



ULPGC

**Universidad de
Las Palmas de
Gran Canaria**

**Escuela de
Ingeniería Informática**



Verificación en escenarios multinivel con presencia de robots asistentes

Manual de Usuario

Proyecto creado por Kevin David Rosales Santana

Supervisado por:

Modesto Fernando Castrillón Santana

José Javier Lorenzo Navarro

Grado en Ingeniería Informática

Julio de 2020

Índice general

1	Distribución del repositorio	1
2	Visualización y ejecución de cuadernos <i>Jupyter</i>	3
2.1	<i>GitHub</i>	3
2.2	<i>IntelliJ IDEA Ultimate</i>	4
2.3	Entornos de <i>Jupyter</i>	7
3	Requisitos para la ejecución	8
3.1	Configuración del fichero de variables de entorno	8
3.2	Bibliotecas	8
3.3	Descarga de modelos	9
3.4	Celdas ejecutables sin <i>AveRobot</i>	9

Índice de figuras

1.1	Repositorio del proyecto	2
1.2	README.md del proyecto	2
2.1	Visualización de cuaderno <i>Jupyter</i> desde el propio repositorio	3
2.2	Instalación del <i>plugin Python</i> en <i>IntelliJ IDEA Ultimate</i>	4
2.3	Apertura de cuaderno dentro del IDE	4
2.4	Arranque del servidor <i>Jupyter</i>	5
2.5	Enlaces para acceder al cuaderno <i>Jupyter</i>	5
2.6	Acceso a los cuadernos del repositorio desde un servidor <i>Jupyter</i>	6
2.7	Ejecución de celdas desde un servidor <i>Jupyter</i>	6

1 Distribución del repositorio

El repositorio contiene la siguiente distribución de ficheros (ver figura 1.1):

- **src/**: comprende todos los módulos desarrollados a lo largo del Trabajo de Fin de Grado. La funcionalidad de los distintos módulos y sus ficheros de código ha sido previamente explicada en la memoria del proyecto.
- **out/**: incluye diversas salidas de los distintos módulos. Dichas salidas sirven como entradas para otros módulos, de manera que el sistema de verificación se encuentre conectado entre sus distintas partes.
- **main-lift.ipynb**: principal cuaderno *Jupyter* [1] desarrollado. Incluye un estudio profundo sobre los prototipos que hacen uso del Subconjunto de Vídeos B (localización del ascensor) y su correspondiente evaluación. Además, como en el resto de cuadernos, utiliza los distintos módulos que se encuentran en la carpeta **src/** para obtener los datos necesarios para realizar la verificación en escenarios multinivel usando *AveRobot* [2] como conjunto de datos.
- **random.ipynb**: cuaderno *Jupyter* que realiza un estudio de los mejores prototipos obtenidos en el cuaderno **main-lift.ipynb** (tanto en la verificación mediante umbral de distancia como en la verificación mediante redes neuronales) en el Subconjunto de Vídeos A (localización aleatoria).
- **corridor.ipynb**: cuaderno *Jupyter* con la misma funcionalidad que **random.ipynb** con la diferencia de realizar el estudio sobre el Subconjunto de Vídeos C (localización del pasillo).
- **stairs.ipynb**: cuaderno *Jupyter* con la misma finalidad que **random.ipynb** diferenciándose en que **stairs.ipynb** realiza el estudio sobre el Subconjunto de Vídeos D (localización de las escaleras).
- **README.md**: fichero Markdown [3] que contiene una guía breve sobre información de contexto del repositorio (ver figura 1.2). Incluye una serie de tablas con los resultados de los distintos prototipos desarrollados a lo largo del proyecto.
- **.env**: fichero de configuración de variables de entorno del proyecto. En la sección 3.1 se explicará cómo se deben definir sus distintas variables en caso de desear ejecutar los cuadernos.

- `.gitignore`: fichero cuya finalidad es especificar de manera intencional qué ficheros se quieren ignorar a la hora de actualizar el contenido del repositorio desde el equipo local.
- `manual-de-usuario.pdf`: el presente manual de usuario.

