UNIVERSITY OF BAHRAIN
COLLEGE OF ENGINEERING
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT

Term Project
AIE604

CONSTRUCTION QUALITY CONTROL CLASSIFICATION & DETECTION APPLICATIONS USING DEEP LEARNING

By

Abdulaziz Sami Abdulla Mahmood Qambar **20112926**

Supervised by
Prof. Mohab Mangoud

Project Report
Submitted in partial fulfillment of the requirements of the course
Deep Learning AIE604

June 2021

ACKNOWLEDGEMENT

Firstly, I would like to thank Allah for his blessings that gave me the capability and strength to complete and submit this project. Then, I would like to extend my gratitude to Prof. Mohab Mangoud for his support, efforts, guidance, kindness, and patience through the whole semester. The advice, information, and lecturing that been provided by him were very critical for writing this report, which made my work and thinking reach to a whole new level. Also, I would like to thank my supervisor in Ministry of Housing Eng. Wala Badow for her usual encouragement, motivation, support, and cooperation through this semester. Without her understanding, I would not be able to finish this semester successfully. Finally, special thanks go to my family and to those who gave me all the support and motivation I needed to complete this project.

ABSTRACT

This project will apply Deep Learning on "5" construction site tasks which are: Construction vehicles detection, license plate detection, helmet detection, face mask detection, and concrete crack classification. First, it will give brief introduction about construction sector in Bahrain. Then it will discuss current applied practice of above-mentioned projects and how to improve it. Next, it will present the details of the "5" algorithms that will be developed including the datasets required for training and testing. Later, the algorithms results will be demonstrated and discussed. The first project detection algorithm achieved 93.95% mAP at 16.5 FPS. Second project detection algorithm achieved 92.30% mAP at 16.7 FPS. Third project algorithm achieved 94.90% mAP at 16.7 – 34.5 FPS. The fourth project detection algorithm achieved 89.53% mAP at 16.7 FPS. While the fifth project classification algorithm achieved 98.90% testing accuracy.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ANN	Artificial Neural Networks
BoF	Bag of Freebies
BoS	Bag of Specials
CNN	Convolutional Neural Networks
GDP	Gross Domestic Product
CPU	Central Processing Unit
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
FPS	Frame Per Second
GPU	Graphics Processing Unit
IoU	Intersection of Union
mAP	Mean Average Precision
mIoU	Mean Intersection of Union
NMS	Non-Maximum Suppression
PAN	Path Aggregation Network
ReLu	Rectified Linear
RoI	Region of Interest
SPP	Spatial Pyramid Pooling
SSD	Single Shot Detection
TN	True Negative
TP	True Positive
YOLO	You Only Look Once
YOLOv2	You Only Look Once Version 2
YOLOv3	You Only Look Once Version 3
YOLOv4	You Only Look Once Version 4

CHAPTER 1 – INTRODUCTION

Construction industry is one of the major industries around the globe. It is the development engine of the country that represents the foundation of providing the required facilities from buildings to infrastructure for all other industries. It reflects the country urbanization and modernity. According to Ministry of Finance & National Economy, construction sector in 2020 contributed 7.7% to Bahrain GDP [1]. A total of 330 million dinars construction projects were awarded during last year that are related to roads, housing, electricity, sewage, and telecommunication projects [1]. Foreign investments in construction also contributed a worth of 148 million BD last year [1].

Meanwhile, Kingdom of Bahrain is witnessing the construction of the biggest project in its history which is Bapco's oil refinery expansion project costing 4.2 billion dollars that is expected to be completed by 2022 [1]. On the other hand, the Kingdom recently delivered one of the mega projects in the region which is the new Bahrain International Airport with a cost of 1.1 billion dollars expected to serve 14 million visitor every year. Other projects are in progress such as Bahrain Exhibition Center in Sakhir and Al Fateh Highway Tunnel. In housing sector, Bahrain is witnessing the construction of 5 new cities which are East Hidd, Madinat Salman, Madinat Khalifa, East Sitra, and Ramli Project that consists of more than 10,000 housing unit. In near future, "2" mega projects construction will start which are King Hamad Causeway connecting to Saudi Arabia with an expected cost of 3 billion dollars [2], and Bahrain Metro project. All these projects require major efforts and complicated management system to monitor their performance, quality, and progress.

Efficiency, automation, optimization, and sustainability are the goals construction sector is interested in for a cost-effective project. In recent years, Artificial Intelligence (AI) integration with construction industry started to emerge due to its capability to achieve these goals. Hence, a new horizon of expectations showed up with unlimited opportunities to develop.

1.1 AI in Construction Industry

Artificial Intelligence been deployed in construction sector to serve in many fields as it proved its added value in terms of time, effort, and financial efficiency by reducing the dependency on humans in performing tasks. Hence, reducing the construction costs.

