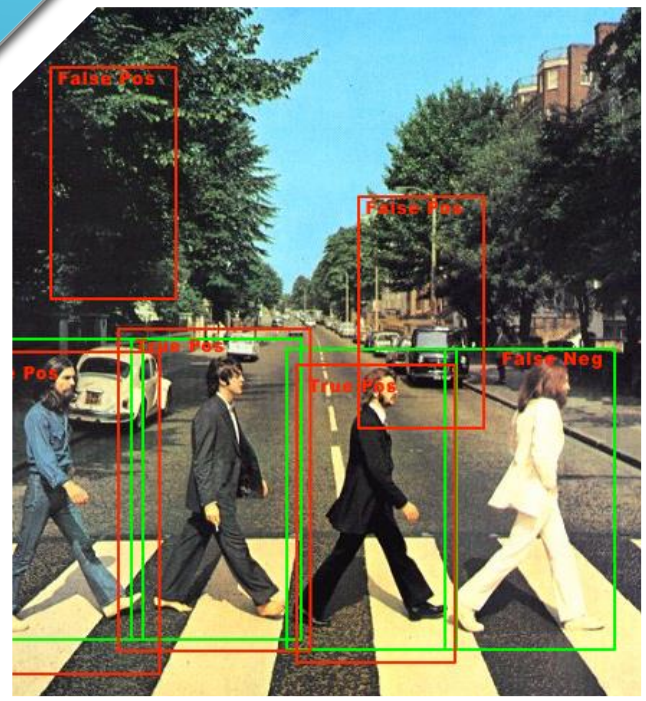
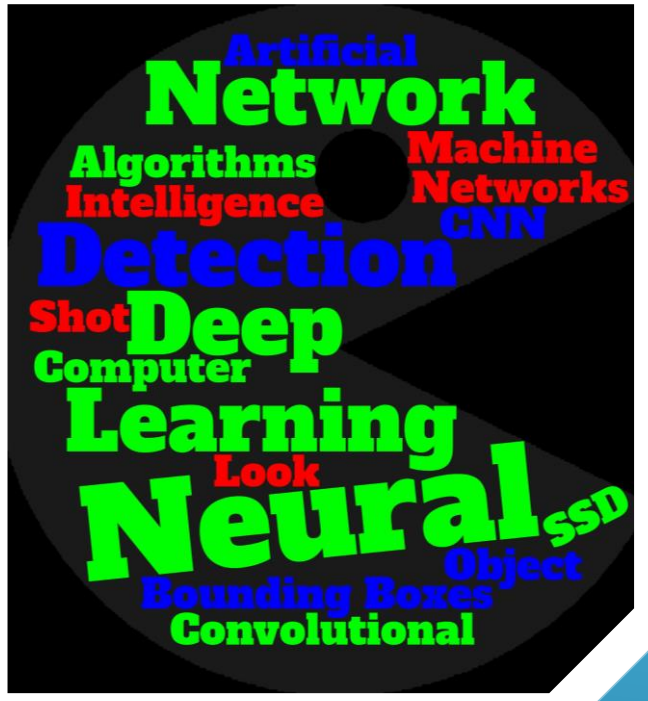
FINAL PRESENTATION

COMPARISON OF OBJECT DETECTION ALGORITHMS: YOLO V3 VS. SINGLE SHOT DETECTION

By Brandon Gilbert





ABSTRACT
(KEYWORDS)

- YOLO v3 – You Only Look Once
- SSD – Single Shot Detection
  - Both are Single Shot Object Detection Algos
  - Make predictions at test time
  - Both use their own versions of Convolutional Neural Networks(CNN)
  - CNN is a deep learning concept that usually relates the computer vision. CNN help power image recognition and related tasks using node layers associated with config files.

INTRODUCTION

LITERATURE REVIEW

- There are a vast number of studies/comparisons between the different object detection Algos.
- These studies focus on mean average precision(mAP), which is an advance metric measuring accuracy, and framerate as a metric for determining speed.

