Universidad de Granada

Sistemas Críticos

Sistema crítico para el reconocimiento facial de terroristas

Curso 2019-2020

Realizado por:

Francisco Solano López Rodríguez
20100444P fransol0728@correo.ugr.es

Máster universitario en Ingeniería Informática

Índice

1. Introducción

El Reconocimiento Facial es un campo que ha sido muy estudiado durante años y el cual ha tomado más importancia en la actualidad gracias al avance de la tecnología y a la gran cantidad de aplicaciones que este tiene en numerosos ámbitos. En particular una de sus aplicaciones se encuentra en el reconocimiento facial de terroristas en tiempo real en cámaras de vigilancia. Este tipo de aplicaciones se trata de sistemas críticos, pues un fallo en este sistema podría provocar graves consecuencias como por ejemplo no detectar a tiempo la presencia un grupo de terrorista y con ello permitir que este cometiera un atentado o por otro lado que una persona inocente fuese identificada como terrorista y ello provocará alguna reacción violenta contra dicha persona por parte de un miembro de seguridad. Es por ello que la confiabilidad es una propiedad muy importante en este tipo de sistemas, no solo en el sentido de que este no cometa errores, sino que por ejemplo este sistema esté activo y en funcionamiento en todo momento.

1.1. Motivación

El motivo de este trabajo es el de aplicar las técnicas de visión por computador para conseguir un sistema de detección de terroristas que sea fiable, en el sentido del número de fallos que comete en el reconocimiento y que además sea lo suficientemente rápido como para poder ejecutarse en una cámara de seguridad en tiempo real.

Este tipo de sistemas son enormemente complejos y en los cuales el primer problema que se nos presenta, no es el de reconocimiento en si mismo, sino el de detectar los rostros que hay sobre la imagen. Una vez detectados los rostros estos pasan al sistema de reconocimiento facial, en el cual debemos de identificar dichos rostros, lo cual presenta numerosos desafíos. Entre algunos de estos desafíos tenemos el de hacer frente a la variación de luz, no solo por la cantidad de luz, sino que esta también puede venir de diferentes direcciones. Además los rostros pueden presentar distintas perspectivas, estar girados, con diferentes expresiones faciales y para colmo el paso del tiempo nos cambia también la cara debido al envejecimiento. Todo esto hace que implementar un sistema de reconocimiento que presente un grado alto de precisión sea extremadamente complicado, ya que los factores a tener en cuenta son muy elevados.

1.2. Objetivos

El objetivo principal de este proyecto va a ser el de implementar un método que combine el uso de varios algoritmos y con ello aumente la robustez del sistema y se consiga aumentar la tasa de acierto. Para ello se propone un método que hará uso de 3 algoritmos de reconocimiento diferentes y la respuesta que dará el sistema ante

la entrada de una imagen facial, vendrá dada por el voto de la mayoría.

Otro de los objetivos será el de aumentar la velocidad de dicho reconocimiento, pues el sistema está planteado para ser utilizado en tiempo real, por lo que el tiempo de respuesta será crucial. Para ello se propone ejecutar los 3 algoritmos de reconocimiento de forma paralela y con ello obtener una notable mejora en el tiempo de ejecución.

Como ya se ha comentado, el número de factores a tener en cuenta para el reconocimiento es muy alto, por ello para simplificar el problema vamos a centrarnos en el caso de que todos los rostros están de frente y además alineados horizontalmente, es decir los rostros no están girados. Respecto a la detección de rostros se va a usar un solo algoritmo, pues esta no es la parte que se pretende investigar, así nos centraremos principalmente en la parte de reconocimiento, es decir la identificación de los rostros.

Además de los objetivos ya explicados, se intentará también que el sistema funcione sobre un conjunto abierto. Muchos sistemas de reconocimiento son implementados para funcionar en un entorno cerrado, en otras palabras, estos sistemas son diseñados de forma que se da por hecho que las imágenes de prueba que se van a evaluar pertenecen a alguna de las identidades que se encuentra en la base de datos. El problema es que en la realidad esto no ocurre, ningún sistema tiene una foto de todas las personas existentes en el mundo, por lo tanto sería necesario aplicar algún método para identificar a esas personas desconocidas.

