# Tell me what to find

## Artificial Intelligence CS 640

Mohit Mohit, Abhishek Rai Sharma, Sahil Gupta

# Introduction

- Given a text and a set of images, we need to detect the object in the image set by the text description.
- We have set of image frames from certain videos and these images are pre-labeled with bounding boxes of objects from LaSOT (Large-scale Single Object Tracking).
- For a particular video we have a set of images captured from timeline, one for “annotation” and the rest for “verification”
- To achieve this task, first of all, we labelled the video with a natural language description of the object in the bounding box.

- Object detection: In this step, we will find important object in the image

- Object matching with description: From the given natural language description of the object we will extract the object and search them in image set.

# Available Object Detectors

There are several object detectors to choose from like Faster R-CNN, SSD and YOLOv3. Let’s make a fair comparison:

- Faster R-CNN (Region based detector) demonstrate a small accuracy advantage but with slow speed.
- SSD have a pretty impressive frame per seconds (FPS) using lower resolution images at the cost of accuracy.
- YOLOv3 has an impressive FPS without the loss of too much accuracy.

After comparing the accuracy and the frames per second of these algorithms, we selected YOLOv3 to move forward with object detection.

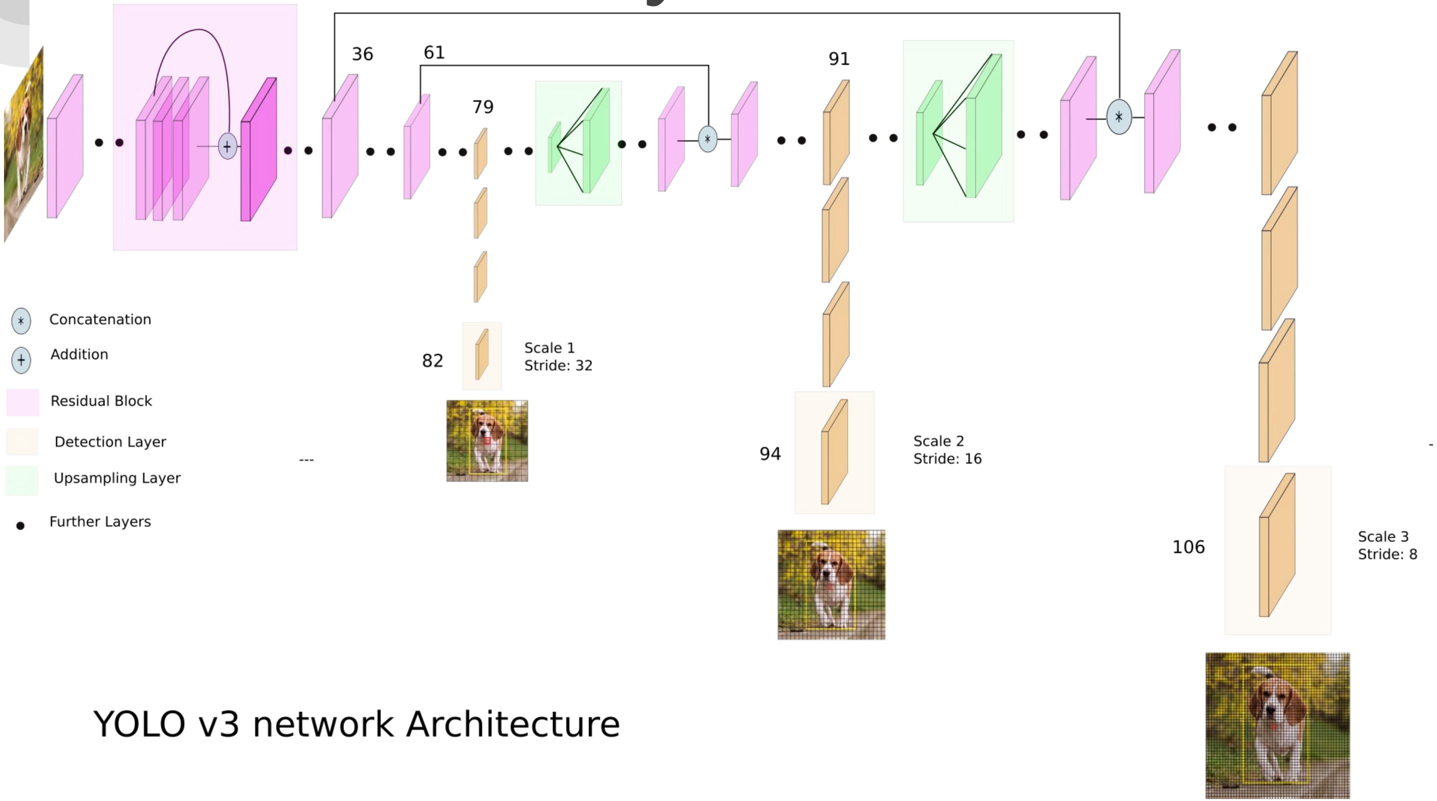
# YOLOv3

- YOLOv3 is a fully convolutional network and its eventual output is generated by applying a 1 x 1 kernel on a feature map.
- We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region.
- Image classification and localization are applied on each grid.
- To identify multiple object in a single grid, we use anchor boxes: using predetermined bounding boxes for checking an object class.
- Later, we calculate the confidence score and predict the bounding boxes for each grid.
- Confidence scores are calculated using Intersection over Union (IoU).
- Higher confidence score demonstrates accuracy of the predicted object in the grid.

# YOLOv3

- YOLO v3 uses a variant of Darknet, which originally has 53 layer network trained on Imagenet. For the task of detection, 53 more layers are stacked onto it.
- The first detection is made by the 82nd layer. For the first 81 layers, the image is downsampled by the network.
- 81st layer has a stride of 32. If we have an image of 416 x 416, the resultant feature map would be of size 13 x 13.
- The feature map from layer 79 is subjected to a few convolutional layers before being up sampled by 2x to dimensions of 26 x 26.
- This feature map is then depth concatenated with the feature map from layer 61.
- the second detection is made by the 94th layer, yielding a detection feature map of 26 x 26x255..
- A similar procedure is followed again, We make the final of the 3 at 106th layer, yielding feature map of size 52 x 52 x 255.

