Segmentación Multimodal para detección de Animales

Jorge Urbón Burgos
777295@unizar.es

Supervisor: Rosario Aragües
raragues@unizar.es

Co-Supervisor: Jesús Bermúdez
bermudez@unizar.es

14 de diciembre de 2024

Resumen

Índice general

Capítulo 1

Introduction

Image segmentation has been a fundamental problem in computer vision tasks since the early days of the field. The main goal of image segmentation is to partition an image into multiple regions or objects

Capítulo 2

Literature Review

Capítulo 3

Metodología

3.1. Datos

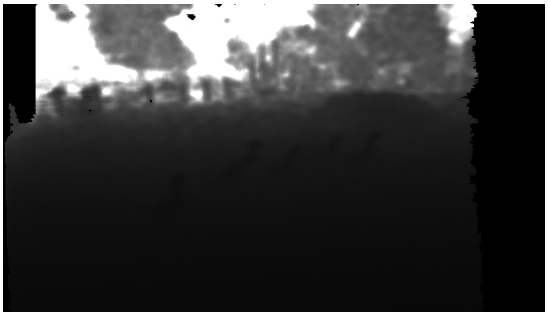
3.1.1. Lindenthal Dataset

Para este trabajo, emplearemos una parte[1] del *dataset* `Lindenthal Camera Traps` obtenido por [1]. Este *dataset* consiste en imágenes de animales en un entorno natural, obtenidas con una cámara estéreo `Intel RealSense D435` y en el que se incluyen imágenes infrarojas (`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 milimetros[2]. 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 3.1 se muestra un ejemplo de una pareja de imágenes `IR` y `D` del *dataset*.

[1] A 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 etiqueraría cada décimo fotograma.

[2] El sensor `Intel RealSense D435` cuenta con la capacidad de obtener imágenes estéreo de forma activa empleando ambas cámaras infrarojas mediante la proyección de un patrón infrarojo en la escena, sin embargo al haberse construido este dataset a partir de imagenes 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.

(a) Imagen `IR`



(b) Imagen de profundidad

Figura 3.1: Ejemplo de una pareja `IR-D` del *dataset* `Lindenthal Camera Traps`

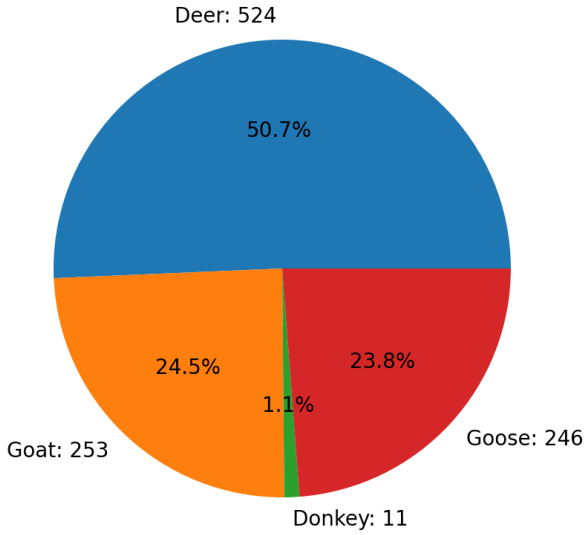
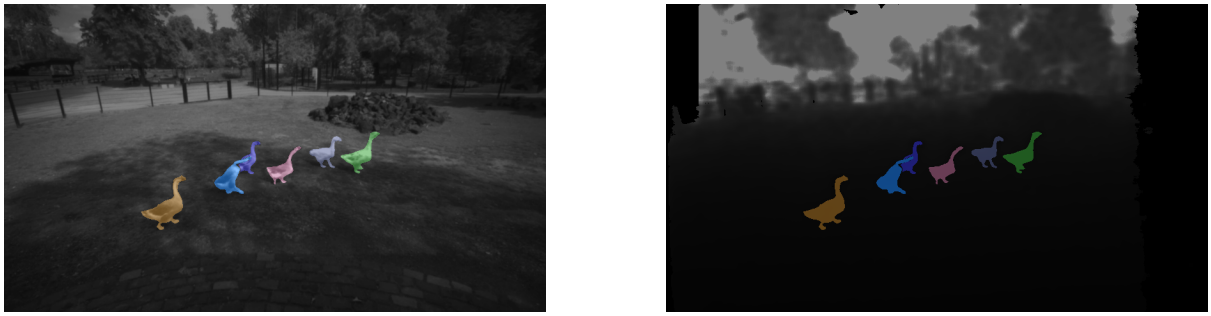


Figura 3.2: Desglose de la frecuencia de aparición de cada categoría en el dataset



(a) Imagen IR etiquetada

(b) Imagen de profundidad

Figura 3.3: Ejemplo de una pareja IR-D del *dataset* Lindenthal Camera Traps

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 el animal. Las categorías de animales presentes en el dataset son **Deer**, **Goat**, **Donkey** y **Goose** y en la figura 3.2 se muestra un desglose de la frecuencia de aparición de cada categoría en el dataset. Por otra parte, este *dataset* se dividirá en dos partes: el 85 % para entrenamiento y el 15 % restante para validación. En la figura 3.3 se muestra un ejemplo de un par IR-D con sus respectivas máscaras de segmentación.

3.2. Modelo

El modelo a implementar fue presentado por [2] y propone una arquitectura de fusión de características de forma ïnteractivaïmplementando rectificación de características inter-modal y bidireccional, así como atención cruzada secuencia a secuencia, lo que permite interacciones inter-modales más ricas y efectivas. La arquitectura propuesta por [2] se muestra en la figura 3.4.

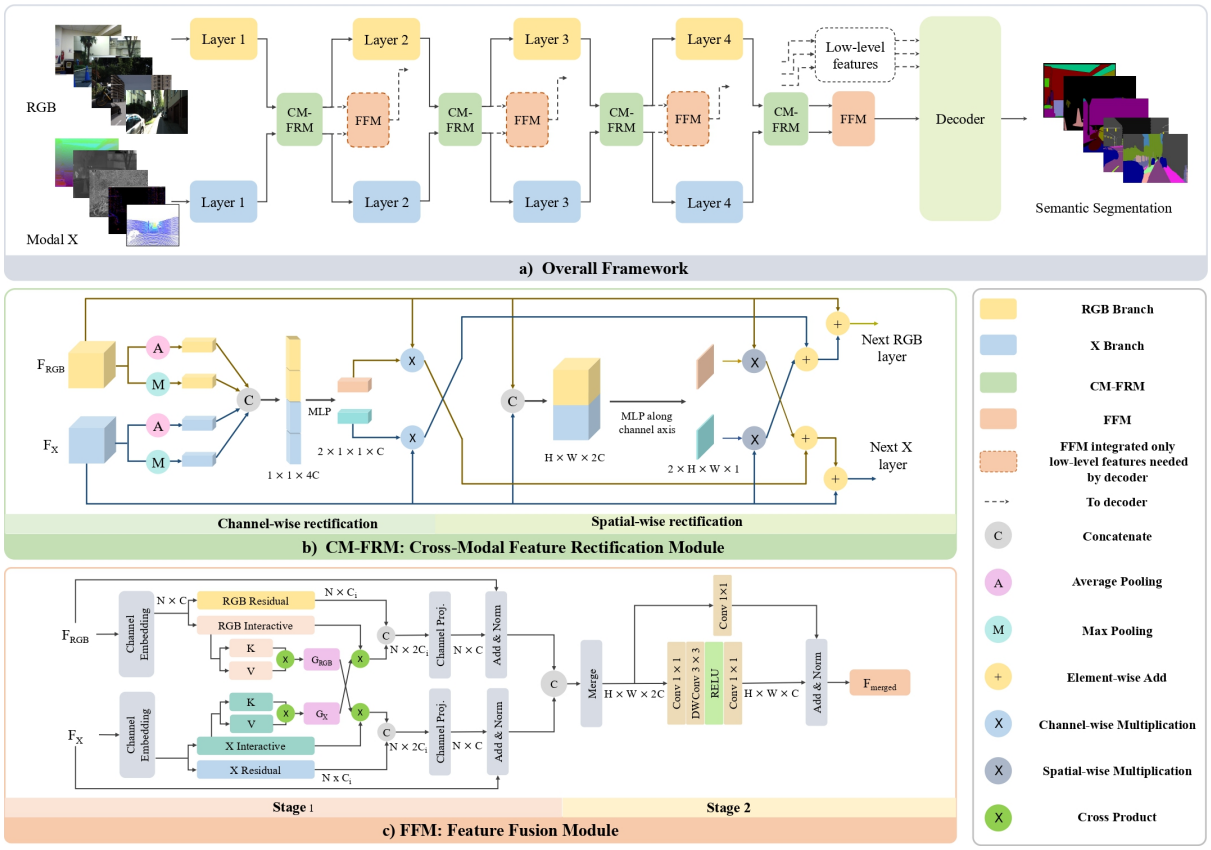


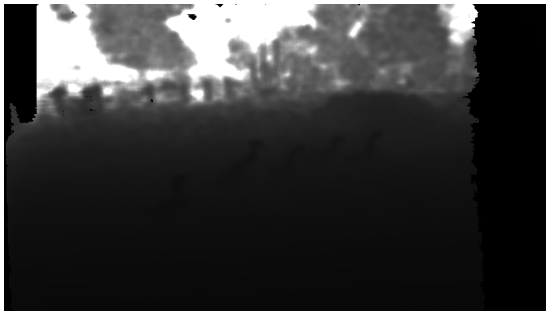
Figura 3.4: Arquitectura propuesta por [2]

3.2.1. Entrenamiento

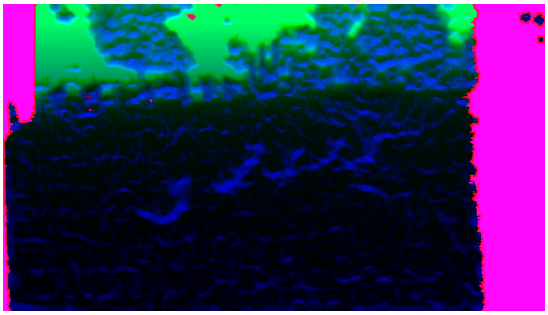
Para entrenar el modelo, se empleará un *backbone* pre-rentrenado en `ImageNet` y se empleará la técnica de *transfer learning* para adaptar el modelo a las características de los datos de entrada. Se empleará la función de pérdida `CrossEntropyLoss` y el optimizador `Adam` con una tasa de aprendizaje de 10^{-6} dinámica. El modelo se entrenará durante 500 épocas con un tamaño de *batch* de 2. Además, para fomentar el aprendizaje de las características inter-modales, se aplicará *modality dropout* con una probabilidad de 0.5 decreciente con el número de épocas. Por último, se analizan las diferentes propuestas en 3.2.2 con el fin de determinar qué técnica de pre-procesado es la más adecuada para el modelo.

3.2.2. 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 se deben considerar las distintas técnicas de preprocesado que se pueden aplicar a cada modalidad para obtener un mejor resultado en la segmentación. 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. Además, al consistir el *dataset* en imágenes obtenidas con una cámara estéreo `Intel RealSense D435`, la cantidad de ruido en las imágenes es considerable, por lo que toda técnica que ayude a destacar los objetos de interés será



(a) Imagen de profundidad base



(b) Profundidad colorizada con HHA

Figura 3.5: Comparison between the base depth image and the HHA encoded image

importante para el correcto desempeño del modelo. En este caso, debido a que la imagen `IR` ya cuenta con una calidad aceptable, se optará por aplicar las técnicas de pre-procesado exclusivamente a la imagen de profundidad `D`.

HHA Encoding

Esta técnica, propuesta por [3] usa los tres canales de la imagen de entrada para codificar las siguientes tres características:

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

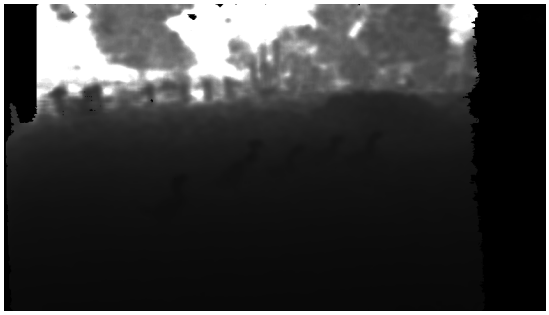
Estas colorización, además de implementar características que difícilmente serían aprendidas por el modelo si no se codificaran en la imagen de entrada, aprovecha los tres canales de entrada al *encoder*, por lo que los pesos ya preentrenados en *ImageNet* pueden ser empleados para una mejor comprensión de la escena.

Colorizacón Jet

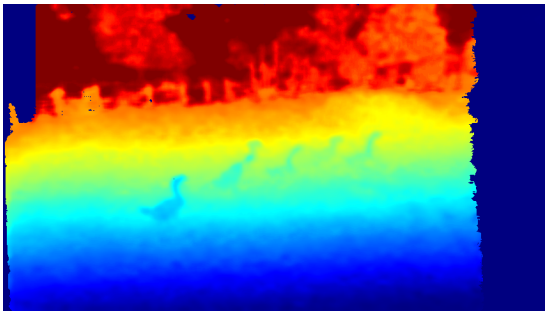
Como se propone en [4], otra posible técnica es la colorización de la imagen de entrada usando el esquema de color *Jet*, lo que implica la asignación de un valor *RGB* a cada píxel dependiendo de su valor de intensidad. Esto, de forma similar a 3.2.2, permite al modelo aprovechar sus pesos ya entrenados en *ImageNet* para identificar con mayor facilidad los objetos a segmentar.

Colorización por Distancia

Considerando los resultados obtenidos por Jet, se propone en una técnica de colorización que, si bien puede no resultar visualmente atractiva para el ojo humano, podría codificar la distancia de forma más efectiva para el modelo. Esta técnica también se basa en asignar un valor `RGB` a cada píxel dependiendo de su valor de intensidad, pero en este caso se determinará 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. La interpolación de los valores de distancia a estos valores podría permitir al modelo aprender de forma más efectiva la distancia de los objetos en la escena y así poder segmentarlos de forma más efectiva.



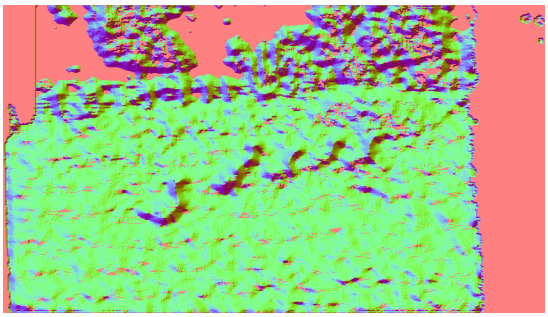
(a) Imagen de profundidad base



(b) Profundidad colorizada con *Jet*

Figura 3.6: Comparación entre la imagen de profundidad y la colorización *Jet*

(a) Imagen de profundidad base



(b) Normales codificadas en la imagen de profundidad

Figura 3.7: Comparación entre la imagen de profundidad y la codificación de las normales

Normales

Siguiendo lo propuesto por [4], una mejor y más rápida solución de postprocesado consiste en 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. Esto permite al modelo aprender características de la escena que de otra forma serían difíciles de aprender, como la orientación de los objetos en la escena.

Comparación de Técnicas

Las distintas técnicas anteriores presentan diferentes resultados a la hora de ser aplicadas al *input* del modelo. La tabla **??** muestra una comparación entre las distintas técnicas de postprocesado.

Técnica	Depth	HHA	Jet	Normales	Distance
Background IoU	99.831	99.807	99.808	99.807	99.810
Deer IoU	81.790	78.884	78.872	79.353	78.987
Goat IoU	78.947	79.282	80.102	74.442	79.442
Donkey IoU	74.109	75.623	76.267	76.166	74.541
Goose IoU	78.009	76.111	75.988	76.486	76.544
Mean IoU	82.537	81.941	82.208	81.251	81.865

Cuadro 3.1: Evaluación `IR-D` entre las distintas técnicas de postprocesado

Técnica	Depth	HHA	Jet	Normales	Distance
Background IoU	99.823	99.801	99.795	99.799	99.800
Deer IoU	81.361	78.797	78.063	78.830	78.457
Goat IoU	78.010	79.187	78.596	77.266	79.026
Donkey IoU	71.146	67.713	75.221	63.912	73.968
Goose IoU	76.326	74.934	73.149	74.297	74.089
Mean IoU	81.333	78.513	80.965	78.821	81.068

Cuadro 3.2: Evaluación `IR` entre las distintas técnicas de postprocesado

Técnica	Depth	HHA	Jet	Normales	Distance
Background IoU	99.598	99.530	99.561	99.552	99.568
Deer IoU	52.348	43.681	46.818	45.351	47.319
Goat IoU	70.057	62.111	67.850	62.962	67.403
Donkey IoU	65.208	64.397	64.213	72.583	71.497
Goose IoU	56.162	49.537	54.352	52.797	54.595
Mean IoU	68.675	63.851	66.559	66.649	68.077

Cuadro 3.3: Evaluación `D` entre las distintas técnicas de postprocesado

Capítulo 4

Results

Capítulo 5

Discussion

Capítulo 6

Conclusion

Bibliografía

[1] T. Haucke and V. Steinhage, "Exploiting depth information for wildlife monitoring," 2021. [Online]. Available: https://arxiv.org/abs/2102.05607

[2] 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

[3] 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

[4] 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