UNIVERSIDAD AUTÓNOMA DE ZACATECAS

"Francisco García Salinas"

"ADAPTACIÓN DE IMÁGENES CON TRANSPORTE ÓPTIMO Y APRENDIZAJE PROFUNDO PARA UNA MEJORA EN LA DETECCIÓN DE ESPIGAS DE TRIGO"

Tesis para obtener el grado de:

Maestro en Ciencias del Procesamiento de la Información

Presenta

Jesús Eduardo Salas Ibañez

Director:
Dr. Gamaliel Moreno Chávez

Co-Directores:
Dr. José Ismael de la Rosa Vargas
Dr. José de Jesús Villa Hernández

Asesores:
Dr. Efren González Ramírez
Dr. Pedro Daniel Alaniz Lumbreras

Zacatecas, Zac., 29 de abril de 2025

Zacatecas, Zac., 3 de abril de 2025.

C. Jesús Eduardo Salas Ibañez
Estudiante de la MCPI
PRESENTE

At´n: Dr. Huizilopoztli Luna García
Responsable de la MCPI

Nos es grato comunicarle que después de haber sometido a revisión académica la propuesta de Tesis titulada "**Adaptación de Imágenes con Transporte Óptimo y Aprendizaje Profundo para una Mejora en la Detección de Espigas de Trigo**", presentada por el estudiante Lic. Jesús Eduardo Salas Ibañez y habiendo efectuado todas las correcciones indicadas por este Comité Tutorial, se **AUTORIZA** el documento de tesis para su impresión.

Sin más por el momento reciban un cordial saludo.

COMITÉ TUTORIAL
PROCESAMIENTO Y ANÁLISIS DE DATOS

Dr. Gamaliel Moreno Chávez

Dr. José de Jesús Villa Hernández

Dr. Efrén González Ramírez

Dr. José Ismael de la Rosa Vargas

Dr. Pedro Daniel Alaniz Lumbreras

c.c.p. Interesado.

c.c.p. Responsable de la Maestría en Ciencias del Procesamiento de la Información.

Carta de similitud núm. 793/IyP
Zacatecas, Zacatecas 29/Abril/2025

Dr. Huizilopoztli Luna García
Responsable de la MCPI – UAZ
Presente

Estimado Dr. Huizilopoztli,

Después de saludarlo, sirva el presente oficio para notificar que el documento

"Adaptación de imágenes con transporte óptimo y aprendizaje profundo para una mejora en la detección de espigas de trigo"
de Jesús Eduardo Salas Ibañez

Fue analizado con el software Copyleaks, con la intención de detectar similitudes; el resultado en cuestión fue

4.2 % de similitud

De acuerdo a lo anterior, el porcentaje se considera **ACEPTABLE** de acuerdo a los estándares internacionales.

Atentamente
"Somos Arte, Ciencia y Desarrollo Cultural"

Dr. Carlos Francisco Bautista Capetillo
Coordinador de Investigación y Posgrado
Universidad Autónoma de Zacatecas

Zacatecas, Zac., 3 de abril de 2025

Carta Cesión de Derechos

A QUIEN CORRESPONDA

El que suscribe C. **Jesús Eduardo Salas Ibañez**, alumno del Programa de **Maestría en Ciencias del Procesamiento de la Información**, con número de matrícula **34151937** y adscrito a la Unidad Académica de Ingeniería Eléctrica de la Universidad Autónoma de Zacatecas, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del **Dr. Gamaliel Moreno Chávez** y cede los derechos del trabajo titulado **"Adaptación de Imágenes con Transporte Óptimo y Aprendizaje Profundo para una Mejora en la Detección de Espigas de Trigo"** a la Universidad Autónoma de Zacatecas para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o directores del trabajo. Este puede ser obtenido escribiendo al correo electrónico eduardo.si@uaz.edu.mx o estableciendo contacto con el responsable del Programa de Maestría, quien turnará la solicitud al autor y directores del trabajo de investigación. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo. Agradezco de antemano su atención a la presente, reciba un cordial saludo.

ATENTAMENTE

Jesús Eduardo Salas Ibañez

Agradecimiento Especial

Expreso mi más sincero y profundo agradecimiento al **Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT)** por el invaluable apoyo económico brindado a través de la convocatoria “Becas Nacionales (Tradicional) 2023-1”, mismo que me permitió dedicarme de tiempo completo a los estudios de maestría y a la realización de este proyecto de investigación dentro del Programa de Maestría en Ciencias del Procesamiento de la Información de la Universidad Autónoma de Zacatecas.

Agradecimientos

Agradezco de manera especial a mi director de tesis, el Dr. Gamaliel Moreno, por permitirme trabajar en este proyecto bajo su guía, así como por transmitirme sus conocimientos y su gran dedicación a la investigación científica, lo que sin duda lo vuelve un ejemplo a seguir. De igual manera, agradezco a mis co-directores, el Dr. Ismael de la Rosa y el Dr. Jesús Villa, así como a mis asesores, el Dr. Efrén González y el Dr. Daniel Alaniz, quienes con sus ideas y retroalimentación me permitieron culminar este trabajo de manera satisfactoria.

A mis profesores de la maestría, por todos los conocimientos que me transmitieron en el aula y que me permitieron formarme en el área de los datos y la inteligencia artificial. A mis compañeros y amigos de la maestría, quienes definitivamente formaron parte importante de esta etapa y de quienes me llevo también algunas enseñanzas y buenas experiencias.

Finalmente, quisiera agradecer enormemente a mis padres Javier y Margarita, a mis hermanos Ingrid y Jayro, y a mi novia Pau, ya que estuvieron presentes durante todo este proceso, apoyándome de muchas maneras y dándome ánimos cuando más lo necesitaba. Ustedes son las personas más importantes en mi vida y mi mayor fuente de inspiración, por lo que mis logros también son de ustedes.

Dedicatoria

Dedico este trabajo con mucho cariño a mis padres, quienes me han apoyado en cada paso de mi trayectoria académica y me han enseñado a siempre dar lo mejor de mí y luchar por mis sueños.

Resumen

El trigo es uno de los cultivos más importantes para la seguridad alimentaria mundial, por lo que aumentar la eficiencia en la producción de este cereal representa una prioridad. La densidad de espigas es uno de los parámetros más importantes para realizar una estimación temprana de la cosecha; por esta razón, se han propuesto diversos métodos para realizar un conteo automático de las espigas de trigo en imágenes a color. Entre estos métodos han destacado las redes neuronales de detección de objetos como YOLO o R-CNN. Sin embargo, la detección precisa de espigas en imágenes provenientes de distintos dominios puede ser un reto para cualquier modelo, pues el aspecto de estas puede variar debido a características ambientales o a la variedad del cultivo y la etapa fenológica en que se encuentre. Las técnicas de adaptación de dominio permiten reducir estas variaciones en el aspecto de las espigas, proporcionando una forma para mejorar el desempeño de los modelos de detección de objetos. En este trabajo se aplicó una metodología de adaptación de dominio basada en el transporte óptimo, la cual permite modificar la distribución de colores de una imagen para que se alinee con la de una imagen al azar de otro dominio. Al aplicar este método a las imágenes de la base de datos Global Wheat Head Detection 2021 y posteriormente entrenar un modelo YOLOv5s, se encontró que la precisión promedio media (mAP50) del modelo aumenta hasta un 4.1 % comparado con el mismo modelo sin aplicar este método. Además, al eliminar imágenes para crear una base de datos balanceada, este aumento en el mAP50 se dispara hasta un 8.4 %. Así, la metodología propuesta presenta una alternativa que mejora la detección de espigas sin aumentar la complejidad de la red neuronal o el tamaño de la base de datos.

Palabras clave: Aprendizaje profundo, detección de objetos, transporte óptimo, adaptación de dominio, espigas de trigo.

Abstract

Wheat is one of the most important crops for global food security; for this reason, developing a highly efficient production of this cereal is a priority. Wheat head density is a key parameter for early yield estimation; therefore, various methods for automatic counting of the heads in color images have been proposed. Object detection neural networks such as YOLO or R-CNN have excelled in this task. However, accurate detection of wheat heads becomes a challenge when dealing with images from different domains, mainly due to differences in environmental conditions, crop variety, and phenological stage. Domain adaptation techniques provide a way to improve the performance of object detection models, as they reduce the variations in the heads' appearance. In this work, a domain adaptation technique based on optimal transport was applied. This technique modifies the color distribution of an image, matching it with a randomly selected image from another domain. Applying this method to the Global Wheat Head Detection 2021 dataset, and then training a YOLOv5s model with these images, led to an increase of up to 4.1 % in the mean average precision (mAP50), when compared to a baseline model in which no domain adaptation is used. Moreover, by removing images to create a balanced dataset, the mAP50 improves by up to 8.4 %. Thus, the proposed methodology provides an alternative to enhance wheat head detection without increasing the model complexity or the size of the dataset.

Keywords: Deep learning, object detection, optimal transport, domain adaptation, wheat heads.

Contenido General

Pág.

Pág.

Índice de figuras

Figura Pág.

Índice de tablas

Capítulo 1

Introducción

En este primer capítulo se presentan los principales antecedentes al problema de investigación, así como la definición del mismo; se plantea la relevancia del trabajo a través de la justificación, se enuncian las preguntas, objetivos e hipótesis que guían el proceso de investigación, y se finaliza con una breve descripción del trabajo a realizar y la estructura de la tesis.

1.1. Antecedentes

A nivel mundial, la agricultura en los próximos 35 años se enfrentará con el reto de asegurar el abasto de alimentos para la población mundial. De acuerdo con la FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura), para 2050 el volumen de producción agrícola debe aumentar en un 70 %, por lo que, en el caso de los cereales, el rendimiento de los cultivos debe incrementar de 1.2 a 4.3 ton/ha. En cuanto a México, de acuerdo al INEGI (Instituto Nacional de Estadística, Geografía e Informática) y SAGARPA (Secretaría de Agricultura y Desarrollo Rural), el área cultivable actual es de 27.8 millones ha, pero se estima que podría disminuir debido al cambio de uso de suelo para la construcción de viviendas. Debido a esto, la mejor forma de aumentar la producción agrícola es seguir la tendencia mundial de aumentar el rendimiento por unidad de superficie [1].

El trigo es el segundo mayor cultivo de cereal en el mundo, solo detrás del arroz, con una producción anual de 735 millones de toneladas. Además, es uno de los principales alimentos de la dieta del ser humano, con una contribución del 19 % de las calorías consumidas y un 21 % de la ingesta de proteína [2]. Su importancia radica también en el hecho de que es cultivado en regiones de distinta naturaleza en todo el mundo y que está muy presente en la cultura e incluso en la religión

de algunas sociedades [3]. El trigo ocupa un rol estratégico en la seguridad alimentaria mundial, pues casi el 25 % de la producción global se comercializa internacionalmente, a diferencia del arroz, donde solo se comercializa alrededor del 0.4 %. Durante los próximos años, la producción de trigo se verá desafiada por el cambio climático, con múltiples estudios que estiman que habrá un decrecimiento del 7 % en la cosecha por cada grado que incremente la temperatura. La adopción de nuevos avances tecnológicos relevantes a la producción de trigo proporcionará una gran oportunidad para mejorar la producción sustentable de este cereal [4].

La fenología del trigo comprende las distintas etapas del ciclo de vida de la planta, por lo que su entendimiento es esencial para maximizar la producción. Se han desarrollado diversas escalas para medir el crecimiento de este cultivo, entre las que se encuentran la de Feekes (11 fases), Haun y Zadoks (100 etapas) [5]. De una manera resumida, los 3-4 meses de duración del ciclo de vida del trigo pueden entenderse a través de las siguientes 7 etapas [6]:

- Germinación: La semilla absorbe agua y la planta comienza a crecer.
- Crecimiento de la plántula: La planta obtiene sus primeras hojas y raíces.
- Macollaje: La planta genera más brotes o macollos, los cuales pueden formar sus propias espigas y aumentar el rendimiento del cultivo.
- Encañado: El tallo crece y aparecen nuevas hojas, marcando el inicio de su etapa de reproducción.
- Espigado y floración: La planta desarrolla sus espigas, mientras que las flores dentro de estas se polinizan y crean el grano.
- Desarrollo y llenado del grano: Los granos crecen y se llenan de almidón y nutrientes.
- Maduración: La planta de trigo alcanza su etapa final, el grano se seca y la planta está lista para la cosecha.

La producción de granos en cereales tiene tres componentes principales: el número de espigas por unidad de área (población de espigas), número de granos por espiga y peso de los granos. Cada una de estas componentes es importante; sin embargo, en estudios que investigaban los factores

que reducen el rendimiento de la cosecha de trigo, se encontró que la población de espigas influía de manera significativa en la producción de trigo [7]. Estimar la población de espigas de manera rápida es una prioridad para monitorear la eficiencia del manejo de los cultivos, así como realizar una predicción temprana de la producción de grano. Actualmente se realizan distintos estudios experimentales con cultivos de trigo en los que el número de espigas se cuenta de manera manual, lo cual es un proceso lento y tedioso para los investigadores, además de que no existe un protocolo estandarizado para realizar este conteo, lo cual lo hace ser más propenso a errores o variaciones entre los resultados experimentales obtenidos al utilizar distintas metodologías [8].

En años recientes, la tecnología de sensores ha ganado terreno en el área del monitoreo automático del crecimiento de los cultivos, ya que proporciona gran precisión y tiene el potencial de reducir los costos e incrementar la producción. Una de las tecnologías más importantes en este aspecto son las cámaras digitales, pues las imágenes a color pueden ser procesadas para estimar distintas métricas relacionadas con el crecimiento de los cultivos. Además del avance en las cámaras, también se ha hecho lo propio en el análisis de las imágenes, principalmente con la introducción de técnicas de aprendizaje automático y posteriormente de las redes neuronales profundas pertenecientes al aprendizaje profundo [9].

El aprendizaje automático o machine learning ha creado la oportunidad de cuantificar y entender los procesos relacionados a datos en el contexto de la agricultura. Las metodologías del aprendizaje automático involucran un proceso de aprendizaje en el que la computadora puede aprender a partir de la experiencia, es decir, aprender con datos de entrenamiento a realizar una tarea específica. Sus aplicaciones en la agricultura se centran principalmente en la predicción de cosecha, detección de enfermedades, detección de maleza, calidad de cultivos y reconocimiento de especies [10]. De acuerdo a estudios recientes sobre el uso de este tipo de tecnologías en la predicción de cosechas [11] y conteo de plantas [12], el uso de algoritmos clásicos de procesamiento de imágenes junto a técnicas de aprendizaje automático en estas tareas ha tenido un éxito limitado, pues la poca disponibilidad de datos, así como la poca variación en las condiciones de obtención de las imágenes que suelen existir en estas investigaciones, generan problemas para obtener modelos que puedan ser utilizados en una mayor diversidad de condiciones ambientales, como las presentes en el campo.

La generación de bases de datos extensas y variadas y una mayor disponibilidad de tarjetas gráficas (GPUs) han tenido un impacto significativo en las investigaciones de esta área, pues han promovido el enfoque hacia el uso de modelos de redes neuronales. Las redes convolucionales han destacado en esta área y los tipos y arquitecturas típicas son: AlexNet, VGGNet, RCNN, ResNet, YOLO, SSD, EfficientDet [12]. Actualmente, las redes neuronales profundas que han ganado mayor popularidad para tareas de detección o conteo de hojas son redes como YOLO o R-CNN, pues realizan un conteo mediante métodos de segmentación y detección de objetos, en donde realizan a la vez una localización y clasificación del objeto dentro de la imagen.

1.2. Planteamiento del problema de investigación

Con el avance del aprendizaje profundo se han logrado crear métodos de detección de objetos más precisos, los cuales se han utilizado en distintas tareas, entre las que destaca la detección y el conteo de espigas de trigo. En este sentido, se han realizado diversos esfuerzos para obtener mejores modelos que detecten las espigas de trigo en una gran variedad de condiciones, buscando modelos generalizables que puedan adecuarse a las condiciones cambiantes que surgen al tratar de aplicarlos en el campo. Uno de los esfuerzos más importantes corresponde al Global Wheat Challenge (GWC) [13], impulsado por David et al. con la creación de la base de datos Global Wheat Head Detection (GWHD) [14], donde se tenía por objetivo encontrar los mejores modelos de redes neuronales que pudieran realizar correctamente la detección de espigas de trigo en escenarios diversos, valiéndose de la gran variabilidad de las imágenes presentes en la base de datos GWHD. Sin embargo, la búsqueda de generalización mediante el uso de este tipo de bases de datos introduce nuevos problemas, pues existe un cambio de dominio donde la variación en las imágenes utilizadas para el entrenamiento de un modelo es distinta a la variación de las imágenes utilizadas en pruebas [15]. Estos cambios de dominio surgen principalmente debido a una diferencia en las condiciones ambientales y de equipo en la captura de las imágenes, e introducen un desafío para los algoritmos, dificultando la detección de las espigas en ciertas imágenes. En este contexto, un dominio representa un conjunto de imágenes que comparten ciertas características. Como se puede ver en la Figura 1.1 mediante imágenes pertenecientes a la base de datos GWHD, estos cambios de

dominio pueden deberse a la diversidad de ubicaciones en donde fueron obtenidas las imágenes, a un cambio de iluminación debido a la hora del día o posición de la cámara, un cambio de equipo de captura, o a un cambio de apariencia de las espigas debido a las distintas fases del ciclo de crecimiento de la planta [16].

Figura 1.1 Cambios de dominio debidos a distintas ubicaciones, condiciones de iluminación y etapas de desarrollo de la espiga

1.3. Justificación del problema de investigación

La automatización en la detección y el conteo de espigas de trigo proporciona una forma más rápida y estandarizada de obtener la población de espigas, la cual es una métrica importante para poder realizar una buena predicción de la cosecha de grano. Las técnicas de aprendizaje profundo, particularmente las redes neuronales convolucionales, han demostrado ser eficaces en la ejecución de esta tarea, teniendo ventajas como una buena precisión y un significativo ahorro de tiempo respecto a las técnicas de conteo manual, además de una mejor capacidad de generalización que las metodologías basadas en algoritmos clásicos de procesamiento de imágenes y aprendizaje automático. Las redes YOLO representan un buen punto de partida en la experimentación de mejores metodologías, ya que en el GWC 2021 fueron el tipo de arquitectura más utilizada entre los modelos ganadores [13].

Los métodos de adaptación de dominio toman en cuenta los cambios de dominio o distribución debidos a la obtención de datos de distintas fuentes [17], por lo que el uso de estos mejora las capacidades de generalización que puede tener un sistema de visión por computadora [18]. Al entrenar una red neuronal en un dominio, esta puede tener un desempeño reducido al aplicarse a otro dominio, por lo que utilizar técnicas de adaptación de dominio para alterar las imágenes y hacer que todas se alineen con un dominio objetivo permite mejorar el desempeño de la red en dominios donde no se desempeñaba correctamente [19].

Para poder realizar los cambios de dominio de las imágenes, el marco definido por el Transporte Óptimo proporciona una buena alternativa, pues permite comparar distribuciones estadísticas tomando en cuenta la información espacial de las modas de la densidad. Esto proporciona una mayor flexibilidad y generalización para comparar histogramas a través del plan de transporte, lo cual es de gran utilidad al realizar una transferencia de color entre imágenes donde se conserve la estructura general de los objetos presentes en éstas [20]. De esta manera, se podrían adaptar todas las imágenes a un mismo dominio y facilitar la detección de espigas a un algoritmo diseñado para esta tarea, como las redes YOLO antes mencionadas. Esto ayudaría a reducir el número de patrones que debe aprender la red, dando una posible mejora a su desempeño sin recurrir al aumento de datos o a aumentar la complejidad de la red.

1.4. Preguntas de investigación

- ¿La introducción de técnicas de adaptación de dominio basadas en transporte óptimo mejora la precisión de detección de una red YOLO?
- ¿Cuánto puede aumentar la precisión de la red YOLO al introducir la adaptación de dominio con transporte óptimo en las imágenes?
- ¿Qué algoritmo de transporte óptimo resulta más viable para realizar la adaptación de dominio de una gran cantidad de imágenes?
- ¿Qué métricas resultan más útiles a la hora de evaluar la detección de espigas de trigo?

1.5. Objetivo general

Aumentar en un 3 % la precisión con la que un modelo YOLO detecta espigas de trigo en imágenes RGB, mediante la introducción de técnicas de adaptación de dominio basadas en la teoría del transporte óptimo.

1.5.1. Objetivos específicos

- Mostrar que la integración de adaptación de dominio basada en transporte óptimo proporciona un aumento en la precisión de detección de espigas de trigo.
- Determinar un dominio objetivo donde se realice de mejor manera la detección de espigas, es decir, un conjunto de imágenes con la distribución de colores adecuada en donde la red YOLO obtenga mayor precisión.
- Identificar un algoritmo que permita aproximar de manera rápida y precisa el transporte óptimo entre imágenes.
- Identificar las condiciones en las que el transporte óptimo puede proporcionar un mayor aumento a la precisión del modelo.
- Definir las métricas que mejor representen el desempeño del modelo YOLO en la detección de espigas y utilizarlas en su validación.

1.6. Hipótesis

La adaptación de dominio en imágenes de espigas de trigo, mediante el uso de técnicas fundamentadas en la teoría del transporte óptimo, permite aumentar la precisión en la detección de espigas de un modelo YOLO en, por lo menos, un 3 %.

1.7. Trabajo a Realizar

La realización del presente trabajo de investigación se llevará a cabo de acuerdo con los siguientes pasos:

- Obtención de una base de datos de imágenes de espigas de trigo, donde se incluyan imágenes de distintas variedades de trigo, tomadas durante distintas etapas fenológicas, con distintas condiciones de iluminación y un número variable de espigas en cada imagen.
- Preprocesamiento de dicha base de datos para que pueda ser utilizada adecuadamente por un modelo YOLO, lo que incluye utilizar el formato adecuado para los archivos de etiquetas de cada imagen y organizar los directorios de archivos de etiqueta e imágenes de manera correcta.
- Selección de una versión adecuada de la arquitectura YOLO, con la cual se obtengan resultados aceptables en la identificación de las espigas de trigo.
- Selección de un dominio objetivo para las imágenes, así como la elección de un algoritmo de transporte óptimo que realice la adaptación de dominio de las imágenes hacia el dominio objetivo, priorizando la rapidez del algoritmo para que pueda ser usado en todo el conjunto de imágenes.
- Integración del algoritmo de transporte óptimo con la versión de YOLO elegida, así como el entrenamiento del modelo.
- Realización de pruebas y refinamiento del modelo, experimentando con distintos parámetros para encontrar los que influyan de mejor manera en la detección de las espigas.
- Presentación de resultados, donde se incluirán todas las métricas con que se evalúa al modelo y se presentarán las conclusiones alcanzadas después de llevar a cabo todo el proceso de investigación.

1.8. Estructura de la tesis

Los capítulos restantes del presente trabajo de tesis se estructuran de la siguiente manera:

- **Capítulo 2**. Aborda el marco teórico de la investigación, describiendo conceptos clave y teorías relacionadas al problema de investigación, entre los que se encuentran la visión computacional, aprendizaje profundo, redes neuronales convolucionales para la detección de objetos y transporte óptimo. Además, se incluye un análisis de algunos trabajos relacionados a esta investigación.

- **Capítulo 3**. Describe la metodología propuesta para la investigación, desde la obtención de las imágenes y su preprocesamiento hasta la arquitectura de la red neuronal elegida, el algoritmo de transporte óptimo utilizado, el diseño de los experimentos propuestos y las métricas utilizadas para evaluar los modelos.

- **Capítulo 4**. Presenta los resultados obtenidos a partir del modelo propuesto y los distintos experimentos realizados, así como una discusión de estos.

- **Capítulo 5**. Concluye la tesis con los objetivos alcanzados, las contribuciones hechas a partir de esta investigación y el planteamiento de futuros trabajos.

Capítulo 2

Marco Teórico

Este capítulo abarca una descripción de los principales conceptos y teorías relevantes para esta investigación, entre los que se encuentran la visión computacional, aprendizaje profundo y transporte óptimo. Además, se incluye un breve análisis de los principales estudios relacionados, sus contribuciones y limitaciones y el cómo la propuesta de investigación se diferencia de estos trabajos, culminando con el modelo de investigación planteado.

2.1. Visión Computacional

La visión computacional es un área que involucra hacer que una computadora pueda "ver", ya que se utiliza una cámara y computadora en lugar del ojo humano para identificar, rastrear y medir distintos objetivos en una imagen [21]. Se puede entender también como una combinación de distintos conceptos, técnicas e ideas de procesamiento digital de imágenes, reconocimiento de patrones, inteligencia artificial y gráficos por computadora. Las tareas principales de la visión computacional se relacionan con el proceso de obtener información acerca de eventos o descripciones a partir de las imágenes digitales. Algunos autores usan los términos de visión computacional y procesamiento de imágenes para referirse a una misma área, aunque el propósito de la primera es crear modelos y extraer datos e información de las imágenes, mientras la segunda tiene como propósito implementar transformaciones computacionales a las imágenes como modificar la nitidez, el contraste, eliminar ruido, entre otras cosas [22].

