National College of Ireland

Deep Learning-based Weapon Detection Using Live Cameras

MSc Research Project
Data Analytics

Abhinav Bhardwaj
Student ID: x20100906

School of Computing
National College of Ireland

Supervisor: Prof Aaloka Anant

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Abhinav Bhardwaj
Student ID:	x20100906
Programme:	Msc Data Analytics
Year:	2021-2022
Module:	MSc Research Project
Supervisor:	Prof Aaloka Anant
Submission Due Date:	31/01/2022
Project Title:	Deep Learning-based Weapon DetectionUsing Live Cameras
Word Count:	7776
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Abhinav Bhardwaj
Date:	31st January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Deep Learning-based Weapon Detection Using Live Cameras

Abhinav Bhardwaj
x20100906

Abstract

The world is surrounded by cameras, but none of us receive alerts when something goes wrong. In the age of automation, human monitoring is still necessary to keep track of things. Images can be processed in a novel way that focuses the detection algorithm on a specific point in time. This paper explains a new approach for detecting weapons and alerting security forces. The author revisited the topic of identification and created associated confusion classes to reduce the incidence of false positives and false negatives. Deep learning-based classification and recognition algorithms were used to test the new self-generated database. Database was built by using computer camera to capture weapons like 'knife' and other weapon-like objects 'vapes', 'keys', and 'pen'. This study uses a variety of algorithms, all based on deep learning and CNN architecture. Transfer learning from TensorFlow 2 Detection Model Zoo was used in the COCO dataset models that have been pre trained to reduce training time.

With this study's groundbreaking work, it is now feasible to identify weaponry in real-time video footage. To discover the best CNN object detector for real-time weapon detection in CCTV video streams. The author has built a new dataset of 411 photos using the OpenCV package captured in various environments such as: bright light, low light, blurred, sharp, reflective surface, dark and light background. In terms of speed and accuracy, On SSD MobileNet V2 FPNLite 640X640, the trained model predicted images in nearly all orientations, aspects, and viewpoints with an accuracy of 71.8 percent.

1 Introduction

1.1 Background

Photo and video recognition is a necessity in many real-world situations. Deep learning models can be used to build useful detectors for such scenarios, particularly when the target elements vary considerably in size, hue, structure, and substance. Because of their small size and similarity in shape and color, the process becomes more challenging when the target objects are small (represented by a reduced number of pixels).Dahlan et al. (2021)

The public usage of lethal firearms has increased in recent years, and this trend is projected to continue. Other countries, outside the US and later Europe, had seen active shooter events. The global rise in crime is partly due to the popularity of firearms and other small, concealable weapons. Law and order issues must be overcome for a

country to develop. We all need peace and security, whether we're trying to attract investment or tourists. Firearms are a crucial component in many countries' high crime rates. Most of the countries on this list allow gun ownership. We live in a global town, and our words and deeds directly affect others. This is especially true given how quickly social media, particularly Facebook and Twitter, disseminate erroneous or fabricated information around the world. Despair and rage are rising, and hate speech is inciting violence. Studies reveal that when confronted with a firearm, a person may lose their senses and become hostile.

The role of CCTV cameras in fixing this issue cannot be emphasized. CCTVs are now installed in practically every public place for safety, crime detection, and other security purposes. CCTV footage is crucial in court. Police officers arrive at a crime scene to acquire evidence, including video footage. Compared to other countries, the UK has around 4.8 million cameras. In 2010, Sweden installed 49598 cameras. With only 456 cameras in Poznan, the Polish authorities reduced drug charges by 58.67 percent and street fights by 41.3 percent. China now has 145 million cameras, and that number is likely to climb to 175 million by 2022. John Sudworth was apprehended by Chinese authorities in under six minutes using their massive CCTV camera network and facial recognition technology.Grega et al. (2016)

1.2 Importance

Using surveillance cameras to keep people safe was not only difficult and unreliable in the past. A human must constantly check the monitors and screens. Ten hours a day, seven days a week, a CCTV operator must monitor 20–25 screens. He must always be on the lookout for anything that could harm persons or property. The capacity to sustain continuous concentration on each screen decreases with the number of screens. It is difficult for the person responsible for monitoring the screens to remain focused at all times. Surveillance cameras that can automatically detect firearms and alert operators or security personnel are the answer. Instead, researchers are focusing on X-ray or millimeter wave imaging, as well as standard machine learning approaches, to detect concealed weapons (CWD). Convolutional neural networks (CNNs) have made significant advancements in object categorization and identification in recent years. It has achieved the best results to yet in terms of grouping, detection, and localisation. A feature is learned from data rather being chosen manually. Deep Convolutions Neural Networks are currently the most accurate detection methods. These models can learn distinguishing traits from vast amounts of labeled data. Mery et al. (2016)

This paper proposes an innovative autonomous detection and classification technique for weapons in real-time using cutting-edge deep learning models. To solve the challenge of detecting weapons in real-time for possible robbers/terrorists using deep learning, this study finished the implementation for knife as a single class for weapon and related confused items such as key, vape, and pen as distinct classes. With this information, we could determine whether someone was in danger or not.

Knives or revolvers were used in over 96 percent of the robberies observed on surveillance cameras, according to YouTube recordings. Through the use of an early warning system that alerts the operator and the appropriate authorities, this could help prevent many robberies and other scenarios like those that occurred in schools and clubs in the United States last year or anywhere else in the world.

1.3 Research Question

To investigate the efficiency and accuracy of the classification model to identify the weapons in the surveillance camera and minimize the false alarms.

1.4 Research Objective

For the purpose of answering the aforesaid research question, the following study objectives will be implemented:

- Using a self-created dataset with varying lighting conditions, backdrops, viewpoints, and multiple objects of the same class.
- To minimize false alarms, created a class that contains things that appear to be weapons.
- Stochastic Gradient descent with momentum is utilized in the fine-tuning process.
- Design and implementation of Deep Neural Network models in SSD MobileNet V2 FPNLite 320x320 and SSD MobileNet V2 FPNLite 640x640 based on feature-engineered weapon data utilizing Mobilenet network model as a baseline model.
- Analyzing and evaluating the accuracy and recall scores to arrive at a final conclusion.

The remainder of this paper is organized as follows: section II includes a related work section. The section after, section III talks about the Methodology, data generation, annotation, Modeling and Evaluation. In Section IV, the design specification explains about the how deep learning techniques are implemented. In Section V, the paper talks about the implementation process. Section VI discusses the evaluation and finally in section VII conclusion and future work.

2 Related Work

Modern CCTV systems, processing technologies, and deep learning models have made real-time object classification increasingly difficult. Most earlier research on this topic has focused on concealed weapon detection (CWD). It was previously used for luggage control and other extra security measures at airport terminals, and relied on imaging techniques like millimeter-wave and infrared scanning. Based on a suggested CWD technique for detecting concealed weapons at airports and other secure sites in the body.CWD is a method that combines infrared and color images in a multi-scale decomposition procedure. Incorporated optical and Infrared or millimeter wave images into a multi-resolution mosaic process to highlight the target's concealed armament.Dahlan et al. (2021)

A photo fusion-based CWD approach was proposed. For this, they used infrared imaging and visual fusion to find hidden weaponry above and below exposed regions. Their technique included using a homomorphic filter with various exposure settings to capture visible and infrared images. Extraction and detection are used extensively in modern systems, from simple intensity descriptors to more complex techniques like boosted cascade classifiers.Xu and Wu (2015)

CWD worked in certain cases, but it had limitations. These systems cannot identify non-metallic weapons since they rely on metal detection. Due to the high expense of X-ray scanners and conveyor belts, these machines were difficult to operate. Their use was limited by their high cost and health risks. Video-based weapon detection can also be used as a backup technique for acoustic gunshot detection. Gesick et al. (2009)

