UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DETECCIÓN DE POSE HUMANA EN IMÁGENES TÉRMICAS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

JAVIER IGNACIO SMITH DE LA CARRERA

PROFESOR GUÍA:
JAVIER RUÍZ DEL SOLAR SAN MARTÍN

MIEMBROS DE LA COMISIÓN:
PATRICIO LONCOMILLA ZAMBRANA
FRANCISCO RIVERA SERRANO

SANTIAGO DE CHILE
2021

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE MAGÍSTER EN CIENCIAS
DE LA INGENIERÍA
POR: **JAVIER IGNACIO SMITH DE LA CARRERA**
FECHA: 2021
PROF. GUÍA: JAVIER RUÍZ DEL SOLAR SAN MARTÍN

DETECCIÓN DE POSE HUMANA EN IMÁGENES TÉRMICAS

La detección de pose humana, tarea que concierne la localización de puntos importantes del cuerpo (*keypoints*) humano en imágenes, tiene una gran variedad de aplicaciones. Estas abarcan áreas desde el cuidado de la salud, hasta la realidad virtual. La mayoría de los sistemas implementados, se han enfocado en la detección de pose sobre imágenes a color. Sin embargo, una oportunidad se presenta en la detección sobre el dominio térmico, el cual presenta ventajas como invarianza a la iluminación y preservación de la privacidad de las personas.

Se propone la aplicación del modelo CenterNet a la detección de pose sobre imágenes térmicas. Este modelo posee una innovadora forma de detección, lo cual lo ha llevado a resultados estado del arte en competencias como COCO. Los objetos y personas se detectan a partir de sus puntos centrales, realizando regresión hacia otras propiedades como los *keypoints*.

CenterNet se pre-entrena sobre la base de datos COCO, cambiando entre las arquitecturas de extracción de características DLA, Hourglass y HRNet. Se entrenan variantes sobre las imágenes originales a color y en escala de gris. Estos modelos, al ser evaluados en un conjunto de 200 imágenes térmicas, alcanzan un máximo de $54,6\,\%$ de precisión (AP) y $64\,\%$ de *recall* (AR), por el modelo Hourglass. El pre-entrenamiento sobre imágenes en gris ha demostrado ser ligeramente mejor que con imágenes a color. Una variante, entrenada sobre imágenes de COCO traducidas a térmicas con el sistema ThermalGAN, ha probado ser fútil.

El desempeño de los modelos es mejorado realizando un *finetuning* sobre 600 imágenes térmicas. Se exploran diferentes combinaciones de *batch size* y *learning rate*, distintos *learning rate schedules* y el congelamiento de capas. La evaluación muestra que parámetros de entrenamiento cercanos a los referidos en la literatura, son óptimos, y que el congelamiento del primer módulo de convolución de los *backbone* incrementa marginalmente la precisión de los modelos. Los mejores modelos, basados en CenterNet DLA y Hourglass, alcanzan una precisión notable del $77\,\%$ y $80\,\%$, respectivamente. La diferencia se identifica como un *trade-off* entre precisión y tiempo de inferencia, donde el primero detecta a 22 FPS, mientras que el último solo a 10 FPS. También, se identifica una deficiencia en la detección de *keypoints* de la cara.

Los mejores modelos CenterNet, se comparan a otros sistemas de detección de pose humana populares, como Simple Baselines y PoseAE. En conclusión, Simple Baselines resulta ser el más preciso, con un $77\,\%$ de AP y $81\,\%$ de AR, a expensas de un largo tiempo de inferencia para imágenes con muchas personas. CenterNet, en cambio, se desempeña con alta precisión y tiempos de inferencia razonables. La elección del modelo depende de las necesidades que presente el usuario en cuanto a precisión y rapidez de detección.

Para mi hermano Esteban.

Agradecimientos

El primer agradecimiento lo quiero dedicar a mis padres María Isabel y Jorge, sin cuyo esfuerzo, apoyo y cariño, me hubiese sido difícil llegar a este punto de la carrera. Muchas gracias por todo.

También quiero agradecer al resto de mi familia, en particular mis hermanos. Por el apoyo durante estos años, tanto en Santiago como cuando estuve en Concepción, y los valiosos aprendizajes que he adquirido de ustedes.

A mi polola Loreto, con quien he compartido unos maravillosos últimos dos años. El tiempo pasado contigo ha endulzado todo aspecto de mi vida.

A mis amigos de la universidad y del colegio. Las risas y experiencias compartidas con ustedes las atesoro de corazón.

A mi profesor guía y co-guía, quienes me encaminaron y ayudaron en el desarrollo de aspectos fundamentales de este proyecto.

¡Un abrazo grande a todas estas personas e infinitas gracias!

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Motivación

En la última década, los principales desafíos de visión computacional se han concentrado en las áreas de clasificación de imágenes [23], detección de objetos [39], segmentación de instancias [40] y segmentación semántica [33]. En cuanto a la tarea de detección de pose humana múltiple, tema al que se aboca este trabajo de memoria, tiene como objetivo localizar las articulaciones (*keypoints*) de una o más personas en la imagen de entrada. Con estas detecciones, se puede construir un "esqueleto" que representa la configuración geométrica de las partes de las personas.

Al igual que otras tareas de visión computacional, la detección de pose se deriva de la detección de objetos, cuyo enfoque es localizar los objetos mediante cajas delimitadoras llamadas *bounding boxes* y categorizarlos en distintas clases. Sin embargo, abarca una variedad de aplicaciones totalmente distinta. La estimación de pose humana ha encontrado cabida en contextos como interacción humano-computador, análisis de movimiento, realidad aumentada, realidad virtual, el cuidado de salud, etc [6].

Actualmente, el enfoque de detección de pose humana múltiple con *deep learning* se ha dividido en dos paradigmas: detectores *top-down* [55, 40, 29] y detectores *bottom-up* [52, 36, 26]. El primer paradigma, lo forman modelos compuestos de una etapa de detección de personas, constituida por un detector de objetos convencional, y una segunda etapa donde se detecta la pose sobre cada una de estas personas identificadas. En cambio, en el segundo paradigma, se prescinde de la fase de detección de personas y se intenta ubicar los *keypoints* sobre la imagen completa y agruparlos en poses individuales para cada persona.

El trabajo descrito en este informe se centra en el segundo paradigma. Sobre este, la experiencia dicta que es menos preciso que el primero, dado que busca detectar los *keypoints* sobre una mayor cantidad de zonas de la imagen sin ellos. No obstante, avances recientes [52, 30, 36] demuestran que este tipo de modelos se pueden desempeñar con resultados competitivos con los métodos *top-down*, y a un costo de tiempo menor. Lo anterior, considerando la dependencia del tiempo de inferencia de los métodos *top-down* con la cantidad de personas presentes en la imagen.

Motivado por los resultados alcanzados por los modelos de detección de pose *bottom-up*, se explora, en esta memoria, la aplicación del modelo *CenterNet* [36] en el ámbito de detección de pose humana sobre imágenes térmicas. El modelo CenterNet, aborda este problema como la detección, simultánea, del punto central de cada persona y la regresión hacia las ubicaciones de los *keypoints*. Es, en esto, un sistema innovador que ha demostrado resultados estado-del-arte en competencias clásicas de detección de pose humana, como *Common Objects in Context* (COCO) [21].

Por otro lado, la idea tras ocupar imágenes térmicas se justifica a través de tres principales razones. Las imágenes térmicas son invariantes a la iluminación, por lo que pueden capturar a las personas aún en entornos de completa oscuridad. Además, son robustas a ciertos grados de oclusión, como cuando las personas están cubiertas por sábanas. La característica anterior, presenta una increíble ventaja en contextos hospitalarios, como ya ha sido explorado en otros trabajos [44, 27]. Aun más, las imágenes térmicas permiten resguardar, en cierta medida, la anonimidad de las personas detectadas. Esto último es conveniente en la actualidad, donde las personas viven reiteradas situaciones en las que su privacidad es vulnerada. Como consecuencia de todo lo descrito, es que se pretende que el trabajo desarrollado aporte al abrir una ventana de investigación y de aplicaciones en el ámbito de detección de pose humana sobre este nuevo dominio.

1.2. Objetivos del Trabajo de Título

1.2.1. Objetivos Generales

El objetivo general del presente trabajo es entrenar y aplicar un modelo de detección de pose humana sobre imágenes del dominio térmico. Para las tareas de detección, se entrenarán variantes del modelo CenterNet [36], con distintas arquitecturas *backbone*. Se busca, también, generar alternativas de detección que cumplan la inferencia en tiempo real. Adicionalmente, la precisión del modelo de detección tiene que ser comparable a resultados encontrados en la literatura para este tipo de problema.

1.2.2. Objetivos Específicos

Lograr el objetivo general propuesto requiere el cumplir varios objetivos específicos, los cuales se listan a continuación.

- Construir una base de datos representativa de imágenes térmicas de personas, etiquetadas con sus *keypoints*
- Entrenar variantes del modelo CenterNet, para detección de pose humana 2D, sobre un conjunto de entrenamiento extraído de la base de datos generada, analizando el efecto de las distintas arquitecturas *backbone*.
- Evaluar los detectores de pose entrenados sobre un conjunto de evaluación y, de acuerdo a estos resultados, proponer contextos donde pueden ser de utilidad los modelos entrenados.
- Seleccionar los mejores modelos de detección de pose humana, e implementar un programa de demostración que permita aplicarlos a imágenes o videos.

Capítulo 2

Antecedentes

En el presente capítulo se da a conocer el marco teórico y estado del arte que soporta el trabajo de título. Primero, se comienza con la descripción de las imágenes térmicas y los problemas de detección de objetos y pose humana. Después, se explican los diferentes paradigmas dominantes para la resolución de cada tarea. Más adelante, se describen las diferentes arquitecturas de redes neuronales profundas utilizadas como *backbone* del modelo CenterNet aplicado. En las últimas secciones, se detallan los métodos de traducción de imágenes, utilizados en este trabajo con el objetivo de generar datos de entrenamiento.

2.1. Imágenes Térmicas

Los sistemas de imagenología térmica permiten extender el campo de visión humano hacia el infrarrojo lejano, al hacer visible las radiaciones infrarrojas emitidas naturalmente por todos los cuerpos. En particular, las cámaras térmicas relevantes en este trabajo capturan la radiación que pertenece al rango de 9-14 [μm] del espectro electromagnético. De acuerdo a la teoría de radiación de cuerpos negros, objetos con mayor temperatura emiten una mayor cantidad de radiación infrarroja [14]. Como consecuencia, las cámaras térmicas permiten detectar e identificar cuerpos que manifestan una diferencia de temperatura con su entorno. Un ejemplo de esto es el cuerpo humano en un contexto urbano, situación que se refleja en la figura 2.1, donde se muestra una imágen del FLIR Dataset [4].

(a) Escala de grises (b) Mapa de calor

Figura 2.1: Ejemplos de imágenes térmicas.

Para efectos de este trabajo de título, se utilizarán imágenes capturadas mediante el sistema denominado *Forward Looking Infra-Red*, abreviado *FLIR*. Este método captura imágenes térmicas a través de un sistema óptico que colecta, filtra espectralmente, y enfoca la escena de radiación infrarroja sobre un arreglo de detectores multi-elemento, escaneados ópticamente [9]. Estos detectores convierten las señales ópticas en señales eléctricas analógicas que luego son amplificadas y procesadas para su visualización en un monitor o almacenamiento como imágenes.

Las imágenes térmicas corresponden a arreglos bidimensionales de pixeles, donde cada pixel corresponde a una intensidad de radiación infrarroja detectada. Valores altos del pixel significan una radiación infrarroja incidente más alta y, por ende, una mayor temperatura en la región referida. Dado que estas imágenes poseen un solo canal, se pueden representar en escala de grises como se muestra en la figura 2.1a, donde un tono más blanco sugiere cuerpos de mayor temperatura. También es posible representar estas imágenes usando un mapa de calor, como se ve en la figura 2.1(b).

2.2. Detección de Objetos y Pose Humana

2.2.1. Detección de Objetos

El problema de detección de objetos, se concentra en reconocer las categorías y predecir las ubicaciones de los objetos en imágenes o videos, ocupando para ello *bounding boxes*. Estas últimas son "cajas delimitadoras" que encierran el objeto en cuestión, como se demuestra en la figura 2.2 [17]. Notar, además, que las *bounding boxes* incluyen la categoría del objeto que encierran. En este caso, las categorías corresponden a un perro, una bicicleta y un camión.

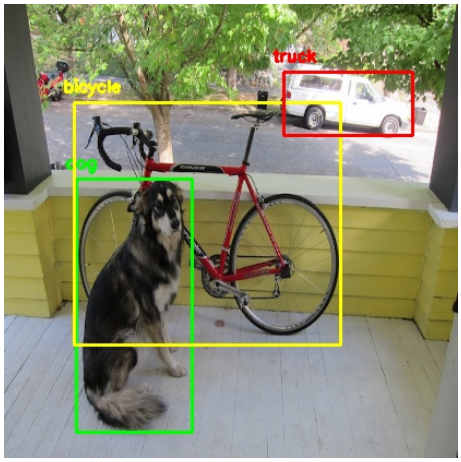


Figura 2.2: Ejemplo de detección de objetos.

Los *bounding boxes* se representan tradicionalmente de dos formas: a través de la ubicación de dos esquinas opuestas o mediante la ubicación de su centro y las dimensiones de la caja. De acuerdo a lo anterior, los modelos de detección entregan como *output* alguna de las dos alternativas por cada objeto detectado. La elección dependerá del tipo de post-procesamiento que incorpore el sistema.

2.2.2. Detección de Pose Humana 2D

Una tarea derivada de la detección de objetos es la detección de pose humana 2D. El objetivo de esta, es predecir la ubicación de las articulaciones de una persona, de aquí en adelante *keypoints*, en una imagen o video. Un ejemplo de esto se aprecia en la figura 2.3 [36]. En ella se puede ver que un modelo se encargó de predecir lo mejor posible los *keypoints* de los ciclistas y de los espectadores. En este caso, el modelo predice un total de 17 *keypoints*, entre los que se encuentran las rodillas, los ojos, los codos, las manos, etc. Luego, para cada una de las personas detectadas, construye una representación de la estructura geométrica de la persona.

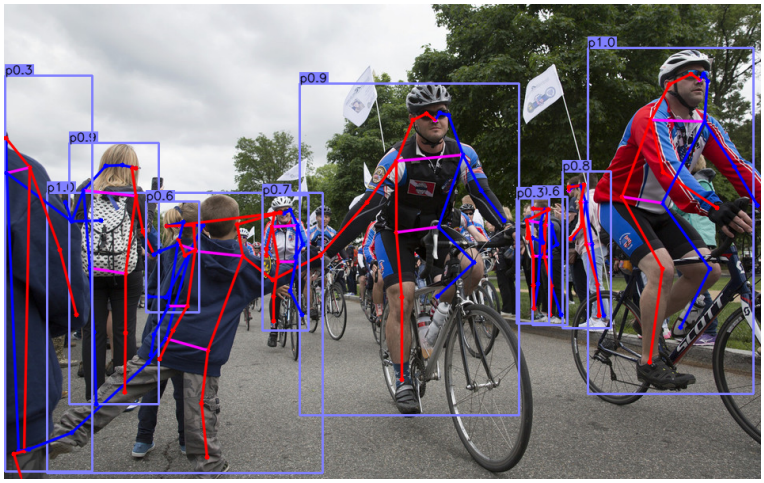


Figura 2.3: Ejemplo de detección de pose humana.

El problema de detección de pose humana, suele ser más complicado que el problema de detección de objetos por una variedad de razones. Primero, se tiene que la cantidad de detecciones es ampliamente mayor, siendo que se tiene que localizar cada *keypoint* de cada persona. Esto último presenta un gran desafío, considerando que las articulaciones suelen ser pequeñas o apenas visibles, presentar oclusión, verse deformadas por efecto de la ropa o estar afectas a los cambios de iluminación. Adicional a esto, los *keypoints* detectados tienen que asignarse de manera apropiada a cada persona, para formar las poses individuales. Problemas comunes en la asignación ocurren cuando hay personas cercanas en la imagen. No obstante lo señalado, herramientas actuales basadas en *deep learning* [20, 36, 52, 55, 40] permiten resolver este problema con un alto grado de confianza.

2.3. Paradigmas de Detección

2.3.1. Detección de Objetos

2.3.1.1. *Two-Stage* Detectors

Los *two-stage detectors* corresponden a un tipo de detectores de objetos que realizan esta tarea en dos etapas. La primera, corresponde a la generación de *region proposals*, donde se identifica zonas de la imagen que pueden ser potenciales objetos. La fase siguiente consiste en la predicción sobre las *region proposals* para clasificar los objetos en las categorías adecuadas. Un modelo popular que sigue este paradigma es Faster R-CNN [15].

2.3.1.2. One-Stage Detectors

A diferencia de los detectores de dos etapas, típicamente, estos modelos consideran cada posición de la imagen o mapa de características como un posible objeto y lo intentan clasificar en alguna de las categorías o como fondo. Usualmente, están conformados por una sola red construida a partir de varias capas convolucionales. Las capas convolucionales se encargan de la extracción de características de alto y bajo nivel (arquitecturas *backbone*), de la localización y etiquetado de los objetos, así como la asignación de un valor confianza en la predicción de estos.

Dada la gran naturaleza convolucional de los *single-stage detectors*, se pueden compartir pesos en distintas partes de la red y reducir la cantidad de parámetros necesarios para ajustar el modelo. A su vez, el hecho de tener una estructura simple, conformada por una o pocas redes convolucionales, facilita el entrenamiento de extremo a extremo. Esto contribuye a que sean modelos rápidos, en comparación a los detectores de dos etapas, y que, incluso, alcancen inferencia en tiempo real [17, 36, 8, 46]. No obstante, la precisión de estos modelos ha demostrado ser menor que su contraparte, constituyendo un *trade-off* entre eficiencia y calidad de predicción.

2.3.2. Detección de Pose Humana

Al igual que para la detección de objetos, en la detección de pose humana existen dos paradigmas de detección importantes. Asimismo, es necesario aclarar que se puede dividir esta tarea en detección de pose *single-person* y *multi-person*. La primera de ellas, concierne la detección de pose sobre una imagen de una persona tipo *bounding-box*. Por otro lado, *multi-person pose detection* se refiere a la detección de pose en una imagen que puede tener $0 - n$ personas. El interés del presente trabajo se concentra en esta última, y es en su contexto que se identifica los dos paradigmas descritos a continuación.

2.3.2.1. *Top-Down*

La detección de pose humana *top-down*, se caracteriza por separar el problema en dos fases. En la primera de ellas, se aplica un detector de personas para identificar las distintas instancias de personas en una imagen, a través de *bounding boxes*. Luego, la segunda fase consiste en aplicar algoritmos de detección de pose *single-person*. Estos algoritmos pueden ser de tipo regresivos, donde se entrena un mapeo desde la imagen a los *keypoints*, o con mapas de calor de *keypoints*, donde se aproximan las ubicaciones de estos, supervisados por mapas de calor *ground truth* de las anotaciones reales. Este proceso se puede interpretar como un sistema compuesto por un detector de objetos que identifica las personas y posteriores detectores de pose sobre cada persona identificada. En la figura 2.4 se muestra un esquema representativo de este tipo de modelos, extraída de [6].

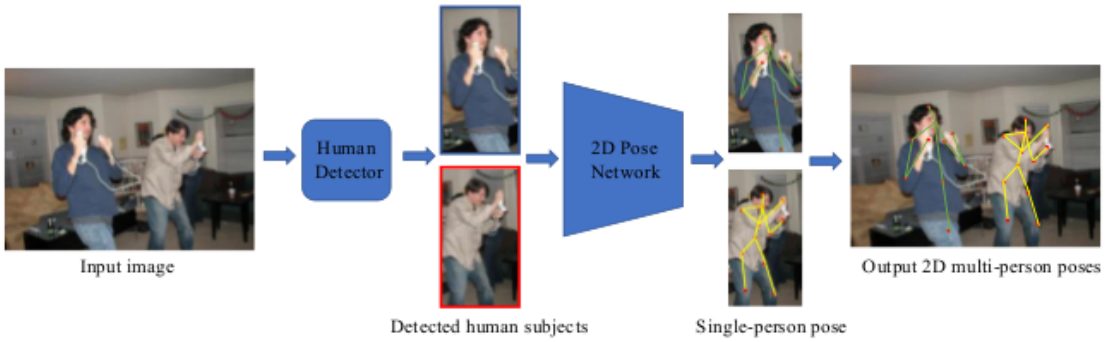


Figura 2.4: Esquema representativo de detección de pose humana por métodos *top-down* [6].

La ventaja de este sistema es que desglosa la detección de pose de múltiples personas en una imagen, a varias detecciones individuales. Como consecuencia, evita falsas detecciones que se pueden producir por considerar erróneamente zonas de la imagen sin personas y la localización de las articulaciones se restringe a áreas muy restringidas. Esto permite que sistemas basados en este paradigma, tales como variantes de *Mask R-CNN* [40] y, más recientemente, *Simple-Baselines* y *HRNet* [55, 29], alcancen resultados estado del arte en *benchmarks* clásicos como *COCO* y *MPII* [21, 19].

Sin embargo, el método *top-down* también posee algunas desventajas importantes. La primera de ellas, siendo la dependencia en los resultados de la detección de objetos. Si este no logra detectar una persona en la imagen, la pose de ella simplemente no se podrá estimar. El problema anterior se conoce como *early commitment*. Asimismo, se ha evidenciado que el tiempo de inferencia aumenta de manera proporcional a la cantidad de personas en la imagen. Esto se explica al considerar que se aplica un detector de pose diferente a cada instancia de persona identificada. También, otra desventaja que se puede señalar de este método es la posible asignación errónea, de articulaciones, a personas que están cercanas y cuyos *bounding boxes* se solapan.

2.3.2.2. *Bottom-Up*

En el paradigma de detección de pose *bottom-up*, se busca detectar cada *keypoint* presente en la imagen y luego agruparlos para formar la pose de cada individuo. Predominantemente, la detección de *keypoints* se realiza utilizando métodos basados en mapas de calor, al igual que algunos modelos de detección de pose *single-person*. Dado que estos sistemas no dependen de un detector de personas inicial, tienen el potencial para romper la relación proporcional entre el tiempo de inferencia y la cantidad de personas presentes en la imagen. Sin embargo, poseen la complejidad adicional de tener que agrupar las detecciones independientes de articulaciones. En la figura 2.5 se muestra un esquema representativo de este tipo de modelos, extraída de [6].

Figura 2.5: Esquema representativo de detección de pose humana por métodos *bottom-up* [6].

El primero de estos métodos [18], propone un enfoque de detección conjunta de articulaciones y asociaciones a individuos, con puntajes pareados obtenidos mediante regresión desde desfaces espaciales de las partes detectadas. Sin embargo, esta forma de asociación corresponde a un problema *NP-hard*, por lo que el tiempo de procesamiento para una sola imagen es del orden de horas. Más recientemente, otros métodos *bottom-up* han logrado popularidad gracias a sus métodos de agrupamiento de articulaciones más eficientes.

En particular, se puede destacar los *Part Affinity Fields* (PAF), introducidos por el modelo OpenPose [52] (ver figura 2.6) y el uso de *associative embedding vectors*, por el modelo PoseAE [26], como enfoques para agrupamiento de articulaciones exitosos. El primero de ellos consiste en un conjunto de flujos de campo que codifican relaciones, por parejas, entre partes del cuerpo de un número variable de personas. De esta manera, se puede generar un puntaje que indica la afinidad de una articulación con otra. Por otro lado, el concepto detrás de los *associative embedding vectors* [26] es asignar una "etiqueta" a cada *keypoint* y agrupar aquellos que tengan etiquetas similares. Estas etiquetas son los *vector embeddings*, que se asocian de acuerdo a una métrica de mínima distancia cuadrática.

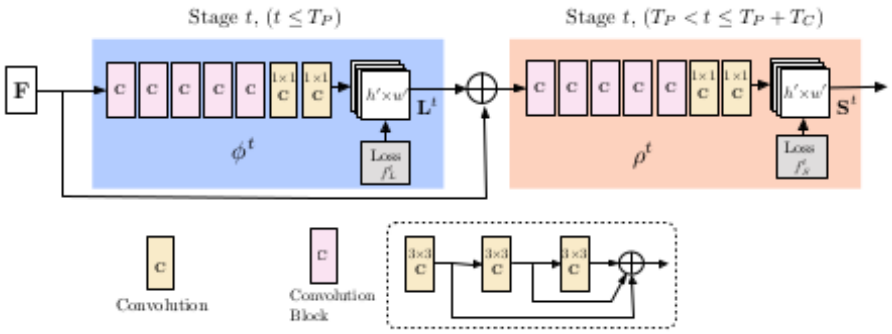


Figura 2.6: Arquitectura de modelo OpenPose [52], con la fase de detección de PAF (superior izquierda) por cada articulación y posterior generación de mapas de confidencia (superior derecha).

Nuevos modelos, como BottomUp-HRNet [30], han avanzado aún más el estado del arte en detección *bottom up*. Aun así, es importante señalar que el desempeño promedio de estos modelos aún está por debajo de otros métodos *top-down* [29, 55] en competencias como COCO y MPII [21, 19]. Sin embargo, pueden realizar detecciones sobre imágenes con múltiples personas de manera más rápida y precisa, y cada vez se acercan más a los mejores resultados obtenidos por el otro paradigma.

2.4. CenterNet

Para entender la detección de pose humana con CenterNet [36], conviene partir explicando la variante que realiza detección de objetos. El modelo CenterNet [36], comprende un detector de objetos *single-stage* que aborda la tarea como una predicción de los centros de los objetos y regresión hacia el tamaño de los *bounding boxes*. La novedad de este método, es que desecha el uso de *anchor boxes*, *priors* de los *bounding boxes* dispuestos a lo largo de la imagen, utilizados tradicionalmente por otros modelos *single-stage* [17, 46]. Esto es ventajoso, dado que el uso de *anchor boxes* introduce sesgos en la detección, el diseño de ellos es laborioso e incrementa el tiempo de inferencia.

Es importante destacar que el modelo discutido en este trabajo es "Objects as Points" (16/4/2019) ideado por X. Zhou, D. Wang y P. Krahënbül [36]. Es diferente al modelo "CenterNet: Keypoint Triplets for Object Detection" (17/4/2019) [37] planteado por K. Duan, S. Bai, L. Xie, H. Qi, Q. Huan y Q. Tian. Ambos modelos se refieren por el nombre CenterNet, pero toman enfoques diferentes.

