

Tribhuvan University

Institute of Science and Technology

Seminar Report

On

OBJECT DETECTION USING YOLO CNN

Submitted to

Central Department of Computer Science & Information Technology

Tribhuvan University, Kirtipur

Kathmandu, Nepal

Submitted by

Mausham Shrestha

Roll No.: 1 / 075

In partial fulfillment of the requirement for Master's Degree in Computer Science and Information Technology (M.Sc. CSIT), 1st Semester

November, 2019



Tribhuvan University

Institute of Science and Technology

**Supervisor’s Recommendation**

I hereby recommend that this Seminar report is prepared under my supervision by **Mausham Shrestha** entitled “**OBJECT DETECTION USING YOLO CNN**” be accepted as fulfillment in partial requirement for the degree of Master's of Science in Computer Science and Information Technology. In my best knowledge, this is an original work in computer science.

…………………………………

Asst. Prof. Bikash Balami

Central Department of Computer Science

and Information Technology

Date: 2019



Tribhuvan University

Institute of Science and Technology

LETTER OF APPROVAL

This is certify that the seminar report prepared by Mr. Mausham Shrestha entitled “**OBJECT DETECTION USING YOLO CNN**” in partial fulfillment of the requirements for the degree of Master's of Science in Computer Science and Information technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Evaluation Committee

………………………….. …………………………

Asst. Prof. Nawaraj Paudel Asst. Prof. Bikash Balami

(HOD) (Supervisor)

Central Department of Computer Central Department of Computer

Science & Information Technology Science & Information Technology

..…………………………………

(External)

Date: 2019

# ACKNOWLEDGEMENT

With immense please, I submit my deepest gratitude to the Central Department of Computer Science and Technology, Tribhuvan University.

First of all, I would like to thank Asst. Prof. **Nawaraj Paudel**, Head of Central Department of Computer Science and Information Technology, Tribhuvan University for his encouragement and guidance.

I would like to thank Asst. Prof. **Bikash Balami** for his valuable guidance in carrying out this work under his effective supervision and enlightenment I am also grateful for his excellent guidance, patience, trust and correction of my seminar work.

I also want to thank Asst. Prof. **Mr. Tej Bahadur Shahi** for his inspiration, insightful discussion, thoughtful guidance and critical comments.

# ABSTRACT

Object detection is the process of identifying the object available in picture or video. Here YOLO stands for you only look once. Compared to other region proposal classification networks (fast RCNN) which perform detection on various region proposals and thus end up performing prediction multiple times for various regions in a image, Yolo architecture is more like FCNN (fully convolutional neural network) and passes the image (nxn) once through the FCNN and output is (mxm) prediction. This the architecture is splitting the input image in mxm grid and for each grid generation 2 bounding boxes and class probabilities for those bounding boxes. Note that bounding box is more likely to be larger than the grid itself.

Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance.

Keywords: *YOLO, bounding boxes, sliding window.*

# TABLE OF CONTENTS

[ACKNOWLEDGEMENT III](#_Toc24979517)

[ABSTRACT IV](#_Toc24979518)

[TABLE OF CONTENTS V](#_Toc24979519)

[LIST OF FIGURES VI](#_Toc24979520)

[LIST OF ABBREVIAITONS VII](#_Toc24979521)

[CHAPTER I 1](#_Toc24979522)

[INTRODUCTION 1](#_Toc24979523)

[1.1 Background 1](#_Toc24979524)

[1.2 Problem Statement 2](#_Toc24979525)

[1.3 Scope and Objectives of the Project 2](#_Toc24979526)

[CHAPTER II 4](#_Toc24979527)

[LITERATURE REVIEW 4](#_Toc24979528)

[2.1 Previous Works, Discussions and Findings 4](#_Toc24979529)

[CHAPTER III 6](#_Toc24979530)

[METHODOLOGY 6](#_Toc24979531)

[3.1 Unified Detection 6](#_Toc24979532)

[3.2 Network Design 7](#_Toc24979533)

[CHAPTER IV 9](#_Toc24979534)

[IMPLEMENTATION 9](#_Toc24979535)

[CHAPTER V 10](#_Toc24979536)

[RESULT AND ANALYSIS 10](#_Toc24979537)

[CHAPTER VI 12](#_Toc24979538)

[CONCLUSION 12](#_Toc24979539)