2.1 An Improved Object Detection Model for Embedded Systems Using Mobilenet-SSDv2

In this paper author states that "Most object detection approaches proposed in the literature are aimed at improving detection accuracy, and this is a problem". It is because of this that efforts to minimize computing complexity are often neglected. To achieve real-time speed, these new object detectors will require a high-end GPU. Mobilenet-v2-based object identification is presented in this publication by the researchers.Chiu et al. (2020) For embedded systems with limited processing power, a real-time object detector has been proposed. This is a major design consideration for modern self-driving systems (ADAS). Additionally, they employ a feature pyramid network (FPN) with the proposed object detection model in order to considerably improve detection accuracy and stability. Using a lightweight object detection model, experimental results show that the proposed method can obtain up to 74.9 percent mAP in the VOC dataset. In comparison to the current Mobilenet-SSD detector, the proposed detector has a detection accuracy of roughly 3.5 percent greater than the current detector. On the Nvidia Jetson AGX Xavier platform, the proposed detector obtains an average frame rate of 19 frames per second when running 720p video streams (FPS). A wide range of applications are possible because of the lightweight object detector's design. Chiu et al. (2020)

Mobilenet-v2 backbone network with enhanced feature extraction was suggested by the authors of this study. Combining Mobilenet-v2 and FPN models allowed them to expand the feature map of the image and hence the detection accuracy of their back-end network. On the Pascal VOC dataset, they suggested detection network had a mAP accuracy of 75.6 percent and a processing speed of 21 FPS. A total of 32MB of storage space is available in the network model, as well.Howard et al. (2017) A major advantage for embedded devices with limited resources is this. Mobilenet-SSDv2 detector results show that the original Mobilenet-SSD detector's fast processing benefit is retained, but the detection accuracy is greatly improved. Benefits like these show that a detection strategy proposed in this paper is more suited for embedded devices.Gopalakrishnan et al. (2017)

2.2 MobileNets: Mobile Vision Applications Using Efficient Convolutional Neural Networks

The author created models known as MobileNets for use in mobile and embedded vision applications. MobileNets is a reduced design that takes advantage of depth-wise separable convolutional convolutions in order to generate lightweight deep neural networks. Both of these fundamental global hyper-parameters are effective in balancing latency and accuracy. Consider the limits imposed by an application while deciding on the optimal amount of hyper-parameters to employ in their model construction. It was proved that their classification models outperformed popular ImageNet classification models after

substantial study on resource and accuracy tradeoffs was provided. By using MobileNets, the authors are able to put their theories to the test in a variety of scenarios and applications, including object identification, fine-grain classification, facial features, and large-scale geo-location. Wang et al. (2018)

Their experiments begin with an examination of the effects of depth-wise convolutions and the prospect of decreasing rather than increasing the number of layers. After that, a variety of common models were compared to demonstrate how the two hyper-parameters, resolution and width multiplier, were affected by reducing net work. Following that, the authors look into MobileNets in a variety of contexts. Xu et al. (2017)

According to the authors, a novel model architecture known as MobileNets, which is built on depth-wise separable convolutions, has been suggested. They investigated some of the most critical design decisions that contribute to the creation of an efficient model. Demonstrated how to reduce the size and latency of MobileNets without sacrificing acceptable accuracy or reliability. The scientists found that when they tested many MobileNets against usual models, they discovered that they had superior size, speed, and accuracy features. In doing so, they demonstrated how MobileNet may be used to a variety of uses. As a future step in the adoption and investigation of MobileNets, they planned to offer models in Tensor Flow as a next phase.Howard et al. (2017)

2.3 A Shape-Based Approach for Detecting Highly Visible Objects Using Deep Learning

Important visual features are detected while unimportant aspects of the image are filtered out. Using convolutional neural networks and form prediction, the researchers have developed a new method for identifying salient objects in photographs. A CNN model is used to understand the appearance of a prominent object in an image. Adding image-specific low-to-medium-level data refines the map further. According to results from an experiment, a new method of recognizing salient objects beats existing best practices.Kim and Pavlovic (2016)

Based on the results, this strategy outperforms all other methods tested. This study's qualitative analysis relies on saliency maps as examples. Lacking shape information, BIN (Binary representation) fails to correctly detect object borders. SSD values are generated solely across specified object portions, while saliency values are determined over the full picture. FCN (fully convolutional network) outperforms SSD (solid state drive) in terms of performance since it can focus on crucial bits while the rest of the data distracts it Less performant than the method provided here because it is not dependent on predefined shape classes. This results in coarse saliency maps but correct item silhouette recognition by SSD.Kim and Pavlovic (2016)

The author provides a revolutionary approach for detecting essential elements in photos using convolutional neural networks (CNNs). The salient object detection problem is a multi-label classification problem. The shape of a significant item is predicted by a CNN trained to anticipate object shape. Using hierarchical segmentation maps, refine the CNN's projected saliency map by adding global information such as spatial consistency and item bounds. The recommended technique beats current best practices in determining saliency across a wide range of benchmark datasets.

2.4 Adaptive Transfer from a Zoo of Models via Zoo-Tuning

Deep networks using enormous datasets have resulted in a vast number of diverse pre-trained models. Using typical single-model transfer learning methodologies does not fully use the zoo's vast amount of data, especially when transferring knowledge between models. To achieve the purpose, this work proposes Zoo-Tuning, a machine learning technique that adjusts the parameters of pre-trained models. Using Zoo-learn-able Tuning's channel alignment and adaptive aggregation layers, Authors improved the quantity of knowledge transferred by simultaneously changing several source models to downstream jobs. Adaptive aggregation reduces the overall cost of compute by two for training and inference. A collection of batch average gating settings was also suggested to save storage costs during inference procedures. Among the tasks tested by the author were incentive learning, photo classification, and facial landmark identification. An adaptive transfer learning strategy may be more effective and efficient than other approaches in conveying zoological information.Shu et al. (2021)

The same team of researchers conducts three tests. The first scenario used previously learned reinforcement learning models to transfer the models to a fresh set of Atari games. The other two examples used a zoo of computer vision models trained on a variety of large datasets. These models were used for face recognition and classification. The tests are run on PyTorch, a Python framework.

Transfer learning from model zoos is possible via Zoo-Tuning, say the authors of this study. An input gate network aligned with the channel alignment layer aggregates each model's parameters. This is because the target task modifies the source pre-trained parameters. To reduce storage costs even further, the authors proposed a temporal ensemble of batch average gating values. There is evidence that Zoo-performance Tuning can boost performance in areas such as reinforcement learning, picture classification, and facial landmark recognition.Gopalakrishnan et al. (2017)

2.5 Deep Convolutional Neural Networks with transfer learning for data-driven pavement distress detection using computer vision

A pre-trained deep learning model and weights are available within Keras Applications for this research. With weights pre-trained on the ImageNet database, the researchers used the Visual Geometry Group's (16-layer DCNN) VGG-16 model in Keras. The ImageNet database contains about 3.2 million well-labeled photos organized according to WorldNet's hierarchical system. When applied to image recognition and classification datasets from diverse domains, the pre-trained VGG-16 model performed well.

Sixteen convolutional layers (3 3), five max-pooling layers (2 2 for spatial pooling), three fully connected layers, and a soft-max layer comprise the VGG-16 DCNN model with 144 million parameters. Rectification Nonlinearity (ReLu) was activated on all hidden layers. Dropout regularization was also used in the model's fully connected layers. A VGG-16 trained on ImageNet. Using the pre-trained VGG-16 network's fully-connected classifier (or bottleneck layer) as a deep feature generator for the pavement photos. These semantic picture vectors were then trained and evaluated for label prediction using another classifier (such as Neural Networks [NN], Support Vector Machine [SVM], Random Forest [RF], and so on) (such as Neural Networks [NN], Support Vector Machine [SVM], Random Forest [RF], and so on).Kim and Pavlovic (2016)

On an Intel® CoreTM i7-5600U CPU, deep transfer learning was achieved. It uses an ImageNet-trained VGG-16 DCNN. Fine if the image is not exactly 48x48. A 224X224 shot yields 25,088 deep transfer learning features. They are 3072 x 1425 pixels. The ImageNet pre-trained VGG-16 DCNN produces a sparse feature matrix. So a 1000X500 input image preserved the pre-processed photos' original aspect ratio. The classifier employed ImageNet's 238,080 VGG-16 DCNN features.Ozaki et al. (2017)

Except for scikit-learn, all other classifiers were written in Python. Several deep transfer learning models detect pavement cracks. Best-performing models for each classifier category are highlighted. Cohen's Kappa and F1 values are used to quantify classification precision and accuracy [5, 6]. The strongest performers tend to be SVM and LR classifiers, followed by DCNN features trained on ImageNet. Lesser ROC curves and AUC values are more accurate.

