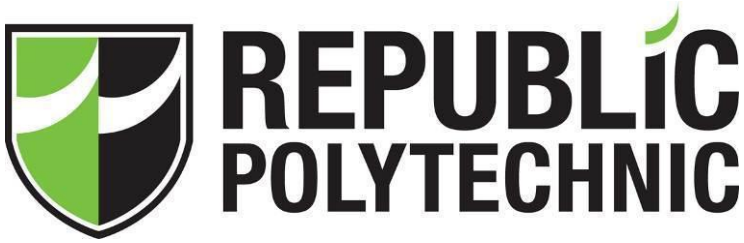


GRAYMATICS – Video Analytics for HELMET detection

Date of Submission: 07-April-2020

Submitted By:

David Tan Leng Poh
Vivek Bhatnagar

This page is Intentionally left blank

Abstract

Personal protection equipment (PPE), such as the HELMET is paramount for the safety of two-wheeler riders and to facilitate enforcement agencies there is a need to systemically detect errant drivers on roads.

Our Capstone project is to address this need through an AI solution which can annotate images of riders without a HELMET in a video stream from traffic cameras on the roads.

Graymatics is targets traffic images and videos of India and uses YOLOv3 models for video analytics and detection.

Graymatics - a cognitive media processing company based in Santa Clara, California, United States and in Singapore. The company is most well-known for its digital video analysis and image-recognition technology, a platform capable of identifying the brands of products within images or video content.

Acknowledgement

This accomplishment was made possible by the timely and valuable support of the project stakeholders. Our sincere gratitude to

Graymatics:

- Mr. Abhijit Shanbhag – Industry partner
- Alex, YT & DonYun – Product Manager & specialists

Republic Polytechnic:

- Dr. Jimmy Goh – Supervisor and guide
- Mr. Khee Wei Seow & Dr. Beng Keat Liew – Mentors

Table of Contents

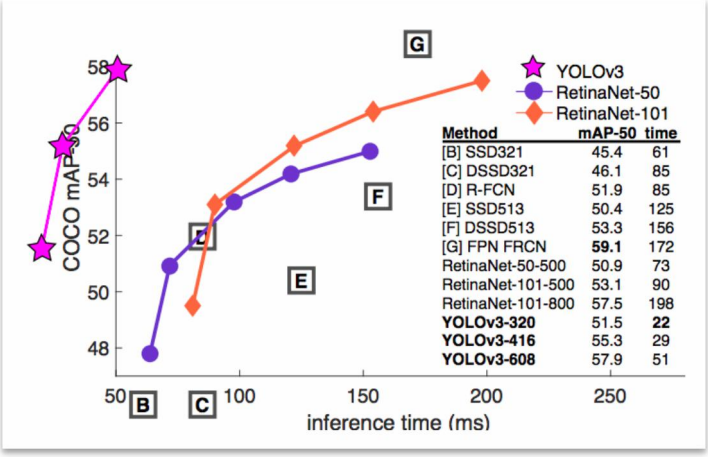
1. Background

1.1. YOLO: Real-Time Object Detection

You only look once (YOLO) was the industry partner's choice and rightfully so as it performs significantly faster than any other detection methods.

1.2. Comparison to Other Detectors[2]

YOLOv3 is extremely fast and accurate. In mAP measured at .5 IOU YOLOv3 is on par with Focal Loss but about 4x faster. Moreover you can easily trade-off between speed and accuracy simply by changing the size of the model, no retraining required!



2. Methodology and Design

2.1. Project specifications

Each box predicts the classes the bounding box may contain using multilabel classification.

In our case, it's 2 classes – helmet and no helmet.

We used a new network for performing feature extraction which had 53 convolutional layers, Darknet-53.

Each network is trained with identical settings and tested at 256_256, single crop accuracy.