Para llevar a cabo el análisis matemático de una imagen, ésta se puede definir como una función de dos dimensiones f(x,y) donde x e y son coordenadas espaciales y la amplitud de f en cada par de coordenadas (x,y) es llamada la intensidad de una imagen en ese punto. Cuando x, y, f son todos finitos y cantidades discretas, la imagen puede ser llamada una imagen digital. Una imagen

digital se compone de un número finito de elementos, donde cada uno de ellos tiene un valor de intensidad, coordenadas particulares y es llamado pixel [23]. En una imagen a escala de grises de 8-bits cada pixel tendrá una intensidad en el rango de 0 a 255, como se muestra en la Figura 2.1.

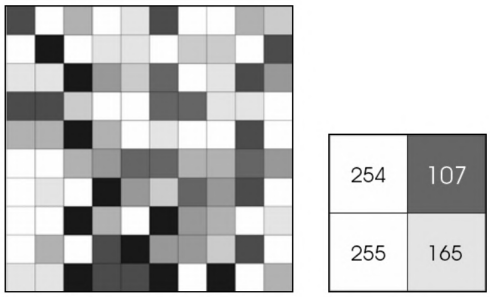


Figura 2.1 Cada pixel tiene un valor de 0 (negro) a 255 (blanco) [24].

Los algoritmos que se han desarrollado en el área de visión computacional pretenden realizar distintas tareas de inspección visual, entre las que se pueden mencionar principalmente tres: reconocimiento de objetos para determinar si una imagen contiene un objeto en específico, detección de objetos para localizar instancias de objetos semánticos de una clase dada, y entendimiento de la escena para dividir la imagen en segmentos significativos para su análisis [25]. A continuación, se presenta una descripción de estas tres tareas, así como del conteo de objetos.

2.1.1. Clasificación de imágenes

La clasificación de imágenes tiene como objetivo clasificar automáticamente las imágenes en clases predefinidas. El trabajo de investigación en esta área se enfocaba anteriormente en diseñar características invariantes de la escala, representaciones de características y clasificadores, lo cual presentaba complicaciones con objetos en imágenes naturales con un fondo complicado, variaciones de color, textura, iluminación y poses cambiantes. Se necesitaba entonces desarrollar técnicas avanzadas que permitieran mejorar la precisión en la clasificación, y fue entonces cuando las técnicas de aprendizaje profundo demostraron gran desempeño en tareas de clasificación, mostrando que los métodos basados en Redes Neuronales Convolucionales (CNN por sus siglas en inglés) se desempeñan mejor que otros métodos convencionales en presencia de variaciones a gran escala [25].

2.1.2. Detección de objetos

La detección de objetos implica determinar y localizar las instancias de un objeto ya sea para un gran número de categorías predefinidas en imágenes naturales o para un objeto en particular. Se relaciona con la clasificación de imágenes en el sentido de que en ambas tareas se debe manejar un gran número de objetos altamente variables, sin embargo, se considera que la detección de objetos presenta mayor dificultad, pues requiere identificar de manera precisa la localización del objeto de interés. Existen bases de datos para evaluar la detección de objetos como Pascal-VOC 2007 con 20 clases o MS-COCO con 80, las cuales fueron creadas para construir sistemas de detección de objetos robustos y de propósito general. La detección en estas dos bases de datos es evaluada a través de dos métricas: Precisión Promedio (AP por sus siglas en inglés) donde se cuentan los cuadros delimitadores detectados correctamente para los cuales la relación de superposición excede 0.5, y Precisión Promedio media (mAP por sus siglas en inglés) donde se promedian los valores de AP asociados con diferentes umbrales de la relación de superposición [25].

El enfoque utilizado en la detección de objetos puede dividirse en dos categorías: los métodos que utilizan aprendizaje automático y los que usan aprendizaje profundo. Los primeros necesitan establecer ingeniería de características, valiéndose de métodos como la transformada general de Hough, el detector de esquinas de Harris y el invariante de escala y rotación (SIFT por sus siglas en inglés). Los segundos no necesitan ingeniería de características y se desempeñan mejor que los métodos basados en aprendizaje automático cuando se trabaja con datos a gran escala. Los principales métodos de detección de objetos basados en aprendizaje profundo son los modelos R-CNN y YOLO, así como sus iteraciones subsecuentes [26].

2.1.3. Segmentación de imágenes

La segmentación de imágenes tiene que ver con cómo se presentan exactamente los objetos en una escena visual, y sus diferencias con la clasificación de imágenes y detección de objetos se pueden apreciar de mejor manera en la Figura 2.2. La segmentación de imágenes se conoce también como clasificación a nivel de pixeles, ya que pretende dividir una imagen en regiones significativas clasificando cada pixel dentro de una entidad específica. La segmentación puede dividirse a su

vez en dos categorías, las cuales corresponden a la segmentación semántica y la segmentación de instancias [25].

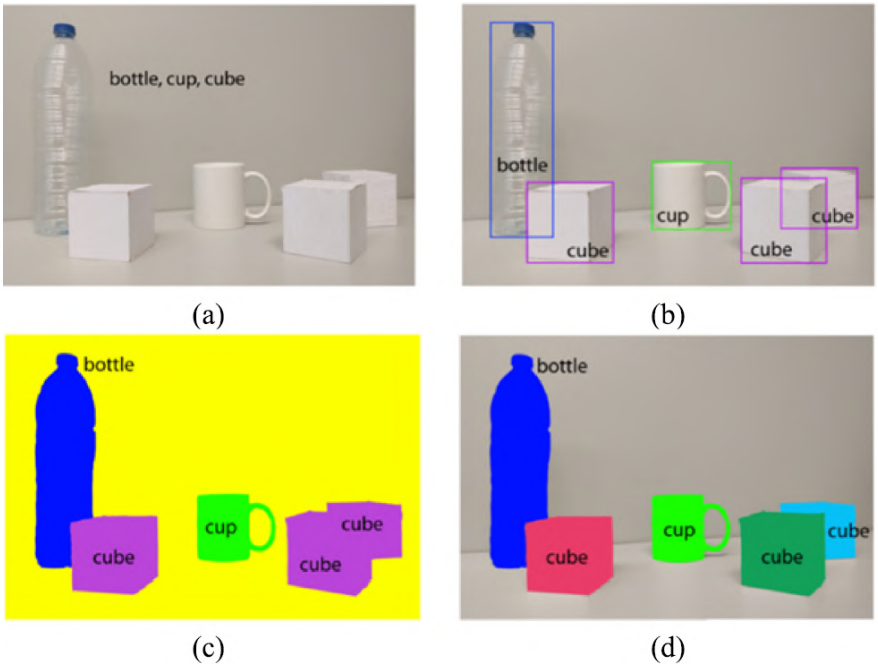


Figura 2.2 Ejemplo de distintas tareas de percepción visual: (a) clasificación de imágenes, (b) detección de objetos, (c) segmentación semántica, (d) segmentación de instancias[25].

La segmentación semántica es una técnica donde se asocia cada pixel de una imagen digital con una etiqueta de clase, clasificando pixeles de la imagen en una o más clases en lugar de objetos reales que no son semánticamente interpretables. La segmentación de instancias detecta todas las instancias de una clase con la funcionalidad extra de demarcar instancias separadas de cualquier clase, por lo que se dice que incorpora las funcionalidades de la detección de objetos y la segmentación semántica [27].

2.1.4. Conteo de objetos

El conteo de objetos es la tarea de predecir con precisión el número de instancias de diferentes categorías de objetos presentes en escenas naturales, como se puede observar en la Figura 2.3. Las categorías comunes de objetos en escenas naturales pueden variar desde frutas hasta animales, y el

conteo debe realizarse en una variedad de escenas distintas [28]. Los algoritmos de conteo de objetos utilizados anteriormente requerían conteo manual y métodos tradicionales de procesamiento de imágenes, por lo cual había un gran margen de mejora en la precisión y generalización de los modelos. Así, el uso del aprendizaje profundo en tareas de conteo trajo ventajas como una mejor precisión y la habilidad de contar objetos en fondos complejos y con una distribución de objetos más densa [29].

Figura 2.3 Ejemplo de conteo de objetos en la base de datos COCO [28].

2.2. Aprendizaje profundo

El deep learning o aprendizaje profundo es un área del aprendizaje automático (o machine learning) basada principalmente en las Redes Neuronales Artificiales (ANN, por sus siglas en inglés), las cuales son un paradigma computacional que se basa en el funcionamiento del cerebro humano [30]. El aprendizaje profundo permite que modelos computacionales de múltiples capas de procesamiento (constituidas por neuronas artificiales) aprendan y representen datos abstractos imitando la forma en que el cerebro humano entiende la información multimodal y capturando la estructura de los datos a gran escala. Esta área ha generado mucho interés en los últimos años ya que ha demostrado lograr un mejor desempeño que otras técnicas clásicas en tareas donde se utilizan datos complejos como imágenes, audio, datos médicos o datos de sensores [31].

En el campo de la visión computacional, el aprendizaje profundo ofrece ventajas sobre los métodos clásicos, principalmente al alcanzar mayor precisión en tareas como la clasificación de imágenes, detección de objetos o segmentación semántica, requiriendo menor análisis de expertos y ajustes del modelo al ser modelos que son entrenados en lugar de programados. Además, ofrece mayor

flexibilidad, pues se puede re-entrenar un modelo con una base de datos personalizada para cualquier caso de uso, en lugar de limitarse a dominios específicos como pasa con algunos algoritmos clásicos de visión computacional [30]. A continuación, se explicará el funcionamiento general de las ANN y de algunas de sus variantes relacionadas con las imágenes y principalmente con la detección de objetos.

2.2.1. Redes Neuronales Artificiales

Las redes neuronales artificiales están inspiradas en la forma en que funcionan las neuronas biológicas del cerebro humano, el cual se compone de un gran número de neuronas interconectadas. Cada neurona realiza una tarea simple como respuesta a una señal de entrada, sin embargo, una red de neuronas conectadas es capaz de realizar tareas complejas como reconocer un rostro. Una neurona biológica está compuesta principalmente por dendritas, el cuerpo de la célula y un axón, como se muestra en la Figura 2.4. Las dendritas son ramas que se conectan al cuerpo de la célula y reciben señales de otras neuronas, el axón es el transmisor de la neurona y envía señales a las neuronas cercanas. La conexión entre el axón de una neurona y las dendritas de otra neurona es llamada sinapsis, y a través de estas sinapsis se transmiten señales electroquímicas. Cuando la señal total que recibe una neurona es mayor al umbral de la sinapsis la neurona se activa, con lo cual manda señales electroquímicas a las neuronas cercanas [32].

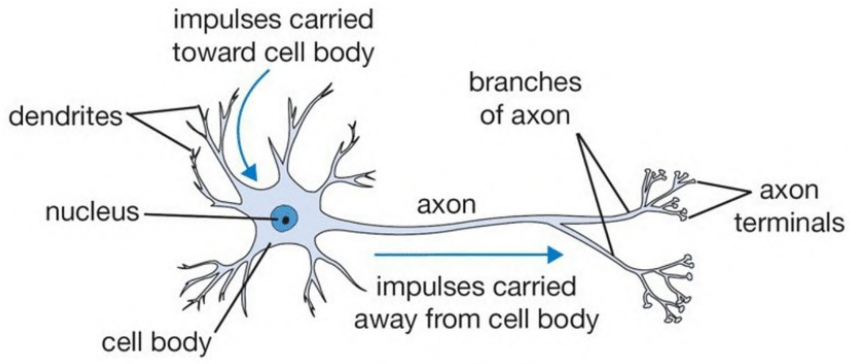


Figura 2.4 Neurona biológica [33].

Una ANN es un modelo computacional con un gran número de nodos o neuronas conectadas entre sí. Cada nodo representa una función de salida o función de activación específica. La conexión entre dos nodos representa un peso para la señal que pasa a través de esta conexión y es equivalente a la memoria de la red neuronal. La salida de la red dependerá de cómo se encuentra conectada la red, el valor de los pesos y la función de activación utilizada [34]. Una neurona artificial se puede representar de acuerdo a la Figura 2.5, donde x_i son las señales de entrada, w_i son los pesos asociados a dichas entradas, b es el bias y f es la función de activación. Cada nodo o neurona recibirá múltiples entradas provenientes de otras neuronas que tendrán pesos asociados, representando la magnitud de la sinapsis. Luego, se calcula la suma ponderada de las entradas multiplicadas por los pesos y este valor pasa por una función de activación para obtener una señal de salida, la cual será enviada a los nodos cercanos [32].

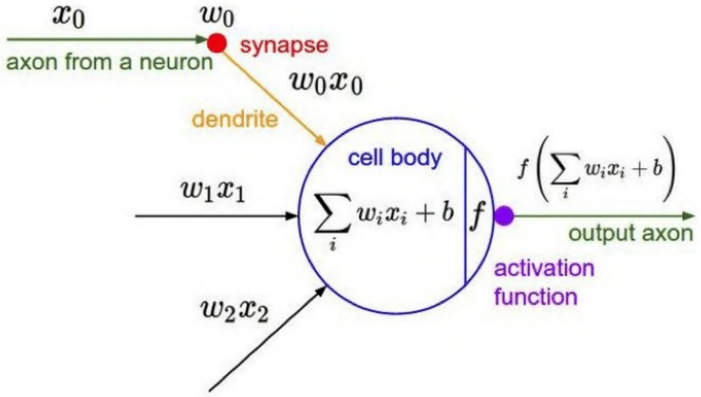


Figura 2.5 Modelo de una neurona artificial con tres entradas y una salida [33].

El tipo de neuronas en una red neuronal se divide en tres categorías: neuronas de entrada, neuronas de salida y neuronas ocultas. Las neuronas de entrada aceptan señales y datos del mundo exterior, mientras que las de salida obtienen el resultado del procesamiento hecho por toda la red. Las neuronas ocultas se encuentran en medio de las de entrada y salida y no pueden ser observadas desde afuera del sistema [34]. Las neuronas suelen estar organizadas en arreglos lineales llamados capas, y en la Figura 2.6 se puede observar una red neuronal con una capa de entrada, dos capas ocultas y una de salida. En una red normalmente se cuenta con una capa de entrada y una de salida, y puede haber desde cero hasta una gran cantidad de capas ocultas. El diseño de una arquitectura

para la red neuronal implica determinar el número de neuronas por capa, el número de capas en la red y la forma en que se conectan los nodos. Esto en un principio suele hacerse por intuición, pero la arquitectura es optimizada al cabo de varios ciclos de experimentos [32].

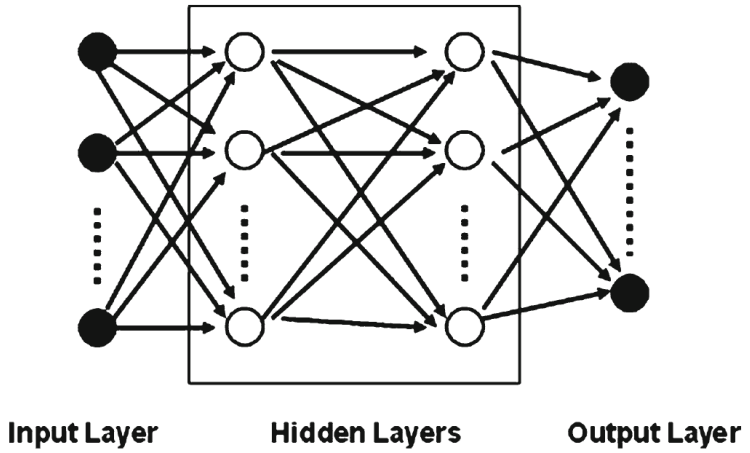


Figura 2.6 Arquitectura general de una red neuronal hacia adelante (feedforward) [32].

2.2.2. Funciones de activación

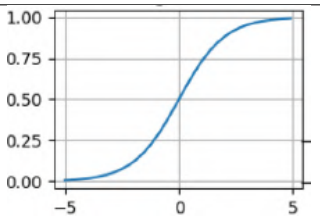
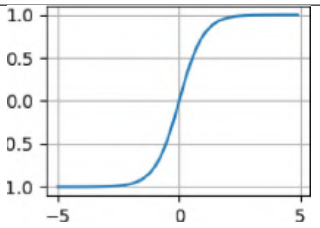
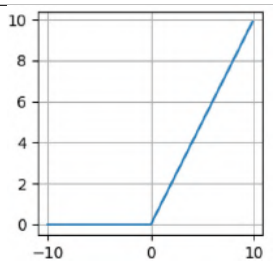
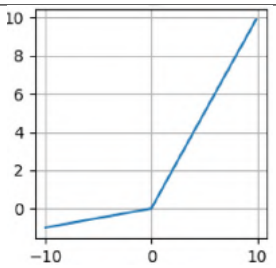
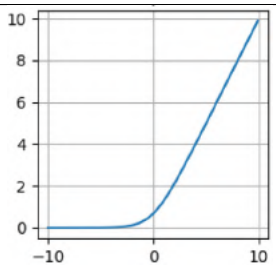
De acuerdo al esquema de un nodo de una red neuronal como el que se puede ver en la Figura 2.5, $\{x_1, x_2, \ldots, x_n\}$ son las entradas de la neurona artificial y $\{w_1, w_2, \ldots, w_n\}$ son los pesos asociados a cada una de las entradas. Si se introduce un sesgo o bias b, entonces la salida y de la neurona puede calcularse a través de una función de activación f mediante la siguiente ecuación [35]:

$$y = f(xw^{\intercal} + b) \tag{2.1}$$

La función de activación representa una parte importante de la arquitectura de una red neuronal, ya que una función de activación no lineal le da a la red propiedades no lineales, permitiéndole diferenciar los datos que no se pueden clasificar linealmente en un espacio de datos determinado. Por esta razón, en la literatura relacionada a las redes neuronales existe un gran interés en identificar y definir funciones de activación que puedan mejorar el desempeño de una red [36]. A continuación,

en la Tabla 2.1 se definen algunas de las funciones de activación más utilizadas en la literatura de redes neuronales.

Tabla 2.1 Algunas funciones de activación utilizadas en redes neuronales [36].

Nombre	**Ecuación**	**Rango**	**Gráfica**
Sigmoide	$\sigma(x) = \dfrac{1}{1+e^{-x}}$	$(0, 1)$	
Tangente hiperbólica	$\tanh x = \dfrac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$(-1, 1)$	
ReLU	$f(x) = \begin{cases} x, & \text{si } x \geq 0 \\ 0, & \text{si } x < 0 \end{cases}$	$(0, \infty)$	
Leaky ReLU	$f(x) = \begin{cases} x, & \text{si } x \geq 0 \\ 0.01x, & \text{si } x < 0 \end{cases}$	$(-\infty, \infty)$	
Softplus	$f(x) = \log(1 + e^{x})$	$(0, \infty)$	

Un conjunto de capas con función de activación lineal equivale a una sola capa lineal donde se multiplican todas las matrices de pesos [37], por lo que funciones no lineales como la sigmoide o

tanh fueron muy utilizadas en el pasado para evitar este problema. Sin embargo, estas funciones introducían el problema de desvanecimiento del gradiente en redes compuestas de un gran número de capas, lo que dificultaba el proceso de entrenamiento de la red. Para evitar este problema, se introdujo la función de unidad lineal rectificada (ReLU, por sus siglas en inglés) y funciones derivadas de ésta, que siguen siendo de las más usadas actualmente en un gran número de arquitecturas de redes neuronales. Existen además otras funciones de activación entrenables, las cuales pueden tener expresiones similares a otras funciones, pero introducen parámetros que son aprendidos durante el entrenamiento de la red con el fin de obtener un mayor desempeño [36].

2.2.3. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) son redes utilizadas en el área de visión computacional para reconocer y clasificar características en las imágenes. La arquitectura de una CNN está influenciada por la organización y funciones de la corteza visual, ya que está diseñada para simular las conexiones entre neuronas del cerebro humano [38]. En la Figura 2.7 se muestran los componentes principales de una CNN, a continuación, se explicará cómo funcionan las capas de convolución, capas de pooling y capas totalmente conectadas, ya que estos tres son los bloques principales en este tipo de redes neuronales.

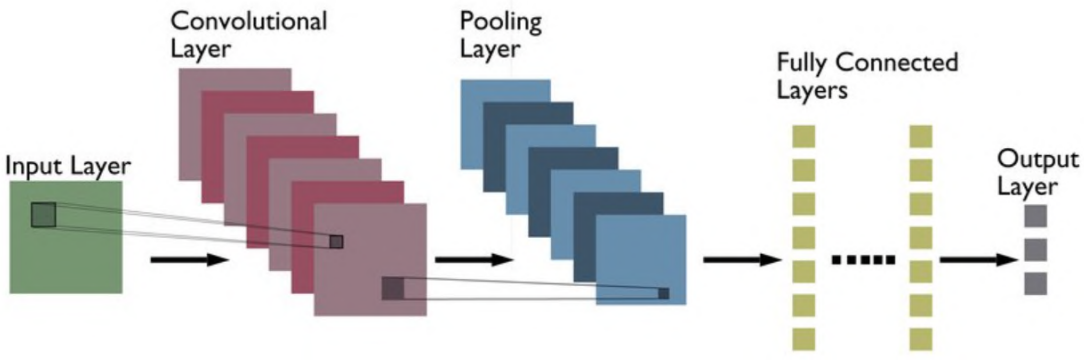


Figura 2.7 Tipos de capas en una arquitectura CNN [39].

Capa de convolución

En las redes convolucionales la entrada será una imagen de tamaño variable W x H x C donde W es el ancho, H la altura y C el número de canales, que en el caso de una imagen RGB son tres. En las redes neuronales mencionadas en la sección anterior se requiere calcular la suma ponderada de cada entrada de la red multiplicada por su peso (que puede verse también como un producto interno), por lo que para una imagen sería necesario utilizar una matriz de pesos distinta para cada tamaño de imágenes, además de requerir una matriz de pesos de gran tamaño. Para resolver estos problemas, las CNN hacen uso de la operación de convolución, en la cual se divide la imagen de entrada en secciones 2D superpuestas y se compara cada sección con un conjunto de pequeñas matrices de pesos, también llamadas filtros o kernels, las cuales suelen reducir el número de parámetros requeridos, ya que suelen ser matrices de 3 x 3 o 5 x 5 [40]. La operación de convolución en 2D puede entenderse a través de la siguiente ecuación:

$$[K \circledast X](i,j) = \sum_{u=0}^{H-1} \sum_{v=0}^{W-1} k_{u,v} x_{i+u,j+v} \tag{2.2}$$

donde el filtro 2D K tiene un tamaño H x W. Por ejemplo, si se realiza la convolución de una imagen de entrada X de 3 x 3 con un kernel K de 2 x 2, la salida Y estará dada a través de la ecuación 2.3, la cual se puede entender también de manera visual en la Figura 2.8.

$$y = \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix} \circledast \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix} = \begin{bmatrix} k_1x_1 + k_2x_2 + k_3x_4 + k_4x_5 & k_1x_2 + k_2x_3 + k_3x_5 + k_4x_6 \\ k_1x_4 + k_2x_5 + k_3x_7 + k_4x_8 & k_1x_5 + k_2x_6 + k_3x_8 + k_4x_9 \end{bmatrix} \tag{2.3}$$

Si se realizara la convolución de una imagen de 5 x 5 con un kernel de 3 x 3, se podría observar que la salida o mapa de características obtenido tendría un tamaño de 3 x 3. Para conservar las dimensiones de la imagen original las CNN emplean el padding, en el cual se agregan valores (usualmente ceros) en los bordes de la imagen con el fin de aumentar su tamaño, como se puede observar en la Figura 2.9, donde la imagen pasa de un tamaño de 5 x 5 a uno de 7 x 7 y al aplicarle un kernel de 3 x 3 se obtiene un mapa de características de 5 x 5, es decir, el mismo tamaño de la

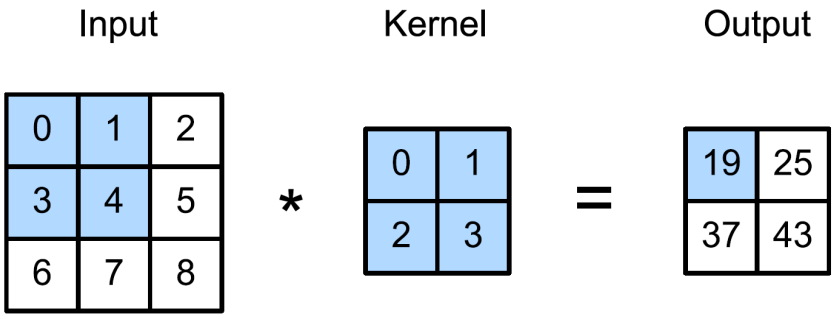


Figura 2.8 Convolución de 3x3 con un kernel de 2x2 [40].

imagen original. Además del padding, se puede aplicar un stride en la operación de convolución, es decir, cambiar cuántos espacios se desliza el kernel después de posicionarse sobre un pixel de la imagen para ir al siguiente, ya que hasta ahora se estaba considerando que se recorre una posición, lo cual corresponde a un stride de 1. Para imágenes cuadradas de dimensión W, kernels cuadrados de dimensión K, utilizando un padding P y stride S, se pueden obtener las dimensiones del mapa de características mediante la siguiente ecuación:

$$W_{out} = \frac{W + 2P - K}{S} + 1 \tag{2.4}$$

Capa de pooling

Luego de obtener los mapas de características, la capa de pooling o agrupamiento es agregada para reducir la dimensionalidad de éstos. Esto ayuda también a extraer las características invariantes a la posición y rotación, además de ayudar a reducir el poder computacional necesario para procesar los datos. Existen dos tipos de pooling, en los cuales se aplica un filtro de tamaño n x n sin pesos a toda la matriz de manera similar a como se hace con la convolución. En el max pooling, se elige el valor máximo que cae dentro del filtro y se coloca en el espacio correspondiente de la matriz de salida, mientras que en el average pooling se promedian todos los valores que caen dentro del filtro y este promedio se coloca en el espacio correspondiente de la matriz de salida [38]. Se puede observar un ejemplo de ambas operaciones de pooling con un stride de 2 en la Figura 2.10.

