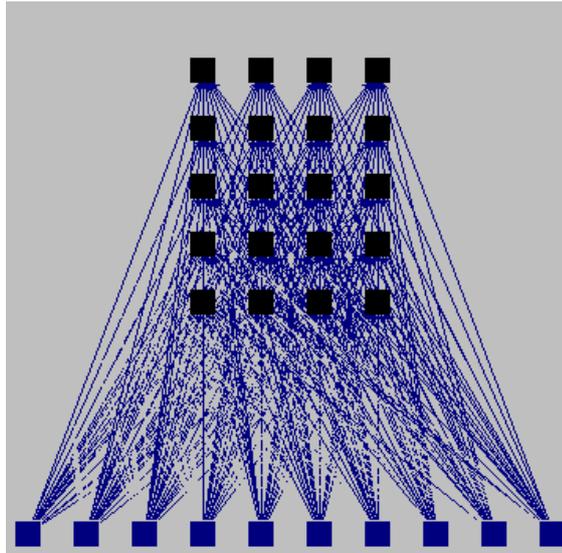


Figura 2.2.3.2.1.4. Red Kohonen

2.3. ANALISIS DE COMPONENTES PRINCIPALES (ACP)

El análisis de componentes principales (ACP)¹³, es una técnica estadística, propuesta a principios del siglo pasado por Karl Pearson como parte del análisis de factores. Sin embargo la complejidad de los cálculos retrasaron 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.

Podría decirse que el objetivo principal que persigue el ACP 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. Dicha representación debe ser tal que al desechar dimensiones superiores, la pérdida de información sea mínima. Las aplicaciones del ACP son

¹³ Oja, E., 1991. "Data compression, feature extraction, and autoassociation in feedforward neural networks" Artificial Neural Networks vol 1 pp.737-746.

numerosas y entre ellas se puede citar la clasificación de individuos, la comparación de poblaciones, la estratificación multivariada, etc.

2.3.1. PRINCIPIOS DEL ACP

Se quiere presentar los fundamentos del ACP¹⁴, tratando de reducir al mínimo el aparato matemático formal que lo sustenta. Se parte con un sencillo ejemplo ficticio en dos dimensiones. Supóngase que se mide el peso en kilogramos y la longitud en centímetros de 20 peces y que se obtienen los siguientes datos:

PEZ	PESO	LONGIT		PEZ	PESO	LONGIT
1	0.5	15		11	3.5	55
2	1.0	30		12	5.0	85
3	2.5	55		13	6.0	75
4	6.5	85		14	4.5	60
5	4.5	70		15	1.5	50
6	1.5	35		16	5.0	75
7	2.5	65		17	7.5	85
8	1.5	45		18	2.5	40
9	3.5	45		19	3.5	65
10	0.5	25		20	4.0	80

Tabla 2.3.1. Datos para el Análisis de Componentes Principales

Los datos anteriores pueden ser dibujados mediante un diagrama de dispersión en un sistema coordenado, obteniéndose una figura como la siguiente:

¹⁴ www.monografias.com

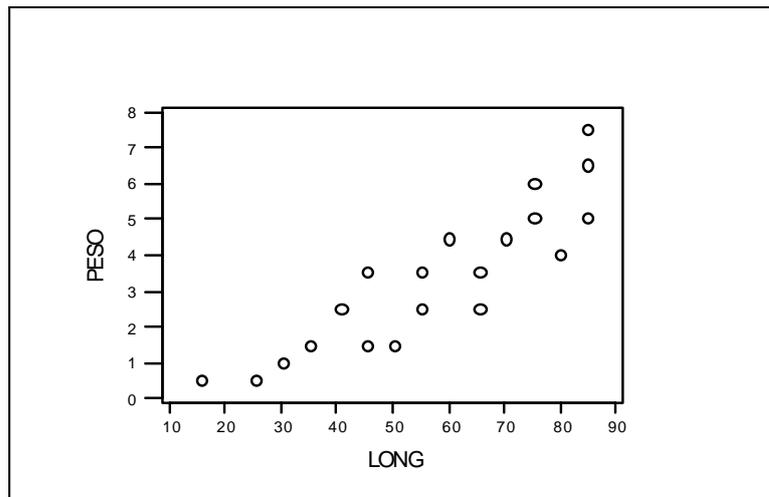


Figura 2.3.1.1. Graficación de los datos

Como se puede apreciar, cada variable puede representarse sobre un eje coordenado y así cada pareja de valores (x_i, y_i) representa las medidas del i-ésimo individuo, los cuales al ser representados en el plano forman la nube de individuos.

Se quiere construir un nuevo sistema de coordenadas ortogonales en el cual los puntos puedan ser representados de una manera tal que sus proyecciones sobre el nuevo primer eje recojan la mayor cantidad posible de variación y las proyecciones sobre el segundo eje recoja el resto de variación. Intuitivamente se encuentra que tales ejes corresponden a las rectas F1 y F2, representadas en la siguiente gráfica cuyo origen se encuentra en el centro de gravedad G de la nube (punto cuyas coordenadas son las medias de las variables consideradas), tal como se ve en la figura siguiente.

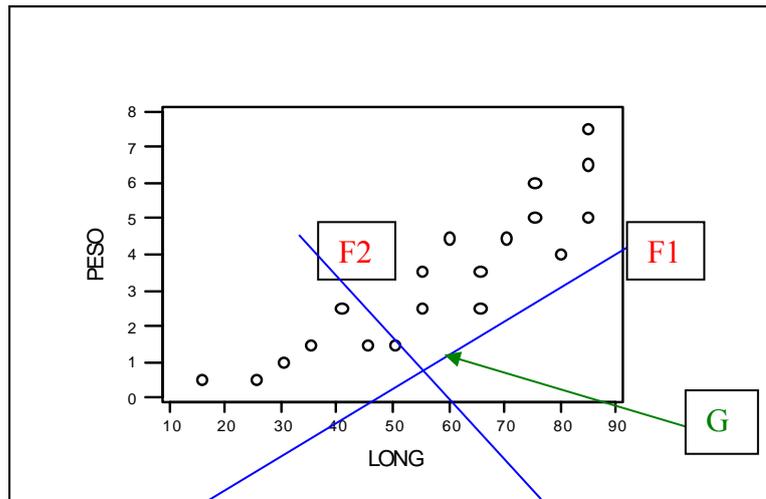


Figura 2.3.1.2. Graficación de Rectas y Gravedad

Con mucha frecuencia se maneja simultáneamente un número p ($p \geq 3$) de variables numéricas. Si cada variable se representa sobre un eje, se necesitaría un sistema de coordenadas rectangulares con p ejes perpendiculares entre sí para ubicar las coordenadas de los puntos y poderlos dibujar. Este dibujo es realmente imposible para el ser humano cuando $p \geq 4$, pero la idea esbozada en el ejemplo anterior sigue siendo válida: buscar un nuevo sistema de coordenadas con origen en el centro de gravedad de la nube de puntos de tal manera que el primer eje del nuevo sistema (F1) recoja la mayor cantidad posible de variación, el segundo eje F2, la mayor cantidad posible entre la variación restante, el tercer eje F3 la mayor posible entre la variación que queda después de las dos anteriores y así sucesivamente.

Observando la figura anterior se puede deducir que el nuevo sistema de coordenadas se logra después de dos movimientos de la nube de puntos: un primer movimiento es una *traslación* que permite situar el nuevo origen en el centro de gravedad de la nube. La nueva nube, obtenida después de esta traslación se llama nube centrada. Un segundo movimiento que se hace sobre la nube centrada es una *rotación*, usando el centro de gravedad como punto pivotal. Esta rotación ha de hacerse de tal manera que el nuevo primer eje del sistema de coordenadas apunte en la dirección de máxima dispersión de la nube centrada, el segundo eje apunte en la dirección con la segunda mayor dispersión perpendicular a la anterior, el tercer eje en la dirección de tercera mayor dispersión perpendicular a las dos

anteriores y así sucesivamente. Es evidente que el nuevo sistema de coordenadas tiene entonces tantos ejes perpendiculares entre sí como tenía el antiguo, es decir, tantos ejes como variables se hayan considerado inicialmente.

En los cursos de álgebra lineal se habla comúnmente de las transformaciones lineales de un espacio vectorial y se demuestra que toda transformación lineal está asociada a una matriz. En particular, las rotaciones de un espacio vectorial son transformaciones lineales del espacio vectorial sobre sí mismo y están asociadas con matrices cuadradas, unitarias y ortogonales. Una matriz de éstas, Q , tiene tantas filas y columnas como sea la dimensión del espacio, sus columnas son vectores unitarios (es decir de longitud igual a la unidad) y tiene la particularidad de que al ser multiplicada por su transpuesta produce la matriz unidad. En otras palabras, $Q^{-1} = Q'$. Las traslaciones no son transformaciones lineales pero tienen la propiedad de no modificar la variabilidad de la nube de puntos. Es decir, las varianzas y covarianzas en la nube son las mismas antes y después de una traslación.

Los resultados expuestos en el párrafo anterior, junto con algunas propiedades de la matriz de varianzas covarianzas Σ , correspondiente a las variables originales y que serán presentadas a continuación, constituyen las bases sobre las cuales descansa la técnica de componentes principales.

Se considera entonces p variables aleatorias de tipo numérico X_1, X_2, \dots, X_p las cuales posiblemente estén correlacionadas entre sí. Se piensa que las p variables anteriores, consideradas conjuntamente, forman un vector aleatorio o variable aleatoria multivariada, denotada por: $\mathbf{X} = (X_1, X_2, \dots, X_p)$

La matriz de varianzas covarianzas para la variable \mathbf{X} está definida como Σ y su entrada en la fila i columna j es el valor de la covarianza entre X_i y X_j . Esto hace que la diagonal de Σ esté conformada por las varianzas $V(X_1), V(X_2), \dots, V(X_p)$ y que Σ sea simétrica. Además se puede probar que es una matriz definida positiva (es decir, la forma cuadrática asociada a ella tiene todas sus raíces positivas). Todas estas propiedades de Σ hacen que esta matriz tenga p valores propios reales y diferentes. Esto en particular

garantiza que Σ sea diagonalizable. Es decir, se puede encontrar una matriz \mathbf{Q} , invertible, tal que $\Sigma = \mathbf{Q}\Lambda\mathbf{Q}^{-1}$ donde Λ es la matriz diagonal formada por los valores propios de Σ , denotados $\lambda_1, \lambda_2, \dots, \lambda_p$

Es posible reordenar de acuerdo con su magnitud los valores propios de Σ de tal manera que λ_1 sea el mayor de ellos, λ_2 el que le sigue, etc y λ_p el menor de todos. Esto simplemente se traduce en un reordenamiento de las columnas de la matriz \mathbf{Q} de manera que la primera sea un vector propio asociado con λ_1 , la segunda un vector propio asociado con λ_2 y así sucesivamente. En particular dichas columnas pueden estar formadas por vectores propios normalizados, es decir, perpendiculares entre sí y de longitud igual a la unidad. De esta manera se construye una matriz que produce la rotación deseada ya que, como puede probarse, el primer vector propio $\mathbf{U}_1 = (u_{11}, u_{12}, \dots, u_{1p})'$ apunta en la dirección de máxima variabilidad de la nube centrada. Esta dirección se llama primera dirección principal. El segundo vector propio $\mathbf{U}_2 = (u_{21}, u_{22}, \dots, u_{2p})'$ apunta en la siguiente dirección de máxima variabilidad de la nube centrada, llamada segunda dirección principal y así sucesivamente.

Una vez resuelto el problema de la rotación, bastará multiplicar la variable centrada $\mathbf{X}_c = \mathbf{X} - \mathbf{G} = (X_1^c, X_2^c, \dots, X_p^c)$ (donde cada fila de \mathbf{G} es el centro de gravedad \mathbf{G}) por la matriz de rotación \mathbf{Q} para obtener la nueva variable $\mathbf{F} = (F_1, F_2, \dots, F_p)$, llamada Variable de Componentes Principales. Cada componente F_i del vector aleatorio \mathbf{F} , se llama una componente principal. Evidentemente se cumple que $F_j = u_{j1}X_1^c + u_{j2}X_2^c + \dots + u_{jp}X_p^c$ es decir, cada componente principal es una combinación lineal de las variables originales centradas.

La traza de Σ , por ser la suma de las varianzas de las variables originales X_i recibe el nombre de varianza total, VT. Resulta claro que $\text{Traza}(\Sigma) = \text{Traza}(\mathbf{Q}\Lambda\mathbf{Q}^{-1}) = \sum_{i=1}^p \lambda_i$. Se

puede probar además que $V(F_i) = \lambda_i$ para $i = 1, 2, \dots, p$ y que $\text{Cov}(F_i, F_j) = 0$. Esto implica varias cosas, a saber:

La varianza total es igual a la suma de los valores propios de Σ e igual a la suma de las varianzas de las componentes principales. Es decir, la varianza total es la misma con las variables originales que con las variables transformadas, F_i

Las componentes principales son variables aleatorias no correlacionadas entre sí obtenidas mediante transformaciones lineales de las variables originales centradas. Esto es:

$$F_j = \mathbf{U}_j \bullet \mathbf{X} = u_{j1}X_1 + u_{j2}X_2 + \dots + u_{jp}X_p \quad \text{para } j = 1, 2, \dots, p$$

Resulta claro que $E(F_j) = 0$ para $j = 1, 2, \dots, p$

Si todas las variables originales X_i son normales entonces todas las componentes principales son normales.

Como puede deducirse de lo anterior, la varianza total se descompone en un número finito de partes disjuntas λ_j de tamaños cada vez menores, lo que en la práctica proporciona un mecanismo para reducir la dimensionalidad de representación de las variables. En efecto, si se descarta las últimas $p - r$ componentes principales, las primeras r tendrán una tasa de

representatividad igual a $\frac{\lambda_1 + \lambda_2 + \dots + \lambda_r}{VT} \times 100\%$ de la varianza total de las variables

originales. Muchas veces este porcentaje es bastante alto con un pequeño valor de r lo que se traduce en una alta representatividad en un espacio de pocas dimensiones.

En la práctica resulta importante el caso $r = 2$ ya que si, en tal caso se obtuviera una tasa de representatividad alta, se habría logrado describir el problema sobre un plano con una pequeña pérdida de información. Por supuesto que si la reducción a un espacio de dos dimensiones conlleva una alta pérdida de representatividad no se habrá logrado un éxito y las técnicas que aquí se propondrán para visualización de individuos y variables no serán muy buenas.

La ecuación $\mathbf{F} = \mathbf{X}_c \mathbf{Q}$ implica $\mathbf{X}_c = \mathbf{FQ}^{-1} = \mathbf{FQ}'$ lo que permite obtener las variables centradas originales como combinaciones lineales de las componentes principales. Esto en

particular va a permitir representar gráficamente las variables originales centradas dentro del espacio de componentes principales, llamado espacio factorial, como puntos cuyas coordenadas son los coeficientes de X_i en la combinación lineal correspondiente. Teniendo en cuenta que sólo las componentes principales iniciales llevan la mayor parte de la representatividad se podrá reducir el espacio factorial a dos o tres dimensiones, lo que lleva a una representación de las variables originales como vectores sobre un plano (plano factorial) o sobre un espacio tridimensional. La representación sobre el plano factorial F_1F_2 es particularmente útil pues permite visualizar relaciones de correlación entre las variables originales y de éstas con los ejes factoriales, lo que rápidamente da una idea de cómo y en cuánto contribuye cada variable a la conformación de los primeros factores (así se llaman también las componentes principales) y qué tan fuertes son las dependencias entre las diferentes variables y los factores. Tal representación plana se llama mapa perceptual de variables. Una alta correlación positiva se traduce en vectores (flechas que unen el origen con el punto representativo de la variable) que forman un ángulo agudo. Una alta correlación negativa se traduce en flechas opuestas que tienden a formar ángulos llanos.. Finalmente, la ausencia de correlación se traduce en flechas que tienden a formar ángulos rectos. Esto sugiere que la correlación entre dos variables se mida a través del coseno del ángulo que ellas forman. Igualmente es factible realizar un mapa perceptual de individuos, es decir, una proyección de la nube de individuos sobre el plano factorial determinado por F_1F_2 , plano que reúne la mayor representatividad de VT.

Puesto que uno de los objetivos que se persiguen con el ACP es la representación de las observaciones o individuos en un espacio de pocas dimensiones, resulta interesante tener una medida de tal representación para cada individuo. Una tal medida está dada por la suma de cosenos cuadrados. Estos valores son los cuadrados de los cosenos de los ángulos formados por el vector que representa a cada individuo con los ejes del sistema de coordenadas factoriales. La suma de todos estos cosenos es igual a la unidad. Sin embargo, si se retienen r factores (componentes), la suma de los r primeros cosenos cuadrados mide el grado de representabilidad de cada individuos, siendo mejor representados aquellos individuos para los cuales la suma de los r primeros cosenos cuadrados esté más cerca de 1.

Las correlaciones entre las variables originales y los factores se conocen comúnmente como cargas factoriales. Es posible calcular analíticamente tales correlaciones lo que da

origen a una matriz \mathbf{L} de orden $p \times p$, llamada matriz de cargas. (algunos paquetes usan otros nombres: Factor Pattern en SAS, Factor Matrix en SPSS, Factor Loadings en Estadística, etc). Se puede probar que la correlación entre la variable original X_i y la componente principal F_j está dada por $l_{ij} = \frac{\sqrt{\lambda_j} u_{ji}}{\sqrt{V(X_i)}}$. Así $\mathbf{L} = (l_{ij})$.

2.4. MÉTODOS BAYESIANOS

Se denomina método bayesiano¹⁵ por basarse originalmente en el teorema de Bayes, publicación póstuma de Thomas Bayes en 1763, que en esencia permite, conocer la probabilidad de que ocurra un suceso, modificar su valor cuando se dispone de nueva información.

Los métodos bayesianos, con una interpretación diferente del concepto de probabilidad, constituyen una alternativa a la estadística tradicional centrada en el contraste de hipótesis. En esencia se diferencian en que incorporan información externa al estudio, con esta información y los propios datos observados estimar una distribución de probabilidad entre la magnitud y el efecto

2.4.1. TEOREMA DE BAYES

Es la probabilidad de que ocurra el suceso cuando es $P(A)$ a la probabilidad de A .
 $P(A.B)$ a la probabilidad de que ocurran los sucesos A y B (ambos).
 $P(A / B)$ a la probabilidad de que ocurra A cuando se sabe que ha ocurrido B (se denomina **probabilidad condicionada**).

La probabilidad de que ocurra A y B es igual a la probabilidad de B multiplicada por la probabilidad de A condicionada a que haya ocurrido B .

$$P(A.B) = P(B) \times P(A / B) = P(A) \times P(B / A)$$

Por simetría es obvio que se cumple la tercera igualdad.

Al tener un conjunto de posibles sucesos A_i ($A_1 \dots A_n$), mutuamente excluyentes se determina que no pueden ocurrir dos de ellos a la vez y que constituyen todas las posibles

¹⁵ <http://www.seh-lilha.org/bayes1.htm#BAYES>

situaciones pueden ser interpretadas de forma que $P(A_1)+P(A_2)+\dots+P(A_n)=1$, el que ocurra alguno de los sucesos A tiene probabilidad 1, suceso seguro. El cuadrado corresponde a todas las situaciones posibles, que en este caso pueden dividirse en tres: A1, A2, A3. El suceso B se puede producir en cualquiera de las tres situaciones.

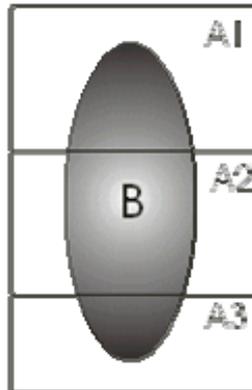


Figura 2.4.1 Teorema de Bayes

Si se reescribe ahora la anterior ecuación por ejemplo para A1 se obtiene

$$P(A_1 \cdot B) = P(A_1/B) \times P(B) = P(B/A_1) \times P(A_1)$$

Aplicando un poco de algebra obtenemos

$$P(A_1/B) = \frac{P(B/A_1) \cdot P(A_1)}{P(B)}$$

$$P(B) = P(B \cdot A_1) + P(B \cdot A_2) + P(B \cdot A_3) = \sum P(B \cdot A_i) = \sum P(B/A_i) \cdot P(A_i)$$

$$P(A_1/B) = \frac{P(B/A_1) \cdot P(A_1)}{\sum P(B/A_i) \cdot P(A_i)}$$

Así se encuentra el Teorema de Bayes el que se utiliza para el reconocimiento de rostros.