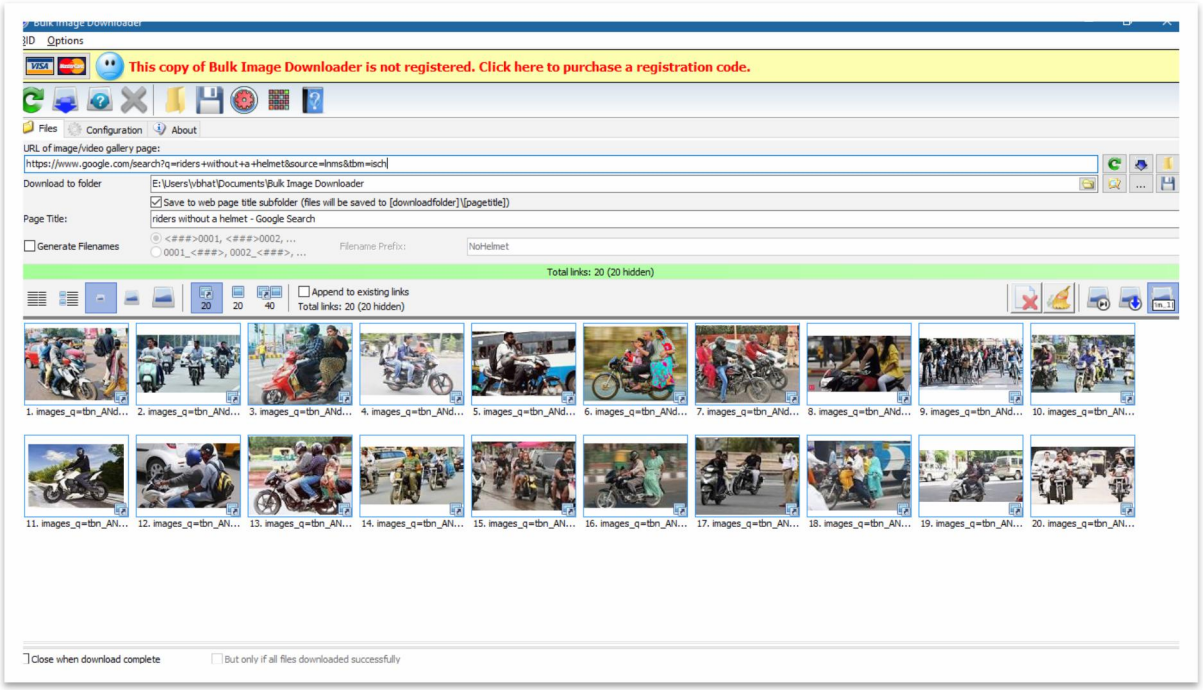
We use the Darknet neural network framework for training and testing.

2.2. Search and download images

Bulk Image Downloader [4] tool was found suitable to download traffic images using the free tier.

Using this search string, the tool found images and then only the allowed ones were downloaded.

https://www.google.com/search?q=riders+without+a+helmet&source=lnms&tbm=isch



2.3. Labelling of images to be used as training dataset.

We have labelled about 70 images to use for training the YOLO model.

YOLO format is:

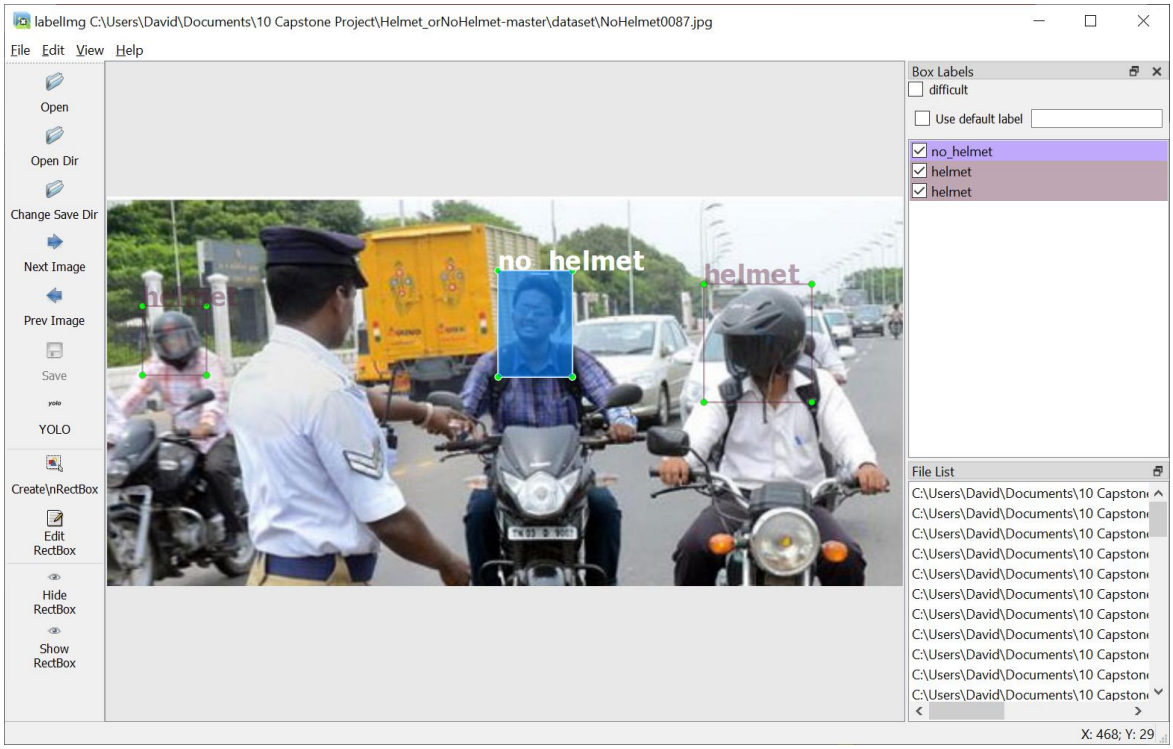
<object-class> <x> <y> <width> <height>
<x_center> = ((X_end + X_start) / 2) / image_width
<y_center> = ((Y_end + Y_start) / 2) / image_height
<width> = (X_end - X_start) / image_width
<height> = (Y_end - Y_start) / image_height