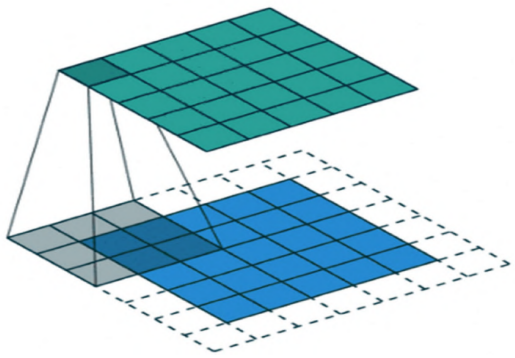


Figura 2.9 Ejemplo de padding en una imagen [38].

Capa totalmente conectada

Después de aplicar varias capas convolucionales y de pooling, el razonamiento de alto nivel de una red neuronal se lleva a cabo mediante las capas totalmente conectadas. Cada neurona en una de estas capas se conecta a todas las salidas de la capa anterior, y su activación puede calcularse a través de una multiplicación de matrices seguida de agregar un sesgo. La capa totalmente conectada eventualmente convierte el mapa de características 2D en un vector de características 1D, el cual puede ser utilizado para realizar una clasificación en múltiples categorías o para realizar un procesamiento adicional [31].

2.2.4. Redes para detección de objetos

Debido al rápido desarrollo de las técnicas de aprendizaje profundo, las redes neuronales convolucionales profundas (DCNNs, por sus siglas en inglés) se han convertido en parte importante de la detección de objetos, ya que las características de las imágenes aprendidas a través de técnicas de aprendizaje profundo son más representativas que las características que se pueden obtener de una extracción manual [41]. De acuerdo a Liu et al. [42], estas redes pueden dividirse en dos categorías:

- De dos etapas o basadas en regiones, en las cuales a partir de una imagen se generan propuestas de regiones sin una categoría definida, se extraen características de estas regiones y

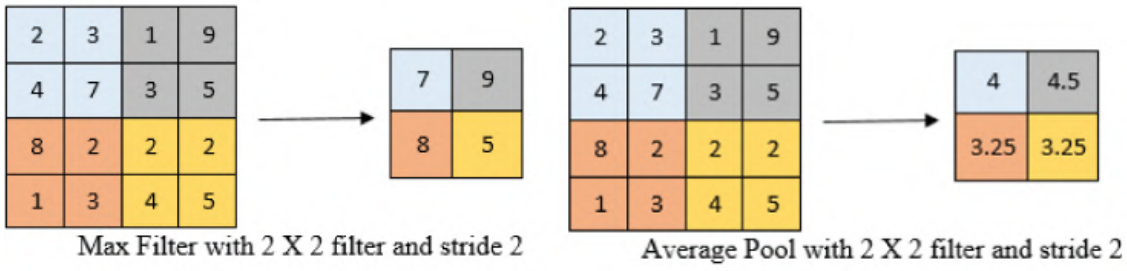


Figura 2.10 Ejemplo de los dos tipos de pooling que pueden aplicarse a un mapa de características [38].

posteriormente se usan clasificadores para determinar la etiqueta de categoría de las regiones. Algunas arquitecturas de este tipo son la RCNN, Fast RCNN, Faster RCNN, RFCN y Mask RCNN.

- De una etapa o unificadas, en las que se predice directamente la probabilidad de una clase y el cuadro delimitador de un objeto utilizando una sola red, sin utilizar una propuesta de regiones. Entre las arquitecturas de este tipo se encuentran OverFeat, SSD y las distintas versiones de YOLO.

Redes YOLO

En las redes YOLO, el problema de detección de objetos es visto como un problema de regresión en lugar de uno de clasificación. Una CNN predice los cuadros delimitadores (bounding boxes) y también las probabilidades de clase para todos los objetos presentes en una imagen. Como el algoritmo identifica los objetos y su posición al mirar la imagen una sola vez, los autores de este modelo lo llamaron YOLO (sólo miras una vez, por sus siglas en inglés) [43].

La primera versión, YOLOv1, fue una red que causó gran impacto debido a que detecta todos los cuadros delimitadores de manera simultánea. Para esto, divide la imagen de entrada en una cuadrícula de $S \times S$ y predice B cuadros delimitadores de la misma clase con un cierto nivel de confianza para C clases distintas en cada elemento de la cuadrícula. Cada predicción de un cuadro delimitador contiene cinco valores: Pc, bx, by, bh, bw, donde Pc es el nivel de confianza para el cuadro, bx y by son los centros del cuadro relativos a la celda de la cuadrícula, y bh y bw son la altura y ancho del cuadro relativos a la imagen. Así, la salida de una red YOLO es un tensor de

$S \times S \times (B \times 5 + C)$ [44]. En la Figura 2.11 se ejemplifica un vector de salida de una red YOLO, considerando una cuadrícula de 3×3, tres clases y una sola clase por celda de la cuadrícula. En este caso, la salida de la red será de $3 \times 3 \times 8$.

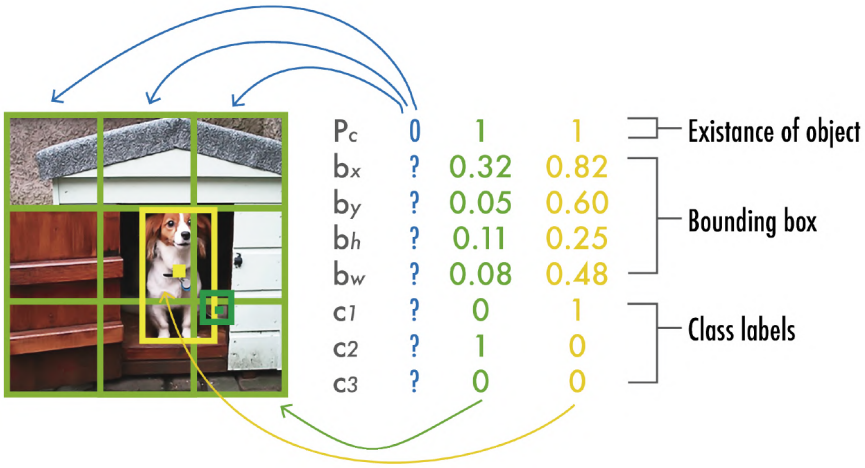


Figura 2.11 Predicción de una red YOLO para una cuadrícula de 3x3, tres clases y una sola clase por elemento de la cuadrícula [44].

Otro concepto fundamental de las redes YOLO es conocido como NMS (supresión no máxima, por sus siglas en inglés), y es un criterio con el cual se descartan los cuadros delimitadores que son menos relevantes para así evitar tener una gran cantidad de cuadros para un mismo objeto. El NMS se vale de la IoU (intersección sobre unión, por sus siglas en inglés), la cual se define para dos cuadros delimitadores B_1 y B_2 de acuerdo a la Ec. 2.5, para eliminar los cuadros menos relevantes. Para realizar esto, primero se selecciona el cuadro delimitador con la máxima puntuación de clase, y luego se eliminan los cuadros que al traslaparse con el cuadro elegido tengan una IoU mayor a un cierto umbral. Este proceso se repite hasta que no queden cuadros delimitadores con puntajes de confianza menores al del cuadro elegido [43]. Una representación visual de cómo funciona la NMS puede verse en la Figura 2.12.

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} \tag{2.5}$$

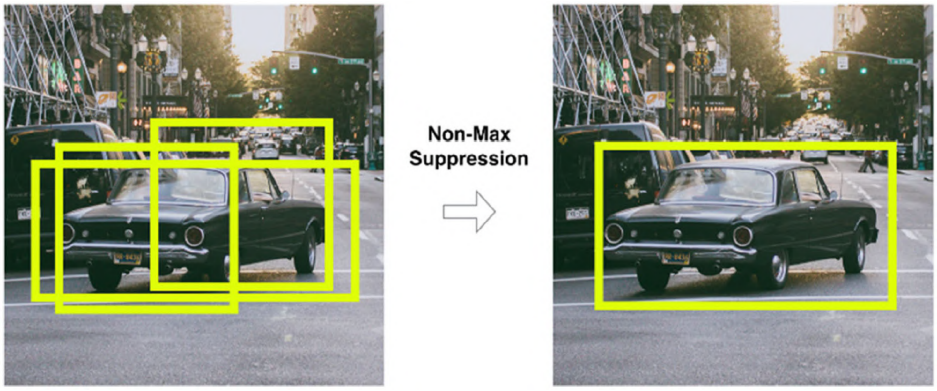


Figura 2.12 Efecto de la Supresión No-Máxima (NMS) en la detección de objetos con YOLO [43].

2.3. Transporte Óptimo

El transporte óptimo busca resolver el problema de transformar una distribución de masa en otra optimizando una función de costo dada [45]. Tuvo sus inicios a partir de la publicación del trabajo Mémoire sur la théorie des déblais et des remblais por el matemático francés Gaspard Monge en 1781. El problema original considerado por Monge puede verse en la Figura 2.13 y consistía en lo siguiente: asumiendo que se tiene cierta cantidad de material que se desea extraer del suelo (déblais) y se quiere transportar a un sitio para realizar una construcción (remblais), se conocen los lugares de donde se extraerá el material y los sitios a los que se transportará, pero debe determinarse a dónde exactamente se mandará el material que se extraiga de cada sitio. Esto es importante ya que, como existe un costo por transportar el material, el objetivo es minimizar el costo total. Más tarde, el matemático ruso Leonid Vitaliyevich Kantorovich redescubrió el problema de Monge en un contexto de economía y probó un teorema de dualidad por medio de análisis funcional, el cual sería útil para llegar a una solución del transporte óptimo. Este problema, llamado de manera más general problema de acoplamiento óptimo (optimal coupling), pasó a llamarse problema de Monge-Kantorovich debido a las contribuciones que ambos realizaron para definirlo y llegar a una solución [46]. A continuación, se describirá la formulación matemática a la que llegó cada uno de ellos y la forma en que se soluciona.

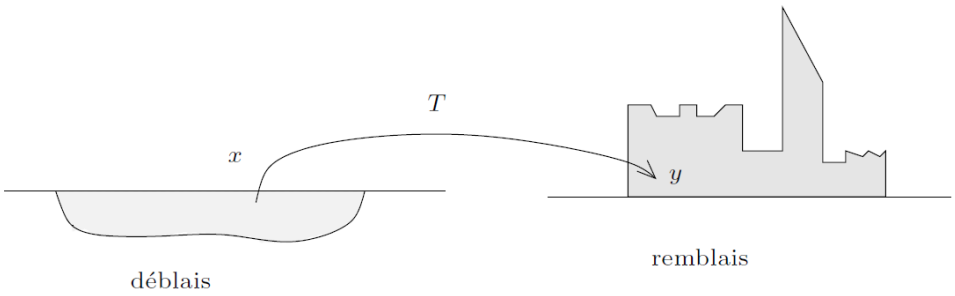


Figura 2.13 El problema de déblais y remblais planteado por Monge [46].

2.3.1. Formulación de Monge

Sea $\Omega \in \mathbb{R}^d$ un espacio medible de entrada de dimensión d, $P(\Omega)$ denota el conjunto de todas las medidas de probabilidad sobre Ω. Si X, Y son un dominio fuente y un dominio objetivo, respectivamente, con elementos x, y, entonces se consideran μ y ν como sus respectivas distribuciones marginales sobre X y Y [47]. También se considera una transformación no lineal del espacio de entrada $T : X \to Y$, la cual transforma la medida μ en su medida imagen denotada por $T_{\#}\mu$, que es otra medida de probabilidad sobre Y, y que satisface la condición:

$$T_{\#}\mu(A) = \mu(T^{-1}(A)), \quad \forall A \subset Y. \tag{2.6}$$

De la ecuación 2.6, T es llamado el mapa de transporte o push-forward de μ a ν si $T_{\#}\mu = \nu$ como se puede ver en la Figura 2.14. Es imposible buscar T en el espacio de todas las transformaciones posibles, por lo que se propuso que T debe ser elegido de tal manera que minimice el costo de transporte $C(T)$ expresado como:

$$C(T) = \int_X c(x, T(x)) d\mu(x), \tag{2.7}$$

donde la función de costo $c : X \times Y \to \mathbb{R}^+$ es una distancia sobre el espacio métrico Ω. $C(T)$ puede interpretarse como la energía requerida para mover una masa de probabilidad $\mu(x)$ desde x hasta $T(x)$ [47].

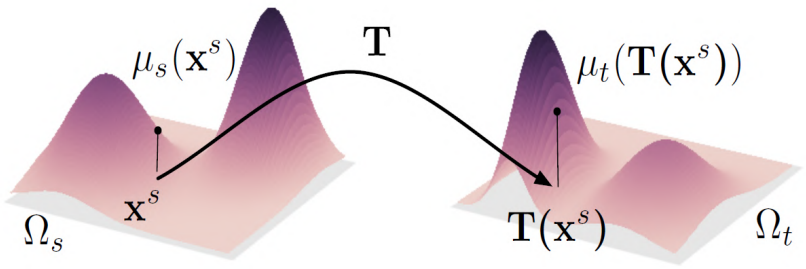


Figura 2.14 El problema de transporte óptimo de Monge sobre dominios 2D, donde Ω_s es el dominio fuente y Ω_t el dominio objetivo [47].

El problema de encontrar este mapa de transporte con el costo mínimo, de acuerdo con la definición de Monge, es la solución del siguiente problema de minimización:

$$\min_T \int_X c(x, T(x)) d\mu(x) \ : \ T_{\#}\mu = \nu. \tag{2.8}$$

Para el caso de medidas discretas de la forma

$$\mu = \sum_{i=1}^{n} \mu_i \delta_{x_i}, \quad \nu = \sum_{j=1}^{m} \nu_j \delta_{y_j}, \tag{2.9}$$

el problema de Monge busca asociar un solo punto y_j a cada punto x_i a través de un mapa de transporte, de manera que este lleve la masa de μ hacia la masa de ν. El mapa $T : \{x_1, ..., x_n\} \rightarrow \{y_1, ..., y_m\}$ debe verificar que

$$\forall j \in [m], \quad \nu_j = \sum_{i:T(x_i)=y_j} \mu_i, \tag{2.10}$$

lo cual es equivalente a la notación compacta $T_{\#}\mu = \nu$ [48]. Este mapa T debe minimizar un costo de transporte parametrizado por una función $c(x, y)$ y definido para puntos $(x, y) \in X \times Y$,

$$\min_T \left\{ \sum_i c(x_i, T(x_i)) : T_{\#}\mu = \nu \right\}. \tag{2.11}$$

En su formulación original, Monge consideró la distancia euclidiana $c(x, T(x)) = |x - T(x)|$ como la función de costo. Para ciertas medidas, la formulación de Monge posee un planteamiento erróneo en el sentido de que no existe mapa de transporte que convierta una función de densidad de

probabilidad (PDF, por sus siglas en inglés) en otra [45]. Por ejemplo, esto puede ocurrir si μ es una delta de Dirac y ν no lo es. Una manera de superar este obstáculo se dio gracias a la formulación de Kantorovich, la cual introduce una relajación al problema original de Monge e involucra encontrar un plan de transporte óptimo en lugar de un mapa de transporte [49].

2.3.2. Formulación de Kantorovich

La formulación del transporte óptimo de Kantorovich es una relajación convexa del problema de Monge, en la que se define $\Pi \in P(X \times Y)$ como el conjunto de todos los acoplamientos probabilísticos con marginales μ y ν [47]. Aquí se busca un plan de transporte $\pi \in \Pi$ entre X y Y que describa cuánta masa está siendo transportada entre distintas coordenadas, es decir, $\pi(A, B)$ nos dice qué tanta masa de un conjunto A es movida a un conjunto B [45]. Esta formulación del problema de transporte óptimo puede describirse entonces mediante la ecuación:

$$\inf_{\pi \in \Pi} \int_{X \times Y} c(x, y) d\pi(x, y), \tag{2.12}$$

donde se tienen las siguientes ecuaciones equivalentes para las marginales:

$$\begin{aligned} \pi(x, Y) = \mu(x), \quad A \subset X, \quad \pi(A, Y) = \mu(A), \\ \pi(X, y) = \nu(y), \quad B \subset Y, \quad \pi(X, B) = \nu(B). \end{aligned} \tag{2.13}$$

La medida de probabilidad π sobre $X \times Y$ es una manera distinta de describir el desplazamiento de las partículas de μ, ya que, en lugar de dar un destino $T(x)$ para cada partícula x, ahora se da el número de partículas que van de x a y para cada par (x, y). Con esta descripción, para un solo punto x, las partículas de este pueden moverse hacia diferentes destinos y. En el caso de que se tengan múltiples destinos, el movimiento no puede ser descrito a través de un mapa de transporte T, por lo que el plan de transporte π ofrece una descripción más adecuada [50].

El problema de Kantorovich puede definirse también para un caso discreto, en el que se consideran PDFs de la forma $\mu = \sum_{i=1}^{M} p_i \delta(x - x_i)$ y $\nu = \sum_{j=1}^{N} q_j \delta(y - y_j)$, donde $\delta(x)$ es la función delta de Dirac. Para estas PDFs no existe un mapa de transporte que lleve la masa de μ hacia ν, por lo que es necesaria la división de masas que proporciona la formulación de Kantorovich. Así, esta formulación puede ser escrita como

$$K(\mu,\nu) = \min_{\pi} \sum_{i} \sum_{j} c(x_i, y_j)\pi_{ij}$$
$$s.t. \sum_{j} \pi_{ij} = p_i, \sum_{i} \pi_{ij} = q_j \tag{2.14}$$
$$\pi_{ij} \geq 0,\ i = 1, ..., M,\ j = 1, ..., N,$$

donde π_{ij} define cuánta masa de la partícula m_i en x_i necesita moverse hacia y_j. La optimización descrita en la ecuación 2.14 tiene una función objetivo lineal y restricciones lineales, por lo que se trata de un problema de programación lineal. Este planteamiento resulta de gran utilidad, pues en la práctica una medida no discreta puede aproximarse a través de una medida discreta [45].

2.3.3. El problema dual de Kantorovich

La formulación de Kantorovich da pie a un problema convexo de optimización lineal con restricciones, por lo que se puede hacer uso de la teoría de la dualidad, la cual es un recurso frecuentemente utilizado en problemas convexos. Consiste en encontrar un problema dual mediante un intercambio de inf-sup y explotar las relaciones entre este problema dual y el primal [50]. Así, el problema dual de Kantorovich se define mediante:

$$\sup \left\{ \int_Y \phi(y) d\nu(y) - \int_X \psi(x) d\mu(x); \quad \phi(y) - \psi(x) \leq c(x,y) \right\}. \tag{2.15}$$

Además, existe una relación entre el problema dual, en la cual se establece que este no puede ser mayor que el problema primal, es decir, al costo de transporte óptimo original:

$$\sup_{\phi - \psi \leq c} \left\{ \int_Y \phi(y) d\nu(y) - \int_X \psi(x) d\mu(x) \right\} \leq \inf_{\pi \in \Pi(\mu,\nu)} \left\{ \int_{X \times Y} c(x,y) d\pi(x,y) \right\}. \tag{2.16}$$

Si podemos encontrar un par (ψ, ϕ) y un plan de transporte π para los cuales se cumpla la igualdad en la ecuación anterior, entonces ambos serán óptimos. El tipo de pares de funciones (ψ, ϕ) que resultan relevantes en el problema dual de Kantorovich son aquellos del tipo:

$$\phi(y) = \inf_{x}(\psi(x) + c(x,y)), \quad \psi(x) = \sup_{y}(\phi(y) - c(x,y)), \tag{2.17}$$

ya que permiten definir una de las funciones en términos de la otra, de modo que sólo exista una incógnita en el problema. Sin embargo, para que el par de funciones (ψ, ϕ) cumpla con ambas

expresiones de la ecuación 2.17, es necesario que ψ sea una función c-convexa y ϕ sea c-cóncava. De esta manera, la transformada-c de estas funciones se define como:

$$\begin{aligned} \forall y \in Y \quad \psi^c(y) &= \inf_{x \in X} (\psi(x) + c(x,y)), \\ \forall x \in X \quad \phi^c(x) &= \sup_{y \in Y} (\phi(y) - c(x,y)). \end{aligned} \tag{2.18}$$

Cuando las funciones cumplen con estas condiciones, la dualidad de Kantorovich puede ser expresada de la siguiente manera:

$$\begin{aligned} \min_{\pi \in \Pi(\mu,\nu)} \int_{X \times Y} c(x,y) d\pi(x,y) &= \max_{(\psi,\phi);\, \phi - \psi \leq c} \int_Y \phi(y) d\nu(y) - \int_X \psi(x) d\mu(x) \\ &= \max_{\psi} \int_Y \psi^c(y) d\nu(y) - \int_X \psi(x) d\mu(x). \end{aligned} \tag{2.19}$$

Como consecuencia de esta dualidad, dado un plan de transporte π, si se logra encontrar un par de funciones (ψ, ϕ) tales que $\phi(y) - \psi(x) = c(x,y)$ a través del soporte en π, entonces el plan π es óptimo [46].

2.3.4. La distancia de Wasserstein

La ecuación 2.12 establece el costo del transporte óptimo entre dos medidas. Se puede pensar en esta expresión como un tipo de distancia entre μ y ν, pero no satisface de manera estricta los axiomas de una función de distancia [46]. Sin embargo, a partir de ella, se puede llegar a una función de distancia que se define de la siguiente manera: sea $p \in [1, \infty)$, para cualesquiera dos medidas de probabilidad μ, ν en un espacio Ω, la distancia de Wasserstein de orden p entre μ y ν se define por la fórmula

$$W_p(\mu,\nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\Omega \times \Omega} |x - y|^p d\pi(x,y) \right)^{1/p}. \tag{2.20}$$

Para cualquier $p \geq 1$, W_p es una métrica en el conjunto de densidades de probabilidad $P(\Omega)$. Para el caso de $p = 1$, la métrica p-Wasserstein es llamada *distancia de Kantorovich-Rubinstein* o *Earth mover's distance*. El espacio métrico $(P(\Omega), W_p)$ definido a partir de esta distancia se conoce como el espacio p-Wasserstein [45]. De manera formal, este espacio puede definirse también como

$$P_p(\Omega) := \left\{ \mu \in P(\Omega); \quad \int_\Omega d(x_0, x)^p \mu(dx) < +\infty \right\}, \tag{2.21}$$

donde $x_0 \in \Omega$ es arbitrario, por lo que el espacio no depende de la elección de este punto. Así, W_p define una distancia finita en $P_p(\Omega)$. En otras palabras, el espacio de Wasserstein es un espacio de medidas de probabilidad que tienen un momento finito de orden p [46].

El hecho de que esta distancia, utilizada en transporte óptimo, convierta una métrica básica entre bins en una métrica entre histogramas de estos bins, la convierte en un método efectivo para comparar histogramas en aplicaciones de visión computacional y aprendizaje automático. De una manera similar, la distancia de Wasserstein también puede ser utilizada sobre algunos espacios de características para desempeñar un análisis de señales e imágenes [48].

