Universidad Zaragoza

Trabajo Fin de Grado

Reconocimiento automático de acciones humanas en secuencias de vídeo con cámara 3D

Automated human actions recognition in 3D video sequences

Autor

Tomás Berriel Martins

Director

Carlos Orrite Uruñuela

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2019

Reconocimiento automático de acciones humanas en secuencias de vídeo con cámara 3D

RESUMEN

El objetivo de este trabajo es determinar las articulaciones más relevantes para el análisis de cada clase de acciones, así como diseñar un clasificador de acciones humanas a partir de la información de un sensor de posición 3D (Kinect). Se parte del análisis de la base de datos, y continua con el estudio de los parámetros de una red Long Short Term Memory y el desarrollo de tres clasificadores distintos basados en dichas redes LSTM.

El trabajo se encuentra dividido en 5 partes:

- Filtrado y normalización de las secuencias de la base de datos.
- Estudio de las componentes más características de cada acción.
- Estudio de la configuración de red neuronal y desarrollo de los clasificadores.
- Ensayos y resultados.
- Conclusiones.

Automated human actions recognition in 3D video sequences

ABSTRACT

The goal of this work is to determine the joints more relevants for the analysis of each class of actions, as well as design a human actions clasificator from the information of a 3D position sensor. It startse with the analisys of the dataset, and goes on studing a Long short term memory paramaters and developing three different clasificators based on LSTM networks.

The work is divided into 5 parts:

- Database sequences filtering and normalizations.
- Studying of the most caracteristics componentes of each action .
- Studying of the neural net configuration, and development of the clasificators.
- Tests and results.
- Conclusions.

Índice

Capítulo 1

Introducción y objetivos

1.1. Marco del TFG

El trabajo surge como respuesta a la necesidad de un sistema que permita automatizar el reconocimiento de actividad humana, en una sociedad tendiente cada vez más a la automatización de sus procesos. El alcance del trabajo es diseñar y estudiar un algoritmo capaz de determinar la acción realizada por una persona mediante la interpretación de las imágenes de profundidad de su esqueleto obtenidas a través de un sensor Kinect. Además, también se va a estudiar la influencia de las distintas características en el reconocimiento de la acción para determinar cuales son las más importantes para clasificar cada acción.

1.2. Estado del arte

El reconocimiento de acciones humanas es un problema muy importante pues tiene un gran abanico de aplicaciones: desde la videovigilancia de espacios públicos y privados, hasta el control de las actividades de personas con poca o ninguna autonomía en hospitales y centros de salud, pasando incluso por la vigilancia del cumplimiento de las normas de seguridad en plantas industriales u otros entornos. Debido al gran interés de este campo existe una gran variedad de algoritmos de clasificación basados en esqueletos humanos. [1] clasifica los algoritmos de reconocimiento de acciones humanas con Kinect en dos grupos, en función de como se extraen las descriptores de las características para representar las acciones humanas.

La primera categoría es la de las características manuales, la cual requiere de dos etapas de diseño manual para obtener el descriptor final: la extracción de las características, y la representación de las mismas.

La segunda categoría es la que incluye los algoritmos de aprendizaje profundo o deep

learning. Estos algoritmos reducen la necesidad de ingeniería de características, pero requieren de una gran cantidad de ejemplos etiquetados y de tiempo de entrenamiento. Shahroudy *et al.* [2] propone un método de aprendizaje basado en una regresión dispersa conjunta (joint sparse regression based learning method) que utiliza la dispersión estructurada para modelar cada acción como una combinación de características multimodales de un conjunto disperso de partes del cuerpo. En [3] propone una nueva red profunda basada en un autoencoder de factorización de características específicas para separar las señales de entradas multimodales en una jerarquía de componentes, además, propone una máquina de aprendizaje automático de estructuras dispersas. En [4] desarrolla a partir de una red Long Short Term Memory, o LSTM, la Part-Aware LSTM, o P-LSTM, una nueva red neuronal recurrente para modelar la correlación temporal a largo plazo de las características de cada parte del cuerpo.

Liu *et al.* [5] [6] añade una nueva puerta en la célula LSTM para aprender la fiabilidad de las características de entrada y ajustar sus efectos en la actualización de la información a largo plazo almacenada en la célula de memoria dando lugar a la Spatio-Temporal LSTM, o ST-LSTM, e introduce una nueva técnica de fusión de características dentro de la unidad LSTM. En [7] [8] propone la Global Context Aware LSTM, o GCA-LSTM, una nueva clase de red capaz de centrarse de forma selectiva en las características informativas de la secuencia de la acción, con la ayuda de la información de contexto global.

1.3. Objetivos

Se busca diseñar un método de reconocimiento de acciones humanas basado en redes neuronales LSTM, que procese la información obtenida de un sensor de posición 3D (Kinect) y la clasifique entre las distintas clases.

Los objetivos específicos son:

- Determinación de las articulaciones más relevantes para cada acción.
- Crear y entrenar un clasificador basado en una red LSTM que permita discernir a que acción corresponde la secuencia evaluada.
- Crear y entrenar un clasificador basado en las articulaciones más destacadas que permita discernir a que acción corresponde la secuencia evaluada.
- Crear y entrenar un clasificador basado en distinguir entre los grupos de acciones más parecidas hasta discernir a que acción corresponde la secuencia evaluada.

– Estudiar el comportamiento de los clasificadores tanto en eficacia como en tiempo de computo.

1.4. Organización del trabajo

Este trabajo expone tres clasificadores distintos de acciones humanas a partir de la información de los esqueletos obtenida por Kinect.

Antes de diseñar los algoritmos, se realiza un filtrado y procesado de la información para normalizarla. Posteriormente se realiza un estudio de los datos de entrenamiento para seleccionar las características más relevantes para cada acción.

Los algoritmos propuestos se basan en una red neuronal LSTM para lo cual se realiza un estudio de distintos parámetros (relleno de las secuencias, número de neuronas, número de capas, etc) para determinar la mejor configuración.

El trabajo se encuentra dividido en 5 partes:

– Filtrado y normalización de las secuencias de la base de datos.

– Estudio de las características más relevantes de cada acción.

– Estudio de la configuración de red neuronal y desarrollo de los clasificadores.

– Ensayos y resultados.

– Conclusiones.

1.5. Herramientas

El trabajo se ha realizado sobre dos plataformas. El estudio de la base de datos y la selección de las características se ha realizado sobre **Matlab R2018a** debido a que es un entorno muy potente y versátil. Para entrenar las redes neuronales se ha trabajado con **Python 3.7** implementando las redes mediante la biblioteca de redes neuronales **Keras**. La implementación se ha realizado en la plataforma **Google Colaboratory**, la cuál mediante una **máquina virtual online** permite trabajar con CPUs, GPUs y TPUs, y cuenta con las librerías de **Tensoflow** y **Keras** ya instaladas, aunque también presenta limitaciones con respecto al **máximo tiempo de procesamiento seguido (8 horas)**, y la necesidad de estar conectado a internet para poder trabajar, lo cual ha supuesto una limitación en cuánto a la extensión de los cálculos realizables.

Capítulo 2

Base de datos NTU RGB+D 60

En este trabajo se ha utilizado la base de datos de esqueletos humanos NTU RGB+D [4] la cual presenta 56.880 secuencias de vídeo, obtenidas de 40 individuos distintos, con 80 puntos de vista distintos, mediante Microsoft Kinect v2. Esta base de datos contiene 60 clases de acciones diferentes incluyendo acciones diarias, acciones mutuas y acciones relacionadas con la salud (Lista de acciones en el Anexo A). En cada secuencia se ha recolectado el vídeo RGB, la profundidad, la información del esqueleto (ubicación 3D de 25 articulaciones predefinidas), y los fotogramas infrarrojos.

Las edades de los sujetos se encuentran entre los 10 y los 35 años, lo que ayuda a generalizar la información sobre las acciones. Además, las secuencias han sido capturadas con varios fondos distintos para asegurar la inconstancia ambiental.

Los esqueletos que realizan las acciones de las distintas clases están compuestos por 25 articulaciones que representan los puntos fundamentales del movimiento del cuerpo humano según el autor de dicha base. Dichas articulaciones están recogidas en la figura 2.3.

Está base de datos presenta problemas como la presencia de esqueletos incompletos, objetos reconocidos como esqueletos, o secuencias incompletas, para lo cual ha hecho falta aplicar una etapa de filtrado en el preprocesado de las características.

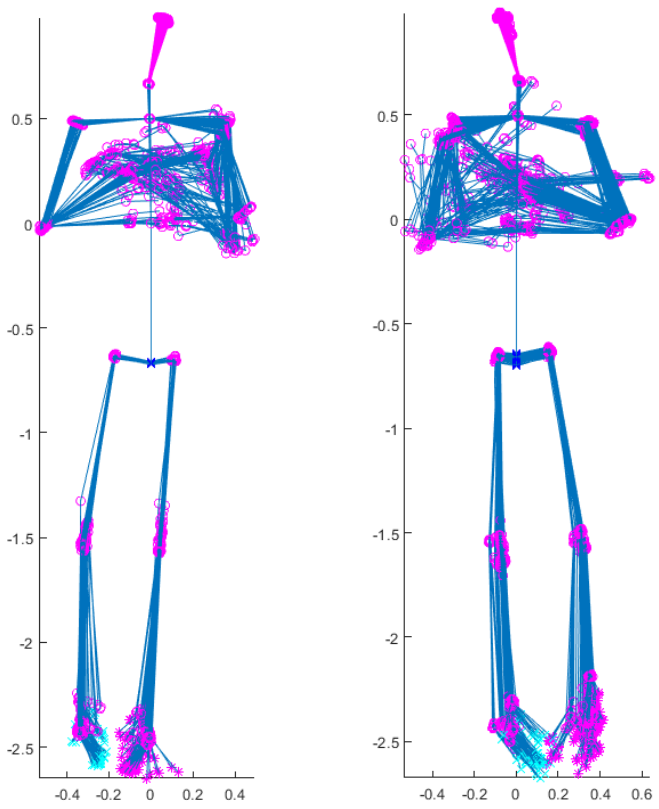
2.1. Preprocesado y filtrado

Primero se han filtrado todos los esqueletos incompletos a partir de la lista "samples_with_missing_skeletons.txt" aportada por [4]. Posteriormente se ha realizado un segundo filtro que descartaba aquellas secuencias donde varía el número de esqueletos a lo largo de la secuencia debido a que consideramos que debía tratarse de ruido.

A diferencia de [4] que trabaja con dos esqueletos rellenando con ceros el esqueleto de aquellas secuencias en las que solo hay un actor, en este trabajo **se ha estudiado**

un solo esqueleto por secuencia para simplificar el problema abordado **reduciendo el espacio de características a la mitad**. Al igual que [4], en las secuencias con más de un actor (desde la clase 50 a la 60, ver Anexo A) se ha considerado como actor principal, es decir el que será analizado por el método de clasificación para asignarle una clase, aquel que presenta un mayor movimiento en 3D. La cantidad de movimiento en 3D de cada actor se ha calculado como la suma de las varianzas de X, Y y Z de cada una de las articulaciones del esqueleto del actor.

En la etapa de preprocesado se ha aplicado una normalización para transformar los puntos de las coordenadas 3D del esqueleto, desde el sistema de coordenadas de la cámara. Para ello primero se trasladan hasta el sistema de coordenadas del cuerpo con origen en el centro de la espina dorsal (articulación 2 en la figura 2.3), posteriormente se realiza un giro para fijar el eje X paralelo con el vector formado desde el hombro derecho al izquierdo (articulaciones 5 y 9 de la figura 2.3), y el eje Y con el vector formado desde la base de la espina hasta el centro de la espina.

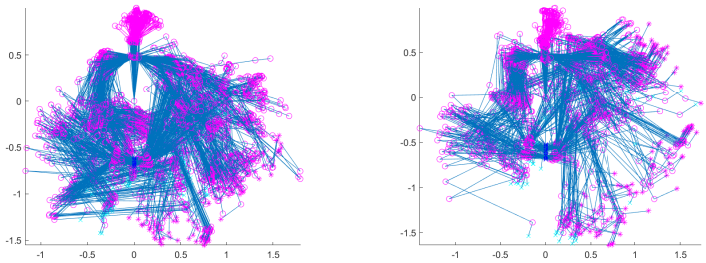


(a) Clase 11 - Leer (b) Clase 12 - Escribir

Figura 2.1: Esqueletos de dos de las clases más parecidas.

La extensión de esta base de datos da lugar a acciones muy similares entre sí que

se confunden con facilidad por los distintos sistemas de clasificación, como por ejemplo las clases 11 y 12, “Leer” y “Escribir” (figuras 2.1a y 2.1b), o a las clases 16 y 17, “Ponerse un zapato” y “Quitarse un zapato” (figuras 2.2a y 2.2b).



(a) Clase 16
- Ponerse un zapato

(b) Clase 17
- Quitarse un zapato

Figura 2.2: Esqueletos de dos de las clases más parecidas.

Por último, cabe destacar que la gran cantidad de información de las articulaciones (25 articulaciones por 3 dimensiones por el número de fotogramas) y la presencia de acciones muy similares entre si (como veremos en el capítulo 4.2) da lugar a un espacio de variables muy complejo donde las redes se sobre ajustan con facilidad. Por ello se va a realizar un análisis y selección sobre las 75 componentes para reducir el espacio de características, y mejorar la clasificación.

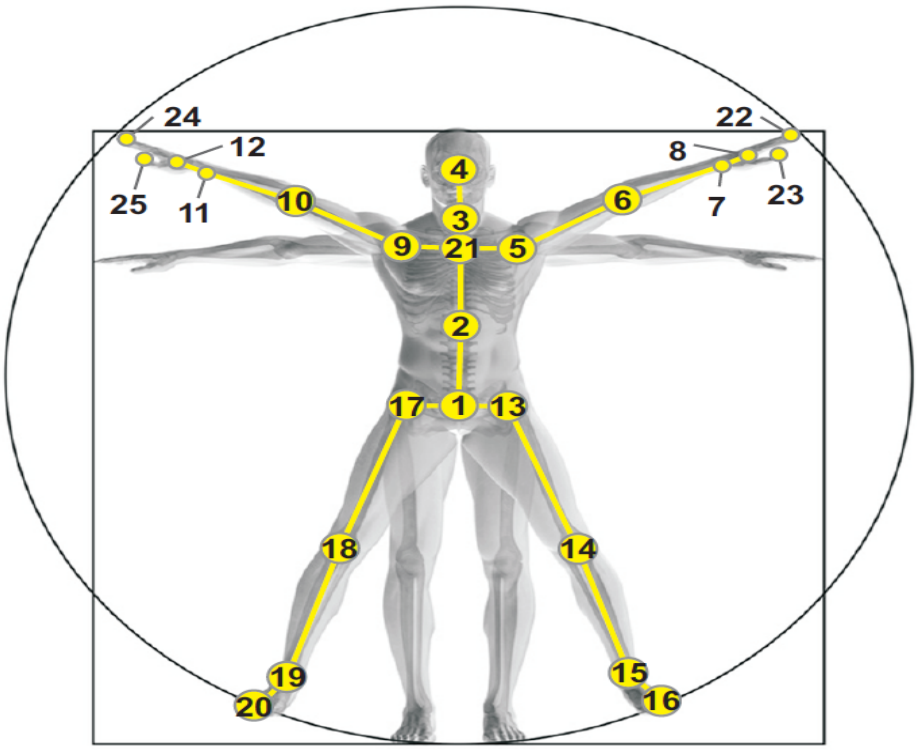


Figura 2.3: Configuración de las 25 articulaciones del cuerpo humano de la base de datos [4]. Las etiquetas de las articulaciones son: 1-base de la espina dorsal 2-mitad de la espina dorsal 3-cuello 4-cabeza 5-hombro izquierdo 6-codo izquierdo 7-muñeca izquierda 8-mano izquierda 9-hombro derecho 10-codo derecho 11-muñeca derecha 12-mano derecha 13-cadera izquierda 14-rodilla izquierda 15-tobillo izquierdo 16-pie izquierdo 17-cadera derecha 18-rodilla derecha 19-tobillo derecho 20-pie derecho 21-extremo superior de la espina dorsal 22-punta de la mano izquierda 23-pulgar izquierdo 24-punta de la mano derecha 25-pulgar derecho

Capítulo 3

Selección Características

La base de datos utilizada en este trabajo presenta las coordenadas X, Y y Z de las 25 articulaciones, siendo un total de 75 características por cada fotograma. Este espacio de características tan grande es muy fácil que sobre ajuste una red neuronal sin la suficiente cantidad de muestras, para ello en este capitulo se va a realizar un análisis y selección de características. **En el primer apartado se van a estudiar las características más relevantes de los esqueletos de cada clase** con el fin de obtener las componentes más características de cada clase y así reducir la complejidad del problema abordado. **En el segundo se estudiarán las características más discriminantes entre distintos grupos de acciones** con el objetivo de mejorar la clasificación de clases similares entre sí.

