
This is the **published version** of the bachelor thesis:

Zhang, Rongrong; Carrabina Bordoll, Jordi, dir. Aplicación de técnicas IA a sensorización FLIR para camas robotizadas para evitar úlceras. 2023. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/280722>

under the terms of the  license

This is the **published version** of the bachelor thesis:

Zhang, Rongrong; Carrabina Bordoll, Jordi, dir. Aplicación de técnicas IA a sensorización FLIR para camas robotizadas para evitar úlceras. 2023. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/280722>

under the terms of the  license

Aplicación de técnicas IA a sensorización FLIR para camas robotizadas para evitar úlceras

Rongrong Zhang

Resumen—En el ámbito médico, el tratamiento de las úlceras por presión siempre ha sido de vital importancia para los pacientes que padecen enfermedades que los mantienen encamados durante periodos prolongados. Este proyecto en colaboración con la empresa Control-Live SL, se enfoca en el desarrollo de un sistema de identificación de los puntos de presión utilizando el aprendizaje profundo, aprovechando una base de datos pública de imágenes llamada SLP (*Simultaneously-collected multimodal Lying Pose*). Este proyecto utiliza la red neuronal convolucional HRNet para construir el esqueleto y determinar los puntos clave de las articulaciones del cuerpo humano. Estos puntos son fundamentales para identificar y localizar los puntos de presión del paciente.

Palabras claves—Úlceras por presión (UPP), estimación de pose humana (HPE), aprendizaje profundo, HRNet, puntos claves de articulaciones, esqueletización, HRpose, puntos de presión, base de datos SLP

Resum— En l'àrea mèdica, el tractament de les úlceres per pressió sempre ha estat de vital importància per als pacients que pateixen malalties que els tenen enllitats durant períodes llargs de temps. Aquest projecte, en col·laboració amb l'empresa Control-Live SL, s'enfoca al desenvolupament d'un sistema d'identificació dels punts de pressió utilitzant l'aprenentatge profund, aprofitant una base de dades pública d'imatges anomenada SLP (*Simultaneously-collected multimodal Lying Pose*). Aquest projecte utilitza la xarxa neuronal convolucional HRNet per construir l'esquelet i determinar els punts clau de les articulacions del cos humà. Aquests punts són fonamentals per identificar i localitzar els punts de pressió del pacient.

Paraules claus—Úlceres per pressió (UPP), estimació de posició humana (HPE), aprenentatge profund, HRNet, punts claus d'articulacions, esqueletització, HRpose, punts de pressió, base de dades SLP

Abstract—In the medical domain, the treatment of pressure ulcers has always been of vital importance for patients suffering from illnesses that let them lying in bed for longer periods of time. This project, partnered with Control-Live SL company, focuses on the development of a pressure point identification system using deep learning. We use images stored in the public database called SLP (*Simultaneously-collected multimodal Lying Pose*). This project uses the convolutional neuronal network HRNet to build the skeleton and determine the key joint points of the human body. These points are essential to identify and locate the patient's pressure points.

Index Terms—Pressure ulcers (PUs), human pose estimation (HPE), deep learning, HRNet, joint hotspots, skeletonization, HRpose, pressure points, dataset SLP

1 INTRODUCCIÓN - CONTEXTO DEL TRABAJO

UNA de las enfermedades que sufren las personas mayores son las úlceras por presión (UPP) o “úlceras de decúbito”. Este tipo de úlceras suelen ser padecidas por personas que se encuentran en una postura fija durante un periodo de tiempo prolongado, tal que algunas zonas del cuerpo se mantienen presionado y bloquean la circulación de la sangre. En consecuencia, se

generan las úlceras por presión y estas zonas empiezan a necrosarse [1], [2], [3].

La situación más habitual en la que se producen es en personas enfermas que deben estar en una cama o las que utilizan sillas de ruedas (por su movilidad altamente limitada). En la Fig. 1, se muestran las zonas más comunes del cuerpo en las que desarrollan las úlceras por presión: las caderas, las nalgas, los talones, los codos, los hombros, las orejas y la parte posterior de la cabeza. Llamaremos *pin-points* a estos puntos. La afectación a cada uno de los puntos de presión del paciente cambia en función de su posición de reposo. Por ejemplo, cuando esté sentado las

- E-mail de contacto: rongrong.zhang@autonoma.cat
- Mención realizada: Ingeniería de Computadores
- Trabajo tutorizado por: Jordi Carrabina Bordoll y Marc Codina Barberà (Dpt. de Microelectrònica i de Sistemes Electrònics)
- Curso 2022/23

zonas de riesgo localizadas en los glúteos, piernas y pies se verán más afectadas, dado que son las que reciben mayor presión.

Monitorizar la posición y su duración en los pacientes encamados es una parte importante para el cuidado y tratamiento de las UPP en el ámbito de medicina y enfermería clínica. Una forma eficiente de prevenir las úlceras por presión es mediante el movimiento de la posición del cuerpo con frecuencia dependiendo de la afectación de cada una ellas, lo que permite disminuir la presión en los puntos de riesgo de ulceración. En la actualidad, existen superficies hinchables que permiten prevenir, reducir o aliviar este efecto de manera global y continua [4], [5].

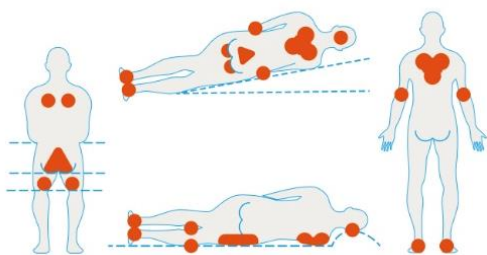


Fig. 1. Los puntos de presión comunes del cuerpo [3]

La empresa Control-Live S.L. ha desarrollado una cama modular con movimiento longitudinal y transversal (Fig. 2), V-Chair [6], que permite automatizar la posición y duración de cada estado de manera que se puede adaptar al tratamiento óptimo de las UPPs. La llamada *Smart Therapeutical Surface (S.T. Surface)*, y tiene la funcionalidad de convertirse en una cama o una silla de ruedas e incluso permite realizar movimientos laterales del paciente. Con la V-Chair, se puede mover fácilmente el paciente de una postura a otra, basta con hacer el movimiento de V longitudinal hacia la izquierda y derecha, proporciona un nivel alto de comodidad y soporte al usuario.

V-Chair se ha concebido especialmente para los pacientes encamados de alta dependencia o movilidad limitada que tienen que estar bajo control [25]. Todos los movimientos o los comandos que modifican la posición de la V-Chair están controlados por el usuario o el/la cuidador/a mediante un panel de control táctil o por comandos hablados mediante un sistema de reconocimiento de voz.

