Pose Detection

Marc Picazos Carrillo

Resum—Resum del projecte, màxim 10 línies. ....
....
....
....
....
....
....
....
....
....

Paraules clau—Paraules clau del projecte, màxim 2 línies. ....
....

Abstract—Versió en anglès del resum. ....
....
....
....
....
....
....
....
....
....

Index Terms—Versió en anglès de les paraules clau. ....
....

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

Alexa, ¿qué tiempo hace hoy? , Ok google, activar alarma a las 8:00 son frases que se utilizan más en el día a día de las personas y esto no sería posible sin el Machine learning. En la vida cotidiana se utilizan multitud de aplicaciones que aplican Machine Learning para ofrecer una mejor experiencia o resultados más similares a los que podría dar una persona, y es que en los últimos años la evolución del Machine Learning ha permitido que campos como procesamiento de lenguaje natural o computer vision creen aplicaciones cada vez más complejas y potentes.

La siguiente aplicación es muy utilizada por los estudiantes, cuando se está realizando un trabajo y se necesita traducir un texto se acude a traductores de texto como el Google Translate [1] o Deepl [2].

- *E-mail de contacte: marc.picazos@e-campus.uab.cat*
- *Menció realitzada: Enginyeria de Computació*
- *Treball tutoritzat per: Coen Antens (CVC)*
- *Curs 2020/21*

Estos traductores han ido mejorando mucho en los últimos años gracias al Machine Learning, pasando de ofrecer una traducción directa a una respuesta más estructurada según el contenido del texto, esto es posible a que la inteligencia artificial que utilizan ya no sólo traduce el contenido del texto, sino que además les permite comprender el texto para ofrecer una mejor traducción. Una tecnología muy emergente y que sin la evolución de la visión artificial sería posible es la conducción autónoma, un ejemplo es el Autopilot de Tesla [3]. Mediante un sistema de cámaras en el automóvil permiten captar imágenes de todo su alrededor que una vez procesadas, le permite a la inteligencia artificial obtener datos como señales de tráfico, detectar las líneas de la calle, coches u obstáculos que se encuentran a su alrededor y a la distancia que se encuentran para poder tomar decisiones de forma autónoma y conducir el coche con seguridad.

La visión artificial o visión por computador no ha sido siempre tan potente como lo es ahora, en los inicios de siglo veintiuno se realizaban concursos, en los que se retaba a los investigadores a leer bases de datos de imágenes y clasificarlas según su contenido. Una de las bases de datos más conocidas con la que se realizaban estos retos y que sirve a los investigadores para comprobar la

eficacia de sus proyectos es ImageNet [4]. Y es que en el 2012 se realiza un gran avance en este concurso, la utilización de redes neuronales convolucionales. Alex Krizhevsky juntos con sus compañeros presenta un sistema de redes convolucionales conocido por AlexNet [5] que les permite clasificar en 1000 clases diferentes los 1,2 millones de imágenes de alta resolución de la base de datos de Image-Net con un porcentaje de error de 17%, mejorando así mucho el estado del arte actual en cuanto a la clasificación de imágenes. Esto da el pistoletazo de salida para el uso de las redes neuronales convolucionales en la visión por computador atrayendo las miradas de grandes empresas, que aprovechando los recursos que tienen en sus manos han llevado estos sistemas a otro nivel en el que prácticamente obtienen un 99% de error al clasificar imágenes.

Dentro de la visión por computador encontramos que una de sus utilidades más destacables es la pose detection, cómo es el caso de la aplicación creada por amazon para mantener la distancia de seguridad entre sus empleados debido a la COVID-19 [6]. En esta aplicación mediante la detección de las personas en las imágenes obtenidas por las cámaras y la creación de los puntos de unión de las articulaciones de la persona, es posible calcular la distancia de seguridad de cada persona y, en el caso que no se cumpla dicha distancia, se comunica con un elemento visual conforme no se está manteniendo la distancia de seguridad necesaria.

