**Universidad Rey Juan Carlos**

**Máster Oficial en Visión Artificial**

**“Biometría Informática”**

**Curso académico 2012/2013**

**Orión García Gallardo**

**Práctica 3: Influencia de la dimensionalidad en el reconocimiento facial 2D.**

**Introducción**

El reconocimiento facial 2D es una técnica de la biometría para identificar o verificar la identidad de personas a partir de imágenes 2D de sus caras. Dicha técnica empezó a usarse a partir de los años 60 [1]. Fue entonces cuando se desarrolló el primer sistema semiautomático para reconocimiento facial, que requería del administrador para localizar rasgos. En la actualidad reconocimiento facial está ampliamente extendido, algunos ejemplos se pueden ver en el tránsito de fronteras, usos forenses, identificación de sospechosos y criminales por parte de la policía, etc.

Aunque a los humanos nos resulta relativamente sencillo reconocer personas, automatizar este proceso en un algoritmo es bastante complejo y costoso computacionalmente. Uno de los factores que más dificultan este reconocimiento, además del típico de la iluminación, es la expresión facial que tenga el sujeto en el momento de tomar la instantánea. Aunque algunos trabajos se han realizado en este sentido [2][3][4], una determinada expresión facial (alegría, risa, enfado, tristeza, etc) puede hacer que una misma persona parezca totalmente distinta.

El objetivo de esta práctica es comprobar el efecto de PCA en el reconocimiento facial 2D. El Análisis de Componentes Principales (PCA) [5] es un método basado en apariencia que genera una representación compacta de la cara proyectando la imagen a un sub-espacio de dimensión menor. Para evaluar esta técnica lo que se realizará es:

1. Entrenar una red neuronal de tipo perceptrón multicapa (MLP) usando como patrones las imágenes completas de la cara (es decir, sin realizar extracción de características). Probar el clasificador entrenado usando el conjunto de patrones de test, para calcular los % de reconocimiento correcto y erróneo.
2. Representar los resultados de clasificación usando una matriz de confusión.
3. Repetir los pasos 3) y 4) pero aplicando PCA sobre las imágenes de entrenamiento y test, antes de realizar el entrenamiento y test de la red perceptrón multicapa. Indicar resultados de reconocimiento aplicando la combinación: PCA-MLP. Mostrar algunas imágenes que forman la base del subespacio reducido.
4. Comparar resultados obtenidos con y sin aplicación de PCA. Mostrar algunas imágenes correcta eincorrectamente clasificadas en cada caso.

# Descripción de la solución

# Entre el amplio abanico de opciones disponibles para implementa esta práctica se ha optado por el lenguaje de programación C++ y la librería de manejo de imágenes OpenCV. Esta librería contiene una gran cantidad de funcionalidad principalmente orientada a la visión por computación en tiempo real. Está desarrollada por Intel pero es libre bajo la licencia open source BSD. OpenCV permite hacer uso de forma fácil y sencilla de numerosos algoritmos ya implementados, entre ellos, los clasificadores que vamos a usar en esta práctica. De esta manera uno puede centrarse en seleccionar los parámetros más convenientes para los algoritmos y en organizar los entrenamientos para obtener los resultados más óptimos.

# Dentro de las funcionalidades desarrolladas por la librería OpenCV se encuentra una implementación del algoritmo PCA sobre matrices. También se encuentra implementado un algoritmo que usa los PCA para el reconocimiento facial en imágenes. Este algoritmo se denomina Eigenfaces [6] y tiene la ventaja sobre otros algoritmos de reconocimiento facial, de ser muy rápido y eficiente. Usar eigenfaces permite operar con una cantidad grande de imágenes en un tiempo relativamente corto. Sin embargo, la eficiencia de esta técnica se reduce cuando tratamos con caras vistas desde distintos ángulos y diferentes niveles de intensidad [7].

# Para la realización de esta práctica hemos usado la base de datos de imágenes de caras Extended Yale Facedatabase en su primera versión. Esta base de datos nos suministra 16352 imágenes de 28 personas tomadas con distintos tipos de iluminación y ángulos. Para simplificar la evaluación se han cogido tan sólo 10 sujetos, lo que reduce el algoritmo a procesar 5840 imágenes (584 por persona). Por otro lado, y con el objetivo de hacer más ligero el sistema se ha reducido la resolución de las imágenes de 168x192 a 30x30. Esto nos permite reducir bastante a nivel computacional el número de operaciones que tienen que realizar las redas neuronales.