Example:

0 0.848780 0.160156 0.143089 0.242188
1 0.446341 0.191406 0.128455 0.216146
1 0.173984 0.222656 0.139837 0.205729

Classes.txt: (object-class)

helmet
no_helmet



2.4. Clone and Build Darknet in server

git clone https://github.com/AlexeyAB/darknet

Cloning into 'darknet'...
remote: Enumerating objects: 46, done.
remote: Counting objects: 100% (46/46), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 13021 (delta 20), reused 21 (delta 10), pack-reused 12975
Receiving objects: 100% (13021/13021), 11.82 MiB | 16.45 MiB/s, done.
Resolving deltas: 100% (8881/8881), done.

cd darknet
make

2.5. Configuring Files for Training

We have to change the cfg/vochelmet.data config file to point to the data:

```
1 classes= 2
2 train  = /home/txt/trainH.txt
3 valid  = /home/txt/2020_testH.txt
4 names = data/vochelmet.names
5 backup = backupH/
```

We also changed the filters and classes in the cfg/yolov3-train_helmet.cfg file.
Formula is filters = (classes+5) * 3 in yoloV3 (3 masks only)

filters=21 (2+5) x 3

classes=2

Prepare trainH.txt and testH.txt for the image dataset.
Sample:
/home/graymatics/models/vivek/dataset/NoHelmet0003.jpg
/home/graymatics/models/vivek/dataset/NoHelmet0004.jpg
/home/graymatics/models/vivek/dataset/NoHelmet0006.jpg

Download Pretrained Weights:
wget https://pjreddie.com/media/files/darknet53.conv.74

2.6. Train the Model

Run the command with GPU:
./darknet detector train cfg/ vochelmet.data cfg/yolov3-train_helmet.cfg darknet53.conv.74 -gpus 0

2.7. Test the Model using python codes.

The final weight is used to test images and videos. The following 2 programs were used to test images and videos. For the video detection, only frames with helmet/no_helmet detected are saved to an output AVI file.


Helmet_detection_image.py


Helmet_detection_video.py

```
# Load names of classes
classesFile = "models/vochelmet.names"

# Give the configuration and weight files for the model and load the network using them.
modelConfiguration = "models/yolov3_deploy_helmet.cfg"
modelWeights = "models/yolov3-helmet_final.weights"

net = cv.dnn.readNetFromDarknet(modelConfiguration, modelWeights)
net.setPreferableBackend(cv.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv.dnn.DNN_TARGET_CPU)
```

3. Findings

Key findings are as follows

3.1. CCTV Images/Video for training model

Was not readily available with the industry partner and in the essence of time we started arranging for it from the internet. We needed lots of training data and to get those we needed a tool which could download in bulk. Bulk Image Downloader [4] tool was just right for the job. We were then able to select relevant images suitable for training and download them efficiently.

3.2. Training model on the server

Without GPU the training was extremely slow and often generated errors. As we later found out through our research and assistance from the industry partner, the parameters to be changed and use of DOCKER containers to address the speed and errors.

3.3. Viewing output images/videos

Was not possible on the server. This was needed to try out the model detection results. So as a workaround we used alternatives like running on Google COLAB and personal laptops machines while the industry partner resolved it on the server.

