Transformers and Attention-Based Deep Networks Assignment - 2 Object Detection

Nesil Bor
2030336 - Graduate School of Informatics
DI725 – Transformers and Attention-Based Deep Networks
Ankara, Türkiye
nesil.bor@metu.edu.tr

***Abstract*—This study evaluates two transformer-based models, DETR and YOLOS-tiny, for object detection on the AU-AIR dataset, comprising aerial imagery with eight object categories. DETR was tested in baseline and optimized configurations, while YOLOS-tiny was fine-tuned with a modified classifier head. Both models were trained for 5–10 epochs and evaluated using COCO metrics (mAP@0.5:0.95). Results show near-zero mAP for both models, underperforming baselines (YOLOv3-Tiny: 30.22, MobileNetV2-SSDLite: 19.50). Label mismatches, insufficient training, and dataset complexity contributed to poor performance. Future work should focus on extended training and robust category mapping.**

***Index Terms*—Object Detection, Transformers, AU-AIR Dataset, DETR, YOLOS-tiny**

I. Introduction

Object detection in aerial imagery is essential for applications such as surveillance, urban planning, and autonomous navigation. The AU-AIR dataset, featuring eight object categories including Human, Car, and Truck, presents challenges due to small object sizes, diverse viewpoints, and complex backgrounds. Baseline models, specifically YOLOv3-Tiny with a mean Average Precision of 30.22 and MobileNetV2-SSDLite with a mean Average Precision of 19.50, provide performance benchmarks for this task. This study applies transformer-based models, DETR and YOLOS-tiny, aiming to improve detection performance through fine-tuning. An exploratory data analysis was conducted to understand dataset characteristics, informing preprocessing and model design. This report details the dataset, exploratory data analysis, methodology, evaluation, results, and limitations of the experiments.

II. Dataset

The AU-AIR [1] dataset comprises 32823 aerial images captured by Parrot Bebop 2 drones, accompanied by JSON annotations that specify bounding boxes in the format of x_min, y_min, x_max, y_max and category labels for eight classes, including Human, Car, Truck, Van, Motorbike, Bicycle, Bus, and Trailer. Pre-processing involved converting annotations to COCO format to ensure compatibility with standard evaluation tools. AU-AIR category IDs, ranging from 0 to 7, were mapped to COCO IDs, such as Human to 1 and Car to 3, for YOLOS-tiny compatibility. The dataset was divided into an 80% training set and a 20% validation set. Missing iscrowd fields in the annotations were addressed by setting them to 0. For DETR's optimized configuration, images were resized to 384x384 pixels to reduce memory usage. A custom dataset class was implemented to handle image loading and ensure proper scaling of bounding boxes to match resized images, mitigating potential misalignment issues during training and evaluation.

Exploratory data analysis was performed to gain insights into the AU-AIR dataset's characteristics, guiding model design and preprocessing decisions. The dataset contains 32823 images and 132031 annotations across eight categories. Analysis of class distribution revealed significant imbalances, with categories like Car and Human having the highest number of annotations, while Motorbike and Trailer were underrepresented, potentially impacting model performance on these classes. Image sizes were consistent, with a mean width and height of 1920 and 1080 pixels, respectively, and standard deviations near zero, indicating uniform resolution across the dataset. Bounding box sizes showed variability, with a mean width of approximately 200 pixels, mean height of 150 pixels, and mean area of 30000 square pixels, confirming the presence of small objects that are challenging for detection models. On average, each image contained four objects, with a standard deviation of 2.5, and some images had up to 20 objects. Visualizations, including histograms of image sizes, bounding box dimensions, and objects per image, were generated and saved as PNG files, highlighting the need for strategies like image resizing to handle small objects effectively.

III. Method

Two transformer-based models were implemented using PyTorch and the Hugging Face Transformers library, with experiments tracked on Weights and Biases for transparency. The codebase, including preprocessing, training, and evaluation scripts, is publicly available on GitHub with multiple commits across different days to meet version control requirements. DETR, based on the facebook/detr-resnet-50 model, uses a

ResNet-50 backbone and a transformer encoder-decoder architecture. It was tested in two configurations: a baseline setup trained for 5 epochs with a learning rate of 5e-5, batch size of 2, and full model training, and an optimized setup trained for 10 epochs with a learning rate of 1e-4, batch size of 4, frozen backbone, mixed precision training, and images resized to 384x384 pixels. YOLOS-tiny, based on the hustvl/yolos-tiny model, was fine-tuned for 6 epochs with a learning rate of 1e-4, batch size of 4, and gradient clipping with a maximum norm of 1.0 to stabilize training. Its classifier head was modified to a Linear layer mapping 768 input dimensions to 9 output classes, including a background class. A step learning rate scheduler reduced the learning rate by a factor of 0.1 every 3 epochs. Training was conducted on a GPU-enabled system, with losses logged to Weights and Biases and model checkpoints saved for analysis.