En el ámbito de un proyecto de investigación con el Centre de Prototips i Solucions HW/SW (CEPHIS) de la UAB, la empresa está desarrollando el sistema electromecánico de movimiento de la cama que permita un control en lazo cerrado. Utilizando este sistema, se propone diseñar e implementar un prototipo que sea capaz de controlar el movimiento de la V-Chair de forma automatizada (sin necesidad de intervención personal) en función del nivel de riesgo de ulceración personalizable a las necesidades y morfología de cada paciente.

Para ello, se sensorizará la V-Chair para capturar la imagen del paciente encamado y su duración. Se estimarán los *pin-points* mediante técnicas de estimación de poses human (HPE), que permiten la comprobación del riesgo de ulceración en cada *pin-point*. Si supera un umbral especificado, se moverá la V-Chair a una nueva posición para la que se deberá evaluar que el riesgo en ese *pin-point* ha disminuido suficientemente (o desaparecido).



Fig. 2. El movimiento longitudinal y transversal del V-Chair

Este proceso implica diferentes tareas: (1) sensorizar la V-Chair para capturar la posición del paciente sobre la V-Chair mediante diferentes sensores (visible, infrarrojos, 3D ToF, superficie de presión); (2) identificar mediante un algoritmo la posición del esqueleto del paciente (y sus *pin-points*) sobre la V-Chair; (3) evaluar la presión en cada *pin-point* sobre un modelo virtual de la V-Chair; (4) determinar que *pin-points* superan el umbral de riesgo predefinido por el personal clínico; (5) mover la V-Chair a una nueva posición y finalmente (6) evaluar que en la nueva posición el riesgo de ulceración se ha reducido por debajo del umbral.

Tanto la posición de los *pin-points* en función de la morfología del paciente, como su nivel de gravedad y los umbrales de riesgo de ulceración correspondientes se establecerán mediante el conocimiento y las métricas obtenidas en la literatura médica, de manera personalizable para cada paciente, por parte de personal clínico experto: fisioterapeutas, enfermeras, médicos, etc.

Este proyecto se centra en la parte de construcción de esqueleto del paciente e identificación de los *pin-points* mediante aprendizaje profundo. Se aplicará la estimación de pose humana a una imagen, obteniendo los puntos claves de articulación y basado en ello, con un algoritmo se determinarán los *pin-points*.

2 OBJETIVOS

El objetivo principal de este proyecto es aplicar un algoritmo de IA sobre una base de datos pública etiquetada de posturas de personas sobre la cama, para obtener el esqueleto de un cuerpo humano. Sobre los puntos de referencia del esqueleto se mapearán los *pin-points* para las diferentes posiciones del paciente.

Se prevé disponer de diferentes tipos de sensores que

permitan la estimación de la posición del paciente con diferentes condiciones de iluminación (día-noche) y bajo situaciones donde el paciente pueda estar cubierta o no por una manta y de la manera menos invasiva (más anónima) posible. Entre ellos se consideran: sensores de infrarrojos de longitud de onda larga (LWIR), sensores de profundidad 3D ToF (*time-of-flight*), sensores radar, superficies de presión (integrados con sensores), juntamente con sensores CMOS de imagen visible para construir el *ground-truth*.

En primera instancia se considera el uso de la cámara CMOS visible para realizar la implementación del algoritmo. En la versión final, se substituirá por el sensor que de mejores resultados y que mantenga la privacidad personal y la protección de datos.

Se plantea alcanzar el objetivo mediante el desarrollo de diferentes tareas específicas:

- Análisis de *dataset* público SLP [7].
- Definición de la metodología y selección de recursos a utilizar del SLP.
- Aplicación de algoritmo de construcción del esqueleto del cuerpo del paciente a partir de una imagen o una serie de imágenes (*dataset* SLP) mediante técnicas de inteligencia artificial.
- Anotación de los *pin-points* sobre el esqueleto con representación gráfica y/o textual.

Más adelante, se diseñará un programa que permitirá a los especialistas clínicos introducir los umbrales de cada *pin-point* de cada paciente mediante una interfaz gráfica. Esta información permitirá calcular la duración máxima de cada postura en función de la presión en cada *pin-point*, y enviará la orden para realizar el movimiento de la V-Chair. Por tanto, el cambio de posición de la V-Chair se decidirá de manera personalizada para cada paciente para minimizar la posibilidad de ulceración según los criterios de los especialistas clínicos. Una vez realizado, se repetirá la evaluación de la posición del paciente y estimación de presión de sus *pin-points* para asegurar que los cambios de presión han sido efectivos. Si no han cambiado por debajo del criterio establecido, se enviará un aviso.

Esta metodología será integrada con una base de datos propia que recoja las posiciones de la V-Chair que no están contempladas en la base de datos SLP, así como el mismo procedimiento de esqueletización, identificación de *pin-points* y la gestión de los movimientos de la V-Chair en función de los umbrales.

3 ESTADO DEL ARTE

3.1 Estimación de posición humana HPE

La estimación de posición humana (HPE acrónimo de *human pose estimation*) se ha abordado en el ámbito de la Inteligencia Artificial mediante técnicas de visión por

computador para detectar las articulaciones clave del cuerpo a partir de imágenes o videos y mostrar una postura o un grafo de la estructura del cuerpo humano [26], [27]. Tiene una amplia aplicación en multitud de campos como por ejemplo en películas, animación, realidad virtual, interacción hombre-máquina, atención médica, etc. [28].

En la literatura científica, existen técnicas de HPE para la estimación de una única persona o de múltiples personas en la misma imagen. En la mayoría de las aplicaciones en entornos reales, la estimación para una única persona no es suficiente. Sin embargo, en este trabajo estamos interesados en la estimación de la posición de la única persona que se encuentra en reposo en una cama.

Los modelos de HPE encontrados para una única persona han sido entrenados mediante bases de datos de imágenes y videos públicos como MPII, COCO, HumanEva, Human3.6M y SURREAL (*Synthetic hUmans foR REAL tasks*) que se pueden utilizar para explorar y construir el modelo [29]. Hemos seleccionado la base de datos SLP (*Simultaneously-collected multimodal Lying Pose*) [7] debido a que contiene posturas del cuerpo humano en reposo en una cama, lo que coincide directamente con el objetivo de este trabajo. SLP ha sido construido por un grupo de investigadores de la Universidad Northeastern de Boston. Contiene imágenes de las posturas capturadas de 109 participantes con múltiples modalidades de sensores incluyendo RGB, LWIR (*Long Wavelength Infrared*), profundidad (*MS Kinect*) y mapa de presión superficial. Las imágenes se han obtenido en condiciones diversas, incluyendo personas cubiertas por una manta o una sábana [30].

Los modelos HPE suelen estar basados en redes neuronales convolucionales que realizar el aprendizaje automático a partir de los datos de las BBDD.