2.3.5. Regularización entrópica

La formulación de Kantorovich del transporte óptimo puede resolverse aproximando su solución a través de varios métodos. Uno de estos es la adición de una penalización por regularización entrópica al problema original, la cual tiene importantes ventajas, pues la minimización de este problema regularizado puede resolverse utilizando un simple esquema alternativo de minimización, el cual se traduce en iteraciones que son productos entre matrices y vectores, volviéndolo adecuado para ejecutarse en una GPU y permitiendo acelerar el cálculo de la distancia de transporte óptimo [48]. De manera general, la regularización entrópica para valores positivos de la temperatura $\varepsilon > 0$ se define como

$$OT_\varepsilon(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} \int_{X \times Y} c(x, y) d\pi + \varepsilon KL(\pi | \mu \otimes \nu), \tag{2.22}$$

donde la penalización entrópica

$$KL(\pi | \mu \otimes \nu) = \int_{X \times Y} \log \left(\frac{d\pi}{d\mu d\nu} \right) d\pi \tag{2.23}$$

se refiere a la divergencia de Kullback-Leibler [51]. El problema entrópico OT_ε es también llamado el *problema de Schrödinger*, ya que fue introducido originalmente por Erwin Schrödinger en una memoria que discutía las interpretaciones físicas de la mecánica cuántica [52].

El algoritmo de Sinkhorn está estrechamente relacionado con el problema entrópico, ya que es el método que se suele utilizar para resolverlo. Existen diversas maneras de interpretar las iteraciones de este algoritmo, pero la mayoría de los investigadores lo hacen a través de la introducción de variables auxiliares, como el kernel de Gibbs k_ε y las funciones de escala u y v, las cuales se

definen de la siguiente manera:

$$k_\varepsilon(x,y) = e^{-C(x,y)/\varepsilon}, \quad u(x) = e^{f(x)/\varepsilon}, \quad v(y) = e^{g(y)/\varepsilon} \tag{2.24}$$

y están codificadas con una matriz kernel y vectores de escala positivos:

$$K_{i,j} = e^{-C(x_i,y_j)/\varepsilon} \in \mathbb{R}^{N\times M}_{>0}, \quad U_i = e^{f_i/\varepsilon} \in \mathbb{R}^{N}_{>0}, \quad V_j = e^{g_j/\varepsilon} \in \mathbb{R}^{M}_{>0}. \tag{2.25}$$

En este sistema de coordenadas exponenciales, las iteraciones de Sinkhorn son representadas mediante las siguientes ecuaciones:

$$\begin{aligned} u \leftarrow \frac{1}{k \star (v\nu)} \qquad & y \qquad v \leftarrow \frac{1}{k \star (u\mu)}, \\ i.e. \quad U_i \leftarrow \frac{1}{K(V\nu)} \qquad & y \qquad V_j \leftarrow \frac{1}{K^\top(U\mu)}. \end{aligned} \tag{2.26}$$

Esto es equivalente a hacer que se cumplan de manera alternada las restricciones de las marginales:

$$(\pi 1)_i = \sum_{j=1}^{M} \pi_{i,j} = \mu_i \qquad y \qquad (\pi^\top 1)_j = \sum_{i=1}^{N} \pi_{i,j} = \nu_j, \tag{2.27}$$

en el plan de transporte

$$\pi_{i,j} = \exp \frac{1}{\varepsilon}[f_i + g_j - C(x_i,y_j)] \cdot \mu_i\nu_j = \mu_i U_i K_{i,j} V_j \nu_j, \tag{2.28}$$

codificado de manera implícita por los vectores de escala (U_i) y (V_j). Este algoritmo fue llamado así por Richard Sinkhorn, quien fue el primero en demostrar la convergencia del procedimiento de normalización descrito en la ecuación 2.26 para matrices arbitrarias positivas K [52].

2.3.6. Divergencia de Sinkhorn

A partir del problema entrópico OT_ε, se puede construir el siguiente costo al cual se le conoce como divergencia de Sinkhorn:

$$S_\varepsilon(\mu,\nu) = OT_\varepsilon(\mu,\nu) - \frac{1}{2}OT_\varepsilon(\mu,\mu) - \frac{1}{2}OT_\varepsilon(\nu,\nu). \tag{2.29}$$

Esta ecuación satisface $S_\varepsilon(\nu,\nu) = 0$ y permite interpolar entre el transporte óptimo y la discrepancia media máxima (MMD, por sus siglas en inglés), la cual es más fácil de calcular, ya que permite escalar a lotes más grandes con una baja complejidad de muestreo.

$$OT_0(\mu,\nu) \xleftarrow{0\leftarrow\varepsilon} S_\varepsilon(\mu,\nu) \xrightarrow{\varepsilon\to+\infty} \frac{1}{2}\|\mu-\nu\|^2_{-C}\,. \tag{2.30}$$

Los términos de autocorrelación $OT_\varepsilon(\mu,\mu)$ y $OT_\varepsilon(\nu,\nu)$ aparecen debido a que, para valores positivos de ε, $OT_\varepsilon(\nu,\nu) \neq 0$, de manera que minimizar $OT_\varepsilon(\mu,\nu)$ con respecto a μ resulta en una solución sesgada, como se puede ver en la Figura 2.15, donde el gradiente de $OT_\varepsilon(\mu,\nu)$ lleva a μ hacia una medida reducida cuyo soporte es más pequeño que el de la medida objetivo ν. La ecuación 2.29 se introdujo para arreglar este sesgo entrópico presente en el costo OT_ε, pues al tener una estructura similar a la de una norma de un kernel cuadrático, S_ε conlleva una función de pérdida definida positiva, la cual es apropiada para aplicaciones en aprendizaje automático [51].

(a) $\mathrm{L} = \mathrm{OT}_\varepsilon$ (b) $\mathrm{L} = \mathrm{S}_\varepsilon$

Figura 2.15 Solución μ (en rojo) del problema de ajuste $\min_\mu L(\mu,\nu)$ para una medida ν mostrada en azul [51].

La regularización entrópica y la divergencia de Sinkhorn pueden extenderse al transporte óptimo no balanceado, el cual consiste en relajar las restricciones $(\pi_1 = \mu, \pi_2 = \nu)$, las cuales imponen una conservación de masa. Así, en el caso no balanceado se tiene entonces que $(\pi_1,\pi_2) \neq (\mu,\nu)$, por lo que se aumenta la robustez del plan de transporte óptimo ante valores atípicos [53]. Si $\varepsilon, \rho > 0$ son dos parámetros de regularización, el costo no balanceado se puede escribir como:

$$OT_{\varepsilon,\rho}(\mu,\nu) = \min_{\pi\in\Pi(\mu,\nu)} \int_{X\times Y} c(x,y)d\pi + \varepsilon KL(\pi|\mu\otimes\nu) + \rho KL(\pi_1,\mu) + \rho KL(\pi_2,\nu). \quad (2.31)$$

Este problema está bien definido incluso si μ y ν no tienen la misma masa total, y coincide con el problema de Schrödinger OT_ε cuando $\rho = +\infty$ [52]. También puede definirse una divergencia de Sinkhorn no balanceada mediante la ecuación:

$$S_{\varepsilon,\rho}(\mu,\nu) = OT_{\varepsilon,\rho}(\mu,\nu) - \frac{1}{2}OT_{\varepsilon,\rho}(\mu,\mu) - \frac{1}{2}OT_{\varepsilon,\rho}(\nu,\nu) + \frac{\varepsilon}{2}(\langle\mu,1\rangle - \langle\nu,1\rangle)^2. \quad (2.32)$$

2.4. Principales estudios relacionados

A continuación, se presentan algunos trabajos relacionados con esta investigación, donde se realiza la detección de espigas de trigo utilizando redes neuronales y en los cuales se utiliza también la base de datos GWHD (Global Wheat Head Detection). Se analiza la arquitectura de redes neuronales utilizada y las técnicas de adaptación de dominio empleadas para los casos en que se aplicó esta metodología a las imágenes.

El trabajo de Wang et al. [29] propone un modelo de detección de objetos basado en la arquitectura EfficientDet-D0 que se enfoca en resolver los problemas de oclusión de las espigas de trigo presentes en las imágenes de la base de datos GWHD. Para esto, proponen una metodología de tres pasos que consiste en lo siguiente:

- Emplear transferencia de aprendizaje para pre-entrenar la red troncal del modelo, permitiéndole extraer las características semánticas de las espigas de trigo.
- Utilizar la técnica de recorte aleatorio para aumento de imágenes, en la cual se borran rectángulos de la imagen de acuerdo al número y tamaño de las espigas de trigo para simular oclusión en imágenes reales de espigas.
- Introducir un módulo de atención de bloque convolucional (CBAM, por sus siglas en inglés) luego de la red troncal de la EfficientDet-D0, con lo cual el modelo presta mayor atención a las espigas y elimina información irrelevante del fondo.

Con esta metodología lograron obtener una precisión de conteo de 94 %, mejorando en un 2 % la precisión original del modelo EfficientDet-D0, además de lograr una tasa de detección falsa del 5.8 %.

Zhang et al. [54] diseñaron un modelo de detección de espigas de trigo basado en la arquitectura YOLOv5, en el cual agregaron algunas modificaciones, como un módulo de atención para mejorar la extracción de características de las espigas y un módulo de fusión de características en la red troncal, además de mejorar la función de pérdida de la red introduciendo términos específicos para los módulos de reconocimiento de las espigas de trigo y de la imagen de fondo. Con el modelo que

propusieron lograron obtener un mAP de 0.688, con lo cual lograron un mejor desempeño que el de otros modelos de detección de objetos como YOLO, EfficientDet, Mask-RCNN o SSD.

Liu et al. [16], por su parte, proponen un modelo para modificar los colores de las imágenes al cual llaman transformación de color dinámica (DCT, por sus siglas en inglés), y mencionan que puede ser integrado a cualquier detector de objetos, por lo que integran este modelo a la arquitectura Scaled-YOLOv4. El modelo DCT corresponde a una transformación lineal de color que se implementa como una red dinámica y permite corregir variaciones en la iluminación de manera adaptativa. Con este modelo, logran obtener una precisión promedio de dominio (ADA) de 0.821, lo cual les otorgó un lugar entre los ganadores del Global Wheat Challenge (GWC) 2021.

En la investigación realizada por Hartley y French [19] se propone utilizar adaptación de dominio por medio de redes adversarias generativas para transformar imágenes sintéticas en imágenes de apariencia realista de acuerdo a los dominios en la base de datos GWHD. Para esto, primero se generan modelos 3D de escenas que contienen espigas de trigo por medio de Blender, para luego utilizar el modelo CycleGAN para realizar la adaptación de dominio de las imágenes sintéticas. Además, se utiliza regresión de mapas de calor como una red adicional y algoritmos de agrupamiento para separar la base de datos original en 4 dominios y adaptar las imágenes sintéticas a cada uno de ellos. Con esta propuesta, implementada en conjunto con la red Detectron2, se obtiene una intersección sobre unión media (Mean IoU) de 0.8642 y una distancia euclidiana media de 10.5617.

Shen et al. [55] proponen un modelo ligero basado en YOLOv5s para detectar y contar espigas de trigo, al cual llaman S-YOLOv5s. Este modelo tiene por objetivo ser desplegado en sistemas embebidos y dispositivos móviles, por lo que el trabajo se centra en la optimización del modelo YOLOv5s a través de algunas modificaciones en la arquitectura: se reemplaza la red troncal CSPDarknet por ShuffleNetV2 para reducir el tamaño del modelo; se introduce el operador de sobremuestreo CARAFE para reemplazar el operador de la PANet, con lo cual se mejora la extracción de características y la resolución espacial de los mapas de características; y, finalmente, se introduce el cabezal de detección dinámica de objetivo DyHead que permite adaptarse a la diversidad de tamaños, formas y orientaciones de las espigas de trigo. Con estas modificaciones, logran

reducir el tamaño del modelo en 11.6 MB manteniendo un buen desempeño, pues el mAP50 se reduce en sólo 1.3 %.

En la investigación realizada por Okafor et al. [15], se propone utilizar adaptación de dominio de Fourier (FDA), corrección adaptativa alfa-beta-gamma (AABG) y filtro guiado aleatorio (RGF) como técnicas de preprocesamiento en las imágenes de epigas de trigo. Con la FDA se reduce la variación entre los distintos dominios transformando la imagen al dominio de Fourier, para posteriormente alinear su distribución con la de una imagen de otro dominio; la AABG, por su parte, permite ajustar las propiedades de la imagen basadas en estadísticas locales de los parches de imagen; y el RFG permite un filtrado de la imagen que toma en cuenta los bordes. Al probar distintas combinaciones de estas tres técnicas con el modelo EfficientDet-D5, lograron mejorar el mAP del modelo en 2.42 % en las imágenes de la base de datos GWHD, utilizando una combinación de FDA y RFG.

2.5. Comparación entre los trabajos relacionados y la propuesta de investigación

En la Tabla 2.2 se presenta una comparación de los trabajos relacionados mencionados anteriormente, así como de esta investigación. Se pueden observar de manera resumida las redes de detección de objetos utilizadas en cada estudio, si utilizaron o no alguna técnica de adaptación de dominio y el uso de alguna otra técnica relevante, además de las métricas evaluadas en cada una de las investigaciones.

Tabla 2.2: Comparación de trabajos relacionados.

Estudio	**Autor**	**Modelo de detección de objetos**	**Adaptación de dominio**	**Otras técnicas utilizadas**	**Métricas evaluadas**
Oclussion Robust Wheat Ear Counting Algorithm Based on Deep Learning	Wang et al. (2021) [29]	EfficientDet-D0	No fue utilizada	Transferencia de aprendizaje, recorte aleatorio, CBAM	Tasa de precisión de conteo, tasa de detección falsa

Estudio	**Autor**	**Modelo de detección de objetos**	**Adaptación de dominio**	**Otras técnicas utilizadas**	**Métricas evaluadas**
High-Precision Wheat Head Detection Model Based on One-Stage Network and GAN Model	Zhang et al. (2022) [54]	YOLOv5	No fue utilizada	Módulo de atención, módulo de fusión de características, modificación de la función de pérdida	Precisión promedio media (mAP)
Dynamic Color Transform Networks for Wheat Head Detection	Liu et al. (2022) [16]	Scaled-YOLOv4	No fue utilizada	Transferencia de color dinámica (DCT)	Precisión promedio de dominio (ADA)
Domain Adaptation of Synthetic Images for Wheat Head Detection	Hartley & French (2021) [19]	Detectron2	CycleGAN	Generación de datos sintéticos	Mean IoU, distancia euclidiana media
A lightweight network for improving wheat ears detection and counting based on YOLOv5s	Shen et al. (2023) [55]	YOLOv5s	No fue utilizada	Reemplazo de la red troncal por ShuffleNetV2, uso de operador CARAFE, DyHead	mAP, coeficiente de determinación (R^2)
Enhanced Wheat Head Detection in Images Using Fourier Domain Adaptation and Random Guided Filter	Okafor et al. (2024) [15]	EfficientDet-D5	Adaptación de dominio de Fourier	Corrección adaptativa alfa beta gamma (AABG), filtro guiado aleatorio (RGF)	mAP
Esta propuesta	Salas Ibañez	YOLOv5s	Transporte óptimo	Transferencia de aprendizaje	Precisión, sensibilidad, mAP

2.6. Modelo o esquema general de investigación

El presente trabajo seguirá el enfoque de una investigación cuantitativa experimental de acuerdo a lo planteado por Hernández Sampieri [56] en su interpretación del método científico. Además,

su alcance es de tipo correlacional, ya que se busca encontrar una relación de aumento en la precisión de detección de espigas en un modelo de redes neuronales al utilizar un algoritmo de transporte óptimo para realizar una transferencia de color en las imágenes. La investigación es cuantitativa, ya que se busca medir la precisión alcanzada por el modelo de redes neuronales como variable dependiente, mientras que es experimental porque se plantearán experimentos donde se probarán distintos dominios objetivo y parámetros del algoritmo de transporte óptimo, y se evaluará la precisión de detección de espigas en cada uno de estos escenarios, utilizando la misma arquitectura en la red neuronal. Para diseñar los experimentos, una vez elegida una determinada arquitectura de redes neuronales, se evaluará su precisión en la tarea de detección de espigas de trigo. Luego, se utilizará la misma red neuronal para realizar diversos modelos, en donde en cada uno se realizará una transferencia de color en las imágenes, variando los parámetros del algoritmo de transporte óptimo y utilizando un dominio objetivo distinto. Por cada modelo se obtendrá la precisión obtenida en la detección de espigas, para así determinar el modelo que se desempeñó mejor y, por lo tanto, el dominio objetivo y parámetros del algoritmo de transporte óptimo más adecuados.

Capítulo 3

Método y propuesta de investigación

En este capítulo se presenta la metodología utilizada para llevar a cabo la investigación. Comienza por definir brevemente el proceso realizado y, posteriormente, se detallan cada uno de los pasos: la adquisición de las imágenes, su preprocesamiento, la arquitectura del modelo de redes neuronales, el algoritmo de adaptación de dominio con transporte óptimo, el planteamiento de los dos casos experimentales y, finalmente, una descripción de las métricas de validación empleadas.

3.1. Modelo de investigación

La metodología utilizada en esta investigación consiste principalmente en 5 etapas, las cuales se pueden observar en la Figura 3.1 y, además, se puede hallar bajo cada una de ellas una breve descripción.

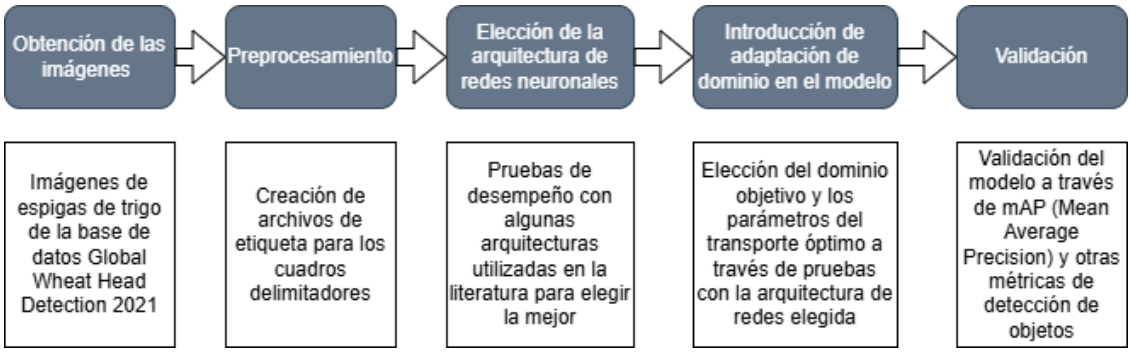


Figura 3.1 Esquema general de los pasos seguidos durante la investigación.

La base de datos utilizada para esta investigación es Global Wheat Head Detection (GWHD) 2021 [14], la cual consta de 6515 imágenes RGB de 1024x1024 píxeles donde se encuentran 275187 espigas de trigo, ya que las imágenes están etiquetadas con las posiciones donde se encuentran los cuadros delimitadores de cada una de las espigas. Las imágenes provienen de 16 instituciones

distribuidas en 12 países y representan imágenes de distintas etapas de desarrollo del trigo, por lo que estas variaciones aportan una gran cantidad de casos que deben ser considerados por el modelo.

3.2. Obtención de las imágenes

Las imágenes utilizadas en este trabajo corresponden a las de la base de datos GWHD 2021, las cuales fueron adquiridas con una distancia de muestreo del suelo de entre 0.2 y 0.4 mm. La distancia de muestreo del suelo (GSD, por sus siglas en inglés) se define como la distancia entre los centros de dos píxeles adyacentes medidos en el suelo. Se relaciona con la distancia focal de la cámara, su resolución y la distancia de la cámara al objeto fotografiado. Usualmente se describe en términos de centímetros por píxel (cm/px) [57].
La base de datos de imágenes está compuesta de 47 sub-datasets, donde cada uno de estos contiene un conjunto de imágenes adquiridas en el mismo lugar, con el mismo sensor y desde la misma plataforma, a los cuales también se les ha definido como dominios [14]. La Tabla 3.1 describe los lugares de donde fueron obtenidos cada uno de los 47 dominios, así como la plataforma en que se montó la cámara y el número de dominios por país.

3.3. Preprocesamiento

El preprocesamiento de la base de datos se basó en estructurar los directorios de las imágenes y etiquetas, convertir las etiquetas a un formato distinto y crear un archivo de configuración YAML para poder entrenar un modelo YOLO mediante el paquete de Python 'Ultralytics' [58]. La estructura de los directorios se hizo como sigue: en la carpeta principal del proyecto se tienen tres carpetas para cada uno de los conjuntos de imágenes (entrenamiento, validación y prueba) y dentro de cada una de estas carpetas hay dos carpetas más, donde una contiene las imágenes y la otra contiene los archivos de etiquetas correspondientes a esas imágenes. Los conjuntos de imágenes para entrenamiento, validación y prueba que se utilizaron son los definidos por los autores de la base de datos GWHD 2021, donde se tienen 3657 imágenes para entrenamiento, 1476 para validación y 1382 para pruebas.

Tabla 3.1 Descripción del origen de las imágenes de la base de datos GWHD 2021 [14].

País	**Ubicación**	**Institución**	**Número de dominios**	**Plataforma de obtención**	**Número de imágenes**
Suiza	Usak	ETHZ	1	Spidercam	747
Reino Unido	Rothamsted	Rothamsted	1	Pórtico	432
Bélgica	Gembloux	ULiège/Gembloux	1	Carro	30
Noruega	NMBU	NMBU	2	Carro	180
Francia	Gréoux, Villiers le Bacle, Villers-Saint-Christophe, Mons, Toulouse	Arvalis, INRAe	13	Manual	2268
Canadá	Saskatchewan	USaskatchewan	1	Tractor	200
Estados Unidos	KSU, Maricopa, Arizona	Kansas State University, TERRA-REF project	6	Tractor, pórtico	605
México	Ciudad Obregón	CIMMYT	3	Carro	206
Japón	NARO-Tsukuba, NARO-Hokkaido, Kyoto	UTokyo, UKyoto	4	Carro, manual	1174
China	Baima	NAU	3	Carro, manual	220
Australia	Gatton, McAllister, Brookstead	UQueensland	11	Tractor, manual	423
Sudán	Wad Medani	ARC	1	Manual	30

La base de datos tiene sus etiquetas distribuidas en archivos .csv, donde en la primera columna se encuentran los nombres de las imágenes y en la segunda columna la etiqueta de los cuadros delimitadores en formato *x_min*, *y_min*, *x_max*, *y_max*, separando por un punto y coma cuando se encuentra más de un objeto en la imagen y escribiendo *no_box* cuando no hay espigas en la imagen. A diferencia de esto, los modelos YOLO requieren un archivo .txt de etiquetas por cada imagen, el cual debe tener el mismo nombre de la imagen y tener la etiqueta de un cuadro delimitador en formato *clase*, *x_centro*, *y_centro*, *ancho*, *altura* y además normalizados de 0 a 1. Para manejar

varios objetos en una imagen, en este formato se debe escribir la etiqueta de un objeto por línea, y en el caso de no tener objetos en una imagen, no es necesario crear un archivo de etiquetas para esa imagen. En la Tabla 3.2 se puede ver un ejemplo de etiquetas en ambos formatos.

Tabla 3.2 Comparación entre el formato de las etiquetas de la base de datos GWHD 2021 y el formato usado por los modelos YOLO.

Etiqueta original		**Etiqueta en formato YOLO**	
imagen1.png	896 911 977 955	imagen1.txt	0 0.914 0.911 0.079 0.043
imagen2.png	481 820 604 922; 655 957 732 1024	imagen2.txt	0 0.530 0.851 0.120 0.099 0 0.677 0.967 0.075 0.065
imagen3.png	no_box	No hay archivo de etiqueta	

En la Figura 3.2 se pueden observar cuadros delimitadores para tres objetos distintos. Las etiquetas proporcionadas en la base de datos GWHD 2021 corresponden a las coordenadas de la esquina superior izquierda y la esquina inferior derecha del cuadro, mientras que las requeridas por los modelos YOLO son las coordenadas del centro del cuadro, su altura y ancho normalizados de 0 a 1. Para realizar esta conversión entre etiquetas se utilizaron las siguientes ecuaciones:

$$\begin{aligned} x_{center} &= \frac{x_{min} + x_{max}}{2W}, \qquad y_{center} = \frac{y_{min} + y_{max}}{2H}, \\ w &= \frac{x_{max} - x_{min}}{W}, \qquad h = \frac{y_{max} - y_{min}}{H} \end{aligned} \tag{3.1}$$

donde H es la altura total y W es el ancho total de la imagen, y su uso es para poder realizar la normalización de los valores entre 0 y 1.

Como último paso del preprocesamiento, se creó un archivo YAML, el cual es un formato utilizado comúnmente para escribir archivos de configuración y representa datos estructurados con una sintaxis basada en sangrías, pares clave-valor y convenciones intuitivas [59]. En el caso de los modelos YOLO, estos archivos son necesarios para indicar al modelo los datos a utilizar en distintas etapas, por lo que el archivo contiene las rutas a las carpetas donde se encuentran las imágenes para el entrenamiento, la validación y las pruebas, así como un diccionario de los nombres de las clases que detectará el modelo y sus índices correspondientes, empezando en 0.