2.4.2. METODOLOGÍA BAYESIANA

Con las metodologías estadística clásicas se calcula la probabilidad de observar un resultado suponiendo que la realidad sea de una manera determinada en otras palabras la hipótesis es nula, sin embargo en la práctica se necesitan los conocimientos para tomar decisiones, y lo que realmente interesa es conocer la probabilidad de que las cosas sean de una manera determinada dados los datos por medio de condiciones. Esta es la diferencia que radica en el enfoque bayesiano. En el caso de las pruebas diagnósticas lo que importa en la práctica es el valor, positivo o negativo, de la prueba no la sensibilidad o especificidad de éstas.

Al estar habituados a la presentación de los métodos bayesianos con sucesos binarios, también son aplicables cuando los resultados son continuos. Para la metodología bayesiana es fundamental el concepto de probabilidad a priori o prior, equivalente al conocimiento de la prevalencia en las pruebas diagnósticas. El análisis comienza resumiendo cuantitativamente la información previa existente y externa al estudio, cuyo origen puede ser diverso, desde datos de laboratorio, otros estudios, opinión de expertos o incluso las propias creencias.

Al pretender realizar una comparación y esta cuantificada mediante el logaritmo del odds ratio. Al llamar a esa magnitud que interesa calcular θ . Lo primero que hay que determinar es la distribución de probabilidad de esa magnitud con la información externa de la que se dispone, es lo que se denomina probabilidad a priori y representarlo como $P(\theta)$. Seguidamente se cuantifica la información mediante lo que se denomina función de verosimilitud o likelihood, que se presentará como $P(\text{Datos} / \theta)$. La verosimilitud representa la probabilidad de los datos observados para cualquier valor del parámetro θ . Se utilizará el teorema de Bayes para actualizar el valor a priori $P(\theta)$. A la luz de los datos obtenidos y calcular $P(\text{Datos} / \theta)$, la denominada función de probabilidad a posteriori, es decir cómo de probables son los diferentes valores posibles de θ una vez obtenidos nuestros datos. Según el teorema de Bayes muestra que:

$$P(\theta/\text{Datos}) \propto P(\theta) * P(\text{Datos}/\theta)$$

Donde el símbolo \propto indica que el lado de la izquierda es proporcional al lado de la derecha, es decir que son iguales salvo por un término constante llamado denominador del teorema de Bayes que no depende del parámetro θ .

Así pues los resultados se expresan como una función de la probabilidad a posteriori de los diferentes valores de θ . Al realizar una comparación.

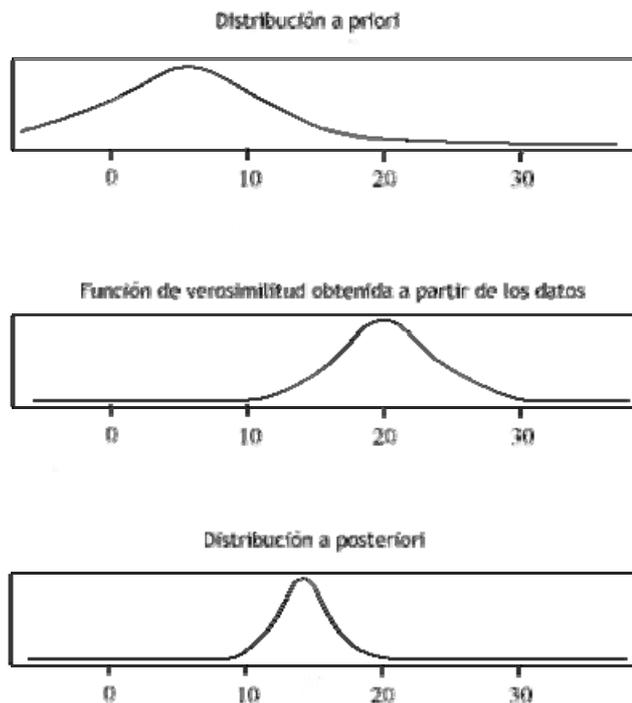


Figura 2.4.2 Metodología Bayesiana

En la figura 2.4.2. La curva de la parte superior representa la distribución de probabilidad que se espera para la diferencia entre las medias a priori, de acuerdo con la información de la que se dispone. La curva de la parte central representa la función de verosimilitud obtenida a partir de los datos del estudio: probabilidad para cada posible valor de esa diferencia en función de los datos. Por último la curva de la parte inferior corresponde a la distribución de probabilidad a posteriori obtenida al combinar ambas curvas, la correspondiente a la información previa y la obtenida en nuestro estudio.

Se denomina Intervalo de credibilidad a quien garantiza e incluye el verdadero valor de θ con una probabilidad dada.

También puede utilizarse la distribución a posteriori para evaluar la probabilidad de que θ tenga un valor igual o mayor que un valor concreto para tomar una decisión.

2.4.3. CÓMO CUANTIFICAR LA INFORMACIÓN A PRIORI

En la cuantificación de la distribución de probabilidad a priori radica el principal punto de controversia de los métodos bayesianos implica una pérdida de objetividad.

Hay que tener presente que el término *a priori* no implica necesariamente una relación temporal en el sentido de que corresponda a una información obtenida con anterioridad. Existen diferentes procedimientos para formalizar la distribución de probabilidad a priori. Es lo que se conoce como análisis de sensibilidad.

Desde el punto de vista probabilístico o matemático de la cuestión existen tres métodos fundamentales para establecer la distribución de probabilidad a priori: distribución no informativa o de referencia, que corresponde a una ausencia de opinión a priori y por lo tanto no aporta información a lo que se observa en los datos. Distribución a priori escéptica, que considera que la probabilidad de que la hipótesis alternativa sea cierta o exista diferencia entre los grupos, es muy pequeña. Y distribución a priori entusiasta, que tiene razones fundadas para encontrar diferencias, por lo que determina que la probabilidad de que éstas sean 0 o peor en el grupo de interés tiene una probabilidad muy baja.

El problema radica en que la especificación y cuantificación de la distribución a priori no es una tarea sencilla, especialmente cuando se trata de modelos con más de un parámetro, como pueden ser los modelos de regresión. Por otro lado existe una cierta reticencia por parte de los investigadores a incorporar una distribución a priori con suficiente información, por temor a la posibilidad de que se les acuse de subjetividad.

2.5. EIGENFACES

Las *eigenfaces* permiten encontrar un espacio vectorial común óptimo de un conjunto de caras; luego se estima la distancia a ese espacio de cualquier cara a reconocer.

Eigenfaces es una técnica de representación para reducción de dimensionalidad y generalización.

Normaliza y selecciona una región de lo rostro en varias imágenes, trata cada sub-imagen de lo rostro como un vector de intensidad y encuentra momentos de los vectores en una base de datos.

El reconocimiento funciona según principios tales como “eigenfaces” o “eigenfeatures”. El eigen alemán se refiere en este caso a las matemáticas usadas para analizar características faciales únicas. Un sistema *eigenface_based* considera cada imagen facial como conjunto en dos dimensiones de las áreas claras y oscuras dispuestas en un modelo determinado. El algoritmo del reconocimiento salva cada imagen como una combinación de eigenfaces y después compara las características del eigenface de la cara actual con éstas en la base de datos.

Un sistema *eigenfeature_based* se centra en características específicas tales como la nariz, ojos, boca, cejas, y curvaturas del hueso, y las distancias relativas entre ellas. El sistema analiza la cara actualmente explorada y extrae eigenfeatures determinados, entonces compara éstos con otros análisis en la base de datos. Los sistemas de Eigenfeature trabajan típicamente en conjunto con sistemas del eigenface para producir la identificación más exacta posible.

En general, los sistemas del eigenfeature son más exactos en identificar caras a pesar de variaciones substanciales tales como barbas y anteojos.

2.6. FISHERFACES

Una variación del Análisis de Componentes Principales denominada “Fisherfaces” fue propuesta en P.Belhumeur, como una mejora de la primera. Teniendo en cuenta que el conjunto de entrenamiento está etiquetado, es posible utilizar esta información para reducir la dimensionalidad del espacio de características. Aplicando el Análisis Discriminante Lineal (Linear Discriminant Analysis) de Fisher, es posible construir una matriz de proyección en la cual la razón entre la dispersión intra-clase y la inter-clase sea máxima. Los resultados muestran que tanto PCA como LDA o Fisherfaces, obtienen un buen rendimiento si las imágenes de prueba son similares a las imágenes de entrenamiento. Sin embargo, los experimentos desarrollados, muestran que el método LDA obtiene mejores resultados en caso de que haya variaciones en las condiciones de iluminación y de gesto con el 15.3% de reconocimiento incorrecto para PCA frente al 7.3% por el método de Fisherfaces.

En P.Belhumeur se afirma que los autovectores que corresponden a los autovalores de mayor valor a menudo contienen información relativa a las variaciones de iluminación, más que a la identidad de los individuos. Por lo tanto, si estos autovectores se descartan habitualmente los tres primeros, entonces la proyección de la imagen en los restantes autovectores debería tener unos valores de los cuales se han eliminado los cambios debidos a variaciones en la iluminación. Sin embargo, tanto se comprueba que al eliminar esos autovectores el rendimiento del sistema decae. Tal vez estos tres autovectores no sólo codifican información de las condiciones de iluminación, sino que también contengan información útil para el reconocimiento.

2.7. METODOS HIBRIDOS

Son métodos conformados con principios de otros métodos para el análisis de componentes los que se describe a continuación:

2.7.1. MÉTODOS BASADOS EN KERNELS

Este grupo de métodos son una generalización a los métodos de análisis de componentes como por ejemplo análisis de componentes principales (PCA)¹⁶, análisis de componentes independientes (ICA), análisis lineal discriminante (LDA), etc. En los métodos de componentes, se construye un subespacio que cumpla determinadas restricciones y luego se elige una base que lo genere de alguna manera particular los componentes de mayor potencia, mejor discriminación. El Método de Kernels es poder tener en cuenta momentos de mayor orden sin tener un costo computacional excesivamente grande. Buscan llevar el problema de clasificación a un espacio de mayor dimensión donde las clases sean linealmente separables. Para esto se realiza lo siguiente:

- Se mapean los vectores de entrenamiento a través de una función no lineal que lleva los puntos a un espacio de mayor dimensión.
- Se plantea un problema equivalente al problema de PCA en dicho espacio. Existen algunas variantes que intentan resolver el problema de ICA o LDA en su lugar.
- Se resuelve el problema equivalente, utilizando el **Kernel Trick**, que es una manera simplificada de resolver el problema de PCA en el espacio de mayor dimensión. Si el mapeo no lineal cumple determinadas condiciones particulares, se pueden realizar todos los cálculos de la resolución del problema equivalente, sin necesidad de mapear los vectores en el espacio de mayor dimensión. Para esto existen diferentes funciones, llamadas núcleos que posibilitan dicho atajo. Entre ellas se destacan los núcleos **gaussianos** y **polinómicos**.

¹⁶ Ming-Hour Yang. Face recognition using kernel methods, NIPS'01, volume 14, pages 1457–1464, Vancouver, British Columbia, Canada, December 3-8 2001, MIT Press.

2.7.2. EVOLUTIONARY PURSUIT (EP)

Este método, al igual que PCA, ICA y LDA se basa en el análisis de componentes. EP plantea una manera novedosa de obtener una base de vectores eficiente para la representación de las imágenes de rostros. Para encontrar la base, se realiza una búsqueda de manera de maximizar una función **fitness** que mide al mismo tiempo la precisión de la clasificación y la habilidad de generalización del sistema. Como el problema de buscar la base óptima es un problema de dimensión alta se utiliza un tipo de Algoritmos Genéticos (GA) al cual se le llama **Evolutionary Pursuit**¹⁷.

Para encontrar las EP-caras se realiza lo siguiente:

- Se reduce la dimensión de los datos mediante PCA.
- Se transforma el espacio anterior mediante una transformación **deWhitening**.
- En el espacioWhitened-PCA, se realiza el siguiente procedimiento hasta cumplir un número máximo de iteraciones o la actuación buscada.
 - Se realizan varias rotaciones entre pares de vectores de una base del espacio y luego se selecciona un conjunto de ellos. Se codifica la rotación efectuada mediante representación en palabras de bits, utilizando 10 bits para cada ángulo rotado.
 - Se calcula la función fitness para medir la precisión y la generalización del sistema. Precisión hace referencia al buen desempeño del algoritmo hasta ahora, mientras que por generalización se entiende la performance esperada en pruebas futuras.
 - Se calculan los ángulos y los vectores que maximizan la función fitness y se guardan como la mejor solución hasta el momento.
 - Se itera a un nuevo subconjunto de ángulos y por lo tanto de vectores rotados, mediante Algoritmos Genéticos.

¹⁷ Chengjun Liu and Harry Wechsler. Evolutionary pursuit and its application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(6):570–582, 2000.

- Con la base óptima hallada se realiza reconocimiento de caras siguiendo un procedimiento análogo al de los métodos por subespacios lineales.

2.7.3. SUPPORT VECTOR MACHINE (SVM)

Es una herramienta genérica para resolver problemas de reconocimiento de patrones planteada en la década pasada por **Cortes y Vapnik**¹⁸. Dado un conjunto de puntos en un determinado espacio que pertenecen a dos clases distintas, SVM encuentra el hiperplano que separa la mayor cantidad de puntos de la misma clase del mismo lado. Esto se realiza maximizando la distancia de cada clase al hiperplano de decisión denominado OSH (**Separación Óptima de Hiperplano**). Los puntos más cercanos al hiperplano, de cada conjunto en cuestión, son los llamados vectores soporte. Para aplicar SVM a un problema particular es necesario reducir la dimensionalidad de los datos para lo que se utilizan herramientas como PCA o LDA.

En general se obtienen resultados superiores a los métodos clásicos de análisis de componentes y también se logra cierta robustez frente a cambios en la iluminación.

2.7.4. ACTIVE APPEARANCE MODEL (AAM)

Active Appearance Model¹⁹ (AAM) es un modelo estadístico de la forma y la apariencia en niveles de gris del rostro que se puede generalizar a casi cualquier ejemplo válido de dicho rostro. Ajustar dicho modelo a una imagen implica encontrar los parámetros del modelo que minimizan la diferencia entre la imagen y una síntesis del modelo, proyectado en la imagen. La gran cantidad de parámetros presentes en el modelo hacen de éste un problema complicado. El modelo se genera combinando modelos de variación de la forma y la apariencia de la cara. Estos se construyen a partir de imágenes de prueba donde se marcan puntos de interés, puede ser en forma manual o automática.

¹⁸ G. Guo, S. Li, and K. Chan. Face recognition by support vector machines. In Proceedings of the International Conferences on Automatic Face and Gesture Recognition, FG '00, pages 196–201, 2000.

¹⁹ Gareth J. Edwards, Timothy F. Cootes, and Christopher J. Taylor, Face recognition using active appearance models. In Proceedings of the 5th European Conference on Computer Vision

Los modelos de forma y apariencia, se generan considerando distintos puntos marcados en la imagen como vectores y aplicando PCA a dicha información. Cada ejemplo de forma y apariencia puede entonces ser representado por los coeficientes correspondientes. Dado que existe correlación entre la variación de la forma y la apariencia, se aplica PCA a la concatenación de los vectores de los modelos anteriores y con ello se construye el modelo final.

Teniendo un modelo combinado de forma y apariencia y una posición inicial aproximada, se propone un algoritmo para ajustar los parámetros del modelo y sintetizar un rostro lo más similar posible a la original. Aplicado al Reconocimiento de rostros para cada imagen conteniendo un rostro se ajusta un modelo de forma y apariencia y los coeficientes resultantes se toman como representantes de la identidad tanto para el registrado como para el reconocimiento. Se utiliza para dicho ajuste información extra, proveniente de la relación existente entre la imagen error y el error en los parámetros del modelo.

Dentro de las ventajas se destaca poder obtener representación de vistas no conocidas del rostro a partir de la interpolación de poses conocidas. Como aspecto negativo se menciona la complejidad en la implementación. Además los resultados son fuertemente sensibles a la correcta selección de los puntos principales.

2.7.5. MODELADO 3D

La idea central de estas técnicas es ajustar un modelo genérico 3D a cada imagen en cuestión. Para esto es necesario contar con imágenes 3D. Existen distintas técnicas de adquisición de imágenes 3D, entre ellas: cámaras estéreo, escaners láser, SLS (Structure Light System), secuencia de imágenes 2D.

Primero mediante una serie de imágenes 3D de unos pocos individuos se construye un modelo de cara genérico 3D. Luego dicho modelo es ajustado de manera iterativa, para cada individuo de la base completa. Finalmente se registra las variaciones respecto al modelo genérico y a su vez alguna información de textura.

Los métodos basados en modelos 3D son menos sensibles a variaciones en la pose o en cambios en la iluminación que otros métodos. Su principal contrapartida es que necesitan

imágenes de adquisición compleja, además de ser extremadamente costosos computacionalmente.

2.7.6. ELASTIC BUNCH GRAPH MATCHING (EBGM)

Este método consiste en extraer una representación del rostro en forma de grafo y el reconocimiento se realiza comparando los grafos correspondientes a las distintas imágenes. Se define un conjunto de puntos principales, como por ejemplo las pupilas, esquinas de la boca, etc. Un grafo etiquetado representando una cara consiste en lo siguiente: N nodos ubicados en los puntos principales y las aristas que se forman entre parejas de nodos. Cada nodo es etiquetado con los denominados **jets** y cada arista es etiquetada con la distancia entre los nodos correspondientes.

Los jets se basan en una transformada **wavelet** definida como la **convolución** de la imagen con una familia de núcleos de **Gabor** con distintas frecuencias y orientaciones.

Para encontrar los puntos principales en nuevos rostros se necesita una descripción general del objeto rostro. Dicha descripción debe cubrir una amplia gama de variaciones como son diferentes formas de ojos, boca, diferentes sexos, razas, etc. Para ello se combina un grupo representativo de grafos modelo en una estructura del tipo stack llamada **Face Bunch Graph**²⁰ (FBG).

Cada grafo modelo tiene la misma estructura y los nodos refieren a los mismos puntos principales. El objetivo en una imagen de prueba es encontrar los puntos principales y obtener el grafo que maximice la similitud con el FBG considerado. El grafo resultante es la representación de la cara presente en la imagen.

Luego de extraído el grafo de la imagen de prueba, se utiliza una función de similitud para evaluar la similitud de dicho grafo con los grafos de la base de datos. Un criterio de decisión puede ser tomar el máximo valor de similitud como cara encontrada.

²⁰ Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching.

Si bien es necesaria una etapa de marcado manual de puntos para la generación del modelo estadístico, el algoritmo logra buenos resultados con una única imagen por persona. El problema principal de este algoritmo es el costo computacional de convolucionar a la imagen con una familia de núcleos de Gabor, lo cual implica tiempos de procesamiento elevados.

2.7.7. MODELOS DE MARKOV ESCONDIDOS (HMM)

Se trata de un conjunto de modelos estadísticos utilizados para caracterizar las propiedades estadísticas de una señal. Estos modelos son de gran utilidad para la representación de dependencias estadísticas en problemas que tienen una temporalidad inherente, por lo que han tenido éxito en aplicaciones como reconocimiento de voz y de gestos. Para el caso de imágenes, las dependencias temporales se transforman en dependencias espaciales, por lo que es necesario encontrar una forma de recorrer una señal que en vez de ser unidimensional, como es en el caso de la voz, sea de dos dimensiones.

Formalmente un HMM²¹ consiste en los siguientes procesos interrelacionados:

- Un proceso subyacente, modelado por una cadena de Markov no observable con un número finito de estados, una matriz de transición de estados y una distribución de probabilidad de estados inicial.
- Un proceso observable dado por los vectores de observación y relacionado por un conjunto de densidades de probabilidad asociadas a cada estado.