3 Methodology

An open standard process model known as cross-industry data mining is a good analogy for this research technique (CRISP-DM). The figure below demonstrates the project architecture and the systematic approach to create the classification methodology, which involves the following steps: business knowledge, data acquisition, data prepossessing, modeling, evaluation, and anticipated output. The generated results could be put to good use in the classification of counterfeits.

3.1 Business Understanding

When it comes to keeping the public safe, the most current usage of object recognition technology is to identify the weapons used by criminals. Using artificial intelligence, we can create high-performance systems that can analyze imagery data faster and with better accuracy than ever before. A few AI-powered solutions have already proven to be more accurate at identifying dangerous weapons than experienced security professionals and systems. As part of the study, the author is proposing a better understand of safety and monitoring, which could lead to better procedures and a new level of safety. In many cases, it is impossible to adequately monitor security without the assistance of humans. Security officials are warned and alerted to the presence of a weapon by this method of identification. This reduces the time it takes to respond and saves both lives and property.

3.2 Data Acquisition

The self-generated dataset for the project was created by the author using objects from day-to-day life. He categorizes the objects into two categories: those that are hazardous and those that are non-hazardous. The concept behind creating two classes is that it can be further extended in the future by adding new photos. Moreover, these are sufficiently versatile from the standpoint of coding that anyone can add additional classes to either class. For hazardous classes, the author utilized kitchen knives, whereas for non-hazardous classes, he used pens, keys, and a vape. The hazardous class represents the types of weapons that we must identify, and it will serve as the primary object for identification during the course of the project. The non-hazardous class symbolizes the confusion class, which may appear to be a weapon or hazardous and may create a false

alert if it is mistaken for either. For this reason, in order to reduce false alarms in real time, we must increase the number of things that appear to be weapons, and we must continue to train our model in order to achieve more accuracy. The images were taken using two type of cameras. One is a web camera built inside the system, while the other is an external camera that is connected to the system. This assisted the researcher in creating a variety of settings and surfaces for all of the items, which will reduce the need for preprocessing in the future.

3.3 Image Pre-processing and Feature Extraction

Author first acquired a total of 20 photos, for each of the four objects in each of the both classes. When the data was introduced into the training phase, the findings were less than promising. Because of this, author decided to increase the amount of photographs to at least 100 each object, resulting in a total of 411 images for the two classes combined. The majority of them have excellent picture quality and large pixels that allow items to be clearly seen. Some of them are on a table with a white or black background and varied lighting conditions, such as low light and bright light, while others are on a hand-held. Some are intentionally blurred, while others exhibit objects that are not in the frame of the image. There are a few photos that have two items in a single frame.

The next thing the author did was label the objects the project needed to detect by using a graphical image annotation tool created by Darren Tzutalin from his public GitHub repository named LabelImg. Using LabelImg, users can label the image's object boundary boxes graphically. It is developed in Python and has a graphical user interface that is implemented using Qt. In PASCAL VOC format, ImageNet saves annotated images as XML files. Please refer the figure 1. Figure has been self made using following link in foot note. [1] We will create a new directory inside our working folder and clone the GitHub repository. Labeling the photos is a time-consuming and tedious operation that must be completed manually for each and every image in order to detect the objects in them. Then, they were subsequently subdivided into 75-25 training and testing images, respectively.

3.4 Modelling

MobileNetV2 is the latest iteration of lightweight neural network models, following in the footsteps of its predecessor MobileNetV1. MobileNetV2 further decreases the number of operations performed and the amount of memory consumed with maintaining accuracy. The contribution of the inverted residual with linear bottleneck is the important concept in this development. Kindly refer the figure 2 for the complete architecture. Kindly refer the footnote for the source of figure. [2]

- Bottleneck in a straight line: Non-linearity inside a deep neural network causes information loss, and MobileNetV2 makes use of linear bottleneck layers to keep that information safe. It is based on the assumption that a layer's information can be found in a low-dimensional subspace of the input space. Because of the non-linearity of the RELU transformation, significant information is typically lost, but

[1] draw.io

[2] https://medium.com/@techmayank2000/object-detection-using-ssd-mobilenetv2-using-tensorflow-api-can-detect-any-single-class-from-31a31bbd0691

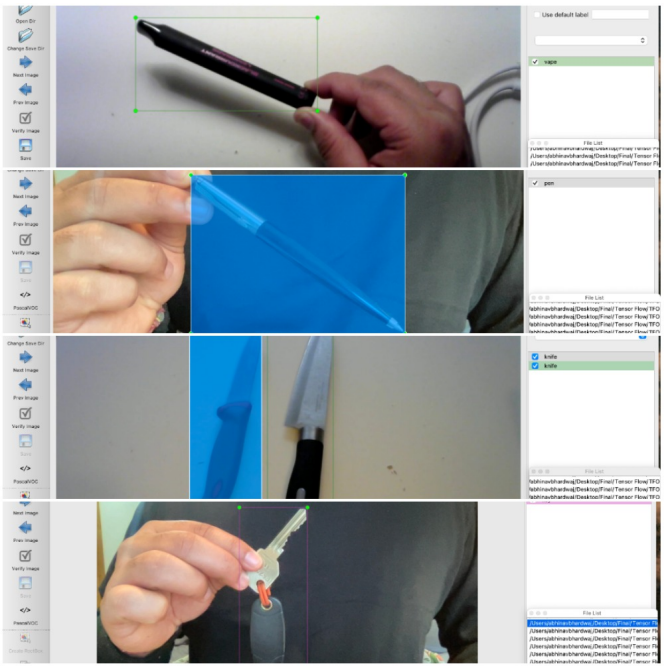

Figure 1: Labelling of Objects

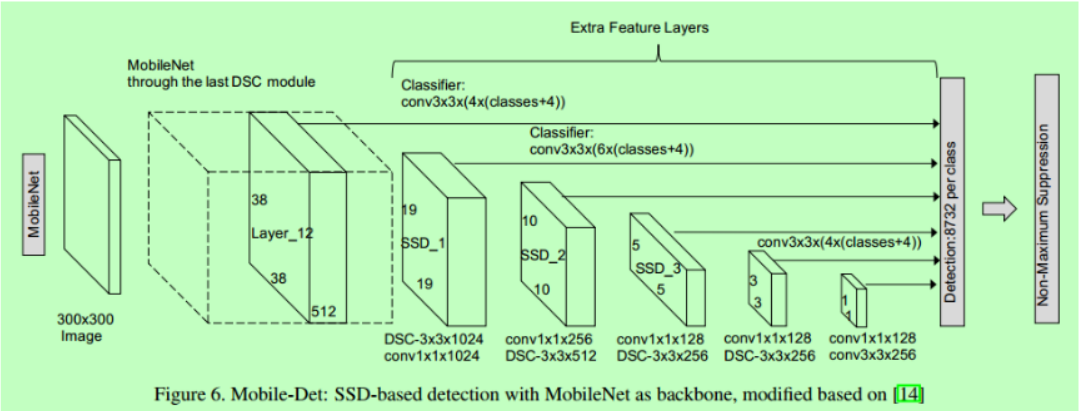

Figure 6. Mobile-Det: SSD-based detection with MobileNet as backbone, modified based on [14]