[REFERENCES](#_Toc24979540)

# LIST OF FIGURES

[Figure 1. 1: The YOLO Detection System 1](#_Toc24979360)

Y[Figure 3. 1: The Model 7](#_Toc24979300)

[Figure 3. 2: The Architecture 8](#_Toc24979301)

Y[Figure 4. 1: Before and after processing of image ………………………………………..10](#_Toc24979469)

# LIST OF ABBREVIAITONS

DPM Deformable Parts Models

FPS Frame Per Second

GPU Graphics Processing Unit

IOU Intersection Of Union

mAP Mean Average Precision

R-CNN Regional Convolution Neural Network

RoI Region Of Intersection

YOLO You Only Look Once

# CHAPTER I

# INTRODUCTION

## 1.1 **Background**

Humans glance at an image and instantly know what objects are in the image, where they are, and how they interact. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little conscious thought. Fast, accurate algorithms for object detection would allow computers to drive cars without specialized sensors, enable assistive devices to convey real-time scene information to human users, and unlock the potential for general purpose, responsive robotic systems. Current detection systems repurpose classiﬁers to perform detection. To detect an object, these systems take a classiﬁer for that object and evaluate it at various locations and scales in a test image. Systems like deformable parts models (DPM) use a sliding window approach where the classiﬁer is run at evenly spaced locations over the entire image. More recent approaches like R-CNN use region proposal. Methods to ﬁrst generate potential bounding boxes in an image and then run a classiﬁer on these proposed boxes. After classiﬁcation, post-processing is used to reﬁne the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene [4]. These complex pipelines are slow and hard to optimize because each individual component must be trained separately. We reframe object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities.

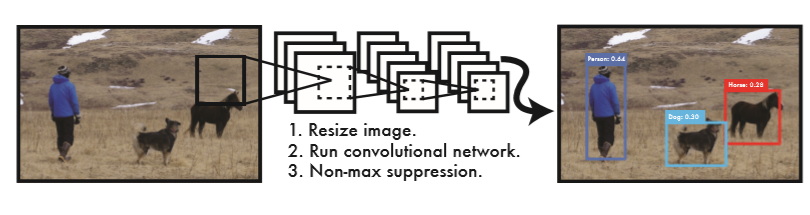


Figure 1. : The YOLO Detection System

Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448×448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model’s conﬁdence.

Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are. YOLO is refreshingly simple. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This uniﬁed model has several beneﬁts over traditional methods of object detection. First, YOLO is extremely fast. Since the detection is frame as a regression problem it does not require a complex pipeline. Simply run neural network on a new image at test time to predict detections. The base network runs at 45 frames per second with no batch processing on a Titan X GPU and a fast version runs at more than 150 fps. This means we can process streaming video in real-time with less than 25 milliseconds of latency. Furthermore, YOLO achieves more than twice the mean average precision of other real-time systems[5].

Second, YOLO reasons globally about the image when making predictions. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance. Fast R-CNN, a top detection method [14], mistakes background patches in an image for objects because it can’t see the larger context. YOLO makes less than half the number of background errors compared to Fast R-CNN. Third, YOLO learns generalizable representations of objects. When trained on natural images and tested on artwork, YOLO outperforms top detection methods like DPM and R-CNN by a wide margin. Since YOLO is highly generalizable it is less likely to break down when applied to new domains or unexpected inputs. YOLO still lags behind state-of-the-art detection systems in accuracy. While it can quickly identify objects in images it struggles to precisely localize some objects, especially small ones.

## 1.2 Problem Statement

Object detection in real-time was not much implemented before the introduction of YOLO since before this no any algorithm was able to process the video in real time. The algorithm was only able to process 1 image per 2 second which was not enough to process real-time data.

