PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

SEGMENTACIÓN SEMÁNTICA DE ESCENAS URBANAS DE LA PROVINCIA DE HUAMANGA

Tesis para optar el título profesional de Ingeniero Electrónico

AUTOR:

Lui Gustavo Pasapera Huaman

ASESOR:

Donato Andrés Flores Espinoza

Lima, agosto, 2024

Informe de Similitud

Yo, Donato Andres Flores Espinoza, docente de la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú, asesor(a) de la tesis titulada, SEGMENTACIÓN SEMÁNTICA DE ESCENAS URBANAS DE LA PROVINCIA DE HUAMANGA del autor Lui Gustavo Pasapera Huamán dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 16 %. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 18/08/2024.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima, 18 de agosto de 2024

<table>
<tr><td colspan="2">Apellidos y nombres del asesor:
Flores Espinoza, Donato Andres</td></tr>
<tr><td>DNI:
06017817</td><td rowspan="2">Firma
  </td></tr>
<tr><td>ORCID:
https://orcid.org/0000-0003-2092-7666</td></tr>
</table>

RESUMEN

La presente tesis se enfoca en la identificación y clasificación de objetos en escenas urbanas de la provincia de Huamanga, explorando un entorno diferente al de las ciudades desarrolladas y otras bases de datos existentes. Se estudiarán las escenas urbanas de Huamanga para segmentar imágenes en 7 clases de datos: personas, vehículos, motociclistas, edificios, veredas, pistas y otros, que incluyen detalles de cielo y cables de energía eléctrica. El enfoque principal de la tesis estará centrado en la visión por computadora, específicamente en la segmentación semántica para la clasificación de objetos. Para ello, se emplearán arquitecturas de aprendizaje profundo pre-entrenadas adaptadas a Deeplabv3+, y se utilizarán imágenes de la provincia de Huamanga como base de datos local.

La investigación se inicia con un análisis del estado del arte, destacando la importancia de la clasificación de objetos en escenas urbanas y los beneficios del aprendizaje profundo en comparación con métodos tradicionales. Se enfatiza la necesidad de utilizar bases de datos locales sobre las existentes, así como la base teórica para la clasificación de imágenes locales utilizando Deeplabv3+ y redes de aprendizaje profundo mediante la transferencia de aprendizaje. Posteriormente, se describe el diseño, la recopilación y el enfoque de la base de datos locales en comparación con conjuntos de datos como Imagenet y CityScapes, utilizando la arquitectura Deeplabv3+ junto con redes de aprendizaje profundo en los datos locales. Finalmente, se presentan los resultados basados en el incremento del número de datos, analizando la precisión, el Índice de Jaccard (IoU) y el mBFScore tanto a nivel global como por clase, junto con un análisis comparativo con la base de datos Cityscapes. Se proporcionan tablas sumarias que verifican los resultados de cada red de aprendizaje profundo y se propone hardware para dispositivos capaces de ejecutar tareas de segmentación semántica.

Dedicado a mis padres Lidia y David por su gran dedicación, enseñanzas y su apoyo incondicional

Agradecimiento por su gran apoyo y enseñanzas a mi asesor Andrés Flores

“Puede que haya perdido todo, pero jamás dejaré de pelear por lo que creo”

Akira Toriyama

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

La tecnología actual ha estrechado su relación con la visión por computadora y la clasificación de imágenes, generando avances notables en la capacidad de las computadoras para interpretar el entorno que las rodea. En este contexto, la segmentación semántica ha surgido como una herramienta poderosa que no solo permite la detección de objetos en imágenes, sino también la comprensión de su contexto y significado [1], [2]. Sin embargo, muchos entornos y conjuntos de datos existentes no se adaptan adecuadamente a localidades específicas [54], [55], lo que resalta la importancia de crear y evaluar conjuntos de datos locales para contribuir al avance tecnológico.

Estudios recientes, como los llevados a cabo en Huailai, China [56] y Chennai, India [57], han demostrado que el uso exclusivo de conjuntos de datos generales puede no ser suficiente para obtener resultados óptimos al segmentar imágenes locales. Por lo tanto, se busca mejorar la precisión en la identificación de características locales mediante la implementación de la arquitectura Deeplabv3+ y datos específicos de la región. Además, la experiencia de investigaciones como las realizadas en Bandung, Indonesia [50], resalta la importancia del uso de conjuntos de datos locales para obtener resultados más precisos. Esta perspectiva orienta la propuesta centrada en el estudio de los escenarios urbanos de Huamanga, lo que a su vez contribuye a la investigación en entornos andinos. De esta manera, facilita una comprensión más profunda de las necesidades particulares de la región y promueve el desarrollo de soluciones más efectivas y contextualizadas.

El presente trabajo de investigación se enfoca en la identificación y clasificación de objetos en escenas urbanas de la provincia de Huamanga, utilizando segmentación semántica con Deeplabv3+ y arquitecturas de aprendizaje profundo pre-entrenadas. Se emplearán imágenes locales como base de datos para segmentar imágenes en 7 clases de

datos, incluyendo personas, vehículos, motociclistas, edificios, veredas, pistas y otros, con detalles como cielo y cables de energía eléctrica.

El primer capítulo aborda la relevancia de la clasificación de objetos en escenas urbanas y se destacan las ventajas del aprendizaje profundo en comparación con los métodos tradicionales. Además, se enfatiza en la importancia de utilizar e integrar bases de datos locales sobre las ya existentes. Finalmente, se detallan los objetivos y alcances del presente trabajo de investigación.

En el segundo capítulo se abordan fundamentos teóricos sobre la utilización y creación de conjuntos de datos locales, así como su aplicación en el reentrenamiento mediante la transferencia de aprendizaje en una arquitectura de segmentación semántica Deeplabv3+. Esta arquitectura utiliza redes de aprendizaje profundo como clasificadores. Finalmente, se presenta el modelo de solución propuesto.

En el tercer capítulo comprende los diseños y criterios para la creación de un conjunto de datos local de la provincia de Huamanga, así como las fases del proceso de entrenamiento y métricas de evaluación a considerar.

En el cuarto capítulo se presentan las pruebas y resultados, comenzando con el aumento progresivo del número de datos de entrenamiento y su impacto en la eficiencia del entrenamiento. Se analizan los resultados en términos de precisión, Índice de Jaccard (IoU) y mBFScore tanto a nivel global como por clase, además de realizar un análisis comparativo con el conjunto de datos Cityscapes. Se examinan las tablas sumarias y se llevan a cabo estudios sobre posibles hardware para ejecutar tareas de segmentación semántica. Finalmente, la última sección presenta las conclusiones, recomendaciones y posibles direcciones para futuras investigaciones.

CAPÍTULO 1

Importancia de la clasificación de objetos en escenas urbanas.

Para una computadora, la detección de objetos se realiza a través del procesamiento de imágenes digitales capturadas por cámaras, las cuales sirven como medio de adquisición de datos. Este proceso se realiza a través de la extracción de características y clasificación de las imágenes y/o objetos. En este sentido, la clasificación de objetos en escenas o entornos urbanos es una tarea fundamental para aplicaciones como la conducción autónoma, gestión de tráfico, planificación urbana, entre otros, ya que permite reconocer y diferenciar entre diferentes tipos de objetos en su entorno.

1.1 La clasificación de objetos en escenas urbanas.

En el ámbito de la segmentación semántica y clasificación de objetos en escenas urbanas, se han empleado tradicionalmente métodos convencionales para abordar este desafío. Entre ellos se encuentran SVM (*Support Vector Machine*), *Random Forest* y KNN (*K-Nearest Neighbors*), que han sido utilizados en diversas aplicaciones para la identificación y categorización de objetos en imágenes urbanas. Sin embargo, la implementación de estos métodos tradicionales plantea complicaciones significativas. Factores como la variación en la forma de los objetos, las condiciones de iluminación y otros aspectos impactan de manera considerable tanto en el proceso de entrenamiento como en la aplicación de dichas técnicas [6]. La crítica principal a estos enfoques radica en que no resulta óptimo depender de métodos que requieren la extracción manual de características, como es el caso de SVM, Random Forest y KNN. Este proceso de extracción manual puede resultar tedioso y consume mucho tiempo. Además, la precisión de la clasificación se ve directamente afectada por la calidad de las características extraídas [18].

Los enfoques del aprendizaje profundo eliminan en gran medida la necesidad de aplicar métodos tradicionales de caracterización de objetos al permitir que el modelo aprenda automáticamente las representaciones relevantes de los datos durante el entrenamiento. El aprendizaje profundo ha demostrado ser una técnica muy efectiva para la clasificación de objetos en escenas urbanas. Su integración en la visión por computadora tiene un alto potencial de poder generar soluciones sólidas y de mayor accesibilidad para las nuevas tendencias, tales como la conducción autónoma, gestión de tráfico y planificación urbana [3], [4], [5], [7], [8], [45].

1.2 El aprendizaje profundo y los conjuntos de datos en escenas urbanas.

El aprendizaje profundo se fundamenta en la construcción de redes neuronales profundas, las cuales aprenden automáticamente a partir de grandes cantidades de datos. De esta manera, se obtiene un conocimiento más extenso a nivel computacional y entrenamiento por parte de una arquitectura de aprendizaje profundo. No obstante, si la cantidad de datos de entrenamiento es limitada, el modelo puede sobreajustarse a los datos de entrenamiento y no generalizar de manera efectiva nuevos datos. En ese sentido, existe una gran diferencia entre utilizar datos tales como *Cityscapes Dataset* e *Imagenet,* y los datos locales que diferencien de estos.

La efectividad limitada de arquitecturas de aprendizaje profundo entrenadas con datos diferentes al querer estudiar un área local específica se debe, en parte, a la variación de los objetos presentes. La diversidad en color, forma y otras características específicas de un área local puede no estar bien representada en modelos entrenados con conjuntos de datos más generales. Esto destaca la necesidad de considerar la variabilidad local al implementar arquitecturas de aprendizaje profundo, ya que la adaptación a las

particularidades específicas de un entorno local puede ser crucial para un rendimiento óptimo.

El análisis detallado de la clasificación de objetos en entornos asiáticos, como China y Taiwán, como se destaca en los estudios [54], [55], resalta la importancia de considerar las particularidades locales al estudiar ciudades específicas. Estas investigaciones señalan que las bases de datos existentes pueden no abordar completamente los desafíos únicos presentes en estos entornos, lo que subraya la importancia de realizar análisis específicos para ciudades que enfrentan mayores retos y diferencias culturales.

La implementación de redes pre-entrenadas en nuevos entornos, como áreas rurales o distintas a las habituales, puede presentar un desafío significativo debido a la variabilidad en los datos. En consecuencia, puede resultar esencial entrenar la red en un conjunto de datos específico que refleje las características únicas del entorno local de interés. Este enfoque permite adaptar el modelo a las peculiaridades de la región, optimizando así su desempeño y precisión en la clasificación de objetos en entornos nuevos o diferentes a través de un conjunto de datos específico.

1.3 Justificación

La clasificación de imágenes en entornos urbanos se ha convertido en un tema de gran interés en la visión por computadora. El desafío clave en la visión por computadora es la capacidad de los sistemas para interpretar y comprender su entorno para tomar decisiones informadas y seguras. En este contexto, el aprendizaje profundo se ha establecido como una herramienta esencial para la clasificación de imágenes en escenas urbanas [19], [46]. Una de las técnicas más utilizadas es la segmentación semántica, que permite la identificación y clasificación precisa de objetos en una imagen. La segmentación

semántica basada en aprendizaje profundo ha demostrado resultados impresionantes en la identificación y clasificación de objetos en imágenes en tiempo real en escenas urbanas [20], [21], [22]. Dentro de las arquitecturas de segmentación semántica, DeepLabV3+ ha demostrado ser capaz de adaptarse a diferentes entornos urbanos, lo que la hace adecuada para su uso en sistemas tales como la conducción autónoma, gestión de tráfico y planificación urbana. Además, ha demostrado ser muy eficiente en términos de uso de recursos computacionales, lo que la hace adecuada para su implementación en sistemas con aplicaciones a la segmentación de objetos en entornos urbanos [45], [46].

Una de las desventajas al implementar métodos de aprendizaje profundo o sus extensiones en arquitecturas de segmentación semántica es que, al utilizar una red pre entrenada en un conjunto de datos específico que difiere del entorno nuevo, se enfrenta al desafío de que los modelos previamente entrenados pueden no ser adecuados para identificar con precisión los elementos en estas nuevas imágenes, dado que carecen de información adecuadas para los nuevos datos. A continuación, se mostrarán algunos casos de otros países en los cuales implementaron nuevos conjuntos de datos locales en contraste a los conjuntos de datos con el uso de Deeplabv3+:

El primer caso, en el estudio [56] propusieron una nueva base de datos de Huailai, China, que discernía con respecto a bases de datos existentes. Se contrastaron diferencias entre alturas de edificios y el efecto que estos tenían sobre el resto de clases presentes en tales entornos urbanos; en este estudio tuvieron mejoras del 7.42% al 18.82% en cuanto a precisión con respecto a utilizar bases de datos existentes.

En el segundo caso, en el estudio [57] se contrastaron diferencias culturales y la distribución de clases en Chennai, India. En este estudio de segmentación semántica con

Deeplabv3+ se hizo con el fin de generar resultados más precisos. De este estudio destaca notablemente la diferencia y detección en calzadas con bordes casi invisibles.

Como último caso, en el estudio [50] destacaron la importancia del uso de su propia base de datos de la ciudad de Bandung, Indonesia, en referencia a la base de datos de CityScapes, teniendo mejores resultados en la precisión en base al uso de la segmentación semántica con Deeplabv3+.

De manera análoga a los estudios vistos, si se toma como ejemplo el modelo de red entrenada con el conjunto de datos “CamVid”, en la ciudad de Cambridge, los resultados son los siguientes:

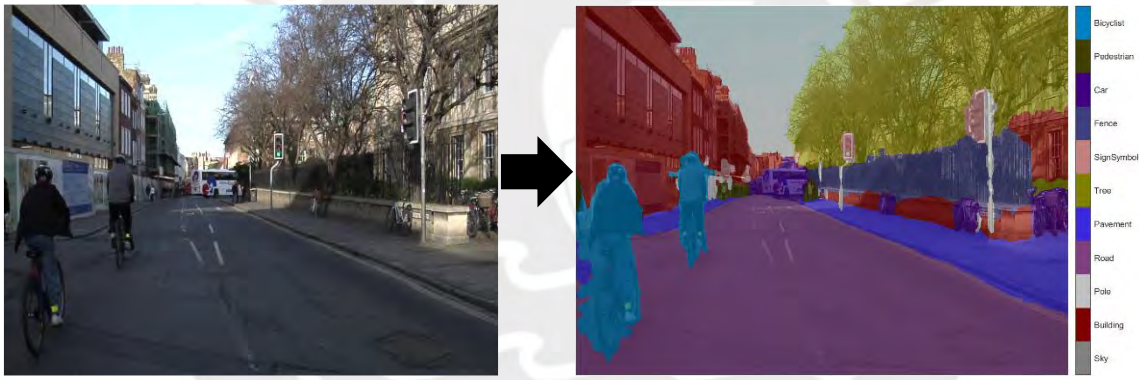

Figura 1.1. Segmentación semántica en escenas urbanas de Cambridge con red existente. Fuente: Propia.

Como se puede apreciar en la Figura 1.1, la segmentación de las escenas urbanas en la ciudad de Cambridge se lleva a cabo de manera muy eficiente en comparación con la imagen propuesta en el lado izquierdo. Esto se debe a que el modelo de red ha sido entrenado con escenas urbanas que presentan rasgos y características similares a los de dicha ciudad, utilizando el conjunto de datos "Camvid". Este hecho demuestra la adaptabilidad del modelo a este tipo específico de entornos.

Por otro lado, cuando se realiza la predicción sobre escenas urbanas de la provincia de Huamanga, el resultado es el siguiente:

Figura 1.2. Segmentación semántica en escenas urbanas de Huamanga con red existente. Fuente: Propia.

Como se puede apreciar en la Figura 1.2, la segmentación de las escenas urbanas en la provincia de Huamanga se realiza de manera diferente a la de la Figura 1.1, ya que segmenta incorrectamente clases que no corresponden con respecto a la imagen propuesta en el lado izquierdo. Es importante destacar que estas pruebas se realizaron con varias imágenes de la provincia de Huamanga, donde se obtuvieron resultados similares.

Los resultados evidencian que el modelo de red ha sido entrenado con escenas urbanas que presentan rasgos y características distintas a las de la provincia de Huamanga, lo cual demuestra la limitada adaptabilidad en este tipo de entornos. Por esta razón, se subraya la necesidad de estudiar cada tipo de entorno, ya que los métodos y entrenamientos aplicados en experimentos previos fueron diseñados para sistemas específicos [9]. Además, el estudio de nuevos entornos contribuye al conocimiento sobre la adaptabilidad de los modelos en contextos diversos.

Otra de las principales desventajas de utilizar métodos de aprendizaje profundo o sus extensiones en arquitecturas de segmentación semántica se encuentra en la necesidad de contar con una gran cantidad de datos en el entrenamiento para que la arquitectura de red

pueda ser eficiente y de alto rendimiento. Esta problemática se puede solucionar mediante la transferencia de aprendizaje. La transferencia de aprendizaje es una forma eficiente de utilizar el conocimiento adquirido en un conjunto de datos grande y general para mejorar el rendimiento en un conjunto de datos más específico y limitado [23]. Para implementar el estudio en un nuevo entorno, es importante recolectar nuevos datos que, si bien no necesariamente son abundantes, deben reflejar la variabilidad de un entorno determinado. Así, se puede reentrenar la red y generar arquitecturas de red más eficientes y de alto rendimiento sin la necesidad de utilizar grandes cantidades de datos.

En la presente tesis, se plantea implementar la arquitectura de red de segmentación semántica Deeplabv3+ basada en redes de aprendizaje profundo a fin de estudiar un entorno de la comunidad andina frente a otros de primer mundo tal como Cityscapes dataset, Camvid u otros. El estudio se centrará en la provincia de Huamanga de la ciudad de Ayacucho a fin de contribuir con la investigación como parte de la clasificación de objetos en entornos andinos.

1.3 Objetivos

1.3.1 Objetivo general

- Evaluar la segmentación de objetos con Deeplabv3+ en las escenas urbanas de la provincia de Huamanga a través de redes pre-entrenadas de aprendizaje profundo.

1.3.2 Objetivos específicos

- Recopilar y preparar de datos de la provincia de Huamanga para el conjunto de datos de entrenamiento, validación y pruebas.

- Implementar la arquitectura Deeplabv3+ utilizando redes pre-entrenadas ResNet18, ResNet50, MobileNetV2 y Xception mediante el uso del programa Matlab.
- Entrenar y evaluar la arquitectura de segmentación semántica modificada junto con el nuevo conjunto de datos utilizando el programa Matlab.
- Comparar el desempeño de las redes ResNet18, ResNet50, MobileNetV2 y Xception aplicables a la arquitectura Deeplabv3+ utilizando el nuevo conjunto de datos.
- Realizar estudios para la implementación de hardware en la arquitectura de segmentación semántica.

CAPÍTULO 2

Fundamentos de la visión por computadora

2.1 Visión general

La visión por computadora es un campo interdisciplinario que combina principios de procesamiento de imágenes y aprendizaje automático para permitir que las computadoras analicen y comprendan datos visuales. Desde la adquisición de imágenes hasta la interpretación de escenas complejas, la visión por computadora abarca una amplia gama de conceptos esenciales, incluida la extracción de características, la segmentación de imágenes, el reconocimiento de patrones y la comprensión de la estructura tridimensional. Estos fundamentos son esenciales para aplicaciones que van desde la detección de objetos en tiempo real hasta el análisis de imágenes médicas, y constituyen la base de la capacidad de las máquinas para interactuar y comprender el mundo visual que las rodea.

En cuanto al conjunto de datos, es de suma relevancia que estos varíen en forma, tamaño, texturas, brillo, etc. Esto se puede realizar tanto manualmente como a nivel de computadora. Para el primer caso, se adquieren los datos en diferentes entornos o lugares de tal manera que se diferencien entre sí, lo que contribuye a que la arquitectura de red sea más eficiente y robusta al momento de variar las características de un determinado lugar. Por ejemplo, se podrían tomar datos de una determinada avenida en diferentes horarios para obtener imágenes con diferentes brillos, sombras y formas. En el segundo caso, se podrían modificar los datos mediante procesos computacionales que incluyan cambios en el brillo, texturas y formas mediante el uso de funciones de rotación, brillo, entre otros. A este método se le llama el aumento de datos que se define como un conjunto de técnicas que generan datos sintéticos a partir de un conjunto de datos existente. Estos nuevos datos suelen incluir pequeñas variaciones respecto a los datos originales, con el propósito de hacer que las predicciones del modelo sean consistentes ante esos cambios.

Además, estos datos sintéticos pueden representar combinaciones entre ejemplos que están distantes en el espacio de características, lo que sería difícil de inferir de otro modo [60].

