Journal of Electronics and Information Science (2021) 6: 21-26
Clausius Scientific Press, Canada

DOI: 10.23977/jeis.2021.060204
ISSN 2371-9524

Multi-Person Detection of Drivers Based on Yolo Network

Xiaoyu Xian, Yin Tian, Haichuan Tang, Qi LIU

Crrc Academy Co., Ltd., Beijng 100070, China

Keywords: Objective detection, Image process, Deep learning

Abstract: Subway train drivers abide by the operations requirements to routinely check a myriad of system parameters and indicators to ensure safe operation. It is important to ensure that the driver have correctly performed the entire set of routine operations without omission. It is therefore hoped that introducing real-time monitoring to the on-board surveillance system can replace human efforts in favor for improved safety on the driver's side. In this paper we investigate the objective detection methods to accomplish open pose estimation. We take a good method in doing such task as it satisfies all the requirements: real-time, high accuracy, works for both RGB and greyscale input, multi-person detection, invariant to rapid switch from darkness to brightness, consistent performance in low or even middle noise input situation.

1. Introduction

The You Once Look Once (Yolo)[1] is a state of art, read-time objective detection system. Since 2015 when R-CNN[2] was published, using convolutional neural network for objective becomes the trend as convolutional neural network has the advantage of invariant to twist or transformation and has extremely good performance on image classification. The backbone of most objective detection methods like R-CNN at the time was ImageNet (Like AlexNet, and VGG-16)[3] . Those detection systems are two-stage objective detection method as they repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections. Yolo is the first algorithm that use one stage objective detection, namely Yolo 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. These bounding boxes are weighted by the predicted probabilities. This makes Yolo 1000x faster than R-CNN, and 100x faster than faster R-CNN. In Yolo v2, it switches from the backbone VGG-16 to Darknet -19 to achieve better accuracy. In Yolo v3, it builds the Darknet-53 to guarantee high speed and at the same time, increases its performance on accuracy especially on small objective detection.

Note that RetinaNet takes like 3.8x longer to process an image. YOLOv3 is much better than SSD variants and comparable to state-of-the-art models on the AP50 metric.

2. Model Architecture and Training

We uses the backbone DarkNet 53 to extract the features, its structure is attached on the right. One could also train Yolo other dataset like on VGG-16 or COCO, that will result in different pre-trained weights. It is highly flexible and could be adjusted to different tasks and settings. Here for our project, we restrict the detection objective to be person, resulting in an even easier training process as we only need the feature extraction of person. Note that a good thing about Yolo is that the tradeoff between speed and accuracy is extremely easily by changing the size of model, it does not require retraining the whole model.

3. Implementing Yolo on Grayscale and Noisy Input

As the output of the detection will be used later for the task of pose estimation and status learning, we need to make sure that the result of Yolo is robust and constituent in greyscale input and noisy environment.

The result shows that even with full greyscale input, Yolo still gives good detections. Later we notice that the rapid shifting from light to dark requires that the detection system to be invariant to color changes. We once tried to retrain the classifier with greyscale image dataset. However, this is not a good idea as the input is actually RGB videos and images, thus training with greyscale image will result in some information loss in prediction. Additionally, by experiments we notice that the prediction for greyscale input is quite good even with classifier of RGB trained weights.

The testing platform is a personal computer with a Nvidia GeForce GTX 1080 graphics card, 16GB Memory, Intel 8700 CPU. Note that Yolo detects objective in video frame by frame, and as Yolo works at speed more than 30fps for general 1080p image, it produces the real-time objective detection for video under 30Hz to 60Hz, depending on the size of the input video and hardware performance.

We will demonstrate the experiment that we have done with Yolo v3 in different situations. First, we will show the detections in normal RGB image and its comparison in greyscale image.