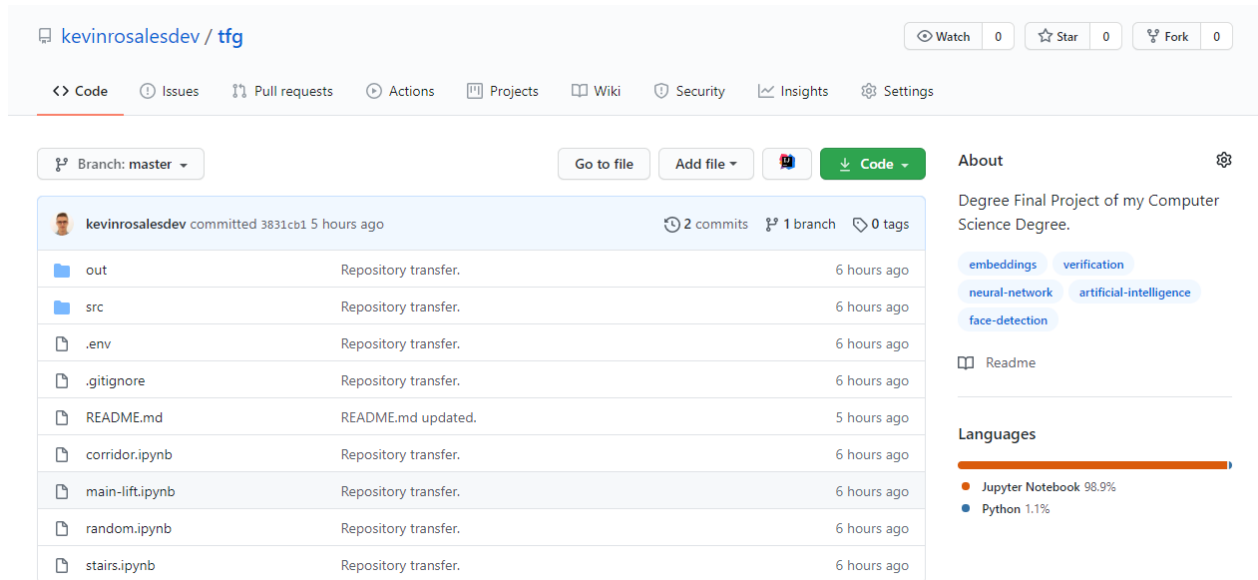


Figura 1.1: Repositorio del proyecto [4]



Figura 1.2: README.md del proyecto [Fuente]

2 Visualización y ejecución de cuadernos *Jupyter*

Respecto a la visualización y ejecución de los cuatro cuadernos *Jupyter* [1] presentes en el proyecto, se proponen los siguientes tres medios:

2.1 *GitHub*

GitHub [5] permite la visualización de los cuadernos *Jupyter* desde la propia página. Tan solo es necesario realizar *click* sobre uno de los cuadernos en cuestión para acceder a su contenido.