4. Evaluation and Analysis

- Command used to measure model performance in terms of mAP (mean average precision)

  ./darknet detector map data/obj.data cfg/yolov3_helmet.cfg
  /mydrive/yolov3/backup/yolov3_helmet_last.weights

- Output is as follows:

  [yolo] params: iou loss: mse (2), iou_norm: 0.75, cls_norm: 1.00, scale_x_y: 1.00

  Total BFLOPS 65.312
  avg_outputs = 516922
  Allocate additional workspace_size = 52.43 MB
  Loading weights from /mydrive/yolov3/backup/yolov3_helmet_last.weights...
  seen 64, trained: 640 K-images (10 Kilo-batches_64)
  Done! Loaded 107 layers from weights-file

  calculation mAP (mean average precision) ...
  8

  detections_count = 21, unique_truth_count = 13

  class_id = 0, name = helmet, ap = 78.98% **(TP = 10, FP = 2)**
  class_id = 1, name = no_helmet, ap = 12.50% **(TP = 0, FP = 1)**

  for conf_thresh = 0.25, **precision = 0.77, recall = 0.77, F1-score = 0.77**
  for conf_thresh = 0.25, **TP = 10, FP = 3, FN = 3**, average IoU = 60.07 %

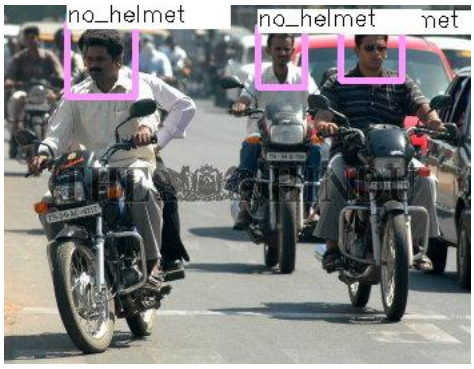
  IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
  mean average precision (**mAP@0.50**) = 0.457386, or 45.74 %
  Total Detection Time: 1 Seconds

- mean Average Precision (mAP): mean average precision (mAP@0.50) = 45.74 %

- Confusion matrix: precision = 0.77, recall = 0.77, F1-score = 0.77

5. Results

The results are very promising given the time and resources. In fact, our model is able to fairly (almost 80%) annotates riders riding with or without a HELMET on the roads.



Conclusion

Wearing “Personal Protection Equipment” (PPE) like the HELMET is critical for the safety of road users – especially for the two-wheelers and detecting them through AI video analytics equally crucial to the success of the smart-city project.

Road traffic video through CCTV and such are fast paced in terms of subject moving in the frames fast and appearing for a short period of time. “You Only Look Once” (YOLO)v3 AI model handles it well and is able to detect objects quickly.

Recommendations

There is always space for improvement in terms of accuracy and recall.

- CCTV/ Traffic camera video feeds if used as training data will improve the accuracy of the model.

For the Traffic regulating agency to enforce compliance

- Alerts from this AI model to feed into a dashboard along with alerts from all other algorithms generating complete information of the errant drivers.

Appendices

I. TIPP_AAI Capstone Project - Graymatics Handover document.

TIPP_AAI Capstone Project - Graymatics Handover document.

VB **Vivek Bhatnagar <vivek_bhatnagar@outlook.com>**
2/4/2020 10:21 AM

To: Alex Kaiser Cc: ags@graymatics.com; Jimmy Goh; David Tan

Handover.docx
20.44 KB

Dear Alex,

Please find attached the handover document summarising details of the CODE, CONFIG, DATA and the RESULTS.

We are pleased to inform that we have completed the necessary YOLOv3 model training, deployment and detection of the HELMET as part of the smart-city project.

Our model processes a traffic video file and is able to fairly (based on the available training data) annotates riders riding with or without a HELMET on the roads.

Today is our last day at Graymatics for our TIPP_AAI Capstone project and we wish to thank the entire team for the support and guidance provided.

best, Vivek & David

References

[1] Redmon, J. a. (2018). YOLOv3: An Incremental Improvement. *arXiv*.

[2] Redmon, J. (2020, March). Retrieved from YOLOv3: https://pjreddie.com/darknet/yolo/

[3] singh, v. (n.d.). Retrieved from github: https://github.com/BlcaKHat/yolov3-Helmet-Detection

[4] Bulk Image Downloader https://bulkimagedownloader.com/

[5] GitHub, Inc. https://github.com/AlexeyAB/darknet

[6] Redmon, J. (2020, March) Retrieved fom https://pjreddie.com/media/files/darknet53.conv.74