Reconocimiento de Patrones

Informe de Proyecto

Autores:
Agustín Navcevich 4.646.135-9
Alfredo Solari 4.952.867-3

Mail:
agustin.navcevich@fing.edu.uy
alfredo.solari@fing.edu.uy

Instituto de Ingeniería Eléctrica

Facultad de Ingeniería

Universidad de la República

3 de diciembre de 2017

Índice

1. Descripción del problema

En este trabajo se presenta un posible enfoque para el diseño de clasificadores automáticos punta a punta para la base de datos FMNIST.

La base de datos FMNIST contiene 60000 muestras donde el contenido de cada muestras son imágenes de distintas prendas de vestir que pertenecen a 10 clases diferentes. Es una base de datos basada en el MNIST[1] pero generada con la consigna de que la clasificación de las clases sea de mayor dificultad que la base original.

Esta base consta de muestras con 784 características que corresponden a una matriz de 28x28 pixeles de una imagen en escala de gris de 8 bits de resolución.

1.1. Objetivo

Diseñar un sistema de reconocimiento de patrones que obtenga el mejor resultado posible de clasificación de la clase de prenda a la que corresponde cada imagen, en base a las características establecidas en la parte anterior.

Para esto se evaluarán técnicas de selección y/o extracción de características de los datos proporcionados. Luego de esto se estudiará la matriz de confusión que obtenga para el sistema propuesto y se analizará como mejorar los resultados.

2. Método y paradigma utilizado

2.1. Método

Lo que se utilizó para poder hacer la clasificación de los patrones dados por las imágenes, fue un sistema de reconocimiento de patrones que fuera de punta a punta. Esto es, que para un patrón de entrada en el sistema, la salida sea el patrón clasificado. Para esto en el sistema a pesar de ser punta a punta debe contar con distintos módulos.

Un sistema de reconocimiento de patrones de punta a punta esta compuesto por tres principales módulos. El primero de ellos es donde se pre-procesan los datos para quitar todos aquellos en los cuales pueden traer un mal rendimiento en el clasificador. Luego como en el reconocimiento de patrones no se clasifican directamente las entidades u objetos sino su descripción, el segundo módulo es el de extracción y selección de características. Luego de haber hecho esta extracción y selección de características se llega a la última etapa del proceso que es la parte de los clasificadores. En este sistema se pueden diferenciar dos etapas distintas; la cual en la primera entrenamos el sistema para que reconozca los patrones que queremos y la segunda etapa es una etapa de clasificación propiamente dicha, donde podemos tener una etapa de validación y luego una etapa de test. En la parte de validación se utilizo en nuestro caso un sistema de retroalimentación en cuanto a los dos últimos bloques.

Como se vió que en todos los casos ya existentes no se había hecho mucha ingeniería de datos, eligiendo buenas características que mejoraran el sistema; entonces se planteó trabajar en este aspecto para atacar mejor la clasificación y así poder mejorar el sistema final. En la parte de entrenamiento del sistema se trabajo como un sistema realimentado, en el sentido que dadas características que eran extraídas, se fijaban si mejoraban la clasificación y a partir de ahí se inferían nuevas características que pudieran mejorar los clasificadores. entonces el método utilizado se basa en seleccionar y extraer características, clasificar con algún método, buscar nuevas características que mejoren la clasificación y volver a iterar el proceso. Para esto La principal idea es luego de haber extraído las características y haber hecho una clasificación, analizar la matriz de confusión, y pensar que características se podrían extraer para de alguna manera "desconfundir" las clases con menor acierto o menor puntaje.

2.2. Paradigma

A partir de lo planteado como metodología nos planteamos un paradigma a seguir, este fue enfocarse en la extracción de características globales y no en el paradigma de "un pixel, una característica" porque se estimó que es más realista trabajar sobre características globales que sobre una combinación lineal de valores de pixel, ya que en un problema **real** las imágenes podrían llegar a tener una cantidad muy elevada de píxeles y/o cada

[1] Base de imágenes de digitos escritos a manos, de tamaño 28x28 píxeles.

muestras(imagen) podría tener una distinta cantidad de píxeles que la anterior[2]. Por lo cual para cada imagen se le extrajo una cantidad de características menor que la cantidad de píxeles, esto también teniendo en mente la maldición de la dimensionalidad.

3. Extracción de Características

Se siguieron dos enfoques para la extracción de las características. El primer enfoque fue un enfoque orientado al problema que se tenía, esto es: a partir de conocimientos que teníamos sobre imágenes y algunas de las formas de poder extraer información de las mismas, poder extraer características cómo un resultado de un análisis global de cada imagen, de forma que fueran relevantes a la hora de mejorar la separación entre clases. El segundo enfoque fue un enfoque mas automático en el sentido de que a partir de los píxeles de la imagen se corrieron distintos algoritmos que seleccionan los caracteres de acuerdo a distintos criterios y con esta selección de los mejores píxeles se formaron distintas características.

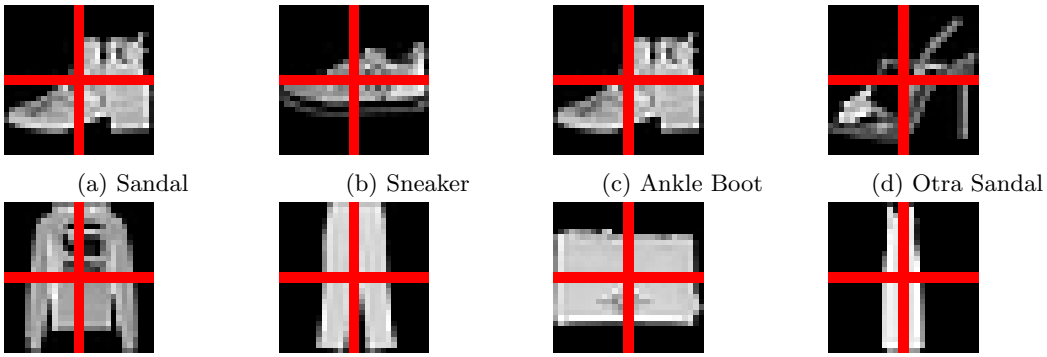
Primeramente se describirán los distintos métodos para el primer enfoque mencionado.

3.1. Momentos Estadísticos

Cómo primera aproximación se han utilizado los momentos estadísticos. Según el libro de González[2], plantea que este tipo de características sirven para clasificar texturas, es decir objetos que tienen cierta estructura que se repite a lo largo de un misma muestra. Hemos observado que para el caso de las muestras de ropa de la base FMNIST no alcanza solamente con momentos estadísticos.

3.2. Momentos Estadísticos v2

Observando las imágenes, se puede observar que hay clases que se localizan en determinadas zonas. Por ejemplo si partimos la imagen en 4 cuadrantes, como en la fiugra:(2)



(a) Sandal (b) Sneaker (c) Ankle Boot (d) Otra Sandal

(e) Pullover (f) Trousers (g) Bag (h) Dress

Figura 1: Máscaras utilizadas para el análisis de frecuencia.

Figura 2

Se observa que las Camisetas, Pullovers y vestidos presentan componentes en las 4 áreas. Pero los Zapatos se encuentran mayoritariamente en 2, 3 y 4 y no en 1. Por lo cual esta pareció ser una buena forma de seguir aumentando la separabilidad, el calcular los mismos momentos que en la parte anterior pero localmente, dividiendo la imagen en cuatro cuadrantes[3]. en la práctica esto generó un mejor poder clasificación que los momentos globales, pero no lo suficiente como para dejar de buscar otras características distintas.

3.3. Área y Perímetro

Continuando con las características que dan una noción de forma se buscan: el área, el perímetro y el factor de forma(que es una combinación no lineal de ambos). Esto nos da una noción de tamaño y de circularidad.