Un esquema de la estructura de CenterNet [36] se ve en la figura 2.7. En ella, se puede ver que la imagen pasa primero por una arquitectura *backbone*, que se encarga de extraer representaciones y características de la imagen que faciliten la detección. La salida de esta red se alimenta a un módulo de predicción, compuesto de tres cabezas de clasificación que detectan características distintas de los *bounding box* detectados. Cada cabeza de clasificacion contiene una convolución con *kernel* de 3×3, una función ReLU y otra convolución con *kernel* 1×1. Incorporando otras cabezas de clasificación, se puede extender la capacidad de CenterNet para que realice detección de pose humana, segmentación semántica y más.

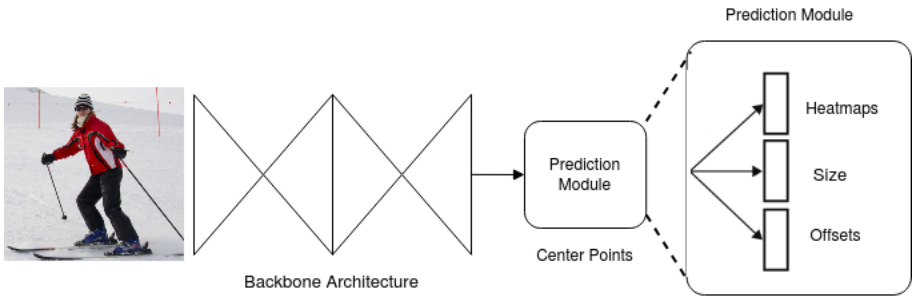


Figura 2.7: Esquema General de CenterNet (adaptado a partir de un esquema de CornerNet [8]).

2.4.1. Predicción de Centros

Sea la entrada de la red una imagen I de tamaño $W \times H \times 3$, donde W es el ancho, H el alto y tiene 3 canales. Uno de los objetivos es producir C mapas de calor (*heatmaps*) $\hat{Y} \in [0,1]^{\frac{W}{R} \times \frac{H}{R}}$, donde R es el *output stride* y C la cantidad de categorías. En el trabajo original, utilizan $R = 4$. Cada mapa de calor, para cada una de las categorías de objetos, contiene probabilidades de las localizaciones de los centros del objeto correspondiente, dentro de la imagen. De esta manera, tomando los *peaks* del mapa de calor que se encuentren sobre un valor umbral, se puede saber la ubicación de todos los centros de los objetos, como se muestra en el ejemplo de la figura 2.8.

Figura 2.8: Ejemplo de generación de mapas de calor para localizar centros de objetos. Imagen extraída de [36].

Para entrenar la generación del mapa de calor, se computa un equivalente de baja resolución $\widetilde{p} = \lfloor \frac{p}{R} \rfloor$ para cada punto de interés (centro) *ground truth* p de la clase c. Alrededor de cada punto $\widetilde{p}$, se computa un mapa de calor usando un *kernel* Gaussiano: $exp\left(-\frac{(x-\widetilde{p}_x)^2+(y-\widetilde{p}_y)^2}{2\sigma_p^2}\right)$, donde σ_p es una desviación estándar adaptiva al tamaño del objeto en cuestión. El objetivo de entrenamiento es una regresión logística con *focal loss* [34] de penalización reducida, desplegado en el sistema de ecuaciones 2.1. En esta ecuación, α y β son hiper-parámetros del *focal loss* [34] y N el número de objetos en la imagen. En el trabajo original [36], los autores escogen $\alpha = 2$, $\beta = 4$ para todas las pruebas.

$$L_k = -\frac{1}{N}\sum_{xyc}\begin{cases}(1-\hat{Y}_{xyc})^{\alpha}log(\hat{Y}_{xyc}), & \text{si } Y_{xyc}=1\\ (1-Y_{xyc})^{\beta}(\hat{Y}_{xyc})^{\alpha}log(1-\hat{Y}_{xyc}), & \text{otra manera}\end{cases} \tag{2.1}$$

2.4.2. Afinación Mediante *Offsets*

Para recuperar el error de discretización causado por el submuestreo con *stride* R, se genera un *offset* $\hat{O} \in \mathbb{R}^{\frac{W}{R}\times\frac{H}{R}\times 2}$ por cada centro detectado. Todas las clases comparten el mismo *offset*, el cual es entrenado utilizando una función de pérdida L1, exhibida en la ecuación 2.2. Como se aprecia en la figura 2.9, la función de este *offset* es corregir la ubicación del centro de los objetos, y con ello el *bounding box*, para la detección sobre la imagen de tamaño original.

$$L_{off} = \frac{1}{N}\sum_{p}\left|\hat{O}_{\tilde{p}} - \left(\frac{p}{R} - \tilde{p}\right)\right| \tag{2.2}$$

Figura 2.9: Ejemplo de generación de *offset* para corregir localización de centros de objetos. Imagen extraída de [36].

2.4.3. Predicción de Tamaño de *Bounding Boxes*

Para explicar la transformación de cada punto central p_k a un *bounding box* $(x_1^{(k)}, y_1^{(k)}, x_2^{(k)}, y_2^{(k)})$ que encasille el objeto de categoría c_k, se debe considerar que la transformación necesaria es $p_k = \left(\frac{x_1^{(k)}+x_2^{(k)}}{2}, \frac{y_1^{(k)}+y_2^{(k)}}{2}\right)$. Como se vio anteriormente, CenterNet utiliza el mapa de calor $\hat{Y}$ para detectar cada centro, extrayendo los puntos altos para cada categoría independientemente. Adicionalmente, el modelo predice mediante una regresión el tamaño de la *bounding box* $s_k = \left(x_2^{(k)} - x_1^{(k)}, y_2^{(k)} - y_1^{(k)}\right)$ para cada objeto k. El resultado de esto se puede ver en la figura 2.10. Se utiliza una sola predicción de tamaño $\hat{S} \in \mathbb{R}^{\frac{W}{R}\times\frac{H}{R}\times 2}$ para todas las categorías de objetos. Este predictor se entrena usando una función de pérdida L1, mostrada en la ecuación 2.3.

$$L_{size} = \frac{1}{N}\sum_{k=1}^{N}\left|\hat{S}_{p_k} - s_k\right| \tag{2.3}$$

Figura 2.10: Ejemplo de regresión al tamaño del objeto, a partir del centro detectado. Imagen extraída de [36].

2.4.4. Estimación Final de los *Bounding Boxes*

El objetivo general del entrenamiento se define como la ecuación 2.4. λ_{size} y λ_{off} son hiperparámetros que controlan la incidencia de las funciones de pérdida del tamaño y desfase, respectivamente. En el trabajo original [36], estos se configuraron en $\lambda_{size} = 0.1$ y $\lambda_{off} = 1$ para todos los experimentos. Recapitulando de las figuras 2.8, 2.9 y 2.10, CenterNet predice un mapa de calor para cada una de las C categorías, un $offset$ de dimension 2 y un tamaño de dimensión 2 para cada centro detectado.

$$L_{det} = L_k + \lambda_{size} L_{size} + \lambda_{off} L_{off} \tag{2.4}$$

2.4.5. Estimación de Pose Humana con CenterNet

CenterNet [36] realiza la estimación de pose humana como parte del paradigma *bottom-up*. Para ello, primero es necesario especificar la posición de cada *keypoint* de las personas en una imagen. En la base de datos COCO [21], hay $k = 17$ articulaciones totales. CenterNet considera la pose como una propiedad $k \times 2$ dimensional del centro, parametrizando cada *keypoint* por un *offset* al centro de la misma forma que la detección del tamaño del *bounding box* descrita anteriormente. De esta manera, se realiza una regresión $\hat{J} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times k \times 2}$ a la ubicación de cada *keypoint*, entrenándola con una función de pérdida L1.

Para mejorar las localizaciones, se estiman k mapas de calor $\hat{\Phi} \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times k}$ de articulaciones. Las predicciones iniciales se transforman al *peak* más cercano en el mapa de calor detectado. Aquí, el *offset* del centro actúa como una señal de agrupamiento para asignar cada detección de articulación a la instancia de persona más cercana. Al igual que para los mapas de calor de los centros de los objetos, se generan mapas de calor *ground truth* a partir de una expansión Gaussiana de los *keypoints* anotados. Los mapas de calor se entrenan con un *focal loss* [34] similar a la ecuación 2.1.

Específicamente, si $(\hat{x}, \hat{y})$ es un centro detectado, se realiza una regresión a todas las ubicaciones de articulaciones $l_j = (\hat{x}, \hat{y}) + \hat{J}_{\hat{x}\hat{y}j}$ para $j \in 1, .., k$. También, se extraen todas las articulaciones $L_j = \{\widetilde{l}_{ji}\}_{i=1}^{n_j}$ con una confianza > 0.1 para cada tipo de articulación j del mapa de calor $\hat{\Phi}_{..j}$. Luego, se asigna cada localización obtenida mediante regresión l_j a la articulación más cercana $\text{argmin}_{l \in L_j}(l - l_j)^2$, considerando solo las detecciones de articulación dentro del *bounding box* de la persona detectada.

Igual que en la detección de objetos, se predicen k *offsets* $\hat{O}_k \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times 2}$ que corrigen el error de sub-muestreo del mapa de características asociado a cada *keypoint* detectado. Estos *offsets* se entrenan utilizando una función de pérdida L1.

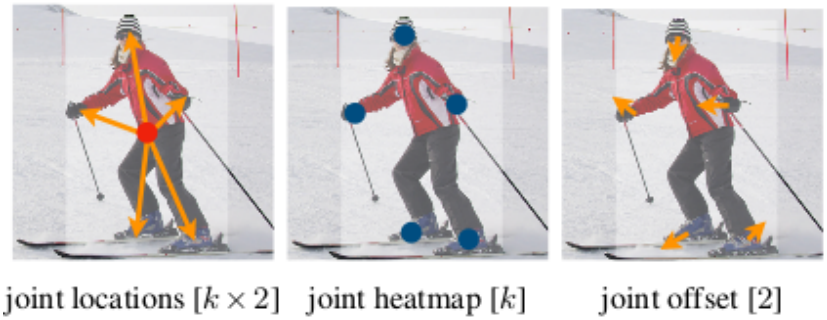


Figura 2.11: Predicción de pose humana mediante CenterNet [36].

Con todo lo anterior, el lector habrá notado que la extensión de CenterNet [36], desde la detección de objetos a la detección de pose humana, es en gran parte repetir el procedimiento de la detección de los centros para la detección de cada *keypoint*. Esto es, detectar un mapa de calor para cada *keypoint* y predecir un *offset* que corrija los mapas de calor. Luego, los *keypoints* se agrupan para cada persona de acuerdo a la predicción de los *offsets* del centro, la cual es similar a la predicción del tamaño de los *bounding boxes* en la detección de objetos. Esto, en términos de la estructura de la red, se traduce a incorporar tres nuevas cabezas de clasificación (dentro del módulo de predicción de la figura 2.7) que realice cada predicción. Cada predicción se ve esquematizada en la figura 2.11.

2.5. Arquitecturas *Backbone*

Las arquitecturas *backbone*, en el marco de detección de objetos y tareas similares con redes neuronales profundas, son redes artificiciales tradicionalmente convolucionales que se ocupan, primordialmente, en la extracción de características de una imagen. Permiten obtener representaciones de la imagen, idealmente de alto y bajo nivel, a partir de las cuales el modelo de detección puede envolver su funcionalidad. Debido a esto, son de gran importancia para el desempeño de los modelos.

En esta sección, se discuten algunas arquitecturas *backbone* relevantes para los modelos de detección empleados en el presente trabajo de título. La implementación del modelo CenterNet [36] facilita el intercambio entre distintos tipos de estas redes. Cambiar entre una y otra, tiene una incidencia significativa sobre la precisión que alcanza el sistema de detección y en el tiempo que tarda en realizar las inferencias.

2.5.1. ResNet

La arquitectura ResNet [28] se creó para resolver el problema de los gradientes desvanecientes en redes neuronales muy profundas. Hasta la fecha de su publicación, experimentos como [5, 16] demostraron que añadir más capas a una red que ya es profunda, conducía a errores de entrenamiento más altos. Como respuesta a esto, ResNet habilitó la construcción de redes neuronales más profundas, a través de la incorporación de módulos residuales, que incrementan el desempeño de los modelos.

En redes *feed forward*, los módulos residuales se crean mediante *skip connections*, lo cual está ilustrado en la figura 2.12 [28]. Estas son conexiones que permiten "saltar" una o más capas, sumando la entrada de esta a su salida a través de una matriz de identidad. Estas conexiones no añaden parámetros extras ni complejidad computacional pero permiten a la red que le sea más fácil optimizar los pesos de las capas.

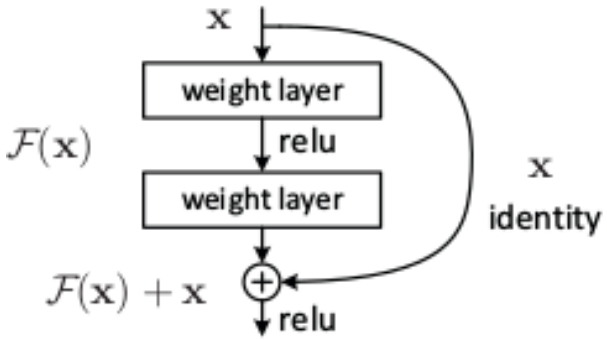


Figura 2.12: Conexión *skip* en Bloque residual [28].

2.5.2. DLA

La proposición de *Deep Layer Aggregation* (DLA) [47], es entregar una forma novedosa de conexión entre capas de una red. Para ello, se centra en la fase de agregación de una red, donde se combinan las capas. Utiliza un método que permite fusionar información espacial y semántica, capturada por las distintas capas para tareas de reconocimiento y agregación. Esto permite a la red capturar representaciones más profundas y que son compartidas a través de ella. Específicamente, DLA apunta a la agregación de los módulos de una arquitectura.

En DLA, la agregación se realiza a través de dos estructuras denominadas *Iterative Deep Aggregation* (IDA) y *Hierarchical Deep Aggregation* (HDA). Estas estructuras son herramientas, como los módulos residuales, que se pueden incorporar a una arquitectura *backbone* existente, como ResNet [28]. Estas estructuras se ilustran en las figuras 2.13 y 2.14, respectivamente.

IDA sigue la encadenación iterativa de los modulos y divide las cadenas de estas en etapas de acuerdo a la resolución del mapa de características sobre el cual trabajan. Agrega progresivamente la representación de los módulos, como se ve en la figura 2.13, incrementando la profundidad de la red.

Este proceso de agregación comienza en la escala más pequeña y superficial del espacio de características, para luego ir iterando en escalas mayores y más profundas. De esta manera, características superficiales de la imagen son refinadas mientras se propagan a través de las distintas etapas de agregación.

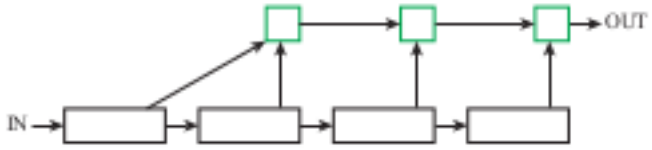


Figura 2.13: Iterative Deep Aggregation (IDA) [47].

HDA, por su parte, fusiona etapas y bloques de la red en un árbol para preservar y combinar los canales de características. Agrega capas superficiales y profundas de la red entre sí, posibilitando el entrenamiento hacia representaciones más ricas que abordan una mayor parte de la jerarquía de características. Aquí, la salida de un nodo de agregación se alimenta a la entrada del siguiente sub-arbol de la red, como se demuestra en la figura 2.14 [47].

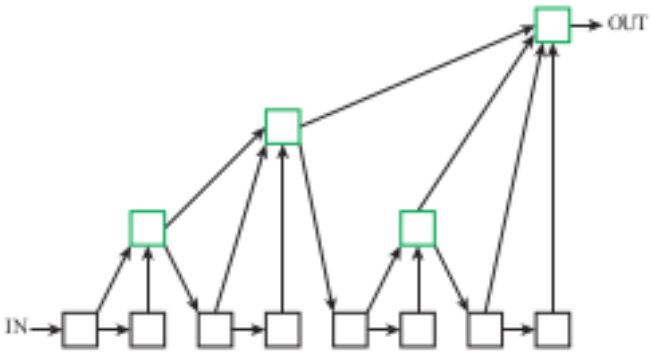


Figura 2.14: Hierarchical Deep Aggregation (HDA) [47].

La implementación de las dos estructuras anteriores, sobre una red *backbone* simple, se ve ilustrada en la figura 2.15 [47]. Las conexiones iterativas unen etapas vecinas para progresivamente profundizar y afinar, espacialmente, las representaciones. Las conexiones jerárquicas cruzan etapas con árboles de conexiones iterativas que abarcan un espectro de capas para propagar mejor las características y los gradientes de la red.

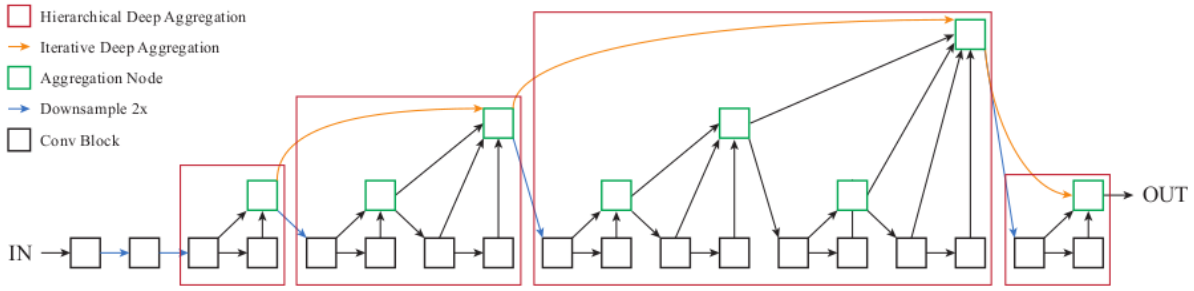


Figura 2.15: Ejemplo de estructuras IDA y HDA aplicadas sobre una red convolucional simple [47].

2.5.3. Hourglass

La arquitectura Hourglass [12] fue creada inicialmente para aportar en la solución al problema de estimación de pose humana. Su diseño está inspirado en la necesidad de disponer de una red que capture información en el mayor rango posible de escalas. Esto es específicamente útil para el problema señalado, donde las articulaciones de personas muchas veces se encuentran distribuidas en un gran rango de escalas. A pesar del propósito original para el cual fue creado esta red, se ha adaptado para resolver otros problemas como la detección de objetos [8][36]. En la figura 2.16 [12], se aprecia una ilustración de un módulo "hourglass". Notar la forma tipo reloj de arena (Hourglass) que le da el nombre característico a esta red.

Si bien en la literatura es común referirse a esta arquitectura como "Hourglass", el nombre correcto de la red es "Stacked Hourglass Network" [12]. Esto, puesto que está compuesta por varios módulos denominados Hourglass dispuestos en cadena. Cada módulo posee una distribución simétrica, en capacidad, entre la parte de procesamiento *bottom-up* y el procesamiento *top-down*. Dado que la topología del módulo es simétrica, cada capa que por un lado lleva el mapa de características hacia una resolución menor se le corresponde con una capa en el otro lado que lo lleva hacia una resolución mayor en la misma proporción.

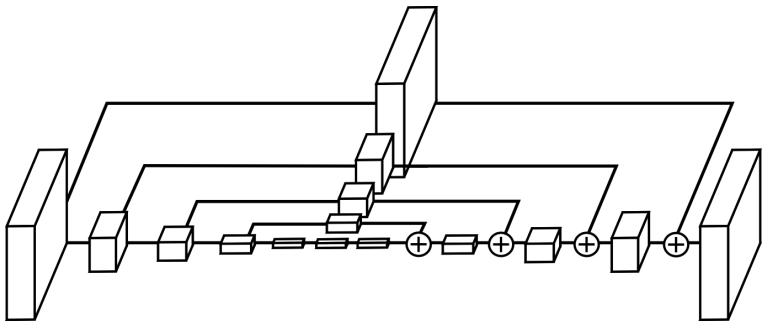


Figura 2.16: Ilustración de un módulo "hourglass".

El procesamiento *bottom-up* en un módulo hourglass se efectúa mediante capas convolucionales y capas de *max pooling*. Antes de cada capa de *pooling*, la red se ramifica y aplica convoluciones sobre el mapa de características en su resolución original. Esto es lo que se ve en la parte trasera de la figura 2.16. A su vez, el procesamiento *top-down* lo efectúa utilizando *nearest neighbour upsampling* y conexiones *skip*. Esto lo diferencia de otras redes tipo *encoder-decoder*, donde se utilizan capas de *unpooling* y de-convolucionales para las transformaciones.

A través del encadenamiento de módulos hourglass, se logra realizar inferencias *bottom-up* y *top-down* de manera repetitiva. Esto permite una forma de reevaluación de la estimaciones iniciales y características extraídas a través de la imagen. La clave de este enfoque es la predicción de mapas de calor intermedios, sobre los cuales se aplica una función de pérdida. Esta técnica se conoce como supervisión intermedia (ver figura 2.17 [12]), y permite que la red pueda evaluar las predicciones después de cada módulo, tomando en cuenta características a nivel local y global de la imagen. Las predicciones se reintegran de vuelta al espacio de características con la aplicación de una capa convolucional adicional, que les incrementa el número de canales.

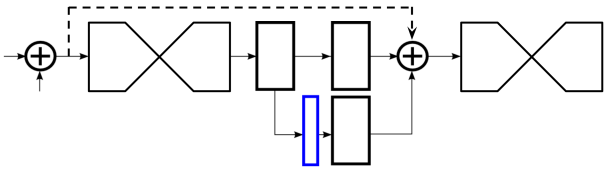


Figura 2.17: Supervisión intermedia entre los módulos hourglass.

2.5.4. HRNet

Por último, la arquitectura HRNet (*Higher Resolution Network*) [29] fue creada también con foco en el problema de estimación de pose humana. La novedad de esta red, es la preservación de la alta resolución de entrada a lo largo de todo el proceso. Esto lo logra a través de una estructura de sub-redes que trabajan en distintas resoluciones y que en cada etapa están conectadas de manera paralela. Ello le permite fusionar las representaciones de distintas escalas, como se puede apreciar en la figura 2.18. Como consecuencia, no sigue la separación del *pipeline* de detección en fases de alta-baja resolución y baja-alta resolución, que se ve en otros trabajos [12, 28, 47, 55].

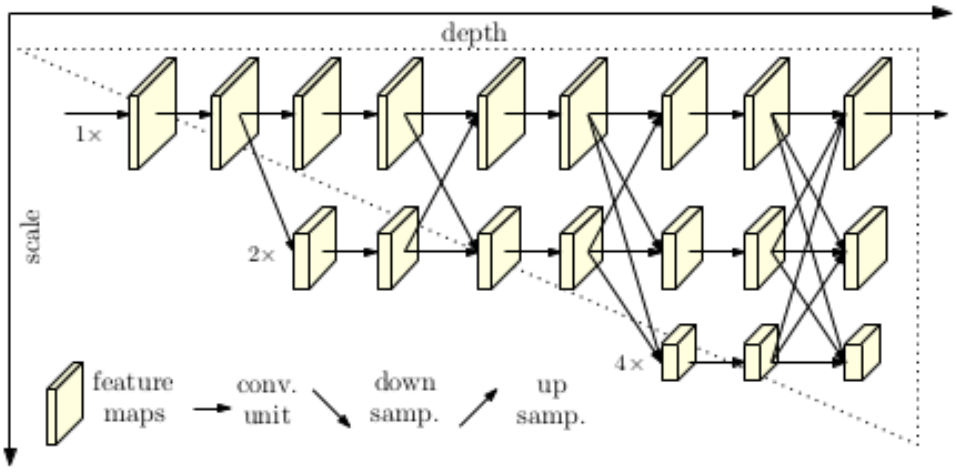


Figura 2.18: Arquitectura de HRNet [29].

De la figura 2.18, se ve como la red está compuesta por múltiples subredes, inicializadas a medida que se avanza en la profundidad de la red. Estas subredes trabajan a distintas resoluciones, y cada una de ellas comparte sus representaciones con las otras redes, fusionando de esta manera la información de distintas escalas (fusión multi-escala). El diseño sigue, además, lo propuesto por ResNet [28], distribuyendo la profundidad a cada etapa y el número de canales a cada resolución.

En total, hay 4 subredes y 8 unidades de intercambio a lo largo de la red donde se comparte la información extraída de distintas escalas. Los autores [29], proponen dos redes de distinto tamaño (HRNet-W32 y HRNet-W48). Los números 32 y 48 representan los anchos (canales) de las subredes de alta resolucion (red superior en la figura 2.18) en las últimas tres etapas. Para las otras subredes, de menor resolución, HRNet-W32 tiene anchos de $[64, 128, 256]$ y HRNet-W48 anchos de $[96, 192, 384]$.

2.6. Traducción de Imágenes Utilizando GANs

Un problema importante en la detección de pose humana sobre imágenes térmicas es la escasez de bases de datos anotadas. Al conocimiento del autor, el *SLP dataset* [44] es el único de libre acceso que contiene una cantidad suficiente de imágenes térmicas (cerca de 15.000) de personas etiquetadas con sus articulaciones. Sin embargo, esta base de datos solo contiene escenas de personas en camillas clínicas, lo cual limita el contexto de aplicación de algún modelo entrenado sobre ella a entornos similares.

En contraste, en el dominio RGB existe una variedad considerable de bases de datos anotadas con *keypoints*, siendo quizás las más populares COCO [21] y MPII [19]. En virtud de esto, se plantea aplicar un modelo que transforme estas imágenes al dominio térmico con una precisión suficiente. Esto permitiría utilizar la vasta cantidad de anotaciones de pose humana de estos *datasets*, para entrenar un modelo con imágenes de una distribución similar al dominio térmico. A continuación, se introducen algunos sistemas que se pueden utilizar potencialmente para lograr este objetivo.

2.6.1. Generative Adversarial Networks (GANs)

GAN [53] corresponde a un sistema de configuración de redes neuronales profundas para la estimación de modelos generativos. En el ámbito de visión computacional, esto significa, por ejemplo, crear imágenes artificiales de gatos a partir de una distribución de probabilidad p_{train} aprendida de un conjunto de entrenamiento de imágenes de gatos reales. Esto lo realiza a través de la competencia de dos redes, una generativa G y una discriminativa D. Está competencia se basa en que el generador trata de producir imágenes lo suficientemente parecidas al conjunto de entrenamiento como para engañar al discriminador, el cual tiene que discernir entre falso y real.

