Universidad Zaragoza

Trabajo Fin de Grado

Sistema de detección de emociones a partir de secuencias de audio, vídeo y mapa de profundidad

Autor

Mario Subías Pérez

Director

Antonio Miguel Artiaga

Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza

2015

Gracias a mi tutor Antonio,
por su tiempo, dedicación y ayuda,
sin él no hubiera sido posible.

Gracias a mis compañeros,
y amigos, los 'esmochaos',
por apoyarme durante estos años,
ha sido un gustazo encontraos.

Gracias a mis amigos de siempre,
por seguir estando ahí a pesar
del paso del tiempo como
si nada hubiera cambiado.

Y gracias a mis padres y hermano,
por aguantar lo bueno y lo malo
y apoyarme en todo momento.

SISTEMA DE DETECCIÓN DE EMOCIONES A PARTIR DE SECUENCIAS DE AUDIO, VÍDEO Y MAPA DE PROFUNDIDAD

RESUMEN

El objetivo del presente proyecto es el de crear un sistema de reconocimiento de emociones mediante secuencias de audio, vídeo y mapa de profundidad utilizando técnicas de Machine Learning o aprendizaje automático.

Para dicho fin, se procedió a la realización de un curso[1] sobre el tema con el fin de comprender qué metodologías utilizar y con qué fines. En este curso se veían los aspectos más teóricos de esta rama, que si bien no eran imprescindibles para la utilización de dichas técnicas, daban una visión globalizada sobre cómo proceder y qué tipo de clasificadores existían. Además, se realizó un estudio del estado del arte sobre las bases de datos actuales y artículos académicos de índole similar al tema a tratar, con el fin de conocer los procedimientos que más éxito tenían.

Para la captación de dichas secuencias, se decidió emplear en principio la herramienta de Kinect[2], pues está dotada de sensores capaces de

[1] https://es.coursera.org/learn/machine-learning

[2] http://www.kinectfordevelopers.com/

detectar el audio, la imagen y el mapa de profundidad. Sin embargo, a medida que avanzó el proyecto, se dio la constancia de que el sensor de profundidad no provee de suficiente resolución para el problema que pretendíamos abordar, por lo que se decidió no centrarse en esa parte. Una vez reconocido este problema y con vistas del tiempo limitado que tiene el trabajo de fin de grado, se optó por renunciar a la creación de la base de datos propia, pues ya no estaba equipada con el factor de novedad que suponía el sensor de profundidad.

Ya iniciado el punto de partida y con un estudio del estado del arte previo, se procedió a la extracción de características. Para la parte de la imagen se buscó parámetros que fueran relevantes en la detección de emociones. Se utilizó puntos característicos faciales determinados por algoritmos como Active Appareance Model [1] o Active Shape Model[2], que surten de un mapa de puntos localizados de la cara indicando la posición de las distintas partes de la cara. En cuanto al audio, se utilizaron características utilizadas habitualmente en procesado de voz como pueden ser el pitch para reconocer con qué tono está hablando la persona o indagando más las características frecuenciales localizadas, el cepstrum localizado o la predicción lineal localizada (lpc)[3].

Una vez obtenidas estas características, se procedió a utilizar algoritmos de aprendizaje automático que permiten la generalización de estos datos para la clasificación de futuras muestras no vistas previamente. Estos algoritmos aprenden de los datos entregados y crean un nuevo algoritmo en función de lo relevantes que son estas características con el fin de realizar una correcta clasificación para una nueva muestra dada.

ÍNDICE

1. INTRODUCCIÓN

1.1. OBJETIVOS Y MOTIVACIÓN

El objetivo principal de este trabajo de fin de grado fue la elaboración de un sistema de clasificación de emociones utilizando características extraídas de la imagen, del audio y de la profundidad mediante algoritmos de aprendizaje automático.

Actualmente la mayoría de soluciones para abordar el problema de la detección de emociones se centran tan sólo en un campo, en imagen o en audio. Por ello, una de las motivaciones para la realización de este trabajo fue el crear un clasificador que pudiera combinar varias características de información provenientes de distintas fuentes, en este caso la imagen, el audio y el mapa de profundidad. Se pretendía, también, crear un sistema que pudiera ser utilizado de la manera más estandarizada posible. Desde este punto de vista, el dispositivo kinect resultaba perfecto al ser asequible en cuanto a carácter económico y contar con suficientes sensores para la extracción de características que se utilizarían para realizar la clasificación.

1.2. ESTADO DEL ARTE

Una de las primeras fases de este proyecto fue la investigación del área de conocimiento en la que se iba trabajar, es decir, localizar aquellos aspectos más relevantes de investigaciones similares que se habían llevado a cabo en otras universidades.

La búsqueda consistió en encontrar bases de datos de carácter similar al deseado. Se encontraron diversas bases de datos [4] [5] [6] [7] [8] [9] [10] [11] [12] con un número de emociones variable, optando finalmente por intentar clasificar las emociones que la mayoría de los expertos reconocen como las emociones básicas: alegría, miedo, tristeza, asco, enfado, sorpresa y neutral (o sin expresión). En cuanto a la imagen, se encontraron varias bases de datos muy amplias, diferenciando por sexos y por nacionalidades. También se pudieron ver que algunas tomaban fotografías extra para estudiar más casos, como pueden ser el ángulo respecto al cual está fotografiado el sujeto o la cantidad de luminosidad presente en las fotografías tomadas en la base de datos. Aunque finalmente, estos supuestos no se han tenido en cuenta con el fin de acotar el proyecto e intentar centrarse en la mayor cantidad de partes posible.

Ya en el audio, se encontraron algunas bases de datos referidas a emociones no todo lo ideales que pudieran ser para este proyecto. Algunas contaban con demasiadas emociones y se encontraban en un

formato difícil de utilizar en las herramientas a emplear, por ejemplo, que toda la base de datos estaba tomada en un solo archivo de audio. Por lo que finalmente se utilizó una base de datos que, a pesar de no contar con las mismas emociones que en el campo de la imagen, nos ofrecía una gama significativa: alegría, tristeza, miedo, enfado y neutral.

Además, de forma paralela, se investigó algunas publicaciones científicas relacionadas con el tema a tratar para ver qué tipo de tecnologías habían usado. Para la detección de bordes en la imagen era muy común ver el filtro de Gabor[8] como herramienta a utilizar en el preprocesado de la imagen. Los filtros de Gabor detectan diferencias entre áreas en la imagen con píxeles altamente correlados entre sí. Con ellos, se puede calcular un promedio para parametrizar la cara. Además, este filtro permite ajustarse en función de la orientación de la cara. Otros filtros de este estilo que se utilizaron en otros proyectos fueron el Lambertiano o el edge[3]. Con este tipo de filtros se trabaja con todo el conjunto de la imagen como tal, pixel a pixel. Esto, además de ser más costoso computacionalmente, puede no ser tan preciso para el problema que queremos abordar. Otras técnicas utilizadas en proyectos de índole similar, fueron las de Active Shape Model[13][14] y Active Appareance Model. Estos modelos buscan localizar las coordenadas de los puntos clave de la cara.

[3] http://es.mathworks.com/help/images/ref/edge.html

También se vio que las características frecuenciales del audio contienen información acerca de las emociones mostradas. Uno de los parámetros más utilizados para la detección de emociones es el pitch[15], con el cual se puede ver que el tono de la voz está relacionado con la emoción expresada. También era común el uso de los MFCC (Mel Frequency Cepstral Coefficients)[16], un método de extracción de características del habla más complejo que el pitch. Además se vieron algunos proyectos de clasificación de emociones y sus maneras de proceder[17] en cuanto al audio re refiere.

1.3. HERRAMIENTAS UTILIZADAS

Para la realización de este trabajo, se utilizaron los siguientes programas y herramientas:

I. Kinect

Como herramienta hardware de captador de datos se utilizó la Kinect. Este dispositivo desarrollado por Microsoft consta de un sensor de profundidad, una cámara RGB y un array de micrófonos colocados a lo largo del dispositivo. Para poder integrarla con el mismo ordenador, se descargó el Toolkit[4] de desarrollo de kinect en su versión 1.8. De esta forma, programas como Matlab podían manipular la herramienta de forma directa.

[4] http://www.kinectfordevelopers.com/

II. Matlab

Para el procesado de los datos, así como para las primeras pruebas, se utilizó la herramienta de software matemático de Matlab. En ella se encontró la posibilidad de integrar algoritmos de imagen y sonido útiles para los fines a desarrollar. La versión utilizada fue la R2014a, pues algunos de los algoritmos utilizados requerían de las últimas versiones de Matlab para ser utilizados en la plataforma. La razón de utilizar este tipo de software y no otro es el conocimiento del mismo ya adquirido en la carrera, su potencia de cálculo y la gran comunidad que hay en la red del mismo.

III. Weka

Este programa de software gratuito se utilizó para la parte de aprendizaje automático. El programa lee los datos mediante archivos de texto como pueden ser los .arff o .csv y los interpreta ordenando estos valores según la clase a la que pertenezcan. La herramienta de clasificación ofrece la posibilidad de utilizar un gran número de clasificadores gracias la implementación de código abierto que proporciona, además permite modificar los parámetros más destacables de los mismos de una forma cómoda y la visualización de las características. La librería que se utilizó finalmente para el sistema, es libsvm[5].

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

1.4. ORGANIZACIÓN DE LA MEMORIA

La memoria cuenta con una estructura dividida en cuatro partes:

- Apartado 2: se analizan los métodos utilizados para la extracción de características empleados.
- Apartado 3: se explican las técnicas de aprendizaje automático utilizadas y cómo se han utilizado, así como las bases de datos utilizadas.
- Apartado 4: se detalla cómo ha sido la parte de implementación del sistema.
- Apartado 5: se recogen las conclusiones obtenidas con pruebas en sujetos reales.

2. EXTRACCIÓN DE CARACTERÍSTICAS

Para realizar un sistema de clasificación, la primera fase consiste en elegir las características que se van a utilizar, pues deben tener información suficiente para poder distinguir entre las clases elegidas en la clasificación. A continuación se exponen las técnicas utilizadas para esta extracción de características. Las secciones nombradas como Viola Jones y Project-Out Cascaded Regression (PO_CR) pertenecen al campo de la imagen, mientras que Parámetros básicos de audio, MFCC y LPC están enfocados a la parte de sonido.

2.1. VIOLA JONES

El algoritmo de Viola Jones[18] nos permite realizar detección de objetos con un coste computacional muy bajo. En nuestro caso, este algoritmo se emplea para la detección de caras y, posteriormente, selección de distintos atributos faciales, como pueden ser la zona de los ojos o de la boca.

La información de la imagen es procesada en escala de grises y no se emplea directamente la imagen, sino lo que se conoce como el método de la imagen integral. Este método es capaz de evaluar la suma (integral) del valor de los píxeles de una región de una forma muy eficiente,

permitiendo su operación en tiempo real. La suma del valor de los píxeles encerrados en la región definida por un rectángulo ABCD, se puede calcular sabiendo la suma de los valores de los píxeles contenidos en los rectángulos correspondientes entre cada punto y el origen de la imagen. Como se puede apreciar en la figura 2.1.1, con tan sólo la localización de 4 puntos y mediante sumas y restas, podemos determinar la suma de los valores de los píxeles encerrados en la región deseada. El único coste que tiene este método es el cálculo inicial de las sumas hasta el origen.

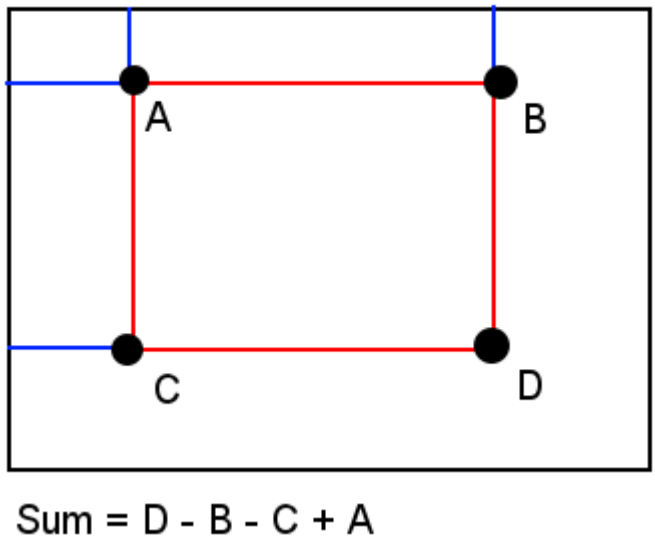

Figura 2.1.1: Cálculo de píxeles mediante imagen integral

Esto permite al algoritmo recorrer la imagen de una forma rápida y eficiente, y localizar las caras independientemente de lo alejada o cercana que esté en relación al número de píxeles totales, así como la posibilidad de encontrar varias caras en una misma imagen.

El algoritmo trata de localizar distintas características, conocidas como características Haar[19], que buscan niveles de luminosidad distribuidos de una forma concreta en la imagen. Por ejemplo, la región de los ojos será más oscura que la región de la nariz o que la zona de las mejillas. En la figura 2.1.2 se muestran algunos ejemplos de las características Haar.

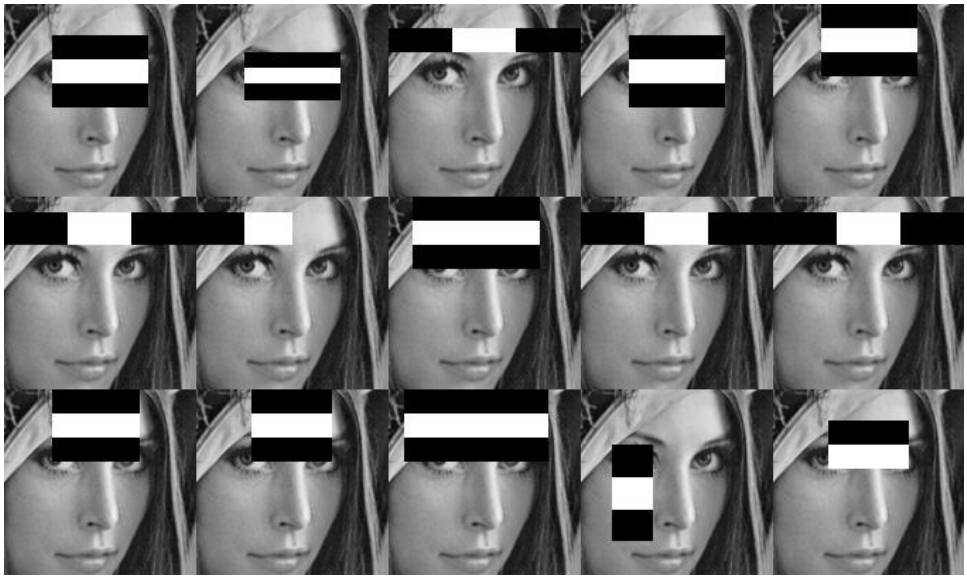

Figura 2.1.2: Ejemplos de características Haar en la imagen de Lena

El algoritmo de Viola Jones utiliza una modificación del algoritmo AdaBoost[20] para seleccionar las características y entrenar los clasificadores posteriores. El algoritmo AdaBoost es un algoritmo de aprendizaje automático capaz de construir un clasificador fuerte mediante una combinación ponderada de clasificadores débiles. En el caso de Viola Jones, se proporcionan al algoritmo de AdaBoost ejemplos de imágenes con las etiquetas correspondientes: cero si no hay cara y uno si la hay.

Para utilizar este algoritmo con una imagen nueva, Viola Jones utiliza un sistema de clasificadores en cascada, en los cual cada uno de los clasificadores está dotado de un conjunto de características visuales. El principio básico del algoritmo de detección de Viola Jones es aplicar el detector muchas veces con la misma imagen (cada vez con un nuevo tamaño). Cada clasificador determinará si la región tomada es cara o no, de tal forma que si un clasificador decide que la región no es cara, el algoritmo se detendrá y probará con una nueva región. El esquema del clasificador en cascada a seguir es el siguiente:

Figura 2.1.3: Esquema del algoritmo Adaboost

2.2. PROJECT-OUT CASCADED REGRESSION (PO_CR)

Para la clasificación de emociones, en este proyecto se ha pretendido trabajar mediante características de más alto nivel, como son las coordenadas de ciertos puntos de interés (landmarks) en la imagen, en lugar de con los píxeles directamente, pues el tiempo de cómputo es menor y pueden proporcionar información más fácil de procesar por un clasificador si nuestro objetivo es clasificar emociones. El algoritmo de PO_CR[21] se basa en los modelos estadísticos de computer vision ASM (Active Shape Model)[2] y AAM (Active Appearance Model)[1] mediante

los cuales se buscan puntos de referencia de la cara o FCP (facial characteristic points) que nos permitirán determinar la forma de la boca, ojos o nariz.

Una vez tenemos una imagen en la cual queremos localizar los FCPs, el algoritmo lo primero que hace es colocar estos puntos en una situación aleatoria, manteniendo la forma típica de una cara aunque, probablemente, mal situada respecto al rostro original. El algoritmo parte de unos puntos iniciales y mediante un algoritmo iterativo va desplazando los puntos a los píxeles próximos más probables en función de la textura y forma actual y buscada. En los anexos A y B se puede encontrar un ejemplo de funcionamiento, así como las ecuaciones matemáticas en las que se basa el algoritmo.

Previamente, este algoritmo debe ser entrenado con una base de datos que contenga imágenes relacionadas con sus puntos faciales. En este caso, en la documentación del toolkit[6] sugería utilizar las bases de datos AFW, Helen, LFPW e iBug[13][14] aunque partan de un total de 68 puntos, el algoritmo utiliza los 49 principales de la cara. De tal forma que para una imagen nueva, devuelve 49 FCPs distribuidos por la cara como se muestra en la figura 2.2.1. Por lo general, el software de ASM se distribuye sin modelos previos, por lo que se deben entrenar los modelos partiendo de cero.