3.1. Componentes más características

Para obtener una información general sobre las secuencias se realiza un filtrado de aquellas que más difieren del resto. Primero se calcula la distancia euclídea entre los conjuntos consecutivos de las 75 características de cada acción colocando todas las secuencias de una misma clase en fila y calculando la distancia euclídea entre todas las parejas de fotogramas consecutivos para cada secuencia (Figura 3.1). Después se extrae la distancia máxima de cada conjunto de 75 características y se calcula un intervalo de confianza del 90 %, desechando aquellas secuencias que presentan un valor máximo fuera de dicho intervalo (Figura 3.2), obteniendo de esta forma la variación de las 75 componentes para cada secuencia de una clase.

Tras filtrar las secuencias, calculando la variación media de cada componente a lo largo de todas las secuencias de la clase estudiada, se obtiene el histograma de actividad de las 75 características (Figura 3.3a). A partir del cual se seleccionan como características más relevantes aquellas que presenten una variación a lo largo de la secuencia igual o mayor que la mitad de la variación que presenta la característica que

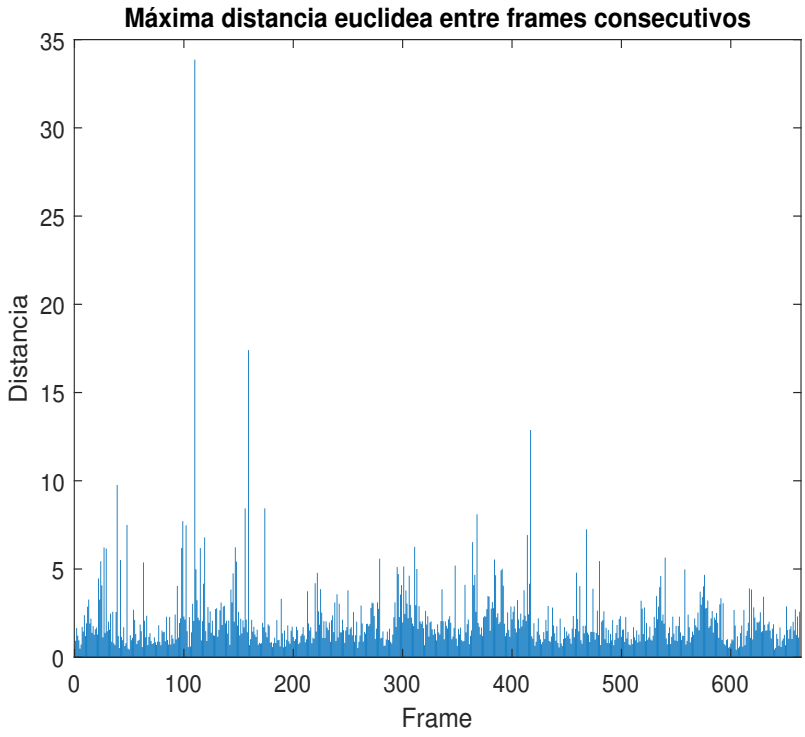


Figura 3.1: Máxima distancia euclídea entre fotogramas consecutivos de la secuencia 11

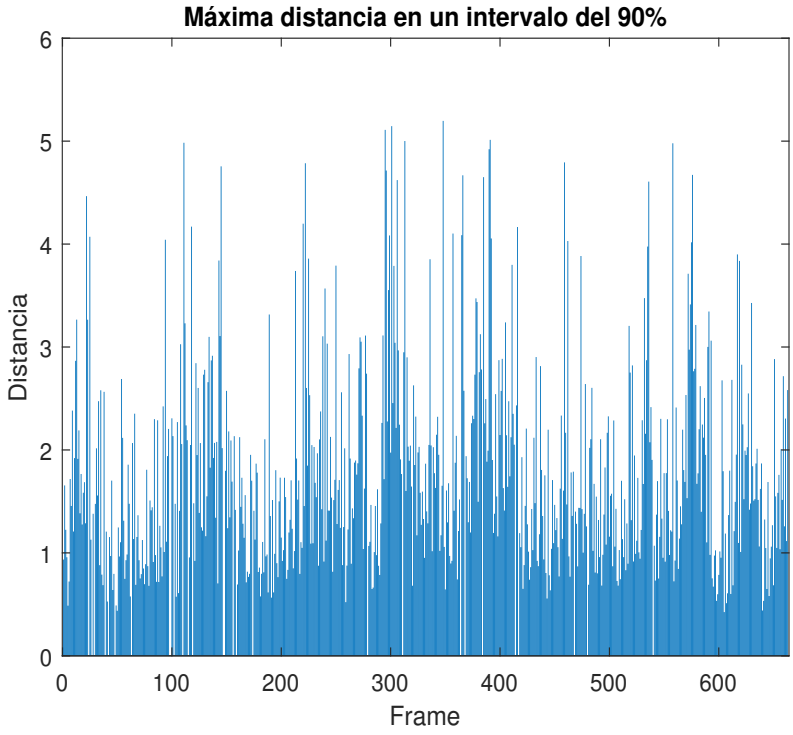
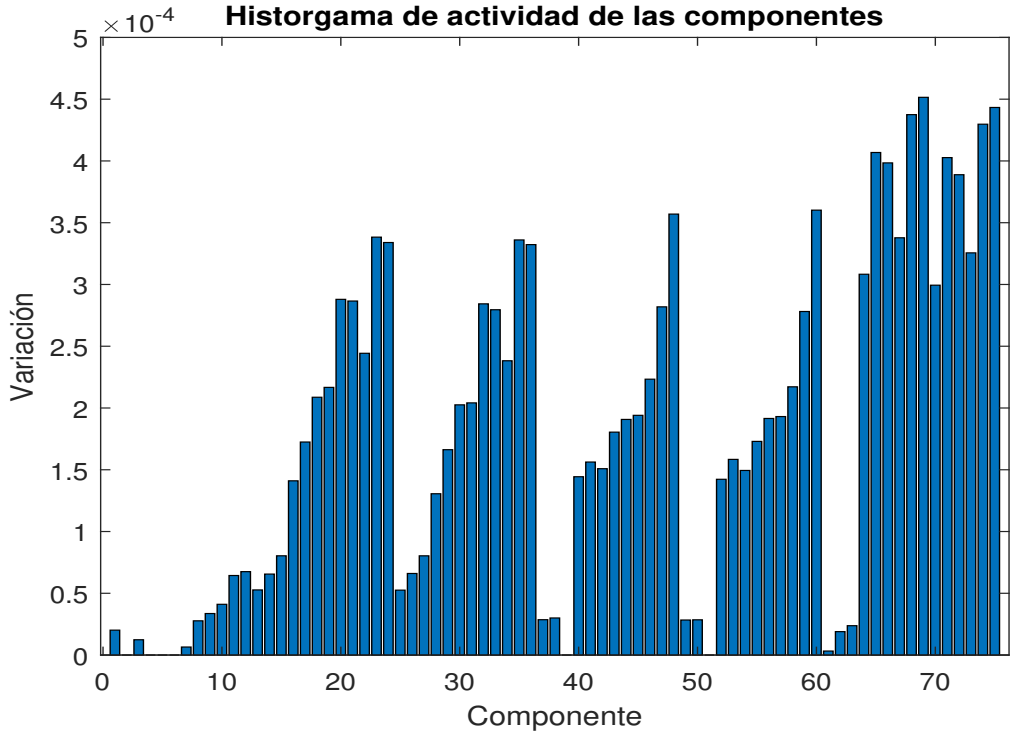
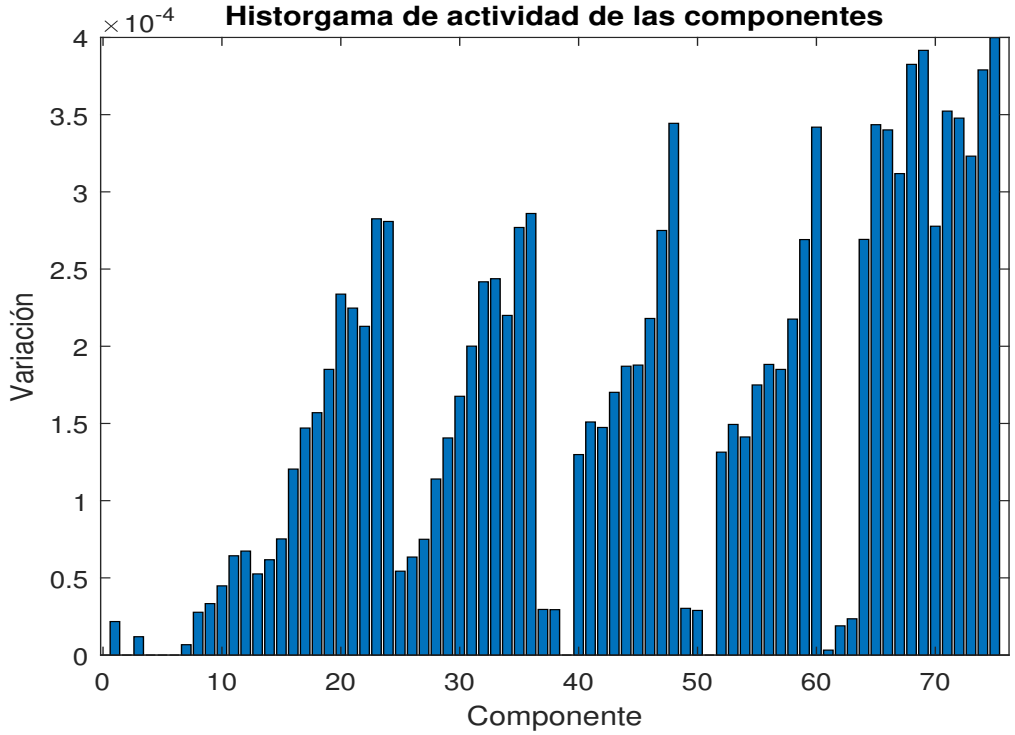


Figura 3.2: Máxima distancia euclídea entre fotogramas consecutivos con un intervalo de confianza del 90 % de la secuencia 11

más varía, ordenándolas en orden decreciente de aportación.



(a) Clase 11 - Leer



(b) Clase 12 - Escribir

Figura 3.3: Histograma de actividad de las 75 componentes.

Comparando los histogramas de actividad de las clases 11 y 12, “Leer” y “Escribir”

(figuras 3.3a y 3.3b), se puede observar como comparten las componentes más características, algo que se debe a que ambas acciones se realizan moviendo las manos de forma similar, como se ha observado anteriormente en la figura 2.1, y que se confirma siendo las componentes más características las equivalentes a las distintas partes de las manos (tablas 3.1 y 3.2).

Componente	Articulación equivalente (Eje)
69	Pulgar izquierdo (Z)
75	Pulgar derecho (Z)
68	Pulgar izquierdo (Y)
74	Pulgar derecho (Y)
65	Punta de la mano izquierda (Y)
71	Punta de la mano derecha (Y)
66	Punta de la mano izquierda (Z)
72	Punta de la mano izquierda (Z)
60	Pie derecho (Z)
48	Pie izquierdo (Z)
23	Mano izquierda (Y)
67	Pulgar izquierdo (X)
35	Mano derecha (Y)
24	Mano izquierda (Z)
36	Mano derecha (Z)
73	Pulgar derecho (X)
64	Punta de la mano izquierda (X)
70	Punta de la mano derecha (X)
20	Muñeca izquierda (Y)
21	Muñeca izquierda (Z)
32	Muñeca derecha (Y)
47	Pie izquierdo (Y)
33	Muñeca derecha (Z)
59	Pie derecho (Y)
22	Mano izquierda (X)
34	Mano derecha (X)

Tabla 3.1: Componentes características de la clase Leer (11)

3.2. Componentes más discriminantes

Ante la presencia de clases que representan acciones parecidas, por lo que sus componentes más relevantes son muy similares, se propone la idea de comparar dichas clases en función de las componentes que más se diferencian entre una clase y la otra. De esta forma se pretende centrar el foco en el grupo de componentes que no comparten, y por lo tanto distingue, acciones muy similares. Para ello se restan los histogramas de las acciones similares entre si, y se seleccionan como componentes más discriminantes

Componente	Articulación equivalente (Eje)
75	Pulgar derecho (Z)
69	Pulgar izquierdo (Z)
68	Pulgar izquierdo (Y)
74	Pulgar derecho (Y)
71	Punta de la mano derecha (Y)
72	Punta de la mano izquierda (Z)
48	Pie izquierdo (Z)
65	Punta de la mano izquierda (Y)
60	Pie derecho (Z)
66	Punta de la mano izquierda (Z)
73	Pulgar derecho (X)
67	Pulgar izquierdo (X)
36	Mano derecha (Z)
23	Mano izquierda (Y)
24	Mano izquierda (Z)
70	Punta de la mano derecha (X)
35	Mano derecha (Y)
47	Pie izquierdo (Y)
64	Punta de la mano izquierda (X)
59	Pie derecho (Y)
33	Muñeca derecha (Z)
32	Muñeca derecha (Y)
20	Muñeca izquierda (Y)
21	Muñeca izquierda (Z)
34	Mano derecha (X)
46	Pie izquierdo (X)
58	Pie derecho (X)
22	Mano izquierda (X)
31	Muñeca derecha (X)

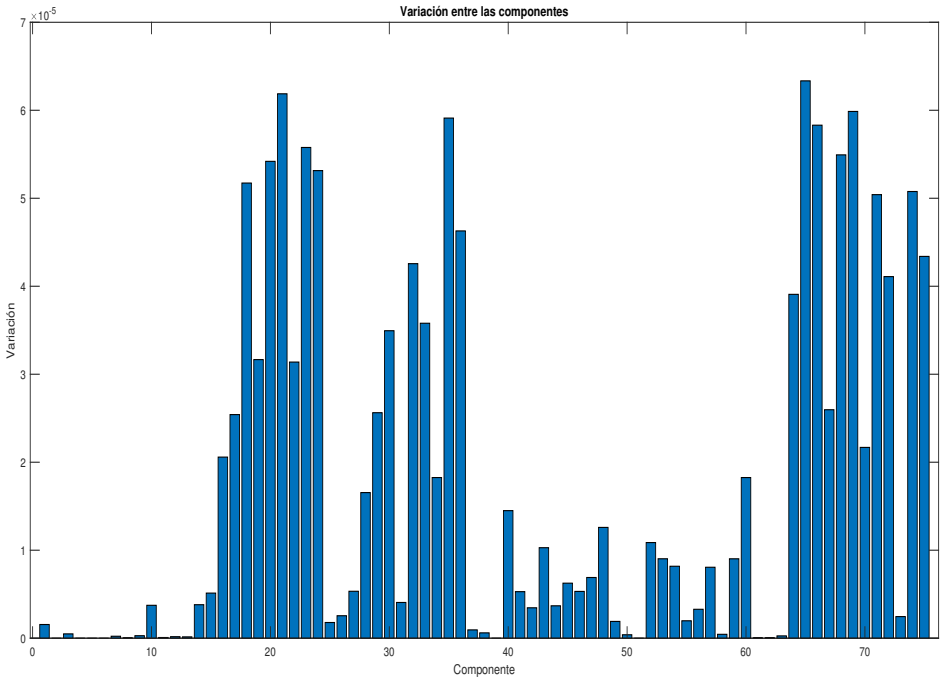
Tabla 3.2: Componentes características de la clase Escribir (12)

aquellas cuyo valor absoluto de la resta, es igual o mayor que la mitad de la diferencia de la característica que más varía, ordenándolas en orden decreciente de aportación.

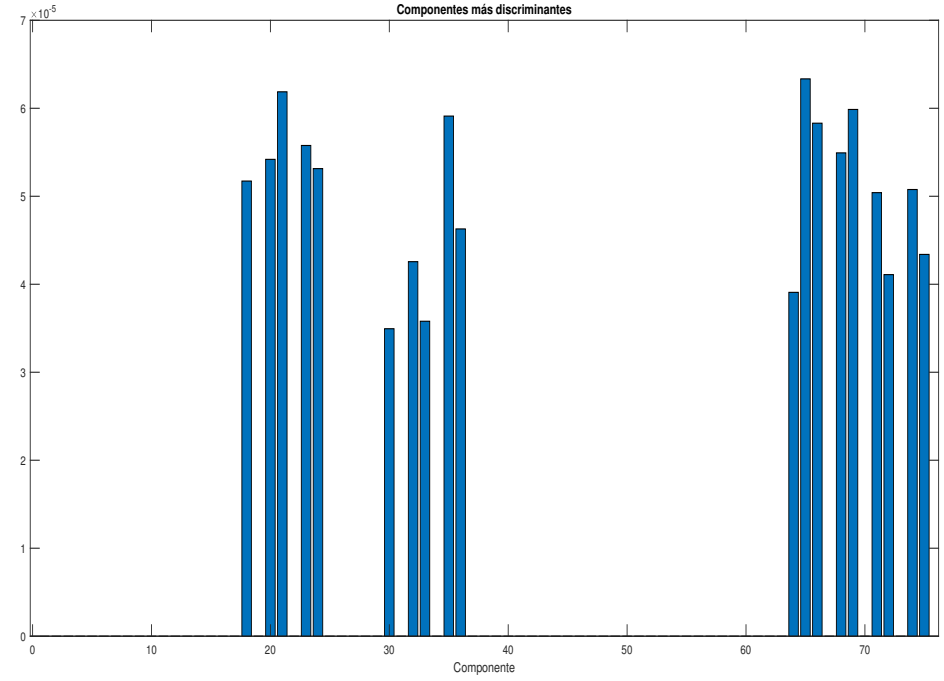
En la figura 3.4a se observa la diferencia entre las componentes de las clases 11 y 12. Seleccionando solo las que presentan una diferencia de como mínimo la mitad de la diferencia de la componentes más distinguida se obtiene la figura 3.4b, donde las componentes más discriminantes, ordenadas de mayor a menor, son las de la tabla 3.3.