En el contexto de este trabajo, son de interés tanto el modelo GAN, como el modelo GAN condicional denominado *conditional generative adversarial network* (cGAN) [43]. En GAN, el generador se entrena para aprender una transformación desde un vector de ruido aleatorio z y la imagen de entrada x, a una imagen y objetivo. El discriminador se entrena para discernir entre la imagen generada y la imagen objetivo, como se ve en la ecuación 2.5. cGAN, por su parte, también incluye x como variable de entrada en el discriminador D. x es un vector que contiene información a priori de la transformación que se desea realizar. La función objetivo de un cGAN es la que se muestra en la ecuación 2.6. Estas funciones objetivos también suelen llamarse funciones de pérdida adversarias. Para ambas ecuaciones mostradas abajo, el generador busca minimizarlas y el discriminador, maximizarlas.

$$L_{GAN} = \mathbb{E}_y[logD(y)] + \mathbb{E}_{x,z}[log(1 - D(G(x, z))] \tag{2.5}$$

$$L_{cGAN} = \mathbb{E}_{x,y}[logD(x, y)] + \mathbb{E}_{x,z}[log(1 - D(G(x, z))] \tag{2.6}$$

2.6.2. Pix2pix

Pix2pix [3] aborda el problema de traslación imagen-a-imagen. Un ejemplo de esto, es la transformación de una imagen en escala de gris a una imagen a color. Este fue uno de los primeros trabajos en proveer un sistema general de traducción que se puede optimizar independiente de la traducción específica que es buscada. Pix2pix asume que, en el conjunto de entrenamiento, las imágenes de ambos dominios son del mismo tamaño y están alineadas.

Para abordar el problema, hace uso de las cGANs [43], optimizando la función objetivo mostrada en la ecuación 2.6, considerando x como la imagen del espacio de entrada, y la imagen del espacio objetivo y z un vector de ruido aleatorio. Adicionalmente, se incluye, en la función objetivo, la función de pérdida L1 que se ve en la ecuación 2.7. El objetivo final de entrenamiento es el que se muestra en la ecuación 2.8, con λ un hiperparámetro del modelo (definido en 100 para la mayoría de los experimentos en el trabajo original [3]).

$$L_{L1} = \mathbb{E}_{x,y,z}[||y - G(x,z)||] \tag{2.7}$$

$$L_{pix2pix} = \arg \min_G \max_D L_{cGAN}(G,D) + \lambda L_{L1}(G) \tag{2.8}$$

Para la red generadora, Pix2pix [3] utiliza una red similar a las redes *encoder-decoder* denominada U-Net. U-Net se podría definir como una variante de este tipo de red, donde se añaden *skip connections* entre las capas gemelas de la parte *encoder* y la parte *decoder*, como se ve en la figura 2.19 [3]. En cuanto a la red discriminadora, Pix2pix [3] utiliza una arquitectura denominada PatchGAN. Esta modela la imagen como un campo Markoviano aleatorio, segmentándola en una grilla de celdas de tamaño $N \times N$. El discriminador clasifica cada una de estas celdas en falsa o real, convolucionando a través de toda la imagen

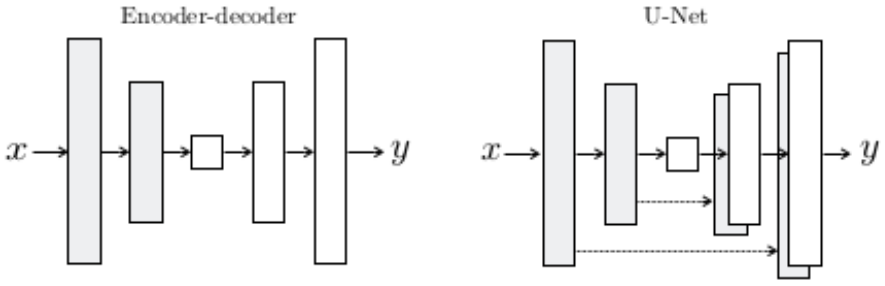


Figura 2.19: Red encoder-decoder (izquierda) y red U-Net[3] (derecha).

2.6.3. CycleGAN

CycleGAN [13] consiste en una modificación y extensión del modelo Pix2pix [3]. A diferencia de este último, presenta una alternativa de traducción imagen-a-imagen que no requiere imágenes alineadas. Está compuesto por dos sistemas GAN, uno para traducir la imagen de un dominio X a otro dominio Y, y otro para traducir una imagen del dominio Y al dominio X. Sigue la intuición de que una imagen traducida de un dominio a otro, al ser traducida de vuelta al dominio original, debería ser similar a la imagen original.

La formulación de CycleGAN [13] parte por dos modelos GAN con generadores $G : X \rightarrow Y$ y $F : Y \rightarrow X$, y dos discriminadores D_Y y D_X, respectivamente. Cada uno de estos modelos optimiza un objetivo del tipo mostrado en la ecuación 2.5, los cuales se llamarán $L_{GAN}(G, D_Y, X, Y)$ y $L_{GAN}(F, D_X, Y, X)$. De acuerdo a los resultados de [13], el vector de ruido z no aporta a la solución, por lo que se omite de la ecuación 2.5. Las funciones de pérdida, al igual que en Pix2pix [3], buscan guiar cada modelo a una traducción coherente con cada uno de los dominios objetivo. El ciclo se puede ver resumido en la figura 2.20(a) [13].

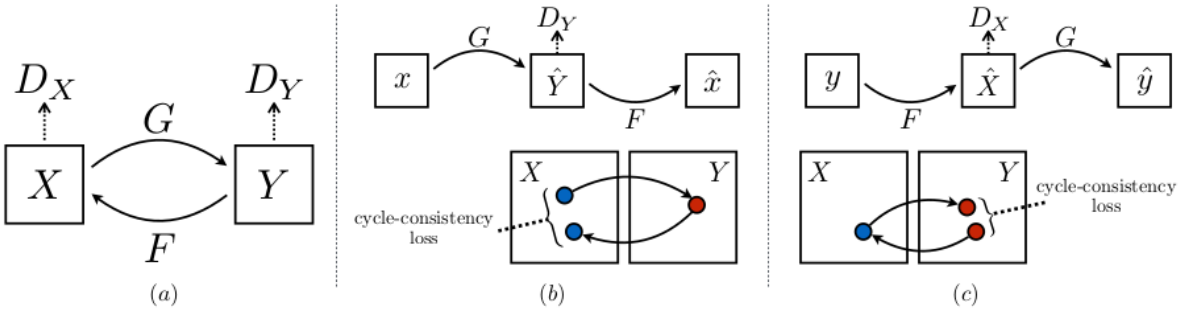


Figura 2.20: (a) Esquema del proceso CycleGAN (b) *forward cycle consistency loss* (c) *backward cycle consistency loss* [13].

Para reducir el espacio de funciones de traducción, propone un *cycle consistency loss* para asegurar que las traducciones sean coherentes. Esta función se aprecia en la ecuación 2.9. Puede desglosarse en dos partes: un *forward consistency loss* (figura 2.20(b)) que compara la imagen de entrada x con la imagen generada por $F : F(G(x))$, y un *backward consistency loss* (figura 2.20(c)), que compara la imagen de referencia y con la imagen generada por $G : G(F(y))$. El objetivo final de entrenamiento para CycleGAN [13] es el que se muestra en la ecuación 2.10, donde λ controla la importancia relativa de los dos objetivos adversariales (definido en 10 en [13]).

$$L_{cyc}(G, F) = \mathbb{E}_x([||F(G(x)) - x||] + \mathbb{E}_y([||G(F(y)) - y||] \tag{2.9}$$

$$L_{cyclegan} = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F) \tag{2.10}$$

En cuanto a las arquitecturas de las redes, las generativas se conforman de una red con dos capas convolucionales, varios módulos residuales [28] y dos convoluciones finales. Por el lado del discriminador, se imita a Pix2pix [3], utilizando redes PatchGAN con celdas de tamaño 70×70.

2.6.4. ThermalGAN

El modelo ThermalGAN [50], se construye también sobre pix2pix [3] y un modelo llamado BicycleGAN [22], basado en CycleGAN [13]. El contexto de creación se asocia al problema de re-identificación de personas utilizando sistemas multi-modales. Sin embargo, es potencialmente útil para aplicaciones donde se necesite la transformación de imágenes de color al dominio térmico.

La estructura de ThermalGAN se muestra en la figura 2.21. Este sistema recibe como entrada una imagen a color A y un vector de temperatura T, el cual contiene la temperatura deseada de los objetos y el fondo. Ambas son alimentadas a una red generadora G_1, adaptada del *framework* de BicycleGAN [22], la cual produce un mapa $\hat{S}$ con temperaturas promedio para cada objeto de la escena. A este mapa se le llama "imagen de segmentación térmica".

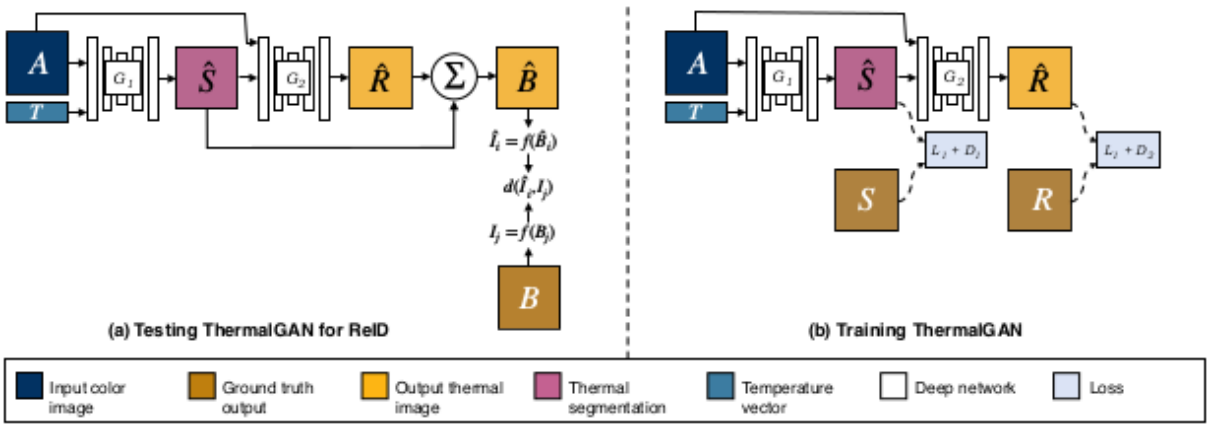


Figura 2.21: Esquema general del sistema ThermalGAN [50].

La imagen de segmentación térmica $\hat{S}$, junto a la imagen a color A, se introducen a otra red generadora G_2, la cual produce una imagen de contrastes térmicos relativos $\hat{R}$. La red G_2 es una red pix2pix [3], con modificaciones para aceptar un cuarto canal (correspondiente a la imagen de segmentación térmica producida por G_1). Finalmente, la suma entre la imagen de segmentación térmica $\hat{S}$ y la imagen de contrastes térmicos relativos $\hat{R}$, producen la imagen térmica absoluta $\hat{B}$ final.

Capítulo 3

Metodología

3.1. Bases de Datos

Para entrenar y evaluar los modelos, son importantes las bases de datos con imágenes con anotaciones para pose humana. A continuación, se describen las fuentes y bases de datos recopiladas del internet que sirven para el propósito de este trabajo.

3.1.1. COCO

La base de datos COCO [21], corresponde a un conjunto de imágenes a color con anotaciones para detección de objetos, pose humana, segmentación de instancias y otras tareas clásicas de visión computacional. Esta base de datos ha logrado popularidad como un *benchmark* de entrenamiento y evaluación para los modelos desarrollados en el contexto de las tareas anteriormente descritas [40, 17, 52, 36]. En este caso, se utiliza la versión del año 2017 para el entrenamiento y evaluación de los modelos de detección de pose humana.

COCO es una base de datos de gran escala, que contiene más de 200.000 imágenes, con cerca de 250.000 instancias de personas etiquetadas con *keypoints*. La mayoría de las personas etiquetadas son de escala mediana o larga, con máximo 17 *keypoints* anotados por persona. Estos *keypoints*, corresponden a la nariz, ojos, orejas, hombros, codos, muñecas, caderas, rodillas y tobillos. Los modelos, entrenados en este trabajo, apuntan a la detección de estos mismos *keypoints*, lo cual difiere de competencias asociadas a otras bases de datos como MPII [19], donde se detectan solamente 15 *keypoints*.

3.1.2. Imágenes Térmicas

Las bases de datos de imágenes térmicas, utilizadas en este contexto, provienen de distintas fuentes [4, 50, 38, 54, 41, 2, 32, 10, 45, 31, 24, 25, 11, 49, 51, 48]. Lo anterior, debido a que no existe una base de datos de imágenes térmicas con las anotaciones, variedad de escenas y de objetos como las que se encuentran en el dominio a color [21, 19]. Más aún, imágenes térmicas anotadas para pose humana son virtualmente inexistentes.

Para la construcción de la base de datos con imágenes térmicas anotadas para detección de pose humana, se extraen imágenes de distintas bases de datos de imágenes térmicas. La distribución de estas, se aprecia en los gráficos de las figuras 3.1 y 3.2, para el conjunto de entrenamiento y el conjunto de prueba, respectivamente. Se debe notar la diferencia en el aporte de imágenes de las distintas fuentes consultadas.

El diferente aporte de cada fuente se explica al considerar que muchas de ellas (ej. Boutiv, SLP y Biloudet [32, 45, 38]) provienen de videos que capturan escenas estáticas. Por lo tanto, para tratar de disminuir la correlación entre las imágenes del *dataset*, se extrae solo un conjunto acotado de imágenes de estas fuentes. Otras bases de datos, como FLIR y ThermalWorld [4, 50], no presentan este problema y pueden aportar más imágenes. Asimismo, se intenta incorporar imágenes de escenas variadas, tanto de interiores como exteriores. Un detalle más extenso de esto se puede encontrar en el anexo A.

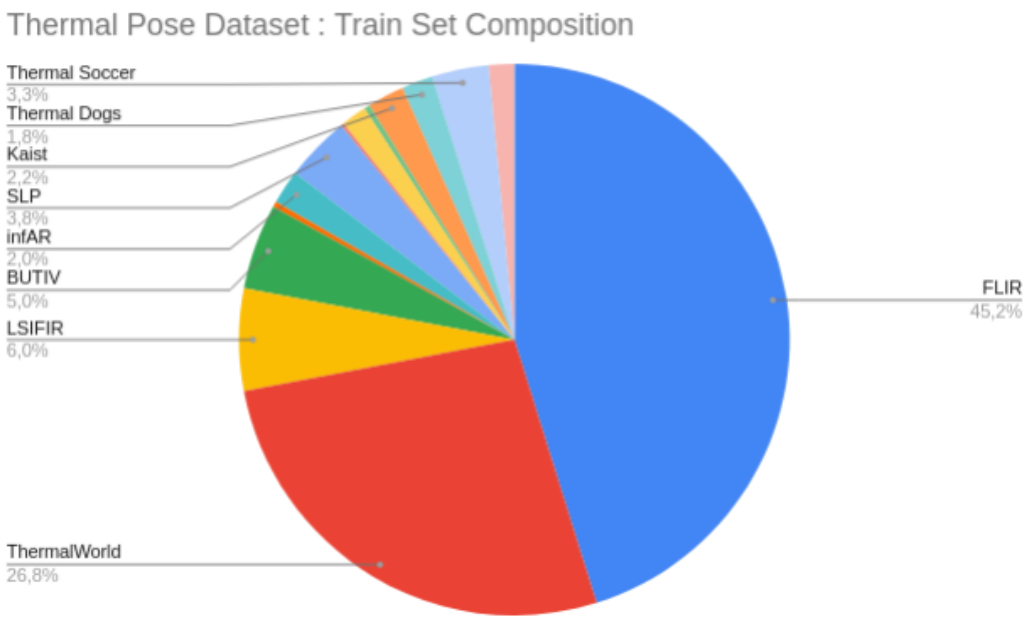


Figura 3.1: Distribución de imágenes, para el conjunto de entrenamiento del *dataset* construido, de acuerdo a la fuente

Para el caso de la transformación de imágenes a color al dominio térmico, se utilizan solo algunas de las fuentes referidas anteriormente. Lo anterior, dado que se necesitan *datasets* que contengan pares de imágenes color-térmico. En particular, para el entrenamiento de CycleGAN [13] se utiliza FLIR, AAU VAP Trimodal y Biloudet [4, 41, 38], escogiendo solo un grupo reducido de imágenes de las últimas dos fuentes. Por otro lado, para el entrenamiento de Pix2pix, dado que requiere pares de imágenes alineados, se utiliza Kaist [25] así como algunas imágenes de AAU VAP Trimodal [41].

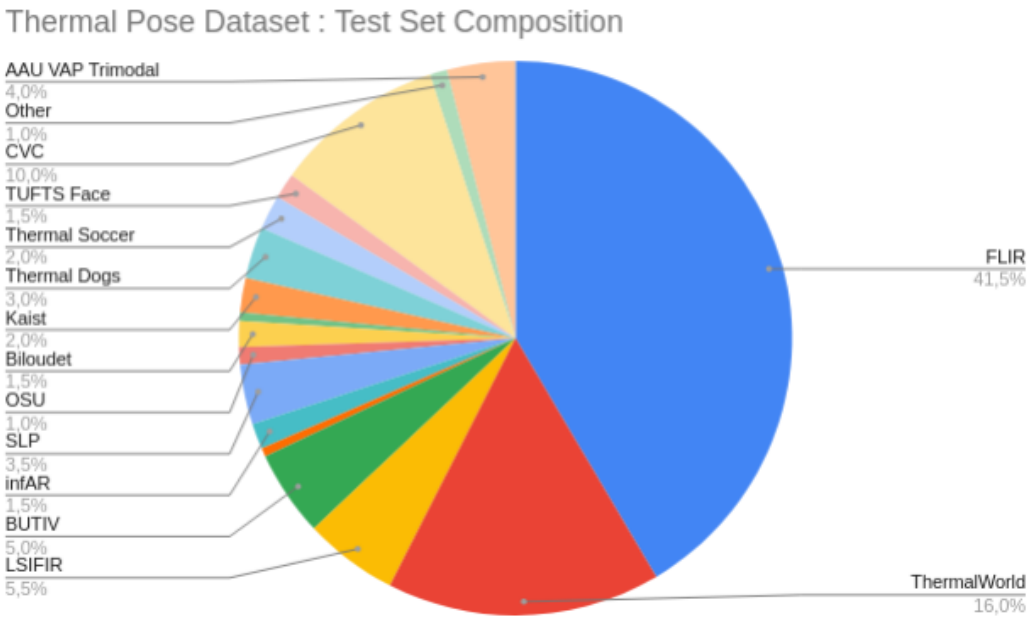


Figura 3.2: Distribución de imágenes, para el conjunto de prueba del *dataset* construido, de acuerdo a la fuente

3.2. Etiquetado de Imágenes Térmicas

Con el fin de disponer de una base de datos de imágenes térmicas anotadas con pose humana, se implementan dos alternativas. La primera de ellas, concierne el etiquetado manual de 600 imágenes para *finetuning* y 200 imágenes para evaluación de los modelos. La segunda alternativa, corresponde a la generación de imágenes térmicas artificiales utilizando sistemas tipo GAN. A continuación se describen ambas.

3.2.1. Etiquetado Manual

Para el etiquetado manual de imágenes, se utiliza la herramienta *labelme* [7]. Esta herramienta, permite marcar las ubicaciones de las articulaciones de las personas en la imagen a través de puntos, como se ve en la figura 3.3. Cada punto se categoriza según el nombre de la articulación que representa. Asimismo, se agrupa cada conjunto de articulaciones según una identificación numérica que represente la persona a la cual pertenecen. Por último, se añade una etiqueta para indicar el grado de visibilidad de la articulación, siendo 0 no presente en la imagen, 1 presente pero no visible, y 2 visible.

En total, se anotan 17 articulaciones por persona, sobre 800 imágenes. Estas imágenes son seleccionadas a partir de las fuentes discutidas en la sección anterior (ver figuras 3.1 y 3.2). 600 de estas imágenes se utilizan para formar un conjunto de entrenamiento y 200 de ellas para el conjunto de prueba.

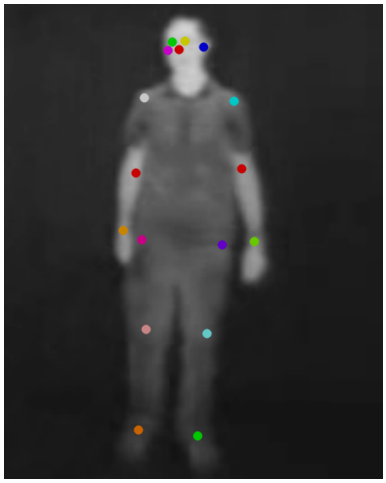


Figura 3.3: Ejemplo de anotación manual de articulaciones utilizando herramienta *labelme* [7].

3.2.2. Transformación de Imágenes de Color al Dominio Térmico

Los modelos de traducción de imágenes utilizados, corresponden a Pix2pix, CycleGAN y ThermalGAN [13, 3, 50]. Se utilizan los primeros dos modelos puesto que han sido exitosos en tareas como la transformación de imágenes en escala de gris a color. Esto sugiere que pueden ser útiles en la transformación de imágenes desde el dominio RGB al dominio térmico. ThermalGAN [50], por su parte, fue diseñado específicamente para la transformación de imágenes de color al dominio térmico.

3.2.2.1. ThermalGAN

Dado que ThermalGAN [50] posee un modelo entrenado para la traducción de imágenes a dominio térmico, no se necesita entrenar. Sin embargo, la transformación la realiza ocupando una "máscara de segmentación térmica" $\hat{S}$ (ver figura 2.21). Esta es generada a partir de la misma imagen a color y un vector de temperatura, utilizando el *framework* BicycleGAN [22]. Como no se dispone de un vector de temperatura para la base de datos que se desea transformar (COCO), esta máscara de segmentación se estima empíricamente a partir de los valores de la máscaras de ejemplo, disponibles en el repositorio del código (https://github.com/vlkniaz/ThermalGAN).

Para construir $\hat{S}$, se utiliza como base la máscara de segmentación anotada. Dado que el objetivo principal es emular la distinción entre las personas y el fondo, para construir la máscara $\hat{S}$ se utilizan dos valores posibles de pixeles. Uno de ellos, para las regiones que contienen personas (conocidas gracias a la máscara de segmentación anotada) y otro para el resto de la imagen.

(a) Imagen a Color (b) Máscara de Segmentación

Figura 3.4: Máscara de segmentación para una imagen a color. La máscara de segmentación térmica $\hat{S}$, tiene una forma parecida a (b) pero con diferentes valores de pixel para las personas y el fondo.

Se experimenta con una serie de combinaciones distintas para formar este par de valores para los pixeles. Si se define pix_p como el valor de la máscara en las regiones con personas y p_f como el valor de la máscara en las regiones sin personas, entonces, se exploran los valores $pix_p = [0, 6, 8, 10, 12]$ y $pix_f = [0, 3, 6, 8, 10, 12]$. Recordar, que la máscara de segmentación térmica $\hat{S}$ contiene, generalmente, intensidades de píxeles más altas para las regiones con humanos. Esto representa una mayor temperatura de las personas en comparación al entorno. No obstante, también se experimenta con valores de píxel más intensos para el fondo. Una máscara de segmentación térmica, generada a través de este método, se vería como la máscara de segmentación mostrada en la figura 3.4 (b), para la imagen a color (a).

3.2.2.2. Entrenamiento de CycleGAN y Pix2pix

El código base para Pix2pix y CycleGAN está disponible en el enlace: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix [3, 13]. Su implementación se basa en Python, utilizando la librería Pytorch. Para utilizar este modelo, es necesario clonar el repositorio anterior, e instalar las dependencias señaladas en el enlace.

Para preparar las imágenes, cada par se escala, manteniendo la relación de aspecto, a un ancho o alto fijo. En particular, se escalan las imágenes a una altura de 256 píxeles. Esto, siguiendo las sugerencias en [3, 13].

Para el modelo Pix2pix [3], se separa la base de datos en dos carpetas: `A` y `B`. La carpeta `A` contiene las imágenes RGB y la carpeta `B` las imágenes térmicas. Cada una de estos directorios se dividen en carpetas `train`, `val` y `test` que concentran los conjuntos de entrenamiento, validación y prueba, respectivamente. Una vez realizado esto, se debe mover el directorio completo (dentro del cual están las carpetas `A` y `B`) a la carpeta `datasets` del repositorio clonado.

Pix2pix se entrena en 200 épocas, siguiendo las recomendaciones de los autores [3]. Se entrenan modelos con la arquitectura generadora UNet-256, donde el número especifica la cantidad de capas convolucionales de la red. A su vez, se cargan las imágenes en su tamaño original y se especifican recortes aleatorios de 256×256 sobre estas. El resto de las configuraciones se mantiene en sus valores predefinidos. Durante evaluación, sobre el conjunto de prueba, se utilizan las mismas configuraciones que en la etapa de entrenamiento.

Para el modelo CycleGAN [13], se separa la base de datos en 4 carpetas: `trainA`, `trainB`, `testA` y `testB`. Las carpetas que terminan en `A` corresponden a imágenes del dominio RGB y las que terminan en `B` a imágenes del dominio térmico. A su vez, las carpetas `train` contienen las imágenes del conjunto de entrenamiento y las carpetas `test` del conjunto de prueba. Se utilizan las mismas configuraciones especificadas para pix2pix para entrenar este modelo.

3.2.2.3. Evaluluación Cualitativa

Ante la falta de una alternativa de evaluación cuantitativa para los modelos de traducción de imágenes, se decide evaluar cualitativamente los resultados sobre un conjunto de imágenes del *dataset* AAU VAP Trimodal [41]. Este contiene pares de imágenes alineados, de color y térmicas, por lo que se puede comparar el resultado de la transformación de la imagen a color con la imagen térmica real. Principalmente, se busca que las zonas de las imágenes transformadas, que contienen personas, sean distinguibles del entorno de manera similar a como sucede en las imágenes térmicas. Además, dado que se planea utilizar las anotaciones de pose de las imágenes a color, se busca que no se produzca un desplazamiento significativo de los sujetos de la imagen transformada con respecto a la de color.

