

Avance de Proyecto

Identificación facial con pocas muestras por clase

Martín Villanueva¹

¹Departamento de Informática
Universidad Técnica Federico Santa María

Máquinas de Aprendizaje, 2015

Outline

1 El Problema

- Identificación facial
- Datasets

2 Propuestas de solución

- Linear Discriminant Analysis
- Multi-class SVM
- Convolutional Neural Networks
- Dissimilarity SVM

Outline

1 El Problema

- Identificación facial
- Datasets

2 Propuestas de solución

- Linear Discriminant Analysis
- Multi-class SVM
- Convolutional Neural Networks
- Dissimilarity SVM

Definición del Problema

El problema tratado consiste en, dadas una *pocas* muestras de imágenes faciales de distintos individuos, generar un algoritmo que permita identificar correctamente a cada individuo, dada una imagen distinta a las anteriores. Más formalmente:

- Se define $S_{tr} = \{\mathbf{s}_1, \dots, \mathbf{s}_M\}$ como el conjunto de entrenamiento con imágenes faciales de K individuos.
- El problema de **identificación** es, dada una imagen de *test* \mathbf{p} , determinar cuál de las K clases del conjunto de entrenamiento S_{tr} corresponde.
- Se restringe el problema a casos donde cada clase posee pocas muestras en S_{tr} , lo cual es una situación bastante realista.

Datasets

Los datasets a ocupar corresponden a Faces94 y Faces95. Los cuales tiene las siguientes características respectivamente:

- **Faces94:** 153 individuos, 20 imágenes por individuo, resolución 200×180 , hombres y mujeres, fondo fijo, poca variación en escalamiento, poca variación de posición de la imagen y condiciones de iluminación, y considerables variaciones en expresiones faciales.
- **Faces95:** 72 individuos, 20 imágenes por individuo, resolución 200×180 , hombres y mujeres, fondo con sombra, considerable variación en escalamiento, poca variación en posición de imagen, iluminación variable, y variaciones considerable en expresiones faciales.

Faces94



Figura: Imágenes características de un individuo en Faces94

Faces95



Figura: Imágenes características de un individuo en Faces95

Otras consideraciones

Para emular las restricciones del problema, se sigue la siguiente metodología

- Para cada uno **Faces94** y **Faces95** se generan 20 training y testing sets respectivos.
- Cada training set se genera tomando 3 – 5 muestras aleatorias por cada individuo, dejando el resto de las 15 – 17 imágenes restantes en el testing set.
- Luego, para cada dataset (**Faces94** o **Faces95**) hay en total 60 sets de training y testing respectivos.
- La presentación de resultados, se realiza con Errorbars (t-student 95 % de confianza) del error rate sobre los 20 datasets respectivos.

Outline

1 El Problema

- Identificación facial
- Datasets

2 Propuestas de solución

- Linear Discriminant Analysis
- Multi-class SVM
- Convolutional Neural Networks
- Dissimilarity SVM

LDA

- La implementación ocupada corresponde la de Scikit-Learn. Para cada una de las clases (personas) genera un función discriminante lineal δ_k , que permite diferenciar a cada una de las clases.
- Supuestos Fuertes: 1) La probabilidad multivariada de las características $P(x_m|y = k)$ se distribuye normal, y 2) La matriz de covarianza para cada una de las clases es igual.
- Como LDA es un modelo generativo (sin hiperparámetros), no es necesario realizar cross-validation para el modelo. Permite ahorrar gran tiempo de computación.