Comparando las componentes más discriminantes entre las acciones 11 y 12 (tabla 3.3) con las componentes más características de estas acciones (tablas 3.1 y 3.2) se observa que muchas de la componentes se repiten debido a que aunque el movimiento este centrado en la misma articulación cambia la magnitud con la que varia. También aparecen nuevas componentes como la 18 (Codo izquierdo Z) o la 30 (Codo derecho Z)

las cuales distinguen un movimiento de otro y aportan nueva información con respecto a la que se podía obtener estudiando las componentes más características.



(a) Diferencia entre las componentes de las clases Leer(11) y Escribir(12)



(b) Componentes más discriminantes de las clases Leer(11) y Escribir(12)

Figura 3.4: Histograma de actividad de las 75 componentes.

Componente	Articulación equivalente (Eje)
65	Punta de la mano izquierda (Y)
21	Muñeca izquierda (Z)
69	Pulgar izquierdo (Z)
35	Mano derecha (Y)
66	Punta de la mano izquierda (Z)
23	Mano izquierda (Y)
68	Pulgar izquierdo (Y)
20	Muñeca izquierda (Y)
24	Mano izquierda (Z)
18	Codo izquierdo (Z)
74	Pulgar derecho (Y)
71	Punta de la mano derecha (Y)
36	Mano derecha (Z)
75	Pulgar derecho (Z)
32	Muñeca derecha (Y)
72	Punta de la mano derecha (Z)
64	Punta de la mano izquierda (X)
33	Muñeca derecha (Z)
30	Codo derecho (Z)

Tabla 3.3: Componentes discriminantes entre las clases Leer(11) y Escribir(12)

Capítulo 4

LSTM y variantes de la técnica

A diferencia de los seres humanos que utilizan sus conocimientos previos, o de contexto, sobre un tema para comprender los conocimientos nuevos que adquieren sobre él, las redes neuronales prealimentadas carecen de la capacidad de utilizar la información recibida en el pasado de forma conjunta con la nueva información recibida, siendo ineficientes para interpretar secuencias, sea de palabras o de imágenes. Para solventar este problema se han desarrollado las redes neuronales recurrentes las cuales presentan bucles internos que les permiten utilizar la información pasada de forma conjunta con la información presente. Los métodos desarrollados en este trabajo se basan en la utilización de redes neuronales LSTM, un tipo de red neuronal recurrente cuyas características le permiten recordar información a largo y a corto plazo.

4.1. RNN

Las redes neuronales recurrentes, o RNN [9], son redes especializadas en procesar secuencias de valores. Cada parte de la salida de la red es una función de las partes previas de la salida. Cada parte de la salida se produce usando la misma regla de actualización aplicada a la salida anterior. Esta formulación recursiva da lugar al intercambio de parámetros a través de un grafo computacional profundo que incluye ciclos. Estos ciclos representan la influencia del valor actual de la variable en su valor en en el siguiente paso temporal.

Estas redes operan sobre una secuencia que contiene vectores x_t, con el índice de paso temporal t que abarca de 1 hasta τ, siendo τ la longitud de la secuencia. El índice temporal no tiene porque ser una referencia literal al paso del tiempo en el mundo real, sino que puede hacer referencia a la posición en la secuencia.

Este tipo de redes neuronales presenta un problema con las dependencias temporales a largo plazo, puesto que los gradientes propagados a la largo de varias etapas tienden a desvanecer o a crecer desmesuradamente.

Una breve explicación de su funcionamiento se encuentra recogida en el anexo B.1.

4.2. LSTM

Las redes neuronales LSTM, o Long-Short Term Memory [10], introducen auto bucles que generan caminos por los cuales el gradiente puede propagarse durante largos periodos sin desvanecer. Esto se consigue introduciendo una unidad de puerta de entrada multiplicativa para proteger los contenidos de la memoria almacenados en la célula de perturbaciones producidas por entradas irrelevantes, y una unidad de puerta de salida multiplicativa para proteger otras células de los contenidos de la memoria de la célula actualmente irrelevantes. El resultado es una unidad más compleja llamada célula de memoria (Figura 4.1), y cuyo funcionamiento se modela en el anexo B.2.

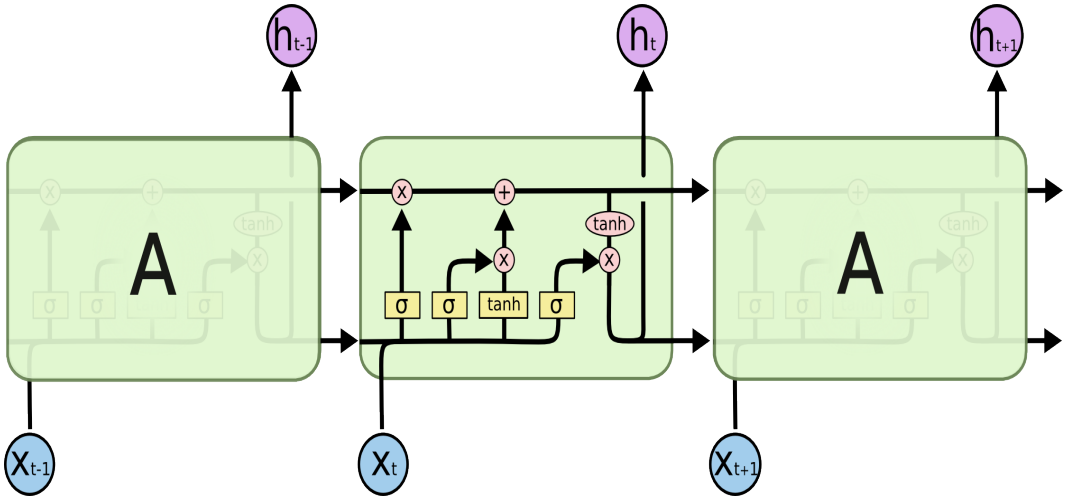


Figura 4.1: Célula LSTM

4.2.1. LSTM Bidireccional

Una variedad de la red LSTM es la LSTM-Bidireccional en la cuál la información se propaga tanto desde el principio de la red hacia el final, como desde el final hasta el principio, obteniendo así una bidireccionalidad en la transmisión de la información al coste de duplicar los pesos de la red.

4.3. Método de las componentes características

El primer método desarrollado es el método de las componentes características. Este método se basa en crear una red neuronal especializada en las componentes más características de cada una de las clases a distinguir, creando de esta forma un clasificador con tantas redes neuronales como clases a distinguir, todas las redes con la

misma configuración, pero con distintos vectores de entrada y datos de entrenamiento. Cada una de estas redes recibe como entradas las características más destacadas de la acción sobre la que se especializan (obtenidas en la sección sección 3.1), y proporciona como salida un único valor que representa la probabilidad de que la secuencia evaluada pertenezca a la clase sobre la que se ha especializado dicha red o no.

Para clasificar una secuencia, el comparador la evaluará en todas las redes neuronales y asignará como predicción la acción que corresponda a la red cuyo valor de salida sea más alto.

4.4. Método del árbol

El segundo método desarrollado es el método del árbol de decisiones. Este método se basa en dividir las clases en grupos según la similitud de las clases, y discernir si la secuencia a clasificar pertenece a un grupo u a otro. Esta comparación se repite siguiendo un dendograma que clasifica los distintos grupos, hasta llegar a clasificar la secuencia en un grupo formado por una sola clase.

Para definir el árbol de decisiones se ha comparado todas las secuencias de la base de datos de entrenamiento (actores del grupo B de la tabla 5.1) 1 contra 1 en una red LSTM Bidireccional especializada en esas dos secuencias. A partir de los resultados de dicha comparación se ha construido una matriz de confusión para conocer que secuencias se confunden más entre si, y se ha extraído el valor del coeficiente kappa de Cohen, una medida estadística que ajusta el efecto del azar en función de la concordancia observada, restando el efecto de los aciertos aleatorios. Este índice sirve para dar una idea de la calidad de los clasificadores, siendo buenos clasificadores aquellos con una K > 0.7, y clasificadores aleatorios aquellos con una $K \approx 0$ (ecuaciones 4.1 y 4.2).

$$DeteccionesPonderadas = \frac{MuestrasCorrectas * MuestrasDetectadas}{MuestrasTotales} \tag{4.1}$$

$$K = \frac{DeteccionesCorrectas - DeteccionesPonderadas}{MuestrasTotales - DeteccionesPonderadas} \tag{4.2}$$

A partir de la matriz de kappas se ha construido un dendograma donde se pueden observar los distintos grupos ordenados por la calidad con la que un clasificador binario podría discernir a que grupo pertenece una determinada secuencia. Cuanto más fácil es distinguir dos grupos más arriba se encuentran en el dendograma (Figura 4.2).

Una vez obtenido el dendograma se ha entrenado una red neuronal para cada una de las bifurcaciones, estando la red especializada en distinguir entre los dos grupos que

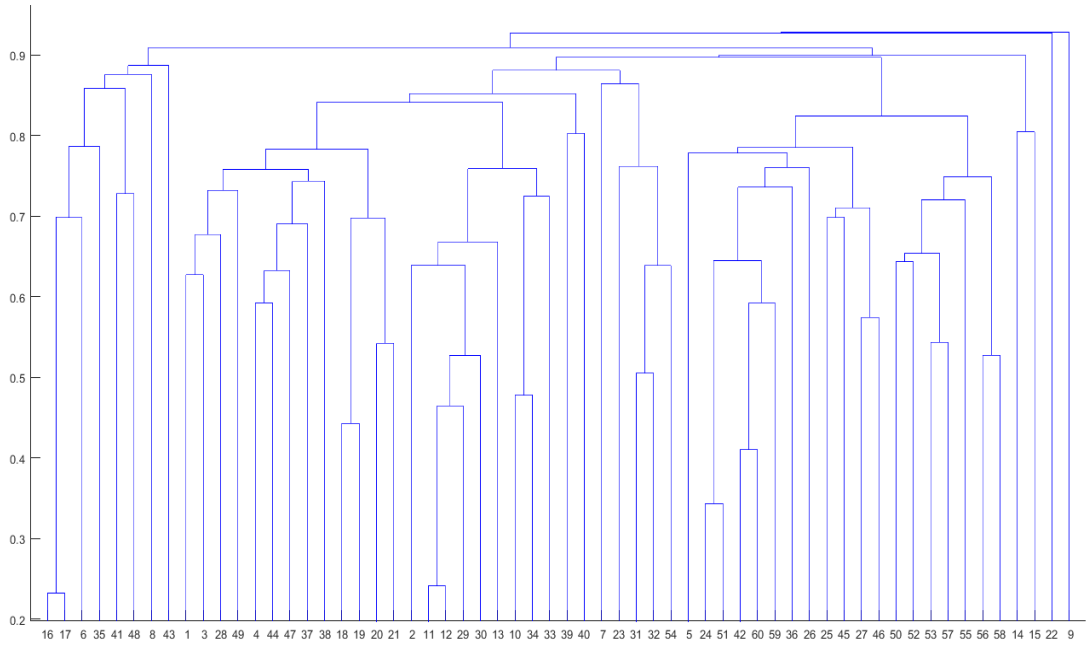


Figura 4.2: Dendograma 60 clases

forman la bifurcación, y presentando como salida la probabilidad de que la secuencia evaluada pertenezca a un grupo u al otro.

Siguiendo el dendograma, recogido en el anexo C.1, en cada bifurcación el clasificador evaluara la secuencia a clasificar, en la red correspondiente a dicha bifurcación. Si el resultado de la evaluación es un valor inferior a 0.5 se considerara que la secuencia es más parecida al grupo de clases de la izquierda y el clasificador procederá por dicha rama del dendograma. Si el resultado de la evaluación es un valor igual o mayor que 0.5 se considerará que la secuencia es más parecida al grupo de clases de la derecha de la bifurcación procediendo en este caso por la rama derecha del dendograma. Esta comparación se realizara en cada bifurcación hasta que el grupo de clases de uno de los lados este formado por una sola clase la cual será considerada la predicción del clasificador para la secuencia evaluada.

En dicho esquema se observa como la clase que tiene que pasar por menos redes para ser clasificada es la clase 9 que solo pasa por una red, mientras que las clases que pasan por más redes son las clases 4, 11, 12 y 44 las cuales pasan por 14 redes para ser clasificadas.

Cuantas más redes tengan que evaluar a una secuencia antes de que sea clasificada, peor probabilidad de acierto habrá debido a que el error de la red de cada nivel se acumula de forma multiplicativa.

4.4.1. Árbol + Componentes discriminantes

Para intentar mejorar el rendimiento del método del árbol se ha desarrollado una variante que se basa en entrenar los modelos de las bifurcaciones a partir de las componentes más discriminantes (sección 3.2) entre los dos grupos de acciones de cada modelo. Por simplicidad este nuevo modelo se ha desarrollado únicamente para las acciones con un solo actor (clases 1 a 49), pensando en expandirlo para el resto de clases en función de los resultados obtenidos.

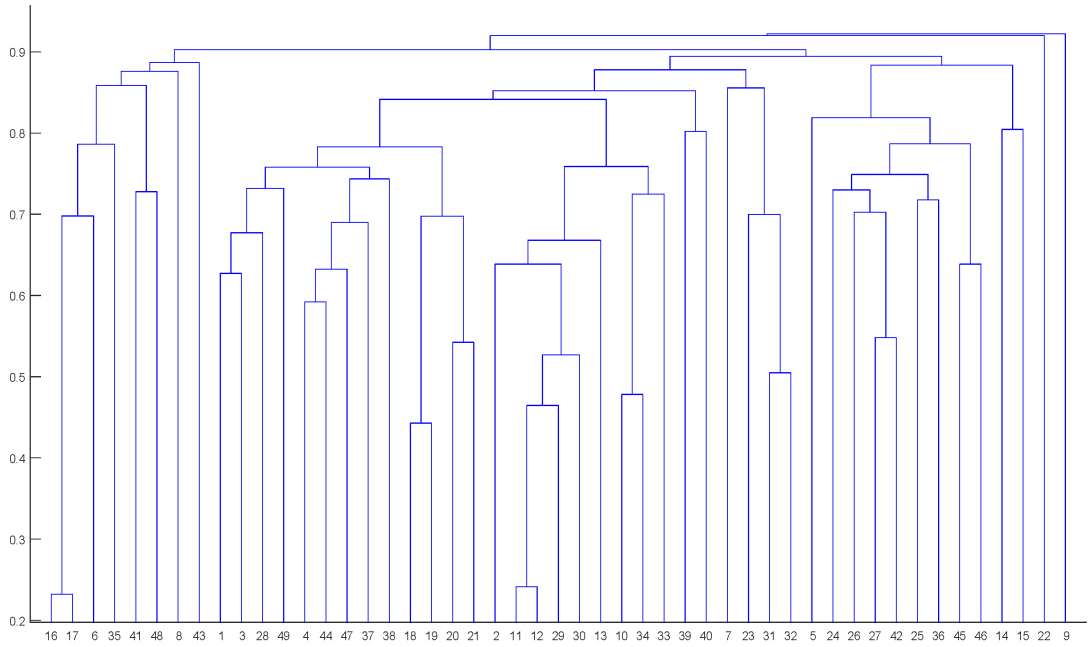


Figura 4.3: Dendograma clases 1 a 49

En el anexo C.2 se puede observar un esquema del árbol de decisiones con los distintos niveles. En dicho esquema se observa como la clase que tiene que pasar por menos redes para ser clasificada es la clase 9, mientras que las clases que pasan por más redes son las clases 4, 11, 12 y 44 las cuales pasan por 13 redes para ser clasificadas, un nivel menos que con respecto al método del árbol y que se debe a la ausencia de las clases desde las 50 hasta la 60.

Capítulo 5

Ensayos y resultados

Este apartado se divide en tres secciones. La primera estudia las distintas opciones de normalizar todas las secuencias a una misma longitud. La segunda evalúa distintas arquitecturas de red neuronal. Y en la tercera se estudia el comportamiento de los clasificadores.