2. Trabajo desarrollado

2.1. Herramientas utilizadas

El lenguaje de programación utilizado ha sido C++, el cual ha sido ampliamente utilizado en sistemas críticos de seguridad y tiempo real. Uno de los factores clave de elegir este lenguaje para sistemas en tiempo real, sin duda es la eficiencia que ofrece, además de permitir la programación a bajo nivel necesaria para el acceso al hardware o el código de alto rendimiento.

La biblioteca utilizada para las tareas de visión por computador ha sido OpenCV, una biblioteca open source orientada a la visión artificial originalmente desarrollada por Intel. Esta biblioteca dispone de un gran número de funcionalidades que nos van a permitir preprocesar imágenes, detectar rostros y realizar el proceso de identificación facial. A nivel computacional es muy eficiente, lo que la hace idónea para aplicaciones en tiempo real, precisamente como es el caso de este proyecto.

También se ha recurrido a la API de programación paralela OpenMP, la cual nos va a permitir ejecutar los 3 algoritmos de reconocimiento de forma concurrente.

Veamos las características del equipo utilizado para la evaluación de los modelos de reconocimiento facial.

- Sistema Operativo: Ubuntu 18.04.4 LTS
- Memoria RAM: 15,4 GiB
- Procesador: Intel® Core™ i5-4200U CPU @ 1.60GHz 4
- Disco duro: SSD 256 GB

2.2. Conjunto de datos utilizado

Debido a que no encontré ninguna base de datos de calidad que contuviera rostros de terroristas, finalmente opté por hacer uso de una base de datos genérica, que contiene imágenes de rostros de distintos sujetos. Esta base de datos se llama Yale (puede consultarse en la bibliografía) y contiene múltiples fotos del rostro de frente de cada sujeto, con distintas variaciones de luz y de expresiones faciales. Tan solo se han utilizado dos de los sujetos de esta base de datos, los cuales serán los supuestos terroristas. De cada sujeto se han utilizado 40 imágenes para entrenar los modelos de reconocimiento facial y 15 imágenes para validación. Además se ha obtenido otro conjunto de rostros pertenecientes a personas famosas, que se ha utilizado para la identificación de personas no registradas en la base de datos, es decir cuyo rostro no coincide con ninguno de los dos rostros utilizados para entrenar y por lo tanto estos rostros no serán identificados como supuestos terroristas. Todas las imágenes utilizadas, tanto las de la base de datos de Yale, como las fotos de celebridades, ya

venían debidamente cortadas, de modo que solo contiene el rostro de frente de los individuos, además todas han sido reescaladas a un tamaño de 128x128.

En la imagen inferior se muestran algunas imágenes del conjunto Yale, que como podemos apreciar ya se encuentran recortadas, de forma que queda enmarcado el rostro frontal de la persona.

Figura 1: Rostros de la base de datos Yale

2.3. Modelos de entrenamiento utilizados

Los algoritmos de reconocimiento utilizados han sido los siguientes:

- **Fisherfaces**: es un método que se encarga del reconocimiento de caras, teniendo en cuenta como se refleja la luz y las expresiones faciales. Este algoritmo maximiza la relación entre la distribución de las clases y la distribución intraclases. Fisherface clasifica y reduce la dimensión de las caras utilizando el método Discriminante Lineal de Fisher(FLD) y PCA (conocido como Eigenfaces). Este método crea una proyección lineal que maximiza las diferentes imágenes de caras proyectadas.
- **Local binary patterns**: es un algoritmo que ha mostrado muy buenos resultados como descriptor de texturas en escala de grises. Es muy eficiente y además presenta robustez con respecto a la expresión facial y posición de la cabeza, debido a que está basado en características locales del rostro.
- **Eigenfaces**: es uno de los algoritmos más populares y tiene como base el Análisis de Componentes Principales. El método es muy eficiente en tiempo de procesamiento y almacenamiento, debido a que PCA reduce considerablemente el tamaño de la dimensionalidad.