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000 × 600
Fast YOLO	52.7	155	1	98	448 × 448
YOLO (VGG16)	66.4	21	1	98	448 × 448
SSD300	74.3	46	1	8732	300 × 300
SSD512	76.8	19	1	24564	512 × 512
SSD300	74.3	59	8	8732	300 × 300
SSD512	76.8	22	8	24564	512 × 512

Speed is measure with a batch size of 1 or 8 during inference.

LITERATURE REVIEW (CONTINUED)

- Using mAP to compute the Average Precision is a more precise method of measuring accuracy then what I will be using in this study.
- However, the data extracted, and the implementation is quite extension
- Comparing Algos by framerate also give a better understanding by which Algo performs better
- In this study, I am working on this implementation but have ran into some roadblocks due to Nvidia's CUDA association with OpenCV

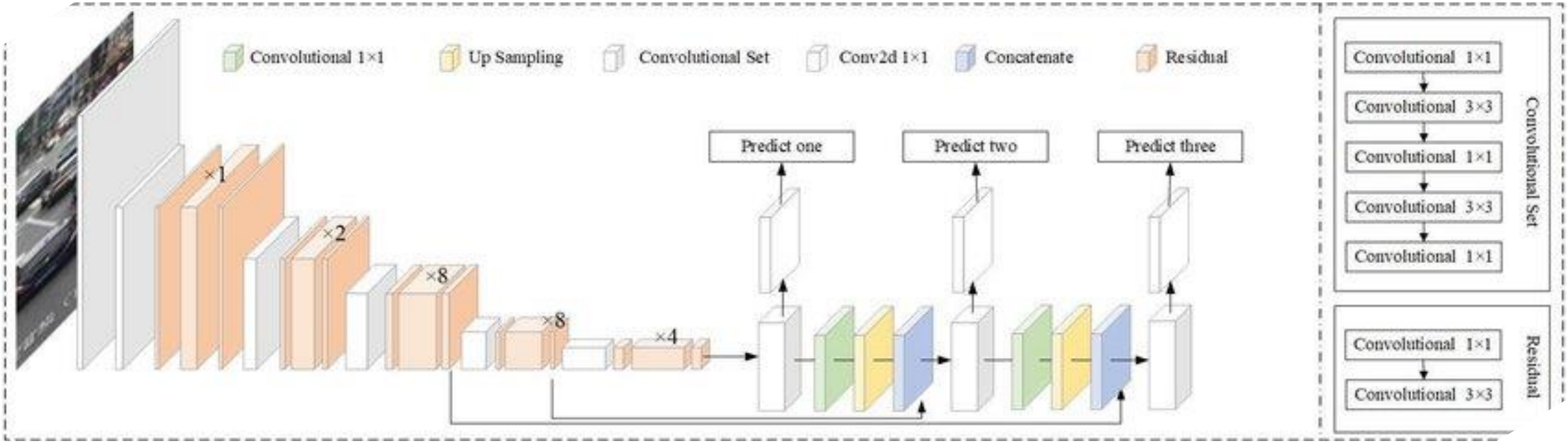
- Compare two state-of-the art single shot object detection models
- Use same data set – COCO
- Using same confidence threshold of 60%
- Same 11 second dashcam video(330 Frames) comparing by frame not frame rate

METHODOLOGY

YOLO VS SSD

- YOLO(You only look once): State of the art object detection algorithm runs better on GPU unless you run YOLO tiny(Not accurate, but faster and easier to run). Takes the whole image at test time and makes predictions with a single network evaluation unlike other systems like R-CNN which run thousands for an image.
- SSD(Single Shot Detector): Uses a convolutional network on image only once and computes a feature map. Has a good balance between speed and accuracy. Like YOLO, SSD is faster than R-CNN by taking a single shot to detect multiple object per image compared to a two-shot common(RPN) approach.