Para el entrenamiento de las redes neuronales se han utilizado las secuencias representadas por los actores del grupo A, mientras que para la evaluación se han utilizado las secuencias representadas por los actores del grupo B (tabla 5.1).

Grupo	Actores
A	1,2,4,5,8,9,13,14,15,16,17,18,19,25,27,28,31,34,35,38
B	3,6,7,10,11,12,20,21,22,23,24,26,29,30,32,33,36,37,39,40

Tabla 5.1: Grupos de actores

En cada sección se presentarán los ensayos realizados, los resultados obtenidos y se discutirán los mismos. Todos los ensayos se han realizado entrenando las redes hasta que la función de perdida (Loss function) dejaba de mejorar durante diez épocas seguidas, y guardando el modelo del entrenamiento que había obtenido mejores resultados.

Debido al alto coste computacional que implica trabajar con toda la base de datos, los estudios de las secciones 5.1 y 5.2 se realizan únicamente sobre dos de las 60 clases.

5.1. Normalización de la longitud de las secuencias

Una de las limitaciones que presentan las redes LSTM es que todas las secuencias tienen que tener la misma longitud. Para ello se estudian seis métodos distintos de normalizar las secuencias a la misma longitud:

- **Zero Padding:** Es el método más utilizado en la literatura. Consiste en definir una longitud mayor o igual que la de la secuencia más larga, y rellenar todas las secuencias con ceros hasta que alcanzan la longitud definida. Existen dos

variaciones: rellenando las secuencias con ceros al principio (Zero Padding Top), o rellenándolas con ceros al final (Zero Padding Bottom).

- **Frame padding:** Al igual que en el Zero Padding, en este método se define una longitud mayor o igual que la de la secuencia más larga. La diferencia radica en que en este método en vez de rellenar las secuencias con ceros, se rellenan con un fotograma, que será el primero en el caso de rellenar el principio de la secuencia (Frame padding Top), o el último en el caso de rellenar el final de la secuencia (Frame padding bottom).

- **20 Timesteps:** Es el método utilizado por [4] y [6]. Se basa en dividir las secuencias en $T = 20$ pasos temporales de igual longitud, y escoger aleatoriamente un fotograma de cada intervalo. Esta estrategia añade variación a la base de datos y mejora la capacidad de generalizar de la red neuronal.

- **Longitud media:** Por último desarrollamos un método propio que se basa en calcular la longitud media de todas las secuencias (N) y ajustar todas las secuencias a dicha longitud. Las secuencias de mayor longitud se dividen en N pasos temporales y se escoge un fotograma de cada paso. Las secuencias más cortas se alargan repitiendo algunos fotogramas.

Para analizar el rendimiento de los distintos métodos propuestos se va a entrenar una red para que distinga entre dos de las acciones más parecidas de la base de datos, leer y escribir (números 11 y 12), y se va a comparar su porcentaje de acierto.

La red entrenada consta de una única capa oculta: una capa LSTM con 100 neuronas y un coeficiente de dropout de 0.5.

Debido a que el resultado del entrenamiento varía por la aleatoriedad de los pesos iniciales, en cada entrenamiento se realizara primero 30 entrenamientos para cada método, se medirá la media y la variación estándar de la exactitud, se realizara un estudio de la t de student para conocer el numero de muestra necesarias para conseguir un error absoluto de 0.5 con una confianza del 95 %, y se realizaran los entrenamientos extra para tener el número suficiente de muestras.

Tras un primer análisis de los resultados obtenidos (Tabla 5.2) se observan principalmente dos cosas: el método de ZPB no es valido, pues proporciona un acierto del 49.8 %, el cuál es peor que tirar una moneda al aire, pues en la clasificación de dos secuencias la aleatoriedad corresponde a un 50 % de acierto; y que los métodos que mejores resultados proporcionan son FPT y TS, 64.87 % y 64.88 % respectivamente, aunque con muy poca diferencia con los otros métodos teniendo en cuenta el margen de error.

Método (repeticiones)	Acc (%)	Desv. Estándar	Margen de error
ZPT (30)	64.422	1.257	0.45
ZPB (30)	49.803	0.160	0.06
FPT (42)	64.87	1.590	0.48
FPB (47)	62.943	1.566	0.45
TS (48)	64.875	1.677	0.47
AVG (38)	63.037	1.425	0.45

Tabla 5.2: Análisis relleno secuencias 11 y 12

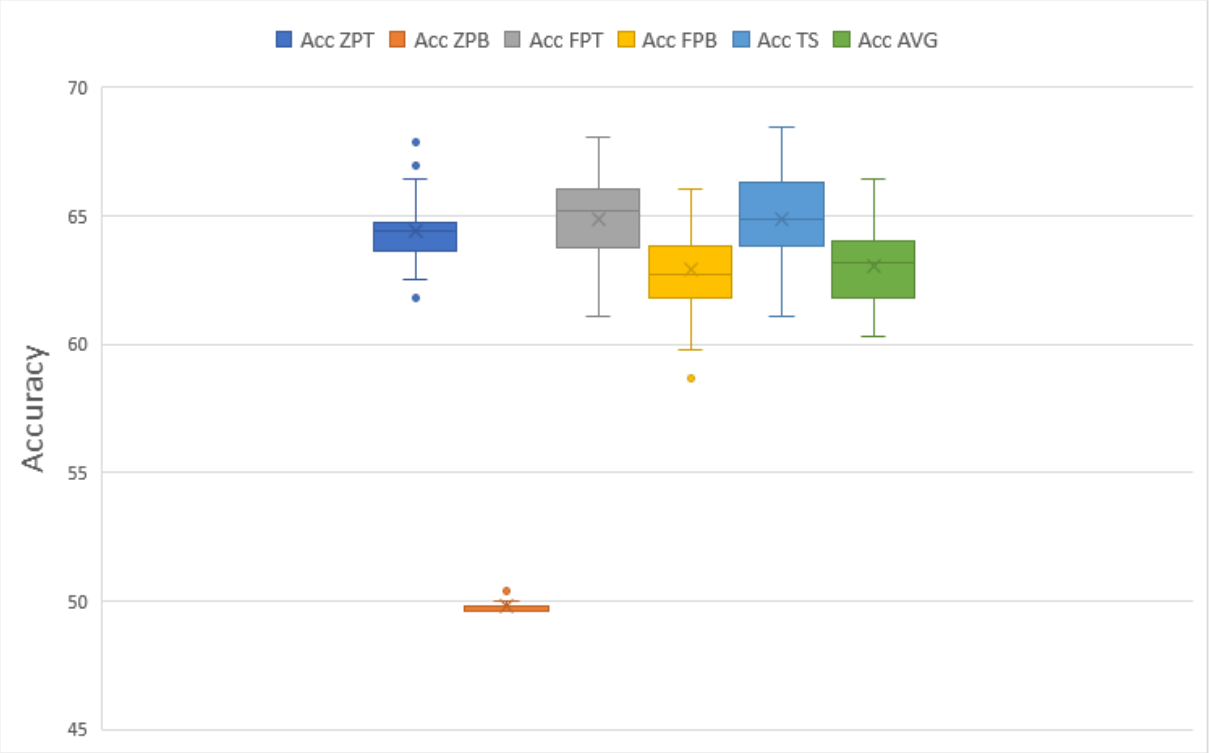


Figura 5.1: Acierto de los distintos rellenos para las secuencias 11 y 12

Para contrastar los resultados obtenidos se repite el experimento realizarlo sobre las dos acciones con mayor diferencia en la longitud media de las secuencias:

– Ponerse una chaqueta (secuencia 14)

– Cruzar las manos de frente (secuencia 40)

Finalmente tras contrastar los resultados (Tabla 5.3) se observa que el mejor resultado es el obtenido por el método de los pasos temporales (TS), con un 96.19 % de acierto. Por lo tanto, este será el método utilizado para normalizar la longitud de las secuencias en los siguientes apartados.

Método (repeticiones)	Acc (%)	Desv. Estándar	Margen de error
ZPT(30)	95.150	1.188	0.43
ZPB(10)	48.975	0.485	0.35
FPT(30)	94.495	1.140	0.41
FPB(59)	90.142	2.455	0.64
TS(30)	96.190	0.561	0.2
AVG(30)	95.229	0.678	0.42

Tabla 5.3: Análisis relleno secuencias 14 y 40

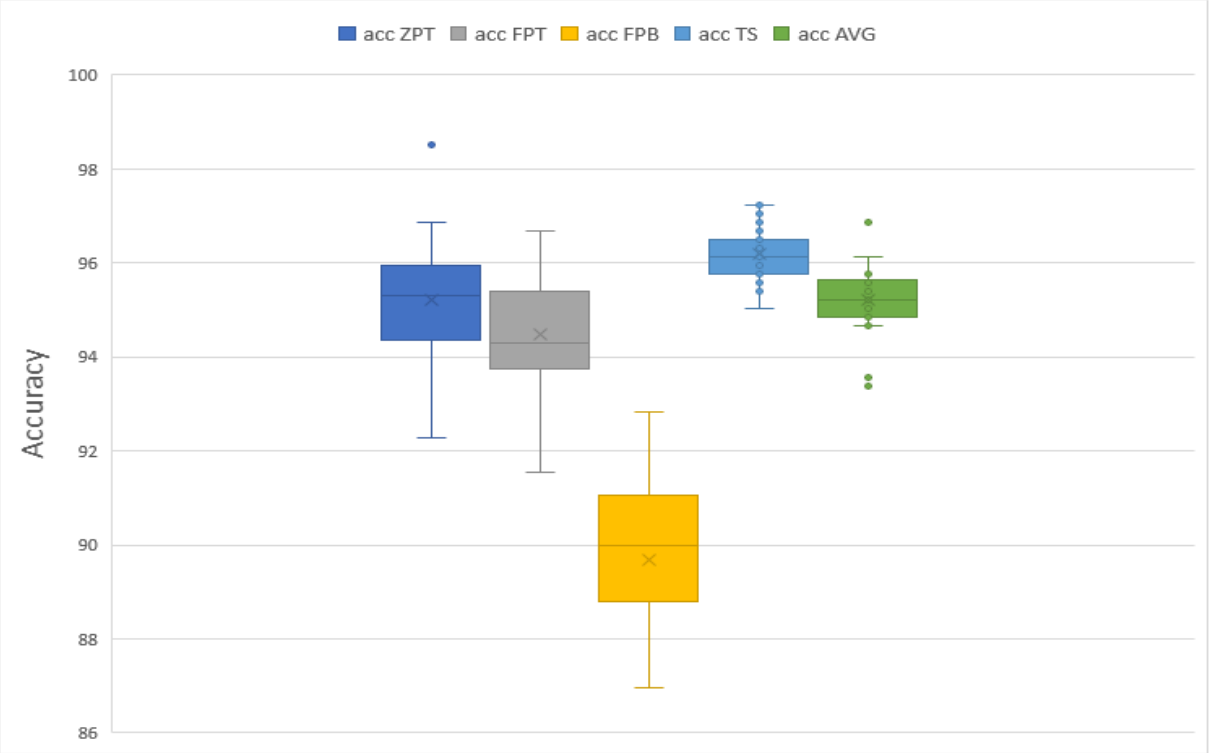


Figura 5.2: Acierto del análisis del relleno para las secuencias 14 y 40

5.2. Estudio arquitectura

En este apartado se evalúan diversas arquitecturas de red neuronal, con el objetivo de escoger la óptima sobre la cual implementar los métodos propuestos. El estudio se centra en dos arquitecturas, (**LSTM y LSTM Bidireccional**), variando la cantidad de neuronas(**10, 50 , 100 y 200**).

El estudio se realiza sobre dos de las secuencias más parecidas de la base de datos, leer y escribir (números 11 y 12), y se evalúa comparando los porcentajes de acierto obtenidos.

Tras un primer estudio se observa que los mejores resultados se obtienen con 100 y 200 neuronas para ambas arquitecturas, obteniendo la arquitectura LSTM Bidireccional resultados ligeramente mejores (Tabla 5.4). Al igual que en el apartado anterior se va

Método (repeticiones)	Acc (%)	Desv. Estándar	Margen de error
LSTM 10 (46)	64.875	1.677	0.498
LSTM 50 (40)	65.655	1.482	0.474
LSTM 100 (40)	66.697	1.221	0.390
LSTM 200 (27)	66.838	1.168	0.462
BiLSTM 10 (81)	64.024	2.126	0.470
BiLSTM 50 (45)	66.806	1.443	0.434
BiLSTM 100 (25)	67.018	1.064	0.581
BiLSTM 200 (16)	67.216	0.896	0.477

Tabla 5.4: Análisis arquitectura secuencias 11 y 12

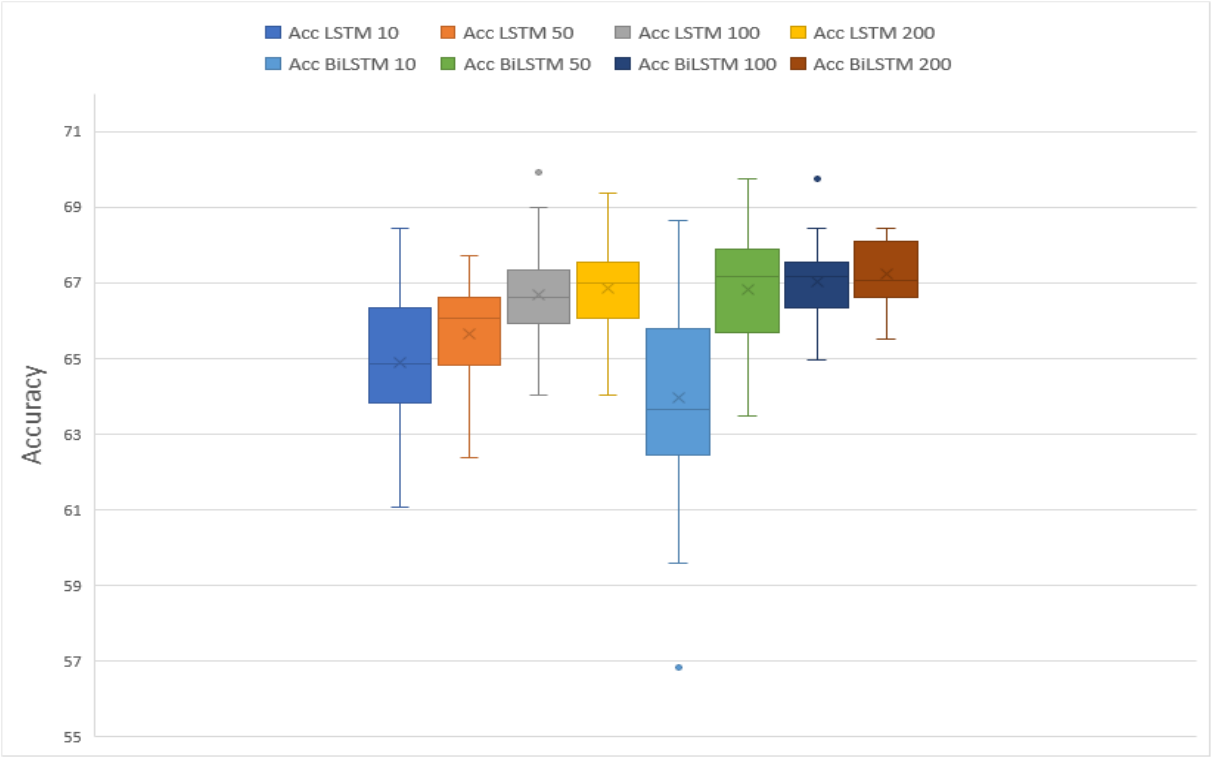


Figura 5.3: Acierto del análisis de la arquitectura para las secuencias 11 y 12

a contrastar los resultados obtenidos repitiendo el estudio para las secuencias 14 y 40.

