



Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Eléctrica

**Desarrollo de un método de clasificación de edad para
imágenes faciales basado en análisis antropométrico y de
texturas**

MEMORIA PARA OPTAR AL TITULO DE INGENIERO CIVIL ELECTRICISTA

JACOB ALEJANDRO SARAVIA ESTRADA

PROFESOR GUÍA:
CLAUDIO PÉREZ FLORES

MIEMBROS DE LA COMISIÓN:
HÉCTOR AGUSTO ALEGRÍA
PABLO ESTÉVEZ A.

SANTIAGO DE CHILE
OCTUBRE 2011

AGRADECIMIENTOS

En primer lugar debo agradecer a mis padres, por su incondicional y constante apoyo, quienes a pesar de la distancia se mantuvieron siempre presentes alentándome desde pequeño a entregar lo mejor de mí y superarme día a día.

A mis hermanos, por compartir alegrías y experiencias que forjaron en gran medida la persona que soy, y también por brindarme la ayuda necesaria en los momentos precisos.

A mis compañeros de laboratorio, quienes ayudaron en gran medida a concretar el presente trabajo, compartiendo de paso su entrañable amistad.

A mi profesor guía por su orientación y colaboración durante el desarrollo de la investigación.

El trabajo realizado para esta tesis fue financiado gracias al proyecto FONDEF D08I-1060.

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA
POR: JACOB ALEJANDRO SARAVIA ESTRADA
FECHA: 29/12/2011
PROF. GUÍA: CLAUDIO PÉREZ FLORES

Desarrollo de un método de clasificación de edad para imágenes faciales basado en análisis antropométrico y de texturas

Un sistema que ayude a determinar la edad de las personas se transforma en una gran ayuda para facilitar una serie de tareas, como estudios de mercadeo, selección de contenidos aptos para ciertos grupos etarios, entre otros sistemas basados en interacciones humanas.

El presente trabajo expone el desarrollo de un algoritmo que sea capaz de estimar edad y clasificar en rangos etarios utilizando rostros frontales, realizando para ello un análisis antropométrico de proporciones faciales en conjunto con un análisis de textura de la piel en determinadas zonas claves del rostro. Para ello es que se calculan ciertas proporciones del rostro y, por otro lado, se determina el nivel de arrugas, mediante la obtención de la intensidad, cantidad y distribución espacial de dichos surcos en la piel. Con tales datos se efectúa un procedimiento de clasificación, contrastando un par de algoritmos de aprendizaje computacional conocidos como redes neuronales artificiales y máquinas de soporte vectorial. Para ajustar los parámetros del algoritmo anteriormente descrito, se aplica una optimización por enjambre de partículas.

Con todo esto en mente, el clasificador con mejor desempeño es el que en función de la cantidad de grupos previamente definidos (e.g. niño, joven o adulto) entregue la mayor tasa de aciertos, o el que mediante una aproximación de la edad real (en cantidad de años) presente un menor error. Para dichas tareas se obtuvo una clasificación con una tasa de aciertos de un 98% en promedio para 2 clases, y 6 años de error en promedio para la estimación de la edad real del sujeto.

Contenido

1. Introducción	6
1.1. Antecedentes generales	6
1.2. Antecedentes específicos	7
1.3. Trabajos previos.....	7
1.3.1. Transformada wavelets lineal	8
1.3.2. Mapas binarios de campos receptivos gaussianos	9
1.3.3. Modelo de ranking	9
1.3.4. Clasificación mediante características gabor y lógica difusa	10
1.3.5. Clasificación automática de edad usando LBP	10
1.4. Fundamentación y objetivos.....	11
1.5. Objetivos específicos	12
1.6. Herramientas	12
2. Clasificación de edad utilizando características faciales.....	13
2.1. Preprocesamiento.....	13
2.2. Extracción de coordenadas de puntos de interés	14
2.3. Cálculo de proporciones biométricas	18
2.4. Extracción y cálculo de arrugas.....	19
2.4.1. Filtro Canny.....	21
2.5. Clasificación vía redes neuronales.....	22
2.5.1. Neuronas Artificiales	22
2.5.2. Redes Neuronales Artificiales.....	23
2.5.3. Backpropagation Networks (BPN).....	24
2.5.4. Método de entrenamiento.....	25
2.6. Mejoras propuestas	26
2.6.1. Filtros	27

2.6.2.	Clasificación vía Support Vector Machines	29
2.6.3.	Optimización por PSO.....	37
3.	Implementación	43
3.1.	Bases de datos	43
3.2.	Obtención de la imagen.....	45
3.3.	Algoritmo de Clasificación	46
3.3.1.	Clasificadores	48
3.4.	Optimización	50
4.	Resultados	52
4.1.	Primeros resultados	52
4.2.	Optimización de resultados para la red neuronal	54
4.3.	Optimización de resultados para SVM	57
4.4.	Análisis de resultados	59
5.	Conclusiones.....	61
5.1.	Alcances y trabajos futuros.....	65
6.	Referencias.....	66

1. Introducción

1.1. Antecedentes generales

Dentro del marco de la historia humana, los rostros han desempeñado una importante fuente de información en las comunicaciones sociales, ya sea que entreguen información inmediata con cierta carga emocional, o en un sentido más personalizado, con información propia sobre el individuo. Específicamente, además de expresar emociones, el rostro puede comunicar el género, etnia de origen, edad e incluso a veces experiencias de vida. [16]

Es así como el rostro en sí mismo representa una invaluable fuente de información y punto común para establecer una comunicación en base a un contexto determinado, entregando una referencia entre los individuos en cuanto a su situación y cómo deben reaccionar frente a su interlocutor. [16]

Bajo esta mirada, cuando se tiene en escena un interlocutor de edad avanzada se sabe de antemano que sus reacciones y necesidades serán diametralmente distintas, por ejemplo, a las que podría tener un niño. Por lo cual, la tarea de lograr diferenciar entre grupos etarios constituye una labor muy importante en el contexto de las comunicaciones humanas ya que permite que exista un flujo de información que sea más efectivo entre los interlocutores. [17]

Por esta razón, el reconocimiento de las variaciones faciales ha sido ampliamente estudiado, con especial énfasis en la identificación facial, la identificación de características del rostro, clasificación por género y expresiones faciales. Sin embargo, hay pocos estudios que aborden el tema de la clasificación de edad, especialmente en cuanto a estimación se refiere [17]. La solución del problema pasa por tener categorías específicas para cada rango de edad y saber diferenciarlas, lo cual incluso para un ser humano representa una tarea no trivial.

1.2. Antecedentes específicos

Dentro del marco del presente trabajo se utilizará como base el método de detección e identificación de rostros ya desarrollado en el Laboratorio de Imágenes del Departamento de Ingeniería Eléctrica de la Universidad de Chile, esto para delimitar una base de trabajo con tal de aplicar el algoritmo a diseñar y añadir funcionalidad al software de identificación.

Luego, el trabajo consiste en crear un sistema que sea capaz de entregar una aproximación de la edad del sujeto con tal de clasificarlo dentro de algún rango etario, es decir, la finalidad no es obtener la edad exacta sino más bien, decir con un alto grado de certeza si es, por ejemplo, un niño, un joven, un adulto o un adulto mayor.

Por otro lado, el dato relevante sobre el computador con el cual se trabajará, es que posee un procesador Intel Core 2 Duo a 2.66 GHz con 2 Gb de ram.

1.3. Trabajos previos

La mayoría de las clasificaciones automáticas de la edad a través de imágenes combinan dos componentes: una representación de la imagen y un proceso de estimación de la edad, luego la diferencia se basa en el enfoque utilizado para conseguir dicho objetivo.

Hay cinco diferentes enfoques que son comúnmente usados hasta la fecha para la representación de la imagen:

- a) Modelos antropométricos: su objetivo es obtener una aproximación de la edad e implementar simulaciones para retratar los cambios de la apariencia con el paso del tiempo.
- b) Modelos activos de apariencia: son ampliamente utilizados como una característica para la estimación de edad.

- c) Los colectores de edad donde la subyacente estructura de la edad facial se proyecta en un espacio de características de menor dimensión.
- d) Modelos basados en características que describen la edad del rostro con un conjunto de texturas características discriminantes.
- e) Modelos de simulación de edad, etc.

Con tal de tener un mejor contexto de las bases teóricas es que se entregará un breve resumen de trabajos que han utilizado los diferentes enfoques y posteriormente se procederá a explicar en detalle el método que establecerá la base teórica para el presente desarrollo. A continuación se entrega un breve resumen de algunos de estos métodos.

1.3.1. Transformada wavelets lineal

La transformada wavelets lineal puede ser utilizada para clasificar grupos de gente adulta (16 - 85 años), identificando cambios topológicos de la piel facial, así, este método se basa en la premisa de que la piel de la cara tiende a cambiar drásticamente con el paso del tiempo. Las wavelets poseen la ventaja de entregar una análisis rápido sin tener que utilizar máscaras más grandes que de 5x5 píxeles en varias orientaciones. De acuerdo a los autores del trabajo [3], se obtienen resultados muy precisos y que son indistintos de la pose o la rotación de la cabeza en la fotografía analizada. El método desarrollado alcanza una tasa de aciertos del 90.8% en la base FG-NET, para 5 grupos de edad comprendidos entre los 16 a los 75 años. [3]

1.3.2. Mapas binarios de campos receptivos gaussianos

Los mapas binarios de campos receptivos gaussianos utilizan una pirámide gaussiana de escala invariante de media octava. La representación de la imagen se construye usando operadores gaussianos derivativos. Se asume que estos mapas son una representación robusta de la imagen que codifica cambios faciales a través del proceso de envejecimiento [5]. Una representación tensorial se usa para codificar los cambios de forma del proceso de envejecimiento conservando la estructura tridimensional del espacio. Se utiliza *Principal Component Analysis* (PCA) multilinear para reducir la dimensionalidad del espacio tensorial. Y a mediante *Relevance Vector Machine* (RVM) se realiza el método de regresión para estimar la edad. Éste método en particular entrega una aproximación de la edad del individuo con un error medio de 5.16 años. [5]

1.3.3. Modelo de ranking

El envejecimiento puede ser visto como un problema de series de tiempo [6]. Obviamente, el incremento de edad es monótono con el paso del tiempo, además las características importantes pueden ser útiles para distinguir la edad en diferentes estados con un determinado sujeto. En este trabajo se utiliza un método denominado RankBoost para la extracción de características, que se basan en la relación ordinal entre pares de datos de la imagen. Con ello se puede organizar fácilmente los datos con tal de obtener la información ordinal acorde a la edad de cada sujeto. Se obtiene una tasa de aciertos de un 91% considerando 3 grupos etarios. [6]

1.3.4. Clasificación mediante características gabor y lógica difusa

Se aplica un algoritmo de reducción dimensional lineal para crear un mapa de los rostros humanos desde el espacio de intensidades de los píxeles a un espacio menor, el que facilita la clasificación de edad usando una función cuadrática o SVR. Además como el problema en sí es bastante complejo además implementan las características mediante la transformación *Local Binary Pattern* (LBP) y un algoritmo de Adaboost para construir un clasificador que sea capaz de reconocer 3 clases diferentes: niños, adultos jóvenes y adultos mayores. El método alcanza un 91% de precisión. [7]

1.3.5. Clasificación automática de edad usando LBP

La transformación LBP [8] extrae propiedades fundamentales de la textura de la imagen y a través de un histograma se obtiene un poderoso descriptor de las texturas. Una vez se ha obtenido el mapa LBP de la imagen del rostro, se pasa a la etapa de clasificación donde a través de los métodos de mínima distancia y vecino más cercano, se compara con respecto a los datos de la base de entrenamiento, entregando un resultado que dependerá de que tan bueno sea el entrenamiento que exista para cada una de las distintos grupos etarios. Con 5 grupos etarios distintos, el método alcanza tasa de un 80% de clasificación. [8]

1.4. Fundamentación y objetivos

Como ya se ha mencionado, a través del presente documento se entregará una visión sobre cómo resolver el problema planteado de clasificación etaria, con énfasis en encontrar una solución que sea capaz de entregar un resultado en forma automática y en un tiempo real.

Una detección automática de la edad puede ser utilizada para mejorar sistemas de seguridad, buscar personas perdidas, evitar acceso de menores a ciertas instancias no apropiadas, para focalizar publicidad, entre otros sistemas de interacción humana. Principalmente se persigue ayudar a aplicaciones relacionadas con mercadeo, es decir, en la oferta selectiva de productos o algún estudio del comportamiento de determinados grupos etarios en cuanto a sus preferencias de consumo.

La estimación de edad es un problema complejo puesto que los efectos de la edad dependen mucho de la genética, estilo de vida, lugar de residencia y condiciones climatológicas [1]. Además, hombres y mujeres envejecen de maneras distintas, y los efectos aparentes de envejecimiento son encubiertos con maquillaje o accesorios faciales. Así, añadiendo la influencia de las diferencias individuales, establecer un método general que entregue una estimación de la edad seguirá siendo un problema abierto, ya que aún no se ha logrado un dominio en el tema de parte de algún grupo de expertos. Finalmente, los alcances de este proyecto son muchos al sentar un precedente para futuras investigaciones en un campo que crece día a día.

La estructura del trabajo comprende un estudio teórico a partir del cual se desprende una posible solución al problema. Una vez sentadas las bases de la solución propuesta, se mostrarán los detalles de su implementación práctica y las mejoras que se han realizado a partir de los trabajos en los cuales está basado el proyecto. Para culminar se mostraran los resultados del método desarrollado, analizando su desempeño y alcances.

1.5. Objetivos específicos

- Desarrollar un método para estimar la edad en base a fotografías frontales del rostro, con una tasa de aciertos mínima del 85% y una tasa de falsos positivos no superior a un 10%.
- Estudiar alternativas que permitan ejecutar el algoritmo de estimación en tiempo real.
- Implementar el algoritmo de tal manera que sirva como extensión para softwares de detección de rostros.

1.6. Herramientas

Una de las restricciones fundamentales de este trabajo es que debe operar en tiempo real. Por este motivo se desarrollará un software de procesamiento de las imágenes, el cual será íntegramente programado en el lenguaje C++, utilizando una librería especial denominada OpenCV para el filtrado de la imagen y el proceso de entrenar el clasificador. Por otro lado, para optimizar los resultados entregados por la clasificación, se utilizarán funciones programadas en lenguajes con base en Java. Además se utilizarán diversas bases de datos internacionales como FG-NET y MORPH, especiales para trabajos relacionados con clasificación facial.

