Bird Detection with YOLO: Crow, Pigeon and Seagull

Project

JUILLARD Thibaut
thibaut.juillard@univ-lyon2.fr

GHIZLAN Moqim
moqim.ghizlan@univ-lyon2.fr

Université Lumière Lyon 2
M2 MALIA

March 14, 2025

Abstract

The objective of this project is to develop an object detection model based on a convolutional neural network (CNN) architecture to identify three bird species: crows, pigeons, and seagulls. We used YOLO, an efficient real-time object detection framework. A custom dataset was created using a Python script to collect images from Google and annotated via the Roboflow platform. The model was trained on this annotated dataset and evaluated using both quantitative and qualitative metrics to assess its performance. This report details the data collection and annotation process, the model configuration, and the results obtained.

1 Introduction

Object detection is a key domain in computer vision with applications ranging from surveillance to ecological research. In this project, we trained a YOLO (You Only Look Once) model to identify three bird species: crows, pigeons, and seagulls. These species are common in urban and coastal environments, and their automatic recognition can have applications in behavioral studies and environmental monitoring.

The objective is to design a model capable of identifying these birds in diverse images, including those with multiple species coexisting. The dataset was constructed by collecting images through an automated Python script and annotating them via the Roboflow platform. The trained model was evaluated using performance metrics and visual analysis of the results.

2 Dataset and Preprocessing

2.1 Dataset Collection

The dataset was built by collecting images of three bird species: crows, pigeons, and seagulls. A Python script was developed to automate image retrieval from Google, ensuring a variety of perspectives, lighting conditions, and backgrounds. All images were resized to a uniform resolution of 640×640 pixels to maintain consistency and optimize training efficiency.

2.2 Data Annotation and Augmentation

Once collected, the images were annotated using the Roboflow platform, where bounding boxes were manually drawn around each bird and labeled accordingly. This ensured that the model could learn to distinguish between different species accurately. Data preprocessing techniques were applied to normalize pixel values between 0 and 1, stabilizing the learning process. Augmentation techniques, including horizontal flipping, brightness adjustments, and random rotations, were implemented in Python to artificially expand the dataset and enhance model generalization.

2.3 Dataset Splitting

To enable effective training and testing, the dataset was divided into three subsets. The training set comprised 70% of the data and was used to optimize model parameters. The validation set (20%) was used for hyperparameter tuning to improve generalization, while the remaining 10% was reserved for testing. This structured approach ensured a robust evaluation of the model's performance.

3 Model and Training

3.1 Model Architecture

The YOLO object detection framework is renowned for its speed and efficiency in real-time applications. We used the YOLOv11n variant, which provides a balance between accuracy and computational efficiency. The model consists of 100 layers, with approximately 2.58 million parameters, and processes images at a resolution of 640×640. The architecture incorporates residual connections, a single detection head, spatial pyramid pooling, and attention mechanisms to enhance feature representation and improve detection accuracy.

3.2 Training Setup

The model was trained on a dataset of annotated images of three bird species. Training was performed on a Tesla P100 GPU using the PyTorch-based Ultralytics YOLO framework. The dataset was split into training (70%), validation (20%), and test (10%) sets. The training process spanned 50 epochs, with real-time monitoring of key losses, including bounding box loss, classification loss, and distribution focal loss (DFL). Metrics such as accuracy, recall, and mean average precision (mAP) at various IoU thresholds were logged throughout training.

3.3 Hyperparameter Tuning

Extensive hyperparameter tuning was conducted to optimize performance. We experimented with SGD and Adam optimizers, using learning rates of 0.01 and 0.0005, respectively. The batch size was varied between 16 and 32 to evaluate its impact on stability and memory consumption. The best results were achieved with a learning rate of 0.0005, Adam optimizer, and batch size of 32, yielding a mAP@50 of 0.956. The final model achieved a precision of 0.964, recall of 0.849, and mAP@50-95 of 0.715, demonstrating strong generalization to unseen data.

3.4 Evaluation Strategy

The trained model was evaluated on the test set using standard metrics such as precision, recall, and mAP. Post-training validation included confusion matrix analysis, precision-recall curves, and confidence score distribution to assess model behavior. The final model was saved for deployment and real-world testing, confirming its effectiveness in detecting and classifying bird species.

4 Evaluation

The trained YOLO model was tested on a separate dataset containing images of crows, pigeons, and seagulls. The evaluation focused on key performance metrics, including mAP@50, precision, and recall. The final evaluation on the test dataset yielded an overall mAP@50 of 91.1%, precision of 70.5%, and recall of 88.9%.

4.1 Performance Metrics

The model exhibited high classification accuracy across all classes. Per-class analysis showed that crows achieved the highest detection accuracy with a precision of 99.1% and recall of 100%, followed by pigeons with precision of 92.2% and recall of 81.8%. Seagulls performed slightly lower, with a precision of 95.2% and recall of 85.4%.

4.2 Precision-Recall Analysis

Precision-recall curve analysis demonstrated a strong trade-off between these two metrics. The PR curve confirmed that the model maintained high precision across a broad range of recall values, and the F1-score curve indicated a well-balanced detection capability.

4.3 Confusion Matrix

A confusion matrix was computed to analyze classification errors. The matrix showed minimal misclassifications, with most predictions aligning with ground-truth labels. However, some overlap between pigeons and seagulls was observed, suggesting occasional confusion due to similar visual features.

4.4 Confidence Score Distribution

The histogram of confidence scores revealed that most predictions were made with high confidence, indicating reliable detection. The majority of confidence scores were above 0.8, confirming the model's ability to distinguish between bird species effectively.

5 Discussion

Evaluation results confirm that the YOLO model effectively detects and classifies three bird species. The final mAP@50 of 91.1% demonstrates strong localization accuracy. While precision (70.5%) and recall (88.9%) indicate high detection capabilities, some false positives and false negatives remain. The confusion matrix highlights the model's ability to distinguish crows with near-perfect accuracy, though distinguishing pigeons from seagulls remains challenging.

Dataset quality significantly influenced performance. Automated image collection and augmentation improved generalization, allowing the model to adapt to diverse conditions. Future improvements could involve fine-tuning augmentation parameters, increasing dataset diversity, or experimenting with alternative architectures.

6 Conclusion

This project successfully developed a real-time object detection model for bird species identification using YOLO. Through automated dataset collection, structured training, and extensive evaluation, the model achieved high accuracy and demonstrated strong generalization capabilities. Future work could focus on improving dataset diversity, fine-tuning hyperparameters, and exploring ensemble methods to enhance classification accuracy further. The insights from this project contribute to the advancement of automated wildlife monitoring and species recognition.

7 References

References

[1] J. Redmon, A. Farhadi, *YOLOv3: An Incremental Improvement*, arXiv preprint arXiv:1804.02767, 2018.

[2] Ultralytics, *YOLOv11 Documentation*, `https://github.com/ultralytics/yolov11`, 2024.

[3] Roboflow, *Roboflow Platform for Dataset Annotation and Augmentation*, `https://roboflow.com`, 2024.

[4] DeepL, *DeepL Translator for Multilingual Reports*, `https://www.deepl.com`, 2024.

[5] OpenAI, *ChatGPT for Technical Writing and Report Structuring*, `https://openai.com`, 2024.

[6] C. F. Crispim Junior, *Computer Vision and Deep Learning Course*, Université de Lyon 2, 2024.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, *ImageNet: A Large-Scale Hierarchical Image Database*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[8] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, Advances in Neural Information Processing Systems (NeurIPS), 2019.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, *Microsoft COCO: Common Objects in Context*, European Conference on Computer Vision (ECCV), 2014.

[10] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A. A. Kalinin, *Albumentations: Fast and Flexible Image Augmentations*, Information, 2020.

[11] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016.