[2]Incluso podría tratarse de que las muestras en lugar de ser imágenes fueran regiones de interés dentro de una imagen (O como se denominan usualmente por su sigla en inglés ROIs).

[3]Una opción evaluada fue seguir partiendo en NxN trozos la imagen. Se descartó esta opción para seguir buscando características de otra índole. Pero podría ser una buena forma de seguir aumentando la separabilidad de las clases.

Logrando separar muy bien los zapatos y sandalias, que al tener tiras poseen una relación perímetro/area menor que por ejemplo un pullover.

Ambiciosamente también se espera que ayude a distinguir camisetas de pullovers ya que el largo de las mangas darían lugar a cambios en la relación área-perímetro[4]. Para hallar estas características primero se hace una umbralización de las imágenes, y se halla área y perímetro.

Para hallar el área y el perímetro: para el área simplemente la suma de la región umbralizada. Para el perímetro se hallan los bordes de la región con morfología matemática y se suma sobre toda la imágen, la cantidad total será la cantidad de píxeles de borde[5].

3.4. Template Matching

Otro de los métodos que se usaron para poder distinguir y desconfundir las clases fue un "template matching". Es decir generar un prototipo representativo para cada clase. De esta manera podemos obtener un puntaje asociado a cada muestra proyectándolas en un template y sumando todos los valores de las muestras.

El procedimiento fue entonces:

- hallar los templates como un promedio de muchas muestras de cada clase.
- Umbralizar a las imágenes (muestras) y hacerles un blur gausseano.
- Proyectar la imagen umbralizada y blurreada en el espacio de los templates para obtener el score, la característica.

Se muestra gráficamente parte de los resultados:

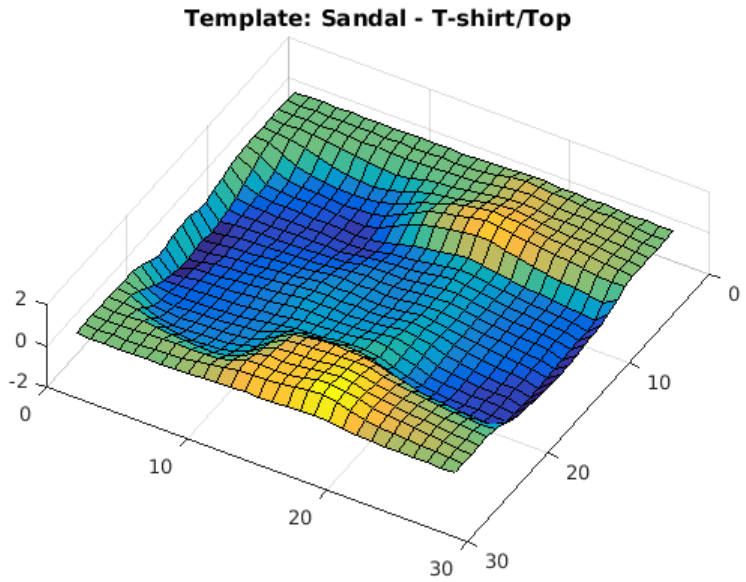
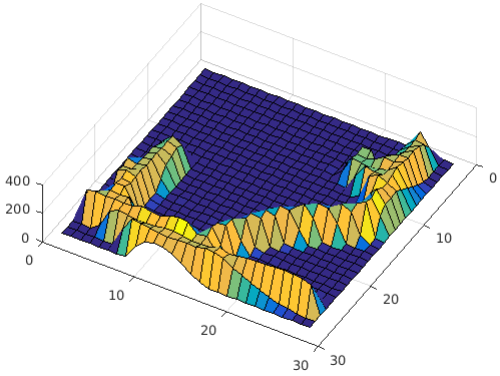
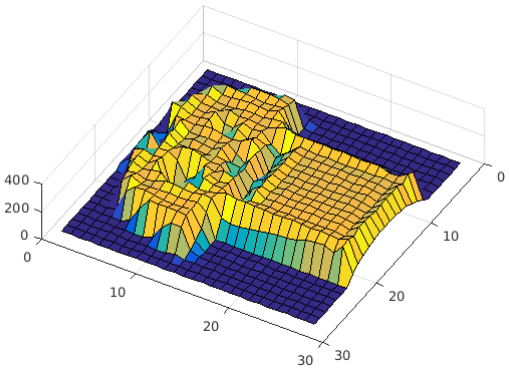


Figura 3: Template "Sandalia - Camiseta/Top"

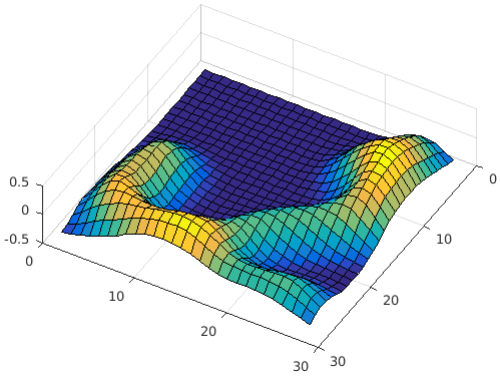
[4]En la realidad no se encontró una umbralización global que separara bien las mangas del resto del pullover o shirt o coat.
[5]También se usa algo de morfología para remover puntos que pueden haber quedado como .ºutliers", fuera del contorno de la prenda.



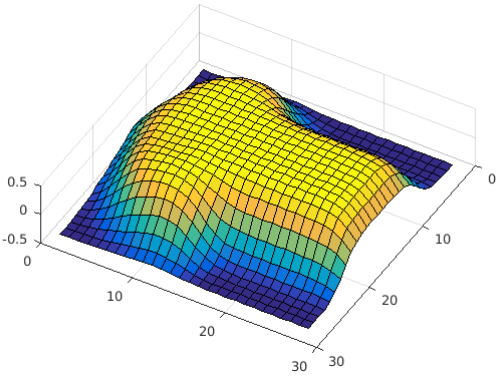
(a) Muestra original: sandalia



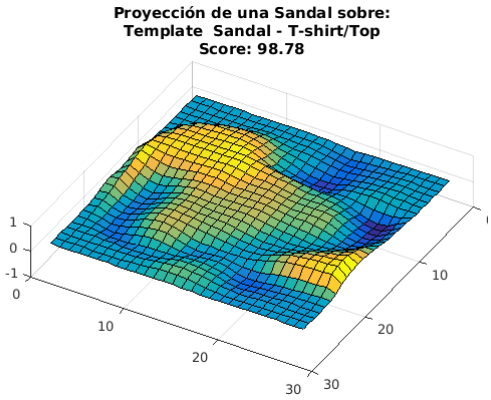
(b) Muestra original: Tshirt



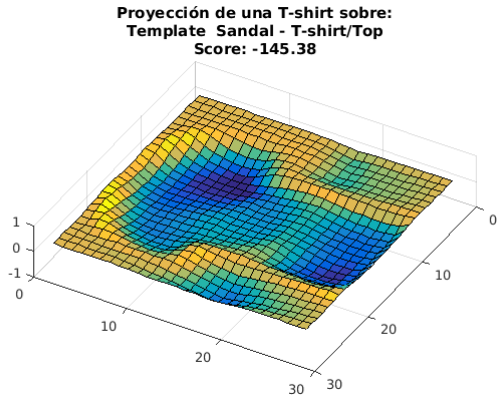
(c) Sandalia + Umbral + Blur



(d) Tshirt + Umbral + Blur



(e) Proyección de la sandal sobre el template



(f) Proyección de la Tshirt sobre el template

Figura 4: Dos muestras de clases distintas proyectadas en el Template “Sandalia - Camiseta/Top”

3.5. Características en el espacio de Fourier

3.5.1. Potencia espectral en bandas en bandas

Otro campo explorado en la extracción ha sido la potencia espectral en bandas. Es decir un análisis de potencia en el dominio Fourier. Se observa que las camisas de manga larga y los pullover y coat deberían aportar ciertas frecuencias en el eje horizontal que las camisetas no, ya que las manga aportarían ese componente. También los pantalones se diferenciarían de los vestidos de esa manera, ya que se generarían componentes distintos en frecuencias espaciales en el eje x.