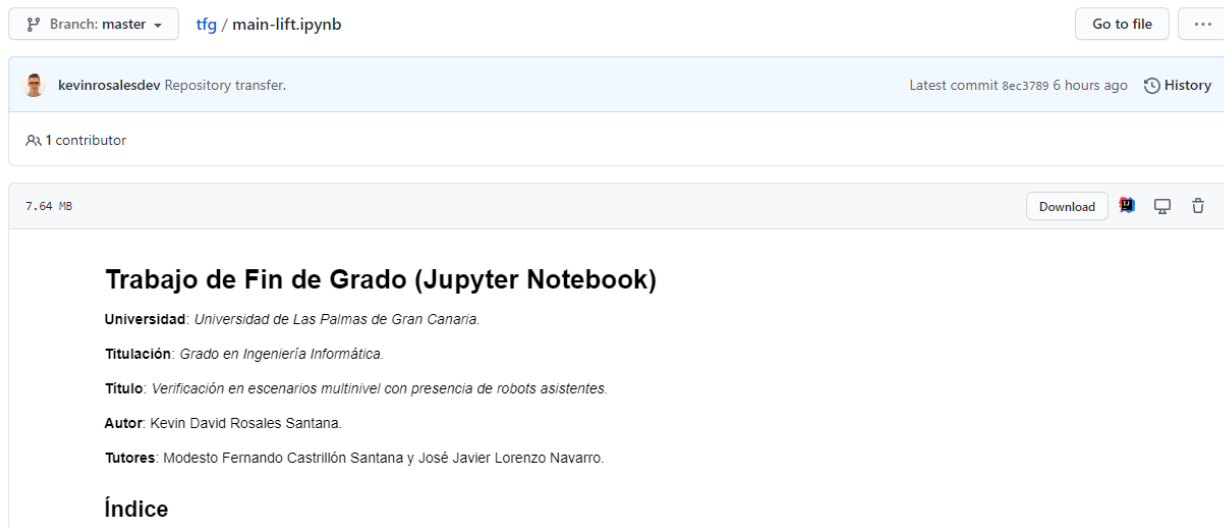


Figura 2.1: Visualización de cuaderno *Jupyter* desde el propio repositorio [Fuente]

Sin embargo, no se recomienda esta opción debido a que la visualización del cuaderno es algo desestructurada. Además, este medio tampoco permite la ejecución de las celdas pertenecientes a los cuadernos.

2.2 *IntelliJ IDEA Ultimate*

IntelliJ IDEA Ultimate [6] [7] es un IDE que permite la visualización y ejecución de cuadernos *Jupyter* [8]. Siendo la manera recomendada para la ejecución de los cuadernos en este proyecto, el primer paso que se debe hacer es añadir la extensión de *Python* al IDE tras obtenerlo, como se indica en la siguiente figura:

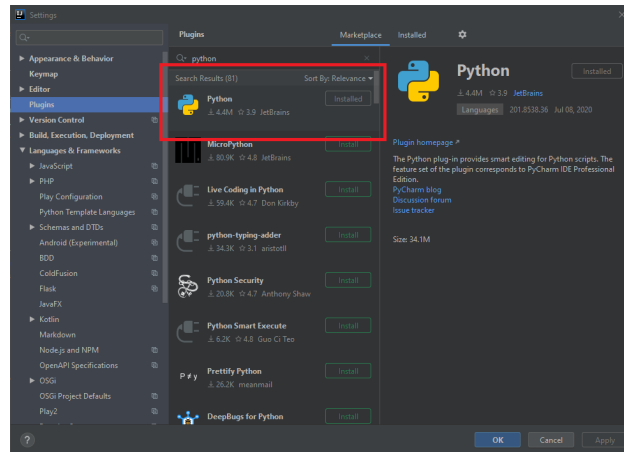


Figura 2.2: Instalación del *plugin Python* en *IntelliJ IDEA Ultimate* [Fuente]

Otra opción alternativa a *IntelliJ IDEA Ultimate* es tratar de usar *PyCharm* [9], creado por los mismos desarrolladores (JetBrains [10]).

Tras ello y después de realizar la configuración del intérprete de *Python* para el proyecto, se puede proceder a ejecutar el cuaderno en cuestión abriendo dicho cuaderno desde el IDE, tal y como se muestra en las figura 2.3:

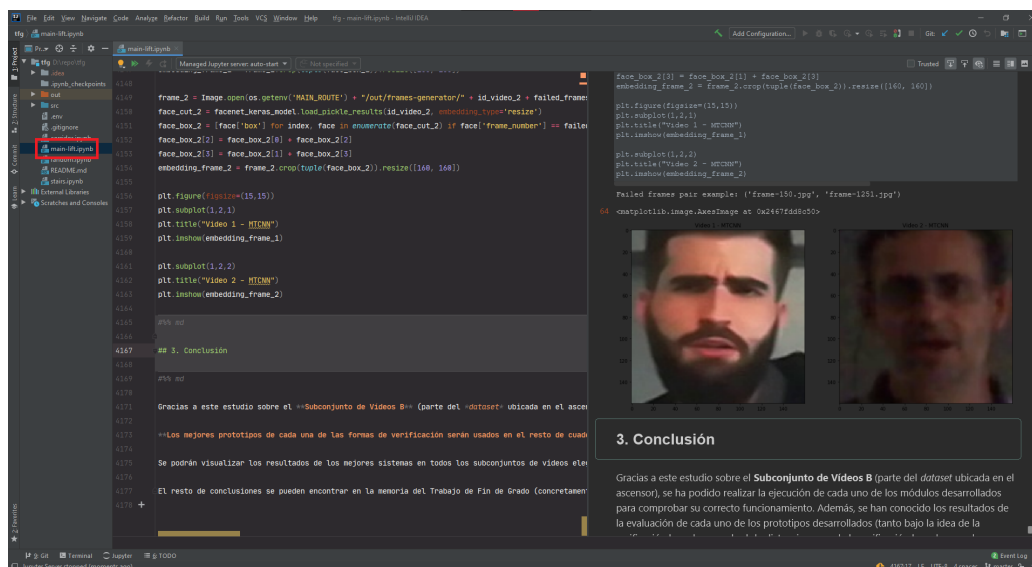


Figura 2.3: Apertura de cuaderno dentro del IDE

Para a continuación arrancar el servidor *Jupyter*, como se muestra en la figura 2.4:

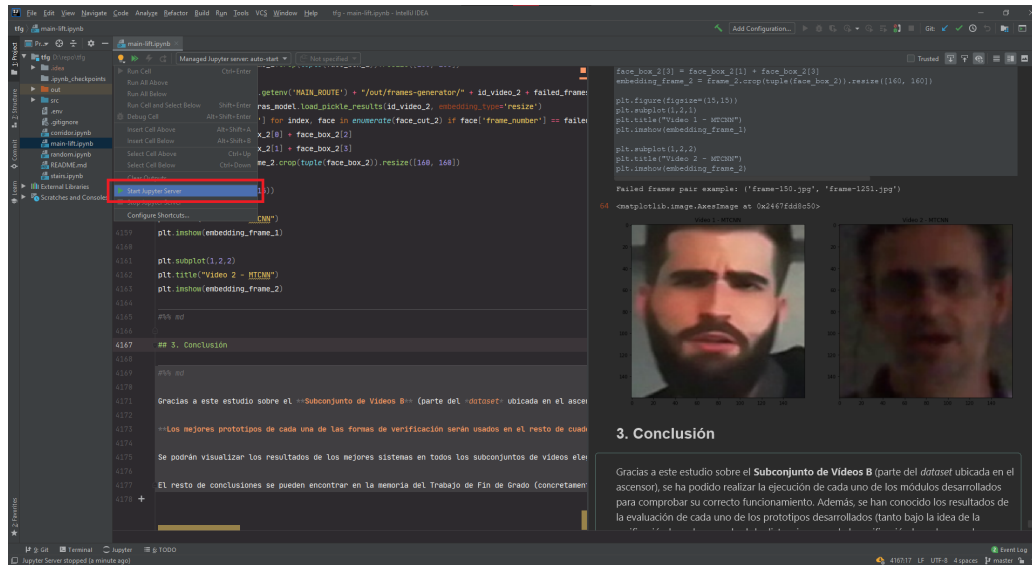


Figura 2.4: Arranque del servidor *Jupyter*

Y por último acceder al enlace para el acceso al contenido del cuaderno a través del servidor, tal y como se puede observar en la figura 2.4:

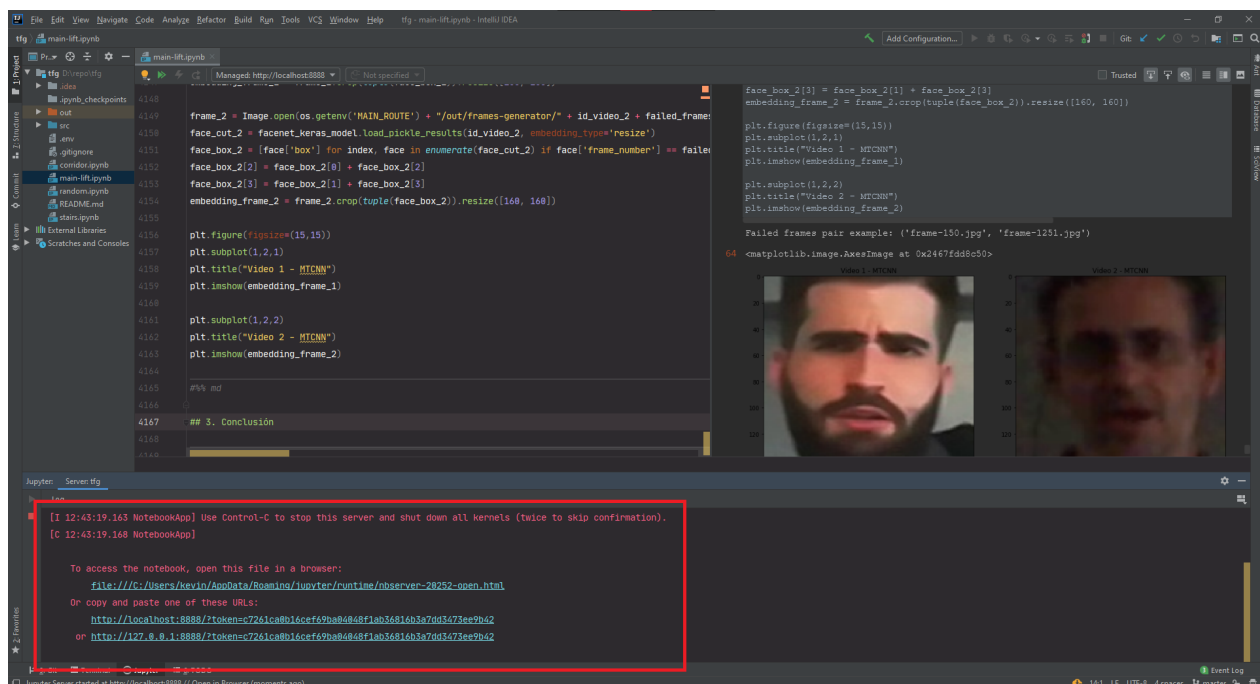


Figura 2.5: Enlaces para acceder al cuaderno *Jupyter*

Teniendo un directorio de ficheros (alojado por defecto en la dirección <http://localhost:8888/tree>) donde poder acceder a los distintos cuadernos desarrollados a lo largo del proyecto como es posible analizar en la figura 2.6:

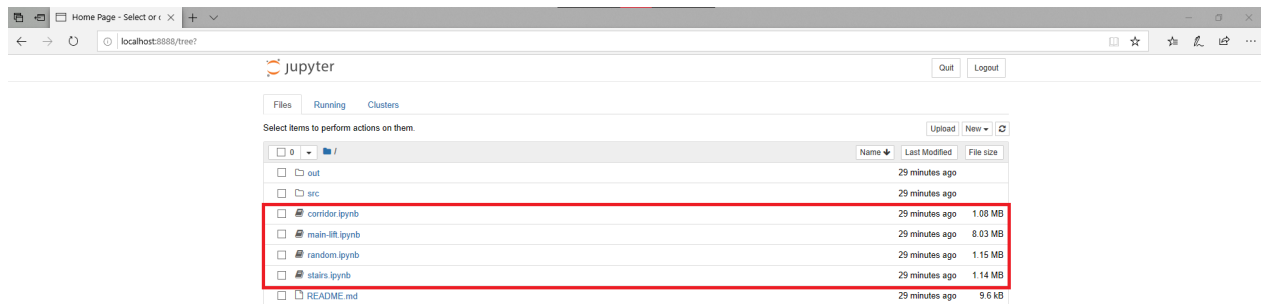


Figura 2.6: Acceso a los cuadernos del repositorio desde un servidor *Jupyter*

Realizando *click* en cualquiera de ellos, se entrará al entorno que permitirá la ejecución de las distintas celdas del cuaderno, además de la visualización que ya permitía *GitHub* en la sección anterior, como se muestra en la figura 2.7.