# You Only Look Once

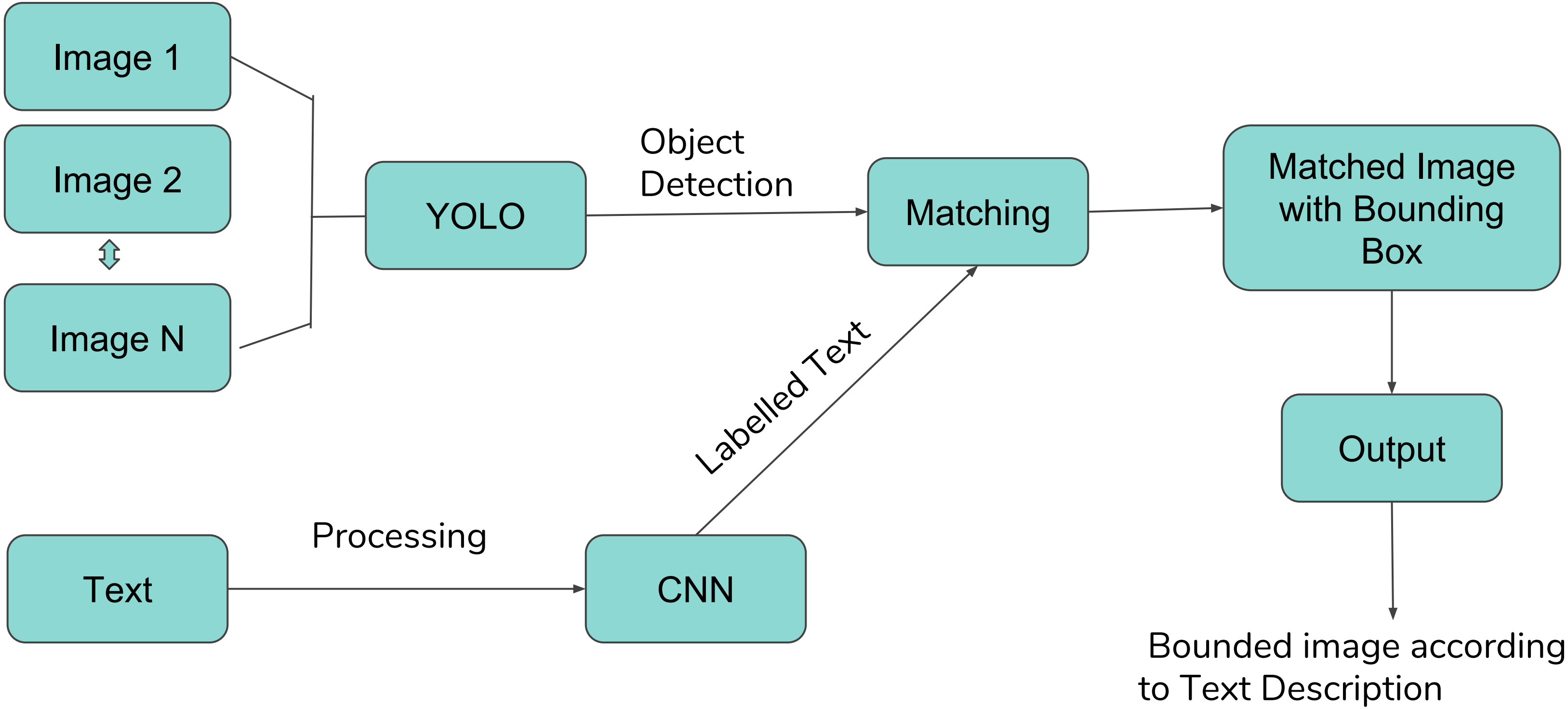


YOLO v3 network Architecture

# To go beyond just detection!

- YOLOv3 or any object detector simply lists all possible bounding boxes for any image.
- To find a selective object in a given set of images, we need to come up with a technique to decide which object class to focus.
- Thus, to predict the object a user is interested in tracking, we are using a language model.

# Object detection using NLP



# NLP using CNN with GloVe Embedding

- Glove(Global Vectors for word representation) is an unsupervised learning algorithm for obtaining vector representations of words.
- Dense vector representations of word forms where similar words are close in the vector space is done through word embedding.
- Through glove embeddings to convert a word into a 100 dimensional word embedding vector.
- We passed these stacked vectors which include padding to input of CNN
- CNN for Text Classification with GloVe embeddings is modelled to get the class for each text description.

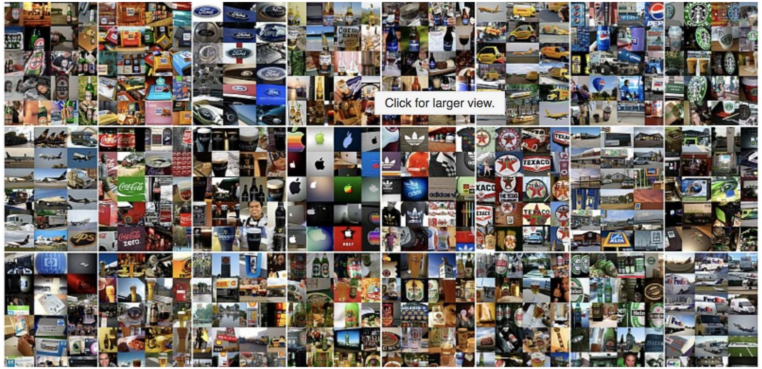