Para extraer estos datos visualizados se decidió utilizar tres máscaras: una para bajas frecuencias otra para medias frecuencias y otra para frecuencias altas. Se observa en la figura(6) las mascaras utilizadas. Luego de hacer un AND entre cada una de las máscaras y las transformadas de Fourier de las imágenes se integra la

potencia, es decir, se suma el valor absoluto del resultado de dicho AND. [6]

Figura 5: Máscaras utilizadas para el análisis de frecuencia.

3.5.2. Características de forma en el espacio de Fourier

Otras características extraídas fueron, luego de una umbralización de la FFT un conteo de la cantidad de objetos en el eje x y en el eje y, así como la cantidad de objetos totales. Obtenidos los primeros con una simple cuenta de la cantidad de subidas y bajadas restrictos al eje Ox y Oy respectivamente en las imágenes umbralizadas. En el segundo caso se realizó utilizando un etiquetado (o "labeling") y se obtuvo dicha cantidad extrayendo la cantidad de etiquetas. Además se extrajeron las componentes x e y de las direcciones principales de la FFT umbralizada.

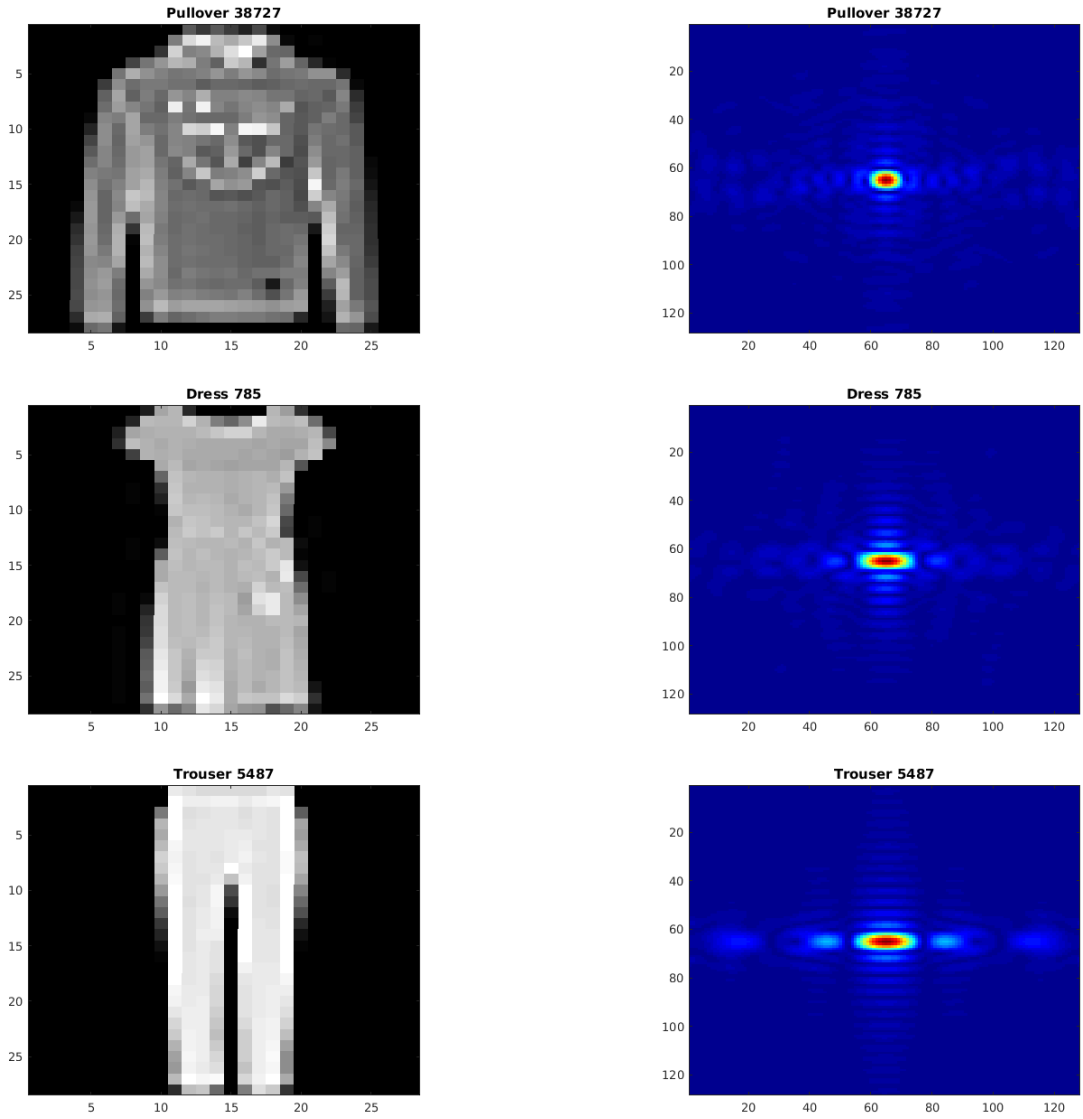


Figura 6: Máscaras utilizadas para el análisis de frecuencia.

[6] Las transformada de fourier que se hicieron fueron de tamaño 128x128.

3.5.3. Conclusiones sobre características en el espacio de Fourier

Ha sido inesperado para nosotros que los resultados de la clasificación no mejoraran luego de agregar las características en el espacio de Fourier. Nos proponemos un momento para analizar el por qué.

Una razón posible del fracaso de las características de forma puede ser, por ejemplo, que se utilizó un umbral porcentual referido al valor máximo del valor absoluto de la FFT, en lugar de un umbral automático[7]. A pesar de que se probó con varias imágenes y el resultado era el esperado, se hace notable que esos resultados observados en la etapa de decisión del umbral no eran representativos de la mayor parte del conjunto.

Dentro de las características de potencia por bandas: no se ha podido extraer de la transformada la información visualizada. Una posible mejora de esto, en caso de que quisiéramos insistir en características basadas en la FFT, sería hallar la relación entre las potencias, es decir la división entre la potencia en la banda baja y media, o entre la banda baja y alta, etc. Notesé que luego de observar la baja capacidad de discriminación de estas características se probó normalizar la potencia total de cada muestra antes de enmascarar sin lograr ningún cambio en el resultado.

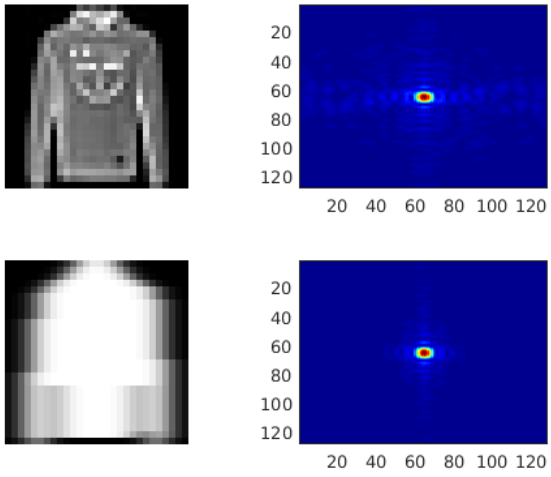


Figura 7: Transformada de Fourier de un pullover antes y después de aplicarle un umbral y un borroneado gausseano

Otra posibilidad para considerar a futuro sería hacer el mismo preprocesamiento que se ha hecho antes del template matching[8] y obtener una imagen umbralizada y borroneada(filtro gaussiano). Esto, que ha probado ser efectivo para el template matching disminuiría el valor de las altas frecuencias ya que se obtendría un pasabajos. Y por lo tanto se eliminaría (en parte) el efecto observado para las mangas y pantalones. Cómo se observa en la figura(7).