LDA: Resultados en Faces94

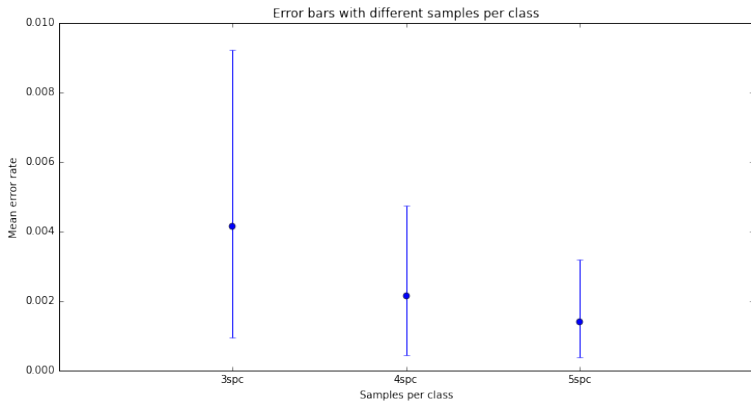


Figura: Resultados obtenidos con LDA en **Faces94**

LDA: Resultados en Faces95

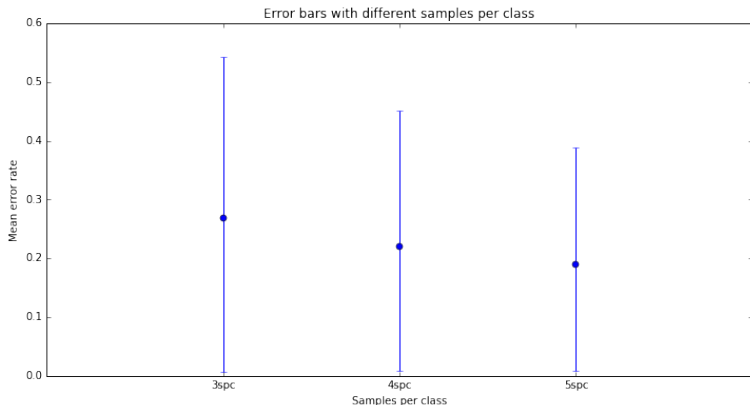


Figura: Resultados obtenidos con LDA en **Faces95**

Multi-class SVM

- Support Vector Machines (SVM) es un método de clasificación binario (adaptable a problemas de múltiples clases), que encuentra la frontera de decisión lineal óptima (hiperplano óptimo) separadora de las clases.
- Implementación y entrenamiento de multiclass SVM's con kernels tanto lineales como RBF. La implementación de Scikit-Learn advierte: "SVC implement one-against-one" approach (Knerr et al., 1990) for multi- class classification".
- Se utilizan ν -SMV's, debido a que facilita la configuración del parámetro de holgura $\nu \in (0, 1]$, y a su interpretación.
- Selección de hiperparámetros ν y γ (en kernels rbf) se realiza stratified cross-validation. Para que esto tenga sentido, el número de folds debe ser igual al número de muestras por clase.

Reducción de dimensionalidad

- **Problema:** Las dimensiones de las imágenes ($200 \times 180 = 36000$) corresponden al total de features de cada foto. Esto hace necesario ocupar reducción de dimensionalidad para tomar las características realmente importantes (aquellas que permiten diferencias entre las clases) y mejorar los tiempos de entrenamientos y eficiencia.
- Como técnica de reducción de dimensionalidad, se ha decidido ocupar LDA, representación también conocida como Fisher-Faces (reducción de dimensionalidad supervisada).
- **Idea:** Proyectar la data en espacio donde se maximice la inter-class variance y minimice la intra-class variance.

Reducción de dimensionalidad (2)

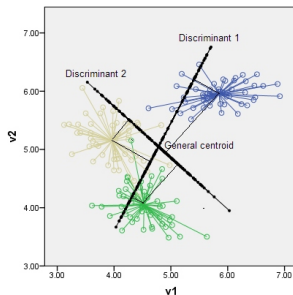


Figura: LDA como reducción de dimensionalidad

- El número de máximo de discriminantes es $\min(\text{dim}, K - 1)$, por lo tanto en un problema con muchas más dimensiones (features) que clases, la reducción de dimensionalidad es considerablemente.

Resultados linear SVM Faces94

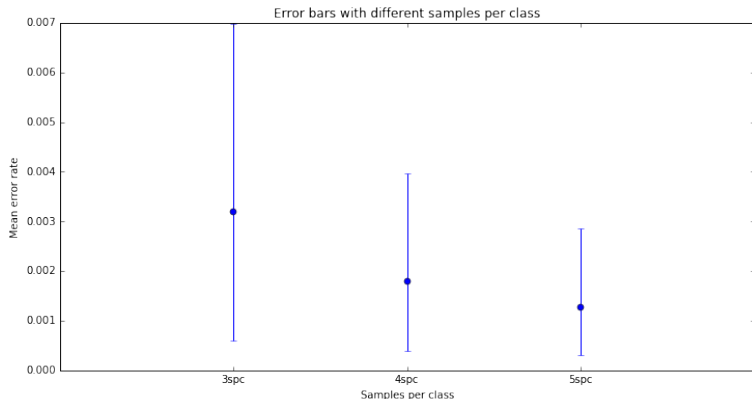


Figura: Resultados obtenidos con linear SVM en **Faces94**

Resultados linear SVM Faces94 (2)