Para la clasificación de objetos se utilizará la segmentación semántica que es una técnica avanzada en el campo de la visión por computadora que implica la asignación de etiquetas semánticas a píxeles individuales en una imagen, con el objetivo de discernir y diferenciar regiones y objetos específicos según sus propiedades visuales. Dentro de las arquitecturas de segmentación semántica, Deeplabv3+ se destaca por su alta capacidad para capturar detalles finos y contextuales; además de tener excelentes resultados en pruebas con diversos conjuntos de datos [24], [58], [59]. Su arquitectura de red permite el uso de redes pre-entrenadas de redes de aprendizaje profundo. Las redes de aprendizaje profundo que se pueden integrar en esta arquitectura de red son las de Mobilenetv2, Resnet18, Resnet50, Xception e Inceptionresnetv2. De estas, las que tuvieron mayor performance en aplicaciones de transferencia de aprendizaje aplicables a segmentación de escenas urbanas fueron las de Mobilenetv2, Resnet18, Resnet50, Xception con Resnet50 en el primer lugar, Resnet18 en el segundo, Mobilenetv2 en el tercero y Xception en el último lugar, llegando a tener resultados en la precisión entre el 88 al 95% tanto en la etapa de prueba como entrenamiento [24], [25], [26], [27], [28].

2.2 Visión por computadora

El término 'visión por computadora', dentro del campo de la clasificación de objetos, se basa en un conjunto de técnicas y el uso de modelos orientados al procesamiento, análisis e interpretación de imágenes digitales en un entorno o medio específico que se desea estudiar [10]. Sus aplicaciones pueden dirigirse hacia la medicina, la agricultura, los vehículos autónomos, el análisis de tráfico, la seguridad pública, entre otros. En este

contexto, se presentarán una serie de conceptos esenciales que constituyen y complementan la visión por computadora y sus aplicaciones en la segmentación de imágenes. A continuación, se detallan los conceptos a tratar:

2.2.1 Aprendizaje profundo

El término aprendizaje profundo se refiere a un tipo de aprendizaje automático que permite a las máquinas realizar tareas de clasificación directamente a partir de datos digitales. Esta técnica utiliza redes neuronales con múltiples capas para detectar patrones en los datos. Estas redes están diseñadas para imitar la estructura del cerebro humano y son capaces de aprender de manera automática a partir de ejemplos y datos de entrada proporcionados. En particular, las redes neuronales convolucionales son altamente eficaces en el reconocimiento de objetos. El aprendizaje profundo se ha convertido en una herramienta indispensable en el campo de la visión por computadora, ya que permite la detección y clasificación de objetos de manera altamente eficiente. De manera general, cada arquitectura de red posee 3 partes principales: La capa de entrada, las capas ocultas y la capa de salida [61]. La detección de características posee capas tales como las convolucionales, unidad lineal rectificada y de agrupación. Su estructura y orden jerárquico depende de cada arquitectura.

- Convolución. – Esta capa toma matrices de un determinado tamaño y las opera a través de un producto escalar con otra matriz llamada kernel. Este procedimiento, genera a su salida una nueva matriz.
- Unidad lineal rectificada (ReLU). – Asigna el valor de cero (0) a los valores negativos.

- Agrupación máxima (Max Pooling). –Simplifica la salida con un muestreo no lineal en la forma de reducir la cantidad de parámetros que una red necesita para realizar el proceso de aprendizaje.

La clasificación utiliza las siguientes capas.

- Capa totalmente conectada (Fully connected). – genera un número 'N' de conexiones que representas el número de clases que la red podrá predecir.
- Capa de función exponencial normalizada (Softmax). – Proporciona una salida para la clasificación.

2.2.2 Transferencia de aprendizaje

La transferencia de aprendizaje implica utilizar conocimientos existentes para generar nuevos aprendizajes. En el campo de la inteligencia artificial, es posible reutilizar una red neuronal pre-entrenada al reentrenarla con clases similares dentro de una categoría específica. Por ejemplo, si se dispone de una red pre-entrenada para la detección de ciertas variedades de flores, pero no incluye todas las variedades, es posible reutilizar esta red para detectar las nuevas variedades mediante un proceso de reentrenamiento.

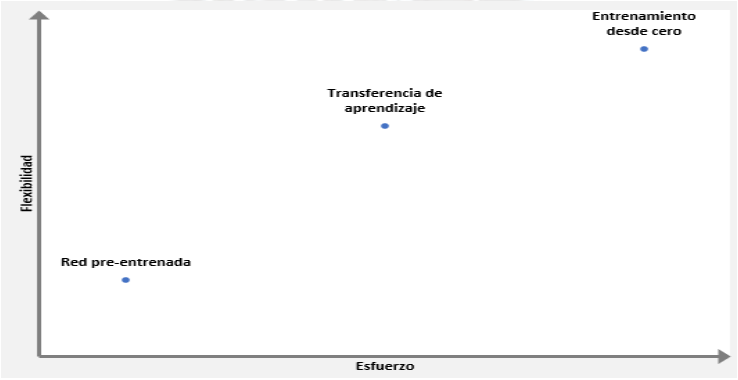

Figura 2.1. Métodos para la aplicación del aprendizaje profundo. Fuente: Propia.

En la Figura 2.1 se muestra una representación con respecto a la flexibilidad y esfuerzo con lo que respecta al uso de una red pre-entrenada, transferencia de aprendizaje y entrenamiento desde cero. La implementación de solo una red pre-entrenada es de poco esfuerzo; sin embargo, posee poca flexibilidad ya que se depende directamente del proceso de entrenamiento de una red, sin la posibilidad de cambio o reajuste. El entrenamiento desde cero implica una mayor flexibilidad; su uso e implementación es favorable siempre y cuando no existan redes pre-entrenadas que puedan clasificar imágenes de interés por clase; sin embargo, se perdería el aprendizaje, medido en el peso por cada clase, de una determinada arquitectura de red. En tal sentido, con la transferencia de aprendizaje se aprovecha el conocimiento adquirido en tareas previas para mejorar la eficiencia, la generalización y el rendimiento de los modelos en nuevas tareas. Para un mejor entendimiento, el proceso de transferencia de aprendizaje se detalla a continuación:

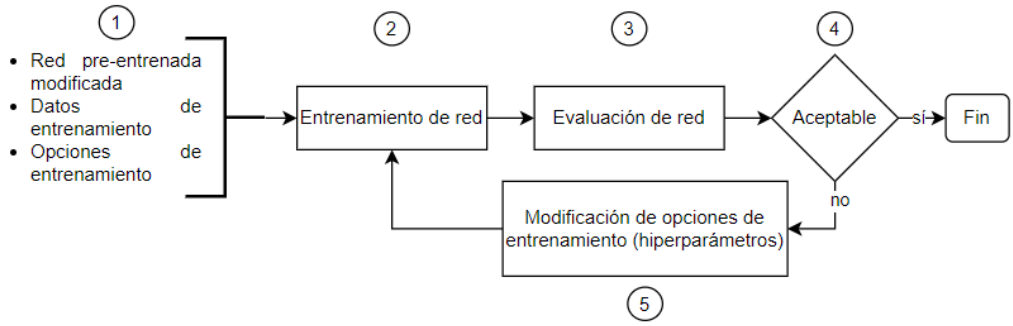

Figura 2.2. Proceso para la transferencia de aprendizaje. Fuente: Propia.

En la primera parte de la Figura 2.2 se presenta la red de aprendizaje pre-entrenada modificada, junto con los datos de entrenamiento y las opciones de entrenamiento correspondientes. Estos elementos conforman una arquitectura ya modificada, la cual consiste en una adaptación de las capas de la arquitectura de segmentación

semántica Deeplabv3+. Esta adaptación implica cambios en el número y tipo de clases de interés, así como en el tamaño de las imágenes utilizadas para el entrenamiento. Para llevar a cabo estas modificaciones, se utiliza el conocimiento adquirido sobre las redes de aprendizaje profundo, en particular en lo referente a la capa de clasificación a nivel de píxel. Posteriormente, la red se somete a un proceso de reentrenamiento y se evalúa su precisión, el índice de superposición de píxeles (IoU) y el puntaje de segmentación (BFScore) tanto en métricas globales como por clase, hasta alcanzar un resultado satisfactorio en la etapa de entrenamiento.

2.2.3 Aumento de datos

El aumento de datos es una estrategia utilizada para mejorar la capacidad y el rendimiento de los modelos de aprendizaje automático, especialmente en tareas de visión por computadora. El aumento de datos implica aplicar transformaciones diversas y aleatorias a los datos de entrenamiento existentes con el objetivo de crear nuevas instancias de datos que sean perceptualmente similares a las originales, pero que presenten variaciones que ayuden al modelo a generalizar mejor. En cuanto a la variación a nivel de computadora del conjunto de datos, se muestra a continuación, funciones que permiten obtener imágenes rotadas, reflejadas, trasladas y escaladas.

2.2.3.1Rotación. - Es el proceso en el cual se permite cambiar el sentido de una imagen a través de un ángulo especificado. Dentro del preprocesamiento de imágenes digitales, se utiliza para obtener una mejor vista de la imagen. Se muestra a continuación su forma matemática.

$$Xn = cos(\theta) * (Xa - Xo) - sin(\theta) * (Ya - Yo) + Xo \qquad (1)$$

$$Yn = sin(\theta) * (Xa - Xo) + cos(\theta) * (Ya - Yo) + Xo \quad (2)$$

donde *(Xo,Yo)* son las coordenadas del centro de rotación, $\boldsymbol{\theta}$ es el ángulo de rotaciones en sentido horario que tienen los ángulos positivos, *(Xa, Ya)* representan la posición inicial de un elemento de una imagen y *(Xn,Yn),* la posición nueva para un elemento de una imagen [11].

2.2.3.2Reflexión. - Es el proceso que permite hacer el efecto espejo para una determinada imagen. Se muestra a continuación su forma matemática.

- Reflexión sobre el eje vertical de abscisa Xo.

$$Xn = -Xa + (2 * Xo) \quad (3)$$

$$Yn = Ya \quad (4)$$

- Reflexión sobre el eje vertical de abscisa Yo.

$$Xn = Xa \quad (5)$$

$$Yn = -Ya + (2 * Yo) \quad (6)$$

- Reflexión sobre el eje orientado en cualquier dirección arbitraria $\boldsymbol{\theta}$ que pasa por (Xo, Yo).

$$Xn = Xa + 2 * \Delta * (-sin(\theta)) \quad (7)$$

$$\Delta = (Xa - Xo) * sin(\theta) - (Ya - Yo) * cos(\theta) \quad (8)$$

donde *(Xo,Yo)* son las coordenadas de las abscisas por las cuales se hará la reflexión, $\boldsymbol{\theta}$ es el ángulo de rotaciones en sentido horario que tienen los ángulos positivos, *(Xa, Ya)* representan la posición inicial de un elemento

de una imagen y *(Xn,Yn)*, la posición nueva para un elemento de una imagen [11].

2.2.3.3Traslación. - Es el proceso en el cual se le asigna a cada elemento de la imagen, una nueva posición representada en la imagen de salida los pixeles de los datos. Se muestra a continuación su forma matemática.

$$Xn = Xa + Xx \tag{9}$$

$$Yn = Ya + Yy \tag{10}$$

donde *(Xx,Yy)* representan el desplazamiento para cada elemento de la imagen, *(Xa, Ya),* la posición inicial de un elemento de una imagen y *(Xn,Yn),* la posición nueva para un elemento de una imagen [11].

2.2.3.4Escalamiento. - Se utiliza para ampliar o reducir el tamaño o parte de una determinada imagen. Puede ampliarse mediante dos métodos: la replicación de pixeles o la interpolación. El primero reemplaza cada pixel de la imagen original por un grupo de pixeles del mismo valor, mientras que el segundo, lo hace a través de un grupo expandido de píxeles.

2.2.4 Segmentación semántica

La segmentación semántica es el proceso de clasificar cada píxel en una imagen que pertenece a una determinada clase y, por lo tanto, puede considerarse como un problema de clasificación por píxel. Además, tiene la principal característica del reconocimiento de múltiples objetos presentes en una determinada imagen o cuadro de video. Dentro de estás arquitecturas de red, se encuentran la Deeplabv3+ que tiene alta performance en la detección de objetos en carreteras [24]. Esta arquitectura se puede reutilizar para el entrenamiento a través de redes

de aprendizaje profundo. Es decir, aprovechan el entrenamiento previo por parte de las redes de aprendizaje profundo y ajustan su propia arquitectura al entrenamiento en el campo de la segmentación semántica.

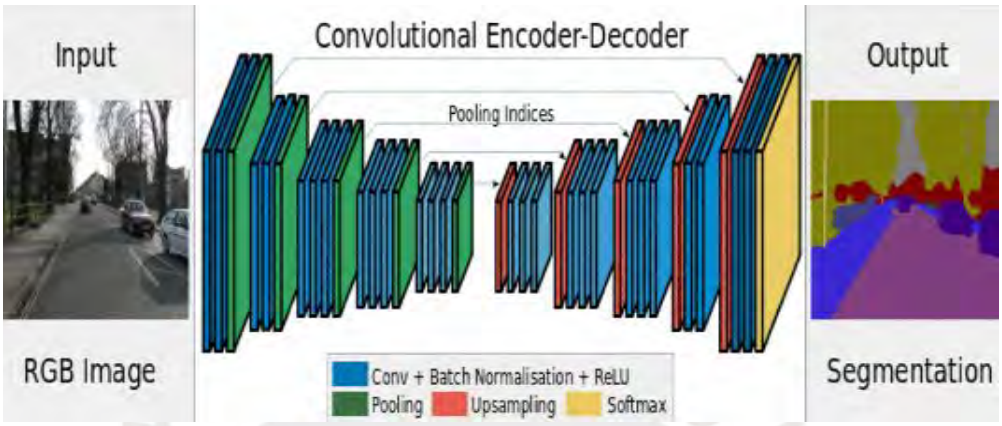

Figura 2.3. Estructura de una arquitectura Deeplabv3+ basada en Segnet. Fuente: [12].

En la Figura 2.3 se logra apreciar una estructura codificadora (*encoder*) – decodificadora (*decoder*). El codificador extrae características de una imagen digital a través de una secuencia de filtros cada vez más estrechos y profundos. En el codificador utiliza arquitecturas de aprendizaje profundo pre-entrenadas con el fin de implementar la transferencia de aprendizaje y aprovechar los pesos de cada red. El aprendizaje profundo se sitúa como clasificador. La capa de muestreo (*Upsampling*) realiza el proceso inverso a la de agrupación que se explicó en el punto 2.2.1. Esto hace que el decodificador realice un proceso inverso al codificador hasta llegar a una salida de imagen segmentada.

2.3 Modelo de solución

La propuesta del modelo de solución se muestra en la Figura 2.4. Se basa en la utilización de los conocimientos adquiridos por cada red de aprendizaje profundo, como Resnet18, Resnet50, Mobilenetv2 y Xception. El objetivo es aprovechar su acceso e

implementación en la arquitectura de segmentación semántica Deeplabv3+. El entorno que se estudiará es la provincia de Huamanga, donde se recolectarán datos variados para hacer que la red sea más robusta frente a las variaciones del entorno y sus características. Asimismo, se emplearán funciones para el aumento de datos, como rotación, traslación y reflexión, como se explica en el punto 2.2.3. Por otro lado, se tendrán en cuenta las opciones de entrenamiento tales como el minibatchsize, el número de épocas, entre otros, según el Anexo B. Se variarán los valores de las opciones de entrenamiento según requerimientos mediante pruebas experimentales en la etapa de entrenamiento de la red. En caso de ser necesario se incrementarán el número de imágenes para el entrenamiento adecuado de cada red de aprendizaje profundo en su implementación a la segmentación semántica. Seguidamente se evaluará cada red y finalmente, se analizará y propondrá hardware existente para la aplicación de las arquitecturas de segmentación semántica.

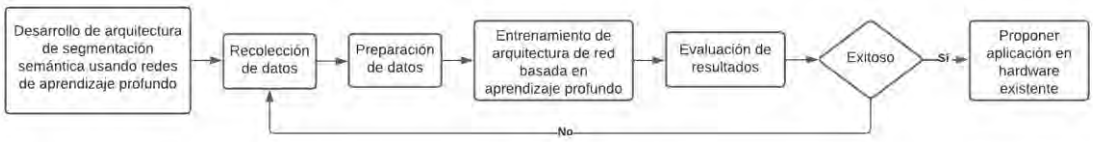

Figura 2.4. Modelo de solución. Fuente: Propia

CAPÍTULO 3

Diseño y entrenamiento de conjunto de datos local

El presente capítulo muestra los criterios y procedimientos necesarios para realizar el proceso de segmentación semántica de datos locales en comparación con los grandes conjuntos de datos, tales como Imagenet y CitySpaces, utilizando arquitecturas pre-entrenadas de aprendizaje profundo. El programa a usar será Matlab versión 2022b y se utilizará una computadora con características principales de RAM de 16 GB y GPU NVIDIA GeForce GTX 3050.

3.1 Base de datos

Con respecto a la base de datos, es importante considerar las variaciones entre entornos, las imágenes recurrentes que se clasificarán, el número necesario y admisible de imágenes para el proceso de transferencia de aprendizaje, así como las métricas y el etiquetado manual necesarios para el desarrollo de la base de datos.

La adquisición de datos desde diferentes fuentes proporciona una mayor variedad en cuanto a luminosidad, contornos y calidad de la imagen [13], [15], [16], [17]. Para este propósito, se utilizarán dos cámaras distintas, ambas con la misma resolución para evitar el efecto de *aliasing* que podría ocurrir al redimensionar las imágenes. La primera captura se realizará con un celular Motorola G60S, que graba en una resolución de video de 360 x 640 x 3. La segunda captura se llevará a cabo con un celular Xiaomi Redmi Note 10S, con la misma resolución de video. Posteriormente, se extraerán imágenes de los videos con intervalos que oscilan entre 12 a 15 cuadros.

3.1.1 Tipos de entorno

En la provincia de Huamanga, se recolectarán datos tanto del centro de la ciudad como de los distritos alejados. Dentro de cada entorno se considerarán cambios

en luminosidad y efecto de sombras que son muy importantes para la creación de la base de datos. Se ha contemplado dentro de cada tipo de entorno, una diversidad de horarios que incluyen mañana, mediodía y tarde, con el fin de generar la mayor variabilidad de datos en cuanto a variaciones tales como brillos y sombras. La variación contribuirá a que el entrenamiento de la red de aprendizaje profundo sea más eficiente y robusto, permitiendo captar cada píxel de manera más precisa en relación con la segmentación semántica [14], [17]. Por lo tanto, se propone realizar un estudio diferenciado utilizando datos separados para la etapa de entrenamiento y la etapa de prueba. Para la etapa de entrenamiento se utilizarán datos del centro de la ciudad y para la etapa de prueba, se utilizarán datos de distritos alejados. Esto permitirá un análisis más preciso del rendimiento final de las nuevas redes creadas. A continuación, se detallan las características tanto del centro de la ciudad como de los distritos alejados en la provincia de Huamanga:

Figura 3.1. Escenas urbanas del centro de Huamanga por horario. Fuente. Propia.

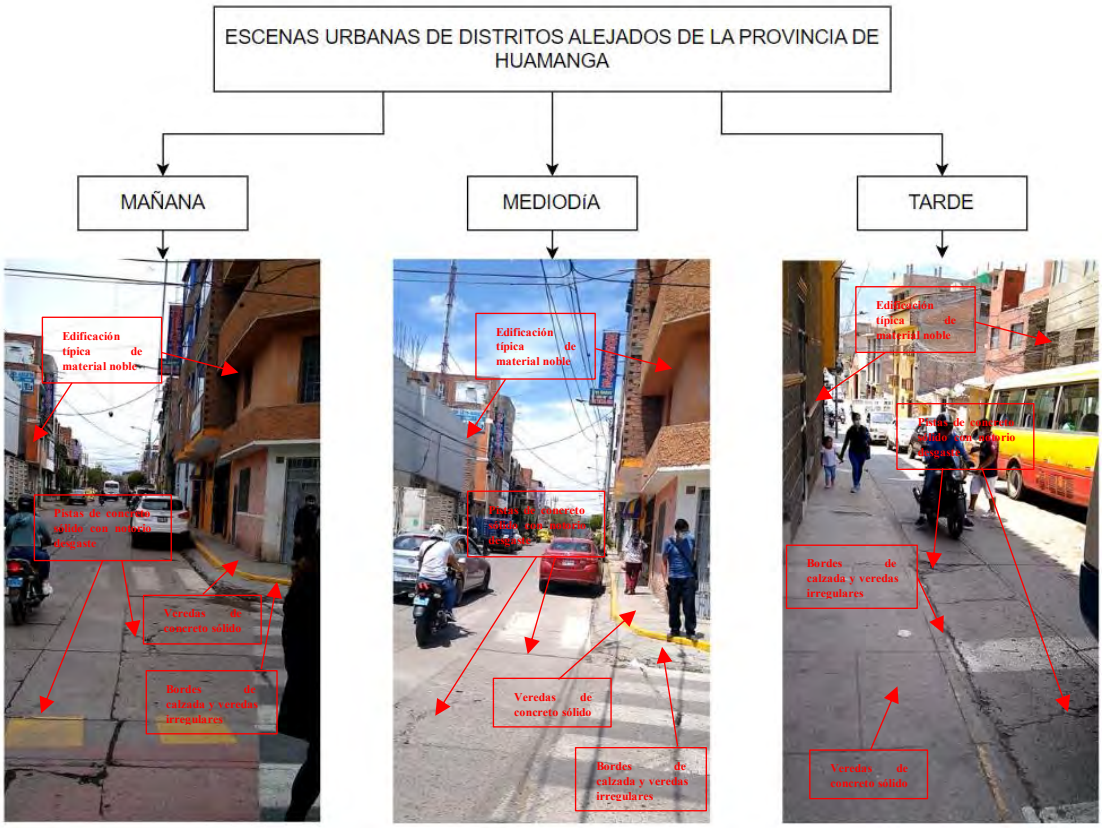

Figura 3.2. Escenas urbanas de los distritos alejados de Huamanga por horario. Fuente: Propia.

