



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
ELÉCTRICA – INSTRUMENTACIÓN

**APLICACIÓN DE REDES NEURONALES PARA EL RECONOCIMIENTO
DE ROSTROS EN ENTORNOS LIBRES.**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
SANDRA BONILLA MEZA

TUTOR (ES) PRINCIPAL(ES)
DRA. TETYANA BAYDYK MYKOLAIVNA, CCADET

México D.F., Ciudad Universitaria, agosto 2016.

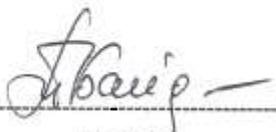
JURADO ASIGNADO:

Presidente: Dr. García Ugalde Francisco
Secretario: Dra. Medina Gómez Lucía
Vocal: Dra. Baydyk Mykolaivna Tetyana
1er. Suplente: Dr. Kussul Mikhailovich Ernst
2do. Suplente: Dr. Lara Rosano Felipe de Jesús

Lugar donde se realizó la tesis: CCADET

TUTOR DE TESIS:

DRA. TETYANA BAYDIK MYKOLAIVNA



FIRMA

MÉXICO, D. F. agosto 2016.

Agradecimientos.

A la Dra. Dra. Tetyana Baydyk Mykolaivna y al Dr. Ernst Kussul Mikhailovich por su apoyo incondicional, experiencia, invaluable aportaciones, asesoría y dirección, para llevar a cabo este trabajo. Así como el apoyo de los proyectos UNAM-DGAPA-PAPIIT IN 102014 y UNAM-DGAPA-PAPIIT IT 102814 para la realización de esta investigación.

Agradecemos también al laboratorio de Inteligencia Artificial del departamento de ingeniería eléctrica de la Fundação Educacional Inaciana (FEI), São Paulo, Brazil por la base de imágenes FEI y a la Universidad de Massachusetts por la base de imágenes LFW.

Agradezco personalmente al Programa de Escritura Académica en Inglés del Centro de Estudios de Posgrado, UNAM por la ayuda proporcionada en la revisión crítica del texto de este trabajo.

Dedicatorias.

A David Atonaltzin, mi Esposo, a mis Hijos David Edmundo y Arturo Alexis, quienes han sido el motivo y apoyo para seguir adelante.

A mis Padres y Suegros quienes nos han proporcionado una buena parte de su tiempo y recursos para alcanzar esta meta.

A mi tutora la Dra. Tetyana Baydyk Mykolaivna y al Dr. Ernst Kussul Mikhailovich, por dedicarme su tiempo y paciencia, pero sobre todo por creer en mí.

A cada uno de mis profesores a lo largo de mi trayectoria académica, por el empeño y dedicación empleados para la formación de nuevos investigadores y mejores seres humanos.

A todos mis compañeros de viaje en esta gran aventura.

Índice.

Agradecimientos.	3
Dedicatorias.	3
Índice.	4
Introducción.	6
Capítulo 1: La clasificación de Rostros Humanos.	8
1.1 Problema de investigación.	9
1.2 Métodos y técnicas empleados para la identificación de rostros humanos.,	10
1.2.1 Métodos de detección de rostros en imágenes fijas.	10
Capítulo 2: Bases de imágenes para el reconocimiento de Rostros Humanos.	24
2.1 Base de Datos Yale.	24
2.2 Base de Datos Yale Extendida B.	24
2.3 Base de Datos CAS-PEAL.	25
2.4 Base de imágenes FEI.	25
2.5 Bases de Imágenes FRAV2D y FRAV3D.	26
2.6 Base de datos FERET.	27
2.7 Base de Datos ORL.	28
2.8 Base de Imágenes LFW.	29
Capítulo 3: Clasificador Neuronal de Permutación de Códigos (Permutation Coding Neural Classifier (PCNC)).	31
3.1 Enmascaramiento.	36
3.2 Codificación.	37
3.3 Entrenamiento.	39
Capítulo 4: Experimentación y resultados obtenidos.	42
4.1 Preparación de las imágenes de trabajo.	43
4.2. Descripción del instrumento elaborado en C++.	44
4.2.1 Módulo 1: Enmascaramiento (Mask).	45

4.2.2 Módulo 2: Codificación.	46
4.2.3 Módulo 3: Entrenamiento.	47
4.2.4 Módulo 4: Reconocimiento.	48
4.3 Proceso de experimentación.	49
4.3.1 Reconocimiento holístico.	50
4.3.2 Reconocimiento de fragmentos de rostros.	51
4.3.2.1 Fragmentos de ojos y cejas (Caso I).	51
4.3.2.2 Fragmentos de boca y barbilla (Caso II).	52
4.4. Resultados	53
Capítulo 5: Conclusiones y propuestas de trabajos a futuro.	55
Apéndice .	59
Apéndice : Código del programa empleado para la aplicación de redes neuronales en el reconocimiento de rostros en entornos libres.	59
Referencias Introducción y Capítulo 1	22
Referencias Capítulo 2.	30
Referencias Capítulo 3	41
Referencias. Capítulo 4.	54
Referencias. Capítulo 5.	57

Introducción.

El reconocimiento de rostros humanos, es un tema actual que involucra diferentes disciplinas. Para la investigación en el ámbito de la visión computacional es una de las tareas más importantes en el área de reconocimiento de imágenes.

Esta importancia se deriva de su aplicación en diferentes ámbitos, tales como: seguridad, administración, industria, autenticación y etiquetado de imágenes en redes sociales, interacción robótica y biometría por mencionar algunos.

El reconocimiento de rostros humanos a partir de fotografías, se remonta a los años sesentas, mediante un sistema semiautomático diseñado por Woodrow Wilson Bledsoe fundador de Panoramic Research Inc. [1]

En el ámbito de seguridad cobra un auge importante en E.U. en el año de 2001. El departamento de defensa busca dotar a los lugares de concentración masiva con sistemas de reconocimiento de rostros como medida de seguridad y prevención para evitar tragedias multitudinarias.

Para el reconocimiento de rostros humanos, se emplean diferentes métodos, procedimientos y algoritmos, por ejemplo el método SVM (*Support Vector Machine*), métodos estadísticos, redes neuronales etc. [2]. Estos métodos y procedimientos se aplican en diferentes bases de imágenes que se han producido para estudiar el reconocimiento de rostros humanos y que han sido creadas en su mayoría bajo condiciones controladas; con lo que se facilita el estudio de parámetros específicos en el problema del reconocimiento de rostros.

En el presente trabajo se lleva a cabo la tarea de clasificación de rostros humanos, esta tarea va más allá que únicamente reconocer la existencia de rostros humanos en una imagen, se lleva a cabo la identificación de un sujeto específico. Para este efecto, partimos del hecho de que en la imagen ya se cuenta con rostros humanos, y lo que se propone es reconocer a un sujeto objetivo, sin importar las condiciones en las que la

imagen se encuentre; por ejemplo con variaciones en la pose, la iluminación, el ángulo en que se toma la imagen, la expresión facial (sonrisa, llanto, enojo, euforia etc.) y/o si se encuentra más de un rostro en una imagen.

Para resolver esta tarea, se eligió utilizar el método de alto nivel, el Clasificador Neuronal de Permutación de Códigos (*Permutation Coding Neural Classifier (PCNC)*) como método para la extracción de características en el reconocimiento de la imagen. Este método se ha probado con buenos resultados en la clasificación de imágenes proveniente del entorno real, tales como larvas de diferentes tipos de insectos y caracteres escritos manualmente. Aplicando este método, se ha observado una menor tasa de errores en comparación con el métodos de Máquinas de Soporte Vectorial (Support Vector Machines (SVM)) e iteración de puntos cercanos (*Iterative Closest Points (ICP)*) [3].

Los criterios de elección de este clasificador neuronal para llevar a cabo esta tarea fueron principalmente su robustez y eficiencia en la identificación de rostros humanos aplicados en bases de imágenes obtenidas bajo condiciones controladas. Este clasificador neuronal, se basa en el concepto del descriptor local (*Random Local Descriptor (RLD)*). El RLD funciona como un extractor general de características, estableciendo conexiones entre neuronas en la capa asociativa y puntos aleatorios de la imagen de entrada, calculando la función de brillo en el punto seleccionado [4].

Este método se aplicó a una base de imágenes extraídas del entorno natural, que se conoce como Rostros Etiquetados en Estado Salvaje (*Labeled Face in the Wild (LFW)*), sin condiciones controladas. La complejidad de esta base de imágenes radica en que estas se obtienen sin preparación alguna de los rostros a identificar, por lo que algunas de estas imágenes contienen más de un rostro, diferentes tipos de iluminación, posición, expresión facial etc. de aquí la dificultad para la identificación de los rostros objetivo [5].

Para mostrar la efectividad del método PCNC en esta tarea, se implementó un programa elaborado en C++ como un instrumento para entrenamiento de una red neuronal artificial que permitiera llevar a cabo la tarea de clasificación de los rostros objetivo.

El lograr identificar y clasificar el rostro del sujeto objetivo en cada imagen, pese a la dificultad que representa el manejo de esta base de imágenes dadas sus condiciones derivadas del entorno libre; es la aportación e innovación de este trabajo.

Capítulo 1: La clasificación de Rostros Humanos.

En el presente capítulo establecemos nuestro problema de investigación: La clasificación de rostros humanos por sujeto de estudio, partiendo de su identificación en una imagen. Se presenta también la revisión de la literatura respecto a los trabajos y métodos más relevantes respecto al reconocimiento y clasificación de rostros humanos en imágenes desde el ámbito de la visión computacional, sin perder de vista el enfoque proporcionado por las neurociencias. Toda vez que algunas de las características que se consideran para el reconocimiento de rostros, se consideran para proponer diversos modelos y métodos que se emplean en el reconocimiento de rostros en imágenes; Considerando características como las siguientes:

- La unicidad de los rostros.
- El reconocimiento holístico o por características locales.
- Análisis y uso de expresiones faciales
- El papel del hemisferio derecho del cerebro en la percepción de rostros

Esta última característica, nos permite introducir el reconocimiento de rostros mediante el modelo de redes neuronales; a partir de la evidencia de que el cerebro humano dedica un sistema específico para el reconocimiento de rostros [6].

En la actualidad existe un gran número de escenarios en donde la identificación y/o reconocimiento de personas se hacen necesarios desde el punto de vista biométrico, para diferentes estudios como sistemas de seguridad y como una tarea fundamental en el reconocimiento de patrones desde el punto de vista de la visión computacional.

Uno de los problemas típicos que presentan los sistemas de seguridad, es el registro de personal que ingresa a un establecimiento. Esta actividad es llevada a cabo en la mayoría de los casos, haciendo uso elementos como llaves, tarjetas o contraseñas para lograr la identificación de los sujetos que ingresan. Sin embargo, el uso de los anteriores elementos no es suficiente para garantizar la seguridad del sistema, puesto que es fácil suplantar identidades consiguiendo de esta manera acceso antes restringido [2].

Para llevar a cabo el reconocimiento de rostros, se han empleado diversos modelos y métodos entre los que encontramos: el uso de elementos geométricos en los rostros, el análisis estadístico, la lógica difusa, selección de componentes principales y las redes neuronales.

Algunos de los instrumentos aplicados al reconocimiento de rostros humanos, utilizan el algoritmo denominado *Eigenfaces* [7] y el método de análisis de características locales (*Local Feature Analysis (LFA)*) [8] ambos basados en el método de análisis de componentes principales.

1.1 Problema de investigación.

Como se ha mencionado, el reconocimiento de rostros es un tema actual con aplicaciones reales desde el punto de vista de la visión computacional. Sin embargo nuestro problema de investigación parte de la existencia de rostros humanos en una imagen fija; para llevar a cabo la clasificación de diversas imágenes del rostro de sujetos objetivo y establecer que estas pertenecen al mismo sujeto independientemente de las condiciones de la imagen.

Se pretende mostrar la efectividad de la aplicación de las redes neuronales para resolver la tarea de identificación de rostros en medios no controlados; aplicando el clasificador neuronal PCNC al reconocimiento facial en entornos libres considerando la base de datos LFW Mediante el entrenamiento de una red neuronal artificial, mediante el desarrollo de un instrumento que nos permita llevar a cabo el entrenamiento de una red neuronal artificial basado en el método PCNC.

A continuación, mencionamos algunos de los modelos y métodos que se han empleado para la identificación de rostros desde el punto de vista de la visión computacional.

1.2 Métodos y técnicas empleados para la identificación de rostros humanos.

En los últimos quince años se han desarrollado diversos métodos de detección de rostros, como describe Yang et al. en su revisión [9]. Este desarrollo se debe a las múltiples dificultades que existen en las imágenes, tales como:

Pose: Las imágenes del rostro varían según la pose frente a la cámara, con lo cual algunas características faciales pueden quedar ocluidas (por ejemplo, un solo ojo puede estar visible para una foto de perfil).

Presencia o ausencia de componentes estructurales: Características faciales tales como barba, bigote y lentes al estar o no estar presentes generan una gran variabilidad en el color, forma y tamaño de los conjuntos de rostros.

Expresión Facial: La apariencia del rostro es directamente afectada por la expresión en el mismo. Se pueden tener expresiones de tristeza, alegría, molestia euforia etc. Que se expresan mediante cambios sutiles, en la posición de los labios, barbilla, ojos etc.

Oclusión: Los rostros pueden estar ocluidos parcialmente por distintos objetos, por ejemplo por una bufanda. Pero también pueden estar ocluidos por otras caras, por ejemplo en una imagen de una multitud de personas.

Condiciones de la Imagen: Cuando la imagen es adquirida, condiciones tales como la iluminación y características de la cámara (tipo de sensor, tipo de lente, obturador etc.) afectan la apariencia del rostro.

A partir de estas características, podemos dividir a los métodos de detección de rostros en imágenes fijas en cuatro categorías principales de acuerdo con Yang et al. [10], sin embargo existen también métodos donde hay una clara superposición en más de una categoría:

1.2.1 Métodos de detección de rostros en imágenes fijas.

a). Métodos basados en Conocimiento: Estos métodos se basan en codificar el conocimiento humano mediante reglas. Generalmente, dichas reglas capturan relaciones entre características faciales.

b). Métodos basados en Características Invariantes: Estos métodos buscan características que no son modificadas a pesar de que el rostro esté sometido a cambios de iluminación, pose y/o ubicación de la cámara, las cuales son utilizadas para la ubicación del rostro.

c). Métodos basados en Enmascaramiento: Estos métodos utilizan la correlación entre una imagen de entrada y patrones almacenados que describen un rostro.

d). Métodos basados en Apariencia: Estos métodos utilizan modelos obtenidos mediante un conjunto de entrenamiento de imágenes, los cuales capturan una representación variable de la apariencia facial.

A continuación se explican en detalle cada una de estas categorías y se describen los métodos más característicos de cada una de ellas.

a). Métodos basados en Conocimiento.

Estos métodos de detección de cara se basan en reglas obtenidas del conocimiento que los investigadores tengan sobre cómo está constituido el rostro de una persona. Existen reglas simples que describen las características de una cara y sus relaciones. Por ejemplo, una cara aparece en una imagen con dos ojos que son simétricos entre ellos, una boca y una nariz. Las relaciones entre las características pueden representarse por las distancias relativas y posiciones.

Primero se extraen las características faciales de la imagen de entrada, y los candidatos a cara son obtenidos mediante un conjunto de reglas. Luego un proceso de verificación es usado para reducir las detecciones falsas.

El mayor problema de este enfoque está en la dificultad que existe en trasladar el conocimiento humano a reglas bien definidas. Si estas reglas son estrictas, es posible que no se detecte ninguna o muy pocas caras debido a que no se podrá satisfacer la totalidad de ellas. Y si las reglas son muy generales, existirá una gran cantidad de falsos positivos. Otra dificultad es extender esta metodología para detectar rostros en distintas poses, ya que es muy difícil definir todos los casos existentes.