Para aplicar HMM al reconocimiento de rostros, lo que se hace es dividir la imagen en bloques y se aplica la DCT o KLT. Luego ciertos coeficientes de la transformada del bloque se utilizan como los vectores de observación.

Se modela el rostro por una cadena de Markov, la cual se entrena a los efectos de encontrar los parámetros, utilizando el algoritmo EM maximizando $P(O/\lambda)$, siendo O el vector de observación y λ la cadena de Markov. De esta forma se obtiene una cadena λ para cada individuo de la base. Luego para tomar la decisión se elige la cadena λ^* que maximice

²¹ A. Nefian. A hidden Markov model-based approach for face detection and recognition

$P(O/\lambda_i)$. Existen distintos modelos para representar el rostro, a los efectos de recorrer la señal bidimensional:

HMM top-down.- Se toman bloques que se corresponden con las distintas regiones horizontales de la cara desde arriba hacia abajo. De esta manera los bloques corresponden a la frente, los ojos, la nariz, la boca y el mentón.

HMM embebidas.- Se hace una cadena de cadenas, donde una de nivel inferior recorre la imagen en sentido horizontal y luego una de nivel superior recorre la imagen en sentido vertical.

La dificultad en este tipo de algoritmos se centra en el entrenamiento de las cadenas. Además es costoso desde el punto de vista computacional.

2.7.8. LOCAL BINARY PATTERNS (LBP)

El operador original LBP es una herramienta interesante como descriptor de textura. Este operador recorre la imagen y etiqueta los píxeles de la misma, umbralizando una vecindad de 3×3 respecto al valor del píxel en el que se encuentra, considerando el resultado como un número binario. Luego el histograma de las etiquetas puede utilizarse como descriptor de textura.

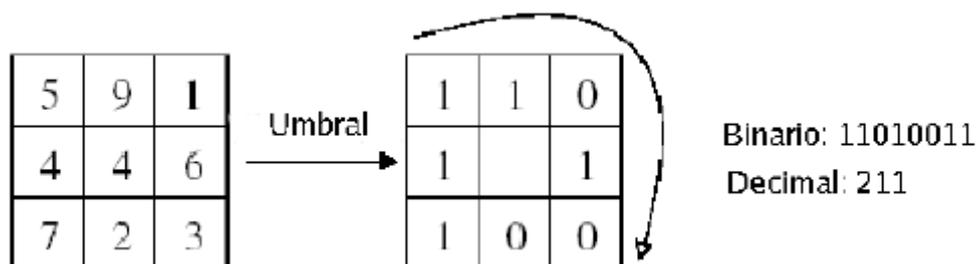


Figura 2.7.8. Operador LBP

En la figura se muestra el funcionamiento del operador LBP básico. Algunas variantes al operador básico fueron introducidas posteriormente, una de ellas es el hecho de considerar entornos de tamaño variable.

Usando vecindades circulares e interpolando los valores de los píxeles en donde cualquier radio es válido. Otra extensión es el tomar en cuenta solamente los patrones uniformes, siendo éstos los que tienen al menos dos transiciones 0-1 o 1-0. Esta idea se apoya en el hecho de que la mayor parte de los patrones en una imagen son de tipo uniforme.

El histograma obtenido mediante este procedimiento contiene información acerca de la distribución local de los micropatrones, como bordes, puntos y otros, sobre la imagen completa. Para tener una representación eficiente de la cara es necesario además almacenar información espacial. La imagen se divide en regiones uniformes por medio de una grilla. Luego se calculan los histogramas para cada bloque y se concatenan todos para obtener el patrón del rostro. De esta forma se logra efectivamente una descripción de la cara en tres escalas o niveles: las etiquetas contienen información de los patrones a nivel de los píxeles, los histogramas en una pequeña región producen información a nivel regional y los histogramas regionales son concatenados para crear una descripción global de la cara.

Como aspecto más positivo se resalta que es una representación simple y eficiente de la cara la cual logra obtener buenos resultados.

CAPITULO III

ALGORITMOS GENÉTICOS

En este capítulo se explicará sobre los algoritmos genéticos, métodos y técnicas que utiliza, sus ventajas y limitaciones, la importancia que tiene para el reconocimiento de rostros, los campos en los que son utilizados, y las funciones de aprendizaje que poseen.

3.1. INTRODUCCIÓN

Un algoritmo genético ²²o AG es una técnica de programación que imita a la evolución biológica como estrategia para resolver problemas. Dado un problema específico a resolver, la entrada del AG es un conjunto de soluciones potenciales a ese problema, codificadas de alguna manera, y una métrica llamada función de aptitud que permite evaluar cuantitativamente a cada candidata. Estas candidatas pueden ser soluciones funcionales, con el objetivo de que el AG las mejore, pero se suelen generar aleatoriamente.

El AG evalúa cada candidata de acuerdo con la función de aptitud. De un conjunto de candidatas generadas aleatoriamente, la mayoría no funcionarán, y serán eliminadas. Pero unas pocas pueden ser prometedoras, pueden mostrar actividad aunque sólo sea actividad débil e imperfecta, hacia la solución del problema.

Estas candidatas prometedoras se van a conservar y se les permitirá reproducirse. Se harán múltiples copias de ellas, pero estas copias no van a ser perfectas; se van a introducir cambios aleatorios durante el proceso de copia. Luego, esta descendencia digital prosigue con la siguiente generación, formando un nuevo acervo de soluciones candidatas, y son sometidas a una ronda de evaluación de aptitud.

²² Holland, John. "Genetic algorithms." *Scientific American*, julio de 1992, p. 66-72.

Las candidatas que han mejorado con los cambios en su código son eliminadas; pero de nuevo, al azar, las variaciones aleatorias introducidas en la población pueden haber mejorado a algunos individuos, convirtiéndolos en mejores soluciones del problema, más completas o más eficientes. De nuevo, se seleccionan y copian estos individuos vencedores hacia la siguiente generación con cambios aleatorios, y el proceso se repite.

Las expectativas son que la aptitud media de la población se incrementará en cada ronda, por lo tanto, al repetirse este proceso cientos o miles de rondas, pueden descubrirse soluciones muy buenas del problema.

Los algoritmos genéticos han demostrado ser una estrategia enormemente poderosa y exitosa para resolver problemas, demostrando de manera espectacular el poder de los principios evolutivos. Se han utilizado algoritmos genéticos en una amplia variedad de campos para desarrollar soluciones a problemas difíciles para los diseñadores humanos.

3.1.1. MÉTODOS DE REPRESENTACIÓN

Para que un algoritmo genético pueda ponerse a trabajar en un problema, se necesita un método para codificar las soluciones potenciales del problema de forma que una computadora pueda procesarlas. Un enfoque común es codificar las soluciones como cadenas binarias: secuencias de 1s y 0s, donde el dígito de cada posición representa el valor de algún aspecto de la solución.

Otro método similar consiste en codificar las soluciones como cadenas de enteros o números decimales, donde cada posición representa algún aspecto particular de la solución. Este método permite una mayor precisión y complejidad que el método de sólo usar números binarios y a menudo, está intuitivamente más cerca del espacio de problemas.

Un tercer método consiste en representar a los individuos de un AG como cadenas de letras, donde cada letra representa un aspecto específico de la solución.

La virtud de estos tres métodos es que facilitan la definición de operadores que causen los cambios aleatorios en las candidatas seleccionadas: cambiar un 0 por un 1 o viceversa, sumar o restar al valor de un número a una cantidad elegida al azar, o cambiar una letra por otra.

Otra estrategia desarrollada principalmente por John Koza, de la Universidad de Stanford, y denominada programación genética, representa a los programas como estructuras de datos ramificadas llamadas árboles. En este método, los cambios aleatorios pueden generarse cambiando el operador o alterando el valor de un cierto nodo del árbol, o sustituyendo un subárbol por otro.

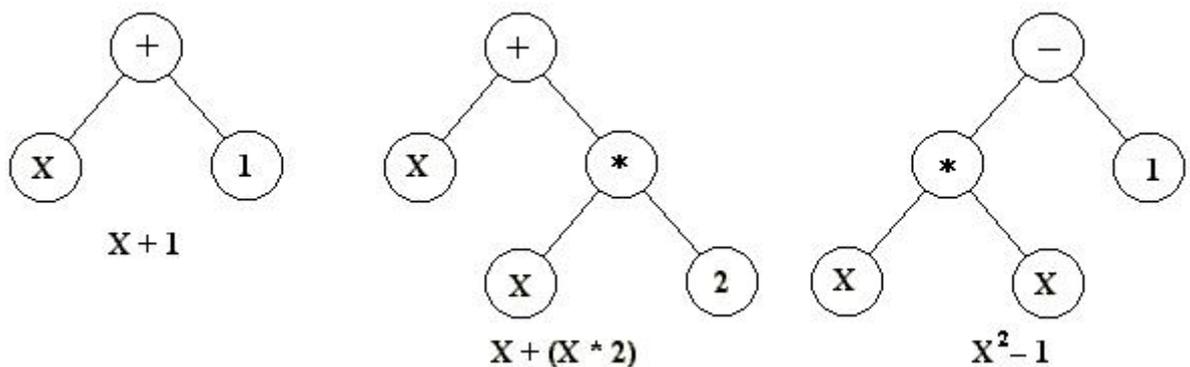


Figura 3.1.1. Árboles utilizados en la programación genética

Los algoritmos evolutivos no necesitan representar las soluciones candidatas como cadenas de datos de una longitud fija. Algunos las representan de esta manera, pero otros no.

3.1.2. MÉTODOS DE SELECCIÓN

Un algoritmo genético puede utilizar muchas técnicas diferentes para seleccionar a los individuos que deben copiarse hacia la siguiente generación, Algunos de estos métodos son mutuamente exclusivos, pero otros pueden utilizarse en combinación, algo que se hace a menudo.

3.1.2.1. SELECCIÓN ELITISTA

Se garantiza la selección de los miembros más aptos de cada generación. La mayoría de los AGs no utilizan elitismo puro, sino que usan una forma modificada para que los mejores individuos sean copiados hacia la siguiente generación en caso de que no surja nada mejor.

3.1.2.2. SELECCIÓN PROPORCIONAL A LA APTITUD

Los individuos más aptos tienen más probabilidad de ser seleccionados. Selección por rueda de ruleta: una forma de selección proporcional a la aptitud en la que la probabilidad de que un individuo sea seleccionado es proporcional a la diferencia entre su aptitud y la de sus competidores.

3.1.2.3. SELECCIÓN ESCALADA

Al incrementarse la aptitud media de la población, la fuerza de la presión selectiva también aumenta y la función de aptitud se hace más discriminadora. Este método puede ser útil para seleccionar más tarde, cuando todos los individuos tengan una aptitud relativamente alta y sólo les distinguen pequeñas diferencias en la aptitud.

3.1.2.4. SELECCIÓN POR TORNEO

Se eligen subgrupos de individuos de la población, y los miembros de cada subgrupo compiten entre ellos. Sólo se elige a un individuo de cada subgrupo para la reproducción.

3.1.2.5. SELECCIÓN POR RANGO

A cada individuo de la población se le asigna un rango numérico basado en su aptitud, y la selección se basa en este ranking, en lugar de las diferencias absolutas en aptitud. La ventaja de este método es que puede evitar que individuos muy aptos ganen dominancia al principio a expensas de los menos aptos, lo que reduciría la diversidad genética de la población y podría obstaculizar la búsqueda de una solución aceptable.

3.1.2.6. SELECCIÓN GENERACIONAL

La descendencia de los individuos seleccionados en cada generación se convierte en toda la siguiente generación. No se conservan individuos entre las generaciones.

3.1.2.7. SELECCIÓN POR ESTADO ESTACIONARIO

La descendencia de los individuos seleccionados en cada generación vuelve al conjunto genético preexistente, reemplazando a algunos de los miembros menos aptos de la siguiente generación. Se conservan algunos individuos entre generaciones.

3.1.2.8. SELECCIÓN JERÁRQUICA

Los individuos atraviesan múltiples rondas de selección en cada generación. Las evaluaciones de los primeros niveles son más rápidas y menos discriminatorias, mientras que los que sobreviven hasta niveles más altos son evaluados más rigurosamente. La ventaja de este método es que reduce el tiempo total de cálculo al utilizar una evaluación más rápida y menos selectiva para eliminar a la mayoría de los individuos que se muestran poco o nada prometedores, y se someten a una evaluación de aptitud más rigurosa y computacionalmente más costosa sólo a los que sobreviven a esta prueba inicial.

3.1.3. MÉTODOS DE CAMBIO

Una vez que la selección ha elegido a los individuos aptos, éstos deben ser alterados aleatoriamente con la esperanza de mejorar su aptitud para la siguiente generación. Existen dos estrategias básicas para llevar esto a cabo.

La primera y más sencilla se llama mutación. Al igual que una mutación en los seres vivos cambia un gen por otro, una mutación en un algoritmo genético también causa pequeñas alteraciones en puntos concretos del código de un individuo.

El segundo método se llama cruzamiento, e implica elegir a dos individuos para que intercambien segmentos de su código, produciendo una descendencia artificial cuyos individuos son combinaciones de sus padres. Este proceso pretende simular el proceso análogo de la recombinación que se da en los cromosomas durante la reproducción sexual. Las formas comunes de cruzamiento incluyen al cruzamiento de un punto, en el que se establece un punto de intercambio en un lugar aleatorio del genoma de los dos individuos, y uno de los individuos contribuye todo su código anterior a ese punto y el otro individuo contribuye todo su código a partir de ese punto para producir una descendencia, y al cruzamiento uniforme, en el que el valor de una posición dada en el genoma de la descendencia corresponde al valor en esa posición del genoma de uno de los padres, elegido con un 50% de probabilidad.

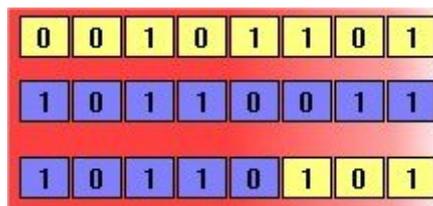


Figura 3.1.3.1. Método de Cruzamiento

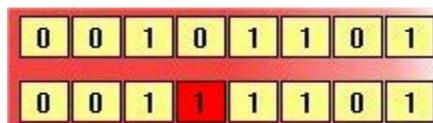


Figura 3.1.3.2. Método de Mutación

Las figuras muestran el efecto de estos dos operadores genéticos en los individuos de una población de cadenas de 8 bits. El diagrama superior muestra a dos individuos llevando a cabo un cruzamiento de un punto; el punto de intercambio se establece entre las posiciones quinta y sexta del genoma, produciendo un nuevo individuo que es híbrido de sus progenitores. El segundo diagrama muestra a un individuo sufriendo una mutación en la posición 4, cambiando el 0 de esa posición de su genoma por un 1.

3.1.4. OTRAS TÉCNICAS DE RESOLUCIÓN DE PROBLEMAS

Con la evolución de la inteligencia artificial y el desarrollo de los métodos heurísticos, han emergido otras técnicas²³ de resolución computerizada de problemas que en algunos aspectos son similares a los algoritmos genéticos.

3.1.4.1. ASCENSO A COLINA (HILL CLIMBING)

Similares a los algoritmos genéticos, aunque más sistemáticos y menos aleatorios. Un algoritmo de ascenso a colina es una solución al problema normalmente elegida al azar. Luego la cadena sufre una mutación y esta proporciona una solución con mayor aptitud. El algoritmo se repite hasta que no se pueda encontrar una mutación que provoque un incremento en la aptitud de la solución actual, y esta solución se devuelve como resultado. El algoritmo de ascenso a colina es lo que se conoce como algoritmo voraz, lo que significa que siempre hace la mejor elección disponible en cada paso, con la esperanza de que de esta manera se puede obtener el mejor resultado global.

3.1.4.2. RECOCIDO SIMULADO (SIMULATED ANNEALING)

El recocido simulado es una técnica de optimización, similar a los algoritmos genéticos. La idea toma prestado su nombre del proceso industrial en el que un material se calienta por encima de su punto de fusión y luego se enfría gradualmente para eliminar defectos en su estructura cristalina, produciendo un entramado de átomos más estable y regular²⁴. En el recocido simulado, existe una función de aptitud que define un paisaje adaptativo; sin embargo, en lugar de una población de candidatas como en los algoritmos genéticos, sólo existe una solución candidata. El recocido simulado también añade el concepto de "temperatura", una cantidad numérica global que disminuye gradualmente en el tiempo. En cada paso del algoritmo, la solución muta (lo que es equivalente a moverse hacia un punto adyacente en el paisaje adaptativo). Luego, la aptitud de la nueva solución se compara con la aptitud de la solución anterior; si es mayor, se conserva la nueva solución. En caso contrario, el algoritmo toma la decisión de conservarla o descartarla en base a la

²³ Koza, John, Martin Keane, Matthew Streeter, William Myrdlowec, Jessen Yu y Guido Lanza. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers, 2003

²⁴ Haupt, Randy y Sue Ellen Haupt. *Practical Genetic Algorithms*. John Wiley & Sons, 1998.

temperatura. Si la temperatura es alta, como lo es al principio, pueden conservarse incluso cambios que causan decrementos significativos en la aptitud, y utilizarse como base para la siguiente ronda del algoritmo, pero al ir disminuyendo la temperatura, el algoritmo se va haciendo más y más propenso a aceptar sólo los cambios que aumentan la aptitud. Finalmente, la temperatura alcanza el cero y el sistema se "congela"; cualquiera que sea la configuración que exista en ese punto se convierte en la solución. El recocido simulado tiene a menudo aplicaciones en la ingeniería del diseño, como determinar la disposición física de los componentes en un chip informático²⁵.

3.1.4. VENTAJAS DE LOS ALGORITMOS GENÉTICOS

El primer y más importante punto es que los algoritmos genéticos son intrínsecamente paralelos. La mayoría de algoritmos son en serie y sólo pueden explorar el espacio de soluciones hacia una solución en una dirección al mismo tiempo, no se puede hacer otra cosa que abandonar todo el trabajo hecho y empezar de nuevo. Sin embargo, ya que los algoritmos genéticos tienen descendencia múltiple, pueden explorar el espacio de soluciones en múltiples direcciones a la vez. Si un camino resulta ser un callejón sin salida, pueden eliminarlo fácilmente y continuar el trabajo en avenidas más prometedoras, dándoles una mayor probabilidad en cada ejecución de encontrar la solución.

La ventaja del paralelismo va más allá de esto, se considera lo siguiente: todas las cadenas binarias de 8 dígitos forman un espacio de búsqueda, que puede representarse como ***** donde * significa 0 o 1.

Por tanto, un algoritmo genético que evalúe explícitamente un número pequeño de individuos está evaluando implícitamente un grupo de individuos mucho más grande de la misma manera que un encuestador que le hace preguntas a un cierto miembro de un grupo étnico, religioso o social espera aprender algo acerca de las opiniones de todos los miembros de ese grupo, y por tanto puede predecir con fiabilidad la opinión nacional sondeando sólo un pequeño porcentaje de la población. De la misma manera, el algoritmo genético puede dirigirse hacia el espacio con los individuos más aptos y encontrar el mejor de ese grupo. En el contexto de los algoritmos genéticos, esto se conoce como teorema del

²⁵ Kirkpatrick, S., C.D. Gelatt y M.P. Vecchi. "Optimization by simulated annealing." *Science*, vol.220, p.671-678 (1983).

esquema, y es la ventaja principal de los algoritmos genéticos sobre otros métodos de resolución de problemas

Debido al paralelismo que les permite evaluar implícitamente muchos esquemas a la vez, los algoritmos genéticos funcionan particularmente bien resolviendo problemas cuyo espacio de soluciones potenciales es realmente grande lo que significa que es demasiado vasto para hacer una búsqueda exhaustiva en un tiempo razonable.

La mayoría de los problemas que caen en esta categoría se conocen como no lineales. En un problema lineal, la aptitud de cada componente es independiente, por lo que cualquier mejora en alguna parte dará como resultado una mejora en el sistema completo. La no linealidad es la norma, donde cambiar un componente puede tener efectos en cadena en todo el sistema, y donde cambios múltiples que individualmente son perjudiciales, que en combinación pueden conducir hacia mejoras en la aptitud mucho mayores. La no linealidad produce una explosión combinatoria.