Como se puede apreciar en las Figuras 3.1 y 3.2, las escenas urbanas del centro de la ciudad presentan pistas con drenajes y estructuras más tratadas y conservadas en comparación con los distritos alejados. Además, mientras que el centro de la ciudad de Huamanga conserva características coloniales, los distritos alejados exhiben rasgos comunes de escenas urbanas.

Se observa en las Figuras 3.1 y 3.2 que, durante las mañanas, las sombras son casi nulas y la coloración de las pistas y los objetos es semioscura. Durante el mediodía, hay pocas sombras y una gran luminosidad. Por último, en la tarde, los objetos y las pistas presentan muchas sombras y una gran luminosidad.

3.1.2 Tipos de clase

Es crucial realizar un análisis empírico para determinar las clases específicas que se utilizarán según la variabilidad y el tipo de entorno a estudiar [50], [51]. Con respecto al tipo de clases, las categorías comunes para la clasificación de objetos en escenas urbanas incluyen carreteras, aceras, personas, ciclistas, vehículos como camiones, automóviles, autobuses y motocicletas, así como edificios, muros, vegetación y el cielo [47], [48], [49].

Existe la posibilidad de fusionar clases dentro de otras que integren el mismo grupo y otras que menos representadas según un entorno de estudio de acuerdo a las necesidades que se requieran [52], [53]; por ejemplo, si se tienen camiones, buses y autos, se pueden integrar dentro de una misma clase como vehículo dentro de la nueva base de datos. Así mismo, si una determinada clase está siempre presente en otra más significativa y no afecta el proceso de clasificación de objetos, esta puede ser incluida dentro de la misma clase más significativa [52], [53]; por ejemplo, clasificar la placa de un carro dentro de la clase "carro" no afectaría el proceso de clasificación de objetos, ya que la clase más significativa sería "carro".

Para la presente tesis, se seleccionaron las siguientes clases, basadas en los dos párrafos anteriores: Peatón, carro, motociclista, edificio, pista, vereda y otros. Las primeras seis clases agruparon categorías similares dentro de la misma clase. La última clase, denominada "otros", agrupó una categoría poco significativa y siempre presente, como las líneas de tensión eléctrica, dentro de la clase cielo.

3.1.3 Número de imágenes

Con respecto al número de imágenes, se vio en la literatura leída [55], [57], [50] que no hay un número exacto de imágenes mínimas o máximas a considerar. En

tal sentido se propone realizar un enfoque empírico considerando variabilidad entre los entornos y datos por tomar. Así mismo, se irá aumentando conforme se realicen los avances y progresos con respecto al entrenamiento. Con respecto al entrenamiento y a la validación, se utilizarán datos de calles o avenidas diferentes a fin de evitar efectos de sobreajuste al momento del entrenamiento. Así mismo, para la etapa de entrenamiento se utilizarán las imágenes junto con el proceso denominado "aumento de datos" o por su nombre en inglés *"data augmentation"*. Este procedimiento facilitará y brindará una mayor eficiencia y variedad en cuanto a entrenamiento.

3.1.4 Preprocesamiento

El proceso de transferencia de aprendizaje requiere trabajar con las métricas mínimas establecidas por las redes pre-entrenadas de aprendizaje profundo. En nuestro caso, las redes Resnet18, Resnet50, Mobilenetv2 y Xception utilizan medidas de 224x224x3 y 299x299x3, donde "3" representa la escala de colores (RGB). Los datos recopilados tendrán medidas de 360x640x3, lo cual justifica que no será necesario redimensionar las imágenes, ya que cumplen con el valor mínimo requerido.

3.1.5 Etiquetado manual

Para una aplicación de segmentación semántica, además de considerar una base de datos de imágenes adquiridas con cámaras, es necesario contar con una base de datos que incluya datos de imágenes segmentadas. Para este proceso, se utilizará la herramienta ImageLabeler de Matlab, que se encuentra en la sección de aplicaciones (APPs). El número total de datos de imágenes segmentadas debe ser igual al número de imágenes tomadas por las cámaras. Esto se debe a que se

realizará la segmentación manual de las imágenes obtenidas a través de las cámaras.

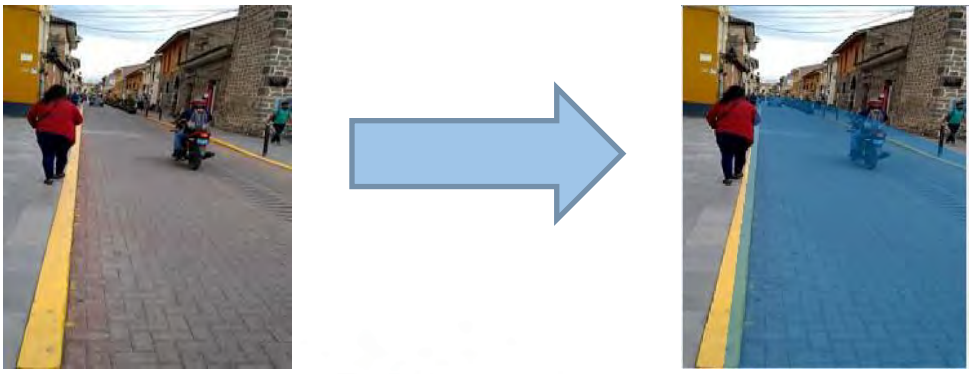

Figura 3.3. Etiquetado manual de imágenes. Fuente: Propia

En la Figura 3.3 se observa la segmentación manual de la imagen. En este caso se muestra una imagen en proceso de segmentación manual de la pista.

3.2 Entrenamiento

La presente tesis utilizará cuatro arquitecturas de aprendizaje profundo pre-entrenadas en nuevas configuraciones de segmentación semántica basadas en Deeplabv3+. A partir de estos tres puntos, se llevarán a cabo sus respectivas evaluaciones y comparaciones para determinar qué arquitectura se ha adaptado mejor a la clasificación de las clases previamente mencionadas (personas, carros, veredas, pistas, edificios, motociclistas y otros).

El proceso de entrenamiento comenzará con la importación de las bases de datos de imágenes en formato RGB e imágenes segmentadas mencionadas en el punto 3.1. Posteriormente, se crearán bases de datos locales utilizando funciones como imageDatastore y pixelLabelDatastore. Estos datos se separarán en conjuntos de entrenamiento, validación y prueba, diferenciados según lo planteado en la toma de datos. Luego, se realizará el aumento de datos para la etapa de entrenamiento utilizando la función imageDataAugmenter. A continuación, se importará una de las redes de

aprendizaje profundo pre-entrenadas y se acoplará dentro de la red de segmentación semántica Deeplabv3+. Finalmente, se ajustarán los parámetros de entrenamiento y se llevará a cabo el entrenamiento con todos los datos correctamente cargados. A continuación, se muestra el diagrama de bloques general del proceso de entrenamiento:

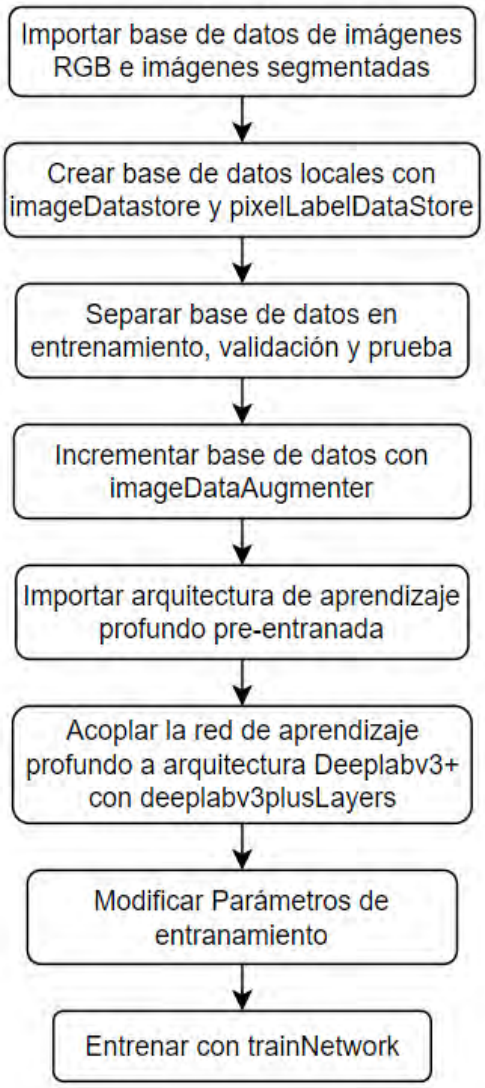

Figura 3.4. Diagrama de bloques de entrenamiento. Fuente: Propia.

Para el nivel de redes de aprendizaje profundo, se importarán por separado las redes proporcionadas por el programa Matlab, que incluyen Resnet18, Resnet50, Mobilenetv2 y Xception. Estas redes están pre-entrenadas para la clasificación de clases como pistas, veredas, edificios, carros, motociclistas y personas. Además, es posible modificar la

arquitectura interna de cada red para adaptarla a una arquitectura de segmentación semántica. En cuanto a la transferencia de aprendizaje, se utilizará cada una de estas redes de manera individual para desarrollar aplicaciones de segmentación semántica.

En relación con el número de imágenes de entrenamiento, se empleará la función imageDataAugmenter proporcionada por Matlab. El siguiente paso será la modificación de los parámetros de entrenamiento. En este nivel, los parámetros se ajustarán en términos del número de clases, imágenes, tipo de equipo (capacidad computacional) utilizado para el proceso de entrenamiento y el tiempo de entrenamiento estimado. Los parámetros de entrenamiento incluyen:

SGDM: Un modelo discriminativo basado en la mezcla gaussiana exhibe una capacidad de ajuste flexible.

Learning rate: Esto permite que la red aprenda rápidamente con una tasa de aprendizaje inicial más alta, al tiempo que puede encontrar una solución cercana al óptimo local una vez que la tasa de aprendizaje desciende.

ValidationData y ValidationPatience: Evita que la red se sobreajuste en el conjunto de datos de entrenamiento.

Minibatchsize: Utilizado para reducir el uso de memoria durante el entrenamiento.

CheckpointPath: Está configurado en una ubicación temporal. Este par nombre-valor permite guardar los puntos de control de la red al final de cada época de entrenamiento.

MaxEpochs: Establece un número de épocas de entrenamiento. Su valor se quiere incrementar la exactitud de la red.

Estos parámetros se acoplan de manera conjunta con la función de Matlab llamada trainingOptions. De manera adicional, dentro de esta función, se agrega la base de datos generada llamada datos de validación. Finalmente, en cuanto a entrenamiento, se pasa a entrenar la arquitectura de red con todos los parámetros mencionados anteriormente en este punto. Para realizar esta tarea, se utiliza la función otorgada por Matlab llamada trainNetwork.

3.3 Evaluación

Tabla 3.1. Métricas de clasificación. Fuente: Propia.

Nombre	Definición	Fórmula
Accuracy	La precisión indica el porcentaje de píxeles correctamente identificados para cada clase.	$\left(\frac{TP+TN}{TP+TN+FP+FN}\right)$
Global - accuracy	Es la proporción de píxeles clasificados correctamente, independientemente de la clase, respecto al número total de píxeles.	$\left(\frac{TP+TN}{TP+TN+FP+FN}\right)$
BFScore	Indica qué tan bien se alinea el límite predicho de cada clase con el límite real.	$\left(\frac{2\times TP}{2\times TP+FP+FN}\right)$
IoU (Intersection over unión)	También conocido como coeficiente de similitud de Jaccard, es la métrica más utilizada para la segmentación semántica y calcula la intersección de imágenes binarias.	$\left(\frac{TP}{TP+FP+FN}\right)$
Weighted - IoU	Es el promedio de cada clase, ponderado por el número de píxeles de esa clase. Se utiliza esta métrica si las imágenes tienen clases de tamaño desproporcionado.	$\left(\frac{\sum_{i=1}^{N} w_i \times \frac{TP_i}{TP_i+FP_i+FN_i}}{\sum_{i=1}^{N} w_i}\right)$

De las fórmulas:

- TP: número de predicciones negativas que son incorrectas.

- TN: número de predicciones negativas que son correctas.
- FP: número de predicciones positivas que son incorrectas.
- FN: número de predicciones negativas que son incorrectas.
- N: Número de clases
- w_i: peso asociado a la clase i
- TP_i: número de predicciones negativas que son incorrectas asociado a la clase i.
- FP_i : número de predicciones positivas que son incorrectas asociado a la clase i.
- FN_i: número de predicciones negativas que son incorrectas asociado a la clase i.

Se realizará el estudio a través de las métricas o sus similares de tal manera de tener varias propuestas y realizar su respectiva evaluación.

CAPÍTULO 4

Pruebas y resultados

4.1 Consideraciones iniciales

Las consideraciones iniciales para la etapa de pruebas y resultados consistió en utilizar 800 y 1400 imágenes para el proceso de entrenamiento con el fin de evaluar si se realizaba correctamente la segmentación semántica.

4.2 Pruebas y resultados iniciales

El número total de imágenes fue de 800 para esta etapa inicial. Dentro de este conjunto, muchas no contenían muchas variaciones con respecto a la toma de datos. Se optó por utilizar la arquitectura Resnet18 en esta primera etapa debido a que su entrenamiento era una de las que requería menos tiempo en comparación con otras arquitecturas de aprendizaje profundo. Se configuró un minibatchsize de 8 y se establecieron 15 épocas de entrenamiento. A continuación, se mostrarán pruebas con sus respectivos análisis de resultados:

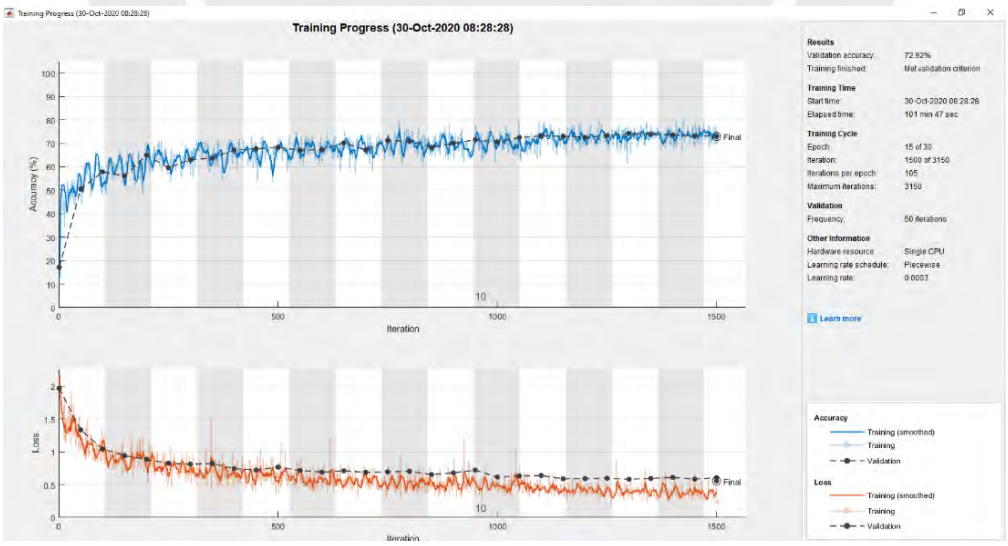

Figura 4.1. Gráfica del proceso de entrenamiento con 800 imágenes basada en la arquitectura Resnet18 con Minibatchsize de 8 y Epocas de 15. Fuente: Propia.

Tabla 4.1. Métricas por clase con 800 imágenes de arquitectura basada en Resnet18 con Minibatchsize de 8 y Epocas de 15. Fuente: Propia.

	IoU	MeanBFScore
Road	0.83656	0.55378
Sidewalk	0.51377	0.37254
Building	0.39164	0.17594
Pedestrian	0.21238	0.21163
Others	0.55509	0.38284
Car	0.29463	0.1547
MotorCycle	0.52035	0.40396

En la Figura 4.1, se observa un resultado de más del 72.92%, el cual fue considerado demasiado bajo. Al analizar este caso con la Tabla 4.1, se pudo observar que cada clase no estaba siendo segmentada correctamente debido a su bajo nivel de IoU. Como ejemplo ilustrativo, se presenta a continuación una imagen segmentada por esta primera red.

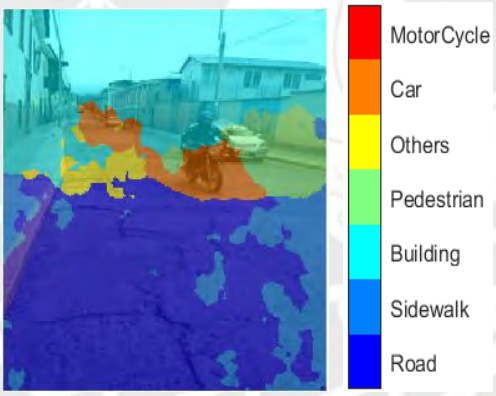

Figura 4.2. Resultado de segmentación semántica con un conjunto de datos de 800 imágenes con Minibatchsize de 8 y Epocas de 15. Fuente: Propia.

El análisis en la Figura 4.2 muestra que tanto la precisión como el IoU (Intersección sobre Unión) no son óptimos, lo que se evidencia en errores como la incorrecta segmentación de una motocicleta como un carro. Ante esta situación, se tomaron medidas para mejorar el rendimiento del modelo. Se incrementó el número de imágenes a 1400, incluyendo una variedad más amplia de tipos de imágenes, con el objetivo de mejorar la precisión final del entrenamiento de la red. Tras ajustar estos parámetros, se procedió con el entrenamiento de la red y se obtuvieron los siguientes resultados:

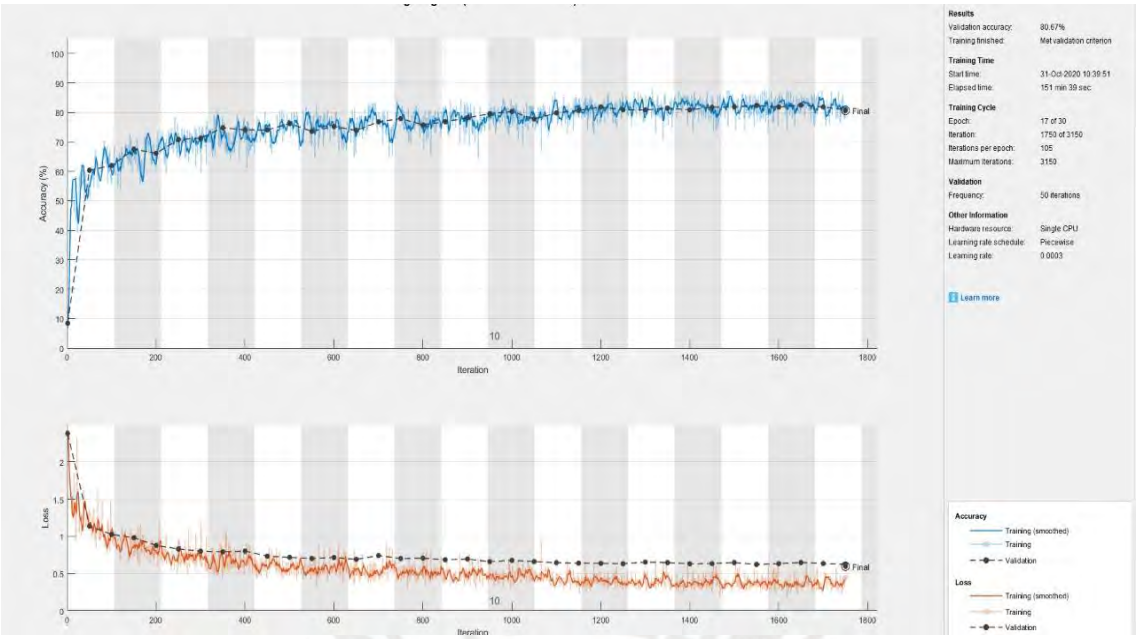

Figura 4.3. Gráfica del proceso de entrenamiento con 1400 imágenes basada en la arquitectura Resnet18 con Minibatch de 25 y Epocas de 50. Fuente: Propia.