## 1.3 Scope and Objectives of the Project

Object detection have numerous scope and application in computer vision few of them are listed below:

* One of the best examples of why you need object detection is for autonomous driving is In order for a car to decide what to do in next step whether accelerate, apply brakes or turn, it needs to know where all the objects are around the car and what those objects are That requires object detection and we would essentially train the car to detect known set of objects such as cars, pedestrians, traffic lights, road signs, bicycles, motorcycles, etc.
* Object detection system is also used in tracking the objects, for example tracking a ball during a football match, tracking movement of a cricket bat, tracking a person in a video. Object tracking has a variety of uses, some of which are surveillance and security, traffic monitoring, video communication, robot vision and animation.
* Iris recognition is one of the most accurate identity verification systems. Identity verification and identification is becoming increasingly popular. However, advances in the field have expanded the options to include biometrics such as iris, retina and more. Among the large set of options it has been shown that the iris is the most accurate biometric. Hence we need object detection system in iris detection.
* Activity recognition aims to recognize the actions and goals of one or more agents from a series of observations on the agents actions and the environmental conditions. This research field has captured the attention of several computer science communities due to its strength in providing personalized support for many different applications and its connection to many different fields of study such as human-computer interaction, or sociology.
* Autonomous assistive robots must be provided with the ability to process visual data in real time so that they can react adequately for quickly adapting to changes in the environment. Reliable object detection and recognition is usually a necessary early step to achieve this goal.

## 

# CHAPTER II

# LITERATURE REVIEW

## 2.1 Previous Works, Discussions and Findings

Various study has been already performed for detection objects in image as well as video. Most of the algorithm seems accurate but seems to be too slow. Some algorithms are DPM (deformable part models), R-CNN, Fast R-CNN.

DPM are deﬁned by a coarse root ﬁlter that approximately covers an entire object and higher resolution part ﬁlters that cover smaller parts of the object. The root ﬁlter location deﬁnes a detection window (the pixels contributing to the part of the feature map covered by the ﬁlter). The part ﬁlters are placed λ levels down in the pyramid, so the features at that level are computed at twice the resolution of the features in the root ﬁlter level. It is found that using higher resolution features for deﬁning part ﬁlters is essential for obtaining high recognition performance. With this approach the part ﬁlters capture ﬁner resolution features that are localized to greater accuracy when compared to the features captured by the root ﬁlter. Consider building a model for a face. The root ﬁlter could capture coarse resolution edges such as the face boundary while the part ﬁlters could capture details such as eyes, nose and mouth [2]. It is also called sliding window technique. Entire image is divided into number of block and each block needs to be processed separately. The execution time is very high. It executes at 0.07FPS i.e. it takes 14 second to process each image with 33.7mAP.

In object detection, methods such as R-CNN have obtained excellent results by integrating CNNs with region proposal generation algorithms such as selective search. The R-CNN method is a chain of conceptually simple steps: generating candidate object regions, classifying them as foreground or background, and post-processing them to improve their ﬁt to objects. R-CNN starts by running an algorithm to extracts from an image x a shortlist of image regions R ∈R(x) that are likely to contain objects. These proposals, in the order of a few thousands per image, may have arbitrary shapes, but in the following are assumed to be converted to rectangles. It could process image at the rate of 0.05FPS i.e. it takes 20s to process one image with 66.0mAP.

A Fast R-CNN network takes as input an entire image and a set of object proposals. The network ﬁrst processes the whole image with several convolutional (conv) and max pooling layers to produce a conv feature map. Then, for each object proposal a region of interest (RoI) pooling layer extracts a ﬁxed-length feature vector from the feature map. Each feature vector is fed into a sequence of fully connected (fc) layers that ﬁnally branch into two sibling output layers: one that produces soft max probability estimates over K object classes plus a catch-all “background” class and another layer that outputs four real-valued numbers for each of the K object classes. Each set of 4 values encodes reﬁned bounding-box positions for one of the K classes [1]. The execution time is very high. It executes at 0.5FPS i.e. it takes 2 second to process each image with 70mAP.

# CHAPTER III

# METHODOLOGY

## 3.1 Unified Detection

Unification of separate components of object detection into done in a single neural network. This network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously. This means the network reasons globally about the full image and all the objects in the image. The YOLO design enables end-to-end training and real time speeds while maintaining high average precision. This system divides the input image into an S ×S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and conﬁdence scores for those boxes. These conﬁdence scores reﬂect how conﬁdent the model is that the box contains an object and also how accurate it thinks the box is that it predicts. Formally conﬁdence is deﬁne as Pr(Object)∗ . If no object exists in that cell, the conﬁdence scores should be zero. Otherwise we want the conﬁdence score to equal the intersection over union (IOU) between the predicted box and the ground truth. Each bounding box consists of 5 predictions: x, y, w, h, and conﬁdence. The (x,y) coordinates represent the center of the box relative to the bounds of the grid cell. The width and height are predicted relative to the whole image. Finally the conﬁdence prediction represents the IOU between the predicted box and any ground truth box. Each grid cell also predicts C conditional class probabilities, Pr(Classi|Object). These probabilities are conditioned on the grid cell containing an object. This algorithm only predicts one set of class probabilities per grid cell, regardless of the number of boxes B.