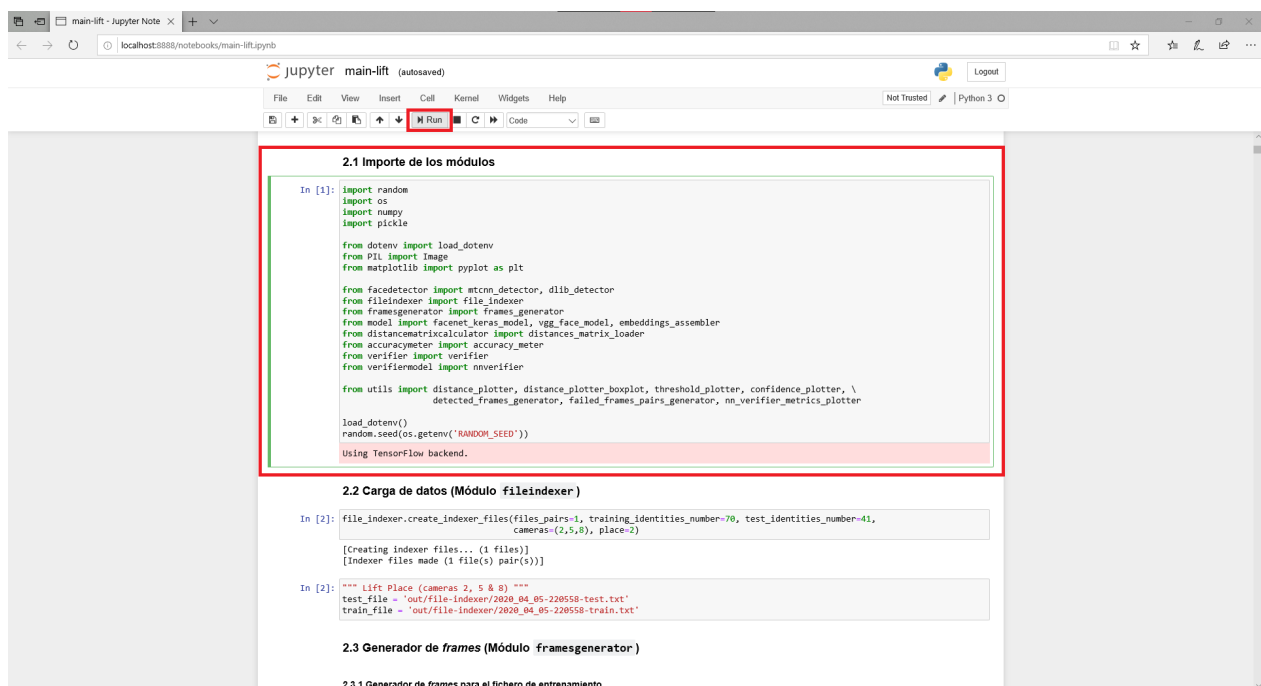


Figura 2.7: Ejecución de celdas desde un servidor *Jupyter*

Cabe destacar que la visualización del documento mejora cuando se utiliza el propio servidor de *Jupyter*, haciendo la lectura más cómoda para el usuario que está haciendo uso de dicho documento. En el caso de *IntelliJ IDEA Ultimate* o *PyCharm*, el documento puede ser visualizado y ejecutado además desde el mismo IDE también, como se muestra en la figura 2.3.

2.3 Entornos de *Jupyter*

La última manera que se propone para visualizar y ejecutar cuadernos *Jupyter* es hacer uso de su propio entorno sin ningún IDE.

Con esta finalidad, se puede hacer uso del comando de consola `pip` [11] para la instalación de *Jupyter*:

```
pip install notebook
```

Para a continuación ejecutar en la misma consola desde el directorio del proyecto lo siguiente:

```
jupyter notebook
```

Con lo que se obtendría una vista muy similar a la figura 2.6 bajo la misma dirección por defecto que se ejecutaba con *IntelliJ IDEA Ultimate*. De esta manera, también es posible visualizar y ejecutar cada una de las celdas disponibles en los cuatro cuadernos desarrollados.