Otra ventaja notable de los algoritmos genéticos es que se desenvuelven bien en problemas con un paisaje adaptativo complejo, aquéllos en los que la función de aptitud es discontinua, ruidosa, cambia con el tiempo, o tiene muchos óptimos locales. La mayoría de los problemas prácticos tienen un espacio de soluciones enorme, imposible de explorar exhaustivamente, el reto se convierte entonces en cómo evitar los óptimos locales soluciones que son mejores que todas las que son similares a ella, pero que no son mejores que otras soluciones distintas situadas en algún otro lugar del espacio de soluciones.

3.1.5. LIMITACIONES DE LOS ALGORITMOS GENETICOS

Aunque los algoritmos genéticos han demostrado su eficiencia y potencia como estrategia de resolución de problemas. Los algoritmos genéticos tienen ciertas limitaciones²⁶.

La primera y más importante consideración al crear un algoritmo genético es definir una representación del problema. El lenguaje utilizado para especificar soluciones candidatas

²⁶ Mitchell, Melanie. *An Introduction to Genetic Algorithms*. MIT Press, 1996.

debe ser robusto, debe ser capaz de tolerar cambios aleatorios que no produzcan constantemente errores fatales o resultados sin sentido.

Hay dos maneras principales para conseguir esto. La primera, utilizada por la mayoría de los algoritmos genéticos, es definir a los individuos como listas de números binarios, enteros o reales donde cada número representa algún aspecto de la solución candidata. Si los individuos son cadenas binarias, un 0 o 1 podría significar la ausencia o presencia de una cierta característica.

Si son listas de números, estos números podrían representar muchas cosas distintas: los pesos de las conexiones en una red neuronal, el orden de las ciudades visitadas en un recorrido dado, la situación espacial de componentes electrónicos, los valores con los que se alimenta a un controlador, los ángulos de torsión de los enlaces péptidos de una proteína, etc.

Así, la mutación implica cambiar estos números, cambiar bits o sumar o restar valores aleatorios. En este caso, el propio código del programa no cambia; el código es lo que dirige la simulación y hace un seguimiento de los individuos, evaluando sus aptitudes y quizá asegurando que sólo se producen valores realistas y posibles para el problema dado. Otro método en la programación genética, en donde la programación genética representa a los individuos como árboles de código ejecutables que pueden mutar cambiando o intercambiando subárboles. Ambos métodos producen representaciones robustas ante la mutación, y pueden representar muchos tipos diferentes de problemas.

Un problema con el que los algoritmos genéticos tienen dificultades son los problemas con las funciones de aptitud engañosas en las que la situación de los puntos mejorados ofrece información engañosa sobre dónde se encuentra probablemente el óptimo global. En este problema un algoritmo genético al igual que la mayoría de los algoritmos no tendría más probabilidad de encontrar un óptimo global que una búsqueda aleatoria.

Un problema muy conocido que puede surgir con un algoritmo genético se conoce como convergencia prematura. Si un individuo que es más apto que la mayoría de sus

competidores emerge muy pronto en el curso de la ejecución, se puede reproducir tan abundantemente que merme la diversidad de la población demasiado pronto, provocando que el algoritmo converja hacia el óptimo local que representa ese individuo, en lugar de rastrear el paisaje adaptativo lo bastante a fondo para encontrar el óptimo global.

Esto es un problema especialmente común en las poblaciones pequeñas, donde incluso una variación aleatoria en el ritmo de reproducción puede provocar que un genotipo se haga dominante sobre los otros.

Los métodos más comunes implementados por los investigadores en algoritmos genéticos para solucionar este problema implican controlar la fuerza selectiva, para no proporcionar tanta ventaja a los individuos excesivamente aptos. La selección escalada, por rango y por torneo, son tres de los métodos principales para conseguir esto, algunos métodos de selección escalada son el escalado sigma, en el que la reproducción se basa en una comparación estadística de la aptitud media de la población, y la selección de Boltzmann, en la que la fuerza selectiva aumenta durante la ejecución de manera similar a la variable temperatura en el recocido simulado.

3.1.6. EJEMPLOS ESPECÍFICOS DE LOS ALGORITMOS GENÉTICOS

Mientras el poder de la evolución gana reconocimiento cada vez más generalizado, los algoritmos genéticos se utilizan para abordar una amplia variedad de problemas en un conjunto de campos sumamente diverso, demostrando claramente su capacidad y su potencial.

Acústica

Se utilizan algoritmos genéticos para diseñar una sala de conciertos con propiedades acústicas óptimas, maximizando la calidad del sonido para la audiencia, para el director y para los músicos del escenario. Esta tarea implica la optimización simultánea de múltiples variables. Comenzando con una sala con forma de caja de zapatos, el algoritmo genético de los autores produjo dos soluciones no dominadas, ambas descritas como forma de hoja, esta solución tiene proporciones similares al Grosser Musikvereinsaal de Viena, el cual

está considerado generalmente como una de las mejores si no la mejor sala de conciertos del mundo, en términos de propiedades acústicas.

Ingeniería aeroespacial

Se utilizó un algoritmo genético de múltiples objetivos para diseñar la forma del ala de un avión supersónico. Hay tres consideraciones principales que determinan la configuración del ala: Minimizar la resistencia aerodinámica a velocidades de vuelo supersónicas, minimizar la resistencia a velocidades subsónicas y minimizar la carga aerodinámica. Estos objetivos son mutuamente exclusivos, y optimizarlos todos simultáneamente requiere realizar contrapartidas. Las soluciones del algoritmo genético son similares a un diseño llamado "ala flecha", sugerido por primera vez a finales de los años 50, pero que finalmente fue abandonado en favor del diseño más convencional con forma de delta.

Astronomía y astrofísica

La utilidad de los algoritmos genéticos para problemas de astrofísica, aplicándolos a un problema: es obtener la curva de rotación de una galaxia basándose en las velocidades rotacionales observadas de sus componentes. El algoritmo genético produjo dos curvas, y ambas estaban bien ajustadas a los datos, observaciones posteriores se pueden distinguir cuál es la que tiene mejor curvatura.

Química

Un pulso láser ultracorto de alta energía puede romper moléculas complejas en moléculas más sencillas, un proceso con aplicaciones importantes en la química orgánica y la microelectrónica. Para moléculas grandes obtener la forma del pulso deseado de manera analítica es demasiado difícil: los cálculos son demasiado complejos y las características relevantes no se conocen con suficiente precisión. Los algoritmos genéticos resolvieron este problema, en lugar de introducir información compleja del problema, sobre las características cuánticas de las moléculas iniciales, para diseñar el pulso conforme a las especificaciones, se dispara un pulso, mide las proporciones de las moléculas producto

resultantes, muta aleatoriamente las características del rayo con la esperanza de conseguir que estas proporciones se acerquen a la salida deseada, y el proceso se repite.

Ingeniería eléctrica

Se utilizó un algoritmo genético para evolucionar antenas de alambre con propiedades especificadas a priori. El diseño de tales antenas es un proceso impreciso, comenzando con las propiedades deseadas y luego determinando la forma de la antena mediante conjeturas o intuición, experiencia, ecuaciones aproximadas o estudios empíricos. Esta técnica requiere mucho tiempo, a menudo no produce resultados óptimos y tiende a funcionar bien sólo con diseños simétricos y relativamente simples. En contraste, con el método del algoritmo genético, el ingeniero especifica las propiedades electromagnéticas de la antena, y el algoritmo genético sintetiza automáticamente una configuración que sirva.

Mercados financieros

Se utilizó un algoritmo genético para predecir el rendimiento futuro de 1.600 acciones ofertadas públicamente. Concretamente, al AG se le asignó la tarea de predecir el beneficio relativo de cada acción, a lo largo del periodo de tiempo en cuestión, 12 semanas en el futuro. Como entrada, al AG se le proporcionaron datos históricos de cada acción en forma de una lista de 15 atributos, se le pidió al AG que evolucionara un conjunto de reglas si y entonces para clasificar cada acción y proporcionar una salida sobre qué hacer con respecto a la acción. Los resultados del AG fueron comparados con los de un sistema establecido, basado en una red neuronal, que los autores habían estado utilizando para pronosticar los precios de las acciones y administrar las carteras de valores durante tres años. El AG y la red neuronal hicieron pronósticos al final de la semana para cada una de las 1.600 acciones. Doce semanas después de cada predicción, se comparó el rendimiento verdadero con el beneficio relativo predicho. El AG superó significativamente a la red neuronal: en una ejecución de prueba, el AG predijo correctamente la dirección de una acción.

Juegos

Una de las demostraciones más novedosas y persuasivas de la potencia de los algoritmos genéticos es la que presentaron Chellapilla y Fogel²⁷, que utilizaron algoritmos genéticos para evolucionar redes neuronales que pudieran jugar a las damas. Las mayores dificultades en este tipo de problemas relacionados con estrategias es el problema de la asignación de crédito en otras palabras, ¿cómo escribir una función de aptitud? Se ha creído ampliamente que los criterios simples de ganar, perder o empatar no proporcionan la suficiente información para que un algoritmo genético averigüe qué constituye el buen juego.

El algoritmo evolutivo comenzó con una población de 15 redes neuronales con pesos y tendencias, generados aleatoriamente, asignados a cada nodo y conexión; luego, cada individuo se reprodujo una vez, generando una descendencia con variaciones en los valores de la red.

Ingeniería de materiales

Se utilizó algoritmos genéticos para diseñar polímeros conductores de electricidad basados en el carbono, conocidos como poli anilinas. Estos polímeros, un tipo de material sintético inventado recientemente, tienen grandes aplicaciones tecnológicas potenciales y podrían abrir la puerta a nuevos fenómenos físicos fundamentales. Los algoritmos genéticos tienen la tarea de diseñar moléculas nuevas con propiedades especificadas a priori, comenzando con una población de candidatos iniciales generada aleatoriamente.

Matemáticas y algoritmia

Aunque algunas de las aplicaciones más prometedoras y las demostraciones más convincentes de la potencia de los algoritmos genéticos se encuentran en el campo de la ingeniería de diseño, también son relevantes en problemas puramente matemáticos. Se utilizó los algoritmos genéticos para resolver ecuaciones de derivadas parciales no lineales

²⁷ Chellapilla, Kumar y David Fogel. "Evolving an expert checkers playing program without using human expertise." *IEEE Transactions on Evolutionary Computation*, vol.5, no.4, p.422-428 (agosto de 2001).

de alto orden, normalmente encontrando los valores para los que las ecuaciones se hacen cero, y dan como ejemplo una solución casi perfecta para los coeficientes de la ecuación de quinto orden conocido como Super Korteweg-de Vries.

Ejército y cumplimiento de la ley

Utilizo algoritmos genéticos para evolucionar planes tácticos para las batallas militares. Planear una batalla militar táctica es una tarea compleja multidimensional que a menudo atormenta a los profesionales experimentados, no sólo porque este tipo de decisiones a menudo se toman bajo condiciones de mucho estrés, sino también porque hasta los planes más sencillos requieren tomar en cuenta un gran número de variables y consecuencias.

Un uso interesante de los algoritmos genéticos en el cumplimiento de la ley, describiendo el software “FacePrints”, un proyecto para ayudar a los testigos a identificar y describir a los sospechosos criminales. La imagen cliché del artista policía haciendo un dibujo del rostro del sospechoso en base a la descripción de los testigos es un método difícil e ineficiente: la mayoría de la gente no es buena describiendo aspectos individuales del rostro de una persona, como el tamaño de la nariz o la forma de la mandíbula, pero sin embargo son mejores al reconocer caras completas. FacePrints aprovecha esto utilizando un algoritmo genético que evoluciona dibujos de caras basándose en bases de datos de cientos de características individuales que pueden combinarse de infinitas maneras.

Biología molecular

En los seres vivos, las proteínas transmembrana son proteínas que sobresalen de una membrana celular. Estas proteínas realizan a menudo funciones importantes como detectar la presencia de ciertas sustancias en el exterior de la célula o transportarlas hacia el interior de la célula. Se utiliza programación genética para diseñar un algoritmo que identificase el dominio transmembrana de una proteína.

Reconocimiento de patrones y explotación de datos

Se utilizaron algoritmos evolutivos para evolucionar un complejo sistema de reconocimiento de patrones con una amplia variedad de usos potenciales. El reconocimiento de patrones es una tarea cada vez más realizada por algoritmos de aprendizaje automático, en particular, algoritmos evolutivos. La mayoría de ellos comienzan con un acervo de características predefinidas, del que un algoritmo genético puede seleccionar combinaciones apropiadas para la tarea en cuestión; en contraste, este método empezaba desde cero, primero evolucionando detectores individuales de característica en forma de árboles de expresiones, y luego evolucionando combinaciones cooperativas de esos detectores para producir un sistema completo de reconocimiento de patrones.

Robótica

Se ha utilizado Algoritmos Genéticos en la programación de robots en el torneo internacional Robocup el cual es un proyecto para promocionar el avance de la robótica, la inteligencia artificial y los campos relacionados, proporcionando un problema estándar con el que probar las nuevas tecnologías concretamente, es un campeonato anual de fútbol entre equipos de robots autónomos.

Se le proporciono al programa genético un conjunto de funciones de control primitivas como girar, moverse, tirar, etcétera. Estas funciones estaban también sujetas al cambio y refinamiento durante el curso de la evolución.

Diseño de rutas y horarios

Se utilizaron algoritmos genéticos para diseñar los horarios de los exámenes universitarios. Se sabe que, en general, el problema del horario es completo, lo que significa que no se conoce un método para hallar con garantías una solución óptima en un tiempo razonable. En un problema así, hay restricciones duras no puede asignarse el mismo aula a dos exámenes a la vez- y restricciones suaves si es posible, no deben asignarse varios

exámenes en sucesión a un mismo estudiante, para minimizar la fatiga. Las restricciones duras deben satisfacerse, mientras que las restricciones suaves deben satisfacerse lo máximo posible. Este se llama algoritmo memético a su método híbrido para resolver este problema.

3.2. FUNCIONES DE APRENDIZAJE

El aprendizaje es el proceso de adquirir conocimientos, habilidades, actitudes o valores, a través del estudio, la experiencia o la enseñanza. Las funciones de aprendizaje son las actividades que realiza la inteligencia artificial para conseguir el logro de los objetivos que pretende alcanzar, para el reconocimiento de rostros se hace un estudio del aprendizaje automático el que se describirá continuación:

3.2.1. APRENDIZAJE AUTOMÁTICO

Es una rama de la Inteligencia Artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos. Por lo tanto es un proceso de inducción del conocimiento. En muchas ocasiones el campo de actuación del aprendizaje automático se cubre con la estadística, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el aprendizaje automático se centra más en el estudio de la complejidad computacional de los problemas. Muchos problemas son de clase NP-hard, por lo que gran parte de la investigación realizada en aprendizaje automático está enfocada al diseño de soluciones factibles a esos problemas.

El aprendizaje automático tiene una amplia gama de aplicaciones, incluyendo motores de búsqueda, diagnósticos médicos, detección de fraude en el uso de tarjetas de crédito, análisis del mercado de valores, clasificación de secuencias de ADN, reconocimiento del habla y del lenguaje escrito, juegos y robótica.

Interacción hombre-máquina

Algunos sistemas de aprendizaje automático intentan eliminar toda necesidad de intuición o conocimiento experto de los procesos de análisis de datos, mientras otros tratan de establecer un marco de colaboración entre el experto y la computadora. De todas formas, la intuición humana no puede ser reemplazada en su totalidad, ya que el diseñador del sistema especifica la forma de representación de los datos y los métodos de manipulación y caracterización de los mismos.

El aprendizaje automático puede ser visto como un intento de automatizar algunas partes del método científico. El primer paso será conocer las características de un ser humano inteligente, se podría considerar como la capacidad de respuesta de un ser inteligente ante determinadas circunstancias.

El aprendizaje automático es la capacidad que depende de:

- La experiencia.
- Las reglas de comportamiento.
- La facilidad de adquirir información externa.
- La rapidez para encadenar pensamientos.
- La lógica en deducir consecuencias.
- Ciertos elementos no definibles.

La facultad de adquirir y manipular conocimientos para obtener razonamientos útiles es otra forma de acercarse a un concepto de inteligencia natural.

Un ser humano inteligente se caracteriza por las siguientes facultades²⁸:

Aprende: Recoge información del exterior, especialmente a través de los sentidos de la vista y el oído, y la guarda en la memoria.

²⁸ http://es.wikipedia.org/w/index.php?title=Aprendizaje_Autom

Razona y Deduce: Manipula los conocimientos e información que posee, aplicando una serie de reglas y experiencias a veces formales y a veces intuitivas para deducir y solucionar los problemas que se presentan.

Realiza las soluciones elaboradas: Comunica sus deducciones al exterior, bien por medio de la palabra, o bien por los movimientos de sus órganos, motores como las manos y los pies.

Genera sentimiento: La alegría, la tristeza e incluso el amor, son sentimientos que se desprenden de los razonamientos inteligentes.

Conocer, comprender y razonar son los términos que más se relacionan con la inteligencia, dentro de la cual existen varios grados o niveles. La inteligencia no es un atributo exclusivo de los hombres, ya que existen otros seres vivos que también se pueden considerar inteligentes.

Un ser inteligente es el que recoge información del mundo exterior, almacena experiencias y razona para resolver nuevas situaciones. El comportamiento inteligente se caracteriza por no producir siempre los mismos resultados.

Según las circunstancias y factores objetivos y subjetivos, ante una misma situación, los seres inteligentes no toman la misma decisión. Es conocida la importancia que tiene en los juegos de desafío, las diversas actitudes que se adoptan cuando se dispone de las mismas cartas, según las diversas fases del juego o el comportamiento de los rivales.

La Inteligencia Artificial ha sido creada por el hombre para copiar, en cierta medida, la inteligencia natural. Para conseguirlo, se ha escogido como herramienta disponible más potente y sofisticado, al computador, máquina capaz de procesar gran cantidad de datos en muy poco tiempo, según las directrices definidas en un programa.

Hoy en día se ha considerado al computador como una máquina tonta, que significa que solo realiza aquello que se haya programado previamente. Este modo de operar elimina una de las cualidades principales de la inteligencia, que consiste en un comportamiento que

no ha sido programado, sino que se ha producido como consecuencia de una manipulación inteligente de los conocimientos y experiencias.

En el ser humano la experiencia y la facultad de razonamiento se alterna a lo largo del tiempo, es muy frecuente que reaccione de forma diferente ante los mismos hechos. Una máquina puede considerarse como inteligente cuando cumple los siguientes requisitos:

- Si es capaz de percibir visualmente los objetos que la rodean y reconocer sus formas.
- Una máquina es inteligente si es capaz de “entender” el lenguaje natural, hablado o escrito, así como de producir respuestas en dicho lenguaje.
- Otra característica propia de una máquina inteligente consiste en elaborar actuaciones de acuerdo con las condiciones cambiantes del entorno y llevarlas a cabo mediante los correspondientes elementos físicos.
- Se dice que una máquina es inteligente cuando puede almacenar información y conocimientos a los que manipula mediante reglas y algoritmos para alcanzar soluciones a los problemas que plantea su funcionamiento.

La inteligencia comprende facultades como:

- Percibir y manipular elementos del mundo real.
- Adquirir, aplicar conocimientos y entender el lenguaje natural.
- Razonar y resolver problemas.

El estado actual de la Inteligencia Artificial está muy lejos de alcanzar la excelente armonía funcional de los seres vivos Inteligentes y, mucho menos, de fabricar robots que se odien o se amen. Pero ya hay máquinas con cierto grado de inteligencia, y frente a las numerosas objeciones que se plantean ante la posibilidad que ingenios artificiales puedan pensar, hay que preguntar:

¿Quién no ha sido vencido por una máquina electrónica de juegos, como las de ajedrez? Conviene señalar que no siempre se persigue que las máquinas sigan lo más fielmente posible el modo de actuar del ser humano, en ocasiones se prefiere que operen en función de una mayor eficacia respecto a la aplicación hacia la que se orientan.

Modelos de un sistema de Inteligencia Artificial

Una técnica para estudiar la Inteligencia Artificial es considerarla como el conjunto de elementos físicos y lógicos necesarios para simular una mente humana e incluirla en una máquina. Con este enfoque se discuten las diferentes partes que configurarían una mente artificial.