At test time multiply the conditional class probabilities and the individual box conﬁdence predictions,

Pr(Classi|Object)∗Pr(Object)∗  = Pr(Classi)∗  (1)

which gives us class-speciﬁc conﬁdence scores for each box. These scores encode both the probability of that class appearing in the box and how well the predicted box ﬁts the object [5].

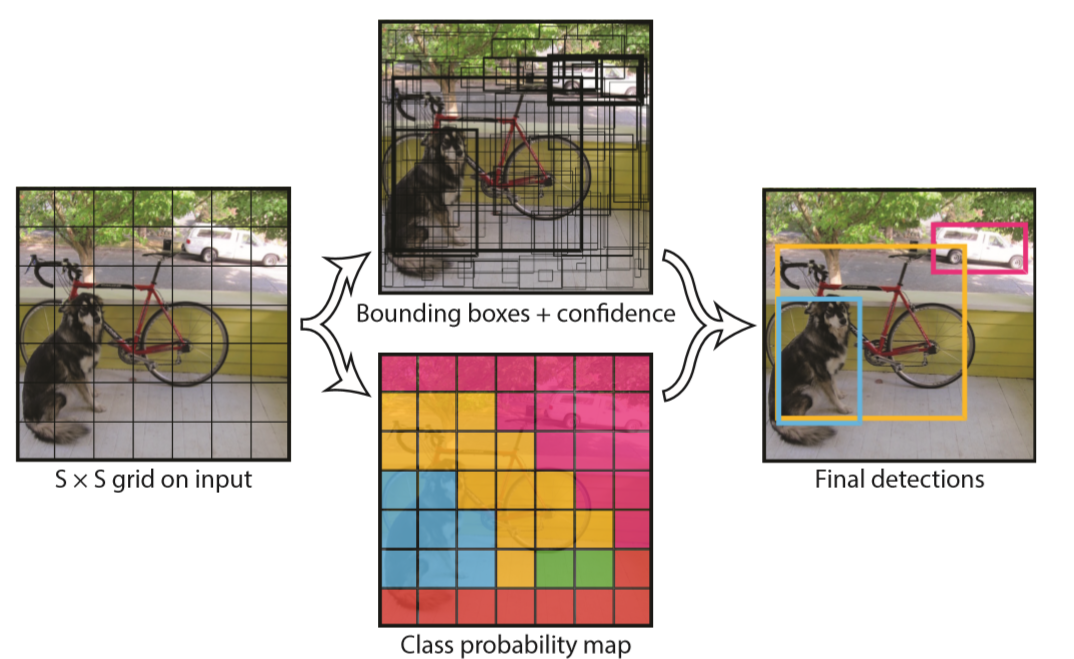


Figure 3. : The Model

This system models detection as a regression problem. It divides the image into an S×S grid and for each grid cell predicts B bounding boxes, conﬁdence for those boxes, and C class probabilities. These predictions are encoded as an S ×S ×(B ∗5 + C) tensor.

## 3.2 Network Design

The initial convolutional layers of the network extract features from the image while the fully connected layers predict the output probabilities and coordinates. The network has 24 convolutional layers followed by 2 fully connected layers.

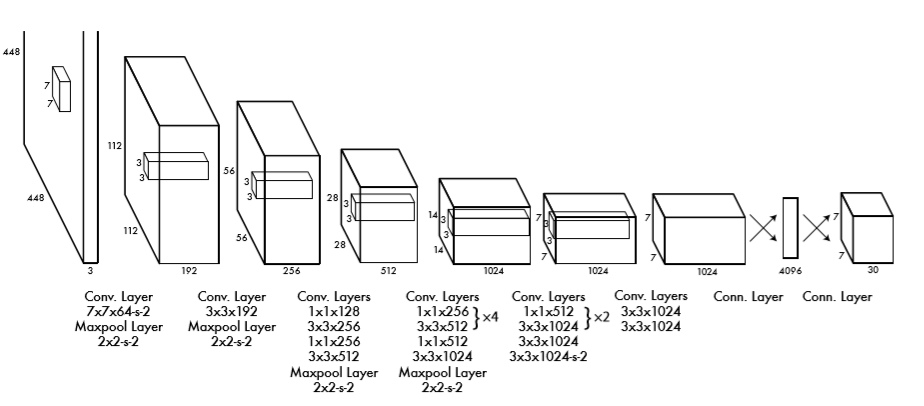


Figure 3. : The Architecture

The detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. Here, convolutional layer is pretrain on the ImageNet classiﬁcation task at half the resolution (224×224 input image) and then double the resolution for detection [5].