Sin embargo, se advierte de que a la hora de importar los distintos módulos del sistema, *IntelliJ IDEA Ultimate* y *PyCharm* ofrecen una gestión de la configuración de los módulos del proyecto que permite indicar el SDK y cuál es la carpeta de ficheros fuente o *source*.

Dicha gestión permite una correcta conexión entre el cuaderno *Jupyter* y los módulos del sistema de verificación. Es por ello que si se quisiera usar un entorno de *Jupyter* sin usar estas herramientas, se debería modificar el importe de los distintos módulos. Por lo tanto, se desaconseja esta opción si se quieren ejecutar las celdas (para la visualización de estas no existe ningún problema).

3 Requisitos para la ejecución

3.1 Configuración del fichero de variables de entorno

Se debe configurar el fichero `.env` presente en el repositorio con los datos del equipo que esté ejecutando el proyecto. Concretamente, las variables que se deben modificar son las siguientes:

- `VIDEOS_ROUTE`: ruta de los vídeos del conjunto de datos *AveRobot* [2]. Para la obtención del conjunto de datos *AveRobot* [2], se debe realizar el procedimiento indicado al final de su página oficial¹. Se recuerda que la distribución de *AveRobot* debe quedar dividida en tres carpetas, correspondiendo cada una con un piso distinto (cada carpeta tiene el siguiente nombre: `‘averobot_floor_0X’`, siendo `X` el número del piso (1, 2 o 3)). Se puede encontrar más información en el capítulo 5 de la Memoria del Trabajo de Fin de Grado.
- `MAIN_ROUTE`: ruta del proyecto en el equipo.

Se debe destacar que, en caso de ejecutarse sobre Windows, las rutas deben usar igualmente la barra `‘/’` en lugar de la barra `‘\’` que usa dicho sistema operativo.

3.2 Bibliotecas

A lo largo de la Memoria del Trabajo de Fin de Grado se nombraron todas y cada una de las bibliotecas que se usaron a lo largo del proyecto (se pueden observar las principales bibliotecas usadas en la sección 3.1.2 de la Memoria del Trabajo de Fin de Grado). En caso de requerir alguna de instalación, se aconseja el uso de la herramienta mencionada en la sección 2.3 `pip` [11].

Todas las bibliotecas se usaron en su última versión bajo la versión de Python 3.7.1, como se mencionó en dicha memoria. Se recomienda tener tanto `cuDNN` [12] como `CUDA` [13] instalado correctamente.

¹Página oficial de *AveRobot* con instrucciones para el acceso al conjunto de datos: <http://mozart.dis.ulpgc.es/averobot/>

3.3 Descarga de modelos

Existe una serie de modelos que se deben descargar desde sus respectivos alojamientos si se quiere hacer uso de ellos. En concreto, se tienen los siguientes:

- `shape_predictor_68_face_landmarks.dat.bz2`²: permite la extracción de elementos faciales en la detección de caras por parte de los detectores de DLIB (ver sección 6.2.2.4 de la Memoria del Trabajo de Fin de Grado). Se debe descomprimir e introducir en la carpeta ‘src/facedetector’.
- `mmod_human_face_detector.dat.bz2`³: permite la inicialización del detector DLIB - MMOD (ver sección 4.2.2 de la Memoria del Trabajo de Fin de Grado). Se debe descomprimir e introducir en la carpeta ‘src/facedetector’.
- `facenet_keras.h5`⁴: permite el uso del modelo FaceNet como generador de descriptores (ver sección 4.3 de la Memoria del Trabajo de Fin de Grado). Se debe introducir en la carpeta ‘src/model’.

3.4 Celdas ejecutables sin *AveRobot*

En caso de no tener acceso a *AveRobot*, se proporciona una serie de descriptores ya generados bajo unos elementos (detectores, normalizaciones, localizaciones, generadores de descriptores...) descritos en su propia nomenclatura en la carpeta ‘out/embeddings-assembler’ dentro de un *DataFrame* de Pandas (ver sección 6.4 de la Memoria del Trabajo de Fin de Grado).