Método (repeticiones)	Acc (%)	Desv. Estándar	Margen de error
LSTM 10 (30)	96.190	0.561	0.209
LSTM 50 (20)	96.624	0.401	0.188
LSTM 100 (20)	96.719	0.381	0.178
LSTM 200 (20)	96.972	0.406	0.190
BiLSTM 10 (20)	96.688	0.574	0.269
BiLSTM 50 (20)	96.789	0.355	0.166
BiLSTM 100 (20)	96.954	0.312	0.146
BiLSTM 200 (20)	97.046	0.385	0.180

Tabla 5.5: Análisis arquitectura secuencias 14 y 40

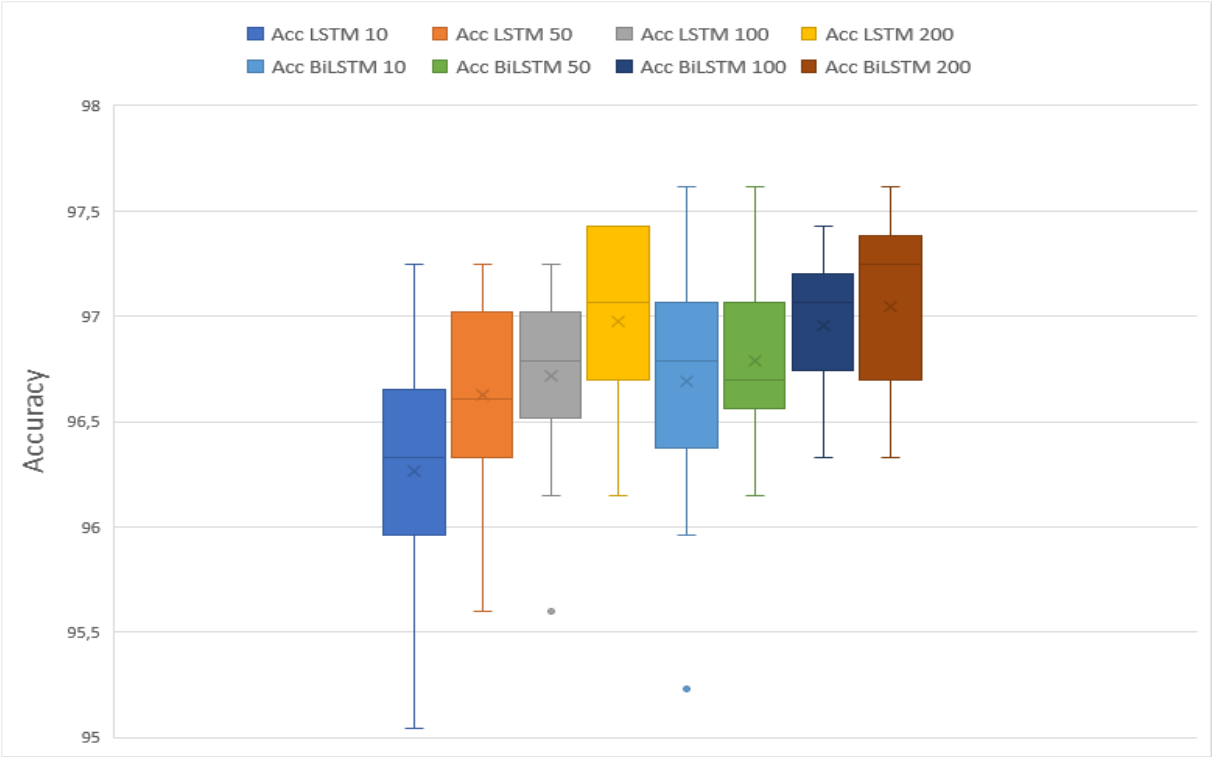


Figura 5.4: Accuracy análisis arquitectura secuencias 14 y 40

Tras repetir el análisis se observa como los resultados obtenidos (Tabla 5.5) coinciden con los anteriores obteniendo los mejores resultados las configuraciones de 100 y 200 neuronas. Aunque en ambos análisis se observa el máximo rendimiento para el mayor número de neuronas, no se ha probado a aumentar el número de neuronas por encima de 200 debido a que el coste computacional es demasiado alto para el escaso beneficio que aportaría.

Debido a la escasa diferencia en el rendimiento entre las configuraciones con mejores resultados, se va a optar por utilizar la arquitectura LSTM Bidireccional con 100 neuronas pues es la que presenta una mejor relación rendimiento-coste computacional, pues al tener menos pesos que las configuraciones con 200 neuronas su tiempo de

entrenamiento será menor debido a que en cada paso de una época de entrenamiento se tendrán que actualizar un menor número de pesos, lo que implica un menor coste computacional.

La tabla 5.6 presenta una comparación entre las 4 configuraciones con mejor acierto, donde se puede observar que al aumentar el número de neuronas aumenta el numero de pesos. Esto se debe a la presencia de una mayor cantidad de conexiones entre las neuronas de las distintas capas producida por la mayor cantidad de neuronas. A su vez también se observa que las redes LSTM Bidirecionales presentan el doble de pesos que la equivalente LSTM con el mismo número de neuronas, lo cual es resultado de que una red Bidireccional está formada por dos redes LSTM cada una de las cuales analiza el flujo de información en un sentido.

Método	Acc (%)	Pesos
LSTM 100 (20)	96.719	70800
LSTM 200 (20)	96.972	221600
BiLSTM 100 (20)	96.954	141600
BiLSTM 200 (20)	97.046	442320

Tabla 5.6: Análisis de los pesos de las distintas arquitecturas

5.3. Clasificadores

En este apartado se estudian los tres clasificadores propuestos. Primero se estudia un clasificador formado por una red neuronal LSTM Bidireccional, posteriormente el clasificador basado en el método de las componentes más destacadas, y por último el clasificador basado en el método del dendograma.

Todos los clasificadores han sido construidos utilizando una o más redes neuronales con una capa oculta de arquitectura LSTM Bidireccional con 100 neuronas, y una capa de salida densa de sesenta neuronas en el caso del primer clasificador, y de una neurona en los otros dos, con activación Sigmoide para clasificar entre dos grupos (Si la salida es mayor que 0.5 pertenece al primer grupo, sino pertenece al segundo grupo).

Las redes han sido entrenadas con las secuencias pertenecientes a los actores del grupo B (tabla 5.1) y validadas con las secuencias de los actores pertenecientes al grupo 2 (Tabla 5.1). El entrenamiento se ha realizado durante un máximo de 100 épocas, parando cuando la función de pérdida dejaba de mejorar durante 10 épocas seguidas y guardando el modelo con mejor prestación durante el entrenamiento.

5.3.1. LSTM Bidireccional

El primer clasificador desarrollado es una red de una sola capa oculta que clasifica la secuencia que recibe entre las distintas clases. Se basa en una capa LSTM Bidireccional de 100 neuronas, con dropout de 0.5, seguido de una capa densa de salida de 60 neuronas (una por clase) y activación softmax. Se utiliza esta función de activación debido a que al ser la capa de salida una distribución de la probabilidad de que la secuencia pertenezca a una de las 60 clases, el conjunto de los 60 valores de salida tiene que sumar uno.

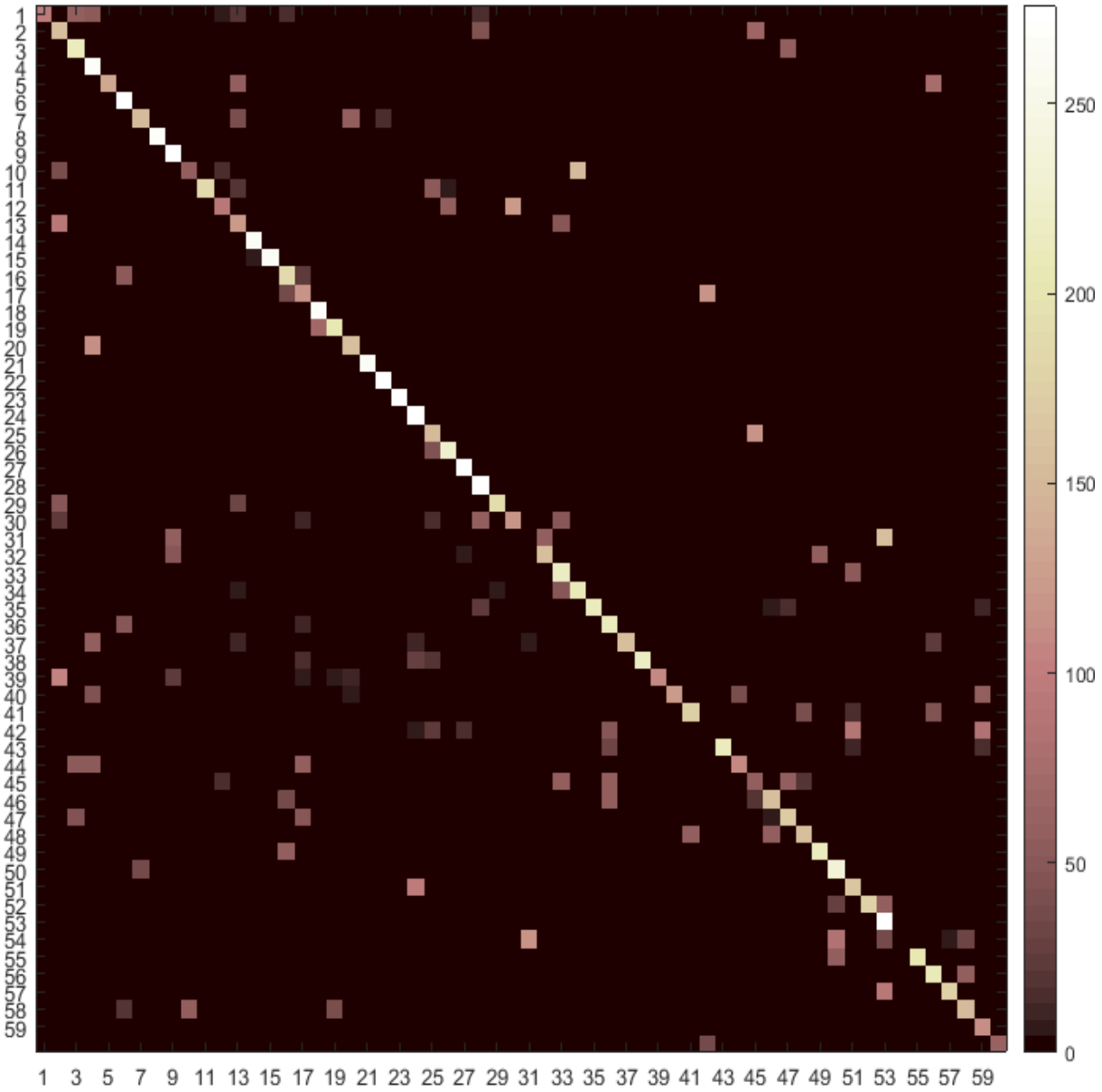


Figura 5.5: Matriz de confusión del clasificador de una red Bi-LSTM

Entrenando la red sobre los actores del grupo B (tabla 5.1) durante un máximo de 100 épocas, parando cuando la función de pérdida dejaba de mejorar durante 10

épocas seguidas y guardando el modelo con mejor prestación durante el entrenamiento, y evaluando su rendimiento sobre los actores del grupo 2, se ha obtenido un porcentaje de acierto del 66.90 %. Como se puede observar en la matriz de confusión (figura 5.5) los peores porcentajes de acierto se corresponden a las clases 31 ("Apuntar algo con un dedo"), 42 ("Asombrarse") y 54 ("Señalar con el dedo a otra persona"), con unos porcentajes de 0.0 %, 0.72 % y 0.0 % respectivamente, y la desviación estándar del acierto de las distintas clases de un 26.50 %.

En el caso de la clase 31, se produce una gran confusión con la clase 53 ("Dar una palmada en la espalda a otra persona"), mientras que la clase 54 se confunde principalmente con la clase 31. Esta confusión se debe principalmente a que al estudiar la red un solo esqueleto no diferencia entre cuando una acción se realiza sobre un objeto (clase 31) o cuando se realiza sobre otra persona (clases 53 y 54) dando lugar a confusiones entre acciones similares. A su vez la clase 42 presenta una baja tasa de acierto debido a que es una acción que cada persona realiza de una forma distinta dando lugar a que se confunda con acciones sin mucho parecido como la clase 36 ("Sacudir la cabeza"), la clase 51 ("Dar una patada a otra persona") o la clase 59 ("Dos personas caminando una hacia otra").

En cuanto al coste temporal, como se puede observar en la tabla 5.9, la red se ha entrenado en un periodo de 5 horas, y cada secuencia ha tardado de media 0.35 segundos en ser evaluada.

5.3.2. Método de las componentes características

El método de las componentes destacadas se basa en una red neuronal especializada para cada clase, las cuales reciben como entrada las componentes más destacadas de dicha clase y asignan una probabilidad de que la secuencia evaluada pertenezca o no a dicha clase. En este apartado primero vamos a estudiar el número ideal de componentes que reciben las redes, y después se evaluará el rendimiento del método.

Vamos a estudiar si las redes obtienen mejor rendimiento con la componente más relevante, las 10 componentes más relevantes, todas las componentes relevantes, o todas las 75 componentes. Para ello se realiza un estudio sobre dos de las secuencias más parecidas de la base de datos, leer y escribir (números 11 y 12), sobre una red de una capa LSTM Bidireccional de 100 neuronas,y se evalúa comparando los porcentajes de acierto obtenidos.

Tras el primer análisis (Tabla 5.7) se observa como utilizando solo la componente más importante la red obtiene las peores prestaciones debido a la poca información recibida. También se observa como las otras tres configuraciones obtienen resultados muy similares, siendo ligeramente mejor las configuraciones con todas las componentes

Componentes	Acc (%)	Desv. Estándar	Margen de error
Componente más relevante	54.268	0.564	0.10
10 componentes más relevantes	66.685	1.144	0.21
Todas las componentes más relevantes	66.925	1.074	0.20
Todas las 75 componentes	66.956	1.144	0.21

Tabla 5.7: Análisis componentes secuencias 11 y 12

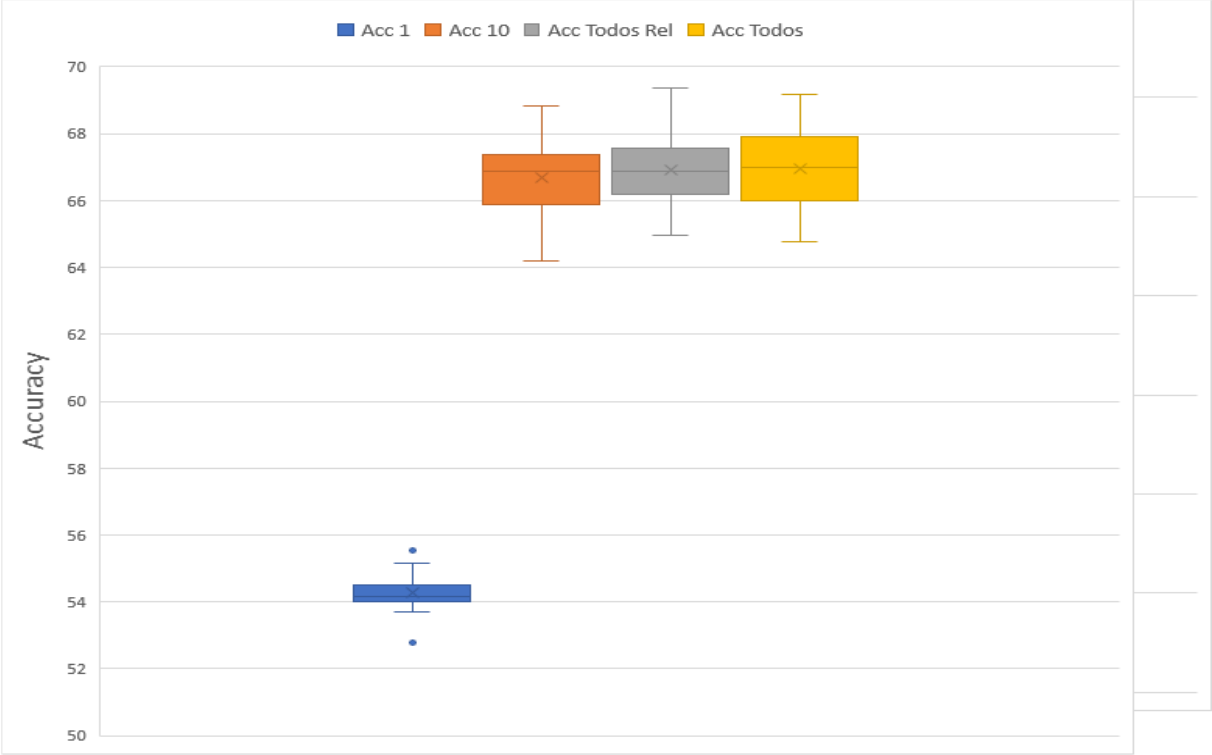


Figura 5.6: Acierto análisis componentes secuencias 11 y 12

relevantes y con todas las componentes.

Para contrastar los resultados se repite el estudio sobre las secuencias 14 y 40, comprobando que el peor resultado se obtiene para la configuración con únicamente la componente más relevante, y volviendo a observar como los resultados de las otras tres configuraciones son estadísticamente equivalentes, lo cuál implica que utilizando solo las componentes relevantes la red recibe la misma información que utilizando todas las componentes, pero a un menor coste computacional pues al disminuir el numero de componentes disminuye el numero de pesos. Por esto en el método utilizaremos la configuración con todas las componentes relevantes pues aunque sea más costosa que utilizando solo las 10 más relevantes también es más estable pues presenta una menor desviación estándar (Tablas 5.7 y 5.8)

Tras estudiar el número de componentes ideales para este método se procede a evaluar su rendimiento sobre el conjunto de datos de validación obteniendo una precisión del 57.75 %.

