Comparative Evaluation of Finetuned Faster R-CNN Model on Dark Images Using Different Architectures

Anket Sah, Amala Chirayil, Ksheeraj Vepuri, Kriti Gupta, Sanmesh Bhosale
(https://github.com/ksheeraj/CS256-AI-ObjectDetection)

Abstract - **Human and object detection is one of many subject areas that is incessantly explored in computer vision. The progress that has been made thus far has given rise to several applications that include video surveillance systems being able to detect humans and other objects within the vicinity. This progress is a direct result of the approaches researchers have developed over the years, from classical approaches to deep learning approaches.**

Even though significant progress has been made in this subject area, an observation made by researchers Yuen Peng Loh and Chee Sang Chen on the lack of low-light images [1] led us to investigate how state-of-the-art (SOTA) convolutional neural networks (CNNs) perform on low-light images. Therefore, in this paper, we are going to be incorporating SOTA CNNs, specifically Faster R-CNN, in distinct architectures and examine the performance of this CNN on low-light/dark images.

Index Terms: Faster R-CNN, EnlightenGAN, Detectron2, CLAHE, USM, Deep learning

I. INTRODUCTION

Optical character recognition (OCR), self-driving cars, pedestrian detection, medical imaging, and visual search engines are few examples of human and object detection applications that exist today. These applications are a result of the persistent progress being made within the human and object detection subject area of computer vision. This persistent progress is revealed by the large number of state-of-the-art (SOTA) convolutional neural networks (CNNs) that have been trained on several available datasets such as PASCAL VOC and MS COCO for human and object detection research.

Many SOTA CNNs have rightfully earned the title of “state-of-the-art” after making significant breakthroughs. However, according to researchers Yuen Peng Loh and Chee Sang Chen, these breakthroughs were a result of the large availability of visible-light images. In fact, “less than 2% of total images were low-light data in successful public benchmark datasets such as PASCAL VOC, ImageNet, and Microsoft COCO” [1]. In order to reduce the lack of low-light images, these researchers produced a low-light, annotated image dataset called Exclusively Dark.

Just as it’s important to detect human and objects in visible-light images, its equally important to recognize, detect, and classify these same objects in low-light images since “low-light environment is an integral part of our everyday activities” [1]. When our surroundings become less visible as the amount of visible light decreases, we are risking our safety and security. For instance, a self-driving car should perform well in a low-light or dark environment because if any human or object goes unnoticed, there is a high probability of a deadly event occurring. Inspired by the observations made by the researchers of University of Malaya [1], we will be examining the performance of a SOTA real-time object detection CNN called Faster R-CNN, on low-light images by incorporating it into different architectures which are constructed from a combination of image filters and another

CNN called EnlightenGAN [6].

The paper is structured as follows. Section II provides a literature survey of classical approaches and deep-learning approaches for human and object detection. Section III then provides a methodology of our project and relevant background information related to our project. Section IV dives deeper into implementation details. Subsequently, Section V provides the results of the implementations. Lastly, Section VI wraps up our project and we also discuss the future work that could be implemented by using our current project as its starting foundation.

II. LITERATURE SURVEY

Prior to using deep learning approaches for addressing human and object detection problem in computer vision, a few classical approaches based on feature extraction and machine learning algorithms were used. We will be examining two classical approaches, Viola-Jones Object Detection Framework [2] and Histogram of Oriented Gradients [3], and a deep learning approach called Faster R-CNN [4] by providing a concise description of each in Sections A and B, respectively.

A. CLASSICAL APPROACH

i. Viola-Jones Object Detection Framework: Paul Viola and Michael Jones proposed and developed the first object detection framework to provide near real-time object detection. This framework is a culmination of three key contributions: integral image, learning algorithm based on Adaboost, and cascade structure of classifiers [2]. Each of these three components play a critical role in the overall framework.

   Integral image is a type of image representation that speeds up the process of feature evaluation so that any feature can be computed in constant time. Adaboost, short for Adaptive Boosting, is a machine learning algorithm that combines weak classifiers into a single, strong classifier. Viola and Jones built a classifier by “selecting a small number of features using Adaboost” [2]. The last contribution of this framework is a cascade structure of classifiers that increased the detector’s speed by focusing on high-probable regions of the image.

   For more information on this framework, please refer to [2] where the authors provide detailed explanations of their experiments.

ii. Histogram of Oriented Gradients: In [3] Surasak et al. proposed human detection technique called Histogram of Oriented Gradients (HOG) for detecting humans in each frame of a video. The primary focus of HOG is to differentiate the objects from the background, thus making the process of human or object detection more efficient and coherent. The results of the authors’ work present an average accuracy of 81.23% with a standard deviation (SD) of 10.95%.

For human and object detection in images, comparing classical approach and deep learning approach, deep learning approach has much better results as shown by Mahony et al. in [5]. Therefore, in our research work we looked further deep into various Deep Learning Approaches for better classification of humans and objects in images.

B. DEEP LEARNING APPROACH

i. Faster R-CNN: Faster R-CNN is a region-based CNN, that is composed of two networks, a Regional Proposal Network (RPN) and an object detection network. The RPN is used to generate region proposals that most likely contain objects and these region proposals are fed into an object detection network layer to detect the objects. In comparison with its predecessor, Fast R-CNN which uses Selective Search to generate region proposals, Faster R-CNN introduce “novel RPNs that share convolutional layers with state-of-the-art object detection networks. [4]” Hence, the time it takes to compute region proposals is small (e.g., 10ms per image).

III. METHODOLOGY

In order to evaluate Faster R-CNN on low-light/dark images, we incorporated the model into different architectures. Each of these architectures can be seen in Section A, below, along with a high-level overview of the architecture diagram itself.

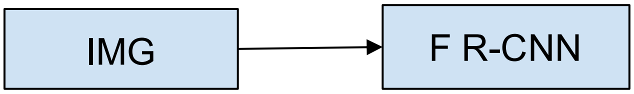
A. ARCHITECTURE DIAGRAMS

In order to understand the architecture diagrams below, refer to the legend below that describes each of the components in the diagrams. The implementation details of each of these components can be found in Section IV of the report.