[7] Umbrales automáticos se podrían elegir muchos, por ejemplo: Otsu, Entropía de Renyi, Li, Huang, Shanbhag, entre otros. Varios de los cuales usan conceptos de teoría de la información para separar una imagen en dos clases.

[8] Y que ha obtenido muy buenos resultados.

A continuación se verán los métodos automáticos:

3.6. PCA

El análisis de componentes principales(PCA) es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos.

Técnicamente, PCA busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados. Esto convierte un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables sin correlación lineal llamadas componentes principales.

PCA construye una transformación lineal que escoge un nuevo sistema de coordenadas para el conjunto original de datos en el cual la varianza de mayor tamaño del conjunto de datos es capturada en el primer eje (llamado el Primer Componente Principal), la segunda varianza más grande es el segundo eje, y así sucesivamente. Para construir esta transformación lineal debe construirse primero la matriz de covarianza o matriz de coeficientes de correlación. Debido a la simetría de esta matriz existe una base completa de vectores propios de la misma. La transformación que lleva de las antiguas coordenadas a las coordenadas de la nueva base es precisamente la transformación lineal necesaria para reducir la dimensionalidad de datos. Es decir, el primero de los componentes principales es un vector en el que se tiene mayor varianza en los datos y los siguientes CP son los vectores que tienen mayor varianza restrictos a que deben ser ortogonales a los anteriores vectores hallados.

En este caso debido a las ventajas de PCA para reducir la dimensionalidad de un grupo de datos, es que se utilizo para retener aquellas características del conjunto de datos que contribuyen más a su varianza, manteniendo un orden de bajo nivel de los componentes principales e ignorando los de alto nivel. Por lo cual como esos componentes de bajo orden a veces contienen el aspecto "más importante"de esa información, es que se pueden usar como características relevantes.

3.7. LDA

El análisis por discriminante lineal (LDA) se utiliza más comúnmente como técnica de reducción de la dimensionalidad en el paso de preprocesamiento para la clasificación de patrones.. El objetivo es proyectar un conjunto de datos en un espacio de dimensiones inferiores con una buena separabilidad de clase para evitar el sobreajuste ("maldición de la dimensionalidad").

El enfoque LDA en general es muy similar a PCA, pero además de encontrar los ejes componentes que maximizan la varianza de nuestros datos, estamos interesados en los ejes que maximizan la separación entre múltiples clases.

Así que, en pocas palabras, a menudo el objetivo de LDA es proyectar un espacio característico (un conjunto de datos muestras n-dimensionales) sobre un subespacio k, más pequeño mientras se mantiene la información discriminatoria de clase.

En general, la reducción de la dimensionalidad no sólo ayuda a el reducir el ajuste excesivo minimizando el error en la estimación de parámetros ("maldición de la dimensionalidad"), por lo cual también es un buen método para seleccionar características, ya que es un elemento que tiene en cuenta las clases adicionalmente.

4. Selección de características

Para esta parte del sistema, a medida que se iban extrayendo características se iban seleccionando las mejores en un proceso iterativo.

Este proceso consto de varias iteraciones donde en cada una de ellas a medida que se obtenían algunas características nuevas, se hacia la selección de algún clasificador, como veremos mas adelante, para así observar la validez de las características extraídas y ver si lo que se había hecho ayudaba a la separabilidad entre clases.

En concreto se llevaron a cabo dos enfoques distintos, el primero de ellos fue usar las características automáticas por separado de las manuales. En el primer enfoque se fue haciendo el proceso incremental de ir agregando las características, mientras que para el proceso automático se utilizaron las características de cada uno por separado, ya que PCA y LDA son métodos complementarios. Luego se realizaron combinaciones de características en cada iteración.

Primeramente, observaremos los resultados que se llego trabajando con los métodos automáticos de extracción de características. En esta parte observaremos los resultados cualitativos a los que se llego, mas adelante junto con los clasificadores se verán resultados en profundidad sobre la separabilidad entre clases y las distintas precisiones.

Comenzando por PCA, para determinar el número de componentes principales óptimo a utilizar en esta transformación, se calculo cuanta varianza captura cada uno de estos. Es de esperar que la varianza por componente sea diferente según las imagenes que compongan el conjunto sobre el cual se entrenará la transformación; por lo tanto realizando múltiples simulaciones de conjuntos iniciales pretendemos tener una visión estadística, que sea robusta a las posibles diferentes configuraciones iniciales y por lo tanto a prendas desconocidas que puedan ser incorporadas.

De acuerdo a esto, se pudo observar que mantener el 90 % de la varianza en los componentes principales era la mejor opción, haciendo que los clasificadores tuvieran mejor poder de discriminación. Se probaron con distintos valores y empíricamente se llego a este valor para el cual se obtuvieron los primero 43 componentes principales que son los que juntos conservan el 90 % de la varianza.

Luego, cuando se trabajo con LDA se pudo ver que los mejores parámetros que se encontraron para reducir las características fue usar la descomposición por valores propios como solución para resolver el problema, y se utilizo la propiedad que el algoritmo elija la cantidad de vectores con los que quedarse. Al final a lo que se llego es que el mejor resultado es que nueve son los resultantes vectores de características.

Como con esta extracción de características se pretendió reducir la dimensionalidad de nuestro problema. Esto se tradujo en una mejor eficiencia (tanto en precisión como en tiempo de cómputo)de los clasificadores a utilizar, siendo este un efecto deseable a la hora de correr el sistema.

En cuanto a la selección de características de a cuerdo al problema, los resultados cualitativos que se obtuvieron fueron los siguientes:
Como se menciono y al usarse el método iterativo primero se probaron primeramente solamente los momentos estadísticos, se pudo observar que a pesar de lo que se había planteado anteriormente de que eran buenos para separar texturas no tuvieron un impacto grande a la hora de la clasificación, obteniendo peores rendimientos que la selección automática de características.
Luego, se introdujeron el área y el perímetro con el factor de forma para el cual se vio una mejoría en la clasificación de los patrones. Introduciendo mejor separabilidad entre las clases.
Luego habiendo agregado también el template matching, se pudo ver una mejoría en los clasificadores pero a pesar de esto todavía no se llegaba a los resultados de referencia que provee Zalando Research, por lo cual se siguieron buscando mas características que separaran las clases.

Luego de haber estudiado todas las posibilidades de nuevas características se agregaron las FFT con las cuales no se pudo mejorar en gran forma los resultados obtenidos hasta el momento.

Luego se estudio como afectaban en la separabilidad las distintas características. Se vio que para las primeras características, los momentos estadísticos, si se le realizaba un método automático para selección de características, la varianza de un 99,99 % permanecía en solo un vector principal por lo cual esto es una medida de que tan buenos eran las primeras caracetristicas. Cuando se agregaron las siguientes características esto mejoro rotundamente, pero igual se vio que PCA en este caso no era lo adecuado para usar por lo mencionado anteriormente entonces se utilizo solamente LDA ya que tiene en cuenta las clases. Usando este método con las características filtradas nuevamente se llego nuevamente a obtener 9 componentes principales. como se vio anteriormente eso mejoraba la precisión y el tiempo de computo por lo cual fue una gran herramienta.

Luego de haberse acercado lo mas posible al valor de precisión del umbral para los distintos clasificadores se probo mejorar la precisión de los algoritmos introduciendo mas características con la combinación de las características introducidas por los métodos automáticos y aquellos relevados. Para esto se utilizaron las características pasadas por el algoritmo LDA para cada uno de los métodos obteniendo así 18 características para cada imagen. En este caso se pudo ver que se pudo estar a la misma altura de los umbrales emitidos por Zalando research, eligiendo entonces estas como nuestras características finales para la etapa de clasificación final.