Componentes	Acc (%)	Desv. Estándar	Margen de error
Componente más relevante	93.364	0.879	0.16
10 componentes más relevantes	97.101	0.649	0.12
Todas las componentes más relevantes	97.021	0.500	0.09
Todas las 75 componentes	97.125	0.206	0.04

Tabla 5.8: Análisis componentes secuencias 14 y 40

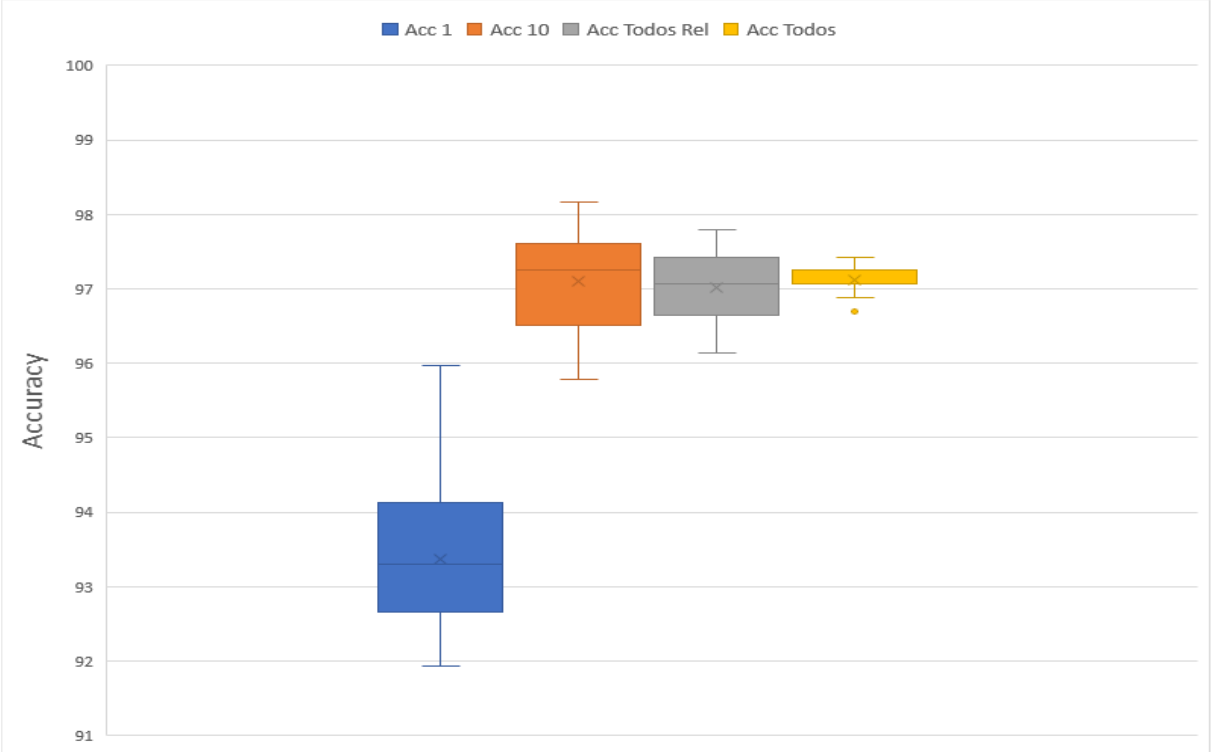


Figura 5.7: Acierto análisis componentes secuencias 14 y 40

Aunque el porcentaje de acierto general de este método es peor que el del clasificador con una sola red, en este método no hay ninguna clase que presente un porcentaje de acierto del 0.0 % como se observa en la matriz de confusión (figura 5.8), siendo los peores porcentajes de acierto para las clases 41 ("Bostezar/toser") y 60 ("Dos personas alejándose una de otra") con un 20.65 % y un 18.0 % respectivamente, y la desviación estándar del acierto de las distintas clases de un 16.64 %.

En esta ocasión es más comprensible que la clase 40 presenta una de las peores tasas de acierto debido a que la acción de bostezar/toser se realiza principalmente mediante un movimiento de la boca/mandíbula y esta no se encuentra reflejada por ninguna articulación, por lo que es un movimiento que no se ve reflejado en el esqueleto. A su vez la clase 60 presenta una de las peores tasas por el problema mencionado anteriormente del estudio de un solo esqueleto puesto que la principal diferencia con la clase 59 ("Dos personas caminando la una hacia la otra") es la relación de acercamiento/alejamiento entre los dos esqueletos.

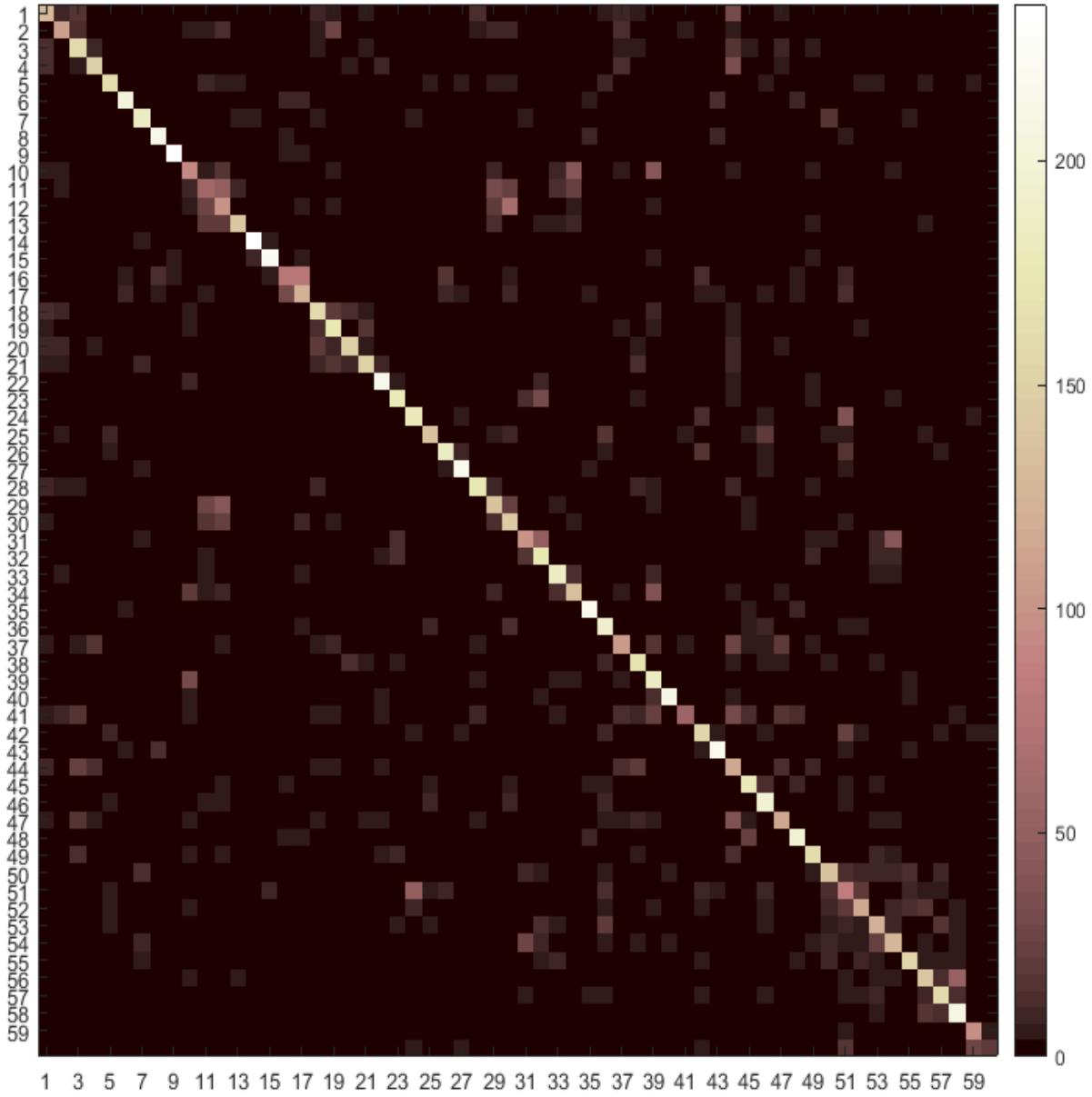


Figura 5.8: Matriz de confusión del método de las componentes características

En cuanto al coste temporal, como se puede observar en la tabla 5.9, cada una de las 60 redes ha tardado de media una hora en ser entrenada, aunque el tiempo total ha sido de casi una semana debido a las limitaciones de la plataforma utilizada. Cada secuencia ha tardado de media de 0.40 segundos en ser evaluada.

5.3.3. Método del árbol

El método del dendograma se basa en comparar la secuencia a clasificar a lo largo de un árbol de decisiones, donde en cada bifurcación se decide si pertenece a un grupo u otro de clases mediante una red especializada en dichos grupos.

Las redes se han entrenando sobre los actores del grupo B (tabla 5.1) durante

un máximo de 100 épocas, parando cuando la función de pérdida dejaba de mejorar durante 10 épocas seguidas y guardando el modelo con mejor prestación durante el entrenamiento.

Para obtener el rendimiento del método se han evaluado todas las secuencias del conjunto de validación, grupo de actores A (tabla 5.1), obteniendo un porcentaje de acierto del 55.77 %. En este método tampoco hay ninguna clase que presente un porcentaje de acierto del 0.0 % como se observa en la matriz de confusión (figura 5.9), siendo los peores porcentajes de acierto para las clases 1, 41 y 60 con un 30.63 %, 32.97 % y un 29.0 % de acierto respectivamente, y la desviación estándar del acierto de las distintas clases de un 13.25 %.

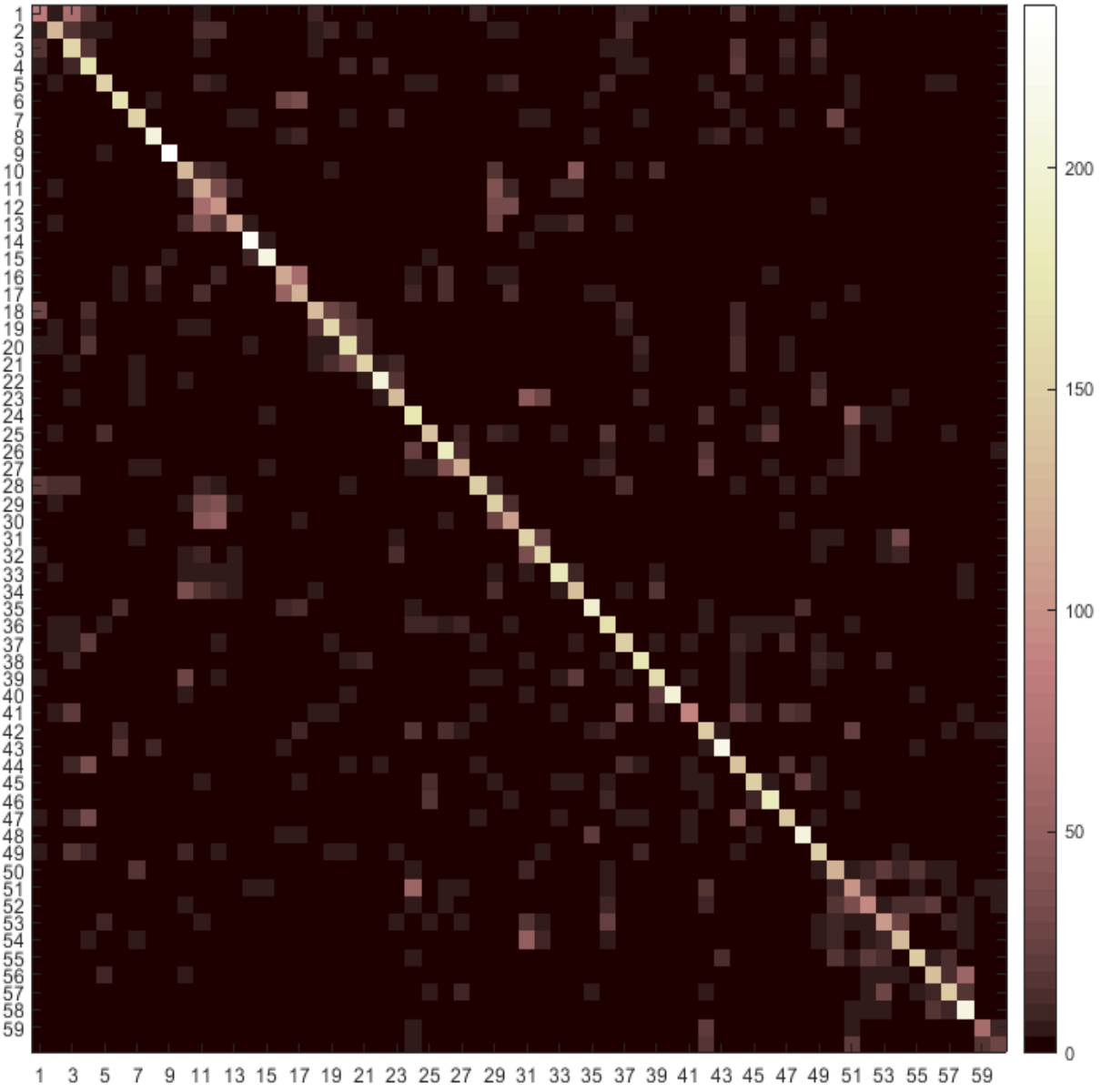


Figura 5.9: Matriz de confusión del método del árbol

En cuanto al coste temporal, como se puede observar en la tabla 5.9, el tiempo de entrenamiento de las redes ha sido muy dispar, tardando las redes que distinguían entre solo dos clases apenas unos 3 minutos, y las redes de los niveles más altos que distinguían entre grupos que incluían todas o casi todas las redes entre 50 minutos y una hora. Con respecto al tiempo de evaluación Cada secuencia ha tardado de media menos de 0.1 segundo en ser evaluada.

5.3.4. Método del árbol + características discriminantes

Como mejora del método del árbol se ha probado a entrenar los modelos del árbol de decisiones sobre las características más discriminantes (sección 3.2) de los distintos conjuntos de clases.

Este método se ha desarrollado primero para las acciones con un solo actor (de la clase 1 a la 49) con la idea de reducir los costes computacionales, y pensando en ampliarlo a todo el conjunto de clases en segunda instancia.

Las redes se han entrenando sobre los actores del grupo B (tabla 5.1) durante un máximo de 100 épocas, parando cuando la función de pérdida dejaba de mejorar durante 10 épocas seguidas y guardando el modelo con mejor prestación durante el entrenamiento.

Para obtener el rendimiento del método se han evaluado todas las secuencias del conjunto de validación, grupo de actores A (tabla 5.1), obteniendo un porcentaje de acierto del 47.79 % , un resultado peor del esperado, sobre todo teniendo en cuenta que al disminuir el número de clases el porcentaje de acierto debería aumentar. En este método tampoco hay ninguna clase que presente un porcentaje de acierto del 0.0 % como se observa en la matriz de confusión (figura 5.10), siendo los peores porcentajes de acierto para las clases 1 y 41 con un 14.39 % y un 8.70 % de acierto respectivamente, y la desviación estándar del acierto de las distintas clases de un 15.13 %.

En cuanto al coste temporal, como se puede observar en la tabla 5.9, el tiempo de entrenamiento de las redes ha sido menor que en el método original, abarcando desde minutos para las redes que distinguen entre solo dos acciones, hasta 30 minutos para las redes del principio del modelo que trabajan con todo el conjunto. Esto se debe a que al utilizar solo las características más discriminantes disminuye el espacio de características disminuyendo los pesos de las redes. Con respecto al tiempo de evaluación este se ha mantenido con respecto al método original, tardando cada secuencia menos de 0.1 segundos de media en ser evaluada.

