

Detección Multimodal de Animales para el Ciber-Pastoreo de Ganado

Jorge Urbón Burgos
777295@unizar.es

Supervisor: Rosario Aragüés
raragues@unizar.es

Co-Supervisor: Jesús Bermúdez
bermudez@unizar.es

21 de enero de 2025

Resumen

Índice general

| | |
|--|-----------|
| 1. Introducción | 2 |
| 1.1. Motivación | 2 |
| 1.2. Objetivos | 2 |
| 1.3. Trabajos Relacionados | 2 |
| 1.3.1. Redes Convolucionales | 3 |
| 1.3.2. UNets | 3 |
| 1.3.3. Mask R-CNN | 3 |
| 1.3.4. Transformers | 3 |
| 1.3.5. Monitorización de Animales | 4 |
| 2. Datos | 5 |
| 2.1. Datasets Disponibles | 5 |
| 2.2. Lindenthal Camera Traps Dataset | 6 |
| 2.2.1. Desglose del <i>Dataset</i> | 6 |
| 3. Cross Modal Fusion | 8 |
| 3.1. Descripción del Modelo | 8 |
| 3.2. Técnicas de Pre-Procesado | 9 |
| 3.2.1. HHA Encoding | 9 |
| 3.2.2. Colorización Jet | 10 |
| 3.2.3. Colorización por Distancia | 11 |
| 3.2.4. Normales | 11 |
| 3.3. Entrenamiento | 12 |
| 3.3.1. Implementación | 13 |
| 3.4. Comparación de Técnicas | 14 |
| 4. DFormer | 15 |
| 4.1. Descripción del Modelo | 15 |
| 4.2. El pre-entrenamiento | 15 |
| 4.3. Resultados | 16 |
| 5. Resultados | 17 |
| 5.1. Cross Modal Fusion | 17 |
| 6. Conclusiones y trabajo futuro | 19 |
| 6.1. Conclusiones | 19 |
| 6.2. Trabajo futuro | 19 |

Capítulo 1

Introducción

1.1. Motivación

Este trabajo parte a raíz de lo propuesto en el proyecto COUNTRYBOTS llevado a cabo por el grupo Ropert, cuyo principal objetivo es la automatización y robotización de tareas agrícolas y ganaderas mediante el uso de sistemas multirobot que combinan imágenes aéreas para una mejor monitorización del entorno.

La motivación del trabajo aquí presentado radica en la necesidad de diferentes barreras que la monitorización de animales puede presentar, siendo el *trackeo* de animales mediante el uso de varias modalidades una posible solución a estos problemas. La segmentación multimodal en visión artificial tiene el potencial de mejorar significativamente el desempeño de las labores de ciber-pastoreo y vigilancia de animales en entornos naturales, ya que la información adicional que diferentes modalidades (o las interacciones entre las mismas) puede aportar al modelo puede ser clave para la correcta segmentación de los animales en la escena.

Este trabajo se centra principalmente en explorar, implementar y evaluar diferentes técnicas multimodales para el *trackeo* de animales.

1.2. Objetivos

El objetivo de este trabajo es, por tanto, la verificación de la utilidad de las técnicas multimodales para la identificación y monitoreo de animales en entornos naturales con el fin de demostrar su utilidad de cara a conseguir características más ricas y completas. Para ello, se analizarán diferentes técnicas basadas en *Deep Learning* y se evaluarán los resultados obtenidos.

1.3. Trabajos Relacionados

El campo de la visión por computador ha sufrido una enorme revolución a lo largo de la última década gracias, principalmente, a los avances en *Deep Learning*. A continuación se presentan algunos de los trabajos más relevantes en lo referente a este trabajo.

1.3.1. Redes Convolucionales

A pesar de la antigüedad del concepto, las redes neuronales convolucionales CNN supusieron un gran avance en el reconocimiento de imágenes desde la introducción de AlexNet en [1]. Desde entonces, las CNN han sido ampliamente utilizadas una gran variedad de tareas relacionadas con la Visión por Computador como la clasificación de imágenes, detección y segmentación de objetos. A lo largo de este trabajo, se emplearán diferentes técnicas que ponen en uso este tipo de redes neuronales.

1.3.2. UNets

Los modelos UNet, presentados originalmente en [2] como una alternativa para la segmentación de imágenes biomédicas, han sido ampliamente utilizados en tareas de segmentación generales debido a su arquitectura sencilla y efectiva consistente en una serie de bloques de *upsampling* y *downsampling* que permiten al modelo aprender características a diferentes niveles de abstracción en la imagen de entrada (mostrado en la figura 1.1). Este tipo de modelos, aunque relativamente antiguos, siguen siendo relevantes en la actualidad e incluso se han llegado a implementar versiones multimodales como [3] o [4].

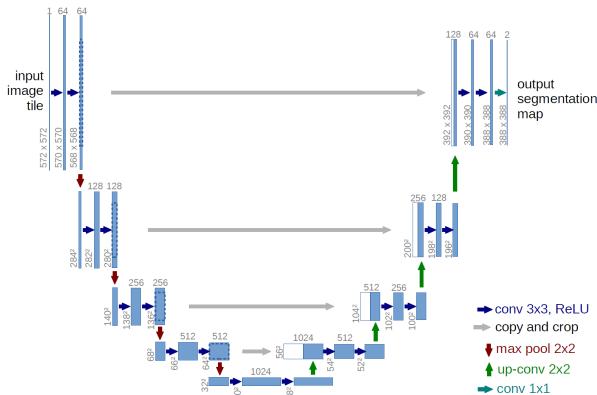


Figura 1.1: Arquitectura de un modelo UNet

1.3.3. Mask R-CNN

Como sucesor a Faster R-CNN (trabajo presentado en [5] que se limita a la detección de regiones de interés RoI), Mask R-CNN ([6]) se presenta como una amplia mejora de su predecesor al incluir una rama de segmentación de instancias en paralelo a la rama de detección de objetos como se muestra en la figura 1.2

1.3.4. Transformers

Uno de los más recientes y posiblemente más relevantes avances en el campo del *Deep Learning* ha sido la introducción de los modelos **Transformers** en [7]. Estos modelos han sido amplia y popularmente empleados en tareas de procesamiento de lenguaje natural (NLP) en los últimos años([8], [9], [10]) debido a la ya demostrada eficacia de su arquitectura basada en la *self attention*, aunque también han supuesto un gran progreso en las tareas de visión por computador tras la demostración en [11] de que pueden incluso superar el rendimiento de las CNN en tareas de clasificación.

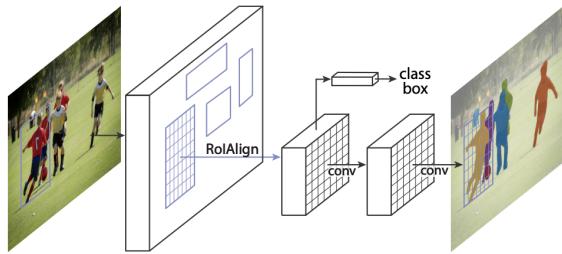


Figura 1.2: Arquitectura de un modelo Mask R-CNN

La capacidad de este tipo de arquitecturas en tareas de segmentación ya ha sido demostrada en una variedad de trabajos ([12], [13], [14]) y se espera que su aplicación en tareas multimodales como la segmentación de imágenes sea igualmente efectiva.

1.3.5. Monitorización de Animales

Aunque no se trate de un tema tan ampliamente estudiado como la segmentación de imágenes, la monitorización de animales también se ha visto beneficiada por los avances en visión por computador dados en los últimos años. Proyectos como [15], [16], [17] o estudios del arte como [18] demuestran ya la utilidad técnicas de localización de animales basadas en *deep learning* mediante el uso de imágenes aéreas.

Capítulo 2

Datos

Dado el objetivo que se tiene en este trabajo de verificar la utilidad de las técnicas de segmentación multimodales en la identificación y segmentación de animales en imágenes en entornos naturales, se focaliza la búsqueda en *datasets* que contengan, idealmente, pares de imágenes multimodales que ya hayan sido sincronizadas, alineadas y etiquetadas previamente con el fin de facilitar el proceso de entrenamiento y validación de los modelos. Además, se busca que dichos posibles *datasets* contengan una variedad de animales y situaciones que permitan una generalización adecuada de los modelos a entrenar.

2.1. Datasets Disponibles

Con el fin de encontrar un conjunto de datos que se adecúe a las necesidades anteriormente expresadas, se realiza una búsqueda exhaustiva de *datasets* disponibles de forma pública en internet. Cabe destacar las particularidades bajo las que se trabaja, ya que no solo se requiere que el *dataset* contenga imágenes de animales, si no que además se necesita que más de una modalidad esté presente y que los datos hayan sido ya etiquetados y estén listos para el entrenamiento. A continuación, se presentan estos *datasets* y se analizan sus características.

NOAA Arctic Seals 2019: este *dataset* presentado en [19] cuenta con alrededor de 80.000 pares de imágenes a color (RGB) e infrarrojas térmicas (IRT) tomadas desde el aire durante vuelos desarrollados en Alaska. A pesar de que este *dataset* cuenta con una gran cantidad de imágenes (incluyendo unas 14.000 *bounding boxes*), la variedad de animales presentes en el mismo es limitada, por lo que no se considera adecuado para este trabajo.

UAV-derived waterfowl thermal imagery dataset: este *dataset* presentado en [20] también consiste en imágenes aéreas de aves acuáticas tomadas con cámaras infrarrojas térmicas (IRT) y cámaras RGB desde un *drone* sobre la región de Nebraska. El *dataset* cuenta con 8.976 *bounding boxes*, sin embargo sus modalidades no se encuentran alineadas, lo que dificultaría mucho el proceso de entrenamiento de los modelos.

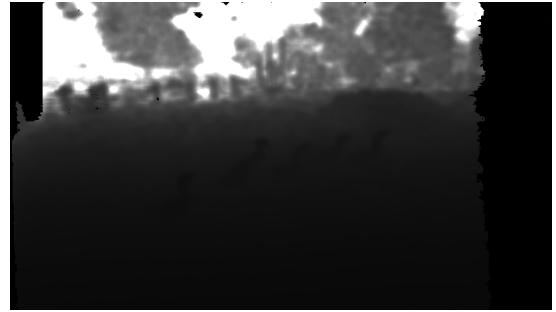
Lindenthal Camera Traps: este otro *dataset* presentado en [21] está construido a partir de pares infrarrojos para visión nocturna (IR) y mapas de profundidad (D) obtenidos a través de una cámara estéreo fija a pocos metros del suelo. Este *dataset* cuenta con un total de 1.038 instancias segmentadas de animales de 4 clases diferenciadas. A pesar de que este *dataset* cuenta con una cantidad de imágenes mucho menor que los anteriores, la calidad de las imágenes y la variedad en el tipo de animales presentes en el mismo lo hacen ideal para este trabajo.

2.2. Lindenthal Camera Traps Dataset

De entre los *datasets* públicos disponibles, el más completo y adecuado para este propósito se trata de **Lindenthal Camera Traps** presentado por [21], por lo que durante este proyecto se empleará una parte¹ del mismo. Este *dataset* está constituido por imágenes de animales en el Zoo de Lindenthal, Colonia, obtenidas con una cámara estéreo Intel RealSense D435 y en el que se incluyen imágenes infrarrojas (IR) con una profundidad de 8 bits y mapas de profundidad (D) con una profundidad de 16 bits en los que el valor de cada pixel representa la distancia en milímetros². Ambas modalidades se encuentran ya alineadas y sincronizadas, lo que permite un fácil acceso a la información de ambas modalidades. Además, el etiquetado de los animales se encuentra en formato COCO. En la figura 2.1 se muestra un ejemplo de una pareja de imágenes IR y D del *dataset*. La captura de estas imágenes se realiza desde el tejado de un establo, lo que permite una vista elevada de los animales en la escena. Además, los datos se capturan exclusivamente cuando se da algún tipo de movimiento, por lo que en el *setup* indicado se incluye un sensor infrarojo pasivo (PIR) que actúa como sensor de movimiento.



(a) Imagen IR



(b) Imagen de profundidad

Figura 2.1: Ejemplo de una pareja IR-D del *dataset* Lindenthal Camera Traps

2.2.1. Desglose del *Dataset*

El *dataset* consiste en doce videos para los que se ha etiquetado cada décimo fotograma, lo que supone un total de 412 pares IR-D con 1038 instancias etiquetadas en formato COCO que incluyen la máscara, la caja delimitadora de la misma, la categoría y un identificador único para cada uno de los animales. Las categorías de animales presentes en el dataset son **Deer**, **Goat**, **Donkey** y **Goose**, siendo considerado todo lo demás descriptable, es decir, parte de la categoría **Background**. En la figura 2.2 se muestra un desglose de la frecuencia de aparición de cada categoría en el dataset y se puede observar el amplio desbalanceo entre distintas categorías, por lo que cabe esperar que el modelo tenga ciertas dificultades a la hora de segmentar las categorías menos frecuentes como por ejemplo **Donkey**. Por otra parte, este *dataset* se mezclará aleatoriamente y se dividirá en

¹A pesar de que este *dataset* cuenta con 775 secuencias de video que incluyen información RGB durante el día o IR durante la noche, así como mapas de profundidad en ambos casos, tan solo fueron etiquetados para su uso como *Ground Truth* 12 videos nocturnos, de los cuales se etiquetería cada décimo fotograma.

²El sensor Intel RealSense D435 cuenta con la capacidad de obtener imágenes estéreo de forma activa empleando ambas cámaras infrarrojas mediante la proyección de un patrón infrarrojo en la escena, sin embargo al haberse construido este dataset a partir de imágenes nocturnas en las que dicho patrón sería visible al sensor IR, se optó por generar los mapas de profundidad D mediante el uso de la cámara estéreo en modo pasivo.

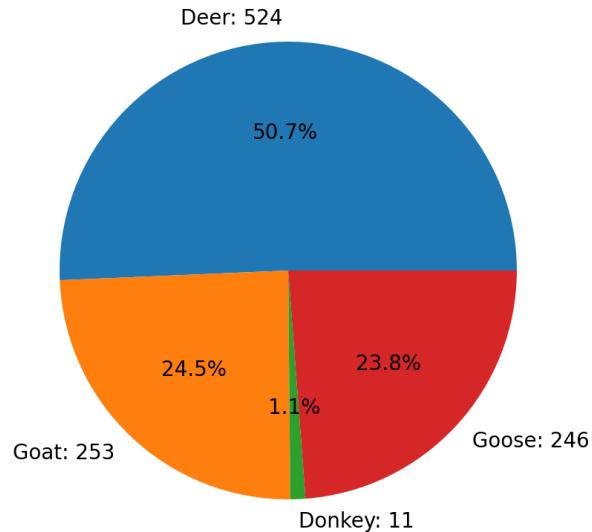


Figura 2.2: Desglose de la frecuencia de aparición de cada categoría en el dataset

dos partes: el 85 % de las imágenes se emplearán para entrenamiento y el 15 % restante para validación de los resultados con el fin de aplicar técnicas como el `EarlyStopping` o simplemente para una verificación manual del buen entrenamiento durante el proceso. En la figura 2.3 se muestra un ejemplo de un par IR-D con sus respectivas máscaras de segmentación extraídas del etiquetado COCO.



(a) Imagen IR etiquetada



(b) Imagen D etiquetada

Figura 2.3: Ejemplo de una pareja IR-D del *dataset Lindenthal Camera Traps*

Capítulo 3

Cross Modal Fusion

3.1. Descripción del Modelo

El primer modelo a implementar en este trabajo se trata de **CrossModalFusion (CMX)**, presentado por [22]. Este modelo propone una arquitectura para la segmentación semántica de pares RGB-X diseñada para aprovechar las características complementarias entre imágenes RGB y diversas modalidades como la térmica, imágenes de eventos o, como es de nuestro interés, imágenes de profundidad.

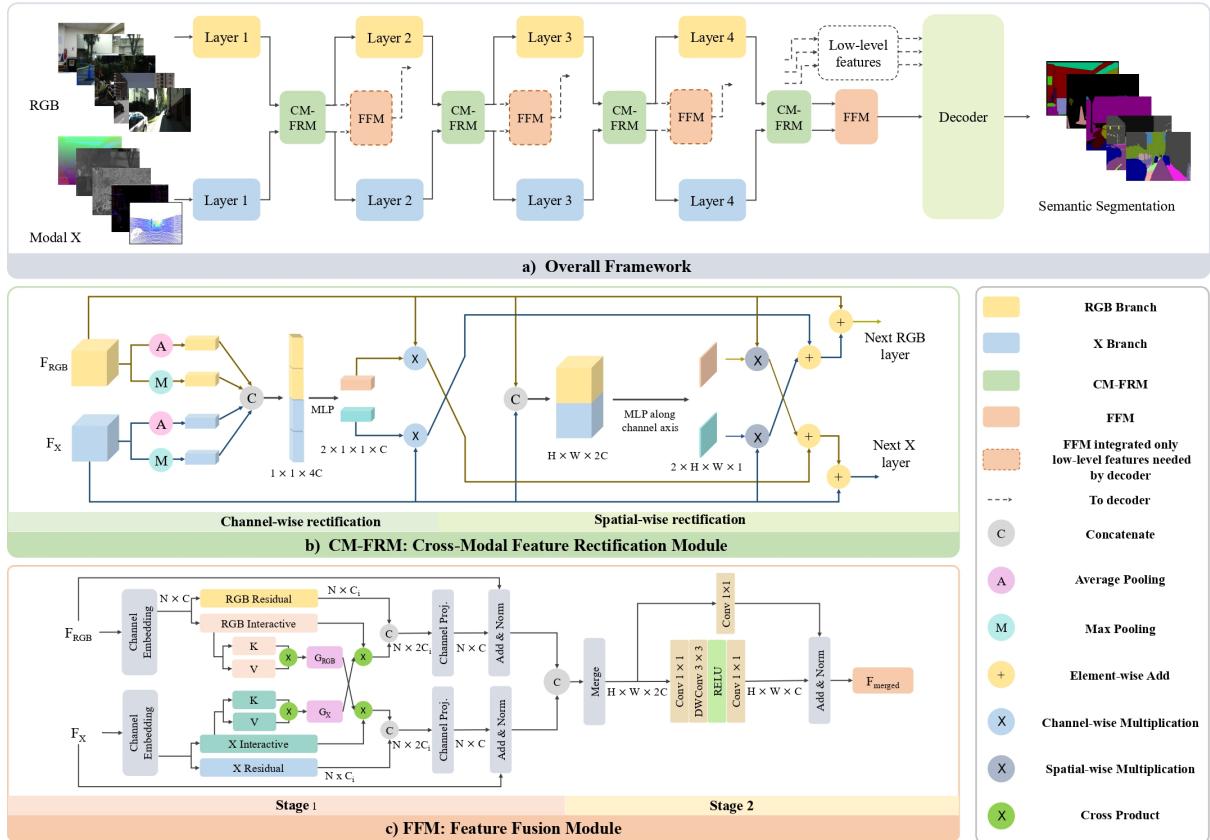


Figura 3.1: Arquitectura propuesta por [22]

La arquitectura de CMX (figura 3.1) hace uso de dos módulos clave para la interacción y fusión de características intermodales: el *Cross-Modal Feature Rectification Module* (CM-FRM), empleado para calibrar las características bimodales aplicando una rectificación

que sería capaz de mitigar ruidos y aprovechar las complementariedades entre modalidades, y el *Feature Fusion Module* (FFM), que somete a las características a un mecanismo de atención cruzada previo a su fusión. Este método de fusión de características supera con creces otras técnicas tradicionales como la fusión de características mediante concatenación o suma ya que permite un análisis más profundo de aquello que correlaciona ambas modalidades.

Además, a pesar de que el *backbone* preentrenado que se ofrece está entrenado en **ImageNet** y, por tanto, sus pesos están adaptados a RGB-RGB, se asegura la adaptabilidad del mismo a una gran variedad de modalidades RGB-X mediante *transfer learning*, por lo que se espera que el modelo sea capaz de adaptarse fácilmente a las necesidades de este trabajo, siendo IR un buen sustituto del original RGB debido a la similaridad de su dominio y D aquella modalidad extra D.

3.2. Técnicas de Pre-Procesado

Cuando vamos a usar una imagen como input para un modelo de segmentación, se debe tener en cuenta el formato en el que se desea representar la información que se le proporciona al modelo. En este caso, se cuenta con dos modalidades de entrada: IR y D, por lo que es interesante el análisis de las posibles técnicas de preprocessado que se pueden aplicar a cada modalidad para obtener un mejor resultado en la segmentación.

Así como aumentar la información que otorga el canal IR es una tarea complicada, ya que la imagen ya cuenta con una calidad aceptable y que difícilmente se mejorará con ninguna técnica de preprocessado (además de ser tan similar a aquellos pesos preentrenados en **ImageNet**), el canal D es el que más margen de operación presenta debido a la gran cantidad de ruido presente y, al contrario que con la modalidad IR, el que el modelo más dificultades tenga a la hora de extraer características debido a su escasa similaridad con los datos de pre-entrenamiento en **ImageNet**. Es por esto que se propone aplicar las técnicas de pre-procesado exclusivamente al canal D con el fin de aumentar la información disponible para el modelo e, idealmente, que este aprenda a emplear las características de forma más efectiva.

Existe una variedad de técnicas de pre-procesado que aprovechan los tres canales de entrada del modelo para así lograr un *input* capaz de aportar mayor cantidad de información al modelo de segmentación. A continuación, se presentan las técnicas de pre-procesado que se proponen para este trabajo.

3.2.1. HHA Encoding

Esta técnica, propuesta por [23] y siendo la técnica recomendada por los autores de CMX para el preprocessado de imágenes de profundidad, emplea los tres canales de la imagen de entrada para codificar las siguientes tres características:

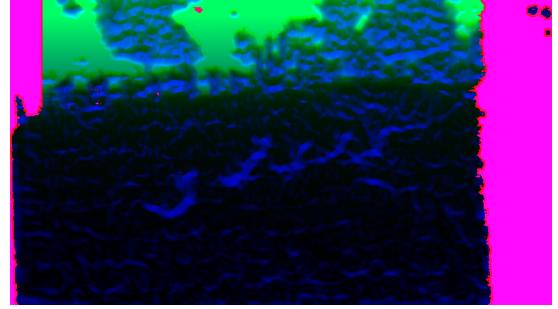
- Ángulo con respecto a la gravedad
- Altura sobre el suelo
- Disparidad horizontal

Esta colorización, además de representar características que difícilmente serían aprendidas por el modelo si no se codificaran en la imagen de entrada, aprovecha los tres canales

de entrada al *encoder*, por lo que más información es aportada en la entrada. A pesar de que esta colorización permite calcular las características mencionadas, es importante destacar que el cambio de dominio es mucho más radical que otras alternativas propuestas en esta sección, por lo que los pesos preentrenados en **ImageNet** no supondrán una base de entrenamiento tan robusta.



(a) Imagen de profundidad base



(b) Profundidad colorizada con HHA

Figura 3.2: Comparación entre la imagen de profundidad base y la imagen de profundidad codificada con HHA

Por otra parte, es importante destacar que el tiempo de cómputo necesario para realizar esta operación puede ser elevado (alrededor de 4.4 segundos en el equipo de testeo), por lo que la aplicación de esta técnica estaría limitada a casos en los que su aplicación en tiempo real no sea necesaria a no ser que se realizase una implementación extremadamente optimizada del algoritmo. Un ejemplo de esta técnica se muestra en la figura 3.2b, donde se puede observar una gran cantidad de ruido presente en la imagen HHA derivado del ruido presente en la imagen de profundidad original, lo que podría suponer una dificultad para el modelo a la hora de aprender las características de la escena ya que esta técnica suele reservarse para imágenes tomadas en interior, donde la cantidad de ruido es mucho menor.

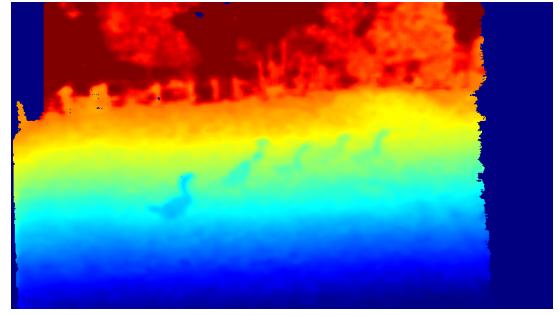
3.2.2. Colorización Jet

En el trabajo desarrollado en [24] se propone como alternativa a HHA la colorización de la imagen de entrada usando el esquema de color *Jet*. Esto, a pesar de no aportar información adicional sobre la imagen de profundidad original, sí que, en cierta forma, “ordena” la información de profundidad en los canales RGB, lo que implica dos posibles ventajas: la imagen resultante se asemeja mucho más a las imágenes **ImageNet** sobre las que el backbone ha sido preentrenado y, por otro lado, facilita el aprendizaje del modelo al asignar valores cromáticos distintos a profundidades diferentes, haciendo más explícitas las diferencias espaciales en la escena.

Como se puede ver en la figura 3.3b, cada píxel es asignado con un valor RGB dependiendo únicamente de su valor de profundidad siguiendo una tabla de colorización estándar *Jet* como la típicamente empleada en **OpenCV**. Este proceso, a pesar de no aportar información adicional como en el caso de HHA, es mucho más rápido y sencillo de implementar. Además, como se aprecia en la figura 3.3, la imagen resultante es mucho más clara y la existencia de ruido no resulta tan crítica. Como se menciona en [24], la aplicación de este preprocesado no necesariamente supone una mejora en el rendimiento del modelo, pero sí que alcanza unos resultados muy similares a cambio de un menor tiempo de cómputo.



(a) Imagen de profundidad base



(b) Profundidad colorizada con *Jet*

Figura 3.3: Comparación entre la imagen de profundidad base y la imagen de profundidad codificada con *Jet*

3.2.3. Colorización por Distancia

Considerando los resultados obtenidos empleando la colorización *Jet*, se propone en una técnica de colorización que, si bien puede no resultar tan visualmente atractiva para el ojo humano como su alternativa, podría codificar la distancia de forma más efectiva para el modelo. En específico, se propone establecer una relación lineal entre los valores de profundidad y el valor de los canales de la imagen final de forma que los objetos más lejanos serán más dominados por el canal R, mientras que los objetos más cercanos serán más dominados por el canal B pasando por el canal G en el rango intermedio. Un ejemplo de esta técnica se muestra en la figura 3.4b.



(a) Imagen de profundidad base



(b) Profundidad colorizada por distancia

Figura 3.4: Comparación entre la imagen de profundidad base y la imagen de profundidad codificada con *Jet*

De esta manera, esta técnica es también extremadamente rápida de aplicar al basarse en asignar un valor RGB a cada píxel dependiendo de su valor de intensidad. La interpolación de los valores de distancia a estos valores podría ser aprendida por el modelo de forma algo más efectiva que en el caso de *Jet* debido a la mayor simplificación en su distribución, ya que no fomenta tanto una imagen placentera a la vista sino una organización en los canales RGB más directa.

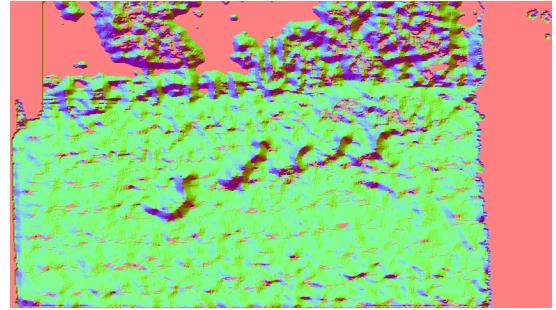
3.2.4. Normales

De nuevo, siguiendo los resultados presentados por [24], una mejor y más rápida solución de postprocesado podría tratarse de la codificación de las normales en la imagen de entrada. Esta técnica consiste en calcular el vector normal de la superficie de cada

pixel partiendo de la imagen de profundidad y codificarlo en los tres canales del *input* del modelo. Para comprender cómo se realiza esta conversión es necesario entender que la imagen de profundidad es tan solo una representación de los diferentes valores de distancia de los objetos en la escena, lo que supone una conversión directa de la imagen de profundidad 2D a una nube de puntos en un espacio 3D. Esto supone que cada uno de los puntos se puede asociar con un vector normal en XYZ que representa la orientación de la superficie en ese punto, siendo estos vectores fácilmente codificables en los canales RGB de la imagen de entrada.



(a) Imagen de profundidad base



(b) Profundidad colorizada por normales

Figura 3.5: Comparación entre la imagen de profundidad base y la imagen de profundidad codificada con normales

De esta forma, este tipo de codificación también aporta información adicional al modelo que difícilmente se podría aprender de otra forma, como puede ser la orientación de los objetos en la escena o incluso los bordes de los mismos (en decremento de la información de profundidad). Un ejemplo de esta técnica se muestra en la figura 3.5b donde se puede apreciar que el método, por razones similares a las de la colorización HHA, presenta cierto nivel de ruido en la imagen resultante.

$$\begin{aligned}\hat{n} &= \frac{(n_x, n_y, 1)}{\sqrt{n_x^2 + n_y^2 + 1}} \\ n_x &= \frac{\partial z}{\partial x}, n_y = \frac{\partial z}{\partial y}\end{aligned}\tag{3.1}$$

La imagen final se obtiene calculando las derivadas parciales respecto a x e y de los valores de profundidad en la modalidad D mediante la aplicación de un filtro Sobel y normalizando el vector resultante como se muestra en la ecuación 3.1.

3.3. Entrenamiento

El entrenamiento del modelo es realizado en un ordenador portátil con una tarjeta gráfica NVIDIA GeForce GTX 1650 con 4GB de memoria VRAM y un procesador Intel Core i5-10300H con 8 núcleos y 16 hilos, así como 16GB de memoria RAM. Además, como se ha mencionado en los anteriores apartados, se empleará un *backbone* pre-entrenado en ImageNet y se aplicará la técnica de *transfer learning* para adaptar el modelo a las características de los datos de entrada. El modelo se entrenará durante 500 épocas realizando validaciones periódicas con el objetivo de verificar el correcto aprendizaje de la red y evitar *overfitting*, además de aplicar *data augmentation* mediante el recorte de las imágenes

originales en imágenes más pequeñas con este mismo fin de aumentar (artificialmente) los datos de entrenamiento para evitar *overfitting* y fomentar el aprendizaje de características generales y no tanto de características específicas de los datos de entrenamiento.

| Entrenamiento | IR+D | Depth | InfraRed |
|----------------------|----------|----------|----------|
| Sin Modality Dropout | 73.902 % | 19.844 % | 60.597 % |
| Con Modality Dropout | 80.907 % | 66.099 % | 80.026 % |

Cuadro 3.1: Comparación de la aplicación de *modality dropout*

Por otra parte, tras observar que un entrenamiento típico puede generar que el modelo aprenda a emplear una de las modalidades de entrada de forma más efectiva que la otra, siendo esta última empleada como una suerte de refuerzo” que tan solo amplía la información de la otra modalidad pero no es capaz de aportar la suficiente información por si misma, y dado que sería ideal que el modelo sea capaz de desenvolverse sin problema cuando, por el motivo que sea, una de las modalidades no esté disponible, se opta por aplicar *modality dropout* en el entrenamiento. Esta técnica consiste en la eliminación aleatoria de una de las modalidades de entrada del modelo durante el entrenamiento. De esta forma, durante los primeros compases del entrenamiento, el modelo aprenderá a emplear las características de forma independiente en ambas modalidades, y a medida que éste avance, se fomentará el aprendizaje de las características de forma conjunta. Como se puede observar comparando las figuras (TODO y TODO) en las que se muestra las matrices de confusión para las distintas combinaciones de modalidades de entrada, el modelo tiende a aprender mucho más de la modalidad IR ya que es la que mayor información aporta, pero aun siendo así, gracias a la aplicación del *modality dropout*, el modelo puede realizar una segmentación adecuada incluso cuando tan solo se cuenta con la modalidad D. Los beneficios de implementar esta técnica se pueden observar en la tabla 3.1, donde se ve una clara mejora en la segmentación cuando falta una de las modalidades de entrada si el entrenamiento se ha realizado con *modality dropout*.

3.3.1. Implementación

Para este trabajo se implementará el backbone *Mix Transformer* (MiT) preentrenado en **ImageNet** y basado en el *framework SegFormer* (ambos presentados en [13]) en su versión más pequeña, el MiT-B0. Este *encoder*, junto al *decoder MLP* para segmentación, suman un total de 3.8M de parámetros entrenables. La implementación del modelo sobre el *dataset* del que se cuenta se realiza adaptando el código disponible en el repositorio de *GitHub* asociado al trabajo original ([22]), siendo los cambios realizados los necesarios para adaptar el *DataLoader* al formato COCO en el que se encuentran los datos de entrenamiento, la implementación de técnicas como el *Modality Dropout* y la validación durante el entrenamiento con el fin de evitar *overfitting*. Además, se respeta la implementación original en PyTorch por simplicidad y por facilidad de uso.

La función de pérdida empleada durante el entrenamiento será *CrossEntropyLoss* y el optimizador se trata de **Adam** con una tasa de aprendizaje de 10^{-6} dinámica con un momento de 0.9. El modelo se entrenará durante 500 épocas con un tamaño de *batch* de 2 pares IR-D (de nuevo, debido a los limitados recursos con los que se cuenta, en este caso una GPU con tan solo 4 GB de VRAM).

Además, la implementación del *modality dropout* se realiza de forma que durante las primeras épocas de entrenamiento exista un 50 % de probabilidad de que una de las dos

modalidades de entrenamiento esté ausente, siendo ese el caso, la modalidad IR faltará con una probabilidad del 65 %, mientras que la D (aquella cuyo aprendizaje de características se desea fomentar) lo hará con una probabilidad del 35 %. A medida que el entrenamiento avance, la probabilidad de que una de las modalidades falte disminuirá de forma lineal hasta llegar a un 0 % en la última época de entrenamiento.

3.4. Comparación de Técnicas

Las distintas técnicas anteriores presentan diferentes resultados tras su evaluación, tal y como se puede ver en la tabla 3.2 y en las diferentes matrices de confusión presentes en el anexo (TODO: referir al anexo!!!). En la tabla 3.2 se muestra la media entre las diferentes clases (**Background, Deer, Goat, Donkey y Goose**) del índice de Jaccard o *Index Over Union* (3.2) obtenida por cada una de las técnicas de pre-procesado propuestas para cada uno de las posibles casuísticas de ausencia o presencia de las modalidades IR y D. Aquí, se puede apreciar que la colorización Jet es la mejor opción cuando ambas modalidades están disponibles, aunque, por otro lado, se puede apreciar como éste no necesariamente es el caso cuando alguna de las modalidades se encuentra ausente.

| Técnica | IR+D | Depth | InfraRed |
|----------------------------|----------|----------|----------|
| Sin Pre-Procesado | 80.907 % | 66.099 % | 80.026 % |
| HHA Encoding | 80.620 % | 30.702 % | 80.293 % |
| Colorización Jet | 81.677 % | 63.274 % | 78.230 % |
| Colorización por Distancia | 81.422 % | 57.659 % | 79.027 % |
| Normales | 79.566 % | 47.340 % | 78.760 % |

Cuadro 3.2: IoU de las técnicas de pre-procesado propuestas

$$\text{IoU} = \frac{\text{Área de Superposición}}{\text{Área de Unión}} \quad (3.2)$$

Además, como se puede ver en las matrices de confusión que se encuentran en el anexo (TODO: referir al anexo!!!) todas estas técnicas comparten una precisión extremadamente alta al diferenciar entre clases, siendo la confusión con el fondo la principal fuente de error en la segmentación de las imágenes. Un ejemplo del resultado de la segmentación de una imagen empleando colorización Jet se muestra en la figura 5.1.

Por último, cabe destacar cómo las técnicas HHA y Normales presentan peores resultados cuando solo se cuenta con la modalidad D. Esto se debe principalmente a la cantidad de ruido presente en las imágenes resultantes, mientras que los otros métodos presentan resultados más consistentes debido a la simplicidad de su implementación, estos son gravemente influenciables por el ruido presente en la imagen original.

Capítulo 4

DFormer

4.1. Descripción del Modelo

Otro modelo relevante para este trabajo es **DFormer**, presentado por [25], que propone un marco innovador respecto al estado del arte en segmentación semántica de imágenes multimodales por la implementación de un modelo ya preentrenado en pares RGB-D (al contrario del habitual preentreno en pares RGB-RGB) con la reducción de costes computacionales que esto supone. Este modelo redefine la forma en que se aprenden las características conjuntas de RGB y Depth, permitiendo mejorar el rendimiento en tareas de segmentación semántica de imágenes RGB-D.

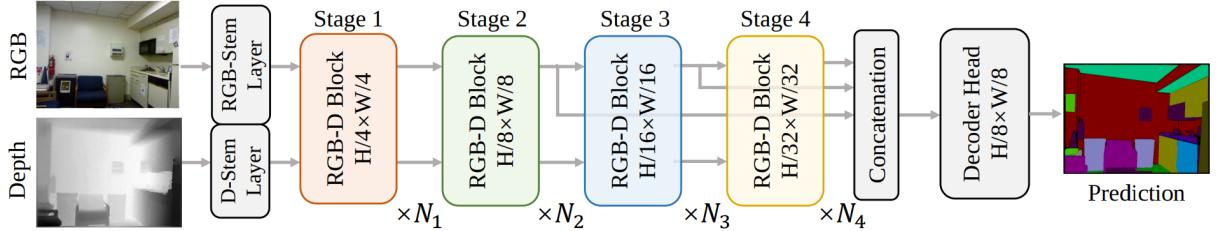


Figura 4.1: Arquitectura propuesta por [25]

La arquitectura de **DFormer** (figura 4.1) se basa en un ya típico esquema *encoder-decoder*, en concreto el *encoder* jerárquico en varias etapas está diseñado para generar características a diferentes escalas. Cada una de estas etapas incluye una serie de bloques RGB-D consistentes en un módulo de atención global (GAA) y un modulo de mejora local (LEA) que potencian la localización de objetos fomentando el entendimiento 3D de la escena y que capturan información local a partir de la información de profundidad (figura 4.2).

4.2. El pre-entrenamiento

Una importante mejora que se implementa en [25] consiste en el preentrenamiento en pares RGB-D. Estos pares son conseguidos aplicando técnicas de estimación de la profundidad sobre los datos presentes en **ImageNet**, lo que permite el pre-entrenamiento del modelo en un gran conjunto de datos ya estandarizado sin necesidad de recurrir al etiquetado de pares RGB-D o a datasets de menor tamaño. Este pre-entrenamiento en pares RGB-D estimados permitirá al modelo aprender a emplear y combinar las características

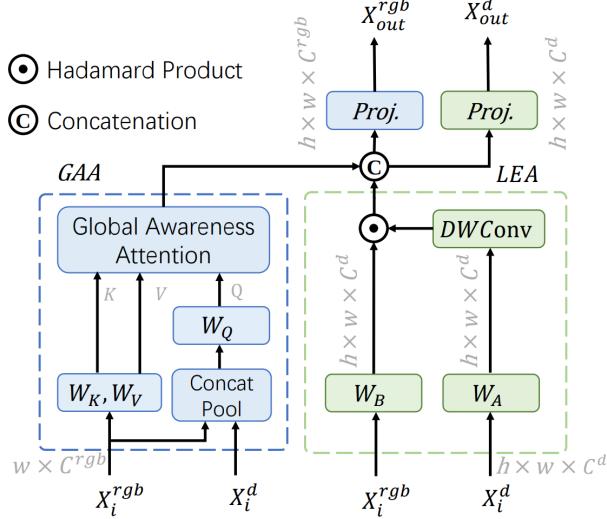


Figura 4.2: Bloques RGB-D propuestos por [25]

de ambas modalidades de forma más efectiva de cara al *transfer learning* sobre el que ya se contará con un *prior* de calidad, lo que es una ventaja sobre otros modelos preentrenados en pares RGB-RGB que deben aprender el cambio de dominio durante el entrenamiento final.

4.3. Resultados

El entrenamiento de este modelo es mucho más directo que en el caso de CMX ya que no se requiere de ningún tipo de preprocessado de la modalidad de profundidad para acercarla a la modalidad RGB original, por lo que se puede emplear directamente el canal D original sin la necesidad de alterarlo de ningún modo. Sin embargo, TODO: POR Q NO FUNCIONA??????

Capítulo 5

Resultados

5.1. Cross Modal Fusion

Empleando el modelo **SegFormer** presentado en [14] como punto de comparación para el desempeño de **CMX** (ya que este último se construye sobre el primero), se puede observar que el uso de técnicas multimodales supone una gran mejora en los resultados de segmentación, siendo el caso de mejor desempeño aquel en el que se aplica un preprocesado mediante colorización *Jet*, en la tabla 5.1 se puede ver como claramente el empleo de dos modalidades aumenta la calidad de las características aprendidas por el modelo, lo que deriva en una segmentación más precisa.

| Modelo | IoU medio |
|-------------|-----------|
| SegFormer | 66.894 % |
| CMX (IR-IR) | 73.770 % |
| CMX (IR-D) | 81.677 % |

Cuadro 5.1: Comparación de los resultados de segmentación entre un modelo **SegFormer** básico, **CMX** entrenado tan solo en la modalidad **IR** y **CMX** entrenado en pares **IR-D**

Así mismo, también cabe destacar que entrenando el modelo **CMX** en pares duplicados **IR-IR** (la misma imagen en ambas entradas) se desarrolla la tarea de segmentación mucho mejor que el modelo **SegFormer** a pesar de estar basados en la misma arquitectura. Esto se debe a que, además de que el modelo **CMX** resultaría ser mas grande al tener dos *encoders* en paralelo, éste extrae características de la entrada duplicada de forma simultánea para más tarde fusionarlas de forma inteligente, lo que, a pesar de contar con la misma información en ambas entradas, permite al modelo comprender mejor la escena. Siendo este el caso, no sería justa una comparación directa entre **SegFormer** y **CMX** multimodal dada la naturaleza más avanzada del segundo, por lo que únicamente se puede concluir la utilidad de la multimodalidad en la segmentación de imágenes si comparamos los resultados de **CMX** en pares **IR-D** con los de **CMX** en pares duplicados **IR-IR**, caso en el que se aprecia una gran mejora de la segmentación cuando existen diferentes modalidades a la entrada del modelo, verificando así la hipótesis inicial.



(a) Output del modelo



(b) Ground Truth

Figura 5.1: Resultados de la segmentación

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

En este trabajo, se ha explorado la aplicación de técnicas de segmentación multimodal en el ámbito con el fin de demostrar su eficacia en posibles aplicaciones de ciber-pastoreo. Se implementaron y evaluaron dos modelos principales: *Cross Modal Fusion (CMX)* y *DFormer*, ambos diseñados para aprovechar la información complementaria de las imágenes infrarrojas (IR) y de profundidad (D). Además, se evaluaron distintas técnicas de preprocesado para el primero de los casos, pretendiendo encontrar formas de adaptar métodos no específicos a esta modalidad, como la colorización de imágenes de profundidad o la codificación de otro tipo de características matemáticas extraíbles de la misma.

Los resultados obtenidos demuestran que la combinación de modalidades IR y D permite una segmentación más precisa y robusta en comparación con el uso de modalidades individuales. Entre las técnicas evaluadas, CMX con preprocesado mediante colorización *Jet* se destacó como una solución eficiente y de bajo costo computacional capaz de rendir mejor que sus alternativas, especialmente en escenarios donde ambas modalidades están disponibles. Por otro lado, la aplicación de *modality dropout* durante el entrenamiento resultó ser fundamental para aumentar la robustez del modelo ante la ausencia de una de las modalidades.

Aunque los resultados son prometedores, se identificaron algunas limitaciones, como la sensibilidad del modelo al ruido presente en las imágenes de profundidad dado a la naturaleza del sensor y el entorno en el que se desarrolla la acción. Estas limitaciones abren la puerta a nuevas líneas de investigación que se describen a continuación.

6.2. Trabajo futuro

A partir de las conclusiones extraídas de este trabajo, se plantean las siguientes líneas de investigación para mejorar y expandir los resultados obtenidos:

- **Optimización del preprocesado:** Investigar nuevas técnicas de preprocesado que reduzcan el ruido en las imágenes de profundidad, paralelización el procesado de imágenes para posibles aplicaciones en tiempo real, algoritmos de filtrado avanzado o técnicas de aprendizaje profundo específicas para la limpieza de datos.
- **Ampliación del dataset:** Utilizar datasets más diversos y extensos que incluyan diferentes tipos de animales, entornos y condiciones de iluminación, con el objetivo

de mejorar la capacidad de generalización de los modelos. Posible inclusión de datos sintéticos como se hace en [21].

- **Implementación en tiempo real:** Adaptar los modelos para aplicaciones en tiempo real mediante la optimización de los algoritmos mediante *frameworks* como TensorRT o ONNX y el uso de hardware especializado, como cámaras integradas con capacidades de procesamiento local.
- **Fusionar modalidades adicionales:** Integrar otras modalidades, como datos infrarrojos térmicos o cámaras de eventos, para enriquecer la información de entrada y mejorar la precisión en escenarios más complejos.
- **Evaluación en entornos reales:** Validar el desempeño de los modelos en situaciones reales de ciber-pastoreo, analizando su efectividad y adaptabilidad en aplicaciones prácticas.
- **Aplicación en imagen aérea:** Aplicación de las técnicas en imágenes aéreas para un seguimiento más amplio y cercano a un escenario real de ciber-pastoreo.

En resumen, este trabajo establece una base sólida para la aplicación de la segmentación multimodal en el ciber-pastoreo y abre nuevas oportunidades para la investigación y desarrollo de sistemas más robustos y eficientes en este campo.

Bibliografía

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- [2] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015. [Online]. Available: <https://arxiv.org/abs/1505.04597>
- [3] J. Huang, X. Li, T. Tan, X. Li, and T. Ye, “Mma-unet: A multi-modal asymmetric unet architecture for infrared and visible image fusion,” 2024. [Online]. Available: <https://arxiv.org/abs/2404.17747>
- [4] Z. Marinov, S. Reiß, D. Kersting, J. Kleesiek, and R. Stiefelhagen, “Mirror u-net: Marrying multimodal fission with multi-task learning for semantic segmentation in medical imaging,” 2023. [Online]. Available: <https://arxiv.org/abs/2303.07126>
- [5] R. Girshick, “Fast r-cnn,” 2015. [Online]. Available: <https://arxiv.org/abs/1504.08083>
- [6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” 2018. [Online]. Available: <https://arxiv.org/abs/1703.06870>
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [8] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” 2018.
- [9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2019. [Online]. Available: <https://arxiv.org/abs/1810.04805>
- [10] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” 2023. [Online]. Available: <https://arxiv.org/abs/1910.10683>
- [11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” 2021. [Online]. Available: <https://arxiv.org/abs/2010.11929>

- [12] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” 2020. [Online]. Available: <https://arxiv.org/abs/2005.12872>
- [13] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” 2021. [Online]. Available: <https://arxiv.org/abs/2105.15203>
- [14] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, “Segmenter: Transformer for semantic segmentation,” 2021. [Online]. Available: <https://arxiv.org/abs/2105.05633>
- [15] G. Nolan, “Sheepcounter dataset,” <https://universe.roboflow.com/sjy/sheepcounter-odziz>, sep 2024, visited on 2025-01-20. [Online]. Available: <https://universe.roboflow.com/sjy/sheepcounter-odziz>
- [16] BIRDSAI, “Birdsai dataset,” <https://universe.roboflow.com/birdsai/birdsai-duqdg>, jun 2022, visited on 2025-01-20. [Online]. Available: <https://universe.roboflow.com/birdsai/birdsai-duqdg>
- [17] S. Guillén-Garde, G. López-Nicolás, and R. Aragüés, “Detection and tracking of livestock herds from aerial video sequences,” *Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain*, 2021.
- [18] Z. Xu, T. Wang, A. K. Skidmore, and R. Lamprey, “A review of deep learning techniques for detecting animals in aerial and satellite images,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 128, p. 103732, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1569843224000864>
- [19] Alaska Fisheries Science Center, “A dataset for machine learning algorithm development,” 2021. [Online]. Available: <https://www.fisheries.noaa.gov/inport/item/63322>
- [20] Q. Hu, J. Smith, W. Woldt, and Z. Tang, “Uav-derived waterfowl thermal imagery dataset,” 2021. [Online]. Available: <https://doi.org/10.17632/46k66mz9sz.4>
- [21] T. Haucke and V. Steinhage, “Exploiting depth information for wildlife monitoring,” 2021. [Online]. Available: <https://arxiv.org/abs/2102.05607>
- [22] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen, “Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers,” 2023. [Online]. Available: <https://arxiv.org/abs/2203.04838>
- [23] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from rgb-d images for object detection and segmentation,” 2014. [Online]. Available: <https://arxiv.org/abs/1407.5736>
- [24] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, “Multimodal deep learning for robust rgb-d object recognition,” 2015. [Online]. Available: <https://arxiv.org/abs/1507.06821>
- [25] B. Yin, X. Zhang, Z. Li, L. Liu, M.-M. Cheng, and Q. Hou, “Dformer: Rethinking rgbd representation learning for semantic segmentation,” 2024. [Online]. Available: <https://arxiv.org/abs/2309.09668>