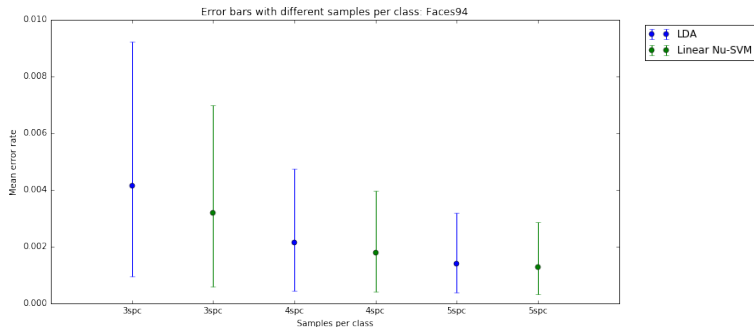


Figura: Comparación de resultados obtenidos con linear SVM en **Faces94**

Resultados linear SVM Faces95

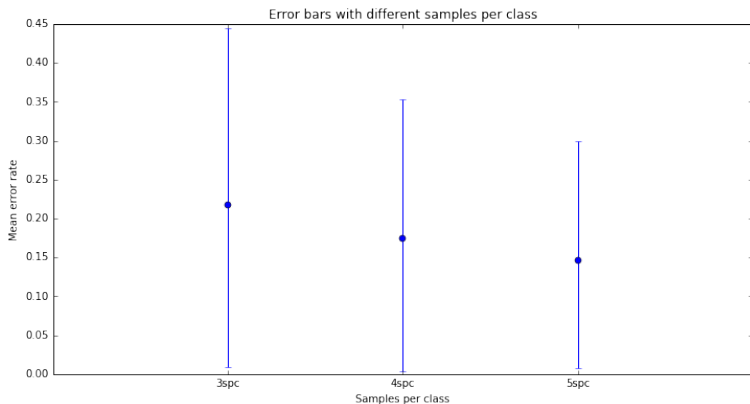


Figura: Resultados obtenidos con linear SVM en **Faces95**

Resultados linear SVM Faces95 (2)

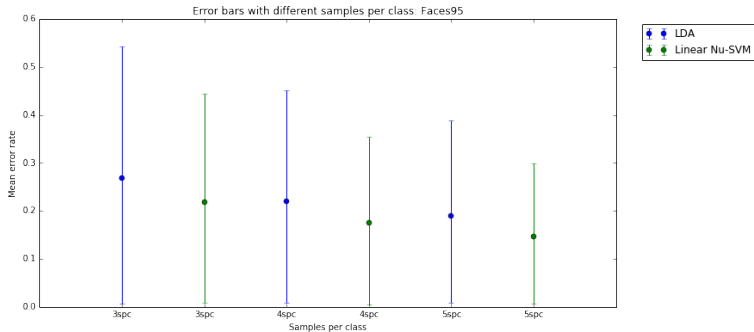


Figura: Comparación de resultados obtenidos con linear SVM en **Faces95**

Resultados RBF SVM Faces94

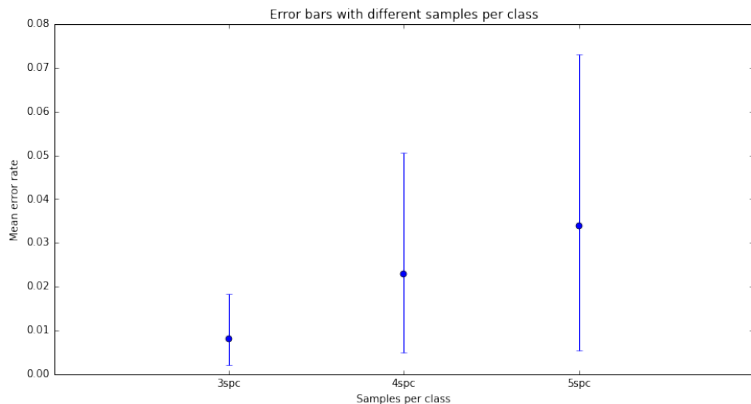


Figura: Resultados obtenidos con RBF SVM en **Faces94**

Resultados RBF SVM Faces94 (2)