One of the interesting AI domains is computer vision which had been utilized intensively in construction sites management. Nevertheless, AI been deployed as a decision support expert system tool that is developed based on human experience to guide employees especially junior workers in executing their tasks. It can be also integrated with regression models to predict delay in project based on current progress. In addition, construction contractors started deploying autonomous dumping truck for repetitive tasks such as: transporting excavated material to different location within the site or performing compaction works. Similarly, drones been equipped with built in algorithms that are able to autonomously survey a specific area and collect data [3]. In 3D printing construction, robots were used as concrete injectors that automatically build the building. In offsite construction, robots been used to assemble building parts and join them together. All these applications present how AI interfered the construction sector.

1.1.1 Health & Safety in Construction Sites

Health & safety in construction sites is very critical concern. Construction sites are known to be hazardous environment [4]. Hence, AI can be used as an intelligent safety officer at sites [4]. Discarding the need to assign a human safety officer by deploying cameras around the site having built-in trained detection algorithms to monitor the labor personal protective equipment wearing including helmets and reflective vest in order to alert unsafe behavior [4]. Also, similar algorithms can be deployed to monitor the buffer distance between moving vehicles and labors [4]. Hence, monitoring the health and safety 24/7 around the site for better health & safety management performance and accidents minimization. In terms of surveillance, the same concept can be applied to protect the site boundaries from intruders by connecting the intelligent cameras with alert system, as the construction site contain expansive materials, equipment's, and tools that are kept for use such as: excavators, manhole steel covers, cables, steel reinforcement, and pumps that are easily stolen. An intelligent eagle eye system will significantly protect the site without the need to assign a security company for patrolling. Similarly, computer vision can be utilized to monitor number of machinery and workers on site by object tracking, in order to verify the daily reports being submitted by the contractor and avoid any contractual claims raised by the contractors blaming Consultants in disturbing progress flow which as actually due to providing insufficient manpower to execute obliged tasks.

1.2 Problem Definition

The problems that will be investigated in this project are observed in Ministry of Housing construction sites, the place I am employed in. They are divided to “2” categories as shown below:

1.2.1 Construction Site Access

Construction sites security and access control is an essential part of construction management. Proper site access mechanism is required to monitor visitors access, material delivery to site, machinery access, and labor deployment. In addition to preventing unauthorized personals from accessing site. In Ministry of Housing projects, manual site access monitoring mechanism is being implemented by assigning a watchman to collect delivery notes and record the dumping trucks license number leaving site “A” to deliver filling materials to site “B”. Thus, to ensure that the same truck had delivered the material to destination. For site security, the current practice in Ministry is to assign a specialized security contractor to control site access with providing patrols to roam the site for any violations or intruders. This practice is being done due to previous cases where material gets stolen, damaged, or unauthorized personal breach the site and get injured. The cost of deploying “4” security personals for 12 hours shift, 7 days a week is 1,300 BD/month. Hence, it will cost minimum more than 23,400 BD for an “18” months project. Currently, the Ministry have more than “10” on-going housing projects with more than 3,000 housing units currently being built around the “4” governorates. Some of the projects have 194 housing units such as Qalali Housing Project. Others have 1,077 housing units such as East Sitra Town – Phase 1 Housing Project. Thus, requires more than “4” security personals to be on site in such huge project. Therefore, the minimum current cost of deploying security in Ministry of Housing projects per month is estimated to be 13,000 BD/month. The current practice is inefficient, costly, time consuming, and requires continuous manual operations monitoring.

1.2.2 Health, Safety and Quality Control

According to global reports, 48% of falls from height in construction sites lead to serious injuries and 30% lead to death [5]. In Bahrain and according to Social Insurance Organization, “45” falls from height accidents been registered in Q4 of 2020, none of them resulted death [6]. Thus, health & safety in construction sites is very critical concern for Ministry of Housing. The Engineers and labors safety shall not be tolerated due to unsafe practice. Hence, the Ministry ensure deploying safety officers to roam the site on daily basis to evaluate contractor workmanship and measures taken including wearing PPE, safety belts, providing warning tapes, barriers, etc., to reduce the accidents and injuries. In addition, with COVID-19 pandemic, the Ministry highly stress on complying with precautionary measures including wearing masks, periodic sanitization, and social distancing. The health & safety officer deployment costs around 1000 BD/month/project (10,000 BD/month for 10 projects). The current practice is inefficient, costly to deploy health & safety officer in every site, time consuming and requires human personal to be deployed on site.

1.3 Motivation

This project is motivated by several reasons:

1. To utilize computer vision and deep learning in developing algorithms to enhance project management, quality monitoring, and control.
2. To create an efficient cost-effective sustainable method for site access control and health & safety management on site.
3. To minimize the need for security subcontractors & human health & safety officer in construction sites.
4. To automate the site access monitoring system.

1.4 Objective

1. To develop CNN based algorithm for real time automatic construction vehicles detection.
2. To develop CNN based algorithm for real time automatic vehicle license plate detection.
3. To develop CNN based algorithm for helmet wearing detection for workers.
4. To develop CNN based algorithm for face mask wearing detection for workers.
5. To develop classification algorithm for crack images classification.
6. To evaluate & compare the results of the developed deep learning model with literature.