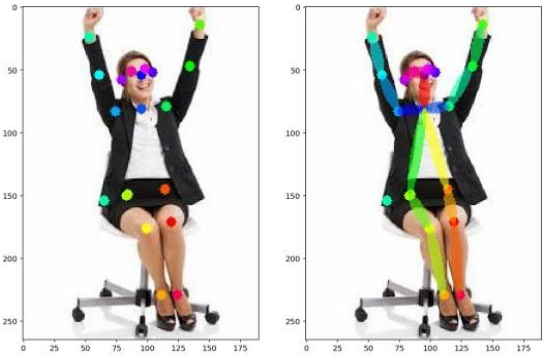

Figure 1 - Detección de puntos clave con pose detection

En la robótica es habitual el uso de pose detection para el uso de tareas colaborativas humano-robot, como es el caso del paper Formulating Intuitive Stack-of-Tasks with Visuo-Tactile Perception for Collaborative Human-Robot Fine Manipulation [7], en el que se utilizan cámaras para trackear a las personas cercanas y mediante human pose estimation obtener los keypoints de la persona. De esta manera el robot puede interactuar de manera segura con la persona al manipular objetos porque puede procesar dónde están los límites y en caso necesario no interactuar para proteger al empleado.

En el artículo Using Kinect™ sensor in observational methods for assessing postures at work aplican la tecnología de pose detection a la Ergonomía en el trabajo [8]. En este proyecto se presenta una tecnología capaz de detectar las malas postura de una persona durante su jornada laboral en oficina mediante cámaras Kinect con el objetivo de mejorar la ergonomía y reducir las lesiones causadas por la mala postura.

La utilización de la visión por computador en el deporte está permitiendo añadir valor al contenido consumido por las grandes audiencias. En el caso proyecto de Leonardo Citraro (2020) Real-Time Camera Pose Estimation for Sports Fields [9], mediante el uso de cámaras permite convertir la escena 2d a 3d detectando en ella a los jugadores del campo de juego, esto puede permitir obtener métricas de los jugadores de forma automática y ofrecer al espectador un valor añadido. Otra utilidad en el deporte es el ejercicio guiado como el que propone Hou (2017) en su proyecto Dancing like a superstar: Action guidance based on pose estimation and conditional pose alignment [10], en el utilizan el posicionamiento de los keypoints de la persona, obtenidos con pose estimation gracias a cámaras Kinect, para crear una plataforma en la que ayuda a mejorar los pasos de baile de manera interactiva.

Estos ejemplos son solo algunos de los avances en el campo de la visión por computador que están permitiendo automatizar procesos. De esta forma, se facilita el trabajo a las personas y a su vez se adapta a los diferentes entornos.
En este trabajo profundizaremos sobre el uso de human pose estimation dentro del ámbito del deporte para analizar qué sistemas se adaptan mejor a los problemas que plantea el deporte para estos sistemas. Con el fin de crear un dataset en un entorno real con el sistema de pose estimation qué mejor rendimiento muestre en la primera fase del proyecto y finalmente aplicar el dataset obtenido para crear un sistema de detección de anomalías que nos visualize cuando se está realizando un movimiento no esperado o mejorable técnicamente durante la realización de una disciplina deportiva.

2 OBJETIVOS

En este apartado se tratarán los objetivos del proyecto con el fin de tener una visión general hasta donde se quiere llegar con el trabajo de fin de grado. Para ello se dividirán los objetivos en apartados para poder explicar con más detalle cada uno de ellos y mostrando en el orden que deberán ser realizados, ya que el cumplimiento de un objetivo dependerá de su sucesor.

2.1 Analizar pose estimation en el deporte

El deporte es uno de los ámbitos en los que hoy en día ya se están utilizando técnicas de pose estimation, algunas de sus aplicaciones actuales son obtener estadísticas en tiempo real de los jugadores en un partido de fútbol, ayudar en el entrenamiento de los deportistas o recrear jugadas en tres dimensiones. Debido a las múltiples apli-

caciones y los diferentes contextos en los que se realiza el deporte, nuestro principal objetivo será encontrar las librerías de pose estimation que mejor se adaptan al ámbito Deportivo

2.3 Dataset en entorno real

Una vez hemos analizado qué sistemas de pose estimation se adaptan mejor al deporte, se aplicará uno de ellos a un entorno real con el objetivo de obtener todos los datos posibles de la realización de una práctica deportiva.