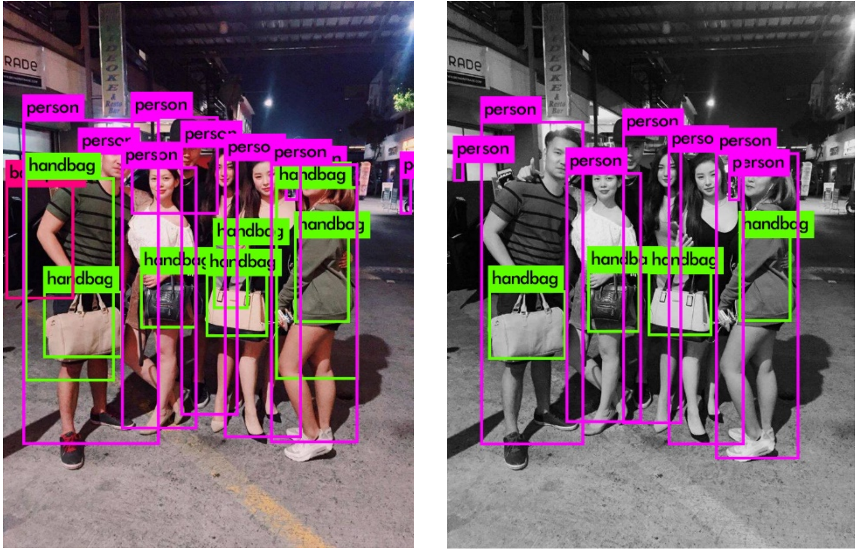

Figure 1. Performance comparison of YOLO v3 on RGB (left) and grayscale (right) input

It’s easy to see that under greyscale, Yolo achieve an even better prediction as it did not square the person head twice. Notice that the case demonstrated here are quite complicated as there are multiple people and most of them has overlapping parts.

Next, we show that the detection of people in extreme light cases: extremely dark and extremely bright.

