

## TRABAJO FIN DE MÁSTER

Máster en Ciencia de Datos e Ingeniería de Datos en la Nube

# Plantilla para la elaboración de la memoria del TFM en el Máster en Ciencia de Datos e Ingeniería de Datos en la Nube (TÍTULO)

**Autor:** Agustín Piqueres Lajarín

**Tutor:** Daniel González Medina

Septiembre, 2022



*Aquí va la dedicatoria que cada cual  
quiera escribir. El ancho se controla  
manualmente*

## Declaración de autoría

Yo, ... , con DNI ... , declaro que soy el único autor del trabajo fin de grado titulado “ ... ”, que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual, y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a ... de ... de 20 ...

Fdo.: Agustín Piqueres Lajarín

## Resumen

resumen

## Agradecimientos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



## Índice general

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación	1
1.2	Objetivos	1
1.3	Estructura del proyecto	1
<b>2</b>	<b>Estado del arte</b>	<b>3</b>
2.1	Deep Learning	3
2.2	Cloud	3
2.3	Trabajos relacionados	3
<b>3</b>	<b>Desarrollo del proyecto</b>	<b>5</b>
3.1	Extracción y recolección de datos	5
3.1.1	<i>Introducción</i>	5
3.1.2	<i>Proceso de extracción</i>	5
3.1.3	<i>Datos obtenidos</i>	6
3.2	Experimentación con deep learning	6
3.2.1	<i>Introducción</i>	7
3.2.2	<i>Preprocesado de los datos</i>	7
3.2.3	<i>Experimentos realizados y resultados</i>	7
3.2.4	<i>Evaluación de los resultados</i>	8
3.3	Cloud y despliegue de la aplicación	8
3.3.1	<i>Introducción</i>	8
3.3.2	<i>Arquitectura cloud (diagrama)</i>	8
3.3.3	<i>Resultado y funcionamiento (alguna imagen)</i>	8



<b>4 Conclusiones y trabajo futuro.....</b>	<b>9</b>
<b>A Anexo 1.....</b>	<b>11</b>

# Índice de figuras

---

3.1    Proceso de extracción de datos    . . . . . 6



## Índice de tablas

---



# 1. Introducción

---

## 1.1. Motivación

texto

## 1.2. Objetivos

texto

## 1.3. Estructura del proyecto

texto

---

## 2. Estado del arte

---

texto

### 2.1. Deep Learning

texto

### 2.2. Cloud

texto

### 2.3. Trabajos relacionados

texto



---

## 3. Desarrollo del proyecto

---

### 3.1. Extracción y recolección de datos

#### 3.1.1. Introducción

En el siguiente apartado se expone la metodología de obtención de los datos de entrenamiento.

Las distintas versiones de MoViNets están entrenadas sobre Kinetics 600 (REFERENCIA: KINETICS), por lo que para reentrenar el modelo necesitamos clips (videos) que representen los movimientos que se quieren predecir.

Si bien no existe un conjunto de datos similar aplicado específicamente a movimientos de crossfit, muchos atletas suben videos a youtube de realizando distintas pruebas y ejercicios. Partiendo de estos videos, se pueden extraer los distintos *clips* (trozos) que representen los movimientos, con los que reentrenar el modelo.

Todos los videos descargados pertenecen a la primera etapa de los Crossfit Games de 2020 (REFERENCIA: CFGAMES2020). Este año fue algo atípico debido a la pandemia. La competición se distinguió en dos etapas, una primera en formato online, y la final a la que solo pudieron acceder 5 hombres y 5 mujeres. Los videos correspondientes a la primera etapa están todos subidos a YouTube, donde todos los atletas están grabados de forma individual y con la cámara estática, lo que reduce enormemente el ruido de los datos.

#### 3.1.2. Proceso de extracción

Para seleccionar la muestra de videos, se ha partido de 5 *wods* (PONER NOTA AL PIE CON LA DEFINICIÓN) que en total dan pie a 9 movimientos (NOTA AL PIE, REFERENCIA A LABELS).

De cada *wod* se han seleccionado 20 videos (10 de hombres y 10 de mujeres), de los cuales se han etiquetado 15 repeticiones de cada movimiento presente. Esto nos deja con una muestra de alrededor de 300 repeticiones por movimiento, 2700 clips.