Ojos: Consisten en un sistema de visión que permite conocer las características de los objetos del entorno.

Oídos: Con los que se captan los mensajes hablados o escritos en lenguaje natural y sirven, posteriormente, para la comprensión de los mismos. Existen sistemas, aun muy limitados en vocabulario y presentaciones, que pueden interpretar palabras o frases habladas o tecleadas por un terminal.

El reconocimiento de modelos, es una de las materias más importantes de la Inteligencia Artificial. Se representa de forma efectiva y compacta el conocimiento para utilizarlo con facilidad. Una de las mayores aportaciones de la Inteligencia Artificial es la posibilidad de reconocer el lenguaje natural en el que se expresan los seres humanos para que cualquier persona pueda acceder el control de las máquinas inteligentes.

Memoria: Se trata de una gran base de datos en la que se almacena hechos e información. Además, hay otra base de datos con los objetivos a conseguir.

Cerebro: Es el encargado de llevar a cabo las siguientes misiones:

- Análisis e interpretación de los procedimientos no almacenados en la memoria.
- Procedimientos de razonamiento, que emplean la información de la base de datos para obtener más información.
- Procedimientos de planificación, que construyen un plan para alcanzar el objetivo.

Órganos Motores: Son los actuadores, como los brazo y las piernas, que ejecutan prácticamente el plan generado.

El modelo humano de la máquina inteligente, que representaría a la Inteligencia Artificial. Es una forma de implementar artificialmente la mente humana y este es considerado como un sistema de Inteligencia Artificial.

3.2.2. TIPOS DE ALGORITMOS

Los diferentes algoritmos de Aprendizaje Automático se agrupan en una taxonomía en función de la salida de los mismos. Algunos tipos de algoritmos son:

Aprendizaje supervisado

El algoritmo produce una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema

Aprendizaje no supervisado

Todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan sólo por entradas al sistema.

Aprendizaje por refuerzo

El algoritmo aprende observando el mundo que le rodea. Su información de entrada es el *feedback* o retroalimentación que obtiene del mundo exterior como respuesta a sus acciones.

Transducción

Similar al aprendizaje supervisado, pero no construye de forma explícita una función. Trata de predecir las categorías de los futuros ejemplos basándose en los ejemplos de entrada, sus respectivas categorías y los ejemplos nuevos al sistema.

Aprendizaje multi-tarea

Métodos de aprendizaje que usan conocimiento previamente aprendido por el sistema.

El análisis computacional y de rendimiento de los algoritmos de aprendizaje automático es una rama de la estadística conocida como teoría computacional del aprendizaje.

El aprendizaje automático acaba de manera automática ya que es un proceso tan sencillo para el hombre que no se da cuenta de cómo se realiza y todo lo que esto implica. Desde que se nace hasta que se muere los seres humanos tienen diferentes procesos entre ellos se encuentra el de aprendizaje por medio del cual se adquiere conocimientos, este desarrolla habilidades para analizar y evaluar a través de métodos y técnicas así como también por medio de la experiencia propia, pero a las máquinas hay que indicarles cómo aprender, ya que si no se logra que una máquina sea capaz de desarrollar sus habilidades entonces el proceso de aprendizaje no se estará llevando a cabo, sino que solo será una secuencia repetitiva.

También se debe tener en cuenta que el tener conocimiento o el hecho de realizar bien el proceso de aprendizaje automático no implica que se sepa utilizar, es preciso saber aplicarlo en las actividades cotidianas y un buen aprendizaje también implica saber cómo y cuándo utilizar nuestros conocimientos.

Para llevar a cabo un buen aprendizaje es necesario tener en cuenta todos los factores que a este le rodean como, la sociedad, la economía, la ciudad, el ambiente, el lugar, etc. Por lo cual es necesario empezar a tomar diversas medidas para lograr un aprendizaje adecuado, y obtener una automatización adecuada del aprendizaje, por lo cual lo primero que se debe tener en cuenta es el concepto de conocimiento, el cual es el entendimiento de un determinado tema o materia en el cual tu puedas dar tu opinión o punto de vista así como también responder a ciertas interrogantes que puedan surgir de dicho tema o materia.

En el aprendizaje automático se obtiene 3 tipos de conocimiento los cuáles son:

Crecimiento: Se adquiere con todo lo que nos rodea, el cual guarda la información en la memoria como si dejara huellas este proceso se lo ve en el funcionamiento del cerebro humano.

Reestructuración: Al interpretar los conocimientos el individuo razona y genera nuevo conocimiento al cual se le llama de reestructuración.

Ajuste: Es el que se obtiene al generalizar varios conceptos o generando los propios.

Los tres tipos se efectúan durante un proceso de aprendizaje automático pero la importancia de cada tipo de conocimiento depende de las características de lo que se está tratando de aprender. El aprendizaje es más que una necesidad, es un factor primordial para satisfacer las necesidades de la inteligencia artificial.

CAPITULO IV

DESARROLLO E IMPLEMENTACION DEL SISTEMA

El presente capítulo hace referencia al proceso de desarrollo de Software que se ha seguido para el análisis, diseño, y construcción del Sistema de reconocimiento de rostros, en primera instancia se analizan los requisitos que orientaron el diseño de la aplicación SIFARE; asimismo, se incluyen los requisitos de software que debe satisfacer la aplicación. Para la construcción del sistema se propone la utilización del proceso unificado rational (RUP).

4.1. ESPECIFICACION DE REQUISITOS DE SOFTWARE

4.1.1. INTRODUCCION

El presente documento es una Especificación de Requisitos Software (ERS) para el sistema de reconocimiento de rostros. Esta especificación se ha estructurado inspirándose en las directrices dadas por el estándar “IEEE Recommended Practice for Software Requirements Specification ANSI/IEEE 830 1998”.

4.1.2. PROPOSITO

El objetivo es definir de manera clara y precisa las funcionalidades y restricciones del producto, y está dirigido al desarrollador y usuarios finales del sistema.

4.1.3. AMBITO DEL SISTEMA

En primera instancia y para el tema de estudio que se está tratando el sistema se desarrollara en la Escuela Politécnica del Ejercito., pero en el futuro el objetivo es que esta aplicación permita que las empresas posean un sistema que les permita realizar un control de acceso, control de personal, etc.

Esta aplicación será totalmente diseñada y codificada con la herramienta de Matlab 7.2.

4.1.4. DEFINICION DE ACRONIMOS Y ABREVIATURAS

En esta sección se definirán todos los términos, acrónimos y abreviaturas utilizadas en la ERS.

4.1.4.1. DEFINICIONES

Administrador	Persona encargada de la Administración del sistema de reconocimiento de rostros
Personas	Individuos que serán registrados en el sistema

Tabla 4.1.4.1. Definiciones

4.1.4.2. ACRONIMOS

ERS	Especificación de Requisitos de Software
PCA	Análisis de componentes principales
RN	Redes Neuronales

Tabla 4.1.4.2. Acrónimos

4.1.4.3. ABREVIATURAS

SIFARE	Sistema de Reconocimiento de Rostros
--------	--------------------------------------

Tabla 4.1.4.3. Abreviaturas

4.1.5. REFERENCIAS

Estándar ANSI/IEEE de Especificación de Requisitos de Software 830/1998.

4.1.6. DESCRIPCION GENERAL

Se presenta las principales áreas de negocio a las cuales el sistema debe dar soporte, las funciones que el sistema debe realizar, la información utilizada, las restricciones y otros factores que afecten al desarrollo del mismo.

4.1.6.1. PERSPECTIVA DEL PRODUCTO

El sistema actualmente será desarrollado en la Escuela Politécnica del Ejercito pero el propósito es que se pueda aplicar a cualquier tipo de empresa que necesite de este sistema.

4.1.6.2. REQUISITOS DE LA APLICACIÓN

Antes de empezar con el proceso de diseño de la aplicación, es importante definir la función de la misma. El análisis de los requisitos constituye uno de los pasos decisivos en el desarrollo de la aplicación. El objetivo de la identificación de los requisitos consiste en crear una solución que se adapte a las necesidades tanto del consumidor como del distribuidor. Estos requisitos se transformaran, en un documento que servirá como guía para el desarrollo del concepto global del proyecto.

En esta sección se indican los requisitos que se identifican para la aplicación SIFARE.

REQUISITOS DE FUNCIONAMIENTO

Si una persona común tuviese que definir el sistema de reconocimiento de rostros, probablemente diría que se trata de identificar una persona por medio de una cámara web. Aunque esta descripción es exacta, no describe completamente los procesos que se encuentran inmersos. En un mundo de continuo y de rápido movimiento como es el de la inteligencia artificial, abarca el campo de seguridad y control empresarial.

Las empresas son cada vez más exigente con este tipo de sistemas y si uno de ellos no cumple con sus expectativas, optará por utilizar un sistema más confiable. Por lo tanto, ¿qué es lo que demanda los clientes? En la siguiente lista se incluyen algunos de los principales aspectos que afectan al diseño de la aplicación.

- Facilidad de uso y exploración
- Rápido rendimiento
- Reducción de tiempo de respuesta
- Aumento de la escalabilidad

En un primer momento, parece que alguno de estos retos pertenece al ámbito del diseñador de la aplicación, mientras que otros son responsabilidad de los directores empresariales. Sin embargo, si estos objetivos se examinan detalladamente, se podrá comprender hasta qué punto influye el diseño de la aplicación en cada uno de ellos.

FACILIDAD DE USO Y EXPLORACIÓN

Para facilitar la utilización del sistema consistirá en emplear analogías conocidas para las tareas comunes. Esto facilita que el administrador que no posee un amplio conocimiento de informática comprenda el funcionamiento del sistema y pueda manejarlo.

Si el sistema ofrece facilidad de uso y exploración, el administrador disfrutara utilizarlo. De lo contrario, si presenta dificultades, probablemente no utilice el sistema y opten por otro distinto.

RÁPIDO RENDIMIENTO

Existen varios factores en el diseño que pueden afectar a su rendimiento. Es obvio que el rendimiento adquiere distintos significados dependiendo de la persona que lo use y que los niveles de rendimiento que se consideran aceptables puede variar de una persona a otra. El rendimiento del sistema puede afectar al administrador en el tiempo que este necesita para realizar la identificación de la persona.

REDUCCIÓN DEL TIEMPO DE RESPUESTA

El factor que se considera como el más indicativo del buen funcionamiento de un sistema es que el tiempo de respuesta sea aceptable. El tiempo de respuesta es la cantidad de tiempo de espera que transcurre desde que se realiza una petición hasta que se obtiene el resultado. Lo ideal sería que todas las acciones del sistema se realizaran de forma instantánea; sin embargo, en la práctica, factores como la velocidad de procesamiento es limitado, la concurrencia a la base de datos y las tareas de procesamiento suelen ser la causa de ligeros retrasos. Uno de los objetivos del proceso de diseño del sistema debe consistir en intentar reducir los factores que afectan negativamente al tiempo de respuesta, aunque dichos factores no se puedan eliminar en su totalidad.

AUMENTO DE LA ESCALABILIDAD

Otro aspecto importante que influye en el rendimiento es la escalabilidad. Se trata de la capacidad del sistema para poder aumentar cuando se agregan recursos. El sistema continuará proporcionando tiempos de respuesta aceptables cuando un gran número de personas sean ingresadas. Muchos desarrolladores se han decepcionado al comprobar que un sistema de prueba que funcionaba extraordinariamente en un equipo de desarrollo no podía hacer frente al número de personas necesario.

4.1.6.3. CARACTERÍSTICAS DE LOS USUARIOS

El sistema debe ofrecer una interfaz de usuario intuitiva fácil de aprender y sencillo de manejar. El sistema debe presentar un grado alto de usabilidad, sería deseable que el usuario se familiarice desde la primera vez que ingresa en el mismo.

4.1.6.4. RESTRICCIONES

El sistema se realizará en base a las políticas y los procesos de negocio establecidos por el desarrollador. Este sistema tiene las funcionalidades generales comunes que se definieron.

Lo que no se considera es:

- El Sistema no eliminara personas ingresadas
- No se comunicará con otros sistemas
- Trabaja independientemente y no tendrá interfaces con otros servicios.

En cuanto a las restricciones de Hardware y Software, por la naturaleza de la aplicación esta deberá funcionar en un ambiente de formularios tanto el modulo de administración de personas. Para el desarrollo del sistema se utilizará la tecnología de Matlab para el desarrollo de Aplicaciones.

4.1.6.5. SUPOSICIONES Y DEPENDENCIAS

En esta sección se encuentran todas las suposiciones, dependencias con otros sistemas.

4.1.6.5.1. SUPOSICIONES

Se asume que los requisitos ya han sido previamente analizados y aprobados cualquier modificación debe ser aprobada por todas las partes involucradas y se debe actualizar este documento ya que en base a este se desarrollara el sistema de reconocimiento de rostros.

4.1.6.5.2. DEPENDENCIAS

SIFARE, funciona autónomamente, sin necesidades de comunicarse con otros sistemas externos, por lo que no hay dependencia respecto a otros sistemas.

4.1.7. REQUISITOS ESPECÍFICOS

En la siguiente sección se describen los requisitos propiamente dichos identificados durante la fase de planeación de este proyecto, así como el modelo de aplicación y el diseño de proceso utilizados en el desarrollo de la aplicación SIFARE.

SIFARE se diseñó para cumplir con los siguientes requisitos de funcionamiento:

- Requisitos funcionales
- Requisitos de interfaces externas
- Requisitos tecnológicos

4.1.7.1. REQUISITOS FUNCIONALES

En las siguientes líneas se especifican los requisitos a un mayor detalle permitiendo un mejor entendimiento del sistema.

4.1.7.1.1. INGRESO DE PERSONA

- Req(01) El administrador debe poder realizar el ingreso de una persona al sistema. El registro incluye la siguiente información: nombre, apellido, teléfono, dirección, cargo e imagen capturada.

4.1.7.1.2. CONSULTAR PERSONA

- Req (02) El sistema permitirá consultar los datos de una persona que este previamente ingresado en el sistema. La consulta se realizara por medio del código de la persona.

4.1.7.1.3. IDENTIFICACIÓN PERSONA

- Req(03) El sistema permitirá realizar la identificación de una persona que este registrada en la base de datos. La identificación se realizara por medio de la imagen capturada de la persona

4.1.7.1.4. IMPRESIÓN DE REPORTE

- Req(04) El sistema permitirá realizar la impresión de reporte de una persona que haya sido identificada y se encuentre en la base de datos.

4.1.7.2. REQUISITOS DE INTERFACES EXTERNAS

En esta sección se describirán los requisitos que afectan a las interfaces del administrador.

4.1.7.2.1. INTERFACES DEL ADMINISTRADOR

La interfaz debe ser orientada a formularios y el manejo del programa se realizará a través de teclado y ratón.

4.1.7.2.2. INTERFACES DE SOFTWARE

En la versión inicial el sistema trabajará autónomamente, y no tendrá ninguna interfaz de software con otros sistemas.

4.1.7.3. REQUISITOS DE DESARROLLO

El ciclo de vida elegido para el desarrollo del sistema es evolutivo de manera que se pueda incorporar fácilmente cambios o agregar mayor funcionalidad al sistema, además en el futuro el sistema podrá relacionar a otros sistemas para prestar mayores servicios.

4.1.7.4. REQUISITOS TECNOLÓGICOS

Son las características técnicas que se requiere para la correcta operación del Sistema para ello son necesarios los siguientes elementos de hardware, software y ambiente:

4.1.7.4.1. REQUERIMIENTOS DE HARDWARE

Computador

La configuración recomendada para este dispositivo es la siguiente:

- Procesador Intel Pentium IV
- Puertos Usb
- 512MB de memoria, 1 Gb recomendado.
- 3 Gb de espacio libre en disco para instalación
- WebCam DELUX

4.1.7.4.2. REQUERIMIENTOS DE SOFTWARE

En esta sección se encuentran los requerimientos de Software bajo los cuales la aplicación va a funcionar.

- Un sistema operativo Microsoft Windows XP Servipack 2.
- Codec de Imágenes DirectX 9.0 o superior
- Matlab 7.2.
- Driver WebCam

4.1.7.4.3. REQUERIMIENTOS DE AMBIENTE

En esta sección se encuentran los requerimientos de ambiente bajo los cuales la aplicación va a funcionar correctamente.

- La distancia mínima de la WebCam con el individuo al que se va a realizar el reconocimiento de rostro debe ser de 1.35 metros.
- La iluminación en la toma de las imágenes digitales de los usuarios debe ser alta.
- El fondo en la toma de las imágenes digitales debe ser de color blanco.

4.2. CONSTRUCCION DEL SISTEMA

Para la construcción del sistema se utiliza Proceso Unificado Rational (RUP) que describe a continuación:

4.2.1. EL PROCESO UNIFICADO RATIONAL (RUP)

El Proceso Unificado Rational es una metodología propuesta por la Rational Software Corporation, desarrollada por Jacobson, Booch y Rumbaugh, y adopta la notación UML (Unified Modeling Language-Lenguaje Unificado de Modelado). Constituye una metodología iterativa e incremental, dirigida por Casos de Uso y centrada en la arquitectura. En un ciclo de vida iterativo e incremental, el desarrollo procede como una serie de iteraciones que evolucionan hasta el sistema final. Cada iteración consiste de los siguientes componentes de proceso: análisis de los requerimientos, análisis, diseño, implementación y pruebas.

El Proceso Unificado Rational se estructura en dos dimensiones diferentes:

- El tiempo: división del ciclo de vida en fases e iteraciones.
- Componentes de Proceso: producción de un conjunto específico de componentes con actividades bien definidas.

Estructurar el proyecto a lo largo de la dimensión del tiempo envuelve la adopción de las siguientes fases:

- Inicio: especificación de la visión del proyecto.

- Elaboración: planear las actividades necesarias y los recursos requeridos; especificando las características y diseñando la arquitectura.
- Construcción: construir el producto como una serie de iteraciones incrementales.
- Transición: proveer el producto a la comunidad de usuarios (construir, entregar y entrenar).

Estructurar el proceso a lo largo de la dimensión de componentes incluye las siguientes actividades:

- Análisis de Requerimientos: descripción de lo que debería hacer el sistema.
- Diseño: arquitectura del sistema. Cómo se realizará el sistema en fase de implementación.
- Implementación: la producción del código que resultará en un sistema ejecutable.
- Test: la verificación del sistema completo.

La interacción entre ambas dimensiones se observa en la figura 4.2.1. Como se puede notar en la figura, cada fase incluye cierto flujo de trabajo correspondiente a cada uno de los componentes de proceso.

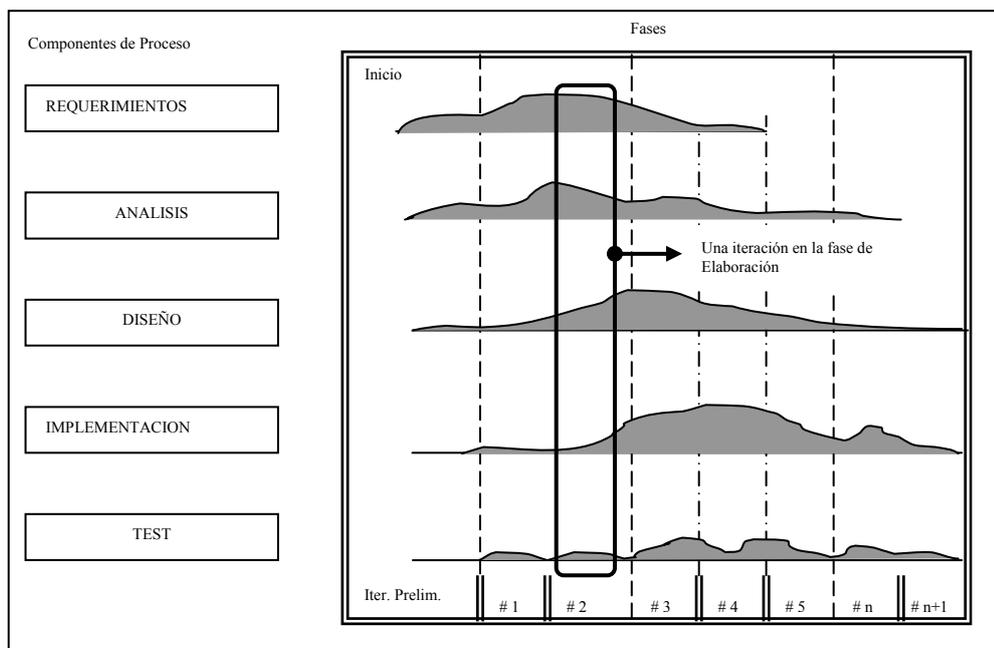


Figura 4.2.1: Interacción entre las dimensiones Tiempo y Componentes de Proceso

4.2.1.1. ARQUITECTURA DEL SISTEMA.

Una vista de la representación de la arquitectura del Sistema se tiene en la figura 4.2.1.1. Aquí se puede observar el hecho de que todo el desarrollo de las distintas vistas del sistema gira en torno a los Casos de Uso.