YOLO V3 USING DARKNET-53 AS BACKBONE NETWORK PRODUCING THREE SCALE PREDICTIONS

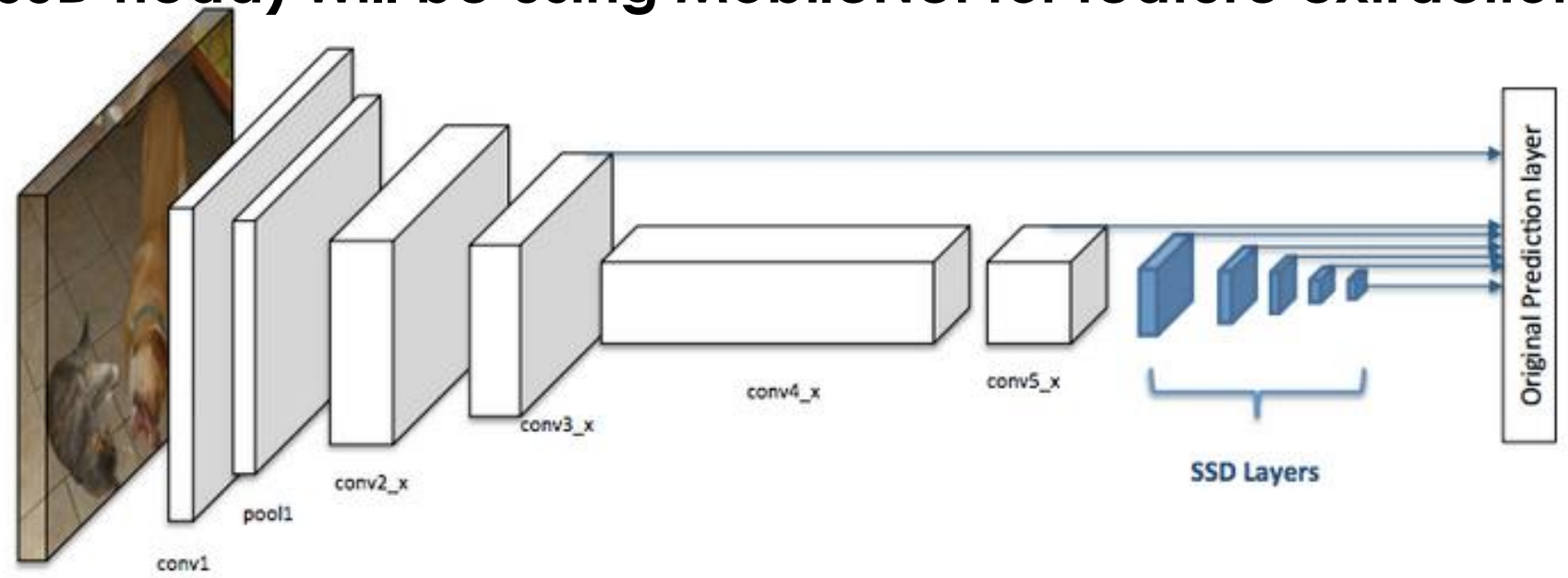


TAKES ORGINAL IMAGE AND GOES THROUGH A SERIES OF CONVOLUTIONAL LAYERS AND RESIZING THE IMAGE IN EACH SET TO GET THREE PREDICTION VALUES

Detection layer:	82	94	106
Network strides	32	16	8
Image Size: 416 x 416 (original input image)	13x13	26x26	52x52
Detection of object size	Large	Medium	small

YOLO V3

Architecture of a convolutional neural network with SSD detector(white boxes are the CNN backbone, blue is the SSD head) Will be using MobileNet for feature extraction



TAKES THE IMAGE AND RESIZES THE IMAGE WITH BOUNDING BOXES OF DIFFERENT ASPECT RATIOS, THEN THE SSD NETWORK COMBINES PREDICTIONS FROM FEATURE MAPS CREATED AT DIFFERENT RESOULTIONS TO DETERMINE THE OBJECT CLASSIFACTION --- In total, SSD makes 8732 predictions using 6 layers.