5. Clasificación

En este apartado se pretende dar resultados mas profundos que los analizados anteriormente, pero también describir los algoritmos utilizados para la clasificación. Teniendo en cuenta que la media aritmética se acercara al 90 %, que fue el primer objetivo planteado se utilizaron distintos clasificadores, evaluando el desempeño de las características seleccionadas.A los algoritmos que mejor se acercaron al umbral deseado, se les realizo un optimización de los parámetros y estudio de robustez. Luego de probar distintos algoritmos se llego a la conclusión de los que mejor funcionaban eran SVC y Random Forest. Otros de los algoritmos probados fueron Knn, Naive Bayes y Arboles de decisión. Para todos los casos se utilizo cross-validation o split con 80 % entrenamiento y 20 % de validación de las muestras de entrenamiento.

Primero se dará un pantallazo general de los algoritmos para los cuales no se trabajo en profundidad, para luego dar un reporte detallado para los algoritmos de SVC y RandomForest.

5.1. Naive Bayes

Es un método de aprendizaje supervisado basado en el teorema de Bayes, este supone independencia entre pares de muestras. Este clasificador se caracteriza por ser sumamente rápido. Para este método se realiza una tabla comparativa donde se varia el paramento "Kernel" en para la estimación de la densidad de probabilidad, se obtuvieron los siguientes valores de media aritmética: 82 % para LDA y 77 % para PCA.

5.2. KNN

Este método de clasificación supervisada que estima la función de densidad de probabilidad o directamente a la probabilidad posterior de que un elemento x que pertenezca a la clase W_j a partir de un conjunto de entrenamiento. Llamados "lazy learning" ya que la función se aproxima solo locamente y todo los cálculos son diferido a la clasificación. La fase de entrenamiento del algoritmo consiste en almacenar los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento. En la fase de clasificación, la evaluación de un elemento de test, es representado por un vector en el espacio característico. Se calcula la distancia entre los vectores almacenados y el nuevo vector, y se seleccionan los k vecino más cercanos. El nuevo dato es clasificado con la clase que más se repite en los vectores seleccionados.

La mejor elección de k depende fundamentalmente de los datos; generalmente, valores grandes de k reducen el efecto de ruido en la clasificación, pero crean límites entre clases parecidas. Un buen k puede ser seleccionado mediante una optimización de uso. Para este método se realizaron diferentes variaciones de k llegando a los valores con mejor valor de cross-validation de 85 % para las características con LDA y 84 % para las características con PCA. Cabe aclarar que para la selección de clasificadores, se usaron solamente las características por PCA y LDA.

Se puede observar como mejora los aciertos por clase en comparación con lo obtenido con NaiveBayes, Para el k elegido la performance aumenta un 1 %, por lo que se obtiene un mejor clasificador de los datos. Al aplicar los selectores de características fueron que se vieron mejorías de la performace del método.

5.3. Arboles de decisión

Otro de los clasificadores que se usaron fueron el calsificador de arbol C4.5 que tiene parametros confienceFactor (factor de confianza utilizado en la poda) y el minNumObj (cantidad mínima de patrones por nodo para crear un hoja) realiza el podado del árbol. Este método genera arboles de decisión para clasificación a partir de datos de entrenamiento, es fácil de interpretar, de rápida clasificación y se le puede incluir conocimiento a priori. Se corrió el arbol en python con distintos métodos de podado, obteniendo los siguientes como mejores valore obtenidos en media, se pudo alcanzar un 85 % en media probando así que se tiene un clasificador parecido a KNN en cuanto a precisión.

5.4. Random Forest

Random Forest es un método de aprendizaje por ensamble para la clasificación, regresión y otras tareas, que funcionan construyendo una multitud de árboles de decisión en el momento del entrenamiento y produciendo a la salida la clasificación según la clase que se obtuvo más veces a la salida de los distintos árboles. Random Forest corrige el hábito de los árboles de decisión de adaptarse a su conjunto de entrenamiento.

El algoritmo de aprendizaje para los bosques al azar aplica la técnica general de bootstrap aggregating, o bagging.Este procedimiento de bootstrapping conduce a un mejor rendimiento del modelo porque disminuye la varianza del modelo, sin aumentar el sesgo. Esto significa que aunque las predicciones de un solo árbol son altamente sensibles al ruido en su conjunto de entrenamiento, el promedio de muchos árboles no lo es, siempre y cuando los árboles no estén correlacionados. El simple hecho de entrenar muchos árboles en un solo conjunto de entrenamiento daría árboles fuertemente correlacionados (o incluso el mismo árbol muchas veces, si el algoritmo de entrenamiento es determinista); el muestreo bootstrap es una forma de des-coincidir con los árboles mostrándoles diferentes conjuntos de entrenamiento.

Los bosques aleatorios se diferencian en una sola forma de este esquema general de boostrap y bagging: utilizan un algoritmo de aprendizaje de árboles modificado que selecciona, en cada división del proceso de aprendizaje, un subconjunto aleatorio de las características. Este proceso a veces se llama "embolsado de características". La razón para hacer esto es la correlación de los árboles en una muestra ordinaria de bootstrap: si una o unas pocas características son predictores muy fuertes para la variable de respuesta (salida objetivo), estas características se seleccionarán en muchos de los árboles B, haciendo que se correlacionen. Por lo cual hacer esto previene este comportamiento mejorando los algoritmos de arboles de decisión.

5.4.1. Resultados

Como se menciono anteriormente se verán los distintos pasos del proceso. Con los métodos automáticos se obtuvieron valores en media con cross-validation de 86 % para LDA y 83 % para PCA.

De estos dos se eligió entonces para trabajar LDA y así poder ver como era la separabilidad entre clases. Podemos ver en la figura 8 la matriz de confusión.

```
Number of mislabeled points out of a total 6000 points : 826
0.862333333333
```