# Matching Text with Image object

- Passing text description through CNN we get the labelled text description.
- YOLOv3 has the labelled images with confidence intervals.
- After finding the desired class by our language model, we will find the object in the image set with the highest confidence.
- The output of the model will be the image with the desired object class along with the bounding box, label, and confidence score.

# Detection of image through text description

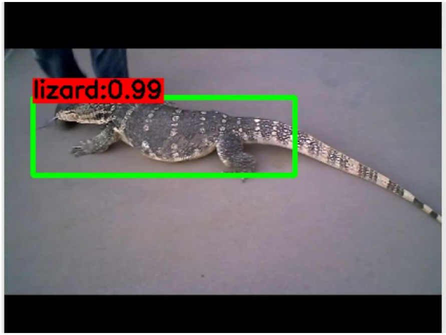
User wants to detect the “ Lizard walking on Road”

Dataset containing 160000 images - from LaSOT

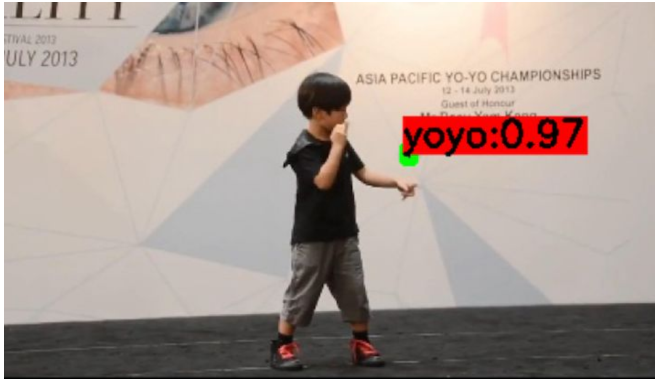


# Using CNN and YOLO model for  object detection for particular text

Choosing the image according to the text description



“Lizard walking on Road”



“ Find a yo yo”

# Cosine Similarity

- We generated the outputs of both Text Classification model and Yolo model.
- We created the softmax vector of all the 70 classes for both NLP description and the group of images.
- We check the cosine similarity between the two non-zero vectors which gave us the similarity between the text class and the image class.