IMPLEMENTATION

- Python 3.9
  - Using PyCharm IDE on Windows 10
  - Using the following Dependencies:
    - Open CV – CV2: library for computer vision
    - NumPy – library for large multi-dim arrays
    - Pickle – for object storing into file
    - Matplotlib – for charting
    - Pandas – for plotting line graphs

IMPLEMENTATION CONTINUED

- Program contains three classes
  - Yolov3.py
    - Main algo run file
  - SSD_MobileNet.py
    - Main algo run file
  - Data.py
    - Stores DetectedObject using namedtuple for both algos
  - analyze.py
    - Main class to run analysis and produce charts

IMPLEMENTATION CONTINUED

1. This study, the test begins by running each algo using the same mp4 dashcam video
2. Then the data gets extracted into files using Data.py and pickle
3. The analysis phase begins by running analyze.py that produces the graphs using the data provided by above

EXPERIMENTAL SETUP AND DATASET

- This study will use COCO dataset to act as a base model for object classification
- Data collected will be the following:
  - Number of objects detected (per frame)
  - Confidence levels (in percentages)
  - Class IDs (name of objects)
  - Framerate (once fixed)

DATASET:

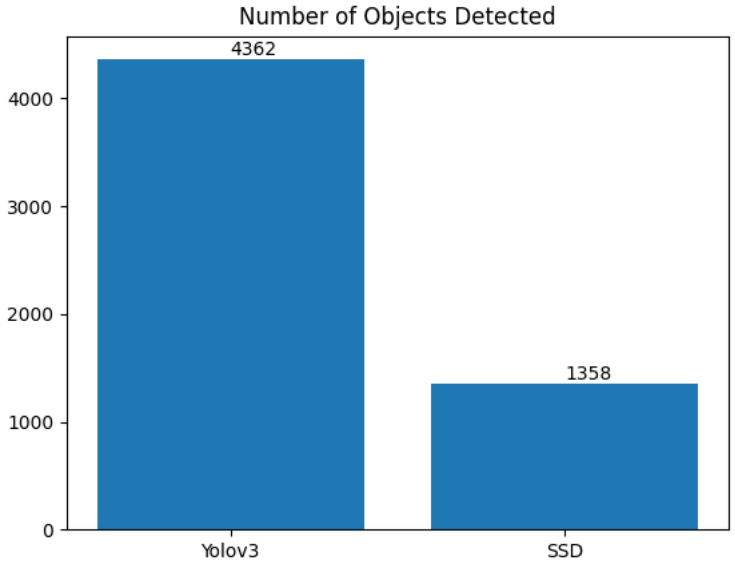
Coco Name Object classifier:

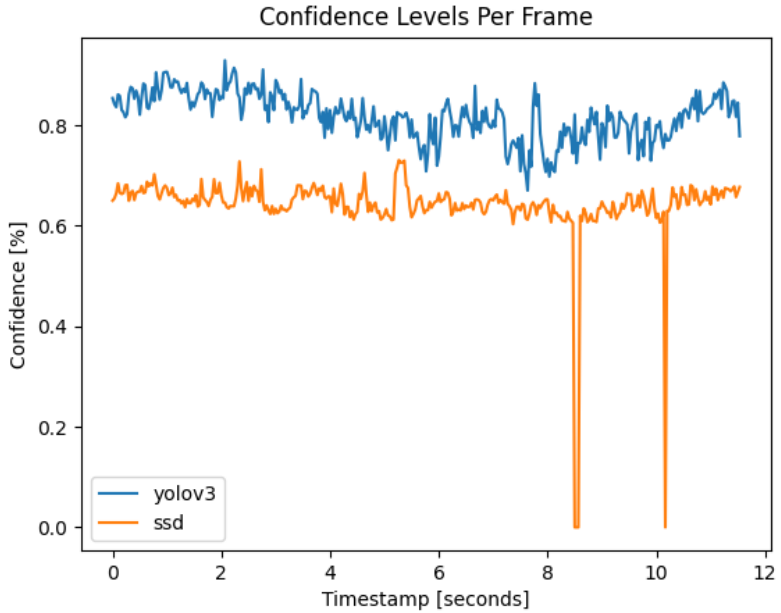
80 different objects

person	bird	suitcase	fork	chair	toaster
bicycle	cat	frisbee	knife	sofa	sink
car	dog	skis	spoon	pottedplant	refrigerator
motorbike	horse	snowboard	bowl	bed	book
aeroplane	sheep	sports ball	banana	diningtable	clock
bus	cow	kite	apple	toilet	vase
train	elephant	baseball bat	sandwich	tvmonitor	scissors
truck	bear	baseball glove	orange	laptop	teddy bear
boat	zebra	skateboard	broccoli	mouse	hair drier
traffic light	giraffe	surfboard	carrot	remote	toothbrush
fire hydrant	backpack	tennis racket	hot dog	keyboard	
stop sign	umbrella	bottle	pizza	cell phone	
parking meter	handbag	wine glass	donut	microwave	
bench	tie	cup	cake	oven	

METRICS

- Number of Objects Detected per frame(Bar graph)
- Accuracy based on Confidence Levels(Plot graph)
- Number of Objects Detected per ClassID(Table)
- Consistency of Algorithms based on the multiple test runs (determine drop-off and standard deviation abnormalities)
- Framerate(working on it)





RESULTS ANALYSIS

CONCLUSIONS

- By far, YOLO v3 outperforms against SSD in the # of objects detected per frame and overall tests. About 4x the amount as shown in Fig.1, that is a 220% increase.
- YOLO v3 also holds a greater confidence levels and can hold bounding boxes on objects for longer by an average of 20%
- YOLO v3 also retains its consistency as in multiple tests the results were within normal deviations when compared to SSDs abnormal results.**
- After researching and comparing the two algorithms in this study, I can confident that YOLO v3 is better among the two.

CONCLUSIONS

REFERENCES

- M. Daily, S. Medasani, R. Behringer and M. Trivedi, "*Self-Driving Cars,*" in Computer, vol. 50, no. 12, pp. 18-23, December 2017, doi: 10.1109/MC.2017.4451204.
- Simhambhatla, Ramesh; Okiah, Kevin; Kuchkula, Shravan; and Slater, Robert (2019) "Self-Driving Cars: Evaluation of Deep Learning Techniques for Object Detection in Different Driving Conditions," *SMU Data Science Review*: Vol. 2 : No. 1 , Article 23..
- Martinez, M., Sitawarin, C., Finch, K., Meincke, L., Yablonski, A., and Kornhauser, A., "*Beyond Grand Theft Auto V for Training*, Testing and Enhancing Deep Learning in Self Driving Cars", arXiv e-prints, 2017.
- Ni J, Chen Y, Chen Y, Zhu J, Ali D, Cao W. *A Survey on Theories and Applications for Self-Driving Cars Based on Deep Learning Methods*. Applied Sciences. 2020; 10(8):2749. https://doi.org/10.3390/app10082749
- Thrun S. (2006) *Winning the DARPA Grand Challenge*. In: Fürnkranz J., Scheffer T., Spiliopoulou M. (eds) Machine Learning: ECML 2006. ECML 2006. Lecture Notes in Computer Science, vol 4212. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11871842_4
- Lin TY. et al. (2014) *Microsoft COCO: Common Objects in Context. In: Fleet D.*, Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48
- What's new in YOLO v3?. A review of the YOLO v3 object… | by Ayoosh Kathuria | Towards Data Science
- YOLO - object detection — OpenCV tutorial 2019 documentation (opencv-tutorial.readthedocs.io)
- J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, 2018. 4
- Review: SSD — Single Shot Detector (Object Detection) | by Sik-Ho Tsang | Towards Data Science
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., & Berg, A. (2016). SSD: Single Shot MultiBox Detector. *ECCV*.

PRESENT DEMO