Es por ello que, en caso de no poseer el conjunto de datos *AveRobot*, se abre la posibilidad de usar dichos *DataFrames* en determinadas secciones de los cuadernos para observar cómo se comporta cada uno de los sistemas. Además, se adjuntan todos los modelos de redes neuronales generados en la verificación haciendo uso de dicha técnica en la carpeta ‘out/verifier-model’ y la nomenclatura de los vídeos pertenecientes al conjunto de entrenamiento y test de cada uno de los 4 subconjuntos de vídeos usados en el proyecto en la carpeta ‘out/file-indexer’.

Concretamente, en caso de **no** tener acceso a *AveRobot*, **no** se podrán ejecutar las secciones de los cuadernos enumeradas como: 2.2, 2.3.1, 2.3.2, 2.4.1, 2.4.2, 2.4.3, 2.5.1, 2.5.2, 2.5.3, 2.6.X.9, 2.6.X.10, 2.7.X.9 y 2.7.X.10. El resto de secciones deberían ejecutarse con los ficheros de salida que se adjuntan en la carpeta ‘out/’ del repositorio.

²Enlace a la descarga de `shape_predictor_68_face_landmarks.dat.bz2`: <https://github.com/davisking/dlib-models>

³Enlace a la descarga de `mmod_human_face_detector.dat.bz2`: <https://github.com/davisking/dlib-models>

⁴Enlace a la descarga de `facenet_keras.h5`: https://drive.google.com/drive/folders/12aMYASGCKvDdkygSv1yQq8ns03AStDO_

Bibliografía

- [1] Project Jupyter. *Página oficial de Project Jupyter*. URL: <https://jupyter.org/>. (accedido: 24.05.2020).
- [2] Mirko Marras. y col. «AveRobot: An Audio-visual Dataset for People Re-identification and Verification in Human-Robot Interaction». En: *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM, INSTICC*. SciTePress, 2019, págs. 255-265. ISBN: 978-989-758-351-3. DOI: 10.5220/0007690902550265.
- [3] Matt Cone. *Getting Started: An overview of Markdown, how it works, and what you can do with it*. URL: <https://www.markdownguide.org/getting-started/>. (accedido: 08.07.2020).
- [4] Kevin Rosales. *Repositorio del código programado para el TFG*. URL: <https://github.com/kevinrosalesdev/tfg>. (accedido: 28.06.2020).
- [5] Microsoft. *GitHub*. URL: <https://github.com>. (accedido: 22.05.2020).
- [6] JetBrains. *IntelliJ IDEA*. URL: <https://www.jetbrains.com/es-es/idea/>. (accedido: 24.05.2020).
- [7] JetBrains. *IntelliJ IDEA: Diferencias entre versiones*. URL: https://www.jetbrains.com/es-es/idea/features/editions_comparison_matrix.html. (accedido: 24.05.2020).
- [8] JetBrains. *Jupyter notebook support (IntelliJ IDEA Ultimate)*. URL: <https://www.jetbrains.com/help/idea/jupyter-notebook-support.html>. (accedido: 08.07.2020).
- [9] JetBrains. *Jupyter notebook support (PyCharm)*. URL: <https://www.jetbrains.com/help/pycharm/jupyter-notebook-support.html>. (accedido: 08.07.2020).
- [10] JetBrains. *Página oficial de JetBrains*. URL: <https://www.jetbrains.com/>. (accedido: 24.05.2020).
- [11] PSF. *Installing Packages*. URL: <https://packaging.python.org/tutorials/installing-packages/>. (accedido: 08.07.2020).
- [12] NVIDIA Corporation. *Manual de instalación de cuDNN*. URL: <https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html#overview>. (accedido: 08.07.2020).
- [13] NVIDIA Corporation. *Manual de instalación de CUDA*. URL: <https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html>. (accedido: 08.07.2020).