[6] http://ibug.doc.ic.ac.uk/resources/facial-point-annotations/

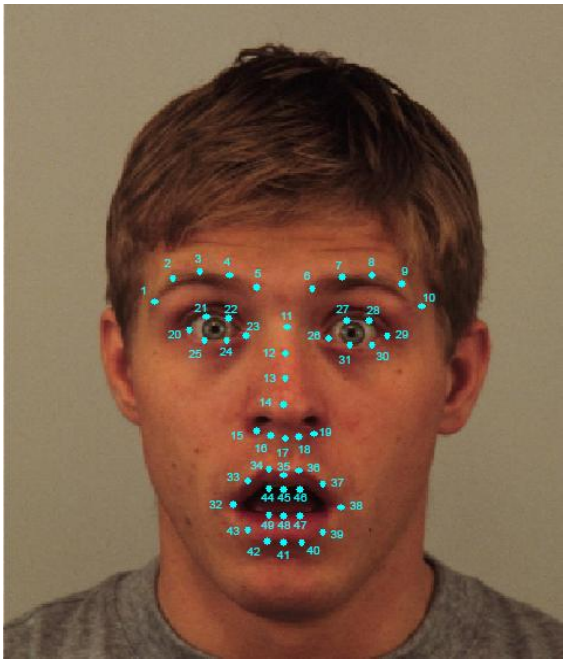

Figura 2.2.1: Ejemplo del algoritmo PO_CR con los puntos obtenidos señalados

Uno de los inconvenientes de este algoritmo es que requiere de una buena inicialización para que esta iteración se produzca con éxito. Por ejemplo, en algunas pruebas se han encontrado errores en la localización de las cejas, pues el algoritmo las situaba a la altura del pelo por tener una textura similar, al encontrarlo antes que las propias cejas, el algoritmo lo daba como bueno siendo que se trataba de un error.

Para evitar este problema, en este proyecto, se ha optado por detectar la región de la cara mediante el algoritmo de Viola Jones y proceder a inicializar los FCPs dentro de esta región. Utilizando Viola Jones nos aseguramos de que la inicialización se produce de forma correcta y por tanto el algoritmo puede proceder con mayor probabilidad de éxito.

2.3. PARÁMETROS BÁSICOS

A la hora de extraer características que nos permitan distinguir emociones, en una primera instancia se probó con algunos parámetros básicos, como son el pitch y la señal muestreada. En el estudio del estado del arte previo que se realizó, se encontraron proyectos que utilizaban características frecuenciales para la clasificación de emociones, uno de los parámetros más utilizados era el pitch.

El pitch es la frecuencia fundamental en cada instante de tiempo. Para la obtención del mismo se utilizó el código generado en Matlab extraído de la web mathworks[22]. Como primera idea, podemos suponer que emociones como la tristeza tendrán unos rangos de frecuencias más bajos que otras como la alegría.

Para caracterizar el pitch con características con las que entrenar el algoritmo de aprendizaje automático, se procedió a realizar un análisis estadístico de la señal devuelta por el algoritmo. Se utilizó la media y la varianza de la señal que marcaba la frecuencia fundamental a través del tiempo.

Otra característica de interés es la energía localizada, para calcularla se procedió a realizar un enventanado de las muestras de 30 ms con un desplazamiento de 10 ms para trabajar con muestras lo más

estacionarias posibles. De estas nuevas muestras se extrajo la varianza de cada una de ellas y se realizó una media total que se empleó como característica a la que entrenar el clasificador.

2.4. Mel Frequency Cepstral Coefficients

Otra de las propiedades que se utilizó para el estudio espectral del sonido es el conocido como los coeficientes Mel-Cepstrum de la señal. El cepstrum se puede definir como la Transformada de Fourier del espectro de la señal en cuestión en escala logarítmica.

Para realizar su cálculo se debe seguir el siguiente esquema adjunto:

Figura 2.4.1: Esquema de MFCC

Basándose en esta idea, los MFCC (Mel Frequency Cepstral Coefficients) se utilizan para la representación del habla inspirándose en la percepción auditiva humana. El objetivo es obtener unos coeficientes que capturen la información de interés mediante un proceso que tiene como pasos destacados: el análisis localizado de frecuencia, el cálculo de

la potencia en una serie de bandas críticas y el decorrelado mediante una transformada DCT (Discrete Cosine Transform). En concreto, el algoritmo describe un proceso a seguir los siguientes pasos:

1- Separar la señal en secciones
2- Aplicar FFT a cada sección
3- Aplicar banco de filtros en escala Mel al módulo de la FFT
4- Tomar el logaritmo
5- Aplicar la DCT

A continuación se muestra mediante un ejemplo cada uno de estos pasos detalladamente.

Como se ha mencionado anteriormente, es difícil extraer características en una señal de audio de voz debido a su naturaleza no estacionaria. Para este apartado también se ha utilizado una separación de ventanas de audio de 30 ms con un paso de 10 ms para agrupar la señal de audio en secciones y se ha enventanado mediante una ventana de Hamming para disminuir el efectos de bordes suavizando la señal.

A continuación se puede observar la forma que tendría el valor absoluto de la FFT de la señal previamente filtrada con un filtro de Hamming en las distintas secciones. Como se aprecia, es complicado

analizar la información en esta etapa del procesado, es por ello que debemos seguir aplicando técnicas de procesado a esta señal.

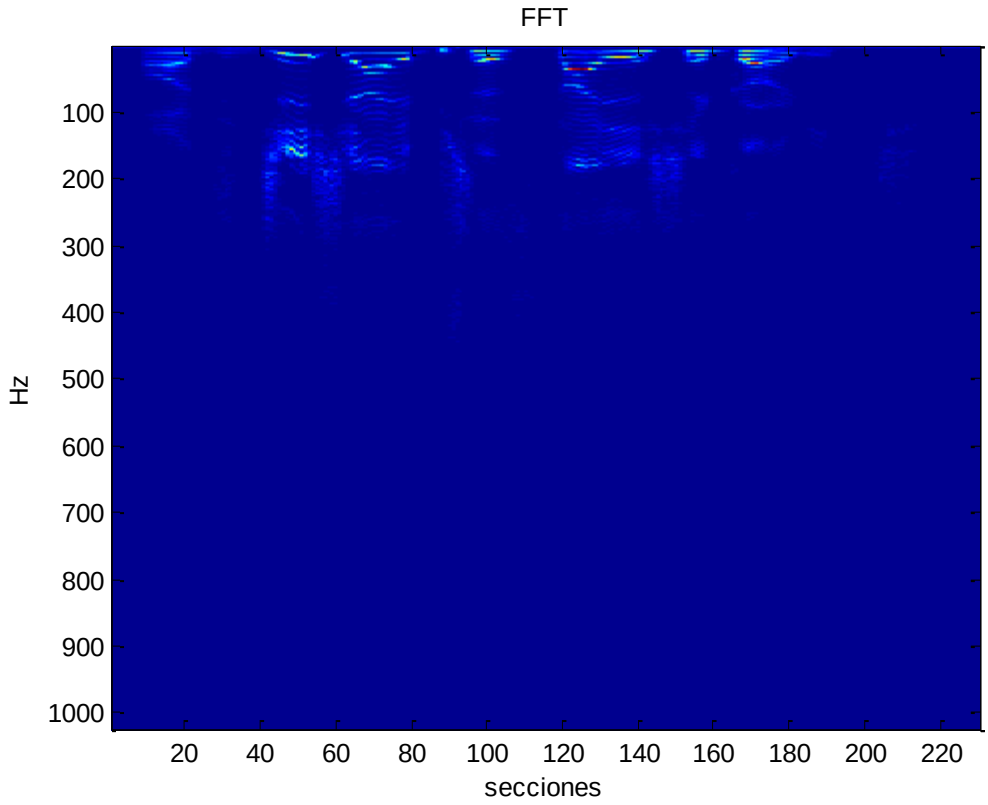

Figura 2.4.2: FFT de señal de audio

En el siguiente paso, se pretende aplicar un banco de filtros de escala Mel a esta señal. El motivo de hacer esto es el de aproximar la señal al modelo de percepción que tenemos los seres humanos. La escala Mel básicamente es una transformación de un espacio a otro que simula la interpretación que nuestro oído hace de las señales de voz. Para realizar este proceso se utiliza la siguiente fórmula:

$$Mel(f) = 2595\, log_{10}\left(1 + \frac{f}{700}\right)$$

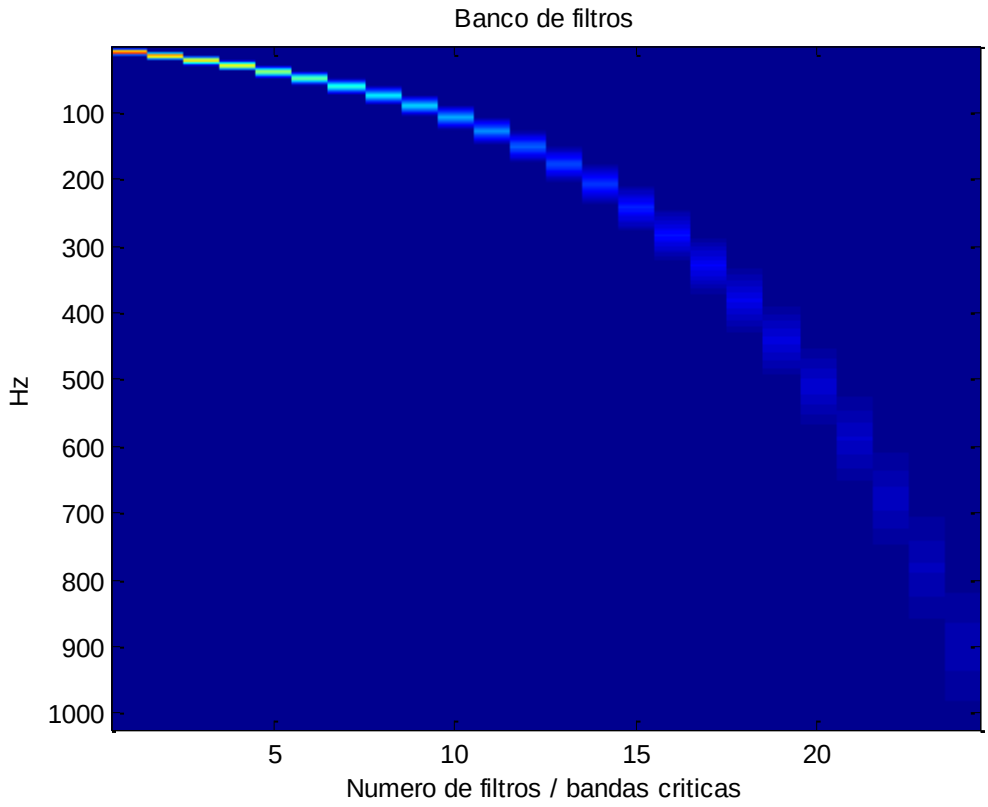

Figura 2.4.3 Banco de Filtros de frecuencia escala Mel en formato matricial

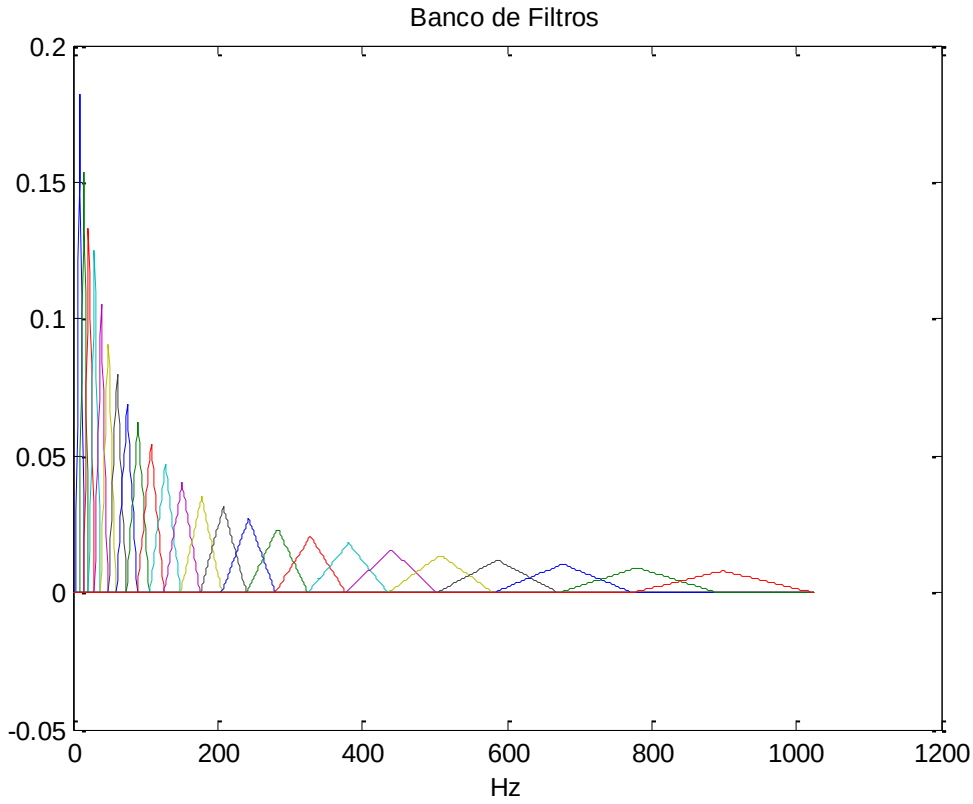

Figura 2.4.4 Banco de Filtros de frecuencia escala Mel

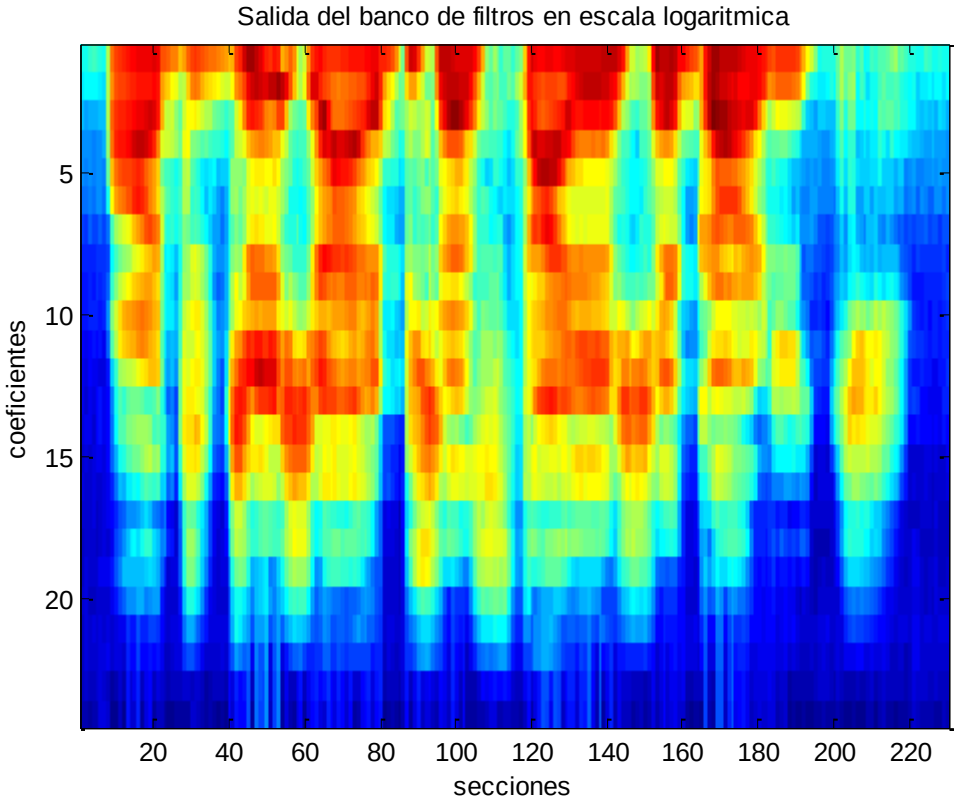

Figura 2.4.5: Señal tras el banco de filtros en escala logarítmica

A la salida del banco de filtros ya se puede intuir algo más de información, pero es a la hora de realizar la operación logaritmo (figura 2.4.5) sobre esta matriz cuando apreciamos realmente información frecuencial de la señal suficientemente útil para poder entrenar un clasificador capaz de distinguir entre dos señales distintas.