Figura 3.2 Ejemplo de etiquetas de un cuadro delimitador en el formato usado por los modelos YOLO [58].

3.4. Arquitectura YOLOv5

El modelo YOLOv5 fue lanzado en 2020 por Glen Jocher, fundador y CEO de Ultralytics, y se desarrolló en el framework de Pytorch, en lugar de Darknet, que era el framework utilizado en versiones previas de YOLO. Esta arquitectura está dividida en tres partes: tronco, cuello y cabeza, que representan partes distintas del proceso de detección de objetos, desde la extracción de características, el refinado de éstas y la obtención de predicciones basadas en las características obtenidas anteriormente. Un diagrama detallado de la arquitectura de esta red puede verse en la Figura 3.3.

El tronco de la red se compone de una CSPDarknet53 que empieza con una Stem, es decir, una capa convolucional escalonada con una ventana amplia para reducir los costos computacionales y de memoria, seguida de capas convolucionales que extraen las características relevantes de la imagen de entrada. La capa SPPF y las siguientes capas de convolución procesan las características a distintas escalas, mientras que las capas upsample incrementan la resolución del mapa de características. Cada convolución dentro de la red es seguida por una normalización por lote (batch

normalization) y una activación con la función SiLU. El cuello utiliza SPPF y una CSP-PAN modificada, mientras que la cabeza de la red sigue la tendencia de YOLOv3 al tener múltiples salidas, con predicciones multiescala hechas para distintos tamaños de malla [44].

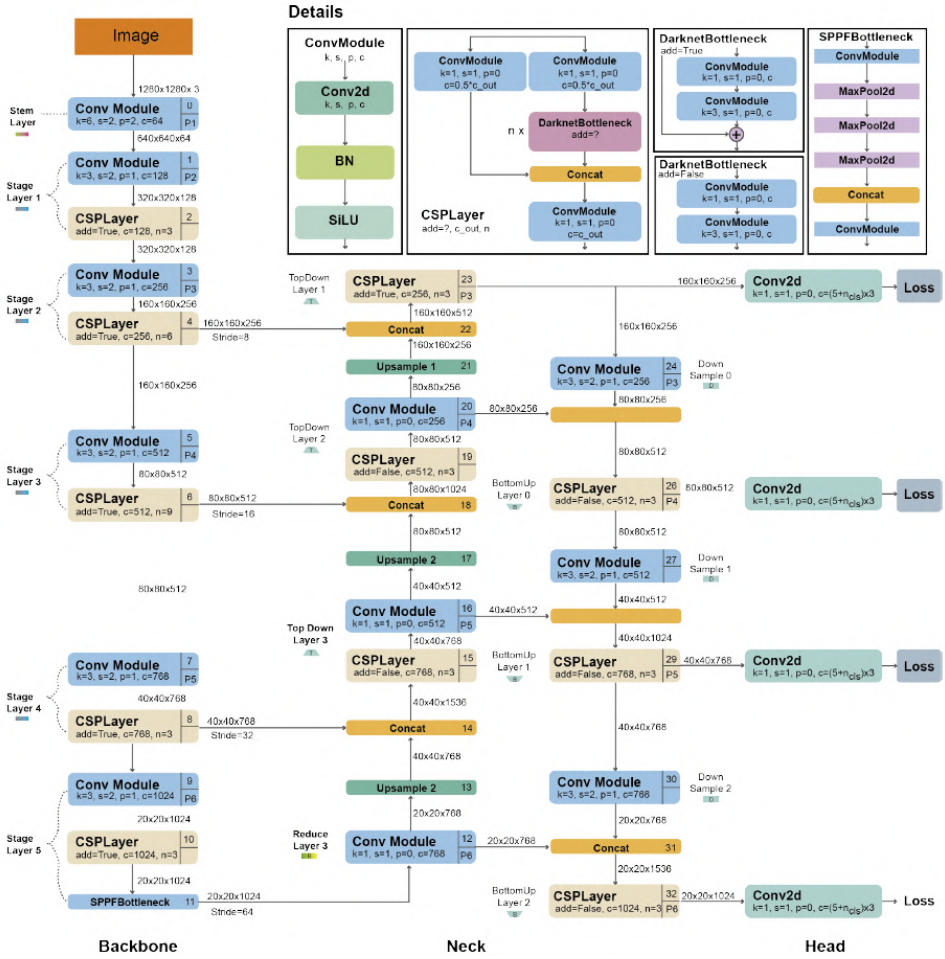


Figura 3.3 Arquitectura de la red YOLOv5 [44].

YOLOv5 posee cinco versiones distintas de acuerdo a su tamaño: YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large) y YOLOv5x (extra large). Estas se diferencian principalmente en el número de parámetros para ajustarse a distintas configuraciones de hardware. El modelo utilizado durante esta investigación fue el YOLOv5s, el cual posee 9.1 millones de parámetros. El objetivo de utilizar este tamaño de modelo es reducir el tiempo de entrenamiento,

ya que para cada prueba con distintos parámetros de transporte óptimo es necesario entrenar el modelo YOLO nuevamente.

En esta investigación fue utilizada la versión más reciente del modelo antes mencionado, la cual es YOLOv5su v7.0 de Ultralytics. Esta versión integra el enfoque sin anchor-boxes introducido en los modelos YOLOv8 [58]. En cuanto a los parámetros de entrenamiento, se eligió la misma configuración para todos los experimentos realizados, la cual consiste en un tamaño de imagen de 640×640, 100 épocas de entrenamiento, un parámetro de paciencia de 10 donde si el modelo no mejora en ese número de épocas se termine prematuramente el entrenamiento, una semilla aleatoria 0, y los demás parámetros se dejaron establecidos a sus valores por defecto. Finalmente, se eligió inicializar el modelo con pesos de un modelo preentrenado, lo cual se explicará más a detalle en la siguiente sección.

3.5. Aprendizaje por transferencia

Un modelo de machine learning generalmente está diseñado para resolver tareas específicas, para lo cual requiere entrenarse desde cero con una gran cantidad de datos. El aprendizaje por transferencia es un método mediante el cual se transfiere el aprendizaje adquirido en una tarea para poder resolver otra generalmente muy similar, por lo que sirve como un punto de partida para reducir el entrenamiento requerido o incrementar la precisión del modelo.

Recientemente se han creado grandes bases de datos para aplicaciones en visión computacional como clasificación, detección de objetos y segmentación, las cuales contienen desde cientos de miles hasta millones de imágenes etiquetadas. Entre estas destacan ImageNet, Microsoft COCO y Google Open Images, que han servido para realizar aprendizaje por transferencia debido a que muchos de los frameworks más usados en machine learning proveen modelos preentrenados con estas bases de datos.

Dentro del aprendizaje profundo, algunas redes neuronales han alcanzado un desempeño inigualable en diferentes tareas, por lo que sus autores suelen compartir el código fuente e incluso los modelos preentrenados para descargarse libremente, ya que, como estos modelos se han desempeñado bien en tareas específicas, pueden ser utilizados como base para tareas relacionadas. Las

arquitecturas CNN suelen componerse de una secuencia de capas convolucionales y de pooling que funcionan como un extractor de características, y se conectan a capas finales responsables de hacer una tarea de regresión o clasificación. Uno de los enfoques del aprendizaje por transferencia utiliza un modelo preentrenado reemplazando estas últimas capas y ajustando el modelo para la tarea deseada. Otro enfoque congela las primeras capas y utiliza parte del modelo original como un extractor de características para las últimas capas en la tarea objetivo [60].

En esta investigación se utilizó el modelo YOLOv5su preentrenado en la base de datos Microsoft COCO y proporcionado por la librería de Ultralytics. Esta base de datos es con la que se entrenan la mayoría de modelos de detección de objetos, ya que contiene fotos de 91 tipos de objetos comunes en su contexto natural, con un total de 2.5 millones de objetos etiquetados en 328 mil imágenes [61]. El uso de este modelo preentrenado permitió reducir el tiempo de entrenamiento al alcanzar un buen desempeño en la detección de espigas en pocas épocas, ya que se inicializa la red neuronal con los pesos del modelo preentrenado.

3.6. Algoritmo de transferencia de color

Para realizar un cambio de colores en las imágenes de espigas de trigo, abordado desde el punto de vista de adaptación de dominio, primero se deben establecer un dominio de origen y un dominio objetivo. Se tomaron como dominios a los ya establecidos por la base de datos GWHD 2021, por lo que se asumió que en cada uno de estos existe una distribución de colores similar en todas las imágenes que los conforman. Así, esta adaptación de dominio se abordó como una 'transferencia de color' de las imágenes del dominio objetivo a las imágenes del dominio de origen, con el fin de que estas últimas se parezcan a las primeras. Este proceso se realizó mediante un algoritmo de transporte óptimo con la librería de GeomLoss para Python [51], mediante el cual se modifican los colores de una imagen para que se parezcan a los de otra.

De acuerdo a lo planteado en el capítulo anterior, el transporte óptimo busca transformar de manera eficiente una distribución de probabilidad en otra distinta. Con esto en mente, para poder llevar a

cabo el transporte óptimo entre dos imágenes, estas deben representarse primero como distribuciones de probabilidad. Feydy et al. [51] describen que esto puede hacerse mediante la definición de dos medidas de probabilidad discretas α y β en el espacio RGB $[0,1]^3$:

$$\alpha = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_i}, \qquad \beta = \frac{1}{M}\sum_{j=1}^{M}\delta_{y_j} \tag{3.2}$$

donde $x \in (\mathbb{R}^3)^N$ y $y \in (\mathbb{R}^3)^M$ son dos tensores con valores reales y de dimensiones $(N,3)$ y $(M,3)$ respectivamente, que representan la información de cada uno de los tres canales de color, pero donde la matriz de cada canal es aplanada para convertirse en un vector. Estas representaciones de imágenes pueden visualizarse como nubes de puntos como en la Figura 3.4, donde se muestra una imagen de origen a la izquierda, a la cual se le cambiará la paleta de colores mediante el algoritmo de transporte óptimo para que termine pareciéndose a los colores de la imagen objetivo de la derecha.

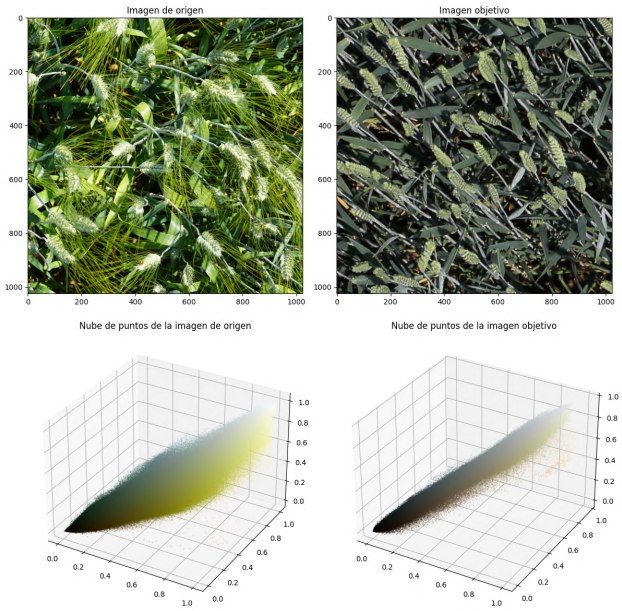


Figura 3.4 Representación de las imágenes como nubes de puntos en el espacio RGB.

El transporte óptimo no se calcula de manera exacta, sino que se aproxima mediante la divergencia de Sinkhorn $S_{\varepsilon,\rho}$. La función de costo se define a través de la distancia de Wasserstein con $p = 2$, y se encuentra el gradiente de la divergencia de Sinkhorn:

$$v_i = \frac{1}{\alpha_i} \nabla_{x_i} S_{\varepsilon,\rho}(\alpha, \beta) \tag{3.3}$$

el cual define un mapeo cuya suavidad y alcance máximos se pueden definir como dos parámetros: desenfoque $b = \sqrt{\varepsilon}$ y alcance $r = \sqrt{\rho}$ respectivamente. Los parámetros $(\varepsilon, \rho) = (0, +\infty)$ definen un transporte óptimo perfecto, pero entre más cercanos a estos valores se elijan los parámetros, más tiempo le llevará al algoritmo converger.

3.7. Introducción de transporte óptimo en el modelo de redes

Para poder medir el efecto del transporte óptimo en el modelo YOLO mencionado anteriormente, se propuso realizar una comparación entre el desempeño del modelo entrenándolo con las imágenes sin alterar y su desempeño realizando una adaptación de dominio con el algoritmo de la sección anterior a todas las imágenes de la base de datos y entrenándolo con estas imágenes modificadas.

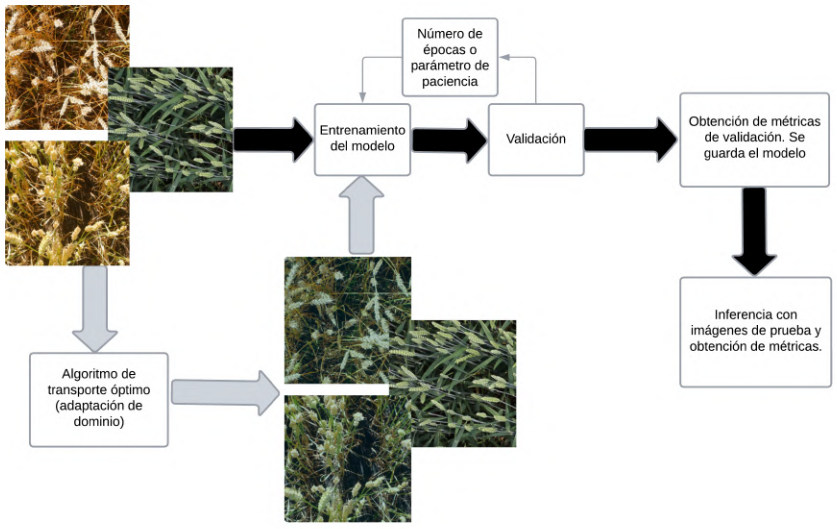


Figura 3.5 Proceso de entrenamiento de la red para las imágenes originales y adaptadas.

El proceso seguido para el entrenamiento del modelo YOLOv5s, así como la inferencia realizada con este modelo y la obtención de métricas se describe mediante la Figura 3.5, en la cual se sigue

un proceso para primero entrenar la red con las imágenes originales de la base de datos, y posteriormente se realiza un proceso similar en que se incluye primero la adaptación de las imágenes con transporte óptimo para realizar un entrenamiento similar de la red con estas imágenes modificadas. Este primer proceso con las imágenes originales, seguido por las flechas negras en la Figura 3.5, consta de los siguientes pasos:

- Introducción de las 3657 imágenes de entrenamiento al modelo YOLOv5s, con los parámetros iniciales mencionados en la sección de la arquitectura. Al igual que en la mayoría de procesos de entrenamiento de redes neuronales, el entrenamiento y validación se realizan de manera conjunta, y este proceso se detiene una vez que se llega al número de épocas establecido o se cumple el parámetro de paciencia.

- Validación con las 1476 imágenes designadas para esta tarea. Esta validación comprueba lo aprendido por el modelo durante el entrenamiento y se realiza en cada época.

- Obtención de dos archivos de guardado para el modelo, con los pesos de este para la última época y la época de mejor desempeño en los datos de validación.

- Inferencia con las 1382 imágenes de prueba y obtención de las métricas de desempeño del modelo. En esta parte se carga el archivo de mejor desempeño obtenido en el paso anterior, y con este se realiza la inferencia en las imágenes de prueba (que el modelo no ha visto hasta este paso) para aproximar el desempeño del modelo en un entorno real.

Por su parte, el proceso de entrenamiento con las imágenes modificadas, seguido en la Figura 3.5 por las flechas grises y posteriormente las negras, consta de los siguientes pasos:

- Elección del dominio objetivo. Se propone que, para reducir el número de imágenes a las cuales se les aplicará la transferencia de colores, se elija como dominio objetivo a alguno de los dominios con mayor número de imágenes.

- Introducción de todas las imágenes de la base de datos en el algoritmo de transporte óptimo, excepto las del dominio objetivo. Como el algoritmo realiza la transferencia de color entre dos imágenes, se itera por todas las imágenes a modificar, y como imagen objetivo se elige

aleatoriamente una de las pertenecientes al dominio objetivo. Para asegurar la reproducibilidad de este experimento, se establece una semilla 0 al inicio de este proceso para asegurar que siempre se elijan las mismas imágenes objetivo.

- Obtención de una versión de la base de datos en que todas las imágenes tienen la misma paleta de colores, la cual se define por los colores de las imágenes pertenecientes al dominio objetivo.
- Repetición de los pasos seguidos en el primer proceso, pero esta vez con las imágenes modificadas en lugar de las originales.

Tabla 3.3 Especificaciones del equipo de cómputo utilizado.

MSI Pulse GL66 12UGKV	
Sistema operativo	Ubuntu 22.04.3 (WSL2)
Procesador	Intel Core i7-12700H 2.70 GHz
RAM	32 GB DDR4
Tarjeta gráfica	NVIDIA GeForce RTX 3070

Para el primer caso de estudio, el procedimiento descrito en el primer proceso se llevó a cabo solamente una vez, ya que esto da como resultado las métricas del modelo con las imágenes originales, lo cual sirve de punto de comparación para determinar el efecto del transporte óptimo en las imágenes. Por su parte, el procedimiento descrito en el segundo proceso se repitió un total de 8 veces, ya que se eligieron dos dominios objetivo distintos y se escogieron 4 combinaciones distintas de los parámetros de desenfoque y alcance para el algoritmo de transporte óptimo. Todos los experimentos fueron llevados a cabo utilizando el mismo hardware, cuyas características se detallan en la Tabla 3.3. Se realizó también un segundo caso de estudio, el cual se describe en la siguiente sección.

3.8. Análisis de cambio de balance en los dominios

Con el fin de determinar la influencia que tiene el número de imágenes del dominio objetivo en el éxito o fracaso de éste para aumentar el desempeño de la red YOLO, además de averiguar si el transporte óptimo funciona mejor o peor con una menor cantidad de imágenes de entrenamiento, se planteó una serie de experimentos en los cuales se eliminó un determinado número de imágenes por dominio en el conjunto de entrenamiento y se fue variando el dominio objetivo utilizado. Este segundo caso de estudio fue planteado luego de obtener los resultados del primero, por lo que se toman en cuenta algunos de los hallazgos realizados en esos experimentos. Así, durante estos experimentos no se variaron los parámetros del algoritmo de transporte óptimo, sino que se dejó la combinación de estos que mejor funcionó durante el caso de estudio anterior.

Para poder plantear esta reducción del número de imágenes en determinados dominios, primero fue necesario saber el número de imágenes por dominio del conjunto de entrenamiento. Así, la Tabla 3.4 muestra este número de imágenes por dominio, así como algunas características importantes de cada dominio, como lo son el país de procedencia de las imágenes y la etapa fenológica en que se encuentran las espigas de trigo presentes en las imágenes. Cabe destacar que la razón por la que se eligió realizar esta disminución de imágenes tan solo en el conjunto de entrenamiento es debido a que no hay dominios con imágenes presentes en los tres conjuntos de la base de datos (entrenamiento, validación y prueba), por lo que manipular los dominios es más fácil tan solo en el conjunto de entrenamiento, que es en el que el modelo aprende a identificar las espigas. Además, dejar inalterados los otros dos conjuntos presenta una forma de poder comparar mejor los resultados de estos experimentos con los obtenidos en el caso de estudio anterior al evaluarse los modelos sobre los mismos conjuntos de imágenes.

Como otra evaluación sobre los dominios, también se pretende averiguar si un dominio objetivo 'funciona bien' (es decir, aumenta el desempeño del modelo) debido a su número de imágenes o a las características de éstas, las cuales vienen dadas en gran medida por la etapa fenológica del trigo. En la Figura 3.6 se presenta una distribución de todos los dominios de la base de datos de acuerdo a la etapa fenológica de las espigas en las imágenes, en la cual también se incluyen imágenes que ayudan a visualizar las características de color de las hojas y espigas en cada una de estas etapas.

Tabla 3.4 Descripción de los dominios que conforman el conjunto de entrenamiento de la base de datos.

Nombre del dominio	**País**	**Etapa de las espigas**	**Número de imágenes**
ETHZ_1	Suiza	Llenado	747
Arvalis_3	Francia	Llenado-maduración	588
Arvalis_5	Francia	Llenado	448
Rres_1	Reino Unido	Llenado-maduración	432
Arvalis_2	Francia	Llenado	401
Arvalis_4	Francia	Llenado	204
Inrae_1	Francia	Llenado-maduración	176
Arvalis_6	Francia	Llenado-maduración	160
NMBU_2	Noruega	Maduración	98
NMBU_1	Noruega	Llenado	82
Arvalis_1	Francia	Post-floración	66
Arvalis_11	Francia	Llenado	60
Arvalis_10	Francia	Llenado	60
Arvalis_9	Francia	Maduración	32
ULiège-GxABT_1	Bélgica	Maduración	30
Arvalis_12	Francia	Llenado	29
Arvalis_7	Francia	Llenado-maduración	24
Arvalis_8	Francia	Llenado-maduración	20

Para llevar a cabo los experimentos, se eliminó un número fijo de imágenes de ciertos dominios específicos para alcanzar una distribución diferente y se consideraron tres casos: que el dominio con mayor número de imágenes sea otro, que los dominios con más imágenes tengan un número igual de estas y que haya una mayor diferencia en el número de imágenes del dominio de mayor número y todos los demás dominios. Luego, se entrenó el modelo YOLOv5s con el conjunto de entrenamiento reducido por la eliminación de imágenes para tener un punto de referencia en cada experimento. Posteriormente, se realizó el transporte óptimo de todas las imágenes con distintos dominios objetivo y con los parámetros $b = 0.6$ y $r = \infty$, tomando una imagen objetivo al

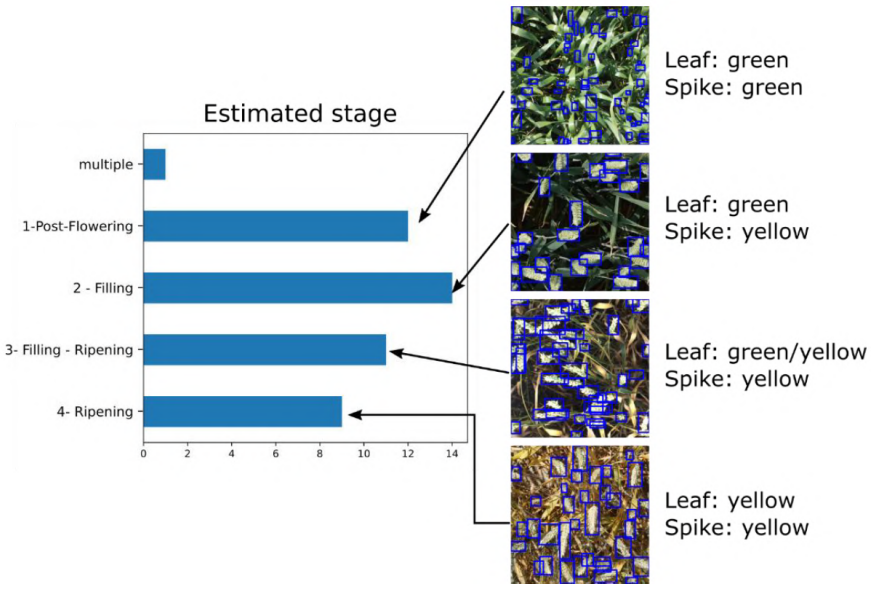


Figura 3.6 Distribución de los dominios en la base de datos de acuerdo a la etapa de desarrollo de las espigas [14].

azar dentro del dominio elegido, como se explicó en la sección anterior. Finalmente, para cada dominio objetivo, se entrenó el modelo YOLO y se obtuvieron las métricas correspondientes para compararlas con las obtenidas sin el transporte óptimo.

3.9. Métricas de validación del modelo

Para poder evaluar el desempeño del modelo YOLO con y sin el transporte óptimo, se eligieron algunas métricas comunes utilizadas por la comunidad científica en las tareas de detección de objetos. La más usada de estas corresponde al mAP o precisión promedio media, la cual se define a través de la precisión y la sensibilidad o recall. Antes de definir estas métricas, es importante mencionar también algunos conceptos clave para todas ellas, así como sus siglas en inglés:

- Verdadero Positivo (TP): Una detección correcta de un bounding box de referencia.
- Falso positivo (FP): Una detección incorrecta de un objeto no existente o una detección mal localizada de un objeto existente.
- Falso negativo (FN): Un bounding box de referencia no detectado.

Cabe destacar que en el contexto de la detección de objetos no se consideran los verdaderos negativos (TN), ya que hay un número infinito de bounding boxes que no deben ser detectados dentro de cualquier imagen.