Dentro de este objetivo, marcamos como objetivo opcional la realización de un pequeño evento deportivo para la recopilación de datos de forma masiva en un mismo evento. Debido a la pandemia que se está sufriendo es posible que este objetivo no se pueda llevar a cabo, es por eso por lo que se indica como objetivo opcional.

2.4 Detección de anomalías

Como objetivo final y objetivo opcional después de crear un dataset con datos reales de la práctica de una disciplina deportiva, se aplicará un algoritmo para detectar posibles fallos durante la práctica de esta, mostrando los fotogramas en los que ha sucedido. De esta manera podremos completar el círculo de analizar los sistemas de pose estimation, escoger el que mejor se ajusta al deporte, llevarlo a la práctica creando un dataset y finalmente utilizar el dataset creado para observar fallos en la técnica.

3 Metodología

La metodología utilizada para el desarrollo del trabajo de fin de grado será Kanban [11] con algunos principios de metodologías ágiles. Esta elección viene debida a la experiencia obtenida durante el grado en diferentes asignaturas en las que hemos practicado con estas metodologías, donde el objetivo es organizar y gestionar el trabajo de equipos de trabajo. En este caso nos encontramos con un proyecto personal y algunas de estas metodologías necesitan de varios integrantes para un buen funcionamiento. Es por esto por lo que utilizaremos Kanban para gestionar y organizarnos, de tal manera que mediante un tablero se pueda ver el estado del proyecto. Para implementar Kanban se utilizará la plataforma Monday, que nos permite crear un tablero con nuestras tareas.

Además, se implementará el principio de trabajo iterativo, en el que mediante sprints de 1 o 2 semanas se irá evolucionando el proyecto y nos permitirá organizar el trabajo necesario para cada fase del proyecto. Al finalizar cada iteración se realizará una reunión con el tutor para verificar el estado del trabajo y planificar próximos pasos de los siguientes sprints.

Como modo de comunicación se utilizará Microsoft Teams para mantener reuniones con el tutor y nos permite crear un canal de comunicación a modo Sandbox en el que poder almacenar documentos de interés. De la misma manera se utilizará un canal a modo de Porfolio para tener un registro del progreso del trabajo realizado durante toda la evolución de este.

4 Planning

En este apartado se detallarán las cargas de trabajo y las tareas a realizar del proyecto para los próximos meses. Para ello se tendrá en cuenta el calendario que disponemos para realizar el proyecto y los informes a realizar durante su transcurso.

Si dividimos todos los meses del calendario escolar para realizar el proyecto en semana, obtenemos un total de 22 semanas para trabajar en él.

Las semanas importantes que se tendrán en cuenta para añadir a la planificación por entregas de un informe o punto de control son las siguientes:

- Semana 4 (08/03 – 14/03): Informe inicial
- Semana 10 (19/04 – 25/04): Informe progreso 1
- Semana (24/05 – 30/05): Informe progreso 2
- Semana 18 (14/06 – 20/06): Propuesta informe final
- Semana 19 (21/06–27/06): Propuesta presentación final
- Semana 20 (28/06 – 04/07): Entrega informe final
- Semana 22 (12/07 – 18/07): Defensa y presentación del proyecto

Según estas semanas y las entregas de control que se tienen que realizar, tenemos aproximadamente 13-14 semanas para trabajar en el core del proyecto, ya que las últimas 5 semanas están destinadas a pulir el informe final y preparar la presentación del proyecto y las primeras 5 semanas son para organizar y plantear el kickoff del proyecto.

Conociendo las semanas que tenemos para trabajar en el proyecto y las fechas de entrega de informes o puntos de control necesarios para realizar el proyecto, pasamos a detallar en las tareas de cada fase del proyecto.

4.1 Tareas del Proyecto

El proyecto tiene tres fases muy diferenciadas que son inicio del proyecto, analizar pose estimation en el deporte, crear un dataset en un entorno real, detección de anomalías y cierre del proyecto. A continuación, se describen las tareas que contiene cada fase del proyecto.

