**Literature Review**

Name: Amr Kenawy

Student # : 500862465

Supervisor’s Name: Dr. Ceni

Date: 18 Feb 2022

Project Abstract

Theme: Deep Learning Model for Multi Object Recognition

Dataset: Berkley Deep Drive Dataset BDD100K, MS COCO Datasets

The Techniques : YOLO (You Look Only Once Convolutional Neural Network), Keras Library

Research questions:

1. Investigate how YOLO algorithm works to speed up Multi Object Detection task
2. Pinpoint the trade-off between model accuracy and processing latency
3. Measure the performance of the model on another dataset and compare KPIs

As one of Autonomous driving essential tasks, Objects detection is one of the potential fields where Deep learning can play a considerable role in mimic human response to continuously changing events and scenarios while driving a vehicle on the road. Accordingly, this model should be able to recognize different objects on the road such as cars, buses, pedestrians, riders etc… stable in different weather conditions such as sunny, cloudy, dark and fast enough to respond correctly at the right time to avoid any accidents.

In order to achieve these features in the model, it needs to be trained on a comprehensive dataset contains rich information such as non-iconic scenes, fully loaded of annotations and includes varieties of roads such as city streets, highways and residential areas, covers different weather conditions at different times of the day, having high resolution frames, it may have 3D in depth views if two cameras are used during data collection plus GPS information that help to identify locations where human driver stop, turn and overtake.

On the other hand, the trained model shall perform multiple tasks: First, detecting lanes to know where to drive and park the car and what other alternative drivable areas in case it decides to change the lane. Second, detecting road objects to avoid collisions and to follow traffic signs by draw bounding boxes around these objects and tagging them with annotations (Classifying the objects), the model should provide well IoU (Intersect Over Union) measuring the overlap of the ground truth and these bounding boxes around the detected objects. Third, generating segmentation from detection, assuming correct detection the model must provide accurate segmentation prediction able to recover object details. Fourth, detecting and tracking multiple objects,

Since these objects are in motion the time must be considered in frames sequences, the output decisions may become obsolete if the processing time is too long. That push for using lower frame resolutions in order to speedup the process, that trade-off between delay and accuracy can be realized either by experimenting different model layers’ structure to optimize inference process or by advancing time axis of the model and use its prediction at “t+1” instead of “t” so use forecasted values instead of real time ones. Fifth, using other information resources such as GPS, 5G IoT (Internet of Things) for enhancing model accuracy by exploiting network effect among the vehicle and other vehicles, people, traffic lights on the road.

Project Scope:

The goal is to apply a deep learning model on a picture contains multiple objects and can do the following: Object detection (Check whether any object does exist?), Localization (Extracting these objects from other surroundings by drawing Bounding Boxes around them) and Classification (Predicting what type of these objects).

I’ll use Yolo ‘You Look Only Once’ version-3 algorithm with Keras library [1] to build a model capable to:

1. Locate the object on the image by finding the bounding box center point coordinates with reference to top left corner of the image
2. The width and height of bounding boxes relative to the width and height of the image
3. Classify these objects with labels and score of confidence

Project Design:

Yolov3 is using multiple convolutional layers to extract feature maps from the input picture, at the same time it has sliding window 3x3 pixels with anchor box located at the center and visit every grid of the picture (picture cells), it creates K (k=9) predictions at each anchor location.

These 9 predictions come from three scales of the input pictures resolutions: 13x13, 26x26 and 52x52 to detect fine grained objects and using three rectangles having different aspect ratios at each scale to find the proper rectangles that maximize the overlap with the true objects (ground truth box) aspect ratios are: [116,90, 156,198, 373,326], [30,61, 62,45, 59,119], [10,13, 16,30, 33,23]

Figure 1 showing prediction Feature Maps at different Scales (Source [5]):

A dog in a cage

Description automatically generated with medium confidence A dog in a cage

Description automatically generated with medium confidence A dog in a cage

Description automatically generated with medium confidence

13 x 13

26 x 26

52 x 52

The above model produces huge number of bounding boxes = w.h.k These bounding boxes undergo another optimization stage to eliminate unnecessary ones:

1. Calculating IOU (Intersection Over Union) score by measuring how much overlap between the proposed bounding box and ground truth box so the boxes with low IOU ratios are ignored
2. Calculating IOU score among proposed boxes to remove redundant bound boxes in Non-Maximal Suppression – NMS function

As the picture subjected to multiple processing through convolutional layers, the output bounding boxes may not fit the original picture size and aspect ratio at the input, so it’s necessary to use a function to correct boxes dimensions back for proper representation.