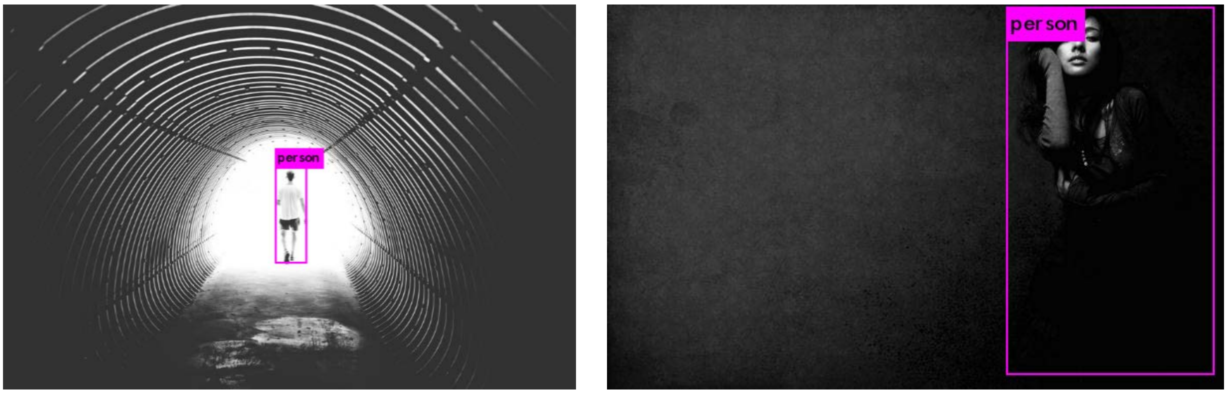

Figure2. Demonstration under extreme lighting conditions

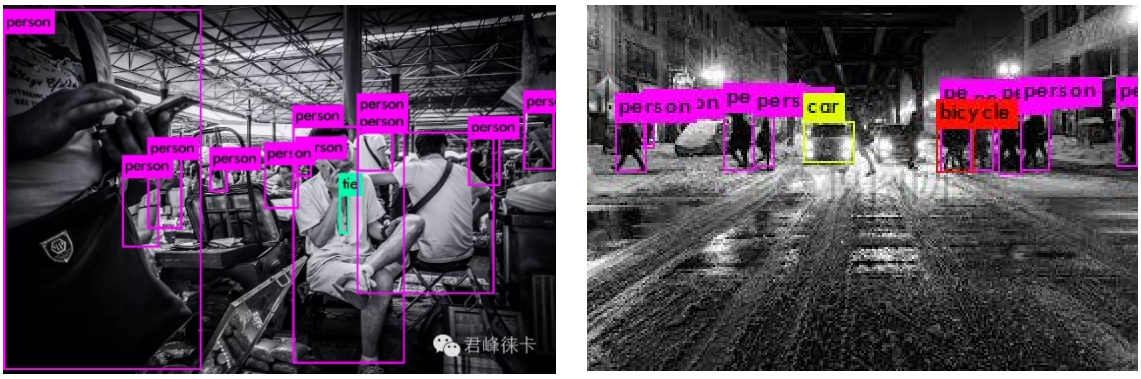

Figure 3. Demonstration under multi-person large size (left) and multi-person small size (right)

Notice that Yolo successfully detects the person in both situations with even greyscale input.

Next, we show multi-person detection in greyscale situation.

Note that the image on the left has a much high resolution than the image on the right, yet they both show good performance of detection on different size of image.

Finally, as dark environment sometimes will bring much of noise into video or image. We test the performance of Yolo on video data and manual add noise to the video and see how it performs. The video we used here is from a greyscale old movie, we test three situations: low noise (20%), moderate noise (40%), and high noise (60%).

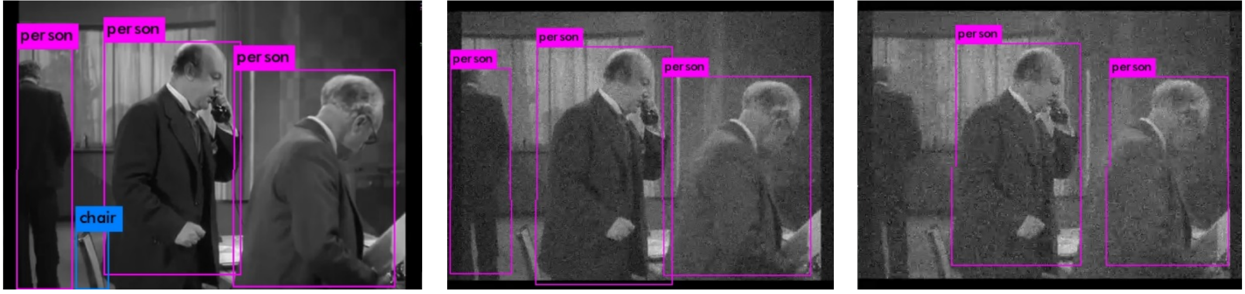

Figure 4. Performance of YOLO v3 under increasing noise level

The result is shown below. Note that in low noise situation, the algorithm could still detect the people yet loses some detections on small objective. In middle noise situation, the algorithm still detects most people while begins to fail on detecting some people whose boundary is mixed with noise and environment, and the square the algorithm gives is not that correct as it did in no noise situation. In high noise situation, the algorithm almost fails on detection on everything. It is quite understandable as in high noise situation, the boundary of objective is covered mostly by noise.

4. Algorithm and Experimental Results

The “State Farm Distracted Driver Detection” dataset (or “State Farm Dataset”) consists of 22424 images in RGB format (here for the sake of convenience, we have transformed all pictures into gray scales)[4]. These images are classified into 10 categories, such as safe driving, texting using the right hand, drinking and so on, each image exactly belongs to one category, and the images are distributed into all the classes approximately evenly, we can check the figure below for illustration. We can find that all the images are shot from a fixed perspective, and the backgrounds are all in the daytime with a good weather and great light condition. Here our main goal is to design algorithms to classify each image to determine their behavior status.