Tabla 4.2. Métricas por clase con 1400 imágenes de arquitectura basada en Resnet18 con Minibatchsize de 25 y Epocas de 50. Fuente: Propia.

	IoU	MeanBFScore
Road	0.86113	0.57716
Sidewalk	0.57248	0.43267
Building	0.71705	0.48348
Pedestrian	0.24187	0.27157
Others	0.69448	0.47898
Car	0.44653	0.24786
MotorCycle	0.26184	0.19279

Aunque la precisión experimentó un aumento de aproximadamente 8 puntos, como se observa en la Figura 4.3, al analizar el Índice de Jaccard (IoU) de la Tabla 4.2, se notó un impacto considerable. Algunas clases disminuyeron significativamente su IoU, alcanzando valores demasiado bajos, mientras que otras solo experimentaron un aumento leve. Este fenómeno se reflejó en el resultado de la segmentación semántica, que se presenta a continuación:

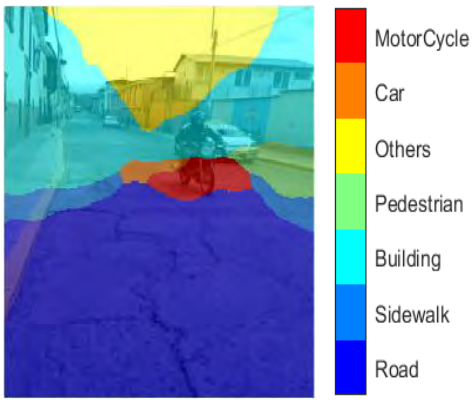

Figura 4.4. Resultado de segmentación semántica con un conjunto de datos de 1400 imágenes con Minibatchsize de 25 y Epocas de 50. Fuente: Propia.

El aumento en el IoU para algunas clases, como "motocicleta" y "otros", observado en la Figura 4.4, no mejoró el rendimiento general de la segmentación semántica. A pesar de intentar ajustes como aumentar el número de épocas a 150 o 200, la precisión no se vio afectada, lo que llevó a reconsiderar el conjunto de datos utilizado.

4.3 Consideraciones finales

Las consideraciones iniciales para la etapa de pruebas y resultados consistieron en utilizar un total de 3600 imágenes para el proceso de entrenamiento, validación y pruebas. De esta base de datos, se destinaron 2000 imágenes para el entrenamiento, 800 imágenes para la validación y 800 imágenes para las pruebas. Las imágenes de entrenamiento y validación fueron seleccionadas del centro de la ciudad y tomadas de diferentes avenidas entre sí para garantizar la diversidad de escenarios y evitar el sobreajuste.

Por otro lado, las imágenes de prueba se tomaron de los distritos alejados del centro de la ciudad. Cabe destacar que todos los datos se recolectaron en diferentes horarios con el fin de capturar la variabilidad de condiciones de entorno. Con respecto a las opciones de entrenamiento, se consideraron de manera general un Minibatchsize de 8 y un número de épocas de 50.

4.4 Resultados finales

A continuación, se presentarán los resultados finales para las redes de segmentación semántica basadas en Deeplabv3+, junto con las redes de aprendizaje profundo Resnet18, Resnet50, Xception y Mobilenetv2. Este análisis incluirá gráficas que ilustren el proceso de entrenamiento, tablas que detallen las métricas tanto a nivel global como por clase, y finalmente, se mostrarán los resultados de las imágenes correspondientes a la etapa de prueba en diferentes horarios. El objetivo es comparar la efectividad y precisión de cada red utilizando las mismas imágenes en todas las pruebas relacionadas con cada red de aprendizaje profundo.

4.4.1 Resnet18

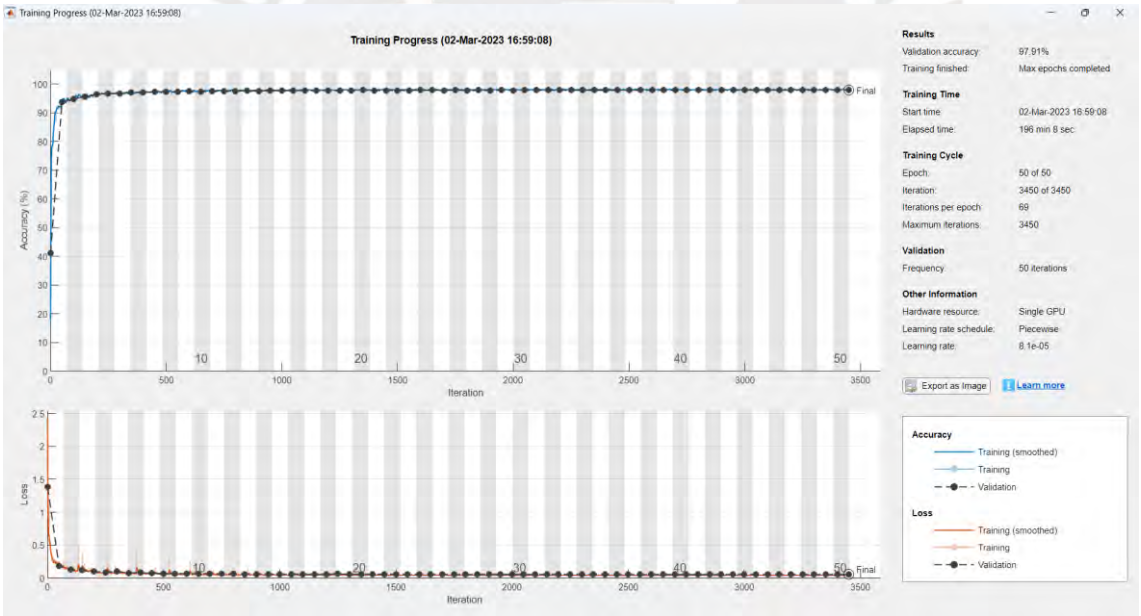

Figura 4.5. Gráfica del proceso de entrenamiento basada en la arquitectura Resnet18. Fuente: Propia.

Durante la etapa de entrenamiento, se alcanzó una exactitud del 97.91%, como se muestra en la Figura 4.5. Este valor es notablemente alto al considerar la métrica de exactitud y en base a la literatura leída y citada en el capítulo 2. En relación de la gráfica *Loss* vs *Iteration* se observa una disminución uniforme y constante, indicando un proceso de

entrenamiento adecuado. Además, se nota que tanto el conjunto de entrenamiento como el de validación exhiben curvas eficientes, sin evidencia de subajuste ni sobreajuste.

Tabla 4.3. Métricas de prueba para arquitectura basada en Resnet18. Fuente: Propia.

GlobalAccuracy	MeanAccuracy	MeanIoU	WeightedIoU	MeanBFScore
0.91941	0.90504	0.7903	0.85557	0.72404

En la Tabla 4.3 se presenta una exactitud global de 0.92 y una exactitud promedio de 0.90 para la etapa de prueba. Esta diferencia en la precisión se debe al hecho de que el conjunto de datos de prueba se seleccionó de manera diferente al conjunto de datos de entrenamiento. Los resultados obtenidos son considerados aceptables según lo establecido en la literatura revisada en el punto 2. En cuanto al IoU promedio, se registra un valor de 0.79, con una IoU ponderada de 0.85. Esto indica que algunas clases tienen una mayor representación en términos de datos o píxeles en el total de las imágenes segmentadas. El BFScore promedio obtenido es de 0.72, lo que indica que los contornos de las segmentaciones se asemejan a la máscara de referencia en esta medida.

Tabla 4.4. Métricas por clase de arquitectura basada en Resnet18. Fuente: Propia.

	Accuracy	IoU	MeanBFScore
Road	0.95145	0.86273	0.70295
Sidewalk	0.73029	0.67458	0.63909
Car	0.87002	0.75602	0.69601
Pedestrian	0.95254	0.71769	0.61851
Building	0.94023	0.91917	0.8431
MotorCycle	0.90902	0.63737	0.61718
Others	0.9817	0.96456	0.94114

El análisis proporcionado en la Tabla 4.4 revela que la clase "others", que representa el cielo y cables de energía eléctrica, obtiene los mayores valores de exactitud, IoU y mBFScore, con un destacado rendimiento en la segmentación. Por otro lado, la clase

"sidewalk" registra el valor más bajo de exactitud, mientras que la clase "motorcycle" presenta el menor IoU y la clase "pedestrian" el menor mBFScore.

En el caso de la clase "others", se destaca una exactitud del 0.98, lo que indica una alta precisión en la clasificación de esta clase. El IoU de 0.96 sugiere que hay una baja presencia de píxeles clasificados incorrectamente en comparación con otras clases, y el mBFScore de 0.94 indica una adecuada delimitación de los contornos.

Para la clase "road", aunque se obtienen valores altos de exactitud e IoU, el mBFScore es significativamente más bajo debido a la mayor cantidad de píxeles en esta clase, lo que afecta la precisión de los contornos clasificados.

La clase "pedestrian" muestra una alta exactitud, pero valores más bajos en IoU y mBFScore, debido a la menor representatividad de píxeles para esta clase en comparación con otras.

En contraste, la clase "sidewalk" presenta resultados más bajos en todos los parámetros, lo que sugiere una menor precisión en la segmentación debido a la variabilidad de datos en la etapa de prueba. A continuación, se analizarán imágenes con los datos de prueba:

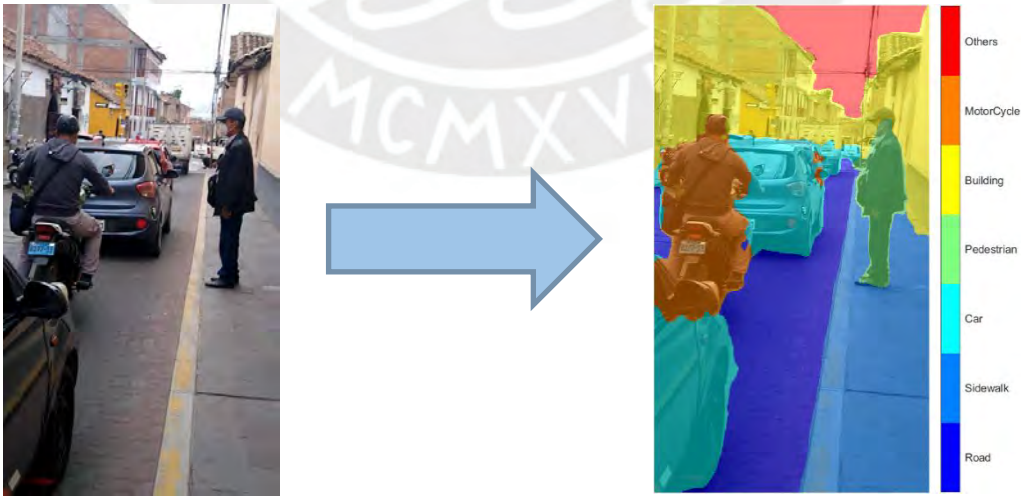

Figura 4.6. Segmentación basada en la red Resnet18 – Horario de la mañana. Fuente: Propia

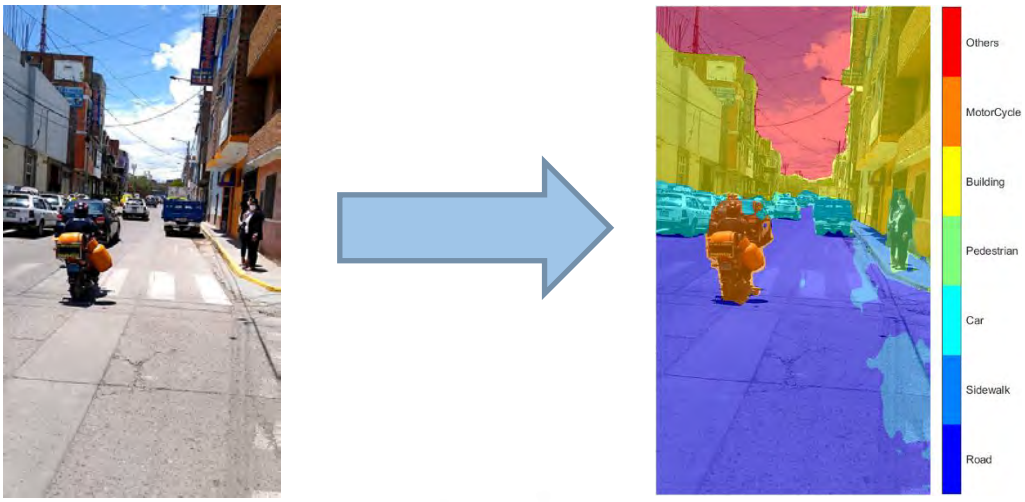

Figura 4.7. Segmentación basada en la red Resnet18 – Horario del medio día. Fuente: Propia

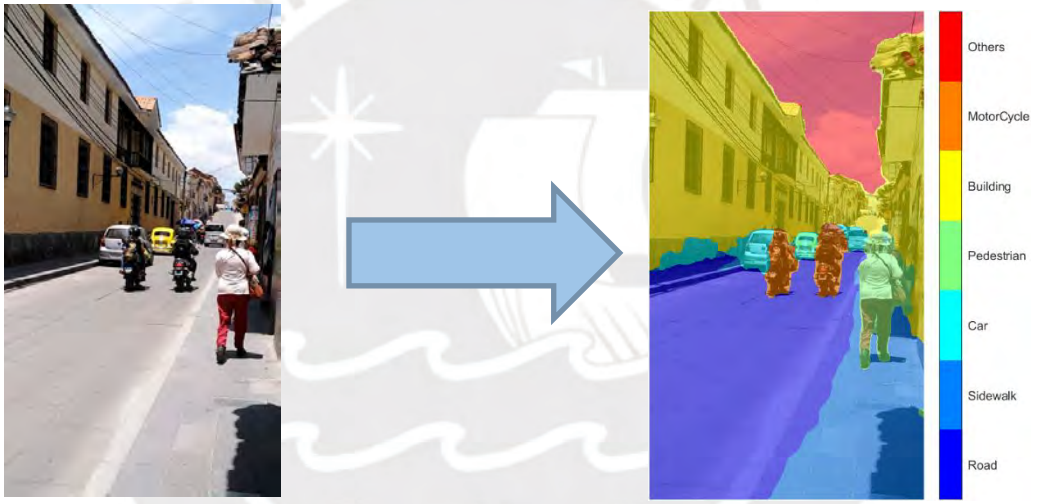

Figura 4.8. Segmentación basada en la red Resnet18 – Horario de la tarde. Fuente: Propia

En las Figuras 4.6, 4.7 y 4.8 se observa que las clases "others" y "building" presentan una mejor segmentación, lo cual se refleja en sus altos valores de exactitud, IoU y mBFScore durante la etapa de prueba.

Respecto a la clase "road", se logra una buena segmentación con base en sus valores de exactitud e IoU. Sin embargo, se nota la presencia de otras clases segmentadas sobre la misma clase y viceversa, con una tendencia a sobrepasar los bordes, lo que resulta en un mBFScore más bajo.

En cuanto a las clases "pedestrian", "car" y "motorcycle", se observa una buena exactitud para cada una. Sin embargo, también se identifica la presencia de otras clases segmentadas, lo cual se traduce en valores más bajos de IoU y mBFScore.

Por último, la clase "sidewalk" muestra la menor exactitud y valores bajos de IoU y mBFScore. Esto se evidencia en los casos 1 y 2, donde la segmentación presenta una baja precisión dentro de su propia clase, además de segmentar incorrectamente otras clases y no delimitar adecuadamente los bordes en comparación con las demás clases.

4.4.2 Resnet50

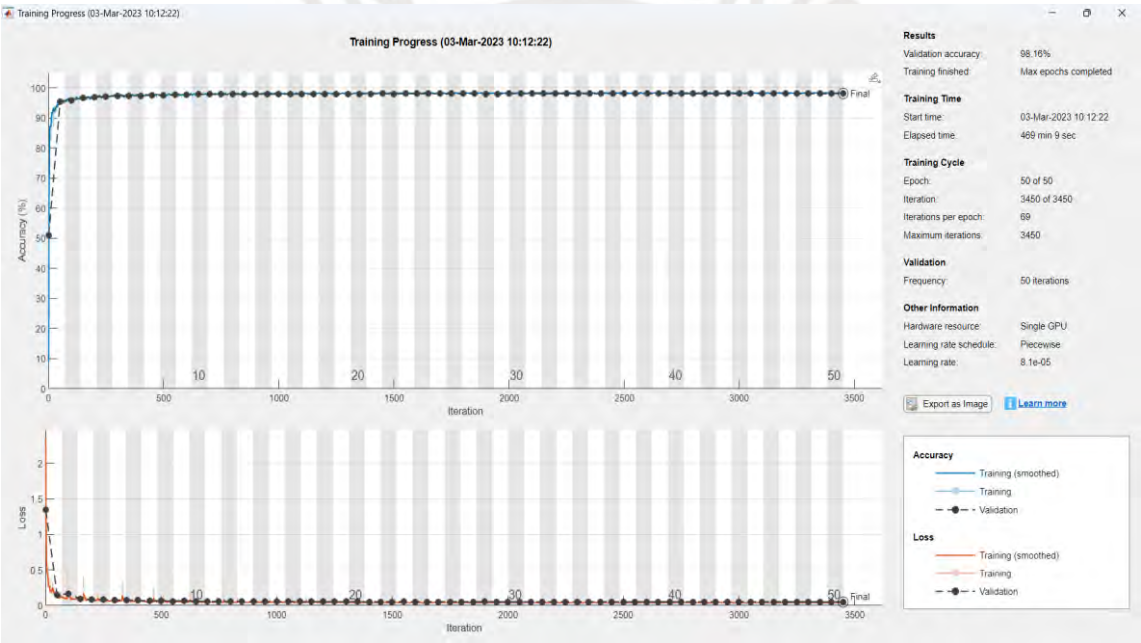

Figura 4.9. Gráfica del proceso de entrenamiento basada en la arquitectura Resnet50. Fuente: Propia

Durante la etapa de entrenamiento, se alcanzó una exactitud del 98.16%, como se muestra en la Figura 4.9. Este valor es notablemente alto al considerar la métrica de exactitud y en base a la literatura leída y citada en el capítulo 2. En relación de la gráfica *Loss* vs *Iteration* se observa una disminución uniforme y constante, indicando un proceso de entrenamiento adecuado. Además, se nota que tanto el conjunto de entrenamiento como el de validación exhiben curvas eficientes, sin evidencia de subajuste ni sobreajuste.

Tabla 4.5. Métricas globales de arquitectura basada en Resnet50. Fuente: Propia

GlobalAccuracy	MeanAccuracy	MeanIoU	WeightedIoU	MeanBFScore
0.92969	0.90577	0.79689	0.87345	0.74382

En la Tabla 4.5 se presenta una exactitud global de 0.93 y una exactitud promedio de 0.90 para la etapa de prueba. Esta diferencia en la precisión se debe al hecho de que el conjunto de datos de prueba se seleccionó de manera diferente al conjunto de datos de entrenamiento. Los resultados obtenidos son considerados aceptables según lo establecido en la literatura revisada en el punto 2. En cuanto al IoU promedio, se registra un valor de 0.79, con una IoU ponderada de 0.87. Esto indica que algunas clases tienen una mayor representación en términos de datos o píxeles en el total de las imágenes segmentadas. El BFScore promedio obtenido es de 0.74, lo que indica que los contornos de las segmentaciones se asemejan a la máscara de referencia en esta medida.

Tabla 4.6. Métricas por clase de arquitectura basada en Resnet50. Fuente: Propia

	Accuracy	IoU	MeanBFScore
Road	0.96013	0.88833	0.70064
Sidewalk	0.75018	0.70958	0.62915
Car	0.90348	0.795	0.6931
Pedestrian	0.85388	0.62462	0.71022
Building	0.96039	0.93605	0.86732
MotorCycle	0.93864	0.66202	0.67095
Others	0.9737	0.96264	0.93352

En base a los resultados de la Tabla 4.6, se observa que la clase con los valores más altos de exactitud, IoU y mBFScore es "others", que representa al cielo y los cables de energía eléctrica. Por otro lado, la clase con la menor exactitud es "sidewalk".

Para la clase "others", se obtiene una buena segmentación con una exactitud de 0.97, lo que indica una clasificación correcta en gran parte de esta clase. Además, el IoU de 0.96

sugiere una baja presencia de otras clases no pertenecientes a "others", y el mBFScore de 0.93 indica una correcta delimitación de la mayoría de los contornos.

En cuanto a la clase "building", se esperan resultados similares a los de "others", ya que presenta valores altos de exactitud (0.96), IoU (0.93) y mBFScore (0.86). Sin embargo, la segmentación semántica de "building" no será equitativa a la de "others", lo que se refleja en valores inferiores de IoU y mBFScore.