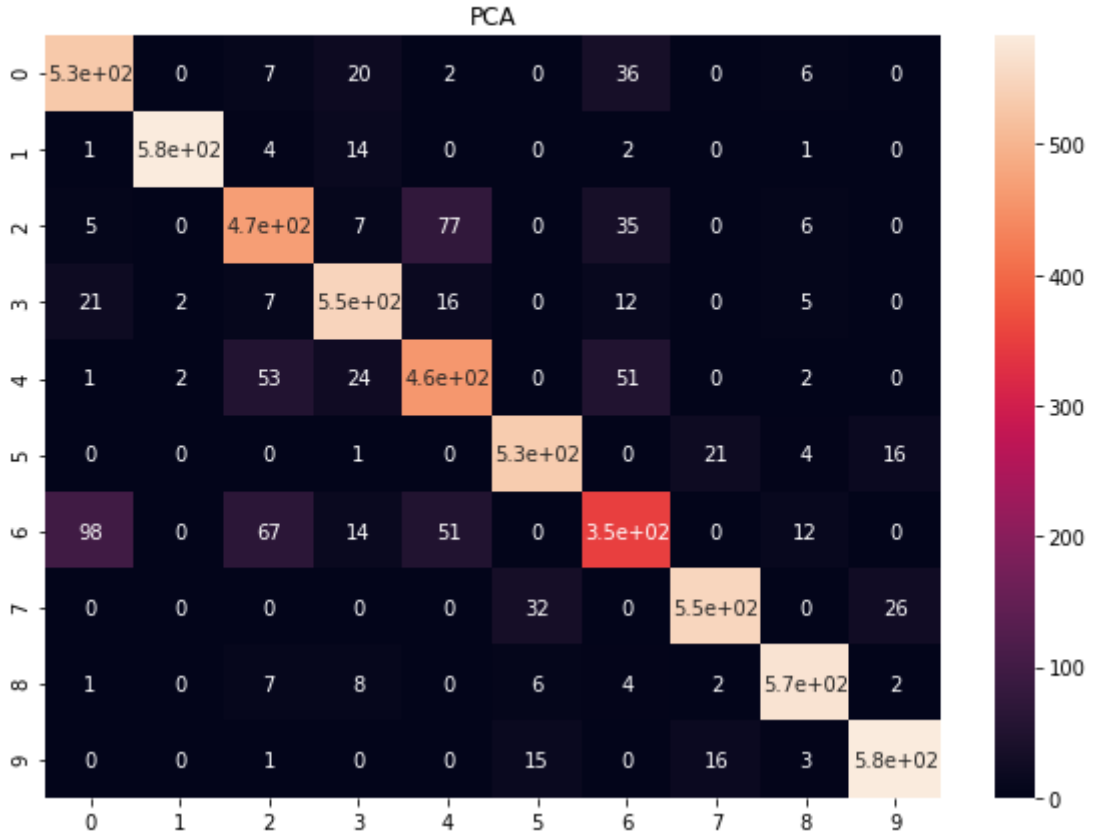


Figura 8: Matriz de confusión de Random Forest con LDA

De esta matriz se puede extraer algunos resultados interesantes los cuales fueron utilizados luego para la creación de las características futuras.

Se pueden ver que las clases que mas se confunden entre si son la $0, 6, 2, 4$. Como era de esperar tres de estas clases son categorías muy parecidas ya que son las camisetas, los pullovers y las camisas. Estas clases pueden tener todas una forma muy parecida, algunas tienen mangas largas y otras mangas cortas pero en general las formas son muy parecidas, y hasta a veces es difícil diferenciar entre clases para un ser humano[9]. En el conjunto de estas cuatro clases tenemos a los bolsos que esta si no tiene mucho sentido en esta comparación. Otra de las clases que se confundieron fue la de de las sandalias con zapatillas donde otra vez podemos ver el parecido en forma.
Es por esto que se propusieron las técnicas manuales, como por ejemplo las FFT para mitigar las diferencias entre energia dado la cantidad de pixeles por arriba del umbral en las distintas zonas, y asi poder lograr mas separabilidad. A partir de esto fue que se crearon las nuevas características y se obtuvieron los siguientes resultados.

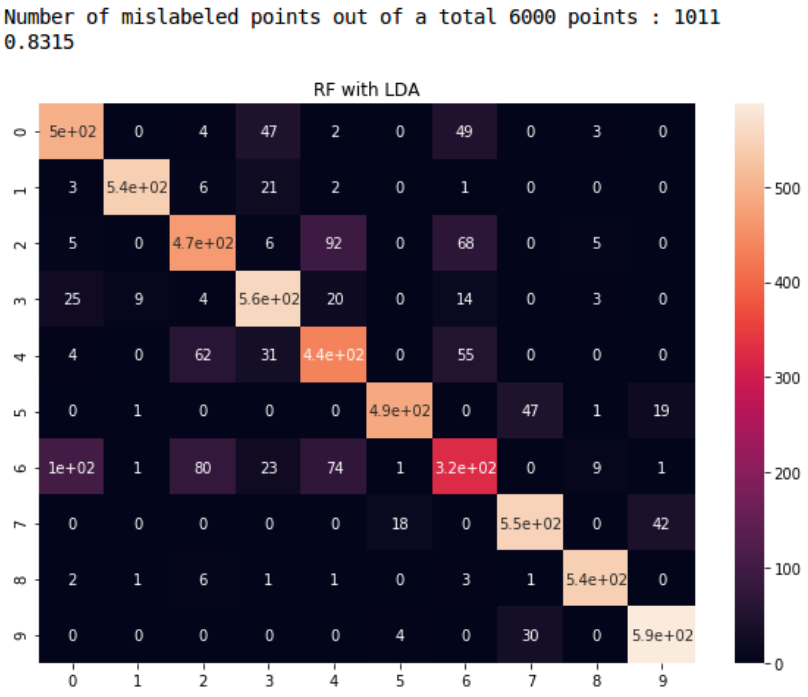


Figura 9: Matriz de confusión de Random Forest con Momentos, Factor de Forma y Template Matching

[9]Que según Richard Duda[1], el ser humano es el mejor sistema de reconocimiento de patrones que existe en el mundo.

```
Number of mislabeled points out of a total 6000 points : 981
0.8365
```

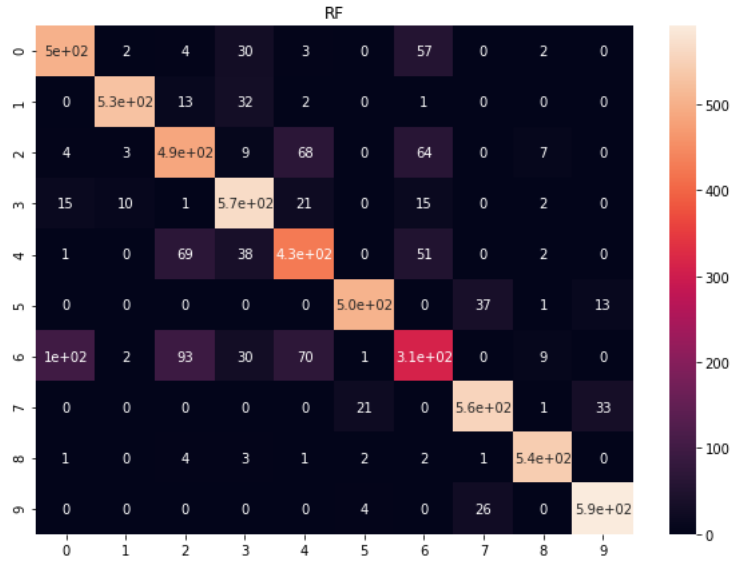


Figura 10: Matriz de confusión de Random Forest con FFT's

En la figura 9 y 10 podemos ver los resultados luego de haber probado primeramente con las características de Momentos, Factor de Forma y Template Matching y luego añadiendo las FFT's. Podemos ver que no se acercan a los resultados que se esperaban, estando lejos del umbral al cual se quería llegar. A pesar de esto se ve que las FFT's introdujeron una leve mejoría en la clasificación aunque no como se esperaba que mejorara la separación de las clases previamente dichas para la clasificación. Podemos ver nuevamente que se siguen confundiendo las clases que antes se confundían con los métodos automáticos.

En un ultimo intento de poder mejorar la clasificación y usar lo mejor de los dos enfoques, como se dijo antes se propuso combinar los dos set de características obteniendo los siguientes resultados:

```
Number of mislabeled points out of a total 6000 points : 745
0.875833333333
```

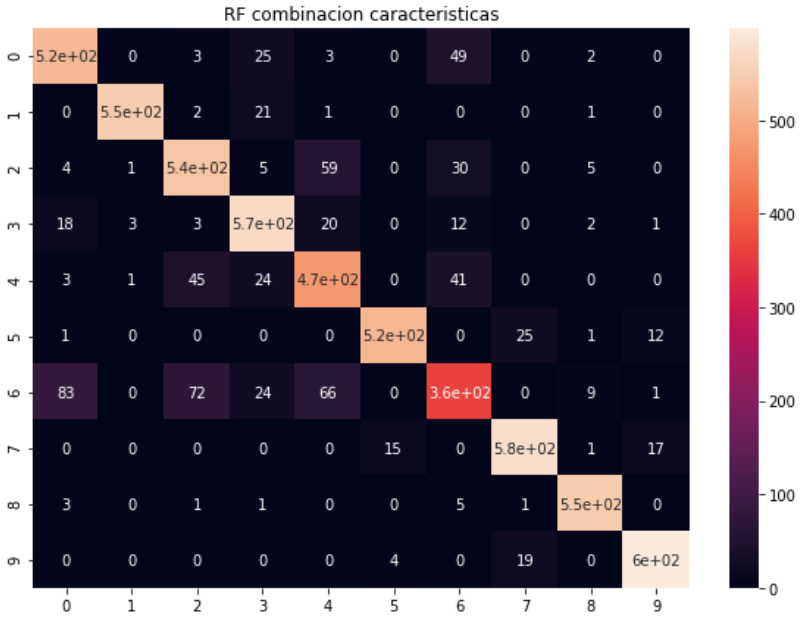


Figura 11: Matriz de confusión de Random Forest con combinacion de características

Para este caso en media usando CV se obtuvo un 87 %. Se puede ver que mejoro la clasificación utilizando

la combinación de ambos set de características. Y vemos que algunas de las clases que se confundían mejoraron a la hora de la clasificación, obteniendo una mejor discriminación.