Yang y Huang [9] [10] propusieron un algoritmo jerárquico basado en conocimiento. Su sistema consiste en 3 niveles de reglas. En el nivel mayor, todos los posibles candidatos a caras son buscados mediante el escaneo de una ventana sobre la imagen de entrada y aplicando un conjunto de reglas en cada ubicación. Las reglas de mayor nivel son descripciones generales de cómo luce una cara mientras que las reglas de menor nivel son descripciones de los detalles de las componentes faciales. Se crea un conjunto de imágenes de distinta resolución.

Kotropoulos y Pitas [11] presentaron un método basado en reglas similar al de Yang y Huang. Este localiza las características faciales mediante un método de proyecciones. Primero se determinan dos mínimos locales de la proyección horizontal, los cuales corresponderían al lado derecho e izquierdo de la cara. Luego, se obtiene la proyección vertical y los mínimos locales son obtenidos para determinar las ubicaciones de los labios, nariz y ojos. Luego estas características constituyen el candidato a cara.

b). Métodos basados en Características Invariantes.

A diferencia de los métodos basados en conocimiento, que buscan relaciones entre las características faciales, los investigadores han intentado buscar características invariantes de las caras para su detección. El supuesto se basa en que los seres humanos son capaces de detectar las caras u objetos en general en distintas poses y condiciones de iluminación, y por lo tanto deben existir propiedades o características invariantes sobre estas variabilidades. Muchos métodos se han propuesto con la intención de detectar primero las componentes faciales y luego inferir la presencia de

una cara. Componentes faciales tales como cejas, ojos, boca, nariz y la línea del pelo son comúnmente extraídas por detectores de borde. Utilizando las componentes faciales extraídas, se construye un modelo estadístico para describir las relaciones y verificar la existencia de una cara. El mayor problema de estos algoritmos es que las componentes faciales obtenidas de la imagen pueden estar seriamente dañadas por efectos de la iluminación, ruido y oclusión, por ejemplo los bordes de las componentes faciales pueden estar debilitados, en cambio los bordes asociados a las sombras pueden ser más fuertes. Componentes Faciales. Estos métodos se han aplicado principalmente a imágenes en escala de grises. Una imagen a color puede ser fácilmente transformada a una en escala de gris.

Leung et al. [12], desarrollaron un método probabilístico para localizar una cara en una escena compleja utilizando un detector de características locales y una correlación grafo-aleatoria. Su motivación fue formular la localización de la cara como un problema de búsqueda, en el cual la meta es encontrar un arreglo de ciertas componentes faciales, que comúnmente pertenecen a un rostro humano. Se consideran cinco características para describir una cara típica: los dos ojos, los dos orificios de la nariz, y la juntura boca-labios. Para cada par de características faciales se calcula su distancia relativa, y sobre un conjunto de imágenes, las distancias son modeladas por una distribución Gaussiana dada una imagen de prueba. Las componentes faciales candidatas son identificadas mediante la correlación de la respuesta de un filtro que pasa sobre cada pixel de la imagen y un vector de respuesta predefinido. Los dos mejores candidatos son seleccionados para buscar las otras características faciales. Dado que las componentes faciales no pueden aparecer en arreglos arbitrarios, las ubicaciones esperadas de las otras componentes faciales son estimadas usando modelos estadísticos de distancias con respecto a éstas. Entonces, las ubicaciones esperadas de las características pueden estimarse con alta probabilidad. Se forman constelaciones (grupos de características) por candidatos que están en ubicaciones adecuadas, y se determina la constelación que mejor represente una cara. Encontrar la mejor constelación es un problema en el cual los nodos de un grafo corresponden a las características de la cara, y los arcos representan las distancias entre diferentes características. El *ranking* de constelaciones se basa en una función de densidad de probabilidad de constelaciones que son cara versus las que no lo son.

En [13] Yow y Cipolla presentan otro método basado en características invariantes. La primera etapa utiliza un filtro Gaussiano de segundo orden sobre la imagen de entrada. Los Puntos de interés permiten detectar las posibles ubicaciones de las características faciales. La segunda etapa, analiza los contornos alrededor de los puntos de interés y los agrupa en regiones. Esta forma de agrupar se basa en la proximidad y similitud en orientación y forma. Las mediciones de las características de las regiones son calculadas y almacenadas en un vector de características. Dado un conjunto de entrenamiento de características faciales, se calculan los promedios y matrices de covarianza para cada característica facial. Una región de la imagen pasa a ser un candidato a característica facial válido cuando la distancia de Mahalanobis [14] entre los correspondientes vectores de características son menores que un umbral. Las características candidatas son agrupadas utilizando un modelo de conocimiento basado en cómo se relacionan las características entre sí. Finalmente, las características faciales y los grupos son evaluados mediante una red Bayesiana. La principal virtud de este método consiste en que es capaz de determinar caras con distintas orientaciones y poses.

Textura.

Las caras humanas son representadas por diferentes texturas, y esta condición puede utilizarse para separarlas de otros objetos. Dai y Nakano [15] desarrollan un método que permite inferir la presencia de una cara mediante la identificación de texturas. Para esto se calculan las texturas utilizando métodos estadísticos de segundo orden; además incorporan información de color a su modelo de texturas. Utilizando dicho modelo se diseña un esquema de escaneo que realza las zonas marcadas de color naranja, entre las cuales se encuentran las caras. La ventaja de este tipo de métodos es que permiten detectar caras con lentes y barba.

Color de Piel.

El color de la piel humana es una característica efectiva en muchas aplicaciones, tales como detección de rostro y seguimiento de manos. Como el color de piel no es suficiente para detectar rostros, estos métodos deben combinar el color de piel con información de movimiento y/o análisis de contornos. Jones y Rehg [16] desarrollan un método que

permite la detección de piel utilizando modelos estadísticos de color. En su trabajo presentan dos implementaciones: una basada en histogramas y otra en suma de gaussianas (MoG: Mixture of Gaussian). En ambos casos se determina la probabilidad de encontrar piel y la de no encontrar piel, pero de distinta manera. En el caso de los histogramas se utiliza un conjunto de entrenamiento, en el cual las regiones de piel son segmentadas manualmente y son agregadas en un acumulador y de forma similar para las regiones de ausencia de piel. En el caso de MoG dichas segmentaciones manuales se utilizan para obtener un modelo compuesto por la suma de 12 gaussianas que representan la probabilidad de piel y ausencia de piel, respectivamente. Se calcula la razón de ambas probabilidades (piel y ausencia de piel), para determinar mediante un umbral si el pixel corresponde a piel o no. Dicho umbral es determinado por el usuario: un umbral riguroso permite muy pocos falsos positivos, pero detecta menos regiones de piel. Recientemente se han desarrollado modificaciones a este método, agregando la característica que los píxeles de piel deben ser vecinos o estar dentro de una vecindad; esta modificación fue propuesta por Ruiz del Solar y Verschae en [17].

C). Métodos basados en Enmascaramiento.

En los métodos basados en enmascaramiento, se predefine manualmente un patrón de una cara estándar. Dada una imagen de entrada, se calcula la correlación entre la imagen y el patrón o máscara para el contorno de la cara, ojos, nariz y boca independientemente. La existencia de una cara es determinada utilizando los valores de la correlación. Este tipo de métodos tienen la cualidad de que son fáciles de implementar. Sin embargo, una sola máscara no es capaz de manejar variaciones de pose, de escala y de forma. Para resolver los problemas de invariancia de escala y forma se proponen máscaras de múltiple-resolución, multiescala, y máscaras deformables como se detalla a continuación.

Máscaras Predefinidas.

Craw et al. [18], presentan un método de localización utilizando una máscara de una vista frontal del rostro. Primero se determinan los bordes de la imagen mediante un filtro

de Sobel. Una máscara del contorno del rostro se ubica sobre la imagen para determinar los bordes que pertenecen a un rostro (un candidato a rostro). Luego se repite el mismo procedimiento para todos los candidatos, pero ahora con máscaras que determinan ojos, cejas y labios. En un trabajo posterior Craw et al. [19], describen un método de localización que utiliza un conjunto de 40 máscaras para buscar características faciales y una estrategia de control para guiar y evaluar los resultados de los detectores basados en máscaras.

Pérez et al. [20], presentan un método de detección de rostro y de iris utilizando múltiples máscaras que permite la detección de rostro en imágenes de video. El método utiliza máscaras antropométricas basadas en características tales como el contorno elíptico del rostro, la ubicación de las cejas, labios y nariz. Este método está probado para imágenes de video de fondos no controlados, y con rotaciones del rostro en el eje coronal entre los $-40\pm$ y 40 . En la Fig.1.1 se muestran las máscaras utilizadas por Pérez et al. [21] para rotaciones coronales de $-40\pm$, $0\pm$ y $40\pm$.

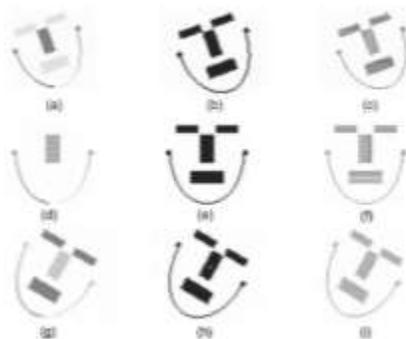


Fig.1.1 máscaras utilizadas para rotaciones coronales de $-40\pm$, $0\pm$ y $40\pm$.

Máscaras Deformables.

Yuille et al. [22], utilizan máscaras deformables para modelar características faciales que se ajusten a un modelo elástico a priori de características faciales. En este método las características faciales se describen por máscaras paramétricas. Se define una función de energía para enlazar los bordes, picos y valles en la imagen de entrada con su correspondiente parámetro en la máscara. Luego, se busca el mejor ajuste al modelo elástico mediante la minimización de los parámetros de la función de energía. Los

resultados experimentales muestran un buen rendimiento en el seguimiento de características no-rígidas. Una desventaja de este método es que debe ser inicializado en las proximidades del objeto de interés.

d). Métodos basados en Apariencia.

Al contrario de los métodos basados en máscaras, los cuales deben ser predefinidos por expertos, las *_máscaras_* en los métodos basados en apariencia son aprendidas mediante un conjunto de patrones elegidos. Por ejemplo, la máscara al utilizar una red neuronal corresponderá a los pesos ajustados de ésta. En general, los métodos basados en apariencia se basan en técnicas de análisis estadístico y *machine learning* para encontrar las características relevantes de imágenes de cara y no-cara. Las características aprendidas corresponden a la forma de los modelos de distribución o las funciones discriminantes que son utilizadas para la detección del rostro. Mientras tanto, se lleva a cabo una reducción de dimensionalidad para mejorar la eficiencia computacional y la eficacia de la detección.

Diversos métodos basados en apariencia se basan a probabilidades. Una imagen o un vector de características derivado de una imagen, se identifica como una variable aleatoria x , y esta variable aleatoria es caracterizada como cara y no-cara mediante una función de densidad de probabilidad condicional $P(x/cara)$ y $P(x/no\ cara)$, respectivamente. Otros métodos basados en apariencia buscan una función discriminante entre las clases cara y no-cara, como por ejemplo una superficie de decisión, hiperplanos separadores, funciones umbrales, etc. Convencionalmente, los patrones de la imagen son proyectados a un espacio de menor dimensionalidad de tal forma que se obtenga una función discriminante para clasificación, o bien una superficie de decisión no-lineal puede ser formada mediante una red neuronal multicapa (MLP).

Entre los métodos basados en apariencia se encuentran: *Eigenfaces* [23], métodos basados en distribuciones, redes neuronales [24], máquinas de soporte vectorial (SVM) [25] [26], Hidden Markov Models (HMM) [27] y cualquier otro método que permita el aprendizaje mediante ejemplos. Si bien los distintos modelos basados en la apariencia pueden variar en construcción, su naturaleza básicamente es la misma.

Eigenfaces.

Sirovich y Kirby [28] demostraron que las imágenes de rostros pueden ser codificadas linealmente utilizando un número reducido de imágenes bases. La demostración se basó en la transformada de Karhunen-Loève, la cual también es conocida por otros nombres, por ejemplo análisis de componentes principales (PCA) y transformada Hotelling. Dado un conjunto de imágenes de entrenamiento de tamaño $n \times m$, donde cada imagen es representada por una matriz de tamaño $m \times n$, se determina un subespacio óptimo de vectores base tal que el error cuadrático medio (MSE) entre la proyección de las imágenes de entrenamiento sobre el subespacio óptimo y las imágenes originales sea minimizado. Al conjunto de vectores base óptimos se les denomina eigenvectores debido a que corresponden a los vectores propios de la matriz de covarianza de las imágenes vectorizadas del conjunto de entrenamiento.

Turk y Pentland [23] utilizaron el PCA para detección y reconocimiento de rostro, de forma similar a Kirby y Sirovich [28]. El PCA es utilizado para determinar las eigenfaces que generan un subespacio (llamado *face-space*) del espacio de imágenes. Las imágenes de rostros son proyectadas en este subespacio y agrupadas en *clusters*. De igual forma, las imágenes de no-rostro son proyectadas sobre el mismo subespacio y agrupadas en *clusters*. Para determinar la presencia de un rostro en una imagen se calcula la distancia entre una región de la imagen y el *face-space* para todas las regiones de la imagen. Dicha distancia se utiliza como medida de *faceness*, y las distancias entre todas las regiones de la imagen y el *face-space* generan un mapa de rostro (*facemap*).

Redes Neuronales.

Las Redes Neuronales han sido aplicadas en muchos problemas de reconocimiento de patrones, tales como reconocimiento de caracteres, reconocimiento de objetos, y conducción de robots autónomos. La detección de rostro se puede tratar como un problema de reconocimiento de patrones de 2 clases (cara vs. no-cara), para lo cual se propusieron muchas arquitecturas de redes neuronales. La ventaja de utilizar redes neuronales para la detección de rostro es que es posible entrenar un sistema que capture las características complejas que describen el rostro.

Sin embargo, encontrar la mejor arquitectura de red neuronal (número de capas, número de nodos, tasas de aprendizaje, etcétera) que permita tener un buen rendimiento es costoso. Existen muchos métodos de detección de rostro basados en redes neuronales. Uno de los trabajos iniciales en este aspecto es el realizado por Rowley et al. [24]. Rowley utilizó una red neuronal multicapa para aprender de un conjunto de imágenes etiquetadas como cara y no-cara. En una primera etapa la imagen original genera una pirámide de imágenes de entrada con resoluciones menores a la imagen original. En una segunda etapa se extraen todas las ventanas de 20x20 píxeles de todas las imágenes de entrada, las cuales pasan a una etapa de pre-procesamiento dónde se realiza una compensación de iluminación seguida de una ecualización basada en histograma. Finalmente, las imágenes pasan por la red neuronal, la que recibe como entrada los valores obtenidos de distintas configuraciones de campos receptivos. Una crítica que se realiza a este tipo de método es que solamente detectan rostros sin rotación, pero extiende su método para aceptar imágenes con rostros rotados, para lo cual utiliza otra red neuronal para determinar la posible orientación del rostro y rotar la ventana.

Máquinas de Soporte Vectorial.

Las Máquinas de Soporte Vectorial (Support Vector Machines (SVM)) fueron desarrolladas por Vapnik [25] y aplicadas por Osuna et al [26]. Mientras muchos métodos entrenan clasificadores (por ejemplo, las redes neuronales) que se basan en la minimización del error de entrenamiento, las SVM se basan en elegir una frontera entre dos clases que minimice el error de generalización. Se dice que un clasificador SVM es lineal cuando separa las clases mediante un hiperplano, el cual es definido por un pequeño sub-conjunto de vectores de entrenamiento denominados vectores de soporte. Estimar el hiperplano óptimo es equivalente a resolver un problema de optimización de programación cuadrática. Sin embargo, computacionalmente es costoso en tiempo y memoria. Osuna et al. [26], desarrollan un método para entrenar SVM para problemas de gran escala, y lo aplican a detección de rostro utilizando dos conjuntos de prueba de 10 millones de patrones de 19x19 píxeles.