- Inicio del proyecto
  - Primera reunión con el tutor.
  - Concretar objetivos del proyecto.
  - Recopilar información necesaria para el proyecto.
  - Informe Inicial.
- Analizar pose estimation en el deporte
  - Escoger 3 sistemas de pose estimation.
  - Analizar sistema pose estimation Alphapose.
  - Analizar sistema pose estimation Den-

sePose.
    - Analizar sistema pose estimation OpenPose.
    - Análisis de una actividad con OpenPose.
    - Detección de patrones con DTW.
    - Tracking de personas en video.
    - Human Activity recognition con DeepLearning.
- Crear un dataset en un entorno real
    - Escoger sistema pose estimation y crear script para preparar dataset.
    - Reunión para organizar control presencial en el que recopilar datos (Opcional según medidas de seguridad del momento).
    - Entrega Informe 1.
    - Realizar control presencial (Opcional según medidas de seguridad del momento).
    - Crear dataset.
    - Analizar errores en el dataset para extraer datos no válidos.
- Detección de anomalías.
    - Desarrollo script para la detección de anomalías.
    - Entrega Informe 2.
    - Test script con datos reales.
- Cierre del proyecto.
    - Entrega del informe final.
    - Finalizar y testear cualquier etapa anterior.
    - Preparar defensa del proyecto.
    - Crear presentación del proyecto.
    - Entrega y finalización del TFG.

4.2 Diagrama de Gantt

A continuación, en este apartado se muestra cómo se han planificado las fases del proyecto, teniendo en cuenta el volumen de tareas por fase y el calendario escolar. Para ello se muestra gráficamente el resultado de la planificación mediante un diagrama de Gantt.

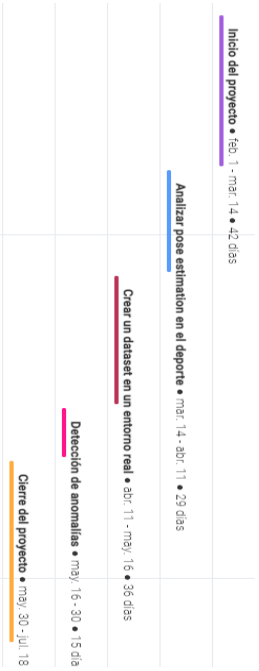

Figure 2 - Diagrama de Gantt del proyecto

5 DESARROLLO

La sección de Desarrollo permite explicar las fases del Proyecto de una manera más detallada. En cada una de ellas podremos ver el trabajo que se ha realizado para cumplir con los objetivos marcados en la planificación.

5.1 Extracción de keypoints

En la primera fase del Proyecto se han utilizado varios sistemas de pose estimation como son OpenPose [14], Detectrone DensePose 2 [15] y AlphaPose [16] para obtener los keypoints de las personas que aparecen en los frames de los videos. Para realizar esta tarea se ha utilizado Google Colab, un producto de Google Research que nos permite escribir y ejecutar código desde el navegador utilizando los recursos computacionales proporcionados por parte de Google de forma gratuita y sin necesidad de una configuración previa. Grácias a estas características encontramos multitud de archivos colbortivos en el entorno del machine learning que nos permiten tener una primera visión del funcionamiento y resultados sin necesidad de pasar por el proceso de instalación en una máquina propia.

En el caso de los sistemas de pose estimation seleccionados, los desarrolladores facilitan un archivo de Google Colab en el que ya contiene las librerías y requisitos necesarios para utilizar su sistema de pose estimation con un conjunto de ejemplos básicos para obtener una primera interacción son los sistemas. Las pruebas se han enfocado en extraer y mostrar los keypoints de un video, como el mostrado en el ejemplo de la figura 3.

Figure 3 – Ejemplo pose estimation aplicado con OpenPose [14]

Todos los sistemas de pose estimation testeados devuelven una vector de valores con la posición x, y del keypoint en el fotograma, además algunos añaden la precisión del punto para tener una referencia de cuan fiable es la posición obtenida. La posición del conjunto de valores x, y e precisión hace referencia a un punto del esqueleto formado por los keypoints, para reconocer a qué parte del esqueleto hace referencia hay que consultar la documentación de cada Sistema de pose estimation. En el ejemplo mostrado se utiliza OpenPose con el formato BODY_25 que representa el esqueleto de la persona con 25 keypoints (Figure 4).