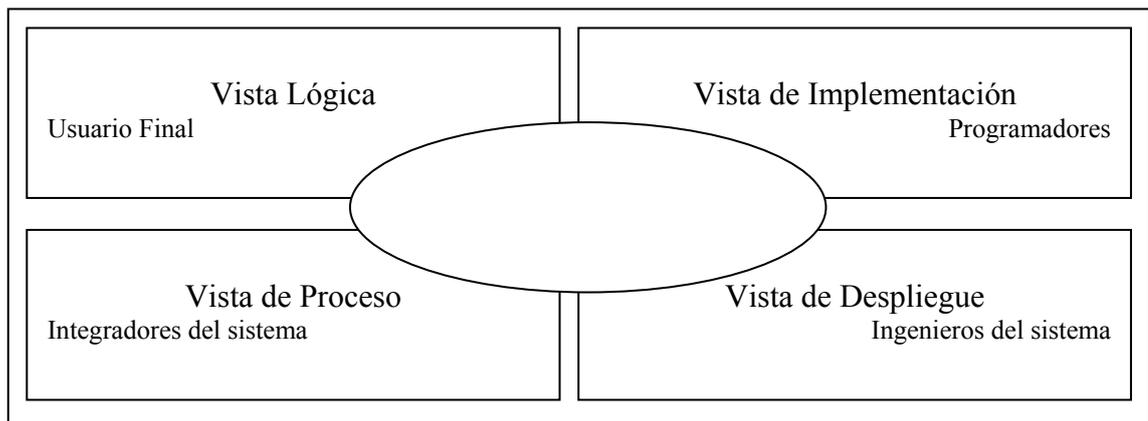


Figura 4.2.1.1. Representación de la Arquitectura del Sistema

Como se puede notar en la figura 4.2.2., la arquitectura de un sistema se organiza en términos de distintas capas. Cada capa representa una abstracción coherente y contiene una interfaz controlada y bien definida. Las vistas o capas identificadas son:

- La vista de *Casos de Uso* describe al sistema como un conjunto de transacciones desde el punto de vista de actores externos.
- La vista *Lógica* contiene una colección de paquetes, clases y relaciones.
- La vista de *Procesos* contiene procesos, hilos de ejecución, mecanismos de comunicación entre procesos y sincronización.
- La vista de *Implementación* contiene módulos y subsistemas.
- La vista de *Despliegue* contiene los nodos físicos del sistema y las conexiones entre nodos.

4.2.2. ANALISIS DEL SISTEMA

El primer paso para el modelado de una aplicación es la identificación de todos los procesos que componen la lógica de negocios. En este punto simplemente se lista los

procesos que se observan en el documento de especificación de requisitos de software para luego abordarlos uno a uno y asignarlos a cada actor en el diagrama de casos de uso.

Procesos:

- Administrar personas

4.2.2.1. DIAGRAMA DE CASOS DE USO

Este diagrama representa la funcionalidad completa de un sistema (o una clase) mostrando su interacción con los agentes externos. Esta representación se hace a través de las relaciones entre los actores (agentes externos) y los casos de uso (acciones) dentro del sistema. Los diagramas de casos de uso definen conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer todos los requerimientos que tiene a su cargo. Esas funcionalidades son representadas por casos de uso. Se pueden visualizar como las funciones más importantes que la aplicación realiza o como las opciones presentes en el menú de la aplicación.

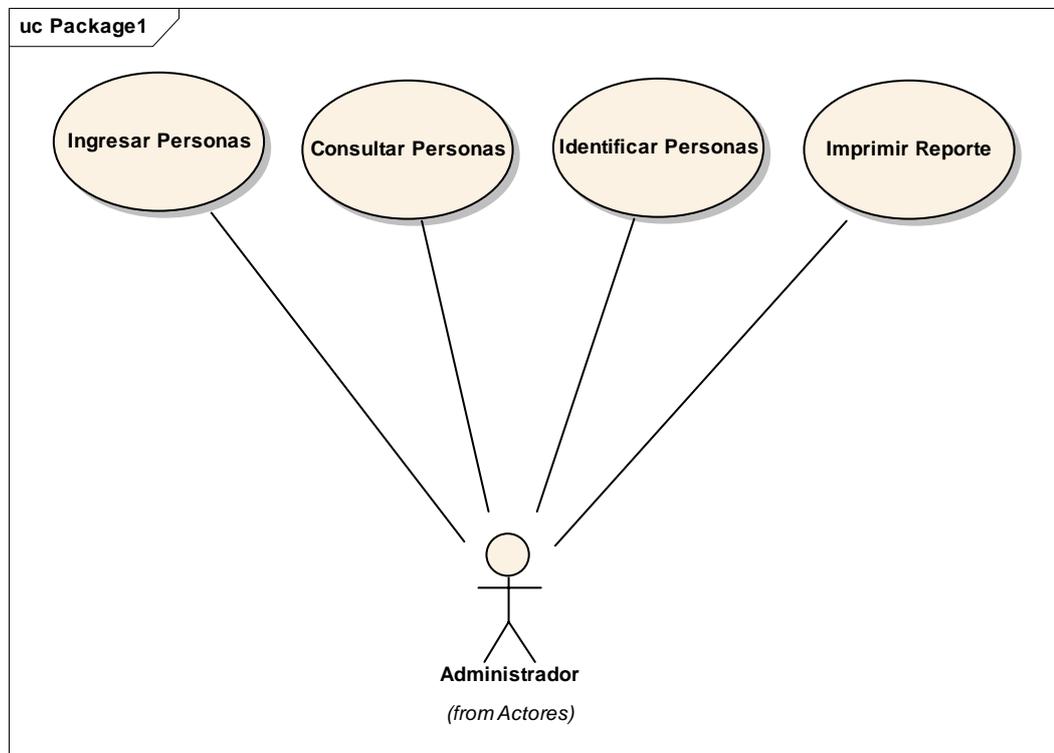


Figura 4.2.2.1. Diagrama de casos de uso

Descripción de Casos de Uso del Administrador

1. Caso de Uso: Ingresar Persona

Nombre: Ingresar Persona

Alias: Ninguno

Actor: Administrador

Función: Permitir registrar a una persona

Descripción: El Administrador añade los datos de la persona: nombre, apellido, teléfono, dirección, cargo y la fotografía.

Referencias: De requerimientos: Req(01)

De casos de uso: Administrar personas

2. Nombre Caso de Uso: Consultar Persona

Nombre: Consultar Persona

Alias: Ninguno

Actor: Administrador

Función: Permitir buscar una persona registrada

Descripción: El Administrador consulta todos los datos de una persona registrada en el sistema por medio del código.

Referencias: De requerimientos: Req(02)

De casos de uso Administrar personas

3. Nombre Caso de Uso: Identificar Persona

Nombre: Identificar Persona

Alias: Ninguno

Actor: Administrador

Función: Permitir identificar a una personas registrada

Descripción: El Sistema Identifica si el usuario esta registrado en el sistema por medio de una imagen.

Referencias: De requerimientos: Req(03)

De casos de uso Administrar personas

4. Nombre Caso de Uso: Imprimir Reporte

Nombre: Imprimir Reporte

Alias: Ninguno

Actor: Administrador

Función: Permitir imprimir el reporte de identificación de una persona registrada.

Descripción: El Administrador imprime el reporte de una persona ya identificada en el sistema.

Referencias: De requerimientos: Req(04)

De casos de uso Administrar Personas

4.2.2.2. DIAGRAMAS DE CASOS DE USO EN FORMATO EXPANDIDO

En este formato se establecen los eventos que pueden ser generados por el actor y van a ser atendidos por cada Caso de Uso. Por evento se entiende la interacción que tiene un actor con la aplicación a través de la interfaz gráfica, tal como el seleccionar por medio del ratón, el ingreso de un texto a un componente, el movimiento de un elemento de la interfaz, etc. Todos los eventos van numerados en orden secuencial de acuerdo a la secuencia lógica como ocurrirían en la aplicación (ciclo de vida del caso de uso). De este formato se obtiene la información para la creación de los diagramas de interacción, más específicamente el de secuencia. También se deben presentar los eventos alternos, los cuales permiten establecer las excepciones que se pueden presentar en la ejecución del programa.

1. Caso de Uso: Ingresar Persona

SIFARE	FORMATOS DE EVENTOS
Nombre Caso de Uso:	Ingresar Persona
Alias:	Ninguno
Actores:	Administrador
Evento:	Respuesta del Sistema
1. El actor ingresa al formulario de ingreso de datos	2. El sistema muestra formulario de ingreso de datos.
	3. El sistema habilita los campos que deben completarse
4. El actor solicita se tome la fotografía	5. El sistema muestra en la pantalla la cámara para tomar la fotografía
6. El actor ingresa los datos de la persona incluyendo su fotografía	
7. El actor indica al sistema que guarde la información	8. El sistema guarda la información de la nueva persona
	9. El sistema limpia la pantalla
EVENTOS ALTERNOS	
Nombre Caso de Uso:	Ingreso de Persona
Alias:	Ninguna
Actores:	Administrador
Número de Línea:	Respuesta del Sistema
Línea 8	Quedan algunos campos vacíos que eran obligatorios. El sistema muestra un mensaje de ingresar los datos correctamente
Línea 8	El formato de un campo no esta correcto. El sistema muestra un mensaje indicando que ingrese los datos correctamente

2. Caso de Uso: Consultar Persona

SIFARE	FORMATOS DE EVENTOS
Nombre Caso de Uso:	Consultar Persona
Alias:	Ninguno
Actores:	Administrador
Evento:	Respuesta del Sistema
1. El actor ingresa al formulario de consulta de persona	2. El sistema muestra formulario de consulta de persona
	3. El sistema habilita el campo código que debe completarse.
4. El actor ingresa el código de la persona	
5. El actor indica al sistema que inicie la consulta	6. El sistema busca la información de la persona
	7. El sistema despliega los datos persona almacenada en el formulario.
EVENTOS ALTERNOS	
Nombre Caso de Uso:	Consultar Persona
Alias:	Ninguna
Actores:	Administrador
Número de Línea:	Respuesta del Sistema
Línea 4	El código no se ingreso en el campo. El sistema muestra un mensaje indicando que ingrese el código de la persona.

3. Caso de Uso: Identificar Persona

SIFARE	FORMATOS DE EVENTOS
Nombre Caso de Uso:	Identificar Persona
Alias:	Ninguno
Actores:	Administrador
Evento:	Respuesta del Sistema
1. El administrador ingresa al formulario de identificar persona	2. El sistema muestra formulario de identificar persona
3. El administrador solicita se tome la fotografía	4. El sistema muestra en la pantalla la cámara para tomar la fotografía
5. El administrador envía fotografía al analizador	6. El analizador solicita imágenes modelos al sistema
	7. El analizador transforma imágenes, compara datos y envía la identidad de la persona al sistema
	8. El sistema despliega los datos de la persona identificada.
EVENTOS ALTERNOS	
Nombre Caso de Uso:	Consultar Persona
Alias:	Ninguna
Actores:	Administrador
Número de Línea:	Respuesta del Sistema
Línea 5	La imagen no es enviada. El sistema muestra un mensaje indicando que envié la imagen de la persona.
Línea 8	La persona no existe. El sistema muestra un mensaje de que la persona no existe en la base de datos

4. Caso de Uso: Impresión de Reporte

SIFARE	FORMATOS DE EVENTOS
Nombre Caso de Uso:	Impresión de Reporte
Alias:	Ninguno
Actores:	Administrador
Evento:	Respuesta del Sistema
1. El actor ingresa al formulario de impresión de reporte.	2. El sistema muestra formulario de impresión de reporte
3. El actor elige el reporte de una persona	4. El sistema busca la información de la persona y muestra.
5. El actor indica al sistema que inicie la impresión	6. El sistema realiza la impresión del reporte.
EVENTOS ALTERNOS	
Nombre Caso de Uso:	Impresión de Reporte
Alias:	Ninguna
Actores:	Administrador
Número de Línea:	Respuesta del Sistema
Línea 3	No se encuentra el reporte requerido.
Línea 5	Se cancele la impresión del reporte

4.2.3. DISEÑO DEL SISTEMA

Para el diseño del sistema se utilizó los diagramas de secuencia que muestran las interacciones de un usuario con el sistema. Interacción es una cadena de mensajes enviados entre los objetos en respuesta a un evento generado por el usuario sobre la aplicación. Los diagramas de interacción pueden ser diagramas de secuencia y diagramas de colaboración. Estos diagramas conforman la etapa del diseño de la aplicación, y se crean a partir de los diagramas de Casos de Uso.

Los Diagramas de Secuencia representan una interacción entre objetos de manera secuencial en el tiempo. Muestra la participación de objetos en la interacción entre sus “líneas de vida” (desde que instancia) y los mensajes que ellos organizadamente intercambian en el tiempo. El responsable o ACTOR es quien inicia el ciclo interactuando inicialmente con la interfaz de usuario: (GUI); en seguida se inician todos los objetos que intervienen en el funcionamiento del aplicativo. En este diagrama se comienza a observar el comportamiento del sistema a partir de los eventos generados por los actores.

Los Diagramas de Colaboración por otra parte dan todas las especificaciones de los métodos. Estos permiten describir una operación específica incluyendo sus argumentos y variables locales creadas durante su ejecución. Se muestran los objetos y mensajes que son necesarios para cumplir con un requerimiento o propósito, o con un conjunto de ellos. Se puede elaborar para una operación o para un Caso de Uso, con el fin de describir el contexto en el cual su comportamiento este inmerso.