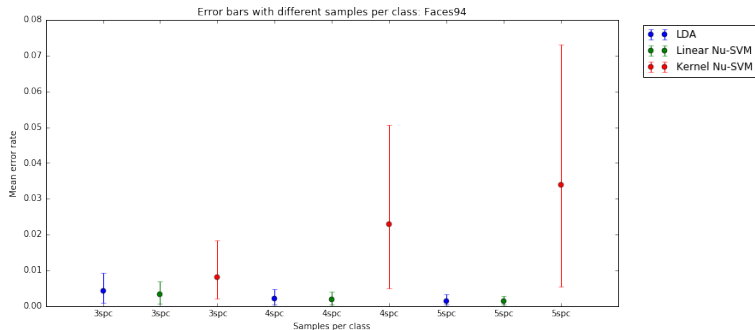


Figura: Comparación de resultados obtenidos con RBF SVM en **Faces94**

Resultados RBF SVM Faces95

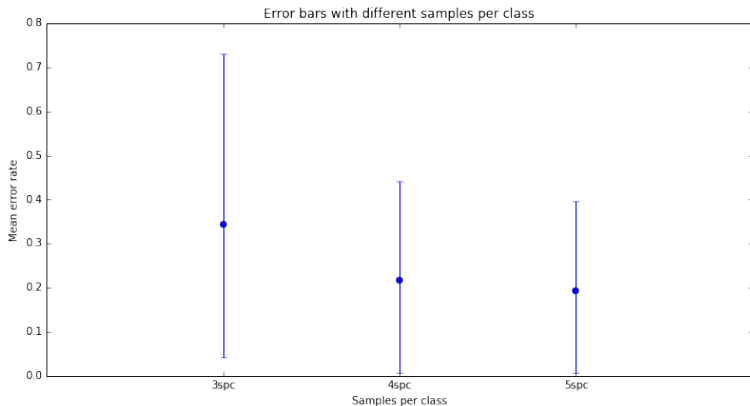


Figura: Resultados obtenidos con RBF SVM en **Faces95**

Resultados RBF SVM Faces95 (2)

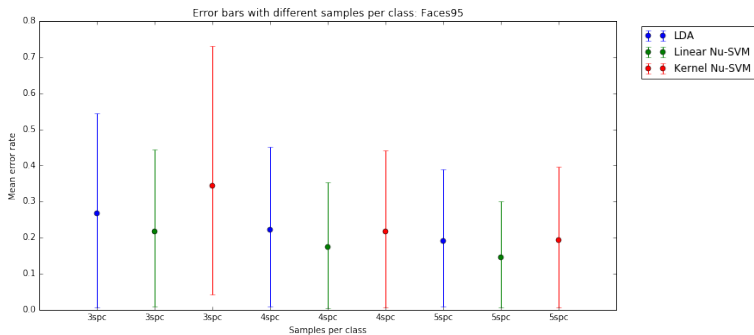


Figura: Comparación de resultados obtenidos con RBF SVM en **Faces95**

Análisis de resultados

- Linear-SVM tiene un buen resultado en general, superando a LDA (mejor capacidad de generalización).
- Para faces94, ocupar modelos que aumenten la complejidad (como RBF SVM) sólo empeora los resultados.
- Para faces95 RBF SVM se comporta mejor, pero no supera a los otros dos algoritmos anteriores.

Preliminares

Las capas que maneja una CNN son básicamente

- **Convolutacional:** Consiste de un arreglo rectangular de neuronas, requiere que la capa previa sea también un arreglo rectangular. Cada una de estas neuronas toma inputs de sección una rectangular de la capa previa (*local receptive field*), siendo los pesos para esta región rectangular los mismos para cada neurona en la capa convolutacional (*weight sharing*). Luego, la capa convolutacional es sólo una convolución de imagen de la capa previa, donde los pesos especifican el filtro de convolución.

Preliminares (2)

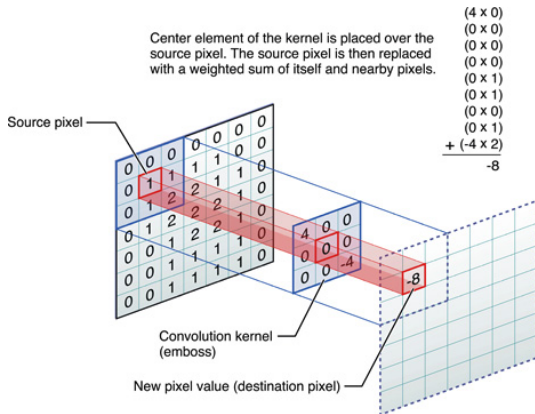


Figura: Esquema de convolución como filtro

Preliminares (3)

- **Pooling:** Después de cada capa convolucional, suele estar una capa de *pooling*. Estas toman una pequeña región rectangular desde el el output de la capa convolucional, y realiza *subsampling* produciendo un solo input desde el bloque. El proceso de pooling puede implementarse de varias formas: Tomando el máximo, el promedio, promedio con pesos, etc. El objetivo es dar holgura espacial a la posición de las características computadas.
- **Fully Connected:** Después de cualquier configuración de capas convolucional y de pooling, el *aprendizaje de alto nivel* de la red, sigue siendo por una capa densa.