3.3. Entrenamiento de Modelos de Detección de Pose Humana

En los experimentos, se entrena CenterNet [36] con arquitecturas *backbone* diferentes. En particular, se entrenan modelos con arquitecturas DLA-34, Hourglass-104 y HRNet-W32 [47, 12, 29]. Los números en los nombres de las arquitecturas DLA y Hourglass, indican la profundidad de capas máxima que alcanzan. Siguiendo a [36, 29], todos los entrenamientos se realizan con *fliping* aleatorio, escalamiento aleatorio (entre 0.6 y 1.3), *cropping* aleatorio y con la función Adam [1] para optimizar el objetivo general. El *data augmentation* anterior, permite introducir más variaciones al conjunto de entrenamiento, lo cual puede ayudar a la generalización de los modelos.

También, se entrenan otros modelos de detección de pose humana, para comparar resultados. En particular, se entrena el modelo Simple Baselines [55], perteneciente al paradigma *top-down* y tres modelos (OpenPose, PoseAE y Bottom-up HRNet [52, 26, 30]) pertenecientes al paradigma *bottom-up*. La preferencia por el entrenamiento de modelos *bottom-up*, se justifica al considerar que la detección de pose humana con CenterNet pertenece a este paradigma, por lo que resulta más relevante la comparación con otros modelos del mismo tipo. Todos estos modelos son entrenados utilizando los optimizadores, y los regímenes de *data augmentation*, especificados en los trabajos originales [55, 52, 26, 30].

La totalidad de los entrenamientos, se realizan utilizando dos GPUs Tesla-V100. Cada una de ellas cuenta con 32GB de memoria. La constancia de la plataforma de entrenamiento, permite que las comparaciones entre los modelos sean más justificables.

3.3.1. Pre-entrenamiento

El pre-entrenamiento de las redes, permite disponer de capas preparadas para computar representaciones generales de las imágenes, previo al entrenamiento a realizar para cumplir la tarea objetivo. Estas representaciones pueden ser, por ejemplo, bordes, manchas y texturas. Luego, se pueden seguir entrenando los pesos utilizando las imágenes térmicas anotadas con pose humana. A continuación, se describen dos formas de pre-entrenamiento ocupadas en este trabajo.

3.3.1.1. ImageNet

Imagenet, es una base de datos de millones de imágenes anotadas para clasificación. En la práctica [40, 17, 15, 52], suele ser utilizada para pre-entrenar las capas de las redes *backbone* que se utilizan para tareas más específicas como detección de objetos y de pose humana. Se sigue a [36], y se utiliza la red DLA [47] inicializada con los pesos de las capas de *down-sampling* entrenados en ImageNet, y los pesos de las capas de *up-sampling* inicializados aleatoriamente. La red Hourglass [12], por su parte, se extrae del modelo ExtremeNet [35] entrenado también en ImageNet. La red HRNet, si bien no se incluye en el trabajo original de CenterNet, se inicializa con los pesos pre-entrenados sobre ImageNet, siguiendo el ejemplo de [29] y [30]. Los otros modelos entrenados (Simple Baselines, OpenPose, PoseAE y Bottom-HRNet [55, 52, 26, 30]), también poseen pre-entrenamiento de sus redes *backbone* sobre ImageNet.

3.3.1.2. Modelos de Detección de Objetos

Siguiendo las recomendaciones del trabajo original de CenterNet [36], los modelos con *backbone* DLA y Hourglass [47, 12] se entrenan para detección de pose a partir de los pesos de los modelos entrenados para detección de objetos sobre COCO [21]. Esto es, un pre-entrenamiento sobre COCO de 230 épocas para DLA y de 50 épocas para Hourglass. Para el caso del *backbone* HRNet [29], no se dispone de un modelo entrenado sobre COCO para detección de objetos con el *framework* de CenterNet. Sin embargo, se utilizan los pesos de la red entrenada con el *framework* del trabajo que la presentó [29], para emular las condiciones de las otras redes.

3.3.2. Entrenamiento Sobre Imágenes Transformadas con GAN

El mejor modelo de transformación de imágenes a color a dominio térmico, se utiliza para transformar las imágenes del *dataset* COCO [21]. Esta elección se efectúa en base a los resultados de la evaluación cualitativa de los modelos, discutida anteriormente. Utilizando las imágenes transformadas, se entrena CenterNet con *backbone* DLA por 140 épocas, con un *learning rate* inicial de 0.0005 (con disminuciones a un 10 % en las épocas 90 y 120), resolución de entrada de 512×512 y *batch size* de 56.

3.3.3. Entrenamiento sobre COCO

Adicional al entrenamiento de los modelos sobre las imágenes térmicas artificiales, se entrenan otros modelos CenterNet con las imágenes originales (a color) y con las imágenes transformadas a escala de gris. Entrenar con imágenes en escala de gris, se cree que puede ayudar a la detección de pose humana sobre imágenes térmicas dada su mayor similitud. Considerar que las imágenes térmicas son arreglos de píxeles de distintas intensidades pero con un solo canal, lo cual se asemeja al formato de las imágenes térmicas.

En la tabla 3.1, se adjuntan los parámetros de entrenamiento de los diferentes modelos CenterNet. Notar que se incluye una columna de nombre "régimen" con los valores $1x$ y $3x$. Se puede observar que el entrenamiento utilizando el régimen $3x$ considera una mayor cantidad de épocas. Este tipo de entrenamiento se reserva para el final, una vez que se definen los mejores parámetros y condiciones de entrenamiento, con tal de ahorrar recursos computacionales y tiempo.

Tabla 3.1: Parámetros de entrenamiento para CenterNet sobre COCO.

Backbone	**Régimen**	***Lr***	**Épocas**	***Lr steps***	***Batch size* 1**	***Batch size* 2**
DLA-34	1x	5,00E-04	140	90;120	8	56
Hourglass-104	1x	2,50E-04	50	40	4	24
HRNet-W32	1x	1,00E-03	140	90;120	12	12
DLA-34	3x	5,00E-04	320	270;300	8	56
Hourglass-104	3x	2,50E-04	150	130	4	24
HRNet-W32	3x	1,00E-03	320	270;300	12	12

Todos los modelos se entrenan, siguiendo el régimen $1x$, sobre COCO con imágenes a color y con las imágenes en escala de gris. La variante de COCO, que entrega los modelos con mejor desempeño sobre el conjunto de evaluación de imágenes térmicas (luego del *finetuning*), se utiliza para entrenar los modelos con el régimen $3x$. Esto permite optimizar los tiempos de los experimentos, dado que aquellos entrenamientos con régimen $3x$ demoran cerca de 5 días por cada variante.

Asimismo, es importante destacar que los modelos con *backbone* DLA y Hourglass, se entrenan con diferentes *batch size* entre ambas GPUs. La razón de utilizar un *batch size* menor en una GPU, es que permite actualizar los pesos de las redes de manera más frecuente, otorgando mayor importancia al grupo específico de imágenes con las que se está entrenando. A su vez, el *batch size* en la GPU restante es mayor para inducir una señal de entrenamiento general y robusta al ruido. Esta práctica sigue lo especificado en [36]. Para el caso de HRNet, el entrenamiento se realizó con un *batch size* idéntico en ambas GPUs, para emular la metodología utilizada en el trabajo original [29].

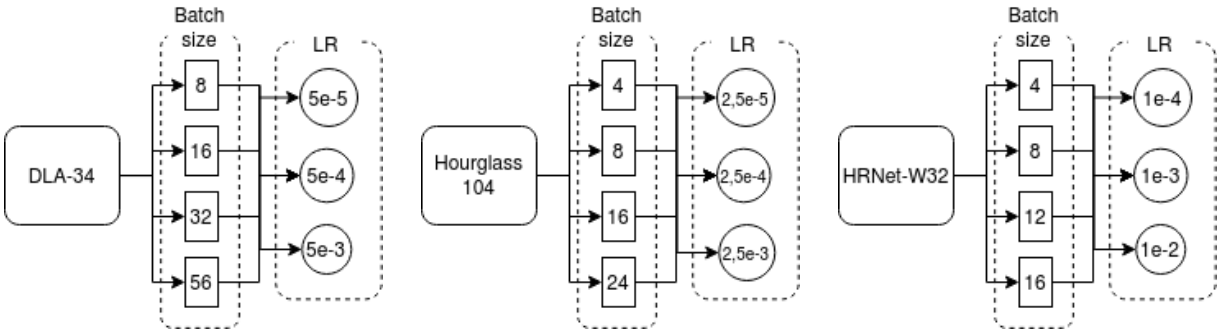
3.3.4. *Finetuning* sobre Imágenes Térmicas

Los modelos, pre-entrenados con el régimen 1x sobre COCO, se siguen entrenando con un *finetuning* sobre 600 imágenes térmicas. Esto permite "afinar" los pesos de las redes para la detección de pose humana en el dominio térmico. Dado que estas pruebas tienen una duración menor, se puede experimentar con distintos parámetros de entrenamiento

3.3.4.1. Diferentes Combinaciones de *Learning Rate* y *Batch Size*

Probablemente, los parámetros más importantes en el entrenamiento de las redes neuronales profundas son el *learning rate* y el *batch size*. Acorde a lo anterior, se realizan varios experimentos de *finetuning* con diferentes combinaciones de *learning rate* y *batch size*. Las variaciones se realizan en torno a los parámetros sugeridos por el *paper* de CenterNet [36], para el entrenamiento de los modelos con arquitecturas DLA y Hourglass. En cuanto a HRNet, la exploración se realiza en torno a los parámetros sugeridos en [30].

Para cada arquitectura, los parámetros de entrenamiento referidos son distintos. Luego, la búsqueda de los *batch size* y *learning rate* óptimos se realiza en la vecindad de los parámetros referenciales de cada uno. Esto se aprecia, para DLA, en la figura 3.5(a). Se puede ver que se experimenta con 4 *batch size* distintos, donde para cada uno se prueban 3 *learning rates* distintos. Para Hourglass y HRNet, la búsqueda se realiza sobre un espacio distinto, como se ve en las figuras 3.5(b) y 3.5(c). Las épocas de entrenamiento, en estas pruebas, son de 10, 5 y 10 épocas para DLA, Hourglass y HRNet, respectívamente.



(a) DLA (b) Hourglass (c) HRNet

Figura 3.5: Diferentes combinaciones de *batch size* - *learning rate* se evalúan para las arquitecturas DLA, Hourglass y HRNet.

Cabe decir, en lo que respecta a las redes DLA y Hourglass, que el *batch size* indicado corresponde al de la GPU 2. Para la GPU principal, se mantiene el *batch size* original de 8 para DLA y 4 para Hourglass. Además, no se experimenta con valores más grandes para esta variable debido a las limitaciones de memoria de la plataforma de entrenamiento. Es más, utilizar un *batch size* de 56 para DLA y de 24 para Hourglass, consume casi la totalidad de los 32GB de memoria de la GPU.

3.3.4.2. Diferentes *Learning Rate-Schedules*

Seleccionando la mejor combinación de *batch size* y *learning rate* para cada arquitectura, se procede a experimentar con diferentes *learning rate schedules*. Esto último, hace referencia a la o las épocas (*learning rate steps*) donde se disminuye el *learning rate*. Para efectos de este trabajo, la disminución es a un 10 % del *learning rate* anterior en las épocas señaladas. Esto sigue la práctica general para el entrenamiento de modelos de detección de pose humana que se puede encontrar en la literatura [36, 29, 52, 26].

Los modelos, esta vez, se entrenan en una mayor cantidad de épocas. CenterNet DLA se entrena en 50 épocas, primero sin disminución de *learning rate*, luego con disminución en las épocas 35 y 45 y, finalmente, con disminución en las épocas 45 y 47. Para Hourglass, se sigue un procedimiento similar, entrenando en 20 épocas y experimentando sin disminución, con disminución en las épocas 10 y 17 y un último experimento con disminución en las épocas 15 y 18. En el caso de CenterNet con HRNet, se sigue la misma exploración que en DLA, dado la similaridad de los *learning rate schedules* sugeridos en el *paper* para el entrenamiento sobre COCO (ver tabla 3.1).

3.3.4.3. Congelamiento de Capas

El congelamiento de capas puede ayudar al desempeño de los modelos, al preservar las representaciones capturadas por el entrenamiento de los modelos en el *dataset* COCO más extenso. También, para plataformas de entrenamiento con capacidades de computación limitada con respecto a los trabajos originales, admite menores tiempo de entrenamiento así como un menor gasto de recursos computacionales. Esto, dado que se paraliza el entrenamiento de ciertas capas de la red *backbone*, por lo que se deben actualizar una menor cantidad de parámetros en cada iteración.

Aquí, se aborda una estrategia de congelamiento de capas como la que se esquematiza en las figuras 3.6 y 3.7, para las arquitecturas DLA y Hourglass. Estas figuras, que ilustran versiones simplificadas de las redes de detección, muestran varios tipos de congelamiento experimentados para las arquitecturas, llamados *freeze 1*,...,*freeze n*. Estas pruebas se realizan considerando las mejores combinaciones de *learning rate*, *batch size* y *learning rate schedule* para cada arquitectura.

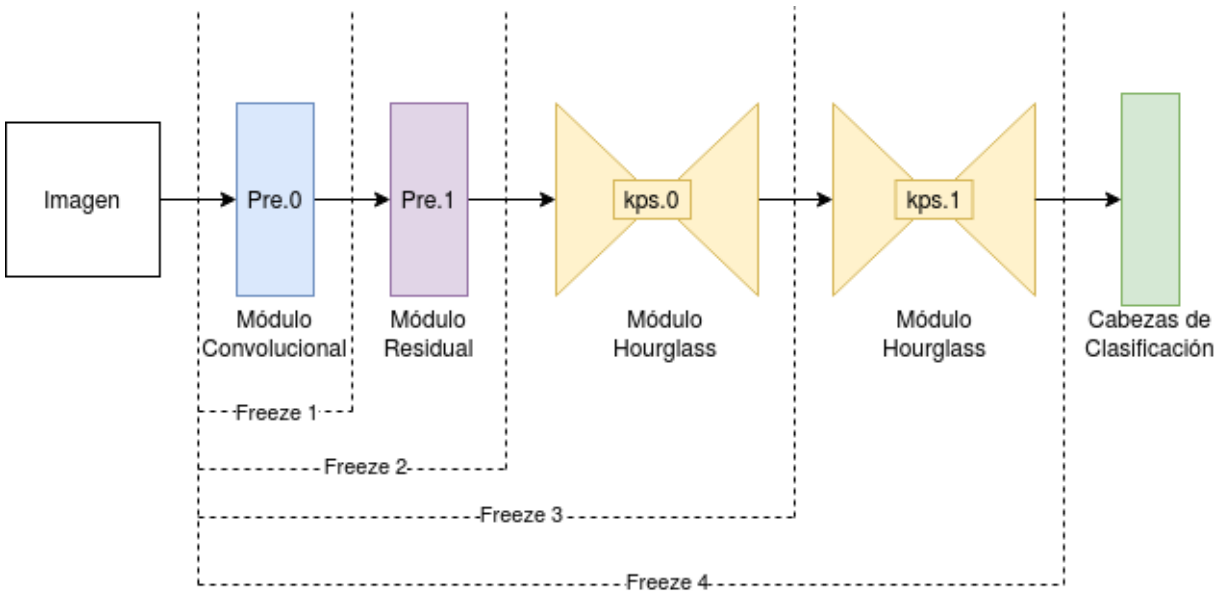


Figura 3.6: Pruebas de *finetuning* con diferentes niveles de congelamiento de la red *hourglass*.

Con respecto a CenterNet Hourglass, primero se experimenta realizando un *finetuning* con el primer módulo convolucional (*Pre.0*) congelado. Luego, se entrena con congelamiento hasta el módulo residual *Pre.1*. Un nivel de congelamiento adicional involucra los dos modulos anteriores y el primer módulo *hourglass* (*kps.0*). Finalmente, se prueba congelando toda la red *backbone* (Pre.0 hasta kps.1). Estos experimentos reciben el nombre de *freeze 1*, *freeze 2*, *freeze 3* y *freeze 4*, respectivamente.

El congelamiento de capas en la red DLA es algo más complejo que en Hourglass, como se muestra en el esquema de la figura 3.7. Este esquema es una versión muy simplificada de la estructura de la red, dado que la real considera conexiones de agregación entre los módulos DLA jerárquicos. Sin embargo, para efectos de este trabajo, la red se puede descomponer en tres módulos convolucionales iniciales, seguidos de 4 módulos DLA jerárquicos (ver *Hierarchical Deep Layer Aggregation* en la figura 2.15).

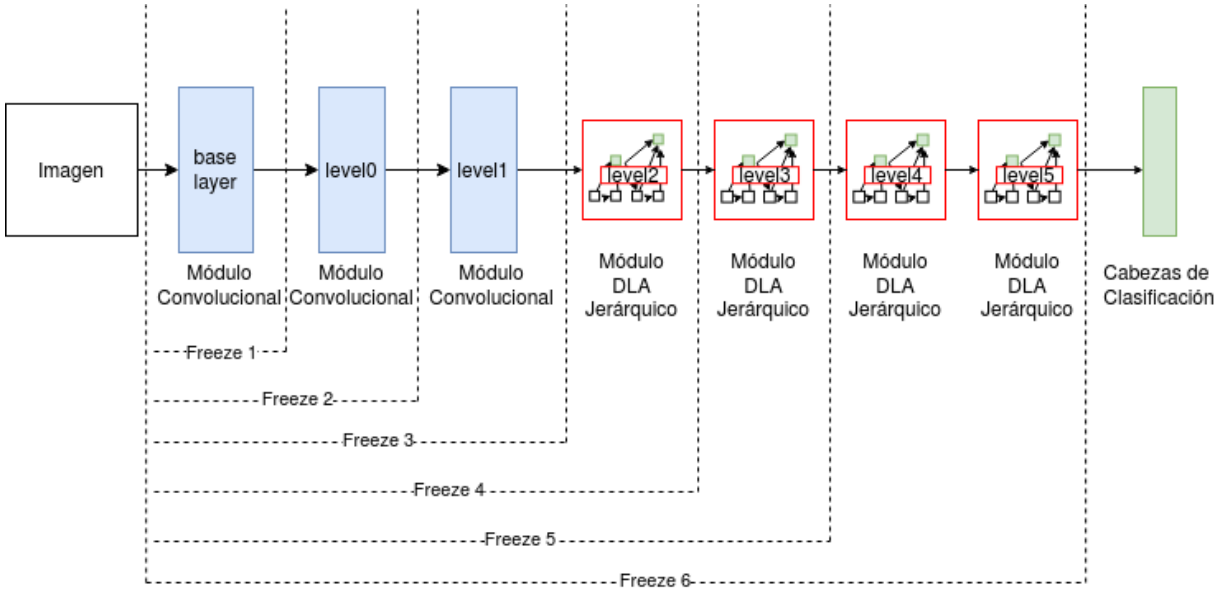


Figura 3.7: Pruebas de *finetuning* con diferentes niveles de congelamiento de la red DLA.

Los experimentos con congelamiento de la red DLA van progresando desde el congelamiento del primer módulo convolucional (*base layer*), hasta abarcar la totalidad de la red *backbone* (*base layer* hasta *level5*). No se incluye un nivel de congelamiento intermedio entre los módulos *level4* y *level5* dado que el nivel de interconexión de la red en esta zona es muy alto. Al igual que en el caso de Hourglass, se denominan *Freeze 1* hasta *Freeze 6* cada uno de los grados de congelamiento de la red.

3.4. Evaluación de Modelos de Detección de Pose Humana

Cada uno de los modelos Centernet y de comparación entrenados, se evalúan en un conjunto de 200 imágenes térmicas anotadas manualmente. Estas imágenes no se utilizan para entrenar los modelos. Por lo mismo, son una oportunidad para ver la precisión real y capacidad de generalización de los modelos.

3.4.1. *Object Keypoint Similarity*

Para el cálculo de las métricas de evaluación, se utiliza una herramienta, similar a *Intersection Over Union* (IOU), llamada *Object Keypoint Similarity* (OKS). Esta métrica, fue introducida por el equipo de desarrollo de la base de datos COCO [21]. Permite extender las nociones de las métricas *Average Precision* (AP) y *Average Recall* (AR), utilizadas en el contexto de detección de objetos, a la detección de *keypoints*.

Recordar, que para una persona de una imagen, se tienen anotaciones de *keypoints* de la forma: $[x_1, y_1, v_1, ..., x_k, y_k, v_k]$, donde x, y son las locaciones de las articulaciones, y v el grado de visibilidad de estas. Además, cada persona anotada tiene un factor s de escala, que se calcula como la razón entre el área del *ground truth bounding box* y la imagen. Los modelos, a su vez, predicen las locaciones $\hat{x}, \hat{y}$ de las articulaciones, y un puntaje general para la confianza en la predicción.

Considerando lo anterior, la métrica OKS computa las distancias d_i entre cada articulación predicha y su anotación real. Estas distancias las combina junto al factor s de escala y una constante k_i, asociada a cada tipo de articulación. Estas constantes k_i, fueron calculadas empíricamente por el equipo de COCO [21] para cuantificar la varianza en la anotación de los distintos *keypoints* de un humano. Por ejemplo, dos personas pueden anotar las caderas en lugares diferentes para una misma persona en una imagen. Todo lo anterior, conduce a la ecuación 3.1, la cual considera solo aquellos *keypoints* que se encuentran dentro de la imagen (i.e $v_i > 0$). El valor de OKS varía entre 0 y 1, siendo $OKS = 1$ una anotación perfecta.

$$OKS = \frac{\sum_i e^{\frac{-d_i^2}{2s^2k_i^2}} \delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{3.1}$$

Las métricas de *average recall* (AR) y *average precision* (AP) de un modelo de detección de pose humana, se calculan en base a la métrica OKS. Se definen umbrales, de puntajes OKS, sobre los cuales se considera una detección como correcta. Por ejemplo, estableciendo un umbral de $OKS = 0.5$, una predicción que obtenga un puntaje sobre este valor se consideraría como correcto, en este caso. Definiendo umbrales mayores de OKS, se puede interpretar como que el criterio de selección de casos positivos es más estricto.

Tabla 3.2: Valores de constantes k_i, de cada tipo de articulación, para el cálculo de OKS.

Articulación	k_i
cadera	0.107
tobillos	0.089
rodillas	0.087
hombros	0.079
codos	0.072
muñecas	0.062
orejas	0.035
nariz	0.026
ojos	0.025

3.4.2. *Average Precision* (*AP*)

La precisión se calcula de acuerdo a la ecuación 3.2, donde N_{TP} es el número de detecciones con un OKS mayor o igual al umbral definido. Por otro lado, N_T es el número total de detecciones hechas. La métrica de precisión muestra cuantas predicciones son verdaderamente reales según el criterio del umbral OKS definido, de la cantidad total de predicciones hechas. Permite evaluar la capacidad del modelo para detectar la pose humana solamente a entidades que sean personas dentro de la imagen.

$$Precision = \frac{N_{TP}}{N_T} \tag{3.2}$$

Siguiendo la literatura [21, 36, 29], se calculan 5 categorías de precisión promedio. La más importante de ellas, llamada AP, está compuesta por el promedio de la precisión sobre 11 umbrales distintos: $OKS = [0.5, 0.55, ..., 0.90, 0.95]$. A su vez, AP^M y AP^L corresponden al promedio de precisiones utilizando los mismos umbrales que AP pero distinguiendo entre personas de mediano y largo tamaño, relativo a las dimensiones de la imagen, respectivamente. Por último, AP^{50} y AP^{75} corresponden al promedio de precisión para umbrales OKS de $0,50$ y $0,75$, respectivamente.

3.4.3. *Average Recall* (*AR*)

El *recall* se calcula de acuerdo a la ecuación 3.3, donde N_{TP} es el número de detecciones con un OKS mayor o igual al umbral definido. N_P, por su parte, es la cantidad de ejemplos positivos totales. Intuitivamente, esta métrica permite evaluar que tan bueno es el modelo para detectar los casos que son realmente positivos. Modelos con una alta tasa de falsos negativos desempeñan mal en esta métrica. La métrica general de AR, se calcula promediando el *recall* sobre 11 umbrales distintos: $OKS = [0.5, 0.55, ..., 0.90, 0.95]$. Adicionalmente, al igual que el AP, se calculan AR^M y AR^L para objetos de distinta escala, junto a $AR^{0.5}$ y $AR^{0.75}$ para distintos umbrales mínimos de OKS.

$$\text{Recall} = \frac{N_{TP}}{N_P} \tag{3.3}$$

3.4.4. Evaluación a Nivel de *Keypoint*

Adicional a las métricas AP y AR, que cuantifican la precisión y el *recall* como un promedio de las detecciones de los diferentes *keypoints*, se calcula el desempeño de los modelos a nivel de cada *keypoint*. Para esto, es necesario quitar la sumatoria de la ecuación 3.1 y promediar la precisión y el *recall* de las detecciones sobre todas las imágenes para cada *keypoint* de manera independiente. Esto, permite comprender que tan precisos son los modelos al predecir las ubicaciones de cada *keypoint* por separado. Por ejemplo, la detección de los hombros de una persona puede ser más precisa que la detección de los ojos.

3.4.5. *Frames Per Second* (FPS)

La métrica FPS, permite evaluar la velocidad de inferencia de los modelos, al calcular la cantidad de imágenes que el modelo es capaz de procesar en un segundo. Esto es importante, puesto que la intención es implementar un sistema de detección de pose humana en tiempo real. En general, un modelo de detección se dice que funciona en tiempo real [42] si $FPS \geq 10$. Para la medición de esta métrica, se calcula el tiempo, en segundos, que le toma al algoritmo procesar una imagen, y se invierte el resultado.

3.4.6. Evaluación Cualitativa

Para complementar la evaluación de los modelos, se analiza el desempeño de estos de manera cualitativa. Esto es, escoger del conjunto de prueba una cantidad fija de imágenes detectadas y analizar visualmente como son las detecciones. Además, el modelo se utiliza para predecir la pose de personas en imágenes capturadas con la cámara térmica FLIR FC-690 S del laboratorio, lo cual, dada la falta de anotaciones en este caso, es evaluado de esta manera.

3.5. Implementación del *Pipeline* Final de Detección

El mejor modelo entrenado, se utiliza como parte del sistema de detección de pose humana en imágenes térmicas final. Este sistema, esquematizado en la figura 3.8, engloba primero la imagen térmica a detectar, el procesamiento de ella mediante el modelo de detección de pose humana y el reporte de resultados. Para la implementación de este sistema, se construye un programa de demostración en Python, que permite especificar la imagen o video de entrada y producir las detecciones. Los resultados de las detecciones se entregan de dos formas: como imágenes con los esqueletos detectados, y como datos más específicos, que incluyen las coordenadas de los *bounding boxes*, la ubicación de cada *keypoint* en la imagen y el puntaje de confianza de cada detección (*confidence score*).