Para la clase "road", se observa una alta y aceptable exactitud e IoU, pero un mBFScore significativamente más bajo debido a la mayor cantidad de píxeles por imagen, lo que indica una segmentación menos precisa de los contornos.

En el caso de las clases "pedestrian", "car" y "motorcycle", se presenta una alta exactitud, pero valores más bajos de IoU y mBFScore, lo que sugiere una segmentación incorrecta de otras clases además de las mencionadas.

La clase "sidewalk" muestra resultados significativamente más bajos en precisión, IoU y mBFScore en comparación con otras clases, indicando una segmentación menos precisa y errores en los bordes de las imágenes segmentadas, lo cual puede atribuirse a la variabilidad en los datos de prueba. A continuación, se analizarán imágenes con los datos de prueba:

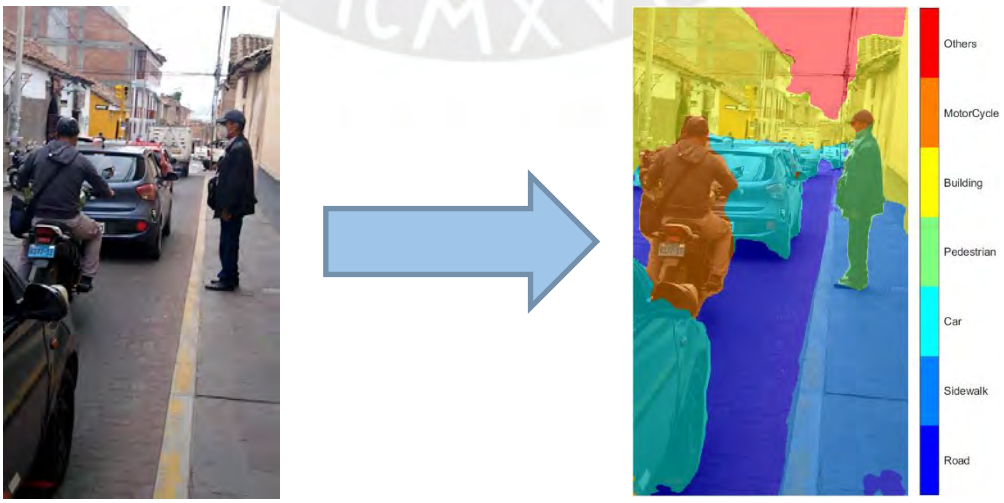

Figura 4.10. Segmentación basada en la red Resnet50 – Horario de la mañana. Fuente: Propia

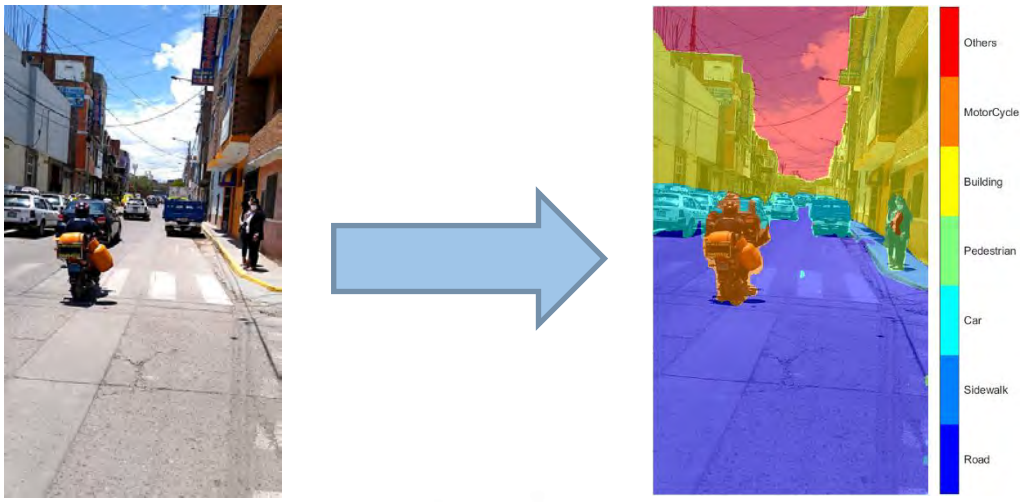

Figura 4.11. Segmentación basada en la red Resnet50 – Horario del mediodía. Fuente: Propia

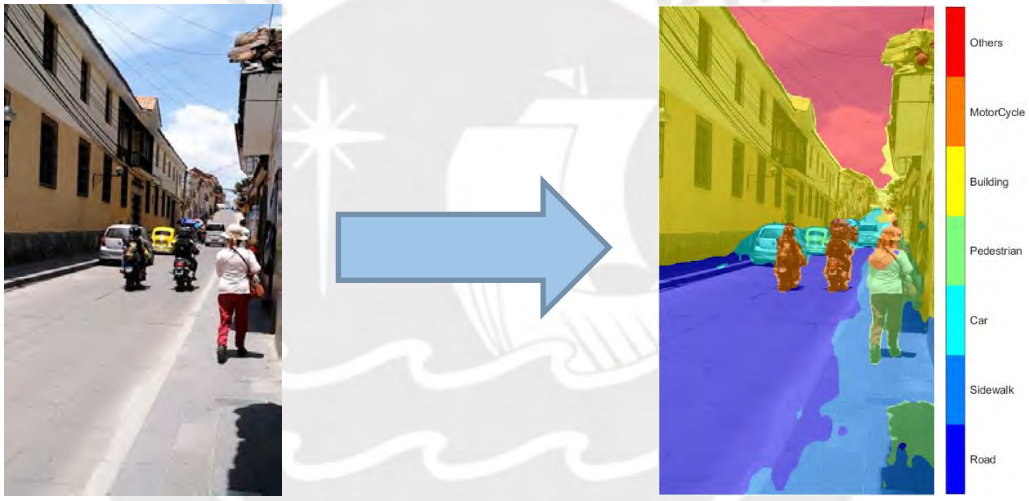

Figura 4.12. Segmentación basada en la red Resnet50 – Horario de la tarde. Fuente: Propia

En las Figuras 4.10, 4.11 y 4.12 se puede observar que, para los tres casos analizados, las clases "others" y "building" presentan una mejor segmentación, lo cual se refleja en sus altos valores de exactitud, IoU y mBFScore en la etapa de prueba. Estos valores indican una clasificación correcta y una delimitación precisa de los contornos para estas clases.

En cuanto a la clase "road", se aprecia una buena segmentación gracias a sus valores de exactitud e IoU, aunque se observa una superposición con otras clases y viceversa, lo que

resulta en un mBFScore más bajo debido a la falta de precisión en la delimitación de los contornos.

Para las clases "pedestrian", "car" y "motorcycle", se logra una buena exactitud, pero se observa la segmentación incorrecta de otras clases además de las mencionadas, lo que se refleja en los valores más bajos de IoU y mBFScore.

Por último, la clase "sidewalk" muestra los resultados más bajos en términos de exactitud, IoU y mBFScore. Se observa una segmentación incorrecta en los bordes de la clase y una superposición con otras clases, lo que indica una menor precisión en la delimitación de los contornos y una clasificación menos precisa.

4.4.3 Mobilenetv2

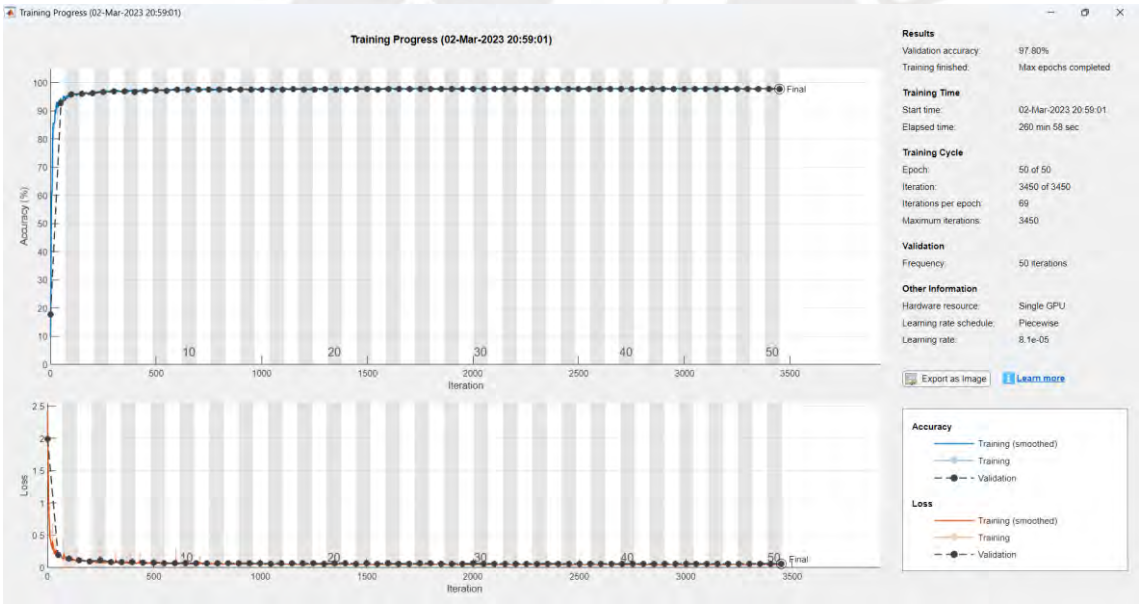

Figura 4.13. Gráfica del proceso de entrenamiento basada en la arquitectura Mobilenetv2. Fuente: Propia

Durante la etapa de entrenamiento, se alcanzó una exactitud del 97.80%, como se muestra en la Figura 4.13. Este valor es notablemente alto al considerar la métrica de exactitud y en base a la literatura leída y citada en el capítulo 2. En relación de la gráfica *Loss vs Iteration* se observa una disminución uniforme y constante, indicando un proceso de

entrenamiento adecuado. Además, se nota que tanto el conjunto de entrenamiento como el de validación exhiben curvas eficientes, sin evidencia de subajuste ni sobreajuste.

Tabla 4.7. Métricas globales de arquitectura basada en Mobilenetv2. Fuente: Propia

GlobalAccuracy	MeanAccuracy	MeanIoU	WeightedIoU	MeanBFScore
0.89995	0.86828	0.73755	0.82542	0.66683

En la Tabla 4.7 se presenta una exactitud global de 0.89 y una exactitud promedio de 0.87 para la etapa de prueba. Esta diferencia en la precisión se debe al hecho de que el conjunto de datos de prueba se seleccionó de manera diferente al conjunto de datos de entrenamiento. Los resultados obtenidos son considerados aceptables según lo establecido en la literatura revisada en el punto 2. En cuanto al IoU promedio, se registra un valor de 0.73, con una IoU ponderada de 0.82. Esto indica que algunas clases tienen una mayor representación en términos de datos o píxeles en el total de las imágenes segmentadas. El BFScore promedio obtenido es de 0.66, lo que indica que los contornos de las segmentaciones se asemejan a la máscara de referencia en esta medida.

Tabla 4.8. Métricas por clase de arquitectura basada en Mobilenetv2. Fuente: Propia

	Accuracy	IoU	MeanBFScore
Road	0.94945	0.84803	0.63862
Sidewalk	0.68006	0.62797	0.53034
Car	0.8149	0.66901	0.58338
Pedestrian	0.80614	0.62984	0.68362
Building	0.92874	0.89758	0.78407
MotorCycle	0.93138	0.54646	0.57203
Others	0.96731	0.94395	0.87578

Según los resultados de la Tabla 4.8, la clase con los mayores valores de exactitud, IoU y mBFScore es "others", que representa al cielo y cables de energía eléctrica. En contraste,

la clase "sidewalk" muestra los valores más bajos en cuanto a exactitud y IoU, mientras que en mBFScore, la clase con el valor más bajo es también "sidewalk".

Para la clase "others", se puede concluir que la segmentación es buena, con una exactitud de 0.96, lo que indica que la mayoría de esta clase se clasifica correctamente. Además, el IoU de 0.94 sugiere que hay poca presencia de otras clases en las segmentaciones de esta clase, y un mBFScore de 0.87 indica que la mayoría de los contornos están bien delimitados.

En cuanto a la clase "building", se esperan resultados similares a los de la clase "others", con valores de 0.93 para la exactitud, 0.89 para el IoU y 0.78 para el mBFScore. Sin embargo, la segmentación semántica puede no ser equitativa en comparación con la clase "others", ya que puede haber más clases ajenas a "building" presentes dentro de la misma.

Para la clase "road", se observa una alta y aceptable exactitud e IoU, pero un mBFScore significativamente más bajo (0.63). Esto se debe a que hay una mayor cantidad de píxeles por imagen para esta clase, lo que influye en el bajo valor del mBFScore.

En el caso de la clase "pedestrian", se tiene una exactitud alta y aceptable, pero valores muchos menores de IoU y mBFScore, lo que sugiere que la cantidad de píxeles para esta clase no es tan representativa como para otras clases.

Finalmente, para la clase "sidewalk", se presentan resultados bajos en comparación con otras clases, lo que indica que la precisión, IoU y BFScore no son óptimos, a pesar de tener una gran cantidad de píxeles representativos por imágenes. Esto se debe a que esta clase varía significativamente con respecto al nuevo conjunto de datos para la etapa de prueba.

A continuación, se analizarán imágenes con los datos de prueba en los diferentes horarios de mañana, mediodía y tarde:

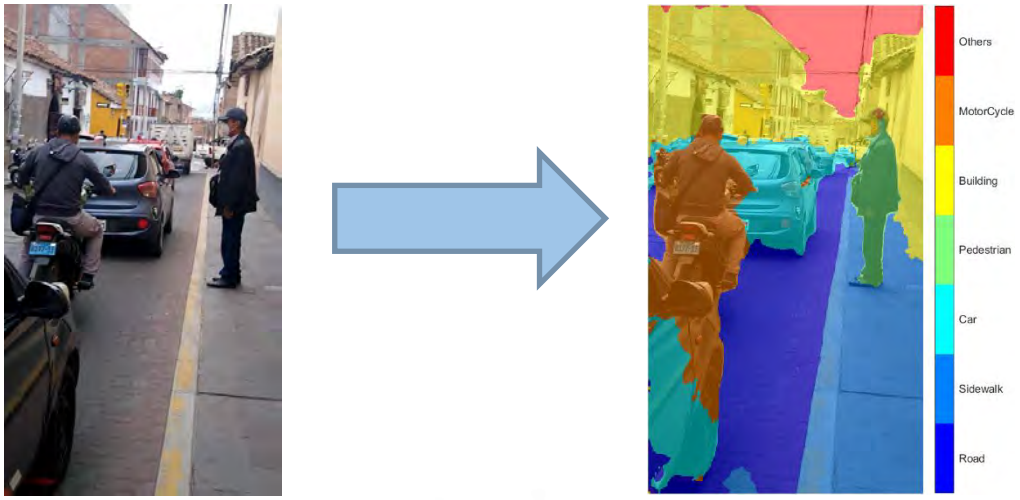

Figura 4.14. Segmentación basada en la arquitectura Mobilenetv2 – Horario de la mañana. Fuente: Propia

Figura 4.15. Segmentación basada en la arquitectura Mobilenetv2 – Horario del mediodía. Fuente: Propia

Figura 4.16. Segmentación basada en la arquitectura Mobilenetv2 – Horario de la tarde. Fuente: Propia

Se puede apreciar en las Figuras 4.14, 4.15 y 4.16 que las clases "others" y "building" presentan una mejor segmentación, como se evidencia por sus altos valores de exactitud, IoU y mBFScore en la etapa de prueba.

En cuanto a la clase "road", también muestra una buena segmentación en términos de exactitud e IoU, aunque se observa alguna superposición con otras clases, lo que afecta al mBFScore.

Respecto a las clases "pedestrian, car y motorcycle", muestran buena exactitud, pero también presentan segmentación en otras clases, lo que se refleja en sus bajos valores de IoU y mBFScore.

Por último, la clase "sidewalk" es la menos precisa, con bajos IoU y mBFScore, además de segmentar incorrectamente en los bordes en comparación con otras clases.

4.4.4 Xception

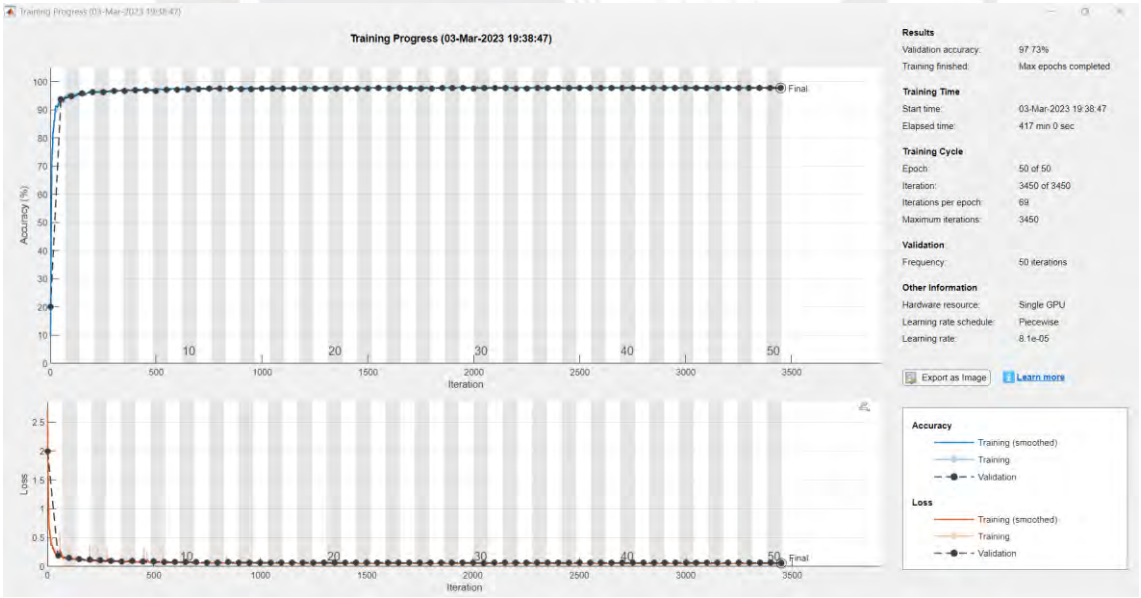

Figura 4.17. Gráfica del proceso de entrenamiento basada en la arquitectura Xception. Fuente: Propia

La exactitud de prueba llega a un valor de 97.73% en la etapa de entrenamiento según lo mostrado en la Figura 4.17 este valor es muy alto cuando se desea medir el parámetro de

exactitud. En cuanto a la pérdida, se muestra una caída uniforme y descendente que conlleva a un correcto proceso de entrenamiento. Se aprecia también que entre el entrenamiento y validación se presenta una curva eficiente que no presenta subajuste ni sobreajuste.

Tabla 4.9. Métricas globales de arquitectura basada en Xception. Fuente: Propia

GlobalAccuracy	MeanAccuracy	MeanIoU	WeightedIoU	MeanBFScore
0.88606	0.8761	0.74908	0.8074	0.66694

En la Tabla 4.9 se presenta una exactitud global de 0.88 y una exactitud promedio de 0.87 para la etapa de prueba. Esta diferencia en la precisión se debe al hecho de que el conjunto de datos de prueba se seleccionó de manera diferente al conjunto de datos de entrenamiento. Los resultados obtenidos son considerados aceptables según lo establecido en la literatura revisada en el punto 2. En cuanto al IoU promedio, se registra un valor de 0.74, con una IoU ponderada de 0.80. Esto indica que algunas clases tienen una mayor representación en términos de datos o píxeles en el total de las imágenes segmentadas. El BFScore promedio obtenido es de 0.66, lo que indica que los contornos de las segmentaciones se asemejan a la máscara de referencia en esta medida.

Tabla 4.10. Métricas por clase de arquitectura basada en Xception. Fuente: Propia

	Accuracy	IoU	MeanBFScore
Road	0.87113	0.77485	0.59176
Sidewalk	0.67964	0.52413	0.46111
Car	0.86292	0.75683	0.65716
Pedestrian	0.90722	0.66369	0.61119
Building	0.94125	0.91603	0.82794
MotorCycle	0.88754	0.64283	0.57268
Others	0.98299	0.96519	0.94233

Según los resultados mostrados en la Tabla 4.10, la clase con mayores valores (exactitud, IoU, mBFScore) sería "others", que representa al cielo y cables de energía eléctrica. En cuanto a la exactitud, la clase "sidewalk" presenta el menor valor. En cuanto al IoU, nuevamente la clase "sidewalk" tiene el menor valor, y finalmente, en mBFScore se obtiene el menor valor para la clase "sidewalk".

Se concluye que al analizar la clase "others", se observa una buena segmentación, con un valor de 0.98 en la exactitud, lo que indica que gran parte de esta clase se clasifica correctamente. Además, se obtiene un IoU de 0.96, lo que sugiere una presencia mínima de otras clases no pertenecientes a "others", y un mBFScore de 0.94, indicando que la mayoría de los contornos están correctamente delimitados.