Los 3 algoritmos descritos están implementados en OpenCV. Todos ellos nos permiten entrenar un modelo de predicción, que posteriormente podremos usar para identificar rostros. Además, no solo devuelven la predicción realizada, sino que también devuelven un valor de confiabilidad de dicha predicción, el cual usaremos

para identificar rostros desconocidos (no terroristas). El valor de confiabilidad se interpreta al contrario de lo que uno podría pensar, cuanto mayor sea el valor, peor será la predicción realizada. Esto se debe a que el cálculo de la confiabilidad, se hace en base a la distancia euclídea que existe entre la imagen a evaluar y las imágenes de los sujetos utilizados para el entrenamiento. Así cuanto mayor sea el valor de la confiabilidad, mayor será la distancia y por tanto peor será la predicción.

2.4. Arquitectura del sistema

Teniendo los modelos de reconocimiento facial entrenados, estamos ya en condiciones de montar la arquitectura del sistema. En la siguiente imagen se muestra el diseño lógico de la arquitectura de nuestro sistema, donde podemos ver el flujo de las diferentes tareas que lo componen. La entrada de nuestro sistema de reconocimiento facial se corresponderá con una imagen, que podrá contener diversos rostros y la salida será la clasificación de cada uno de dichos rostros. La imagen de entrada en principio provendrá de la captura de un frame de un video en tiempo real.