2. Clasificación de edad utilizando características faciales

Existen muchos trabajos que utilizan un enfoque basado en el reconocimiento de características faciales, pero a grandes rasgos el procedimiento es el mismo. La base para el presente trabajo se puede encontrar en el trabajo desarrollado por Mohammad Mahdi Dehshibi y Azam Bastanfard [1], quienes alcanzaron una tasa de aciertos de un 86.64% para 4 clases de edad (0 a 15 años, 16 a 35 años, 36 a 50 años y 50 o más años).

A continuación se muestra un diagrama de flujo de la figura 1, donde se grafica el procedimiento estándar para los algoritmos de clasificación de edad y que marcará la pauta de trabajo a desarrollar y posteriormente se explican en detalle cada una de estas etapas.

2.1. Preprocesamiento

Esta etapa se entrega una imagen facial estándar para la extracción de características. Debe ser aplicado a todas las imágenes que sean procesadas por el algoritmo. Generalmente se traspa de una imagen en colores a una en escala de grises. Luego, la región donde se encuentre ubicado el rostro es recortada, redimensionada y filtrada con tal de reducir el ruido, para luego pasar a las siguientes etapas.

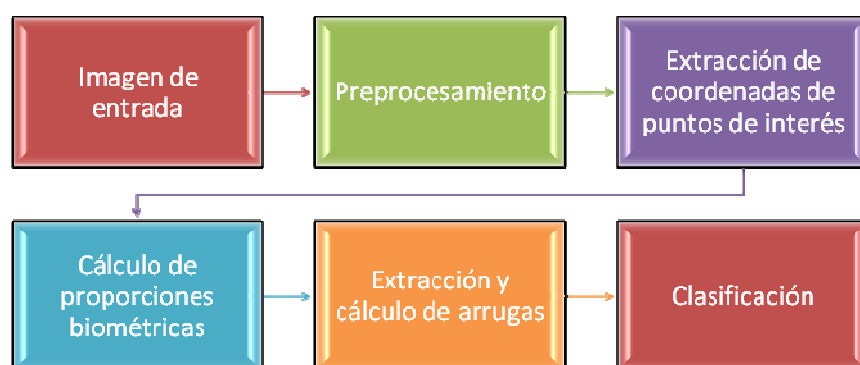


Figura 1. Diagrama del método estándar de clasificación de edad basados en proporciones antropométricas y análisis de texturas

Para la extracción de características se utilizará un detector basado en wavelets gabor desarrollado en el Laboratorio de Imágenes del departamento de ingeniería eléctrica. Dicho software detectará y ubicará el rostro dentro de la imagen y al mismo tiempo determinará la posición del punto de interés más importante para el procedimiento: los ojos del individuo. El área de interés es la correspondiente al rostro, la cual comprende desde la línea en la frente desde la cual comienza a crecer el cabello, hasta el mentón, incluyendo los bordes laterales de la cara.

Posteriormente, con tal de entregar una imagen con menos ruido para el análisis de arrugas y compensada en términos de iluminación, se procede a aplicar filtros pasa-bajos y en algunos casos a ecualizar el histograma de la imagen.

2.2. Extracción de coordenadas de puntos de interés

Para la extracción algunos trabajos utilizan un detector de bordes y a través del ajuste con una malla de puntos predeterminada, hacen calzar los moldes y obtienen las coordenadas deseadas [2], hasta llegar a los métodos más simples los cuales hacen un marcado manual [4]. Sin embargo, a continuación se entrega la metodología más detallada encontrada con respecto a éste procedimiento que resulta crucial para todo el análisis subsiguiente.

El propósito de la extracción de características faciales es el cálculo de razones entre diversas distancias del rostro humano, es decir, realizar un análisis antropométrico del cual se pueda realizar una aproximación de la edad del individuo, hecho postulado por Farkas[9]. Los puntos de interés son los que se detallan en la figura 2 y de estos derivan las siete proporciones que serán utilizadas para extraer la información de la geometría craniofacial de los individuos.

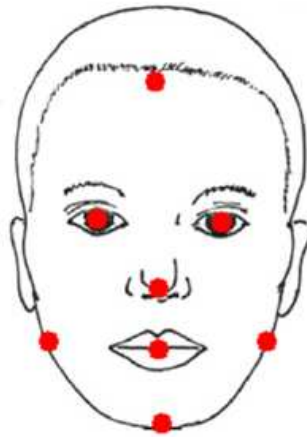


Figura 2. Puntos de interés.

$$\text{Razón 1} = \frac{\text{distancia entre los ojos}}{\text{distancia punto medio entre ojos y nariz}} \quad (1)$$

$$\text{Razón 2} = \frac{\text{distancia entre los ojos}}{\text{distancia punto medio entre ojos y boca}} \quad (2)$$

$$\text{Razón 3} = \frac{\text{distancia entre los ojos}}{\text{distancia punto medio entre ojos y mentón}} \quad (3)$$

$$\text{Razón 4} = \frac{\text{distancia punto medio entre ojos y nariz}}{\text{distancia punto medio entre ojos y boca}} \quad (4)$$

$$\text{Razón 5} = \frac{\text{distancia punto medio entre ojos y boca}}{\text{distancia punto medio entre ojos y mentón}} \quad (5)$$

$$\text{Razón 6} = \frac{\text{distancia entre los ojos}}{\text{distancia entre frente y mentón}} \quad (6)$$

$$\text{Razón 7} = \frac{\text{distancia entre los lados de la cara}}{\text{distancia entre frente y mentón}} \quad (7)$$

Para reducir el tiempo de procesamiento del algoritmo de localización, el rostro es dividido en pequeñas áreas de búsqueda para cada una de las características. Los ojos son la característica más importante y su correcta localización permite identificar todas las otras características faciales de interés. Por lo tanto, el primer paso es la localización del iris y luego basados en propiedades craneofaciales del rostro son ubicadas las áreas de mayor probabilidad, para la búsqueda de las demás características. [1]

Las funciones de proyección son tipos específicos de geometría basado en aproximaciones que usan relaciones antropométricas para localizar dichos puntos [1]. Estas permiten encontrar variaciones de intensidad dentro de la imagen y extraer vectores de características. A pesar del uso extendido de *Variance Projection Function* (VPF) y *Mean Integral Projection function* (MIPF), estos vectores tienen sus limitaciones. MIPF podría fallar cuando la suma de los elementos horizontales y verticales no presentan cambios, mientras que VPF puede fallar en caso de se presente la misma varianza pero diferente media entre sus elementos. Además, las marcas que tengan un contraste relativamente alto podrían no se extraídas correctamente. Para solucionar en parte estos problemas, se construyó una nueva función de proyección, llamada *Combined Projection Function* (CPF). [1]

Matemáticamente, los operadores ortogonales de proyección en un espacio vectorial es cualquier operador que ubica cada punto dentro de su proyección ortogonal en un hiperplano (línea en \mathbb{R}^2 , un plano en \mathbb{R}^3) a través del origen. Si el valor de la información proyectada cambia rápidamente desde cierto punto, entonces dicho punto podría no representar adecuadamente el límite entre dos regiones homogéneas. Esta propiedad de las funciones de proyección es utilizada para la extracción de las características faciales. [1]

Supongamos que $I(x,y)$ representa la intensidad de un píxel en la posición (x,y) , entonces las funciones de proyección se pueden definir como:

$$MIPF_v(y) = \frac{1}{y_2 - y_1} \int_{y_1}^{y_2} I(x, y) dy \quad (8)$$

$$MIPF_h(x) = \frac{1}{x_2 - x_1} \int_{x_1}^{x_2} I(x, y) dx \quad (9)$$

$$VPF_v(y) = \frac{1}{y_2 - y_1} \sum_{y_i=y_1}^{y_2} [I(x, y) - MIPF_v(y)] \quad (10)$$

$$VPF_h(x) = \frac{1}{x_2 - x_1} \sum_{x_i=x_1}^{x_2} [I(x, y) - MIPF_h(x)] \quad (11)$$

Mientras que la función CPF, se puede formular de la siguiente manera:

$$CPF_v(x) = (1 - \mu)VPF'_v(x)^2 + \mu MIPF'_v(x)^2 \quad (12)$$

$$CPF_h(y) = (1 - \mu)VPF'_h(y)^2 + \mu MIPF'_h(y)^2 \quad (13)$$

Donde VPF' y MIPF' son las normalizaciones de VPF y MIPF respectivamente, y μ es un coeficiente de control que se determinará posteriormente. En la figura 3 se muestra cual debiera ser la magnitud esperada de las funciones de proyección para ubicar el centro del ojo.

Además de CPF, se utiliza análisis de componente principal (PCA por sus siglas en inglés) para ubicar las características del rostro. PCA involucra un procedimiento matemático que transforma un número de posibles variables correlacionadas a un pequeño número de variables no correlacionadas llamadas componentes principales. La primera componente principal muestra toda variabilidad relevante que pueda contener, y así sucesivamente las componentes van tomando las variaciones más relevantes que vayan quedando en la imagen. De hecho basta con tomar las primeras dos componentes, el análisis de dichas extracciones ayudará a localizar con precisión las características deseadas. [1]

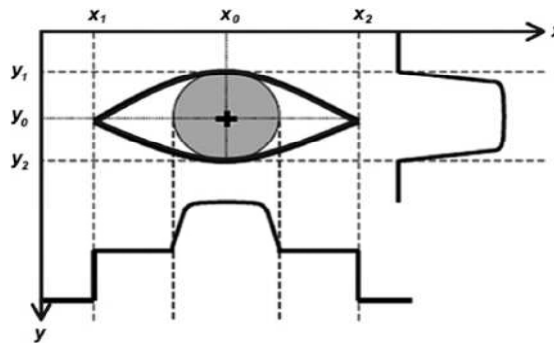


Figura 3. Funciones de Proyección. El resultado gráfico de las funciones de proyección se muestra bajo la imagen (para el eje horizontal) y al lado derecho (para el eje vertical)

2.3. Cálculo de proporciones biométricas

La antropometría es el estudio científico de las medidas y proporciones del cuerpo humano. La antropometría facial nos entrega una descripción cuantitativa de la complejidad craneofacial utilizando las medidas tomadas de las marcas de los puntos de interés del rostro humano a medida que pasa el tiempo. Estudios sobre el crecimiento de las proporciones craneofaciales han demostrado que la forma de la cara va cambiando desde una circunferencia hasta un óvalo a medida que la persona crece. Estos cambios producen cambios de luz en los principales rasgos faciales. En adultos, la posición relativa de los bordes laterales de la cara cambia ligeramente, mientras que las demás distancias relevantes permanecen fijas. Así, estas marcas permiten diferenciar rostros de personas jóvenes del resto. [2]

A continuación se muestra en las figuras 4 y 5 la manera en que se extrae la información de la ubicación de los puntos de interés gracias a la aplicación de las funciones de proyección mencionadas en las ecuaciones 8 a 12 y se muestra cuales son las coordenadas espaciales del centro del ojo dentro de la zona de búsqueda previamente determinada.

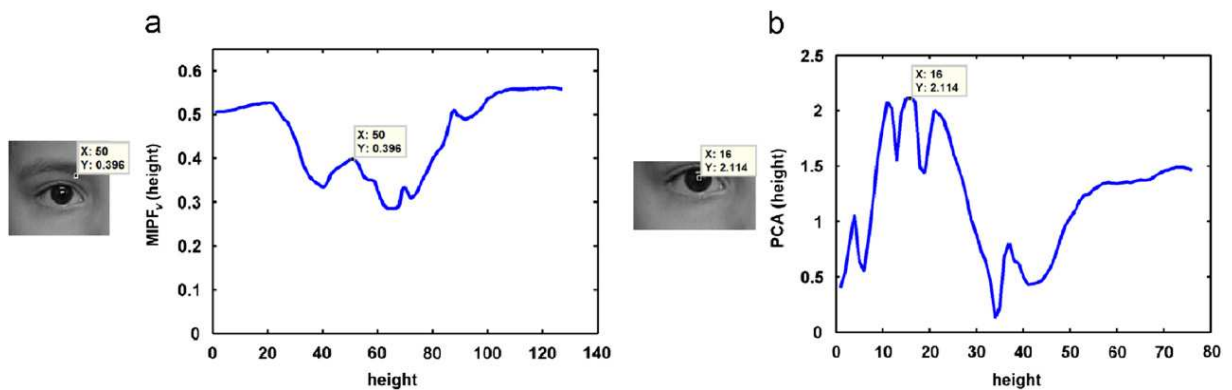


Figura 4. Comparación MIPF (vertical) vs PCA.

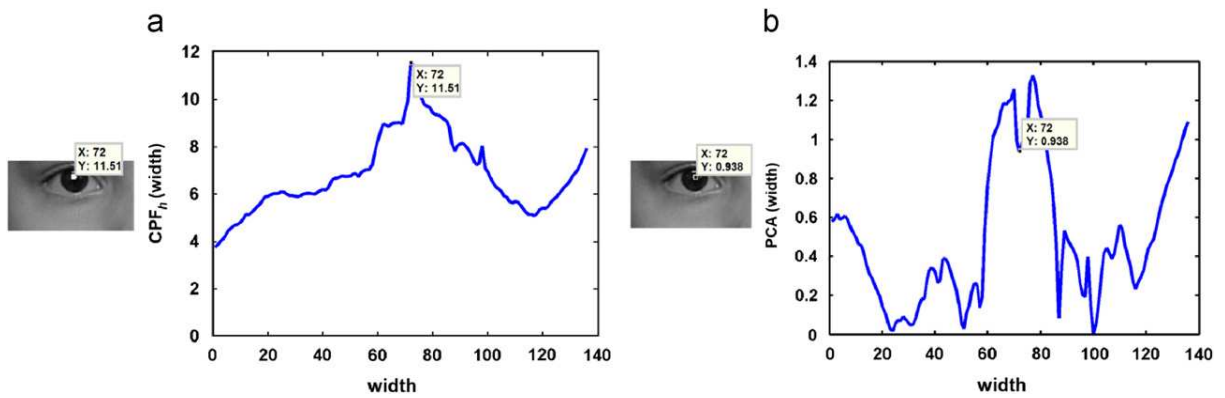


Figura 5. Comparación CPF (horizontal) vs PCA.