Para la clase "building", se esperan resultados similares a los de la clase "others", ya que presenta valores de 0.94 para la exactitud, 0.92 para el IoU y 0.83 para el mBFScore. Sin embargo, la segmentación semántica de "building" no es tan equitativa como en "others", lo que se refleja en los valores de IoU y mBFScore.

En cuanto a la clase "road", se observa una buena exactitud e IoU, pero un mBFScore más bajo debido a la mayor cantidad de píxeles por imagen, lo que afecta la delimitación de los contornos.

Para la clase "pedestrian", se tiene una alta exactitud, pero IoU y mBFScore más bajos debido a la cantidad de píxeles no representativos.

Finalmente, para la clase "sidewalk", se presentan resultados más bajos en comparación con las otras clases, lo que indica una menor precisión debido a la variabilidad en los nuevos datos de la etapa de prueba.

A continuación, se analizarán imágenes con los datos de prueba en los diferentes horarios de mañana, mediodía y tarde:

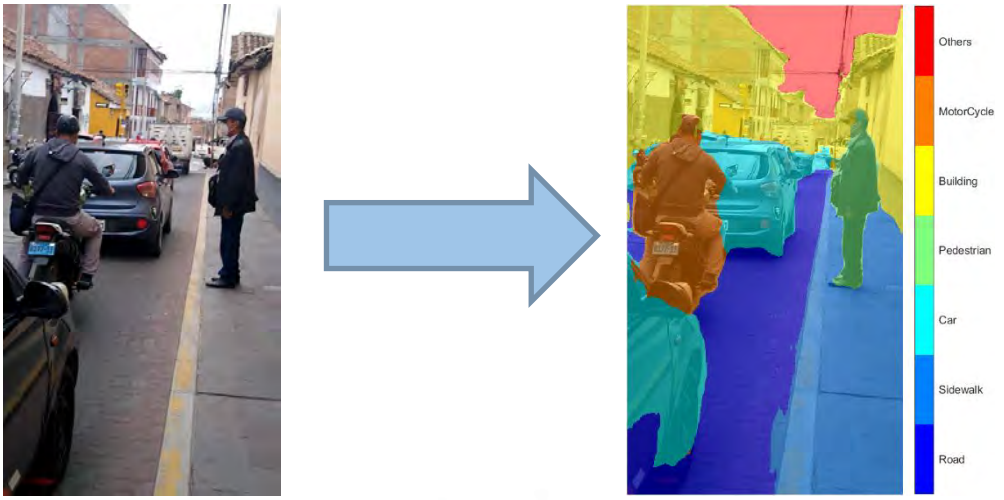

Figura 4.18. Segmentación basada en la arquitectura Xception – Horario de la mañana. Fuente: Propia

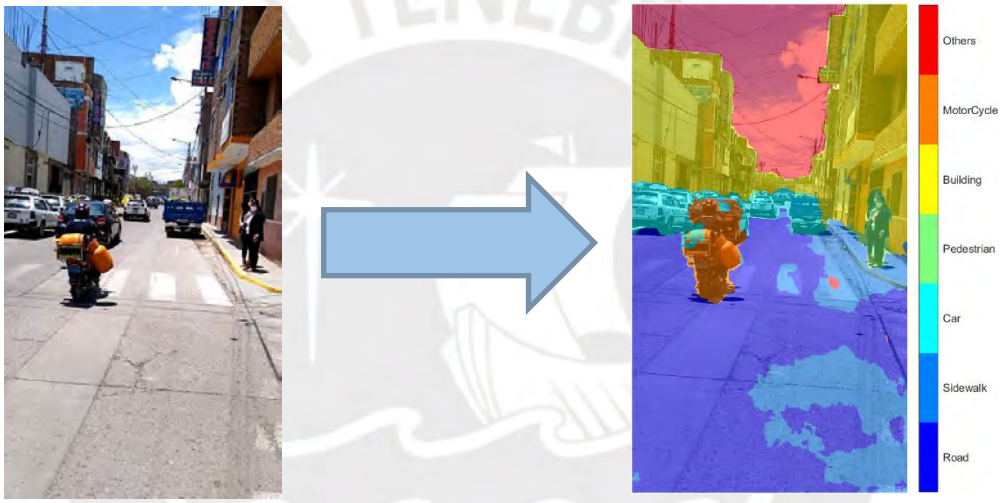

Figura 4.19. Segmentación basada en la arquitectura Xception – Horario del mediodía. Fuente: Propia

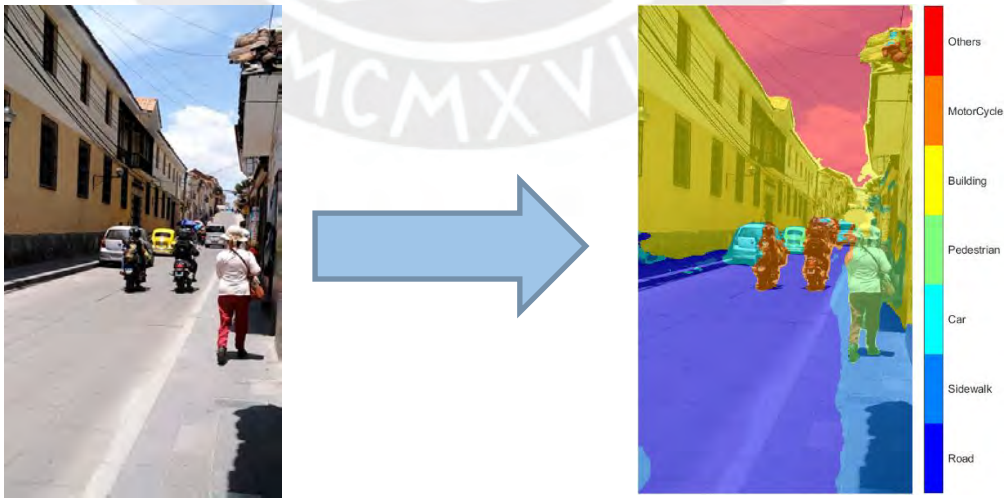

Figura 4.20. Segmentación basada en la arquitectura Xception – Horario de la tarde. Fuente: Propia

Se observa en las Figuras 4.18, 4.19 y 4.20 que las clases "others" y "building" presentan una mejor segmentación, lo cual se evidencia por sus valores superiores de exactitud, IoU y mBFScore en la etapa de prueba.

En cuanto a la clase "road", se presenta una buena segmentación gracias a sus valores aceptables de exactitud e IoU. Sin embargo, también se observa que sobre esta misma clase se segmentan otras clases y viceversa, lo que resulta en un bajo valor de mBFScore debido a la superposición en los bordes.

Para las clases "pedestrian", "car" y "motorcycle", se logra apreciar una buena exactitud. Sin embargo, también se identifica la presencia de segmentación de otras clases en estas regiones, lo que se refleja en los bajos valores de IoU y mBFScore.

Finalmente, la clase "sidewalk" muestra la menor precisión, con bajos valores de IoU y mBFScore, especialmente notables en el caso 1 debido a su baja exactitud en comparación con su propia clase. Además, se evidencia una segmentación incorrecta en los bordes en comparación con las otras clases.

4.5 Comparativa con base de datos Cityscapes

Con respecto al análisis comparativo de las arquitecturas de segmentación semántica basadas en redes de aprendizaje profundo pre-entrenadas, se llevará a cabo la evaluación y comparación bajo redes entrenadas utilizando datos locales de la provincia de Huamanga en contraste con el conjunto de datos Cityscapes. Esto con referencia a la necesidad de demostrar que los resultados específicos de una determinada área local son fundamentales para comprender y mejorar la efectividad de los modelos de segmentación semántica en entornos urbanos. A continuación, se presentan los siguientes resultados:

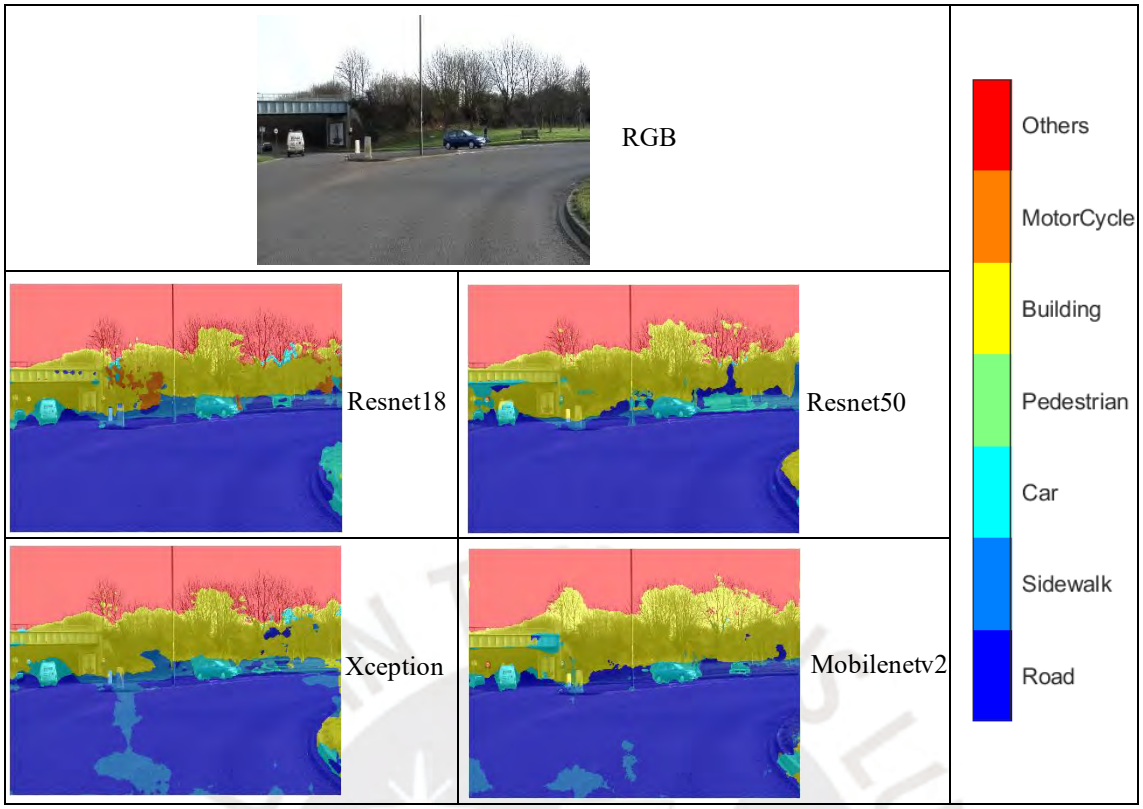

Figura 4.21. Pruebas en Cityscapes dataset con arquitecturas de segmentación semántica en horario de la mañana. Fuente: Propia

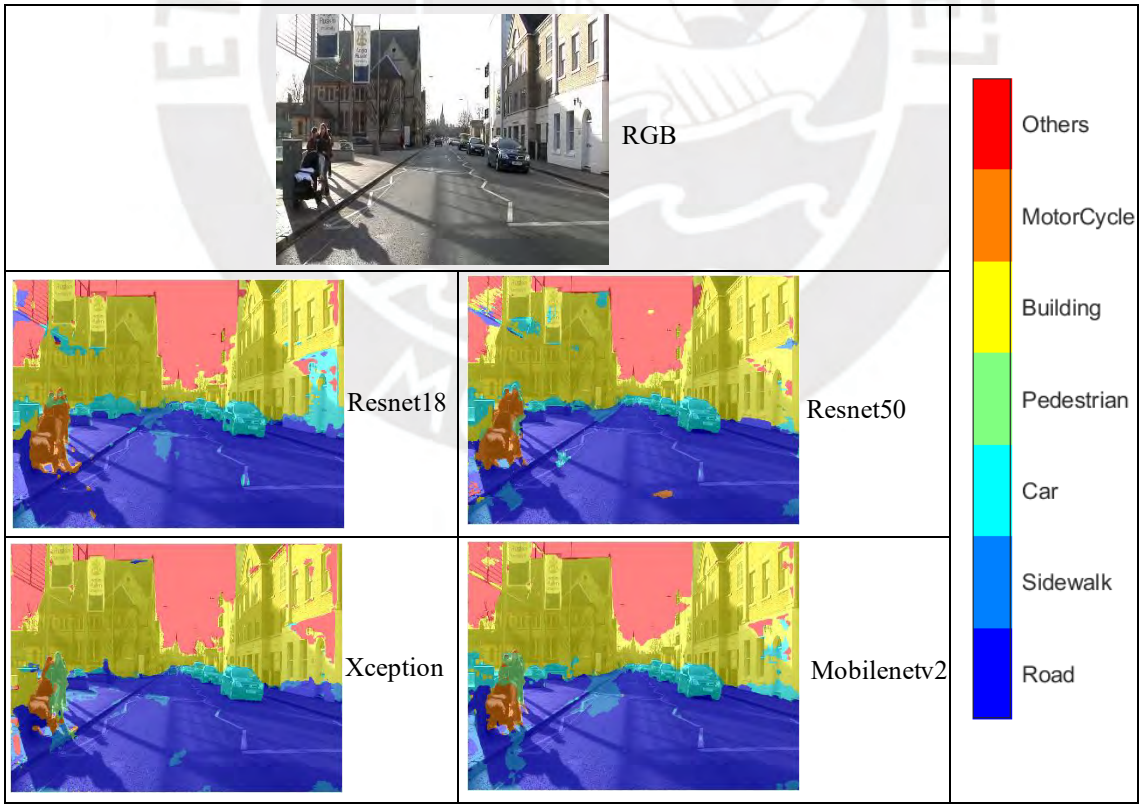

Figura 4.22. Pruebas en Cityscapes dataset con arquitecturas de segmentación semántica en horario del mediodía. Fuente: Propia

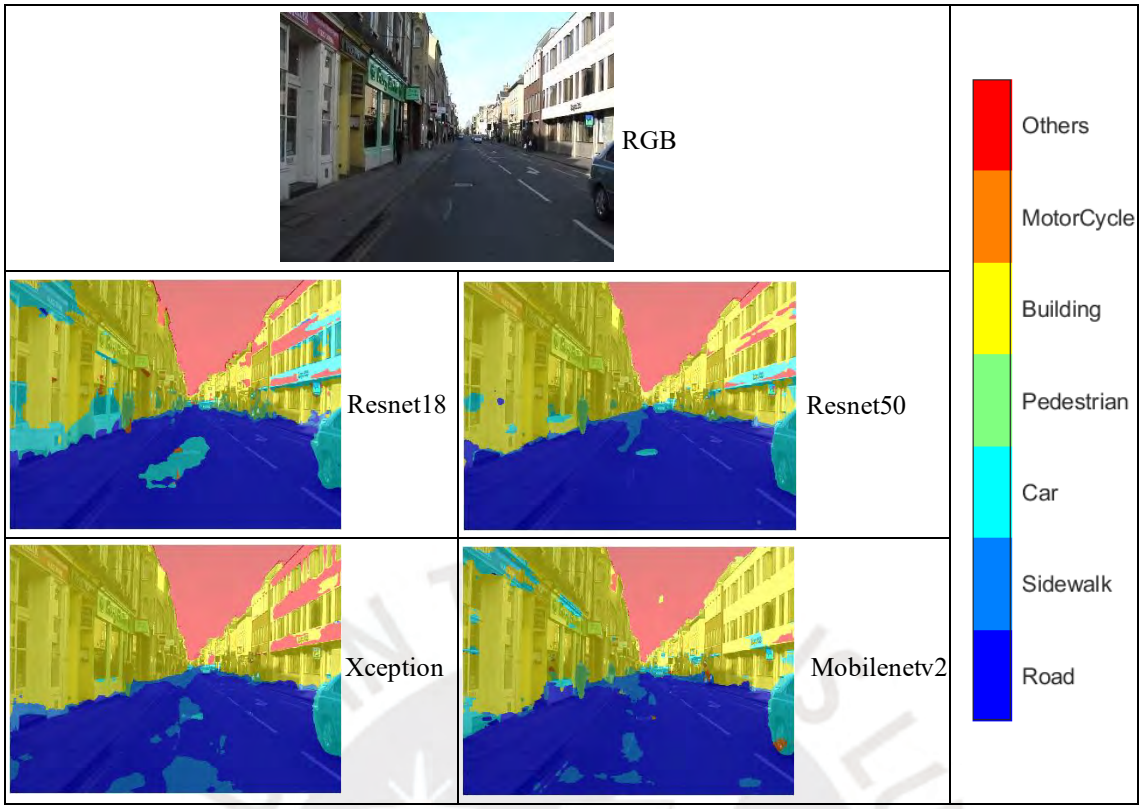

Figura 4.23. Pruebas en Cityscapes dataset con arquitecturas de segmentación semántica en horario de la tarde. Fuente: Propia.

Como se puede apreciar en las Figuras 4.21, 4.22 y 4.23, las arquitecturas de segmentación semántica basadas en Deeplabv3+ entrenadas con datos locales de Huamanga no se ajustan completamente a entornos urbanos más desarrollados. La disparidad entre las infraestructuras y otros elementos puede variar, lo que resalta la necesidad de crear una base de datos específica para ciudades en desarrollo, ya que las redes y datos existentes pueden mostrar poca adaptación en estos entornos.

4.6 Tablas sumarias según estudio y resultados.

Tabla 4.11. Tabla comparativa general con Deeplabv3+ utilizando redes de aprendizaje profundo en la etapa de prueba. Fuente: Propia

Métricas / Red	Global Accuracy	Mean Accuracy	Mean IoU	Weighted IoU	Mean BFScore
Resnet18	0.91941	0.90504	0.79030	0.85557	0.72404
Resnet50	0.92969	0.90577	0.79689	0.87345	0.74382
Mobilenetv2	0.89995	0.86828	0.73755	0.82542	0.66683
Xception	0.88606	0.87610	0.74908	0.80740	0.66694

En la Tabla 4.11 se presenta una comparación del desempeño de cuatro arquitecturas de redes neuronales en una tarea específica. En general, se observa que ResNet50 tiene el mayor Global Accuracy y Mean IoU, seguido de cerca por ResNet18. Esto sugiere que ResNet50 y ResNet18 son las arquitecturas más efectivas en términos de precisión global y capacidad para capturar la superposición de predicciones y verdades de segmentación. Además, ResNet50 tiene el mayor Mean BFScore, lo que indica un mejor equilibrio entre precisión y exhaustividad en la segmentación de objetos. Por otro lado, MobileNetV2 y Xception muestran valores más bajos en todas las métricas en comparación con ResNet18 y ResNet50, lo que sugiere un desempeño inferior en esta tarea específica.

El rendimiento superior de ResNet50 y ResNet18 en la tarea de segmentación semántica mediante transferencia de aprendizaje puede deberse a su capacidad para capturar representaciones profundas y discriminativas de los datos, así como a su diseño específico como arquitecturas para abordar problemas en redes profundas. Además, la efectividad de la transferencia de aprendizaje y la naturaleza de la tarea y del conjunto de datos utilizados también contribuyen a su desempeño superior. Estos factores hacen que las arquitecturas ResNet50 y ResNet18 sean más adecuadas para la tarea en comparación con MobileNetV2 y Xception. Estos resultados resaltan la importancia de elegir la arquitectura adecuada de la red neuronal para maximizar el desempeño en tareas de segmentación de imágenes.

Tabla 4.12. Tabla comparativa de exactitud (Accuracy) por clase con Deeplabv3+ utilizando redes de aprendizaje profundo. Fuente: Propia.

Red / Clase	Resnet18	Resnet50	Mobilenetv2	Xception
Road	0.95145	0.96013	0.94945	0.87113
Sidewalk	0.73029	0.75018	0.68006	0.67964
Car	0.87002	0.90348	0.81490	0.86292
Pedestrian	0.95254	0.85388	0.80614	0.90722
Building	0.94023	0.96039	0.92874	0.94125
MotorCycle	0.90902	0.93864	0.93138	0.88754
Others	0.98170	0.97370	0.96731	0.98299

Los resultados observados en la Tabla 4.12 revelan que, en su mayoría, la arquitectura basada en ResNet50 exhibe la mayor precisión para las diversas clases, destacando su capacidad superior para la clasificación precisa en la mayoría de las categorías. Sin embargo, para las clases "pedestrian" y "others", se observa un mejor desempeño en las arquitecturas basadas en ResNet18 y Xception, respectivamente. Este fenómeno puede atribuirse a la capacidad de ResNet18 para capturar características más finas y detalladas, especialmente relevantes para la clase "pedestrian", debido a su menor profundidad en comparación con ResNet50. Por otro lado, la arquitectura Xception, con su estructura de bloques de convolución separables en profundidad y anchura, puede ser más efectiva para representar características complejas presentes en la clase "others". Aunque ResNet50 generalmente lidera en términos de precisión, ResNet18 sigue siendo una opción competitiva, colocándose en segundo lugar en muchos casos. Por último, las arquitecturas como Xception y MobileNetV2, muestran valores más bajos de precisión en promedio y para la mayoría de las clases, posiblemente debido a su capacidad limitada para capturar características semánticas complejas en comparación con ResNet18 y ResNet50. Estos resultados resaltan la importancia de considerar las características específicas de las clases y las capacidades de las diferentes arquitecturas al seleccionar un modelo para tareas de clasificación de imágenes.