# CHAPTER IV

# IMPLEMENTATION

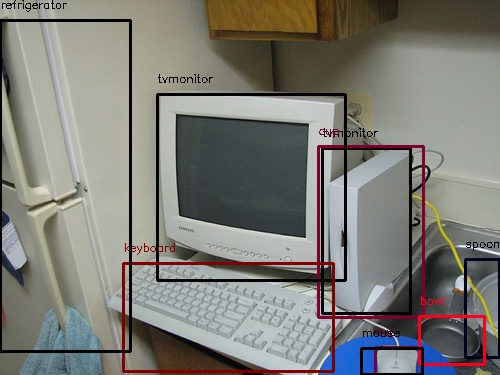
YOLO was written in darknet which is custom deep learning framework which is written C and CUDA. Since, C is not very user friendly there is python version called darkflow. The pretrained model was retrieved from darkflow and implement it on python command line interface. To achieve high speed we can use GPU which support tensorflow. Otherwise it becomes little slow.

Here using YOLO, we can perform real time implementation. Which include providing video input form web camera and at the same time it would provide object detected video. Both image and video were passed through implementation.

# CHAPTER V

# RESULT AND ANALYSIS

The number of photos and video were processed. The video would be provided as Demo and few processed images are:

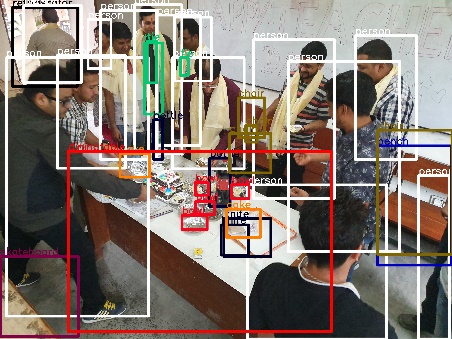
 





Figure 4. : Before and after processing of image

Here the image was processed very fast. The processing of video also does not take a lot of time. Since it processes entire image at a time it make less background error. YOLO imposes strong spatial constraints on bounding box predictions since each grid cell only predicts two boxes and can only have one class. This spatial constraint limits the number of nearby objects that our model can predict. Our model struggles with small objects that appear in groups, such as ﬂocks of birds. Since our model learns to predict bounding boxes from data, it struggles to generalize to objects in new or unusual aspect ratios or conﬁgurations. Our model also uses relatively coarse features for predicting bounding boxes since our architecture has multiple down sampling layers from the input image. Finally, while we train on a loss function that approximates detection performance, our loss function treats errors the same in small bounding boxes versus large bounding boxes. A small error in a large box is generally benign but a small error in a small box has a much greater effect on IOU. Our main source of error is incorrect localizations.

# CHAPTER VI

# CONCLUSION

We introduce YOLO, a uniﬁed model for object detection. Our model is simple to construct and can be trained directly on full images. Unlike classiﬁer-based approaches, YOLO is trained on a loss function that directly corresponds to detection performance and the entire model is trained jointly. Fast YOLO is the fastest general-purpose object detect or in the literature and YOLO pushes the state-of-the-art in real-time object detection. YOLO also generalizes well to new domains making it ideal for applications that rely on fast, robust object detection.

# REFERENCES

[1] K. Lenc and A. Vedaldi. R-cnn minus r. arXiv preprint arXiv:1506.06981, 2015

[2]P.F.Felzenszwalb,R.B.Girshick,D.McAllester,andD.Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010.

[3] R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 580–587. IEEE, 2014. 1, 4, 7

[5] R. Joshep, D. Santosh, G. Ross, F. Ali. You Only Look Once: Uniﬁed, Real-Time Object Detection, University of Washington, Allen Institute for AI, Facebook AI Research, 9 May 2016.