Por ejemplo, en bebés la distancia entre ambos ojos es muy cercana a la distancia entre la nariz y los labios. Así es como esta información ayudaría a distinguir a grupos de jóvenes [1]. Luego, se utilizarán 8 marcas importantes (ver figura 2) y 6 distancias relevantes para obtener 7 razones decisivas. Al utilizar dichas proporciones se eliminará la dependencia con la escala de la imagen. [1]

2.4. Extracción y cálculo de arrugas

Cuando las características primarias han sido detectadas, el mapa geográfico de las arrugas (figura 6) se utiliza para determinar donde será aplicado el algoritmo de detección de bordes con tal de encontrarlas [8]. Para cuantificar el grado de los pliegues de piel, se definen tres parámetros, dentro del área A: densidad (cantidad de píxeles no nulos, dividido en el total), valor medio y desviación estándar de los píxeles en la zona luego de aplicar los respectivos filtros. [1][2]

Los adultos mayores a menudo presentan arrugas en la frente, borde de los ojos, bajo las mejillas y en la parte superior de los labios [1]. Luego, se estudiarán las arrugas del rostro que se encuentran remarcadas en la figura 6.

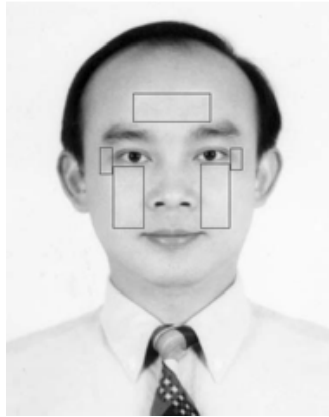


Figura 6. Mapa de las zonas de extracción arrugas en el rostro.

Debido a la variación de los niveles de gris, las arrugas presentan cambios obvios en la intensidad a algunas presentan líneas muy claras. Es importante que los bordes de la imagen no sean pasados por alto o errados. Para encontrar las arrugas se utilizará en principio un detector de bordes Canny, debido a que tiene una baja tasa de error. El método Canny difiere de otros pues utiliza 2 umbrales para detectar bordes débiles, y aquellos que aparecen en la imagen procesada son solo aquellos que se conectan a bordes más claros. [1]

Para este procedimiento, se toma una imagen en escala de grises "I" como entrada, y se retorna una imagen binaria del mismo tamaño. La función devuelve un "1" cuando encuentra alguna arruga y "0" en cualquier otro caso.

Luego, para comparar el grado de pliegues de la piel de los adultos jóvenes con respecto a los más viejos, se calculan los valores antes mencionados. El análisis de las arrugas también puede verse afectado por el uso de accesorios faciales como lentes, o simplemente algo de pelo que se sobreponga a las áreas analizadas (ver figura 6). Por esta razón imágenes que en las zonas de interés presentan anomalías, tales como graves enfermedades de la piel, cicatrices y expresiones faciales particulares podrían inducir a un error y no serán estudiadas.

Ahora, en el trabajo original [1] se utilizaba solo un filtrado con *Canny*, los demás filtros mencionados fueron introducidos al método con tal de mejorar los resultados, pues en un trabajo comparativo [14] se muestran las ventajas de utilizar distintos tipos de filtrados para clasificar los rostros por edad.

2.4.1. Filtro Canny

En 1986, Canny propuso un método para la detección de bordes, el cual se basaba en tres criterios, estos son:

- Un criterio de detección expresa el hecho de evitar la eliminación de bordes importantes y no suministrar falsos bordes.
- El criterio de localización establece que la distancia entre la posición real y la localizada del borde se debe minimizar.
- El criterio de una respuesta que integre las respuestas múltiples correspondientes a un único borde.

Uno de los métodos relacionados con la detección de bordes es el uso de la primera derivada, la que es usada por que toma el valor de cero en todas las regiones donde no varía la intensidad y tiene un valor constante en toda la transición de intensidad. Por tanto un cambio de intensidad se manifiesta como un cambio brusco en la primera derivada, característica que es usada para detectar un borde, y en la que se basa el algoritmo de Canny. [14]

El algoritmo de Canny consiste en tres grandes pasos:

- Obtención del gradiente: en este paso se calcula la magnitud y orientación del vector gradiente en cada píxel.
- Supresión no máxima: en este paso se logra el adelgazamiento del ancho de los bordes, obtenidos con el gradiente, hasta lograr bordes de un píxel de ancho.
- Histéresis de umbral: en este paso se aplica una función de histéresis basada en dos umbrales; con este proceso se pretende reducir la posibilidad de aparición de contornos falsos.

2.5. Clasificación vía redes neuronales

La etapa decisiva después de todo el preprocesamiento, consiste en la clasificación definitiva del individuo. Una de las técnicas más ampliamente utilizadas consiste en la implementación de redes neuronales artificiales, el cual resultó el método escogido por el trabajo original [1].

2.5.1. Neuronas Artificiales

Fueron diseñadas para "emular" las características del funcionamiento básico de las neuronas biológicas. En esencia, se aplica un conjunto de entradas a la neurona, cada una de las cuales representa una salida de otra neurona. Una neurona recibe un número determinado de entradas y obtiene una combinación lineal de ellas. Cada entrada está asociada a un peso individual o ponderación correspondiente al análogo del grado de conexión de la sinapsis. Luego, el resultado de la combinación entregará un valor (denominado grado de certeza) entre 1 y -1 (al menos en la implementación de OpenCV). Esto es importante pues permite a las neuronas calcular funciones lógicas como AND, OR o NOT. Las neuronas que hacen cálculos lógicos se combinan en una red neuronal. La red puede ajustar los pesos para asociar a cada entrada el valor deseado de salida. Esta habilidad de adaptación resulta especialmente útil para problemas donde hay un número finito de salidas, pero no una manera realista de representar todas las posibles entradas. En la figura 8 se puede ver una representación gráfica de una neurona artificial. [2]

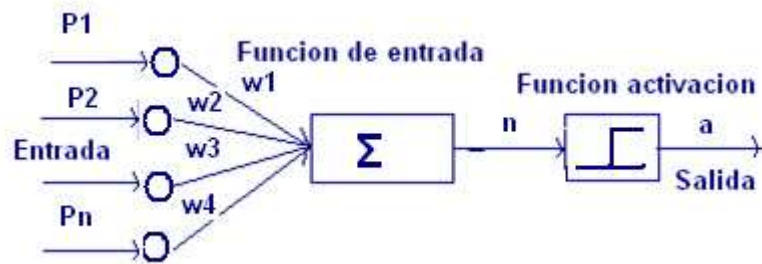


Figura 7. Neurona típica. Se muestran las entradas, la función de combinación y una de activación que determinará el grado de certeza.

2.5.2. Redes Neuronales Artificiales

La capacidad de cálculo y potencia de la computación neuronal proviene de las múltiples conexiones de las neuronas artificiales que constituyen las redes ANN. La red más simple es un grupo de neuronas ordenadas en una capa. Los nodos circulares (ver figura 8) sólo son distribuidores de las entradas y no se consideran constituyentes de una capa.

Las redes multicapa se forman con un grupo de capas simples en cascada. La salida de una capa es la entrada de la siguiente capa. Se ha demostrado que las redes multicapa presentan cualidades y aspectos por encima de las redes de una capa simple. La figura 8 muestra una red de dos capas.

En el contexto de la clasificación de edad, se crea una red neuronal en una estructura que contiene 4 capas. La primera capa consiste en neuronas que son responsables de ingresar adecuadamente la información la imagen del rostro a la red. La segunda y tercera capas se encuentran ocultas y permiten a la red reducir los errores de manera tal de llegar a la salida esperada. La última capa es en donde cada salida posible es representada por una neurona en particular. [2]

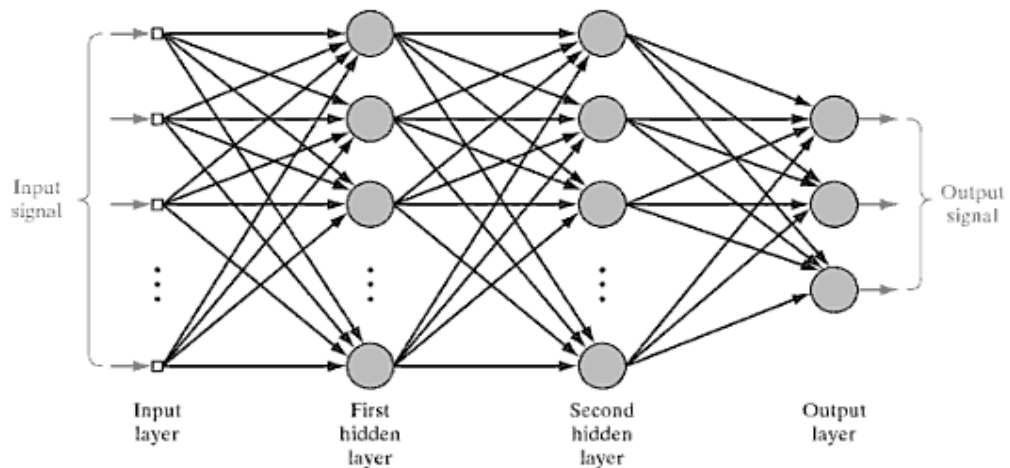


Figura 8. Ejemplo Red Neuronal Multicapa de N entradas y 3 salidas.

2.5.3. Backpropagation Networks (BPN)

Backpropagation es un método de entrenamiento supervisado. A la red se le presenta parejas de patrones, un patrón de entrada emparejado con un patrón de salida deseada. Por cada presentación los pesos son ajustados de forma que disminuya el error entre la salida deseada y la respuesta de la red.[1]

Las neuronas de entrada están totalmente conectadas con la capa oculta y asimismo las salidas de esta se conectan con la capa de salida. La tasa de error de una neurona afecta a la red completa, es por esta razón que minimizar el error resulta crucial, un proceso que eventualmente llevará a la identificación del valor deseado. En backpropagation los datos de entrada serán propagados a través de la red produciendo una salida, esta se compara con el resultado esperado, entregando una tasa de error para la última capa.[2]

Las tasas de error de una neurona en determinada capa se transmiten a la capa siguiente, por lo que los errores también se van propagando. Para evitar este efecto cada neurona variará levemente sus pesos, repitiendo el proceso para cada uno de los valores de entrada. Además de la característica de capas, una red de backpropagation se define por tener proalimentación, lo que significa que todos los lazos son unidireccionales y la red es acíclica.

2.5.4. Método de entrenamiento

El algoritmo de aprendizaje backpropagation conlleva una fase de propagación hacia adelante y otra fase de propagación hacia atrás. Ambas fases se realizan por cada patrón presentado en la sesión de entrenamiento.

En primer lugar los lazos de proalimentación son entrenados utilizando el algoritmo de backpropagation. Esta fase de propagación hacia adelante se inicia cuando se presenta un patrón en la capa de entrada de la red. Cada unidad de la entrada se corresponde con un elemento del vector patrón de entrada. Las unidades de entrada toman el valor de su correspondiente elemento del patrón de entrada y se calcula el valor de activación o nivel de salida de la primera capa. A continuación las demás capas realizarán la fase de propagación hacia adelante que determina el nivel de activación de las otras capas.[4]

Una vez se ha completado la fase de propagación hacia adelante se inicia la fase de corrección o fase de propagación hacia atrás. Los cálculos de las modificaciones de todos los pesos de las conexiones empiezan por la capa de salida y continua hacia atrás a través de todas las capas de la red hasta la capa de entrada. El lazo de retroalimentación sirve para alterar la contribución entregada por las entradas, esto significa que debe ser diseñado de tal manera que produzca diferentes salidas dependiendo de la primera salida del algoritmo de proalimentación, además, el entrenamiento de este lazo se realiza en base a un par de imágenes, el uso de estos pares facilita el ajuste de los pesos del lazo. [4]

La fase de entrenamiento estará completa cuando los lazos de pro y retroalimentación hayan sido correctamente entrenados por separado, es decir, no debe haber interferencias entre ambos procesos. La convergencia es un proceso en el que el valor RMS del error de la red tiende cada vez más al valor 0. La convergencia no siempre es fácil de conseguir porque a veces el proceso puede requerir un tiempo excesivo o bien porque la red alcanza un mínimo local y deja de aprender. [4]

El método propuesto básicamente es un algoritmo supervisado de una red neural con backpropagation, entonces, se ingresa la imagen al sistema, se extraen las características y se clasifica la imagen en uno de los grupos etarios previamente definidos. Generalmente se usa una clasificación basada en 4 grupos etarios: niños, jóvenes, adultos y adultos mayores. Esta es la clasificación a grandes rasgos y no se requiere mayor precisión debido a que siempre surgirán problemas de clasificación incluso para un ser humano (que se considera como un "sistema experto" en cuanto a este tipo de clasificación). [4]

2.6. Mejoras propuestas

Hasta el momento se han revisado paso a paso las etapas del algoritmo utilizado como base para la clasificación de edad, sin embargo, con tal de aportar nuevas herramientas y procedimientos que pudieran traducirse en mejores resultados, es que se propone lo siguiente:

- Agregar mayor cantidad de información de textura al método de aprendizaje computacional, a través de la aplicación de 3 filtros más.
- Estudiar el comportamiento de un algoritmo de aprendizaje computacional alternativo, con la idea de contrastar resultados de clasificación.
- Implementar un algoritmo de optimización con tal de escoger la mejor combinación posible de parámetros tanto para el filtrado de la imagen como ara el algoritmo de aprendizaje, con el objeto de aumentar la tasa de aciertos y/o disminuir el error de clasificación.

2.6.1. Filtros

A continuación se detallan tres filtros extra (con respecto al trabajo base) que se utilizarán para el desarrollo posterior, de modo de obtener una mayor cantidad de información sobre las arrugas presentes en el rostro de los individuos.

2.6.1.1. Filtro Sobel

El filtro Sobel sirve como detector de bordes, que calcula el gradiente de la intensidad de una imagen. Éste consiste en un par de kernels de convolución de 3x3, uno para los cambios verticales y otro para los cambios horizontales. En cada punto de la imagen el resultado aproximado de las gradientes puede ser combinado para entregar la magnitud del gradiente:

$$G = \sqrt{G_x^2 + G_y^2} \quad (14)$$

Donde G_x y G_y son las magnitudes del gradiente calculadas por los respectivos operadores horizontal y vertical. [14]

2.6.1.2. Diferencia entre la imagen original y la suavizada

La imagen original es filtrada (suavizada) por un filtro gaussiano, y luego se le sustrae esta imagen a la original. Así se obtiene una nueva imagen en la cual las diferencias de textura quedan intactas. En otras palabras se obtienen las componentes de alta frecuencia de la imagen original.