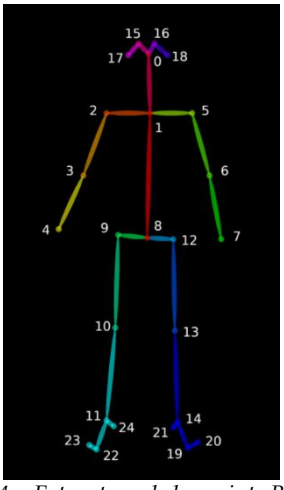

Figure 4 – Estructura de keypoints BODY_25

5.2 Tracking y detección de personas

Un problema observado durante las pruebas de los sistemas de pose estimation es la necesidad de relacionar los conjuntos de keypoints de un frame con el frame anterior, para ello se ha implementado un sistema de detección y tracking de personas para trackear las personas durante todo el video con el fin de relacionar cada conjunto de keypoints con la persona pertinente.

El sistema de multitracking implementado utiliza como base las librerías de Dlib para trackear y hacer la correlación entre detecciones, OpenCV [17] para la gestión y tratamiento de frames del video y crear una red neuronal con el modelo MobileNetSSD para la detección de persona.

El funcionamiento está basado en 3 fases, la primera de ellas obtención de un frame del video, en segundo lugar, detección de personas en el frame, en caso de no tener personas trackeadas se crea un tracker y en caso de existir se actualiza el tracker para situar la posición obtenida en el frame actual. Finlmente se indica en el frame la persona detectada con un cuadro para poder visualizar la detección y tracking, como en el ejemplo de la figura 5.

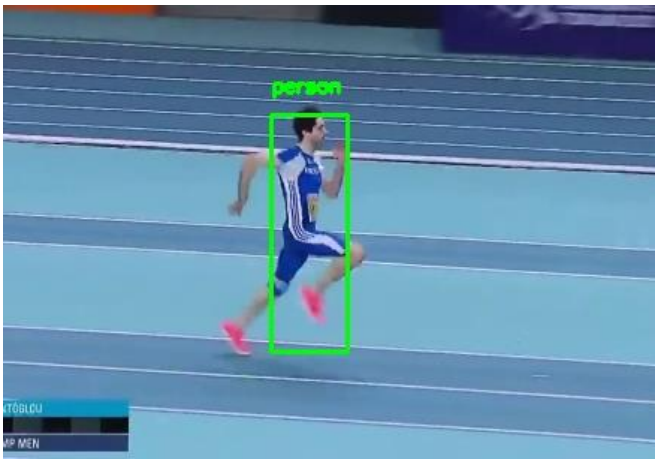

Figure 5 – Detección y tracking de persona

5.3 Reconocimiento de actividad

Una parte esencial del Proyecto es poder reconocer que actividad se está realizando en cada momento para poder extraer del video una parte en concreto y poder analizarla más específicamente según la actividad que se realice o para clasificar la actividad de las personas que aparecen en el fotograma y poder eliminar aquellos keypoints que pueden crear ruido ya que son keypoints de personas que no necesitamos analizar.

Para realizar la prueba de la sección de reconocimiento de actividad se ha utilizado un subset del dataset Berkeley MHAD [18], el cual nos proporciona una serie de videos clasificados según la actividad que se realiza en el video, pudiendo ser una de las siguientes actividades saltos, saltos de tijera, boxeo, mover una mano, mover las dos manos o aplaudir. Mediante el Sistema de pose estimation OpenPose se han extraído los keypoints de los videos, para obtener los keypoints de cada fotograma de los videos con su correspondiente clasificación.

Una vez se han tratado los datos y se han preparado en datos de entrenamiento y de testing se ha entrenado una red neuronal RNN-LSTM. Las redes LSTM son un tipo de red neuronal recurrente que permite traspasar información de una conexión a otra, esto las hace muy valiosas para aprender de secuencias de datos como por ejemplo en tareas de speech recognition [19]. Aplicado al caso que se esta estudiando, este tipo de red neuronal permite tener en valor la secuencia de datos de cada movimiento para clasificar la actividad.

Los resultados que se han obtenido con el modelo de clasificación de actividad entrenado han sido satisfactorios obteniendo un accuracy superior al 90%, clasificando de manera satisfactoria entre un 80-95% de las clases que contiene el dataset, como se puede observar en la matriz de confusión de la figura 6.

