**Literature Review**

Name: Amr Kenawy

Student # : 500862465

Supervisor’s Name: Dr. Ceni

Date: 18 Feb 2022

GitHub : <https://github.com/Amrkenawy/Multi-Object-Recognition.git>

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, it is using Feature Pyramid Network (FPN) for RPN (Regional Proposal Network is a sliding -window class-agnostic object detector) to detect smaller and bigger objects in the picture by scanning the Model over both positions and pyramid levels, imagine looking for a restaurant on a google map while driving a car, first we zoom out to find the main roads then we zoom in when getting closer to the restaurant in order to be able to see smaller streets that leads to it

*Figure 1 showing Feature Pyramid Maps at different Scales (Source [11])*

Diagram

Description automatically generated

Top-down pathway and lateral connections for coarse resolution feature map

Three Independent Predictions at every Feature Map Layer in FPN

Single Stage Feature map in Regular ConvNet

Sliding 3x3 Window

Feed-Forward Computation of the ConvNet

Input Picture

The sliding window in RPN is 3x3 pixels with anchor box located at the center, the anchor box shall visit every grid of the picture and detect whether it is an Object or non-Object (Object-Agnostic), it creates K (k=9) predictions at each anchor location.

These 9 predictions come from:

1. three input picture scales as shown in the feature pyramid (13x13, 26x26 and 52x52) to detect bigger objects using 13x13 scale and fine-grained ones using 52x52 scale
2. three rectangles with different aspect ratios at each scale pursuing the proper rectangles that maximize the overlap with the true objects (ground truth box). Note: aspect ratios are: [116,90, 156,198, 373,326], [30,61, 62,45, 59,119], [10,13, 16,30, 33,23]

*Figure 2 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 = width x height x 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 individual proposed bounding box and ground truth box so the boxes with low IOU ratios are ignored
2. Calculating IOU score among proposed boxes (output from first stage) to remove redundant bound boxes in Non-Maximal Suppression process – NMS function

As the original input picture subjected to multiple processing through convolutional layers, the output bounding boxes after NMS cannot be applied directly to this picture without correcting boxes’ dimensions for proper object representation.

At this stage the model still Object-Agnostic, identified how many objects and the location of these objects (x, y coordinates and w, h of each box), the next step is to identify what type of Objects inside these boxes through classification process. Classification is done over a provided set of classes (80 classes in COCO dataset), the label is attached along with its confidence score to the bounding box surrounding every object in the picture

*Figure 3 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 4 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)
11. Feature Pyramid Networks for Object Detection Tsung-Yi Lin1,2, Piotr Dollar´ 1 , Ross Girshick1 , Kaiming He1 , Bharath Hariharan1 , and Serge Belongie2

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