Waring y Liu [29] presentan un método de detección de rostro basado en histogramas

espectrales y SVM. Los histogramas espectrales son histogramas de distintas imágenes filtradas de ventanas de la imagen original, los cuales entregan una representación diferente para distintos objetos que puedan estar contenidos en la imagen, un ejemplo de ello se muestra en la Fig.1. 2. Dichos histogramas forman vectores de características, los que son utilizados para entrenar una SVM que permite la separación de cara vs. no-cara.

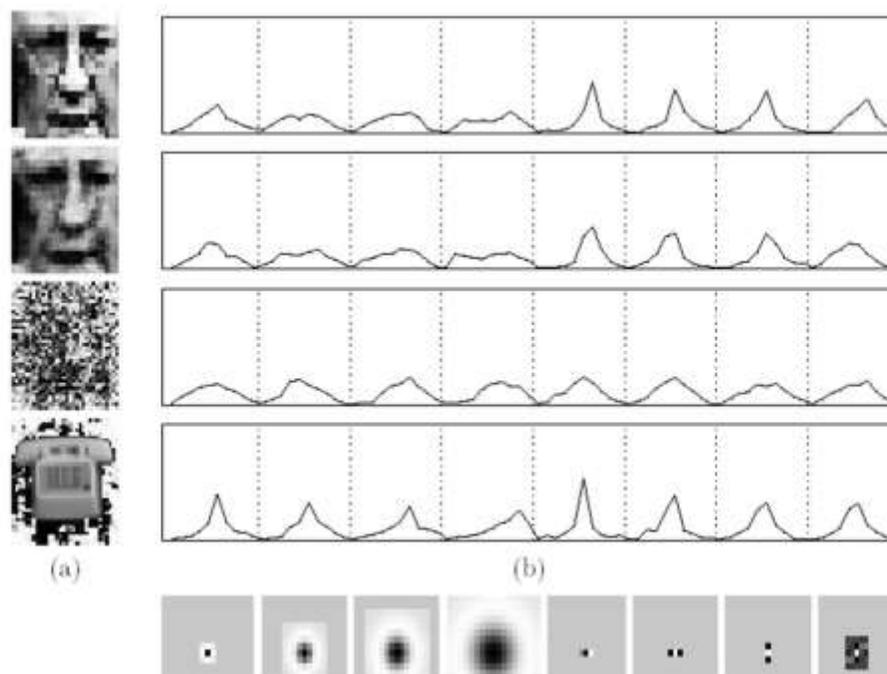


Fig.1. 2 Histograma espectral para diversas imágenes

Hidden Markov Models.

Rajagopalan et al. [27], presentan un método de detección de rostro, en el cual se utiliza un Hidden Markov Model (HMM) para aprender transiciones de rostro a no-rostro y viceversa dentro de una imagen. Los parámetros del HMM representan una secuencia de observación de la imagen, dicha secuencia se construye al computar las distancias a los centros de 12 clusters de cara y no-cara, los que fueron estimados calculando las distribuciones de probabilidad.

Redes Neuronales Min-Max Difusas.

Las redes neuronales min-max difusas FMM fueron propuestas inicialmente por Simpson [30]. Se basan esencialmente en conjuntos difusos del tipo hipercajas. Las hipercajas definen una región en el espacio n-dimensional de los patrones de entrada solamente con dos puntos (mínimo y máximo) como se muestra en la Fig.1.3.

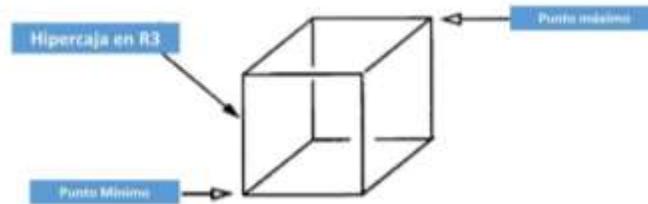


Fig.1. 3 Puntos que definen la hipercaja.

Los patrones contenidos al interior de la hipercaja tienen pertenencia completa a dicha hipercaja, es decir toman el valor 1. En cambio, los patrones que no se encuentran dentro de la hipercaja tendrán pertenencia a esa hipercaja entre 0 y 1, dependiendo de la distancia a ésta. La combinación de los puntos min-max y la función de pertenencia de la hipercaja de_nen un conjunto difuso o *cluster*. En otras palabras se puede decir que una hipercaja es una región del espacio de entrada, delimitada sólo por dos puntos, la cual tiene asociada una función de pertenencia.

El aprendizaje de las redes neuronales min-max difusas consiste en la creación, expansión y contracción de las hipercajas en el espacio de entrada, modificando la función de pertenencia asociada a cada hipercaja. El proceso de aprendizaje comienza al seleccionar un patrón de entrada y buscar la hipercaja más cercana a dicho patrón, la cual puede ser expandida para contenerlo, dependiendo de un criterio de expansión.

Existe la posibilidad que el patrón esté ya contenido en la hipercaja, y por lo tanto no sea necesario expandir la hipercaja para contenerlo. Si la hipercaja que se quiere expandir no cumple el criterio de expansión se crea una nueva hipercaja la cual consiste en ese patrón. Este proceso de crecimiento permite que los clusters existentes sean redefinidos en todo momento y que nuevos clusters sean agregados sin volver a entrenar. Uno de los defectos no deseados de la expansión de hipercajas son los traslapos entre éstas, porque causan ambigüedad al permitir que un patrón pertenezca

a dos o más *clusters* al mismo tiempo. Para eliminar los traslajos se usa una etapa de contracción.

Referencias Introducción y Capítulo 1.

- [1] C. Villegas Quezada, "Reconocimiento de rostros utilizando análisis de componentes principales", Universidad Iberoamericana, México D.F., 2005.
- [2] P. Álvarez Corrales, "Prototipo de sistema piloto para control de acceso basado en el reconocimiento de rostros", Universidad Militar Nueva Granada, Granada, España, 2013.
- [3] T. Baidyk, E. Kussul, Z. Cruz Monterrosas y A. J. Ibarra, "Face recognition using a permutation coding neural classifier", *The Natural Computing Applications Forum*, pp. 1- 15, 2015.
- [4] E. Kussul, T. Baidyk y D. C. Wunsch, «Permutation Coding Neural Classifier (PCNC)», *Neural Networks and Micromechanics*, Heidelberg, Germany, Springer, 2010, pp. 54-69.
- [5] G. B. Huang, M. Ramesh, T. Berg y E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying", University of Massachusetts, Amherst., Amherst, 2007.
- [6] V. Bruce y A. W. Young, "Understanding face recognition", *British Journal of Psychology*, 1986, 77, 305--327., vol. 77, nº 1, pp. 305-307, 1986.
- [7] L. Sirovick y M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Face", *Journal of the Optical Society of America*, vol. 4, nº 1, pp. 519-524, 1987.
- [8] P. S. Penev y J. J. Atick, "Local Feature Analysis: A general statistical theory for object representation", Computational Neuroscience Laboratory, Rockefeller University", 1996 . [En línea]. Available: <http://venezia.rockefeller.edu/group/papers/full/LFA/PenevPS.LFA.ps>. [Último acceso: 2016].
- [9] G. Yang y T. S. Huang, "Human face detection in a Scene", de *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, N.Y., 1993.
- [10] J. M. Yang, D. Kriegman y N. Ahuja, "Detecting faces in images: a survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, nº 1, pp. 34-58, 2002.
- [11] C. Kotropoulos y I. Pitas, "Rule-based face detection in frontal views en Acoustics, Speech, and Signal Processing", de *ICASSP-971 IEEE International Conference*, Munich, Alemania, 1997.
- [12] T. Leung, M. Burl y P. Perona, "Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching", de *Fifth IEEE Int'l Conf. Computer Vision*, Cambridge, Massachusetts, USA, 1995.
- [13] K. Yow y R. Cipolla, "Feature-Based Human Face Detection", *Image and Vision Computing*, vol. 15, nº 9, pp. 713-735, 1997.
- [14] Mitchell y Krzanowski, "The Mahalanobis distance and elliptic distributions", *Biometrika*, vol. 72, nº 1, pp. 464-467, 1985.
- [15] Y. Dai y Y. Nakano, "Face-Texture Model Based on SGLD and Its Application in Face Detection in a Color Scene", *Pattern Recognition*, vol. 29, nº 6, pp. 1007-1017, 1996.
- [16] M. Jones y J. Rehg, "Statistical color models with application to skin detection", *Int. Journal of Computer Vision*, vol. 46, nº 1, p. 81, 2002.

- [17] J. R. del Solar y R. Verschae, "Skin detection using neighborhood information", de *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, Southampton, UK, 2004.
- [18] I. Craw, D. Tock y A. Bennett, "Finding Face Features", de *2nd European Conference on Computer Vision*, Springer, Verlag, 1992.
- [19] D. Craw, D. Tock y A. Bennett, "Finding Face Features", de *Proc. Second European*, Aberdeen, 1992.
- [20] C. Pérez, V. Lazcano., P. Estévez. y C. Held, "Real-time iris detection on rotated faces", de *SPIE - The International Society for Optical Engineering*, San Diego, California., 2003.
- [21] C. Pérez, V. Lazcano, P. Estévez y C. Held, "Real-time iris detection on faces with coronal axis rotation", de *n Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, The Hague, Netherlands, 2004.
- [22] A. Yuille, P. Hallinan y D. Cohen, "Feature Extraction from Faces Using Deformable Templates", *Journal of Computer Vision*, vol. 8, nº 2, pp. 99 -111, 1992.
- [23] M. Turk y A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol. 3, nº 1, pp. 71-86, 1991.
- [24] H. Rowley, S. Baluja y T. Kanade, "Neural network-based face detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, nº 1, pp. 23-28, 1998.
- [25] V. Vapnik, *The Nature of Statistical Learning Theory.*, New York,: Springer-Verlag Inc., 1995.
- [26] E. Osuna, R. Freund y F. Girosi, "Training Support Vector Machines: An Application for Face Detection", de *IEEE Conf. Computer Vision and Pattern Recognition*, San Juan de Puerto Rico., 1997.
- [27] A. Rajagopalan, A. Kale y N. Cuntoor, "Gait-based recognition of humans using continuous HMMs", *Fifth IEEE international Conference on automatic Face and Gesture Recognition*, pp. 336-341, 2002.
- [28] M. Kirby y L. Sirovich, "Application of the Karhunen-Loève Procedure for the Characterization of Human Faces", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, nº 1, pp. 103-108, 1990.
- [29] C. Waring y X. Liu, "Face detection using spectral histograms and SVMs", *IEEE on Systems, Man and Cybernetics.*, vol. 35, nº 3, pp. 467-476, 2005.
- [30] P. K. Simpson, ""Fuzzy min-max neural networks. Part I. Classification", *IEEE Transactions on Neural Networks*, vol. 3, nº 5, pp. 776-786, 1992.
- C. Waring y X. Liu, «Face detection using spectral histograms and SVMs", *IEEE on Systems, Man and Cybernetics.*, vol. 35, nº 3, pp. 467-476, 2005.

Capítulo 2: Bases de imágenes para el reconocimiento de Rostros Humanos.

La tarea de reconocimiento de rostros humanos, se ha apoyado en diversas bases de imágenes que se han producido para estudiarlos. La mayoría de estas bases han sido creadas bajo condiciones controladas, para facilitar el estudio de parámetros específicos en el problema del reconocimiento de rostros. En este capítulo presentamos alguna de las bases de imágenes que se han empleado para realizarla.

2.1 Base de Datos Yale.

La base de rostros Yale, construida por Belhumeur et al. [31] está formada por 165 imágenes en escala de gris de 15 individuos. Cada sesión de 11 imágenes por individuo contempla variaciones en la expresión facial, fuente de iluminación y uso de anteojos en los individuos. En la Fig 2.1 se muestra una sesión completa para un sujeto de la base de imágenes.



Fig. 2.1: Ejemplo de sesión completa para un individuo de la base de datos de Yale.

2.2 Base de Datos Yale Extendida B.

La base de datos de rostros de Yale Extendida B construida por Georghiades et al. [32]; Lee et al. [33], contiene 16128 imágenes de 640×480 en escala de gris de 28 individuos bajo 9 poses y 64 condiciones de iluminación diferentes. Se incluye también un conjunto de imágenes recortadas con únicamente el rostro de los individuos. En la figura 2.2 se muestran algunas imágenes recortadas de la base de rostros de Yale Extendida B.



Fig. 2.2: Imágenes de la base de rostros de Yale Extendida B.

2.3 Base de Datos CAS-PEAL.

La base de imágenes de rostros CAS-PEAL construida por Cao y Shan y Gao et al [34] [35] contiene 99,594 imágenes de 1040 individuos (595 hombres y 445 mujeres) con variaciones en la pose, expresión, accesorios e iluminación.

Nueve cámaras fueron colocadas en un semicírculo para capturar simultáneamente en pose a cada individuo. También se capturaron 18 imágenes variando la pose del rostro hacia arriba y hacia abajo. Se consideraron 5 clases de gesticulación en los rostros, tales como cambio de posición en las comisuras de los labios, apertura de los ojos, cambio de posición en las cejas etc., 6 clases de accesorios (3 lentes y 3 gorras) y 15 direcciones de iluminación. En la Fig. 2.3 se muestran ejemplos de imágenes de esta base de rostros.



Fig. 2.3. Ejemplo de imágenes de CAS-PEAL

2.4 Base de imágenes FEI.

La base de imágenes FEI, es una base de rostros brasileña que contiene un conjunto de imágenes tomadas entre junio de 2005 y marzo del 2006 en el laboratorio de Inteligencia Artificial de la Fundação Educacional Inaciana (FEI) (Educational Foundation of Ignatius) en São Bernardo do Campo, São Paulo, Brasil.

Esta base está conformada por 2800 imágenes, correspondientes a 200 individuos distintos y 14 imágenes de cada uno de ellos. Todas las imágenes se tomaron a todo color contra un fondo blanco homogéneo, en posición frontal con un perfil de rotación del orden de 180° . La escala varía de alrededor del 10% del tamaño original de cada imagen que es de 640X480 píxeles. Todos los rostros corresponden principalmente a estudiantes y personal de FEI, con edades de entre 19 a 40 años con distinta apariencia, corte de cabello y accesorios. El número de sujetos hombres y mujeres es exactamente el mismo para un total de 100. En la Fig. 2.4 se muestran algunos ejemplos de las variaciones en la posición de los rostros en la base de datos FEI [36] [37]



Fig. 2.4. Ejemplo de las variaciones de la posición de los rostros en la base de datos FEI.

2.5 Bases de Imágenes FRAV2D y FRAV3D.

Las bases de datos FRAV2D y FRAV3D fueron desarrolladas por la Universidad Rey Juan Carlos en Madrid, España. La característica principal de esta base de imágenes es que cada una de ellas se presenta con diferentes tipos de inclinación, expresiones faciales y condiciones de iluminación [38].

La base de datos FRAV2D se conforma por 109 sujetos (75 hombres y 34 mujeres) con 32 imágenes a color por persona. Cada imagen tiene una resolución de 320 x 240 píxeles, la mayoría de las imágenes se presentan en posición frontal. En todos los casos las imágenes tienen un fondo azul oscuro. Las 32 imágenes fueron clasificadas en grupos de acuerdo a la pose y a las condiciones de iluminación, así como con diferentes direcciones.

La base FRAV3D se conforma por 106 sujetos con aproximadamente una mujer por 3 varones. Las imágenes se adquirieron con un scanner Minolta VIVID 700, el cual nos proporciona información respecto a la textura. Se capturaron 16 imágenes por persona con diferentes poses direcciones, gestos y cambios de luz. En cada caso se modificó sólo un parámetro por captura lo que provee una ventaja con respecto a otras bases. En la Fig. 2.5 se muestra un ejemplo de las imágenes que conforman la base de imágenes FRAV2D y FRAV3D.