La imagen suavizada se obtiene convolucionando la imagen original con los kernels gaussianos, definidos (en 2 dimensiones) por:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (15)$$

Donde, x e y son las coordenadas dentro del kernel, y σ^2 es la desviación estándar del filtro. [14]

2.6.1.3. Diferencia de Gaussianas

En visión computacional, la diferencia de gaussianas es un algoritmo de mejora de imágenes en escala de grises que involucra la sustracción entre un par de imágenes suavizadas con distintas desviaciones estándar. Con lo cual se obtiene un filtrado pasa bandas de la imagen. [14]

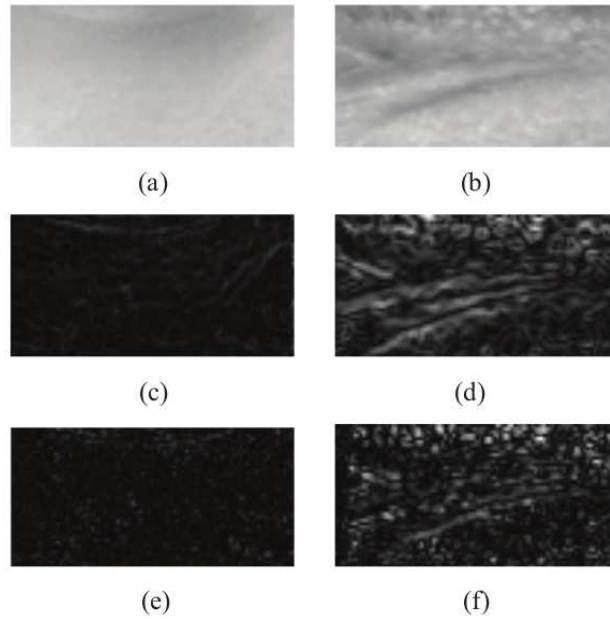


Figura 9. Imagen a) área bajo el ojo de un niño. Imagen b) área bajo el ojo de un adulto mayor. Imágenes c) y d) representan el filtrado con Sobel. Imágenes e) y f) representan las imágenes filtradas por diferencia entra la original y la suavizada.

2.6.2. Clasificación vía Support Vector Machines

Los *Support Vector Machines* (SVM) fueron inventados por Vladimir Vapnik, consistente en un método para la creación de funciones de un conjunto de datos entrenados. La función puede ser una función de clasificación (la salida es binaria: pertenencia o no a una determinada categoría) o puede ser una función de regresión general. [10]

Para clasificación, los SVM's operan encontrando un hiperplano en el espacio de posibles entradas. Este hiperplano intentará separar muestras positivas de muestras negativas. Esta separación escogerá la distancia más grande del hiperplano entre las muestras positivas y negativas más cercanas a esta. Intuitivamente, esta hace la clasificación correcta para analizar los datos que están cerca. Por lo tanto, un hiperplano de separación ideal debe maximizar el margen de separación y minimizar el error de clasificación. Sin embargo, no siempre es posible cumplir los dos objetivos simultáneamente. Para salvar esta dificultad se plantea un problema de optimización cuya función objetivo combina ambos objetivos. Este problema de optimización resulta ser un problema de minimización cuadrático convexo. En el caso que el número de objetos a clasificar es mayor que el número de atributos de cada objeto, lo que usualmente sucede, este problema tiene una única solución óptima. [10]

El reconocimiento de patrones binarios envuelve construcciones de reglas de decisión para clasificar vectores dentro de una de dos clases basadas en conjuntos de vectores entrenados cuya clasificación es conocida a priori. Los SVM's implican un mapeo de datos entrenados dentro de espacios característico de dimensión grande. Un hiperplano (superficie de decisión) es el resultado de la bisección de las dos categorías dentro del espacio característico construido, el cual además maximiza el margen de separación entre sí mismo y sus puntos quedando cerca de esta (llamados support vector). Esta superficie de decisión puede entonces usarse como base para clasificación de vectores de clasificación desconocida. [11]

Las ventajas principales del enfoque los SVM's son las siguientes:

- El enfoque a encontrar un compromiso entre la minimización de riesgo empírico y la prevención de sobreajuste.

- No existe un mínimo global y el problema se resuelve usando técnicas de programación cuadrática.
- El clasificador resultante puede ser especificado completamente en términos de sus support vectors y su función tipo kernel.

2.6.2.1. Obtención del Hiperplano Óptimo de Separación

Consideremos un problema de clasificación binaria para el cual ya se ha definido el conjunto de entrenamiento. En este caso, los objetos a clasificar son los rostros. Supongamos que para cada rostro se han definido n variables a estudiar y que se tiene m individuos en el conjunto de entrenamiento. Entonces, cada rostro es representado por un vector característico de dimensión $n+1$ cuyas primeras n coordenadas corresponden a las variables del estudio y la última corresponde a la clase de pertenencia. En particular, denotaremos por (X_i, Y_i) al vector característico correspondiente al individuo i (para $i = 1, \dots, m$). [10]

Un supuesto inicial para este trabajo es que el conjunto de entrenamiento es linealmente separable. Es decir, existe un hiperplano en \mathfrak{R}^N que deja a todos los vectores X asociados a los individuos de una clase de un lado del hiperplano y a los de la otra del otro lado. Formalmente, existe un par $(\alpha, b) \in \mathfrak{R}^{N+1}$ tal que $\alpha \cdot X_i + b > 0$ si el individuo i está en la clase positiva y $\alpha \cdot X_i + b < 0$, si está en la clase negativa. [10]

De esta manera, dados un vector no nulo normal al hiperplano de separación $\alpha \in \mathfrak{R}^N$ podemos definir una función de clasificación $f : X \rightarrow \mathfrak{R}^N$ por la expresión $f(x) = \alpha \cdot X + b$. Entonces si $f(X_i) > 0$, el individuo i está en la clase positiva y si $f(X_i) < 0$, i está en la clase negativa. Para definir el margen de clasificación, consideramos las distancias d^+ y d^- . La distancia d^+ es la distancia euclídeana entre el hiperplano y la clase positiva. Es decir, la distancia entre el hiperplano y el punto en la clase positiva más cercano a este. Una vez determinadas estas distancias, definimos el margen de separación (del hiperplano) como la suma $d^+ + d^-$.

2.6.2.2. Caso linealmente separable

Supongamos que nos han dado un conjunto S de puntos etiquetados para entrenamiento como se aprecia en la Figura 10, cada punto de entrenamiento $X_i \in \mathfrak{R}^N$ pertenece además a alguna de dos clases y se le ha dado una etiqueta $Y_i \in \{-1, 1\}$ para $i = 1, \dots, l$. En la mayoría de los casos, la búsqueda de un hiperplano adecuado en un espacio de entrada es demasiado restrictivo para ser de uso práctico. Una solución a esta situación es mapear el espacio de entrada en un espacio de características de una dimensión mayor y buscar el hiperplano óptimo allí. Sea $z = \mathfrak{Z}(x)$ la notación del correspondiente vector en el espacio de características con un mapeo \mathfrak{Z} de \mathfrak{R}^N a un espacio de características Z . [11]

Deseamos encontrar el hiperplano $w \cdot z + b = 0$. Definido por el par (w, b) , tal que podamos separar el punto X_i de acuerdo a la función:

$$f(x_i) = \text{sign}(w \cdot z_i + b) = \begin{cases} 1 & y_i = 1 \\ -1 & y_i = -1 \end{cases} \quad (16)$$

Donde $w \in Z$ y $b \in \mathfrak{R}$. Más precisamente, el conjunto S se dice que es linealmente separable si existe (w, b) tal que las inecuaciones

$$\begin{cases} (w \cdot z_i + b) \geq 1, & y_i = 1 \\ (w \cdot z_i + b) \leq -1, & y_i = -1 \end{cases} \quad i=1, \dots, l \quad (17)$$

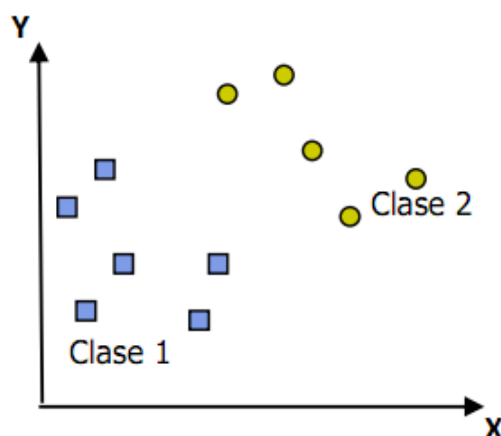


Figura 10. Representación gráfica de datos linealmente separables de 2 clases en un determinado plano.

Sean válidas para todos los elementos del conjunto S . Para el caso linealmente separable de S , podemos encontrar un único hiperplano óptimo, para el cual, el margen entre las proyecciones de los puntos de entrenamiento de dos diferentes clases es maximizado. [11]

2.6.2.3. Caso linealmente no separable.

Si el conjunto S no es linealmente separable, violaciones a la clasificación deben ser permitidas en la formulación de la SVM. Un ejemplo gráfico de ello lo podemos observar en la figura 11.

Para tratar con datos que no son linealmente separables, el análisis previo puede ser generalizado introduciendo algunas variables no negativas $\xi_i \geq 0$ de tal que (17) es modificado a:

$$y_i(w \cdot z_i + b) \geq 1 - \varepsilon_i, \quad i = 1, \dots, l. \quad (18)$$

Los $\xi_i \neq 0$ en (e.18) son aquellos para los cuales el punto X_i no satisface (e.17). Entonces el término $\sum_{i=1}^l \varepsilon_i$ puede ser tomado como algún tipo de medida del error en la clasificación. El problema del hiperplano óptimo es entonces redefinido como la solución al problema:

$$\begin{aligned} \min \left\{ \frac{1}{2} w \cdot w + C \sum_{i=1}^l \varepsilon_i \right\} & \quad (19) \\ \text{s. a } y_i(w \cdot z_i + b) \geq 1 - \varepsilon_i, & \quad i=1, \dots, l \\ \varepsilon_i \geq 0, & \quad i=1, \dots, l \end{aligned}$$

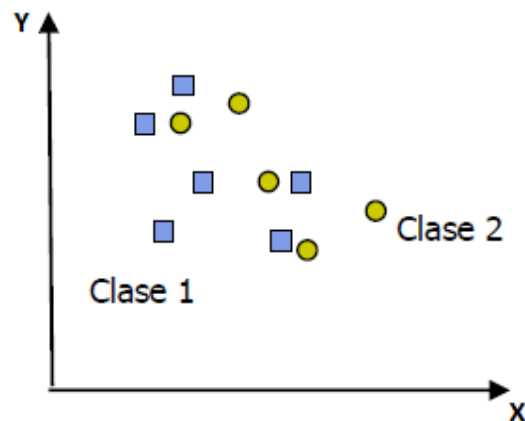


Figura 11. Caso linealmente no separable. Las muestras de las distintas clases se traslapan en el plano observado.

Donde C es una constante. El parámetro C puede ser definido como un parámetro de regularización. Este es el único parámetro libre de ser ajustado en la formulación de la SVM. El ajuste de éste parámetro puede hacer un balance entre la maximización del margen y la violación a la clasificación. [11]

Luego, buscar el hiperplano óptimo en (e.19) es un problema QP, que puede ser resuelto construyendo un Lagrangiano y transformándolo en el dual:

$$\text{Max } W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j z_i \cdot z_j \quad (20)$$

$$\text{s. a } \sum_{i=1}^l y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i=1, \dots, l$$

Donde $\alpha = (\alpha_1, \dots, \alpha_l)$ es un vector de multiplicadores de Lagrange positivos asociados con las constantes en (e.18).

El teorema de Khun-Tucker juega un papel importante en la teoría de las SVM. De acuerdo a este teorema, la solución $\bar{\alpha}_i$ del problema (e.20) satisface:

$$\bar{\alpha}_i (y_i (\bar{w} \cdot z_i + \bar{b}) - 1 + \bar{\varepsilon}_i) = 0, \quad i=1, \dots, l \quad (21)$$

$$(C - \bar{\alpha}_i) \varepsilon_i = 0, \quad i=1, \dots, l \quad (22)$$

De esta igualdad se deduce que los únicos valores $\bar{\alpha}_i \neq 0$ (e.22) son aquellos que para las constantes en (e.18) son satisfechas con el signo de igualdad. El punto X_i correspondiente con $\bar{\alpha}_i > 0$ es llamado vector de soporte. Pero hay dos tipos de vectores de soporte en un caso no separable. En el caso $0 < \bar{\alpha}_i < C$, el correspondiente vector de soporte X_i satisface las igualdades $Y_i (\bar{w} \cdot z_i + \bar{b}) = 1$ y $\xi_i = 0$. En el caso $\bar{\alpha}_i = C$, el correspondiente ξ_i es diferente de cero y el correspondiente vector de soporte X_i no satisface las ecuaciones iniciales. Nos referimos a estos vectores de soporte como errores. El punto X_i correspondiente con $\bar{\alpha}_i = 0$ es clasificado correctamente y está claramente alejado del margen de decisión, ver figura 12. [11]

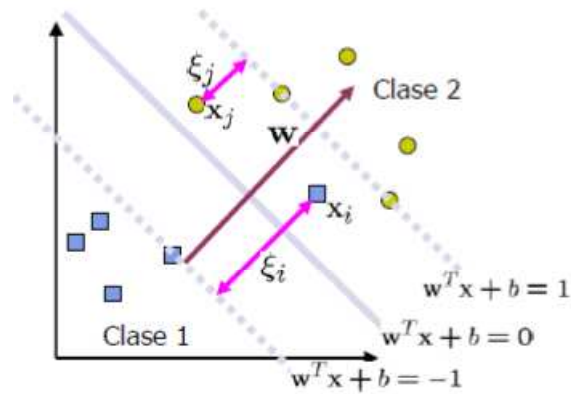


Figura 12. Aparición del parámetro ξ_i en el error de clasificación.

Para construir el $(\bar{w} \cdot z_i + \bar{b})$ hiperplano óptimo, se utiliza:

$$\bar{w} = \sum_{i=1}^l \bar{\alpha}_i y_i z_i \quad (23)$$

Y el escalar b puede ser determinado de las condiciones de Kuhn-Tucker. La función de decisión generalizada es tal que:

$$f(x) = \text{sign}(w \cdot z + b) = \text{sign}\left(\sum_{i=1}^l \alpha_i y_i z_i \cdot z + b\right) \quad (24)$$

2.6.2.4. Kernel para casos no linealmente separables.

Debido a las limitaciones computacionales de las máquinas de aprendizaje lineal, éstas no pueden ser utilizadas en la mayoría de las aplicaciones del mundo real. La representación por medio de funciones Kernel ofrece una solución a este problema, proyectando la información a un espacio de características de mayor dimensión el cual aumenta la capacidad computacional de la máquina de aprendizaje lineal [15].