Legend	
Component Name	Description
IMG	IMG refers to the low-light/dark image data from the Exclusively Dark Dataset [1] that we will be feeding into Faster R-CNN, EnlightenGAN, CLAHE, and USM in their respective architecture diagrams.
COCO Imgs	MS COCO Dataset
F R-CNN	F R-CNN refers to Faster R-CNN. Background information on Faster R-CNN can be found in Section B of the Introduction (Section I).
EG	EG refers to EnlightenGAN [3], which is an unsupervised generative adversarial network, that proposes to regularize the unpaired training data using the information extracted from the input itself, along with a global-local discriminator structure, an attention mechanism and self-regularized perceptual loss function.
CLAHE	CLAHE, which stands for Contrast Limited Adaptive Histogram Equalization, is a variant of adaptive histogram equalization in which the contrast amplification is limited to reduce the problem of noise amplification. CLAHE computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. Hence it is suitable for improving the local contrast and enhancing the definitions of the edges in each section of the image [7].
USM	USM is an image sharpening technique. Unsharp mask is a filter that amplifies the high-frequency components of an image. The resulting image is less blurry than the original image with higher contrast and brightness. Although the resulting image is clearer it may be a less accurate representation of the image’s subject [8].
Finetune	Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is

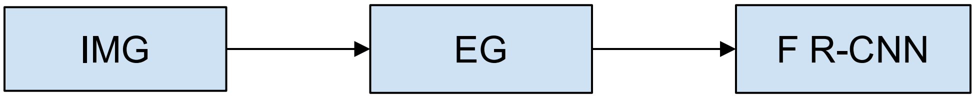
	a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems [9].

i. Architecture Diagram 1



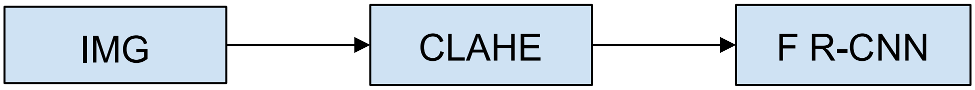
We will be running a dark image through Faster R-CNN to see how it performs on dark images.

ii. Architecture Diagram 2



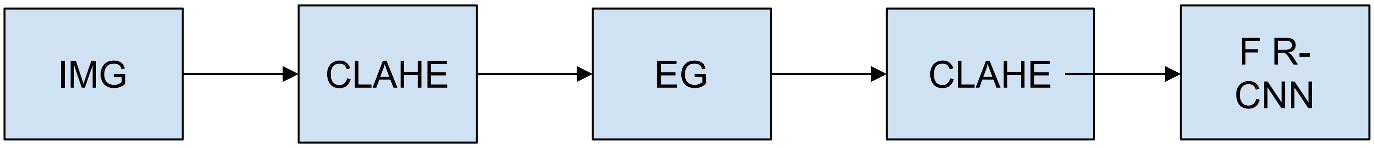
We will be feeding in dark images through EnlightenGAN in order to improve/enhance the lighting conditions in the input dark images. Then, the output of the EnlightenGAN will be fed into Faster R-CNN for object detection. As you will see in the results section (Section V), this step produces the best results for low-light images.

iii. Architecture Diagram 3



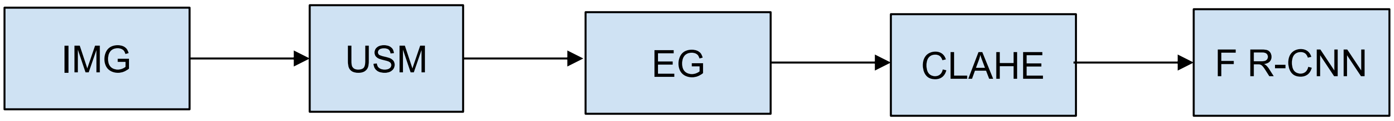
We will be feeding in dark images through CLAHE and then take the output produced by CLAHE in order to feed into Faster R-CNN for object detection.

iv. Architecture Diagram 4



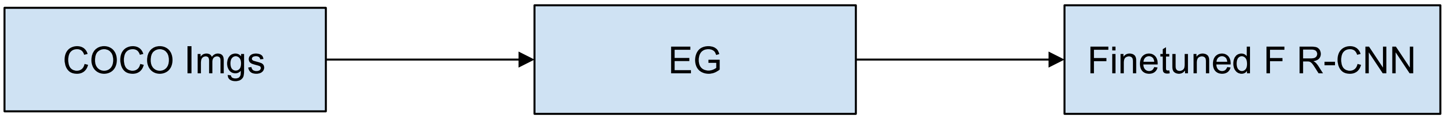
We will be feeding in dark images through CLAHE to enhance the image and output of CLAHE through EnlightenGAN which result in a bright image. Then, taking this bright image, we will run it through CLAHE again and finally through Faster R-CNN.

v. Architecture Diagram 5



We will be feeding in dark images through a sharp bright filter called USM. Then, taking the output from USM we will be feeding it into EnlightenGAN which will result in a bright image. The resultant image after these two steps is fed into CLAHE and finally through Faster R-CNN.

vi. Architecture Diagram 6



We will be feeding in 5000 images from MS COCO dataset from 2017 into EnlightenGAN in order to produce bright images of the MS COCO dataset. Then we will be taking this output produced by EnlightenGAN and use it to finetune the Faster R-CNN model.

IV. IMPLEMENTATION DETAILS

An abundance of Faster R-CNN implementations exist that use different deep learning frameworks, such as PyTorch and TensorFlow. For our project, we decided to use the Faster R-CNN model provided by Detectron2, which is a PyTorch-based modular object detection library developed by Facebook AI Research (FAIR). Detectron2 supports many variations of the Faster R-CNN model. The model that we decided to use was implemented with Residual Network 50 (ResNet50) as the backbone and Feature Pyramid Network (FPN) and uses 3x as its learning rate.

Prior to using Google COLAB to execute the Faster R-CNN model provided by Detectron2, we executed the model on AWS. An AWS EC2 instance with the Deep Learning Base AMI (Ubuntu 16.04) Version 19.3 that used the p2.xlarge GPU instance was set up in order to run this model. We utilized the official documentation provided by Facebook Research to run the model [10]. In order to save on cost, we also executed Faster R-CNN on Google COLAB to run inference as well finetune the model on our custom dataset. Further details can be seen on our GitHub page (*https://github.com/ksheeraj/CS256-AI-ObjectDetection*).

We finetuned the Faster R-CNN model by applying transfer learning. We implemented transfer learning on the pre-trained Faster R-CNN model using two different datasets. The first dataset consists of 100 images which is composed of images from the Exclusively Dark dataset [1] and our own set of low-light images. For this dataset, we used ImgLab, which is an image annotation tool, to annotate our custom dataset into COCO data format [11]. The reason we decided to format the annotation using COCO data format is because Detectron2 supports this format. The second dataset consists of 5000 images from the COCO_val2017 dataset. You can find the finetuned model on our GitHub page (*https://github.com/ksheeraj/CS256-AI-ObjectDetection*).