1.5 Project Outline

The report is divided to 5 chapters that will discuss and develop a deep learning algorithm for "5" sub-projects. Chapter 2 will provide a brief illustration of Convolutional Neural Networks (CNN), transfer learning, object detection, and YOLO algorithm. Chapter 3 will present the details of each sub-project including their goal, dataset analysis, and proposed model architecture. While chapter 4 will demonstrate and discuss the results obtained from the trained models and will compare them with other datasets and other paper findings. At the end, chapter 5 will provide conclusion, summary, and recommendations for further investigation.

CHAPTER 2 – DEEP LEARNING OBJECT DETECTION

Human vision is extremely fast and precise enabling us to execute quick response tasks through processing a lot of information in the images within milliseconds in an efficient manner such as vehicles driving task.. From this concept, computer vision interest emerged to start the search & development journey to find an algorithm that enable the computers detect and classify a wide variety of objects like human, and even better. This chapter will set the project foundation by introducing the background information in object detection algorithms along with literature review done in construction sites security control and health & safety control.

2.1 Literature review

Xiang, X. *et al.* [7] proposed an intelligent monitoring system that uses Faster R-CNN algorithm for detecting intruder vehicles around overhead electrical transmission lines restricted zones including the ability to detect construction vehicles such as: bulldozers and excavators that have various shapes and sizes compared to the normal vehicles, which create risk of hitting power towers and disturbing electricity supply. The authors have utilized transfer learning technique and removed the classifier part, in addition to modifying the region of interest pooling layer position in the algorithm for better detection in complex scenes. The algorithm been deployed on HD cameras 6m to 10m above ground at 300m distance from each other connected with 4G cellular connection for data transmission. The authors have created their own training and testing dataset from images taken from various locations. The authors demonstrated that their detection algorithm achieves good results, and it is robust against various brightness levels and weather conditions, and it can be used for early warning applications especially in rural areas.

Fang, W. *et al.* [8] proposed an improved Faster R-CNN algorithm to detect construction site workers and excavators. The algorithm been trained on 8,500 images and tested against 1,500 images. It managed to achieve a detection rate of 0.101 second per image with AP of 91% and 95% for workers and excavators, respectively. The authors highlighted that the object scale and occlusion had an adverse impact on algorithm detection accuracy. Thus, they recommended to study the optimum placement of cameras around the construction site to mitigate such effect.

Xiao, B., *et al* [9] have developed a 10,000 images dataset manually labeled for 10 classes of construction machinery called Alberta Construction Image Dataset (ACID). The dataset includes images for excavator, compactor, bulldozer, grader, dump truck, concrete mixer truck, wheel loader, backhoe loader, tower crane, and mobile crane that were collected from online and on-site sources. The pictures were captured either through an on-site camera, or UAV, or personal camera. The dataset was tested for 200,000 epochs on "4" object detection algorithms achieving 83% mAP with Inception-SSD, 87.8% with YOLO-v3, 88.8% with R-FCN-ResNet101, and 89.2% with Faster R-CNN-ResNet101. The average object detection speed for all algorithms was 16.7 FPS.

Fang, W. *et al.* [5] developed "2" object detection algorithms to classify workers wearing safety belt to hold them from falling while working at heights. The first algorithm uses Faster R-CNN to detect persons. The second algorithm is CNN based model to detect safety belts. The algorithms proved their reliability by achieving a precision and recall scores of 99% & 95%, respectively for Faster R-CNN, and 80% & 98%, respectively for safety-built detection algorithm. The authors have indicated that their algorithm is limited to only few tasks of workers activities from height. They have also pointed that the dataset created is small. Thus, the algorithm was not able to detect objects in certain situations. In addition, they have highlighted that the dense environment of construction site further complicates the object detection.

Chen, S. *et al.* [10] proposed an improved Faster R-CNN algorithm that uses ResNet-101 as a backbone for helmet detection integrated with K-means algorithm for improving detection accuracy. The detection algorithm been trained for 20,000 epochs on 1,065 images dataset distributed over "2" classes of helmet and head that were split to 90% training and 10% testing. The results obtained showed that the algorithm achieved 94.3% mAP at 11.6 FPS. The same dataset was tested on YOLO and standard faster R-CNN algorithms which achieved 85.6% mAP and 86.2% mAP, respectively.

Fang, Q. *et al.* [4] proposed Faster R-CNN algorithm to detect construction workers not wearing helmet from far-field surveillance cameras. The algorithm been trained on 100,000 manually labeled far images dataset taken from 25 different construction site over a period of 1 year during different weather conditions based on 5 environment categories that are further split to 19 subcategories. The 5 environments categories are weather, illumination, person posture, range, and occlusions. The dataset was divided to

81,000 training image and 19,000 testing images. The testing results achieved 95.7% precision and 94.9% recall.

Nagrath, P. *et al.* [11] proposed a Single Shot Multibox Detector algorithm based on ResNet-10 backbone and MobileNetV2 classifier for face mask detection. The algorithm been trained on both real and artificial mask images dataset that consists of 5,521 images that were split to 80% / 20% with applying data augmentation techniques. The testing results obtained achieved 92.64% accuracy at 15.71 FPS.