Es decir, mapearemos el espacio de entradas X a un nuevo espacio de características de mayor dimensionalidad (Hilbert):

$$F = \{\phi(x) | x \in X\}$$

$$X = \{x_1, x_2, \dots, x_n\} \rightarrow \phi(x) = \{\phi(x)_1, \phi(x)_2, \dots, \phi(x)_n\}$$

A continuación se muestra brevemente la expresión matemática de algunos de los kernels comúnmente utilizados:

- **Polinomial-homogénea:**

$$K(x_i, x_j) = (x_i \cdot x_j)^n \quad (25)$$

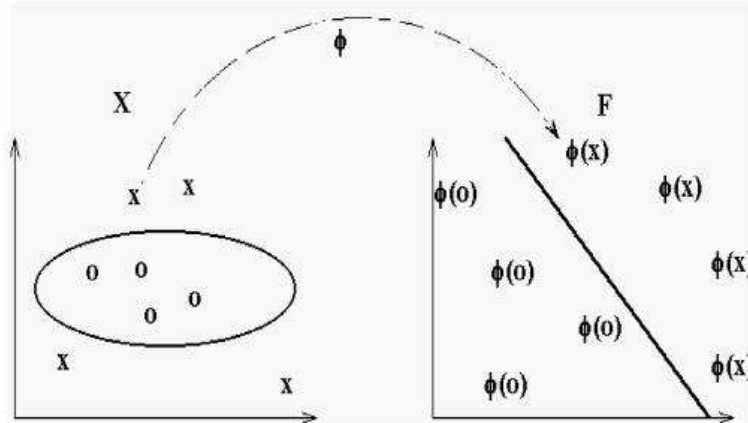


Figura 13. Función con kernel Polinomial. [12]

- **Perceptrón:**

$$K(x_i, x_j) = ||x_i - x_j|| \quad (26)$$

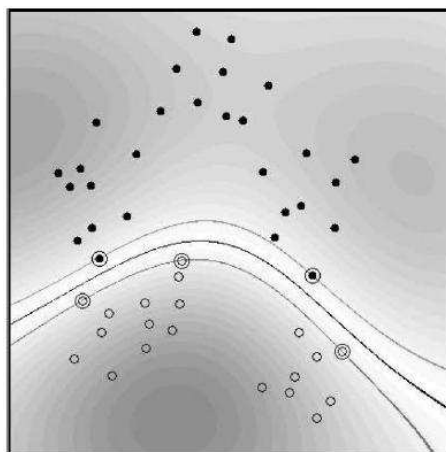


Figura 14. Función con kernel Perceptrón. [12]

- **Sigmoide:**

$$K(x_i, x_j) = \tanh(x_i \cdot x_j - \theta) \quad (28)$$

- **Función de base radial Gaussiana:**

$$K(x_i, x_j) = e^{-\frac{(x_i - x_j)^2}{2\sigma^2}} \quad (27)$$

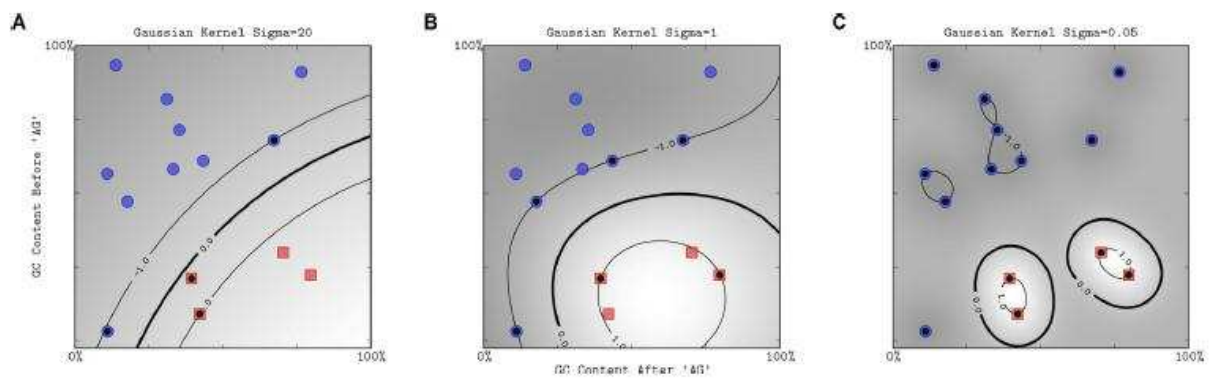


Figura 15. Función con kernel Gaussiano. [12]

2.6.3. Optimización por PSO

Como se podrá intuir, un procedimiento tan complejo como el propuesto requerirá además una etapa exclusivamente dedicada a la selección de parámetros, dentro de los cuales se consideran los siguientes:

- Umbrales y tamaños de cada uno de los filtros.
- Características de la red neuronal.

Dichos parámetros determinarán el desempeño del algoritmo de clasificación, y su selección quedará en manos de un método de optimización computacional, denominado *Particle Swarm Optimization (PSO)*.

Un algoritmo basado en PSO es una técnica metaheurística basada en poblaciones e inspirada en el comportamiento social del vuelo de las bandadas de aves o el movimiento de los bancos de peces. PSO fue originalmente desarrollado por el psicólogo y sociólogo James Kennedy y por el ingeniero electrónico Russell Eberhart en 1995, basándose en un enfoque conocido como la metáfora social [13], que describe a este algoritmo y que se puede resumir de la siguiente forma: los individuos que conviven en una sociedad tienen una opinión que es parte de un conjunto de creencias (el espacio de búsqueda) compartido por todos los posibles individuos. Cada individuo puede modificar su propia opinión basándose en tres factores:

- Su conocimiento sobre el entorno (su valor de fitness).
- Su conocimiento histórico o experiencias anteriores (su memoria).
- El conocimiento histórico o experiencias anteriores de los individuos situados en su vecindario.

Siguiendo ciertas reglas de interacción, los individuos en la población adaptan sus esquemas de creencias al de los individuos con más éxito de su entorno. Con el tiempo, surge una cultura cuyos individuos tienen un conjunto de creencias estrechamente relacionado.

El principio natural en el que se basa PSO es el comportamiento de una bandada de aves o de un banco de peces: supongamos que una de estas bandadas busca comida en un área y que solamente hay una pieza de comida en dicha área. Los pájaros no saben dónde está la comida pero sí conocen su distancia a la misma, por lo que la estrategia más eficaz para hallar la comida es seguir al ave que se encuentre más cerca de ella. PSO emula este escenario para resolver problemas de optimización. Cada solución (partícula) es un ave en el espacio de búsqueda que está siempre en continuo movimiento y que nunca muere. [13]

El cúmulo de partículas (swarm) es un sistema multiagente, es decir, las partículas son agentes simples que se mueven por el espacio de búsqueda y que guardan (y posiblemente comunican) la mejor solución que han encontrado. Cada partícula tiene un fitness, una posición y un vector velocidad que dirige su movimiento. El movimiento de las partículas por el espacio está guiado por las partículas óptimas en el momento actual.

Los algoritmos basados en cúmulos de partículas se han aplicado con éxito en diferentes campos de investigación. Algunos ejemplos son: optimización de funciones numéricas, entrenamiento de redes neuronales, aprendizaje de sistemas difusos, registrado de imágenes, problema del viajante de comercio e ingeniería química. [13]

2.6.3.1. Descripción del Algoritmo PSO

Un algoritmo PSO consiste en un proceso iterativo y estocástico que opera sobre un cúmulo de partículas. La posición de cada partícula representa una solución potencial al problema que se está resolviendo. Generalmente, una partícula p_i está compuesta de tres vectores y dos valores de fitness:

- El vector $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ almacena la posición actual (localización) de la partícula en el espacio de búsqueda.
- El vector $pBest_i = (p_{i1}, p_{i2}, \dots, p_{in})$ almacena la posición de la mejor solución encontrada por la partícula hasta el momento.
- El vector de velocidad $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$ almacena el gradiente (dirección) según el cual se moverá la partícula.
- El valor de fitness $fitness_x_i$ almacena el valor de adecuación de la solución actual (vector x_i).
- El valor de fitness $fitness_pBest_i$ almacena el valor de adecuación de la mejor solución local encontrada hasta el momento (vector $pBest_i$).

El cúmulo se inicializa generando las posiciones y las velocidades iniciales de las partículas. Las posiciones se pueden generar aleatoriamente en el espacio de búsqueda (quizás con ayuda de un heurístico de construcción), de forma regular o con una combinación de ambas formas. Una vez generadas las posiciones, se calcula el fitness de cada una y se actualizan los valores de $fitness_x_i$ y $fitness_pBest_i$. En la imagen de la figura 16 se puede ver una representación gráfica de la inicialización de los puntos de búsqueda en un determinado plano.

[13]

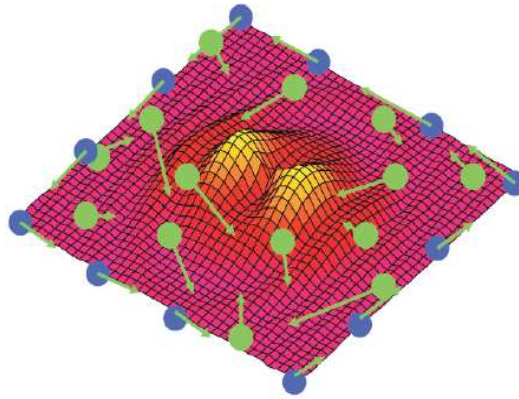


Figura 16. Inicialización del cúmulo en el espacio de búsqueda.

Las velocidades se generan aleatoriamente, con cada componente en el intervalo $[-v_{\max}; v_{\max}]$, donde v_{\max} será la velocidad máxima que pueda tomar una partícula en cada movimiento. No es conveniente dejarlas en cero pues no se obtienen buenos resultados. [13]

Inicializado el cúmulo (Figura 16), las partículas se deben mover dentro del proceso iterativo. Una partícula se mueve desde una posición del espacio de búsqueda hasta otra, simplemente, añadiendo al vector posición x_i el vector velocidad v_i para obtener un nuevo vector posición:

$$x_i \leftarrow x_i + v_i \quad (29)$$

Una vez calculada la nueva posición de la partícula, se evalúa actualizando fitness_{x_i} . Además, si el nuevo fitness es el mejor que el encontrado hasta el momento, se actualizan los valores de mejor posición $pBest_i$ y fitness fitness_{pBest_i} . El vector velocidad de cada partícula es modificado en cada iteración utilizando la velocidad anterior, un componente cognitivo y un componente social. El modelo matemático resultante y que representa el corazón del algoritmo PSO viene representado por las siguientes ecuaciones:

$$v_i^{k+1} = w \cdot v_i^k + \varphi_1 \cdot rand_1 \cdot (pBest_i - x_i^k) + \varphi_2 \cdot rand_2 \cdot (g_i - x_i^k) \quad (30)$$

$$x_i^{k+1} = x_i^k + v_i^k \quad (31)$$

La ecuación de velocidad refleja la actualización del vector velocidad de cada partícula i en cada iteración k . El componente cognitivo está modelado por el factor $\varpi_1 \cdot \text{rand}_1 \cdot (p\text{Best}_i - x_i^k)$ y representa la distancia entre la posición actual y la mejor conocida por esa partícula, es decir, la decisión que tomará la partícula influenciada por su propia experiencia a lo largo de su vida. El componente social está modelado por $\varpi_2 \cdot \text{rand}_2 \cdot (g_i - x_i^k)$ y representa la distancia entre la posición actual y la mejor posición del vecindario, es decir, la decisión que tomará la partícula según la influencia que el resto del cúmulo ejerce sobre ella. [13]

Una descripción más detallada de cada factor se realiza a continuación:

- v_i^k : velocidad de la partícula i en la iteración k ,
- w : factor inercia,
- $\varpi_1; \varpi_2$: son ratios de aprendizaje (pesos) que controlan los componentes cognitivo y social,
- $\text{rand}_1; \text{rand}_2$: números aleatorios entre 0 y 1,
- x_i^k : posición actual de la partícula i en la iteración k ,
- $p\text{Best}_i$: mejor posición (solución) encontrada por la partícula i hasta el momento,
- g_i : representa la posición de la partícula con el mejor $p\text{Best_fitness}$ del entorno de p_i ($l\text{Best}$ o $l\text{ocalbest}$) o de todo el cúmulo ($g\text{Best}$ o $g\text{lobalbest}$).

La ecuación para X modela el movimiento de cada partícula i en cada iteración k . En la figura 17 se muestra una representación gráfica del movimiento de una partícula en el espacio de soluciones.

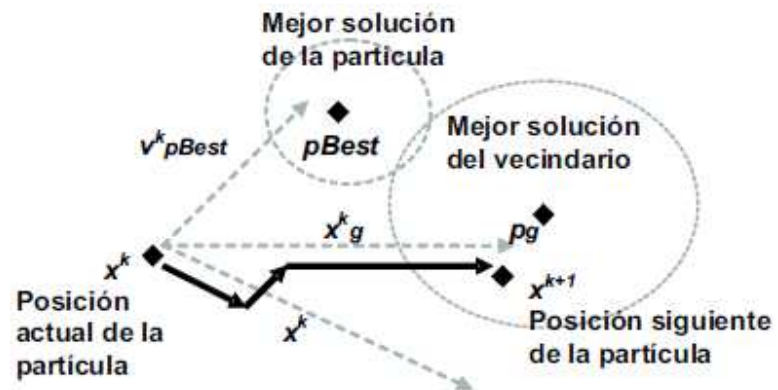


Figura 17. Movimiento de una partícula en el espacio de soluciones.

En esta gráfica, las flechas de línea discontinua representan la dirección de los vectores de velocidad actual: v_{pBest}^k es la velocidad de la mejor posición tomada por la partícula, v_g^k es la velocidad de la mejor partícula encontrada en el vecindario y v^k es la velocidad actual de la partícula. La echa de línea continua representa la dirección que toma la partícula para moverse desde la posición x^k hasta la posición x^{k+1} . El cambio de dirección de esta echa depende de la influencia de las demás direcciones (gradiente) que intervienen en el movimiento. [13]

3. Implementación

Aclarados los conceptos teóricos de los cuales se deriva el trabajo realizado, se detalla la implementación misma del método teórico elegido, que consistente en la clasificación de edad vía características faciales del individuo. Para tener en claro en qué consiste todo el proceso que será detallado a continuación se presenta el diagrama de las respectivas etapas de la figura 18.