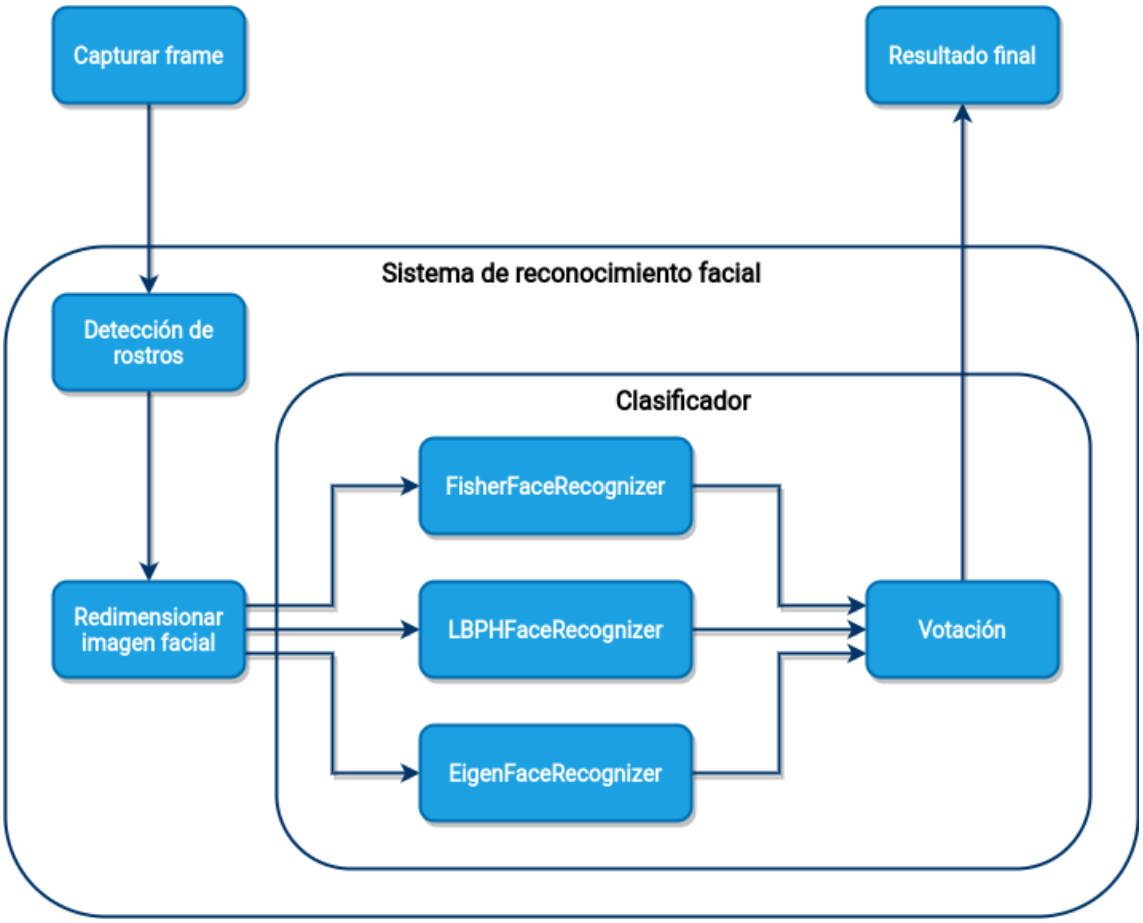

Figura 2: Arquitectura del sistema

Detección de rostros

La detección de rostros se ha logrado mediante el uso de un clasificador en cascada, este es un caso particular de Ensemble learning basado en la concatenación de varios clasificadores , utilizando toda la información recopilada de la salida de un clasificador dado como información adicional para el siguiente clasificador en la cascada. A diferencia de los conjuntos de votación o apilamiento, que son sistemas multiexpertos, la conexión en cascada es de etapas múltiples. OpenCV cuenta con varios clasificadores en cascada ya entrenados, cada uno de los cuales es capaz de reconocer un rostro en cierta perspectiva. Para el caso de nuestro proyecto, como por simplificar el problema optamos por centrarnos solo en rostros frontales, se ha hecho uso de un clasificador en cascada que OpenCV trae entrenado para reconocer rostros de frente.

Redimensionamiento de la imagen

Las imágenes que se utilizaron para entrenar los modelos de reconocimiento facial tenían una dimensión de 128x128, es por ello que la entrada de dichos clasificadores debe ser de la misma dimensión. Por este motivo es necesario reescalar las imágenes de los rostros encontrados, para que tengan la dimensión que nuestro clasificador acepta.

Clasificador

Una vez han sido reescaladas las imágenes de los rostros estas son pasadas a los 3 clasificadores entrenados, cada uno de lo cuales hará una predicción. Para mejorar los tiempos de respuesta y que no haya retardos en la visualización en tiempo real de cámara de vigilancia en la identificación de los rostros, los 3 algoritmos son ejecutados de forma concurrente para realizar la predicción. El tiempo que tarda en obtener cada algoritmo su predicción es bastante bajo, a pesar de ello debemos tener en cuenta que debemos de hacer predicciones en cada frame, lo cual nos limita mucho los tiempos de respuesta. A esto se le suma que por ejemplo en una cámara de vigilancia de un aeropuerto, podrían aparece muchas personas en un solo frame, lo cual dificulta aún más el objetivo de cumplir con los tiempos de respuesta.

Uno de los problemas que se presentó a la hora de definir un umbral para el valor de la confiabilidad, es que este depende de cada modelo y para fijar el valor de este, tuve que hacerlo en base a ensayo y error. Finalmente di con unos valores que maximizaban los resultados obtenidos. Si el valor de la confiabilidad supera el valor del umbral definido para cada uno de los algoritmos, este devolverá como predicción -1, lo cual significará que el rostro no se identifica con ninguno de los rostros utilizados para entrenar, es decir el algoritmo predice que dicho rostro no se corresponde con el de un terrorista.

Votación

Una vez los clasificadores obtiene su predicción esta se pasa a un sistema de votación, en el cual la decisión final se obtiene por el voto de la mayoría. En caso de empate se elige el resultado del clasificador Local binary patterns, debido a que es el que obtiene un mejor rendimiento en la tasa de acierto.