3.2 Red Neuronal Convolucional (CNN)

Antes de adentrarnos en las redes neuronales convolucionales, introduciremos las redes neuronales o redes neuronales artificiales (ANN). Se trata de modelos de aprendizaje y procesamiento automático que emulan el funcionamiento del cerebro humano. Estos métodos surgieron en los años 1950. Sin embargo, no se popularizaron debido a la baja potencia de las computadoras existentes y a la escasez de algoritmos de aprendizaje eficientes. Actualmente, los principales algoritmos son: el perceptrón múltiples capas (MLP), las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN) [8].

El algoritmo MLP es un algoritmo desarrollado por Frank Rosenblatt en el año 1958. Está compuesto por una capa de entrada, una/unas capas ocultas y una capa de salida [8].

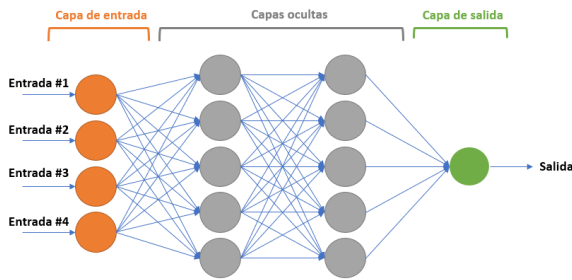


Fig. 3. Estructura de Multicapas Perceptrón (Red Neuronal Tradicional)

Estos algoritmos, aplicados a procesamiento de imágenes, tiene una deficiencia grave: la pérdida de información espacial. Es decir, las relaciones entre píxeles próximos. No será capaz de distinguir el mismo objeto en diferentes zonas de la imagen.

Las redes neuronales convolucionales solucionan este problema. Esto se debe a que permite aprender y extraer las características relevantes de las imágenes [9] con la aplicación de la operación convolución.

Las redes neuronales recurrentes (RNN) se orientan al tratamiento de información temporal o secuencial y no se utilizan en nuestra investigación.

3.2.1 Arquitectura de CNN

La arquitectura de las CNN está constituida por tres diferentes tipos de capas: convolución (*convolucional*), submuestreo/de agrupación (*pooling*) y completamente conectada (*fully connected*) [10].

Capa de convolución: es el bloque inicial de la CNN. Consiste en crear una ventana de usualmente 3x3 o 5x5 píxeles que realiza la convolución con la imagen de entrada. Esta ventana se llama filtro o kernel de la CNN. Este filtro se aplica a toda la imagen píxel a píxel. Usualmente, tanto cuando los datos proceden de una base de datos como cuando proceder de un sensor de imágenes, estos se introducen en la CNN mediante un barrido (esto es, primero toda la fila 1, luego la fila 2, etc.), desde la parte superior hasta la parte posterior. Después de la aplicación del filtro obtendremos una imagen de menor dimensión (en caso de no aplicar *padding* que es el proceso de añadir filas o columnas para no cambiar el tamaño de la imagen resultante). Posteriormente, cada capa suele estar compuesta de un banco de filtros donde cada filtro extrae una característica específica.

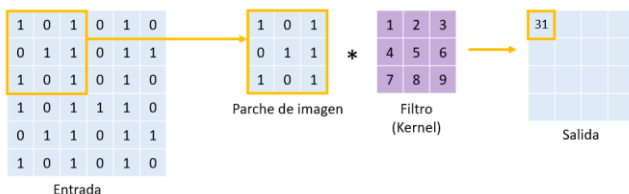


Fig. 4. Operación convolución de una imagen

Capa de submuestreo: después de pasar por el banco de filtros, se aplica una función *pooling*, que puede ser *max-pooling*, promediado, parametrizados, etc. En la mayoría de los casos, se le aplica el *max-pooling* para reducir la dimensión de la imagen y manteniendo a su vez las características importantes. También se conoce como "*downsampling*".

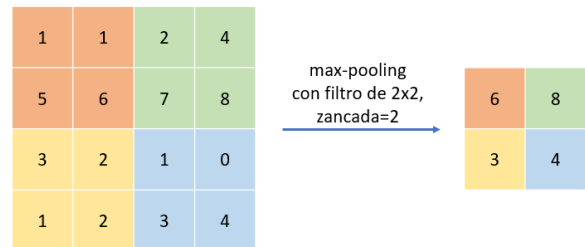


Fig. 5. Operación max-pooling de una imagen

Capas completamente conectadas: las características extraídas de la imagen de la capa de submuestreo (los mapas de características resultantes), se conectan a una red completamente conectada de tipo MLP (habitualmente) para llegar a la capa de salida.

Las capas de convolución y submuestreo forman un bloque básico de una CNN que se encargan de extraer las características relevantes. Antes de entrar en la capa completamente conectada, podemos tener múltiples bloques básicos, tantos como sean necesarios para el modelo que ha sido implementando [11], [12].

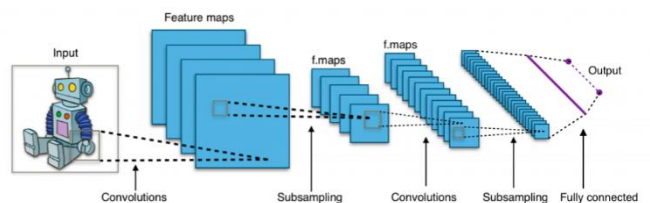


Fig. 6. Arquitectura típica de una Red neuronal convolucional [13]

Como resumen, una red neuronal convolucional es un tipo de red neuronal artificial profunda compuesta por bloques básicos extraen y reducen la dimensión del mapa de características y una capa de clasificación que permite la identificación para la cual se ha diseñado la red.

3.3 HRNet

HRNet es acrónimo de "*High-Resolution Network*", una red neuronal propuesta por Microsoft Research Asia en 2019. Es un algoritmo de aprendizaje profundo (*Deep learning*) basado en la arquitectura CNN, introduciendo elementos arquitectónicos nuevos para procesar imágenes de alta resolución.

Las redes tradicionales utilizadas para reconstrucción o mejora de imágenes (no las de clasificación) tienen una estructura de *downsample* y *up-sample* para tratar las

imágenes. Con *downsample* reducen la dimensión (resolución) de la imagen hasta una dimensión adecuada y después con *up-sample* recupera la resolución original, mediante un proceso secuencial. En cambio, la red HRNet, mantiene las representaciones de alta resolución durante todo el proceso de cada imagen.

3.31. Arquitectura de HRNet

La arquitectura de HRNet se basa en una estructura en cascada de múltiples ramas paralelas. Cada rama tiene una resolución diferente que aporta una información distinta de las demás. Existen dos tipos de conexiones en HRNet, las conexiones residuales y las fusiones de características [14].

- Conexiones residuales: se utilizan dentro de cada rama para mejorar el flujo de información y evitar problemas de desvanecimiento del gradiente.
- Fusiones de características: se encargan de fusionar las características extraídas de diferentes ramas de resolución, combinando informaciones entre ellos.