4.2.3.1. DIAGRAMA DE SECUENCIA: INGRESAR PERSONA

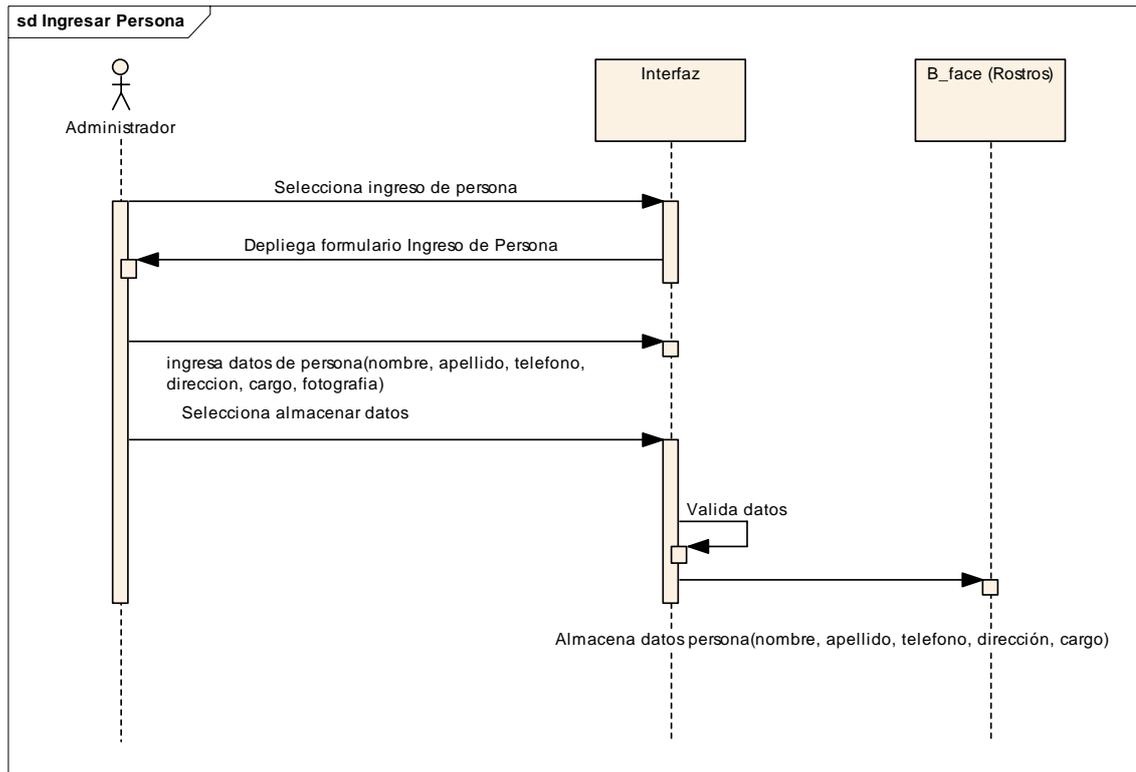


Figura 4.2.3.1. Diagrama de Secuencia Ingresar Persona

4.2.3.2. DIAGRAMA DE SECUENCIA: CONSULTAR PERSONA

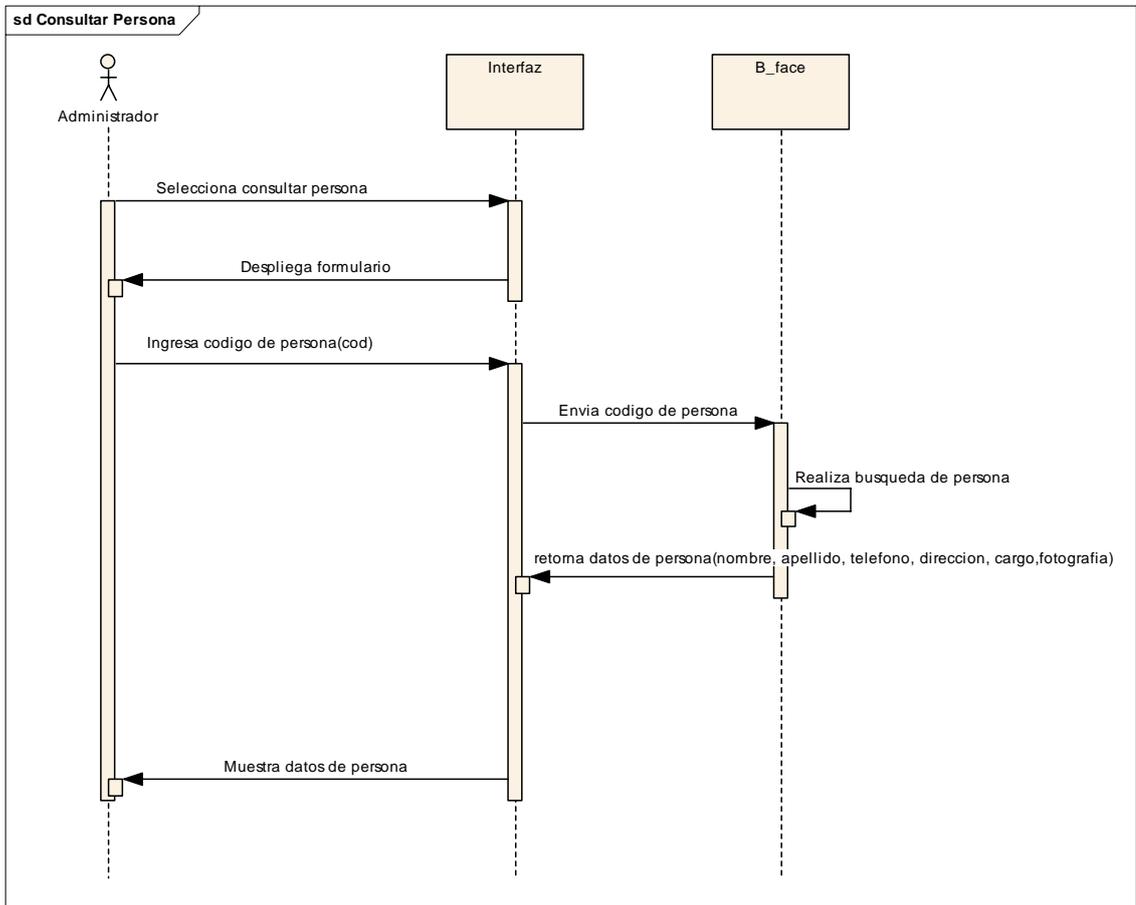


Figura 4.2.3.2. Diagrama de Secuencia Consultar Persona

4.2.3.3. DIAGRAMA DE SECUENCIA: IDENTIFICAR PERSONA

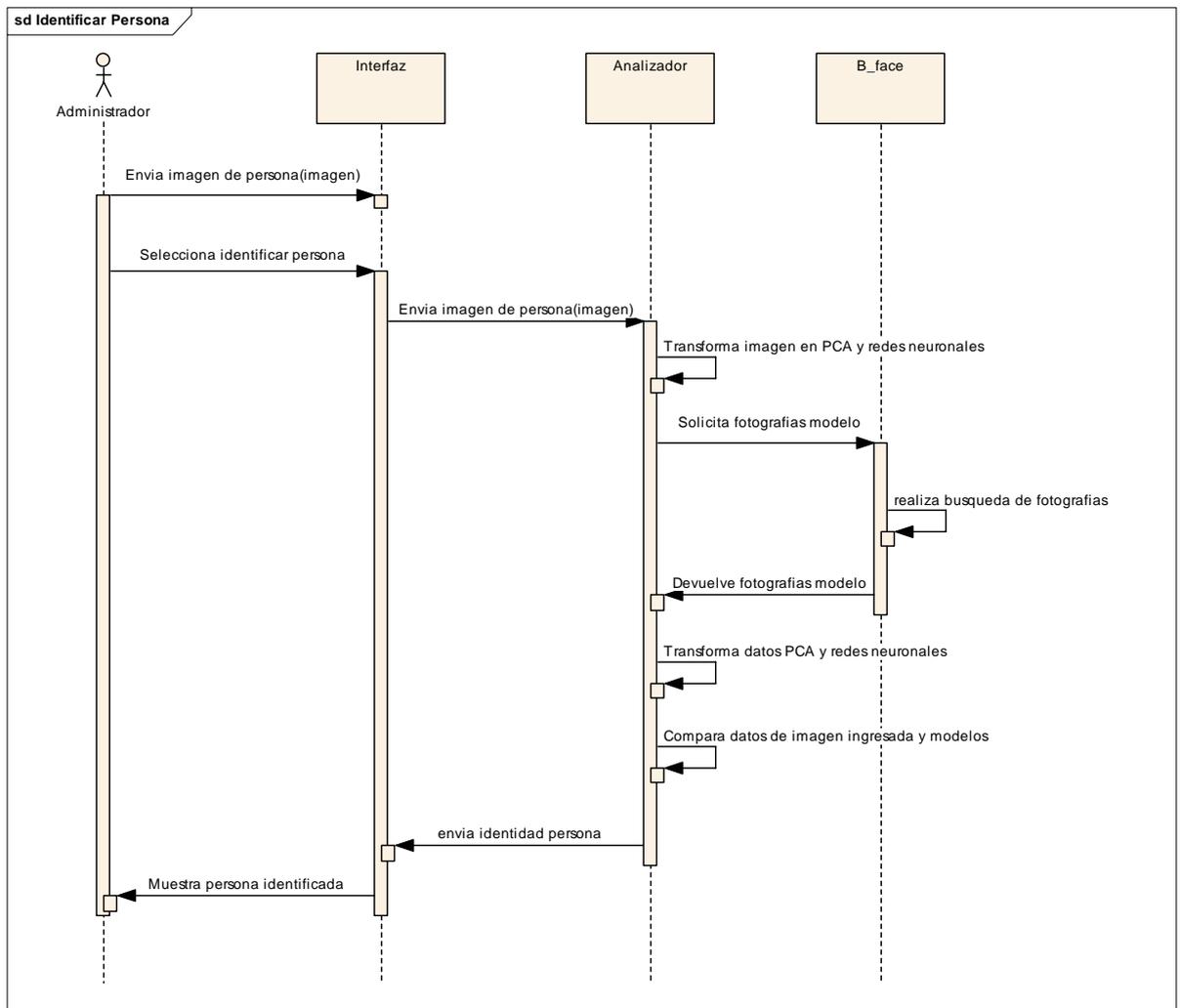


Figura 4.2.3.3. Diagrama de Secuencia Identificar Persona

4.2.3.4. DIAGRAMA DE SECUENCIA: IMPRIMIR REPORTE

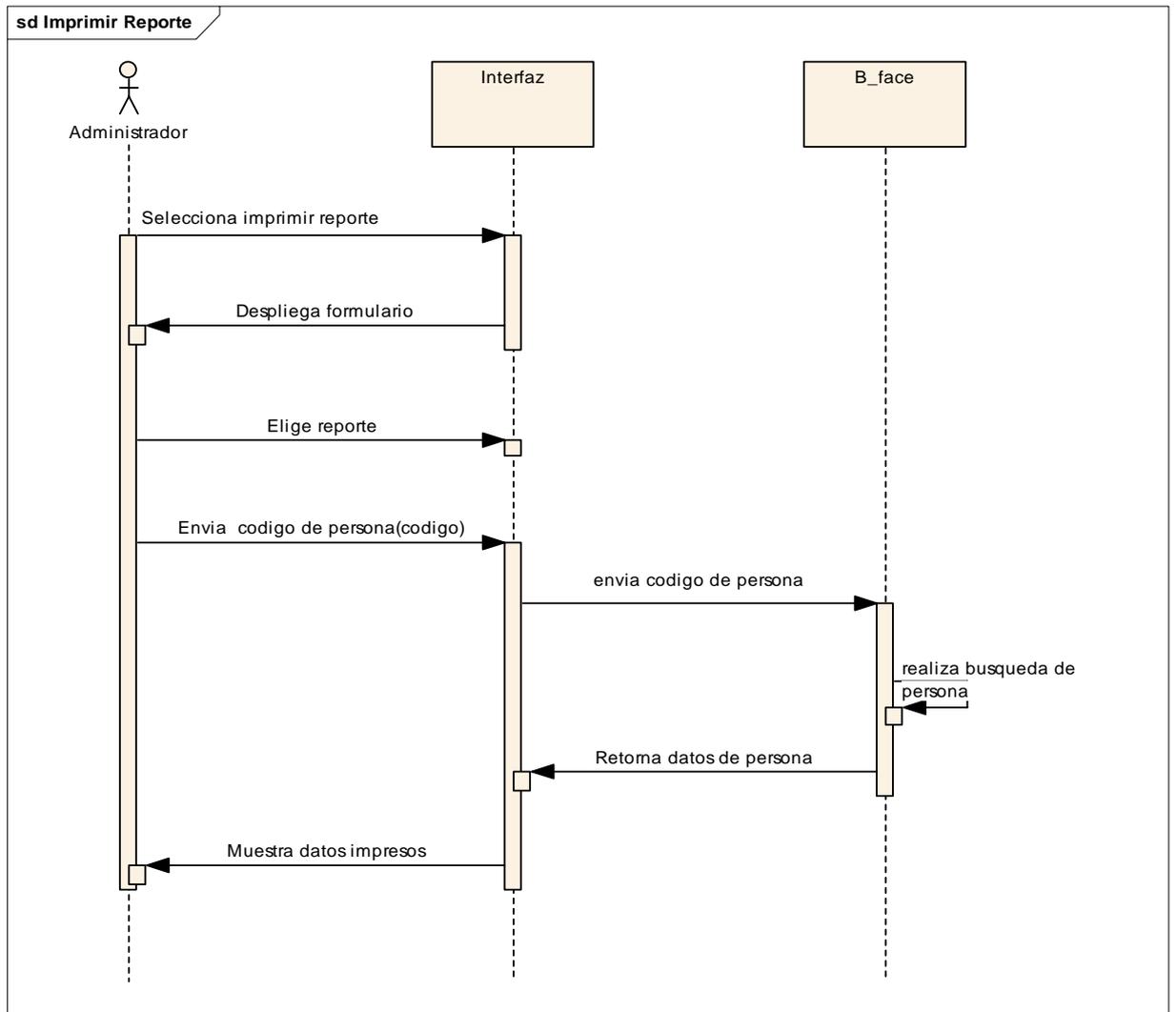


Figura 4.2.3.4. Diagrama de Secuencia Imprimir Reporte

4.2.3.5. DIAGRAMA DE COLABORACIÓN: INGRESAR PERSONA

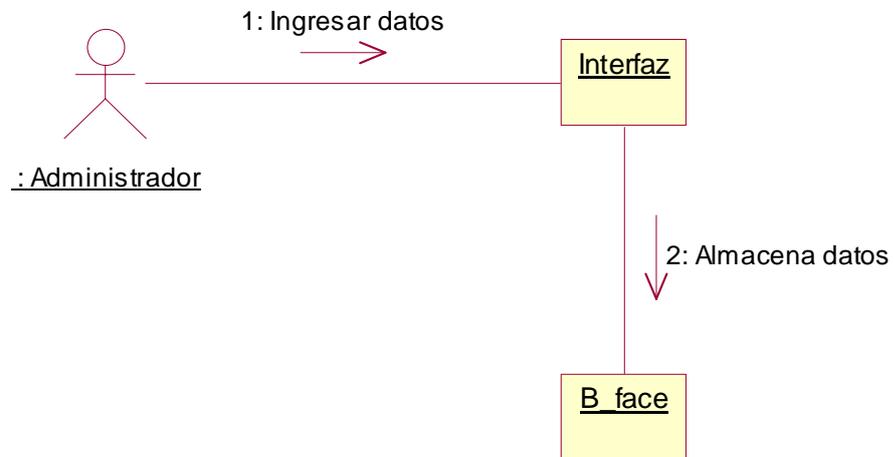


Figura 4.2.3.5. Diagrama de Colaboración Ingresar Persona

4.2.3.6. DIAGRAMA DE COLABORACIÓN: CONSULTAR PERSONA

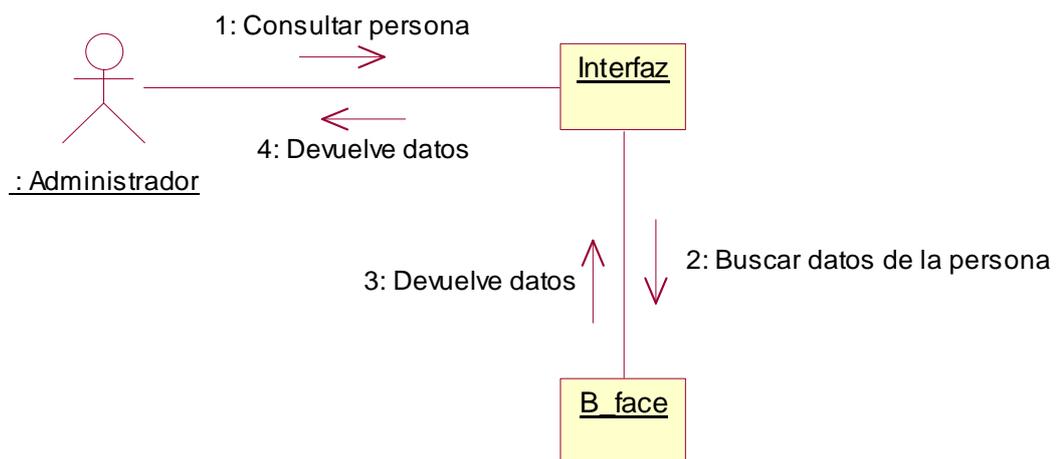


Figura 4.2.3.6. Diagrama de Colaboración de Consultar Persona

4.2.3.7. DIAGRAMA DE COLABORACIÓN: IDENTIFICAR PERSONA

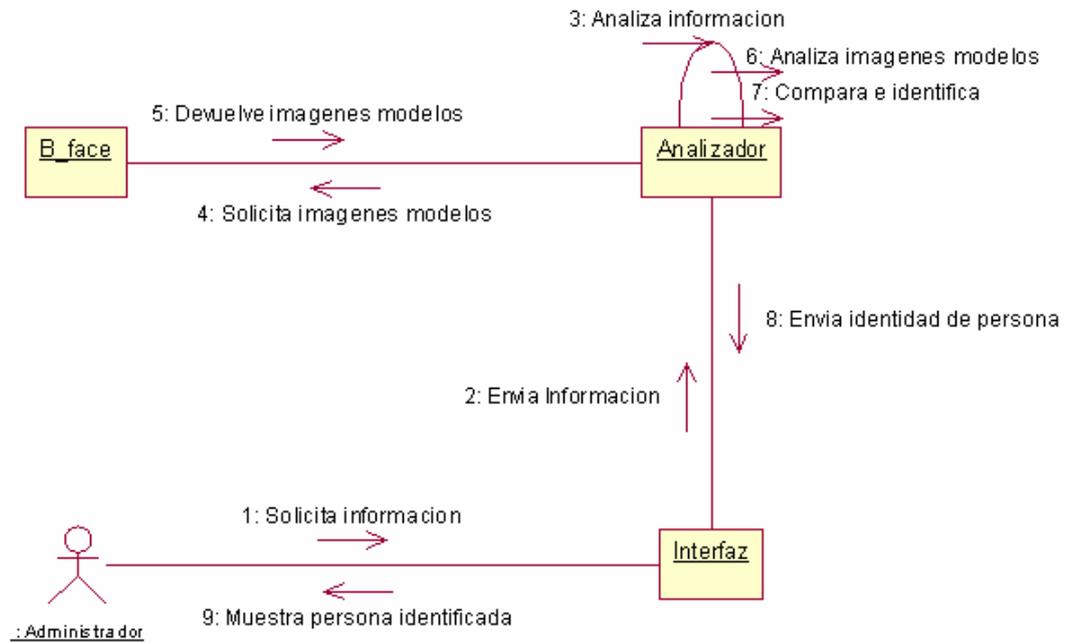


Figura 4.2.3.7. Diagrama de Colaboración de Identificar Persona

4.2.3.8. DIAGRAMA DE COLABORACIÓN: IMPRIMIR REPORTE

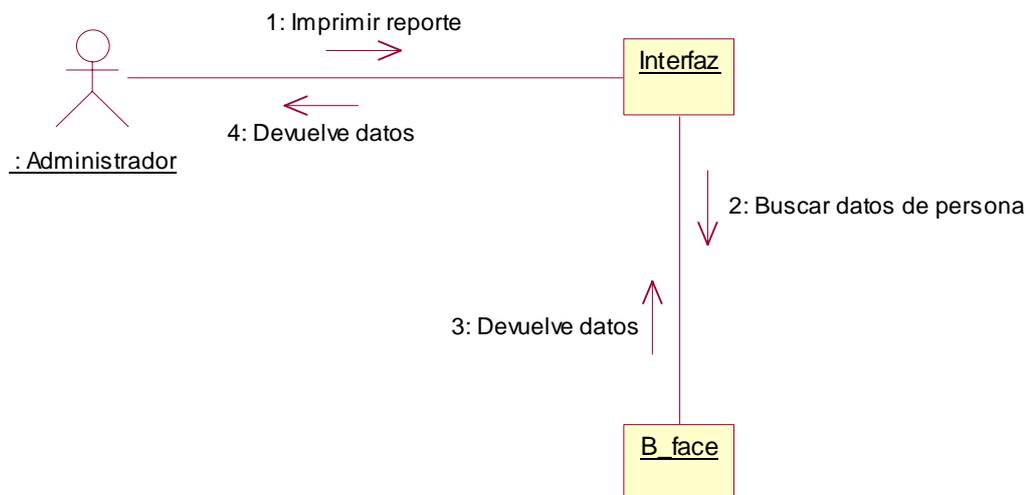


Figura 4.2.3.8. Diagrama de Colaboración de Imprimir Reporte

4.2.4. DIAGRAMA DE CLASES

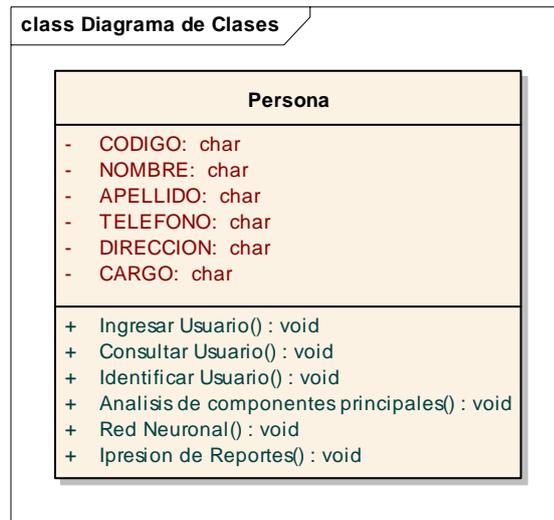


Figura 4.2.4. Diagrama de Clases del Sistema SIFARE.

4.3. IMPLEMENTACIÓN

La codificación del sistema fue realizado de acuerdo a las especificaciones descritas por el usuario en las etapas de análisis y diseño del sistema, los script creados para el sistema están realizados con el lenguaje de programación de MATLAB 7.2.

4.3.1. EJEMPLO DE EXTRACTO DE CÓDIGO PARA EL ANÁLISIS DE COMPONENTES PRINCIPALES DE UNA IMAGEN.

```
% Cargar la Imagen
Ing_imagen = imread('foto.jpg');

% Se amplia los valores de la imagen a un formato estándar de matriz de 250x250
Ing_imagen = imresize(Ing_imagen,[250 250]);

% Transformar la imagen a un vector de 62500 x 1 debido a que se formara un
contenedor con todas las imágenes transformadas.
Ing_imagen = reshape(temporal',ifila*icolumna,1);
```

%Calculamos la media de la matriz de imágenes para luego calcular la desviación de cada imagen con respecto a la media y finalmente lo almacenamos en un contenedor.

```
m = mean(Ing_imagen,2);  
temp = double(Ing_imagen(:,i)) - m;  
A = [A temp];
```

Se procede a utilizar el método de eigenvalores para obtener el PCA de las imágenes de la base de datos.

%Se obtiene la matriz de covarianza y la diagonal de elementos para luego eliminar los eigenvalores menos significativos.

```
L = A'*A;  
[V D] = eig(L);
```

% Se calcula los eigenvectores de la matriz de covarianza así se obtiene los vectores únicos de la matriz de imágenes para luego ser entrenados en la red neuronal.

```
L_eig_vec = [L_eig_vec V(:,i)];  
Eigenfaces = A * L_eig_vec;
```

%Se utiliza el mismo procedimiento para la imagen que se usara para el reconocimiento de rostros, los vectores únicos obtenidos se almacenaran en el vector ProjectedTestImage.

4.3.2. EJEMPLO EXTRACTO DE CÓDIGO PARA CREAR LA UNA RED NEURONAL

Una vez obtenido los vectores únicos de la imagen, se crea la red neuronal con los vectores únicos de cada imagen dentro de la base de imágenes, luego se procede a entrenar la red.

La red neuronal que se utilizara es backpropagation debido que en pruebas realizadas se ha obtenido buenos resultados, para el sistema se utilizara la siguiente arquitectura:

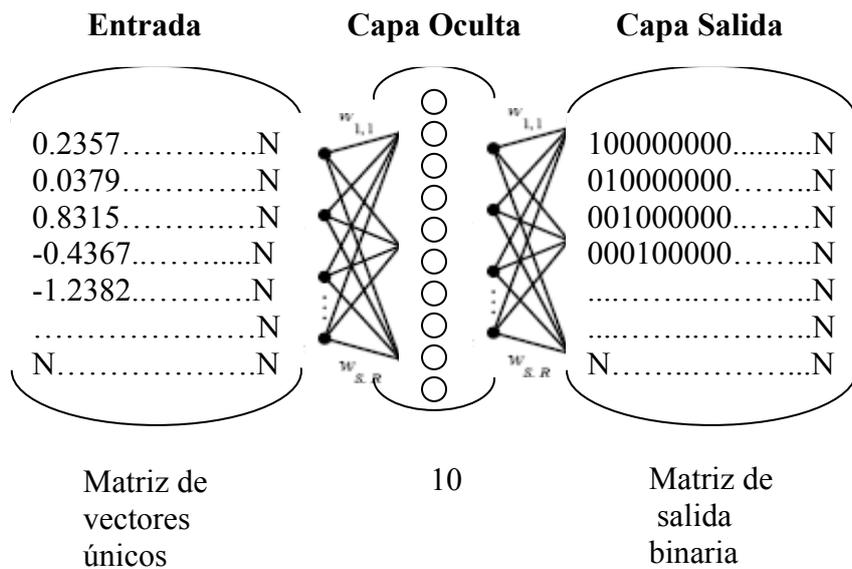


Figura 4.3.2.1. Arquitectura de la red neuronal

En donde:

El número de entradas es la cantidad de elementos de la matriz de vectores únicos que se obtuvo con el análisis de componentes principales.