Yu, J. *et al.* [12] proposed an improved YOLOv4 algorithm for face mask detection. The improvement was applied on the algorithm backbone by introducing a computation reduction technique integrated with an adaptive image scaling algorithm along with an improved Path Aggregation Network for an enhanced semantic information collection in the feature layer with utilizing Hard-Swish activation function instead of Leaky ReLu. The images dataset used consist of 10,855 images distributed over 3 classes (mask, no mask, irregular mask) which are split to 72% training, 8% validation, and 20% testing. The testing results obtained achieved 98.2% recall, 98.3 mAP @ 0.5 reaching 54.57 FPS.

Table 1: Literature summary of construction sites control and health & safety

Paper	Algorithm	Task	Dataset	Recall	mAP@0.5	FPS
Xiao, B., *et al* [9]	SSD YOLOv3 R-FCN Faster R-CNN	Construction Vehicles	10,000	-----	83.0% 87.8% 88.8% 89.2%	26.3 20.8 8.3 11.5
Fang, W. *et al.* [8]	Improved Faster R-CNN	Excavators Workers	10,000	81.0% 79.0%	95.0% 91.0%	9.90
Fang, W. *et al.* [5]	Faster R-CNN CNN	Persons Safety Belts	770	95.0% 98.0%	-----	-----
Chen, S. *et al.* [10]	YOLO Faster R-CNN Improved Faster R-CNN	Helmets	1,065	-----	85.6% 86.2% 94.3%	11.6
Fang, Q. *et al.* [4]	Faster R-CNN	Non-helmet wearing	100,000	94.9%	-----	4.95
Nagrath, P. *et al.* [11]	SSD	Mask No Mask	5,521	85.0% 100%	92.6%	15.71
Yu, J. *et al.* [12]	Imp. YOLOv4 YOLOv4 YOLOv3 SSD Faster R-CNN	Mask No Mask Irregular	10,855	98.2% 92.3% 88.1% ----- -----	98.3% 95.2% 94.8% 97.2% 95.6%	54.57 23.83 21.39 34.69 2.44

2.2 Convolutional Neural Network

Image classification is defined by "*the task of assigning an image to a predefined category*" [13]. The best type of algorithms to perform such task are the Convolutional Neural Networks (CNN). In classification tasks, the images contain only single object which is first preprocessed to standardize its configuration. Then, the features that represent the object properties such as curves, lines or color are extracted and fed to a classifier that predicts the class of the image based on earlier provided labeled training data [13].

Convolutional Neural Networks (CNN) consist of "2" parts as shown in figure (1). The first part is locally connected convolutional layers where each neuron is connected to a group of pixels instead of connecting each neuron to a pixel. This part is concerned with feature extraction. i.e., it will produce a new image that contains the input image features using kernels (filters). The kernels weight represents the hidden layer weights. In fact, kernel weights reflect how significant a pixel in determine the classification of a picture. The second part consists of fully connected layers that are responsible for the classification task. CCN are distinguished from Artificial Neural Networks (ANN) that they consider the spatial features of the pixels which enhance the network performance.

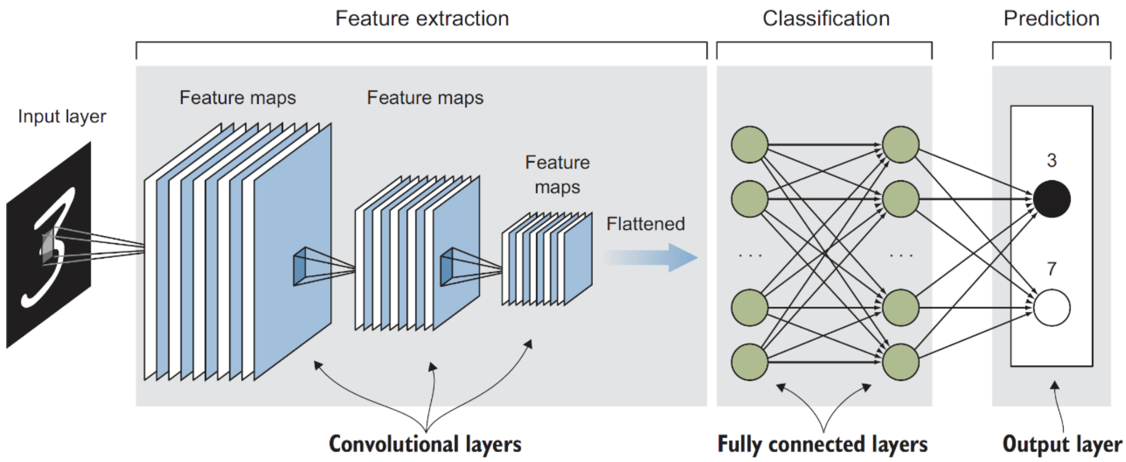

Figure 1: Typical Convolutional Neural Network architecture [13]

During the first part, each convolutional layer will extract the features from the input image and produce feature maps. Each feature map learns more complicated features such as lines, than circles, than face. Next, the features will be flattened to form a long 1D vector that will be fed to the fully connected layer (Classification part).