Con este ejemplo aplicado se puede confirmar que mediante un input de pose estimation, se puede detectar la actividad que se está realizando en la imagen. Y por lo

tanto con un dataset especializado en un deporte se puede detectar las fases o técnicas que están aplicando en todo momento los deportistas.

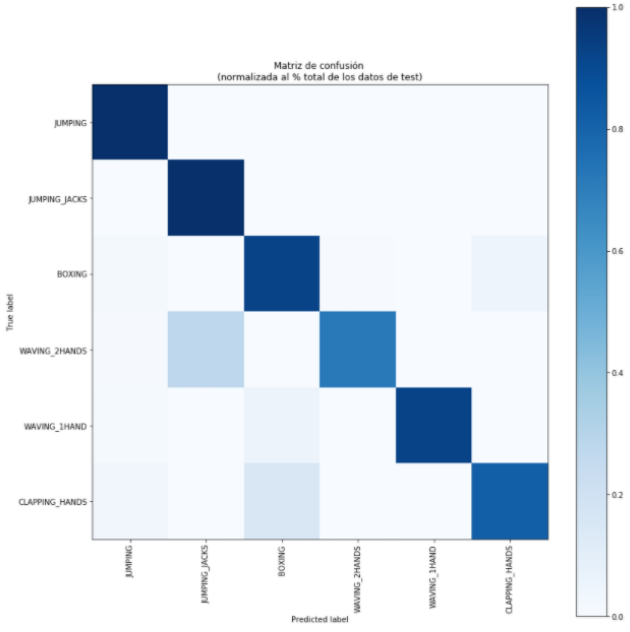

Figure 6 – Matriz de confusión reconocimiento de actividad

5.4 Detección de patrones

Con el fin de encontrar patrones y relaciones entre dos secuencias de una actividad se ha utilizado el algoritmo Dynamic Time Warping o DTW. DTW es un algoritmo que permite medir la similaridad entre dos secuencias temporales pudiendo variar en la velocidad. Speech recognition es un ejemplo aplicado de DTW, en el que mediante las ondas de sonido producidas al hablar permite encontrar patrones de similitud independientemente de la velocidad de pronunciación o accento (Figure 7), como se puede observar en el artículo de 1994 de Donald J. Berndt and James Clifford, Using Dynamic Time Warping to Find Patterns in Time Series [20].

Aplicando el algoritmo de DTW al problema del proyecto permite comparar una misma actividad realizada por dos personas. Pudiendo comparar por ejemplo una actividad realizada por un profesional con una de realizada por un amateur y detectar las fases de la actividad que tienen en común y las fases que se pueden mejorar.

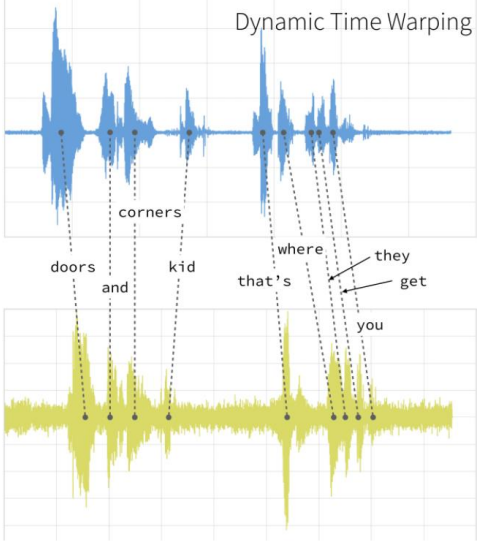

Figure 7 –Ejemplo Dynamic Time Warping para la frase "Doors and corners kid that's where they get you"

En una primera fase se han preparado los datos utilizando OpenPose sobre dos videos grabados lateralmente de un salto de longitud, de cada fotograma se ha extraído los keypoints de toda la pierna izquierda de cada saltador, cadera, rodilla y tobillo. Con estos datos se calcula el ángulo de la pierna obteniendo un array con los ángulos de la pierna izquierda del saltador durante todo el video.