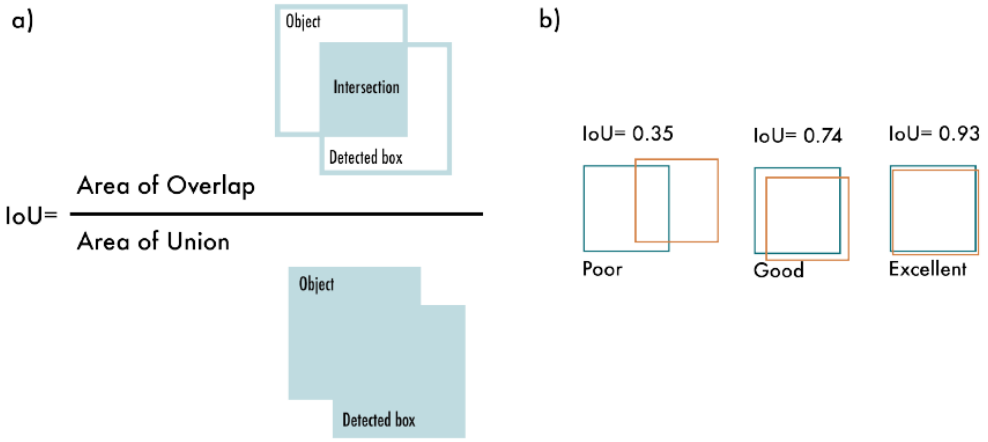


Figura 3.7 Definición de la intersección sobre unión y ejemplos de umbrales [44].

Para que las definiciones anteriores tengan sentido, es necesario establecer qué es una detección correcta y una incorrecta. Esto se hace a través del IoU, definido a través de la Ec. 2.5 y también a través de la Figura 3.7, ya que mide el área de empalme entre el bounding box predicho y el de referencia, dividido entre el área de unión entre ellos. El IoU se puede comparar con un umbral dado u, con el cual se clasifica si la detección es correcta o incorrecta. Si $IoU \geq u$ la detección se considera como correcta, mientras que si $IoU < u$ la detección es incorrecta [62].

3.9.1. Precisión y Sensibilidad

De las métricas definidas a través de las detecciones positivas y negativas, tanto falsas como verdaderas, la precisión y la sensibilidad o recall son las dos más utilizadas en la detección de objetos, pues a través de ellas se define el mAP que es la técnica estándar con la cual se evalúa el desempeño de estos modelos. La precisión P se define como la proporción de predicciones positivas hechas correctamente del total de predicciones, mientras que la sensibilidad R mide la proporción de verdaderos positivos que se predijeron de entre todos los bounding boxes de referencia [63]. Ambos valores van de 0 a 1 y se pueden describir matemáticamente a través de las

siguientes ecuaciones:

$$P = \frac{TP}{TP + FP} \tag{3.4}$$

$$R = \frac{TP}{TP + FN} \tag{3.5}$$

Con estas dos métricas puede construirse una curva de precisión-sensibilidad que se puede ver como un compromiso entre los valores de precisión y sensibilidad para distintos valores de confianza asociados con los bounding boxes generados por el detector. Por lo tanto, un buen detector de objetos será aquél que logre obtener una precisión y un recall altos [62]. Anteriormente, el desempeño de un detector se medía con el área bajo la curva (AUC, por sus siglas en inglés) de la curva precisión-sensibilidad, pero debido a su comportamiento en zigzag, se ha optado por aproximar esto de otras maneras, dando paso a definiciones de la precisión promedio (AP) y, por consiguiente, el mAP.

3.9.2. mAP50 y mAP50-95

Para calcular el mAP se utilizan principalmente dos enfoques definidos por las dos bases de datos más importantes a lo largo de los años en el estudio de la detección de objetos: PASCAL VOC y Microsoft COCO. En la base de datos VOC se utiliza una interpolación de 11 puntos para calcular el AP, en la cual se interpolan los valores de precisión con un IoU de 0.5 en 11 valores equidistantes de sensibilidad (0, 0.1, 0.2, ... , 1) de acuerdo a la ecuación:

$$AP = \frac{1}{11} \sum_{R \in \{0, 0.1, ..., 0.9, 1\}} P_{interp}(R), \tag{3.6}$$

donde

$$P_{interp}(R) = \max_{\tilde{R}:\tilde{R} \geq R} P(\tilde{R}) \tag{3.7}$$

Así, en lugar de utilizar los valores de precisión observados en cada valor de sensibilidad, el AP se calcula considerando la precisión interpolada, que es la precisión máxima correspondiente al valor de sensibilidad mayor que el valor actual. Esto se puede entender a través de la Figura 3.8, donde también se define a la precisión interpolada como el máximo valor de precisión a la derecha [63].

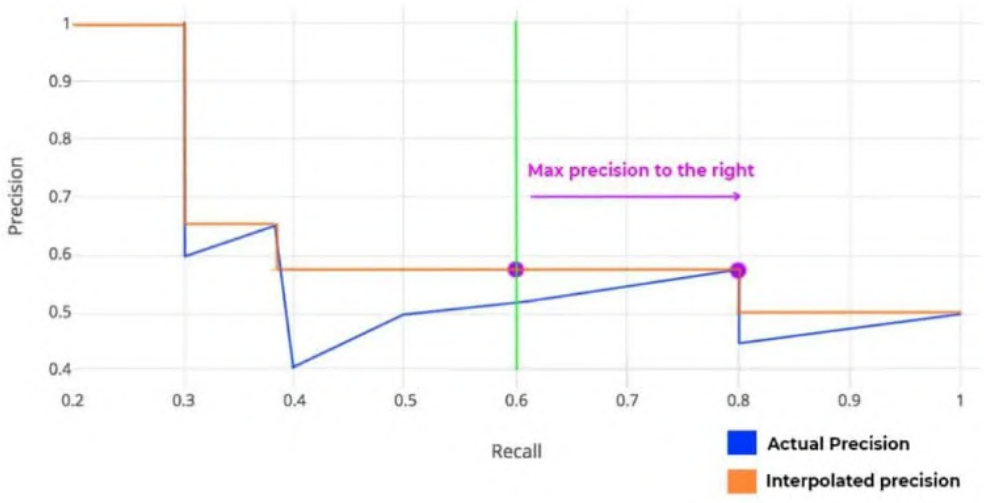


Figura 3.8 Ejemplo de la interpolación de valores de precisión [63].

El mAP es la métrica que se utiliza para medir el desempeño de un modelo de detección de objetos en todas las clases, por lo que se calcula simplemente promediando el AP sobre todas las clases:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i, \tag{3.8}$$

donde AP_i es el AP de la i-ésima clase y N es el número total de clases evaluadas [62].

En la base de datos COCO se utiliza una interpolación de 101 puntos en lugar de la de 11 puntos usada en VOC, de modo que se calcula la precisión para 101 valores de sensibilidad de 0 a 1 y con incrementos de 0.01, siguiendo un proceso similar al descrito anteriormente. Además, el mAP de COCO (también conocido como mAP50-95) se obtiene promediando sobre distintos valores de IoU en lugar de sólo uno, excepto por el mAP50 que es el mAP para un sólo umbral de IoU de 0.5 [44]. Para calcular el mAP de acuerdo a COCO, deben seguirse los siguientes pasos:

- Para cada clase evaluada, obtener la curva de precisión-sensibilidad variando el nivel de confianza de las predicciones del modelo.
- Calcular el AP de cada clase utilizando la interpolación de 101 puntos.

- Calcular el AP para diferentes umbrales de IoU, que van desde 0.5 a 0.95 con incrementos de 0.05. Esto es, para un mayor umbral de IoU se requiere una predicción más acertada para que sea considerada un verdadero positivo.
- Para cada umbral del IoU, tomar la media de los AP en todas las clases para calcular el mAP.
- Calcular el mAP total o mAP50-95 promediando los valores de mAP calculados para cada umbral del IoU.

Finalmente, cabe destacar que en esta investigación se consideran equivalentes el mAP y AP, ya que la detección de espigas considera una sola clase de objeto. Además, a pesar de que se reportan tanto los valores de mAP50 como de mAP50-95, el valor más importante a considerar en todas las evaluaciones es el mAP50 debido a que se presenta un gran número de espigas en las imágenes y estas son muy pequeñas, por lo que interesa más detectarlas de manera aproximada para permitir un conteo que realizar una detección muy acertada de cada una de estas como se haría al considerar umbrales de IoU mayores a 0.5. Así, las métricas que se reportan para evaluar los modelos son Precision, Sensibilidad, mAP50 y mAP50-95, reportando esto tanto para las imágenes de validación como para las de prueba, con un mayor énfasis en estas últimas al ser lo que nos da una mayor idea de cómo se desempeñaría el modelo con imágenes nuevas.

Capítulo 4

Resultados y limitaciones

En este capítulo se presentan los resultados obtenidos a lo largo de esta investigación, así como su interpretación, importancia, alcance y limitaciones. Primero, se aborda una comparativa visual entre los algoritmos POT y GeomLoss; luego se presentan los resultados del desempeño de la red YOLOv5s con el conjunto de imágenes original y cómo se ve afectado por el transporte óptimo, para finalmente mostrar un análisis de cómo un desbalance en la base de datos genera cambios en el desempeño de la red y lo que esto implica.

4.1. Resultados visuales del transporte óptimo

Como una primera prueba para visualizar las capacidades del algoritmo elegido para la transferencia de color mediante transporte óptimo, se compara el aspecto de algunas imágenes al aplicarles este algoritmo perteneciente al paquete GeomLoss con algunas combinaciones de parámetros distintas, y al aplicarles otros algoritmos pertenecientes al paquete de Python Optimal Transport (POT) [64], que es el paquete de transporte óptimo más completo y más utilizado en la literatura, al contener una gran cantidad de métodos que permiten resolver problemas de este tipo.
El objetivo de esta comparación es mostrar ejemplos de los resultados visuales obtenidos por ambos algoritmos y mostrar que GeomLoss representa una buena alternativa a POT, ya que este último realiza un cálculo exacto de la solución al problema de transporte óptimo, mientras que GeomLoss realiza una aproximación y está optimizado para utilizarse en GPU, lo que le da una ventaja en cuanto al tiempo de ejecución. Además, GeomLoss hace uso del paquete KeOps [65] para resolver problemas de transporte óptimo con millones de muestras en segundos, lo que lo vuelve útil para trabajar con imágenes de mayor resolución y representa la principal razón de elegirlo para este estudio, pues puede trabajar con la imagen completa sin requerir muestreo y realiza el transporte

óptimo en poco más de un segundo, por lo que, al usarse en todas las imágenes de la base de datos, no toma demasiado tiempo realizar la transferencia de color entre estas.

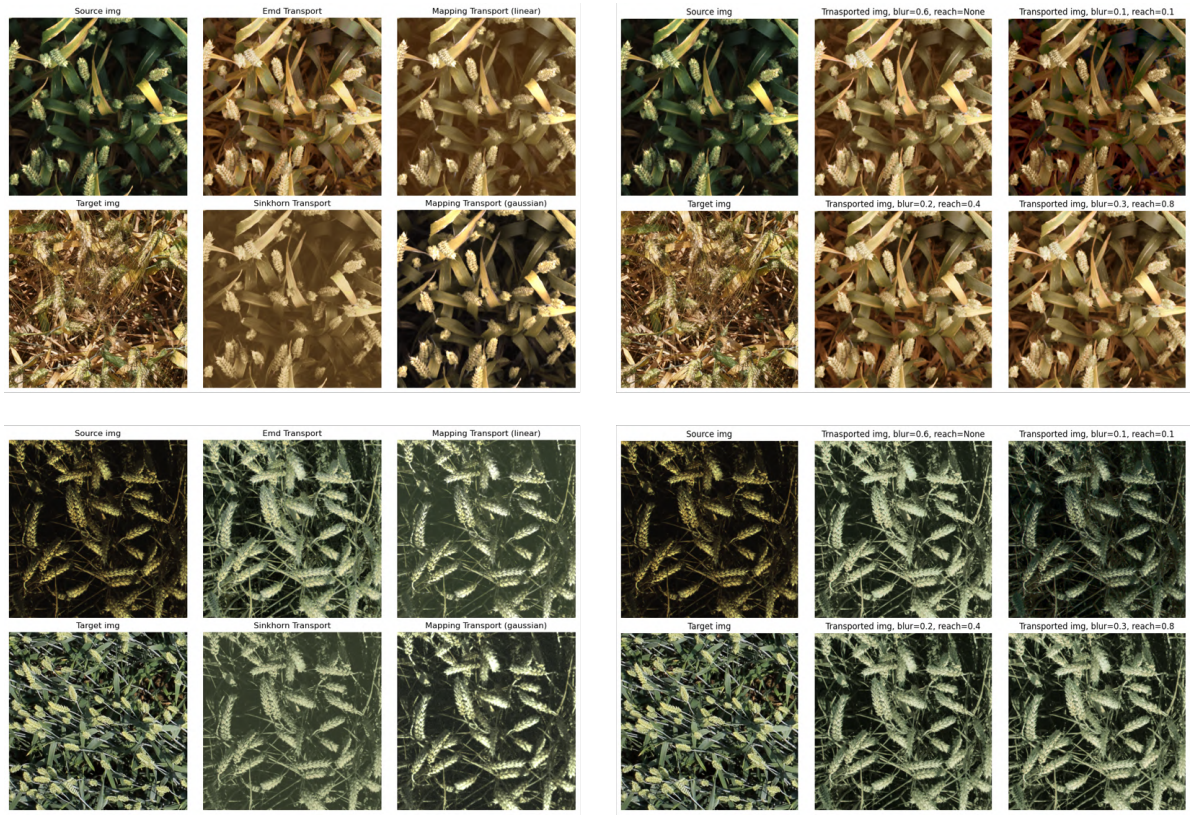


Figura 4.1 Pruebas de transporte óptimo en las imágenes con el algoritmo de Geomloss y los algoritmos de Python Optimal Transport.

En la Figura 4.1 se muestra la comparación de realizar la transformación de colores de dos imágenes de origen hacia los colores de dos imágenes objetivo. En cada uno de los paneles se incluye la imagen de origen que se transforma para parecerse a la imagen objetivo, y se incluyen cuatro transformaciones distintas. En el caso de los dos paneles de la izquierda, estas transformaciones fueron realizadas con distintos métodos disponibles en el paquete POT: transporte con la 'Earth Mover's Distance' (EMD) o la distancia de Wasserstein con $p = 1$, transporte regularizado mediante el algoritmo de Sinkhorn y transporte con un mapeo lineal y uno gaussiano. Por su parte, en los paneles de la derecha se incluyen transformaciones realizadas con cuatro combinaciones distintas de valores para los parámetros de desenfoque b y alcance r del algoritmo de GeomLoss.

En los paneles de la izquierda puede verse que los algoritmos de POT realizan bien el transporte óptimo, destacando principalmente el transporte EMD y el mapeo gaussiano, ya que con estos métodos se generan imágenes con mejores colores que lucen más parecidas a una fotografía real. En los paneles de la derecha puede verse que todas las combinaciones de parámetros del algoritmo de GeomLoss generan imágenes muy parecidas, a excepción de las imágenes con $b = 0.1$ y $r = 0.1$, pues las otras tres imágenes son más parecidas a lo obtenido por el transporte EMD de POT. Estos ejemplos muestran que visualmente se pueden lograr resultados similares con ambos paquetes, por lo que el uso de cada uno dependerá más del caso a tratar. En esta investigación, GeomLoss resultó una elección natural debido a que se priorizó tener un algoritmo que tomara el menor tiempo posible en realizar la transferencia de color de todas las imágenes.

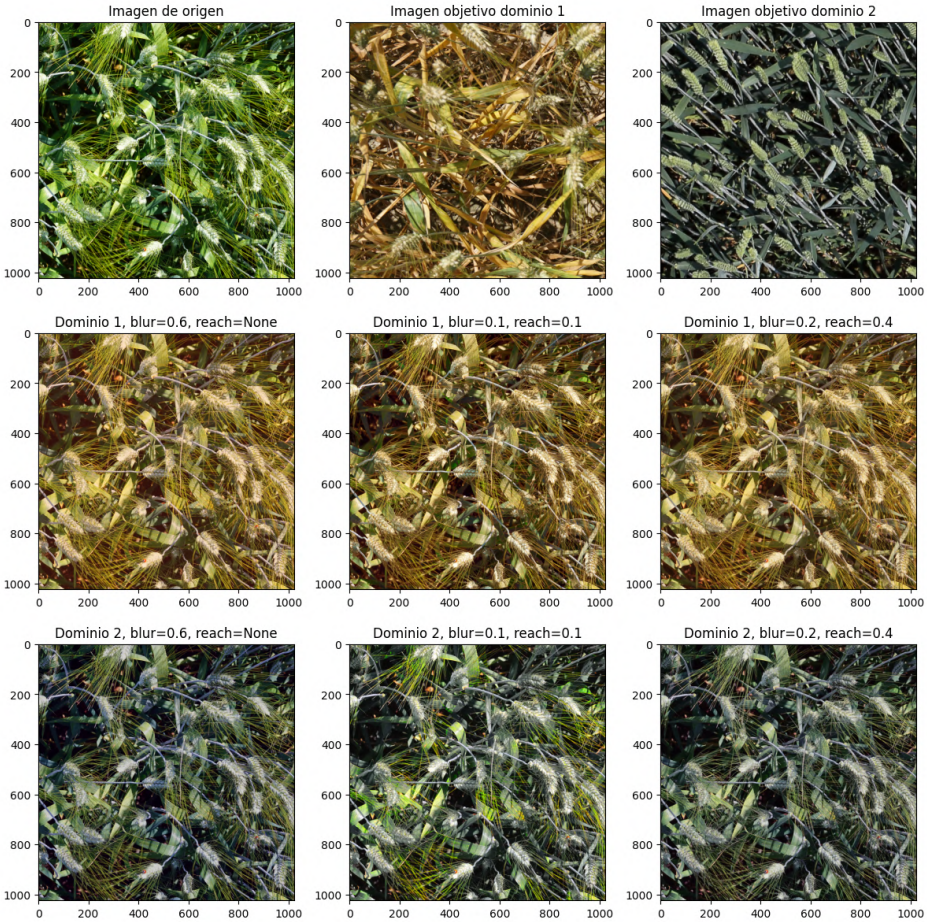


Figura 4.2 Resultado de la adaptación de dominio con el algoritmo elegido y tomando dos dominios objetivo y distintos parámetros.

Finalmente, se realizaron pruebas con otra imagen de origen y dos imágenes objetivo pertenecientes a distintos dominios de la base de datos, eligiendo tres combinaciones de parámetros al aplicar el algoritmo de GeomLoss. Estos ejemplos se pueden observar en la Figura 4.2, y permiten visualizar la diferencia que existe al emplear diferentes dominios objetivo en el transporte óptimo con la misma imagen de origen. Los dominios 1 y 2 corresponden a los dominios Arvalis_3 y ETHZ_1 de la base de datos, respectivamente, y son también los dominios objetivo elegidos en la comparación del desempeño de la red YOLO tratada en la siguiente sección. Para cada uno de los dos dominios, los resultados que se observan en la Figura 4.2 son parecidos entre las tres combinaciones de parámetros, dando ligeras variaciones de color en algunos elementos de la imagen. Sin embargo, estos resultados son sólo un ejemplo visual del transporte óptimo en una imagen y no representan una tendencia a ocurrir en todas las imágenes de la base de datos.

4.2. Comparación del desempeño del modelo con distintos parámetros de transporte óptimo

En este caso de estudio, se abordan los resultados obtenidos al comparar el desempeño de la red YOLO cuando ésta es entrenada con imágenes a las que se les aplicó transporte óptimo y cuando se entrena con las imágenes sin modificar. El 'modelo base', o aquel donde se utiliza la base de datos sin modificar las imágenes, sirve como punto de comparación para saber cuánto cambia el desempeño de la red al aplicar el transporte óptimo con distintos dominios objetivo y parámetros del algoritmo. Inicialmente se incluyen ejemplos donde, a través de tres imágenes donde se representan distintas situaciones, se compara cualitativamente cómo cambian las detecciones de espigas con un modelo donde el transporte óptimo sí mejora el desempeño de la red YOLO. Posteriormente, se presentan también las tablas de resultados donde se realiza la comparación de manera más general, de acuerdo a los experimentos que se realizaron siguiendo la metodología planteada en la sección 3.7.

Para obtener ejemplos que permitan observar en imágenes la diferencia que genera el transporte óptimo, se realizó una inferencia sobre algunas imágenes del conjunto de prueba, comparando las detecciones de objetos que obtiene el modelo base y el mejor modelo obtenido de acuerdo a los

resultados que se discuten más adelante, es decir, el que se entrenó con las imágenes adaptadas al dominio ETHZ_1 y con los parámetros $b = 0.6$ y $r = \infty$. La inferencia tomó un tamaño de imagen de 640×640 como en el entrenamiento, y un IoU de 0.5 para coincidir con el umbral utilizado en la métrica mAP50. Se eligieron imágenes que ilustran el efecto de la adaptación de dominio en estas situaciones: una imagen con una gran cantidad de espigas, pero donde éstas son pequeñas y están más separadas; una imagen con pocas espigas pequeñas, separadas y con una zona con exceso de luz; y finalmente una imagen con gran cantidad de espigas de diversos tamaños donde éstas están superpuestas. En todas las imágenes se presentan las detecciones realizadas en la imagen original a la izquierda y en la imagen con transporte óptimo a la derecha, además de que se incluyen las espigas reales encerradas en rectángulos de color verde y las detecciones de la red YOLO en color azul.

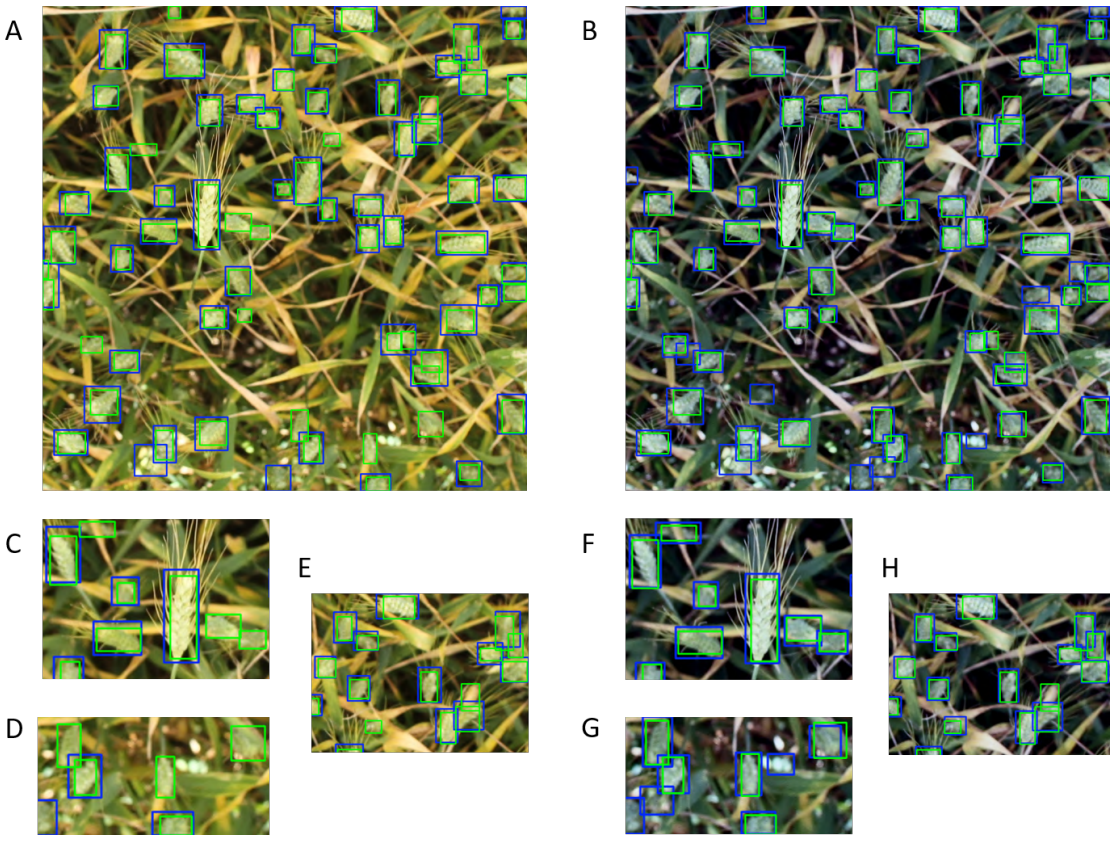


Figura 4.3 Comparación de la detección realizada por el modelo base y por el mejor modelo obtenido en una imagen con gran cantidad de espigas pequeñas; se incluyen acercamientos a algunas zonas de la imagen. En azul se muestran las espigas detectadas por el modelo y en verde las espigas reales.