Este sencillo método de votación, permite combinar los resultados de los 3 algoritmos considerados, obteniendo una notable mejora en la predicción realizada, gracias a que combina lo mejor de cada uno. Los algoritmos Eigenfaces y Fisherfaces están basados en apariencia y LBP está basado en características locales, de modo que cada uno de los dos enfoques utiliza un esquema diferente para la extracción de características y representación de los rostros a reconocer. Esto hace que ambos enfoques se complementen perfectamente, al unir sus resultados, pues los fallos de uno pueden ser corregidos por el otro y viceversa.

2.5. Ejemplo de ejecución del sistema

El objetivo del sistema crítico desarrollado, es su funcionamiento en cámaras de videovigilancia. Para probarlo en mi equipo, se ha implementado un pequeño programa que hace uso de la Cam, para capturar los frames y mandárselos al sistema de reconocimiento.

En primer lugar, como ya se ha comentado anteriormente, este sistema reconoce los rostros que hay sobre la imagen y posteriormente trata de identificar dichos rostros. En el caso de este programa si un rostro coincide con alguno de la base de datos con alta confiabilidad, entonces dibujará un cuadrado de color rojo sobre el rostro, indicando que se trata de un terrorista. En caso de que no se identifique con ninguno de los rostros, se pintará de color verde, en señal de verificación positiva, es decir el rostro no supone una amenaza, al no tratarse de un delincuente o terrorista. Tras dibujar los rectángulos de colores identificativos, el sistema devuelve dicha imagen, para que pueda ser representada por pantalla.

Para el ejemplo que se muestra a continuación, he puesto el rostro del supuesto terrorista en la pantalla de mi teléfono móvil. A pesar de que la calidad de la imagen es bastante mala, al estar grabando una pantalla con brillo, vemos que realiza el reconocimiento del terrorista correctamente. Además a identificado correctamente mi cara como rostro desconocido para el sistema.