IV. Evaluation

Evaluation followed the COCO protocol, using the pycocotools library to compute metrics including mAP@0.5:0.95 as the primary metric, along with AP50, AP75, and per-class Average Precision. Ground truth and prediction data were formatted as COCO JSON files. DETR's baseline configuration applied a confidence threshold of 0.5, while the optimized configuration used a threshold of 0.001 to maximize recall, resulting in 3.28 million predictions. YOLOS-tiny used no confidence threshold, generating 42845 predictions to capture all potential detections. Results were compared against the baseline models, where YOLOv3-Tiny achieved a mean Average Precision of 30.22 with class APs ranging from 4.80 for Motorbike to 51.78 for Bus, and MobileNetV2-SSDLite achieved a mean Average Precision of 19.50 with APs ranging from 0.01 for Motorbike and Bicycle to 39.63 for Bus.

V. Results

During training, DETR's baseline configuration reduced its loss from 3.0062 to 2.8491 over 5 epochs, indicating slow convergence. The optimized configuration improved losses from 2.1709 to 1.5982 by epoch 5 but showed fluctuations, reaching 1.8217 by epoch 10, suggesting limited feature adaptation. YOLOS-tiny exhibited stable training, with losses decreasing from 1.2621 to 0.7026 by epoch 6, comprising classification loss of 0.2342, bounding box loss of 0.0450, and GIoU loss of 0.1217, with a validation loss of 0.7045. Evaluation results were poor for both models. DETR's baseline achieved a mAP@0.5:0.95 of 0.0003, with AP50 at 0.0014 and AP75 at 0.0001, detecting minimal instances of Human and Car with an AP of 0.0009 each. The optimized DETR configuration and YOLOS-tiny both recorded a mAP of 0.0000, with YOLOS-tiny producing invalid labels, such as category 91, not present in AU-AIR. Both models significantly underperformed the YOLOv3-Tiny and MobileNetV2-SSDLite baselines.

VI. Discussion

The near-zero mAP scores for DETR and YOLOS-tiny highlight substantial challenges in applying transformer-based models to the AU-AIR dataset. Exploratory data analysis indicated class imbalances, with Car and Human dominating annotations, and small bounding box sizes, which likely contributed to detection difficulties. Label mismatches were a critical issue, with DETR's optimized configuration outputting generic labels and YOLOS-tiny predicting invalid categories, likely due to errors in category ID mapping or classifier head configuration. The short training durations of 5 to 10 epochs for DETR and 6 epochs for YOLOS-tiny were insufficient for adapting to the dataset's small objects and aerial perspective, unlike the baseline models, which likely benefited from longer training and architectures optimized for small object detection. DETR's frozen backbone and image resizing to 384x384 pixels reduced computational demands but limited feature extraction, while YOLOS-tiny's full model training risked overfitting to incorrect classes. Dataset complexities, such as potential annotation inconsistencies and the inherent difficulty of detecting small objects in aerial imagery, further impacted performance. These findings suggest that transformer models require extensive fine-tuning and careful configuration to compete with traditional models like YOLOv3-Tiny and MobileNetV2-SSDLite in this context.

VII. Conclusion

DETR and YOLOS-tiny failed to achieve effective object detection on the AU-AIR dataset, with mAP scores near zero, far below the YOLOv3-Tiny mean Average Precision of 30.22 and MobileNetV2-SSDLite mean Average Precision of 19.50. Exploratory data analysis underscored class imbalances and small object sizes as significant challenges. Primary limitations included label mismatches, insufficient training duration, and dataset-specific complexities. To improve performance, future work should focus on extending training to 50 or more epochs, verifying and correcting category mappings, implementing data augmentation techniques like random crops and rotations, and exploring higher image resolutions to better handle small objects. Additionally, alternative architectures such as YOLOv8, which are designed for small object detection, may yield better results. The implementation and experiments are fully documented in public GitHub and Weights and Biases repositories, ensuring transparency and reproducibility of the work.

You can access my code and files from [4]. Also you can access my WANDB results from [5].

References

[1] AU-AIR Dataset, https://github.com/ozgurdemir/auairdataset
[2] N. Carion et al., "End-to-End Object Detection with Transformers," ECCV, 2020.
[3] Y. Fang et al., "You Only Look at One Sequence: Rethinking Transformer in Vision," ICLR, 2021.
[4] GitHub Repository, [https://github.com/adigew/DI725_Assignment2_2030336]
[5] Weights & Biases, [https://wandb.ai/adigew-middle-east-technical-university/di725-assignment2?nw=nwuseradigew]