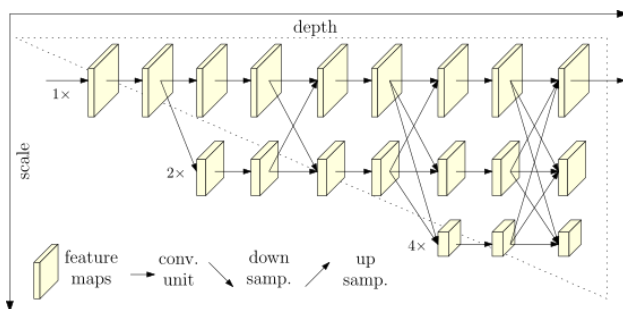


Fig. 7. Arquitectura HRNet [14]

En cada rama, los mapas de características se conectan mediante conexiones residuales mientras que ramas diferentes se conectan con fusiones de características. De esta forma se llega a obtener más información que la que es capaz de extraer una sola rama.

4 METODOLOGÍA

Para el desarrollo del proyecto se plantea a utilizar el método *Scrum* que, aunque se ha desarrollado para la gestión de proyectos de equipos grandes, también se adapta para organizar los proyectos individuales.

El proyecto se divide en diferentes *Sprints* y cada sprint contiene un subconjunto de las tareas y subtareas a realizar. Cada *Sprint* tiene una duración de entre 1 a 4 semanas. Cada tarea o subtarea tiene asociada un tiempo de estimación y un estado del trabajo. Se planificarán las tareas a procesar con el diagrama de Gantt para visualizar el progreso del proyecto y controlar de forma más detallada cada paso.

El Software que se utiliza para la gestión es la

herramienta Jira de la empresa Atlassian. Se aplica la plantilla de *Scrum* que proporciona y define los *sprints* con los objetivos a alcanzar. En la finalización de cada Sprint se hará una revisión de seguimiento de las tareas y se decidirá el avance del proyecto. Con *sprint planning* replanificará las tareas retrasadas y se definirán los trabajos imprevisibles.

La planificación inicial del proyecto se divide en seis *sprints*:

- **Sprint I.** Búsqueda de recursos (2 semanas): Esta etapa se centra en búsqueda de información y artículos que se relacionen con el tema. Se explorarán los bases de datos que pueden servir para encontrar las articulaciones clave, los algoritmos existentes y los entornos de desarrollo a aplicar.
- **Sprint II.** *Framework* IA (3 semanas): Se trata de seleccionar el *framework* de desarrollo de algoritmos IA y dentro de este estudiar las implementaciones disponibles. En particular se profundizará en las librerías que permiten la conversión de las imágenes a la resolución de los algoritmos AI.
- **Sprint III.** Pruebas con Algoritmos IA (4 semanas): Se hace la implementación de los algoritmos existentes, sobre la BBDD SLP para seleccionar el algoritmo con mejores resultados.
- **Sprint IV.** Construcción esqueleto (3 semanas): Con el algoritmo se identificarán las articulaciones clave y se construirá el esqueleto del cuerpo.
- **Sprint V.** Estimación de los *pin-points* (2 semanas): Basándose en el esqueleto obtenido, se estimarán los *pin-points* del cuerpo.
- **Sprint VI.** Documentación (1 semana): Documentar el informe del proyecto y los resultados obtenidos.

El proyecto utiliza en el data-set SLP para conseguir el objetivo principal. Las imágenes que se utilizan son imágenes RGBs e imágenes infrarrojas de baja resolución.

El algoritmo de estimación de pose (puntos de articulación del esqueleto y *pin-points*) se implementará mediante redes neuronales convolucionales (CNN).

5 ENTORNO DEL DESARROLLO Y COMPONENTES

En esta sección se explica el entorno de trabajo y los componentes usados para este proyecto.

5.1 Entorno de desarrollo

Se desarrolla el proyecto utilizando el entorno de desarrollo integrado Visual Studio Code [15], un editor desarrollado por Microsoft que proporciona un amplio ecosistema de extensiones para desarrollo de múltiples lenguajes de programación. Es ágil para compilar, depurar y ejecutar los códigos.

5.2 Herramientas y librerías

Para el desarrollo del proyecto se ha utilizado el lenguaje Python, juntamente con Jupyter Notebook y las librerías de tratamiento de imágenes y de visión por computador. Estos nos permiten mostrar fácilmente los resultados mediante imágenes y gráficas. A continuación, se listan todas las herramientas y librerías utilizadas que se considera relevantes:

- Jupyter Notebook [16]: software libre fácil de manejar, integrado con un servicio web (una interfaz flexible) para computar, visualizar y codificar. Interactúa con VS Code mediante la instalación de la extensión en el entorno.
- Python [18] 3.11.3: lenguaje interpretado y de código abierto que permite transportar fácilmente el programa a otras plataformas. Es simple y se utiliza mucho en la programación de aprendizaje automático.
- PyTorch [19] 2.0.0+cu117: *framework* de código abierto de aprendizaje automático. Proporciona una completa funcionalidad para construir modelos y entrenar redes neuronales profundas incluso para CNNs.
- Dataset SLP [7]: base de datos pública, contiene 14.715 imágenes de diferentes tipos y modalidades [17]. Las imágenes RGB e IR se han utilizado para el desarrollo del proyecto.
- Modelo HRPose del SLP [20]: modelo que se ha implementado a base del HRNet.
- GPUs NVIDIA GeForce RTX 2080 Ti [23]: tarjeta gráfica con 11GB de memoria diseñada por NVIDIA, especialmente dedicada para procesamiento de las imágenes.
- CUDA 11.7 [21]: plataforma con modelo de memoria compartida para la programación paralela de aplicaciones que permite acelerar la ejecución de los códigos fuente sobre las GPU de NVIDIA.
- CUDA Python [22]: proporciona envoltorios Cython/Python para CUDA y las API de tiempo de ejecución. Se puede instalar usando PIP y Conda. Los desarrolladores de Python aprovechan la computación en GPUs para lograr mayor precisión y resultados más rápidos.
- Librerías:
 - OpenCV [24] 4.7.0: *Open Computer Vision Library* es una librería de visión por computador desarrollada por Intel. Proporcionan una amplia gama de funciones para el procesamiento de imágenes y videos.
 - Scikit-image o Skimage 0.20.0: librería de procesamiento de imágenes de Python que ofrece

herramientas para el análisis de las imágenes.

- Numpy 1.24.2: librería fundamental en Python, para procesamiento numérico de datos científicos. Proporciona arrays, matrices, funciones matemáticas y estadísticas para cálculo numérico de forma eficiente y rápida.
- Matplotlib 3.7.1: librería de visualización de datos que proporciona herramientas y funcionalidades para crear gráficas, diagramas, etc.