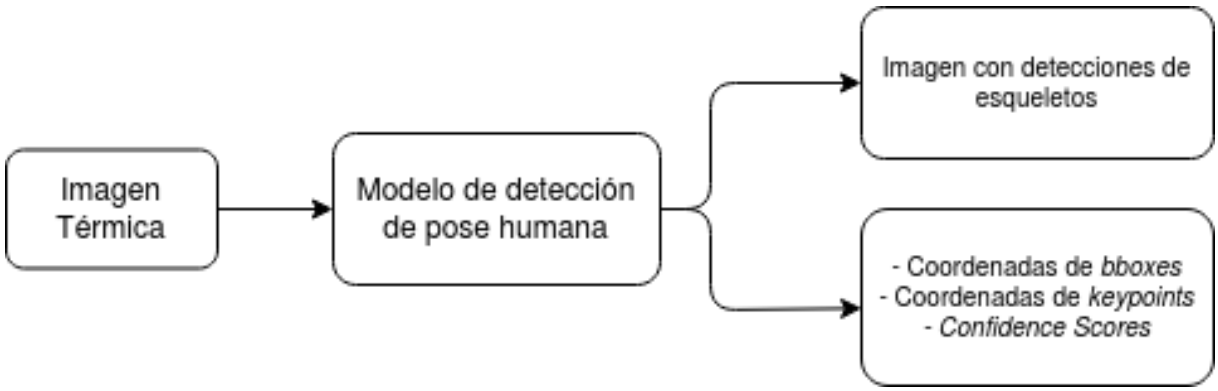


Figura 3.8: *Pipeline* del sistema de detección de pose humana sobre imágenes térmicas.

Capítulo 4

Resultados y Análisis

4.1. Transformación de Imágenes de Color al Dominio Térmico

A continuación, se presentan los resultados de transformación de imágenes de color al dominio térmico. Se incluye, también, un análisis del desempeño de cada modelo y observaciones con respecto a los experimentos. Recordar que la evaluación de los modelos de transformación son estrictamente cualitativas. Se presentan ejemplos de transformación sobre el *dataset* AAU VAP Trimodal [41] en la figura 4.1, el cual tiene pares de imágenes a color y térmicas.

(a) Color (b) pix2pix (c) CycleGAN (d) ThermalGAN (e) Térmica Real

Figura 4.1: Ejemplos de transformación de imágenes de color, de la base de datos AAU VAP Trimodal [41], al dominio térmico.

4.1.1. Pix2pix

La transformación realizada por el modelo pix2pix [3] entrenado, se muestra en la columna (b) de la figura 4.1. Como se ve ahí, el modelo realiza una transformación satisfactoria de las zonas de la imagen que no contienen humanos, pero falla en las partes que sí. Si bien logra capturar algunas características relevantes de la imagen térmica real, como por ejemplo la cabeza de la persona en la tercera fila, la forma de los humanos no es coherente. En particular, es difícil distinguir apropiadamente las formas de las personas en las imágenes de la primera y segunda fila. A raiz de estos resultados, se puede decir que la transformación que realiza este modelo no es exitosa.

4.1.2. CycleGAN

Para el caso de CycleGAN [13], los resultados de transformación se muestran en la columna (c) de la figura 4.1. En esta, se puede evidenciar que la transformación se asemeja más a una imagen en escala de gris, de la imagen a color, que una imagen térmica propiamente tal. Asimismo, se aprecia que en las imágenes de la segunda y tercera fila, aparecen manchas negras en las zonas superiores de las imágenes. Una posible razón de esto, es que el fondo de la imagen a color activa una señal en la red para oscurecer esa zona de acuerdo a lo que aprendió en el entrenamiento. Como consecuencia, al igual que el modelo pix2pix [3], se puede decir que la transformación que realiza este modelo tampoco es exitosa.

4.1.3. ThermalGAN

Por último, se tiene la transformación realizada por el modelo ThermalGAN [50], representada por la columna (d) de la figura 4.1. Como fue mencionado en la sección de metodología, se realizaron pruebas con varias máscaras de segmentación térmica artificiales diferentes. Finalmente, se concluyó que utilizar un valor de 12 para los píxeles correspondientes a las personas y un valor de 1 para los píxeles del fondo, permite lograr los mejores resultados. Esto es contra intuitivo, considerando que generalmente el fondo está a una temperatura menor que las personas. Sin embargo, se nota que la imagen térmica absoluta obtenida por este método ($\hat{B}$ en la figura 2.21), se aleja más de la imagen térmica real que la imagen de contrastes térmicos relativos ($\hat{R}$), puesto que fuerza notablemente el polígono de segmentación sobre la imagen. Luego, dado que $\hat{R}$ posee una distribución de píxeles relativamente invertida con respecto a $\hat{B}$, se alcanzan mejores resultados donde el fondo está a una intensidad mayor que las personas. Una demostración más detallada de esto se presenta en el anexo B.

Continuando, se ve que en la columna (d) de la figura 4.1 se alcanza una transformación más fidedigna a la imagen térmica real que los otros métodos. La figura de los humanos se mantiene y resalta sobre el fondo de manera similar a la imagen térmica real. No obstante, el uso de la máscara de segmentación térmica homogeneiza las texturas de las personas, como se aprecia al comparar las texturas de la imagen de la tercera fila con la imagen térmica real. Adicionalmente, se introduce una serie de artefactos en el fondo que alejan la transformación de la relativa oscuridad que se presenta en la imagen térmica real. Aun con todo, ThermalGAN [50] se posiciona como el mejor método de transformación de imágenes del dominio de color al dominio térmico, de entre los estudiados.

4.1.4. Comparación de Traducción sobre Imágenes de COCO

De manera adicional a los experimentos presentados en la sección anterior, se presenta en la figura 4.2 algunos ejemplos de transformación sobre la base de datos COCO [21]. Dado que, en este caso, no se dispone de las imágenes térmicas reales, no se puede comprobar a ciencia exacta la transformación de los modelos. Sin embargo, los resultados que se presentan en la figura permiten ratificar las observaciones indicadas en las secciones pasadas. Nuevamente, se puede apreciar como ThermalGAN [50] es el modelo que alcanza una transformación más coherente. Esto es particularmente cierto en las imágenes de la segunda y tercera fila. En ellas, detecta que los píxeles de la ropa de los sujetos están a un nivel inferior que las zonas de piel expuestas, lo cual es algo que se puede esperar de las imágenes térmicas reales.

(a) Color (b) pix2pix (c) CycleGAN (d) ThermalGAN

Figura 4.2: Ejemplo de transformación de imágenes de COCO [21] al dominio térmico.

4.2. Descripción de Base de Datos con Imágenes Térmicas Manualmente Anotadas

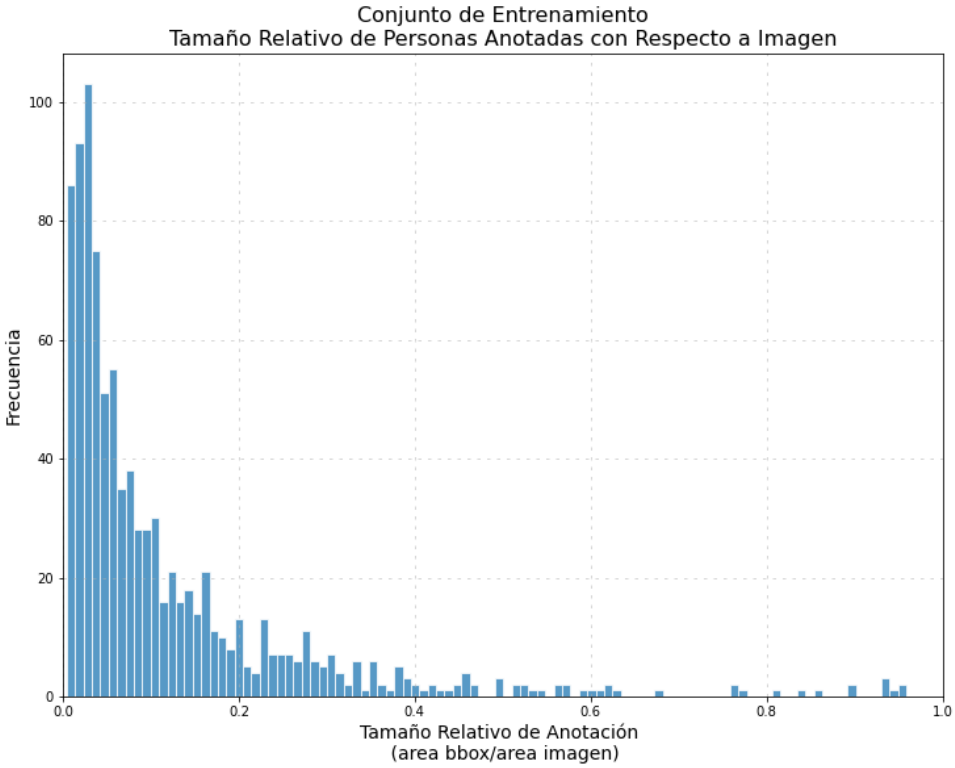
En total, se anotaron 800 imágenes térmicas, las cuales contienen 1218 personas en total. Este valor se desglosa en 927 y 291 personas para el conjunto de entrenamiento y prueba, respectivamente. Con ello, se tiene un promedio de 1.545 personas por imagen anotadas para el conjunto de entrenamiento y de 1.455 personas por imagen para el conjunto de prueba.

Tabla 4.1: Frecuencia de anotación de cada *keypoint*, para conjuntos de entrenamiento y prueba de imágenes térmicas manualmente anotadas.

Keypoint	Entrenamiento	Prueba
nariz	543	194
ojo izquierdo	404	146
ojo derecho	411	165
oreja izquierda	392	136
oreja derecha	392	163
hombro izquierdo	767	236
hombro derecho	765	250
codo izquierdo	673	216
codo derecho	655	242
muñeca izquierda	572	196
muñeca derecha	573	211
cadera izquierda	739	223
cadera derecha	759	241
rodilla izquierda	759	239
rodilla derecha	735	251
tobillo izquierdo	706	229
tobillo derecho	724	238

En cuanto a los *keypoints* anotados, se tiene la tabla 4.1. En esta tabla, se puede ver que hay una menor cantidad de *keypoints* de la cara anotadas, con respecto al resto del cuerpo. Este antecedente es cierto para ambos conjuntos. La razón principal es que muchas de las imágenes muestran personas que dan la espalda a la cámara. Asimismo, se ve que hay una baja cantidad de anotaciones de muñecas. Lo anterior se explica considerando que, en general, las manos de las personas sufren oclusión de manera más fácil que otros *keypoints* del cuerpo humano. De hecho, si se analiza la progresión en cantidad de anotaciones desde el hombro de un lado a la muñeca correspondiente, se nota que las anotaciones disminuyen progresivamente.

Adicionalmente, en los gráficos de la figura 4.3 se pueden ver histogramas con las frecuencias de los tamaños relativos de anotaciones de las personas. Estos tamaños relativos, se calculan dividiendo el área de los *bounding boxes* de las personas con el área de la imagen que las contiene. Los gráficos muestran que la mayoría de las personas anotadas, cubren menos de un 10 % del área de la imagen. Aun así, se dispone de anotaciones de mayor tamaño, algunas incluso cubriendo más del 80 % de la imagen. La distribución de los histogramas son similares para el conjunto de entrenamiento y el de prueba.



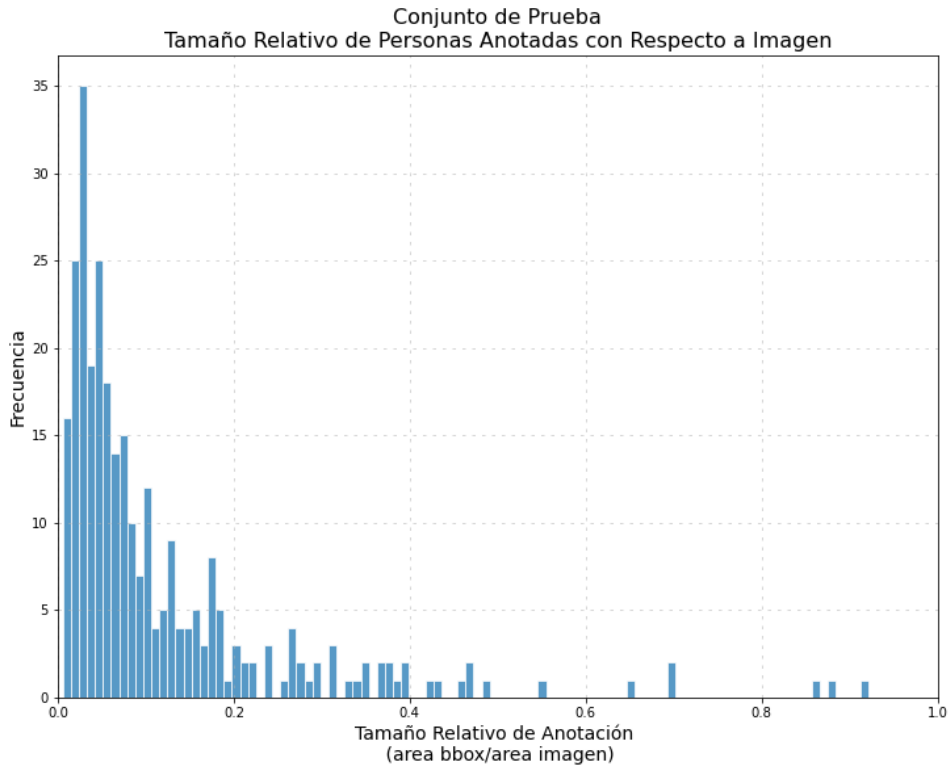


Figura 4.3: Tamaño relativo de anotaciones de personas con respecto al tamaño de la imagen.

En las imágenes de la figura 4.4, se pueden ven ejemplos de anotaciones de personas. Los puntos con bordes negros son aquellos *keypoints* anotados con visibilidad 1 (ocluidos) y los otros con visibilidad 2 (visibles). Los rectángulos corresponden a los *bounding boxes* de las personas.

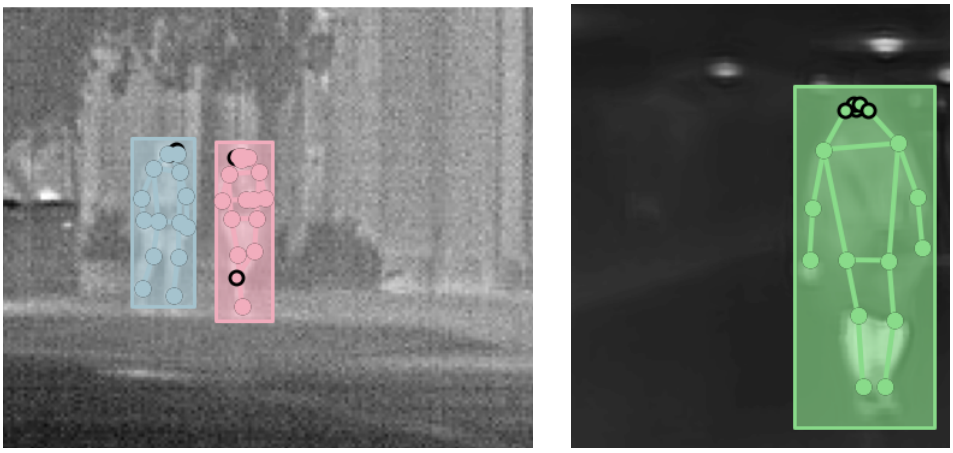


Figura 4.4: Ejemplo de anotaciones de personas sobre imágenes térmicas.

4.3. Desempeño de Modelos Entrenados sobre Dataset COCO

4.3.1. Imágenes Térmicas Artificiales

Utilizando el modelo ThermalGAN [50], el cual tuvo los mejores resultados de transformación de imágenes al dominio térmico, se transforman 2000 imágenes aleatorias de la base de datos *COCO*. Estas imágenes se utilizan para entrenar el modelo CenterNet, con *backbone* DLA-34 [47], empleando los parámetros correspondientes al régimen 1x de la tabla 3.1. Los resultados de la evaluación sobre el conjunto de prueba de imágenes térmicas, se ve en las tablas 4.2 y 4.3, para las métricas AP y AR, respectivamente. Se incluye, también, la evaluación del modelo original entrenado por los autores de [36] sobre COCO a color.

Tabla 4.2: AP de CenterNet (DLA) entrenado sobre conjunto de imágenes térmicas artificiales de COCO.

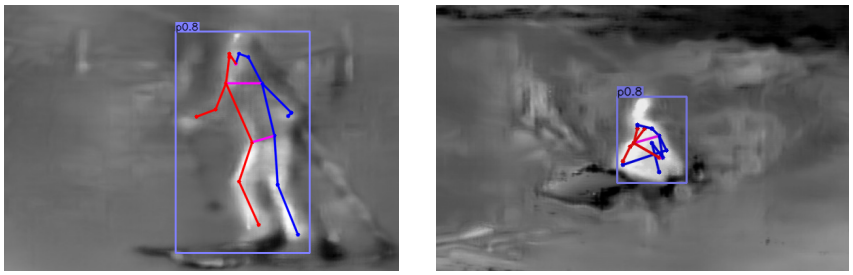
Tipo de imagen	Épocas	AP	$AP_{0.5}$	$AP_{0.75}$	AP_M	AP_L
color	140	0,550	0,786	0,664	0,369	0,673
térmica artificial	70	0,003	0,012	0,001	0,001	0,006
térmica artificial	140	0,020	0,044	0,013	0,021	0,028

Tabla 4.3: AR de CenterNet (DLA) entrenado sobre conjunto de imágenes térmicas artificiales de COCO.

Tipo de imagen	Épocas	AR	$AR_{0.5}$	$AR_{0.75}$	AR_M	AR_L
color	140	0,646	0,872	0,777	0,520	0,748
térmica artificial	70	0,049	0,142	0,020	0,008	0,082
térmica artificial	140	0,082	0,189	0,068	0,027	0,127

De las tablas anteriores, se puede concluir que el modelo entrenado sobre las imágenes térmicas artificiales no alcanza un buen desempeño sobre el conjunto de evaluación. El AP del modelo se mantiene bajo un 5 %, incluyendo en la evaluación con $OKS > 0.5$, lo cual demuestra la baja precisión de las predicciones. Algo similar sucede para el AR, donde se alcanzan métricas más altas pero aún considerablemente más bajas que el modelo original entrenado sobre COCO a color. Aun cuando se aprecia una diferencia significativa en el desempeño de los modelos de las épocas 70 y 140, esta no es suficiente como para justificar que el entrenamiento va a converger a un valor razonable.

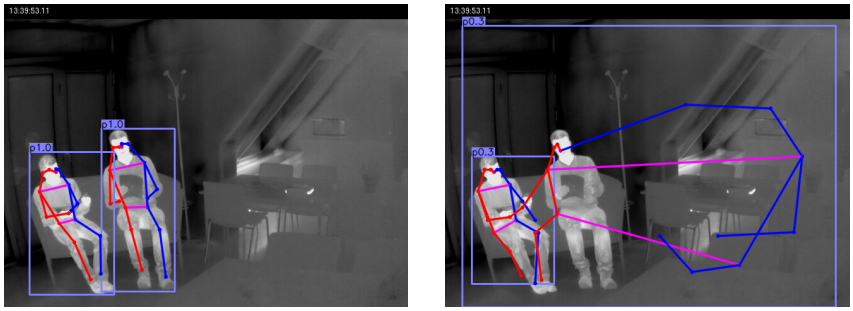
Ahora bien, si se considera que la comparación se está realizando con un modelo entrenado en la totalidad de la base de datos COCO, se puede justificar que el modelo entrenado sobre imágenes térmicas artificiales no dispuso de suficientes ejemplos para generalizar buenas predicciones. Sin embargo, este argumento se cae al examinar que las imágenes transformadas son, supuestamente, más cercanas al dominio térmico que su contra-par a color. Luego, este debería tener un desempeño razonable aún con una menor cantidad de imágenes utilizadas para el entrenamiento.



(a) Detección "correcta" (b) Detección errónea

Figura 4.5: Ejemplos de detección del modelo CenterNet sobre imágenes térmicas artificiales.

Con el objetivo de analizar más en detalle los resultados, se presenta en la figura 4.5 algunos ejemplos de detección sobre un subconjunto de imágenes de evaluación de COCO [21], transformadas con ThermalGAN [50]. Como se ve ahí, en la imagen de la izquierda se evidencia una detección, si bien no perfecta, altamente precisa para la persona retratada. Por otro lado, en la figura de la derecha, se discierne una detección completamente errónea. El problema es que la persona representada perdió completamente la coherencia de forma y los detalles de textura que caracterizan a un humano. Luego, se prevé que la repetición de esta condición, para las imágenes de entrenamiento, pudo incidir en un entrenamiento erróneo sobre figuras con formas y texturas no-humanas. Es decir, la distribución de las imágenes térmicas artificiales se aleja significativamente del dominio térmico, contra lo deseado.



(a) Modelo original

(b) Modelo entrenado sobre imágenes térmicas artificiales

Figura 4.6: Ejemplos de detección del modelo CenterNet, entrenado sobre imágenes transformadas y entrenado sobre imágenes a color, sobre imágenes térmicas reales.

En las imágenes de las figuras 4.6, se puede ver un ejemplo de detección sobre una imagen térmica real. Notar como el modelo original (a) predice, con alta precisión, la pose de las personas retratadas, mientras que el otro modelo (b) predice bien solo una de ellas y falla notablemente en la otra. Esto puede ser una indicación de que el entrenamiento sobre las imágenes térmicas artificiales guió al modelo a ajustarse a patrones que no se manifiestan en el dominio térmico real. Debido a esto, así como para ahorrar recursos computacionales y tiempo, se decide por no seguir los experimentos con los modelos entrenados sobre imágenes térmicas artificiales.

4.3.2. Imágenes a Color

En la tabla 4.4, se muestran las métricas de AP y AR, obtenidas de la evaluación sobre las imágenes térmicas con los modelos entrenados sobre COCO en color. Esta tabla incluye también los modelos originales, entrenados por los autores de CenterNet [36]. Es fácil ver, al comparar los modelos del régimen 1x, que aquellos entrenados en este trabajo son superiores a los originales. En el caso de HRNet , dado que la implementación original de CenterNet no incorpora esta arquitectura, no se tiene un punto de comparación.

Tabla 4.4: Evaluación de modelos entrenados con imagenes de COCO a color. Incluye también modelos originales de los autores de CenterNet [36].

Backbone	Régimen	AP	AR
DLA-34 (original)	1x	0,429	0,517
Hourglass-104 (original)	1x	0,465	0,568
DLA-34 (original)	3x	0,446	0,535
Hourglass-104 (original)	3x	**0,501**	0,584
DLA-34	1x	0,461	0,558
Hourglass-104	1x	0,471	0,570
HRNet-W32	1x	0,441	**0,601**

Otra observación es que los modelos, con arquitectura Hourglass, tienen mejor desempeño que los otros modelos en cuanto a precisión, lo cual se condice con lo señalado en el *paper* de CenterNet [36]. En el caso de DLA, se puede notar que el modelo entrenado es superior al original en ambas métricas, considerando también el modelo entrenado con régimen 3x. El modelo con la arquitectura HRNet, por su parte, alcanza el mayor *recall*. Notablemente, todos los modelos evaluados están por sobre el 42 % de AP y el 50 % de AR. Esto sugiere que el entrenamiento sobre imágenes a color acerca a los modelos a una detección de pose favorable en imágenes térmicas.

4.3.3. Imágenes en Escala de Gris

Por el lado de los entrenamientos con imágenes en escala de gris, evaluados sobre el conjunto de prueba de imágenes térmicas, se llega a los resultados presentados en la tabla 4.5. En este caso, DLA sobrepasa a los otros modelos en cuanto a *precision*. HRNet, al igual que con los modelos entrenados a color, es aquel que alcanza el mayor *recall*. Dados estos resultados, se puede decir que los modelos con este entrenamiento alcanzan desempeños favorables en cuanto a estimación de pose en el dominio térmico.

Tabla 4.5: Evaluación de modelos entrenados con imagenes de COCO en gris.

Backbone	Régimen	AP	AR
DLA-34	1x	**0,485**	0,582
Hourglass-104	1x	0,484	0,575
HRNet-W32	1x	0,400	**0,589**

4.3.4. Comparación

Analizando los resultados de los modelos 1x desplegados en los apartados anteriores, se puede ver que aquellos con arquitecturas Hourglass y DLA, entrenados sobre imágenes en escala de gris, sobrepasan aquellos con entrenamiento sobre imágenes a color. En el caso de DLA, la diferencia es de $\approx 2\,\%$ en AP y AR. Una situación similar, pero más atenuada, se da en el modelo Hourglass, con una diferencia de $\approx 1\,\%$ y $\approx 0.5\,\%$ en AP y AR, respectivamente. Estas diferencias, si bien parecen pequeñas, son importantes en el contexto de la tarea que se intenta resolver. Por ejemplo, un 2 % de mejora en AP, se traduce en 4 imágenes más detectadas precisamente en el conjunto de evaluación reducido que se ocupa aquí (200 imágenes). Para bases de datos más grandes, esta diferencia en precisión abarca, evidentemente, una mayor cantidad de personas correctamente detectadas.

El fenómeno manifestado en el parráfo anterior, no se cumple para CenterNet con *backbone* HRNet. En este caso, el modelo entrenado con imágenes a color supera a aquel entrenado con imágenes en escala de gris. Esto, por $\approx 4\,\%$ en AP y $\approx 1\,\%$ en AR.

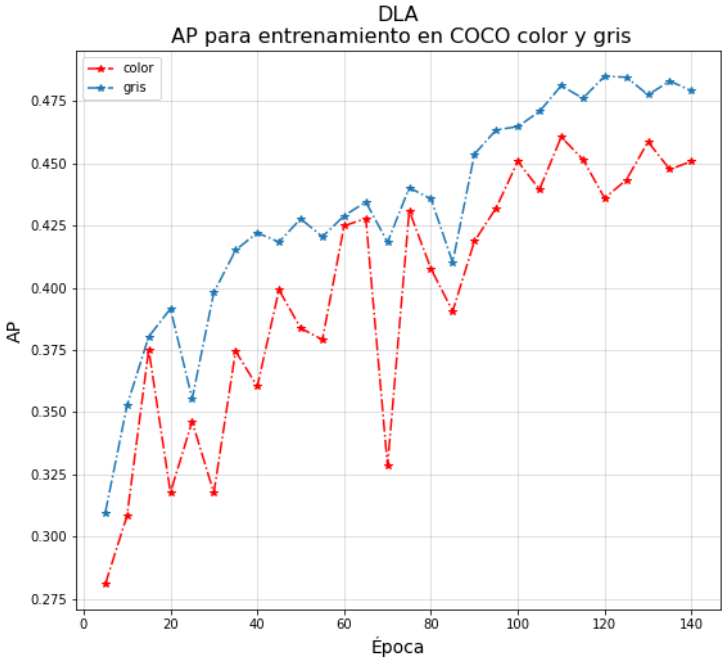


Figura 4.7: Desempeño, por época, de CenterNet DLA entrenado en COCO y evaluado sobre imágenes térmicas.