En la segunda fase se ha aplicado el algoritmo DTW utilizando la librería fastdtw [21], la cual necesita recibir las dos cadenas de datos a comparar para devolver un valor que es la distancia mínima y una cadena de tuplas que indican la relación de un fotograma del primer video con un frame del segundo video.

Esta sección de la primera fase de desarrollo del Proyecto todavía se encuentra por finalizar y no nos permite sacar conclusiones de ella, pero sí que gracias a ella se han podido detectar algunas necesidades como el de utilizar un sistema de tracking y detección de personas para relacionar los keypoints obtenidos con las personas en el video con el fin de evitar ruido.

BIBLIOGRAFIA

[1] AutoML translation [Online] Disponible: https://cloud.google.com/translate
[2] Deepl [Online]: https://www.deepl.com
[3] Tesla,Autopilot and Full Self-Driving Capability [Online]. Disponible: https://www.tesla.com/support/autopilot
[4] Image-net [Online]. Disponible: http://image-net.org
[5] Alex Krizhevsky, Ilya Sutskever,Geoffrey E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks [Online]. Disponible :

https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
[6] Brad Porter (2020). Amazon introduces 'Distance Assistant' [Online]. Disponible: https://www.aboutamazon.com/news/operations/amazon-introduces-distance-assistant
[7] Sunny Katyara, Fanny Ficuciello, Tao Teng, Fei Chen, Bruno Siciliano, Darwin G. Caldwell (2021). Formulating Intuitive Stack-of-Tasks with Visuo-Tactile Perception for Collaborative Human-Robot Fine Manipulation [Online] Disponible: https://arxiv.org/abs/2103.05676
[8] Jose Antonio Diego-Mas, Jorge Alcaide-Mazal (2013). Using Kinect sensor in observational methods for assessing postures at work [Online]. Disponible: https://doi.org/10.1016/j.apergo.2013.12.001
[9] Leonardo Citraro, Pablo Márquez-Neila, Stefano Savarè, Vivek Jayaram, Charles Dubout, Félix Renaut, Andrés Hasfura, Horesh Ben Shitrit, and Pascal Fua (2020). Real-Time Camera Pose Estimation for Sports Fields [Online]. Disponible: https://arxiv.org/pdf/2003.14109.pdf
[10] Hou, Y., Yao, H., Li, H., & Sun, X. (2017). Dancing like a superstar: Action guidance based on pose estimation and conditional pose alignment. [Online] Disponible: https://ieeexplore.ieee.org/document/8296494
[11] Henrik Kniberg & Mattias Skarin(2010). Kanban y Scrum – obteniendo lo mejor de ambos [Online]. Disponible: https://www.academia.edu/download/38261265/KanbanVsScrum_Castellano_FINAL-printed.pdf
[12] MPII Human Pose Dataset [Online]. Disponible: http://human-pose.mpi-inf.mpg.de
[13] Monday [Online]. Disponible: http://monday.com
[14] Documentación OpenPose[Online]: https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/
[15] Documentación Detectron 2 DensePose [Online]: https://detectron2.readthedocs.io
[16] Documentación AlphaPose [Online]: https://www.mvig.org/research/alphapose.html
[17] OpenCv [Online]: https://opencv.org
[18] Berkeley MHAD dataset [Online]: https://tele-immersion.citris-uc.org/berkeley_mhad
[19] Douglas Coimbra de Andrade (2018). Recognizing Speech Commands Using Recurrent Neural Networks with Attention [Online] Disponible: https://towardsdatascience.com/recognizing-speech-commands-using-recurrent-neural-networks-with-attention-c2b2ba17c837
[20] Donald J. Berndt and James Clifford (1994). Using Dynamic Time Warping to Find Patterns in Time Series [Online] Disponible: https://www.aaai.org/Library/Workshops/1994/ws94-03-031.php
[21] FastDtw[Online]: https://github.com/slaypni/fastdtw
[22] Adrian Rosebrock (2018). Multi-object tracking with dlib [Online] Disponible: https://www.pyimagesearch.com/2018/10/29/multi-object-tracking-with-dlib/
[23] Ricardo Portilla, Brenner Heintz and Denny Lee (2019). Understanding Dynamic Time Warping [Online] Disponible: https://databricks.com/blog/2019/04/30/understanding-dynamic-time-warping.html