Preliminares (4)

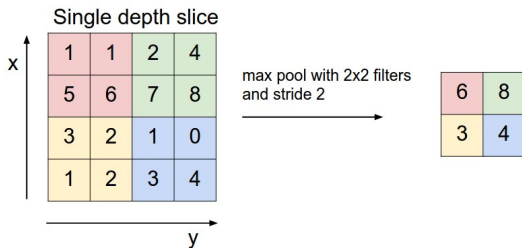


Figura: Esquema de max pooling

Preliminares (5)

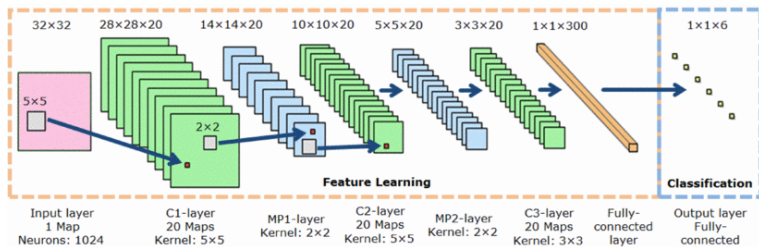


Figura: Esquema clásico de CNN

Configuración utilizada

- **Convolution Layer 1:** ReLU activation, 12 channels, 5×5 filter.
- **Pooling Layer 1:** Pooling type: Max pooling, Pool shape: 2×2 .
- **Convolution Layer 2:** ReLU activation, 8 channels, 4×4 filter.
- **Pooling Layer 2:** Pooling type: Max pooling, Pool shape: 2×2 .
- **Fully Connected Layer:** ReLU activation, 128 neurons.
- **Softmax Layer.**

Adicionalmente se ocupa como algoritmo de optimización *RMSprop*, número máximo de épocas de 500 y *dropout* como regularización.

Resultados CNN en Faces94 (1)

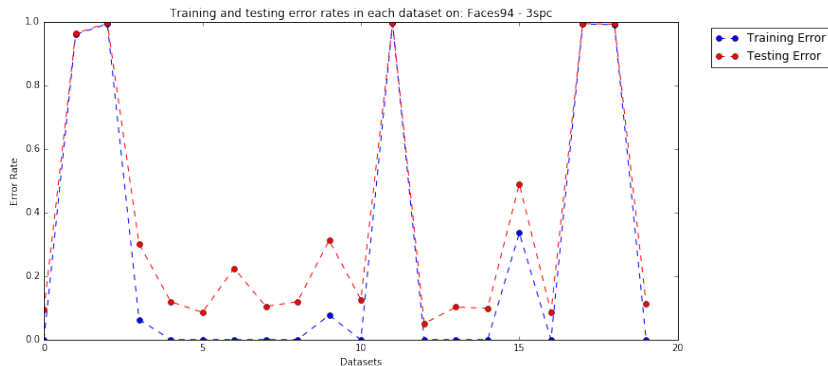


Figura: Resultados obtenidos con CNN en **Faces94** - 3spc

Resultados CNN en Faces94 (2)

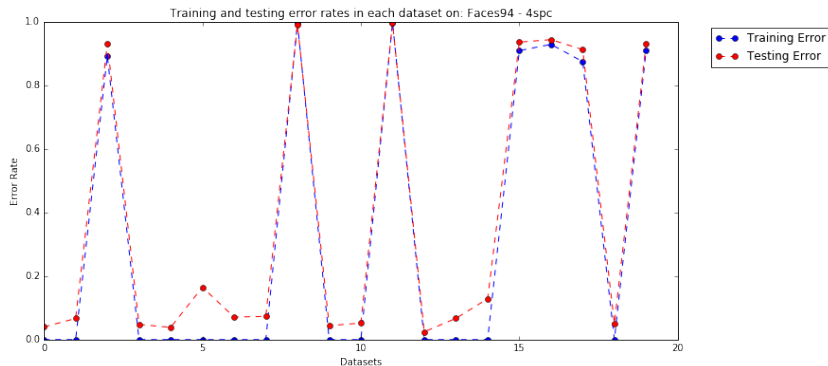


Figura: Resultados obtenidos con CNN en **Faces94** - 4spc

Resultados CNN en Faces94 (3)

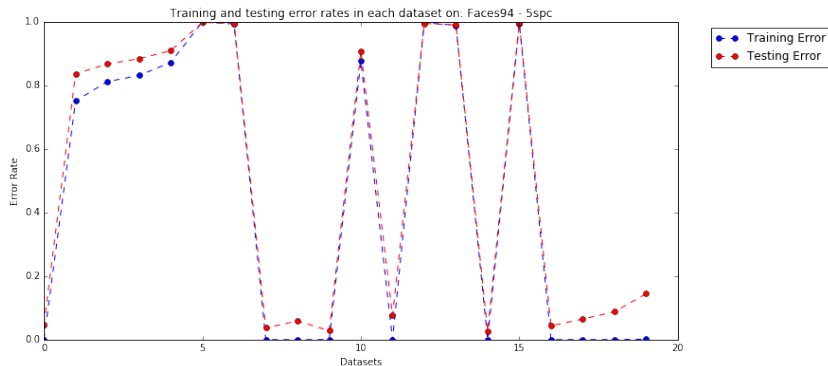


Figura: Resultados obtenidos con CNN en **Faces94** - 5spc

Resultados CNN en Faces95 (1)

