Analysis and Comparison of VGG16 and ResNet-50 for Image Recognition

Nemekhbayar Nomin[1],
[1]Computer Science Engineering Dep. of PNU.

1 Introduction

The paper introduces YOLO (You Only Look Once), a novel, unified approach to real-time object detection. Unlike traditional methods that repurpose classifiers for detection through complex pipelines, YOLO treats object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. It predicts bounding boxes and class probabilities in a single evaluation, achieving high-speed processing at up to 45 frames per second with the base model and 155 frames per second with the Fast YOLO variant [1].

2 Network Architecture

YOLO's architecture is built around a convolutional neural network (CNN) that divides the input image into an SxS grid. Each grid cell predicts multiple bounding boxes, their confidence scores, and class probabilities. The architecture consists of 24 convolutional layers, inspired by GoogLeNet but adapted for object detection. Instead of inception modules, YOLO uses a combination of 1x1 and 3x3 convolutional layers for feature extraction. At the end, two fully connected layers predict bounding box coordinates and class probabilities for each cell, forming a final output tensor of size SxSx(Bx5+C), where B is the number of bounding boxes per grid cell, and C is the number of classes.

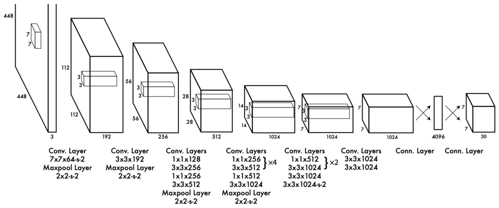

Figure 1: Network Architecture Design [1]

3 Unified Detection and Bounding Boxes

Each grid cell is responsible for predicting objects whose center lies within it, with B bounding boxes per cell. For each bounding box, YOLO predicts the (x, y) coordinates relative to the cell, the width and height relative to the image, and a confidence score, calculated as $\Pr(\text{Object}) \times \text{IOU}_{\text{truth}}^{\text{pred}}$, which reflects the probability of the object and the accuracy of the predicted bounding box. Additionally, each grid cell predicts conditional class probabilities for all possible classes, conditioned on the cell containing an object.

During inference, YOLO multiplies the conditional class probabilities by the bounding box confidence scores, resulting in class-specific confidence scores for each bounding box. This provides both the likelihood of each class and the bounding box's accuracy in one step.

4 Training and Loss Function

YOLO trains on full images, optimizing a multi-part loss function that combines classification, localization, and confidence errors. To address the imbalance between object-containing and non-object cells, the loss function uses weighted parameters, for λ_{coord} localization and λ_{noobj} for confidence in non-object cells. Additionally, YOLO applies square root scaling to the bounding box dimensions, mitigating the effect of small errors in large boxes.

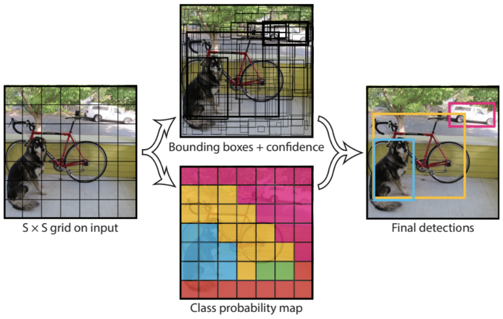

Figure 2: Model Detection and Prediction [1]

5 Comparison Analysis and Experiments

On the PASCAL VOC dataset, YOLO achieves 63.4% mAP, outperforming other real-time systems. YOLO generalizes well to different domains, maintaining accuracy on artwork datasets where other detectors falter. The system's speed is due to its single evaluation pipeline, which contrasts with multi-stage methods like R-CNN and DPM. YOLO, however, has more localization errors, particularly with small or clustered objects, due to its strong spatial constraints.

To further examine the differences between YOLO and state-of-the-art detectors, Error Analysis was made comparing YOLO to Fast R-CNN on the VOC 2007 dataset, using the methodology of Hoiem et al. Each prediction was classified based on errors such as Correct, Localization ($0.1 < \text{IOU} < 0.5$), Similar Class ($\text{IOU} > 0.1$), Other Class ($\text{IOU} > 0.1$), and Background ($\text{IOU} < 0.1$).

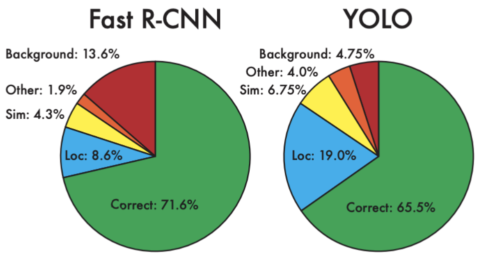

Figure 3: Error Analysis: Fast R-CNN vs. YOLO [1]

Results show that YOLO has more localization errors but fewer background errors compared to Fast R-CNN. In fact, YOLO is less likely to mistakenly classify background as objects, with Fast R-CNN producing almost three times as many background false positives.

6 Conclusion

YOLO's unified model simplifies object detection by eliminating complex pipelines and directly optimizing detection performance. Despite some limitations in precise localization, YOLO's speed and generalization ability make it a powerful tool for applications needing real-time detection and adaptability across domains, such as autonomous systems and robotics.

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2016. URL https://doi.org/10.48550/arXiv.1506.02640.