2.3 Transfer Learning

Transfer learning is the transfer of learned knowledge in a model that collected the features of a large dataset to another new model to perform new task [13]. It can be called the learning shortcut in algorithms. The idea behind it is to utilize an already trained and tuned model and apply it on a dataset for a certain task instead of training a new model from scratch. Or it can be utilized if a model is required to be developed with insufficient data or the data is difficult to be collected. Transfer Learning technique will reduce the training process duration and will achieve higher results. A variety of models can be used as a pretrained model such as: VGG16, ResNet and MobileNet. These models were trained on huge datasets like ImageNet which consists of millions of images [13]. Hence, it can contribute to generalizing the new model and avoid overfitting. It should be noted that the basic features learned in the pretrained model are usually similar to the basic features that will be learned from another dataset as they represent edges and lines. Transfer learning is being deployed in "3" different situations. Either they are deployed as classifiers or feature extractor or fine tuning [13]. In classifier case, the pretrained model is utilized as it is without training or removing any part of it. In the feature extractor situation, the pretrained model feature extractor part will be frozen, and the classifier part will be removed. A new classifier part will be inserted and trained on the new dataset. However, in the fine-tuning case, the user removes the classifier part and part of the feature extractor and replace them with new configuration to train them [13].

2.4 Object Detection & Hyperparameters

Object detection is defined as the ability to obtain a specific object position (localization) in an image with classifying it [13]. Object detection task is executed by predicting the bounding box coordinates of the identified object in terms of pixels along with stating its classification. The most famous algorithms are: Faster R-CNN, Single Shot Detection (SSD) & You Only Look Once (YOLO) [13].

2.4.1 Region of Interest (RoI)

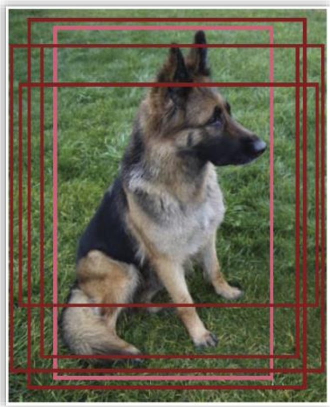

Figure 2: Multiple RoIs detected [13]

Region of Interest (RoI) are the area's that the model think it might have an object. The model will produce thousands of RoI's in the image (figure 2). Each RoI proposed will be evaluated based on its objectness score that represent how likely these RoI's contain an object.

$$P_0 = P_r(Probability\ containing\ object) \times IoU$$

$$Where\ P_r = p(Class|object)$$

The RoI's objectness score that exceed certain threshold will be passed for further processing as these are considered as foreground. Others will be neglected as they will be considered as background. The objectness score threshold is considered as hyperparameter that can control the trade-off between expansive computations and number of RoI's passed.

2.4.2 Network Predictions

For object detection, it is always preferred to use a pretrained model to extract the image features. During this process, the network will predict the bounding box coordinates that consist of 4 numbers (x, y, w, h) representing the RoI center coordinates, RoI width, and RoI height respectively through using a regressor such as a fully connected neural network. In addition to predicting the RoI classification.

Nowadays, the feature extractor model most preferred is the ResNet models rather than VGG models [13]. ResNet models are distinguished by being more complex and deeper than other models. Hence, being more capable to extract and learn features. In addition, ResNet uses new techniques that are used in other models such as: residual connections and batch normalization.

2.4.3 Non-Maximum Suppression (NMS)

After extracting the features of every qualified bounding box, it is high likely that the algorithm predicted several bounding boxes of the same object that overlap each other. Thus, the Non-maximum suppression (NMS) technique objective is to ensure that every object is only detected once by combining the overlapping areas of the bounding boxes and neglecting the non-overlapping areas to produce a single bounding box for the identified object (figure 3). This process is executed by analyzing all proposed RoIs. If the RoI exceed "confidence threshold", the RoI will be kept, otherwise will be neglected. Next, the remaining RoIs will be compared to find the RoI having the highest probability that it contains an object. NMS then will calculate the overlapping using a metric called "Intersection over Union (IoU)" between RoIs having the same classification prediction and merge them together. As a usual practice, IoU hyperparameter is kept at 0.5.

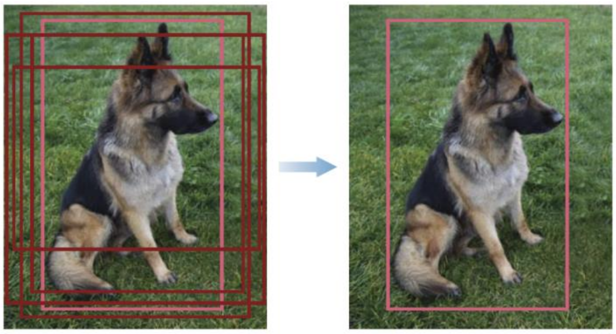

Figure 3: RoIs Before NMS & after applying NMS [13]

2.4.4 Hyperparameters

As discussed above, there are several hyperparameters that can be tuned during object detection implementation. For instance, IoU measures the area of the predicted bounding box overlapping with the ground truth bounding box that is labeled manually (figure 4).