# El siguiente paso ha sido dividir la base de datos de imágenes en conjunto de test y conjunto de entrenamiento. Para realizar esto se ha realizado una [K-fold cross-validation](http://en.wikipedia.org/wiki/Cross-validation_(statistics)#K-fold_cross-validation) donde K se ha asignado al 25% de los datos o lo que es lo mismo a 146 imágenes por sujeto.

# A continuación se han creado dos redes redes neuronales que se han evaluado por separado. Una para emplearse con el conjunto bruto de los datos y la otra para aplicarse con los datos después de haber usado PCA sobre ellos. Para hacer ambas redes lo más parecidas posibles se han implementado las dos con 3 capas. La primera red se ha diseñado de la siguiente forma: primera capa con 900 neuronas, segunda capa con 20 neuronas y tercera capa con 10 neuronas. Por otra parte, la segunda red se ha diseñado como sigue: la primera capa con 80 neuronas, la segunda capa con 20 neuronas y la tercera capa con 10 neuronas. Se han elegido 10 neuronas de salida porque se ha codificado de tal manera que sólo una neurona debería activarse por cada imagen. Así si se activara la primera neurona significaría que la imagen representaría al primer sujeto, si se activara la segunda representaría al segundo sujeto y así sucesivamente. En cuanto a las neuronas de entrada, en la primera red representan el número de pixeles (30x30), mientras que en la segunda representan el número de componentes al que se han reducido la imagen. Esto implica que antes de pasarse los datos por la segunda red se ha aplicado la reducción de dimensionalidad PCA que nos aporta la librería OpenCV. Posteriormente cada una de las imágenes se han proyectado sobre el espacio PCA obtenido antes de usarlas para el entrenamiento.

# Para que no resulte muy tedioso la ejecución de las redes neuronales se han establecido los parámetros de estas de tal manera que los entrenamientos duren sobre el minuto y medio.

# Por otro lado, también se han procesado los datos con el algoritmo Eigenfaces al que se hizo referencia en la introducción. Este procesamiento se ha realizado para hacer una comparación en la evaluación con las otras dos técnicas.

# Datos aplicados

# Para la realización de esta práctica hemos usado la base de datos de imágenes de caras Extended Yale Facedatabase en su primera versión. Esta base de datos nos suministra 16352 imágenes de 28 personas tomadas con distintos tipos de iluminación y ángulos. Para simplificar la evaluación se han cogido tan sólo 10 sujetos, lo que reduce el algoritmo a procesar 5840 imágenes (584 por persona). Por otro lado, y con el objetivo de hacer más ligero el sistema se ha reducido la resolución de las imágenes de 168x192 a 30x30.

# Resultados

# A continuación se muestran las tablas con las matrices de confusión de los tres algoritmos descritos anteriormente. A saber, la red neuronal con los datos en bruto, la red neuronal con los datos con dimensionalidad reducida (PCA) y el algoritmo de Eigenfaces.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sujeto/Predicción | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total | % |
| 0 | 579 |  | 5 |  |  |  |  |  |  |  | 584 | 0,9914 |
| 1 | 270 | 314 |  |  |  |  |  |  |  |  | 584 | 0,5377 |
| 2 | 414 |  | 170 |  |  |  |  |  |  |  | 584 | 0,2911 |
| 3 |  |  |  | 584 |  |  |  |  |  |  | 584 | 1 |
| 4 | 29 |  |  | 3 | 552 |  |  |  |  |  | 584 | 0,9452 |
| 5 | 160 |  |  |  |  | 415 | 7 |  |  | 2 | 584 | 0,7106 |
| 6 | 182 |  |  |  |  | 32 | 369 | 1 |  |  | 584 | 0,6318 |
| 7 | 256 |  |  |  |  |  | 2 | 326 |  |  | 584 | 0,5582 |
| 8 | 91 |  |  |  |  |  |  |  | 448 | 45 | 584 | 0,7671 |
| 9 | 106 |  |  |  |  |  |  |  | 42 | 436 | 584 | 0,7466 |
| Total |  |  |  |  |  |  |  |  |  |  | 5840 | 0,718 |