6 DESARROLLO

El desarrollo del sistema se divide en dos fases para obtener la estimación de los *pin-points*. La primera fase consiste en aplicar un algoritmo de aprendizaje profundo, el modelo HRNet para la construcción del esqueleto del paciente, una vez determinado el esqueleto (puntos clave de articulación). Basado en ello dependiendo de la pose del paciente en particular se le aplica algoritmo para detección de los *pin-points*.

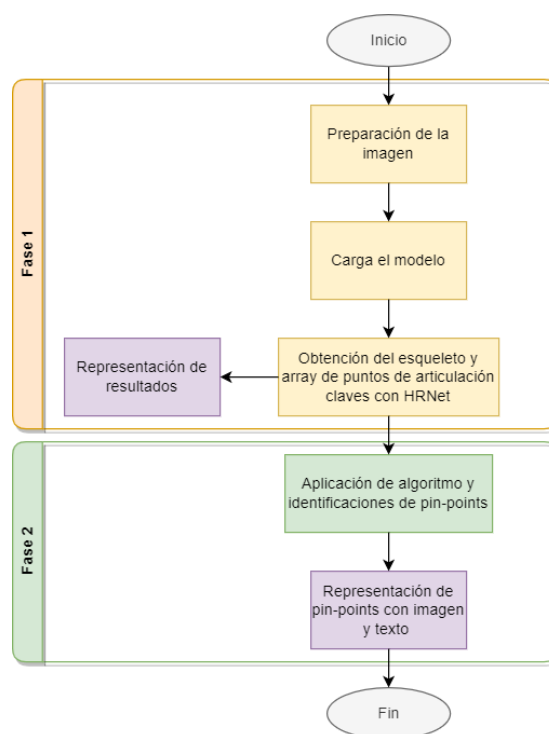


Fig. 8. Diagrama de flujo del sistema

6.1 Aplicación de HRNet preentrenado para creación del esqueleto del paciente

El diseño del modelo de CNN se basa en el modelo HRPose del SLP, (código abierto disponible en [20]) que entrenamos con el *data-set* SLP para imágenes RGBs e infrarrojas. Una vez obtenemos el modelo entrenado, se realiza la inferencia de la imagen construyendo el esqueleto y detectando las articulaciones claves. El programa sigue los siguientes pasos para llegar obteniendo la

predicción de esqueleto del paciente.

- 1) Tratamiento de la imagen de entrada, mediante la librería Scikit-image y OpenCV, consistente en la transformación de la imagen para alimentar el modelo.
- 2) Carga del modelo HRpose entrenado utilizando la clase de PyTorch.
- 3) Entrada las imágenes preparadas al modelo que nos permite obtener los 14 puntos claves de articulación del cuerpo humano a su salida. Estos puntos permiten reconstruir el esqueleto del paciente.
- 4) Recogida de la imagen de salida y superposición sobre la imagen de entrada para mostrar el resultado o guardarlo.

6.1.1 Preparación de la imagen

SLP contiene diferentes modalidades de imágenes con diferentes resoluciones. Se transforman las imágenes originales a una resolución de 256x256x3 para RGBs o 256x256x1 para otros casos.

Dependiendo de la resolución de la imagen se diseña una función para convertir la imagen a la resolución fijada para el modelo.

6.1.2 Carga el modelo HRPose del SLP

Una vez preparadas las imágenes, se carga el modelo HRPose. Este modelo está diseñado con PyTorch. Una vez entrenado el modelo se genera un fichero con extensión .pth. en el que se guardan los parámetros del modelo. Se carga la red con siguiente código:

```
model=HRpose.get_pose_net(in_ch=in_ch, out_ch=n_jt)
model.load_state_dict(torch.load(pthfilename))
model.eval()
```

La instrucción `model.eval()` inicia la inferencia que permite hacer la predicción de los puntos claves de articulación.

6.1.3 Obtención de resultado

A la salida del modelo, se obtiene un array de las articulaciones clave con las coordenadas x, y de la imagen. Estas se desnormalizan y dibujan con la librería cv2 para recuperar la resolución. Se representa el resultado de imagen RGB e IR como se observa en las Fig 9 y 10.

Los puntos de articulación tienen un promedio de 93,5% (Pckh@0.5) de *Accuracy*. Se considera un valor elevado en los modelos de HPE. Se procede a determinar los *pin-points* basándose en los puntos obtenidos.



Fig. 9. Representación del esqueleto sobre imagen RGB

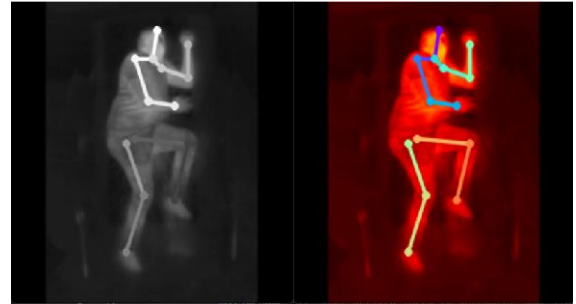


Fig. 10. Representación del esqueleto sobre de la imagen IR. Imagen izquierda es sobre imagen original, imagen de derecha con el filtro para visualizar mejor el resultado

6.2 Algoritmo para detección de *pin-points*

En esta sección se diseña un algoritmo que utiliza los puntos clave obtenidos del modelo para determinar los *pin-points* del paciente. Se divide el algoritmo en 3 pasos:

- 1) Predicción de la posición en la que está acostado el paciente: lado izquierdo, lado derecho, supina (descartamos la posición prona, ya que no se pondrá un paciente discapacitado de movimiento en esta posición).
- 2) Determinar la relación entre cada *pin-point* con las líneas creadas por los puntos clave de articulación dependiendo del caso y calcular la distancia entre algunos puntos con la línea. Se utilizará un umbral (*threshold*) para evitar la superposición en algunas zonas.
- 3) Identificar los *pin-points* combinando los pasos 1) y 2). Se calcularán nuevamente las coordenadas de los *pin-points* basándose en los puntos clave debido a que, con la diferente posición de reposo, los *pin-points* no son exactamente los puntos clave de articulaciones que ha encontrado el modelo.

6.2.1 Detección de posición del paciente

Para poder detectar la posición del paciente, se ha creado una línea del eje central del cuerpo humano mediante cuatro puntos claves detectados por HRPose: hombro izquierdo, hombro derecho, cadera izquierda y cadera derecha.

A continuación, se comparan las posiciones de los codos y las rodillas respecto el eje central. 1) Si la persona

está en posición lado derecho, la mayoría de los puntos tienen que estar en lado derecho del eje central. 2) En el caso de posición lado izquierdo, la mayoría de los puntos tienen que estar en lado izquierdo del eje central. 3) En el resto de los casos, podemos definir que es muy probable que el paciente está en la posición supina, no podemos dar una afirmación, ya que en algunos casos extremos no lo son.