Figure 4: IoU examples (Green box = Ground truth – Red box = Predicted box) [13]

If the IoU value exceeds certain thresholds which usually kept at 0.5, the RoI will be considered as "True Positive" object detection prediction. Others will be considered as "False Positive". Once both these values are calculated for each class identified in the image, both precision and recall for each class can be found. Hence, a precision-recall curve (figure 5) for every class can be plotted which shows how the recall is affected by the precision. The more the precision value stays high as the recall increase, the object detection is considered acceptable. In order to combine the object detection evaluation of all classes identified, a new evaluation metric will be introduced which is called "Mean Average Precision (mAP)", that is concerned with object recognition. mAP can be measured by calculating the area under the precision-recall curve of all classes identified and then taking the average. On the other hand, the most important evaluation metric is the Frames Per Second (FPS). FPS measures the object detection speed of the algorithm. For instance, Faster R-CNN operates at 7 FPS while SSD operates at 59 FPS. However, it should be noted that there is a trade-off between detection speed and detection accuracy. The more detection speed required; the less accuracy detection becomes.

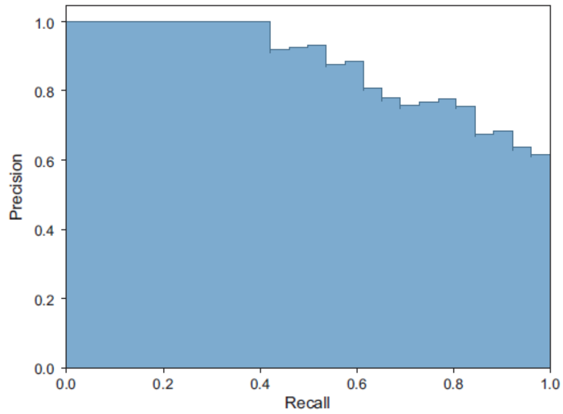

Figure 5: Precision-recall curve sample [13]

2.4.5 Anchors

Bounding boxes are defined as boxes that point for the existence of an object. It is identified by "4" points which are: (x,y) describing the box center coordinates, and (w, h) describing the width and height of the box. However, it was found that identifying the center coordinates of an object within the bounding box is complicated. Thus, the anchor boxes are identified. Anchor boxes idea is to locate an anchor at the center of every sliding window that generates from the last feature map a specified number of anchor boxes of different aspect ratio having the same center. If the sliding

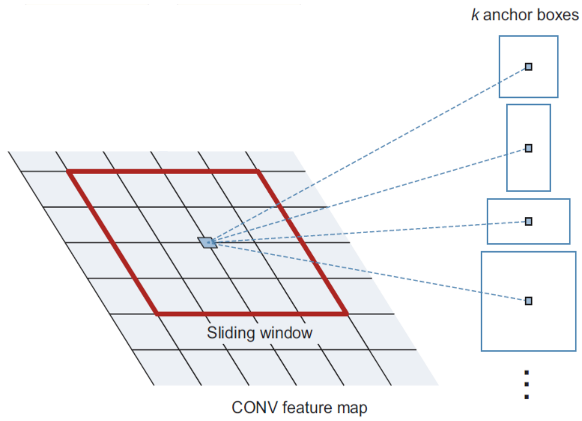

Figure 6: Anchor boxes [13]

window is suspected to contain an object, from each anchor box generated in every sliding window, an offset will be calculated to measure the difference between the ground truth and anchor box edges. The information will be generated by the fully connected layer to be used as a reference to adjust the fit on the object detected. Then, the IoU will be calculated for each anchor box to select the best bounding box covering the object most for classification. As it is shown in figure 6, several sizes of anchor boxes will be produced from every sliding window.

2.5 You Only Look Once (YOLO) Algorithm

You Only Look Once object detection algorithm was proposed in 2016 by Joseph Redmon *et al*. after been inspired by GoogLeNet architecture [14]. The authors called it as DarkNet [14]. The original YOLO algorithm consists of 24 convolutional layers for feature extraction, followed by 2 fully connected layers for class & bounding box coordinates prediction [14]. YOLO algorithm uses 1 x 1 filter as feature reducer for the previous layer. In order to increase the algorithm efficiency, YOLO train the classification convolution layers on half of the input image size. While it trains the detection convolutional layers on original input image size. All layers use leaky rectified linear (Leaky ReLu) activation function except the last layer that uses a linear activation function. It is distinguished by having only single neural network that predicts the object bounding box coordinates and its class from the image [14]. The algorithm was trained on PASCAL VOC detection dataset [14].

YOLO object detection algorithm is preferred over the other algorithms such as: R-CNN and SSD for the following reasons:

1. YOLO achieves high object detection frames per second reaching 45 FPS in traditional YOLO algorithm version 1 [14].
2. YOLO doesn't use the window sliding concept used in R-CNN as it analyzes the features of the whole image simultaneously during training. Leading to less false positives [14] and enabling a better feature context understanding.
3. YOLO algorithm results are considered more generalized compared to other algorithms as it can be applied on new domains and perform well [14].