Fig.2.5 Ejemplo de imágenes de la base de datos FRAV2D y FRAV3D.

2.6 Base de datos FERET

La base de datos FERET es una de las más grandes bases de imágenes públicas disponibles. El programa FERET se desarrolló entre los años de 1993 en 1997, apoyado por el departamento de control de drogas y desarrollo tecnológico a través de la agencia

de desarrollo de productos avanzados (DARPA). Su misión primaria era desarrollar capacidades de reconocimiento facial de forma automática que pudieran utilizarse como apoyo al funcionamiento de tareas relacionadas con seguridad, inteligencia y reforzamiento de la ley [39].

El conjunto de entrenamiento contiene 1,002 imágenes. La fase de prueba se tiene una galería con 1,196 imágenes provenientes de 1,196 sujetos y cuatro conjuntos de prueba (fb,fc, dup I y dup II), incluyendo expresión, iluminación y variaciones temporales. Todas las imágenes están rotadas, escaladas y recortadas en tamaños de 150X130 de acuerdo a las coordenadas oculares proporcionadas. Algunos ejemplos de las imágenes recortadas se muestran en la Fig. 2.6.



Fig.2.6 Ejemplos de rostros recortados provenientes de la base FERET.

2.7 Base de Datos ORL.

Llamada por sus creadores 'La Base de Datos de Rostros (ORL) [40]. Contiene un conjunto de 40 personas y 10 imágenes de rostro de cada una de ellas, tomadas entre Abril de 1992 y Abril de 1994 en el laboratorio AT&T de Cambridge. Esta base de imágenes de rostros se integra por 10 imágenes 40 individuos diferentes. Estas imágenes fueron tomadas en diferentes periodos de tiempo, variando la iluminación, la expresión facial y los accesorios. Todas las imágenes fueron tomadas con un fondo oscuro y homogéneo con el sujeto en posición frontal. El tamaño de cada imagen es de 92×112 píxeles en escala de grises. Un ejemplo de imágenes tomadas para un individuo de esta base de rostros se muestra en Fig. 2.7



Figura 2.7. Ejemplo de imágenes de ORL.

2.8 Base de Imágenes LFW

Rostros Etiquetados en estado salvaje: Una base de imágenes para el estudio del reconocimiento de rostros en entornos libres [41]. Se conforma por 13, 233 imágenes de rostros objetivo. Algunas de estas imágenes contiene más de un rostro, sin embargo el rostro objetivo que contiene los pixeles centrales de la imagen es el que se considera para definir el rostro de la imagen.

El problema del reconocimiento de rostros, es el reconocer a un individuo específico, más allá de únicamente detectar la presencia de un rostro humano. A partir de esta consideración, los rostros diferentes a los rostros-objetivo se ignoran como fondo.

Las imágenes se encuentran disponibles con un tamaño de 250X250 pixeles en formato JPG. La mayoría de las imágenes se encuentran en color, sin embargo algunas se encuentran solamente en escala de grises.

Una de las motivaciones para crear esta base de imágenes es proveer un gran conjunto de imágenes relativamente libre (salvaje). Donde entendemos como por libre, rostros que muestran un gran rango de variaciones que se muestran en la vida cotidiana: Variación de pose, iluminación, fondo, raza, etnicidad, edad, género, ropa, peinado, calidad de la cámara, saturación de color, foco, etc. En la Fig. 2.8 se muestran algunos ejemplos de estas variaciones.



Fig. 2.8. Ejemplos de variaciones en las imágenes de la base LFW.

Referencias Capítulo 2.

- [31] P. N. Belhumeur, J. Hespanha y D. J. Kriegman, "Eigenfaces vs. Fisherfaces", *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 19, nº 1, pp. 711-720, 1997.
- [32] A. S. Georghiades, P. N. Belhumeur y D. J. Kriegman, "From few to many. Illumination cone models for face recognition under variable lighting and pose", *IEEE Trans. Pattern Anal. Mach. Intelligence.*, vol. 23, nº 1, pp. 643-660, 2001.
- [33] K. C. Lee, J. Ho y D. Kriegman, "Acquiring linear subspaces for face recognition under variable lightin", *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 27, nº 5, pp. 684-698, 2005.
- [34] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang y D. Zhao, "The cas-peal largescale chinese face database and baseline evaluations", *IEEE Trans. on System Man, and Cybernetics (Part A).*, vol. 1, nº 1, pp. 149-161, 2008.
- [35] B. Zhang, S. Shan, X. Chen y W. Gao, "Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition", *IEEE Trans. Image Processing*, vol. 16, nº 1, pp. 57-68, 2007.
- [36] Image Processing Laboratory, department of electrical Engineering., "FEI Face Database", Centro Universitario da FEI, Sao Bernardo do Campo, Sao Paulo, Brazil, 2006.
- [37] C. E. Thomaz y G. A. Giraldi, "A new Ranking method for principal components analysis and its applications to face image analysis", *Image and Vision Computing*, vol. 28, nº 6, pp. 902-913, 2010.
- [38] E. Kussul, T. Baidyk, C. Conde y I. Cabello, "Face recognition improvement with distortions of image in training set", de *Procedings of the 2013 International Joint Conference on Neural networks.*, Dallas, Texas, USA., 2013.
- [49] P. J. Phillips, H. Moon, P. J. Rauss y S. Rizvi, "The FERET evaluation methodology for face recognition algorithms", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, nº 10, pp. 1090-1104, 2000.
- [40] A. L. Cambridge., "The ORL Database of Faces", oct 2002. [En línea]. Available: http://ftp.uk.research.att.com:pub/data/att_faces.zip <http://www.cam-orl.co.uk/facedatabase.html>. [Último acceso: diciembre 2015].

Capítulo 3: Clasificador Neuronal de Permutación de Códigos (*Permutation Coding Neural Classifier (PCNC)*)

En este capítulo se explican los fundamentos, estructura, funcionamiento y aplicación del PCNC a la tarea de clasificación de rostros humanos aplicado a la base imágenes LFW.

El método de permutación de códigos, hace posible tomar en cuenta no solo características detectadas en el reconocimiento de un rostro, sino también la posición de cada característica en las imágenes; lo que nos permite obtener una descripción de la imagen a reconocer. Las características extraídas se codifican en forma binaria, este código binario se almacena en la memoria de la computadora para cada característica. Posteriormente se codifica información acerca de la localización de esta característica en la imagen. Al utilizar permutaciones de código binario, el método basado en el clasificador neuronal de permutación de código, ha mostrado tener una mejor eficiencia, debido que es capaz de reconocer imágenes con pequeñas distorsiones o desplazamientos en sus características [42].

El clasificador neuronal PCNC se basa en el concepto del descriptor local (*Random Local Descriptor (RLD)*) [43]. El RLD funciona como un extractor general de características estableciendo conexiones entre neuronas en la capa asociativa y puntos aleatorios en la imagen de entrada, calculando la función de brillo en el punto seleccionado [44]. En la Fig. 3.1 mostramos la representación de un RLD. Cada RLD contiene varias neuronas, del orden de miles; todas las neuronas del RLD se encuentran conectas al área local de la capa S (delimitada en el rectángulo de tamaño $H \times W$).

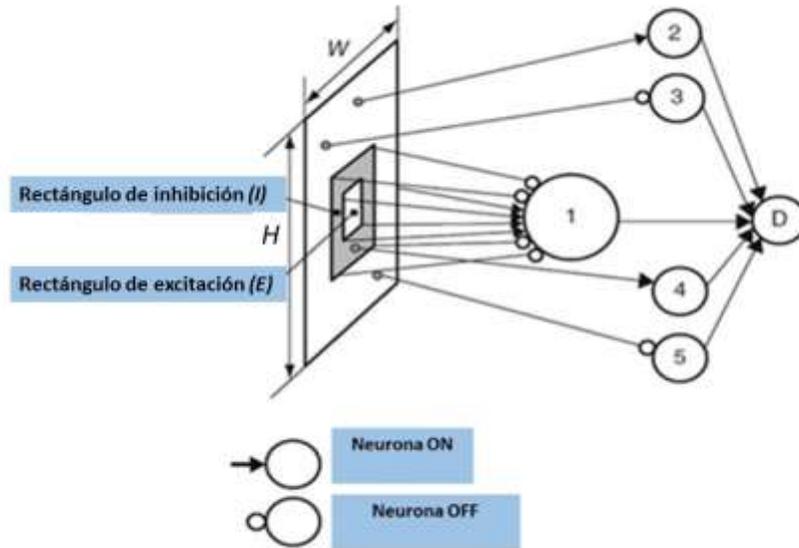


Fig. 3.1 Representación de un RLD.

Las neuronas numeradas del 2 al 5 sirven para probar los pixeles elegidos en la capa S , de forma aleatoria en el rectángulo de tamaño $H \times W$. Clasificamos a estas neuronas como neuronas ON o neuronas OFF, de forma similar a las neuronas que conforman una red neuronal natural. La salida de una neurona simple es 0 ó 1. La neurona ON tiene la salida 1 si el brillo b_i del al pixel i es mayor al del umbral T_i ($b_i \geq T_i$), mientras que la neurona OFF tiene salida 1 si el brillo b_i es menor que el umbral T_i ($b_i \leq T_i$). El umbral T_i se selecciona de forma aleatoria en el rango de brillo de la imagen.

La neurona numerada como 1 es una neurona compleja, Tiene conexiones excitatorias con todos los pixeles que conforman el rectángulo de excitación E en la Fig. 3.1, y las conexiones inhibitorias con los pixeles localizados alrededor; pero dentro del rectángulo de inhibición I . Los pesos de las conexiones pueden ser diferente en diferentes problemas de reconocimiento de imágenes. Estos pesos pueden determinarse por el sistema de reconocimiento diseñado y no son modificables durante el entrenamiento.

En el caso más simple, la neurona numerada con el número 1, puede extraer los puntos de contorno, en este caso los pesos de las conexiones excitatorias pueden ser inversamente proporcionales al área del rectángulo interno y los pesos de la conexión inhibitoria pueden ser inversamente proporcionales al área del rectángulo externo. Otro

caso simple, corresponde a la extracción de áreas de brillo de la imagen. Para este caso los pesos de las conexiones excitatorias deben ser inversamente proporcionales al área del rectángulo interno y todas las conexiones inhibitorias deben ser "0". Utilizamos el primer caso en los problemas de reconocimiento de rostros.

La salida de la neurona numerada como 1, corresponde a "1" si la suma algebraica de todas las señales de entrada es positiva y corresponde a "0" si la suma de todas las señales es negativa. Asumimos que las neuronas complejas detectarán los puntos que proporcionen mayor cantidad de información sobre la imagen.

El descriptor de la neurona D, tiene una salida igual a "1" si todas las neuronas de la 1 a la 5 tienen salida "1". Si al menos una de estas neuronas tiene salida "0" la salida de la neurona D es "0", por lo que la salida de la neurona D puede denominarse como neurona AND. Para explicar esto, consideremos la Fig. 3.2.

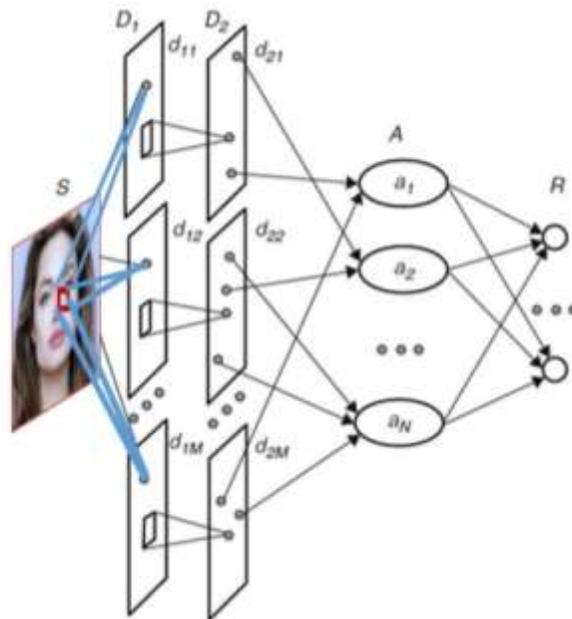


Fig. 3.2 Estructura de un sistema para el reconocimiento de imagen.

La base de este sistema es una red neuronal multicapa. La primera capa se constituye la capa de sensado S y corresponde a la entrada de la imagen. La segunda capa D_1 contiene los RLDs de bajo nivel, mientras que la capa D_2 contiene los RLDs de alto nivel. La capa asociativa A contiene elementos asociativos, los cuales pueden ser

representados por grupos de neuronas. Y finalmente la capa R que contiene a las neuronas de salida.

La capa neuronal D_1 consiste de un gran número de planos, $d_{11}, d_{12}, \dots, d_{1M}$. Cada plano contiene un número de neuronas AND igual al número de píxeles en la imagen de entrada. El plano d_{1j} preserva la topología de la imagen de entrada; lo que significa que cada neurona del plano d_{1j} corresponde a un píxel localizado en el centro del rectángulo de dimensión $W \times H$ que se observa en la Fig. 3.1. La topología de las conexiones entre la capa de sensado y las neuronas 2 a 5 en la Fig. 3.1, es la misma en el rango de cada plano d_{1j} , que se observa en la Fig. 3.2. La topología de las conexiones entre la capa de sensado y la neurona numerada como 1, es la misma para todas las neuronas en todos los planos de la capa D_1 . El objetivo de cada plano es detectar la presencia de una característica en concreto en cualquier parte de la imagen. El número de planos corresponde al número de características extraídas. Un mayor número de características, permite obtener una mejor descripción de la imagen a reconocer.

La capa D_2 en la Fig. 3.2, contiene también M planos de neuronas, cada neurona del plano d_{2j} se conecta a otras neuronas del plano d_{1j} localizado en el rectángulo. La salida de cada neurona es "1" si al menos una de las neuronas conectadas d_{1j} tiene salida "1". Esta neurona se denomina como neurona OR. La topología de las neuronas d_{2j} corresponde a la topología de las neuronas d_{1j} y a la topología de la capa S . Se ha determinado llamar a todas las neuronas que tiene salida "1" como neuronas activas por convención.

La capa asociativa A contiene N elementos asociativos que recolectan las actividades de las neuronas D_2 seleccionadas aleatoriamente. Otra de las funciones de los elementos asociativos es reducir el número de neuronas activas que sobrepasan un valor determinado. Esta función en la capa asociativa se conoce como normalización. Esta capa también desempeña la función de enlace [44], [45].

El PCNC se basa en el paradigma derivado del modelo del Perceptrón Rosenblatt [46]. La estructura del PCNC se muestra en la Fig. 3.3

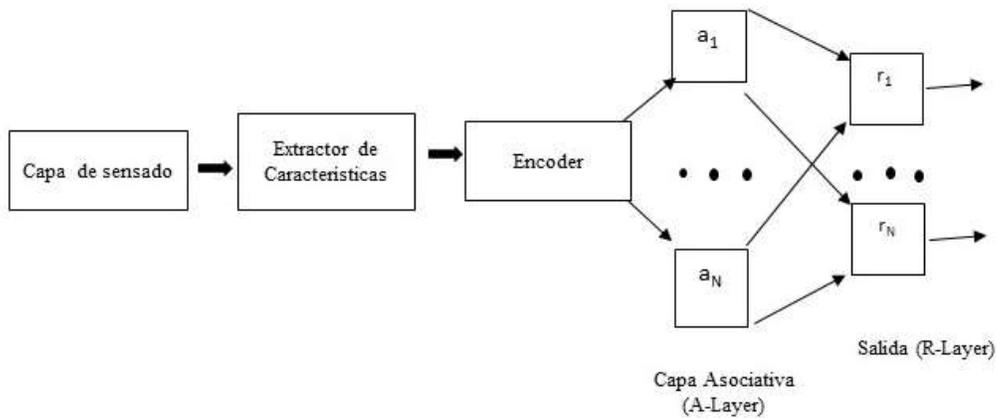


Fig. 3.3 Estructura del PCNC.