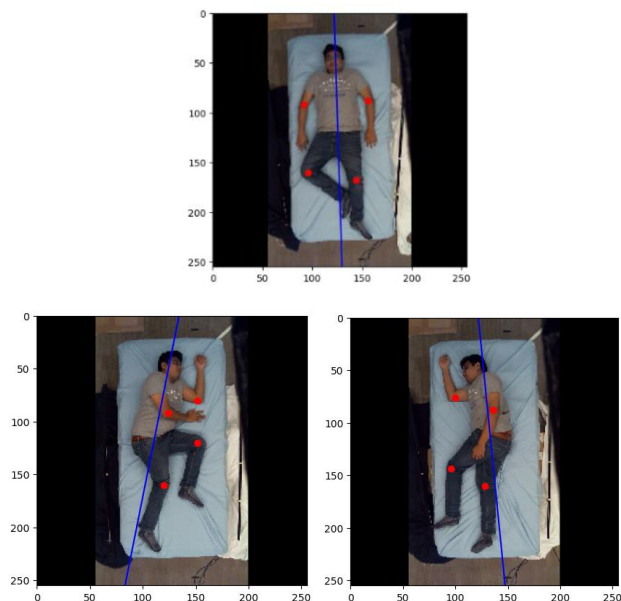


Fig. 11. Imagen original con eje central (la línea con color azul), los codos, rodillas pintadas (puntos rojos)

Se observan las relaciones de los puntos respecto al eje central. En las poses habituales se puede identificar la posición humana correctamente (sin considerar de los casos extremos).

6.2.2 Identificación de los *pin-points*

La identificación de los *pin-points* se hace separadamente. Cada punto se determina en función de su posición relativa a las líneas creadas y puntos clave del esqueleto.

Para facilitar el desarrollo, se divide el cuerpo por 3 zonas posibles: (1) del hombro hacia arriba; (2) de la cadera hasta el hombro y (3) de la cadera hacia abajo. En adelante las llamaremos *zona_cabezal*, *zona_cuerpo_superior* y *zona_cuerpo_inferior* respectivamente.

Es necesario verificar cuáles de los puntos clave de las articulaciones obtenidos en el paso anterior consideramos que están presionando la superficie del colchón, ya que estos son los que tienen posibilidades de generar UPPs. Se estiman los siguientes puntos por separado:

1) **Cabeza y tórax:** los *pin-points* de la cabeza pueden producirse en la oreja derecha, oreja izquierda u occipital. Para deducir cuál es el *pin-point* se necesita

saber la posición del paciente, calculada previamente. Existen los siguientes tres casos:

- Si el paciente está acostado al lado derecho, el *pin-point* es la oreja derecha.
- Si el paciente está acostado al lado izquierdo, el *pin-point* es la oreja izquierda.
- Si el paciente está en tendido supino, el *pin-point* es occipital.

En los tres casos, se calcula nuevamente la coordenada del *pin-point* de la misma forma, es el punto intermedio de la cabeza y tórax.

2) **Hombros y caderas:** en este caso se calcula el ancho de los hombros y de las caderas, poniendo un umbral (*threshold*) personalizable, ya que puede ser distinto para cada paciente. Se realizan los siguientes pasos:

- Si la anchura es mayor o igual que el *threshold*, se consideran *pin-points* los 4 puntos (hombro izquierdo y derecho, cadera izquierda y derecha).
- Si es menor, hay que verificar la posición del paciente:
 - Si está acostado lado izquierdo, los *pin-points* son el hombro y cadera izquierdos.
 - Para el derecho, el hombro y cadera derechos.

No se calculan nuevamente las coordenadas, ya que son las mismas del esqueleto obtenido.

3) **Codos y muñecas:** se aplican técnicas diferentes

Codos: Para verificar la presión en el codo, se crea una línea del hombro y la cadera y se identifica el codo en la posición de la línea. Para el codo izquierdo, se verifica con la línea creada por el hombro y cadera izquierdos.

- Si el codo está en la parte izquierda de la línea, se considera como el *pin-point*.
- Si está en la parte derecha de la línea, se debe analizar si el codo está en la zona cabezal y verificar que el codo no está sobre la cabeza o debajo. En este segundo caso se considera un *pin-point*.

Para el codo derecho se repite el proceso anterior, pero con la posición opuesta.

Muñecas: el proceso para verificar la presión en las muñecas es más complejo que el codo. Se comprueban las mismas condiciones que en el caso del codo, pero además se comprueba en qué zona está la mano.

4) **Rodillas y Tobillos:**

Rodillas: para verificar la presión en la rodilla se tiene que calcular la distancia entre las dos rodillas. Se siguen

los siguientes pasos:

- Verificar que la distancia entre las dos rodillas supera un cierto *umbral*. En este caso, consideramos que las rodillas están con presión.
- En el caso contrario (cuando las dos rodillas se solapan, una sobre otra), se mira la posición del paciente: (i) si es de lado izquierdo, la rodilla izquierda tiene la presión y (ii) si es de lado derecho, es la rodilla derecha que la tiene.

Tobillos: para verificar la presión en el tobillo, el primer paso consiste en calcular la distancia entre los dos tobillos, y las relaciones entre las líneas (línea creada por cadera y rodilla del mismo lado y línea creada por rodilla y torillo del lado contrario) y la distancia entre tobillo y rodilla. Con estos datos se comprueban que tengan las piernas cruzadas, dobladas, puesta uno por encima de otro, etc.

Por último, para marcar los *pin-points*:

- en el caso de la posición decúbito lateral son las mismas coordenadas obtenidas, mientras que
- en el caso de decúbito supino, se crea el *pin-point* mediante el cálculo del punto intermedio de rodilla y tobillo.

También se considera un caso específico, el caso en que las rodillas están dobladas hacia la orientación perpendicular de la superficie del colchón. Estos puntos no son identificados como *pin-points*.

6.2.3 Obtención de resultado

El último paso consiste en representar el resultado con la imagen y mostrar el texto por la pantalla, con el Jupyter Notebook fácilmente obtenemos los resultados. Los puntos marcados en color rojo de la imagen son los *pin-points* del paciente. También se muestra una representación textual junto con la imagen.