On the other hand, the classification is done over a provided set of classes, the label is attached along with its confidence score to the bounding box surrounding every object in the picture

Figure 2 showing a schematic diagram of building the model, input, and output

Bound boxes greater than a threshold

Object Classification

CNN Blocks

Load Model’s Weights

Input Picture

encoded Bounding Boxes

Bounding Boxes & Size Correction

Output Picture

CNN Block

CNN Block

Prepare The Image

Optimized Bound boxes

- Image resizing

- convert it to numpy array

- normalize /255

Label and confidence score

Class Prediction

Dimension: Width x Height

Figure 2 The Model (the picture source [3])

Graphical user interface

Description automatically generated

A brief descriptive Statistics of Dataset:

1. BDD100K: 100,000 images, 3.3M Boxes and 40 object classes
2. MS COCO: 123,287 images, 886,284 instances and 80 object classes

References:

1. Marius Cordts1*,*2Mohamed Omran3 Sebastian Ramos1*,*4 Timo Rehfeld1*,*2 Markus Enzweiler1Rodrigo Benenson3 Uwe Franke1 Stefan Roth2 Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding, 2016.
2. Tsung-Yi Lin Michael Maire Serge Belongie Lubomir Bourdev Ross Girshick James Hays Pietro Perona Deva Ramanan C. Lawrence Zitnick Piotr Dollar´. Microsoft COCO: Common Objects in Context, 2015.
3. Songyang Zhang1,\* Lin Song2,∗ Songtao Liu4,∗ Zheng Ge3 Zeming Li4 Xuming He1 Jian SunShanghaiTech University Xi’an Jiaotao University Waseda University Megvii Technology. Workshop on Autonomous Driving at CVPR 2021:Technical Report for Streaming Perception Challenge, 2021.
4. Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, Oscar Beijbom nuTonomy: an APTIV company. nuScenes: A multimodal dataset for autonomous driving, 2020.
5. Fisher Yu1 Haofeng Chen1 Xin Wang1 Wenqi Xian2[[1]](#footnote-1) Yingying Chen[[2]](#footnote-2)Fangchen Liu3∗ Vashisht Madhavan4∗ Trevor Darrell1 UC Berkeley Cornell University UC San Diego Element, Inc. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning, 2020.
6. Mengtian Li1, Yu-Xiong Wang1,2, and Deva Ramanan CMU, UIUC, and Argo AI. Towards Streaming Perception, 2020.
7. the most widely used project for using pre-trained the YOLO models is called “[keras-yolo3: Training and Detecting Objects with YOLO3](https://github.com/experiencor/keras-yolo3)” by [Huynh Ngoc Anh](https://www.linkedin.com/in/ngoca/) or experiencor.
8. How to Perform Object Detection With YOLOv3 in Keras [How to Perform Object Detection With YOLOv3 in Keras (machinelearningmastery.com)](https://machinelearningmastery.com/how-to-perform-object-detection-with-yolov3-in-keras/) by [Jason Brownlee](https://machinelearningmastery.com/author/jasonb/) on May 27, 2019 in [Deep Learning for Computer Vision](https://machinelearningmastery.com/category/deep-learning-for-computer-vision/)
9. You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon , Santosh Divvala, Ross Girshick , Ali Farhadi∗† University of Washington∗ , Allen Institute for AI , Facebook AI Research
10. How to Implement a YOLO (v3) Object Detector from Scratch in PyTorch: Part 1By [Ayoosh Kathuria](https://www.linkedin.com/in/ayoosh-kathuria-44a319132/" \t "_blank), [How to Implement a YOLO (v3) Object Detector from Scratch in PyTorch: Part 1 - KDnuggets](https://www.kdnuggets.com/2018/05/implement-yolo-v3-object-detector-pytorch-part-1.html)

1. [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)