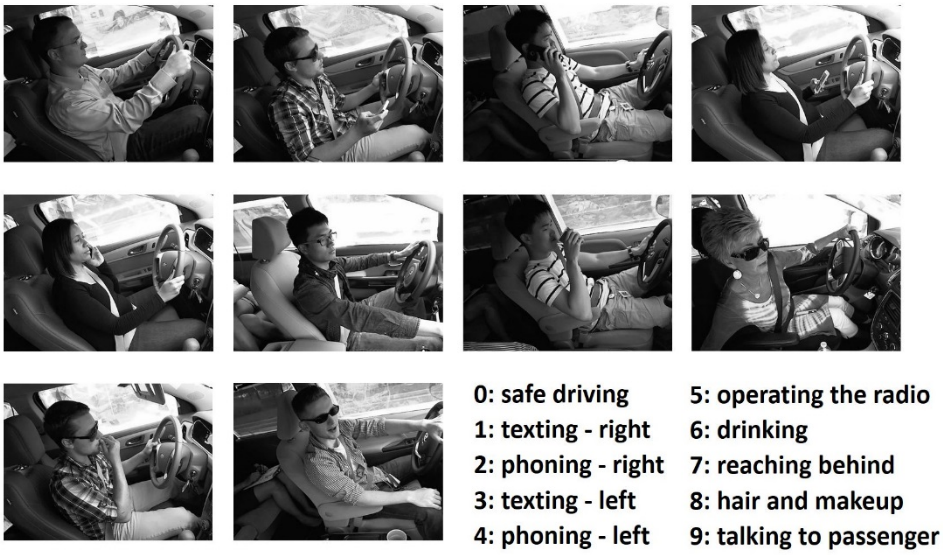

Fig.5 State Farm Dataset Example Images in Grayscale

Here we will use Convolutional Neural Network (CNN) to design the recognition system. CNN is mainly composed of input layers, convolution layers, pooling layers, activation functions and fully-connected layers, here we will not discuss the technical details of CNN, the figure below provides an example of CNN architecture.

Here we used Keras to build CNN with three convolution layers. The size of input layer of the input layer is 150×150, the three convolution layers attain sizes as 32×3×3, 32×3×3 and 64×3×3, respectively. The activation functions are all ReLU, a pooling layer are attached with each convolution layer. Then we split the State Farm dataset into training and test set, and feed into the network. After the previous process, we can output the trained model and use it for real tasks.

In State Farm dataset, there are 22424 images in total, we randomly select a certain proportion of images as the training set and the rest works as the testing set. Each training will proceed for 10 epochs. Here we mainly focus on two problems: first, how is the accuracy of our CNN architecture; second, how the accuracy is affected by the size (proportion) of the training set. We illustrate our experimental result in the following figure.

The six lines in the figure can be roughly categorized into three classes based on the amount of the training size: using roughly 90% of the dataset (20924 images), roughly 30% of the dataset (7324 images) and roughly 20% of the dataset (4424 images) as the training set. We can find from the result that: first, after 10 epochs, all the accuracy (progress) have basically become stable; second, from the tendency of each line, we can find that larger size of training set can effectively increase the prediction accuracy, for the 90% case, the final accuracy is about 98%, while for the 30% and 20% cases, the values are about 90% and 80%. But from another perspective, we should notice that smaller training size can effectively decrease the workload and increase the training speed. Also, the prediction accuracy does not have a huge downfall, so more investigations can be executed to check the possibility of using smaller dataset for training.