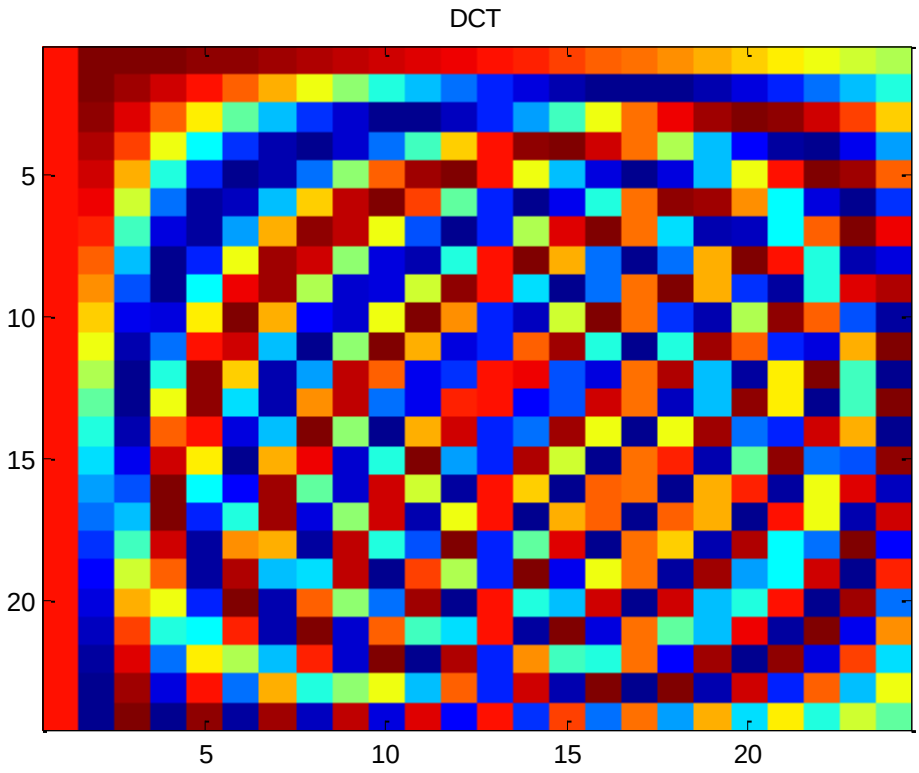

Figura 2.4.6: DCT aplicada

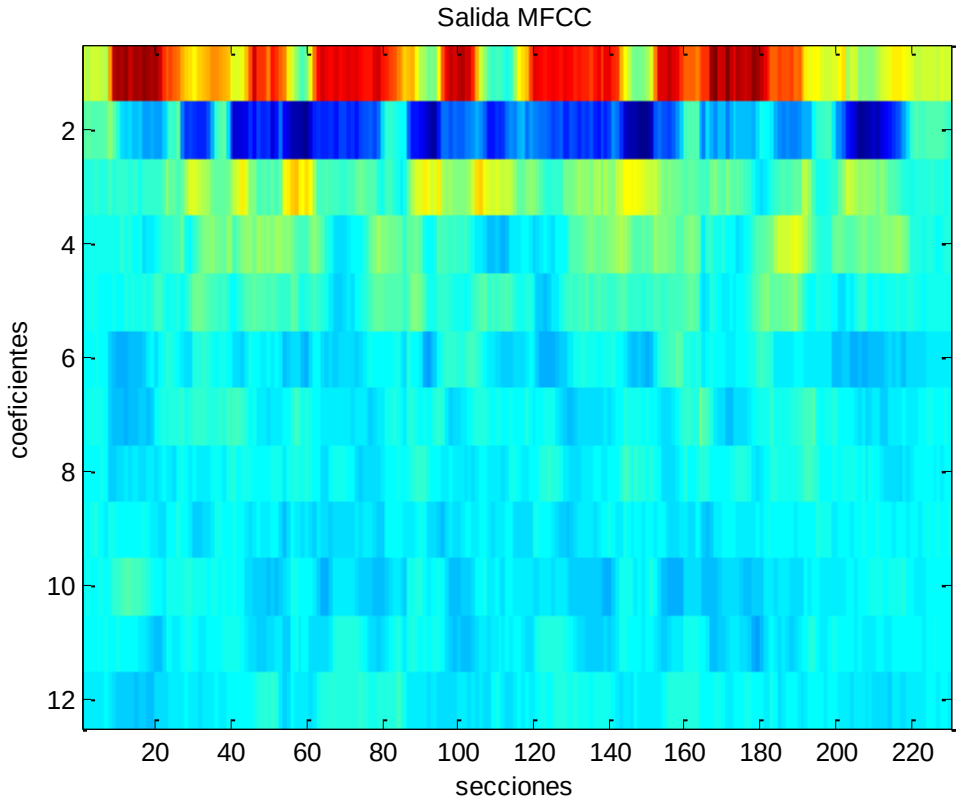

Figura 2.4.7: Salida de Mel Frequency Cepstral Coefficients

Tras aplicar la transformada de cosenos discreta truncada, la información resultante está mucho más compactada y decorrelada. Además, ahora podemos extraer características de tipo estadístico como son la varianza y la media a lo largo de los coeficientes resultantes. Esto nos permitirá analizar una señal de audio independientemente de la cantidad de muestras o secciones que presente, pues esta cantidad es una información que a priori el sistema desconoce.

2.5. Linear Predictive Coding

La predicción lineal codificada es una de las herramientas más utilizadas para el análisis de audio. Una de las razones por la que es tan común su uso, es la posibilidad de obtener información de la envolvente espectral.

Este método, se basa en la idea de la predicción lineal, en el cual se intenta modelar una señal que se supone que se ha producido mediante un modelo todo polos, cuyos coeficientes tenemos que averiguar.

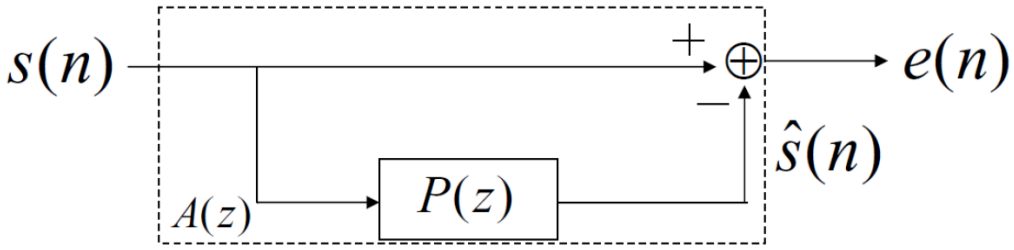

Figura 2.5.1: Sistema de predicción lineal

En la figura 2.5.1 se puede ver el funcionamiento del sistema. Dada una señal (s), el filtro FIR P(z) de orden N, tratará de replicar la señal para contrarrestarla, de tal forma que el error final, e(n), sea lo más pequeño posible. El objetivo del sistema es hallar los coeficientes de P(z) para los cuales el error es más bajo. Para ello se pretende disminuir el error cuadrático:

$$\sum_{n} e[n]^2 = \sum_{n} (s(n) - \sum_{i=1}^{N} a_i x[n-i])^2$$

Esto también se puede interpretar como que el filtro extrae las características para luego replicarlo de la forma más fiel posible. Esta última idea es la que se utiliza en el algoritmo LPC.

El sistema irá aprendiendo de este comportamiento a medida que la señal vaya acentuando las muestras dadas. Es por ello que, nuevamente, no trataremos la señal de audio de forma completa, sino que la enventanaremos para estudiar las propiedades de sus secciones. Cada sección tendrá una longitud equivalente al muestreo de 30 ms y se desplazará la ventana entre sección y sección un total de 10 ms.

2.6. PROFUNDIDAD

Desde un primer momento la idea era extraer información de profundidad mediante el sensor que tiene la Kinect. Este sensor está formado por un sensor de infrarrojos (emisor y receptor). El emisor emite en la banda del infrarrojo y espera a que el receptor reciba el rebote de esta onda en objetos situados frente al dispositivo. Una vez conocido el tiempo que ha tardado la onda en viajar desde que ha sido emitida hasta que ha sido captada por el sensor, y conociendo la velocidad a la que viaja la onda, resulta sencillo calcular la distancia a la que se encuentra el objeto interferido.

El principal problema que se encontró utilizando esta tecnología es la baja resolución que ofrece el dispositivo en esta versión. Si bien resulta suficiente para localizar nuestro rostro, resulta insuficiente para localizar zonas relevantes de la cara como los ojos o la boca y mucho menos para detectar la forma que tiene la cara. En la figura 2.6.1 se puede ver un ejemplo de lo ocurrido.

Figura 2.6.1: Esquema del algoritmo Adaboost

Por este motivo, finalmente se decidió seguir con el trabajo centrándose en la parte de imagen y sonido. Del mismo modo, la creación de una base de datos propia ya no resultaba ser tan necesario pues se pudieron encontrar bases de datos de audio y video referidas a otros artículos científicos.

3. ENTRENAMIENTO

3.1. BASES DE DATOS

Para este proyecto, y tras la previa investigación del estado del arte, se han seleccionado las siguientes bases de datos que se pueden obtener de forma gratuita para fines académicos:

-KDEF (The Karolinska Directed Emotional Faces)[6]

-Database of Polish Emotional Speech[7]

-JAFFE (Japanese Female Facial Expression)[8]

-AFW (Annotated Faces in-the-wild)[13][14]

-Helen (The Helen Database)[13][14]

-LFPW (The Labeled Face Parts in-the-wild)[13][14]

- iBug. [13][14]

Para utilizar el algoritmo de PO_CR anteriormente nombrado, es necesario entrenarlo con una base de datos que contenga imágenes asociadas a un mapa de FCP. Para ello el algoritmo se entrenó con las bases de datos de AFW, Helen, LFPW e IBug que constan de un total de 1026 imágenes con 68 puntos faciales asociados (aunque el algoritmo sólo utiliza 49).

Las primeras pruebas que se realizaron para la detección de emociones en imágenes, se efectuaron con la base de datos JAFFE. Esta base de datos contiene 213 imágenes de 7 emociones (neutral, alegría, sorpresa, enfado, miedo, asco y tristeza) asociadas a 10 modelos japonesas. Aunque para las primeras pruebas resultó muy útil porque al tener un número reducido de muestras las conclusiones de las pruebas se obtenían con mayor rapidez, para un experimento final no era suficiente, pues sólo contenían imágenes de mujeres. Por ello, finalmente se optó por utilizar la base de datos de KDEF que contiene un total de 4900 imágenes con 7 emociones. Estas imágenes fueron tomadas a 70 individuos (35 mujeres y 35 hombres) fotografiando las 7 emociones (2 veces por persona) desde 5 ángulos distintos. En este proyecto sólo se ha tenido en cuenta las fotografías tomadas de perfil, pues el algoritmo PO_CR es sensible a los cambios de orientación de la cara, lo que hace que tengamos un total de 980 imágenes, 140 por emoción.

En cuanto a la parte de audio, se ha empleado la Database of Polish Emotional Speech. Esta base de datos está compuesta por 240 muestras de audio, distribuidas en 6 emociones (enfado, aburrimiento, miedo, alegría, neutral y tristeza). Para mantener una mayor similitud con la base de datos de la parte de imagen, se ha descartado la parte de 'aburrimiento' por no ser considerado como una emoción como tal.

3.2. TIPOS

Para realizar la clasificación de emociones, es necesario contar con un algoritmo de aprendizaje automático. Para ello, se ha utilizado la plataforma de software Weka (Waikato Environment for Knowledge Analysis). A continuación se adjuntan una lista con una breve descripción de los algoritmos empleados más importantes:

I. Support Vector Machine (SVM)

Este algoritmo busca separar de forma óptima las clases de nuestro problema buscando el hiperplano que tenga la máxima distancia con los puntos más cercanos al mismo.

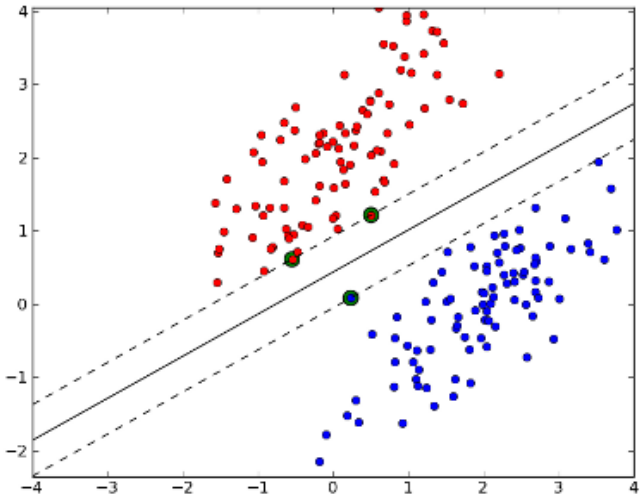

Figura 3.2.1: Ejemplo SVM lineal con 2 clases y 2 características

Esta separación puede medirse sobre una frontera lineal como en el ejemplo de la figura 3.2.1 o no lineal como en la figura 3.2.2, en cuyo caso se recurre al uso de kernels, que permiten la transformación no lineal de los datos a un espacio de mayor dimensión en el que sí que son separables linealmente. Por lo tanto cuando se exploran SVMs hay que buscar qué tipos de kernel se adaptan mejor a la situación de las muestras. Los kernels que se usan habitualmente son los de tipo polinomial, radial o sigmoide. En la figura 3.2.2 se puede ver algunos ejemplos en los que claramente el funcionamiento lineal sería pésimo en comparación con los de carácter no lineal.

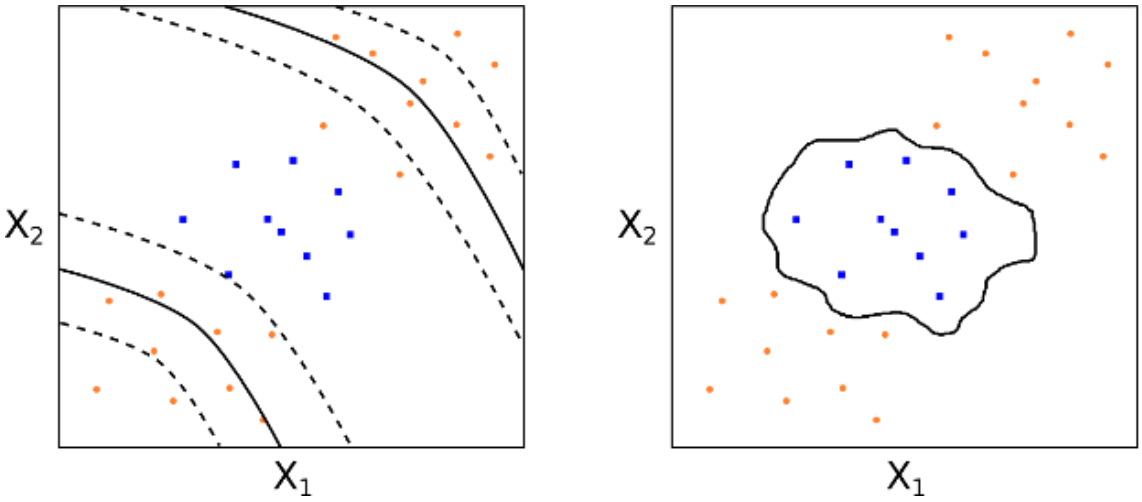

Figura 3.2.2: Ejemplo SVM no lineal con 2 clases y 2 características

II. Árboles

Estos algoritmos buscan generar un árbol de decisión para realizar la clasificación de clases. Se analizan las características disponibles para cada sistema y se van eligiendo los nodos del árbol en función de la cantidad de información que contengan. Para determinar esta cantidad de información, se utilizan medidas como la entropía o la ganancia de información. Algunos de los algoritmos más utilizados para generar las reglas de decisión son ID3[23] y C4.5[24].

Una vez generado el árbol por el algoritmo, para una nueva muestra, se irán recorriendo los nodos del árbol hasta concluir con un resultado. En la figura 3.2.3, se puede observar un ejemplo en el que al entrar el dato CHMIN, el algoritmo se desplazará a una rama si es mayor que 7.5 y a la otra si es menor o igual a este valor y repetirá este proceso hasta decidir a qué clase pertenece.

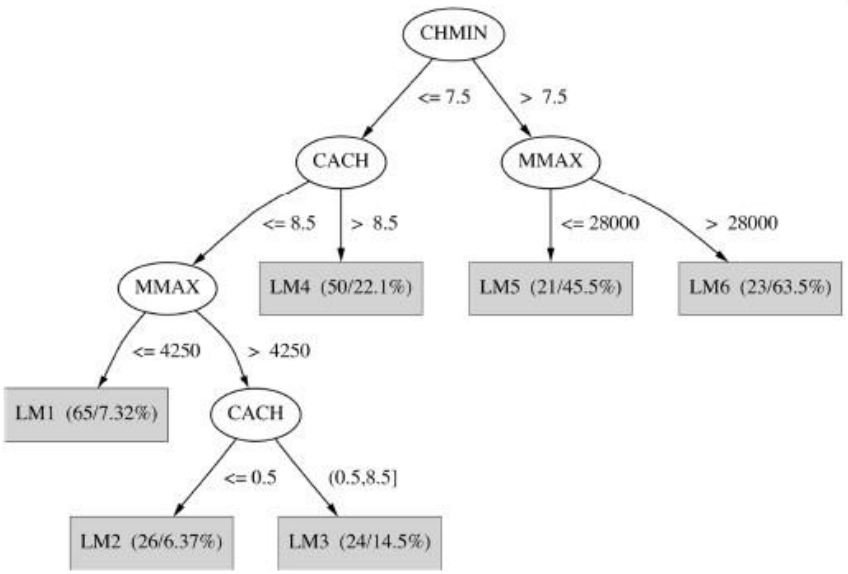

Figura 3.2.3: Ejemplo SVM árbol

Algunos de los sistemas de árboles empleados en este trabajo y que mejor resultado han dado son: LMT (Logistic Model Tree), Random Forest, FT (Functional Trees).

III. Redes Bayesianas

Las redes bayesianas realizan la clasificación mediante un modelo grafo probabilístico que representa un conjunto de variables aleatorias y sus dependencias condicionales. Dadas unas características concretas, el algoritmo computa la probabilidad de la presencia de varias clases para proceder a la clasificación más probable.

El programa Weka nos permite utilizar varios algoritmos de búsqueda de la estructura de dependencia de la red, como pueden ser K2, TAN, TabuSearch, HillClimber, etc.

IV. Bagging y Adaboost

Estos algoritmos nos permiten exprimir más las funciones de otros clasificadores. Se basan en crear varios modelos del mismo clasificador para después combinarlos.

En el caso del Bagging, tras entrenar distintos clasificadores con diferentes muestras o características, para una muestra nueva, cada clasificador tratará de averiguar a qué clase pertenece esta muestra y

posteriormente se someterá a un proceso de votación entre todos los clasificadores para decidir cuál es con mayor probabilidad.

La técnica de Adaboost en cambio, pretende descartar situaciones mediante clasificadores débiles. Como se ha explicado anteriormente, el algoritmo de Viola Jones utiliza esta técnica para determinar si un conjunto de píxeles se puede clasificar como cara o no.

3.3. Evaluación de prestaciones

Para medir cómo de buena es la clasificación de una prueba concreta a realizar, se han utilizado los parámetros de precisión, recall y el parámetro F. La precisión es el coeficiente entre las muestras que el clasificador considera verdaderos positivos y la suma de estos verdaderos positivos con las falsas alarmas, es decir, estamos midiendo las veces que se acierta correctamente la clase respecto al total de veces que el clasificador predice esa clase (con los correspondientes errores cometidos). El recall, en cambio, nos mide la cantidad de verdaderos positivos en función de los que realmente existían en la prueba, es decir, los aciertos entre el total. El parámetro F se utiliza para medir lo buen clasificador que es en función de los dos parámetros anteriores.

$$Precision = \frac{True\ positive}{\#predicted\ positive} \qquad Recall = \frac{True\ positive}{\#\text{actual positive}}$$

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

4. IMPLEMENTACIÓN

Una vez presentadas las características que se van a extraer de las bases de datos para su posterior entrenamiento, se explicarán los pasos que se siguieron en la parte de pruebas y resultados. Para poder entrenar un algoritmo de aprendizaje automático es necesario que las características extraídas de la base de datos presenten una correlación entre ellas y además su distribución sea distinta en función de la clase a la que pertenezcan. El esquema que se seguirá en este apartado será el mostrado en las figuras 4.1 y 4.2. Para este esquema y el resto de los apartados, se ha seguido un convenio de colores para el cual se relaciona un color con un sentimiento para su mejor visualización e interpretación.