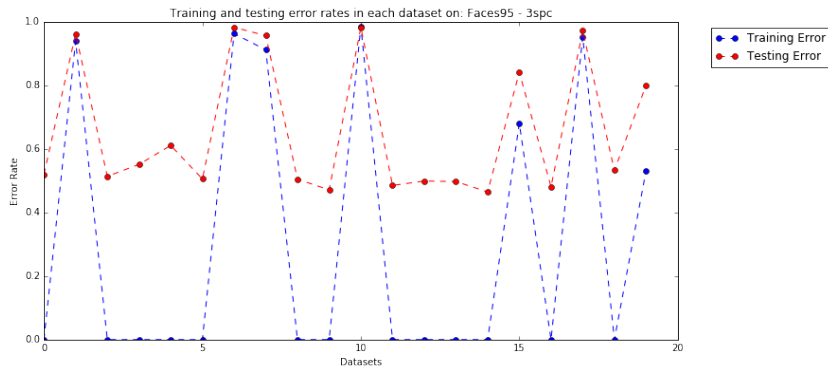


Figura: Resultados obtenidos con CNN en **Faces95** - 3spc

Resultados CNN en Faces95 (2)

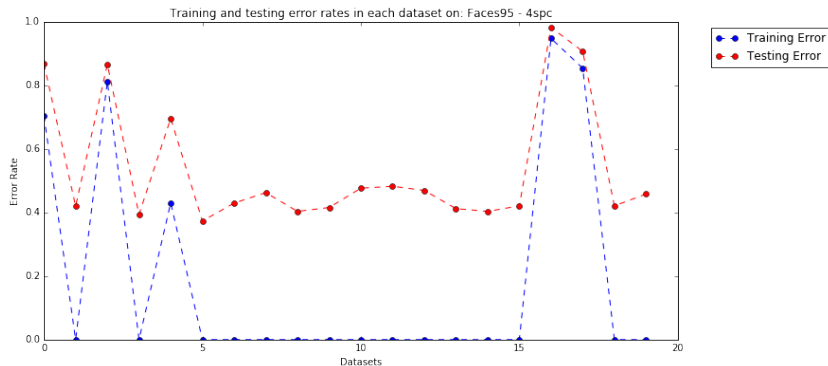


Figura: Resultados obtenidos con CNN en **Faces95** - 4spc

Resultados CNN en Faces95 (3)

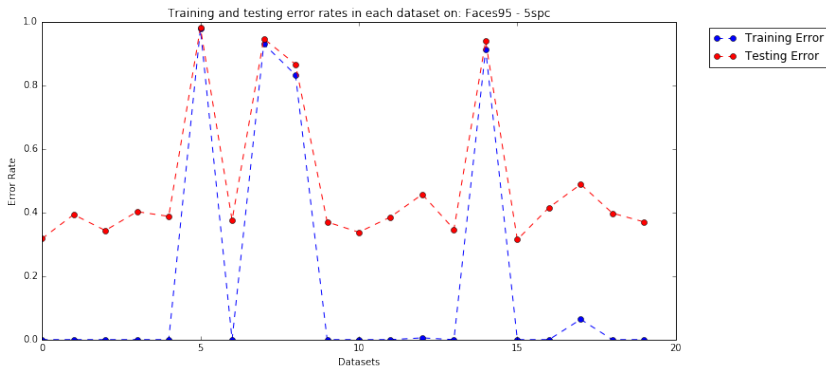


Figura: Resultados obtenidos con CNN en **Faces95** - 5spc

Comentarios finales CNN

- No fue posible realizar *cross-validation*, pues cada *epoch* toma una cantidad muy alta de tiempo.
- Por lo tanto parámetros fueron configurados siguiendo la arquitectura de la red LeNet5 extendiendo el número de canales y unidades de la capa densa.

D-SVM

- Hasta ahora, todos los métodos tratados resuelven el problema de identificación como K -class classification.
- El método aquí propuesto, transformar el espacio de representación de los datos, reduciendo el problema a clasificación binaria.
- Para esto se debe mapear los datos desde el *Image Space* al *Difference Space*.
- Se definen entonces dos conjuntos, C_1 y C_2 ; El primero corresponde al *within-class differences set* y contiene las disimilitudes entre datos de la misma clase. El segundo es *between-class difference set* y contiene las disimilitudes entre datos de distinta clase.