Los elementos que conforman la estructura del PCNC trabajan de forma serial. El PCNC inicia su trabajo cuando una imagen en escala de grises se presenta a la entrada del extractor de características, estas características se presentan al codificador que a su vez las transforma en un vector binario de gran dimensión. Este vector será procesado por la capa asociativa, ya sea para propósitos de entrenamiento prueba o de reconocimiento de alguna clase previamente entrenada.

3.1 Enmascaramiento.

La imagen corresponde a la entrada del extractor de características y se aplica a la entrada del encoder. El encoder produce un vector binario de grandes dimensiones, el cual es presentado a una capa de la entrada del clasificador neuronal. La salida de este clasificador nos proporciona la clase de reconocimiento.

El extractor de características inicia su trabajo seleccionando los puntos de interés en la imagen, en principio pueden proponerse varios métodos de selección de puntos, por ejemplo, los puntos de contorno pueden ser seleccionados como los puntos de interés. Múltiples características se extraen de la imagen delimitada por el rectángulo mostrado en la Fig. 3.1. Los puntos positivos p y negativos n determinan cada característica. Los puntos positivos de la capa S se conectan con la neurona ON y los puntos negativos se conectan con la neurona OFF.

Estos puntos positivos y negativos se distribuyen aleatoriamente en el rectángulo. Cada punto P_{mk} tiene un umbral T_{mk} que se selecciona de manera aleatoria en el rango:

$$T_{min} \leq T_{mk} \leq T_{max}. \quad (3.1)$$

La característica existe en el rectángulo si todas sus neuronas ON y OFF se encuentran activas, de otra forma la característica se encuentra ausente en el rectángulo establecido.

El encoder transforma las características extraídas en un vector binario:

$$V = \{v_i\} (i = 1, \dots, N), \quad (3.2)$$

donde $v_i = 0$ o 1 para cada característica extraída F_s , el encoder crea un vector binario auxiliar:

$$U = \{u_i\} (i = 1, K, N), \quad (3.3)$$

donde $u_i = 0$ o 1 .

Este vector contiene $K1$'s, donde $K \ll N$. La posición de los 1's en el vector U_s , se selecciona de manera aleatoria para cada característica F_s . Este procedimiento genera la lista de posiciones de los 1's para cada característica y guarda esta lista en la memoria. Denominamos como máscara de la característica F_s al vector U_s . En el siguiente estado del codificación, es necesario transformar el vector auxiliar U en el vector U^* , que corresponde a la posición de la característica en la imagen. Esta transformación se lleva a cabo con la permutación de los componentes del vector U .

3.2 Codificación.

El número de permutaciones depende de la localización de las características en la imagen, las direcciones de la permutación horizontal X y vertical en Y tiene diferentes esquemas de permutación y se aplican al vector U de forma secuencial. En primera instancia aplicamos la permutación X , la cual nos da el vector U'' , posteriormente aplicamos la permutación Y al vector U'' ; obteniendo como resultado el vector U^*

El problema es entonces obtener los códigos binarios de las características que se encuentran más fuertemente relacionadas cuando la distancia entre la localización de las características es pequeña y aquellas que se encuentran débilmente relacionadas si

la distancia entre características es mayor [44], [47]. Por ejemplo, si las características de F_s se extraen en un punto de la parte superior de la imagen, y la misma característica se extrae de un punto en la parte inferior de la imagen, estos deben ser codificados por diferentes vectores binarios U_{s1}^* y U_{s2}^* que tienen una correlación débil. Si las mismas características se extraen en la vecindad de los puntos, estos deben ser codificados con prácticamente el mismo vector U_{s3}^* y U_{s4}^* . Esta propiedad permite que el sistema de reconocimiento sea menos sensible a pequeños desplazamientos de los puntos en la imagen. En la Fig. 3.2 mostramos el patrón de permutación para el eje X y en la Fig. 3.3 mostramos el patrón de permutación para el eje Y.

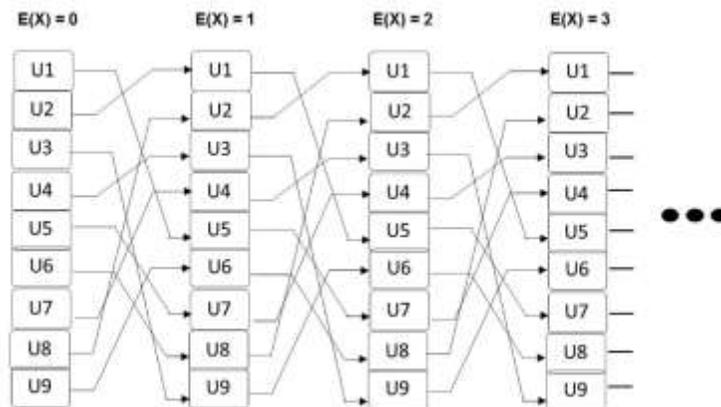


Fig. 3.2 Patrón de permutación para el eje X

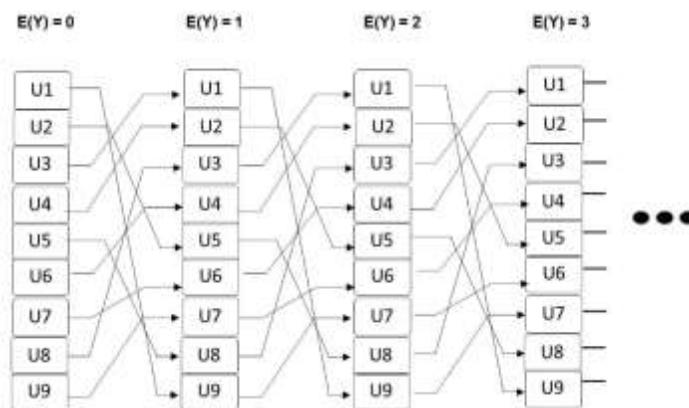


Fig. 3.3 Patrón de permutación para el eje Y.

Para codificar la localización de la característica F_s en la imagen, es necesario seleccionar la distancia de correlación D_c . Permitiendo que la misma característica F_s sea detectada en dos puntos diferentes P_1 y P_2 . Si la distancia d entre estos puntos es $d < D_c$, los códigos correspondientes serán correlacionados. Si la distancia $d > D_c$, los códigos no serán correlacionados. Para obtener esta propiedad, será necesario calcular los siguientes valores:

$$X = j/D_c , \quad (3.4)$$

$$E(X) = (int)X , \quad (3.5)$$

$$R(X) = j - E(x) \cdot D_c , \quad (3.6)$$

$$Y = i/D_c , \quad (3.7)$$

$$E(Y) = (int)Y , \quad (3.8)$$

$$R(Y) = i - E(y) \cdot D_c , \quad (3.9)$$

$$P_x = (int) \left(\frac{R(X) \cdot N}{D_c} \right) , \quad (3.10)$$

$$P_y = (int) \left(\frac{R(Y) \cdot N}{D_c} \right) , \quad (3.11)$$

donde $E(x)$ es la parte entera de X ; $R(X)$ es la parte fraccionario de X . j es la coordenada horizontal de la característica detectada y i es la coordenada vertical de la característica detectada. N es el número de neuronas y P_x , P_y son las fracciones de las neuronas para las cuales se requiere una permutación adicional [46].

La máscara de la característica F_s se considera como un código de la característica y se localiza en la esquina superior izquierda de la imagen. Para cambiar la localización de la característica en la dirección horizontal, es necesario realizar sus permutaciones $E(X)$ veces y realizar una permutación adicional para las P_x componentes del vector. Después de esto es necesario cambiar el código vertical realizando las permutaciones $E(Y)$ veces y una permutación adicional para las componentes P_y .

A continuación describiremos el proceso básico para realizar el reconocimiento de imágenes.

3.3 Entrenamiento.

En la primera etapa, el procedimiento de entrenamiento inicia con la presentación de la primera imagen al PCNC, la imagen es codificada y se calculan las neuronas excitadas E_i de la capa R . E_i se define como:

$$E_i = \sum_{j=1}^N a_j * W_{ji}, \quad (3.12)$$

donde E_i corresponde a la excitación de la i -ésima neurona de la capa R ; a_j es la excitación de la j -ésima neurona de la capa A ; W_{ji} es el peso de la conexión entre la j -ésima neurona de la capa A y la i -ésima neurona de la capa R .

En la segunda etapa, requerimos un reconocimiento robusto. Después de calcular las excitaciones neuronales de la capa R , el nombre correcto de la imagen presentada es leída del archivo correspondiente en la base de imágenes LFW. La excitación de E que corresponde a la neurona la cual se recalcula de acuerdo a la fórmula:

$$E_c^* = E_c * (1 - T_E), \quad (3.13)$$

donde c corresponde a la excitación correcta y $0 < T_E < 1$ es la excitación de reserva la cuál debe tener una neurona ganadora para decir que ganó con seguridad.

En la tercera etapa, denotamos a la neurona ganadora con el número j y el número de neurona, la cual en realidad corresponde a la entrada de la imagen como c . Si $j = c$, no se requiere realizar ninguna acción. Si $j \neq c$, se realizan los cálculos de acuerdo con las siguientes ecuaciones:

$$\begin{aligned} w_{ic}(t+1) &= w_{ic}(t) + a_i, \\ w_{ij}(t+1) &= w_{ij}(t) - a_i \text{ si } (w_{ij}(t+1)) < 0, w_{ij}(t+1) = 0, \end{aligned} \quad (3.14)$$

donde $w_{ij}(t)$ es el peso de la conexión entre la i -ésima neurona de la capa A y la j -ésima neurona de la capa R antes del reforzamiento, $w_{ij}(t+1)$ es el peso después del reforzamiento y a_i es la señal de salida (0 o 1) de la i -ésima neurona de la capa A .

Si consideramos un número de neuronas activas, es conveniente representarlas mediante un vector binario, para salvar los códigos de la imagen en forma compacta y calcular rápidamente la actividad de las neuronas de la capa de salida. Esto después de ejecutar proceso del código, cada imagen tiene su lista correspondiente de neuronas activas.

El proceso de entrenamiento se realiza de forma iterativa. Después de la representación de todas las imágenes a partir del subconjunto de entrenamiento, se

calcula el número total de errores de entrenamiento. Si este número es mayor que el 1% del total del número de imágenes, entonces el siguiente ciclo de entrenamiento continua. Si el número de errores es menor que el 1% entonces se detiene el proceso. El proceso también se detiene cuando el número de ciclos es mayor que en número de valores preestablecidos. En la Fig. 3.4 se muestra el diagrama de flujo para el proceso de entrenamiento.

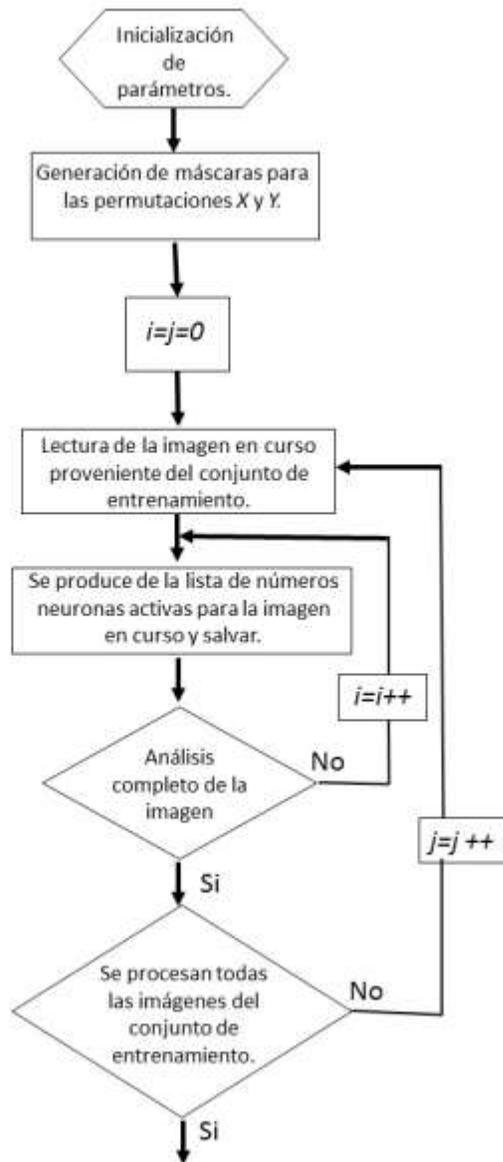


Fig. 3.4 Diagrama de flujo para el proceso de entrenamiento.

Referencias Capítulo 3

- [42] E. Kussul y T. Baidyk, "Face Recognition Using Special Neural Networks", *Neural Networks (IJCNN)*., vol. 1, nº 1, pp. 1-7, 2015.
- [43] E. Kussul, T. Baidyk, D. Wunsch, O. Makeyev y A. Martín, "Permutation Coding Technique for Image recognition systems", *IEEE Transactions on Neural Networks*, vol. 17, nº 6, p. 1566–1579, 2006.
- [44] D. Rachkovskij y E. Kussul, "Binding and Normalization of Binary Sparse Distributed Representations by Context-dependent Thinning", *Neural Computation*, vol. 1, nº 1, pp. 411-452, 2001.
- [45] F. Rosenblatt, "Principles of Neurodynamics", New York: Spartan Books, 1962.
- [46] E. Kussul y T. Baidyk, "Neural Networks and Micromechanics", New York: Springer-Verlag, 2010.

Capítulo 4: Experimentación y resultados obtenidos.

En este capítulo se presenta la metodología que se llevó a cabo para el desarrollo de la tarea. Desde la preparación de las imágenes a partir de la base de imágenes LFW y la definición de las áreas de trabajo para llevar a cabo la clasificación de rostros mediante redes neuronales utilizando el método de PCNC mediante el instrumento desarrollado en C++.

4.1 Preparación de las imágenes de trabajo.

Al emplear una base de datos a partir de entornos libres con características no controladas, se dificulta la clasificación de rostros. Debido a esta situación, para la fase de experimentación se consideran únicamente aquellos sujetos de prueba que cuenten con al menos 10 imágenes para garantizar la efectividad del método PCNC.

Para este fin se lleva a cabo una revisión exhaustiva de la base de imágenes LFW, que contiene 13,233 imágenes con una dimensión de 250X250 pixeles almacenadas en formato JPG, seleccionando aquellos sujetos que cuentan con al menos 10 imágenes para llevar a cabo el proceso de identificación y clasificación [47]. En la Fig. 4.1 se muestra un ejemplo de la selección de imágenes que se utilizaron.



Fig. 4.1 Ejemplo de imágenes seleccionadas para emplear en el experimento.

Las imágenes se convirtieron al formato BMP a 24 bits, debido a su calidad y facilidad de manejo sin importar el tipo de dispositivo que se emplea. Su dimensión al transformarlas a BMP es de 183 KB (183,054 bytes) por imagen. Una vez convertidas las imágenes se procedió a renombrarlas para uniformizar la base nombrando cada carpeta como sujeto con un número consecutivo para cada carpeta identificada y análogamente se renombraron las imágenes del mismo sujeto con un número consecutivo dentro de cada carpeta contenedora, tal que es posible identificar a cada imagen con única. En la Fig.4.2 se muestra la organización actual de las imágenes con las que se llevarán a cabo los experimentos de identificación.



Fig. 4.2 Organización actual de la base de imágenes a implementar en la experimentación.

