

Object Detection with YOLO

[[Project-Repo](#)]



Contents

- Objective
- Dataset
- Architectures
- Inference and Results
- Demonstration
- Conclusion and Future works

Objective

To Compose/Implement :

- YOLOv3 & YOLOv4 algorithm on MS COCO dataset
- Object Detection on Images
- Object Detection on Videos | Live-Cam
- Evaluation Metrics (to Futureworks)

Dataset

Microsoft COCO(Common Objects in Context) Dataset 2017

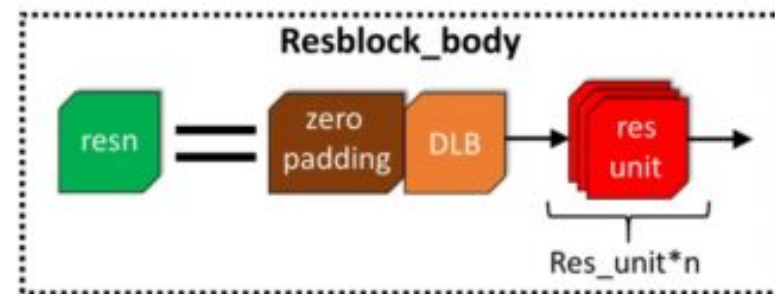
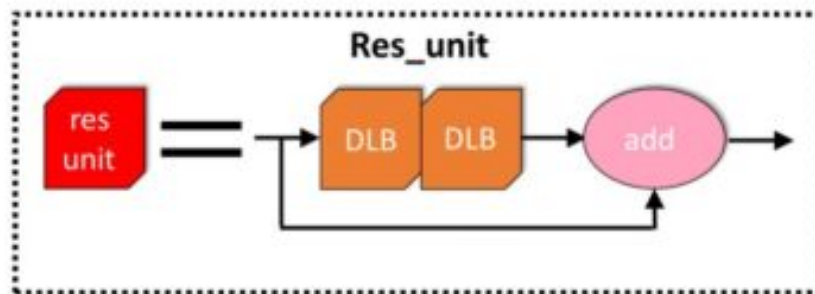
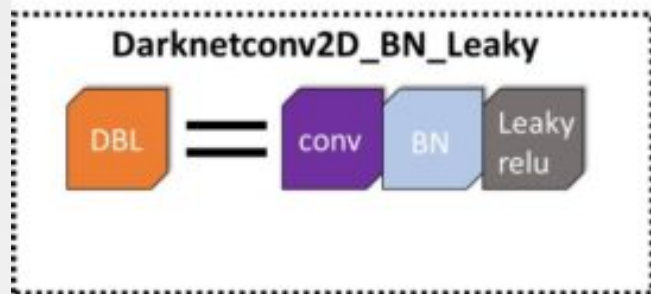
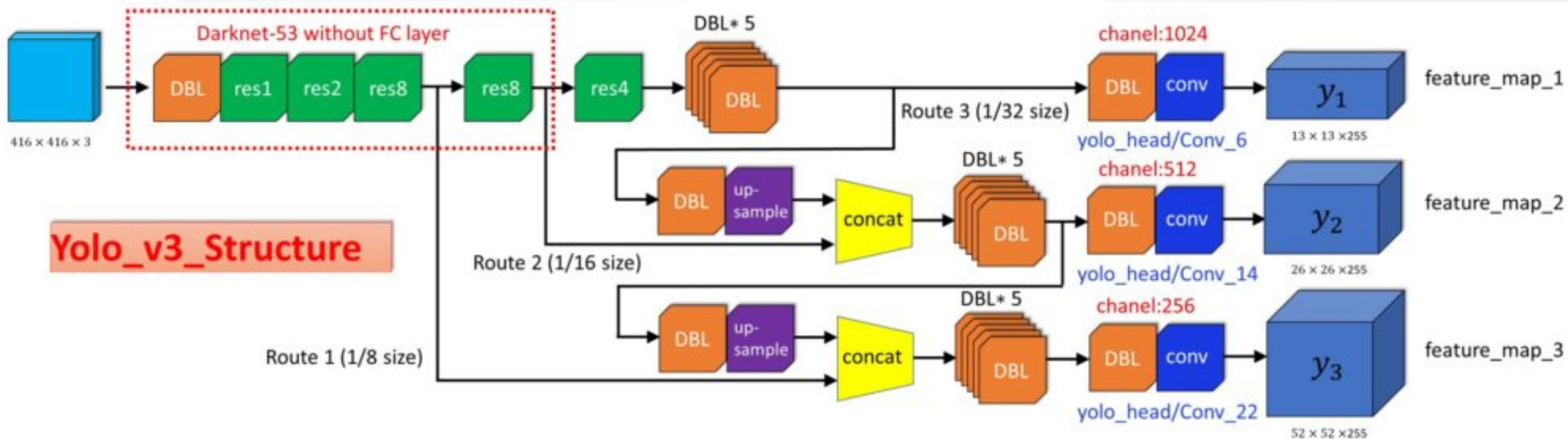
- 80 Common Object Classes
- Pre-trained Darknet weights and configs
 - YOLOV4 [[Weights](#)], [[Configs](#)].
 - YOLOV3 [[Weights](#)], [[Configs](#)].
- MS COCO [[Labels](#)].