En el caso de la Figura 4.3, donde se comparan las imágenes A y B y se proporcionan algunas ampliaciones de ciertas zonas en particular, se puede observar en las imágenes C, D y E que el modelo base tiene dificultades para detectar algunas de las espigas, pues en cada una de éstas existen 3 espigas no detectadas por el modelo. En cambio, el modelo entrenado con las imágenes adaptadas logra detectar la mayoría de estas espigas, pues en las imágenes F y G el modelo detectó las espigas que no se detectaron en las imágenes C y D, mientras que en la imagen H detecta dos de las tres espigas no detectadas en la imagen E. Sin embargo, en la imagen G se pueden observar dos detecciones falsas que no estaban presentes en la imagen D, por lo que, a pesar de que el modelo aumenta las detecciones de espigas verdaderas, también introduce algunas detecciones de espigas falsas. Finalmente, en las zonas ampliadas se puede observar una tendencia del modelo de la derecha a reducir el área existente entre la detección y la espiga real, por lo que en general la adaptación de dominio es beneficiosa en este tipo de situaciones.

Figura 4.4 Comparación de espigas detectadas por el modelo base y por el mejor modelo obtenido, en una imagen con pocas espigas y una zona excesivamente iluminada. Las espigas reales se muestran en color verde y las detectadas en azul.

En la Figura 4.4 es más sencillo observar que la adaptación de dominio mejora significativamente las detecciones, pues a pesar de que en la imagen de la derecha aún existen tres espigas no detectadas, se logran detectar dos espigas más que en la imagen de la izquierda y se eliminan las tres detecciones falsas presentes en la zona iluminada. Este ejemplo muestra cómo una adaptación

de dominio apropiada puede aminorar problemas presentes por un exceso de iluminación en las imágenes, trasladando estos tonos a otros donde la iluminación afecta de menor manera al modelo.

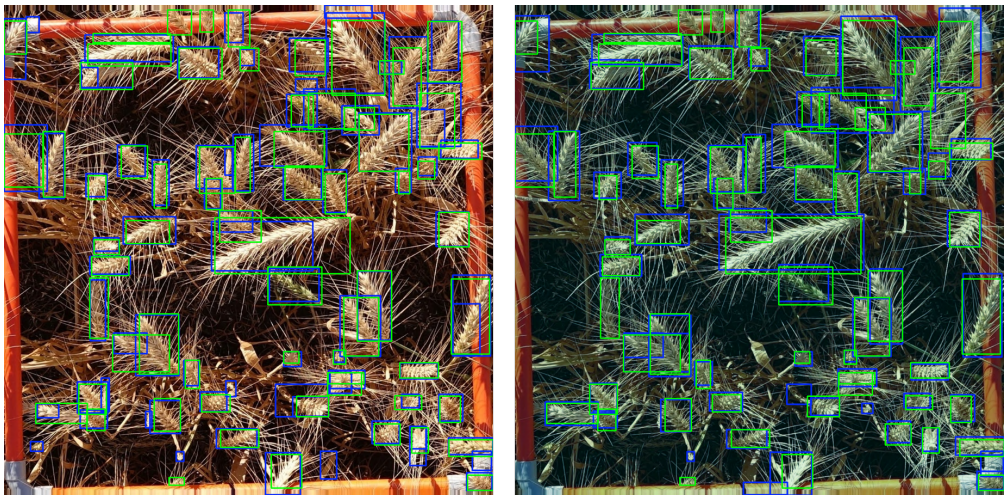


Figura 4.5 Comparación de la detección realizada por el modelo base y por el mejor modelo obtenido en una imagen con muchas espigas superpuestas. En azul se muestran las espigas detectadas por el modelo y en verde las espigas reales.

En el caso de la Figura 4.5 el modelo se enfrenta a una escena complicada, pues las espigas presentes son de tamaños diversos y cubren gran parte de la imagen, estando también algunas de ellas superpuestas. Aquí se observan en realidad pocas diferencias notables entre las detecciones obtenidas por ambos modelos, pues en ambos quedan espigas sin detectar y también ambos modelos realizan algunas detecciones falsas, por lo que no existe una tendencia clara que indique que el modelo de la derecha funciona mejor o peor que el de la izquierda. Esta situación muestra que la adaptación de dominio no logra resolver todo tipo de problemas en la detección de las espigas, por lo que problemas como la superposición requieren de una estrategia distinta.

Para obtener los resultados generales, siguiendo la metodología propuesta en la sección 3.7, primero se encontró el desempeño del modelo de detección de objetos para las imágenes sin modificar. Estos resultados se reportan en la Tabla 4.1 y corresponden a las métricas de Precisión, Sensibilidad, mAP50 y mAP50-95, obtenidas para los conjuntos de imágenes de validación y de prueba. Estos resultados sirven como una base para evaluar al modelo con las imágenes modificadas por el transporte óptimo, pero dentro de esta evaluación se considera mayormente el mAP50 en las imágenes de prueba, pues esta métrica resulta de más utilidad para el conteo de espigas en el que

se enmarca esta investigación. Para este propósito, se requiere detectar las espigas existentes sin importar que el cuadro delimitador no rodee a la espiga de una manera tan precisa.

Tabla 4.1 Resultados obtenidos para la detección de espigas con las imágenes sin modificar.

Conjunto de imágenes	**Precision**	**Sensibilidad**	**mAP50**	**mAP50-95**
Validación	0.908	0.827	0.905	0.496
Prueba	0.792	0.618	0.701	0.336

En las Tablas 4.2 y 4.3 se pueden observar los resultados obtenidos con los dominios Arvalis_3 y ETHZ_1 como dominios objetivo utilizando las cuatro combinaciones de parámetros del algoritmo de transporte óptimo que ahí se describen, donde los resultados en negritas son los mejores que se obtuvieron.

De la Tabla 4.2 se puede ver que, a pesar de superar el mAP50-95 de validación del modelo base en las cuatro ocasiones, el modelo con Arvalis_3 como dominio objetivo para el transporte óptimo sólo logra superar al modelo base en las otras tres métricas en una ocasión. En el caso de los resultados de prueba ocurre algo similar, pero el mAP50-95 sólo es superado dos veces. Este dominio objetivo solamente logra mejorar la detección de espigas en todas las métricas de prueba con los parámetros $b = 0.6$ y $r = \infty$, mientras que con las demás combinaciones de parámetros empeora la detección.

Tabla 4.2 Resultados de la detección de espigas realizando el transporte óptimo con diferentes parámetros y con el dominio objetivo Arvalis_3.

Parámetros		**Validación**				**Prueba**			
b	**r**	**Precisión**	**Sens.**	**mAP50**	**mAP50-95**	**Precisión**	**Sens.**	**mAP50**	**mAP50-95**
0.1	0.1	0.903	0.799	0.889	0.506	0.758	0.564	0.645	0.322
0.6	-	**0.914**	**0.841**	**0.914**	0.516	**0.806**	**0.630**	**0.709**	**0.337**
0.6	0.3	0.906	0.815	0.894	**0.517**	0.744	0.594	0.671	**0.337**
0.2	0.4	0.906	0.799	0.886	0.497	0.758	0.554	0.637	0.309

Ahora, los resultados con ETHZ_1 como dominio objetivo muestran un comportamiento distinto, pues en la Tabla 4.3 se observa que tanto en las imágenes de validación como en las de prueba se

Tabla 4.3 Resultados de detección de espigas considerando distintas combinaciones de parámetros de transporte óptimo y con el dominio objetivo ETHZ_1.

Parámetros		**Validación**				**Prueba**			
b	**r**	**Precisión**	**Sens.**	**mAP50**	**mAP50-95**	**Precisión**	**Sens.**	**mAP50**	**mAP50-95**
0.1	0.1	0.912	0.850	0.917	0.496	0.802	0.633	0.712	0.295
0.6	-	0.912	0.857	0.923	**0.511**	0.824	**0.663**	**0.742**	**0.315**
0.6	0.3	**0.920**	**0.862**	**0.924**	0.498	**0.829**	0.648	0.732	0.302
0.2	0.4	0.909	0.833	0.907	0.490	0.794	0.613	0.690	0.290

mejoran los resultados del modelo base en casi todas las métricas, excepto en el mAP50-95 en el conjunto de prueba. Esto quiere decir que muy probablemente disminuye el nivel de confianza en las detecciones de espigas; sin embargo, se obtiene un mayor número de detecciones con umbrales de 0.5 del IoU, por lo que el mAP50 se eleva mientras que el mAP50-95 disminuye. El mejor resultado se obtuvo nuevamente con los parámetros de $b = 0.6$ y $r = \infty$, donde se logra una mejora significativa del 4.1 % en el mAP50 de prueba. Así, se puede ver que este dominio sirve para aumentar el desempeño del modelo de manera notable, siempre y cuando se acepte la limitación de que la detección de las espigas no será tan precisa, lo cual no resulta un problema cuando sólo se requiere realizar un conteo de espigas.

4.3. Cambio en la distribución de los dominios y su efecto en el desempeño del modelo

Como se mencionó en la sección 3.8, para este segundo caso de estudio se plantearon tres experimentos distintos con el fin de determinar la influencia que tiene la distribución de los dominios en si un dominio funciona o no como dominio objetivo, además de determinar en cuál de estos casos el transporte óptimo otorga un mayor aumento en el desempeño del modelo YOLO, comparándolo con el desempeño que obtiene el modelo sin aplicar adaptación de dominio a las imágenes. En estos experimentos se varió principalmente el dominio objetivo, por lo que, de acuerdo a lo encontrado en el caso de estudio anterior, se mantuvieron los parámetros del algoritmo de transporte óptimo que mejor funcionaron, es decir, $b = 0.6$ y $r = \infty$.

Para llevar a cabo cada uno de los tres experimentos, se eliminó un número específico de imágenes de algunos dominios en particular. Posteriormente, se encontró un modelo base distinto en cada experimento, es decir, un modelo donde la red YOLO fue entrenada con las imágenes sin modificar, pues en cada experimento varía el número total de imágenes para el entrenamiento dependiendo de cuántas se eliminan. Finalmente, se entrenó el modelo varias veces con el transporte óptimo aplicado a las imágenes, considerando distintos dominios objetivo. A continuación, se detalla cada uno de los experimentos realizados y se presentan los resultados obtenidos para el desempeño del modelo.

4.3.1. Reducción del número de imágenes en el dominio predominante

Para llevar a cabo el primer experimento, se eliminaron 300 imágenes del dominio ETHZ_1, que es el que originalmente posee un mayor número de imágenes, con el fin de averiguar si este dominio nuevamente logra aumentar el desempeño del modelo YOLO de manera significativa, como ocurrió cuando se utilizó como dominio objetivo en el anterior caso de estudio. Esto ayudaría a comprender si un dominio puede funcionar igual de bien cuando no es el dominio predominante (el de mayor número de imágenes), o si, por el contrario, el qué tan bien funciona un dominio objetivo depende más de su número de imágenes que de las características de estas.

En la Figura 4.6 se puede observar cómo cambia la distribución de imágenes por dominio en este experimento respecto a la distribución original de imágenes en el conjunto de entrenamiento. En este caso, se puede ver que al eliminar las 300 imágenes, el dominio ETHZ_1 pasa a tener una cantidad de imágenes similar a Arvalis_5, quedando su número de imágenes por debajo de las que tiene el dominio Arvalis_3. El total de imágenes en este conjunto pasa entonces de 3657 a 3357, por lo que se entrenó un modelo base con esta cantidad de imágenes para obtener sus métricas de desempeño. Se eligieron tres dominios objetivo distintos para aplicar el transporte óptimo a las imágenes y comparar sus métricas de desempeño con las del modelo base. En la Tabla 4.4 se pueden observar las métricas de estos 4 modelos, donde el guion (-) en el dominio objetivo indica el modelo base que se entrenó en este experimento.

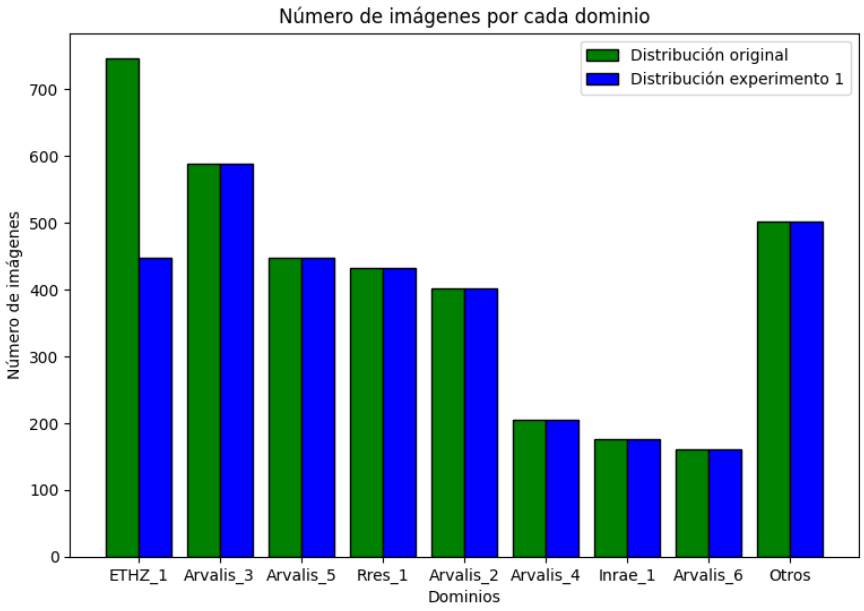


Figura 4.6 Distribución del número de imágenes por dominio para el experimento 1. La leyenda 'Otros' incluye a todos los dominios con menos de 100 imágenes.

De acuerdo con los resultados obtenidos en la Tabla 4.4, el modelo que logró una mayor mejora en el desempeño fue aquél en el que se realizó adaptación de dominio de las imágenes hacia el dominio objetivo Arvalis_3, pues en las imágenes de prueba se muestran los mejores resultados en las 4 métricas, siendo bastante notoria la mejora en el mAP50 y mAP50-95, donde se consiguen aumentos de 6.1 % y 5.7 %, respectivamente, si los comparamos con lo obtenido por el modelo base. En los tres dominios objetivos analizados se mejoran las métricas obtenidas por el modelo sin adaptación de dominio, pero, como se puede ver en la Figura 4.6, el dominio Arvalis_3 fue el que tenía un mayor número de imágenes en este experimento, por lo que se mantiene lo encontrado en el caso de estudio anterior, donde el dominio con mayor número de imágenes presentó el mayor aumento en el mAP50.

4.3.2. Dominios más equilibrados

En el segundo experimento, se eliminaron 346 imágenes del dominio ETHZ_1, 187 de Arvalis_3, 47 de Arvalis_5 y 31 de Rres_1, esto con el fin de que los dominios con mayor número de imágenes quedaran con el mismo número de estas, logrando algo cercano a una base de datos

Tabla 4.4 Resultados de la detección de espigas cuando cambia el dominio con mayor número de imágenes. Se muestran los resultados para el modelo sin adaptación de dominio y con adaptación de dominio empleando dominios objetivo distintos.

Dominio	**Validación**				**Prueba**			
objetivo	**Precisión**	**Sens.**	**mAP50**	**mAP50-95**	**Precisión**	**Sens.**	**mAP50**	**mAP50-95**
-	**0.915**	0.826	0.909	**0.517**	0.791	0.582	0.659	0.294
ETHZ_1	0.910	**0.844**	**0.914**	0.510	0.790	0.624	0.700	0.313
Arvalis_3	0.907	0.839	0.911	0.500	**0.804**	**0.637**	**0.720**	**0.351**
Arvalis_8	0.912	0.841	0.913	0.510	0.792	0.609	0.689	0.306

balanceada en la que se tiene el mismo número de imágenes de cada dominio. Así, este caso nos ayudaría a encontrar de qué manera se mejora o empeora el desempeño de la red YOLO cuando se aplica transporte óptimo a la base de datos balanceada, además de qué dominio objetivo podría funcionar mejor en un caso como este.

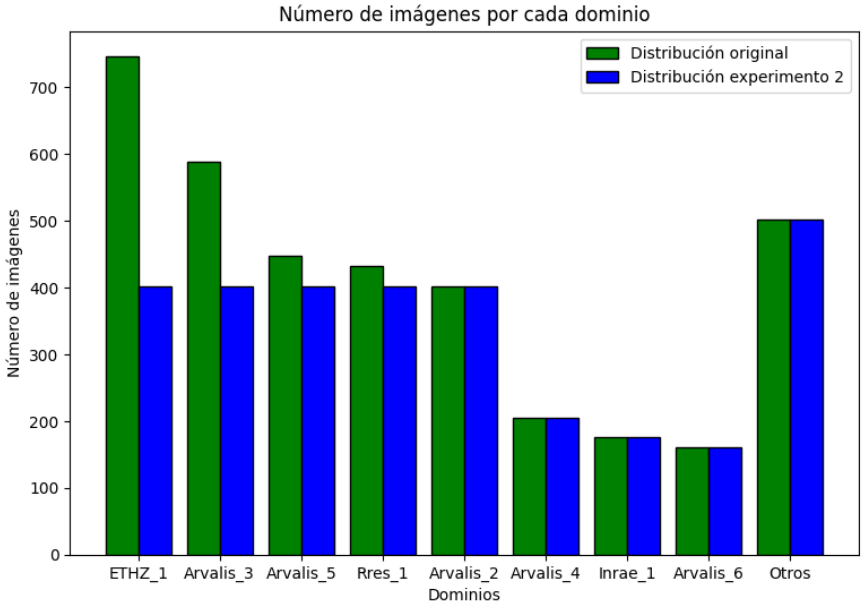


Figura 4.7 Distribución del número de imágenes por dominio para el experimento 2. La leyenda 'Otros' incluye a todos los dominios con menos de 100 imágenes.

En la Figura 4.7 se puede observar cómo los cuatro dominios antes mencionados, así como Arvalis_2, tienen un número igual de imágenes, lo que hace que la base de datos tenga una distribución

más equilibrada. Sin embargo, quedan aún varios dominios con un número distinto de imágenes, pero representan una menor proporción del total que los cinco dominios que se equilibraron. El total de imágenes para entrenamiento en este experimento fue de 3046, por lo que igualmente se obtuvo un modelo base entrenado con estas imágenes, y otros cuatro modelos en los que se aplicó la transferencia de color en las imágenes con diferentes dominios objetivo.

Tabla 4.5 Resultados de detección de espigas cuando hay un número igual de imágenes en varios de los dominios. Se incluyen los resultados del modelo base y aplicando adaptación de dominio con varios dominios objetivo.

Dominio	**Validación**				**Prueba**			
objetivo	**Precisión**	**Sens.**	**mAP50**	**mAP50-95**	**Precisión**	**Sens.**	**mAP50**	**mAP50-95**
-	0.906	0.813	0.898	0.490	0.757	0.564	0.638	0.279
ETHZ_1	0.910	0.841	0.914	**0.510**	0.795	0.622	0.696	0.301
Arvalis_3	0.888	0.792	0.880	0.487	0.719	0.563	0.629	0.266
Arvalis_5	**0.914**	**0.860**	**0.922**	0.509	**0.823**	**0.640**	**0.722**	**0.333**
Res_1	0.912	0.829	0.908	0.507	0.799	0.617	0.699	0.320

Luego de obtener el desempeño de los 5 modelos comparados, se puede observar que, de acuerdo a la Tabla 4.5, el dominio Arvalis_5 logra el mayor aumento del mAP50 en el conjunto de prueba conseguido en esta investigación, siendo del 8.4 % respecto al modelo base, y logrando también un aumento de 5.4 % en el mAP50-95. Esto refleja un mayor aumento en el desempeño del modelo utilizando adaptación de dominio cuando se tienen dominios más equilibrados, es decir, cuando varios de los dominios poseen un número de imágenes similar. Además, cabe destacar que el modelo con dominio objetivo Arvalis_3 empeora el desempeño del modelo base en este caso, por lo que no todos los dominios funcionan bien en una situación con dominios equilibrados y esto podría deberse más a las características de las imágenes de cada dominio.

4.3.3. Aumento de la brecha entre el dominio predominante y los demás dominios

Para realizar el tercer experimento, se eliminaron 384 imágenes del dominio Arvalis_3, 244 de Arvalis_5, 228 de Rres_1 y 197 de Arvalis_2, con la finalidad de que estos dominios tuvieran

igual número de imágenes que el dominio Arvalis_4, como se puede observar en la Figura 4.8, y que además la diferencia entre el número de imágenes de estos y el dominio ETHZ_1 fuera más pronunciada que antes. El objetivo de este experimento fue determinar el impacto que tiene la adaptación de dominio en un caso como este, donde el dominio predominante representa una gran proporción de los datos y su número de imágenes es mucho mayor que el de cualquier otro dominio. Con las imágenes eliminadas, el conjunto de entrenamiento quedó con 2604 imágenes, las cuales se utilizaron para obtener un modelo base que se pudiera comparar con otros tres modelos donde se aplicó el transporte óptimo. Estos resultados pueden observarse a continuación en la Tabla 4.6.

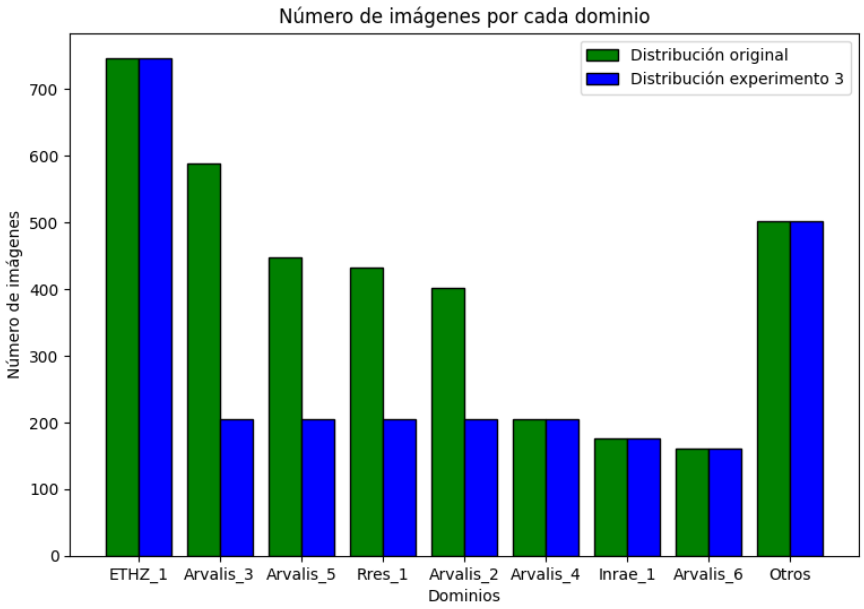


Figura 4.8 Distribución del número de imágenes por dominio para el experimento 3. La leyenda 'Otros' incluye a todos los dominios con menos de 100 imágenes.

En la Tabla 4.6 se puede observar que, para este experimento, los aumentos en mAP50 y mAP50-95 fueron menores que en los dos experimentos anteriores. En este caso, como era de esperarse, el dominio con mejor desempeño fue el que poseía un mayor número de imágenes, es decir, el dominio ETHZ_1. Este dominio logró un aumento de 2.9 % en el mAP50 respecto al modelo base, mientras que los otros dominios objetivo lograron un aumento de menos del 1 %. Esto parece indicar que la adaptación de dominio impacta menos en el desempeño del modelo YOLO cuando

existe un dominio que predomina sobre los demás, pues en este caso el dominio ETHZ_1 tenía 747 imágenes, mientras que los siguientes dominios en número de imágenes poseían tan solo 204.

Tabla 4.6 Resultados obtenidos en la detección de espigas cuando existe una gran brecha entre la cantidad de imágenes de un dominio y los demás. Se incluyen resultados del modelo base y de modelos que incluyen adaptación de dominio con distintos dominios objetivo.

Dominio	**Validación**				**Prueba**			
objetivo	**Precisión**	**Sens.**	**mAP50**	**mAP50-95**	**Precisión**	**Sens.**	**mAP50**	**mAP50-95**
-	**0.907**	0.811	0.895	0.494	0.763	0.584	0.661	0.285
ETHZ_1	0.889	0.818	0.897	0.493	0.767	**0.617**	**0.690**	0.303
Arvalis_5	0.896	0.811	0.894	**0.496**	0.750	0.597	0.669	**0.304**
Res_1	0.899	**0.825**	**0.900**	0.492	**0.776**	0.597	0.670	0.289

Capítulo 5

Conclusiones

En este trabajo se investigó el efecto de la adaptación de dominio por transporte óptimo en el desempeño de la detección de espigas de trigo de una red YOLOv5s. Los resultados muestran que el transporte óptimo aplicado a las imágenes de trigo mejora la detección de las espigas. En los diversos experimentos realizados con la base de datos GWHD 2021 obtuvimos mejoras en el mAP50 entre 4.1 % y 8.4 %. Encontramos que la mejora depende, además de los parámetros del algoritmo, de las características de los dominios fuente y objetivo. En lo que resta del capitulo se detallan los hallazgos de la investigación, se revisan los objetivos e hipótesis planteados, se discuten las contribuciones más relevantes de este estudio y se cierra con una perspectiva de las direcciones en que se puede ampliar esta investigación a través de futuros trabajos.