Una vez organizada la nueva base de datos se consideraron 5 imágenes para llevar a cabo al proceso de entrenamiento y 5 imágenes para llevar a cabo el reconocimiento e identificación. Para el reconocimiento de rostros se seleccionan puntos de interés P_{ij} que tienen un brillo b_{ij} mayor que el umbral determinado B . De aquí que para llevar a cabo nuestra experimentación consideramos ventanas de dimensiones $h*w$ que se formaron alrededor de los puntos de interés que de los que se extraen múltiples características de la imagen en esta ventana, que se determinan a partir de los puntos positivos y negativos que se encuentran aleatoriamente distribuidos dentro de esta ventana [39]. De aquí que para probar la eficiencia del método PCNC se consideran tres casos de experimentación para la clasificación de los rostros objetivo en las imágenes: el análisis del rostro completo, el análisis del área de la barbilla y boca y el análisis de los ojos y cejas. Este análisis se realizó empleando un programa elaborado en C++, que se describe a continuación.

4.2. Descripción del instrumento elaborado en C++

El sistema operativo que se emplea para este instrumento es Windows XP, V.2002, utilizando un procesador Pentium D a 3GHz con una memoria de 232 GB. El programa de entrenamiento de la red neuronal considera 4 módulos de programa elaborados en C++. Mismos que se modificaron para llevar a cabo la tarea de entrenamiento para el reconocimiento y clasificación de los rostros objetivo. En la Fig. 4.3, se muestra el menú principal de nuestro instrumento así como los 4 diferentes módulos que se emplean para llevar a cabo la tarea de reconocimiento y clasificación de los rostros objetivo [49].

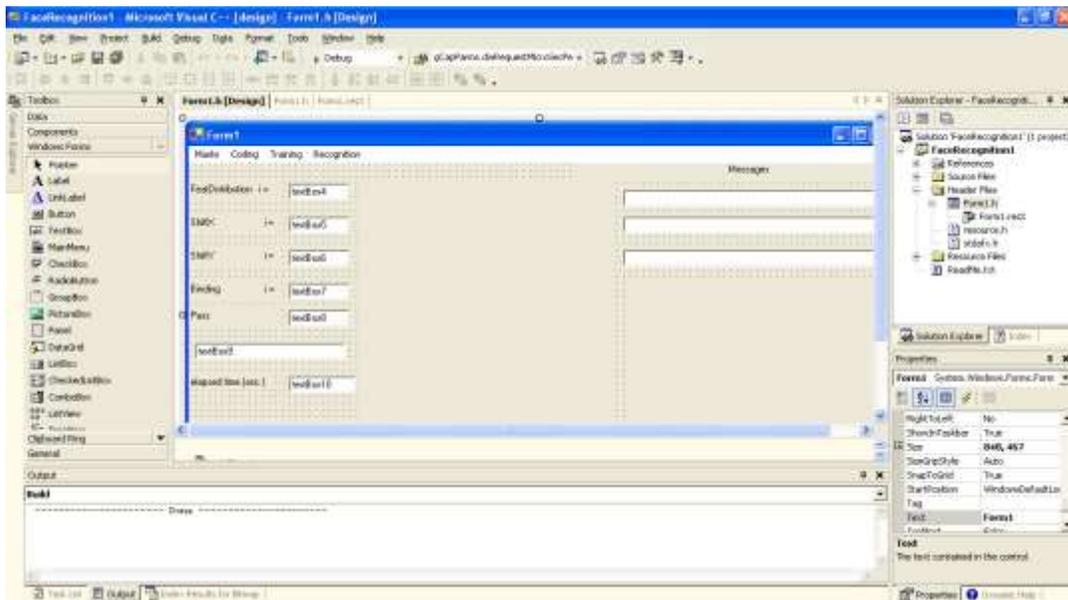


Fig. 4.3 Menú de desarrollo de forma para visual estudio C++

Los diferentes módulos corresponden a cada uno de los elementos que conforman el clasificador neuronal PCNC y permiten interactuar con cada una de las etapas del proceso de clasificación de forma secuencial.

4.2.1 Módulo 1: Enmascaramiento (Mask).

El proceso de enmascaramiento involucra a la capa de sensado, al extractor de características y al encoder. Como se mencionó en el capítulo 3, al presentar una imagen en escala de grises (capa de sensado o sensorial), el extractor de características (descriptor de características aleatorias locales (*Random Local Descriptor (RLD)*)) se inicia con la selección de los puntos de interés en la imagen. Para obtener un mayor número de características se considera el rectángulo o ventana de área $h*w$, que se forma alrededor de cada punto de interés.

Los puntos p positivo y n negativo que determinan cada característica se seleccionan de manera aleatoria dentro de esta ventana. Para obtener una distribución uniforme de estos puntos, en el procedimiento del programa se considera una selección aleatoria de las coordenadas (i, j) que definen la posición de cada uno de los puntos de interés dentro de la ventana.

Una vez extraídas las características de la imagen, el encoder transforma estas características en un vector binario que describe el estado de la capa asociativa y crea un vector binario auxiliar, en donde se definen el número de 1s que contiene cada vector. La posición de estos 1s en el vector auxiliar se selecciona de manera aleatoria para cada característica F_m . Este procedimiento generó la lista de posiciones que tiene cada 1 para cada característica, cada una de estas listas se guarda en memoria. A este vector auxiliar se le denomina máscara de la característica.

El resultado de este enmascaramiento se observa en la Fig. 4.4 Generación de la máscara para la imagen (*Mask generation*).

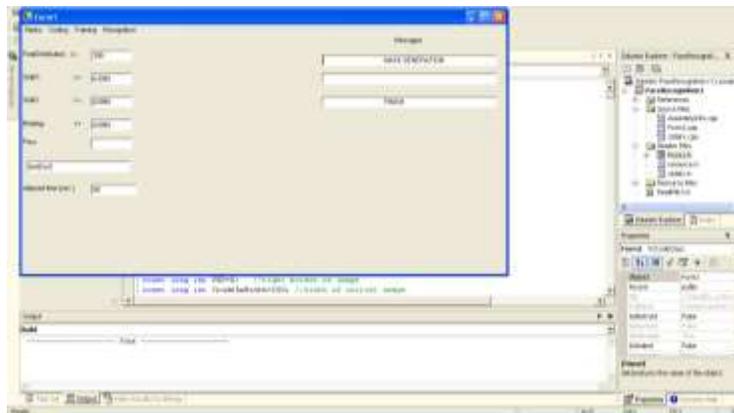


Fig. 4.4. Generación de la máscara para la imagen (*Mask generation*).

4.2.2 Módulo 2: Codificación.

En nuestro siguiente módulo se llevó a cabo el proceso transformación del vector auxiliar en un nuevo vector auxiliar* que corresponde a la localización de las características en la imagen, esta transformación se lleva a cabo mediante el proceso de permutación de los componentes del vector auxiliar original, como se explica en el capítulo 3 y se muestra en la Fig. 3.2. En la Fig. 4.5 se muestra el resultado de la codificación de un rostro completo.

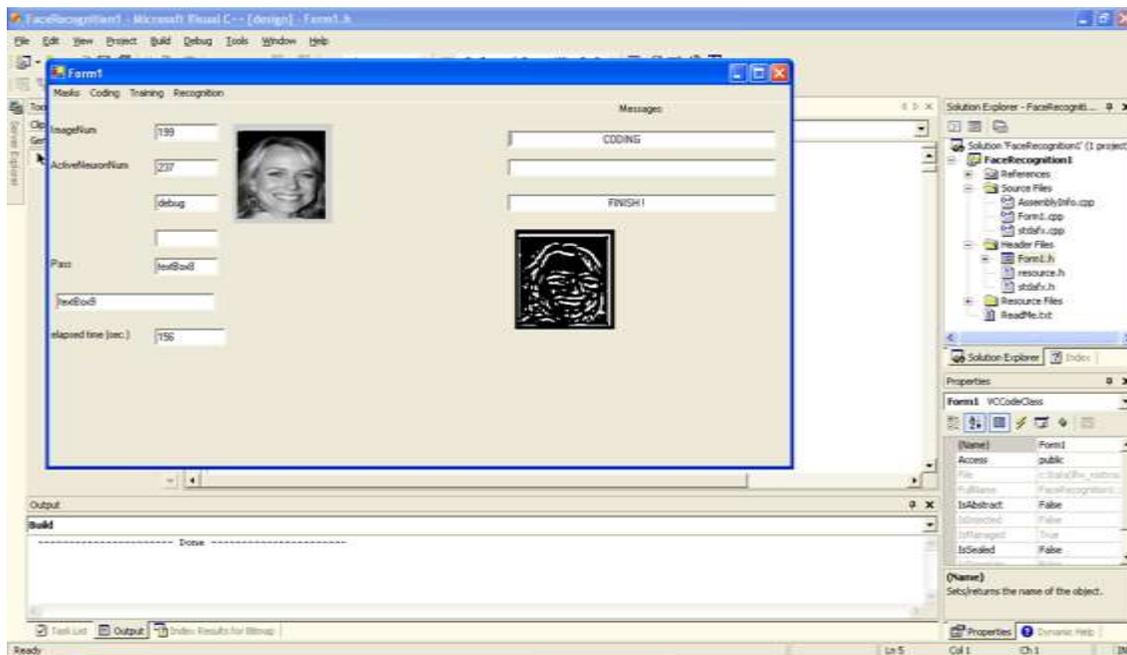


Fig. 4.5 Codificación del rostro completo.

4.2.3 Módulo 3: Entrenamiento.

Una vez codificada la imagen, se inicia la etapa de reconocimiento para el entrenamiento en donde se utilizan 100 imágenes, 5 imágenes de cada sujeto objetivo a clasificar, para llevar a cabo el proceso de entrenamiento de la red neuronal artificial. Se considera un total 20 sujetos en esta experimentación. Para establecer los pesos correspondientes a las conexiones entre las capas A y R se aplica la regla de *Hebb* o ley fundamental de entrenamiento para una red neuronal como sigue: “Si la neurona A excita a la neurona B y este proceso se repite, entonces el peso sináptico de la conexión

entre estas neuronas se incrementa” [50]. En nuestro caso estos pesos son susceptibles de modificarse durante todo el proceso de entrenamiento. La respuesta del sistema de entrenamiento corresponde al número de neuronas con máxima excitación en la capa *R*. En la Fig. 4.6 se muestra el menú de correspondiente a la etapa de entrenamiento.

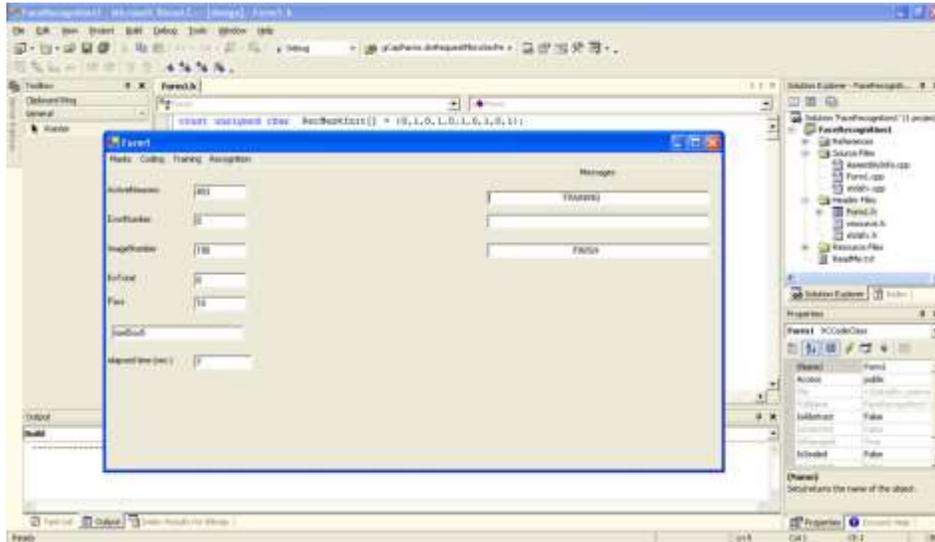


Fig. 4.6 Etapa de entrenamiento.

4.2.4 Módulo 4: Reconocimiento.

Para evaluar la calidad del reconocimiento, se calcula el número de errores obtenidos para las 5 imágenes por sujeto que no participaron en el proceso de entrenamiento. En la Fig. 4.7 Se muestra la etapa de reconocimiento de rostro completo, en donde puede apreciarse el número de errores obtenidos.

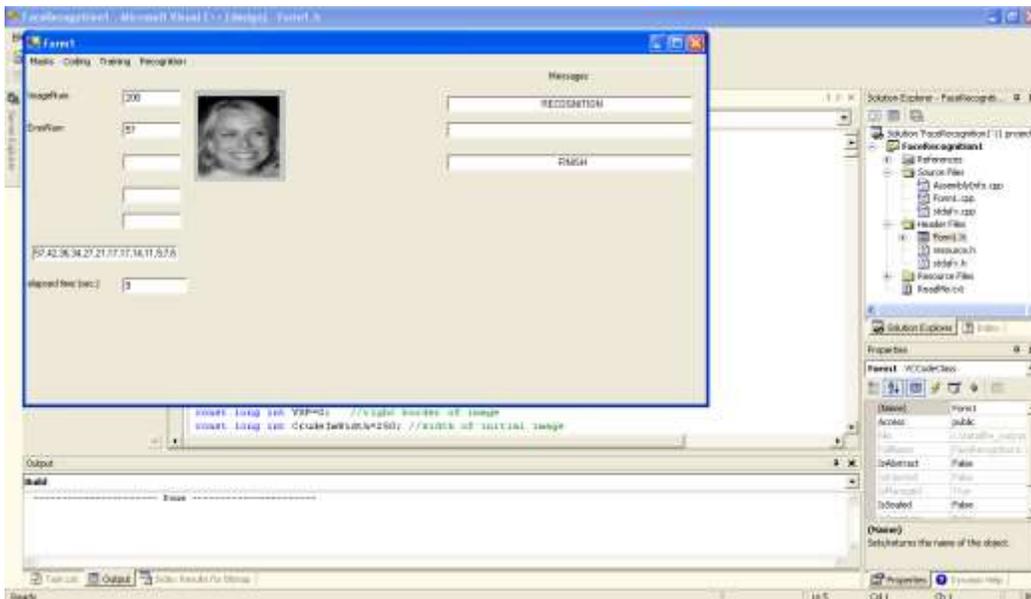


Fig. 4.7 Reconocimiento para el rostro completo.

Este instrumento, nos permite clasificar tanto el rostro completo, como fragmentos del mismo. Para este propósito es necesario modificar los parámetros en el programa y delimitar un área o fragmento en la imagen. En la Fig. 4.8 se muestra el resultado de codificar un fragmento de rostro (barbilla) y en la Fig. 4.9 se muestra el resultado de codificar un fragmento de rostro diferente (ojos y cejas).