On the other hand, due to the high detection speed in YOLO, the accuracy results are affected and are lower than R-CNN and SSD when compared to the original YOLO specially in localization and recall parameters [15].

The algorithm concept is based on dividing the image to a grid of cells. Each cell will predict number of bounding boxes coordinates, objectness score ranging between [0,1], and the object classification. The cell responsible for detecting the object is the cell that the ground truth bounding box center coordinates fall in it. Also, each cell only predicts "1" class probability, regardless of bounding box number in that cell [14]. Then, the identified bounding boxes will be passed to NMS layer to suppress the overlapping bounding boxes as shown below in figure 10.

Figure 7: YOLO version 3 architecture [13]

During testing, each identified object bounding box confidence will be multiplied by its predicted class probability as per the features detected. Both equation sides will be compared to determine the prediction accuracy as shown below [14]:

$$P_r(Class|Object) \times P_r(Object) \times IOU = P_r(Class) \times IOU$$

Since that time, several versions of YOLO been introduced that solve the issues of the preceding versions. For instance, batch normalization, increased processing resolution, and anchor boxes been introduced in YOLOv2 [15]. The network size was increased to 106 fully convolutional layers in YOLOv3. "53" layers are for feature extraction that are trained on ImageNet dataset. In addition to "53" layers for object detection. Also, a Sigmoid activation function was used for anchor box center coordinates prediction instead of Softmax activation function [16]. The reason behind this change is that Softmax activation function is based on predicting single class per bounding box [16]. Which is not always correct. Thus, Sigmoid activation function is preferred as it allow multi-labeling approach [16]. It should be highlighted that it contradicts with the general practice of using Softmax activation function for multi label prediction. However, these updates pushed the performance to achieve similar performance to SSD but 3 times faster, better and 1.5 times faster than ResNet-101, and similar performance to ResNet-152 but 2 times faster and efficient due to the significant smaller size compared to ResNet-152 (30% less size) [16]. However, according to the authors, YOLOv3 suffered slightly with detecting medium to large objected compared to YOLOv1 that suffered from detecting small object [16].

2.6 YOLO v4 Algorithm

In 2020, YOLOv4 was released by Alexey Bochkovskiy *et al* [17], which can achieve 10% better mAP results and 12% better FPS results compared to YOLOv3 as shown in figure 11 [17]. It was developed to be efficient and able to operate on conventional Graphics Processing Unit (GPU) by applying specialized techniques such as: Weighted Residual Connections, Cross Stage Partial Connections, Cross mini batch normalization, and self-adversarial training that enables a fast object detection process [17]. It should be noted that YOLOv5 was released few days after YOLOv4 release. However, YOLOv5 is developed only for PyTorch application.

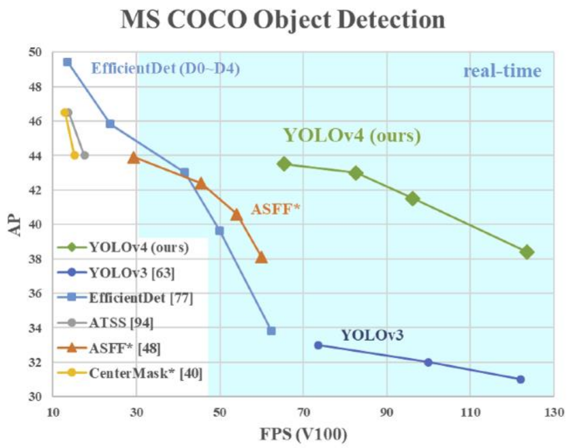

Figure 8: YOLO version 4 performance [17]

2.6.1 YOLO Version 4 Architecture

YOLOv4 architecture is divided to 4 parts which are: Backbone, Neck, Dense Prediction, and Sparse Prediction [17] as shown in figure (9).

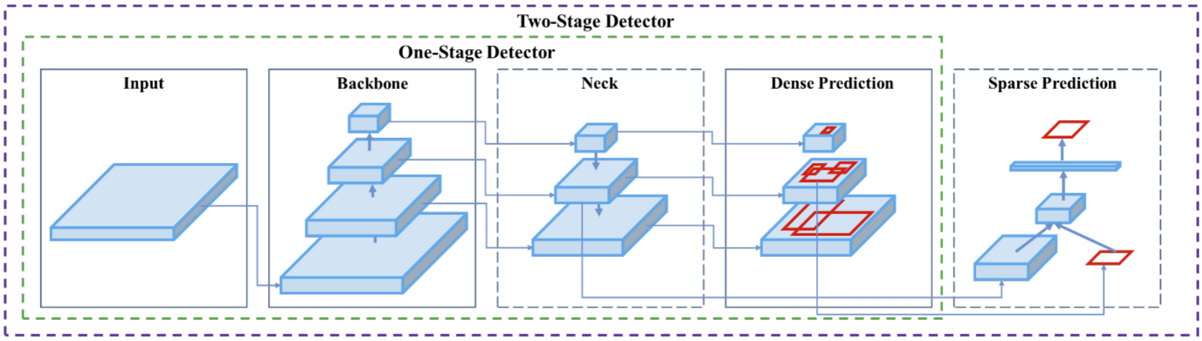

Figure 9: YOLO version 4 architecture [17]

The backbone of YOLOv4 is concerned with image features extraction. YOLOv4 backbone is based on CSPDarknet53 architecture that is trained on ImageNet dataset. The chosen architecture is distinguished for utilizing the parallel computations to expedite the feature extraction process [17]. However, the developer can use VGG or ResNet architecture as a YOLOv4 backbone if needed. YOLOv4 authors highlighted that CSPDarknet53 performance is the best for object detection tasks compared to others [17].