Negro → Neutral
Cian → Sorpresa
Amarillo → Felicidad
Azul → Tristeza
Magenta → Miedo
Verde → Asco
Rojo → Enfado

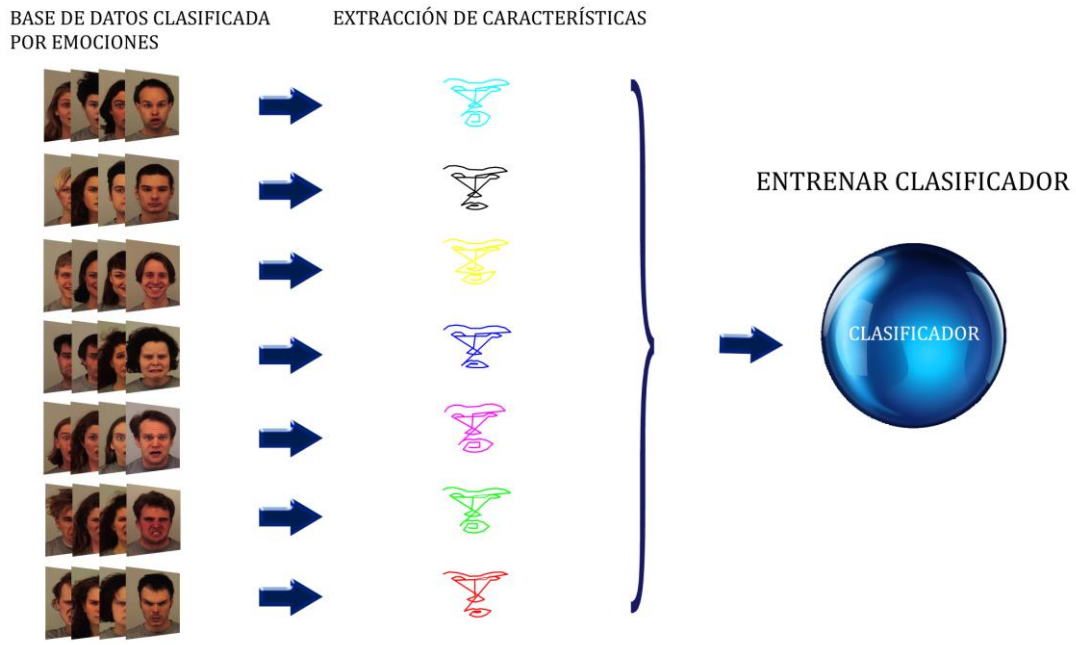

Figura 4.1: Esquema de entrenamiento de un clasificador

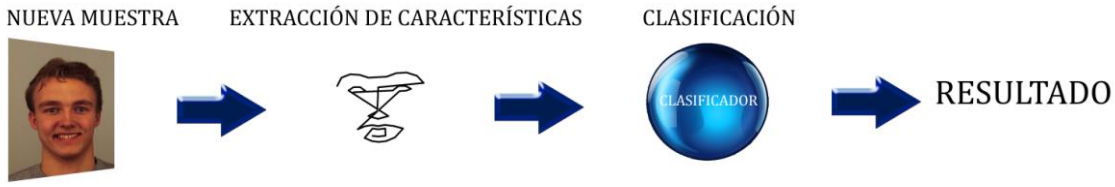

Figura 4.2: Esquema de una clasificación

La primera parte consistió en ordenar la base de datos para poder extraer las características de forma correcta. Una vez seleccionadas las características a extraer, se entrenaron en un clasificador. Posteriormente y para una nueva muestra, se extraen las mismas características que el clasificador procesará dando como resultado la clase a la que el clasificador cree que pertenece esta muestra.

4.1. IMAGEN

Como primera idea se procedió a realizar un cálculo de distancia euclidea entre puntos, por ejemplo, entre los puntos 45 y 48 la distancia será mayor si la emoción mostrada es sorpresa que si es neutral. En la figura 4.1.1 se ve como la distancia entre estos puntos (señalados en rojo) es menor para la clase neutral que para la clase sorpresa.

Figura 4.1.1: Emoción sorpresa (iza) vs Emoción neutral (der) con algoritmo PO_CR

Los resultados que se probaron utilizando una parte de la base de datos como subconjunto de pruebas fueron satisfactorios. Sin embargo, a la hora de proceder a utilizar este algoritmo con ejemplos propios grabados mediante el instrumento de la kinect se producían errores de clasificación. Esto es debido a que estas imágenes tomadas con el dispositivo tienen una resolución de píxeles distinta a las de la base de datos empleada, y por tanto las distancias calculadas no eran similares. Estas distancias se normalizaron para paliar este error. Se llegó a la conclusión de que debían ser normalizadas, pues de lo contrario estaríamos trabajando con una distancia absoluta que dependería de otros factores como la resolución de la imagen o si la persona se

encuentra en el primer plano de la fotografía o más alejada. Para despreciar estos efectos, se ha utilizado como referencia la distancia entre las esquinas inferiores de los ojos, correspondiente a la distancia entre los puntos 23 y 26, pues es una cantidad que se mantiene fija entre los sujetos. Una vez se normalizaron las características extraídas, el entrenador conseguía unos resultados más óptimos en la clasificación.

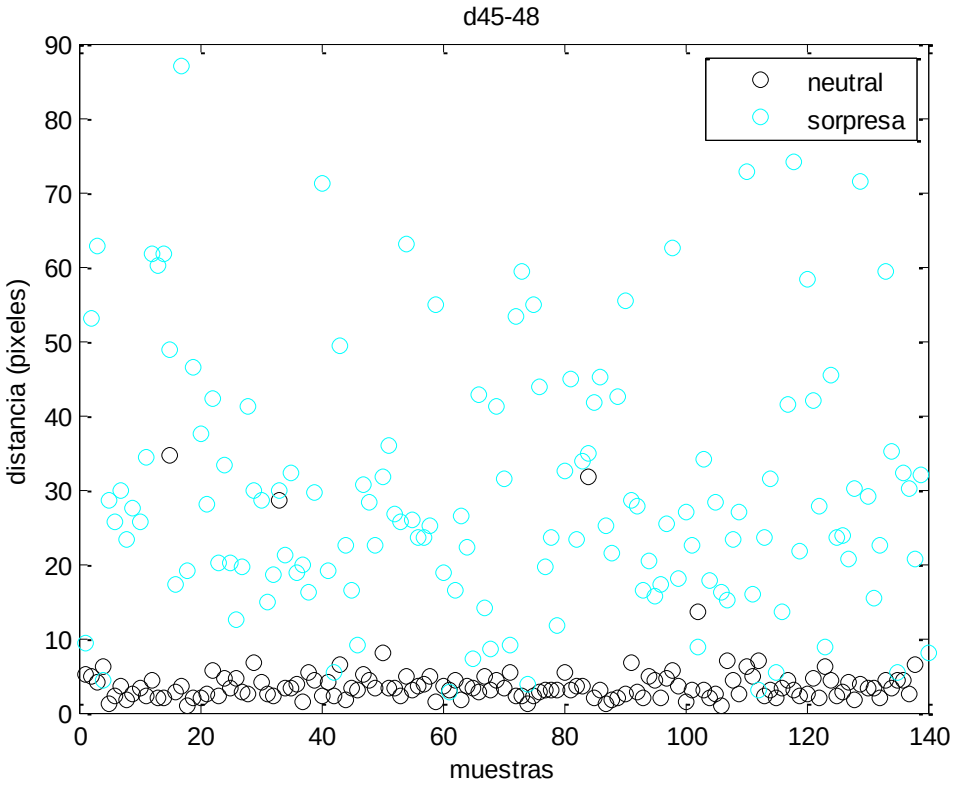

Figura 4.1.2: Valores de distancia d45_48 sin normalizar

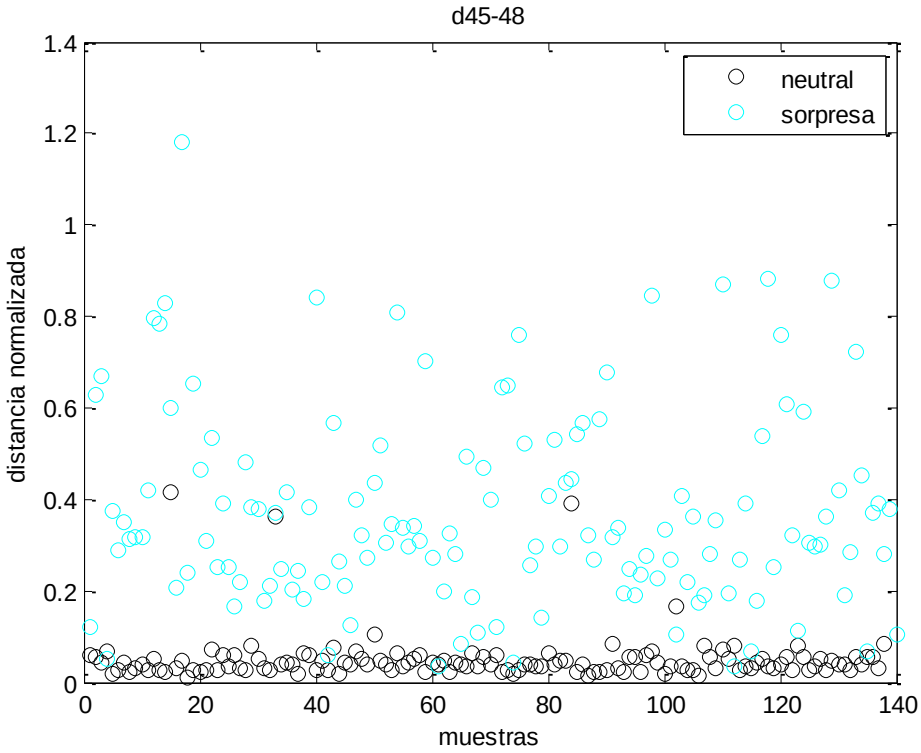

Figura 4.1.3: Valores de distancia d45_48

Una de las primeras pruebas que se hizo fue parametrizar las partes de las caras que a priori parecen más relevantes a la hora de hacer una clasificación de emociones. El objetivo era parametrizar zonas de la cara concretas mediante el cálculo de distancias entre puntos de esas zonas. Se utilizaron los siguientes datos:

- Parametrización boca abierta/cerrada
  - d32_38
  - d35_41
  - d45_48

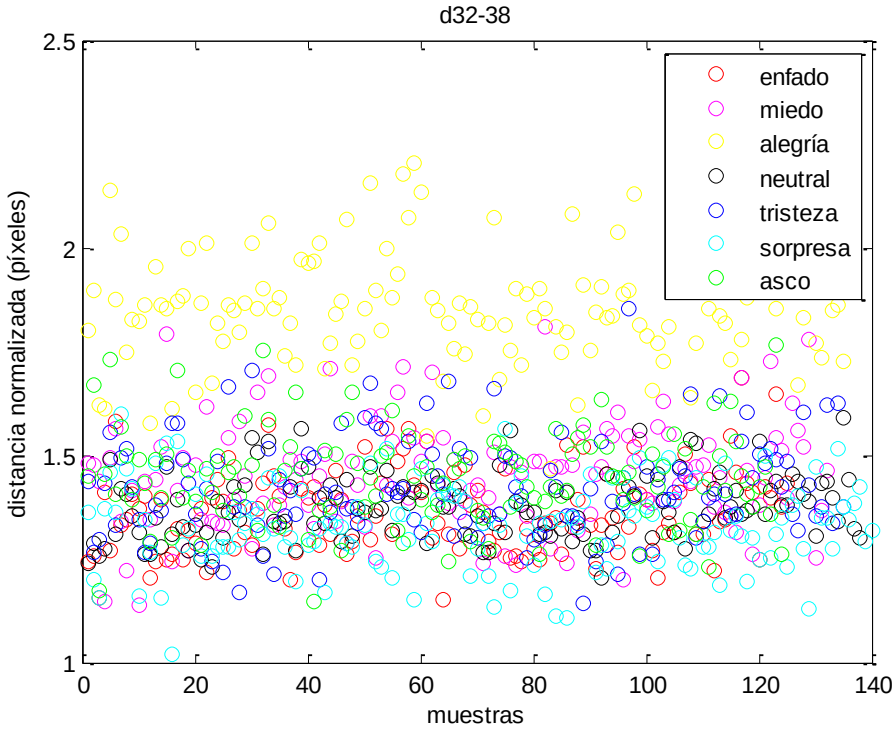

Figura 4.1.4: Valores de distancia d32_38

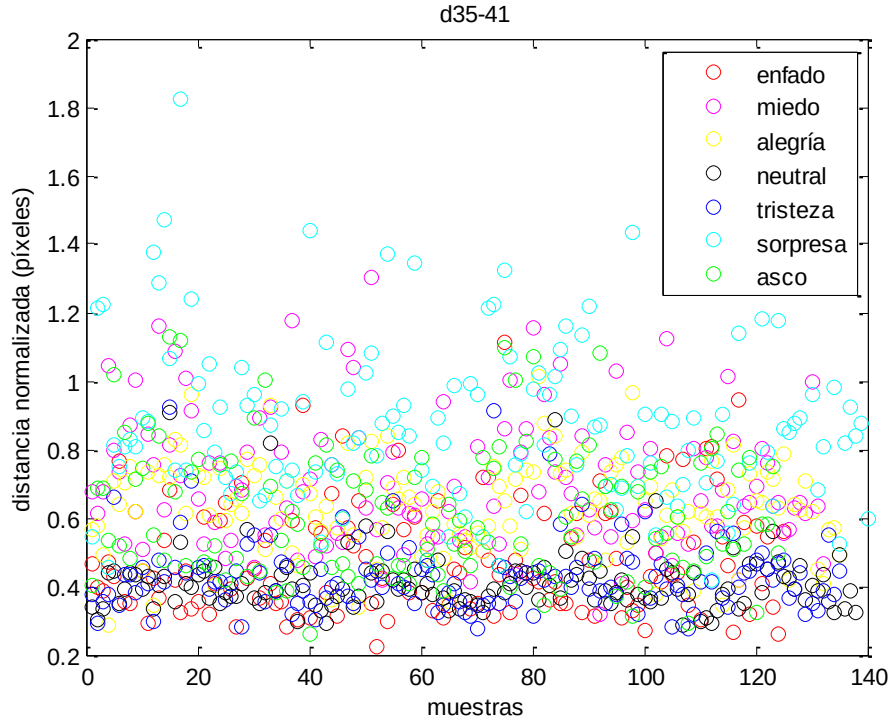

Figura 4.1.5: Valores de distancia d35_41

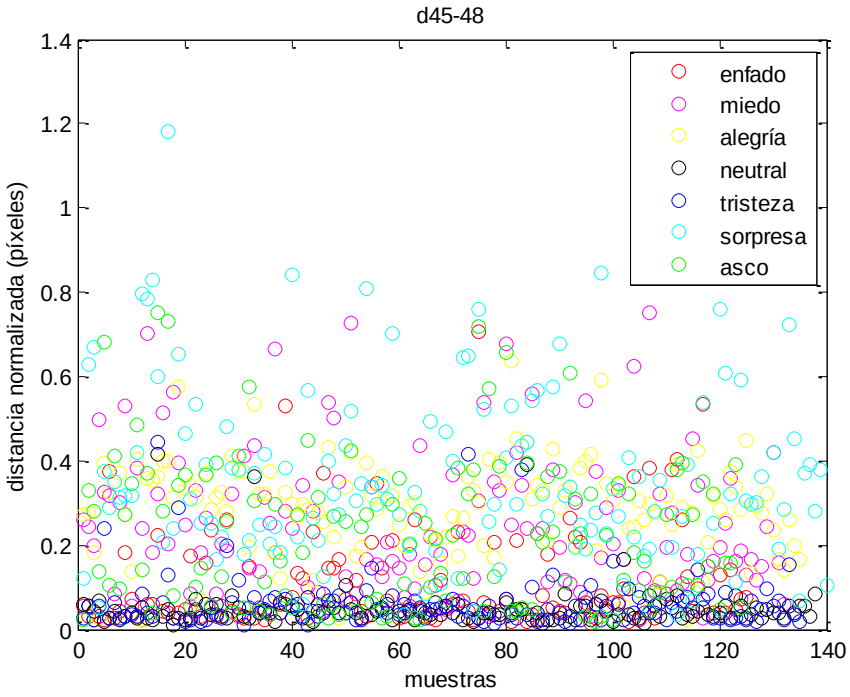

Figura 4.1.6: Valores de distancia d45_48

- Parametrización sonrisa
  - d32_41
  - d38_41

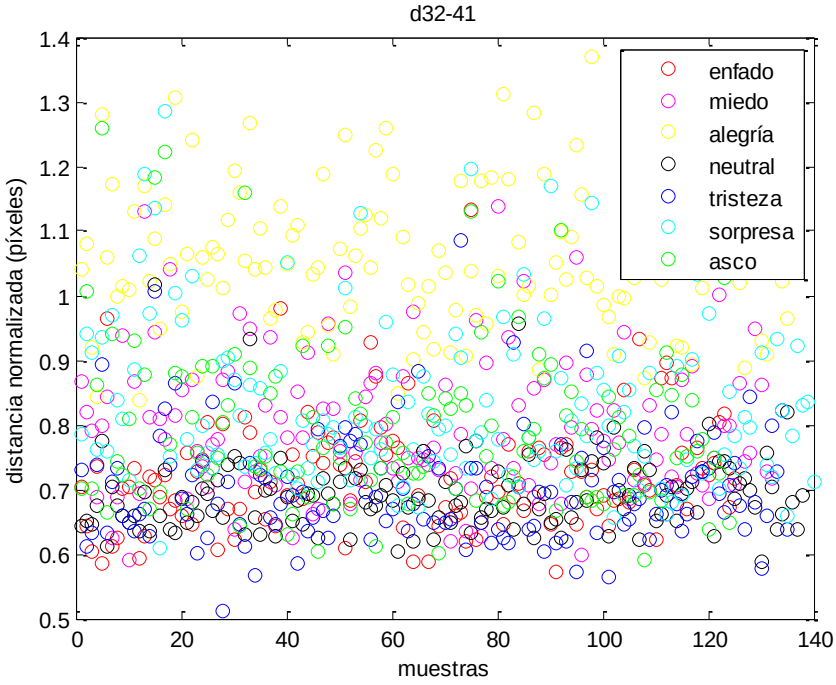

Figura 4.1.7: Valores de distancia d32_41

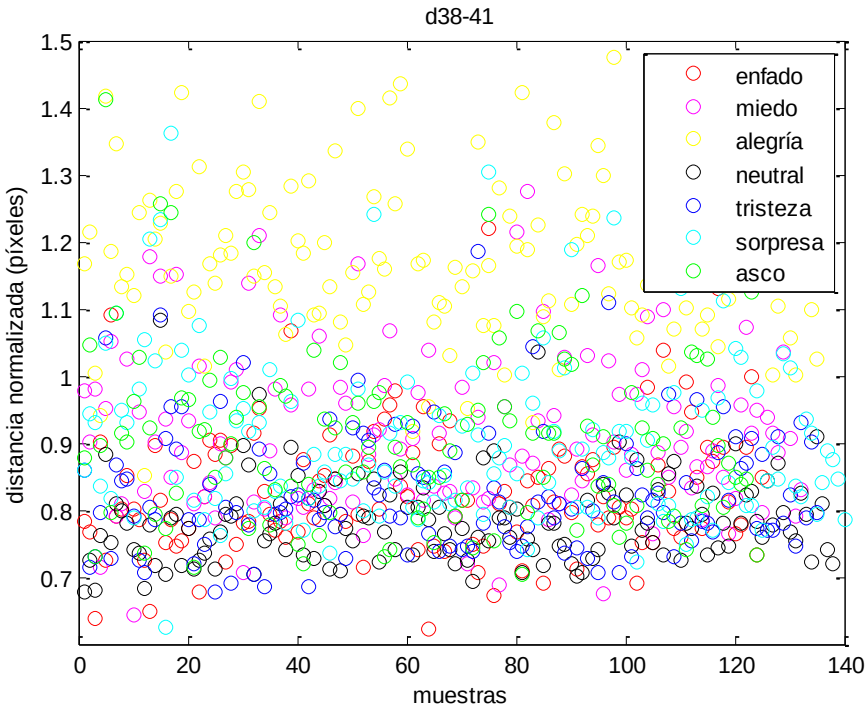

Figura 4.1.8: Valores de distancia d38_41

- Parametrización levantamiento de cejas
  - d4_22
  - d7_27

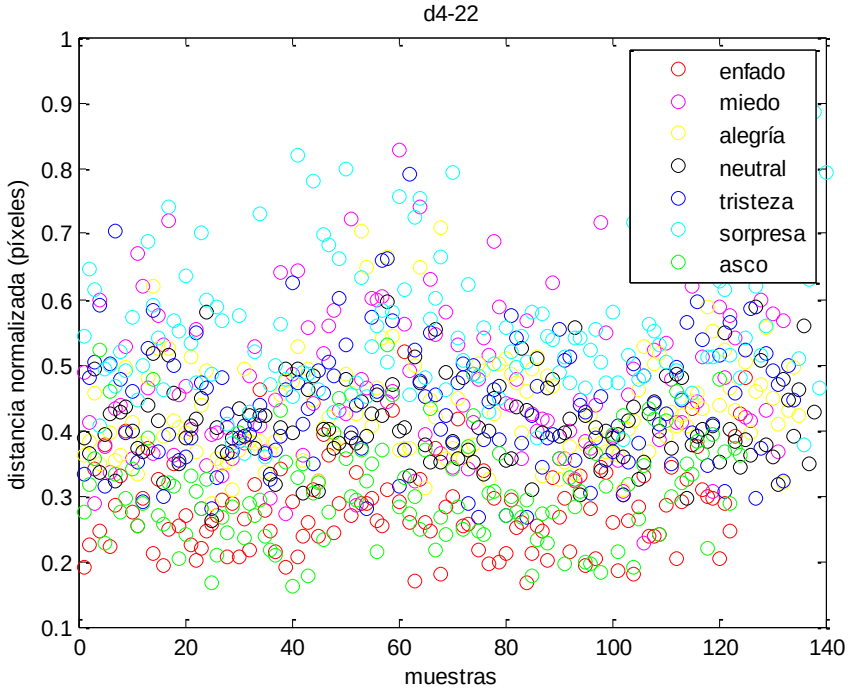

Figura 4.1.9: Valores de distancia d4_22

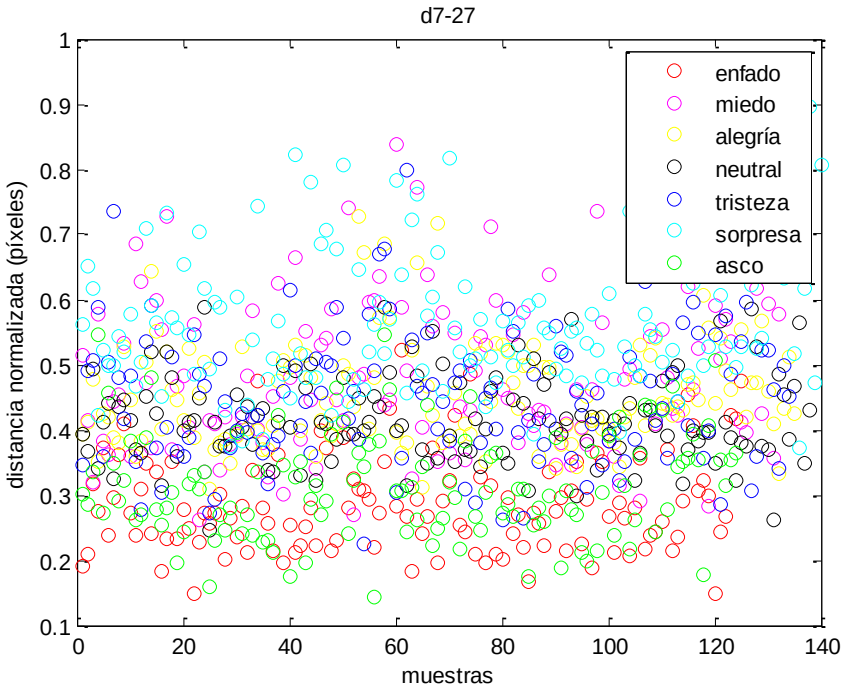

Figura 4.1.10: Valores de distancia d7_27

- Parametrización apertura ojo izquierdo y derecho
  - d22_24
  - d27_31

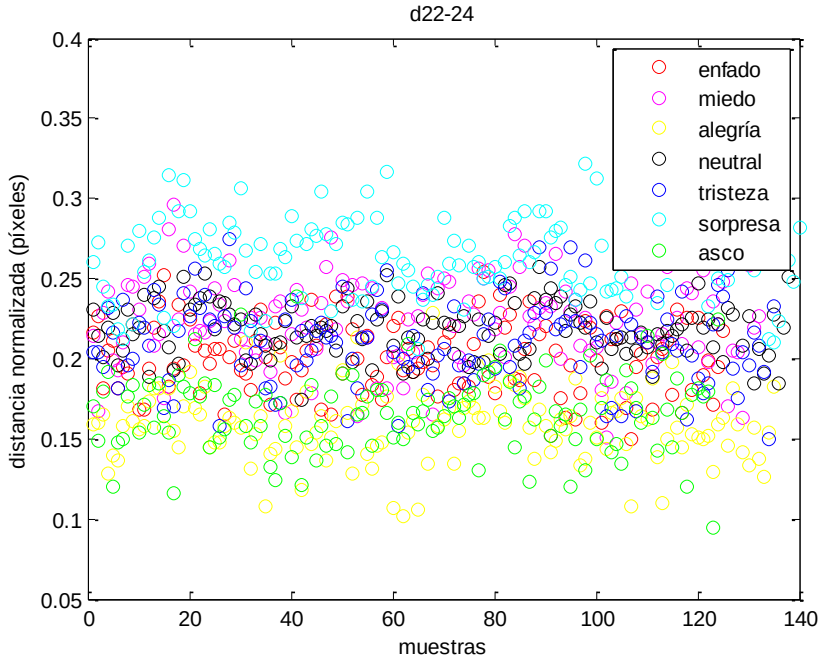

Figura 4.1.11: Valores de distancia d22_24

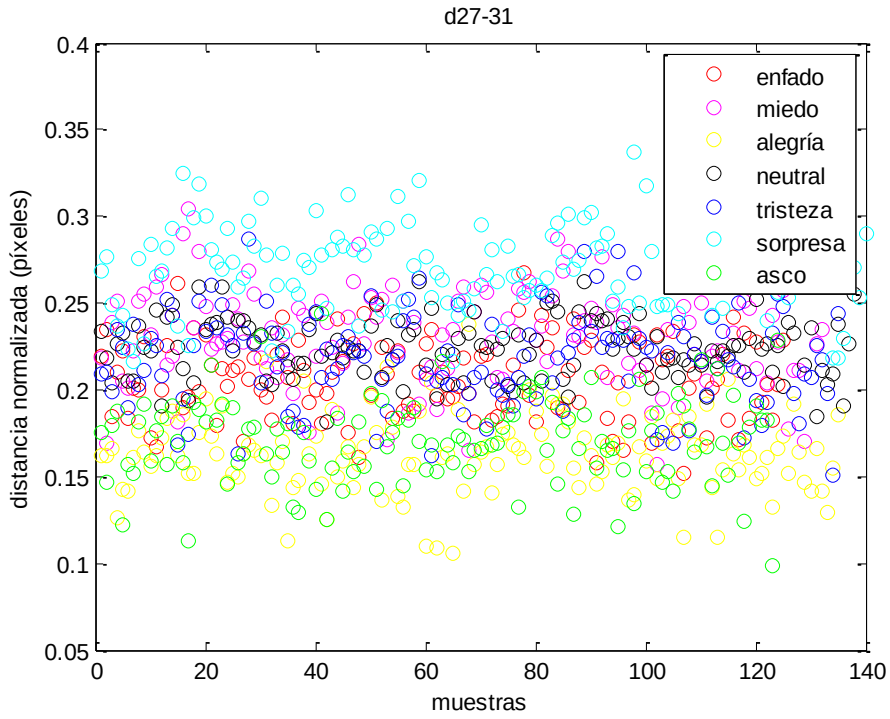

Figura 4.1.12: Valores de distancia d27_31

Con estos datos y el algoritmo de aprendizaje automático LMT proporcionado por weka[25][26] se llegaron a los siguientes resultados:

	Neutral	Alegría	Tristeza	Miedo	Enfado	Asco	Sorpresa
Neutral	90	0	35	4	8	0	1
Alegría	2	125	1	5	0	2	0
Tristeza	36	4	58	10	12	12	3
Miedo	8	5	11	68	6	7	27
Enfado	11	0	10	5	82	17	0
Asco	3	3	8	6	15	88	1
Sorpresa	4	0	1	20	0	0	115

Figura 4.1.13: Matriz de confusión

	Precisión	Recall	F
Neutral	0.584	0.652	0.616
Alegría	0.912	0.926	0.919
Tristeza	0.468	0.430	0.448
Miedo	0.576	0.515	0.544
Enfado	0.667	0.656	0.661
Asco	0.698	0.710	0.704
Sorpresa	0.782	0.821	0.801

Figura 4.1.14: Precisión Recall y F de cada emoción

Correctamente clasificado	**67.38%**

Sin embargo, finalmente se optó por utilizar la mayor cantidad de datos posibles para mejorar el entrenador. Como se ha mencionado anteriormente, el algoritmo de PO_CR nos permite extraer un total de 49 puntos, lo que, en términos de distancias posibles, nos da un total de 1176 combinaciones posibles entre puntos.

Se realizó la misma prueba que en el caso anterior, utilizando el algoritmo LMT para poder comparar ambos resultados:

	Neutral	Alegría	Tristeza	Miedo	Enfado	Asco	Sorpresa
Neutral	105	0	22	4	6	0	1
Alegría	0	131	1	3	0	0	0
Tristeza	16	2	90	9	7	9	2
Miedo	8	2	7	80	7	9	19
Enfado	10	1	4	6	92	12	0
Asco	4	1	11	5	11	91	1
Sorpresa	3	0	1	13	0	0	123

Figura 4.1.15: Matriz de confusión

	Precisión	Recall	F
Neutral	0.719	0.761	0.739
Alegría	0.956	0.970	0.963
Tristeza	0.662	0.667	0.664
Miedo	0.667	0.606	0.635
Enfado	0.748	0.736	0.742
Asco	0.752	0.734	0.743
Sorpresa	0.842	0.879	0.860

Figura 4.1.16: Precisión Recall y F de cada emoción

Correctamente clasificado	**76.64%**

Como se puede apreciar, la mejora es significativa, destacar la mejora en tristeza y miedo que, a pesar de seguir siendo las peores clasificadas, son también las que más mejoran respecto al caso anterior.

Por tanto, se procedió a probar con otros clasificadores con el mismo número de características. En el anexo C se encuentra una lista de los clasificadores utilizados y el porcentaje de muestras correctamente clasificadas.

Finalmente, el que mejor resultado dio fue el SVM[27] ya que mediante la manipulación de los valores de gamma y cost se consigue un importante grado de mejora. Se realizó un barrido logarítmico a lo largo de ambos parámetros.

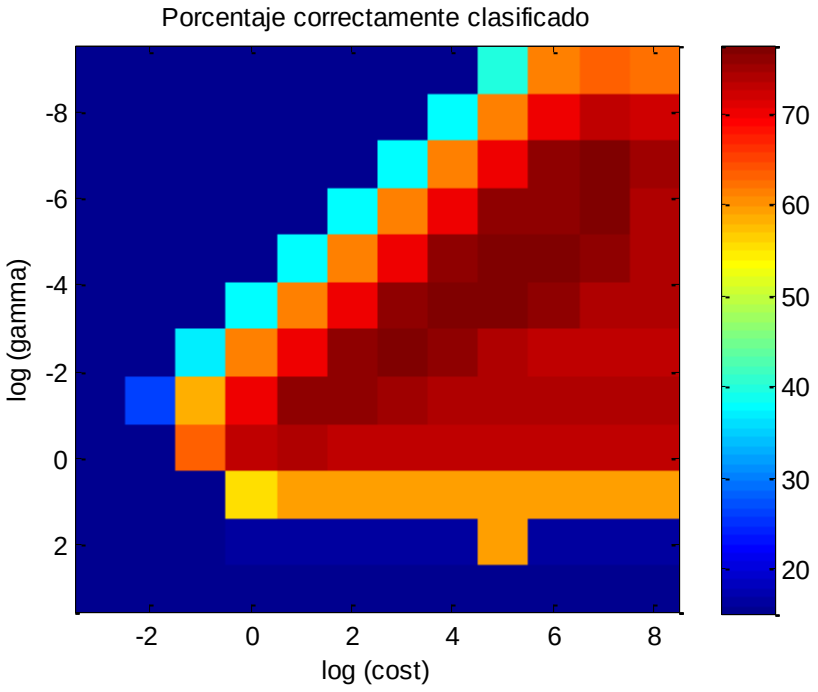

Figura 4.1.17: Análisis de los parámetros gamma y cost en escala logarítmica

En la figura 4.1.17 se puede apreciar que el punto óptimo lo encontramos para los valores de gamma = 0.01 y cost = 1000, dando lugar a los siguientes resultados:

	Neutral	Alegría	Tristeza	Miedo	Enfado	Asco	Sorpresa
Neutral	116	1	14	0	5	0	2
Alegría	0	130	1	4	0	0	0
Tristeza	17	0	94	10	7	7	0
Miedo	7	1	11	80	6	5	22
Enfado	9	0	6	7	88	15	0
Asco	3	1	13	6	9	91	1
Sorpresa	3	0	0	16	0	0	121

Figura 4.1.18: Matriz de confusión

	Precisión	Recall	F
Neutral	0.748	0.841	0.792
Alegría	0.977	0.963	0.970
Tristeza	0.676	0.696	0.686
Miedo	0.650	0.606	0.627
Enfado	0.765	0.704	0.733
Asco	0.771	0.734	0.752
Sorpresa	0.829	0.864	0.846

Figura 4.1.19: Precisión Recall y F de cada emoción

Correctamente clasificado	77.5027%

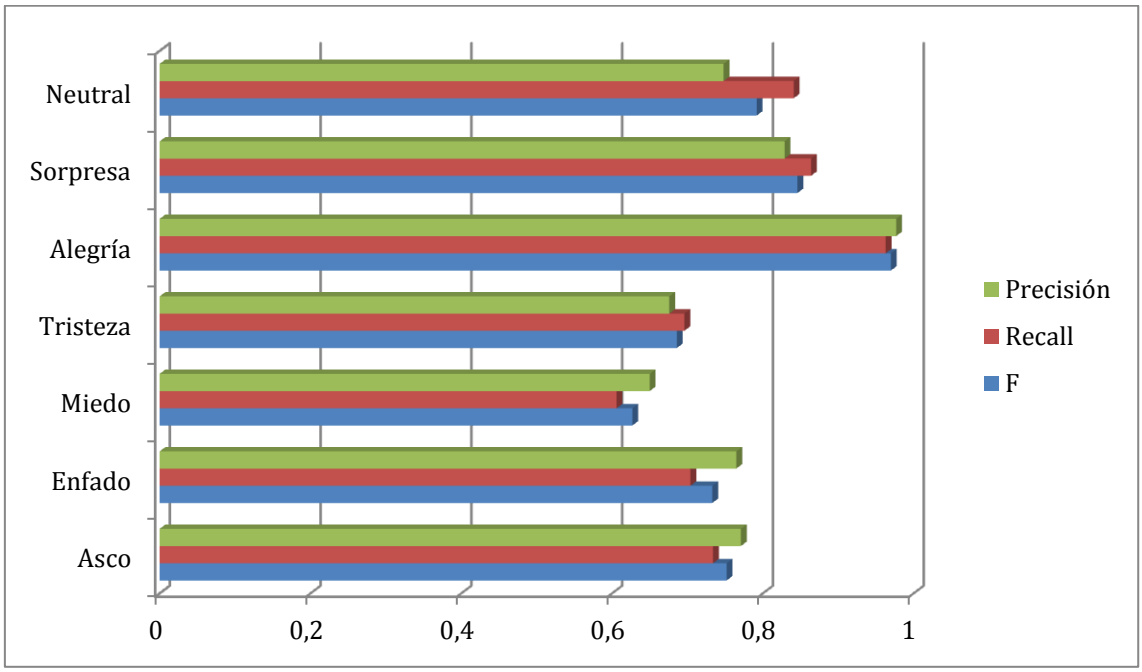

Figura 4.1.20: Gráfica de Precisión Recall y F de cada emoción

El resultado es similar, al del caso anterior de árbol LMT, como conclusión final podemos destacar que la emoción mejor clasificada es la alegría (0.977 de precisión y 0.963 de recall) y la que se clasifica con más dificultad es el miedo (0.650 de precisión 0.606 de recall).