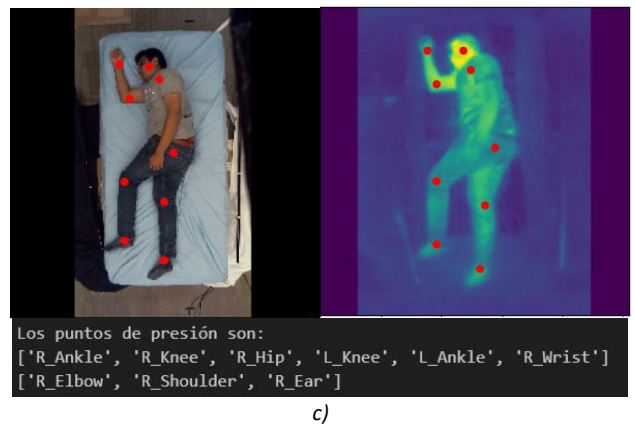
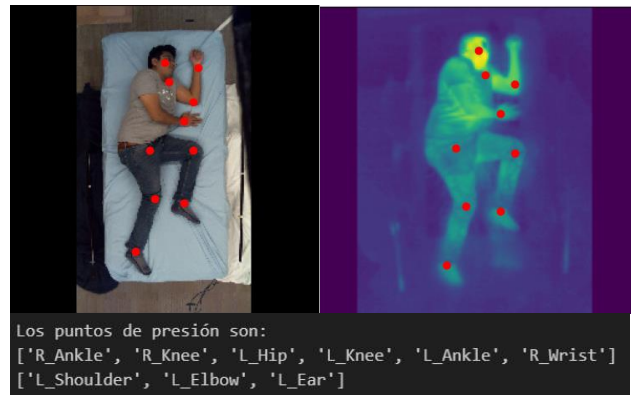
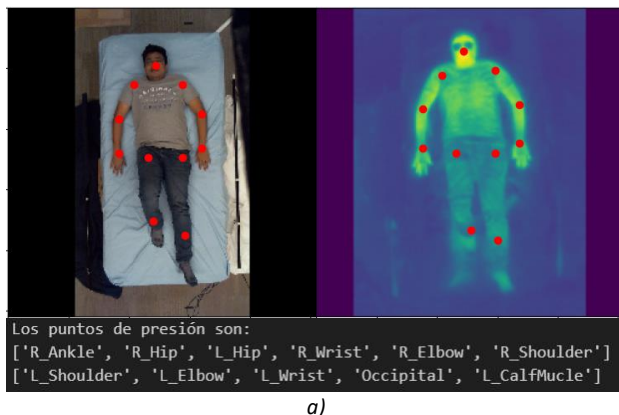


Fig. 12. Representación de la imagen original con *pin-points* marcados con puntos rojos y debajo con texto a) posición supina: imagen izquierda es la imagen RGB y derecha imagen IR, debajo de las imágenes con representación textual. b) posición lado izquierdo (observación por perspectiva personal) c) posición lado derecho

7 CONCLUSIONES

El programa implementado permite estimar los *pin-points* del paciente a partir de una imagen mediante el uso de técnicas de *Deep Learning*. La aplicación de un modelo de aprendizaje automático permite construir un esqueleto sobre el que se identifican los *pin-points* del paciente en función de su postura en la cama. Los resultados representados están validados por los expertos clínicos de la empresa Control Live S.L.

Como trabajo futuro, se debe extender este trabajo para el resto de las posturas de la V-Chair, es decir, las que no son la cama plana. Este trabajo se puede realizar realizando una transformación de la imagen capturada dado que se conoce la posición de todos los elementos mecánicos de la cama en cada caso. Una vez transformada la imagen se puede aplicar el algoritmo desarrollado en este trabajo.

Otra opción es la generación de una base de datos propia, capturada sobre el sistema V-Chair sobre cada uno de sus movimientos o posiciones.

AGRADECIMIENTOS

Quiero expresar mi más profundo agradecimiento a mis tutores Jordi Carrabina y Marc Codina por darme la oportunidad de realizar el proyecto. Especialmente, por las guías expertas y valiosas orientaciones. Me gustaría agradecer al profesor David Castells que me ha dado soporte y ayuda en la fase inicial en desarrollo del modelo del proyecto y a Ganyong Mo, que juntamente hemos investigado y analizado los códigos del modelo.

También quiero agradecer a la empresa Control Live S.L. y a sus trabajadores por todos los recursos y herramientas que me han aportado y el tiempo que han dedicado para el desarrollo del proyecto.

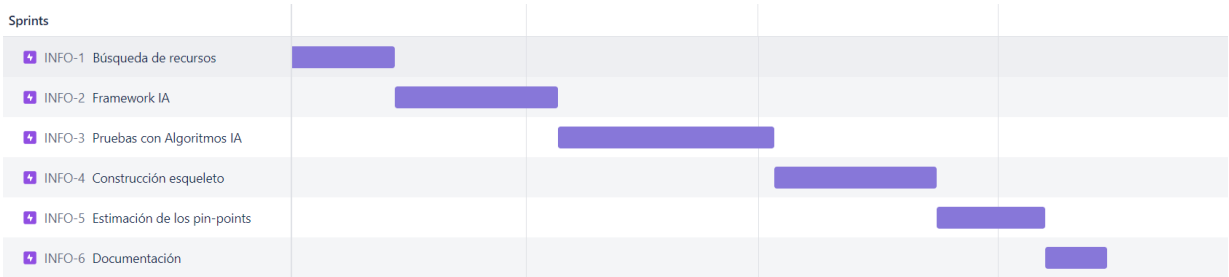
Finalmente, quiero dar mis agradecimientos a mi familia y los compañeros que siempre me han acompañado durante todo el estudio universitario.