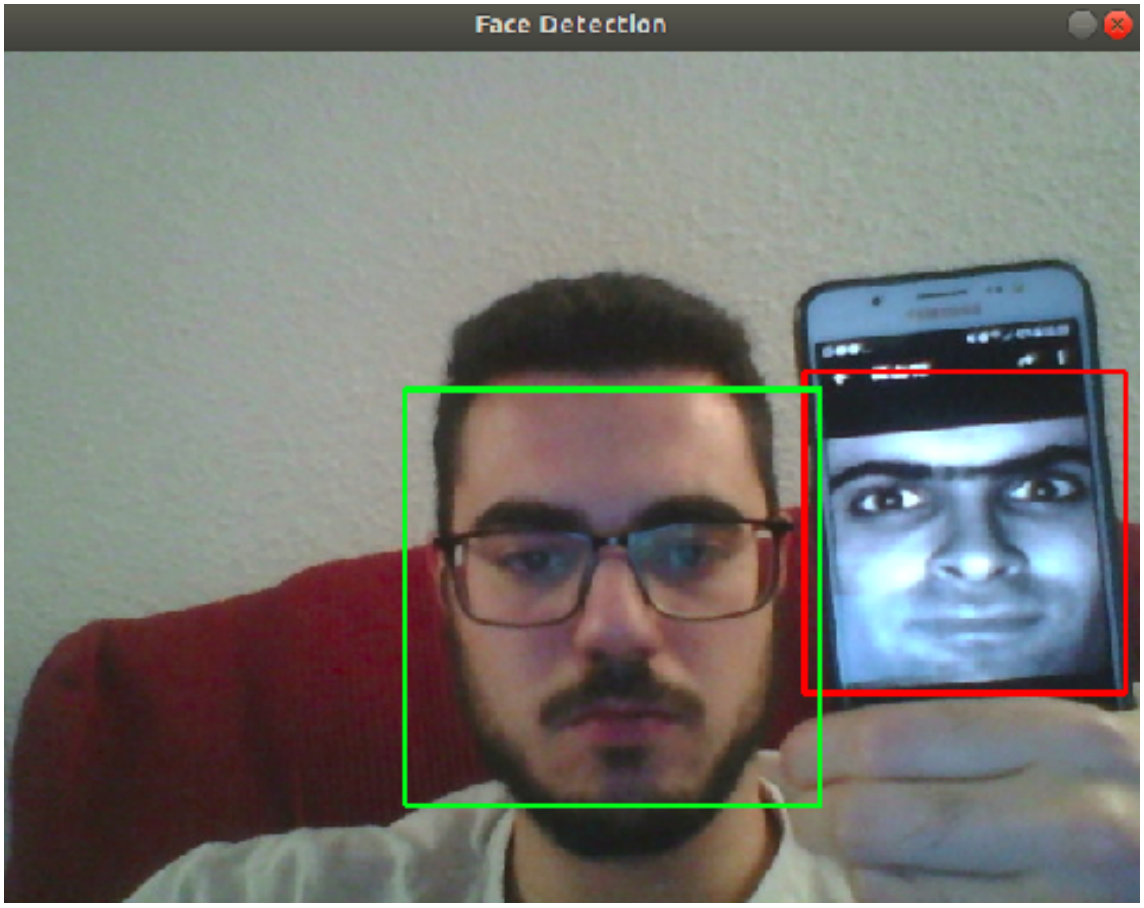

Figura 3: Sistema desarrollado en funcionamiento

3. Pruebas y resultados

En esta sección vamos a ver los resultados de los distintos algoritmos, así como los resultados del algoritmo compuesto.

Comencemos analizando la precisión y la matriz de confusión de los 3 algoritmos considerados, más el algoritmo compuesto.

- **Fisherfaces**

  Este modelo ha obtenido una tasa de acierto del 80,35 %. Su matriz de confusión es la siguiente:

		Observaciones		
		−1	0	1
Predicciones	−1	115	23	0
	0	6	9	0
	1	4	0	11

- **Local binary patterns**

  Este modelo ha obtenido una tasa de acierto del 95,83 %. Su matriz de confusión es la siguiente:

		Observaciones		
		−1	0	1
Predicciones	−1	135	1	2
	0	2	13	0
	1	1	1	13

- **Eigenfaces**

  Este modelo ha obtenido una tasa de acierto del 89,88 %. Su matriz de confusión es la siguiente:

		Observaciones		
		−1	0	1
Predicciones	−1	128	3	7
	0	3	10	2
	1	2	0	13

- **Algoritmo compuesto**

  Este modelo ha obtenido una tasa de acierto del 97,61 %. Su matriz de confusión es la siguiente:

		Observaciones		
		−1	0	1
Predicciones	−1	137	1	0
	0	2	10	2
	1	1	0	13

Como podemos observar el mejor de los 3 algoritmos considerados (sin tener en cuenta el algoritmo compuesto) es el Local Binary Patterns. Este ha obtenido un accuracy de 95,83 %. Como podemos observar en su matriz de confusión ha cometido el error de identificar a una persona inocente como terrorista en 3 ocasiones, e igualmente en 3 ocasiones se ha confundido a un terrorista con una persona inocente. Ambos fallos pueden traer graves consecuencias, por lo que reducir ambos tipos de error es crucial para el sistema crítico. Si observamos los otros dos algoritmos vemos que la tasa de error es aun peor.

A pesar de los fallos cometidos por los 3 algoritmos tenidos en cuenta, cuando los combinadmos, los fallos de alguno de ellos son compensados por los otros y conseguimos un sistema mucho mas robusto, con una tasa de error más baja. Como podemos comprobar el porcentaje de acierto del algoritmo combinado es de un 97,61 %, mejorando los resultados de los 3 algoritmos. De nuevo vemos que se han cometido 3 fallos, al identificar a 3 personas como terroristas, cuando en realidad no lo eran, no obstante podemos ver que con respecto a la identificación de terroristas como personas inocentes ahora tan solo tenemos un error. Esto supone una mejora significativa, pues la identificación a tiempo de un terrorista es crucial para impedir un atentado terrorista.