# Matriz de confusión red neuronal datos en bruto

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sujeto/Prediccion | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total | % |
| 0 | 584 |  |  |  |  |  |  |  |  |  | 584 | 1 |
| 1 | 277 | 307 |  |  |  |  |  |  |  |  | 584 | 0,5257 |
| 2 | 292 |  | 292 |  |  |  |  |  |  |  | 584 | 0,5 |
| 3 | 217 |  |  | 367 |  |  |  |  |  |  | 584 | 0,6284 |
| 4 | 152 |  |  | 1 | 431 |  |  |  |  |  | 584 | 0,738 |
| 5 | 226 |  |  |  |  | 356 | 2 |  |  |  | 584 | 0,6096 |
| 6 | 207 |  |  |  |  | 16 | 361 |  |  |  | 584 | 0,6182 |
| 7 | 226 |  |  |  |  |  |  | 358 |  |  | 584 | 0,613 |
| 8 | 218 |  |  | 1 |  |  |  |  | 329 | 36 | 584 | 0,5634 |
| 9 | 205 |  |  |  |  |  |  |  | 16 | 363 | 584 | 0,6216 |
| Total |  |  |  |  |  |  |  |  |  |  | 5840 | 0,6418 |

# Matriz de confusión red neuronal datos con PCA

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sujeto/Prediccion | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Total | % |
| 0 | 583 | 1 |  |  |  |  |  |  |  |  | 584 | 0,9983 |
| 1 |  | 584 |  |  |  |  |  |  |  |  | 584 | 1 |
| 2 | 1 |  | 583 |  |  |  |  |  |  |  | 584 | 0,9983 |
| 3 |  |  |  | 584 |  |  |  |  |  |  | 584 | 1 |
| 4 |  |  |  | 1 | 583 |  |  |  |  |  | 584 | 0,9983 |
| 5 |  |  |  |  |  | 552 | 23 | 5 | 4 |  | 584 | 0,9452 |
| 6 |  |  |  |  |  |  | 584 |  |  |  | 584 | 1 |
| 7 |  |  |  |  |  | 1 |  | 583 |  |  | 584 | 0,9983 |
| 8 |  |  |  |  |  |  |  |  | 544 | 40 | 584 | 0,9315 |
| 9 |  |  |  |  |  |  | 4 |  | 47 | 533 | 584 | 0,9127 |
| Total |  |  |  |  |  |  |  |  |  |  | 5840 | 0,9783 |

# Matriz de confusión Eigenfaces

# Conclusiones

# Como puede observarse los mejores resultados se han obtenido con el algoritmo de Eigenfaces implementado en OpenCV. Aunque este algoritmo ha tardado sobre los 5 minutos de media en entrenarse con los datos proporcionados, los resultados obtenidos demuestran que el algoritmo es ampliamente más eficiente con un valor cercano al 98%. Los algoritmos en los que se han empleado redes neuronales, aunque en media se han ejecutado durante menos tiempo (sobre el minuto y medio), están claramente por debajo. Se han obtenido mejores resultados con los datos en bruto con un valor cercano al 72%. El problema principal para la segunda red es que el PCA ha cogido muchas componentes del primer sujeto y esto produce que muchas de las caras se clasifiquen como de este sujeto por error. Si se pudiera evitar este problema los valores de eficiencia de la red neuronal combinada con el PCA estarían cercanos a los obtenidos por el algoritmo de Eigenfaces.

**Referencias**

[1] R. Chellappa, C. Wilson, and S. Sirohev, “Human and machine recognition of faces: A survey,” in Proceedings of IEEE, May 1995, vol. 83, pp. 705-740.

[2] Saket S Kulkarni, Narender P Reddy y SI Hariharan (2009). “[Facial expression (mood) recognition from facial images using committee neural networks](http://www.biomedical-engineering-online.com/content/8/1/16)”. BioMedical Engineering OnLine (Department of Biomedial Engineering, University of Akron, Akron, OH, USA).

[3] Ekman P, Friesen WV (1978). “[The Facial Action Coding System A Technique for the Measurement of Facial Movement](http://face-and-emotion.com/dataface/facs/manual/TitlePage.html)”. Consulting Psychologists Press (San Francisco, USA).

[4] Philippos Vanger, Robert Hoenlinger, Hermann Haken (1998). “[Computer aided generation of prototypical facial expressions of emotion](http://www.dgps.de/fachgruppen/methoden/mpr-online/issue4/art3/article.html)”. Methods of psichological Research.

[5] Duda, Richard O. and Hart, Peter E. and Stork, David G., Pattern Classification (2nd Edition) 2001.

[6] L. Sirovich and M. Kirby (1987). "Low-dimensional procedure for the characterization of human faces". Journal of the Optical Society of America A 4 (3): 519–524.

[7] M. Turk and A. Pentland (1991). ["Face recognition using eigenfaces"](http://www.cs.ucsb.edu/~mturk/Papers/mturk-CVPR91.pdf). Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 586–591.