D-SVM (2)

Definition

Dea $S_{tr} = \{\mathbf{s}_1, \dots, \mathbf{s}_M\}$ el conjunto de entrenamiento con K individuos. Para indicar que dos individuos pertenecen a la misma clase se denota $\mathbf{s}_i \sim \mathbf{s}_j$, y en caso contrario $\mathbf{s}_i \not\sim \mathbf{s}_j$. Se define adicionalmente la función de disimilitud $\phi : R^N \times R^N \rightarrow R^S$ con $S \leq N$, como aquella función que mapea dos imagenes, hacia el *difference space*. Luego es posible definir

$$C_1 = \{\phi(\mathbf{s}_i, \mathbf{s}_j) \mid \mathbf{s}_i \sim \mathbf{s}_j\} \quad (1)$$

$$C_2 = \{\phi(\mathbf{s}_i, \mathbf{s}_j) \mid \mathbf{s}_i \not\sim \mathbf{s}_j\} \quad (2)$$

D-SVM (2)

- El entrenamiento de la D-SVM (Dissimilarity SVM), tiene como entradas los conjuntos C_1 y C_2 .
- Como resultado de la clasificación se obtiene la superficie de decisión

$$f(\mathbf{x}) = \sum_m^{M_s} \alpha_m y_m K(\mathbf{v}_m, \mathbf{x}) + b = 0$$

donde la función $f(\mathbf{x})$ puede ocuparse de discriminante para determinar si un dato pertenece a una clase u otra.

D-SVM (3)

El proceso de identificación es como sigue:

- Sea \mathbf{p} la imagen a identificar sobre un conjunto de imágenes conocidas S_{tr} . Se computa la representación en el *difference space*: $\phi(\mathbf{p}, \mathbf{s}_i) \forall \mathbf{s}_i \in S_{tr}$.
- Evaluar la función de discriminante $f(\mathbf{x})$ sobre cada uno de los resultados anteriores.
- La clase de pertenencia será aquella para la cual $f(\phi(\mathbf{p}, \mathbf{s}_i))$ sea la menor posible:

$$\underset{i}{\operatorname{argmin}} f(\phi(\mathbf{p}, \mathbf{s}_i)) \quad (3)$$

Local Binary Pattern (1)

- **Idea:** Forma de representación de imágenes, como una composición de micro-patrones que describen las estructuras presentes en estas. Los distintos patrones que emergen corresponden a las características presentes en las imágenes.

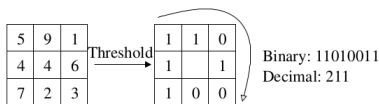


Fig. 1. The basic LBP operator.



Figura: Local Binary Pattern Operator

Local Binary Pattern (2)

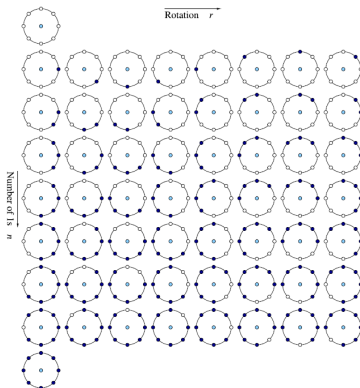


Fig. 2.4 The 58 different uniform patterns in $(8, R)$ neighborhood

Figura: Uniform Local Binary Patterns

Histograma Espacial de Características

- Cada uno de estos 59 patrones representan una característica, y la frecuencia o distribución (histogramas) de estas definen las estructuras presentes en las imágenes.
- Sin embargo se requiere también de información espacial, acerca de donde se encuentran estas características.

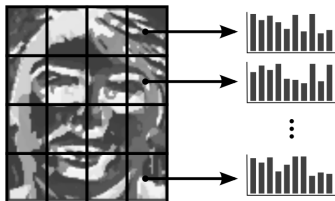


Figura: Histograma espacial de características

Método Propuesto

El modelo propuesto difiere de aquel propuesto por P. Jonathon Philips en *Support Vector Machines Applied to Face Recognition*, en la función $\phi(\cdot, \cdot)$ de mapeo al *difference space*. Sean dos imágenes representadas en la forma de histograma espacial ($\mathbf{h}_1, \mathbf{h}_2$) con S histogramas, luego la función es