Por otro lado, resulta importante recordar que el método desde la etapa de obtención de la imagen hasta la clasificación se confeccionó utilizando el lenguaje C++ con librerías de OpenCV, las cuales proveen las herramientas tanto para la manipulación de la imagen como de aprendizaje computacional.

3.1. Bases de datos

Con tal de confeccionar un procedimiento de aprendizaje computacional, se requiere una base de datos con la cual entrenar y testear el desempeño del algoritmo, con este motivo se escogieron dos bases de datos ampliamente utilizadas para este tipo de investigaciones (clasificación de edad). Estas son FG-NET y MORPH, de las cuales, dadas las limitaciones del algoritmo se seleccionó un subgrupo, el cual compartía las características de presentar un individuo sin imperfecciones serias (tales como cicatrices o muchos accesorios) y además una pose frontal del rostro. Así se confeccionó una base ad-hoc consistente en un total de 1500 imágenes, en la cual los individuos tienen una distribución de edad como se muestra a continuación.



Figura 18. Diagrama del proceso de clasificación de edad.

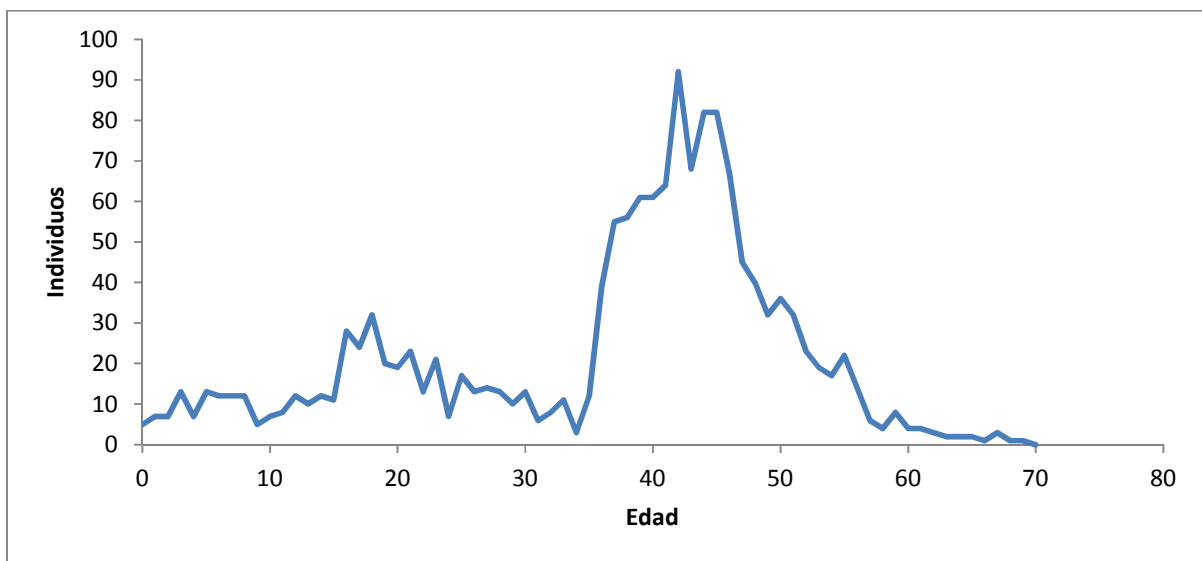


Figura 19. Distribución de edad de los individuos de la base ad-hoc.

Es importante señalar que dentro de la selección realizada no existen individuos mayores a 70 años, lo cual implica otra fuente de sesgo para el clasificador, sin embargo, dadas las condiciones del envejecimiento, se espera que para la clasificación de personas mayores no resulte en un perjuicio ya que las características de textura debieran resultar aún más acentuadas.

Así, en función de la cantidad de las clases que se utilizarán para el entrenamiento del clasificador, se hizo necesario acotar la cantidad de individuos presentes en determinados rangos de edad, pues como se ve en la distribución, dentro del rango de los 35 a 55 años aproximadamente, se presenta el grueso del total de individuos, así en función de las particiones de edad que se usarán, se presentan a priori la cantidad de individuos que conformarán las bases de entrenamiento (60%), validación y testeo (20% respectivamente) para diferentes clasificaciones de edad.

Tabla 1. Sub-base para 3 clases de edad.

Años	Total	Entrenamiento	Test
0-20	257	180	77
21-45	470	329	141
46-70	470	329	141
Total	1197	838	359

Tabla 2. Sub-base para 4 clases de edad.

Años	Total	Entrenamiento	Test
0-16	153	107	46
17-30	254	178	76
31-50	254	178	76
51-70	204	143	61
Total	865	606	259

Tabla 3. Sub-base para 5 clases de edad.

Años	Total	Entrenamiento	Test
0-10	93	65	28
11-20	164	115	49
21-36	203	142	61
37-50	203	142	61
51-70	204	143	61
Total	867	607	260

3.2. Obtención de la imagen

Dado el enfoque del trabajo realizado, el método debe ser compatible con la salida otro software el cual aplica un método de detección de rostros, entregando una imagen con las coordenadas de la cara y posición de los ojos del individuo. Este será el punto de partida para las operaciones subsecuentes, por lo tanto, se tiene una fuerte restricción para el método a desarrollar, puesto que depende directamente de qué tan efectivo sea el detector y qué tan precisas sean las coordenadas de ubicación tanto del rostro como de ambos ojos y finalmente del grado de rotación del rostro dentro de la imagen comparado con una postura netamente frontal de la cara ante la cámara que obtendrá en primera instancia las imágenes.

3.3. Algoritmo de Clasificación

Una vez obtenida la imagen, el rostro será redimensionado a un tamaño de 200x200 píxeles con tal de realizar el procesamiento antes mencionado. Por otro lado, como ya son conocidas las posiciones de ambos ojos, detectar el resto de los puntos de interés será una tarea algo más fácil, puesto que se puede crear una máscara donde buscar el resto de los puntos de interés.

Se procede entonces a delimitar las áreas de búsqueda (con base a propiedades antropométricas) de los rasgos faltantes, estos son: nariz, boca, mentón, borde lateral del rostro y frente (delimitado por la parte superior del recorte del rostro), siguiendo el procedimiento de búsqueda detallado en el capítulo anterior. En la figura 20 es posible apreciar los puntos de interés y además las zonas de búsqueda respectivas.

Una vez se han obtenido las posiciones de estas características y en base a las mismas, se determinan las 5 áreas de extracción de arrugas (ver figura 21): frente, lado externo de los ojos, y las zonas bajo estos hasta la altura de la boca, dicho sea de paso éstas áreas son proporcionales al tamaño del recorte original de la cara, con tal de hacer el algoritmo invariable a la resolución de la imagen. Se procede entonces a extraer la información de textura a través de 4 tipos de filtrado: Canny, diferencia de Gaussianas, Gaussiano pasa altos y Sobel.

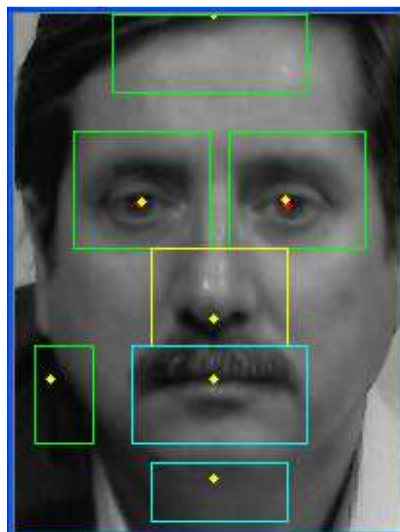


Figura 20. Máscara antropométrica de búsqueda de puntos de interés.

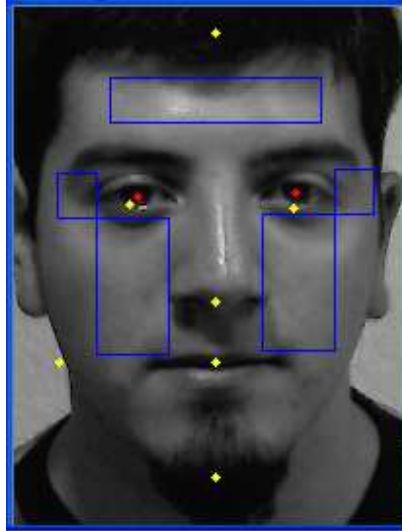


Figura 21. Puntos de Interés y áreas de extracción de arrugas.

En base a los procesos mencionados se obtienen entonces 7 proporciones geométricas (detalladas en el capítulo anterior), además de 12 características de textura por cada una de las 5 zonas, los cuales representan la entrada al clasificador final. Las 12 características de textura corresponden a 3 valores por cada uno de los 4 filtros aplicados dentro de cada área, estos son:

- a) Razón de píxeles nulos versus el total del área.
- b) Valor medio de los píxeles.
- c) Desviación estándar.

Cabe señalar que luego de la normalización de las características, necesaria para ingresar cada una de ellas al clasificador, los valores de a) y b) son idénticos para el filtro Canny, por lo tanto, el total de valores ingresados al clasificador asciende a las 62 características (7 proporciones geométricas más 11 características de textura por cada una de las 5 áreas de extracción).

3.3.1. Clasificadores

3.3.1.1. Red neuronal

Permite obtener una aproximación numérica de la edad del individuo, por lo que se puede utilizar como estimador de edad. Esto ocurre por la naturaleza de sus salidas, las cuales pueden asignar grados de pertenencia a cada una de las clases (valores entre 1 y -1), lo cual combinado al hecho de ingresar los valores esperados para la salida en la etapa de entrenamiento, puede resultar en una implementación muy flexible que permita cumplir 2 objetivos a la vez: clasificación (a través del grado de pertenencia a las respectivas clases) y estimación de edad (a través de una combinación de las salidas de la red para una determinada imagen).

Entonces, en función de la edad real del individuo, al momento de entrenar se entregarán 2 valores objetivos, correspondientes al grado de pertenencia a las 2 clases más representativas de la edad del individuo, de modo tal que el valor más representativo (el promedio de la clase correspondiente) sea 1 en caso de caer justo al medio de la clase, y 0.5 en caso de caer justo en el límite, luego se asignará un valor tal que multiplicado por el valor medio de la clase más cercana complete la edad del individuo; finalmente ambos valores serán normalizados de tal manera que sumen 1. Esto se hará para entregar el vector objetivo de entrenamiento de la red, y luego para reconstruir el valor de la edad del individuo en la etapa de evaluación, se normalizarán los valores de pertenencia a cada clase, luego se multiplicarán por el valor medio correspondiente a cada una, y se sumarán. Esta técnica fue elegida ya que presentaba los menores errores de estimación de edad; así, luego de la aproximación se asigna la clasificación que corresponda.

Dentro de las pruebas de optimización, que serán explicadas en breve, se decidió que la mejor configuración está dada por una red que tenga 2 capas ocultas, cuyo número de neuronas constituyentes, constante de momento y paso fueron determinados en las pruebas que se detallan más adelante. En cuanto a los grupos etarios que es capaz de clasificar, dado que se trata de un clasificador multiclase, no habrá problema para clasificar 3, 4 o 5 grupos etarios que se escojan.

3.3.1.2. SVM

Sus características como clasificador son superiores a las de la red neuronal, por lo cual puede resultar muy útil al ir desglosando el problema por grupos etarios. La desventaja es su naturaleza binaria, lo cual a priori complica la tarea de clasificación al enfrentarse a más de 2 clases.

Así para el presente trabajo se optó por utilizar como máximo 3 clasificadores binarios con tal de no comprometer mayormente el tiempo de procesamiento, con lo cual, si se tienen clasificadores en paralelo, es posible clasificar 3 grupos etarios (C_1 , C_2 y C_3) utilizando las siguientes combinaciones:

- Enfoque uno a uno: $SVM_1 = C_1$ vs C_2 , $SVM_2 = C_1$ vs C_3 y $SVM_3 = C_2$ vs C_3 .
- Enfoque uno contra todos: $SVM_i = C_i$ vs Resto ($i = 1,2,3$)

Por otro lado, si estos 3 clasificadores son ordenados en cascada, pueden clasificarse 4 grupos etarios distintos (C_1 , C_2 , C_3 y C_4), utilizando la siguiente configuración:

- $SVM_1 = (C_1$ y $C_2)$ vs (C_3 y C_4).
- $SVM_2 = C_1$ vs C_2 .
- $SVM_3 = C_3$ vs C_4 .

Como el clasificador SVM entrega una salida binaria (pertenece a una (1) u otra clase (-1)) no permite una aproximación directa de edad, sin embargo, a priori entrega resultados más refinados, ya que es posible ver el comportamiento directo entre clases, e incluso permite un entrenamiento diferenciado para cada sub problema.

En cuanto a los parámetros utilizados por el SVM, dentro de las opciones presentadas por la implementación en OpenCV, se utilizaron los siguientes:

- a) SVM para clasificación: C-SVM
- b) Kernel: función de base radial (RBF)
- c) Error objetivo: 0.01

En cuanto al resto de los parámetros para utilizar el clasificador de forma expedita, se utilizó una opción de entrenamiento automática, la cual hace una validación cruzada de 6 grupos, y determina respectivamente los valores óptimos de las variables de entrenamiento C , γ , p , nu , $coef0$ y $degree$, vistas en la sección 2.6.2.

Por lo tanto, una vez obtenidos y normalizados los valores de las respectivas características viene la fase clave del trabajo, la clasificación. Para este objetivo, y tal como se detalló en el capítulo anterior se aplicaran los 2 clasificadores mencionados, pues cada uno presenta sus ventajas comparativas. Así una vez estudiados los resultados para cada tipo de clasificador, en las diversas modalidades que ofrecen (es decir, para 3, 4 o 5 clases según sea posible), se podrá determinar el tratamiento óptimo para el problema de clasificación, e incluso implementar una solución que combine las capacidades de ambos.

3.4. Optimización

Inicialmente se estableció la edad límite entre clases de forma que cada una poseyera una cantidad de individuos equiparada, con tal de optimizar tan solo los parámetros de cada uno de los filtros a utilizar y además los parámetros que modelan el comportamiento de los clasificadores. Así, el objetivo prioritario era obtener una buena tasa de clasificación global (del orden del 85%) y como objetivo secundario se encontraba la aproximación de la edad cuyo error absoluto se esperaba que fuera menor a 10 años como promedio. Por lo tanto, el primer set de parámetros a optimizar consistía en 11 parámetros para los 4 filtros a aplicar, más otros 3 para la red neuronal:

- a) Canny: 2 umbrales y el tamaño del filtro.
- b) Gaussiano pasa altos: el umbral y el tamaño.
- c) Diferencia de Gaussianos: el par de umbrales más el tamaño.
- d) Sobel: el nivel horizontal, vertical y el tamaño.
- e) Red Neuronal (cuando corresponda): Número de neuronas ocultas, paso y momentum.