5.1. Objetivos alcanzados

El objetivo general de la investigación pudo concretarse de manera satisfactoria, ya que se logró mostrar que la metodología de adaptación de dominio con transporte óptimo aumenta el desempeño del modelo YOLO en diversas métricas, principalmente el mAP50, en el cual se logró aumentar más del 3 % que se tenía propuesto. En cuanto a los objetivos específicos, estos también fueron cumplidos de la siguiente manera:

- Se mostró que integrar la adaptación de dominio mediante transporte óptimo aumenta la precisión de detección de las espigas. Sin embargo, también se mostró que esto no siempre ocurre, pues se debe realizar una elección adecuada de los parámetros del algoritmo y del dominio objetivo para obtener dicho aumento.

- Se logró determinar un dominio objetivo donde funcionó mejor la detección de espigas. Este dominio fue el ETHZ_1, y se concluyó que la razón principal de su buen funcionamiento fue debido a que es el dominio con mayor número de imágenes.

- Se identificó un algoritmo de transporte óptimo rápido y preciso, el cual corresponde a un algoritmo del paquete GeomLoss que emplea la divergencia de Sinkhorn. Este permite realizar el transporte óptimo entre imágenes más rápido que otros paquetes como POT, lo cual permitió utilizarlo en toda la base de datos para realizar múltiples experimentos.

- A través de un caso de estudio donde se analizaron situaciones donde se altera el número de imágenes en los dominios, se identificó que una base de datos balanceada (dominios con una cantidad de imágenes similar) puede proporcionar las condiciones para que el transporte óptimo aumente en un mayor porcentaje la precisión del modelo.

- Mediante un análisis de literatura relevante, se identificaron 4 métricas utilizadas comúnmente en problemas de detección de objetos: Precisión, Sensibilidad, mAP50 y mAP50-95. De estas 4, se encontró que el maP50 representaba de mejor manera el desempeño del modelo para el conteo de espigas.

5.2. Hipótesis demostradas

La hipótesis de la cual partió esta investigación fue demostrada con los resultados obtenidos en el capítulo anterior, pues en ambos casos de estudio se obtuvieron aumentos en el mAP50 del modelo YOLOv5s mayores al 3 % al aplicar adaptación de dominio basada en transporte óptimo en las imágenes de espigas de trigo. En el primer caso de estudio, el aumento máximo obtenido fue del 4.1 %, mientras que en el segundo caso de estudio fue del 8.4 %, por lo cual se superaron las expectativas en cuanto a las capacidades del transporte óptimo como método para mejorar la detección de espigas.

5.3. Contribuciones de la investigación

En este trabajo, se propuso una metodología que permite mejorar el desempeño de un modelo YOLO en la detección de espigas de trigo, utilizando la adaptación de dominio basada en transporte óptimo en las imágenes de la base de datos GWHD 2021. La metodología propuesta tiene el enfoque de mejorar la detección mediante la adaptación de dominio, a diferencia de otros estudios que aumentan la complejidad del modelo o generan imágenes adicionales. Esto puede resultar de utilidad cuando no se pueden tener bases de datos muy grandes o no se dispone del tiempo o los recursos necesarios para entrenar un modelo muy complejo.

A través de las situaciones analizadas, se encontró que para toda la base de datos GWHD 2021, se puede llegar a un aumento del 4.1 % en el mAP50, mientras que si se eliminan imágenes para balancear la base de datos, se llega a obtener un aumento del 8.4 % del mAP50, todo esto cuando se compara con el desempeño del modelo YOLO sin aplicar adaptación de dominio a las imágenes. A pesar de que en algunos casos se logró un aumento en la métrica mAP50-95, se consideró más relevante el mAP50, pues éste se relaciona con un único umbral de IoU y, por lo tanto, no requiere de un traslape preciso de la caja de detección, lo que lo convierte en una métrica adecuada para aplicaciones de conteo, como es el caso de las espigas, donde es más necesario detectar la mayoría de espigas que hacerlo de una manera muy precisa.

También, se encontró que la metodología propuesta es sensible a la elección de parámetros del algoritmo de transporte óptimo y la elección de un dominio fuente y objetivo, pues estas dos variables influyen de manera significativa en si se mejora o no el desempeño de la red YOLO y en qué porcentaje, siendo el dominio objetivo el factor más importante. Además, se encontró que el transporte óptimo puede llevar a una mejora más significativa en el desempeño de la red YOLO cuando se tiene una base de datos con un número de imágenes similar en los dominios del conjunto de entrenamiento, y por el contrario, su impacto es menor cuando uno de los dominios posee un número de imágenes mucho mayor al de los demás dominios.

Finalmente, como producto de esta tesis, se realizó el artículo de investigación titulado 'Adaptación de Dominio en Imágenes para una Mejor Detección de Espigas de Trigo', el cual se presentó en el 18° Coloquio de Posgrado de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro

y posteriormente fue aceptado para su publicación como capítulo del libro 'Innovación sustentable: IA al servicio del planeta'.

5.4. Trabajos futuros

Como trabajo futuro, se tienen varias posibles direcciones, las cuales representan distintos aspectos de esta investigación que pueden ayudar a definir mejor la metodología, aumentar aún más el desempeño de las redes YOLO en la detección de espigas o encontrar distintas áreas de oportunidad donde se puedan aplicar estos métodos. Estas tres posibles vertientes se podrían desarrollar de la siguiente manera:

- Realizar una redefinición de los dominios mediante una técnica de agrupamiento, ya que durante la investigación se usaron los dominios definidos por los autores de la base de datos y se asumió que las imágenes dentro de cada uno de estos son similares entre sí. Así, si se definen nuevos dominios, puede proponerse alguna medida de la similitud entre las imágenes de un mismo dominio, lo que ayudaría a identificar los dominios con mejor o peor similitud interna y guiar hacia una mejor selección del dominio objetivo.

- Integrar la metodología propuesta en esta investigación con otras arquitecturas de redes neuronales, especialmente con algunas que hayan realizado una buena detección de espigas de trigo dentro de la literatura. Así, podría aumentarse más el desempeño de los modelos, pues se incorporaría esta metodología como preprocesamiento de las imágenes, para seguir con alguna arquitectura que haya identificado de buena manera las espigas y haya resuelto problemas que quedan fuera del alcance de esta metodología, como la superposición de las espigas.

- Aplicar estos métodos en otras bases de datos relacionadas con la detección de objetos donde se considere una sola clase de objetos, lo que permitiría saber qué tan generalizables son los hallazgos de esta investigación y si pueden ayudar en otras áreas donde también se requiera mejorar el desempeño de un modelo de detección de objetos.

Referencias

[1] A. S. Baldivia and G. R. Ibarra, "La disponibilidad de alimentos en méxico: Un análisis de la producción agrícola de 35 años y su proyección para 2050," *Papeles de Poblacion*, vol. 23, pp. 207–230, 2017.

[2] M. Z. Ihsan, F. S. El-Nakhlawy, S. M. Ismail, S. Fahad, and I. Daur, "Wheat phenological development and growth studies as affected by drought and late season high temperature stress under arid environment," *Frontiers in Plant Science*, vol. 7, p. 191357, 6 2016. [Online]. Available: www.frontiersin.org

[3] P. R. Shewry, "Wheat," *Journal of Experimental Botany*, vol. 60, pp. 1537–1553, 4 2009.

[4] P. Langridge, M. Alaux, N. F. Almeida, K. Ammar, M. Baum, F. Bekkaoui, A. R. Bentley, B. L. Beres, B. Berger, H. J. Braun, G. Brown-Guedira, C. J. Burt, M. J. Caccamo, L. Cattivelli, G. Charmet, P. Civáň, S. Cloutier, J. P. Cohan, P. J. Devaux, F. M. Doohan, M. F. Dreccer, M. Ferrahi, S. E. Germán, S. B. Goodwin, S. Griffiths, C. Guzmán, H. Handa, M. J. Hawkesford, Z. He, E. Huttner, T. M. Ikeda, B. Kilian, I. P. King, J. King, J. A. Kirkegaard, J. Lage, J. L. Gouis, S. Mondal, E. Mullins, F. Ordon, J. I. Ortiz-Monasterio, H. Özkan, İrfan Öztürk, S. A. Pereyra, C. J. Pozniak, H. Quesneville, M. C. Quincke, G. J. Rebetzke, J. C. Reif, T. Saavedra-Bravo, U. Schurr, S. Sharma, S. K. Singh, R. P. Singh, J. W. Snape, W. Tadesse, H. Tsujimoto, R. Tuberosa, T. G. Willis, and X. Zhang, "Meeting the challenges facing wheat production: The strategic research agenda of the global wheat initiative," *Agronomy*, vol. 12, 11 2022.

[5] J. Hyles, M. T. Bloomfield, J. R. Hunt, R. M. Trethowan, and B. Trevaskis, "Phenology and related traits for wheat adaptation," *Heredity 2020 125:6*, vol. 125, pp. 417–430, 5 2020. [Online]. Available: https://www.nature.com/articles/s41437-020-0320-1

[6] K. Bozhurin, "Wheat life cycle and growth: All you need to know," 11 2024. [Online]. Available: https://growplant.org/blog/wheat-life-cycle/

[7] J. G. Hampton, B. L. M. Cloy, and D. R. M. Millan, "Ear populations and wheat production," *New Zealand Journal of Experimental Agriculture*, vol. 9, pp. 185–189, 4 1981.

[8] J. A. Fernandez-Gallego, S. C. Kefauver, N. A. Gutiérrez, M. T. Nieto-Taladriz, and J. L. Araus, "Wheat ear counting in-field conditions: High throughput and low-cost approach using rgb images," *Plant Methods*, vol. 14, 3 2018.

[9] S. Rasti, C. J. Bleakley, N. M. Holden, R. Whetton, D. Langton, and G. O'Hare, "A survey of high resolution image processing techniques for cereal crop growth monitoring," *Information Processing in Agriculture*, vol. 9, pp. 300–315, 6 2022.

[10] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors 2018, Vol. 18, Page 2674*, vol. 18, p. 2674, 8 2018.

[11] T. van Klompenburg, A. Kassahun, and C. Catal, "Crop yield prediction using machine learning: A systematic literature review," *Computers and Electronics in Agriculture*, vol. 177, p. 105709, 10 2020.

[12] A. Sanaeifar, M. L. Guindo, A. Bakhshipour, H. Fazayeli, X. Li, and C. Yang, "Advancing precision agriculture: The potential of deep learning for cereal plant head detection," *Computers and Electronics in Agriculture*, vol. 209, p. 107875, 6 2023.

[13] E. David, F. Ogidi, D. Smith, S. Chapman, B. de Solan, W. Guo, F. Baret, and I. Stavness, "Global Wheat Head Detection Challenges: Winning Models and Application for Head Counting," *Plant Phenomics*, vol. 5, 6 2023. [Online]. Available: https://spj.science.org/doi/10.34133/plantphenomics.0059

[14] E. David, M. Serouart, D. Smith, S. Madec, K. Velumani, S. Liu, X. Wang, F. Pinto, S. Shafiee, I. S. A. Tahir, H. Tsujimoto, S. Nasuda, B. Zheng, N. Kichgessner, H. Aasen, A. Hund, P. Sadhegi-Tehran, K. Nagasawa, S. Dandrifosse, A. Carlier, B. Dumont, B. Mercatoris, B. Evers, K. Kuroki, H. Wang, M. Ishii, M. A. Badhon, C. Pozniak, D. S. Lebauer, M. Lillemo, J. Poland, S. Chapman, B. D. Solan, F. Baret, I. Stavness, and W. Guo, "Global wheat head dataset 2021: more diversity to improve the benchmarking of wheat head localization methods," 5 2021. [Online]. Available: https://arxiv.org/abs/2105.07660v2

[15] S. C. Okafor, L. Wei, S. Boamah, L. Zhang, and M. B. Diallo, "Enhanced Wheat Head Detection in Images Using Fourier Domain Adaptation and Random Guided Filter," *Canadian Journal of Remote Sensing*, vol. 50, no. 1, 2024.

[16] C. Liu, K. Wang, H. Lu, and Z. Cao, "Dynamic color transform networks for wheat head detection," *Plant Phenomics*, vol. 2022, 2022.

[17] M. Zhang, H. Marklund, N. Dhawan, A. Gupta, S. Levine, and C. Finn, "Adaptive Risk Minimization: Learning to Adapt to Domain Shift," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 23 664–23 678. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/c705112d1ec18b97acac7e2d63973424-Paper.pdf

[18] M. H. Tanveer, Z. Fatima, S. Zardari, and D. Guerra-Zubiaga, "An In-Depth Analysis of Domain Adaptation in Computer and Robotic Vision," *Applied Sciences 2023, Vol. 13, Page 12823*, vol. 13, no. 23, p. 12823, 11 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/23/12823/htmhttps://www.mdpi.com/2076-3417/13/23/12823

[19] Z. K. Hartley and A. P. French, "Domain adaptation of synthetic images for wheat head detection," *Plants*, vol. 10, 12 2021.

[20] N. Papadakis, “Optimal transport for image processing,” 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:125551356

[21] H. Tian, T. Wang, Y. Liu, X. Qiao, and Y. Li, “Computer vision technology in agricultural automation —a review,” *Information Processing in Agriculture*, vol. 7, pp. 1–19, 3 2020.

[22] V. Wiley and T. Lucas, “Computer vision and image processing: A paper review,” *International Journal of Artificial Intelligence Research*, vol. 2, pp. 29–36, 6 2018. [Online]. Available: https://ijair.id/index.php/ijair/article/view/42

[23] B. Chitradevi, P. Srimathi, and A. Professor, “An overview on image processing techniques,” *International Journal of Innovative Research in Computer and Communication Engineering (An ISO*, vol. 3297, 2007. [Online]. Available: www.ijircce.com

[24] “Introduction to image processing.” [Online]. Available: https://esahubble.org/static/projects/fits_liberator/image_processing.pdf

[25] X. Feng, Y. Jiang, X. Yang, M. Du, and X. Li, “Computer vision algorithms and hardware implementations: A survey,” *Integration*, vol. 69, pp. 309–320, 11 2019.

[26] X. Zou, “A review of object detection techniques,” *Proceedings - 2019 International Conference on Smart Grid and Electrical Automation, ICSGEA 2019*, pp. 251–254, 8 2019.

[27] M. Walia, “Semantic segmentation vs. instance segmentation: Explained,” 10 2022. [Online]. Available: https://blog.roboflow.com/difference-semantic-segmentation-instance-segmentation/

[28] H. Cholakkal, G. Sun, F. S. Khan, and L. Shao, “Object counting and instance segmentation with image-level supervision,” pp. 12 397–12 405, 2019.

[29] Y. Wang, Y. Qin, and J. Cui, “Occlusion robust wheat ear counting algorithm based on deep learning,” *Frontiers in Plant Science*, vol. 12, 6 2021.

[30] N. O’Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, “Deep learning vs. traditional computer vision,” *Advances in Intelligent Systems and Computing*, vol. 943, pp. 128–144, 2020. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-17795-9_10

[31] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” *Computational Intelligence and Neuroscience*, vol. 2018, 2018.

[32] J. Zou, Y. Han, and S. S. So, “Overview of artificial neural networks,” *Methods in Molecular Biology*, vol. 458, pp. 14–22, 2008. [Online]. Available: https://link.springer.com/protocol/10.1007/978-1-60327-101-1_2

[33] M. Ullmo, “Emulation and prediction of cosmic web simulations through deep learning,” Theses, Université Paris-Saclay, Feb. 2022. [Online]. Available: https://theses.hal.science/tel-03663099

[34] Y. chen Wu and J. wen Feng, "Development and application of artificial neural network," *Wireless Personal Communications*, vol. 102, pp. 1645–1656, 9 2018. [Online]. Available: https://link.springer.com/article/10.1007/s11277-017-5224-x

[35] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks," *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, pp. 1836–1841, 7 2018.

[36] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Neural Networks*, vol. 138, pp. 14–32, 6 2021.

[37] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. [Online]. Available: http://probml.github.io/book2

[38] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "Cnn variants for computer vision: History, architecture, application, challenges and future scope," *Electronics 2021, Vol. 10, Page 2470*, vol. 10, p. 2470, 10 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/20/2470/htmhttps://www.mdpi.com/2079-9292/10/20/2470

[39] A. Kumar, "Different types of cnn architectures explained: Examples," 12 2023. [Online]. Available: https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/

[40] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. [Online]. Available: probml.ai

[41] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, and X. Lan, "A review of object detection based on deep learning," *Multimedia Tools and Applications*, vol. 79, pp. 23 729–23 791, 9 2020. [Online]. Available: https://link.springer.com/article/10.1007/s11042-020-08976-6

[42] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, pp. 261–318, 2 2020. [Online]. Available: https://link.springer.com/article/10.1007/s11263-019-01247-4

[43] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using yolo: challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, pp. 9243–9275, 3 2023. [Online]. Available: https://link.springer.com/article/10.1007/s11042-022-13644-y

[44] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," 4 2023. [Online]. Available: http://arxiv.org/abs/2304.00501http://dx.doi.org/10.3390/make5040083

[45] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde, "Optimal mass transport: Signal processing and machine-learning applications," *IEEE Signal Processing Magazine*, vol. 34, pp. 43–59, 7 2017.

[46] C. Villani, *Optimal transport, old and new*. Springer, 6 2008.

[47] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1853–1865, 7 2015. [Online]. Available: https://arxiv.org/abs/1507.00504v2

[48] G. Peyré and M. Cuturi, "Computational Optimal Transport," *Foundations and Trends in Machine Learning*, vol. 11, pp. 355–607, 2019. [Online]. Available: http://arxiv.org/abs/1803.00567

[49] L. Ambrosio and N. Gigli, "A user's guide to optimal transport," *Lecture Notes in Mathematics*, vol. 2062, pp. 1–155, 1 2013.

[50] F. Santambrogio, *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*, 1st ed., ser. Progress in Nonlinear Differential Equations and Their Applications. Birkhäuser, 10 2015, vol. 87. [Online]. Available: https://link.springer.com/10.1007/978-3-319-20828-2

[51] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouve, and G. Peyré, "Interpolating between optimal transport and mmd using sinkhorn divergences," in *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019, pp. 2681–2690.

[52] J. Feydy, "Analyse de données géométriques, au delà des convolutions," Theses, Université Paris-Saclay, Jul. 2020. [Online]. Available: https://theses.hal.science/tel-02945979

[53] T. Séjourné, J. Feydy, F.-X. Vialard, A. Trouvé, and G. Peyré, "Sinkhorn Divergences for Unbalanced Optimal Transport," 10 2019. [Online]. Available: https://arxiv.org/abs/1910.12958v3

[54] Y. Zhang, M. Li, X. Ma, X. Wu, and Y. Wang, "High-precision wheat head detection model based on one-stage network and gan model," *Frontiers in Plant Science*, vol. 13, 6 2022.

[55] X. Shen, C. Zhang, K. Liu, W. Mao, C. Zhou, and L. Yao, "A lightweight network for improving wheat ears detection and counting based on YOLOv5s," *Frontiers in Plant Science*, vol. 14, p. 1289726, 12 2023. [Online]. Available: http://labelme.csail.mit.edu/

[56] R. H. Sampieri, S. M. Valencia, C. P. M. Torres, and A. C. Romo, *Fundamentos de Investigación*, 1st ed. McGraw-Hill, 2017.

[57] "What is ground sample distance (gsd)? | vision aerial." [Online]. Available: https://visionaerial.com/what-is-ground-sample-distance/

[58] G. Jocher, "Ultralytics yolov5," 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[59] T. Khan and M. Goodwin, "¿qué es yaml?" 12 2023. [Online]. Available: https://www.ibm.com/es-es/topics/yaml

[60] R. Ribani and M. Marengoni, "A survey of transfer learning for convolutional neural networks," *Proceedings - 32nd Conference on Graphics, Patterns and Images Tutorials, SIBGRAPI-T 2019*, pp. 47–57, 10 2019.

[61] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *Lecture Notes in Computer Science*, vol. 8693 LNCS, pp. 740–755, 5 2014. [Online]. Available: https://arxiv.org/abs/1405.0312v3

[62] R. Padilla, S. L. Netto, and E. A. D. Silva, "A survey on performance metrics for object-detection algorithms," *International Conference on Systems, Signals, and Image Processing*, vol. 2020-July, pp. 237–242, 7 2020.

[63] Kukil, "Mean average precision (map) in object detection," 8 2022. [Online]. Available: https://learnopencv.com/mean-average-precision-map-object-detection-model-evaluation-metric/

[64] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer, "Pot: Python optimal transport," *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-451.html

[65] B. Charlier, J. Feydy, J. A. Glaunès, F.-D. Collin, and G. Durif, "Kernel operations on the gpu, with autodiff, without memory overflows," *Journal of Machine Learning Research*, vol. 22, no. 74, pp. 1–6, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-275.html

Anexos

Se participó en el 18° Coloquio de Posgrado de la Facultad de Ingeniería de la UAQ, llevado a cabo del 19 al 23 de noviembre de 2024 en Querétaro, México. En este congreso, se presentó como ponencia el artículo “Adaptación de Dominio en Imágenes para una Mejor Detección de Espigas de Trigo”.

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA
DIPFI POSGRADO INGENIERÍA

COLOQUIO 18° POSGRADO FI

Se otorga la presente

CONSTANCIA a:

Jesús Eduardo Salas Ibañez y Gamaliel Moreno Chávez

Por su participación en el
XVIII Coloquio de Posgrado de la Facultad de Ingeniería
de la Universidad Autónoma de Querétaro con el artículo:

Adaptación de Dominio en Imágenes para una Mejor Detección de Espigas de Trigo

Noviembre de 2024
Facultad de Ingeniería

Dra. María de la Luz Pérez Rea
Directora
Facultad de Ingeniería

Dr. Juan Carlos Jáuregui Correa
Jefe de la División de Investigación y Posgrado
Facultad de Ingeniería

Adaptación de Dominio en Imágenes para una Mejor Detección de Espigas de Trigo

Domain Adaptation in Images for a Better Wheat Head Detection

Salas Ibañez, Jesús Eduardo; Moreno Chávez, Gamaliel
Unidad Académica de Ingeniería Eléctrica
Universidad Autónoma de Zacatecas
Zacatecas, México
eduardo.si@uaz.edu.mx

Palabras clave: IAR, espigas de trigo, adaptación de dominio, detección de objetos, visión por computadora, aprendizaje profundo.

Resumen- La densidad de espigas es un componente importante a la hora de determinar la cosecha de trigo. Por esta razón, se ha propuesto estimarla mediante un conteo automático de las espigas de trigo en imágenes a color, tarea en la cual los modelos de redes neuronales para detección de objetos han demostrado gran capacidad. Sin embargo, estos modelos pueden enfrentar problemas para identificar correctamente las espigas cuando existe mucha variación visual en su aspecto en distintas imágenes. Este trabajo presenta una forma de atacar este problema mediante la aplicación de un algoritmo de adaptación de dominio basada en transporte óptimo, con el cual se puede cambiar la paleta de colores de una imagen para que sea visualmente más parecida a otra, reduciendo así parte de esta variación visual. Al aplicar este algoritmo a las imágenes de la base de datos Global Wheat Head Detection 2021, se encontró que se puede aumentar el mAP50 de un modelo YOLOv5s hasta en un 4.1%, lo cual muestra el potencial que tienen las técnicas de adaptación de dominio en la mejora del desempeño de un modelo de detección de objetos.

Keywords: IAR, wheat heads, domain adaptation, optimal transport, object detection, neural networks.

Abstract- Spike density is an important parameter when determining the wheat yield. For this reason, it has been proposed to estimate it through the automatic counting of wheat heads in color images, a task in which object detection neural network models have demonstrated great capability. However, these models may face difficulties in correctly identifying wheat heads when there is significant visual variation in their appearance across different images. This work presents a way to address this issue by applying a domain adaptation algorithm based on optimal transport, which allows for changing the color palette of an image to make it visually more similar to another one, thereby reducing part of this visual variation. By applying this algorithm to the images from the Global Wheat Head Detection dataset 2021, it was found that the mAP50 of a YOLOv5s model can be increased by up to 4.1%, demonstrating the potential of domain adaptation techniques in improving the performance of an object detection model.

1. Introducción

El trigo posee un rol estratégico en la seguridad alimentaria mundial ya que es el cultivo más extendido en el mundo, con un estimado de 216 millones de hectáreas de área sembrada en 2019 y una producción anual por país de más de 10000 toneladas. Sin embargo, en los siguientes años, la producción de trigo se verá desafiada a consecuencia del cambio climático, por lo que la adopción de nuevas tecnologías en esta área representa una