Definition

$\phi(\mathbf{h}_1, \mathbf{h}_2) = \mathbf{d} = (d_1, \dots, d_S)$ tal que \mathbf{d}_i corresponde a la diferencia existente entre los histogramas i -ésimos de las respectivas imágenes. Tal diferencia se computa con la función:

$$\chi^2(x, y) = \sum_k \frac{(x_k - y_k)^2}{x_k + y_k}$$

Detalles de Implementación

- Cuando existen s muestras por clase, se pueden hacer $\binom{s}{2}$ combinaciones para formar el conjunto C_1 .
- El conjunto C_2 se forma tomando aleatoriamente igual cantidad $\left(\binom{s}{2}\right)$ de pares de datos de distinta clase.
- El valor que obtiene la función discriminante $f(\mathbf{x})$, se promedia para los datos que pertenecen a la misma clase en S_{tr}

Resultados D-SVM Faces94

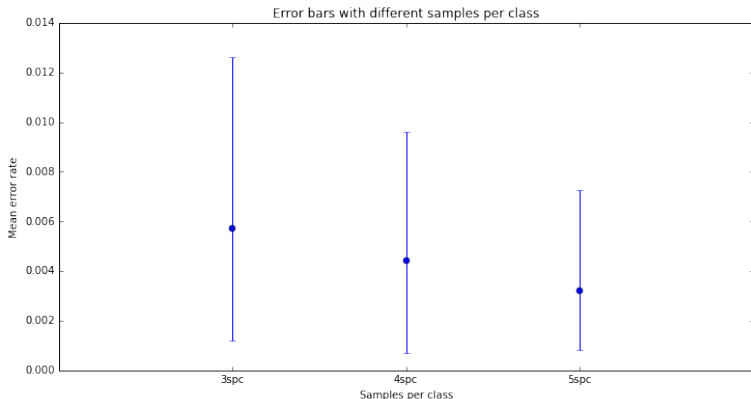


Figura: Resultados obtenidos con D-SVM en **Faces94**

Resultados D-SVM Faces94 (2)

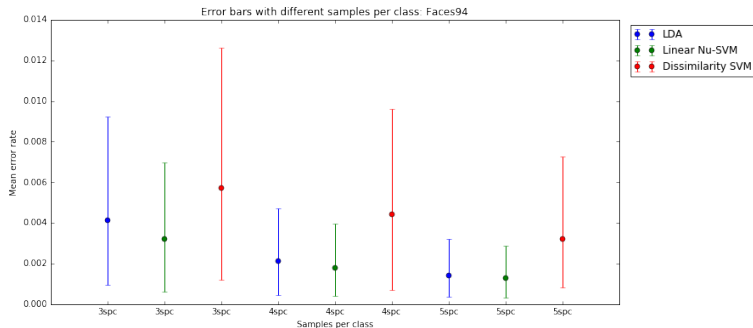


Figura: Comparación de resultados obtenidos con D-SVM en **Faces94**

Resultados D-SVM Faces95

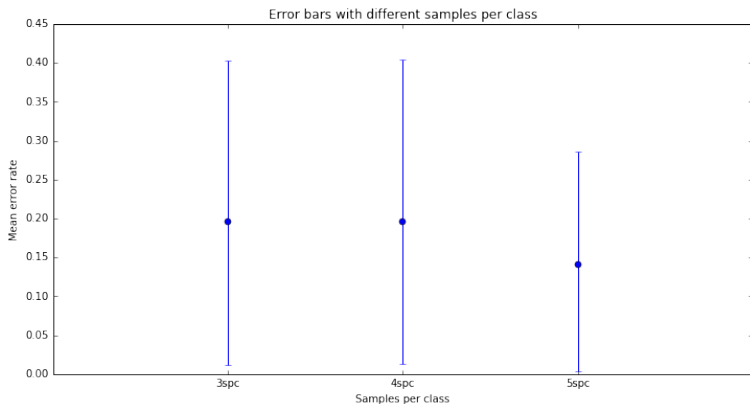


Figura: Resultados obtenidos con D-SVM en **Faces95**

Resultados D-SVM Faces95 (2)

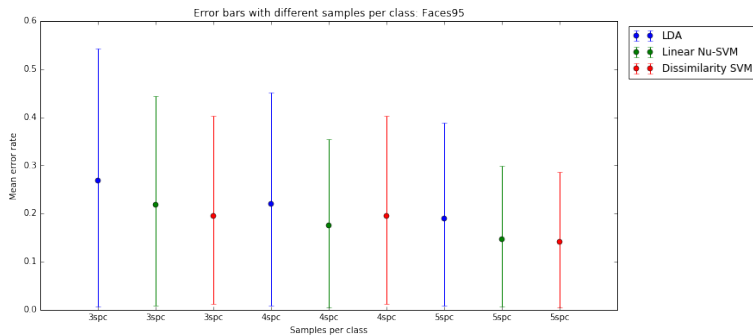


Figura: Comparación de resultados obtenidos con D-SVM en **Faces95**