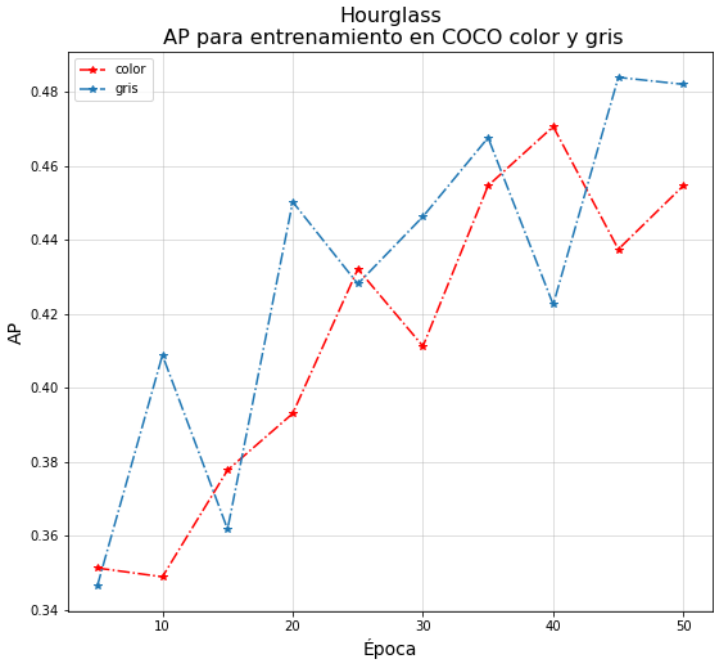


Figura 4.8: Desempeño, por época, de CenterNet Hourglass entrenado en COCO y evaluado sobre imágenes térmicas.

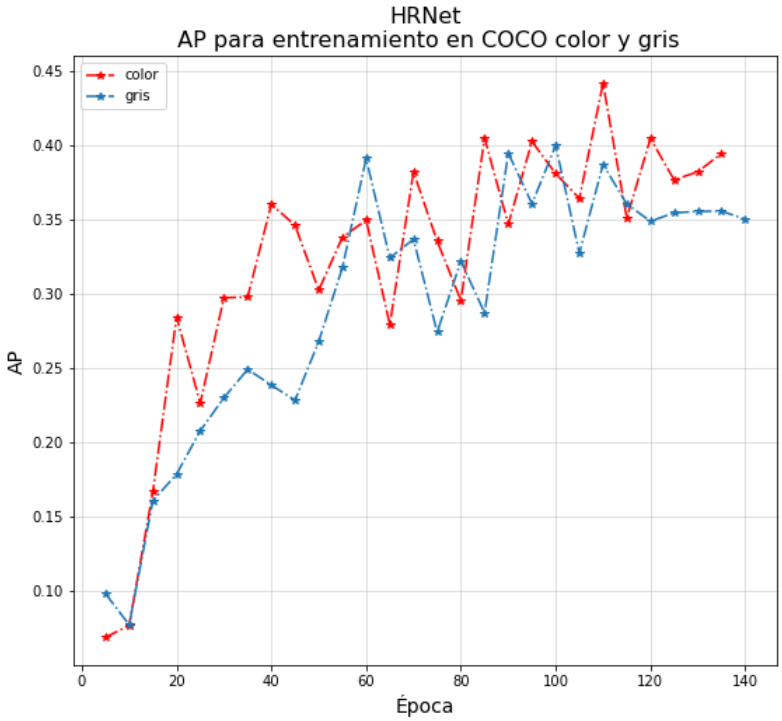


Figura 4.9: Desempeño, por época, de CenterNet HRNet entrenado en COCO y evaluado sobre imágenes térmicas.

Con esto, se puede deducir que cada arquitectura responde de manera diferente al dominio de las imágenes que se utilizan para pre-entrenar. Hourglass y DLA responden de mejor manera a un entrenamiento con imágenes en escala de gris, mientras que HRNet alcanza mejor desempeño con entrenamiento en imágenes a color. Esto también se puede visualizar en los gráficos de las figuras 4.7, 4.8 y 4.9. Aquí se ve que, en la mayoría de las épocas, los modelos DLA y Hourglass entrenados con imágenes en escala de gris sobrepasan al resto. Por otro lado, en HRNet las curvas están más entrelazadas y, finalmente, vence el entrenamiento con imágenes a color.

En las tablas 4.6 y 4.7, se adjuntan los resultados de AP y AR completos para los mejores modelos del régimen 1x entrenados. En la primera tabla, se puede evidenciar que la arquitectura con *backbone* DLA genera los mejores resultados en *precision*, salvo al evaluar solo los objetos de mediano tamaño, donde HRNet es superior. Los resultados de DLA son seguidos por Hourglass, por poca diferencia.

Tabla 4.6: AP de modelos entrenados sobre COCO, sobre el conjunto de evaluación de imágenes térmicas.

Backbone	Pre-entrenamiento	AP	$AP_{0.5}$	$AP_{0.75}$	AP_M	AP_L
DLA-34	gris	**0,485**	**0,735**	**0,558**	0,382	**0,595**
Hourglass-104	gris	0,483	0,725	0,550	0,392	0,581
HRNet-W32	color	0,441	0,652	0,512	**0,401**	0,512

Analizando los resultados de la métrica AR, se puede ver que HRNet es el superior en todas las evaluaciones, con excepción de las detecciones con $OKS > 0.5$. Este modelo se desempeña pobremente en *precision*, pero supera a todos en *recall*. En otras palabras, acierta medianamente bien a la pose de las personas contenidas en el conjunto de evaluación, pero incide con ello en una mayor cantidad de falsos positivos. Esta situación se repite con los modelos evaluados a lo largo del trabajo, como se puede ver al comparar las tablas 4.7 y 4.6, donde el *recall* es siempre mayor a la precisión.

Tabla 4.7: AR sobre el conjunto de evaluación de imágenes térmicas, de modelos entrenados sobre COCO.

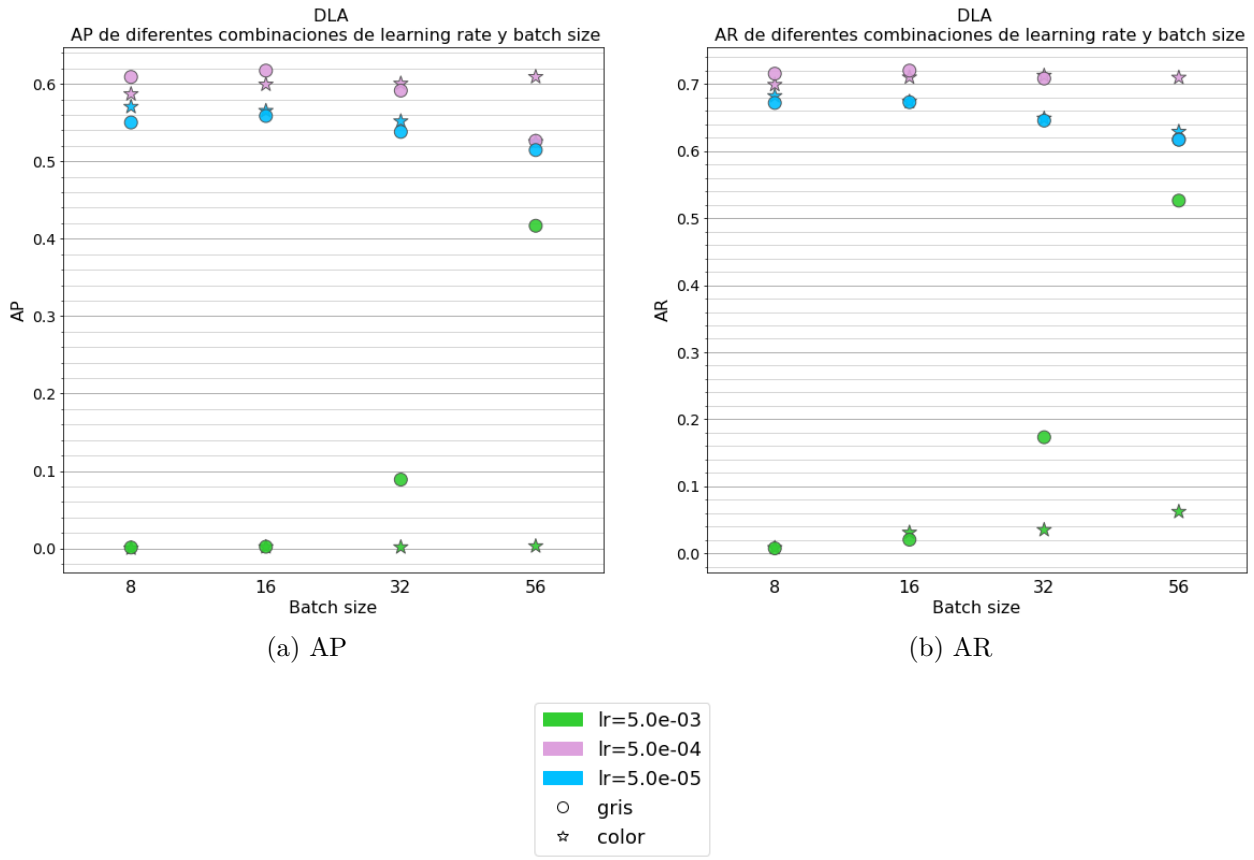
Backbone	Pre-entrenamiento	AR	$AR_{0.5}$	$AR_{0.75}$	AR_M	AR_L
DLA-34	gris	0,582	**0,836**	0,650	0,519	0,655
Hourglass-104	gris	0,575	0,814	0,643	0,530	0,640
HRNet-W32	color	**0,601**	0,832	**0,693**	**0,540**	**0,670**

4.4. Resultados Finetuning

A continuación, se presentan los resultados de los diferentes experimentos de *finetuning* realizados sobre los modelos. El *finetuning*, se realiza sobre las 600 imágenes térmicas del conjunto de entrenamiento y se evalúa cada experimento en las 200 imágenes del conjunto de prueba. Es importante destacar, que los experimentos con distintas combinaciones de *learning rate* y *batch size* y con distintos *learning-rate schedules* se realizaron sobre los modelos CenterNet de régimen 1x pre-entrenados con imágenes a color y en escala de gris. De ahí en adelante, los experimentos se realizan sobre modelos pre-entrenados en régimen 3x sobre la base de datos COCO, utilizando el tipo de imagen que produjo el mejor desempeño después del *finetuning*.

4.4.1. Diferentes Combinaciones de *Learning Rate* y *Batch Size*

En la figura 4.10, se resumen los resultados de los experimentos con diferentes combinaciones de *learning rate* y *batch size*, realizados para CenterNet con arquitectura DLA. En estos gráficos, se puede ver que el comportamiento de ambas métricas es similar para los distintos experimentos realizados. Por otro lado, el *finetuning* sobre las imágenes térmicas redujo la diferencia en el desempeño de los modelos pre-entrenados con imágenes a color con respecto a aquellos pre-entrenados con imágenes en escala de gris.

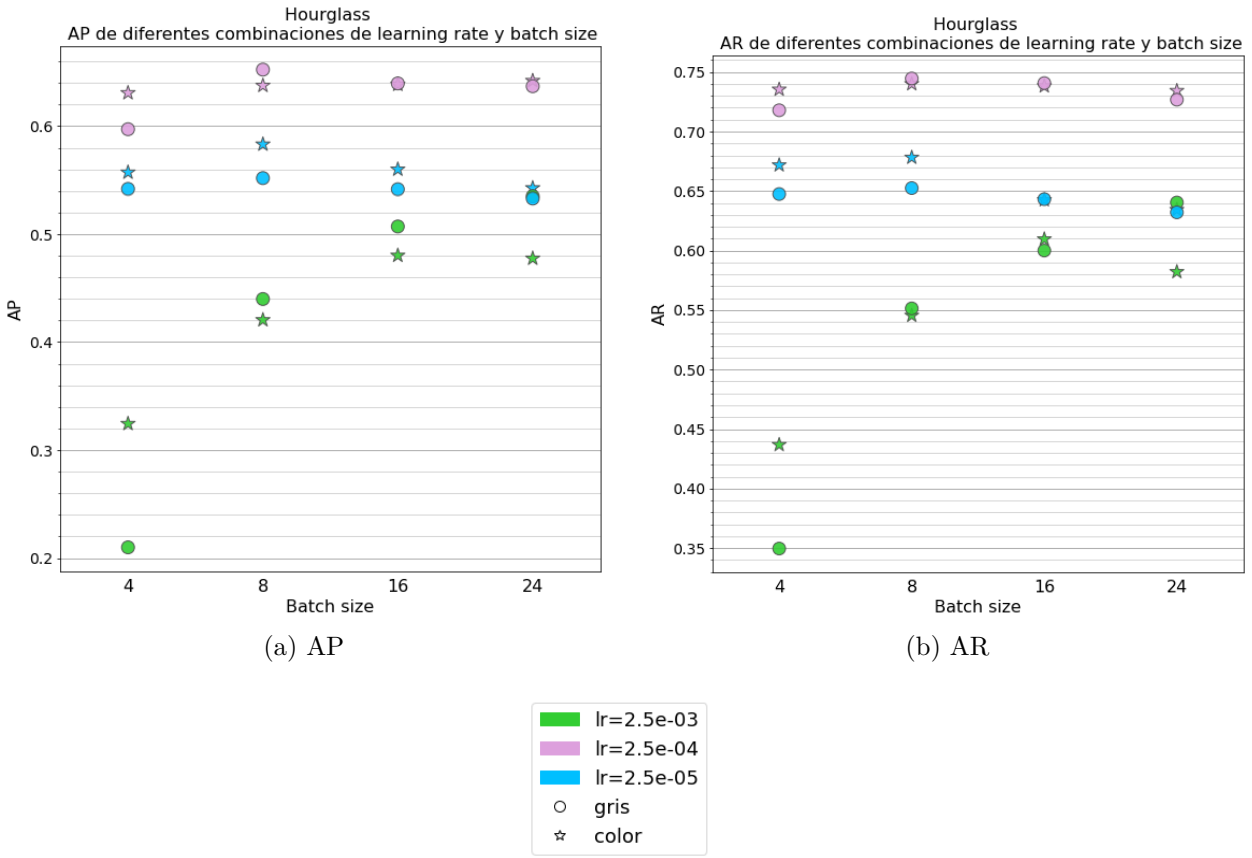


(a) AP (b) AR

Figura 4.10: Desempeño del *finetuning* de CenterNet DLA con diferentes combinaciones de *learning rate* y *batch size*.

Asimismo, se ve que al utilizar un *learning rate* de 0.0005 se alcanzan los mejores resultados para cualquier *batch size*. Dentro de los experimentos con este *learning rate*, se puede evidenciar que para los *batch size* de 8 y 16, los modelos pre-entrenados con imágenes en escala de gris son superiores, mientras que para *batch size* de 32 y 56, el pre-entrenamiento con imágenes a color es superior. No obstante, considerando ambas métricas, el mejor resultado se alcanza utilizando la combinación de *batch size* igual a 16 con un *learning rate* de 0.0005, con pre-entrenamiento en imágenes en escala de gris.

En el caso de los modelos con arquitectura Hourglass, se obtienen los gráficos mostrados en la figura 4.11. Nuevamente, se aprecia la importancia del *learning rate* para el desempeño de los modelos en ambas métricas. Utilizar un *learning rate* de 0.00025 lleva a los mejores resultados, para cualquier *batch size*. Considerando el *learning rate* anterior, los experimentos con *batch size* de 8 y 16 muestran una preferencia por el pre-entrenamiento con imágenes en escala de gris, mientras que los experimentos con *batch size* de 4 y 24 alcanzan los mejores desempeños con un pre-entrenamiento con imágenes a color. De todas maneras, el experimento con un *learning rate* de 0.00025 y *batch size* de 8, con pre-entrenamiento con imágenes en escala de gris, es el que muestra los mejores resultados, tanto en AP como en AR.

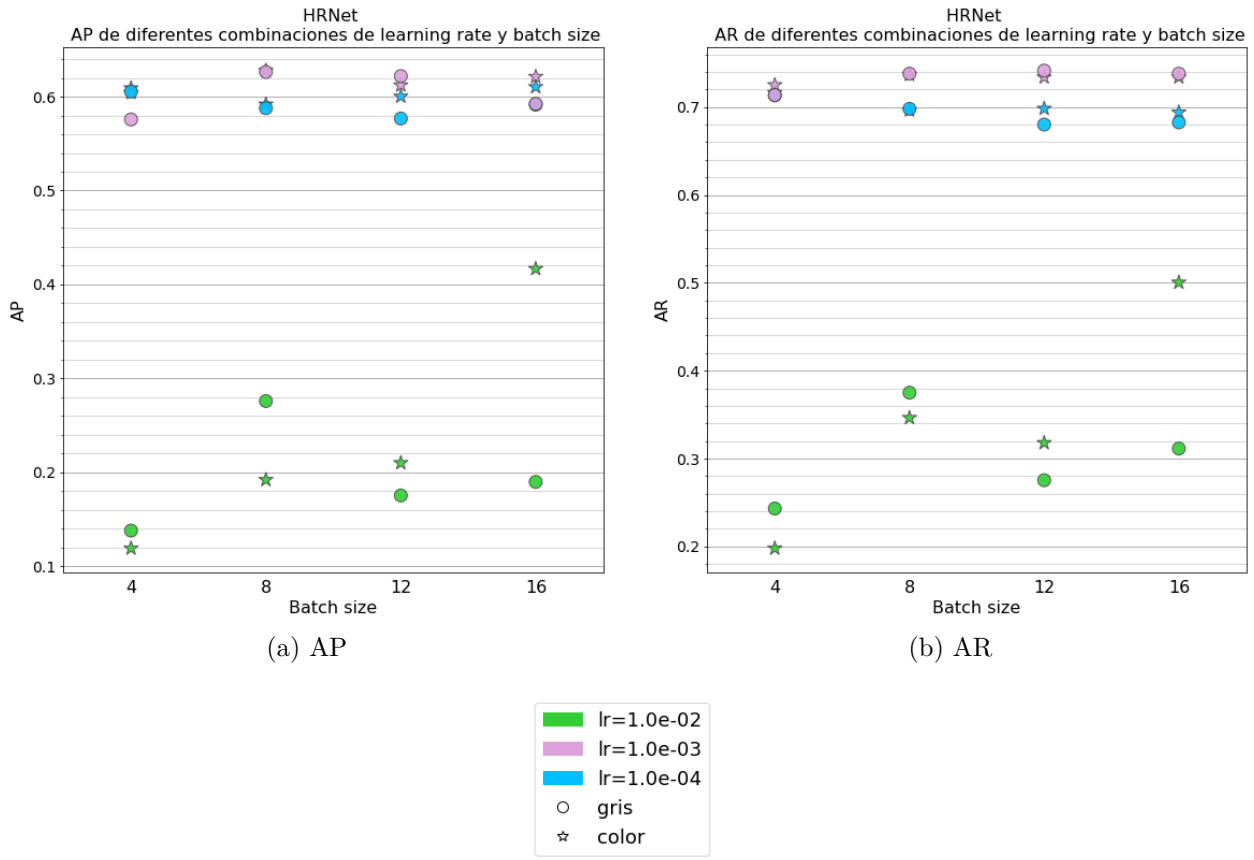


(a) AP (b) AR

Figura 4.11: Desempeño del *finetuning* de CenterNet Hourglass con diferentes combinaciones de *learning rate* y *batch size*.

Por último, para CenterNet con *backbone* HRNet, se obtienen los gráficos que se muestran en la figura 4.12. Estos demuestran que hay dos alternativas de *learning rate* que derivan en resultados cercanos: 0.001 y 0.0001. Sin embargo, utilizar un *learning rate* de 0.001 produce, en general, los mejores resultados para los distintos *batch size* estudiados. Considerando este valor para el *learning rate*, se tiene que el AP y AR de los modelos entrenados con *batch size* de 8 y 12 son altamente similares. Tomando en cuenta AP, se alcanza el mejor resultado ocupando un *batch size* de 8, con una leve diferencia entre los modelos pre-entrenados con imágenes a color y en escala de gris. Por otro lado, para la métrica AR, el mejor resultado se alcanza ocupando un *batch size* de 12 y con pre-entrenamiento con imágenes en escala de gris.

La elección de los mejores parámetros en el caso de CenterNet HRNet es difícil, considerando la similitud de los resultados. Sin embargo, se puede reducir a dos alternativas: utilizar un *batch size* de 8, con pre-entrenamiento con imágenes a color, o utilizar un *batch size* de 12, con pre-entrenamiento con imágenes en escala de gris. Si se compara métrica por métrica, el modelo pre-entrenado con imágenes a color es 0.59 % mejor que el modelo pre-entrenado con imágenes en escala de gris en AP; mientras que el último modelo es un 0.54 % mejor que el primero en AR. Esto demuestra que, con el *finetuning* sobre las imágenes térmicas, el tipo de pre-entrenamiento estudiado (con imágenes a color vs en escala de gris) pierde considerable relevancia.



(a) AP (b) AR

Figura 4.12: Desempeño del *finetuning* de CenterNet HRNet con diferentes combinaciones de *learning rate* y *batch size*.

Con el fin de resumir la exploración de parámetros presentada en los apartados anteriores, se incluye la tabla 4.8. Esta muestra las mejores combinaciones de *batch size* y *learning rate* encontradas para cada modelo y tipo de pre-entrenamiento.

Tabla 4.8: Mejores combinaciones de *learning rate* y *batch size*, para las distintas arquitecturas *backbone* y tipo de imágenes utilizadas para pre-entrenamiento en COCO.

Backbone	Pre-entrenamiento	*Batch size*	*Learning rate*
DLA-34	gris	16	0,0005
DLA-34	color	32	0,0005
Hourglass-104	gris	8	0,00025
Hourglass-104	color	16	0,00025
HRNet-W32	gris	12	0,001
HRNet-W32	color	8	0,001

4.4.2. Diferentes *Learning-Rate Schedules*

Utilizando los parámetros de *learning rate* y *batch size* especificados en la tabla 4.8, se realiza una exploración de distintos *learning rate schedules* para cada arquitectura. Los resultados de esta exploración se muestran en la tabla 4.9 para DLA, en la tabla 4.10 para Hourglass y en la tabla 4.11 para HRNet. Cada tabla muestra, destacado en negro, los experimentos con mejores resultados en cada métrica.

Tabla 4.9: Experimentos con diferentes *learning rate schedules* para CenterNet DLA.

Pre-entrenamiento	*Learning rate steps*	*AP*	*AR*
gris	-	**0,670**	0,746
gris	[35,45]	0,669	0,751
gris	[40-48]	0,653	**0,754**
color	-	0,657	0,751
color	[35,45]	0,628	0,726
color	[40-48]	0,651	0,742

Tabla 4.10: Experimentos con diferentes *learning rate schedules* para CenterNet Hourglass.

Pre-entrenamiento	*Learning rate steps*	*AP*	*AR*
gris	-	0,654	0,757
gris	[10,17]	**0,672**	0,761
gris	[15,18]	**0,672**	**0,764**
color	-	0,668	0,759
color	[10,17]	0,663	0,759
color	[15,18]	0,670	0,755

Tabla 4.11: Experimentos con diferentes *learning rate schedules* para CenterNet HRNet.

Pre-entrenamiento	*Learning rate steps*	*AP*	*AR*
gris	-	0,626	0,738
gris	[35,45]	0,584	0,754
gris	[40-48]	0,576	**0,769**
color	-	**0,628**	0,737
color	[35,45]	0,584	0,711
color	[40-48]	0,576	0,709

En la tabla 4.9, se puede ver que CenterNet con *backbone* DLA alcanza el mejor AP con pre-entrenamiento en imágenes grises y sin un *learning rate schedule*. Asimismo, el mejor AR se produce con el mismo pre-entrenamiento y con disminución del *learning rate* en las épocas 40 y 48. Por otro lado, al utilizar un *learning rate schedule* con disminución en las épocas 35 y 45, se obtiene un AP y un AR muy cercano a los valores máximos destacados. Por esta razón, se elige este tipo de *learning rate schedule* como el óptimo para esta arquitectura.

En el caso de CenterNet con *backbone* Hourglass, se ve en la tabla 4.10 que el mejor *learning rate schedule* corresponde a la disminución de la tasa de aprendizaje en las épocas 15 y 18. Esto, al igual que en DLA, con un pre-entrenamiento sobre imágenes en escala de gris. Si bien, se podría debatir que la diferencia entre los resultados de los pre-entrenamientos con imágenes grises y a color es pequeña, el pre-entrenamiento con imágenes en escala de gris es consistentemente mejor para esta arquitectura, aunque sea por un bajo margen.

Por último, en estos experimentos con *backbone* HRNet, un modelo pre-entrenado con imágenes en escala de gris alcanza el mejor AR, y otro modelo pre-entrenado con imágenes a color alcanza el mejor AP. El modelo pre-entrenado con imagenes a color alcanza ese AP sin un *learning rate schedule*, mientras que el modelo pre-entrenado con imágenes grises lo alcanza con disminución del *learning rate* en las épocas 40 y 48. También, se puede ver que el modelo pre-entrenado con imágenes grises y sin un *learning rate schedule*, alcanza resultados muy similares al modelo pre-entrenado a color. Luego, comparando también los otros experimentos entre pre-entrenamientos con imágenes grises y a color, se decide que el pre-entrenamiento en régimen 3x, para esta arquitectura, se realiza con imágenes en escala de gris. Esta decisión se justifica si se toma en cuenta los resultados de la métrica AR, donde el pre-entrenamiento con imágenes en escala de gris ha demostrado ser, en general, mejor.

Tabla 4.12: Resultados de evaluación, sobre imágenes térmicas, de modelos pre-entrenados con régimen 3x en COCO con imágenes en escala de gris (sin *finetuning*).