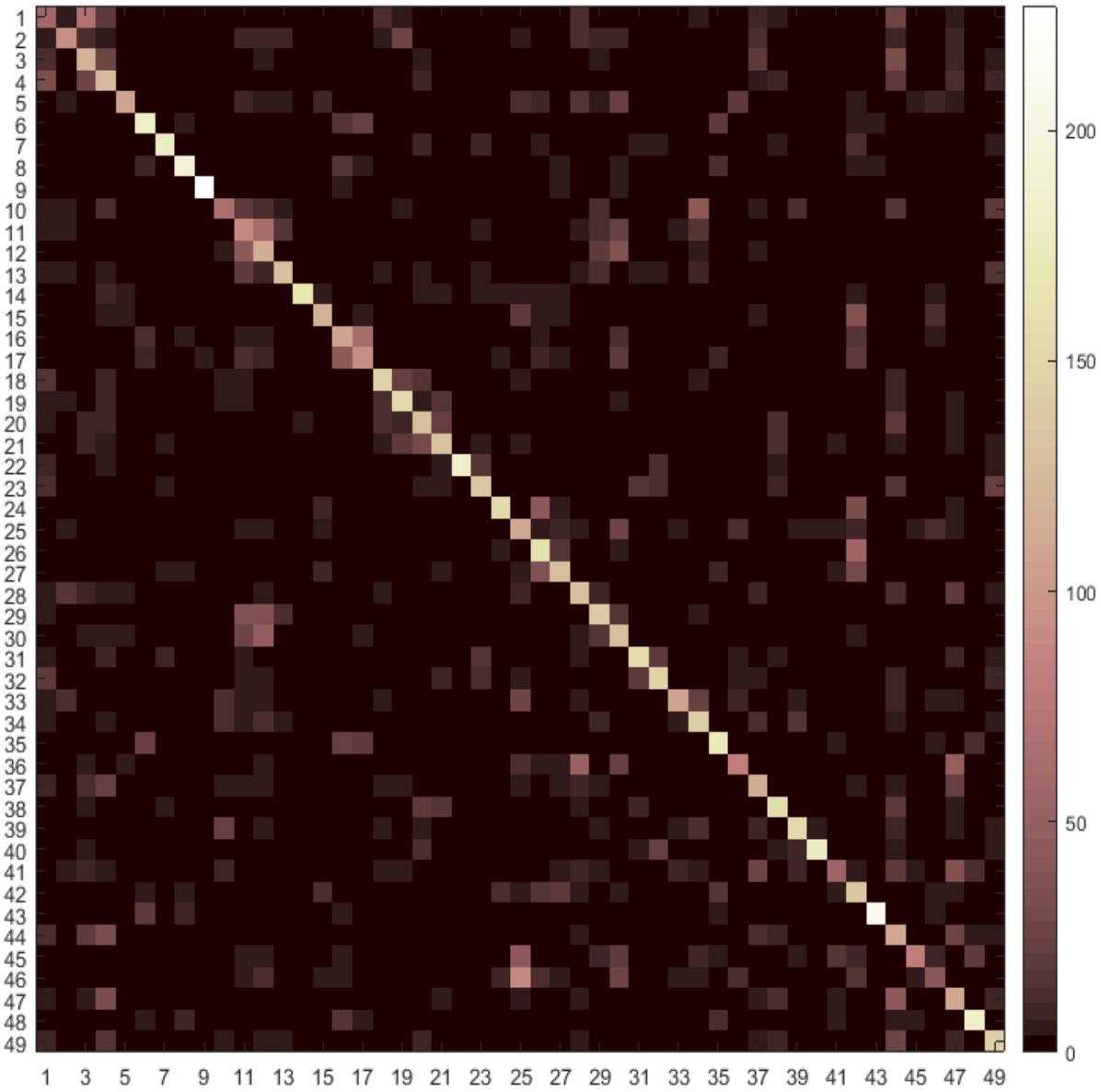


Figura 5.10: Matriz de confusión del método del árbol

Método	Acierto	Coste entrenamiento	Coste evaluación
LSTM Bidireccional	66.9 %	5 horas	0.35 segundos
Componentes características	57.75 %	60 horas	0.40 segundos
Árbol	55.77 %	30 horas	0.1 segundos
Árbol + componentes discriminantes	47.79 %	15 horas	0.1 segundos

Tabla 5.9: Coste computacional de los distintos métodos

Capítulo 6

Conclusiones y trabajo futuro

El reconocimiento de acciones humanas es un campo que puede aportar mucha información para facilitar tareas de videovigilancia en distintos ámbitos. Sin embargo, es un problema cuyas soluciones aún presentan margen de mejora.

Se ha estudiado una de las bases de datos más grande y completas de esqueletos humanos en 3D. A partir de dicho estudio se ha obtenido las componentes más características de cada clase, lo que ha permitido un ahorro del coste computacional sin apenas pérdida de información. También se ha intentado obtener las características más discriminantes entre distintos grupos de clases pero sin obtener una mejora de los resultados.

Se han desarrollado un método propio de normalización de la longitud de las secuencias basado en la longitud media, y tres clasificadores distintos, con un minucioso estudio de los parámetros de cada clasificador, aunque obteniendo resultados lejos del estado del arte (76.1 % GCA-LSTM [8]).

Estos estudios han servido para comprobar la gran cantidad de hiperparámetros a ajustar para conseguir el máximo rendimiento de una red neuronal, la complejidad que puede implicar querer estudiarlos todos, y como la extracción de características realizada automáticamente por una red neuronal es considerablemente mejor que la realizada manualmente, algo que se observa al comparar los resultados de la red LSTM-Bidireccional con los de los métodos propios.

Por todo ello se puede concluir que los objetivo propuestos han sido cumplidos. Sin embargo aún queda trabajo por realizar. El estudio de estos algoritmos sobre otras bases de datos, la selección de características que permitan facilitar la clasificación de las acciones más similares, la combinación del método de una capa LSTM-Bidireccional con algoritmos que permitan evitar acciones con un 0 % de acierto, la disminución del número de bifurcaciones en el método del árbol con el fin de disminuir la probabilidad de error, o la implementación física de estos algoritmos son trabajos para el futuro.

Capítulo 7

Bibliografía

[1] Lei Wang, Du Q Huynh, and Piotr Koniusz. A Comparative Review of Recent Kinect-based Action Recognition Algorithms. *CoRR*, 2019.

[2] Amir Shahroudy, Tian Tsong Ng, Qingxiong Yang, and Gang Wang. Multimodal Multipart Learning for Action Recognition in Depth Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2123–2129, 2016.

[3] Amir Shahroudy, Tian Tsong Ng, Yihong Gong, and Gang Wang. Deep Multimodal Feature Analysis for Action Recognition in RGB+D Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1045–1058, 2018.

[4] Amir Shahroudy, Jun Liu, Tian-tsong Ng, and Gang Wang. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[5] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-Temporal LSTM with Trust Gates For 3D Human Action Recognition. *European Conference on Computer Vision (ECCV)*, 2016.

[6] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Skeleton-Based Action Recognition Using Spatio-Temporal LSTM Network with Trust Gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[7] Jun Liu, Gang Wang, Ping Hu, Ling-yu Duan, and Alex C Kot. Global Context-Aware Attention LSTM Networks for 3D Action Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdyeva, and Alex C. Kot. Skeleton Based Human Action Recognition with Global Context-Aware Attention LSTM Networks. *IEEE Transactions on Image Processing (TIP)*, 2018.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning Book*. MIT Press, 2016.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural computation*, 1780:1735–1780, 1997.

[11] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-yu Duan, and Alex C Kot. NTU RGB + D 120 : A Large-Scale Benchmark for 3D Human Activity Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Lista de Figuras

Lista de Tablas

Anexos

Anexos A

Lista clases NTU RGB+D

Las clases presentes en la base de datos NTU RGB+D 60 son:

- A1. Beber agua.
- A2. Comer.
- A3. Cepillarse los dientes.
- A4. Cepillarse el pelo.
- A5. Dejar caer.
- A6. recoger.
- A7. Tirar.
- A8. Sentarse.
- A9. Levantarse (desde posición sentada).
- A10. Aplaudir.
- A11. Leer.
- A12. Escribir.
- A13. Romper un papel.
- A14. Ponerse la chaqueta.
- A15. Quitarse la chaqueta.
- A16. Ponerse un zapato.
- A17. Quitarse un zapato.

- A18. Ponerse gafas.
- A19. Quitarse las gafas.
- A20. Ponerse un sombrero/gorro.
- A21. Quitarse el sombrero/gorro.
- A22. Alegrarse.
- A23. Sacudir la mano.
- A24. Patear algo.
- A25. Meterse la mano en el bolsillo.
- A26. Saltar sobre una pierna.
- A27. Saltar.
- A28. Hacer una llamada/responder al teléfono.
- A29. Jugar con el móvil/la tablet.
- A30. Escribir en un teclado.
- A31. Apuntar algo con un dedo.
- A32. Hacerse una foto (autofoto).
- A33. Mirar la hora del reloj de mano/bolsillo.
- A34. Frotar las dos manos juntas.
- A35. Asentir con la cabeza.
- A36. Sacudir la cabeza.
- A37. Limpiarse la cara.
- A38. Saludar.
- A39. Juntar las palmas de las manos.
- A40. cruzar las manos de frente (decir stop).
- A41. bostezar/toser.
- A42. Asombrarse.

- A43. Caerse.
- A44. Tocarse la cabeza (dolor de cabeza).
- A45. Tocarse el pecho (dolor de estomago/dolor de pecho).
- A46. Tocarse la espalda (dolor de espalda).
- A47. Tocarse el cuello (dolor de cuello).
- A48. Nausea o situación de vómito.
- A49. Abanicar (con la mano o un papel)/sentir calor.
- A50. Dar un puñetazo/una bofetada a otra persona.
- A51. Dar una patada a otra persona.
- A52. Empujar a otra persona.
- A53. Dar una palmada en la espalda a otra persona.
- A54. Señalar con el dedo a otra persona.
- A55. Abrazar a otra persona.
- A56. Dar algo a otra persona.
- A57. Tocar el bolsillo de otra persona.
- A58. Dar la mano.
- A59. Dos personas caminando una hacia la otra.
- A60. Dos personas alejándose una de la otra.

Anexos B

Redes neuronales

B.1. RNN

En una red neuronal recurrente, ecuaciones B.1 y B.2, el estado oculto de cada instante (h_t) se actualiza en cada paso temporal (t) como una función lineal (f) dependiente del estado anterior (h_{t-1}) y de la entrada actual (x_t), seguida de una función de escala no-lineal (σ). La salida de la red de cada instante (Y_t) se obtiene como una función no lineal del estado oculto (h_t).

$$h_t = \sigma\left(f\begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}\right) \tag{B.1}$$

$$Y_t = \sigma\left(V h_t\right)\epsilon_0 \tag{B.2}$$

Donde $\sigma\epsilon\{Sigm, Tanh\}$ es una función no lineal y V representa la matriz de pesos.

B.2. LSTM

El funcionamiento de una celula de memoria de una red LSTM se modela con las ecuaciones B.3, B.4 y B.5, donde c_t representa la célula de memoria, i representa la puerta de entrada, f representa la puerta de olvide, o representa la puerta de salida, g representa la puerta de modulación, y el operador $\odot$ un producto elemento a elemento.

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} Sigm \\ Sigm \\ Sigm \\ Tanh \end{pmatrix}\left(W\begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}\right) \tag{B.3}$$

$$c_t = f \odot c_{t-1} + i \odot g \tag{B.4}$$

$$h_t = o \odot Tanh\left(c_t\right) \tag{B.5}$$

Anexos C

Método del árbol

C.1. Esquema del método del árbol

Debido a la extensión del esquema se ha dividido el árbol en tres figuras, representando la figura C.1 el principio del árbol de decisiones en el modelo 58; la figura C.2 la bifurcación a partir del modelo 42; y la figura C.3 la bifurcación a partir del modelo 48. En todas las bifurcaciones la izquierda representa un valor de salida de la red de la bifurcación menor que 0.5, y la derecha un valor mayor o igual que 0.5.

En el esquema los modelos de redes neuronales que clasifican las secuencias entre una de las dos bifurcaciones se representan como M_{Num} en círculos naranja. Los modelos cuya representación continua en otra figura se representan en círculos azules, y las clases que marcan el final del proceso de reconocimiento de las secuencias se representan como C_{Num} en círculos rojos.

C.2. Esquema del método del árbol + variantes discriminantes

Este método se ha desarrollado solo para las clases con un solo esqueleto (de la 1 a la 49).

Debido a la extensión del esquema se ha dividido el árbol en tres figuras, representando la figura C.4 el principio del ñ árbol de decisiones en el modelo 47; la figura C.5 la bifurcación a partir del modelo 42; y la figura C.6 la bifurcación a partir del modelo 36. En todas las bifurcaciones la izquierda representa un valor de salida de la red de la bifurcación menor que 0.5, y la derecha un valor mayor o igual que 0.5.

En el esquema los modelos de redes neuronales que clasifican las secuencias entre una de las dos bifurcaciones se representan como M_{Num} en círculos naranja. Los modelos cuya representación continua en otra figura se representan en círculos azules, y las clases

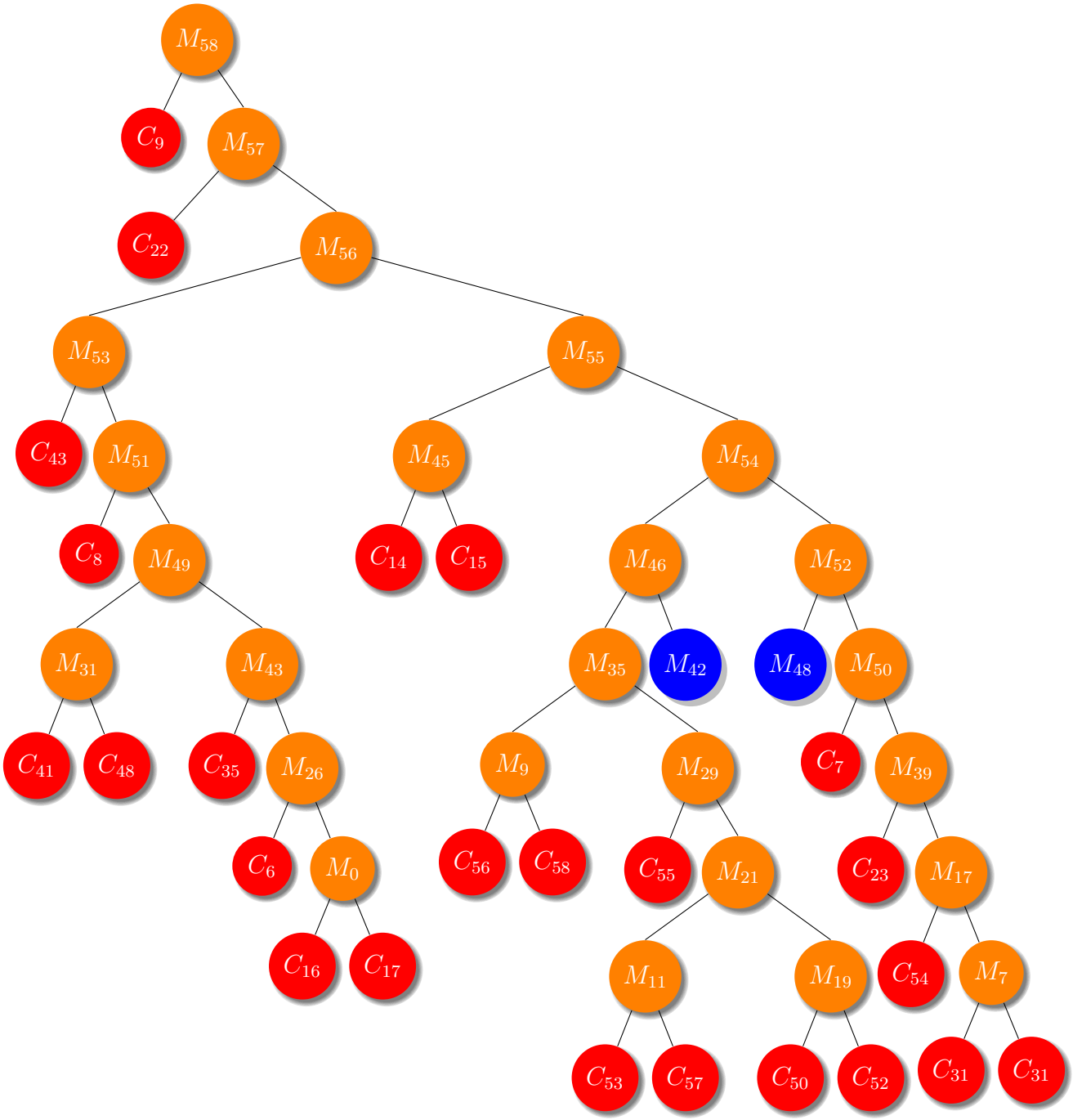


Figura C.1: Árbol de decisiones del modelo del árbol. Representación desde el modelo 58 (el primero).

