YoMAMBA:
A Mamba-based Backbone for YOLO

Dylan Lu

1 Overview

Nvidia's recent release of MambaVision[1] opens the door for an exciting new YOLO[2] architecture. I present YoMAMBA, a novel object detection model that outperforms traditional YOLOv1 models.

2 Implementation

For comparison purposes, I attach the same YOLO detection head to a ResNet18[3] backbone and a MambaVision-T-1k backbone. I then recreate the original YOLOv1 loss function, but using cross-entropy for classification instead of MSE.

2.1 Improvements & Results

By using residual connections, batch norm, Adam optimizer, and a carefully crafted learning rate scheduler, I greatly improve training speed and stability. I also extensively augment training data with random scaling, translation, saturation, and exposure—without which my model overfits and results in 0.9 train mAP but 0.12 val mAP.

After 140 epochs, YoMAMBA's mAP=0.34 versus YOLO ResNet's mAP=0.31 demonstrates the strength of MambaVision as a superior backbone. The mAP is calculated and averaged across all classes with an IOU threshold of 0.5.

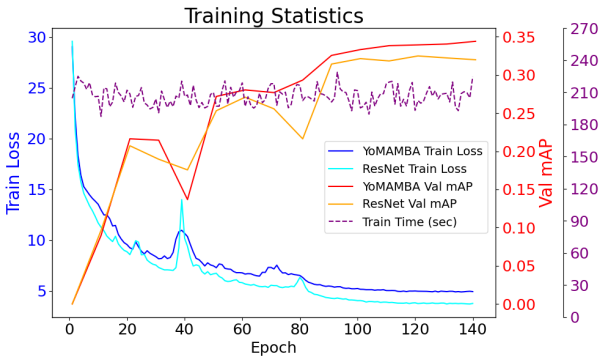
2.2 Dataset & Weights

All models were trained on Pascal VOC 2012, and pretrained backbone weights were obtained from PyTorch and HuggingFace.

2.3 Testing on CSIL

```
cd ~/cs-190i-s25/YoMAMBA
python test_image.py
python test_video.py
```

[1]Hatamizadeh and Kautz 2024.
[2]Redmon, Divvala, et al. 2015.
[3]He et al. 2015.



3 Lessons Learned

Model architectures are easy to reproduce, but in practice, training them is difficult. I initially implemented YOLOv2[4], but ran into issues with exploding gradients and poor mAP scores. I eventually realized that I could stabilize training by pretraining my backbone and then using a carefully tuned learning rate scheduler, both of which were invaluable when I pivoted to YoMAMBA. Most importantly, I learned how to organize a robust machine learning workflow from scratch, starting with data preparation, to model/loss definitions, and finally my training loops; this will undoubtedly prove useful for my future.

[4]Redmon and Farhadi 2016.