Por otro lado, la manera en que se efectuó la correspondiente elección de parámetros óptimos tuvo 2 variedades, una hecha con algoritmo genético puro y otra con la versión modificada como es PSO, en primera instancia optimizando los parámetros de los filtros y clasificadores respectivos para cada implementación. Finalmente por el desempeño se mantuvo el uso de la técnica PSO. Así, se efectuó la optimización de parámetros persiguiendo un total de 6 objetivos distintos, 3 por cada tipo de clasificador:

- Red Neuronal: Minimizar el error medio en años (MAE por sus siglas en inglés), para un clasificador de 3, 4 y 5 clases (3 optimizaciones).
- SVM: Maximizar la tasa de aciertos para 3 clases, el primer resultado parcial ($[C_1$ y $C_2]$ vs $[C_3$ y $C_4]$) y el total de aciertos para 4 clases (3 optimizaciones).

En cuanto a la variante de PSO utilizada se trató de una denominada PSO común con inercia, que formaba parte de un toolbox en el cual se utilizaron los siguientes parámetros de configuración por defecto:

- | | |
|---|-------|
| a) Constante de aceleración local: | 2 |
| b) Constante de aceleración global: | 2 |
| c) Total de partículas: | 50 |
| d) Peso inicial de inercia: | 0.9 |
| e) Peso final de inercia: | 0.4 |
| f) Mínimo global del gradiente del error: | 1e-25 |

4. Resultados

4.1. Primeros resultados

En primera instancia se exploró el problema utilizando tan solo la red neuronal y tratando de seleccionar los parámetros correspondientes sin optimización, lo cual dado el gran número de combinaciones posibles no condujo a buenos resultados, en el general de la clasificación se llegó apenas a un **65%** de clasificación de positivos verdaderos.

Luego, se procedió a implementar la red neuronal optimizada, lo cual llevó a un resultado general de un **72.93%** de clasificación correcta, el detalle se muestra en la siguiente tabla de confusión.

Tabla 4. Red Neuronal para 4 clases

%	Edades	Clase 1	Clase 2	Clase 3	Clase 4
Entrenamiento	0 - 16	95.37	0.95	0.83	0.28
	17 - 30	0.00	100.00	0.00	0.00
	31 - 50	0.00	1.06	98.40	0.53
	51 - 70	0.00	0.24	0.81	99.15
Validación	0 - 16	75.65	24.35	0.00	0.00
	17 - 30	8.42	85.13	6.45	0.00
	31 - 50	0.00	7.89	86.71	5.39
	51 - 70	0.16	8.03	53.28	38.52

Por otro lado, utilizando la cascada de clasificadores SVM, se llegó a un **77.22%** de clasificación global para el mismo conjunto de entrenamiento, lo cual dicho sea de paso, confirmaba el hecho de que SVM resulta más eficiente para problemas de clasificación.

Tabla 5. SVM para 4 clases.

%	Edades	Clase 1	Clase 2	Clase 3	Clase 4
Entrenamiento	0 - 16	88.66	11.34	0.00	0.00
	17 - 30	1.60	98.40	0.00	0.00
	31 - 50	2.25	6.74	73.03	17.98
	51 - 70	0.00	0.00	38.46	61.54
Validación	0 - 16	86.67	11.11	2.22	0.00
	17 - 30	3.90	93.50	2.60	0.00
	31 - 50	3.95	5.26	71.05	19.74
	51 - 70	0.00	6.56	36.07	57.38

Sin embargo, aún ninguno de los resultados obtenidos resultaba contundente, estando aún cerca de un 10% por debajo de la tasa objetivo de clasificación global, por lo cual fue necesario replantear la manera en que se enfrentaba el problema, planteando alternativas específicas para cada método de clasificación.

Por un lado, dado que la intención de utilizar una red neuronal viene por la posibilidad de entregar una aproximación numérica para la edad del individuo, el objetivo para este método fue pasar de un neto clasificador a un estimador de edad y minimizar el error medio de la edad calculada, para luego efectuar una clasificación utilizando una división de edades que resultara más efectiva, es decir, no limitarse necesariamente a trabajar con 4 clases, ya sea para la etapa de entrenamiento, como para la clasificación final.

En cuanto al SVM, se apuntaba a generar los óptimos para cada uno de los clasificadores por separado, aumentar su tasa de acierto y luego combinarlos con tal de potenciar la clasificación global de los individuos. Además, como se puede apreciar de la tabla también se da la posibilidad de generar un clasificador de 2 clases, que también puede resultar útil dependiendo del problema puntual a resolver, por lo cual también se exploró dicha alternativa.

4.2. Optimización de resultados para la red neuronal

El trabajo consistió en encontrar los parámetros óptimos para 3 particiones diferentes (3, 4 y 5 clases etarias), teniendo como objetivo alcanzar el mínimo error posible para la estimación de la edad del individuo en años (MAE). Para estas pruebas se optimizaron solamente los parámetros de los filtros y la red neuronal, con lo cual se observa el siguiente desempeño:

Tabla 6. Resultados preliminares MAE.

		MAE
3 Clases	Entrenamiento	4.96
	Validación	5.99
	Test	6.26
4 Clases	Entrenamiento	4.33
	Validación	6.58
	Test	6.65
5 Clases	Entrenamiento	3.04
	Validación	6.67
	Test	6.82

Como es posible apreciar, los resultados más equilibrados vienen dados por el clasificador entrenado con 4 clases, sin embargo, se encontró un problema a la hora de clasificar, en donde no era posible superar el 75% de acierto para los individuos, por lo cual se optó por utilizar una máscara de clasificación distinta, con tal de asegurar una alta tasa de aciertos, así es como después de muchas pruebas se determinó que la partición óptima para la etapa de clasificación consistía en 3 clases, y así ya los resultados superaron el umbral del 80% por clases, como se refleja a continuación, para los mismos resultados, pero en tasa de aciertos por clase:

Tabla 7. Matriz de confusión para 3 clases de entrenamiento.

	Edades	Clase 1	Clase 2	Clase 3
Entrenamiento	0 - 16	76.23	22.98	0.79
	17 - 35	12.11	82.59	5.31
	36 - 70	0.68	3.57	95.75
Validación	0 - 16	66.67	30.77	2.56
	17 - 35	12.97	81.76	5.27
	36 - 70	0.45	6.54	93.01
Test	0 - 16	66.83	31.46	1.70
	17 - 35	10.22	85.54	4.23
	36 - 70	0.62	6.77	92.60

Tabla 8. Matriz de confusión para 4 clases de entrenamiento.

	Edades	Clase 1	Clase 2	Clase 3
Entrenamiento	0 - 16	91.87	8.13	0.00
	17 - 35	12.14	87.70	0.16
	36 - 70	1.09	1.19	97.72
Validación	0 - 16	88.70	9.13	2.17391
	17 - 35	17.07	77.56	5.37
	36 - 70	1.45	2.44	96.11
Test	0 - 16	83.22	11.78	0.00
	17 - 35	16.34	78.17	5.49
	36 - 70	1.22	2.82	95.95

Tabla 9. Matriz de confusión para 5 clases de entrenamiento.

	Edades	Clase 1	Clase 2	Clase 3
Entrenamiento	0 - 16	79.82	20.18	0.00
	17 - 35	3.64	96.35	0.00
	36 - 70	0.27	4.04	95.68
Validación	0 - 16	72.44	24.88	2.68
	17 - 35	7.50	88.91	3.59
	36 - 70	0.79	8.66	90.55
Test	0 - 16	70.43	27.17	2.39
	17 - 35	10.12	84.15	5.73
	36 - 70	0.76	3.28	95.95

Con estos resultados el último paso consistía en optimizar los límites de corte de los grupos etarios, para lo cual se utilizó la solución previa encontrada como semilla para el algoritmo PSO, dejando como fijos los parámetros de los filtros y la red se optimizando las edades de corte, todo esto bajo la premisa de obtener la mayor tasa de clasificación por clase, con lo cual se pueden ver los siguientes resultados:

Tabla 10. Optimización de límites de grupos etarios para 3 clases de entrenamiento.

MAE	5.29	Clase 1	Clase 2	Clase 3
% Aciertos	94.22			
Entrenamiento	0 - 16	84.21	14.91	0.88
	17 - 35	5.31	91.90	2.79
	36 - 70	0.73	2.48	96.79
MAE	6.55	Clase 1	Clase 2	Clase 3
% Aciertos	89.55			
Validación	0 - 16	81.54	15.90	2.56
	17 - 35	9.32	83.24	7.43
	36 - 70	0.69	6.59	92.72
MAE	6.69	Clase 1	Clase 2	Clase 3
% Aciertos	87.53			
Test	0 - 16	86.09	11.52	2.39
	17 - 35	16.83	74.51	8.66
	36 - 70	0.08	3.74	96.18

Tabla 11. Optimización de límites de grupos etarios para 4 clases de entrenamiento.

MAE	4.70	Clase 1	Clase 2	Clase 3
% Aciertos	93.02			
Entrenamiento	0 - 16	87.85	12.15	0.00
	17 - 35	10.32	89.68	0.00
	36 - 70	0.99	2.21	96.79
MAE	6.78	Clase 1	Clase 2	Clase 3
% Aciertos	88.57			
Validación	0 - 16	84.35	12.83	2.83
	17 - 35	15.61	79.39	5.00
	36 - 70	0.99	3.21	95.80
MAE	6.21	Clase 1	Clase 2	Clase 3
% Aciertos	88.34			
Test	0 - 16	72.17	26.74	1.09
	17 - 35	12.32	84.88	2.80
	36 - 70	0.61	3.21	96.18

Tabla 12. Optimización de límites de grupos etarios para 5 clases de entrenamiento.

MAE	3.46	Clase 1	Clase 2	Clase 3
% Aciertos	94.27			
Entrenamiento	0 - 16	79.02	20.98	0.00
	17 - 35	1.86	96.71	1.43
	36 - 70	0.42	1.12	98.46
MAE	6.02	Clase 1	Clase 2	Clase 3
% Aciertos	89.42			
Validación	0 - 16	73.17	24.88	1.95
	17 - 36	6.29	89.38	4.33
	37 - 70	0.33	4.75	94.92
MAE	6.02	Clase 1	Clase 2	Clase 3
% Aciertos	87.84			
Test	0 - 16	67.32	31.46	1.22
	17 - 35	7.07	89.78	3.15
	36 - 70	0.31	6.61	93.07

4.3. Optimización de resultados para SVM

Como ya se vio anteriormente, el resultado obtenido para 4 clases es inferior a los objetivos planteados, por lo cual, se desechó esta opción focalizando el trabajo en optimizar 2 tipos de clasificadores que podrían resultar útiles, uno de 3 clases, y otro de 2 clases (simplemente porque se descubrió que ofrecía un muy alto rendimiento). Así, persiguiendo como único objetivo incrementar la tasa de aciertos, es que se optimizaron los parámetros de los filtros para encontrar la tasa de clasificación óptima, encontrando los siguientes resultados:

Tabla 13. Resultados preliminares para clasificador SVM de 2 clases.

	Edades	Clase 1	Clase 2
Entrenamiento	0 - 30	100.00	0.00
	31 - 70	4.49	95.51
Validación	0 - 30	97.59	2.41
	31 - 70	7.88	92.11
Test	0 - 30	98.46	1.54
	31 - 70	3.81	96.19

Tabla 14. Resultados Preliminares para clasificador SVM de 3 clases.

	Edades	Clase 1	Clase 2	Clase 3
Entrenamiento	0 - 16	85.71	13.19	1.10
	17 - 35	4.32	93.51	2.16
	36 - 70	0.36	4.33	95.30
Validación	0 - 16	78.57	21.43	0.00
	17 - 35	6.35	93.65	0.00
	36 - 70	0.00	3.53	96.47
Test	0 - 16	79.41	20.59	0.00
	17 - 35	5.08	89.83	5.08
	36 - 70	0.00	5.34	95.42

Por otro lado, con tal de mejorar aún más los resultados, se optimizaron los límites inter-clases, siguiendo el procedimiento anterior, con lo cual se lograron los siguientes desempeños:

Tabla 15. Resultados finales clasificador SVM para 2 clases.

	Edades	Clase 1	Clase 2
Entrenamiento	0 - 35	99.19	0.81
	36 - 70	3.61	96.39
Validación	0 - 35	99.21	0.79
	36 - 70	3.53	96.47
Test	0 - 35	98.31	1.69
	36 - 70	4.58	96.18

Tabla 16. Resultados finales clasificador SVM para 3 clases.

	Edades	Clase 1	Clase 2	Clase 3
Entrenamiento	0 - 16	85.09	14.91	0.00
	17 - 35	9.52	89.12	1.36
	36 - 70	1.73	1.91	96.36
Validación	0 - 16	84.62	12.82	2.56
	17 - 35	21.62	75.68	2.70
	36 - 70	0.41	6.91	92.68
Test	0 - 16	85.09	13.16	1.75
	17 - 35	8.16	90.48	1.36
	36 - 70	1.04	2.60	96.36

Como se puede apreciar, el rendimiento como clasificador neto es superior, razón por la cual además se ha incluido el resultado final de test; pero el clasificador no es capaz de entregar información con respecto a la edad exacta del individuo, lo cual le resta flexibilidad.

Finalmente, se obtuvieron resultados de un 96.44% de aciertos por individuo en caso de utilizar un clasificador de 2 grupos etarios (con un promedio de 97.24% por clases en promedio). En el caso del clasificador para 3 grupos, se obtuvo una clasificación global de un 91.66% (promediando un 90.64% por clase)

4.4. Análisis de resultados

Dados los resultados obtenidos en primera instancia, fue necesario cambiar la estrategia planteada inicialmente, optando por acotar el problema a una clasificación por 3 clases, ya que así se alcanzaba una mayor tasa de clasificación correcta por individuo.

En cuanto a los límites inter-clases utilizados, se observa que son más detallados mientras menor es el individuo, esto es así ya que a edades más tempranas es más fácil diferenciar a las personas por su edad, tarea que conforme pasa el tiempo se vuelve más complicada dado que los cambios ocurren de manera muy diferenciada, ya sea por efectos de la genética del individuo o sus experiencias de vida. Todo esto se vuelve relevante al considerar la principal aplicación del método como herramienta para hacer estudios de marketing, pues tiene directa relación con los estilos de vida de los consumidores.