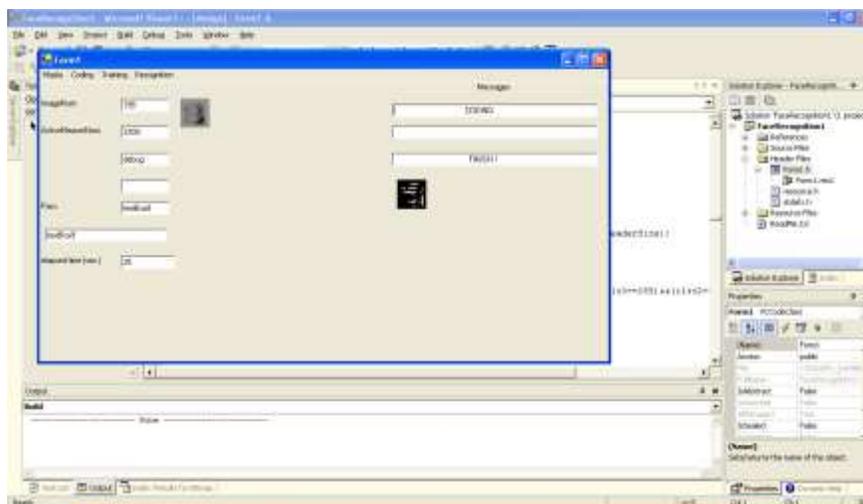
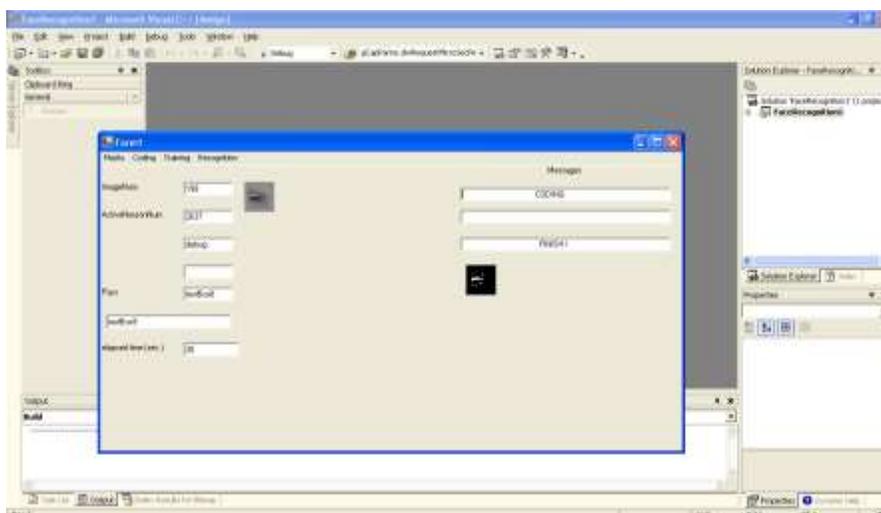


Fig. 4.8 Resultado de codificar la barbilla



4.9 Resultado de codificar ojo y ceja

4.3 Proceso de experimentación.

Utilizamos 20 sujetos objetivo de nuestra base de datos. Se tienen 10 imágenes de cada sujeto. En nuestro primer bloque de experimentos, se emplean las imágenes correspondientes a los números impares para entrenar al clasificador neuronal y los números pares para la clasificación del sujeto; 5 imágenes en cada caso. Posteriormente se consideraron nueve imágenes para el entrenamiento del clasificador y una imagen para reconocer. Se realizaron los siguientes experimentos:

4.3.1 Reconocimiento holístico.

En el capítulo 1, se mencionan diferentes métodos para el reconocimiento de rostros en imágenes fijas, con características establecidas y bien determinadas [10] para que los algoritmos de reconocimiento automático pudieran llevarse a cabo. Sin embargo para nuestro caso particular, debido a la complejidad de la base de imágenes LFW, se realizó una selección manual del área que contuviera el rostro objetivo a clasificar en la imagen con dimensiones $h*w$. Esta selección se muestra en la Fig.4.10



Fig. 4.10 Selección manual del área del rostro a clasificar.

Posteriormente esta selección se transformó en imágenes de 100X100 píxeles en escala de grises como se muestra en la Fig.4.11, para presentarla al encoder que la transformaría en el vector binario para cada característica extraída.



Fig. 4.11 Áreas de rostro seleccionadas y transformadas en escala de grises.

Para obtener resultados confiables, se realizaron cinco experimentos por caso, realizando el proceso de enmascaramiento, codificación entrenamiento y reconocimiento. La generación de las máscaras para el método de clasificación utilizando el método PCNC implica una selección de los puntos de interés en el que se realizaron 100 ciclos de entrenamiento. Para este proceso se consideraron los tiempos de enmascaramiento, el tiempo de codificación, el tiempo de entrenamiento, el tiempo de reconocimiento, el número de errores y el porcentaje de reconocimiento. En la Fig.4.7 se muestra el reconocimiento del rostro completo.

En la Tabla 4.1 se muestran los resultados para nuestros experimentos en el reconocimiento holístico considerando 5 imágenes para el entrenamiento del clasificador y 5 imágenes para llevar a cabo el reconocimiento.

Tabla 4.1. Resultados de experimentos en el caso de reconocimiento holístico utilizando 5 imágenes para entrenar y 5 para reconocer.

Experimento	Tiempo de Enmascaramiento en segs.	Tiempo de codificación en segs.	Tiempo de entrenamiento en segs.	Tiempo de reconocimiento en segs.	No. de errores en el reconocimiento	Porcentaje de reconocimiento
1	79	146	9	9	62	38
2	80	153	7	9	57	43
3	80	156	7	10	61	39
4	79	160	7	10	67	33
5	80	152	7	9	63	37
Promedio	79.6	153.4	7.4	9.4	62	38

4.3.2 Reconocimiento de fragmentos de rostros.

Para el reconocimiento fragmentado del rostro, se consideraron dos casos:

4.3.2.1 Fragmentos de ojos y cejas (Caso I).

Para este experimento se seleccionó manualmente un fragmento de área que contuviera un ojo y una ceja, como se muestra en la Fig. 4.12.



Fig. 4.12 Fragmento de área que contiene ojo y ceja.

Posteriormente se empleó la misma metodología que para el caso holístico, considerando 5 imágenes para el entrenamiento del clasificador y 5 imágenes para reconocer en el caso I. En la Fig. 4.13 se muestra el área empleada para la experimentación.

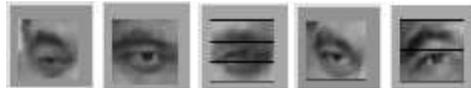


Fig. 4.13 Área seleccionada en escala de grises.

Los resultados para este caso I se muestran en la Tabla 4.2.

Tabla 4.2. Resultados de experimentos preliminares en el fragmentado. Caso I

Experimento	Tiempo de Enmascaramiento en segs.	Tiempo de codificación en segs.	Tiempo de entrenamiento en segs.	Tiempo de reconocimiento en segs.	No. de errores en el reconocimiento	Porcentaje de reconocimiento
1	79	26	139	2	58	42
2	79	22	137	1	56	44
3	79	23	144	2	64	36
4	79	23	143	2	65	35
5	79	21	130	1	65	35
Promedio	79	23	138.6	1.6	61.6	38.4

4.3.2.2 Fragmentos de boca y barbilla (Caso II).

Para este experimento se seleccionó manualmente un fragmento de área que contuviera la boca y barbilla del sujeto, como se muestra en la Fig. 4.14.



Fig. 4.14 Fragmento de área que contiene boca y barbilla.

Posteriormente se empleó la misma metodología que para el fragmentado del ojo y la ceja. En la Fig. 4.16 se muestra el área empleada para la experimentación.



Fig. 4.16 Área seleccionada en escala de grises en el Caso II.

Los resultados para este caso II se muestran en la Tabla 4.3 con 5 imágenes para entrenar y 5 para reconocer.

Tabla 4.3. Resultados de experimentos preliminares en el fragmentado. Caso II

Experimento	Tiempo de Enmascaramiento en segs.	Tiempo de codificación en segs.	Tiempo de entrenamiento en segs.	Tiempo de reconocimiento en segs.	No. de errores en el reconocimiento	Porcentaje de reconocimiento
1	81	24	131	2	73	27
2	80	25	146	2	74	26
3	80	28	148	2	76	24
4	81	24	136	1	74	26
5	80	26	148	2	76	24
Promedio	80.4	25.4	141.8	1.8	74.6	25.4

A partir de los resultados obtenidos, se consideró realizar experimentos aumentando el número de imágenes para entrenar a nuestro clasificador y obtener un mayor porcentaje de reconocimiento. Para este efecto, se consideró el caso holístico con 9 imágenes para llevar a cabo el entrenamiento y 1 para reconocer.

En la Tabla 4.4 se muestran los resultados para nuestros experimentos en el reconocimiento holístico considerando 9 imágenes para el entrenamiento del clasificador 1 imagen para reconocer.

Tabla 4.4. Resultados de experimentos en el caso de reconocimiento holístico (b) utilizando 9 imágenes para entrenar y 1 para reconocer.

Experimento	Tiempo de Enmascaramiento en segs.	Tiempo de codificación en segs.	Tiempo de entrenamiento en segs.	Tiempo de reconocimiento en segs.	No. de errores en el reconocimiento	Porcentaje de reconocimiento
1	80	288	17	2	12	40
2	79	274	17	2	11	45
3	79	291	14	2	12	40
4	79	267	20	2	12	40
5	79	294	12	3	11	45
Promedio	79.2	282.8	16	2.2	11.6	42

4.4. Resultados.

En la experimentación holística, encontramos que al emplear el 50% de las imágenes para entrenar al clasificador PCNC, se obtuvo un porcentaje de reconocimiento para la clasificación del rostro objetivo del 38 %. Para el caso holístico con el 90% de imágenes para entrenar al clasificador, se obtuvo un porcentaje de reconocimiento del 42% que representa un incremento en la tasa de reconocimiento del 4%.

Respecto a la experimentación empleando fragmentos específicos de la imagen del rostro, encontramos que para el caso I (ojo-ceja) el porcentaje de reconocimiento es del 38.4% y para el caso II (boca-barbilla) el porcentaje de reconocimiento es del 25.4%.

Considerando los resultados obtenidos para el caso holístico y el caso I correspondiente al empleo del fragmento específico de ojo-ceja, podemos decir que se obtiene un porcentaje de reconocimiento equivalente.

En la literatura se reporta que se han empleado algunos otros métodos para el reconocimiento de rostros aplicados a la base de imágenes LFW tales como el descriptor de patrones binarios locales (*Local Binary Patterns (LBP)*), el descriptor discriminante de rostros (*Discriminant Face Descriptor (DFD)*), POEM, LQP por mencionar algunos. Sin embargo, estos métodos permiten obtener una representación aceptable para un rostro con un porcentaje de entre el 69% y el 80%, sin embargo no se emplean para llevar a cabo una clasificación de los rostros objetivo como es el caso del método PCNC [51].

Referencias. Capítulo 4.

- [47] G. B. Huang, M. Ramesh, T. Berg y E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying", University of Massachusetts, Amherest, 2007.
- [48] E. Kussul, T. Baidyk, C. Conde y I. Cabello, "Face recognition improvement with distortions of image in training set", de *Proceedings of the 2013 International Joint Conference on Neural networks.*, Dallas, Texas, USA., 2013.
- [49] J. Templeman y A. Olsen, Visual C++.NET, Madrid: McGraw-Hill, 2002.
- [50] D. O. Hebb, "The Organization of Behaviour", New York: Wiley, 1949, p. 319.
- [51] K. Daniilidis, P. Maragos, N. Paragios, "Computer Vision -- ECCV 2010: 11th European Conference on Computer Vision", *Proceedings I, Lecture Notes in Computer Science Image Processing, Computer Vision, Pattern Recognition, and Graphics* Heraklion, Crete, Greece, 2010.

Capítulo 5: Conclusiones y propuestas de trabajos a futuro.

En el presente capítulo se presentan las conclusiones las que se llegó a partir de los resultados obtenidos. Así como las aportaciones de esta investigación y algunas propuestas de trabajos a futuro para esta línea de investigación.

A partir de los resultados obtenidos, podemos concluir que es posible llevar a cabo la clasificación de rostros humanos implementando el método de redes neuronales en ambientes no controlados, en nuestro caso particular aplicando el clasificador neuronal PCNC y la base de imágenes LFW. Lo que representa ventajas importantes para la clasificación de rostros humanos en tareas de seguridad, administración, industria, autenticación, etiquetado de imágenes en redes sociales, interacción robótica y biometría como se mencionó anteriormente. Toda vez que no es necesario recurrir a bases de imágenes creadas en ambientes controlados para obtener información sobre los rostros objetivo, sino tomar las imágenes directamente del medio en el que se encuentren los sujetos sin modificaciones. Lo que permitirá reducir tiempos de operación y costos en la infraestructura para llevar a cabo la clasificación de rostros.

Para el caso del entrenamiento con el 50% de las imágenes en el caso holístico, se encontró que el porcentaje de reconocimiento es del 38% y para los casos de fragmentos de rostros tenemos para el caso I (ojos-ceja) el 38.4% por lo que podemos decir que no encontramos diferencias significativas entre la experimentación holística y la experimentación utilizando el fragmento de ojo-ceja. Esto implica una aportación importante en la implementación de nuestro instrumento de clasificación, toda vez que es posible disminuir los tiempos de entrenamiento y procesamiento para el sistema, así como un ahorro en el almacenamiento de información. Que será beneficioso con bases de imágenes más amplias.

Para el caso II, encontramos un porcentaje de reconocimiento del 25% que es aún menor que los porcentajes reportados para otras condiciones experimentales [51], que es congruente con los resultados esperados para este trabajo, toda vez que las características de la base de imágenes LFW, presentaron un reto en sí mismas. Lo que abre una nueva línea de investigación para la clasificación de rostros en ambientes no

controlados, toda vez que los trabajos de investigación para la clasificación de rostros, se orientan básicamente a bases de imágenes obtenidas en ambientes controlados; que es una de las aportaciones más importantes de esta investigación.

Por lo que respecta a la experimentación empleando el 90% de imágenes para llevar a cabo el entrenamiento del clasificador, encontramos un porcentaje de reconocimiento para la experimentación holística del 42%, Lo que resulta congruente con el entrenamiento del clasificador neuronal PCNC, debido a que al proporcionar al extractor de características un mayor número de puntos de interés, logramos discernir los rostros objetivo con una mayor exactitud, de acuerdo a ley fundamental de entrenamiento para una red neuronal (Ley de Hebb).

Como se menciona en la literatura [52], [53] diferentes fracciones de las imágenes de rostros, aporta diferentes contribuciones para el reconocimiento, independientemente del modelo y la base de datos que se elija para llevar a cabo el reconocimiento y la clasificación de los rostros objetivo.

A partir de esta investigación, se propone continuar ampliando la base de imágenes con la que se cuenta actualmente, así como explorar la posibilidad de implementar métodos automáticos para la fragmentación de rostros. También se considera el realizar una nueva serie de experimentos, para determinar la tasa de reconocimiento con diferentes imágenes a reconocer para un mismo sujeto; seleccionando diferentes fragmentos específicos y con ello establecer los parámetros mínimos con los que deben cumplir las imágenes para ser clasificadas. Así como implementar mejoras a nuestro instrumento de clasificación para optimizar su desempeño y obtener una mayor eficiencia en la clasificación de imágenes en entornos libres, no sólo para la clasificación de rostros humanos sino para tejidos específicos y seres vivos.

Referencias. Capítulo 5

- [50] D. O. Hebb, "The Organization of Behaviour", New York: Wiley, 1949, p. 319.
- [52] S. Ulman y E. y. V. M. Sali, "A Fragment- Based Approach to Object representation and Classification", *C. Arcelli et Al. Eds. IWVF4,LNCS 2059.*, pp. 85-100, 2001.
- [53] J. Tanaka y M. Farah, "Parts and Wholes in Face Recognition", *The Quarterly Journal of Experimental Psychology*, vol. 46A, nº 2, pp. 225-245, 1993.

Apéndice.