Tabla 4.13. Tabla comparativa de IoU (Intersection over Union) por clase con Deeplabv3+ utilizando redes de aprendizaje profundo. Fuente: propia.

Red / Clase	Resnet18	Resnet50	Mobilenetv2	Xception
Road	0.86273	0.88833	0.84803	0.77485
Sidewalk	0.67458	0.70958	0.62797	0.52413
Car	0.75602	0.79500	0.66901	0.75683
Pedestrian	0.71769	0.62462	0.62984	0.66369
Building	0.91917	0.93605	0.89758	0.91603
MotorCycle	0.63737	0.66202	0.54646	0.64283
Others	0.96456	0.96264	0.94395	0.96519

En la Tabla 4.13, ResNet50 y ResNet18 muestran un desempeño sólido en términos de IoU para la mayoría de las clases. ResNet50 tiende a superar ligeramente a ResNet18 en la mayoría de las clases, sugiriendo que su mayor profundidad y complejidad pueden permitir una captura más efectiva de las características semánticas de las diferentes clases en comparación con ResNet18. Sin embargo, para algunas clases como "Pedestrian", ResNet18 muestra un mejor desempeño en términos de IoU, lo que indica que la profundidad adicional de ResNet50 no siempre se traduce en una mejora significativa en la precisión de la segmentación para todas las clases.

Por otro lado, MobileNetV2 y Xception, si bien muestran valores de IoU más bajos en general en comparación con ResNet50 y ResNet18, aún logran resultados significativos en algunas clases. Por ejemplo, MobileNetV2 presenta un desempeño competitivo para la clase "Sidewalk" en comparación con las otras arquitecturas. Sin embargo, en general, MobileNetV2 y Xception tienden a mostrar valores de IoU más bajos, lo que sugiere que su diseño más liviano puede limitar su capacidad para capturar características semánticas complejas y detalladas en comparación con ResNet50 y ResNet18.

Es importante destacar que los valores de IoU varían entre clases, lo que indica que algunas clases son más difíciles de segmentar que otras. Por ejemplo, "Sidewalk" y "MotorCycle" muestran valores de IoU más bajos en general, lo que puede deberse a la variabilidad en las características visuales de estas clases o a la presencia de desafíos específicos en la segmentación de objetos en esas clases.

La variación en los valores de IoU entre las diferentes arquitecturas de redes neuronales puede atribuirse a varias razones. En primer lugar, las arquitecturas más profundas y complejas, como ResNet50 y ResNet18, tienen una mayor capacidad para capturar características detalladas y complejas en las imágenes, lo que resulta en una mejor coincidencia entre las máscaras predichas y las máscaras verdaderas, y por lo tanto, en

valores de IoU más altos. Además, algunas arquitecturas pueden estar diseñadas específicamente para tareas de segmentación semántica, lo que les permite capturar características relevantes de manera más efectiva en comparación con arquitecturas diseñadas para otras tareas, como la clasificación de imágenes. Por último, el desempeño de una arquitectura puede variar dependiendo del conjunto de datos específico utilizado para el entrenamiento y la evaluación, ya que algunas clases pueden ser más difíciles de segmentar debido a la variabilidad en sus características visuales, lo que puede afectar los valores de IoU para esas clases en particular.

Tabla 4.14. Cuadro comparativo de puntuación media de la función BFS (meanBFScore) por clase con Deeplabv3+ utilizando redes de aprendizaje profundo. Fuente: Propia.

Red / Clase	Resnet18	Resnet50	Mobilenetv2	Xception
Road	0.70295	0.70064	0.63862	0.59176
Sidewalk	0.63909	0.62915	0.53034	0.46111
Car	0.69601	0.69310	0.58338	0.65716
Pedestrian	0.61851	0.71022	0.68362	0.61119
Building	0.84310	0.86732	0.78407	0.82794
MotorCycle	0.61718	0.67095	0.57203	0.57268
Others	0.94114	0.93352	0.87578	0.94233

En la Tabla 4.14, ResNet18 y ResNet50 muestran los valores más altos de Mean BFScore en general para la mayoría de las clases. Esto sugiere que estas arquitecturas tienen una capacidad sólida para segmentar objetos con precisión y exhaustividad, lo que resulta en un Mean BFScore más alto. Estas arquitecturas pueden capturar características semánticas detalladas y complejas en las imágenes, lo que les permite realizar una segmentación precisa de una amplia variedad de clases.

Por otro lado, MobileNetV2 y Xception muestran valores más bajos de Mean BFScore en comparación con ResNet18 y ResNet50. Esto podría indicar que estas arquitecturas son menos precisas en la segmentación semántica y pueden perder detalles importantes en la segmentación de objetos. Sin embargo, MobileNetV2 y Xception aún logran

resultados competitivos en algunas clases específicas, lo que sugiere que pueden ser adecuadas para ciertos escenarios de aplicación donde la precisión no es crítica.

Es importante tener en cuenta que el desempeño de las arquitecturas puede variar según las características específicas del conjunto de datos y las clases presentes en él. Por lo tanto, es crucial evaluar el desempeño de diferentes arquitecturas en una variedad de conjuntos de datos y escenarios de aplicación para comprender completamente sus fortalezas y debilidades en términos de segmentación semántica.

La variación en los valores de Mean BFScore entre las diferentes arquitecturas de redes neuronales se debe a diversos factores. En primer lugar, la complejidad y la capacidad de la red juegan un papel crucial. Las arquitecturas más profundas y complejas, como ResNet50 y ResNet18, tienen la capacidad de capturar características más detalladas y complejas en las imágenes, lo que resulta en una segmentación más precisa y completa de los objetos. Esto se traduce en valores más altos de Mean BFScore para estas arquitecturas en comparación con otras más simples.

Además, el diseño específico para la tarea es otro factor importante. Algunas arquitecturas pueden estar diseñadas específicamente para tareas de segmentación semántica, lo que les permite capturar características relevantes de manera más efectiva en comparación con aquellas diseñadas para otras tareas, como la clasificación de imágenes.

Por último, la variabilidad en las clases y características del conjunto de datos utilizado también contribuye a la variación en los valores de Mean BFScore. El desempeño de una arquitectura puede verse afectado por la complejidad y la variabilidad de las clases presentes en el conjunto de datos. Algunas clases pueden ser más fáciles de segmentar debido a características visuales distintivas, mientras que otras clases pueden presentar desafíos adicionales en la segmentación.

Tabla 4.15. Cuadro comparativo de tamaño de red de arquitectura de Deeplabv3+ basada en redes de aprendizaje profundo. Fuente: Propia.

Red / Medida	Resnet18	Resnet50	Mobilenetv2	Xception
Tamaño (MB)	58.20	141.30	9.45	83.40

En la Tabla 4.15, ResNet18 se destaca por su tamaño compacto de 58.20 MB, lo que la convierte en una opción atractiva para aplicaciones con restricciones de almacenamiento o recursos computacionales limitados. Aunque más pequeña en tamaño, ResNet18 aún ofrece una capacidad razonable de extracción de características para la segmentación semántica, lo que la hace adecuada para escenarios donde la eficiencia de almacenamiento es prioritaria.

En contraste, ResNet50 presenta un tamaño mucho mayor de 141.30 MB. Esta arquitectura más profunda y compleja puede capturar características más detalladas en las imágenes, lo que potencialmente mejora la precisión de la segmentación semántica. Sin embargo, su mayor tamaño implica una mayor demanda de recursos computacionales y almacenamiento, lo que puede limitar su aplicabilidad en dispositivos con restricciones de recursos.

MobileNetV2 destaca por su tamaño extremadamente compacto de solo 9.45 MB. Diseñada específicamente para dispositivos móviles o entornos con recursos limitados, esta arquitectura ofrece una excelente eficiencia en términos de almacenamiento y recursos computacionales. MobileNetV2 sigue siendo capaz de proporcionar una segmentación semántica adecuada, lo que la hace especialmente adecuada para aplicaciones en dispositivos con recursos limitados.

Por último, Xception presenta un tamaño de 83.40 MB, posicionándose en un punto intermedio entre ResNet18 y ResNet50 en términos de tamaño de red. Su diseño basado en convoluciones separables en profundidad y anchura puede influir en su tamaño en

comparación con otras arquitecturas. Xception ofrece un equilibrio entre capacidad de extracción de características y eficiencia en términos de tamaño de red, lo que la hace adecuada para una variedad de aplicaciones que requieren un compromiso entre precisión y eficiencia de almacenamiento.

Tabla 4.16. Cuadro comparativo de tiempo de respuesta de arquitectura de Deeplabv3+ basada en redes de aprendizaje profundo en CPU Intel Core i5 11ª generación. Fuente: Propia.

Red / Medida	Resnet18	Resnet50	Mobilenetv2	Xception
Tiempo (ms)	32.54	54.68	26.90	42.51
Cuadros por segundo (fps)	31	18	37	24

En la Tabla 4.16, ResNet18 exhibe un tiempo de respuesta de 32.54 ms, lo que le permite procesar cada cuadro a una velocidad aproximada de 31 cuadros por segundo (fps). Aunque no es la arquitectura más rápida, ResNet18 ofrece un equilibrio adecuado entre tiempo de respuesta y precisión de la segmentación semántica, lo que la hace adecuada para aplicaciones que requieren una velocidad razonable de procesamiento.

ResNet50, con un tiempo de respuesta de 54.68 ms, demuestra un rendimiento ligeramente más lento en comparación con ResNet18, procesando aproximadamente 18 fps. Aunque ofrece una precisión potencialmente mejorada debido a su mayor profundidad, la penalización en términos de tiempo de respuesta puede limitar su aplicabilidad en escenarios que requieren una alta velocidad de procesamiento.

MobileNetV2 destaca por su tiempo de respuesta rápido de 26.90 ms, lo que le permite procesar cada cuadro a una velocidad de aproximadamente 37 fps. Esta arquitectura ofrece una excelente eficiencia en términos de velocidad de procesamiento, lo que la hace especialmente adecuada para aplicaciones en tiempo real o en dispositivos con recursos limitados.

Xception muestra un tiempo de respuesta de 42.51 ms, lo que se traduce en un rendimiento de aproximadamente 24 fps. Aunque no es tan rápido como MobileNetV2, Xception ofrece un equilibrio entre velocidad de procesamiento y precisión de la segmentación semántica, lo que la hace adecuada para una variedad de aplicaciones que requieren un compromiso entre velocidad y precisión.

Finalmente, se puede concluir que la elección de la arquitectura de aprendizaje profundo para la transferencia de aprendizaje dentro de DeepLabV3+ debe considerar no solo la precisión de la segmentación semántica, sino también el tiempo de respuesta y los requisitos de velocidad de procesamiento en la aplicación específica. Cada arquitectura tiene sus propias ventajas y consideraciones en términos de velocidad y precisión, lo que permite una selección adaptada a las necesidades específicas de la aplicación. Es fundamental evaluar cuidadosamente estas características y tomar decisiones informadas para garantizar un rendimiento óptimo en la tarea de segmentación de imágenes.

4.7 Propuesta y análisis de hardware

Matlab ofrece soporte para múltiples dispositivos utilizados en aplicaciones de visión por computadora, incluyendo Raspberry Pi, BeagleBone Black, Arduino, FPGA y tarjetas NVIDIA [44]. La elección de un dispositivo para estas aplicaciones depende de varios factores clave, como el rendimiento, la capacidad de procesamiento, el tiempo de respuesta, portabilidad, eficiencia energética y costo [29], [30], [31], [33], [34], [36], [39], [40], [41], [42], [43].

En el caso de Raspberry Pi, este dispositivo destaca por su reducido tamaño y bajo consumo energético, siendo una opción adecuada para proyectos que requieren dispositivos portátiles y de bajo costo [29]. Sin embargo, debido a sus limitaciones de rendimiento y memoria, puede no ser la mejor opción para aplicaciones que necesitan un

procesamiento intensivo o una gran capacidad de almacenamiento [30], [31], [35], [37], [38].

Las tarjetas NVIDIA, por otro lado, han sido fundamentales en el avance de la visión por computadora gracias a su potente capacidad de procesamiento gráfico y cómputo paralelo [32], [33], [34], [36]. Las tarjetas NVIDIA proporcionan una plataforma robusta para abordar desafíos complejos en el campo de la visión por computadora en aplicaciones tales como la clasificación y/o detección de objetos en escenas urbanas, robótica, seguimiento de objetos, entre otros [31], [33], [35]. Además, su versatilidad, que abarca desde modelos más simples hasta configuraciones avanzadas, las convierte en una opción adecuada para una amplia gama de aplicaciones en este ámbito [34], [35], [36].

Los FPGAs destacan principalmente por su eficiencia energética y su capacidad para paralelizar tareas [39], [40], [41], [42], [43]. Esto los hace particularmente atractivos para aplicaciones que requieren un procesamiento rápido y eficiente en tiempo real, como es el caso de la visión por computadora. Además, su arquitectura reconfigurable y variedad de modelos permite adaptar el hardware según las necesidades específicas del proyecto, lo que los hace altamente versátiles y adecuados para una amplia gama de aplicaciones [39], [40], [41], [42], [43].

Si bien los FPGAs pueden ofrecer ventajas en estos aspectos, es importante tener en cuenta que su programación y configuración pueden requerir conocimientos especializados y tiempo de desarrollo [41]. En contraste, las tarjetas NVIDIA presentan herramientas y bibliotecas específicas para aplicaciones de visión por computadora, lo que las hace más accesibles para estas tareas en comparación con los FPGAs.

Si se analiza la Tabla 4.15, se observa que la red Mobilenetv2 es significativamente más pequeña en comparación con las otras redes, con un tamaño de solo 9.45 MB. Esto la

hace especialmente adecuada para dispositivos con limitaciones de almacenamiento, como dispositivos portátiles o embebidos. Por otro lado, las redes Resnet50 y Xception son más grandes en tamaño, con 141.30 MB y 83.40 MB respectivamente, lo que podría requerir más recursos de almacenamiento y procesamiento.

En cuanto a la Tabla 4.16, se destaca que la red Mobilenetv2 tiene el tiempo de respuesta más rápido de todas las arquitecturas evaluadas, con un tiempo de 26.90 ms. Esto la convierte en una opción atractiva para aplicaciones que requieren un procesamiento rápido en tiempo real, como sistemas de detección de objetos en tiempo real en entornos urbanos. Sin embargo, es importante tener en cuenta que este menor tiempo de respuesta puede estar asociado con una menor precisión en comparación con las otras redes.

Considerando también la red Resnet18, observamos que su tamaño es intermedio entre Mobilenetv2 y las otras redes, con 58.20 MB. En cuanto al tiempo de respuesta, se encuentra entre los valores de Mobilenetv2 y Resnet50/Xception, con 32.54 ms. Esto sugiere que Resnet18 podría ser una opción equilibrada en términos de tamaño de red y tiempo de respuesta para aplicaciones que requieren un compromiso entre precisión y eficiencia computacional en dispositivos con recursos limitados, como dispositivos portátiles o embebidos en entornos urbanos.

Por otro lado, las redes Resnet50 y Xception muestran tiempos de respuesta de 54.68 ms y 42.51 ms respectivamente. Aunque son más lentas que Mobilenetv2, ofrecen una mayor precisión y pueden ser más adecuadas para aplicaciones donde la precisión es prioritaria sobre el tiempo de respuesta, como el reconocimiento de objetos en escenas urbanas complejas.

En conclusión, la elección de la arquitectura de la red de Deeplabv3+ y la selección entre FPGAs, tarjetas NVIDIA y Raspberry Pi para aplicaciones de visión por computadora

dependerán de las necesidades específicas de cada proyecto. Si se prioriza el tiempo de respuesta y se pueden aceptar ciertas compensaciones en precisión, Mobilenetv2 podría ser la opción preferida. En caso de Resnet18, también sería una elección adecuada, especialmente si se valora la precisión y se pueden aceptar tiempos de respuesta ligeramente más largos en comparación con Mobilenetv2. Por lo tanto, para proyectos que requieran un equilibrio entre precisión y tiempo de respuesta, Resnet18 podría ser una opción preferida. Mientras que, si se necesita una mayor precisión y se pueden tolerar tiempos de respuesta más largos, las redes Resnet50 y Xception podrían ser más adecuadas.

Considerando factores como rendimiento, costo y eficiencia energética, cada plataforma tiene sus propias ventajas y desventajas. La elección adecuada dependerá de encontrar el equilibrio óptimo entre estos factores para la aplicación específica en cuestión.

CONCLUSIONES

- El primer objetivo, centrado en la creación de un conjunto de datos específico para la provincia de Huamanga, ha producido resultados sólidos durante el proceso de entrenamiento y evaluación. Estos resultados respaldan la eficacia de la base de datos local, mientras que las métricas favorables obtenidas confirman la calidad de los datos recopilados. Además, la consistencia en el rendimiento del modelo entrenado con estos datos subraya su robustez en el contexto de la provincia de Huamanga.
- El segundo y tercer objetivo, centrados en implementar, entrenar y evaluar la arquitectura Deeplabv3+ utilizando redes pre-entrenadas de aprendizaje profundo con el conjunto de datos en el programa Matlab, han demostrado resultados óptimos y comparables entre sí con respecto a las cuatro redes implementadas a través de la transferencia de aprendizaje. Entre estas redes, Resnet50 y Resnet18 destacaron al obtener mejores resultados globales y en la mayoría de sus clases en comparación con Xception y Mobilenetv2.
- Respecto al cuarto y último objetivo, que implicaba comparar el rendimiento de las redes de aprendizaje profundo pre-entrenadas aplicables a la arquitectura Deeplabv3+ utilizando el nuevo conjunto de datos, y llevar a cabo estudios para la implementación de hardware aplicables a la segmentación semántica, se llega a las siguientes conclusiones:
    - Mobilenetv2 destacó por su menor tamaño y tiempo de respuesta rápido en comparación con otras redes evaluadas en este estudio. Esta característica la hace una opción atractiva para propuestas de hardware con tiempos de respuesta ágiles. Sin embargo, es importante considerar

que esta ventaja puede estar acompañada de compromisos en la calidad de la segmentación en comparación con otras redes.

- Resnet50 sobresale por su excelente rendimiento en la segmentación en comparación con otras redes, lo que la convierte en una opción preferida para obtener mejores resultados. No obstante, su mayor tiempo de respuesta y tamaño sugieren que se necesitará un hardware más robusto en comparación con otras redes.
- Resnet18 y Xception se sitúan en un punto intermedio en términos de tamaño y tiempo de respuesta. Sin embargo, Resnet18 supera a Xception en rendimiento y tiempo de respuesta, mostrando resultados comparables a Resnet50. Esto la convierte en la opción ideal en cuanto a equilibrio entre rendimiento y tiempo de respuesta.

RECOMENDACIONES Y TRABAJO FUTURO

- El tratamiento del conjunto de datos es crucial para el éxito del entrenamiento de la red neuronal, ya que la selección adecuada de imágenes contribuye significativamente a un buen rendimiento del modelo. La variabilidad en los datos de entrenamiento conduce a resultados más sólidos y robustos.
- El tamaño del conjunto de datos debe ajustarse al entorno específico bajo estudio, y se recomienda aumentar gradualmente el conjunto de datos y sus variaciones para mejorar continuamente los resultados del modelo.
- Como trabajo futuro, se podría considerar la implementación de este estudio en hardware para evaluar su eficiencia y rendimiento en un entorno de tiempo real.

BIBLIOGRAFÍA

[1] K. K. Eerapu, S. Lal and A. V. Narasimhadhan, "O-SegNet: Robust Encoder and Decoder Architecture for Objects Segmentation From Aerial Imagery Data," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 6, no. 3, pp. 556-567, June 2022, doi: 10.1109/TETCI.2020.3045485

[2] D. Pan, M. Zhang and B. Zhang, "A Generic FCN-Based Approach for the Road-Network Extraction From VHR Remote Sensing Images – Using OpenStreetMap as Benchmarks," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 2662-2673, 2021, doi: 10.1109/JSTARS.2021.3058347.

[3] P. Viswanath, S. Nagori, M. Mody, M. Mathew and P. Swami, "End to End Learning based Self-Driving using JacintoNet," 2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), Berlin, 2018, pp. 1-4. doi: 10.1109/ICCE-Berlin.2018.8576190.

[4] T. Okuyama, T. Gonsalves and J. Upadhyay, "Autonomous Driving System based on Deep Q Learning," *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*, Singapore, 2018, pp. 201-205. doi: 10.1109/ICoIAS.2018.8494053.

[5] S. Chen, "Multimedia for Autonomous Driving," in *IEEE MultiMedia*, vol. 26, no. 3, pp. 5-8, 1 July-Sept. 2019. doi: 10.1109/MMUL.2019.2935397.