3.1. Análisis de distintas métricas para el algoritmo compuesto

Nuestro sistema evalúa sus resultados sobre un conjunto abierto, donde se tiene que cuando un rostro no coincide con ninguna de las identidades disponibles, este se clasifica como desconocido. Esto hace que en nuestro sistema se puedan dar los siguientes casos:

- Verdadero positivo: el sistema clasifica correctamente el rostro con su identidad.
- Falso positivo: el sistema clasifica incorrectamente un rostro.
- Verdadero negativo: el sistema clasifica correctamente un rostro como desconocido.
- Falso negativo: el sistema clasifica un rostro como desconocido, cuando en realidad este pertenecía a alguna de las entidades de la base de datos.

Los valores anteriores nos va a permitir calcular una serie de métricas que se describen a continuación:

- Sensitividad (recall): nos aporta el porcentaje de aciertos considerando solo la clase positiva.

- Especificidad: nos aporta el porcentaje de aciertos considerando solo las predicciones positivas.
- Precisión: porcentaje de casos positivos acertados con respecto al total de predicciones positivas.
- Accuracy: porcentaje total de acierto.
- F1-Score: nos permite ver como de buenas son las medidas de sensitividad y precisión de forma simultánea, mediante la media armónica de estas.

En la siguiente tabla se muestran los valores de las medidas descritas y algunos más:

Measure	Value	Derivations
Sensitivity	0.9643	TPR = TP / (TP + FN)
Specificity	0.9786	SPC = TN / (FP + TN)
Precision	0.9000	PPV = TP / (TP + FP)
Negative Predictive Value	0.9928	NPV = TN / (TN + FN)
False Positive Rate	0.0214	FPR = FP / (FP + TN)
False Discovery Rate	0.1000	FDR = FP / (FP + TP)
False Negative Rate	0.0357	FNR = FN / (FN + TP)
Accuracy	0.9762	ACC = (TP + TN) / (P + N)
F1 Score	0.9310	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.9175	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

Figura 4: Resultados obtenidos

Como podemos ver la tasa de acierto de este algoritmo es bastante buena, y además mantiene un bajo ratio de falsos positivos. Tanto los valores de sensitividad como de especificidad son bastante aceptables.

3.2. Análisis de los tiempos de ejecución

Los tiempos de ejecución de los algoritmos son muy bajos, por lo que para poder medir el tiempo con mayor precisión, estos se han medido en el reconocimiento facial de una imagen que contiene 57 rostros. Por lo tanto el tiempo obtenido es el de el reconocimiento de los 57 rostros. Si dividimos dicho tiempo por 57, obtendríamos el tiempo que se tardaría en media en detectar un rostro con el algoritmo compuesto. Como el algoritmo compuesto, en realidad se basa en la ejecución de 3 algoritmos de reconocimiento, deberíamos de nuevo dividir por 3 el tiempo de ejecución, para obtener el tiempo que tarda aproximadamente un algoritmo.

- **Exprimento 1**

  En un primer lugar se han medido el tiempo de ejecución sin utiliza ningún tipo de paralelismo. Los resultados han sido:

  - **0.651274 segundos** en el reconocimiento de 57 rostros.
  - **0.01142586 segundos** aproximadamente por rostro el algoritmo compuesto.
  - **0.00380862 segundos** aproximadamente por rostro el algoritmo simple.

- **Experimento 2**

  Esta vez se han medido los tiempos mediante la ejecución de los 3 algoritmos de forma paralela. Los resultados han sido:

  - **0.798642 segundos** en el reconocimiento de 57 rostros.
  - **0.014011263 segundos** aproximadamente por rostro el algoritmo compuesto.
  - **0.004670421 segundos** aproximadamente por rostro el algoritmo simple.

  Al contrario de lo que podríamos imaginar, al ejecutar los 3 algoritmos de forma concurrente, el tiempo de ejecución es mayor. Lo cual nos quiere decir que el tiempo que ahorramos por un lado, lo estamos perdiendo en las sucesivas veces que OpenMP tiene que preparar el entorno de ejecución concurrente.

- **Experimento 3**

  Esta vez en lugar de paralelizar los 3 algoritmos, vamos a paralizar el bucle que itera sobre los 57 rostros. Con ello se espera que al no ser un bucle tan interno como el que paralelizamos en el experimento anterior, los tiempos de ejecución esta vez si se vean reducidos. Veamos los resultados:

  - **0.451358 segundos** en el reconocimiento de 57 rostros.