Backbone	Régimen	AP	AR
DLA-34	1x	0,4850	0,5818
DLA-34	3x	0,5363	0,6204
DLA-34 (384×384)	3x	0,4777	0,5689
Hourglass-104	1x	0,4838	0,5750
Hourglass-104	3x	**0,5465**	**0,6400**
HRNet-W32	1x	0,3996	0,5889
HRNet-W32	3x	0,4748	0,6068

Dados los resultados obtenidos hasta este punto, se puede concluir que el pre-entrenamiento de los modelos con imágenes en escala de gris permite llegar a los mejores resultados. Esto es particularmente cierto para las arquitecturas DLA y Hourglass. Por otro lado, para la arquitectura HRNet, es debatible la veracidad de esta aseveración.

Considerando todo lo anterior, se decide volver a entrenar cada variante de CenterNet sobre imágenes en escala de gris, pero considerando un régimen 3x (ver tabla 3.1). Se entrena, como experimento adicional, CenterNet DLA con una menor resolución de entrada (384×384). La evaluación de estos modelos sobre el conjunto de imágenes térmicas, sin *finetuning*, se puede apreciar en la tabla 4.12. Adicionalmente, se adjunta en la misma tabla los resultados de la evaluación de los modelos entrenados con régimen 1x. Como se puede desprender de esta tabla, entrenar los modelos por una mayor cantidad de épocas mejora sustancialmente su rendimiento.

4.4.3. Congelamiento de Capas

Utilizando las mejores combinaciones de *learning rate*, *batch size* y *learning rate schedules* encontradas, se experimenta con distintos grados de congelamiento de las redes DLA y Hourglass. Es importante recordar, que en lugar de realizar un *finetuning* a los modelos pre-entrenados con régimen 1x, se utilizan aquellos pre-entrenados con régimen 3x. Los resultados de esto se resumen, de manera gráfica, en la figura 4.13. Tomar en cuenta que, para DLA, el grado "Freeze 6" corresponde al congelamiento de la red *backbone* completa, mientras que para Hourglass esto se alcanza en "Freeze 4".

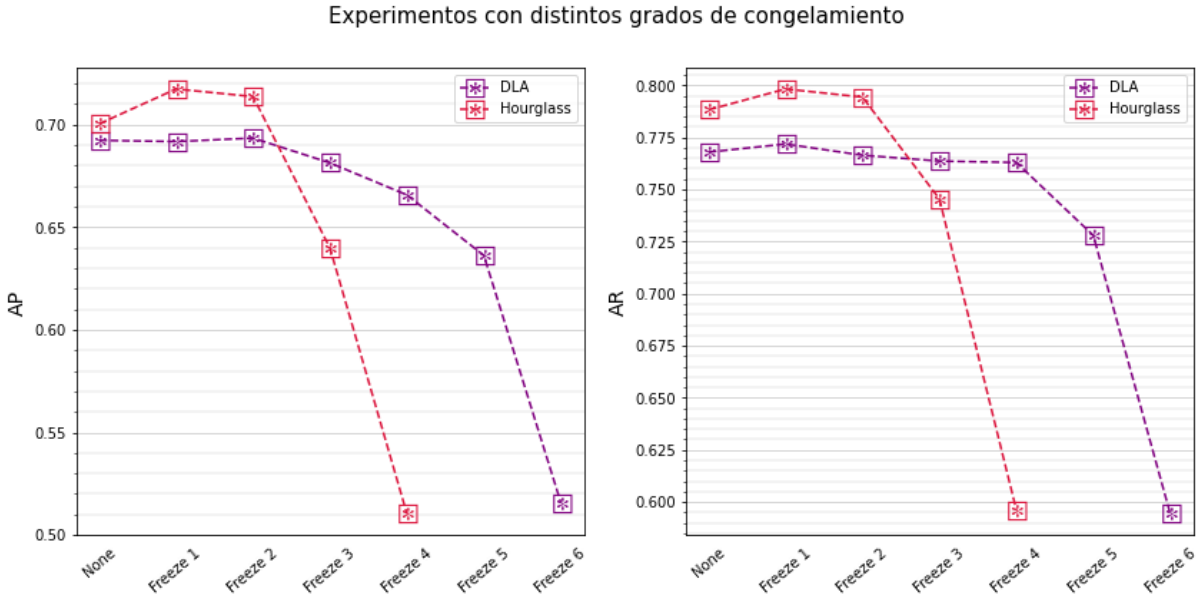


Figura 4.13: Resultados de *finetuning* con diferentes grados de congelamiento de la red *backbone*, para DLA y Hourglass.

Una interpretación de los gráficos muestra que, para ambas arquitecturas, congelar más allá de las primeras capas de la red perjudica el desempeño de los modelos. Considerando tanto AP como AR, para DLA congelar el primer módulo convolucional durante el *finetuning* (Freeze 1) conlleva a los mejores resultados. Aun así, la diferencia con no congelar la red es casi negligible, con una mejoría de un $0,39\,\%$ en AR y una disminución de $0,06\,\%$ en AP. No así para la red Hourglass, donde congelando el primer módulo convolucional (Freeze 1) se logra una mejora de un $1,7\,\%$ en AP y un $1\,\%$ en AR.

Los resultados anteriores también sugieren que el dominio de imágenes térmicas es lejano del dominio de imágenes en escala de gris o, en su defecto, de imágenes a color. En caso contrario, entrenar los modelos con un mayor grado de congelamiento de las redes (por sobre Freeze 2 en ambos casos) se espera que no hubiese afectado con gran magnitud el desempeño de los modelos. Ello dado que se habría logrado "reutilizar" las representaciones entrenadas en la base de datos COCO. Sin embargo, los experimentos señalan que el *finetuning* es importante para adaptar las representaciones hacia el nuevo dominio.

4.4.4. Experimentos Adicionales

En adicional a los experimentos señalados en los apartados anteriores, se realizan pruebas con cambios más pequeños del *learning rate* para las arquitecturas Hourglass y DLA. En el caso de Hourglass, se experimenta con *learning rates* de $0,00005$, $0,00015$, $0,00035$ y $0,00045$, en torno al valor $0,00025$ que ha sido el mejor encontrado hasta este punto. Para DLA, se experimenta con los valores de $0,0003$, $0,0004$, $0,0006$ y $0,0007$, en torno al valor $0,0005$. Los mejores resultados se combinan con el mejor grado de congelamiento de capas, según las figuras 4.13. Adicionalmente, para DLA, se experimenta con el modelo entrenado en imágenes de 384×384 utilizando los mejores parámetros encontrados y diferentes grados de congelamiento. En el caso de HRNet, se realiza un *finetuning* al modelo pre-entrenado con régimen 3x utilizando los mejores parámetros encontrados para esta arquitectura.

Los parámetros finales que conducen a los mejores resultados se presentan en la tabla 4.13, para cada arquitectura. Las evaluaciones respectivas, se adjuntan en las tablas 4.14 y 4.15. Las métricas AP y AR muestran que el mejor modelo es CenterNet con *backbone* Hourglass. Sin embargo, el modelo con arquitectura DLA, entrenado sobre imágenes de 384×384, difiere solo en un $2,1\,\%$ en AP y $2,9\,\%$ en AR.

Tabla 4.13: Parámetros finales de *finetuning* para mejores modelos CenterNet entrenados.

Backbone	Épocas	BS_1	BS_2	*LR*	*LR steps*	Congelamiento
DLA-34	50	8	16	5,00E-04	[35,45]	Freeze 1
DLA-34 (384×384)	50	8	16	5,00E-04	[35,45]	Freeze 1
Hourglass-104	20	4	8	3,50E-04	[15,18]	Freeze 1
HRNet-W32	5	8	8	1,00E-03	-	-

Tabla 4.14: AP para mejores modelos Centernet entrenados.

Backbone	AP	$AP_{0.5}$	$AP_{0.75}$	AP_M	AP_L
DLA-34	0,692	0,935	0,804	0,622	0,769
DLA-34 (384×384)	0,704	0,931	**0,834**	0,651	0,755
Hourglass-104	**0,725**	**0,947**	0,831	**0,677**	**0,786**
HRNet-W32	0,628	0,898	0,738	0,601	0,689

Tabla 4.15: AR para mejores modelos Centernet entrenados.

Backbone	AR	$AR_{0.5}$	$AR_{0.75}$	AR_M	AR_L
DLA-34	0,772	0,971	0,868	0,717	0,837
DLA-34 (384×384)	0,780	0,975	0,889	0,746	0,818
Hourglass-104	**0,801**	**0,979**	**0,893**	**0,770**	**0,848**
HRNet-W32	0,737	0,961	0,839	0,708	0,785

4.4.5. Tiempos de Inferencia

Los resultados presentados anteriormente, se pueden contrarrestar con los tiempos de inferencia evaluados para cada modelo. Estos tiempos de inferencia, son calculados para dos plataformas distintas. Uno sobre una GPU Titan V100 con 32GB de memoria RAM y otro con una GPU GTX 1080 con 8GB de RAM. La razón tras esto, es para acercar los resultados a plataformas computacionales más comunes como lo son la familia GTX de GPUs NVIDIA. Los resultados se muestran en la tabla 4.16. En esta tabla se ve que CenterNet con *backbone* Hourglass es el modelo más lento durante la inferencia. Por otro lado, DLA se presenta como el más rápido, alcanzando una velocidad de inferencia sobre los 20 FPS, para imágenes de 384×384 de resolución. Esto lo califica como un modelo con inferencia en tiempo real.

Tabla 4.16: FPS de mejores modelos CenterNet entrenados.

Backbone	Resolución de entrada	FPS (Titan V100)	FPS (GTX 1080)
DLA-34	[512x512]	18,2	15,9
DLA-34 (384×384)	[384x384]	**22,2**	**22,2**
Hourglass-104	[512x512]	10,0	9,1
HRNet-W32	[512x512]	12,5	16,9

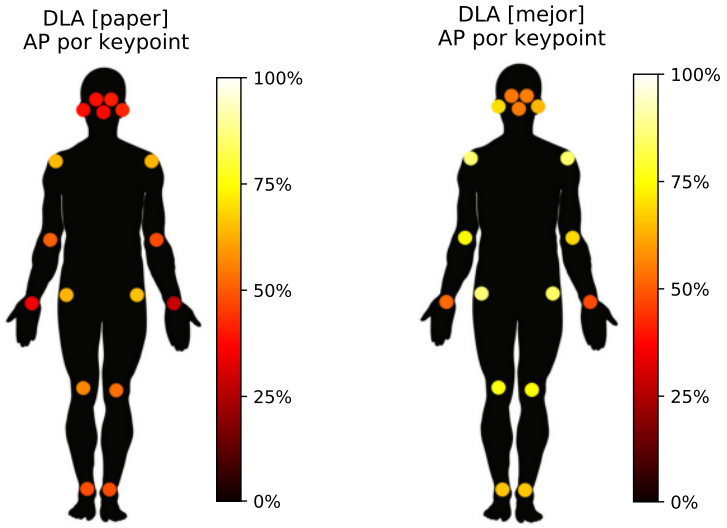
Al comparar las tablas 4.14, 4.15 y 4.16, se puede detectar un *trade-off* entre precisión y rapidez. Si bien la arquitectura Hourglass permite llegar a los mejores resultados en cuanto a precisión y *recall*, es el más lento de los modelos evaluados. Por otro lado, el AP y AR de CenterNet DLA es menor en $\approx 3\,\%$ en comparación a Hourglass, pero con una ganancia de más del doble de rapidez durante inferencia. A su vez, si se consideran las detecciones con un $OKS > 0,5$, tanto DLA como Hourglass poseen desempeños muy similares. Finalmente, la elección dependerá de la aplicación específica a la que se destine el sistema. Cabe destacar que CenterNet HRNet no se recomienda ocupar en ningún caso, dado que posee una peor precisión que DLA con una menor velocidad de inferencia.

4.4.6. Evaluación de *AP* y *AR* a nivel de *keypoint*

Hasta este punto, se han desplegado las evaluaciones de los modelos considerando la detección de pose como un promedio de la precisión y el *recall* de todos los *keypoints* de cada persona. A continuación, en las figuras 4.14 y 4.15, se presentan mapas de calores que representan la precisión de los modelos por cada *keypoint*. Se comparan los desempeños de los modelos originales (*paper*), con los mejores modelos obtenidos en este trabajo.

De estas figuras, se puede apreciar como los modelos originales (*paper*) tienen una baja precision en la mayor parte del cuerpo, salvo las caderas y los hombros. En contraste, los mejores modelos CenterNet DLA y Hourglass entrenados, detectan con alta precisión los hombros, codos, caderas y rodillas. Se destaca, en particular, la alta precisión en los hombros y caderas, cercana al 85 % en ambos casos. También, el hecho de que casi ningún *keypoint*, salvo la muñeca izquierda para el modelo DLA, posee una precisión por debajo del 50 %.

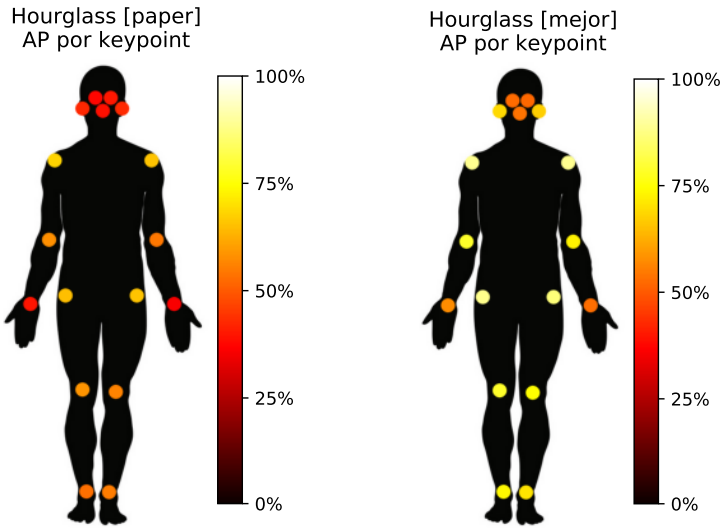
Asimismo, mejoran las detecciones en los *keypoints* de la cara y de los tobillos, con respecto a los modelos originales. En cuanto a los *keypoints* de las muñecas, se ve que no hay mucha mejoría en ambos casos. Comparando los mejores modelos Hourglass y DLA, se ve que en los tobillos y las muñecas el primero presenta una mejoría visible. Con todo, se resalta el hecho de que casi ningún *keypoint*, salvo la muñeca izquierda para el modelo DLA, posee una precisión por debajo del 50 %.



(a) Paper (b) Mejor

Figura 4.14: AP a nivel de *keypoint* para modelo original de CenterNet DLA (paper) y mejor modelo CenterNet DLA entrenado.

A pesar de las mejorías en ambos modelos, la detección es imprecisa para los *keypoints* de la cara y las muñecas. Esto se condice con la cantidad de anotaciones de estos *keypoints* en el conjunto de entrenamiento de imágenes térmicas (tabla 4.1). Sumado a esto, los rasgos de la cara son más difíciles de discernir en las imágenes térmicas en comparación a las imágenes a color, lo cual puede explicar la menor precisión en esta parte del cuerpo. Como consecuencia, se descarta recomendar la aplicación de estos modelos en áreas donde se necesite una alta precisión en la detección de rasgos faciales.



(a) Paper (b) Mejor

Figura 4.15: AP a nivel de *keypoint* para modelo original de CenterNet Hourglass (paper) y mejor modelo CenterNet Hourglass entrenado.

4.4.7. Ejemplos de Detección sobre Imágenes Capturadas en el Laboratorio

En la figura 4.16, se presentan algunos ejemplos de detección de pose humana sobre imágenes capturadas en el laboratorio de la facultad. Se muestran las detecciones de los dos mejores modelos CenterNet entrenados, con arquitecturas DLA y Hourglass. Estas imágenes, no etiquetadas, son independientes de los conjuntos de entrenamiento y evaluación de imágenes térmicas con los que se han entrenado y evaluado los modelos.

En las figuras, se puede ver que los resultados para ambos modelos son, a simple vista, muy similares. Sin embargo, se pueden encontrar ciertos detalles que delatan la mejor precisión del modelo Hourglass. En particular, en la imagen de la última fila, para la detección de la persona en la derecha, Hourglass ubica de mejor manera los *keypoints* de las piernas. Lo mismo sucede para la persona en la derecha de la imagen de la segunda fila. Diferencias pequeñas como esta, brindan a CenterNet con *backbone* Hourglass un mejor desempeño en el conjunto de evaluación considerado.

El problema relacionado a la detección de los *keypoints* faciales también se aprecia en las detecciones. Es más notorio en las imágenes de la primera y cuarta fila. Como se puede ver ahí, a los modelos de ambas arquitecturas se les hace difícil ubicar los ojos, nariz y orejas.

(a) Imagen Térmica (b) DLA (c) Hourglass

Figura 4.16: Ejemplos de detección sobre imágenes capturadas en el laboratorio, para mejores modelos CenterNet DLA y Hourglass entrenados.

4.5. Programa de Demostración

El programa desarrollado en Python, permite detectar la pose humana sobre imágenes y vídeos térmicos. El repositorio del proyecto, que contiene el programa de demostración `detect_people.py` en la carpeta *src*, es el siguiente: https://github.com/jsmithdlc/CenterNet-Thermal-Human-Pose-Estimation.git. En ese enlace, se pueden encontrar instrucciones para configurar las dependencias (librerías) necesarias, armar la estructura del proyecto y conocer las opciones que se brindan para la detección.

A grandes rasgos, el programa permite al usuario elegir entre la detección de pose sobre una imagen, carpeta con imágenes o video. Hay tres modelos disponibles para la detección, correspondientes a los mejores entrenamientos de CenterNet con las arquitecturas DLA, Hourglass y HRNet. Se le otorga al usuario la opción de guardar las imágenes con las detecciones de pose humana dibujadas sobre ella y/o almacenar un archivo en formato *.csv* que contiene las coordenadas de los *bounding box* detectados, la confianza en las predicciones y las coordenadas de los *keypoints* predichas. Un video demostrando los resultados que se pueden obtener utilizando este programa sobre un vídeo térmico, se puede encontrar en el siguiente enlace: https://youtu.be/WRbDxdgnn_o.

4.6. Comparación de Mejores Modelos CenterNet con otros Sistemas de Detección de Pose Humana

A modo de evaluar el desempeño de los modelos CenterNet entrenados, se explora el uso de otros sistemas de detección de pose humana para resolver la estimación de pose en el dominio térmico. Estos modelos, se pre-entrenan utilizando la base de datos COCO con imágenes en escala de gris, al igual que los mejores modelos CenterNet entrenados. Luego, se les realiza un *finetuning* sobre el conjunto de entrenamiento de imágenes térmicas, utilizando para ello los parámetros de entrenamiento reportados en los trabajos originales de cada sistema [55, 26, 30, 52]. En total, son 4 los modelos adicionales explorados.

4.6.1. Modelos Comparados

Dado que la estimación de pose humana en CenterNet cae bajo el paradigma *bottom-up*, tiene más sentido comparar los modelos con otros que sigan el mismo paradigma. Sin embargo, se entrena uno correspondiente al paradigma *top-down* como representante de este tipo de modelos, para contrarrestar con los resultados de los otros modelos. El modelo *top-down* entrenado es Simple Baselines [55], seleccionado debido a sus resultados estado-del-arte, reportados en *benchmarks* clásicos como COCO y MPII [21, 19]. A grandes rasgos, este sistema simplifica el problema de detección de pose utilizando una arquitectura *backbone* simple que tiene como base la red ResNet [28], modificada con la adición de capas deconvolucionales en la fase de *upsampling*.

Igualmente, se entrena el modelo PoseAE [26], correspondiente al paradigma *bottom-up*. Este modelo, a pesar de haber sido introducido el año 2017, sigue siendo competente en las competencias de COCO y MPII. La característica principal de este modelo, es el uso de *embeddings* asociativos como etiquetas para agrupar los *keypoints* detectados. El modelo predice, simultáneamente, mapas de calor con *peaks* en los *keypoints* detectados y estos *embeddings* que permiten agrupar los *keypoints* de cada persona. Utiliza, como *backbone*, una Stacked Hourglass Network [12] con cuatro módulos Hourglass.

Otro modelo de detección de pose *bottom-up* entrenado es *Bottom-Up HRNet* [30]. Este modelo, al igual que CenterNet, se caracteriza por agrupar los *keypoints* detectados a partir de una regresión, a nivel de pixel, del centro de las personas. Sin embargo, en la entrada de la cabeza de regresión, combina tanto el mapa de calor generado para cada *keypoint*, como el mapa de características de entrada generado por la red *backbone*. Adicional a esto, refina las locaciones detectadas utilizando un *spatial transformer network*. Utiliza, como arquitectura *backbone*, la red HRNet [29]. En este caso, se utiliza la versión W32 de esta red, de menor envergadura que W48, puesto que esta última entrega resultados notablemente mejores solo con imágenes de resolución mayor a 512×512. Esta resolución ya es mayor al tamaño promedio de las imágenes térmicas anotadas, por lo que se prevé que no se alcanzará mucha diferencia en el desempeño al utilizar la red más grande.

El último modelo entrenado, corresponde a OpenPose [52]. Este modelo, introducido en el año 2017, suele utilizarse como punto de comparación para otros modelos del paradigma *bottom-up*. Es el mayor representante de los modelos de detección *bottom-up* que agrupan los *keypoints* detectados a través de *Part Affinity Fields* (PAF). Estos, son campos de flujo que codifican los grados de asociación entre pares de *keypoints* de manera no estructurada. La arquitectura de OpenPose se puede separar en dos partes, cada una con estructura similar a la red ResNet [28]. La primera parte de la red detecta los PAF, mientras que la segunda parte detecta los mapas de calor con las locaciones probables de los *keypoints*.

4.6.2. Resultados de Evaluación sobre Imágenes Térmicas

4.6.2.1. *AP* y *AR*

Con todo lo mencionado, se presenta en las tablas 4.17 y 4.18 los resultados para las métricas AP y AR, de los mejores modelos entrenados para cada sistema, junto a los mejores resultados obtenidos para CenterNet. Notar que, para OpenPose [52], se realiza el *finetuning* sobre el modelo original, dado que el entrenamiento de este sistema sobre COCO toma bastante tiempo (>20 días) como para realizarlo en la plataforma computacional utilizada en este trabajo. Detalles adicionales sobre la exploración de parámetros (*learning rate* y *batch size*) efectuada para cada sistema, aparte de CenterNet, se puede encontrar en el anexo C.

Tabla 4.17: *AP* para modelos de referencia, junto a mejores resultados de CenterNet.

AP MODELOS FINALES

Experiment	$\boldsymbol{AP}$	$\boldsymbol{AP_{0.5}}$	$\boldsymbol{AP_{0.75}}$	$\boldsymbol{AP_M}$	$\boldsymbol{AP_L}$
Top-Down					
Simple Baselines	**0,777**	**0,947**	**0,861**	**0,752**	**0,805**
Bottom-Up					
CenterNet-DLA (384 × 384)	0,704	0,931	0,834	0,651	0,755
CenterNet-DLA	0,692	0,935	0,804	0,622	0,769
CenterNet-Hourglass	0,725	0,947	0,831	0,677	0,786
CenterNet-HRNet	0,628	0,898	0,738	0,601	0,689
Bottom-Up HRNet	0,644	0,910	0,805	0,634	0,717
PoseAE	**0,754**	**0,959**	**0,882**	**0,728**	**0,817**
OpenPose	0,679	0,877	0,768	0,615	0,745

Tabla 4.18: *AR* para modelos de referencia, junto a mejores resultados de CenterNet.

Experiment	$\boldsymbol{AR}$	$\boldsymbol{AR_{0.5}}$	$\boldsymbol{AR_{0.75}}$	$\boldsymbol{AR_M}$	$\boldsymbol{AR_L}$
Top-Down					
Simple Baselines	**0,815**	**0,964**	**0,886**	**0,810**	**0,822**
Bottom-Up					
CenterNet-DLA (384 × 384)	0,780	0,975	0,889	0,746	0,818
CenterNet-DLA	0,772	0,971	0,868	0,717	0,837
CenterNet-Hourglass	0,801	**0,979**	0,893	0,770	**0,848**
CenterNet-HRNet	0,737	0,961	0,839	0,708	0,785
Bottom-Up HRNet	0,724	0,950	0,857	0,715	0,743
PoseAE	**0,820**	**0,979**	**0,911**	**0,797**	0,843
OpenPose	0,735	0,900	0,811	0,688	0,780

Las tablas anteriores muestran que el modelo Simple Baselines [55] supera a los otros modelos en las métricas principales de *AP* y *AR*. Esto es de esperar, considerando que los modelos que siguen el paradigma *top-down* suelen ser más precisos. Por otro lado, de los modelos *Bottom-Up*, el sistema PoseAE [26] es aquel que tiene el mejor rendimiento. Le siguen los modelos CenterNet entrenados en este trabajo, encabezados por Hourglass. En particular, se puede ver que, en *recall*, el modelo Hourglass se acerca bastante a PoseAE, superándolo en las detecciones con $OKS > 0,5$ y para objetos de larga escala (AR_L).

Los buenos resultados obtenidos por el modelo PoseAE, se pueden explicar por varios motivos. Sin embargo, el hecho de que utiliza la arquitectura más compleja de todos los sistemas *bottom-up*, con 4 módulos Hourglass, puede ser el factor contribuyente principal. Si se compara esta arquitectura con aquella utilizada en CenterNet, se ve que esta última contiene solo 2 módulos Hourglass. Luego, es de esperar que incrementando la cantidad de módulos Hourglass en la variante de CenterNet, incremente el desempeño de los modelos, dado que incrementa la complejidad del modelo. Más aún, es importante notar que CenterNet supera a todos los otros modelos *bottom-up* entrenados, aparte de PoseAE.

4.6.2.2. Tiempos de inferencia

Otro factor importante a considerar, es la velocidad de inferencia de los modelos. De la misma forma que se hizo para los modelos CenterNet, se evalúa la velocidad de inferencia de los otros modelos sobre dos plataformas computacionales distintas. Los resultados de esto se aprecian en la tabla 4.19. En la tabla, se puede ver que el modelo CenterNet con *backbone* DLA, entrenado y evaluado con imágenes de 384×384, es aquel con mayor velocidad de inferencia en ambas plataformas. Supera por un amplio margen al modelo OpenPose, el cual, aparte de las otras variantes de CenterNet, es el que le sigue en cuanto a rapidez.

Tabla 4.19: FPS de modelos de referencia y mejores modelos CenterNet entrenados.