Figure 2: Architecture of SSD MobileNetV2 FPN Lite

if the information in the input is contained inside a low-level subspace, it is maintained. By incorporating linear bottleneck layers within the depthwise separable convolutional layers, significant information is preserved rather than being lost. It is referred to as bottleneck convolution, and it is employed throughout the entire MobileNetV2 model. Giron et al. (2020)

- Inverted Residues (or Inverted Residues): A normal residual block joins two layers in the same way. The intuition that data is kept through the bottleneck layers suggests that the connections can be routed through them instead. The term for this is invertedresiduals. This approach allows for improved memory utilization and, in general, somewhat greater performance.Giron et al. (2020)

SSD is a single neural network that learns to anticipate and categorize bounding box locations as they are drawn around the image. As a result, SSD can be taught from the start of a program. The SSD network uses the MobileNet concept as a basis, with convolution layers added afterwards. Kindly refer the figure 3 for to understand the architecture of MobileNet. Instead of employing RPN-based systems like the R-CNN series, we can use SSD to recognize several objects in a single picture, minimizing the number of shots necessary. Two-shot RPN techniques are slower than SSD approaches. Kindly refer the footnote for the source of figure.[3]

SSD uses a non-maximum suppression phase after a feed-forward convolutional network generates a fixed-size collection of bounding boxes and scores for the existence of object class instances. The basic network architecture utilized for high-quality picture categorization is decreased before any classification layers are introduced to the early layers. The network gains a new structure that generates detections with the vital features listed below. Akkas et al. (2019)

Convolutional feature layers are added to the truncated base network to build multi-scale feature maps for detection. Multiple detection predictions are possible with decreasing layer sizes. Each feature layer has its own convolutional model for detection prediction.

Detection convolutional predictors Convolutional filters can be used to generate detection predictions for each additional feature layer (or optionally an existing feature layer from the base network). The SSD network design is displayed on top. The fundamental element is a 3x3xp kernel that generates either a score for a category or an offset from the default box coordinates relative to the feature layer's mxn channels. Each mxn position where the kernel is used generates an output value.Eriksson (2021)

When an image is fed into a feature pyramid network (FPN), it produces convolutional feature maps of varied sizes for each layer. Backbone convolutional designs are unrelated. As a general method, it can be utilized to generate feature pyramids in deep convolutional networks for object detection. The pyramid is built using both a bottom-up and a top-down approach. Kindly refer the figure 3 for the architecture. Kindly refer the footnote for the source of figure. [4] The bottom-up pathway results in a feature hierarchy composed of feature maps of varying scales. The feature pyramid comprises levels for each stage. Each stage's final layer serves as a reference set of feature maps. ResNets utilise the feature activations generated by each step's final residual block.

[3]Architecture:`https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classifica`

[4]https://towardsdatascience.com/review-fpn-feature-pyramid-network-object-detection-262fc7482610

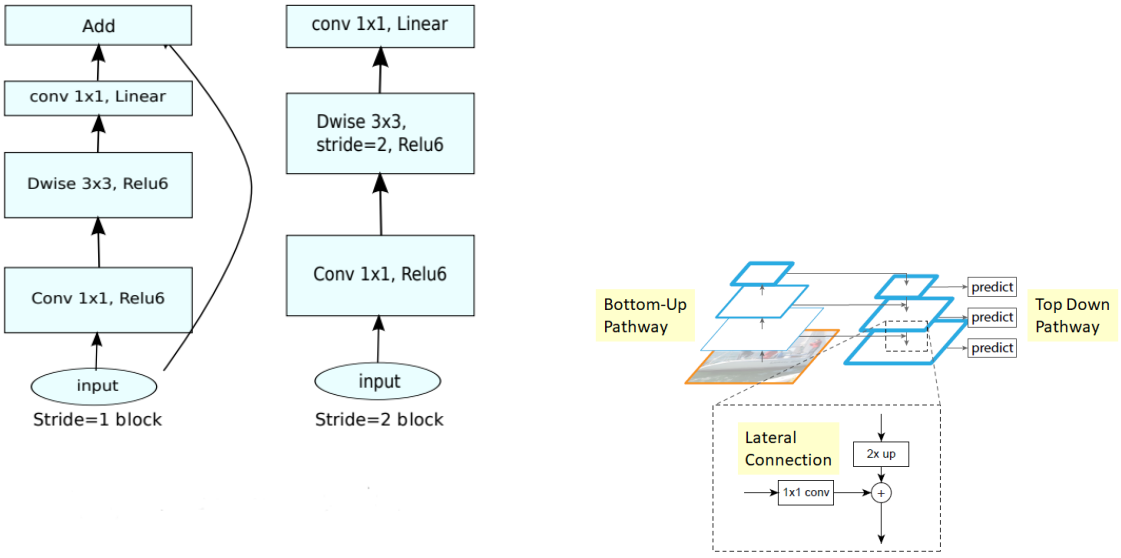

Figure 3: Architecture of MobileNet and FPN

As the name implies, the top-down approach creates hallucinations of higher resolution features by up sampling lower pyramid feature mappings. These attributes are bolstered by lateral connections from the bottom-up pathway. Each lateral link can blend feature maps from the bottom-up and top-down. Because the bottom-up feature map was sub sampled less, its activations are more precisely localized.

3.5 Evaluation and Inference

Precision is the degree to measure if the model's forecasts are on target. i.e. how many of model's predictions are correct. Recall measures how well the model remember the true positives. IoU is used to estimate the distance between two borders. Useful for estimating border overlap between expected and actual areas (the real object boundary). In our dataset, we define an IoU threshold (say, 0.5) for evaluating if the forecast is correct or incorrect.

We only average recall over IoU thresholds between [0.5, 1] because detection performance corresponds to recall at thresholds above 0.5, where at 0.5 boxes loosely localize the objects and at 1 the items are perfectly localized, and because detection performance corresponds to recall at thresholds above 0.5.The area doubled under the Recall x IoU curve is described by average recall. The Recall x IoU gradient depicts recall outcomes one per IoU threshold wherein IoU [0.5,1.0], with IoU thresholds shown on the x-axis and recall plotted on the y-axis. Interpreting the results:

- Even though if all of the test dataset objects have been detected with a high recall rate and low precision, the vast majority of the detections are incorrect which means there are many false positives.
- Because of the poor recall but high accuracy, all predicted boxes are valid; nevertheless, the bulk of test dataset items have been ignored as a result of the low recall but high precision which means there are many false negatives.

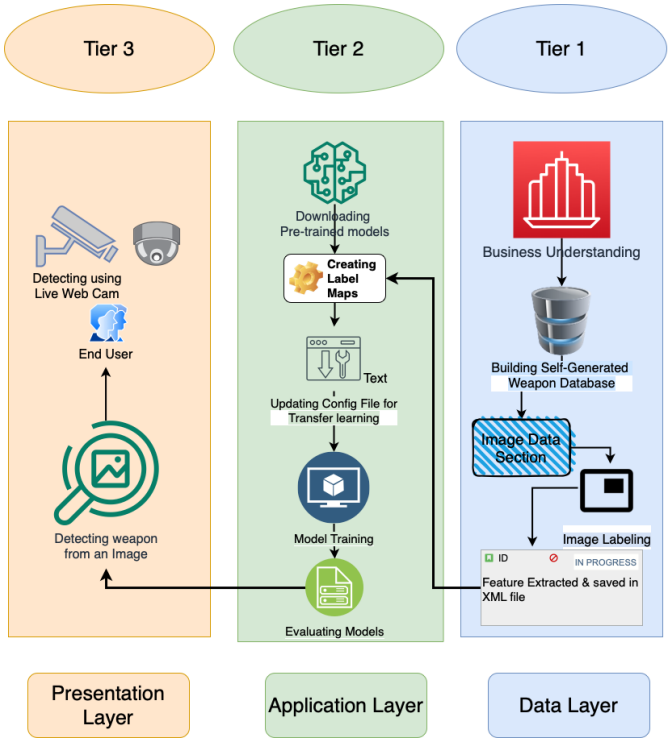

Figure 4: Project Architecture

- The ideal detector detects the vast majority of test datasets items with high precision and recall because of its high precision and recall.

Next, the author considered the classification/localization Loss values as these are the output of loss functions, and they indicate the "price paid for prediction inaccuracy" in classification and localization tasks (respectively). The loss value that has been provided is the sum of the classification and localization losses. The optimization algorithms make an attempt to reduce these loss values until the loss sum reaches a point at which we are satisfied with the results and consider the model to have trained. Loss can be regarded of as a score, with a lower value suggesting a stronger model in general.

4 Design Specification

From the figure 4 Project Architecture it is presented that there are three tiers to the design workflow: the Data Layer, which prepares the information for the model; the second tier, which is known as the Application Layer, which is where model implementation, training, model evaluation, and validation will take place. The final tier, which is known as the presentation layer, is where desired results such as graphs, interpretation/visualization, and live video check will take place.

- In the data preparation stage (Data Layer), using the understating of the business, all methods used for data collection, extraction, feature selection and transformation, as well as augmentation and enrichment of the dataset, are included in this stage. The OpenCV library was used to create the data source for this project. After that, each object in the image is given a graphical annotation with the help

of a tool developed by Darren Tzutalin which is published on his public GitHub repository under the name LabelImg. Each file is then saved in an XML file, together with its associated data-annotation, before being divided into two parts: the 75–25 training component and the testing section.

- The modeling stage (Application Layer) consists of the development of various convolutional neural networks, such as the SSD MobileNet V2 FPNLite 320x320 and the SSD MobileNet V2 FPNLite 640x640. Creation of label maps, TF records and validating the tesnorflow script will pass this stage. It is then further been assessed in terms of average precision, average recall, loss/localization loss, and learning rate. Parameter fine-tuning and model refinement are also carried out during this stage.

- The third tier is the Presentation Layer, also known as the interpretation stage, in which the output is intended to be displayed in TensorBoard. It is an open sourced toolkit which helps viewers to analyze training progress and improve performance of the models by adjusting the hyperparameters. The TensorBoard toolbox presents a dashboard on which logs can be displayed as graphs, pictures, histograms, embeddings, text, and so on. It also aids in the tracking of information such as gradients, losses, metrics, and intermediate outputs. This is the stage at which one can examine an image to determine whether it contains a weapon or not. For a more interactive version, the author has designed a platform where users may use a live feed from a web cam and the output will directly determine whether the user is using a weapon or not.

Each model that has been implemented has undergone modification, cross-validation, and analysis using the proper metrics, which are addressed in greater detail in the following sections.

5 Implementation

5.1 Workflow

In the course of putting the research project into practice, data sampling, cross validation, and fine-tuning parameters using optimization and regularization techniques are all included. This is primarily due to the fact that processing times are much shorter. The generalization error is the most significant problem since it limits the accuracy of a predicting model when it comes to forecasting data that has never been earlier. Last but not least, when it comes to exploratory analysis processes, it is safe to assume that this method will be used as soon as the data is collected, and trained models will be loaded within which interpreters will create realizations, validating and testing their model adjustments as they go along on the way, rather than after the data has been collected. In the first step, the image is input into a convolutional layer, where the kernels of convolutional kernels are used to recognize elements and build feature maps from the images. Following that, the image is fed into a pooling layer, where the kernels of the pooling kernels detect traits and generate feature maps based on them.Following the convolution of the feature maps, the spatial extent of the feature maps is reduced by a subsequent pooling layer. A single series contains a number of repetitions of this pattern. The performance of the pooling system is then flattened and passed on to a number of other fully

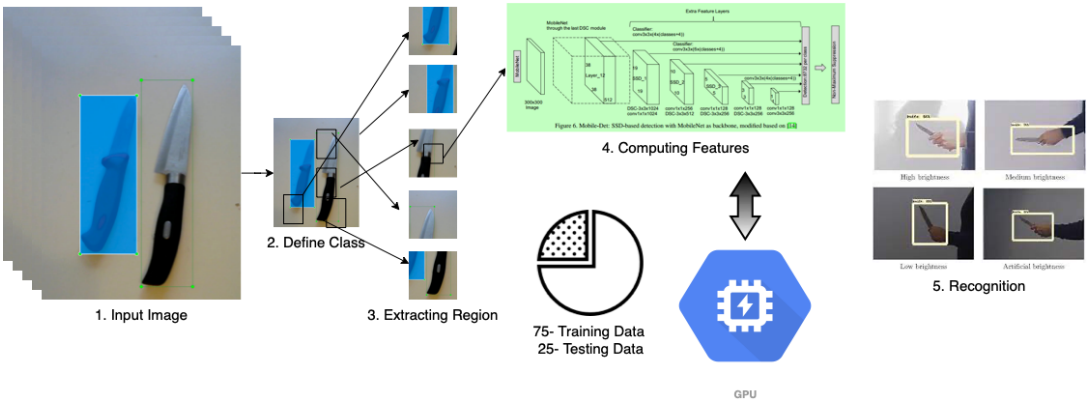

Figure 5: Workflow

connected layers, which result in scores for different classes. The last fully connected layer frequently makes use of a softmax activation to successfully normalize the obtained scores to the scale [0, 1, 2,... n], which provides the prediction class probabilities of the input data for the final fully connected layer. RTX 3070-8GB GPU of a Windows machine (Asus Zephyrus G15 (GA503Qr413)) is used for the model training, which are carried out in TensorFlow and Keras. Please see the figure 5. figure has been self-generated using link kindly refer it in footnote. [5]

5.2 Training and Testing

For each algorithm that was tested during the training phase, a range of settings were tried during the phase. The author made the decision to start with the first 100 photographs in the collection. It was discovered that the outcomes were unsatisfactory. These images are cluttered with distractions, and the main subject, is covered by the clutter in the background, and the person's face, making it impossible for the model to produce a good result in these pictures. The author started by training it for 2000 steps, then increased the number of steps to 4000. In spite of this, the researcher decided to alter the photographs and their background because the average precision and average recall rate for the objects had remained unchanged. When creating the new series of images, the author chose to take 100 photographs of each object, which resulted in a total of 411 photographs taken with the two cameras combined. We moved about 300 images to the training section of the training for both models in order to ensure that the photographs were taken against a variety of backgrounds and lighting scenarios. With the new images the fact that we completed 15,000 steps of training, our accuracy rate was great. In order to progressively raise the speed of our training, we started with 2000 steps and worked our way up to 4000 steps, then 8,000 steps, and finally 15,000 steps. When the number of steps was increased to 20,000, the accuracy score decreased. We used early stopping for both models, however it only stopped after approximately 15,000 steps, indicating that the optimization technique was providing the same results for both models.

Tests were conducted on a quarter of the whole data set, which amounted to approximately 100 pictures that were not included in the training of any of the networks that were constructed. Detailed discussions of model designs and configuration options are

[5]https://draw.io/

provided in the following sections.

5.3 SSD MobileNetV2 FPN Lite

In this study, the two models that are used is from SSD MobileNetV2 FPN Lite version. This has two different blocks available. One of the blocks is a residual block with a one-step stride In addition, there is a second block for downsizing. There are now three layers of blocks for each type. An initial convolution with ReLU6 is used in this case. The second layer of convolution is depth-wise. No non-linearity in the third layer of convolutions. ReLU is utilized again, deep networks will only be as powerful as linear classifiers in non-zero volume areas. There is also an expansions factor t.

- When it comes to major experiments, t = 6 is used. An input with 64 channels would result in an output with 64xt = 64x6 = 384 channels on the internal side.
- According to this study's design, the input image supplied is 320x320 with a learning rate of 0.0, a weight decay rate of 0.01.
- To overcome the oscillations and flat spots of a noisy gradient and cruise across flat areas of the search space, momentum is an extension of gradient descent optimization.
- In order to enhance accuracy, it is necessary to remove ReLU6 from the output of each bottleneck module.
- It outperforms the shortcut between expansions and the one without any residual connections when the bottlenecks are located near one another.

The two models of SSD MobileNet that are used in this study as follows : SSD MobileNetV2 FPN Lite 320x320 and SSD MobileNetV2 FPN Lite 640x640. The distinction between them is in how each model processes the input image. The first reads in 320x320 pixels, while the second reads in 640x640 pixels. The more data that is processed by the natural law of machine learning, the better the outcome. As a result, when we run the data through the 640x640 model, we receive better results. As the model learns to recognize pixels more accurately, accuracy improves. However, there is a trade-off between training time and data input, with the more data fed resulting in a longer training time.

6 Evaluation

The author will compare two experiments utilizing the loss and classification loss matrices, learning rate, average time to train the model, average precision, and average recall. To showcase how accurate the model's forecast is to the observed values And the recall evaluation is how many genuine positives were recalled (found). In order to showcase the differences between MobileNet's two SSD models in tests 6.1 and 6.2, we ran the following tests: MobileNet models with specific hyperparameter tuning and regularization strategies are studied. Using the visualization tool Tensorboard.

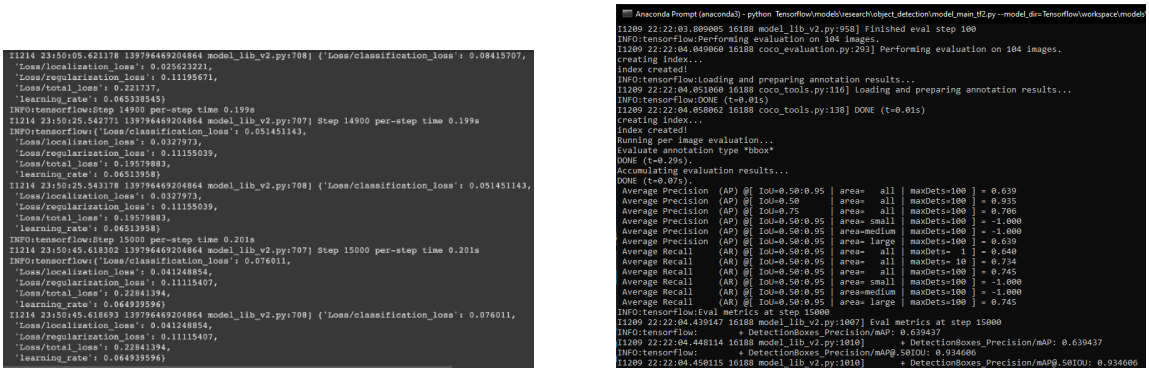

Figure 6: 320x320 model steps@15k and eval matrix

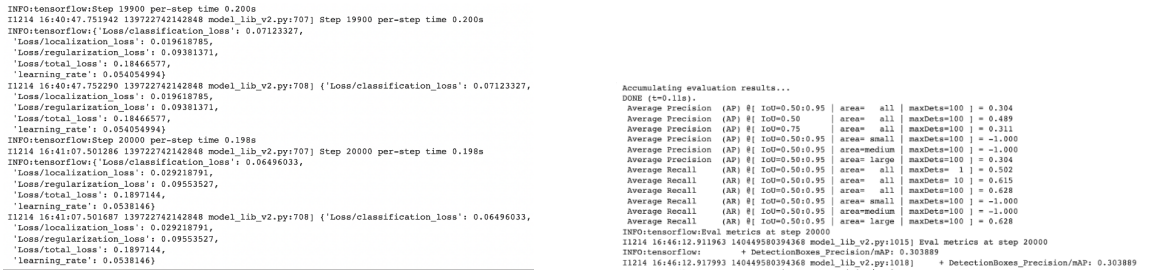

Figure 7: 320x320 model steps@20k and eval matrix

6.1 Experiment 1 : SSD MobileNet V2 FPNLite 320x320

The dataset has been used to evaluate Tensorflow Zoo's MobileNet V2's transfer learning approach. SSD MobileNet V2 FPNLite 320x320 variant was used in the first experiment. The comparison of the test results can be seen below:

Steps	Loss	AvgP	AvgR
5,000	0.107	0.573	0.582
10,000	0.098	0.601	0.623
15,000	0.076	0.639	0.640
20,000	0.064	0.304	0.502

Table 1: Table of experiments run for 320X320

From the table 1, it can be seen that the author, performed test for a total of 20,000 steps and then evaluated the results at every 5000 steps. After first 5000 steps, the loss was greater, coming in at 0.107. The average accuracy (AvgP) and recall (AvgR) values were 0.573 and 0.582, respectively, on the accuracy and recall values. While increasing the number of steps till the author reach 20,000, further he notices a decrease in the loss, but also notice a fall in the average precision and recall values as the number of steps increases. As per the author this situation arises as a result of the weights being wrongly altered during the forward and backward propagation phases of the simulation. Because the expected value falls short of the actual target, the prediction is incorrect. Consequently, the model's best precision and recall values are achieved when the model has completed 15,000 steps. As a result, the model's best precision and recall value is at 15,000 steps. AvgP is 0.639, and AvgR is 0.640. Kindly refer the figure 7.

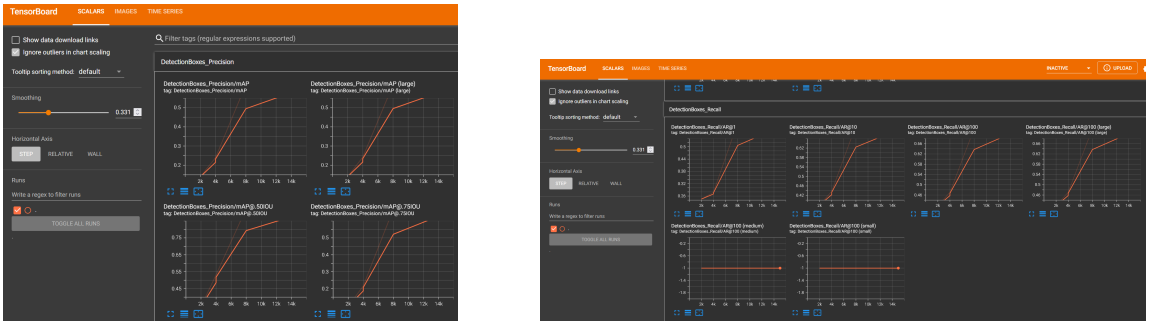

Figure 8: Graphs of AR and AP of 320X320 steps@10k model in Tensorboard

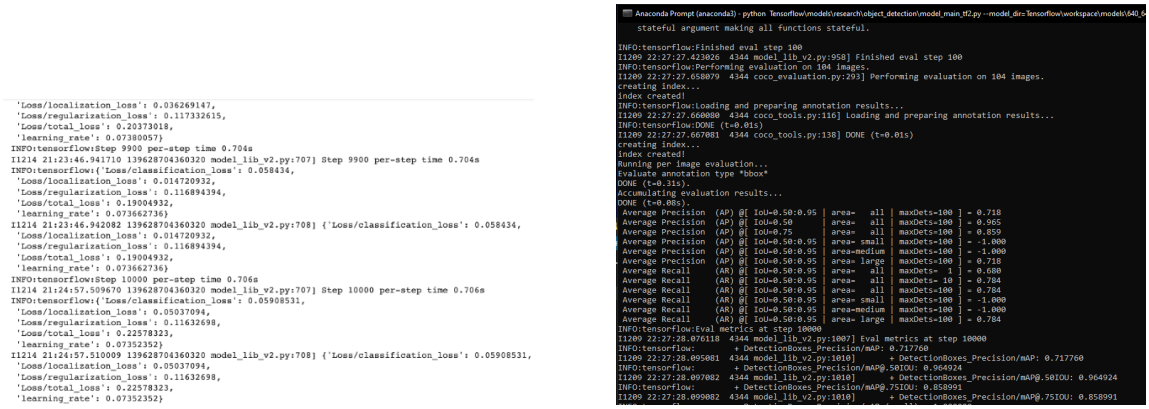

Figure 9: 640x640 model steps@10k and eval matrix

Tensorboard is a tensorflow visualization tool that displays the average precision and average recall metrics for the 320X320 model. This has reached 10,000 steps and is continuing rising, indicating that we need to train more, like we did originally. As a result, after 15,000 steps, we obtained the right accuracy and recall scores. Kindly refer the figure 8.

6.2 Experiment 2 : SSD MobileNet V2 FPNLite 640x640

Experiments have been carried out to test the transfer learning approach of Tensorflow Zoo MobileNet V2. A 640x640 version of the SSD MobileNet V2 FPNLite was utilized in the second experiment. The results of the tests can be found here in the table 2:

Steps	Loss	AvgP	AvgR
5,000	0.098	0.496	0.655
10,000	0.059	0.718	0.680
15,000	0.100	0.554	0.561

Table 2: Table of experiments run for 640X640

In the second experiment, the author used the same methods as in the first, but decreased the number of steps from 20,000 to 15,000. After first 5000 steps, the loss was 0.098, which is only marginally lower than the loss of 0.107 experienced by the first model. The average precision value (AvgP) was 0.496, and the average recall value (AvgR) was 0.655. When compared to the first model, the second model was more complex. As

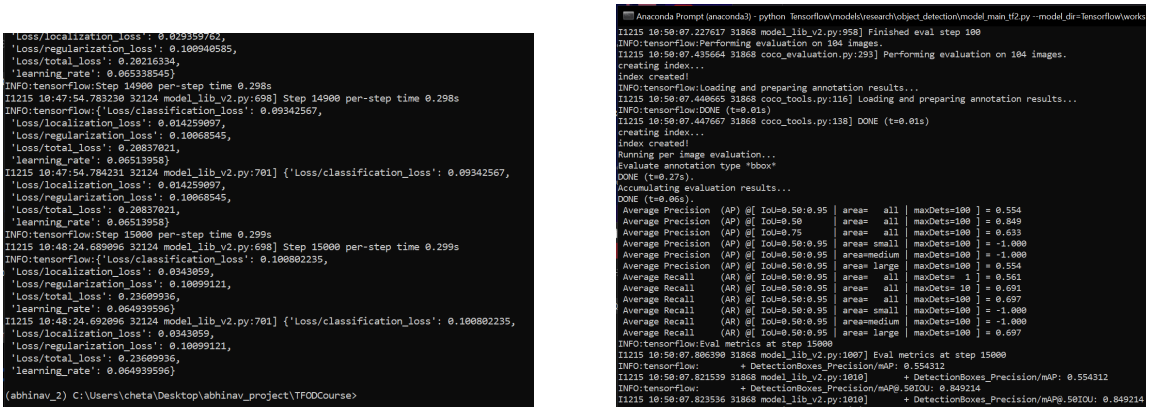

Figure 10: 640x640 model steps@15k and eval matrix

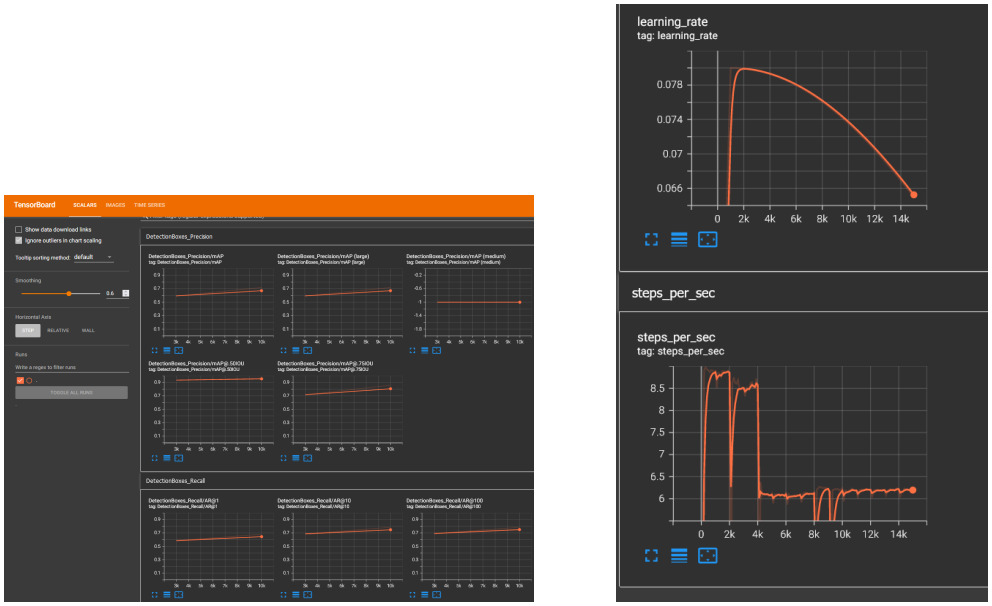

Figure 11: Graphs of AR, AP, descending learning rate and steps@15k of 640X640 model

author increased the number of steps until he reached 15,000, the author saw an increase in loss and a decrease in average precision and recall value, which fell to 0.554 and 0.561, respectively. According to the author, he investigated the same underlying cause in this case as well, which is that the weights are incorrectly updated throughout both forward and backward propagation. One can refer the figure 9and 10 for the same.

Furthermore, the forecast value indicates that it will fall short of the actual target. However, in the second experiment, he finds a faster learning graph as well as a rapid fall in the loss function. As a result, the model's best precision and recall values are found at 10,000 steps. This is 0.718 for AvgP and 0.680 for AvgR.

According to the Tensorboard graphs in figure 11, for the experiment of 640 x 640, for steps 15000, the learning graph is taking a nose dive, indicating that the study does not require any additional training because the model itself has begun to drop the learning rate.

The figure 12 graph shows the difference in learning and loss rates for 10,000 and 15,000 steps. The learning graph in the 10,000 has already begun to bend toward the negative side, but in the second graph, the learning rate is already taking a dip down.

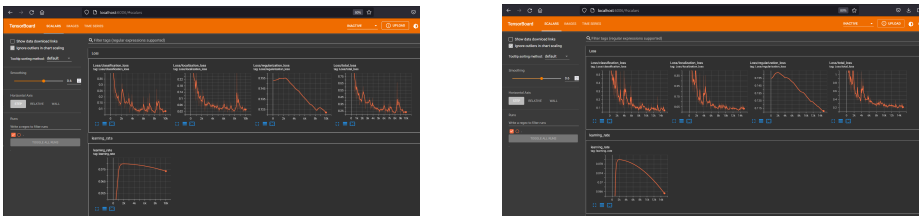

Figure 12: Graphs of Losses vs steps@10k and 15K of 640X640 model in Tensorboard

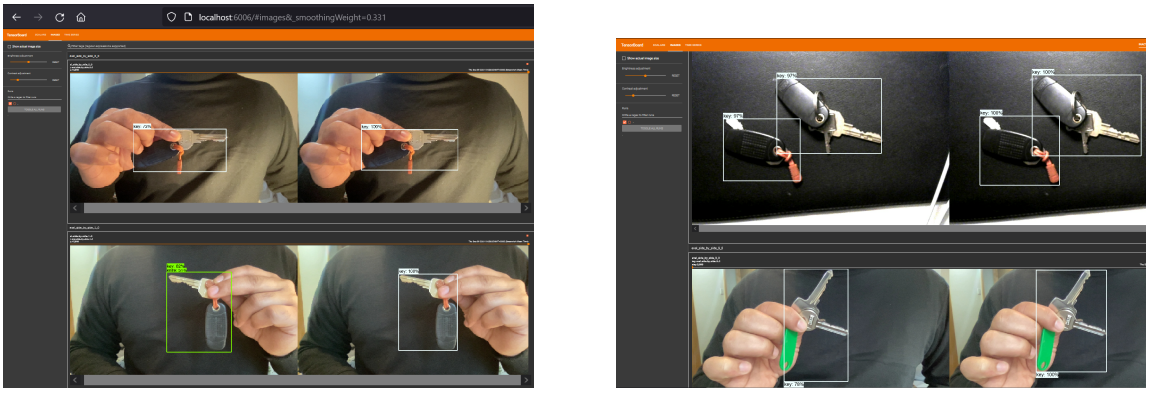

Figure 13: Visual Comparison of Testing VS the Ground Truth

This is the beauty of Tensorflow; one can see all of the phases of the comparison testing data in relation to the ground truth data. On the left side of the tensorboard, there is a tab where you may modify the brightness and number of steps to run through. Kindly refer the figure 13.

6.3 Discussion

- Interception of Image : For SSD MobileNetV2 FPNLite inference outcomes, it is necessary to consider the predictions and dataset used in making these conclusions. An in-depth discussion of the incorrect predictions and threshold value is required in the context of the image predictions. Many incorrect predictions would have been eliminated if the threshold value had been set at 90 percent rather than 70 percent. But on the other hand, some of the most accurate forecasts would have been thrown out. For the threshold value, it depends on whether the desired outcome is high confidence forecasts or if less certain predictions should be allowed. The ambiguity and contradiction between imagery can be used to analyze the images' underlying truths. Simply said, object detection is all about determining the presence of things in a picture using only the information provided by the image. For an instance, as long as the parasite isn't visible, the ground truth won't be used to find parasite-infested bees. In this situation, the ML model will get data that suggests that infected bees appear just like non-infected bees. Some ground truths are marked twice for a single bee with a parasite, where one ground truth is contained within the other. Another example of the ML model's discrepancy in what it should learn to find, in this case it is ambiguous if the right detection is to mark the entire body of a bee with parasite or a portion of the body. Finally, differences in the distance between the camera and the bees need to be addressed. Images of bees that are too far away can result in a loss of crucial information and a decrease in the quality

of the data collected. A lack of consistency in the bee information could have an influence on the trained model's performance.

- Interception of Video : There is currently no way to measure the accuracy of video inference because there is no ground truth data available for video files. The inference time, which must be rapid enough, is one topic for discussion. Constraints imposed by drawing each bounding box in total synchronous with video frames mean detection must be fewer than a camera's frame rate. However, it may not be necessary for the detection problem to have a seamless experience. There would be less need for perfect frame-to-frame synchronization if more frames were virtually identical when hovering the camera over an item. If the detection aim can be met at a specific frame rate without generating redundant frames, then that frame rate is the target. Detection on video would require an inference time of at least a few minutes at the very least. The SSD MobileNetV2 FPNLite's hardware is a major impediment for this operation. For high-end devices, users may expect a much smoother and more dependable experience. A wide range of analysis problems can be solved with video inference, including tracking individuals, motion analysis, and more.

7 Conclusion and Future Work

A weapon detection object detection model is described in this work. It was designed as an alternative to CCTV-based systems and searches for firearms automatically, without the need for human input. SSD MobileNetV2 FPNLite was selected as the model based on the SSD architecture with MobileNetV2 and FPNLite optimizations.‘ To achieve a specific detection purpose, the model has been set up to mimic the appearance of weapons and weapons to other items. Despite its small size and short inference time, it has a surprisingly high mAP. SSD MobileNetV2 FPN Lite 640x640 has shown the better output than its another version of SSD MobileNetV2 FPN Lite 320x320 in terms of average precision with 0.718 and average recall of 0.68. It is implemented in Python-based programming, using still picture inference with experimental webcam streaming inference. In many files, this work's program and its aspects are described. As a result of the findings, a human-free item detection model for public places can be developed. In addition, it is able to distinguish between hazard and non-hazardous objects. It has shown encouraging results in identifying weapons and weapons-like things in public areas. To improve the model's mAP, the weapon data could be improved by taking more photos of the target that are far away and only include visible parameters. Multiple class inference, object tracking and the ability to use these approaches in mobile situations are some potential areas of future research. For example, video inference opens the door to object tracking, which in turn paves the way for movement and pattern detection. The future viability of mobile object detection and related applications is directly tied to the optimization and enhanced efficiency of ML models, which is discussed above in relation to all of the other areas.

Acknowledgement

I would like to thank my mentor, Prof. Aaloka Anant, for his encouragement and assistance during the project's development through lectures and feedback sessions. I would also like to thank my parents for their faith in me during the course.

References

Akkas, S., Maini, S. S. and Qiu, J. (2019). A fast video image detection using tensorflow mobile networks for racing cars, *2019 IEEE International Conference on Big Data (Big Data)*, pp. 5667–5672.

Chiu, Y.-C., Tsai, C.-Y., Ruan, M.-D., Shen, G.-Y. and Lee, T.-T. (2020). Mobilenet-ssdv2: An improved object detection model for embedded systems, *2020 International Conference on System Science and Engineering (ICSSE)*, pp. 1–5.

Dahlan, I. A., Ariateja, D., Arghanie, M. A., Versantariqh, M. A., David, M. and Fatmawati, U. D. (2021). Sistem deteksi senjata otomatis menggunakan deep learning berbasis cctv cerdas, *Jurnal Sistem Cerdas* **4**(2): 126 – 141.
URL: *https://apic.id/jurnal/index.php/jsc/article/view/172*

Eriksson, K. (2021). Detecting parasites on bees with mobile object detection.

Gesick, R., Saritac, C. and Hung, C.-C. (2009). Automatic image analysis process for the detection of concealed weapons, *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*, CSIIRW '09, Association for Computing Machinery, New York, NY, USA.
URL: *https://doi.org/10.1145/1558607.1558630*

Giron, N. N. F., Billones, R. K. C., Fillone, A. M., Del Rosario, J. R., Bandala, A. A. and Dadios, E. P. (2020). Classification between pedestrians and motorcycles using fasterrcnn inception and ssd mobilenetv2, *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, IEEE, pp. 1–6.

Gopalakrishnan, K., Khaitan, S. K., Choudhary, A. N. and Agrawal, A. (2017). Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection, *Construction and Building Materials* **157**: 322–330.

Grega, M., Matiolański, A., Guzik, P. and Leszczuk, M. (2016). Automated detection of firearms and knives in a cctv image, *Sensors* **16**(1).
URL: *https://www.mdpi.com/1424-8220/16/1/47*

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications.

Kim, J. and Pavlovic, V. (2016). A shape-based approach for salient object detection using deep learning, *in* B. Leibe, J. Matas, N. Sebe and M. Welling (eds), *Computer Vision – ECCV 2016*, Springer International Publishing, Cham, pp. 455–470.

Mery, D., Svec, E. and Arias, M. (2016). Object recognition in baggage inspection using adaptive sparse representations of x-ray images, *in* T. Bräunl, B. McCane, M. Rivera and X. Yu (eds), *Image and Video Technology*, Springer International Publishing, Cham, pp. 709–720.

Ozaki, Y., Yano, M. and Onishi, M. (2017). Effective hyperparameter optimization using nelder-mead method in deep learning, *IPSJ Transactions on Computer Vision and Applications* **9**(1): 1–12.

Shu, Y., Kou, Z., Cao, Z., Wang, J. and Long, M. (2021). Zoo-tuning: Adaptive transfer from a zoo of models, *CoRR* **abs/2106.15434**.
URL: *https://arxiv.org/abs/2106.15434*

Wang, Y., Xu, C., Xu, C., Xu, C. and Tao, D. (2018). Learning versatile filters for efficient convolutional neural networks, *Advances in Neural Information Processing Systems* **31**: 1608–1618.

Xu, M., Liu, X., Liu, Y. and Lin, F. X. (2017). Accelerating convolutional neural networks for continuous mobile vision via cache reuse, *arXiv preprint arXiv:1712.01670* .

Xu, T. and Wu, Q. M. J. (2015). Multisensor concealed weapon detection using the image fusion approach, *ICDP*.