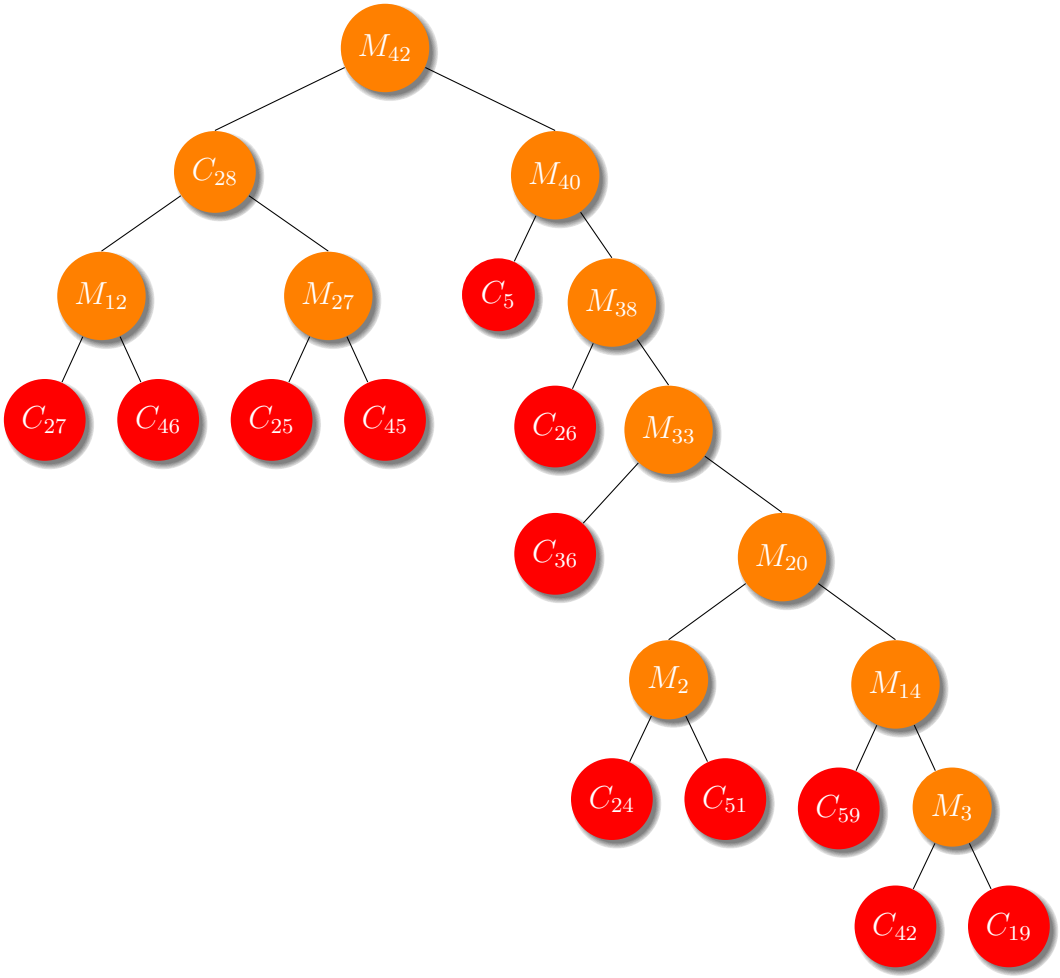


Figura C.2: Árbol de decisiones del modelo del árbol. Representación desde el modelo 42.

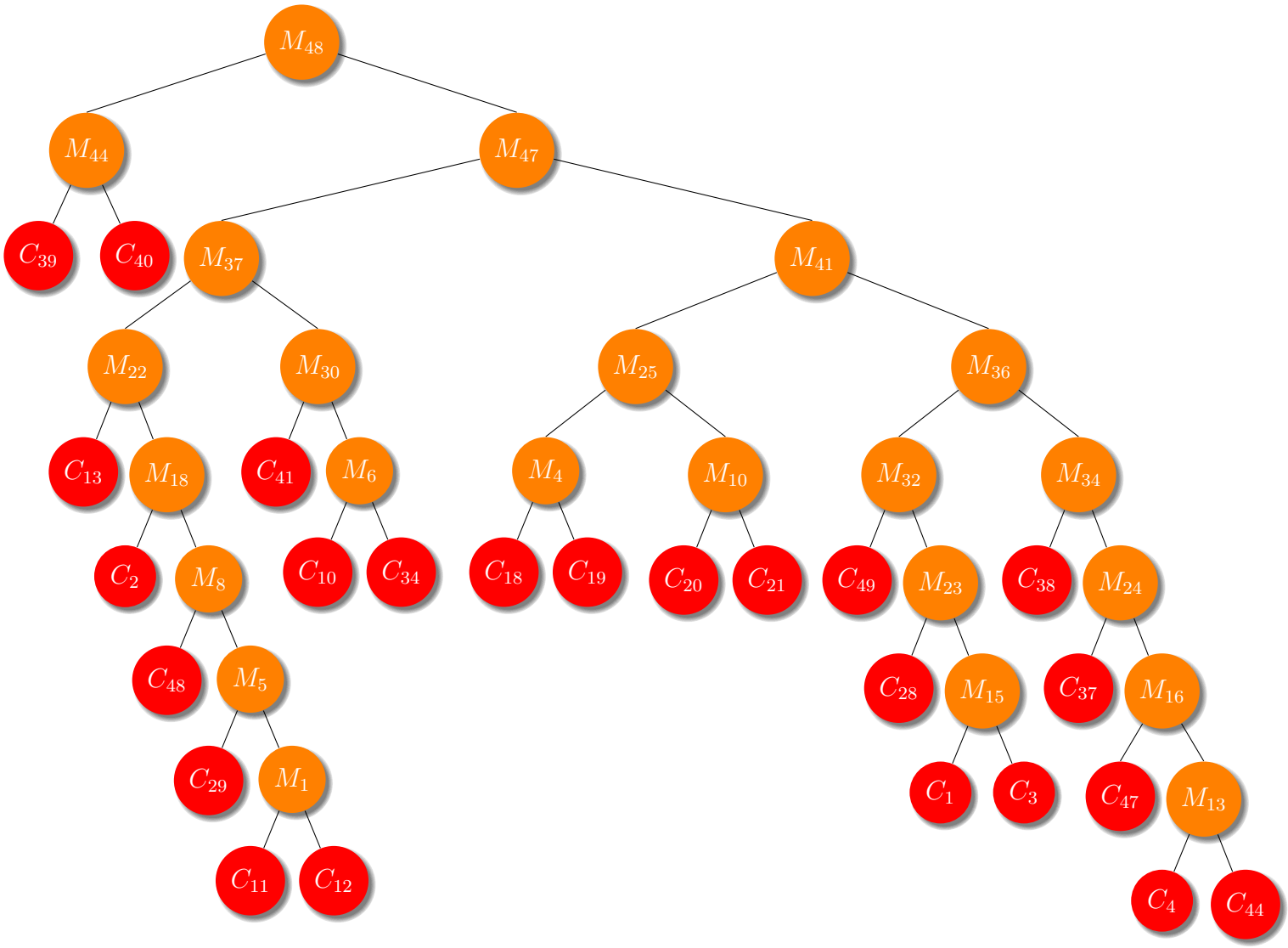


Figura C.3: Árbol de decisiones del modelo del árbol. Representación desde el modelo 48.

que marcan el final del proceso de reconocimiento de las secuencias se representan como C_{Num} en círculos rojos.

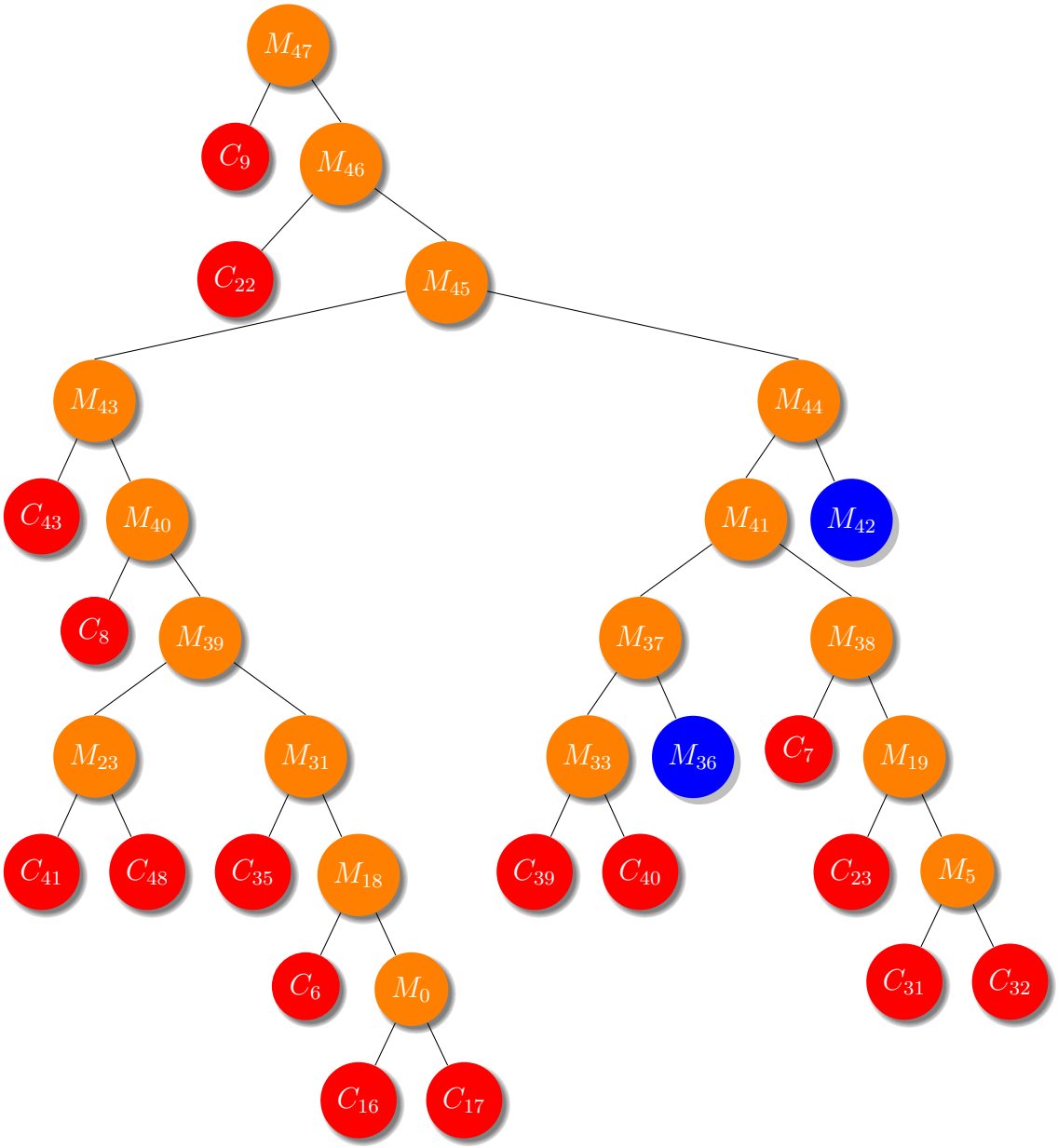


Figura C.4: Árbol de decisiones del modelo del árbol + variantes discriminantes. Representación desde el modelo 47 (el primero).

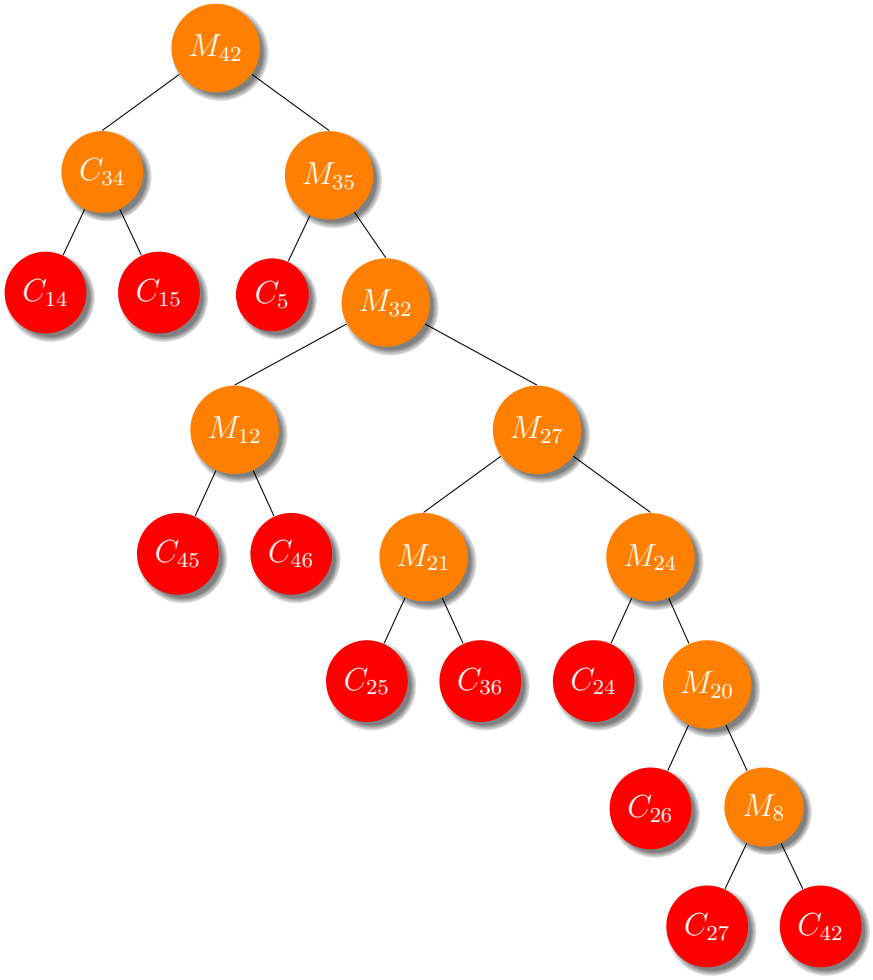


Figura C.5: Árbol de decisiones del modelo del árbol. Representación desde el modelo 42.

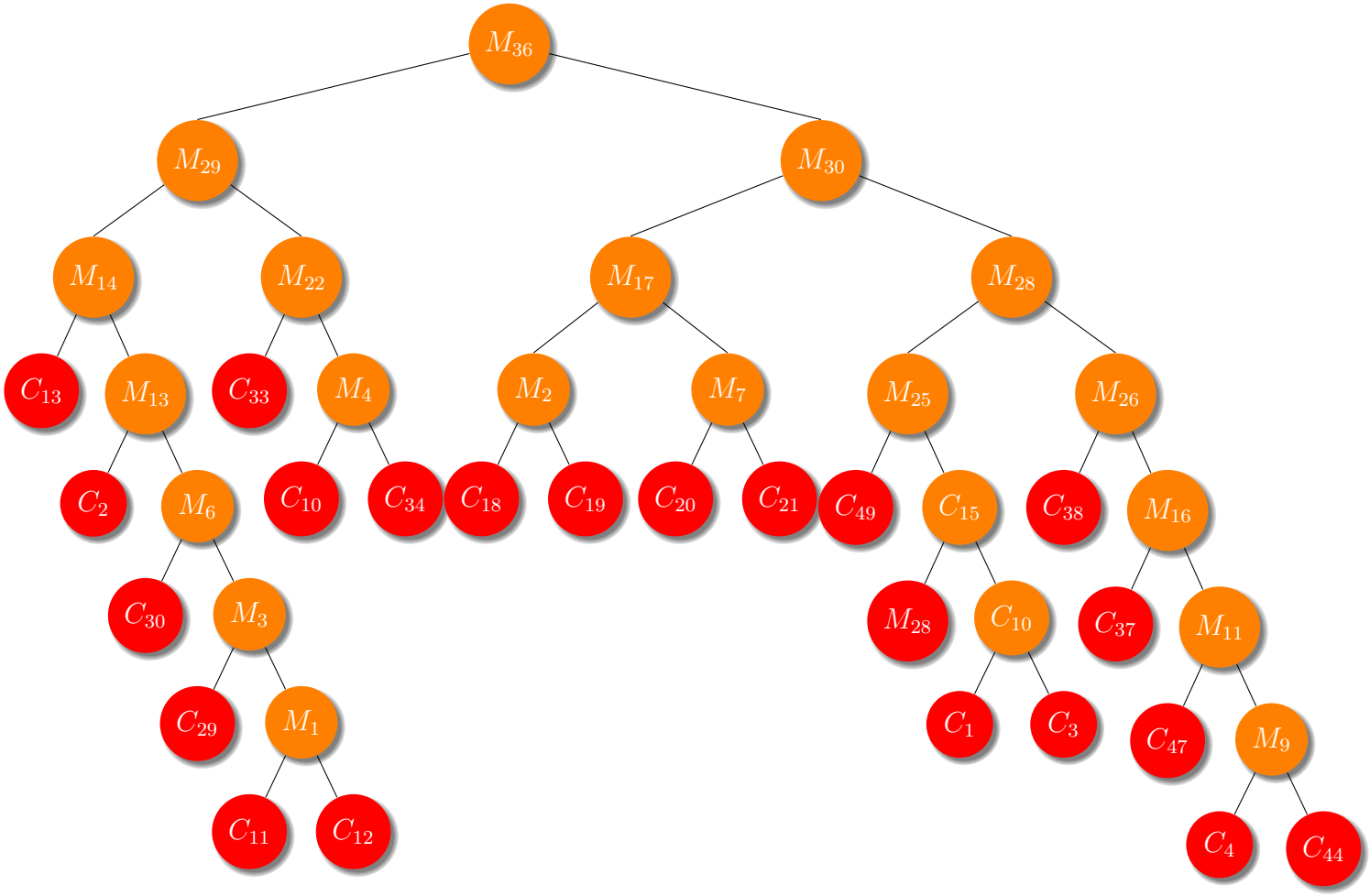


Figura C.6: Árbol de decisiones del modelo del árbol. Representación desde el modelo 36.