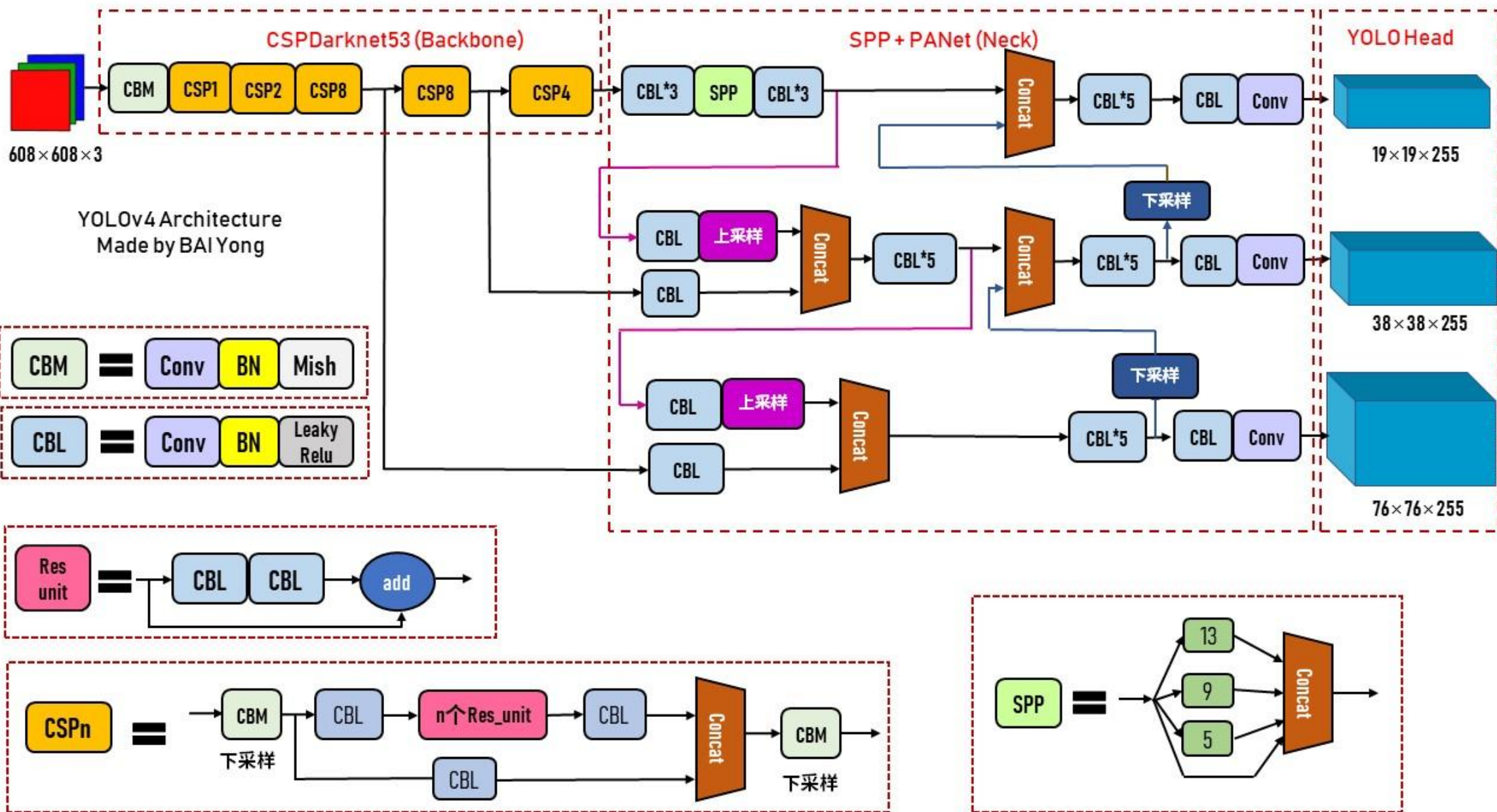
Architectures

Model Architecture, Tech Stack

- YoloV3, (PyTorch | OpenCV)
- Yolov4, (OpenCV)