# Results

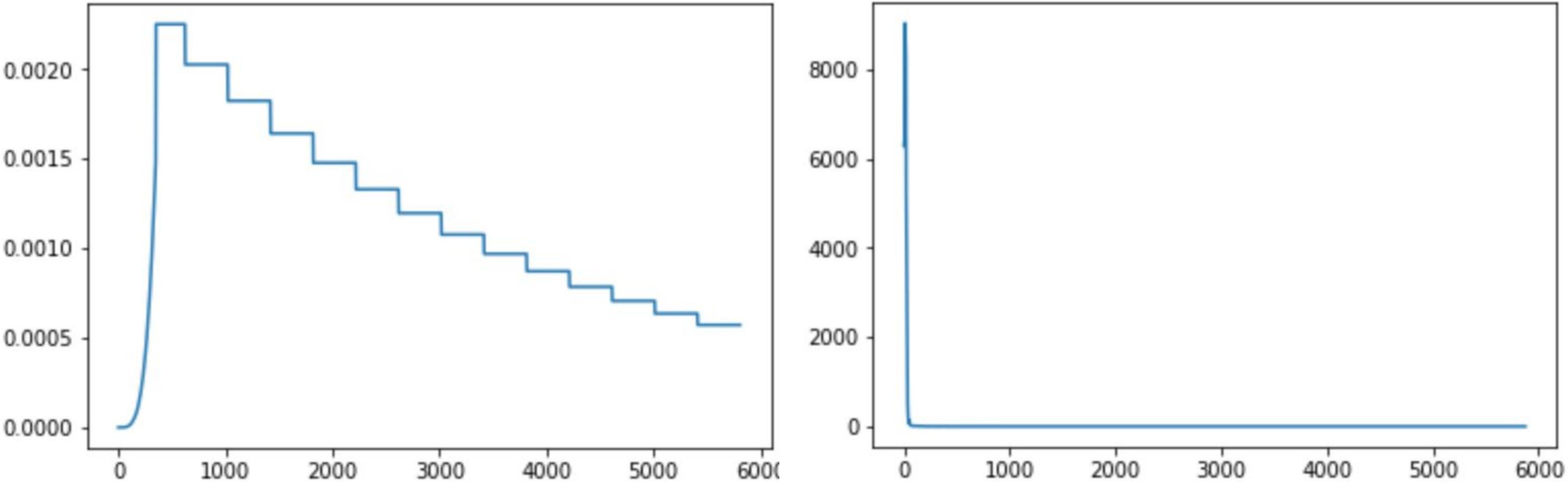
- For evaluation of the model, we created a new dataset that had 70 images
- These images were not used for training
- We text annotated the images
- To evaluate the model, we passed this dataset of 70 images with 1 of the annotated text
- The correct gets score +1 if it correctly identifies the image associated with the label

| | |
|---|---|
| Score for baseline model | 37 |
| Score for model with cosine similarity | 42 |

# Image Resizing maintaining aspect ratio for improving learning rate

- Current yolo implementation takes around 1 day to run ~1k epochs. This is very slow for our purpose because even after 1000 epochs, we do not get bounding boxes as the data set available is huge but context information is limited
- YOLO doesn’t require us to convert the image aspect ratio and automatically resizes it
- We created a method for resizing the image according to input layer of YOLO ( ie 416*416) by putting black labels on images while resizing them and maintaining their aspect ratio
- This led to increase in epoch rate from ~1k to ~3.6k epochs in a day. We trained our model with decreasing epoch rate over 9k epochs

# Learning Rate vs Training Error



# Roadblocks

- Didn’t had any labelled dataset for text description according to LaSOT dataset classes.
- Have to manually create the labelled dataset for text description.
- Comparing the text and image features were quite difficult.
- Resizing of images needed to be done for making it consistent.

# Thank You.