[6] G. Prabhakar, B. Kailath, S. Natarajan and R. Kumar, "Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving," *2017 IEEE Region 10 Symposium (TENSYMP)*, Cochin, 2017, pp. 1-6. doi: 10.1109/TENCONSpring.2017.8069972.

[7] D. Dong, X. Li and X. Sun, "A Vision-Based Method for Improving the Safety of Self-Driving," *2018 12th International Conference on Reliability, Maintainability, and Safety (ICRMS)*, Shanghai, China, 2018, pp. 167-171. doi: 10.1109/ICRMS.2018.00040.

[8] R. Kulkarni, S. Dhavalikar and S. Bangar, "Traffic Light Detection and Recognition for Self-Driving Cars Using Deep Learning," *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, 2018, pp. 1-4. doi: 10.1109/ICCUBEA.2018.8697819.

[9] Z. Unnisa, Z. Akhtar, H. Riaz and T. Zulfiqar, "Obstacle detection for self-driving car in Pakistan's perspective," *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, 2018, pp. 1-8. doi: 0.1109/ICOMET.2018.8346334.

[10] Mathworks, "Computer Vision", 2018. [En línea]. Disponible en: https://es.mathworks.com/products/computer-vision.html. [Accedido: 20-nov-2019]

[11] R. Gonzales y R. Woods, "Digital Image Processing," en *an introduction to the Mathematical Tools Used in Digital Image Processing*, vol 3, New York: Pearson,2008, pp. 93-96

[12] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017. doi: 10.1109/TPAMI.2016.264461.

[13] S. Nam and S. J. Kim, "Modelling the Scene Dependent Imaging in Cameras with a Deep Neural Network," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017, pp. 1726-1734, doi: 10.1109/ICCV.2017.190.

[14] A. R. Rout and S. B. Bagal, "Natural Scene Classification Using Deep Learning," *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, Pune, 2017, pp. 1-5, doi: 10.1109/ICCUBEA.2017.8463727.

[15] T. Chavdarova *et al*., "WILDTRACK: A Multi-camera HD Dataset for Dense Unscripted Pedestrian Detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 5030-5039, doi: 10.1109/CVPR.2018.00528.

[16] T. Chavdarova and F. Fleuret, "Deep Multi-camera People Detection," *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Cancun, 2017, pp. 848-853, doi: 10.1109/ICMLA.2017.00-50.

[17] J. Tsotsos, I. Kotseruba, A. Andreopoulos and Y. Wu, "Why Does Data-Driven Beat Theory-Driven Computer Vision?," *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seoul, Korea (South), 2019, pp. 2057-2060, doi: 10.1109/ICCVW.2019.00260.

[18] G. Wang, Z. Wang, Y. Zhao and Y. Zhang, "Tea Bud Recognition Based on Machine Learning," 2022 41st Chinese Control Conference (CCC), Hefei, China, 2022, pp. 6533-6537, doi: 10.23919/CCC55666.2022.9902610.

[19] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Is Faster R-CNN Doing Well for Pedestrian Detection?," in Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, Oct. 2016, pp. 443-457, doi: 10.1007/978-3-319-46484-8_28.

[20] L. -C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834-848, 1 April 2018, doi: 10.1109/TPAMI.2017.2699184.

[21] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation" in Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, Sep. 2018, pp. 801-818, doi: 10.1007/978-3-030-01234-2_49.

[22] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional Random Fields as Recurrent Neural Networks," in Proceedings of the International Conference on Computer Vision (ICCV), Santiago, Chile, Dec. 2015, pp. 1529-1537, doi: 10.1109/ICCV.2015.178.

[23] M. Oquab, L. Bottou, I. Laptev and J. Sivic, "Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 1717-1724, doi: 10.1109/CVPR.2014.222.

[24] M. N. Mahmud et al., "Altitude Analysis of Road Segmentation from UAV Images with DeepLab V3+," 2022 IEEE 12th International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 2022, pp. 219-223, doi: 10.1109/ICCSCE54767.2022.9935649.

[25] J. Li, "A deeplabv3+-based Investigation on Lane Line Detection," 2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS), Chengdu, China, 2023, pp. 645-648, doi: 10.1109/ISCTIS58954.2023.10213161.

[26] A. Aizatin and I. G. B. B. Nugraha, "Comparison of Semantic Segmentation Deep Learning Methods for Building Extraction," 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 2022, pp. 1-5, doi: 10.1109/CENIM56801.2022.10037426.

[27] S. Das, A. A. Fime, N. Siddique and M. M. A. Hashem, "Estimation of Road Boundary for Intelligent Vehicles Based on DeepLabV3+ Architecture," in IEEE Access, vol. 9, pp. 121060-121075, 2021, doi: 10.1109/ACCESS.2021.3107353.

[28] G. Lili and Z. Jinzhi, "A Lightweight Network for Semantic Segmentation of Road Images Based on Improved DeepLabv3+," 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Chengdu, China, 2022, pp. 832-837, doi: 10.1109/PRAI55851.2022.9904092.

[29] R. N. Lazuardi, D. Sudrajat, N. Aulia and T. Adiono, "A System of Semantic Segmentation on An Autonomous Vehicle," 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Pattaya, Thailand, 2019, pp. 786-789, doi: 10.1109/ECTI-CON47248.2019.8955214.

[30] K. Podbucki, J. Suder, T. Marciniak and A. Dabrowski, "Evaluation of Embedded Devices for Real- Time Video Lane Detection," 2022 29th International Conference on Mixed Design of Integrated Circuits and System (MIXDES), Wrocław, Poland, 2022, pp. 187-191, doi: 10.23919/MIXDES55591.2022.9838167.

[31] P. Bonnin, P. Blazevic, E. Pissaloux and R. Al Nachar, "Methodology of Evaluation of Low Cost Electronic Devices : Raspberry PI and Nvidia Jetson Nano for Perception System Implementation in Robotic Applications," 2022 IEEE Information Technologies & Smart Industrial Systems (ITSIS), Paris, France, 2022, pp. 01-06, doi: 10.1109/ITSIS56166.2022.10118365.

[32] M. Ma, F. Zou, F. Xu and J. Song, "RTSNet: Real-Time Semantic Segmentation Network For Outdoor Scenes," 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Suzhou, China, 2019, pp. 659-664, doi: 10.1109/CYBER46603.2019.9066620.

[33] G. Dong, Y. Yan, C. Shen and H. Wang, "Real-Time High-Performance Semantic Image Segmentation of Urban Street Scenes," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 6, pp. 3258-3274, June 2021, doi: 10.1109/TITS.2020.2980426.

[34] X. Yu, H. Xu and L. Weng, "A Cityscape Image Detail Extraction Enhancement Method for Lightweight Semantic Segmentation," 2022 IEEE 16th International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, China, 2022, pp. 129-133, doi: 10.1109/ASID56930.2022.9995858.

[35] S. İlkin, F. K. Gülağız, M. Akçakaya and S. Şahin, "Embedded Visual Object Tracking System Based on CSRT Tracker," 2022 International Conference on Electronics, Information, and Communication (ICEIC), Jeju, Korea, Republic of, 2022, pp. 1-4, doi: 10.1109/ICEIC54506.2022.9748840.

[36] Y. Zuo, J. Yang, Z. Zhu, R. Li, Y. Zhou and Y. Zheng, "Real-Time Semantic Segmentation of Aerial Videos Based on Bilateral Segmentation Network," 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 2021, pp. 2763-2766, doi: 10.1109/IGARSS47720.2021.9554952.

[37] A. Nour, I. Dogaru and R. Dogaru, "Comparative Study of Extreme Learning Machine using Various Computing Platforms," 2019 6th International Symposium on Electrical and Electronics Engineering (ISEEE), Galati, Romania, 2019, pp. 1-4, doi: 10.1109/ISEEE48094.2019.9136110.

[38] M. R. Ibrahim and T. Lyons, "ImageSig: A signature transform for ultra-lightweight image recognition," 2022 IEEE/CVF Conferenasasdace on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 2022, pp. 3648-3658, doi: 10.1109/CVPRW56347.2022.00409.

[39] H. L. Blevec, M. Léonardon, H. Tessier and M. Arzel, "Pipelined Architecture for a Semantic Segmentation Neural Network on FPGA," 2023 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Istanbul, Turkiye, 2023, pp. 1-4, doi: 10.1109/ICECS58634.2023.10382715.

[40] Y. Sada, N. Soga, M. Shimoda, A. Jinguji, S. Sato and H. Nakahara, "Fast Monocular Depth Estimation on an FPGA," 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), New Orleans, LA, USA, 2020, pp. 143-146, doi: 10.1109/IPDPSW50202.2020.00032.

[41] G. Tatar and S. Bayar, "Real-Time Multi-Task ADAS Implementation on Reconfigurable Heterogeneous MPSoC Architecture," in IEEE Access, vol. 11, pp. 80741-80760, 2023, doi: 10.1109/ACCESS.2023.3300379.

[42] J. Peng et al., "Multi-task ADAS system on FPGA," 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Hsinchu, Taiwan, 2019, pp. 171-174, doi: 10.1109/AICAS.2019.8771615.

[43] Y. Sada, M. Shimoda, A. Jinguji and H. Nakahara, "A Dataflow Pipelining Architecture for Tile Segmentation with a Sparse MobileNet on an FPGA," 2019 International Conference on Field-Programmable Technology (ICFPT), Tianjin, China, 2019, pp. 267-270, doi: 10.1109/ICFPT47387.2019.00044.

[44] Mathworks, “Hardware Support Package System Requirements”, 2018. [En línea]. Disponible en: https://la.mathworks.com/hardware-support/system-requirements.html. [Accedido: 15-mar-2023]

[45] O. L. F. de Carvalho et al., "Bounding Box-Free Instance Segmentation Using Semi-Supervised Iterative Learning for Vehicle Detection," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 3403-3420, 2022, doi: 10.1109/JSTARS.2022.3169128.

[46] S. M. Azimi, P. Fischer, M. Körner and P. Reinartz, "Aerial LaneNet: Lane-Marking Semantic Segmentation in Aerial Imagery Using Wavelet-Enhanced Cost-Sensitive Symmetric Fully Convolutional Neural Networks," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 5, pp. 2920-2938, May 2019, doi: 10.1109/TGRS.2018.2878510.

[47] H. Sheng, R. Cong, D. Yang, R. Chen, S. Wang and Z. Cui, "UrbanLF: A Comprehensive Light Field Dataset for Semantic Segmentation of Urban Scenes," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 11, pp. 7880-7893, Nov. 2022, doi: 10.1109/TCSVT.2022.3187664.

[48] G. Ros, L. Sellart, J. Materzynska, D. Vazquez and A. M. Lopez, "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 3234-3243, doi: 10.1109/CVPR.2016.352.

[49] G. Li, S. Jiang, I. Yun, J. Kim and J. Kim, "Depth-Wise Asymmetric Bottleneck With Point-Wise Aggregation Decoder for Real-Time Semantic Segmentation in Urban Scenes," in IEEE Access, vol. 8, pp. 27495-27506, 2020, doi: 10.1109/ACCESS.2020.2971760.

[50] D. Alexander, H. C. Kurniawan, I. Afifudin and H. Heryanto, "Application of Convolutional Neural Network for Semantic Segmentation of Bandung Urban Scenes," 2022 International Conference on Data and Software Engineering (ICoDSE), Denpasar, Indonesia, 2022, pp. 12-17, doi: 10.1109/ICoDSE56892.2022.9972006.

[51] M. Volpi and V. Ferrari, "Structured prediction for urban scene semantic segmentation with geographic context," 2015 Joint Urban Remote Sensing Event (JURSE), Lausanne, Switzerland, 2015, pp. 1-4, doi: 10.1109/JURSE.2015.7120490.

[52] R. Wenger, A. Puissant, J. Weber, L. Idoumghar and G. Forestier, "Exploring inference of a land use and land cover model trained on MultiSenGE dataset," 2023 Joint Urban Remote Sensing Event (JURSE), Heraklion, Greece, 2023, pp. 1-4, doi: 10.1109/JURSE57346.2023.10144156.

[53] N. Atif, H. Balaji, S. Mazhar, S. R. Ahamad and M. K. Bhuyan, "Semantic Masking: A Novel Technique to Mitigate the Class-Imbalance Problem in Real-Time Semantic Segmentation," 2022 National Conference on Communications (NCC), Mumbai, India, 2022, pp. 407-412, doi: 10.1109/NCC55593.2022.9806776.

[54] M. Li, Y. Wu, A. G. O. Yeh and F. Xue, "HRHD-HK: A Benchmark Dataset of High-Rise and High-Density Urban Scenes for 3D Semantic Segmentation of Photogrammetric Point Clouds," 2023 IEEE International Conference on Image Processing Challenges and Workshops (ICIPCW), Kuala Lumpur, Malaysia, 2023, pp. 3714-3718, doi: 10.1109/ICIPC59416.2023.10328383.

[55] Y. -S. Ni et al., "Summary of the 2022 Low-Power Deep Learning Semantic Segmentation Model Compression Competition for Traffic Scene In Asian Countries," 2022 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Taipei City, Taiwan, 2022, pp. 1-6, doi: 10.1109/ICMEW56448.2022.9859367.

[56] Z. Zhang, T. Jiang, C. Liu and L. Zhang, "An Effective Classification Method for Hyperspectral Image With Very High Resolution Based on Encoder–Decoder Architecture," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 1509-1519, 2021, doi: 10.1109/JSTARS.2020.3046245.

[57] V. E and B. R. Chilukuri, "Carriageway Edge Detection for Unmarked Urban Roads using Deep Learning Techniques," 2023 Smart City Symposium Prague (SCSP), Prague, Czech Republic, 2023, pp. 1-6, doi: 10.1109/SCSP58044.2023.10146209.

[58] M. A. Mikhalkova, V. O. Yachnaya, E. N. Yablokov and V. R. Lutsiv, "Automatic Detection of Pedestrians in Traffic Scene Images," 2020 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), St. Petersburg, Russia, 2020, pp. 1-6, doi: 10.1109/WECONF48837.2020.9131458.

[59] Z. Zhou et al., "A Novel Ground-Based Cloud Image Segmentation Method by Using Deep Transfer Learning," in IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1-5, 2022, Art no. 8010805, doi: 10.1109/LGRS.2021.3072618.

[60] Shorten, Connor, Taghi M. Khoshgoftaar, and Borko Furht. "Text data augmentation for deep learning." Journal of big Data 8 (2021): 1-34.

[61] C. Li, L. Feng, Q. Wang, Z. Wang, L. Liao and J. Lin, "Parameter optimization for three-level inverter m odel Predictive control based on artificial neural network," 2022 IEEE Vehicle Power and Propulsion Conference (VPPC), Merced, CA, USA, 2022, pp. 1-4, doi: 10.1109/VPPC55846.2022.10003303.

[62] X. Liu, B. Dai and H. He, "Real-time object segmentation for visual object detection in dynamic scenes," 2011 International Conference of Soft Computing and Pattern Recognition (SoCPaR), Dalian, China, 2011, pp. 423-428, doi: 10.1109/SoCPaR.2011.6089281.

ANEXOS

A. Código de extracción de imágenes - MATLAB

```
clc;
clear all;
%Cargamos el video en formato mp4.
a=VideoReader('v1.mp4');
%Ponemos un número para la enumeración de archivos a guardar.
%Nos sirve para la secuencialidad de datos y separación del mismo.
d=1;
%Inicializamos un bucle
for img = 1:a.Numframes
    %Se carga el video cada 15 cuadros y definimos desde qué
    %cuadro se quiere empezar.
    if(mod(img,15)==0 && img>=756)
        %Designamos el nombre del archivo con el que se va guardar
        filename = strcat('im', num2str(d),'.jpg');
        %Leemos el cuadro de interés
        b = read(a, img);
        %La guardamos
        imwrite(c,filename);
        %Pasamos al siguiente cuadro
        d = d+1;
    end
end
```

B. Código de entrenamiento - MATLAB

```
clear all;
clc;
% Cargamos la red de aprendizaje profundo para cargar sus pesos de
% pre-entrenamiento.
resnet18();
%Cargamos el conjunto de datos de imágenes
%y de etiquetados de entrenamiento, validación y pruebas.
pxdata_tra = importdata('TRAINING.mat');
pxdata_val = importdata('VAL.mat');
pxdata_test = importdata('TEST.mat');

imdsTrain = imageDatastore('TRAINING_IMAGES');
imdsVal = imageDatastore('VAL_IMAGES');
imdsTest = imageDatastore('TEST_IMAGES');

pxdsTrain = pixelLabelDatastore(pxdata_tra);
pxdsVal = pixelLabelDatastore(pxdata_val);
pxdsTest = pixelLabelDatastore(pxdata_test);

pxds = combine(pxdsTrain,pxdsVal,pxdsTest);
pxdata = [pxdata_tra; pxdata_val; pxdata_test];
tbl = countEachLabel(pxds);
%%
% Extraemos los nombres de conjunto de
% datos etiquetados.
```

```
classes = pxds.ClassNames;

% Creamos el conjunto de datos etiquetados
% para cada grupo.
labelIDs ={[1] ;[2];[3];[4];[5];[6];[7]};

numTrainingImages = numel(imdsTrain.Files);
numValImages = numel(imdsVal.Files);
numTestingImages = numel(imdsTest.Files);
%%
% Especificamos el tamaño de las imagenes a entrenar
% segun la data adquirida.
imageSize = [640 360 3];

numClasses = numel(pxdata.LabelDefinitions.Name);
% Creamos la nueva arquitectura para la segmentación.
lgraph = deeplabv3plusLayers(imageSize, numClasses, 'resnet18');

% Definimos los pesos.
imageFreq = tbl.PixelCount ./ tbl.ImagePixelCount;
classWeights = median(imageFreq) ./ imageFreq;

pxLayer = pixelClassificationLayer('Name','labels','Classes',...
                        tbl.Name,'ClassWeights',classWeights);
lgraph = replaceLayer(lgraph,"classification",pxLayer);

% Definimos los datos de validación.
pximdsVal = pixelLabelImageDatastore(imdsVal,pxdsVal);

% Define las opciones de entrenamiento.
options = trainingOptions('sgdm', ...
    'LearnRateSchedule','piecewise',...
    'LearnRateDropPeriod',10,...
    'LearnRateDropFactor',0.3,...
    'Momentum',0.9, ...
    'InitialLearnRate',1e-2, ...
    'L2Regularization',0.0001, ...
    'ValidationData',pximdsVal,...
    'MaxEpochs',50, ...
    'MiniBatchSize',8, ...
    'Shuffle','every-epoch', ...
    'CheckpointPath', tempdir, ...
    'VerboseFrequency',2,...
    'Plots','training-progress');

%Implementamos el aumento de datos.
augmenter = imageDataAugmenter('RandXReflection',true,...
    'RandXTranslation',[-10 10],'RandYTranslation',[-10 10]);

pximds = pixelLabelImageDatastore(imdsTrain,pxdsTrain, ...
    'DataAugmentation',augmenter);

%Entrenamos la red.

[net, info] = trainNetwork(pximds,lgraph,options);
```

C. Código de métricas de evaluación - MATLAB

```
%Se realiza un nuevo entrenamiento para definir
%parámetros de métricas por clase, global
%de precisión IoU y BFscore
%Se carga la red que hemos entrenado y las
%imágenes de prueba
pxdsResults = semanticseg(imdsTest,Dokinetv40, ...
    'MiniBatchSize',4, ...
    'WriteLocation',tempdir, ...
    'Verbose',false);
metrics = evaluateSemanticSegmentation(pxdsResults,...
                              pxdsTest,'Verbose',false);
metrics.DataSetMetrics

metrics.ClassMetrics
```

D. Código de pruebas - MATLAB

```
%Se carga la imagen que se desea evaluar
%Se carga la red que hemos entrenado
%Se hace la segmentación
I = imread('im821.jpg');
C = semanticseg(I, Dokinetv40);
numClasses = 7;
cmap = jet(numClasses);
B = labeloverlay(I,C,'Colormap',cmap);
imshow(B)

N = 7;
ticks = 1/(N*2):1/N:1;
%Se carga una barra de leyenda de cada clase
colorbar('TickLabels',...
    cellstr(pxdata.LabelDefinitions.Name),...
    'Ticks',ticks,'TickLength',0,...
    'TickLabelInterpreter','none');

colormap(cmap)
```

E. Código para calcular el tiempo de respuesta - MATLAB

```
% Se lee la imagen a segmentar.
I = imread('i4.jpg');
% Se utiliza la funcion tic para grabar el tiempo de inicio.
tic;
% Se realiza la segmentación
C = semanticseg(I, Dokinetv40);
% Se utiliza la función toc para grabar el tiempo transcurrido.
t=toc;
% Se calcula el tiempo de cuadros por segundo.
fps=1/t;
% Se utiliza la función round para mostrar el tiempo en ms.
```

```
t=round((t*1000),2);
% Se muestran los resultados.
disp(['Tiempo de respuesta: ' num2str(t) ' milisegundos']);
fprintf('La red procesa %.2f cuadros por segundo\n', fps);
```