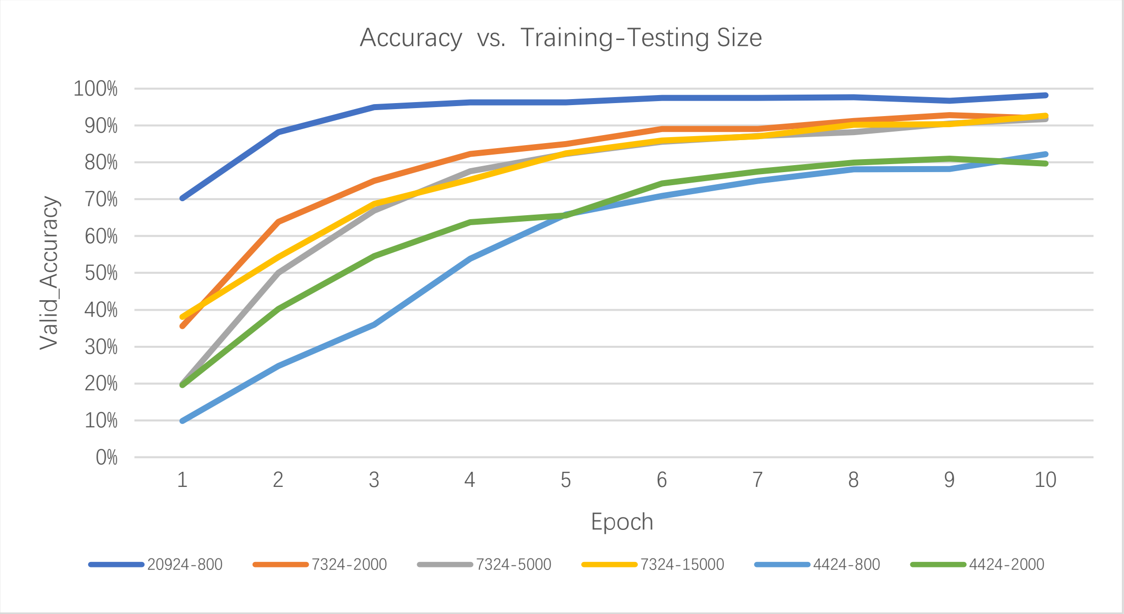

Fig.6 Visualization of Training Results

5. Conclusion

A lot of research and industrial effort have been devoted to objective detection area, and many good algorithms has been developed, like faster like faster R-CNN , F-RCN [5], SSD[6] and Yolo. In order to complete the detection of the driver's behavior, the key issue is how to accurately complete the real-time detection of the driver's body in the complex and changeable subway environment. The multi-person detection of drivers requires real time feedback: namely we need to have at least 30 FPS for just the objective detection part. Due to the specialty of subway driving environment, this detection system needs to have a consistent behavior in the situation of environment rapidly switching from darkness to lightening, RGB scale video input to greyscale video input, and video input with a lot of noise.

Based on the above requirements, we trained a multi-person detection algorithm based on the Yolo network, and completed experiments such as gray-scale input detection, noise testing, and brightness change detection, which proved the adaptability of the model to the changing environment and laid the foundation for subsequent behavior identification.

References

[1] Redmon, Joseph, Ali Farhadi. "Yolov3: An incremental improvement". arXiv preprint arXiv:1804.02767, pp.1-8, April, 2018.

[2] Ren, Shaoqing, He kaiming, Girshick Ross ,et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems, vol.39, no.6, pp. 1137 - 1149, May, 2015.

[3] Ross B. Girshick, Jeff Donahue, Trevor Darrell, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." IEEE conference on computer vision and pattern recognition, abs/1311.2524, pp.1-9, June, 2014.

[4] "State Farm Distracted Driver Detection | Kaggle." [online] Available: www.kaggle.com/c/state-farm-distracted-driver-detection.

[5] Dai, KHJS Jifeng, Yi Li R-fcn. "Object detection via region-based fully convolutional networks" .NIPS, pp.379-387, May,2016.

[6] W Liu, D Anguelow, D Erhan, et al. "Ssd: Single shot multibox detector". European conference on computer vision. Springer, Cham, pp.21-37, October, 2016.