Experiment	**Resolución**	**FPS (TESLA V100)**	**FPS (GTX 1660)**
Top-Down			
Simple Baselines	[384x288]	**5,1**	**5,41**
Bottom-Up			
CenterNet-DLA (384×384)	[384x384]	**22,2**	**22,2**
CenterNet-DLA	[512x512]	**18,2**	15,9
CenterNet-Hourglass	[512x512]	10,0	9,1
CenterNet-HRNet	[512x512]	12,5	16,9
Bottom-Up HRNet	[512x512]	1,5	1,4
PoseAE	[256x256]	5,3	5,8
OpenPose	[368x368]	13,0	6,8

Asimismo, al comparar la velocidad de inferencia de CenterNet-Hourglass con PoseAE, se puede ver que este último es notablemente más lento. Esto incluso considerando que utiliza imágenes cuatro veces más pequeñas que el primero. Lo anterior se puede explicar dada la complejidad de la arquitectura que utiliza, discutida anteriormente.

Por último, se puede comparar la velocidad de inferencia del modelo Simple Baselines con respecto a los métodos *bottom-up*. En esto, se destaca que es el segundo modelo más lento, después de Bottom-Up HRNet, y con unos FPS muy cercanos a PoseAE en ambas plataformas. Más aun, en la tabla 4.20 se pueden ver los tiempos de inferencia de los modelos sobre una imagen con 11 personas. Aquí, es notoria la lentitud del modelo Simple Baselines en comparación a los modelos *bottom-up*, cuyas velocidades no cambian radicalmente con respecto a lo dispuesto en la tabla 4.19.

Tabla 4.20: FPS de modelos de referencia y mejores modelos CenterNet entrenados, sobre imagen con 11 personas.

Experiment	**FPS (GTX 1080)**
Top-Down	
Simple Baselines	**0,8**
Bottom-Up	
CenterNet-DLA (384×384)	**22,2**
CenterNet-DLA	16,4
CenterNet-Hourglass	5,8
CenterNet-HRNet	14,3
Bottom-Up HRNet	1,2
PoseAE	3,0
OpenPose	6,8

4.6.3. Resumen

En vista de los resultados comentados en las secciones pasadas, se puede decir que cada modelo va a servir a aplicaciones distintas. En aquellas donde se necesite una alta tasa de aciertos en las detecciones, sin importar el tiempo de inferencia, es preferible el modelo Simple Baselines. Por otro lado, en las aplicaciones que requieran inferencia en tiempo real, con un grado de acierto menor, se recomienda el uso de CenterNet DLA. Un término medio, entre precisión y tiempo de inferencia, lo alcanza el modelo CenterNet Hourglass. Asimismo, el modelo PoseAE se presenta como una alternativa para aquellas situaciones donde se necesite detectar, simultáneamente, un alto número de personas con precisión y sin comprometer demasiado el tiempo de inferencia.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1. Conclusiones

La detección de pose humana en imágenes térmicas, es una tarea que aún tiene un largo camino de investigación por delante. Sin embargo, el trabajo desarrollado demuestra que se puede obtener un alto grado de éxito al adaptar las más recientes tecnologías de *deep learning* a la resolución del problema. El trabajo aborda varios obstáculos que se encuentran al intentar implementar un sistema de detección de pose en este dominio, desde la construcción de una base de datos hasta la selección de los modelos de detección y su correcto entrenamiento.

Para la construcción de la base de datos con imágenes térmicas anotadas para pose humana, se exploraron métodos basados en redes GAN. Estas técnicas probaron ser fútiles en esta ocasión. No obstante, el constante avance de las tecnologías de traducción de imágenes promete un buen futuro para la implementación de esta idea.

Por otro lado, se logró recolectar y etiquetar una base de datos para el problema, que comprende 600 imágenes térmicas para el conjunto de evaluación y 200 imágenes térmicas para el conjunto de evaluación. Estas imágenes se recopilaron de diversas fuentes, con tal de generalizar las situaciones en las que se pueden encontrar las personas a detectar. Se espera que la base de datos mencionada, liberada al público, facilite e impulse el desarrollo de nuevos modelos para la detección de pose humana en el dominio térmico.

En cuanto al entrenamiento y evaluación de los modelos de detección en sí, los resultados demuestran que el pre-entrenamiento en una base de datos extensa, como COCO, aporta significativamente al problema en cuestión. En particular, si se utilizan imágenes en escala de gris. Un pre-entrenamiento sobre este tipo de imágenes, puede guiar a los modelos a mejores resultados, en comparación a imágenes a color, en la detección sobre el dominio térmico. Esto fue lo que se evidenció, con mayor magnitud, en las variantes de CenterNet con *backbone* DLA y Hourglass.

Más aun, se ve que un *finetuning* sobre un conjunto limitado de imágenes térmicas, permite llegar a resultados de precisión y *recall* competentes para una amplia gama de aplicaciones. La exploración de los parámetros de entrenamiento es importante en este caso. Cambios en el *batch size* y *learning rate*, pueden conducir a resultados totalmente distintos.

Importantes, también, son las arquitecturas *backbone* que se utilizan como parte del proceso de detección. Utilizar la red DLA como *backbone* para CenterNet permite alcanzar niveles de precisión y *recall* por sobre el 70 % en el conjunto de evaluación provisto, con una inferencia en tiempo real de 22 FPS. En cambio, si se cambia la arquitectura a Hourglass, se aprecian mejoras del 2 % en cada métrica, pero con un tiempo de inferencia 10 FPS más bajo. La elección, luego, depende de la aplicación a la que se destine el sistema.

Otro punto explorado, fue realizar el *finetuning* de los modelos CenterNet con distintos grados de congelamiento de la red *backbone*. Ello, con la intención de preservar las representaciones obtenidas tras el pre-entrenamiento de los modelos en la base de datos COCO. Para DLA y Hourglass, se demuestra que congelar el primer módulo convolucional induce a resultados marginalmente superiores. Sin embargo, congelar una extensión mayor de las arquitecturas, no se recomienda puesto que perjudica la precisión de los modelos.

Cabe decir, que se identifica una baja precisión en la detección de los *keypoints* de la cara y las muñecas de las personas. Esto se puede explicar al ver la menor cantidad de anotaciones disponibles para estos *keypoints*, en el conjunto de entrenamiento utilizado para el *finetuning*. Otra explicación puede ser que la identificación de estas partes sea inherentemente más complicado en el dominio térmico, donde la resolución de las imágenes tiende a ser menor y se homogeneizan los rasgos faciales de las personas.

Además de los modelos CenterNet entrenados, se entrenaron y evaluaron otros modelos de detección de pose humana populares para efectos de comparación. En relación a esto, se destaca que el modelo Simple Baselines, perteneciente al paradigma *top-down*, sobrepasa en precisión y *recall* a cualquier otro modelo explorado con ambas métricas por sobre el 80 %. Sin embargo, posee un bajo tiempo de inferencia y este se ve adicionalmente afectado por la cantidad de personas presentes en la imagen.

El fenómeno descrito anteriormente no ocurre en los modelos *bottom-up*, los cuales poseen, en general, altas velocidades de inferencia. De estos, los modelos CenterNet entrenados son los más rápidos, pero PoseAE es aquel más preciso. Esta alta precisión se debe, probablemente, a la complejidad de su red *backbone*, la cual es también la principal razón de que sea el segundo modelo más lento de aquellos de este paradigma, con cerca de 5 FPS. La elección de alguno de estos modelos, luego, depende de las necesidades en precisión y rapidez de inferencia que tengan los usuarios.

5.2. Trabajo Futuro

A continuación, se listan algunos puntos que pueden aportar extendiendo el trabajo realizado y a la resolución del problema de detección de pose humana en imágenes térmicas:

- **Recopilación y anotación de un mayor conjunto de imágenes térmicas**: actualmente, el desempeño de los modelos de detección de pose humana sobre imágenes térmicas depende ampliamente del pre-entrenamiento sobre imágenes de otro dominio. Si bien esto aporta a la solución en cuestión, disponer de una mayor cantidad de imágenes térmicas con personas en distintas poses y entornos aportaría a la obtención de modelos más precisos y generalizados para la detección en el dominio térmico.

- **Entrenamiento de CenterNet con arquitectura Stacked Hourglass Network de 4 módulos Hourglass**: el alto desempeño alcanzado por el modelo PoseAE, plantea la interrogante de si al entrenar CenterNet con una red *backbone* Hourglass con igual cantidad de módulos que la red utilizada en PoseAE, se podría notar una mejora en el desempeño del sistema CenterNet con esta arquitectura. Entrenar tal modelo podría ayudar a validar, o refutar, el sistema CenterNet frente a otros métodos de detección de pose *bottom-up*.

- **Evaluación de modelos separada de acuerdo a la resolución de las imagenes**: los experimentos con el *backbone* HRNet en CenterNet, mostraron que esta red se desempeña peor que las otras redes en el conjunto de evaluación utilizado. Sin embargo, no se evaluaron los modelos en imágenes agrupadas de acuerdo a la resolución. Puede ser, que HRNet se desempeñe mejor en imágenes de menor resolución, al considerar que mantiene una alta resolución del mapa de características a lo largo de su extensión. Identificar esto, podría ayudar a determinar si se puede satisfacer la necesidad de detección de pose humana en imágenes térmicas de muy baja resolución.

- **Optimización de modelos con la librería TensorRT**: esta librería se puede utilizar para mejorar el tiempo de inferencia de los modelos CenterNet entrenados. Dependiendo de los resultados de esta optimización, se puede llegar a mejorar la eficiencia de los modelos para que realicen detecciones en tiempo real o se ejecuten en plataformas computacionales de menor costo que las utilizadas en este trabajo.

Bibliografía

[1] D. Kingma, J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[2] C. Premebida, U. Nunes, J. Armingol y A.de la Escalera D. Olmeda. Lsi far infrared pedestrian dataset. 2013.

[3] P. Isolaand, J. Zhu, T. Zhou, A.A. Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[4] FLIR. Free flir thermal dataset for algorithm training. 2020. URL https://www.flir.com/oem/adas/adas-dataset-form/.

[5] K. He,y J. Sun. Convolutional neural networks at constrained time cost. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5353–5360, 2015.

[6] C. Zheng, W. Wu, T. Yang, S. Zhu, C. Chen, R. Liu, J. Shen, N. Kehtarnavaz and M. Shah. Deep learning-based human pose estimation: A survey, 2021.

[7] W. Kentaro. labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme, 2016.

[8] Hei Law and Jia Bin Deng. Cornernet: Detecting objects as paired keypoints. *International Journal of Computer Vision*, 128:642 – 656, 2019.

[9] J.M LLoyd. *Thermal Imaging Systems*. Plenum Press, New York, 3rd edition, 1975.

[10] R. Miezianko. Terravic research infrared database. *IEEE OTCBVS WS Series Bench*.

[11] J. Nelson. A comprehensive database for benchmarking imaging systems. *Roboflow*, 2020. URL https://public.roboflow.com/object-detection/thermal-dogs-and-people.

[12] A. Newell, and K. Yiang, y J. Deng. Stacked hourglass networks for human pose estimation. *Computer Vision and Pattern Recognition 2016 (CVPR)*, 9912:483–499, 10 2016. doi: 10.1007/978-3-319-46484-8_29.

[13] J. Zhu, T. Park and P. Isola, y A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[14] M. Planck. Ueber das gesetz der energieverteilung im normalspektrum,. *Ann. der Phys*, 4:553–563, 1901.

[15] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, 39(6):1137–1149, 2017.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[17] J. Redmon, S. Divvala, R. Girshick, y A. Farhadi. You only look once: Unified, real-time object detection. *(CVPR)*, pages 779–788, 2016.

[18] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler y B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL http://arxiv.org/abs/1511.06645.

[19] M. Andriluka, L. Pishchulin, P. Gehler, y B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[20] A. Toshev y C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *CVPR*, 2014.

[21] T. Li, M. Maire, S. Belongie, J. Hays , P. Perona , D. Ramanan, P. Dollár, y C.L. Zitnick. *Microsoft COCO: Common Objects in Context*. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10602-1.

[22] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. Efros, O. Wang y E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, 2017.

[23] A. Krizhevsky, I. Sutskever, y G. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012.

[24] A. Akula, R. Ghosh, S. Kumar y H. K Sardana. Moving target detection in thermal infrared imagery using spatiotemporal information. In *JOSA A*, volume 30(8), pages 1492–1501, 2013.

[25] S. Hwang, J. Park, N. Kim, Y. Choi, y I. S. Kweon. Multispectral pedestrian detection: Benchmark dataset and baselines. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[26] A. Newell, Z. Huang y J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. URL https://arxiv.org/abs/1611.05424.

[27] R. Mehra, M. Chetty, y J. K. Kamalu. Multiperson pose estimation using thermal and depth modalities. ., 2017.

[28] K. He, X. Zhang, S. Ren, y J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[29] K. Sun, B. Xiao, D. Liu, y J. Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[30] K. Sun, Z. Geng, D. Meng, B. Xiao, D. Liu, Z. Zhang y J. Wang. Bottom-up human pose estimation by ranking heatmap-guided adaptive keypoint estimates. *arXiv preprint arXiv:*, 2020.

[31] J. W. Davis y M. A. Keck. A two-stage template approach to person detection in thermal imagery. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, volume 1, pages 364–369, 2005. doi: 10.1109/ACVMOT.2005.14.

[32] Z. Wu, N. Fuller, D. Theriault y M. Betke. A thermal infrared video benchmark for visual analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 201–208, 2014. doi: 10.1109/CVPRW.2014.39.

[33] P. Pinheiro, R. Collobert, y P. Dollar. Learning to segment object candidates. *NIPS*, pages 1990–1998, 2015.

[34] T. Lin, P. Goyal, R. Girshick, K. He, y P. Dollár. Focal loss for dense object detection. *ICCV*, pages 2999–3007, 2017.

[35] X. Zhou, J. Zhuo, y P. Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, 2019.

[36] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, y Q. Tian. Centernet: Keypoint triplets for object detection. *ICCV*, pages 6568–6577, 2019.

[37] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, y Q. Tian. Centernet: Keypoint triplets for object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6568–6577, 2019.

[38] P.-L. St-Charles, G.-A. Bilodeau , y R. Bergevin. Mutual foreground segmentation with multispectral stereo pairs. In *International Conference on Computer Vision Workshops (ICCV Workshops)*, Octubre 2017.

[39] P. Piccini, A. Prati, y R. Cucchiara. Real-time object detection and localization with sift-based clustering. *Image and Vision Computing*, 30(8):573–587, 2012. doi: https://doi.org/10.1016/j.imavis.2012.06.004.

[40] K. He, G. Gkioxari, P. Dollár, y R. Girshick. Mask r-cnn. *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[41] C. Palmero, A. Clapés, C-Bahnsen, A. Møgelmose, T. B. Moeslund, y S. Escalera. Multimodal rgb–depth–thermal human body segmentation. In *International Journal of Computer Vision*, pages 1–23, 2016.

[42] X. Wu, D. Sahoo, y S. Hoi. Recent advances in deep learning for object detection. *2019 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] M. Mirza, y S. Osindero. Conditional generative adversarial nets. *ArXiv*, abs/1411.1784, 2014.

[44] S. Liu y S. Ostadabbas. Seeing under the cover: A physics guided learning approach for in-bed pose estimation. In *MICCAI*, 2019.

[45] S. Liu, X. Huang, N. Fu, C. Li, Z. Su y S. Ostadabbas. Simultaneously-collected multimodal lying pose dataset: Towards in-bed human pose monitoring under adverse vision conditions. *arXiv preprint arXiv:2008.08735*, 2020.

[46] W. Liu , D. Anguelov , D. Erhan , C. Szegedy, y Scott Reed. Ssd: Single shot multibox detector. *ECCV*, 2016. doi: 10.1007/978-3-319-46448-0_2.

[47] F. Yu, D. Wang, E. Shelhamer, y T. Darrel. Deep layer aggregation. *Computer Vision and Pattern Recognition (CVPR) 2018*, pages 770–778, 2018.

[48] Y. Socarras, S. Ramos, D. Vazquez, A. Lopez y T. Gevers. Adapting pedestrian detection from synthetic to far infrared images. In *ICCV – Workshop on Visual Domain Adaptation and Dataset Bias*, Sydney, Australia, 2013.

[49] R. Gade y T.B. Moeslund. Constrained multi-target tracking for team sports activities. In *IPSJ Transactions on Computer Vision and Applications*, volume 10. kaggle, 2018. URL https://doi.org/10.1186/s41074-017-0038-z.

[50] V. V. Kniaz, V. A. Knyaz, J. Hladůvka, W. G. Kropatsch, y V. Mizginov. Thermalgan: Multimodal color-to-thermal image translation for person re-identification in multispectral dataset. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 606–624, Cham, 2019. Springer International Publishing. ISBN 978-3-030-11024-6.

[51] K. Panetta, Q. Wan, S. Agaian, S. Rajeev, S. Kamath, R. Rajendran, S. P. Rao, A. Kaszowska, H. A. Taylor, A. Samani y X. Yuan. A comprehensive database for benchmarking imaging systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42 (3):509–520, 2020. doi: 10.1109/TPAMI.2018.2884458.

[52] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, y Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[53] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, y Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, y K. Q. Weinberger, editor, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[54] Y. Liu, Z. Lu, J. Li, C. Yao, y Y. Deng. Transferable feature representation for visible-to-infrared cross-dataset human action recognition. *Complexity*, 2018:1–20, 2018. doi: 10.1155/2018/5345241.

[55] B. Xiao, H. Wu y Y. Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2019.

Anexos

Anexo A

Datos Adicionales de la Base de Datos Construida

En la tabla A.1, se muestra la cantidad de imágenes obtenidas de cada fuente para la construcción de la base de datos de imágenes térmicas. En la tabla A.2, se muestra una caracterización de cada fuente según el tipo de captura (imágenes o vídeo) y el contexto de captura (exterior y/o interior).

Tabla A.1: Cantidad de imágenes, según cada fuente, para los conjuntos de entrenamiento y evaluación de la base de datos de imágenes térmicas construida.

Fuente	Entrenamiento	Prueba
FLIR [4]	271	83
ThermalWorld [50]	161	32
LSIFIR [2]	36	11
BUTIV [32]	30	10
Terravic [10]	2	1
infAR [54]	12	3
SLP [45]	23	7
OSU [31]	1	2
Biloudet [38]	9	3
CSIR [24]	2	1
Kaist [25]	13	4
Thermal DogsAnd People [11]	11	6
Thermal Soccer [49]	20	4
TUFTS Face [51]	9	3
CVC [48]	0	20
AAU VAP Trimodal [41]	0	8
Otro	0	2
TOTAL	**600**	**200**

Tabla A.2: Tipo de formato en que se encuentran las bases de datos de imágenes térmicas y sus contextos de captura.

Fuente	Tipo	Contexto
FLIR [4]	Video	Exterior
ThermalWorld [50]	Imágenes	Interior + Exterior
LSIFIR [2]	Video	Exterior
BUTIV [32]	Video	Interior + Exterior
Terravic [10]	Imágenes	Exterior
infAR [54]	Video	Exterior
SLP [45]	Imágenes	Interior
OSU [31]	Video	Exterior
Biloudet [38]	Video	Interior
CSIR [24]	Imágenes	Exterior
Kaist [25]	Video	Exterior
Thermal DogsAnd People [11]	Imágenes	Exterior
Thermal Soccer [49]	Imágenes	Exterior
TUFTS Face [51]	Imágenes	Interior
CVC [48]	Video	Exterior
AAU VAP Trimodal [41]	Video	Interior
Otro	Imágenes	Exterior

Anexo B

Elección de Máscaras de Segmentación Térmica

Para elegir la mejor máscara de segmentación térmica a utilizar en la transformación con ThermalGAN [50], se experimenta con distintas combinaciones de valores para los píxeles de las personas y el fondo. En las figuras B.1, B.2, B.3, B.4 y B.5 se muestran ejemplos de experimentos con distintas máscaras de segmentación. En la columna de la izquierda, se muestra la imagen original, en la columna media la imagen de contrastes térmicos relativos $\hat{R}$ y en la columna de la derecha la imagen térmica absoluta $\hat{B}$.

Notar, que cuando la diferencia entre los píxeles de las personas y el fondo es pequeña (0-0,3-6 y 6-3 en figuras B.1, B.4 y B.2), no se alcanza a percibir apropiadamente las personas en algunas imágenes $\hat{B}$. Por otro lado, al aumentar la diferencia (10-3 y 3-10 en figuras B.3 y B.5), se fuerza a la transformación a mostrar las personas con un mayor contraste con respecto al fondo. En relación a esto, los experimentos con máscaras donde el fondo tiene un valor de pixel de 3 y las personas un valor de 10 (figura B.3), la imagen $\hat{B}$ muestra apropiadamente los sujetos con una intensidad de píxeles característica del dominio térmico. Sin embargo, a expensas de exagerar los bordes de las personas de una manera poco natural.

Por otro lado, la transformación con las máscaras donde el fondo está a una intensidad de 10 y las personas a una intensidad de 3 (figura B.5), demuestra que las imágenes térmicas absolutas $\hat{B}$ obtenidas con este método presentan un problema parecido al anterior pero a la inversa. En este caso, las personas tienen una intensidad de pixeles significativamente menor que el fondo. No obstante, en la imagen térmica relativa $\hat{R}$ se muestra que las personas poseen características acorde al dominio térmico sobre un fondo negro. Por esta razón, se optó por ocupar este tipo de máscaras de segmentación térmica para la transformación, y extraer, después de la transformación, como imagen térmica final la imagen $\hat{R}$.

(a) Color (b) $\hat{R}$ (c) $\hat{B}$

Figura B.1: Ejemplo de transformación utilizando máscara de segmentación térmica con pixeles de personas en 0 y pixeles de fondo en 0.

(a) Color (b) $\hat{R}$ (c) $\hat{B}$

Figura B.2: Ejemplo de transformación utilizando máscara de segmentación térmica con pixeles de personas en 6 y pixeles de fondo en 3.

(a) Color (b) $\hat{R}$ (c) $\hat{B}$

Figura B.3: Ejemplo de transformación utilizando máscara de segmentación térmica con pixeles de personas en 10 y pixeles de fondo en 3.

(a) Color (b) $\hat{R}$ (c) $\hat{B}$

Figura B.4: Ejemplo de transformación utilizando máscara de segmentación térmica con pixeles de personas en 3 y pixeles de fondo en 6.

(a) Color (b) $\hat{R}$ (c) $\hat{B}$

Figura B.5: Ejemplo de transformación utilizando máscara de segmentación térmica con pixeles de personas en 3 y pixeles de fondo en 10.

Anexo C

Parámetros de Entrenamiento de Modelos de Comparación

Los modelos Simple Baselines, PoseAE y Bottom-Up HRNet [55, 26, 30], se entrenaron sobre COCO con imágenes en escala de gris, utilizando los parámetros de entrenamiento señalados en la tabla C.1. Asimismo, durante el *finetuning*, se exploraron distintas combinaciones de *batch size* y *learning rate* para cada modelo anterior. Los resultados de esto se aprecian en las tablas C.3, C.4, C.2. Para todos los experimentos, se considera un *batch size* identico en ambas GPUs. Cabe destacar, también, que los modelos Simple Baselines, Bottom-Up HRNet y PoseAE se les entrena por 50 épocas durante el *finetuning*. Los primeros dos, con *learning rate steps* en las épocas 35 y 45, mientras que el último sin un *learning rate schedule*.

En el caso de OpenPose, dada la diferente implementación, se entrena durante el *finetuning* por distintas cantidades de iteraciones, sin un *learning rate schedule*. Por la misma razón, solamente se utiliza un *batch size* (16), dado que entrenar con otros *batch size* requiere cambiar el *pipeline* de entrenamiento del sistema provisto, de manera considerable. Aun así, el *batch size* utilizado es el especificado de referencia en el trabajo original. La exploración para este modelo se adjunta en la tabla C.5

Tabla C.1: Parámetros de entrenamiento de modelos Simple Baselines, Bottom-Up HRNet y PoseAE sobre la base de datos COCO.

Modelo	Épocas	*Batch size*	*Learning rate*	*Learning rate steps*
Simple Baselines	140	32	0,001	[90,120]
Bottom-Up HRNet	140	12	0,001	[90,120]
PoseAE	320	32	0,0002	[200]

Tabla C.2: Exploración de *learning rate* y *batch size* para Bottom-Up HR-Net, durante el *finetuning*.

Learning rate	*Batch size*	*AP*	*AR*
0,00075	4	0,590	0,680
0,00075	8	0,636	0,719
0,00075	12	0,607	0,690
0,00075	16	0,604	0,689
0,00100	4	0,425	0,525
0,00100	**8**	**0,644**	**0,724**
0,00100	12	0,492	0,562
0,00100	16	0,622	0,706
0,00125	4	0,558	0,648
0,00125	8	0,617	0,702
0,00125	12	0,628	0,713
0,00125	16	0,638	0,721

Tabla C.3: Exploración de *learning rate* y *batch size* para Simple Baselines, durante el *finetuning*.

Learning rate	*Batch size*	*AP*	*AR*
0,00075	8	0,805	0,835
0,00075	16	0,803	0,833
0,00075	32	0,807	0,836
0,00075	56	0,811	0,839
0,00100	8	0,780	0,813
0,00100	16	0,785	0,816
0,00100	**32**	**0,815**	**0,842**
0,00100	56	0,805	0,836
0,00125	8	0,796	0,827
0,00125	16	0,801	0,831
0,00125	32	0,811	0,841
0,00125	56	0,800	0,831

Tabla C.4: Exploración de *learning rate* y *batch size* para PoseAE, durante el *finetuning*.

Learning rate	*Batch size*	*AP*	*AR*
0,00015	8	0,716	0,789
0,00015	16	0,704	0,795
0,00015	32	0,731	0,807
0,00015	48	0,721	0,799
0,00020	8	0,675	0,773
0,00020	**16**	**0,754**	**0,820**
0,00020	32	0,712	0,794
0,00020	48	0,717	0,794
0,00025	8	0,695	0,797
0,00025	16	0,727	0,804
0,00025	32	0,727	0,800
0,00025	48	0,736	0,805

Tabla C.5: Exploración de *learning rate* y cantidad de iteraciones de entrenamiento para OpenPose, durante el *finetuning*.

Iteraciones	*Learning rate*	*AP*	*AR*
4000	1,50000E-04	0,658	0,724
6000	1,50000E-04	0,669	0,732
2000	1,00000E-04	0,649	0,713
4000	1,00000E-04	0,665	0,727
6000	1,00000E-04	0,644	0,715
2000	5,00000E-05	0,650	0,724
4000	**5,00000E-05**	**0,679**	**0,735**
6000	5,00000E-05	0,669	0,732