Apéndice: Código del programa empleado para la aplicación de redes neuronales en el reconocimiento de rostros en entornos libres.

```
//FaceRecLFW
*/Definición de parámetros de trabajo y encabezados del programa/*
#pragma once #include <time.h>
#include <stdio.h>
#include <string.h>
#include <atime.h>
//Nomenclatura para las imágenes empleadas en el entrenamiento y reconocimiento//
#define ADDRESS1 "imagenes/sujeto%d/sujeto%d__%d%d%d%d.bmp"
#define ADDRESS2 "imagenes/sujeto%d/sujeto%d__%d%d%d%d.bmp"
#define ADDRESS3 "imagenes/sujeto%d/d/sujeto%d%d__%d%d%d%d.bmp"
const unsigned char TrainMaskInit[] = {0,1,0,1,0,1,0,1,0,1}; //50% de imágenes para el entrenamiento
const unsigned char RecMaskInit[] = {1,0,1,0,1,0,1,0,1,0}; //50% de imágenes para el reconocimiento
//Definición de áreas de trabajo//
const signed char DX=5; //ancho de ventana
const signed char DY=5; //altura de ventana al rededor del punto de //interés
const long int ImWidth=30; // clasificación holística
const long int ImHeight=30;
const unsigned delta=3;//parámetro de desplazamiento
const long int HeaderSize=54; //tamaño encabezado del archive de la imagen
const long int ImageWidth=ImWidth+DX+2+2*delta; //Corrimiento para el punto de interés (regla de Hebb)
const long int ImageHeight=ImHeight+DY+2+2*delta; //Corrimiento para el punto de interés (regla de Hebb)
const long int ClassSize=20; //número de clases de salida
const long int SampleNumber=10; //número de imágenes elegidas para cada persona
long int
const long int TrainPart=4;
const long int HSP=0; //borde izquierdo de la imagen
const long int VSP=0; //borde derecho de la imagen
const long int CrudeImWidth=250; //ancho inicial de la imagen
const long int CrudeImHeight=250; //altura inicial de la imagen
const long int UpperLimit=250; // número máximo de líneas en la imagen intermedia (menos de 400)
unsigned char TR; //TR=1 para entrenamiento
unsigned char RC; //RC=1 para reconocimiento
const long int NN=ClassSize*SampleNumber; //conjunto de entrenamiento
const long int PassN=100; //no. de ciclos de entrenamiento
const long int NetSize=64000;// número de neuronas involucradas
const long int ActivSize=NetSize/4;
const long unsigned PositivPoint=2;
const long unsigned NegativPoint=2;
const unsigned RecDist=1;
const float TDS=0.15;
*/(intervalo de seguridad) Diferencia entre la excitación correcta que el //competidor (mayor que el umbral)
De aquí depende la calidad del reconocimiento (aseguramos que //el ganador tiene diferencia mucho mayor
que el siguiente competidor)/*
const int BCN=5; //no. de ciclos
long int meanBr = 128;
const unsigned ShiftRange=12; //rango del ciclo de corrimiento Dc
const unsigned ShiftNumX=(ImageWidth/ShiftRange)+2;
const unsigned ShiftNumY=(ImageHeight/ShiftRange)+2;
const long int FeatNumber=400; //número de características
const long int FDN=16; //número de unos (puntos de interés) para cada característica en el vector de
//enmascaramiento
if(fwrite(TrainRecMask,1,NN,TrainRecMaskFile)!=NN); //almacenamiento de la información para el entrenamiento
//a partir del enmascaramiento

{
```

```

textBox2->Text = "Cannot write TrainRecMaskFile";
    textBox2->Refresh();
    goto end1;}
if (fwrite(FeatMaskX,1,N1,ftMaskX)!=N1)
    {
        textBox2->Text = "Cannot write FeatMaskX";
        textBox2->Refresh();
        goto end1;}
if (fwrite(FeatMaskY,1,N1,ftMaskY)!=N1)
    {
        textBox2->Text = "Cannot write FeatMaskY";
        textBox2->Refresh();
        goto end1;}
if(fwrite(FeatMaskBr,1,N1,ftMaskBr)!=N1)
    {
        textBox2->Text = "Cannot write FeatMaskBr";
        textBox2->Refresh();
        goto end1;}
if(fwrite(ShiftX,sizeof(long),NetSize,shiftX)!=NetSize)
    {
        textBox2->Text = "Cannot write shiftX";
        textBox2->Refresh();
        goto end1;}
if(fwrite(ShiftY,sizeof(long),NetSize,shiftY)!=NetSize)
    {
        textBox2->Text = "Cannot write shiftY";
        textBox2->Refresh();
        goto end1;}
if(fwrite(Binding1,sizeof(long),NetSize,shiftBind)!=NetSize)
    {
        textBox2->Text = "Cannot write shiftBind";
        textBox2->Refresh();
        goto end1;
    }
end1:
    textBox3->Text = "FINISH";
textBox3->Refresh();
fclose(ftMaskX);
fclose(ftMaskY);
fclose(ftMaskBr);
fclose(ftDistribution);
fclose(shiftX);
fclose(shiftY);
fclose(shiftBind);
fclose(TrainRecMaskFile);
private: System::Void label4_Click(System::Object * sender, System::EventArgs * e)
{
private: System::Void Coding_Click(System::Object * sender, System::EventArgs * e)
{
    long unsigned i,j;
    long unsigned q,p;
    long unsigned buff[FDN];
}
textBox1->Text = " CODING"; // ventana de interacción con el usuario
    textBox1->Refresh();
    textBox2->Text = "";    textBox2->Refresh();
    textBox3->Text = "";    textBox3->Refresh();
    textBox4->Text = "";    textBox4->Refresh();
    textBox5->Text = "";    textBox5->Refresh();
    textBox6->Text = "debug";    textBox6->Refresh();
    textBox7->Text = "";    textBox7->Refresh();
        label2->Text = "ImageNum";
        label3->Text = "ActiveNeuronNum";
        label4->Text = "";
        label5->Text = "";

```

```

        label2->Refresh();
        label3->Refresh();
        label4->Refresh();
        label5->Refresh();
fp=fopen("ftmskY.dat","rb");
if(fp==NULL)

if(fread(FeatMaskY,1,FeatSize,fp)!=FeatSize)
{
    textBox2->Text = "Cannot read FeatMaskY";
    textBox2->Refresh();
    goto end;
}

void LetterImageRead(long int letterNum1); //lectura del rostro completo
{
    long int i,j,k,block,p,v,sum1,br1,br2;
    int color;
    unsigned char kk[1];
    char buff2[256];
    int i1,i2,i3,j1,j2,x,X1,X2,Y1,Y2;
    int Xleft[2],Yupper[2];
    long int sumX[CrudelmWidth];
    long int sumY[CrudelmHeight];
    int MarkedWidth, MarkedHeight;
    float   scaleX,scaleY;
    int n,m,pntr;

    for(i=0;i<CrudelmWidth;i++)
    sumX[i]=0;
    for(i=0;i<CrudelmHeight;i++)
    sumY[i]=0;
    Bitmap *CrudelImage = new Bitmap(CrudelmWidth,CrudelmHeight);
    const int ImSize1 = CrudelmWidth*CrudelmHeight*3+5000; //400*400*3+5000;
    const int ImSize2 = CrudelmWidth*CrudelmHeight*3+HeaderSize; //400*400*3+HeaderSize;
    unsigned char ImArray[CrudelmWidth*CrudelmHeight*3+5000];
    static int ImArray1[CrudelmWidth][CrudelmHeight]; #[400][400];
    Color clr;
    p=letterNum1/SampleNumber+1; //número de imagen de rostro
    v=letterNum1-(p-1)*SampleNumber+1;
    i1 =p/100; //definición de número de imágenes que se pueden trabajar como máximo 999
        x = p-i1*100;
        i2 = x/10;
        i3 = x-i2*10;
        j1 = v/10; //definición de número de imágenes por persona hasta 99 imágenes
        j2 = v-j1*10;
    if(i1>0)
    {
    sprintf(buff2,ADDRESS3,i1,i2,i3,i1,i2,i3,4,0,j1,j2); goto m1;}
// nombre de archivo de imagen con código 4000 para barbilla
//0000 imagen completa
        if(i2>0)
        {sprintf(buff2,ADDRESS2,i2,i3,i2,i3,4,0,j1,j2); goto m1;}
        if(i3>0)
        {
        sprintf(buff2,ADDRESS1,i3,i3,4,0,j1,j2); goto m1;
        }
    }
    for(i=0;i<CrudelmWidth;i++) //cambio de tamaño para barbilla 10 y rostro 30
    if(sumX[i]>10) //30
    {
    Xleft[j]=i;i=i+5;j=j+1;
    }
    j=0;

```

```

for(i=0;i<CrudelmHeight;i++)
if(sumY[j]>10) //30
{
Yupper[j]=i; i=i+5;j=j+1;
}
X1=Xleft[0];//búsqueda de marcado manual
X2=Xleft[1];
Y1=Yupper[0];
Y2=Yupper[1];
MarkedWidth = X2-X1;
MarkedHeight = Y2-Y1;
scaleX = float(ImWidth)/float(MarkedWidth);
scaleY = float(ImHeight)/float(MarkedHeight);
for(i=0;i<ImWidth*ImHeight;i++);
GrayIm[i]=0;
for(n=Y1;n<Y2;n++)
for(m=X1;m<X2;m++)
{
x=ImArray1[n][m];
i = int(float(n-Y1)*scaleY);
j = int(float(m-X1)*scaleX);
GrayIm[i*ImWidth+j] = ImArray1[n][m];
}
block=ImageHeight*ImageWidth;
for(j=0;j<block;j++)
{
GrayImage[j]=BB; //(ampliación de la imagen para asegurar el mismo tamaño de la imagen en el formato bmp
GrayImage1[j]=BB;
}
for(i=0;i<ImHeight;i++)
for(j=0;j<ImWidth;j++)
{
x=GrayIm[i*ImWidth+j];
GrayImage[(i+DY/2+delta)*ImageWidth+j+DX/2+delta]=GrayIm[i*ImWidth+j]; //PCNC
GrayImage1[(i+DY/2+delta)*ImageWidth+j+DX/2+delta]=GrayIm[i*ImWidth+j];
}
sum1=0;
for(i=0;i<ImageHeight;i++)
for(j=0;j<ImageWidth;j++)
sum1=sum1+GrayImage1[i*ImageWidth+j];
if(sum1>ImageWidth*ImageHeight)
{
sum1 = sum1/(ImageWidth*ImageHeight);
for(i=0;i<ImageHeight;i++)
for(j=0;j<ImageWidth;j++)
{
br2 = GrayImage1[i*ImageWidth+j];
br1 = (br2*meanBr)/(sum1+1);
if(br1>255)
br1=255;
GrayImage1[i*ImageWidth+j] = br1;
}
}
end:

// (Binding) procedimiento de normalización para reducir método de 1s en el vector
if(BCN==0) //binding cycle coding (vincular elementos comunes en los vectores binarios y disminuye la
cantidad de //neuronas activas). Pero sin perder la información global.
for(i=0;i<NetSize;i++)
bt1[i]=bt[i];
else
for(i=0;i<NetSize;i++)
bt1[i]=0;
for(i=0;i<NetSize;i++)

```

```

if(bt[i]==1)
{
Pos=i;
for(j=0;j<BCN;j++)
{
Pos=Binding1[Pos];
if(bt[Pos]==1)
goto end3;
}
bt1[i]=1;
end3:
Pos=0;
}
// Fin de Binding
for(i=0;i<NetSize;i++)
if(sum1<ActivSize-2)
if(bt1[i]==1)
{
ActivNeuronNumber[sum1++]=i;
}
ActivNeuronNumber[0]=sum1;//se consideran número de neuronas. Se consideran 3 a 5% de neuronas a
//implementar de las que se encuentran activas.
return(sum1);
}
void Distortion(void) // Solo distorsiones de desplazamiento para aumentar el número de muestras debido a las
dificultades de las imágenes.
{
unsigned long u,v;
long unsigned i,j,k2;
unsigned inclin;
unsigned char* work1=new unsigned char [ImageWidth*ImageHeight];

switch(dist)
{
case 0: // imagen original si desplazamiento
{
for(i=0;i<ImageHeight;i++)
for(j=0;j<ImageWidth;j++)
GrayImage[i*ImageWidth+j]=GrayImage1[i*ImageWidth+j];
break;
}
}

*//Rutina de entrenamiento. Inicialización de variables y declaración de la rutina que se repite hasta el caso
26//*
private: System::Void TrainingBegin_Click(System::Object * sender, System::EventArgs * e)
{
long int ii;

textBox1->Text = "TRAINING";
textBox1->Refresh();
textBox3->Text = "";
textBox3->Refresh();
textBox7->Text = "";
textBox7->Refresh();
textBox8->Text = "";
textBox8->Refresh();

label2->Text = "ActiveNeurons";
label2->Refresh();
label3->Text = "ErrorNumber";
label3->Refresh();
label4->Text = "ImageNumber";
label4->Refresh();
label5->Text = "ErrTotal";
label5->Refresh();

```

```

label6->Text = "Pass";
label6->Refresh();
t=CTime::GetCurrentTime();
StartTime = 3600*t.GetHour() + 60*t.GetMinute() + t.GetSecond();
for(ii=0;ii<MatSize;ii++)
    mtr[ii]=0; /* clear */
train();// SI NO HAY ERRORES, no se realiza entrenamiento. Si hay errores se realiza el entrenamiento
//cambiando pesos de conexiones en la matriz (archivo) matrix.dat
}
void train(void) // Entrenamiento (algoritmo de cambio de pesos algoritmo de Hebb
{
FILE* servis;
long int jj,i;
unsigned long k[1];
unsigned char kk[1];
unsigned short kkk[1];
long int sm,p,v;
fp=fopen("TrainRecMask.dat","rb");
if(fp==NULL)
{
textBox2->Text = "Cannot open file TrainRecMask.dat";
textBox2->Refresh();
}
}

// Procedimiento de Reconocimiento//*
private: System::Void RECOGNITION_Click(System::Object * sender, System::EventArgs * e)
{
    unsigned short kkk[1];
    long unsigned i,j;
    long unsigned p,q;
    long unsigned buff[FDN];
    unsigned char errors1[ClassSize];
    textBox1->Text = "RECOGNITION";
    textBox1->Refresh();
    textBox3->Text = "";
    textBox3->Refresh();
    textBox4->Text = "";
    textBox4->Refresh();
    textBox5->Text = "";
    textBox5->Refresh();
    textBox6->Text = "";
    textBox6->Refresh();
    textBox7->Text = "";
    textBox7->Refresh();
    textBox8->Text = "";
    textBox8->Refresh();
    textBox9->Text = "";
    textBox9->Refresh();
    label2->Text = "ImageNum";
    label2->Refresh();
    label3->Text = "ErrorNum";
    label3->Refresh();
    label4->Text = "";
    label4->Refresh();
    label5->Text = "";
    label5->Refresh();
    label6->Text = "";
    label6->Refresh();
    matr=fopen("matrix.dat","rb");
    for(int jj=0; jj<MatSize; jj++)
    {
        if(fread(kkk,sizeof(unsigned short),1,matr)==NULL)
        {
            textBox2->Text = "Cannot open matrix.dat";

```

```

textBox2->Refresh();
exit(1);}
mtr[jj]=kkk[0];
}
fclose(matr);
fp=fopen("TrainRecMask.dat","rb");
if(fp==NULL)
{
textBox2->Text = "Cannot open file TrainRecMask.dat";
textBox2->Refresh();
goto end;
}
for(j=0;j<ShiftNumX;j++); //almacenamiento de resultados en protocol.dat
for(i=0;i<NetSize;i++)
MultyShiftX[j][i]=0;
for(j=0;j<ShiftNumY;j++)
for(i=0;i<NetSize;i++)
MultyShiftY[j][i]=0;
for(i=0;i<NetSize;i++)
{
MultyShiftX[0][i]=ShiftX[i];
MultyShiftY[0][i]=ShiftY[i];
}
for(j=0;j<ShiftNumX-1;j++)
for(i=0;i<NetSize;i++)
{
q=MultyShiftX[j][i];
p=MultyShiftX[j][q];
MultyShiftX[j+1][i]=p;
}
for(j=0;j<ShiftNumY-1;j++)
for(i=0;i<NetSize;i++)
{
q=MultyShiftY[j][i];
p=MultyShiftY[j][q];
MultyShiftY[j+1][i]=p;
}
}

```

i

ⁱ Los módulos originales que se emplearon como base para la implementación de este instrumento, fueron proporcionados por el Dr. Ernst Kussul Mikhailovich a quien agradezco por su valiosa aportación y guía para el desarrollo de esta investigación..