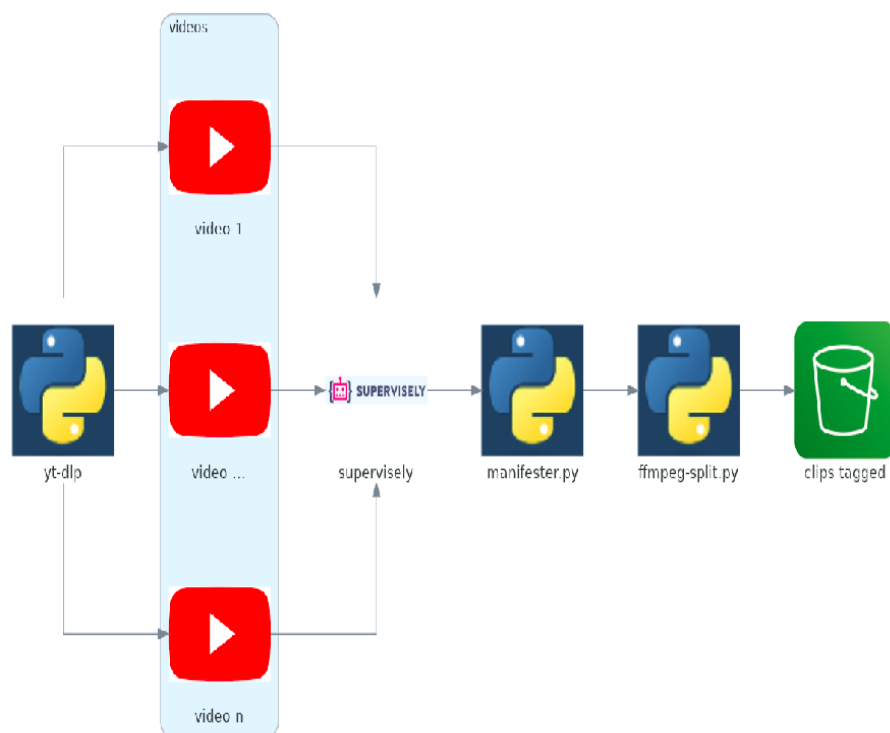
Para descargar los videos se ha utilizado `yt-dlp`, todos con la misma resolución (480x854p) en formato mp4. (INSERTAR NOTA AL PIE CON SCRIPT PARA LA DESCARGA, FALTA CREAR REPO A PARTE PARA LOS SCRIPTS)

---

El etiquetado de los videos se ha hecho con *Supervisely*<sup>1</sup>. Una vez se ha etiquetado un video, *Supervisely* genera un json con las anotaciones correspondientes al movimiento y los frames en los que se produce.

Para obtener cada uno de los clips se ha utilizado (INSERTAR REFERENCIA A `ffmpeg-split.py` MIO)<sup>2</sup>. Las anotaciones se tranforman a un formato que `ffmpeg-split.py` pueda utilizar por medio de (INSERTAR REFERENCIA A `manifestester.py`).

La figura 3.1 resume el proceso para la obtención de los datos. (REFERENCIA A REPO Y DISTINTOS SCRIPTS)



**Figura 3.1:** Proceso de extracción de datos

### 3.1.3. Datos obtenidos

Meter tablas resumen de frames y segundos de los videos.

## 3.2. Experimentación con deep learning

texto

---

<sup>1</sup>Tras probar otras alternativas como *LabelStudio*, *Supervisely* facilita enormemente seleccionar rangos de frames por medio de atajos del teclado

<sup>2</sup>Este script es una adaptación de `ffmpeg-split.py`

### 3.2.1. Introducción

texto

### 3.2.2. Preprocesado de los datos

(ESCRIBIR LA ELECCIÓN DE TENSORFLOW PARA EL DESARROLLO DEL MODELO)

De cara a construir la pipeline para el entrenamiento del modelo, se ha utilizado la API de `tf.data.Dataset` ya que permite la ingesta de un conjunto potencialmente grande de datos, así como el procesamiento de los mismos.

La forma más eficiente (BUSCAR REFERENCIA) de leer los videos en el Dataset es escribirlos como `tf.data.TFRecordDataset`, `tf.data.TFRecordDataset` (REVISAR PARA QUE SALGA LA REFERENCIA) para lo cuál se deben transformar los videos en formato mp4 a `tfrecords`. (INSERTAR REFERENCIA A `movinets_helper/writer.py`, Y LOS DISTINTOS EJEMPLOS)

Al introducir los videos, se cambia el tamaño para igualarlo al utilizado en el entrenamiento original, 224x224p, y se escalan los valores para que se encuentren en el rango  $[0, 1]$ .