La cantidad de neuronas en la capa oculta se ha determinado que sea 10 debido que después de varias pruebas donde se uso esta cantidad de neuronas tuvo un gran rendimiento.

La capa de salida se determina con el numero de imágenes que se encuentran almacenadas en la base de datos.

El objetivo del PCA es dar un valor único al rostro y la red neuronal es determinar que datos en la red llegan a corresponder con la persona que va a ser identificado.

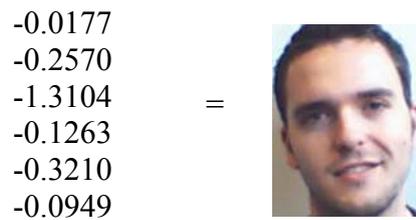


Figura 4.3.2.2. Representación de la imagen con vectores únicos

El proceso de identificación de un rostro utilizando redes neuronales es:

% Crear matriz de salida para la red neuronal debido a que depende del numero de imágenes que estén en la base de datos(es el mismo numero de salidas) y crean una matriz de salidas binarias.

```
dat_salida = [eye(N)];
```

%Crear red neuronal con la matriz de vectores únicos como entrada, 10 neuronas de capa oculta y como capa de salida la cantidad de imágenes que se encuentran almacenadas.

```
net = newff(minmax(Mat_vec_unicos),[10 N],{'tansig' 'logsig'},'trainlm');
```

%Equilibrar los pesos de la red Neuronal

```
net.LW{2,1} = net.LW{2,1}*0.01;
```

```
net.b{2} = net.b{2}*0.01;
```

%Determinar las funciones de la red neuronal que se requieren para su entrenamiento y esta llegue a su meta, mostrando el estado de entrenamiento cada 20 epocas hasta 5000.

```
net.trainParam.goal = 0.1;
```

```
net.trainParam.show = 20;
```

```
net.trainParam.epochs = 5000;
```

%Entrenar la red neuronal con los datos de la matriz de vectores únicos y las salidas binarias.

```
[net,tr] = train(net,Mat_vec_unicos,dat_salida);
```

%Determinar si los vectores únicos de la imagen ingresada que esta almacenada en contenedor ProjectedTestImage coincide con los vectores únicos de una imagen que se encuentre en la red neuronal.

```
[a,b]=max(sim(net,ProjectedTestImage));
```

Si la imagen esta en la base de imágenes la red indicara el código que posee la imagen y despliega los datos que se encuentran almacenados.

4.4. PRUEBAS

Se realizo pruebas al sistema para determina el nivel de error que este posee y determinar si el sistema es fiable.

4.4.1. PRUEBA DE LAS HIPÓTESIS

La tesis es de investigación científica y sus resultados son de característica cualitativa y jerárquica, por lo que se escogió la prueba de Chi cuadrado para la demostración de las hipótesis, siendo pasos a seguir:

- Establecimiento del nivel de significancia
- Selección de la muestra
- Pasos para la recolección de datos
- Tabulación de datos
- Calculo del estadístico
- Grados de Libertad
- Decisión

4.4.1.1. ESTABLECIMIENTO DEL NIVEL DE SIGNIFICANCIA

Se realiza las pruebas al sistema de reconocimiento de rostros y se toma el 5%, que es la probabilidad de cometer un error tipo 1, es decir se rechaza la hipótesis de que el sistema tiene un porcentaje aceptable de eficiencia siendo esta verdadera, la misma que dará el límite para la zona de aceptación y rechazo en la curva de distribución chi cuadrado.

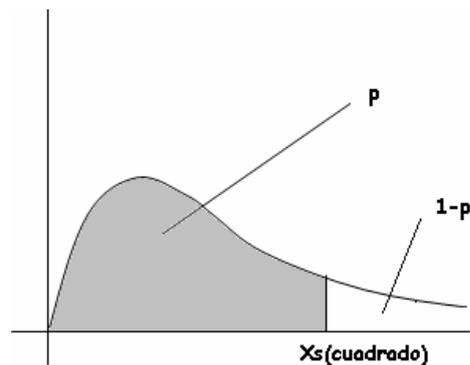


Figura 4.4.1.1. Curva de distribución del chi cuadrado

4.4.1.2. SELECCIÓN DE LA MUESTRA

La muestra fue seleccionada en una base de datos especial, para las pruebas se uso un total de 20 personas con 10 diferentes tipos de imágenes digitales.

4.4.1.3. PASOS PARA LA RECOLECCIÓN DE DATOS

Se elaborará las pruebas con las variables independientes las cuales son:

Normalización: cuando todas personas en las imágenes digitales tienen el mismo formato de fondo y perfil.

Características especiales: cuando las personas en la imagen tienen una característica propia (bigote, barba) o no propia como (lentes y objetos extraños)

Luz: cuando los factores de iluminación se encuentran dentro de patrones normales.

Conocimiento: cuando el administrador tienen un conocimiento básico del sistema

Estas variables son claves ya que el sistema depende de la extracción de características, con la técnica de análisis de componentes principales bajo estas condiciones y determina su eficiencia.

Estas variables serán calificadas con "Muy Buena, Buena, Regular, Deficiente", las mismas que están definidas como:

Muy Buena: cuando el proceso matemático que realizan los algoritmos correspondientes a esa variable dan como resultado una muy buena respuesta o salida.

Buena: cuando el proceso matemático que realizan los algoritmos correspondientes a esa variable dan como resultado bueno o aceptable de respuesta o salida.

Regular: cuando el proceso matemático que realizan los algoritmos correspondientes a esa variable dan como resultado respuesta o salida que no satisface las expectativas.

Deficiente: cuando el proceso matemático que realizan los algoritmos correspondientes a esa variable dan como resultado una respuesta o salida equivocada.

4.4.1.4. TABULACIÓN DE DATOS

En función de de las pruebas aplicadas se obtuvieron los resultados presentados en la tabla de contingencia.

		Muy Bueno	Bueno	Regular	Deficiente
Normalización	Fondo	20	0	0	0
	Perfil	6	9	4	1
Características Especiales	Lentes	15	2	2	1
	Barba	14	4	2	0
	Objetos	8	6	5	1
Luz	Brillo	18	2	0	0
	Intensidad	17	2	1	0
Conocimiento	ACP	11	7	2	0
	Redes Neuronales	9	9	2	0

Tabla 4.4.1.4. Contingencia de los resultados de las pruebas

4.4.1.5. CÁLCULO DEL ESTADÍSTICO

Se establece una ponderación para cada variable independiente en cada subprograma tomando en cuenta los siguientes factores:

- Importancia.
- Complejidad matemática
- Parámetros conocidos.

		Porcentaje de Ponderación	Muy Bueno	Bueno	Regular	Deficiente
Normalización	Fondo	90	18,0	0,0	0,00	0,00
	Perfil	10	0,6	0,9	0,40	0,10
Características Especiales	Lentes	70	10,5	1,4	1,40	0,70
	Barba	25	3,5	1,0	0,50	0,00
	Objetos	5	0,4	0,3	0,25	0,05
Luz	Brillo	80	14,4	1,6	0,00	0,00
	Intensidad	20	3,4	0,4	0,20	0,00
Conocimiento	ACP	60	6,6	4,2	1,20	0,00
	Redes Neuronales	40	3,6	3,6	0,80	0,00

Tabla 4.4.1.5.1. Contingencia ponderada

Luego de la ponderación realizada se aplica la media ponderada, estableciendo el valor medio de cada uno de las variables independientes calificadas en las pruebas por los estudiantes y tabulado en la siguiente tabla .

	Muy Bueno	Bueno	Regular	Deficiente	Total
Normalización	18,6	0,9	0,40	0,10	20,00
Características Especiales	14,4	2,7	2,15	0,75	20,00
Luz	17,80	2,00	0,20	0,00	20,00
Conocimiento	10,20	7,80	2,00	0,00	20,00
Total	61,0	13,4	4,75	0,85	80,00

Tabla 4.4.1.5.2. Frecuencias observadas

Luego se determinan las frecuencias teóricas esperadas, necesarias para aplicar el método del chi cuadrado; para determinar dichas frecuencias se aplicará la siguiente expresión:

$$f_e = \frac{\text{total variable independiente} \times \text{total variable dependiente}}{\text{total de resultados}}$$

Ejemplo:

$$\frac{20 \times 61,0}{80} = 15,25$$

	Muy Bueno	Bueno	Regular	Deficiente	Total
Normalización	15,25	3,35	1,1875	0,2125	20,0
Características Especiales	15,25	3,35	1,1875	0,2125	20,0
Luz	15,25	3,35	1,1875	0,2125	20,0
Conocimiento	15,25	3,35	1,1875	0,2125	20,0
Total	61,0	13,4	4,75	0,85	80,0

Tabla 4.4.1.5.3. Frecuencias teóricas esperadas

Finalmente se determina el valor de chi cuadrado con la siguiente expresión:

$$X^2 = \sum \frac{(f_o - f_e)^2}{f_e}$$

$$\begin{aligned}
 X^2 &= 0,7359 + 0,0474 + 0,4264 + 1,6723 + 1,7918 + 0,1261 + 0,5440 + 5,9112 \\
 &+ 0,5222 + 0,7801 + 0,8212 + 0,5559 + 0,0596 + 1,3596 + 0,2125 + 0,2125 \\
 X^2 &= 15,7787
 \end{aligned}$$

4.4.1.6. GRADOS DE LIBERTAD

La determinación de los grados de libertad, está en función del número de filas y el número de columnas, con la siguiente expresión:

$$\begin{aligned}
 g.l &= (\# \text{ columnas} - 1) * (\# \text{ filas} - 1) \\
 g.l &= (4 - 1) * (4 - 1) = 9
 \end{aligned}$$

4.4.1.7. DECISIÓN

Con el nivel de significancia de valor 0.05, y el valor del grado de libertad se obtiene en la tabla del chi cuadrado el valor límite entre la zona de aceptación y las zonas de rechazo; se obtiene el valor de 15,7787.

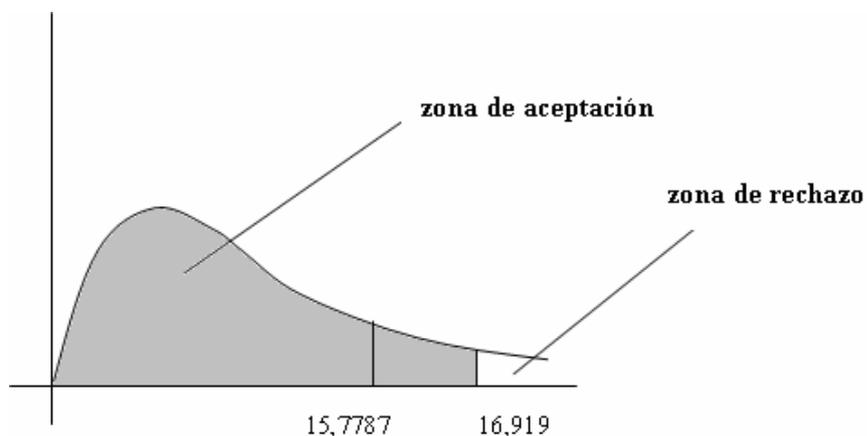


Figura 4.4.1.2. Zona de Aceptación de la hipótesis

Por tanto: $15,7787 < 16,919$

Se Encuentra en la zona de aceptación, por tanto se acepta la hipótesis nula.

Debido que en las pruebas se determino que tiene 15,7787 dentro de la zona de aceptación esto equivale a que el sistema tiene un 95% de efectividad.

4.5 IMPLANTACIÓN

La implantación del sistema se realizo de acuerdo a las especificaciones descritas en las etapas de análisis y diseño del sistema, los ejecutables creados para el sistema se encuentran en la carpeta work del disco provisto en la tesis.

4.5.1. PRERREQUISITOS PARA LA INSTALACIÓN

Para instalar el sistema es necesario conocer los requisitos previos los cuales son necesarios y suficientes para qu el sistema pueda instalarse, es recomendable usar versiones de software estipuladas en este documento ya que el sistema ha sido testeado únicamente con estas versiones.

Para instalar el sistema es necesario:

- Windows XP profesional Servipack 2
- DirectX versión 9.0 o más.
- Matlab 7.2.
- Delux Web Cam Driver

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las conclusiones y recomendaciones a las cuales se ha llegado luego de haber finalizado el presente trabajo de investigación.

5.1. CONCLUSIONES

Como conclusiones se puede decir:

- El PCA es una técnica que permite extraer las características principales de un objeto, con esta técnica y la aplicación de una red neuronal (backpropagation) se implemento un sistema de reconocimiento de rostros, que fue el objeto de estudio de este proyecto de investigación, realizadas las prueba se obtuvo un porcentaje de error del 5% (falsas apreciaciones), lo que implica que el sistema desarrollado esta dentro del margen de aceptación.
- En el estudio del reconocimiento de rostros. Se realizó una importante búsqueda bibliografica sobre las variadas técnicas existentes, intentando además introducirse en problemas relacionados con la detección de rostros, y la extracción de características. Se profundizó en el estudio del análisis de componentes principales (PCA) de una imagen cuyo principio es extraer una serie de características:
Dentro del cual se sigue el siguiente procedimiento
 - Transformar la imagen en una matriz
 - Obtener la media de la matriz
 - Determinar la matriz de covarianza
 - Determinar la diagonal de elementos
 - Extraer los eigenvectores de la matriz de covarianza
 - Crear matriz de vectores únicos

- Aunque las bases de datos utilizadas son de un número de sujetos reducido (por el número de imágenes de prueba 20 individuos por 10 imágenes un total de 200 imágenes) permiten mostrar la validez de la aproximación considerada, mostrando que el sistema de reconocimiento de rostros puede ser una ayuda a la toma de decisiones. Las pruebas realizadas indican que es posible el procesamiento de imágenes en tiempo real, y tener un alto margen de aceptación y colocan a este sistema cerca de los mejores resultados existentes.
- Se logró desarrollar un sistema de reconocimiento de rostros que resuelve la adquisición de la imagen, la detección del rostro, la extracción de características y finalmente el reconocimiento del individuo. El sistema se desarrolló en Matlab, este sistema permite ejecutar las principales operaciones que realiza un sistema de reconocimiento de rostros en el mercado actual.
- El sistema de reconocimiento de rostros puede ser utilizado: en empresas (control de acceso a lugares restringidos), en instituciones (control de asistencia de personal), en aeropuertos (dispositivos de seguridad, cuyo objetivo es el de encontrar personas) lo que permitirá realizar un mejor control y monitoreo de las instalaciones utilizando tecnología de punta.

5.2. RECOMENDACIONES

Como recomendaciones se puede decir:

- Se de mas énfasis a materias que dan soporte a lo que es la Inteligencia Artificial, debido a que esta tecnología esta apareciendo cada vez con mas fuerza, como son la probabilidad y estadística, matemáticas discretas, procesos estocásticos, etc. en la carrera de Sistemas e Informática debido a que en muchas de estas no se indica la aplicabilidad o las aplicaciones que se pueden desarrollar.
- Se propone que la Carrera de Sistemas e Informática realice otros proyectos de tesis, relacionados con este tema de investigación, como es: la búsqueda de personas dentro de una institución o empresa la cual sería una aplicabilidad del sistema realizado, en la cual se utilicen otros métodos y técnicas que se pueden aplicar al reconocimiento de patrones.

BIBLIOGRAFÍA

Andrew Senior, Rein-Lien Hsu, Mohamed Abdel Mottaleb, and Anil K. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.

D.S. Bolme, J.R. Beveridge, M. Teixeira, and B.A. Draper. Evaluation of face recognition algorithms, última actualización: 20 de Abril de 2004.

<http://www.cs.colostate.edu/evalfacerec/algorithms5.html>.

Anil K. Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, 2004.

A. J. Goldstein, L. D. Harmon, and A. B. Lesk. Identification of human faces. *Proceedings of the IEEE*, 59(5):748–760, 1971.

M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

M-H. Yang, D.J Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, January 2002.

W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

C. Havran, L. Hupet, J. Czyz, J. Lee, L. Vandendorpe, and M. Verleysen. Independent component analysis for face authentication. In *Proceedings of the 6th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*.

W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. In *Proceedings of the 3rd International Conference on Automatic Face and Gesture Recognition*.

K. Delac, M. Grgic, and S. Grgic. Independent comparative study of pca, ica, and lda on the feret data set. *International Journal of Imaging Systems and Technology*.

Pablo Navarrete and Javier Ruiz del Solar. Eigenspace-based recognition of faces: Comparisons and a new approach. In *Proceedings of the 11th International Conference on Image Analysis and Processing*.

Baback Moghaddam, Chahab Nastar, and Alex Pentland. Bayesian face recognition using deformable intensity surfaces. In *Proceedings of the 15th Computer Vision and Pattern Recognition Conference*.

Goldberg, David. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

Haupt, Randy y Sue Ellen Haupt. *Practical Genetic Algorithms*. John Wiley & Sons, 1998.

Mitchell, Melanie. *An Introduction to Genetic Algorithms*. MIT Press, 1996.

Dillon W., M. Goldstein; *Multivariate Análisis*. John Wiley & Sons. (1984)

Escofier B., J. Pages; *Análisis Factoriales Simples y Múltiples*. [Universidad](#) del País Vasco. Bilbao (1992)

Lebart, Morineau et Piron; *Statistique Exploratoire Multidimensionnelle*. Dunod. París. 1998.

DIRECCIONES WEB

<http://www.cam-orl.co.uk/facedatabase.html>

<http://citeseer.nj.nec.com/lawrence96what.html>

<http://citeseer.nj.nec.com/update/25947citeseer.html>

www.mathworks.com

<http://www.inegi.gob.mx/difusion/espanol/niveles/jly/nivbien/componentes.html>

http://es.wikipedia.org/wiki/Análisis_de_componentes_principales#Fundamento

www.monografias.com

<http://www.advancedsourcecode.com/facedatabase.asp>

file:///J:/chi/Estadística_archivos/

http://usuarios.lycos.es/guillemat/_derived/home_cmp_Echo000_home.gif

http://sirio.psi.ucm.es/PROYECTOS/IMAGENES/base_pp.gif

<http://www.cs.cmu.edu/~tom/mlbook.html>

Anexo

A

MANUAL DE USUARIO

Información CD SIFARE V.1.0.

Anexo

B

**IMÁGENES DE PRUEBAS REALIZADAS AL
SISTEMA Y RESULTADOS**

Información CD SIFARE V.1.0.

Anexo

C

**CÓDIGO FUENTE DEL SISTEMA DE
RECONOCIMIENTO DE ROSTROS SIFARE
V.1.0.**

Información CD SIFARE V.1.0.

ESCUELA POLITÉCNICA DEL EJÉRCITO

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

DECLARACIÓN DE RESPONSABILIDAD

Yo, CHASIQUIZA MOLINA EDWIN OSWALDO

DECLARO QUE:

El proyecto de Grado titulado "RECONOCIMIENTO DIGITAL DE IMÁGENES APLICADO A ROSTROS HUMANOS BASADO EN PCA UTILIZANDO REDES NEURONALES" ha sido desarrollado en base a un profundo análisis en investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes incorporan en la bibliografía.

Consecuentemente el presente trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga 28 de Enero del 2008

Edwin Chasiquiza
C.I. N° 050276725-4

Latacunga, Enero del 2008

Sr. Edwin Oswaldo Chasiquiza Molina
C.I. 050276725-4

Ing. Edison Espinoza

**COORDINADOR DE LA CARRERA DE INGENIERÍA DE SISTEMAS E
INFORMÁTICA**

Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO