Multi Object Detection using YOLOv8 on PASCAL VOC2012 Dataset

Yu Long Wang
Department of Computer Science
CUNY Queens College
yulong.wang85@qmail.cuny.edu

Abstract

Multi Object detection is a fundamental challenge in computer vision since it requires accurate localization and classification of objects in images. These requirements makes it challenging, especially in real-world scenarios where images can contain multiple objects from different classes with different scales and orientations. This paper presents an implementation and evaluation of YOLOv8 [4], an object detection model, on a subset of the PASCAL Visual Object Classes' [2] dataset. We conducted experiments exploring various configurations of learning rates, loss function weights, and optimization parameters. The model was trained using SGD optimization for 100 epochs with consistent batch size and image dimensions. Our best-performing configuration, utilizing YOLO's default parameters, achieved a mean Average Precision (mAP50) of 0.775 and mAP50-95 of 0.537. Despite significant class imbalance in the training data, the model demonstrated robust performance across all object categories. These results validate the effectiveness of YOLOv8's default configuration for multi-class object detection tasks but also provide valuable insights into the impact of hyperparameter selection on model performance. Our findings suggest that while custom configurations can achieve competitive results, the default parameters offer optimal balance for object detection.

1 Introduction

1.1 Multi-Label Classification

Multi Object Detection is a fundamental task in computer vision that involves the localization, the position of the object by drawing boxes, and classification, identifying the objects in an image. The challenge of multi-object classification is much more complex compared to single label classification. Real world scenarios will more often then not, have multi object of different class in an image. As so, the need for more efficient and accurate detection systems increases to enable its utilization in various applications, such as medical imaging and real time detection.

Convolutional Neural Networks (CNNs) are deep learning architectures designed for processing grid-like data such as images. They use convolutional layers that slide filters across the input data to automatically learn patterns and features. In YOLOv8, the CNN backbone (CSPDarknet) serves as the primary feature extractor, using these convolutional operations to transform raw input images into feature representations for object detection.

Multi-label classification will have two challenges:

1. Label Imbalance: there will often be an imbalance of classes. In the PASCAL VOC the "person" class is the majority class compared to "train" or "motorcycle".
2. **Feature Difference:** Objects may appear in different sizes and location of an image.

1.2 YOLO and Dataset

The YOLO (You Only Live Once) has numerous models for real time object detection that is performant on speed and accuracy by treating it as a single shot regression approach.

This explores the implementation and evaluation of both YOLOv8 and YOLOv11 on the PASCAL VOC 2012 dataset, focusing on five specific object classes: bicycle, motorbike, person, cat, and train. These classes were chosen to represent a diverse range of detection challenges, including:

- varying object sizes (from small bicycles to large trains)
- different aspect ratios (vertical persons vs. horizontal trains)
- complex object structures (bicycles and motorbikes)
- deformable objects (cats and persons)

The primary objectives of this study are:

- evaluation of performance of YOLO models on a focused subset of object classes
- analyze the impact of various training configurations and data augmentation techniques
- assess practical applicability of these models for both accuracy and computational efficiency
- compare the strengths and limitations of different YOLO variants

The implementation utilizes the PASCAL VOC 2012 dataset with a subset that contains roughly 900 training images and validation images. YOLO calculates performance metrics such as, mean Average Precision (mAP), focusing particularly on mAP50 and mAP50-95 to provide an evaluation of detection accuracy.

2 Literature Review

Multi-label image classification is evolving significantly from simple binary classification to complex multi-object recognition systems. This complexity has driven the development of increasingly sophisticated approaches, from traditional computer vision methods to modern deep learning architectures.

2.1 Transformers in Multi-label Classification

Transformers were designed for natural language processing tasks, however recently emerged as a tool in computer vision. The application of Transformers to multi-label image classification can represented by splitting the image into patches and addressing the imbalance.

Liu *et al.* [5] presents Query2Label, which is a two- stage framework that uses transformers decoders to extract features and query the existence of class labels. For an input image x, the model predicts probabilities p = [p_1, ..., p_K] for K categories, where each $p_k \in [0, 1]$ represents the probability of category k being present. Unlike methods that rely on globally pooled features, Query2Label leverages the cross-attention mechanism inherent in Transformer decoders to adaptively extract features relevant to each class label.

The architecture consists of two primary stages:

- **Feature Extraction Stage:** A backbone network (CNN/ViT) processes the input image $x \in \mathbb{R}^{H_0 \times W_0 \times 3}$ to extract spatial features $F_0 \in \mathbb{R}^{H \times W \times d_0}$
- **Query Processing Stage:** Transformer decoders use learnable label embeddings $Q_0 \in \mathbb{R}^{K \times d}$ as queries to process these features through multi-head attention mechanisms

The innovations Liu *et al.* [5] demonstrated:

The transformer decoders query comparing the label embedding with features at each spatial location to generate attention maps and adaptive pooling by checking spatial features and select features of interest, enabling focus on different image regions for different class queries.

Multiple attention heads to enable feature extraction from different viewpoints of an object class.

This Transformer-based approach represents a more flexible and powerful framework for multi-label classification. Benchmarks on PASCAL VOC displays the effectiveness of complex relationships between multiple labels and image features, while maintaining computational efficiency through its two-stage design.

YOLO's CNN based approach to solving multi-label classification is with a single pass to offer computational efficiency. Both still accomplish the challenge of multi-label classification in different ways.

2.2 Multi-Class Attentional Regions

The Multi-Class Attentional Region (MCAR) framework [1] represents another approach to multi-label image classification, addressing the challenges of object localization and feature extraction through a two-stream architecture. Unlike YOLO's single-stage approach, MCAR employs a global-to-local strategy that focuses region analysis.

The framework consists of two primary streams:

- **Global Stream:** Processes the entire image through CNN to extract features
- **Local Stream:** Processes specific regions that it is interested in that were identified by the global stream

Multi-Class Attentional Region innovation is its attentional mapping and efficient interest in regions by creating attention on potential class and selecting the top-n to focus on the most promising class prediction, creating less overhead and computation.

3 YOLO Framework

YOLO (You Only Live Once) is a state of the art real time object detection algorithm that is very performant in terms of speed and accuracy.

The authors of YOLO framed the problem of object detection as a regression by spatially separating boxes and associating it with probabilities using a single convolution neural network (CNN). This implementation is a single-stage object detection framework that processes images in one forward pass, simultaneously predicting object classes and locations.

3.1 Framework

Given an input image $x \in \mathbb{R}^{H \times W \times 3}$, where H and W represent height and width respectively (640×640 pixels), the model predicts a set of bounding boxes and class probabilities predefined categories.

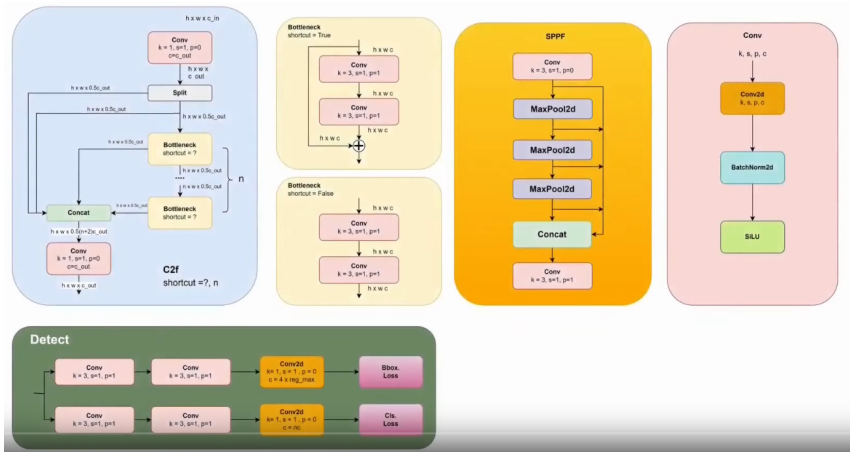

Figure 1: YOLOv8 Architecture details [3]

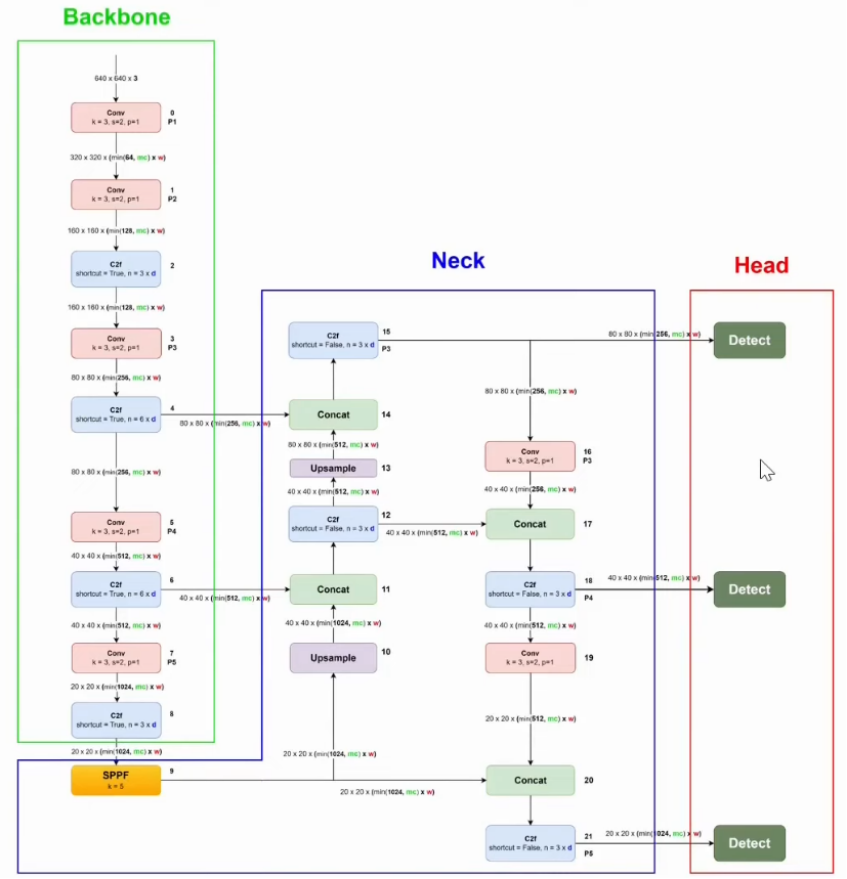

Figure 2: YOLOv8 Architecture [3]

The architecture consists of three main components:

Backbone 2: The backbone is made up of several convolution layer and C2f blocks 1 that progressively downsample the image to extract features.

- P1: 320×320×64 via initial Conv layer
- P2: 160×160×128 through second Conv
- P3: 80×80×256 after C2f and Conv operations
- P4: 40×40×512 with additional C2f processing
- P5: 20×20×512 at final backbone stage

Neck: Combines high resolution features with other features by upsampling and concatenation operations.

- Processes features through C2f modules
- Uses Concat operations for feature fusion
- Implements Upsample operations at 10 and 13
- Maintains three detection scales (80×80, 40×40, 20×20)

Head: Three parallel detection heads at different scales:

- Large object detection: 20×20 feature map
- Medium object detection: 40×40 feature map
- Small object detection: 80×80 feature map

3.2 Loss Function

The YOLOv8 uses three loss functions to optimize the model.

CIoU: This is one of the function loss that YOLOv8 uses. CIoU is for bounding boxes regression to improve localization accuracy. This is represented as **box loss** in the training.

VFL Loss (Varifocal Loss): This is used to address the class imbalance and measure the accuracy of the object in the prediction. This is represented as **cls loss** in the training.

DFL loss (Distribution Focal Loss): This is used help accurately predict object boundaries for the bounding box regression (CIoU).

4 Experiment

4.1 Description of dataset

In this study, we used the PASCAL Visual Object Challenge dataset [2], focusing on five object class: bicycle, motorbikes, people, cats, and train. The dataset was pre-processing and normalization to conform to the YOLO format, with all the images normalized to 640x640 pixels. Our training and validation set is approximately 900 images each. The class distribution with the 'person' class is more prevalent (1,000 images) while the 'cat' class having the lowest representation (200 images). To handle this imbalance I limited the amount of images of 'person' to around 200 also like the other classes.

4.2 Describe Performance Metric

The primary metrics included Mean Average Precision (mAP), which combines both precision and recall across different Intersection over Union (IoU) thresholds. Specifically, we utilized:

Box Precision (P): the accuracy of predicted bounding boxes by measuring the ratio of correct detections to total detections

Recall (R): measures the model's abilities to identify all objects, calculated as the ratio of correct detections to total objects

mAP50: Mean Average Precision at an IoU threshold of 0.5, providing a standard measure of detection quality

mAP50-95: Mean Average Precision across multiple IoU thresholds from 0.5 to 0.95 in steps of 0.05, offering more evaluation of detection quality

4.3 Performance on PASCAL VOC

The experiment was performed through multiple training iterations on PASCAL VOC 2012 [2], with the emphasis on hyperparameter tuning and validation metrics. The process focused on three components that significantly influence model performance: YOLO's optimizer, learning rate, and loss function configuration.

The learning rate was compared with high (0.1) and low (0.01) initial rates. Loss function components included box regression loss weight (ranging from 6.5 to 8.0), classification loss weight (0.8 to 1.5), and distribution focal loss weight (1.5 to 2.0). All experiments maintained consistent training parameters including 100 epochs, batch size of 16, and image size of 640x640 pixels.

4.4 Results

Our experiments revealed several key findings regarding model performance:

The initial high learning rate configuration (lr0=0.1) resulted in suboptimal performance (mAP50: 0.573, mAP50-95: 0.308), while lower learning rates (lr0=0.01) consistently produced better results. The default YOLO configuration achieved the highest performance (mAP50: 0.775, mAP50-95: 0.537).

Loss functions modifications showed that box loss values around 7.5 provided optimal localization accuracy, while classification loss weights (0.8-1.0) performs well. The distribution focal loss weight performed best between 1.7 and 2.0.

Class analysis shows consistent performance across vehicle classes (bicycle, motorbike), with the train class typically achieving the highest accuracy (mAP50: 0.801). Person detection, despite having the most instances, showed moderate performance improvements across experiments, while cat detection maintained robust performance throughout.

Experiment 1 = high learning rate

lr0=0.1	lrf=0.0001	box=7.0	cls=1.2	dfl=1.5
Overall mAP50:	0.573			
Class Precision:				
bicycle:	P=0.622	mAP50=0.532		
Motorbike:	P=0.691	mAP50=0.611		
Person:	P=0.636	mAP50=0.475		
Cat:	P=0.694	mAP50=0.645		
Train:	P=0.623	mAP50=0.601		

Experiment 2

lr0=0.01	lrf=0.0001	box=6.5	cls=1.5	dfl=1.8
Overall mAP50:	0.745			
Class Precision:				
Bicycle:	P=0.796	mAP50=0.709		
Motorbike:	P=0.777	mAP50=0.731		
Person:	P=0.776	mAP50=0.702		
Cat:	P=0.826	mAP50=0.817		
Train:	P=0.825	mAP50=0.768		

Experiment 3 = default YOLO configuration 3

lr0=0.01	box=7.5	cls=0.5	dfl=1.5
Overall mAP50:	0.775		
Class Precision:			
Bicycle:	P=0.861	mAP50=0.737	
Motorbike:	P=0.800	mAP50=0.789	
Person:	P=0.789	mAP50=0.735	
Cat:	P=0.816	mAP50=0.820	
Train:	P=0.873	mAP50=0.791	

Experiment 4

lr0=0.01	lrf=0.005	box=8.0	cls=0.9	dfl=1.7
Overall mAP50:	0.732			
Class Precision:				
Bicycle:	P=0.843	mAP50=0.696		
Motorbike:	P=0.815	mAP50=0.731		
Person:	P=0.829	mAP50=0.701		
Cat:	P=0.825	mAP50=0.770		
Train:	P=0.825	mAP50=0.762		

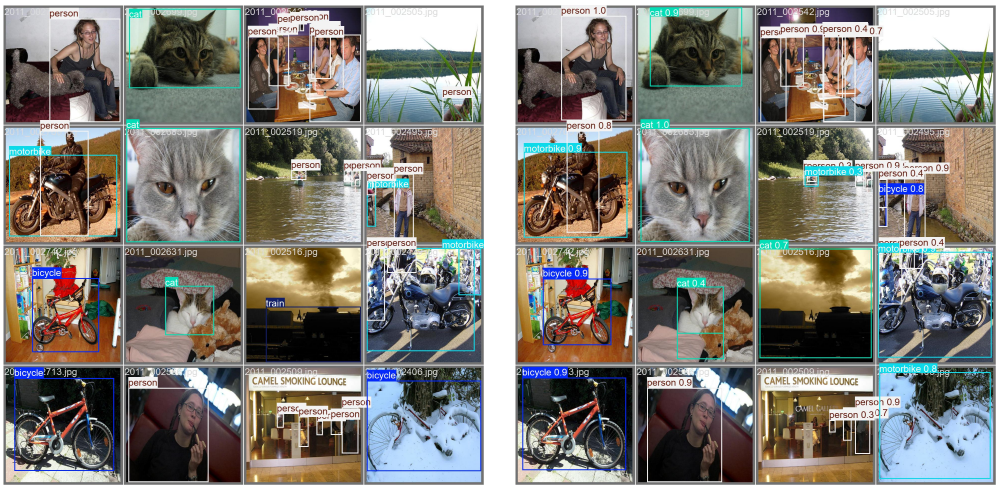

Figure 3: training from YOLO default, Left is labels, Right is prediction

These results demonstrate the importance of careful hyperparameter tuning in object detection, especially on learning rate and loss function. The experiments also validate the effectiveness of YOLO's default configuration while also providing insights to potential optimization for specific use cases.

5 Conclusion

Our experimental study on YOLO object detection revealed several findings regarding hyperparameter optimization and model performance. Through four experiments, we demonstrated that while custom configurations can achieve competitive results, the default YOLO parameters provide optimal performance for general object detection tasks.

Learning Rate Impact:

Lower initial learning rates consistently outperformed higher values. The default learning rate configuration achieved better results with mAP50 of 0.775. Aggressive learning rates degraded performance, as shown by Experiment 1's lower metrics.

Class-Specific Performance:

The model demonstrated class imbalance that objects like train and cat consistently achieved higher precision greater than 0.81, while objects like person, bicycle and motorbike showed more variance but acceptable performance.

References

[1] Hong-Yu Zhou Bin-Bin Gao. Learning to Discover Multi-Class Attentional Regions for Multi-Label Image Recognition. 30:5920–5932, 2021.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[3] Dr. Priyanto Hidayatullah. Yolov8 architecture detailed explanation - a complete breakdown. `https://www.youtube.com/watch?v=HQXhDO7COj8&list=LL`, Oct. 28, 2023.

[4] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.

[5] Xiao Yang Hang Su Jun Zhu Shilong Liu, Lei Zhang. Query2label: A simple transformer way to multi-label classification. *arXiv preprint arXiv:2107.10834v1*, 2021.