BIBLIOGRAFÍA

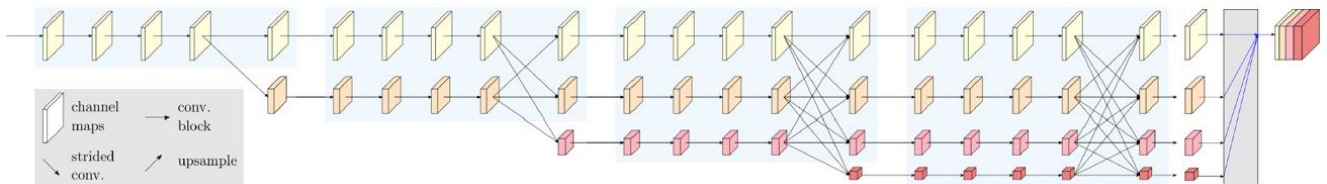
- [1] Abuelo Actual, "Que es una escara por presión?", <https://www.abueloactual.com/que-es-una-escara-por-presion>.
- [2] American Cancer Society, "Úlceras por presión" [En línea]. Available: <https://www.cancer.org/es/tratamiento/tratamientos-y-efectos-secundarios/efectos-secundarios-fisicos/piel-cabello-unas/ulceras-por-presion.html>.
- [3] S. L. Potosí, «Alto a las úlceras por presión: Secretaría de Salud» 20 11 2019. [En línea]. Available: <https://slp.gob.mx/ssalud/Paginas/Noticias/Alto-a-las-úlceras-por-presión-Secretaría-de-Salud.aspx>.
- [4] «Superficies especiales para el manejo de la presión». Doc. técnico nº13 de GNEAUPP (2011).
- [5] Mcinnes E, Jammali-Blasi A, Bell-Syer Se, Dumville Jc, Middleton V, Cullum N (2015). «Support surfaces for pressure ulcer prevention». The Cochrane database of systematic reviews (9): cd001735
- [6] Prevención, úlceras y heridas, «Prevención, úlceras y heridas,», <https://prevencionulcerasyheridas.com/producto/upp-v-chair/>.
- [7] Data set SLP disponible a <https://web.northeastern.edu/ostadabbas/code/>
- [8] "Neuronal Networks", <https://www.ibm.com/topics/neural-networks>
- [9] "Simple introduction to convolutional neural networks", <https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac>
- [10] "Arquitectura Típica Red Neuronal Convolutacional", <https://keepcoding.io/blog/arquitectura-tipica-red-neuronal-convolutacional/>
- [11] "Redes Neuronales Convolutacionales", <https://keepcoding.io/blog/redes-neuronales-convolutacionales/>
- [12] "Redes Neuronales", <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t8neuronales.pdf>
- [13] "Redes Neuronales Convolutacionales", <https://pochocosta.com/podcast/redes-neuronales-convolutacionales-explicadas/>
- [14] Wang, Jingdong, et al. "Deep high-resolution representation learning for visual recognition." IEEE transactions on pattern analysis and machine intelligence 43.10 (2020): 3349-3364.
- [15] "Visual Studio Code", <https://code.visualstudio.com>
- [16] "Jupyter Notebook", <https://jupyter.org/>
- [17] Liu, Shuangjun, and Sarah Ostadabbas. "Seeing under the cover: A physics guided learning approach for in-bed pose estimation." Medical Image Computing and Computer Assisted Intervention—MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22. Springer International Publishing, 2019.
- [18] "Python", <https://www.python.org/>
- [19] "PyTorch", <https://pytorch.org/>
- [20] Modelo HRPose del SLP, "SLP Dataset y Código", <https://github.com/ostadabbas/SLP-Dataset-and-Code>
- [21] "Cuda", <https://developer.nvidia.com/cuda-zone>
- [22] "Cuda Python", <https://developer.nvidia.com/cuda-python>
- [23] "Nvidia GeForce RTX 2080 Ti", <https://www.nvidia.com/es-es/geforce/news/gfecnt/geforce-rtx-2080-ti-reviews/>
- [24] "OpenCV", <https://opencv.org>
- [25] J. M. CASTRO i J. E. TORRA-BOU, «Utilización del sistema V-Chair®, una smart therapeutical surface, en el cuidado de dos pacientes de alta dependencia y movilidad limitada, con discapacidad intelectual en un centro sociosanitario.», *GEROKOMOS*, vol. 33, núm. 1, pp. 53-62, 2022.
- [26] R. R. Abril, «Estimación de pose humana,» *La Máquina Oráculo*, 2021. <https://lamaquinaoraculo.com/computacion/estimacion-de-pose/>.
- [27] M. Camacho, «Estimación de la Pose con Deep Learning,» ENIT-Innova IT business school, [En línea]. Available: <https://eniit.es/estimacion-de-la-pose-con-deep-learning/>.
- [28] Y. C. a. Y. T. a. M. He, «Monocular human pose estimation: A survey of deep learning-based methods,» *Computer Vision and Image Understanding*, 3 2020.
- [29] K. Rastogi, «Know all about 2D and 3D Pose Estimation!», *Analytics Vidhya*, 11 4 2022. [En línea]. Available: <https://www.analyticsvidhya.com/blog/2022/04/comprehensive-guide-for-pose-estimation/>.
- [30] Liu, Shuangjun, et al. "Simultaneously-collected multimodal lying pose dataset: Enabling in-bed human pose monitoring." IEEE Transactions on Pattern Analysis and Machine Intelligence 45.1 (2022): 1106-1118.
- [31] Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., & Sheikh, Y. (2019). «OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. » IEEE Transactions on Pattern Analysis and Machine Intelligence, 172–186. <https://doi.org/10.1109/tpami.2019.2929257>
- [32] He, Kaiming, et al. "Mask R-CNN." IEEE Transactions on Pattern Analysis and Machine Intelligence, June 2018, pp. 386–97, <https://doi.org/10.1109/tpami.2018.2844175>.
- [33] "Algoritmos Cuantitativos", <https://www.lifeder.com/algoritmos-cuantitativos/>
- [34] "downsampling and upsampling in cnn", <https://iq.opengenus.org/downsampling-and-upsampling-in-cnn/>

APÉNDICE

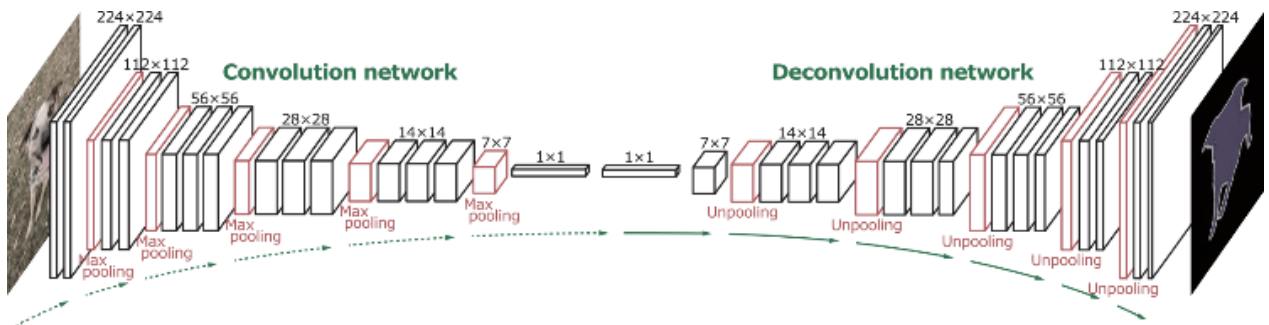
A1. ORGANIZACIÓN SPRINTS DEL PROYECTO EN JIRA



A2. ARQUITECTURA SEGMENTACIÓN HRNET



A2. ESTRUCTURA RED NEURONAL CONVOLUCIONAL



A3. PCK (PERCENTAGE OF CORRECT KEYPOINTS)

PCK, porcentaje de corrección de puntos claves. Es una métrica para evaluar la certeza de la estimación de poses humanas mediante la comparación de la distancia entre la predicción y *ground truth*.

Las métricas comunes aplicables:

- PCKh: porcentaje de puntos clave correctos normalizados por tamaño de cabeza.
- PCK@n: porcentaje de puntos clave correctos en un umbral de distancia normalizado.
- PCKh@n: porcentaje de puntos clave correctos normalizados por tamaño de cabeza en un umbral de distancia normalizado.

A4. ALGORITMO CUANTITATIVO

Los algoritmos cuantitativos son aquellos algoritmos que utilizan operaciones algebraicas y cálculos numéricos específicos para definir un proceso, obteniendo valores concretos [33].