Otra de las funciones que podemos hacer con el programa Weka es clasificar las características con las que vamos a entrenar por relevancia[28]. En el anexo D se encuentra el listado de estas características ordenadas.

En la gráfica 4.1.21 se ve una progresión con las 100 primeras características más relevantes y se puede ver como tiende hacia el valor máximo obtenido.

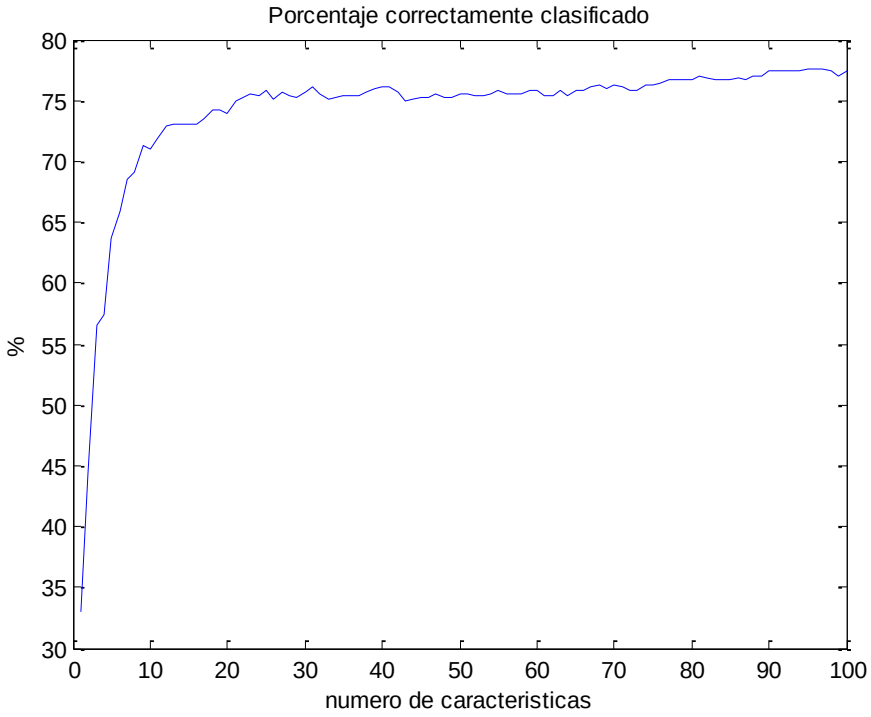

Figura 4.1.21: Mejora del sistema en función del número de características

Por último, se procedió a intentar mejorar este algoritmo SVM[27] con los parámetros gamma y costo adaptados, añadiendo las técnicas de Bagging o Adaboost. Los resultados obtenidos fueron:

Bagging	**Adaboost**
73.63%	75.8%

Con el fin de comprender mejor la idea de la clasificación que intentan realizar los algoritmos de aprendizaje automático, a continuación se expone una gráfica cuyos ejes corresponden a dos de las características utilizadas y cada color está asociado a una emoción concreta de las 7 disponibles.

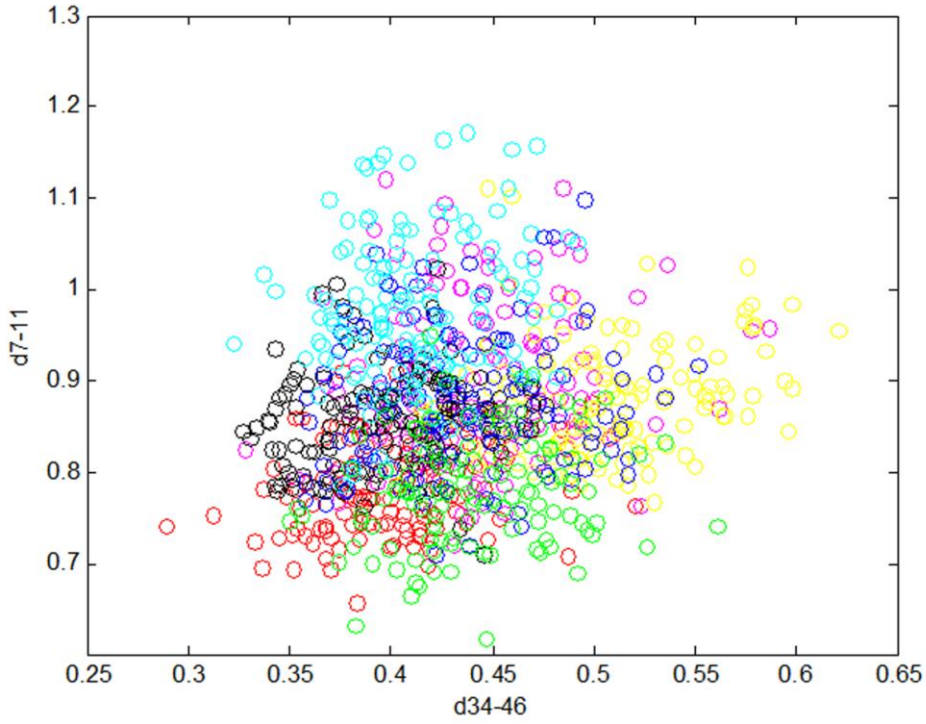

Figura 4.1.22: Ejemplo de clasificación con 2 características

Como se puede apreciar, hay algunas zonas en las que claramente domina una clase en concreto. Estas zonas serían las que el clasificador etiquetaría como pertenecientes a esas clases. De tal forma que para una nueva muestra, el algoritmo extraería estas dos características y las colocaría en el mapa 2D para catalogarlas dentro de una de las clases posibles.

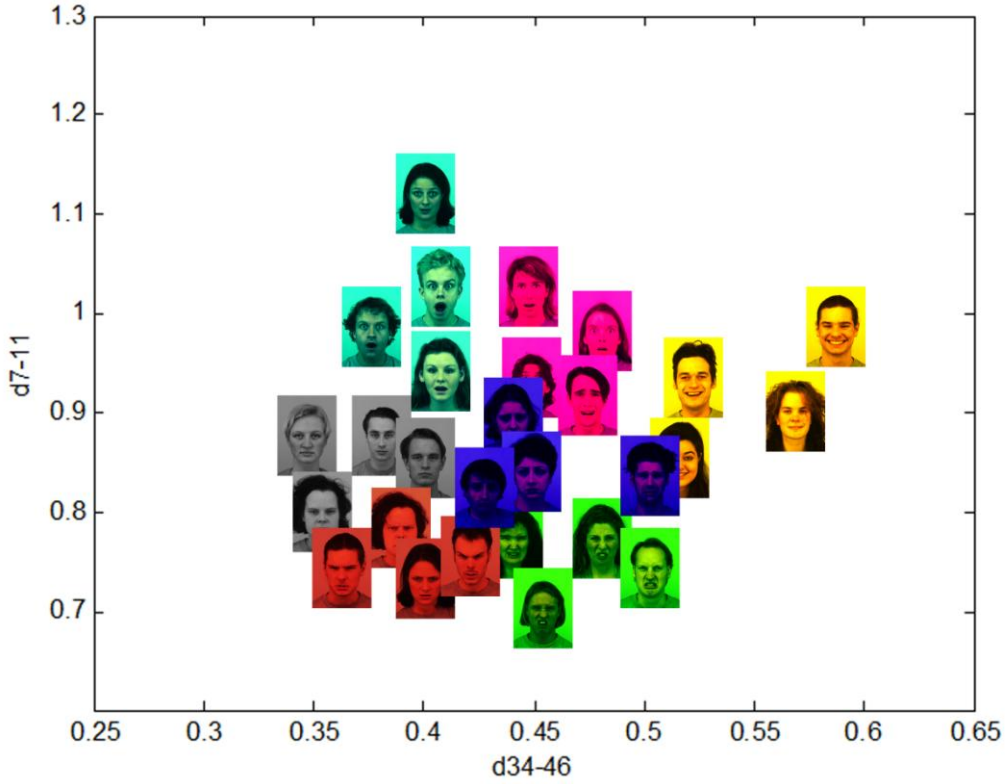

Figura 4.1.23: Ejemplo de clasificación con 2 características

Obviamente este es un pequeño ejemplo de lo que el algoritmo estaría tratando de recrear, con tan sólo dos características es difícil catalogar las clases con gran acierto por falta de información. Es por ello que en la mayoría de las clasificaciones de este tipo se trata de utilizar el mayor número posible de características relevantes.

4.2. AUDIO

De igual modo que con la imagen, en el audio se deben extraer ciertas características para que los clasificadores sepan cómo proceder correctamente.

En una prueba inicial, se procedió a realizar esta clasificación con los parámetros básicos mencionados anteriormente, extraídos directamente del pitch y la señal muestreada.

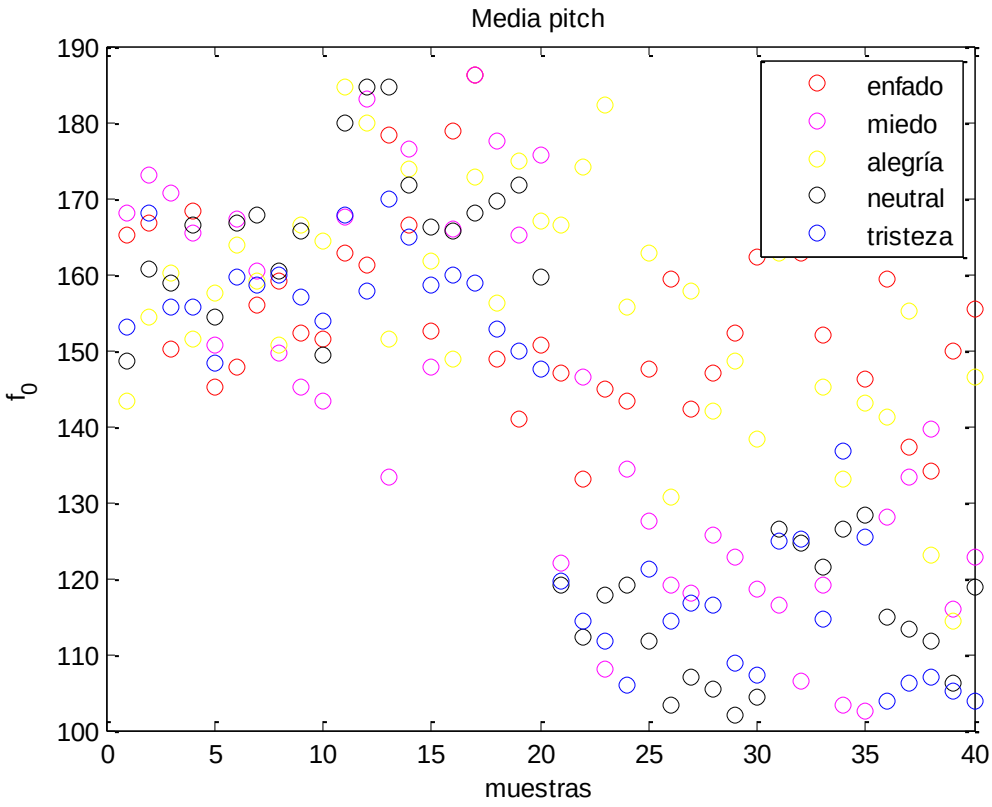

Figura 4.2.1: Valores de media del pitch

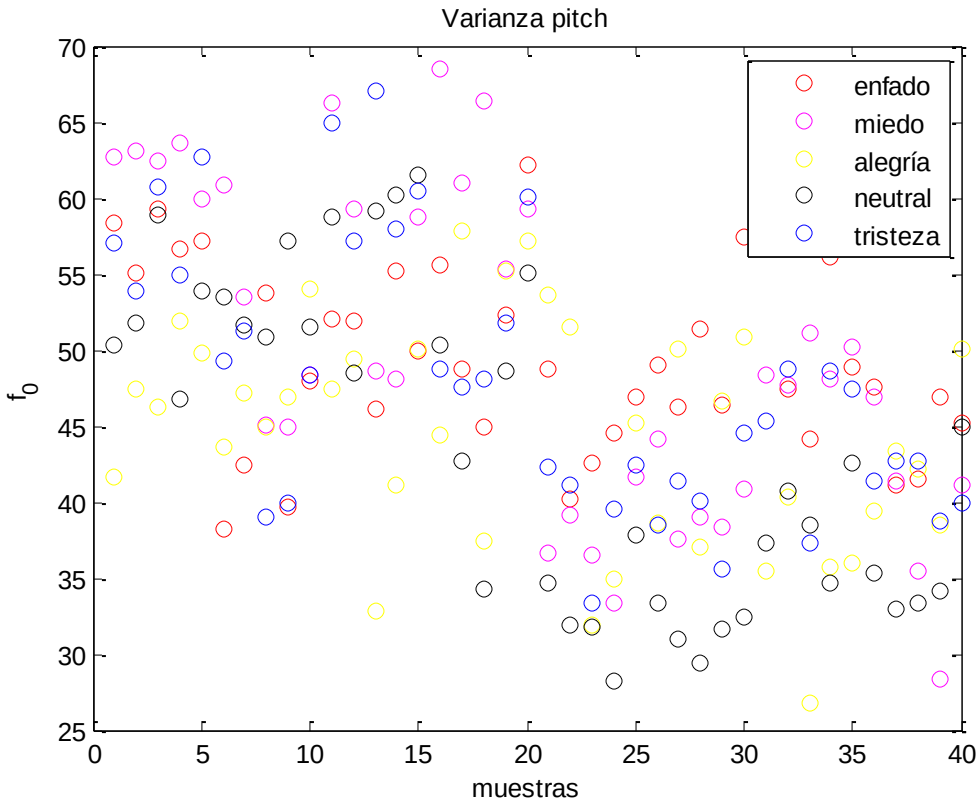

Figura 4.2.2: Valores de varianza del pitch

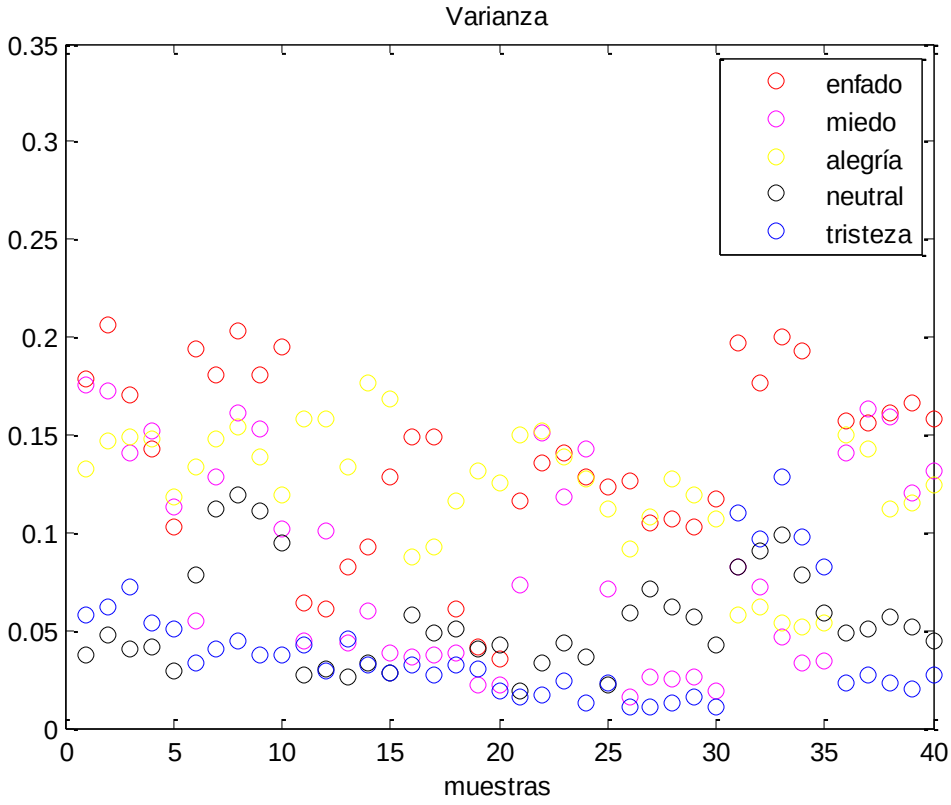

Figura 4.2.3: Valores de varianza de la señal

	Neutral	Alegría	Tristeza	Miedo	Enfado
Neutral	23	1	9	5	2
Alegría	3	25	0	6	6
Tristeza	7	0	26	6	1
Miedo	6	6	12	2	14
Enfado	2	10	2	4	22

Figura 4.2.4: Matriz de confusión

	Precisión	Recall	F
Neutral	0.561	0.575	0.568
Alegría	0.595	0.625	0.610
Tristeza	0.531	0.650	0.584
Miedo	0.087	0.050	0.063
Enfado	0.489	0.550	0.518

Figura 4.2.5: Precisión Recall y F de cada emoción

Correctamente clasificado	**49%**

Como ya se podía intuir en las gráficas anteriores, la emoción peor clasificada de las 5 presentes en la base de datos, es el miedo, clasificando correctamente tan sólo 2 de las 40 muestras existentes, lo que provoca que los parámetros de medida sean tan bajos (0.087, 0.050 y 0.063)

La siguiente prueba que se realizó, fue añadiéndole características derivadas del cepstrum, como son la media o la varianza de cada uno de los coeficientes (MFCC).