As can be seen in the architecture diagrams provided in the subsection A of Section III of the report, Faster R-CNN is a common component in each of those diagrams. Other components that exist in the diagrams as a whole are Contrast Limited Adaptive Histogram Equalization (CLAHE), EnlightenGAN (EG), and Unsharp Mask (USM). Explanations of each of these components can be found in subsection A of Section III. CLAHE is a histogram-based filter that is available in OpenCV, which is a Python library. USM is a filter used to sharpen the edges in an image. The implementation of these two image filters can be found on our GitHub page (*https://github.com/ksheeraj/CS256-AI-ObjectDetection*). EG was implemented on an AWS EC2 instance with the Deep Learning Base AMI (Ubuntu 16.04) Version 19.3 that used the p2.8xlarge GPU instance with torch (0.3.1), torchvision (0.2.0), visdom server, and dominate Python library.

V. RESULTS

The results for each of the architecture diagrams in subsection A of Section 3 can be found in the figures below. In addition, we replaced the Faster R-CNN component of architecture diagrams 1-5 with our finetuned Faster R-CNN model to see how it performed. These figures showcase the result of running Faster R-CNN and finetuned Faster R-CNN on low-light images. Figure 11 showcases sample images from finetuning Faster R-CNN using Transfer Learning on COCO dataset The figures can be found in the index, which is located at the end of the report.

i. Model 1: Figure 1, Figure 2

ii. Model 2: Figure 3, Figure 4

iii. Model 3: Figure 5, Figure 6

iv. Model 4: Figure 7, Figure 8

v. Model 5: Figure 9, Figure 10

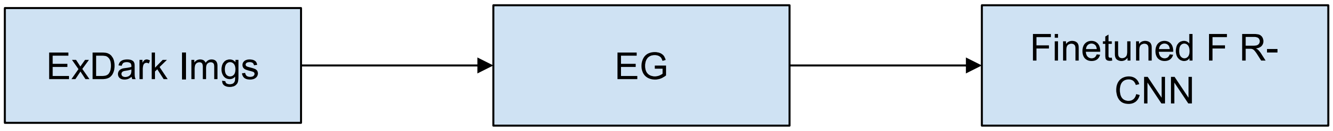
The table below provides an accuracy comparison of running Faster R-CNN and finetuned Faster R-CNN on low-light images. We computed the accuracy by dividing the total number of detected objects by the total number of objects in the image.

Table 1. Accuracy Comparison Table

Model	Faster R-CNN Accuracy	Finetuned Faster R-CNN Accuracy
1	55%	72%
2	33%	75%
3	11%	33%
4	27%	41.66%
5	50%	52.77%

VI. CONCLUSION AND FUTURE WORK

The goal of this project was to evaluate the performance of Faster R-CNN by incorporating it with a combination of different filters and convolutional neural networks. Even though, we were able to get a good accuracy for our finetuned Faster R-CNN model, we know that there is more to discover and work on. For our future work, we would like to finetune Faster R-CNN with the Exclusively Dark dataset as showcased in the architecture diagram below.



VII. REFERENCES

[1] Yuen Peng Loh, Chee Seng Chan, “Getting to Know Low-light Images with the Exclusively Dark Dataset”, 29 May 2018.

[2] Paul Viola, Michael Jones, “Robust Real-time Object Detection”, 13 July 2001.

[3] Thattapon Surasak, Ito Takahiro, Cheng-husan Cheng, Chi-en Wang, Pao-you Sheng, “Histogram of Oriented Gradients for Human Detection in Video”, 2018 5th International Conference on Business and Industrial Research (ICBIR), IEEE.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, 6 Jan 2016.

[5] Niall O’ Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Deniel Riordan, Joeseph Walsh, “Deep Learning vs. Traditional Computer Vision”, Computer Vision Conference (CVC), 2019.

[6] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang, “EnlightenGAN: Deep Light Enhancement without Paired Supervision”.

[7] Victor T. Tom, Gregory J. Wolfe, “Adaptive Histogram Equalization And Its Applications”, 26th Annural Technical Symposium, 17 March 1983.

[8] Artyom M. Grigoryan, Sos S. Again, “Unsharp Masking”, Advances in Imaging and Electron Physics, 2004.

[9] Jindong Wang, Yiqiang Chen, Han Yu, Meiyu Huang, Qiang Yang; ‘EASY TRANSFER LEARNING BY EXPLOITING INTRA-DOMAIN STRUCTURES’

[10] https://github.com/facebookresearch/detectron2

[11] http://cocodataset.org/#download

INDEX

Model 1:

Figure 1: Faster R-CNN

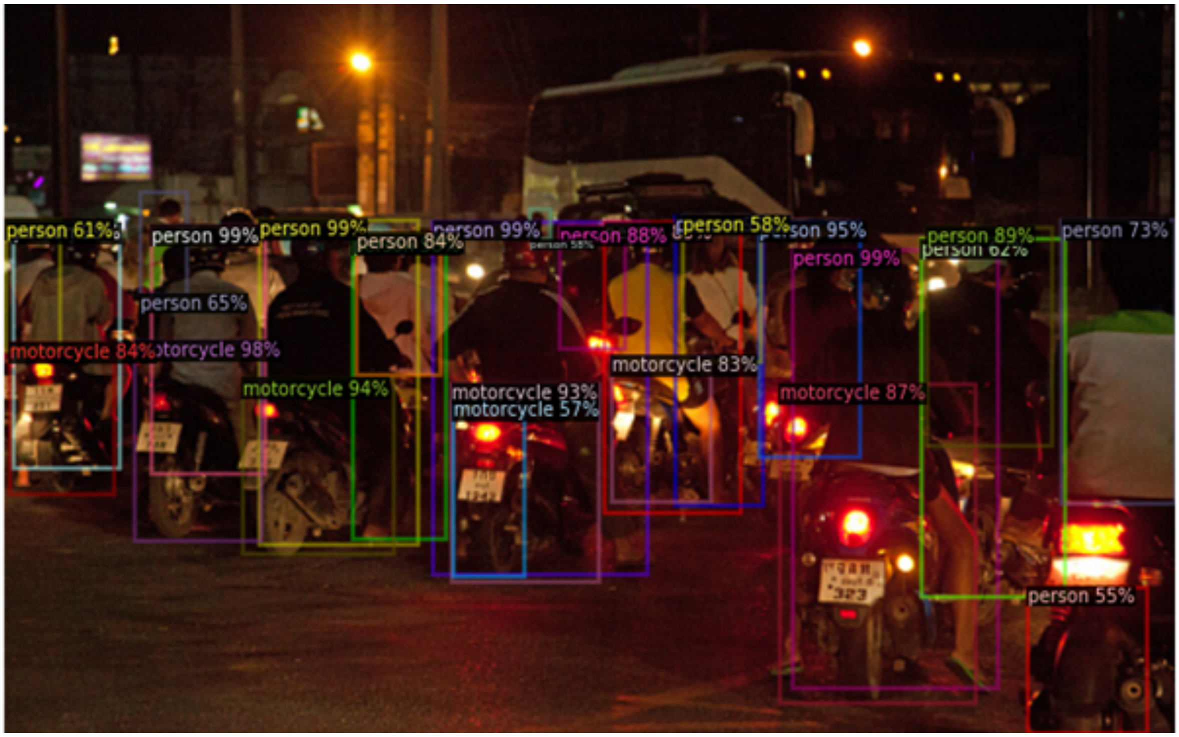
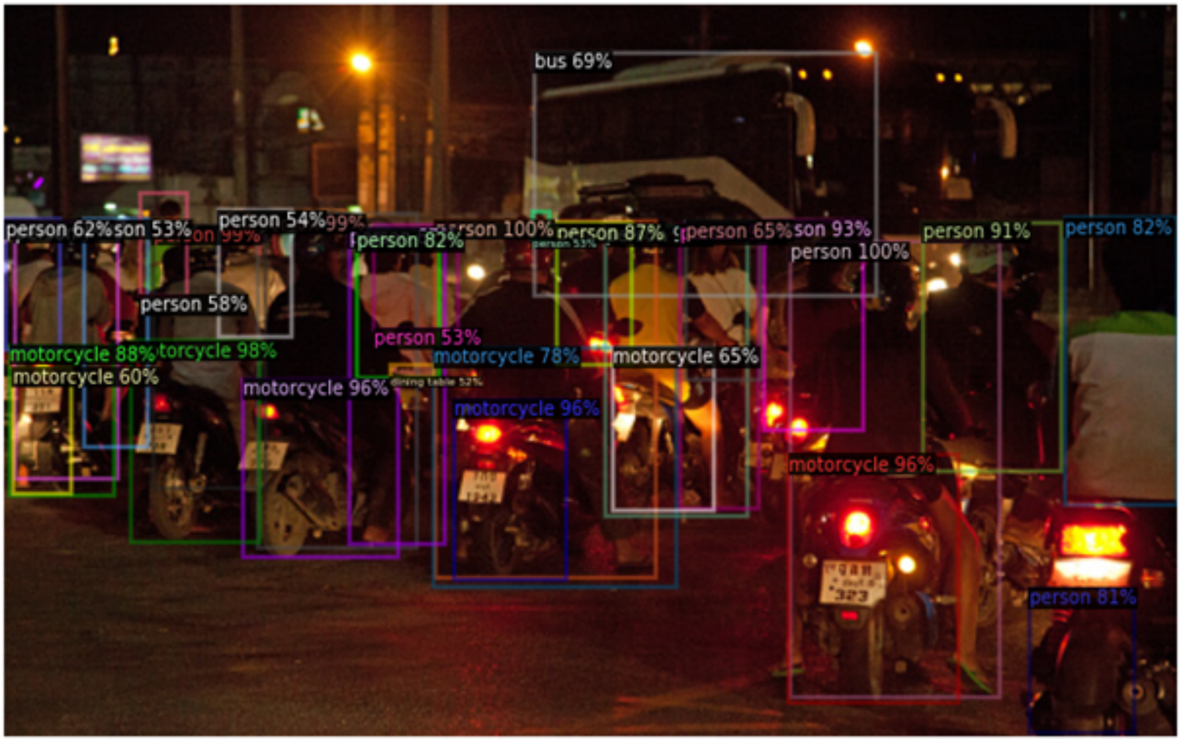


Figure 2: Finetuned Faster R-CNN



Model 2:

Figure 3: Faster R-CNN

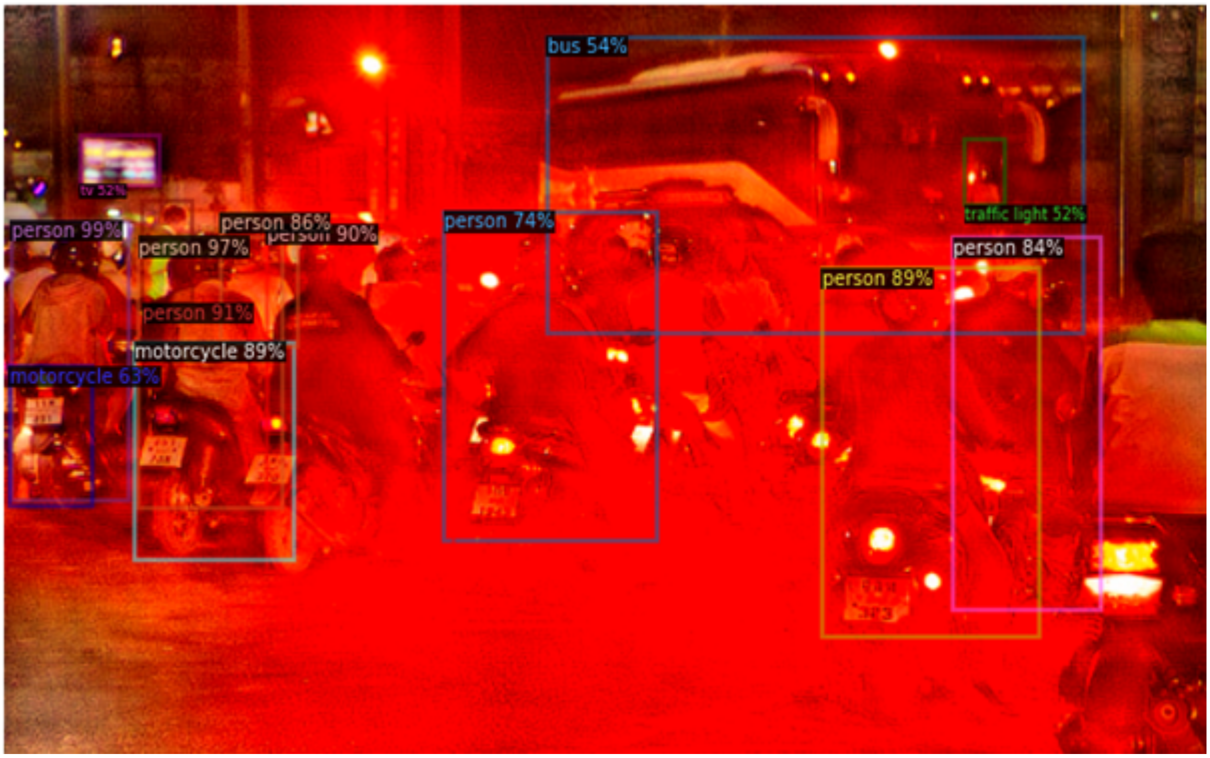
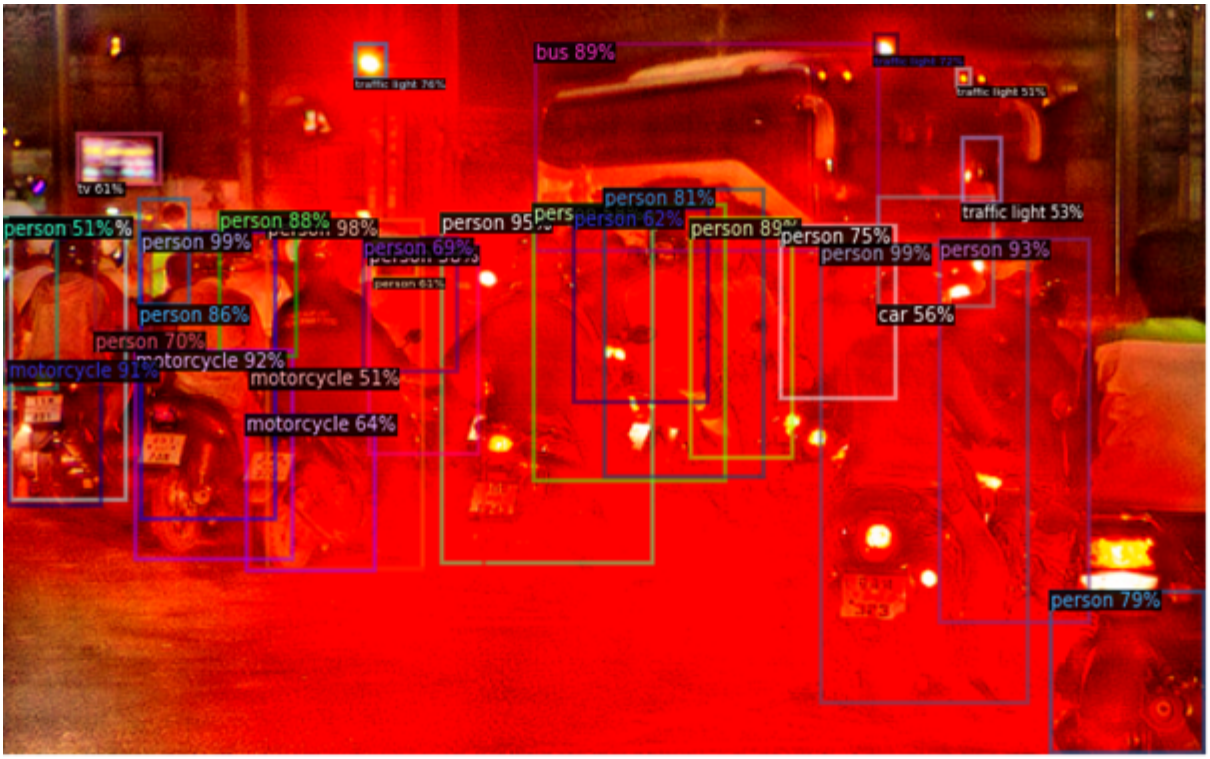


Figure 4: Finetuned Faster R-CNN



Model 3:

Figure 5: Faster R-CNN

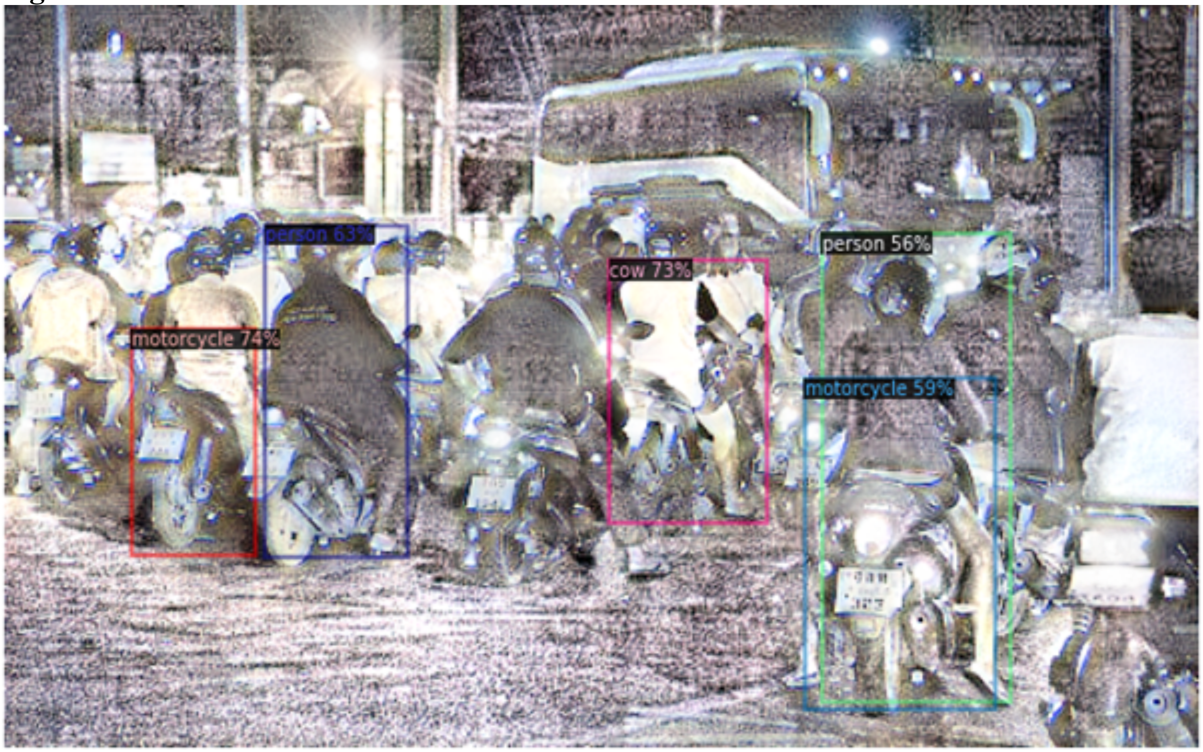
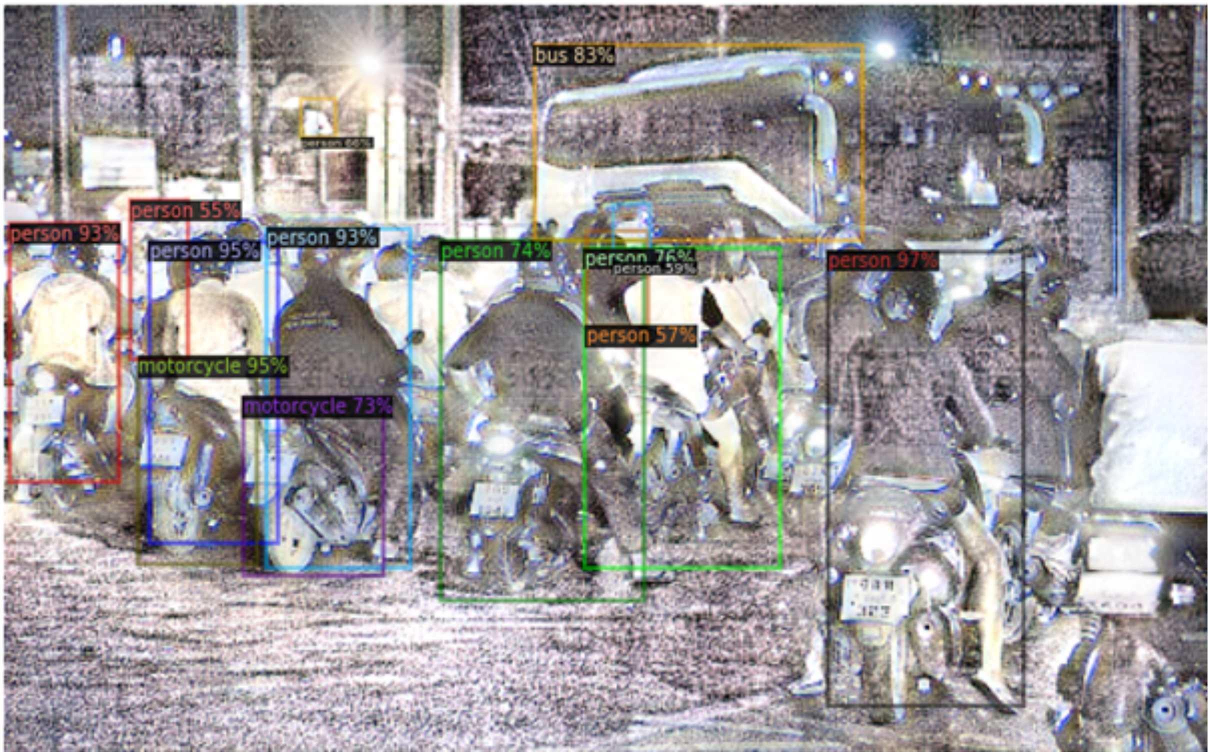


Figure 6: Finetuned Faster R-CNN



Model 4:

Figure 7: Faster R-CNN

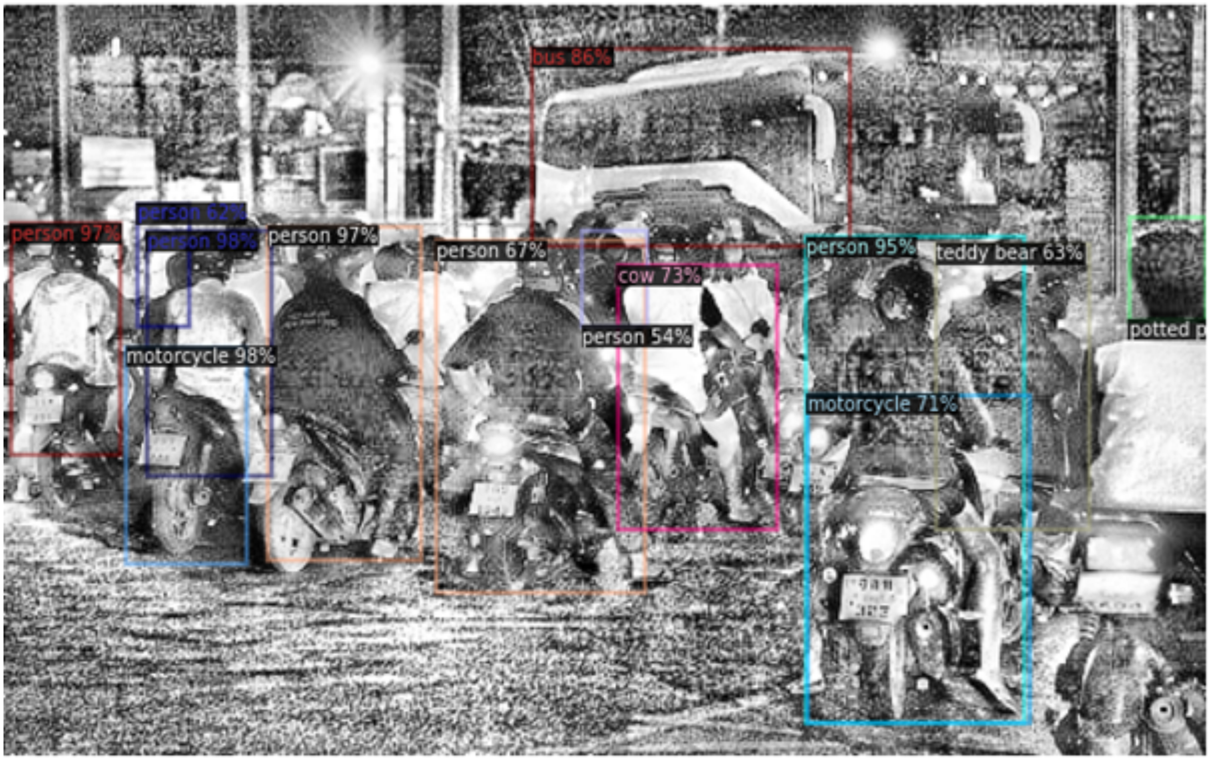
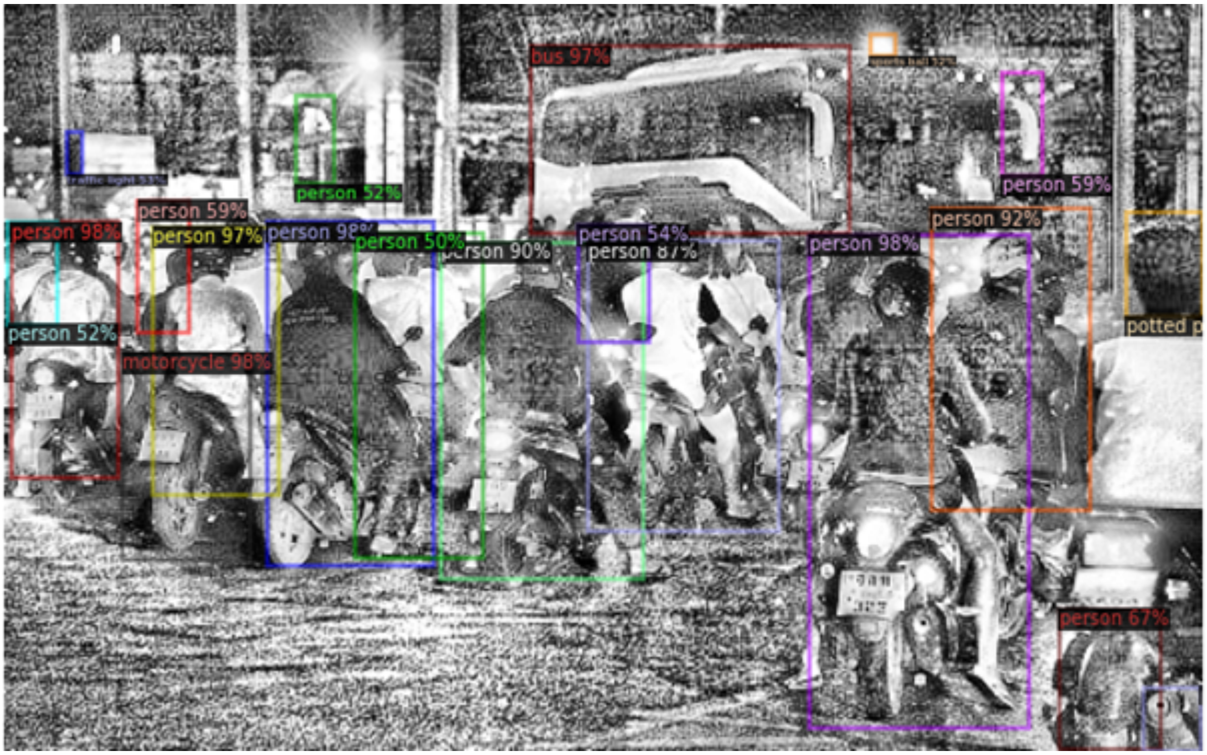


Figure 8: Finetuned Faster R-CNN



Model 5:

Figure 9: Faster R-CNN

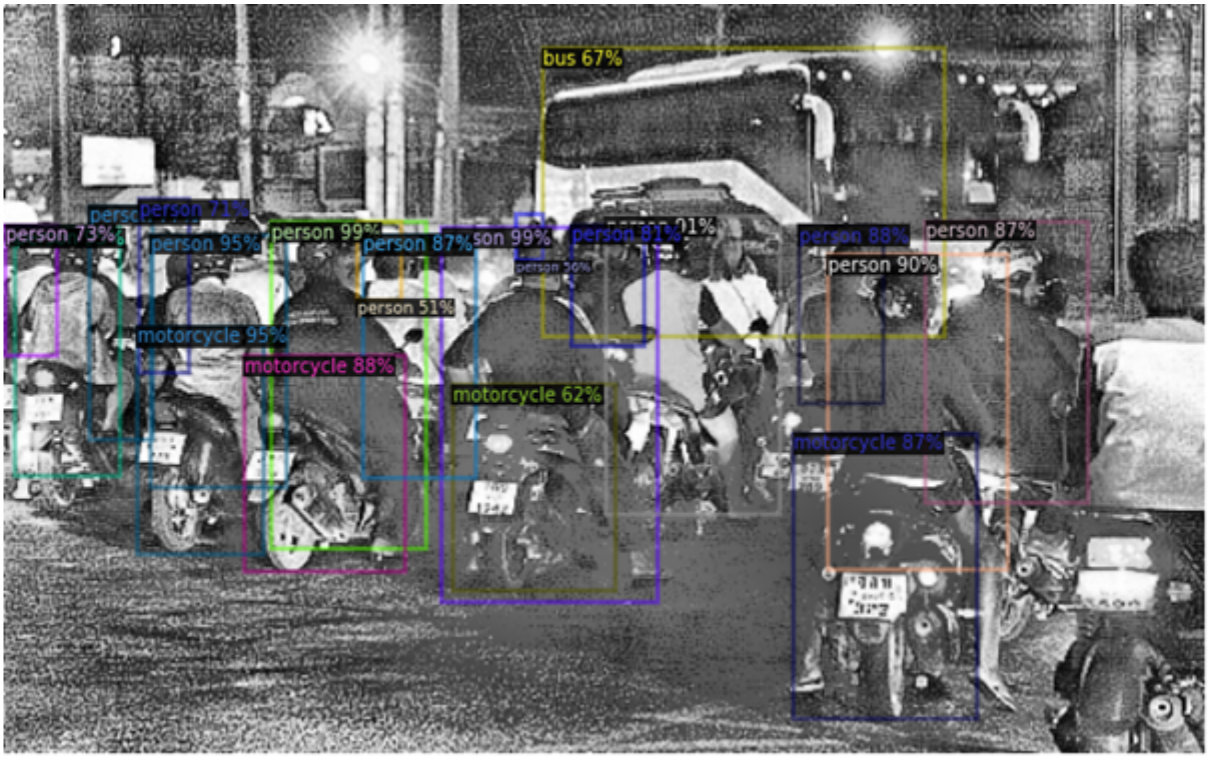


Figure 10: Finetuned Faster R-CNN

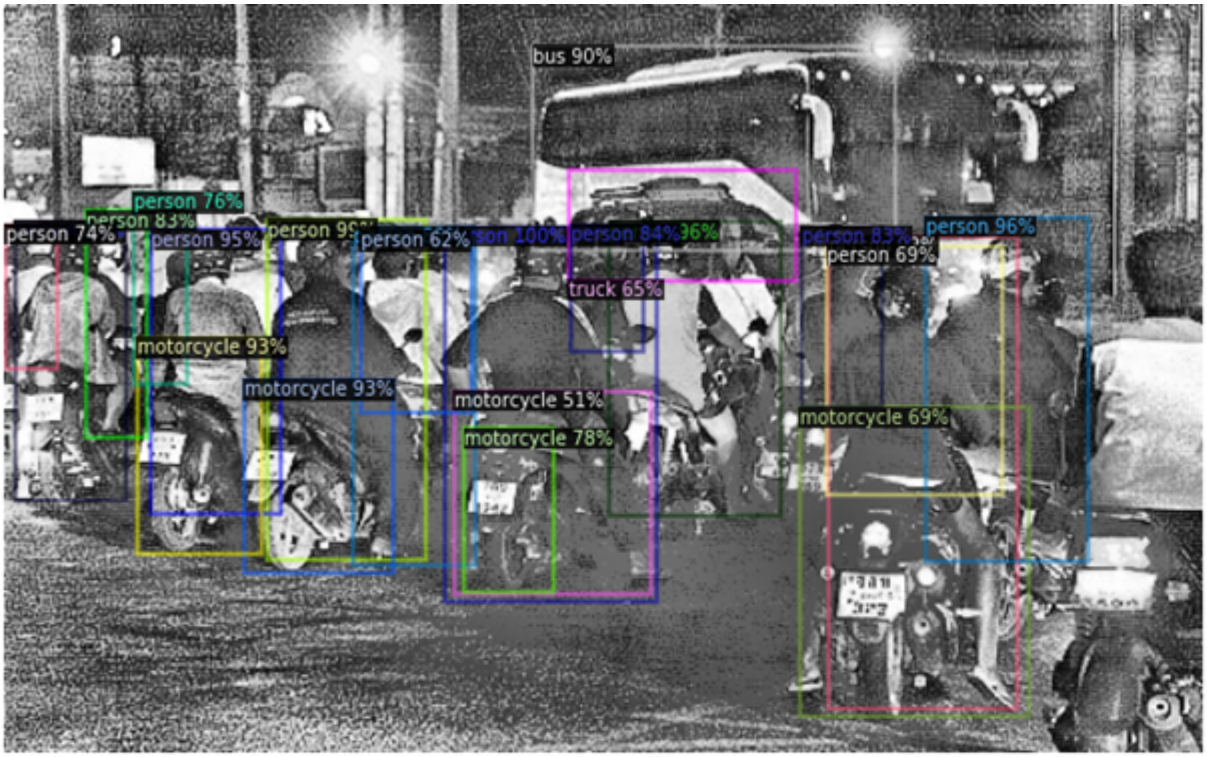
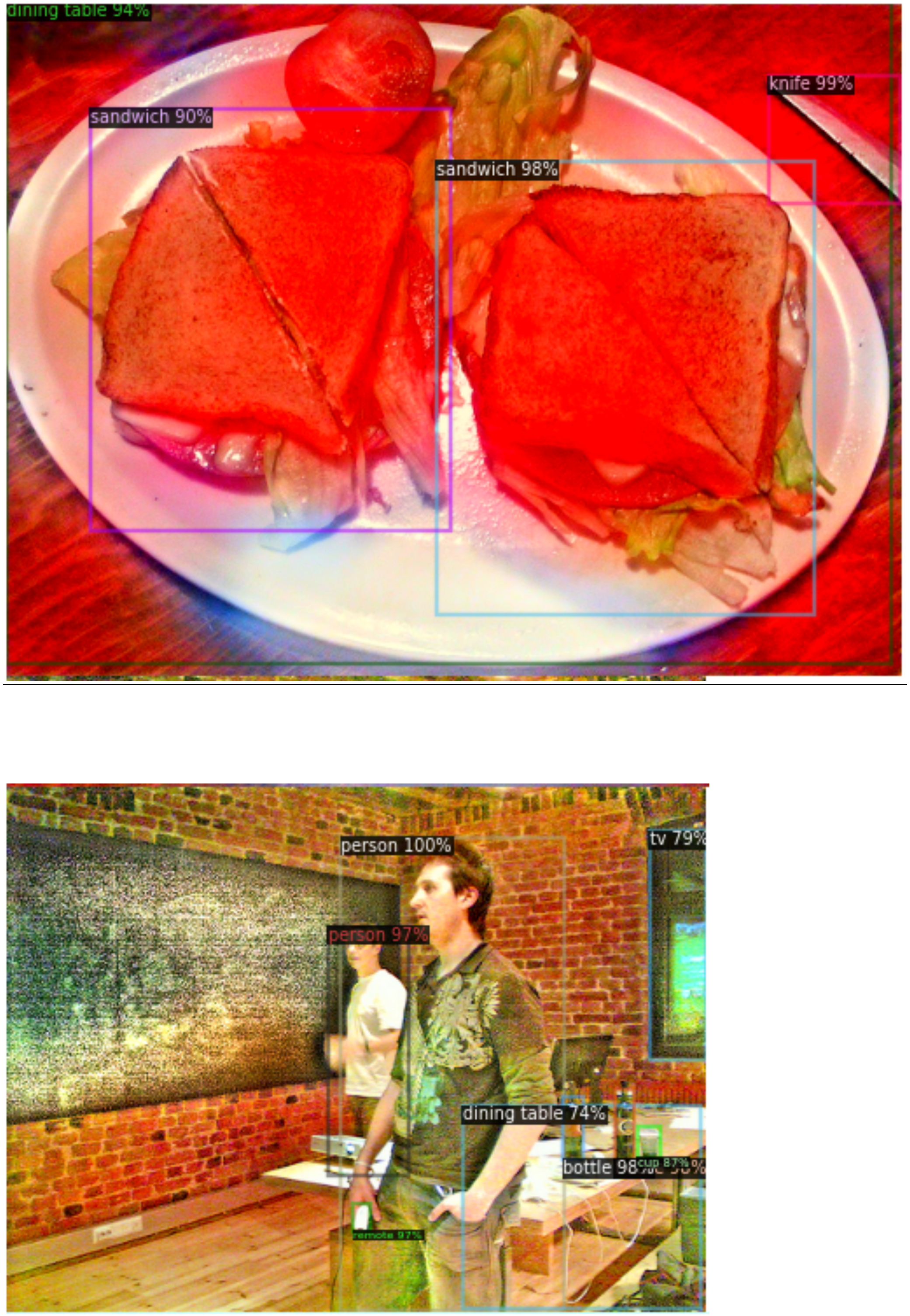


Figure 11: Sample images from Training Faster R-CNN using Transfer Learning on COCO dataset



WELCOME TO THE SHOW
person 99%
sheep 86%
car 93%
person 100%
person 74