En el caso de utilizar un clasificador en base a una red neuronal, el proceso de optimización resultó bastante más engorroso debido a que como clasificador las redes desempeñaron un trabajo pobre para este problema en específico, sin embargo, fue posible explotarlas por el tipo de salidas que ofrecían, puesto que era posible entregar una aproximación numérica de la edad para el individuo en cuestión, lo cual les da una funcionalidad extra como estimador de edad. Así en caso de requerir este tipo de clasificador, quizás su desempeño no sea el óptimo, pero se tendrá una información más detallada sobre la edad de la persona, lo cual a pesar de no ser un objetivo principal es un resultado secundario que podría resultar útil.

Como se observó en el comportamiento del SVM, su relevancia se hace evidente al mostrar excelentes resultados para una clasificación gruesa de edad en 2 grupos, lo cual lo perfila como el mejor algoritmo encontrado para brindar una solución parcial al problema, sin embargo, cuando se quiso evaluar para un mayor número de clases su desempeño fue desigual, ya que por un lado, al tener más clasificadores específicos era difícil sintonizarlos para que rindieran al mismo tiempo de manera óptima, por esto el experimento de clasificación de 4 clases no fue exitoso, aunque el de 3 clases tuvo un desempeño algo más aceptable.

Por estas razones es que lo que se logró configurar es un set de clasificadores optimizados según la tarea específica que se requiera ejecutar, si es una clasificación grosso modo de la edad del individuo, se optará por un clasificador SVM; de lo contrario, lo más aconsejable es perfeccionar el método utilizando redes neuronales.

5. Conclusiones

Como se ha señalado, el problema de clasificar por grupo etario a un individuo es una labor no menor en el campo de la visión computacional y que hoy en día se ha transformado en una gran fuente de investigaciones, esto último dado el auge de los sistemas artificiales de interacción humana, con múltiples propósitos: ya sea desde el marketing selectivo hasta tareas de identificación de individuos. Por esta razón, existe una amplia variedad de acercamientos experimentales alguna solución al problema, cada cual con sus ventajas y desventajas, como su nivel de complejidad y fundamentos teóricos de respaldo.

Para el presente proyecto se ha optado por una alternativa que se basa en estudios de antropometría, los que concluyen que la forma del rostro humano va cambiando su forma durante los primeros años de edad (hasta mediados de la adolescencia), pasando desde una forma aproximadamente circular para terminar con una forma ovalada. Por otro lado, para edades más avanzadas de edad el análisis recae sobre el conocido efecto de la vejez en la piel humana, incorporándose al rostro las respectivas arrugas que se van intensificando a medida que sigue el paso del tiempo, además, éstas suelen mantener ciertas zonas específicas con mayor presencia, como: la frente, la zona del párpado inferior de los ojos, etc.

Entonces, con tal de obtener dicha información sobre la edad del individuo es que se recopilan datos sobre algunas proporciones geométricas del rostro como la asociada en la distancia entre los ojos versus la existente desde ese punto medio hasta la nariz, y otras que se detallan en el capítulo correspondiente; y por otro lado, mediante una serie de filtros, se recopila información sobre la presencia e intensidad de las arrugas en determinadas zonas del rostro.

Con dicha información se confecciona un vector de características, el cual sirve de base para el aprendizaje computacional sobre el proceso de envejecimiento humano, todo esto con el objetivo de entregar una aproximación automática al grupo etario al cual pertenece un determinado individuo. Las componentes de este vector, consistían básicamente en 4 tipos de datos, por un lado estaban las componentes referidas a las proporciones faciales, y otros 3

indicadores de: presencia, distribución e intensidad de arrugas en 5 zonas predeterminadas del rostro, estas son: en la zona de la frente, el área lateral externa y la inferior de ambos ojos.

Luego de obtener los valores clave del rostro del individuo, se requiere un algoritmo que sea capaz de establecer patrones matemáticos sobre la relación entre estos valores y la edad de la persona. Para efectuar esta tarea se presentaron 2 métodos: uno basado en redes neuronales, y otro denominado SVM (o máquina de soporte vectorial), dado que cada uno ofrece sus ventajas comparativas.

Por un lado, se tiene que las redes neuronales poseen salidas con alta variabilidad, las que se pueden utilizar para entregar información más detallada que solamente el grupo de pertenencia de la persona, como por ejemplo ir un paso más allá y aproximar la edad exacta. Esto se hace posible por la manera en que se le pueden ingresar las etiquetas de edad objetivo, las que presentan una cantidad de valores reales idéntico al número de clases con el cual se está trabajando, es decir, se puede codificar la edad del individuo de manera única en función de la edad real; de esta manera el proceso de entrenamiento cuenta con información detallada del individuo, lo cual conforma un enfoque distinto para el aprendizaje computacional. Así es como se estructuró una solución alternativa al problema de clasificación neta, pasando por una estimación de edad, para luego con esa estimación agregar una pequeña etapa de post-procesamiento de dicho valor y entregar una clasificación final.

En la otra mano, se tiene que un clasificador del tipo SVM presenta un desempeño superior en este caso a su par neuronal. Sin embargo, dada su naturaleza binaria tanto como para la manera de representar las etiquetas, como la salida misma del clasificador, es que no ofrecía gran flexibilidad a la hora de la cantidad de clases con las cuales trabajar, optando en última instancia por tener como máximo 3 grupos etarios para la clasificación.

Así es que para ambos clasificadores se fijó el máximo en 3 clases de edad, principalmente por 2 razones: la primera es que dadas las condiciones del método, es decir, la información recopilada del rostro detallada anteriormente, sus desempeños se veía maximizados con dicha cantidad de clases; y la segunda tiene que ver con el hecho de que dada la complejidad de la tarea abordada incluso para un ser humano resulta difícil determinar la

edad de otro individuo, por lo tanto, para la mayoría de los fines que podría tener este método, 3 grupos etarios resulta suficiente, los que se pueden identificar como niño (0 a 15 años), joven (16 a 30 años) y adulto (31 o más años).

Con esto en mente, es que se obtuvo por un lado un clasificador capaz de entregar en promedio un error mínimo de 6.02 años de edad para los sujetos, y que por otro lado permitía una clasificación en 3 grupos con una tasa de aciertos de un 87.84%. Y si lo que se busca es simplemente precisión, también se generó un clasificador cuya tasa de aciertos era de un 96.44% para 2 clases de edad, o si se prefería algo más de detalle, un 90.64% de acierto en la clasificación considerando 3 clases.

Ahora, uno de los principales problemas al momento de clasificar por edad son los casos en que el individuo posee una cercana a los límites inter-clases determinados para los diferentes grupos, ya que esos traspasos entre una clase y otra son continuos y no hay gran diferencia, por ejemplo, entre una persona con 35 años y otra con 36 (las que pueden pertenecer a clases diferentes), luego, esta es una clara fuente de error para el clasificador, lo cual podría ser evitado si es que se incluyera además un grado de pertenencia a las clases, lo cual con la implementación utilizada de SVM resultaba imposible de aplicar y requiere un trabajo extra para modificar el algoritmo de aprendizaje computacional.

Otro gran problema que afecta también a la estimación de edad, viene dado por las características del proceso de envejecimiento de cada ser humano, que si bien es el mismo a grandes rasgos, depende mucho de la genética del individuo y de su propio pasar por la vida, transformando el proceso de envejecimiento en algo que puede llegar a ser tan particular como la propia identidad, como ejemplo de esto se puede considerar la aparición de enfermedades en la piel que puede determinar la manera en que esta irá cambiando a través del tiempo, asimismo dependiendo de la expresividad del rostro de cada individuo en particular los surcos que en ella se formen podrán cambiar su forma o intensidad, o simplemente la raza a la que pertenezca un individuo determinará algunas características predominantes en el proceso de envejecimiento.

Por lo tanto, al momento de intentar identificar el grupo etario al cual pertenece un individuo, inmediatamente se está delimitando y estableciendo un parámetro de generalización para un proceso que resulta único para cada ser humano y que depende de una infinidad de factores que pueden llegar a ser muy aleatorios, por lo tanto, el problema resulta muy complejo y de momento las soluciones encontradas en el presente trabajo a pesar de no resultar impresionantes en cuanto al nivel de aciertos o medidas de error, resultan comparables a otros trabajos actuales. [17]

El presente desarrollo presenta entonces una alternativa para desarrollar la tarea de clasificación gruesa de edad, lo cual resulta muy útil a pesar de lo tosco que pueda parecer a primera vista, teniendo en consideración aplicaciones que requieran tener una interacción con la persona, ofreciendo servicios acorde a su grupo etario, por ejemplo, entregar asistencia específica para niños o adultos mayores en determinadas emergencias, entregar información sobre productos relevantes para un adulto, efectuar estudios de comportamiento, etc. Por otro lado, no está de más destacar el hecho de que el presente método es totalmente automático y presenta un tiempo de respuesta por imagen que no supera la centésima de segundo (con imágenes de 200x200 píxeles, un procesador de doble núcleo a 2.66GHz con la versión 2.3 de OpenCV), lo cual lo hace muy versátil para aplicaciones que requieran una rápida respuesta, como lo es el procesamiento en tiempo real a través de la captura de video.

Finalmente, es importante mencionar que se ha sentado una base para futuros trabajos relacionados, por ejemplo, es posible idear nuevas maneras de codificar la edad en función de las clases etarias que se deseen para que el método de clasificación con redes neuronales resulte aún más preciso; asimismo, se tiene toda la fase de clasificación ya desarrollada, lo que da lugar a que si se cambian los vectores y la respectiva información extraída del rostro se daría lugar a nuevas variaciones de la solución al problema de estimación de edad. Como se entiende, la base de trabajo es muy flexible como para introducir nuevas metodologías y mejoras al actual desarrollo, por lo que los alcances no se ven limitados a los resultados obtenidos durante esta primera fase de un trabajo que aún puede seguir evolucionando.

5.1. Alcances y trabajos futuros

Tal como se ha comentado, el presente método tiene como finalidad el ayudar a objetivos de marketing selectivo, en conjunto con un software de identificación de rostros, con lo cual el proceso de entrega de información al usuario podrá ser cada vez más personalizado, esto complementado con datos del género por ejemplo, podrá derivar en aplicaciones cada vez más amigables y adaptadas al sujeto en cuestión.

Por otro lado, futuras investigaciones pueden tener como objetivo perfeccionar el actual método de clasificación de edad, o simplemente explorar nuevas ideas y fundamentos para lograr un sistema aún más preciso. Una de estas mejoras podría ser el desarrollo de un método que sea independiente de la pose del individuo frente a la cámara, lo que haría que el sistema en general fuera más robusto, sin embargo, como la complejidad aumenta, se requerirían al mismo tiempo métodos más robustos para la detección del rostro y sus características principales, de otro modo se limitan mucho las opciones de investigación

También sería interesante contar con bases de datos de personas de diferentes razas y poblaciones con tal de tener una información más generalizada del proceso de envejecimiento del ser humano bajo diversas condiciones genéticas, geográficas o culturales, ya que el proceso de envejecimiento está muy ligado a factores de este tipo.

Por último, podrían utilizarse algoritmos de aprendizaje no supervisado que dieran cuenta de los cambios que ocurren en el rostro y que no son percibidos por el ojo humano, extrayendo mayor información respecto al proceso de envejecimiento, asimismo, determinar con mayor precisión las zonas del rostro donde se puede encontrar mayor información y también utilizar otro tipo de filtros o técnicas que puedan refinar los datos que se extraen del rostro humano.

6. Referencias

- [1] Mohammad Mahdi Dehshibi, Azam Bastanfard: A new algorithm for age recognition from facial images. *Signal Processing* 90 (2010) 2431–2444.
- [2] Ramesha K, K B Raja, Venugopal K R, L M Patnaik: Feature Extraction based Face Recognition, Gender and Age Classification. (*IJCSE*) *International Journal on Computer Science and Engineering*, Vol. 02, No.01S, 2010, 14-23.
- [3] M. Chandra Moha, V. Vijaya Kumar, V. Venkata Krishna: Novel Method of Adult Age Classification Using Linear Wavelet Transforms. *IJCSNS International Journal of Computer Science and Network Security*, VOL.10 No.3, March 2010.
- [4] Nabil Hewahi, Aya Olwan, Nebal Tubeel, Salha EL-Asar, Zeinab Abu-Sultan: Age Estimation based on Neural Networks using Face Features. *Journal of Emerging Trends in Computing and Information Sciences*. E-ISSN 2218-6301.
- [5] John A. Ruiz-Hernandez, James L. Crowley, Augustin Lux: "How old are you?": Age Estimation with Tensors of Binary Gaussian Receptive Maps. INRIA Grenoble Rhones-Alpes Research Center and Laboratoire d'Informatique de Grenoble (LIG).
- [6] Peng Yang, Lin Zhong, Dimitris Metaxas: Ranking Model for Facial Age Estimation. 2010 *International Conference on Pattern Recognition*.
- [7] Feng Gao, Haizhou Ai: Face Age Classification on Consumer Images with Gabor Feature and Fuzzy LDA Method.
- [8] Asuman GÜNAY, Vasif V. NABIYEV: Automatic Age Classification with LBP.
- [9] L.G. Farkas, *Anthropometry of the Head and Face*, Raven Press, New York, 1994.
- [10] V. Vapnik. *Estimation of dependences based on empirical data*. SpringerVerlag, New York, 1982.

- [11] V. Vapnik. The nature of statistical learning theory. Springer–Verlag, New York, 1995.
- [12] Holland, J., Adaptation in Natural and Artificial Systems, MIT Press, 1995.
- [13] J. Kennedy, R. Eberhart, and Y. Shi. Swarm Intelligence. San Francisco: Morgan Kaufmann Publishers, 2001.
- [14] Sung Eun Choi, Youn Joo Lee, Sung Joo Lee, Kang Ryoung Park, and Jaihie Kim. Kernels. A Comparative Study of Local Feature Extraction for Age Estimation. 2010 11th Int. Conf. Control, Automation, Robotics and Vision, Singapore, 7-10th December 2010.
- [15] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, 5th Annual ACM Workshop on COLT, pages 144–152, Pittsburgh, PA, 1992. ACM Press.
- [16] Tara T. Lineweaver, Paul Hutman, Christopher Ketcham, John Neil Bohannon III. The Effect of Comprehension Feedback and Listener Age on Speech Complexity. Journal of Language and Social Psychology. March 2011.
- [17] Y. Fu, G. Guo, and T.S. Huang, "Age Synthesis and Estimation via Faces: A Survey", presented at IEEE Trans. Pattern Anal. Mach. Intell., 2010, pp.1955-1976.