	Neutral	Alegría	Tristeza	Miedo	Enfado
Neutral	29	0	7	0	4
Alegría	1	30	0	0	9
Tristeza	4	0	27	9	0
Miedo	5	2	5	22	6
Enfado	0	8	2	4	26

Figura 4.2.6: Matriz de confusión

	Precisión	Recall	F
Neutral	0.744	0.725	0.734
Alegría	0.750	0.750	0.750
Tristeza	0.659	0.675	0.667
Miedo	0.629	0.550	0.587
Enfado	0.578	0.650	0.612

Figura 4.2.7: Precisión Recall y F de cada emoción

Correctamente clasificado	**67%**

Comparándolo con la prueba anterior, podemos ver que todo mejora de forma plausible aunque cabe destacar la gran mejoría que se produce en el apartado del miedo, pues de prácticamente no clasificar ninguna bien (tan sólo 2), en este apartado conseguimos un total de 22 muestras clasificadas correctamente.

Como última mejora, se procedió a añadir al clasificador, características del LPC (media y varianza).

	Neutral	Alegría	Tristeza	Miedo	Enfado
Neutral	31	0	5	4	0
Alegría	1	32	0	1	6
Tristeza	6	0	28	6	0
Miedo	2	0	7	27	4
Enfado	1	6	0	1	32

Figura 4.2.8: Matriz de confusión

	Precisión	Recall	F
Neutral	0.756	0.775	0.765
Alegría	0.842	0.800	0.821
Tristeza	0.700	0.700	0.700
Miedo	0.692	0.675	0.684
Enfado	0.750	0.750	0.750

Figura 4.2.9: Precisión Recall y F de cada emoción

Correctamente clasificado	**75%**

Se produce una mejora general en todas las emociones, dando lugar a un total del 75% correctamente clasificado. Al igual que en la parte

de imagen, se procedió a realizar un estudio con diversos clasificadores que el programa Weka ofrece, siendo finalmente el más óptimo el clasificador SVM[27]. Del mismo modo se realizó un estudio en busca de los parámetros de gamma y costo más óptimos.

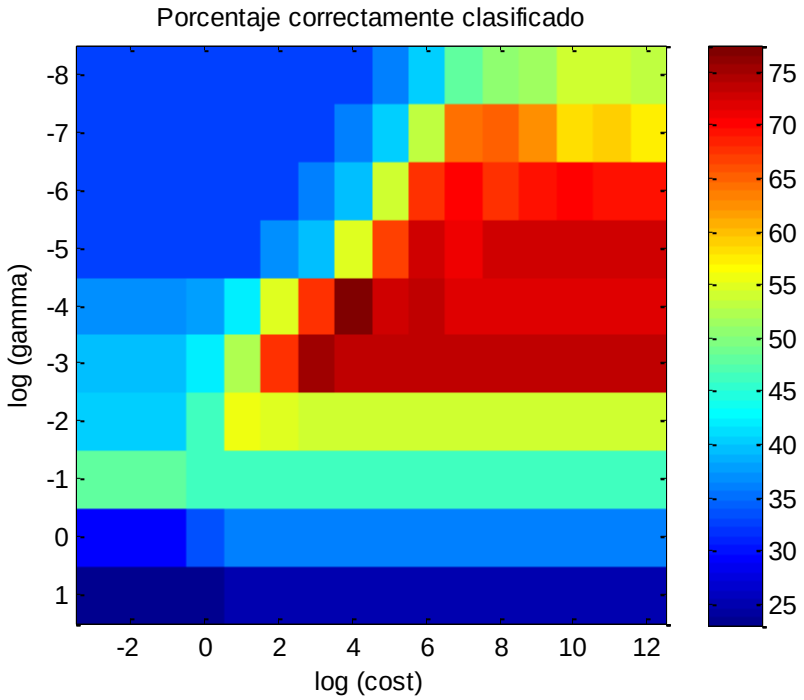

Figura 4.2.10: Análisis de los parámetros gamma y cost en escala logarítmica

En este caso, estos parámetros corresponden a los puntos de coste = 10000 y gamma = 0.0001, dando lugar a un acierto total del 77.5%. Podemos ver las características finales del clasificador en los siguientes gráficos:

	Neutral	Alegría	Tristeza	Miedo	Enfado
Neutral	35	0	3	2	0
Alegría	0	33	0	2	5
Tristeza	4	0	33	3	0
Miedo	0	2	8	27	3
Enfado	1	8	0	4	2

Figura 4.2.11: Matriz de confusión

	Precisión	Recall	F
Neutral	0.875	0.875	0.875
Alegría	0.767	0.825	0.795
Tristeza	0.750	0.825	0.795
Miedo	0.711	0.675	0.692
Enfado	0.771	0.675	0.720

Figura 4.2.12: Precisión Recall y F de cada emoción

Correctamente clasificado	**77.5%**

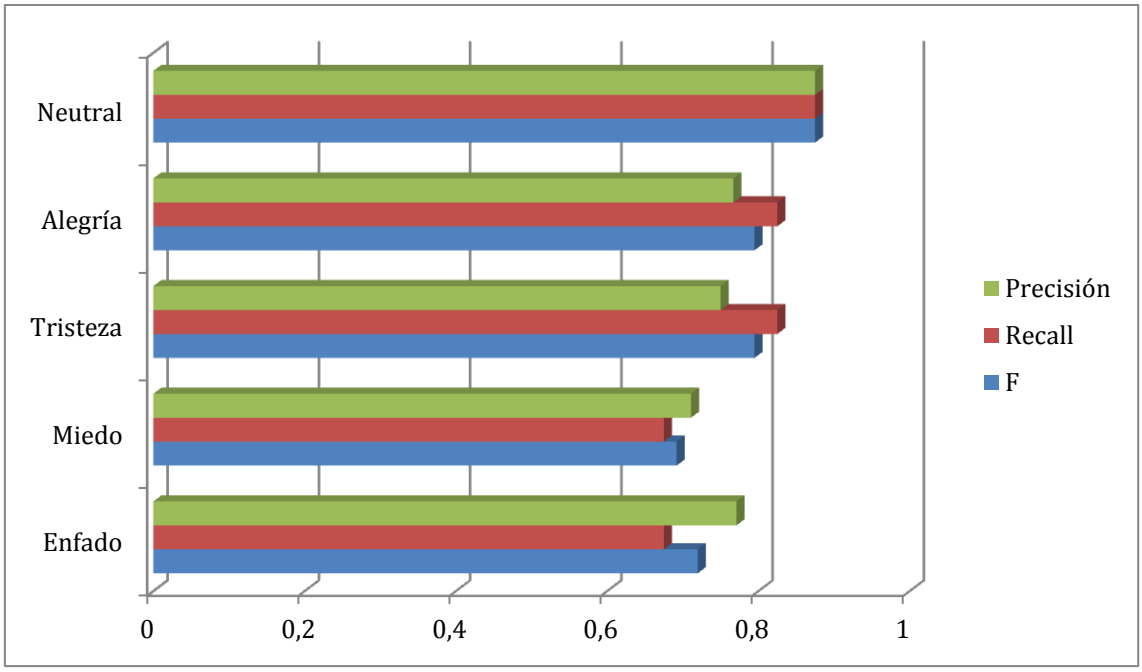

Figura 4.2.13: Gráfica de Precisión Recall y F de cada emoción

También se realizó una clasificación de las características utilizadas en función de su relevancia a la hora de clasificar emociones. En el anexo E se puede ver esta clasificación, y en la figura 4.2.14 se puede apreciar como el algoritmo mejora a medida que aumentamos el número de características.

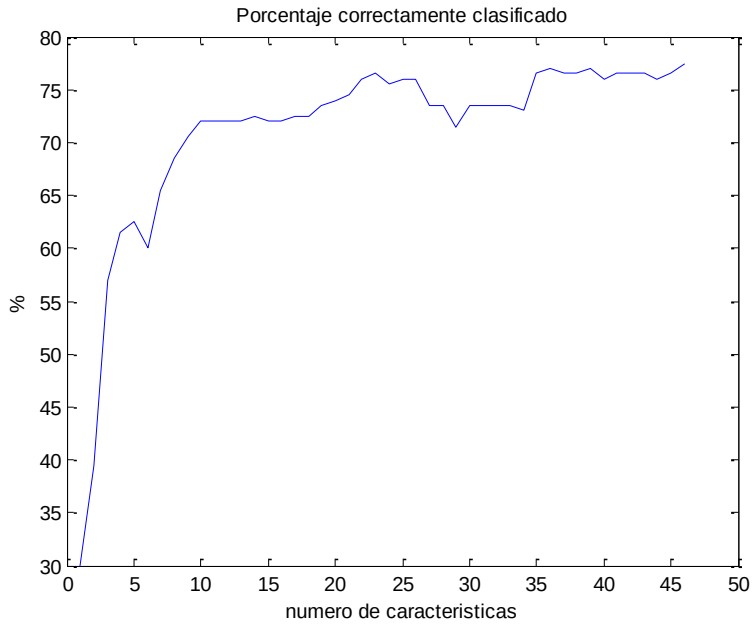

Figura 4.2.14: Precisión Recall y F de cada emoción

Nuevamente, al igual que en la parte de imagen, se procedió a intentar mejorar el resultado obtenido con las técnicas de Bagging y Adaboost, dando lugar a unos resultados presentes en la siguiente tabla:

Bagging	**Adaboost**
71%	77%

5. PRUEBAS Y CONCLUSIONES

Por último, se realizaron unas pruebas con sujetos reales. Fueron un total de 7 individuos, cuyos resultados se adjuntan en la figura 4.1.13. En líneas generales, el comportamiento del sistema es similar al previsto anteriormente, el porcentaje correctamente clasificado es de 73.47% frente al 77.5027%. Sin embargo, encontramos que algunas emociones como neutral y tristeza tienen comportamientos peores al esperado. La primera de ellas, neutral, peca de tener una mala Recall, pues apenas acierta 2 de los 7 sujetos, mientras que en tristeza, a pesar de tener un Recall del 100%, encontramos que la precisión del mismo desciende al 43.75% debido a los falsos positivos que aparecen en otras emociones.

	Neutral	Alegría	Tristeza	Miedo	Enfado	Asco	Sorpresa
Neutral	2	0	5	0	0	0	0
Alegría	0	7	0	0	0	0	0
Tristeza	0	0	7	0	0	0	0
Miedo	1	0	1	5	0	0	0
Enfado	0	0	1	1	5	0	0
Asco	0	1	1	0	1	4	0
Sorpresa	0	0	1	0	0	0	6

Figura 4.1.13: Matriz de confusión

Correctamente clasificado	**73.47%**

Como conclusión final, se puede decir que hemos llegado a unos buenos resultados en la clasificación, similares a otros proyectos de mismo índole. En cuanto a líneas futuras, se podría contemplar la inserción de nuevas características tomadas con un sensor de profundidad de mayor precisión, así como la elaboración de un sistema completo con características provenientes de los 3 sensores. También se podría estudiar la idea de trabajar con los frames previos para mejorar la detección.

Con este trabajo se han aprendido técnicas de aprendizaje automático y procesado de audio y vídeo, así como ver el funcionamiento de una línea de investigación. A nivel personal puedo decir que me siento satisfecho con lo aprendido en este proyecto, de haber explorado una rama de conocimiento nueva para mí y de la forma en la que se llevó a cabo.

BIBLIOGRAFÍA

[1] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[2] B. Van Ginneken, A. F. Frangi, J. J. Staal, M. Bart, H. Romeny, and M. a Viergever, "Optimal Features," vol. 21, no. 8, pp. 924–933, 2002.

[3] B. S. Atal, "The history of linear prediction," *IEEE Signal Process. Mag.*, vol. 23, no. 2, 2006.

[4] P. Belhumeur, J. Hespanha, D. Kriegman, and ÒEigenfaces vs. Fisherfaces, "Recognition Using Class Specific Linear Projection." IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 711–720., 1997.

[5] D. Masip, M. S. North, A. Todorov, and D. Osherson, "Automated prediction of preferences using facial expressions. PLOS One." .

[6] D. Lundqvist, A. Flykt, and A. Öhman, "The Karolinska Directed Emotional Faces," *Psychology*, no. KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, pp. 3–5, 1998.

[7] "Database of Polish Emotional Speech." Medical Electronics Division, Technical University of Lodz, Poland.

[8] M. Lyons and S. Akamatsu, "Coding Facial Expressions with Gabor Wavelets," pp. 200–205, 1998.

[9] M. J. Lyons, "Automatic classification of single facial images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 12, pp. 1357–1362, 1999.

[10] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, and A. van Knippenberg, "Presentation and validation of the Radboud Faces Database. Cognition & Emotion." pp. 1377–1388, 2010.

[11] P. Peer, "CVL Face Database." .

[12] F. Solina, P. Peer, B. Batagelj, S. Juvan, and J. Kova, "Color-based face detection in the '15 seconds of fame' art installation", In: Mirage 2003, Conference on Computer Vision / Computer Graphics Collaboration for Model-based Imaging, Rendering, image Analysis and Graphical special Effects." INRIA Rocquencourt, France, Wilfried Philips, Rocquencourt, INRIA, pp. 38–47, 2003.

[13] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 896–903, 2013.

[14] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark Localization Challenge," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 397–403, 2013.

[15] J. P. Arias, C. Busso, and N. B. Yoma, "Energy and Fo contour modeling with Functional Data Analysis for Emotional Speech Detection," *Interspeech2013*, no. August, pp. 2871–2875, 2013.

[16] O. Pierre-Yves, "The production and recognition of emotions in speech: Features and algorithms," *Int. J. Hum. Comput. Stud.*, vol. 59, no. 1–2, pp. 157–183, 2003.

[17] Y. Liu, "Emotion Recognition From Speech Signals," 2013.

[18] P. Viola and M. Jones, "Robust Real-time Face Detection," vol. 20, p. 2142, 2000.

[19] J. Whitehill and C. W. Omlin, "Haar features for FACS AU recognition," *FGR 2006 Proc. 7th Int. Conf. Autom. Face Gesture Recognit.*, vol. 2006, pp. 97–101, 2006.

[20] R. E. Schapire, "Explaining AdaBoost," pp. 1–16, 2010.

[21] G. Tzimiropoulos, "Project-Out Cascaded Regression with an application to Face Alignment," no. School of Computer Science, University of Nottingham.

[22] "http://www.mathworks.com/matlabcentral/fileexchange/1230-pitch-determination-algorithm/content/shrp.m." .

[23] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[24] J. Ross, Q. Morgan, and K. Publishers, "Book Review : C4 . 5 : Programs for Machine Learning," *Mach. Learn.*, vol. 1, pp. 235–240, 1994.

[25] E. F. and M. H. Marc Sumner, "Speeding Up Logistic Model Tree Induction," *Lect. Notes Comput. Sci.*, vol. 3721, pp. 675–683, 2005.

[26] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Mach. Learn.*, vol. 59, no. 1–2, pp. 161–205, 2005.

[27] C. Chang and C. Lin, "LIBSVM : A Library for Support Vector Machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 1–39, 2011.

[28] X. Wang and J. Tian, "Gene Selection for Cancer Classification using Support Vector Machines," *Comput. Math. Methods Med.*, vol. 2012, p. 586246, 2012.

ANEXO A

A continuación se puede ver una simulación de la implementación del algoritmo PO_CR.

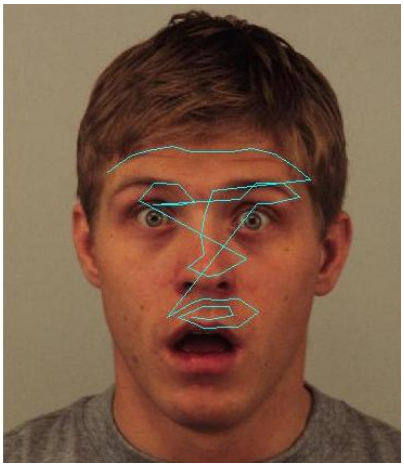

Figura A.1: Primera iteración del algoritmo

En la primera imagen (figura A.1) se ve como sobre la imagen se ha inicializado la forma de la cara con los puntos FCPs, sin embargo no está en la posición correcta.

En las siguientes imágenes se ve como el algoritmo intenta llegar a la situación óptima. De forma iterativa se irán produciendo operaciones de rotación, traslado y escalado, moviendo los puntos por toda la imagen en función de la forma y textura actual en la que se encuentren. Estas operaciones se repetirán hasta que el algoritmo converja.

Figura A.2: Traslación de los puntos hacia posición óptima

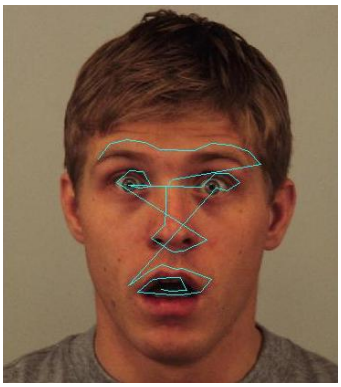

Figura A.3: Traslación de los puntos hacia posición óptima

Finalmente, y si el algoritmo tiene éxito, podemos ver como los puntos de interés casan perfectamente en la cara con los puntos que representan.

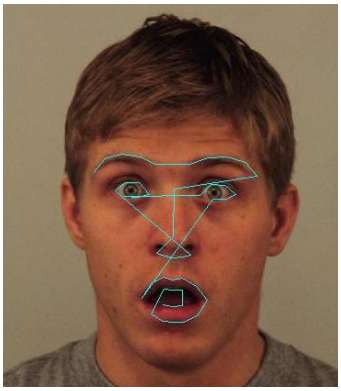

Figura A.4: Posición final de los puntos,

ANEXO B

1) Elegir pares de puntos (x,y) a detectar

$$x_i = (x_{i0}, y_{i0}, x_{i1}, y_{i1}, \dots)$$

2) Minimizar E

$$E = (x_1 - M(s,\theta)[x_2] - t)^T \cdot W \cdot (x_1 - M(s,\theta)[x_2] - t)$$

3) Se define la operación de rotación y escalado

$$M(s,\theta)\begin{bmatrix} x_{jk} \\ y_{jk} \end{bmatrix} = \begin{bmatrix} (s \cdot cos\theta)x_{jk} - (s \cdot \sin\theta) \cdot y_{jk} \\ (s \cdot sin\theta)x_{jk} + (s \cdot \cos\theta) \cdot y_{jk} \end{bmatrix}$$

$$t_j = (t_{xj}, t_{yj}, \dots)^T$$

$$\begin{bmatrix} X_2 & -Y_2 & W & 0 \\ Y_2 & X_2 & 0 & W \\ Z & 0 & X_2 & Y_2 \\ 0 & Z & -Y_2 & X_2 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ C_1 \\ C_2 \end{bmatrix}$$

$$X_i = \sum_{k=0}^{n-1} w_k \cdot x_{ik}$$

$$Y_i = \sum_{k=0}^{n-1} w_k \cdot y_{ik}$$

$$Z = \sum_{k=0}^{n-1} w_k \cdot (x_{2k}^2 + y_{2k}^2)$$

$$W = \sum_{k=0}^{n-1} w_k$$

$$C_1 = \sum_{k=0}^{n-1} w_k \cdot (x_{1k}x_{2k} + y_{1k}y_{2k})$$

$$C_2 = \sum_{k=0}^{n-1} w_k \cdot (y_{1k}x_{2k} - x_{1k}y_{2k})$$

Siendo W la matriz diagonal de pesos para cada punto y $a_x = s \cdot \cos\theta$ y $a_y = s \cdot \sin\theta$ y (θ, s, t_x, t_y) las 4 variables disponibles para cada forma.