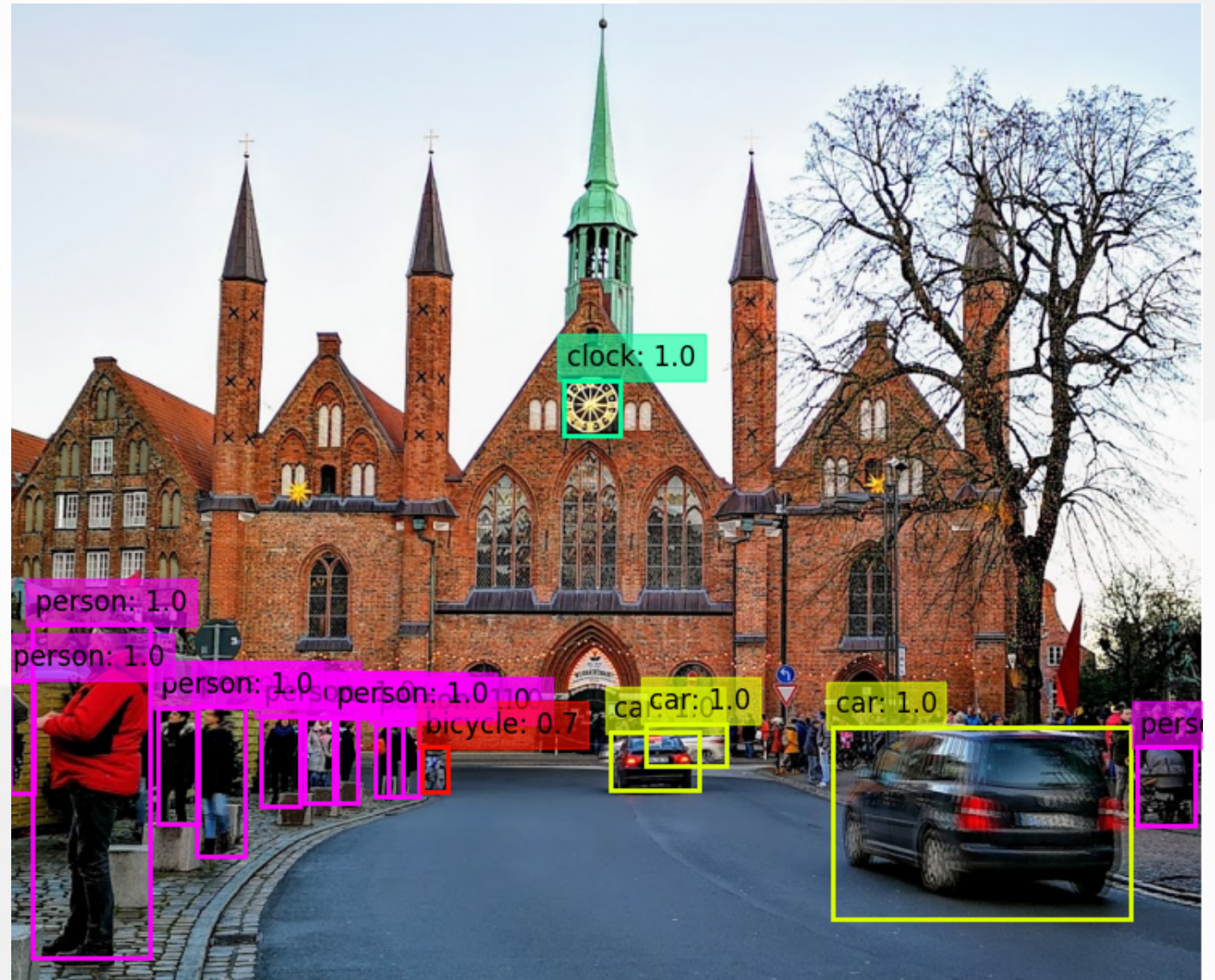
With Pretrained Darknet Backbone.





Results

- Boundary Box
- Confidence Score
- Image Inference [Pytorch]
- Video & Live-Cam Inference [OpenCV]



Demo:



Conclusion and Future works

“ “ Understanding and Implementation of Object Detection ” ”

Future Works

1. Evaluation Metrics for different model comparison
2. Run on CUDA GPU Build
3. Other Yolo Variants with Sensors fusion for 3D object detection.

References

YOLO Algorithms

- [[V4 Apr 2020](#)], [[V3 Apr 2018](#)], [[V2 Dec 2016](#)], & [[V1 May 2016](#)]

Darknet Reference

- [[Weights](#), [Darknet Script](#), [Configs](#)]

Misc.

- Image Yolo Architectures [[V3](#)], [[V4](#)]