Para homogeneizar los datos de entrada se ha decidido fijar al número de frames para todos los videos a 10 ( $\bar{f} = 10$ )(BUSCAR REFERENCIA A LOS DATOS UTILIZADOS EN EL PAPER ORIGINAL). Debido a que los vídeos en este conjunto de datos son más cortos que los utilizados en el caso de *Kinetics 600*, se ha seleccionado el número de frames como  $\bar{f} = \min_v f_v$ , donde  $f_v$  es el número de frames del vídeo  $v$ . El menor número de frames corresponde a uno de los vídeos de *double-unders*. Para aquellos vídeos con un número de frames mayor que  $\bar{f}$ , se seleccionan  $\bar{f}$  frames equiespaciados.

### 3.2.3. Experimentos realizados y resultados

El modelo se ha entrenado en *Google Colab* con *Tensorflow 2*. El modelo seleccionado ha sido MoViNets A2 Base<sup>3</sup>, ya aunque sea necesario utilizar GPU para el fine-tuning, es el modelo más grande (con mejor capacidad predictiva) que permite hacer inferencia CON CPU en un tiempo razonable (BUSCAR REFERENCIA DE LOS AUTORES).

Los hiperparámetros del modelo son los mismos utilizados en el paper original (BUSCAR LOS DATOS EN EL PAPER ORIGINAL), con un tamaño de batch igual a  $8^4$

La muestra se divide en un 80% (2164 clips) para training, y un 20% (541 clips) para test. (INSERTAR FIGURA DE LA DISTRIBUCIÓN DE LAS MUESTRAS AQUÍ).

El modelo se ha entrenado por 10 epochs, evaluando top 1 y top 5 accuracy, obteniendo un accuracy de 0.9995 y 1 en entrenamiento y test respectivamente.

El modelo base generaliza a un gran número de movimientos, incluyendo algunos movimientos similares (REVISAR LABELS DE SQUAT Y CLEAN QUE HAN METIDO), lo que puede facilitar seguir aprendiendo algunos de movimientos similares.

---

<sup>3</sup>No ha sido posible hacer fine-tuning del modelo Stream debido a un error en los modelos guardados, ver issue 10730.

<sup>4</sup>El tamaño del batch seleccionado es el mismo utilizado por los autores en el tutorial a modo de ejemplo sobre el dataset UCF101.

---

Por otro lado, los videos en este caso, aunque se han seleccionado de una muestra distinta de atletas, todos ellos son atletas de élite en su deporte, por lo que la ejecución de los movimientos son muy similares entre si, lo que facilita aprender a clasificar los mismos.

Los resultados del entrenamiento se pueden ver en de forma interactiva en *Tensorboard.dev*

### 3.2.4. Evaluación de los resultados

A la hora de evaluar los resultados, empezamos analizando el heatmap de la matriz de confusión. Como se vio al analizar el accuracy en la muestra de validación, todos los movimientos se clasifican bien.

- INSERTAR heatmap matriz de confusión.

Con vistas a proporcionar un análisis más robusto, con un mayor número de movimientos o muestras de distintos atletas, puede ser útil la información que proporciona el heatmap REF. La construcción de este gráfico se hace de la siguiente forma. Partiendo de las probabilidades asignadas a cada vídeo en el período de test, se fija para cada movimiento (cada *label* correcta) y se calculan las correlaciones. Se selecciona la correlación (de Pearson) con el movimiento correctamente etiquetado, y se concatenan todos los vectores columna, de forma que obtenemos una matriz cuadrada. Estas correlaciones ayudan a ver cuáles son los movimientos que más se confunden entre si. Dado que las probabilidades deben sumar 1, aquellos movimientos con una correlación negativa más cercana a 1 son aquellos en los que se puede observar una mayor intercambio de probabilidades en la predicción.

(POR EJEMPLO, PONER ALGUNOS CASOS PARTICULARES DE LA TABLA Y CONTAR)

- heatmap matriz correlaciones (para ver con que otros movimientos se confunde).

Por último, se puede ver en la siguiente tabla las probabilidades medias y desviación estándar obtenidas para cada movimiento correctamente etiquetado. Esto nos permite ver cuáles son los movimientos que se predicen con una mayor confianza.

INSERTAR GRÁFICA DE LAS PROBABILIDADES POR CADA MOVIMIENTO EN EL ANEXO.

- tabla media/desv de las probabilidades para ordenar confianza en las predicciones.

## 3.3. Cloud y despliegue de la aplicación

En el siguiente apartado se presenta la aplicación para clasificar movimientos y su funcionamiento.

### 3.3.1. Introducción

texto

### 3.3.2. Arquitectura cloud (diagrama)

texto

### 3.3.3. Resultado y funcionamiento (alguna imagen)

texto



## 4. Conclusiones y trabajo futuro

---

texto





## A. Anexo 1

---

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.