4) Elegir pesos

Los pesos son usados para evaluar la importancia de los puntos. A los puntos con mayor estabilidad se les asigna una importancia mayor que al resto. Para ello definiremos:

-R_{kl} : distancia entre los puntos k y l de una forma.

-V_{Rkl} : variación de la distancia sobre el conjunto de formas.

5) Se usa el siguiente algoritmo iterativo para alinear las formas:

-Rotar, escalar y trasladar cada forma para alinear con la primera forma en el conjunto

-Repetir iterativamente:

-Calcular forma media de las formas alineadas

-Normalizar la orientación, escala y origen de la media actual

-Realinear cada forma en la actual media

-Hasta que el proceso converja

ANEXO C

	Imagen	Audio
J48	60%	56%
J48 Consolidated	60.71%	53%
J48 Graft	60.92%	56.5%
LAD Tree	65.12%	55.5%
LMT	76.64%	73%
Random Forest	72.3%	70%
Random Tree	58.1%	55%
REP Tree	62.75%	47%
SimpleCart	62.65%	53.5%
HoeffdingTree	53.37%	60%
FT	73.95%	70.5%
ExtraTree	58.88%	41.5%
DecisionStump	71.47%	39.5%
BayesNet (TAN)	68%	59.5%
NaiveBayes	57%	58.5%
NaiveUpdatable	57.2%	58.5%

ANEXO D

d34_47, d10_42, d32_38, d19_47, d3_42, d7_27, d27_31, d43_49, d16_19, d25_32, d15_49, d2_42, d35_44, d32_44, d35_47, d2_24, d6_35, d20_42, d24_26, d21_25, d5_22, d10_43, d30_38, d10_29, d1_25, d3_11, d40_45, d12_26, d15_19, d38_46, d35_46, d18_34, d3_27, d41_44, d34_48, d1_24, d42_49, d45_47, d15_34, d38_40, d2_26, d16_43, d5_7, d22_26, d14_16, d3_26, d22_24, d2_3, d16_18, d32_45, d6_12, d47_48, d21_22, d40_44, d9_43, d24_32, d1_5, d4_11, d15_47, d15_18, d20_32, d1_49, d40_41, d22_23, d41_48, d42_43, d13_24, d6_26, d16_34, d34_44, d2_25, d5_10, d6_13, d44_48, d3_8, d1_27, d34_40, d33_44, d15_48, d40_43, d12_28, d27_30, d5_21, d17_43, d29_38, d34_35, d18_33, d40_48, d20_25, d46_47, d26_31, d31_38, d6_45, d17_36, d2_6, d20_21, d29_30, d5_9, d33_43, d45_49, d4_26, d39_47, d22_31, d14_48, d32_49, d2_21, d32_46, d19_48, d4_42, d27_28, d36_37, d30_43, d40_47, d6_11, d39_43, d15_35, d21_24, d13_26, d18_19, d13_23, d1_20, d20_22, d3_31, d44_49, d23_32, d6_36, d5_23, d45_48, d32_37, d46_48, d28_30, d48_49, d9_29, d5_8, d23_24, d14_41, d12_29, d2_27, d6_27, d18_43, d32_35, d2_49, d41_49, d8_9, d16_47, d2_23, d34_45, d20_34, d2_20, d32_39, d24_25, d2_11, d41_45, d1_3, d7_22, d12_23, d32_43, d2_31, d16_49, d1_32, d6_19, d2_22, d28_29, d17_38, d44_45, d14_17, d3_12, d1_28, d6_22, d23_31, d1_33 d5_20, d22_49, d2_43, d44_47, d26_38, d6_46, d9_10, d34_39, d17_49, d32_36, d24_49, d39_48, d40_46, d8_22, d4_27, d10_30, d22_25, d35_39, d25_49, d17_33, d21_23, d3_6, d13_31, d21_32, d19_29, d15_33, d13_15, d1_23, d33_38, d41_42, d16_38, d6_18, d11_20, d41_47, d32_42, d30_31, d29_43, d37_45, d6_31, d1_26, d35_48, d22_32, d21_49, d12_24, d4_31, d42_44, d19_23, d38_45, d6_34, d2_8, d35_45, d28_43, d33_34, d24_48, d1_42, d20_23, d17_35, d7_21, d28_38, d5_41, d12_31, d43_48, d31_43, d38_43, d34_46, d10_27, d19_24, d1_4, d1_43, d4_21, d14_23, d2_48, d19_31, d2_9, d37_38, d17_37, d1_34, d41_46, d17_19, d36_46, d39_49, d6_23, d9_30, d3_30, d1_21, d19_43, d3_23, d6_15, d20_24, d22_27, d14_29, d27_38, d38_39, d5_24, d10_20, d7_20, d33_35, d6_44, d40_49, d16_35, d36_38, d33_45, d27_29, d13_29, d32_48, d46_49, d4_10, d24_31, d37_46, d14_26, d18_23, d18_26, d12_13, d2_28, d3_44, d18_36, d38_47, d26_27, d5_6, d3_9, d39_46, d10_49, d4_9, d22_48, d5_42, d4_12, d6_37, d3_10, d45_46, d12_30, d6_21, d14_19, d32_33, d5_12, d13_16, d23_25, d16_37, d42_48, d19_25, d2_47, d20_41, d2_30, d1_6, d3_4, d14_24, d11_28, d5_25, d8_21, d21_31, d17_32, d6_30, d3_24, d10_38, d11_38, d35_49, d21_48, d8_10, d6_16, d4_23, d14_15, d22_47, d25_26, d18_32, d6_14, d15_16, d26_48, d15_43, d38_44, d17_46, d37_44, d3_28, d29_35, d13_45, d2_12, d15_46, d23_27, d19_44, d11_26, d18_38, d10_33, d26_29, d28_31, d6_28, d18_24, d5_40, d1_31, d8_30, d15_38, d19_32, d20_26, d6_24, d16_42, d20_33, d13_19, d33_37, d2_29, d1_38, d13_30, d35_36, d3_49, d3_13, d4_44, d32_34, d11_13, d20_49, d15_36, d4_5, d2_10, d6_20, d26_37, d43_44, d5_47, d2_4, d29_31, d10_28, d3_29, d39_40, d10_32, d13_20, d20_31, d12_38, d35_40, d1_29, d8_20, d35_38, d6_17, d5_48, d19_20, d26_49, d17_42, d32_47, d1_7, d40_42, d7_30, d10_45, d2_32, d4_45, d10_24, d16_48, d27_43, d20_35, d19_22, d13_25, d38_41, d5_19, d7_29, d16_46, d5_27, d20_43, d9_23, d5_34, d14_45, d1_22, d17_22, d2_7, d16_39, d18_42, d26_39, d39_41, d19_39, d21_26, d3_43, d20_36, d31_39, d10_23, d18_44, d16_17, d1_8, d7_28, d41_43, d29_45, d18_41, d16_26, d6_33, d21_47, d4_41, d10_14, d20_44, d4_6, d44_46, d11_12, d5_11, d17_47, d32_40, d31_32, d16_23, d30_39, d19_26, d6_25, d10_44, d6_7, d2_13, d11_27, d15_17, d10_22, d5_43, d20_45, d14_44, d9_42, d33_46, d3_7, d1_35, d13_38, d17_23, d9_38, d36_48, d3_25, d20_29, d7_23, d3_5, d8_29, d2_5, d5_49, d19_38, d26_47, d16_33, d14_20, d34_36, d5_26, d43_47, d2_38, d14_42, d14_35, d13_44, d3_22, d5_15, d36_49, d10_13, d5_35, d29_39, d22_43, d39_42, d32_41, d19_30, d17_34, d14_30, d4_49, d3_14, d39_44, d10_16, d1_9, d12_44, d26_43, d6_38, d12_19, d4_30, d16_36, d18_25, d17_21, d5_31, d12_27, d5_28, d5_13, d4_43, d4_13, d20_46, d9_49, d8_38, d1_44, d10_21, d6_29, d29_36, d1_10, d1_12, d18_31, d11_29, d11_22, d23_47, d10_15, d20_48, d17_18, d3_45, d13_43, d21_43, d14_25, d23_48, d6_39, d39_45, d20_47, d4_25, d19_49, d16_24, d18_49, d10_41, d14_34, d21_34, d14_31, d18_20, d11_44, d4_48, d26_30, d2_15, d30_37, d29_34, d15_39, d12_43, d47_49, d7_11, d23_38, d18_29, d42_47, d33_49, d17_45, d7_25, d18_22, d1_2, d18_48, d5_18, d9_33, d20_30, d7_26, d33_36, d5_46, d17_24, d11_25, d3_21, d4_7, d25_34, d12_15, d7_24, d3_32, d10_17, d3_41, d7_10, d13_14, d21_38, d14_49, d1_30, d7_38, d25_35, d25_31, d31_48, d11_43, d4_8, d7_39, d3_19, d12_45, d15_40, d17_48, d13_28, d37_48, d29_37, d6_10, d14_38, d8_23, d15_37, d36_39, d10_26, d34_38, d7_19, d13_17, d25_36, d5_32, d29_33, d24_47, d12_20, d33_48, d20_37, d4_46, d31_37, d4_20, d23_49, d9_28, d3_48, d37_49, d5_16, d29_48, d11_45, d17_41, d16_44, d35_37, d5_45, d15_26, d28_32, d15_44, d38_42, d21_35, d18_37, d18_21, d33_47, d11_39, d18_45, d21_42, d9_41, d11_14, d7_31, d10_19, d19_35, d17_39, d21_33, d29_44, d4_14, d19_34, d36_45, d6_47, d10_12, d29_46, d4_32, d19_36, d25_30, d9_24, d14_43, d5_36, d11_23, d22_38, d4_24, d24_38, d13_18, d9_20, d29_49, d28_39, d3_20, d25_48, d25_33, d1_45, d27_32, d25_42, d37_47, d13_46, d2_14, d37_39, d11_47, d18_40, d14_18, d14_46, d10_34, d19_33, d19_21, d29_47, d16_32, d10_31, d14_37, d9_48, d20_40, d10_18, d5_14, d16_22, d4_29, d27_39, d23_29, d9_45, d9_22, d25_47, d42_46, d11_32, d8_25, d28_37, d5_39, d11_46, d4_28, d3_18, d30_32, d18_47, d4_19, d8_43, d1_48, d13_22, d5_44, d42_45, d9_21, d11_30, d1_13, d7_47, d38_49, d5_37, d7_9, d16_40, d28_42, d15_24, d27_37, d4_47, d11_15, d21_36, d7_8, d5_33, d34_37, d34_43, d13_41, d8_24, d1_36, d10_48, d22_33, d31_47, d19_45, d21_44, d12_16, d10_25, d12_46, d9_44, d12_25, d24_43, d1_15, d1_11, d6_40, d25_38, d10_11, d9_40, d22_34, d11_49, d6_41, d33_39, d15_23, d18_35, d2_19, d12_14, d1_16, d5_17, d6_48, d14_33, d9_32, d13_34, d4_34, d2_16, d13_48, d31_49, d25_29, d16_41, d13_27, d17_27, d37_42, d7_40, d30_36, d12_39, d23_40, d3_46, d20_38, d38_48, d12_22, d24_33, d16_45, d14_28, d36_47, d14_22, d5_29, d6_8, d11_21, d24_30, d26_41, d11_19, d4_35, d13_35, d6_9, d28_49, d37_43, d33_42, d25_41, d21_45, d26_40, d23_33, d23_37, d34_42, d5_30, d8_31, d6_32, d3_38, d2_34, d11_24, d27_42, d12_47, d10_39, d26_46, d15_25, d9_27, d16_27, d25_37, d25_28, d30_48, d22_28, d19_37, d15_45, d25_44, d23_28, d30_33, d13_42, d3_37, d24_34, d17_31, d15_42, d28_35, d3_15, d9_16, d18_30, d4_22, d7_12, d2_44, d17_25, d9_19, d22_42, d7_41, d21_41, d9_26, d4_40, d12_42, d7_46, d20_39, d17_28, d21_28, d24_35, d8_27, d8_28, d9_17, d22_35, d34_41, d25_43, d3_34, d17_26, d4_33, d6_42, d20_28, d22_30, d15_20, d36_40, d8_11, d11_48, d29_32, d24_27, d16_21, d3_47, d4_38, d4_16, d12_49, d28_45, d7_48, d2_17, d7_45, d2_41, d3_35, d13_49, d3_17, d17_44, d1_14, d2_45, d33_40, d28_33, d12_21, d27_35, d16_31, d36_42, d4_37, d1_47, d10_37, d25_45, d34_49, d13_47, d43_46, d24_29, d4_17, d27_49, d31_36, d17_20, d8_39, d4_18, d14_40, d1_17, d11_40, d26_28, d12_32, d9_35, d21_46, d26_34, d11_42, d19_46, d3_36, d4_15, d35_41, d14_27, d22_44, d31_42, d2_35, d7_37, d30_49, d23_46, d2_39, d9_31, d36_43, d8_46, d16_28, d24_37, d7_18, d3_40, d30_42, d11_16, d7_36, d21_27, d8_47, d20_27, d2_18, d9_34, d12_18, d25_27, d24_36, d22_29, d13_33, d2_36, d9_15, d9_39, d27_33, d23_36, d26_44, d36_44, d18_46, d11_33, d12_34, d14_21, d7_42, d8_45, d26_45, d6_49, d1_46, d30_47, d14_47, d23_34, d4_36, d13_36, d11_31, d25_46, d5_38, d13_40, d16_25, d26_36, d13_39, d10_35, d12_35, d7_35, d21_29, d27_45, d27_46, d12_33, d2_37, d14_39, d15_32, d3_16, d28_34, d25_40, d29_41, d9_25, d21_30, d31_33, d37_41, d8_19, d28_46, d17_29, d13_21, d11_37, d9_46, d15_29, d27_34, d19_40, d8_44, d11_17, d36_41, d30_35, d8_40, d16_29, d12_41, d23_39, 28_41, d30_46, d15_27, d3_33, d9_47, d12_36, d25_39, d11_18, d2_40, d26_42, d35_42, d3_39, d7_49, d13_37, d31_34, d24_44, d19_27, d7_44, d1_41, d28_44, d21_39, d10_36, d17_40, d26_32, d15_31, d22_41, d19_42, d21_37, d8_32, d31_46, d7_34, d28_36, d12_48, d1_37, d23_30, d19_41, d7_13, d23_35, d9_18, d14_36, d24_42, d26_35, d2_46, d21_40, d24_40, d8_41, d9_14, d9_11, d4_39, d7_43, d35_43, d27_47, d26_33, d12_40, d22_36, d7_17, d29_40, d31_35, d28_47, d9_36, d23_41, d28_48, d43_45, d16_20, d15_28, d19_28, d23_42, d8_15, d8_48, d27_41, d18_27, d22_37, d8_42, d10_47, d30_45, d22_39, d12_37, d8_12, d29_42, d11_34, d23_43, d23_44, d11_41, d28_40, d9_37, d30_34, d22_45, d11_35, d27_40, d27_36, d8_26, d15_22, d8_49, d6_43, d12_17, d18_39, d27_44, d27_48, d11_36, d1_40, d7_14, d18_28, d7_15, d9_12, d17_30, d24_41, d30_40, d23_45, d31_41, d33_41, d8_17, d7_33, d9_13, d22_46, d30_41, d8_37, d1_18, d8_33, d14_32, d2_33, d7_32, d7_16, d10_40, d15_41, d1_39, d15_21, d24_46, d31_40, d24_28, d8_18, d16_30, d24_39, d10_46, d24_45, d30_44, d8_35, d8_36, d37_40, d1_19, d13_32, d31_44, d8_34, d22_40, d8_13, d8_14, d15_30, d31_45, d8_16, d23_26.

ANEXO E

media coeficiente 4 cepstrum
media coeficiente 10 cepstrum
media coeficiente 2 lpc
varianza coeficiente 5 cepstrum
media de varianzas de ventanas de la señal
varianza coeficiente 1 cepstrum
media coeficiente 1 cepstrum
varianza coeficiente 4 cepstrum
media coeficiente 2 cepstrum
varianza coeficiente 6 lpc
varianza coeficiente 8 lpc
media coeficiente 8 lpc
media coeficiente 9 lpc
media del pitch
varianza coeficiente 9 cepstrum
varianza coeficiente 9 lpc
media coeficiente 8 cepstrum
media coeficiente 3 lpc
media coeficiente 7 cepstrum
varianza coeficiente 6 cepstrum
varianza coeficiente 3 lpc
varianza coeficiente 2 cepstrum
varianza señal
varianza coeficiente 7 cepstrum
varianza coeficiente 5 lpc
media coeficiente 12 cepstrum
varianza del pitch
media coeficiente 5 cepstrum
varianza coeficiente 4 lpc
media coeficiente 6 cepstrum
media coeficiente 9 cepstrum
media coeficiente 7 lpc
varianza coeficiente 3 cepstrum
varianza coeficiente 10 cepstrum
intensidad máxima
media coeficiente 3 cepstrum
varianza coeficiente 8 cepstrum
media coeficiente 5 lpc
varianza coeficiente 7 lpc
media coeficiente 6 lpc
media coeficiente 11 cepstrum
media coeficiente 4 lpc
varianza coeficiente 11 cepstrum
varianza coeficiente 12 cepstrum
varianza coeficiente 2 lpc
intensidad mínima