5.5. SVC

La idea general de SVM es que busca un hiperplano que separe de forma óptima a los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior, esto es encontrar un espacio donde los datos sean linealmente separables, lo que permite contemplar el caso no linealmente separable.

En ese concepto de 'separación óptima' es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso también a veces se les conoce a las SVM como clasificadores de margen máximo. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado en el caso binario.

Para la extensión al caso multiclase se puede hacer de varias maneras, dos de ellas son:

- Uno contra todos: M clasificadores binarios, una clase contra el resto de las otras en cada uno de los clasificadores binarios.
- Clasificación por pares: se consideran $c(c-1)/2$ pares de clases, y para cada una se diseña un clasificador. Para clasificar una nueva muestra se pasa por todos los clasificadores, y se asigna a la clase más votada.

Otra de las variantes que tiene esta tecnica con respecto a SVM comun es que tenemos el parametro C, este indica a la optimización SVM cuánto se desea evitar clasificar erróneamente cada ejemplo de entrenamiento. Para los valores grandes de C, la optimización elegirá un hiperplano de margen más pequeño si ese hiperplano hace un mejor trabajo de conseguir todos los puntos de entrenamiento clasificados correctamente. Por el contrario, un valor muy pequeño de C hará que el optimizador busque un hiperplano separador de margen más grande, incluso si ese hiperplano clasifica erróneamente más puntos. Para valores muy pequeños de C, debería obtener ejemplos mal clasificados, a menudo incluso si sus datos de entrenamiento son linealmente separables.

5.5.1. Resultados

También se realizo el proceso incremental con este clasificador. Luego de haber probado que el mejor conjunto de características en varios clasificadores fue la combinación de LDA para el conjunto de pixeles junto con LDA para las características manuales. Se procedió a entrenar con este conjuntos de características SVC como final.

En el proceso se pudo ver como se ve reflejado en en la figura 12 que se obtuvo valores similares que con el método de Random Forest para las características de LDA.

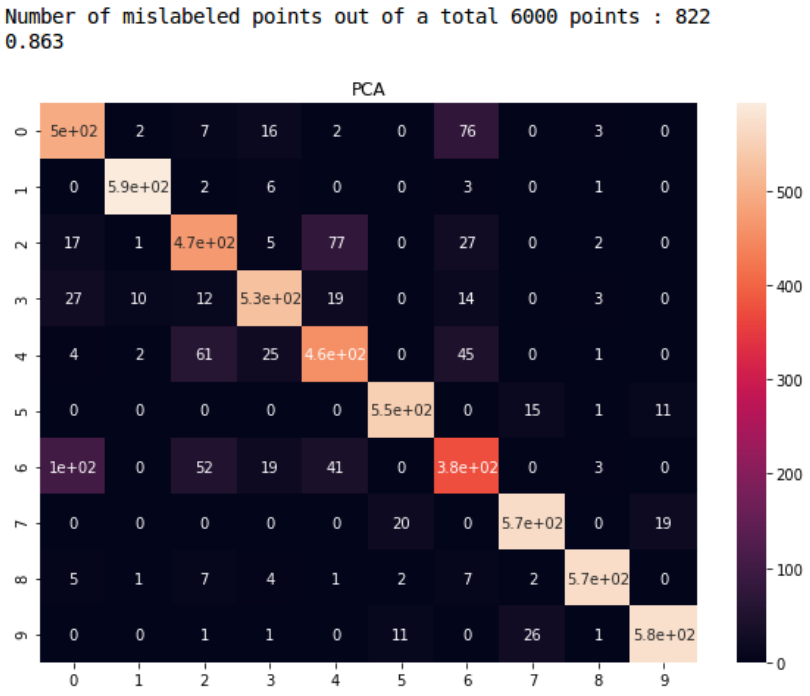


Figura 12: Matriz de confusión de SVC con LDA

Aqui nuevamente se puede apreciar la confusión entre las mismas clases que se mencionaron anteriormente, por lo cual se intento evaluar si nuestra técnica ayudo para este clasificador a separar las clases.

Luego de haber hecho la selección de las características como en la sección anterior se llego al 87,5 % en media. Y se pudo apreciar las mismas mejoras que se mencionaron antes en cuanto a la desconfusión de las clases que eran clasificadas erróneamente. Se puede ver la matriz de confusión en la figura **??**.

Otro de las cosas que ayudaron a este algoritmo a mejorar en media fue el resultado de la estandarización, esto es que las características fueron redimensionadas de manera que tendrán las propiedades de una distribución normal estándar con media cero y varianza 1.
Esto es porque SVM asume que los datos con los que trabaja están en un rango estándar, generalmente de 0 a 1, o de -1 a 1 (aproximadamente). Entonces, la normalización de los vectores de características antes de alimentarlos a la SVM es muy importante. (Esto a menudo se denomina blanqueamiento, aunque existen diferentes tipos de blanqueamiento.) Desea asegurarse de que para cada dimensión, los valores se ajustan a escala aproximadamente dentro de este rango. De lo contrario, si, por ejemplo, la dimensión 1 es de 0-1000 y la dimensión 2 es de 0-1.2, entonces la dimensión 1 es mucho más importante que la dimensión 2, lo que sesgará los resultados.

Despues de todos estos resultados, como este metodo en media se comporto un poco mejor que Random Forest, este fue el algoritmo elegido como mejor para realizar las pruebas de test.

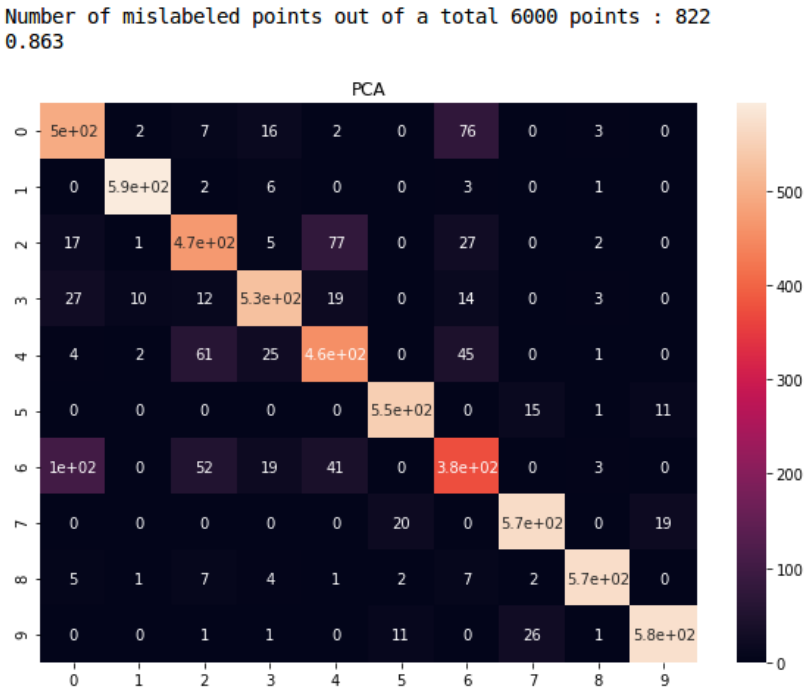


Figura 13: Matriz de confusión de SVC con combinación de características

5.6. Otros metodos

El otro metodo que se utilizo para clasificación como muestra de una forma distinta de clasificación fue usar las redes neuronales. Para esto se uso la biblioteca PyTorch que implementa todas las funciones para aplicar una red neuronal. Se vio y estudio que los algoritmos basados en Deep Learning como lo son las redes neuronales tienen en teoría la capacidad de aproximar cualquier función. Por lo cual es una buen primera hipotesis para poder aproximar la funcion de discriminacion. Se probo usar esto ya que la diferencia con algoritmos usando características diseñadas a mano tienen:

- Performance limitada al escoger modelo
- Requieren conocimiento experto

Se fue escepticos al contar con no mucha cantidad de datos, ya que estas redes funcionan si la cantidad de datos que le entran a la red son muchas. A pesar de esto la red tuvo una buena performance.

Se eligió una red de convolucion, estas consisten en múltiples capas de filtros convolucionales de una o mas dimensiones. Después de cada capa, por lo general se añade una función para realizar un mapeo causal no-lineal.
Como redes de clasificación, al principio se encuentra la fase de extracción de características, compuesta de neuronas convolucionales y de reducción de muestreo. Al final de la red se encuentran neuronas de perceptron sencillas para realizar la clasificación final sobre las características extraídas. Esta fase se compone de capas alternas de neuronas convolucionales y neuronas de reducción de muestreo. Según progresan los datos a lo largo de esta fase, se disminuye su dimensionalidad, siendo las neuronas en capas lejanas mucho menos sensibles a perturbaciones en los datos de entrada, pero al mismo tiempo siendo estas activadas por características cada vez más complejas.
Los ditintos tipos de capas que se utilizaron fueron:

- Convolutiva (feature map): las neuronas están conectadas localmente y comparten pesos; extraen características locales y son invariantes respecto al espacio.
- Agrupación/Submuestreo (pooling): condensan salidas de capas anteriores, aplicando submuestreo y alguna función de reducción (máximo, k-máximo, media, etc.); reducen el espacio.
- Completamente conectadas: al estilo clásico, normalmente son la última capa

La arquitectura que se uso es la mostrada en la figura 14:

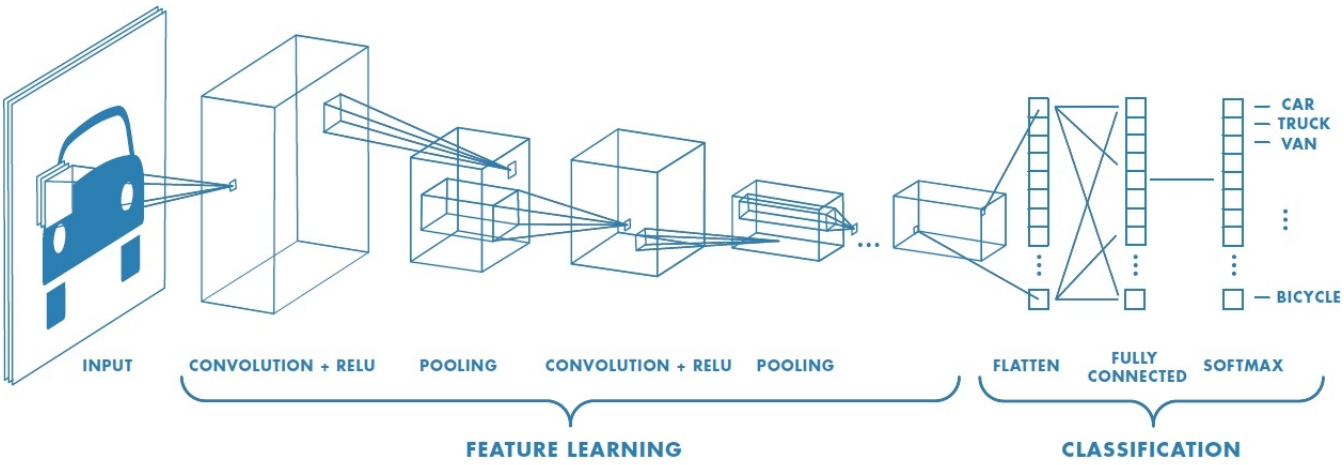


Figura 14: Arquitectura de Red de convolución

El metodo para el entrenamiento fue distinto a el antes visto donde se tiene un conjunto de entrenamiento: El cual permite obtener los pesos de una red dada, Ajustar los parámetros de la red. Un conjunto de validación: que permite comparar el desempeño de distintas arquitecturas de redy volver a hacer ajuste de hiperparámetros. Por ultimo un conjunto de prueba permite estimar el desempeño de la red en datos futuros, el mejor modelo se debe probar en esta parte.

El resultado de utilizar este método fue una precisión del 94 %

6. Conclusiones

Luego de haber seleccionado como algoritmo final SVC, ya que la red de convolución no tenia en cuenta las características que habíamos planteado, se obtuvo en test un 80 % de precisión al clasificar los datos. Esto quiere decir que no se tuvo una buena generalización. Se estudiaron varios de los posibles orígenes del problema y se concluyo que hacían falta características que separaran mejor las clases para obtener mejores resultado.

En cuanto a nuestro enfoque, la extracción de características: podemos decir que no se llegó a un nivel lo suficientemente bueno cómo para competir con una red de convolución o incluso con el nivel de clasificación de una persona normal (∼84 % según el GitHub de FMNIST.[10]). Por lo que no nos satisface el resultado de la clasificación con las características halladas. Dejándonos como conclusión que la base de datos debe ser pequeña y los priors que tiene el humano sobre cómo clasificar las muestras debe ser preciso y capaz de llevarse a una característica concreta(en código) para que sirva este enfoque, de otra forma los métodos automáticos darán los mismos (o mejores) resultados que la extracción de características impuestas por quién implemente el sistema[11].

También se pudo observar que para la red de convolución la precisión fue 91 %. Se puede concluir que este método de aprendizaje automático es mas efectivo que el método que utilizamos en la parte anterior. A pesar de pensar que no iban a ser los datos suficientes para un buen entrenamiento de la red podemos concluir que con la cantidad de datos que teníamos fue suficiente para un buen entrenamiento, que luego se viera reflejado en una buena generalización para los datos de test. Como ventaja se puede ver que no es necesario de conocimiento previo del problema para poder hacer la clasificación, sino que es la red la que aprende las características del problema.

Bibliografía

[1] Richard Duda
Pattern Classification. 2a Edición.

[10] https://github.com/zalandoresearch/fashion-mnist/

[11] A menos que se tengan unas características definidas que se sepa que sirven siempre para clasificar bastante bien, como lo es el caso de textura. Un ejemplo de como funciona un sistema conocido de reconocimiento de patrones para textura es en Fiji(Un programa OpenSource de procesamiento de imágenes muy utilizado en la medicina, biología y astronomía), con el paquete de "Trainable weka segmentation", una adaptación de paquetes de weka con extracción de características y segmentación de textura, que utiliza además de momentos, filtros de sobel, y laplacianos y otros filtros convolutivos para obtener características a partir de las imágenes.

Wiley-Blackwell, 2000.

[2] Rafael González
Digital Image Processing. 3a Edición.
Pearson Prentice Hall, 2008.

[3] ZalandoResearch
https://github.com/zalandoresearch/fashion-mnist/

[4] Imagej - Fiji
https://imagej.net/Trainable_Weka_Segmentation