- **0.007918561 segundos** aproximadamente por rostro el algoritmo compuesto.
- **0.00263952 segundos** aproximadamente por rostro el algoritmo simple.

Esta vez si que se aprecia una importante mejora, en concreto una mejora del 30 %, con respecto al tiempo de ejecución obtenido en el experimento 1, en el que no se práctico ningún método de paralelización.

4. Conclusiones

En este proyecto se propuso la implementación de un método que combinara el uso de varios algoritmos de reconocimiento facial, con el objetivo de mejorar la tasa de acierto. Dicho modelo ha sido implementado correctamente, mostrando unos resultados satisfactorios y superiores a los que obtiene cada uno de los algoritmos por separado. Además se ha conseguido reducir la tasa de falsos negativos, el cual puede considerarse un error más grave (también dependerá del escenario) que un falso positivo, pues en el primer caso podrías estar dejando escapar a un delincuente y en el segundo caso finalmente el ojo humano puede determinar que se trataba de un falso positivo y dejar libre a la persona inocente que fue erróneamente clasificada.

También se ha logrado incrementar la velocidad del algoritmo, aunque no de la forma que se propuso (paralelizando los 3 algoritmos), sino separando los rostros que aparecen en una imagen y analizarlos de forma concurrente, obteniendo una mejora del 30 % en el tiempo de respuesta.

También se ha logrado incluir el sistema de reconocimiento, en un sistema en tiempo real, haciendo para ello uso de la videocámara del portátil.

A pesar de los buenos resultados que se han obtenido, no podemos olvidarnos que han sido logrados en un entorno simplificado del problema, donde los rostros estaban de frente, las caras estaban correctamente alineadas y el número de personas a identificar como terroristas era tan solo de 2, lo cual hace que el modelo de reconocimiento sea muy ligero y los tiempos de predicción sean mucho menores. En la realidad la base de datos de terroristas y delincuentes puede ser enorme lo cual dificultaría el problema. Además la detección de los rostros en la cámara no siempre tiene porque ser frontal, como en nuestro caso, sino que puede estar de perfil, con diferentes perspectivas, etc. Todo ello obligará a utilizar múltiples detectores de rostros, uno por cada perspectiva a considerar, además de que los tiempos de respuesta se verán aumentados por un preprocesamiento que deberá aplicarse a las imágenes.

Además las imágenes analizadas (para simplificar aún más el problema y obtener modelos más ligeros) tenía una dimensión de tan solo 128x128 y el tiempo de detección de rostros también era bajo debido a que la calidad de la cámara de mi portátil es baja. Una imagen de mayor resolución provocará que los tiempos de respuesta aumenten.

La paralelización en un solo ordenador, a pesar de que se haya logrado en nuestro problema simplificado, probablemente se quede corta, para satisfacer los tiempos de respuesta tan ajustados que requiere un sistema crítico en tiempo real como este. Algunas soluciones alternativas, podrían basarse en el uso de tecnologías cloud, para distribuir los procesos y acelerar los tiempos de respuesta.

Respecto a la tasa de error que se ha obtenido, a pesar de ser de tan solo 2 %, probablemente sea demasiado alto como para considerar usar el método en un sistema real. Para mejorar esta tasa de acierto aún más posiblemente, la clave no esté en seguir mejorando aún más los algoritmos utilizados, sino mejorar la calidad del

conjunto de datos, aunque en muchos casos esto no siempre es posible, debido a que no se disponen de suficientes fotografías de algunos de los terrosristas más buscados.

5. Bibliografía

- Método de optimización para reconocimiento facial basado en la fusión de algoritmos y segmentación según las características de las imágenes, Manuel Freire
- Documentación oficial OpenCV
- Heterogeneous Robots Sharing Personal Tours in Multi-Floor Environments, Igor Rodriguez
- The Extended Yale Face Database B