The next part is the Neck. Its purpose is to increase the information aggregated in the feature extraction part by collecting feature maps at different backbone stages and providing them to up sampling and down sampling convolutional layer blocks equipped with specialized techniques such as: Path Aggregation Network (PAN) to increase features extraction performance [17]. It combines the detection results of feature maps scales of 13 x 13, 26 x 26, and 52 x 52 to generate a detection [18].

The final part is the head which is a combination of one stage and two stage object detectors. The one stage object detector is called Dense Prediction that is used to localize the bounding boxes and classify the objects identified. The Dense Prediction functionality is similar to YOLOv3 grid cells division [17]. The two-stage object detector is called Sparse Prediction which is used to improve the object detection & classification results through utilizing “2” group of techniques which are: Bag of Freebies and Bag of Specials [17].

2.6.2 Bag of Freebies (BoF)

Bag of Freebies are combination of training strategies and techniques that aim to improve the algorithm performance without increasing the computational cost. The most familiar example of this category is the data augmentation. The traditional data augmentation is used for creating multiple varieties from training images to enhance the algorithm robustness and reduce texture bias by exposing it to different object context and positions. The data augmentation is divided to “2” types. The first data augmentation technique is called photometric distortions which adjust the brightness, contrast, hue, saturation, and noise [17]. The other data augmentation technique is geometric distortion which adds random scaling, cropping, flipping, and rotating [17]. For example, an advance data augmentation technique called CutOut randomly selects pixel blocks in the image and turn them off to simulate an object partially covered by an obstacle [17]. Similarly, a technique called DropBlock applies the same concept directly on the feature maps rather than the input image [17]. Hence, improving the detector capability to detect objects from partial features. Another data augmentation technique specially developed by YOLOv4 authors called Mosaic. It merges “4” random training images in 1 image, to have an image that contains a combination of object classes in different sizes and context.

Moreover, other techniques evolved the bounding box coordinates prediction which is traditionally performed by applying regression and MSE [17]. The current technique predicts these coordinates as independent points without taking into consideration the object. However, the updated technique integrates predicting the bounding box coordinates with the IoU percentage and ground truth bounding box. Hence, further improving the results [17].

2.6.3 Bag of Specials (BoS)

Bag of Specials are combination of plugins and post processing techniques that significantly enhance the algorithm performance with an increase in the prediction computation cost that is less than the performance improvement in terms of percentage. The commonly used techniques are receptive field enlargement, screening prediction results, and non-maximum suppression. For instance, Spatial Pyramid Pooling (SPP) technique is used for enhancing receptive field by splitting the extracted feature maps to b x b blocks to form a pyramid that is integrated with CNN and max pooling to further enhance algorithm robustness, faster convergence, and improve generalization. SPP technique improved YOLOv3 mAP by 2.7% with an increase of computational cost by 0.5% [17].

2.6.4 YOLO Version 4 Techniques

YOLOv4 uses ReLU and Leaky-ReLU as CNN layers activation function. It also incorporates the updated bounding box technique integrating the MSE & IoU. In addition to CutOut for data augmentation, and both Dropout & DropBlock as regularization techniques. Batch normalization been also used to normalize the values. YOLOv4 is equipped with SPP technique after the CSPDarknet53 architecture. In addition to PAN technique that is used to integrate different level of features in the backbone to different detection levels in the head. Last but not least, residual connections and cross stage partial connections been both deployed to mitigate the effect of gradient dispersion or explosion [18].

Table 2: YOLOv4 architecture & techniques summary

Backbone (Feature Extractor)	**Neck**	**Head (Detector)**
CSPDarknet53	SPP & PAN	YOLOv3
Bag of Freebies (BoF)		**Bag of Freebies (BoF)**
Mosiac DropBlock		MAE – IoU (CIoU Loss) DropBlock Mosiac
Bag of Specials (BoS)		**Bag of Specials (BoS)**
Skip connections: Cross stage partial connections		PAN

Despite Faster R-CNN algorithm is the most accurate in object detection, yet it is not the quickest [8], [4]. On the other hand, YOLOv3 utilizes Feature Pyramid Network (FPN) [19]. While YOLOv4 utilizes Path Aggregation Network (PAN) to enable the object detection in different levels as explained earlier [19]. Hence, an increase in AP and FPS can be achieved which makes it better in performance compared to YOLOv3 [19]. Based on above analysis and our intended projects, we have decided to develop the object detection algorithms using a YOLOv4 with utilizing transfer learning merits and training part of the feature extraction and a whole new classifier as per our objectives.

CHAPTER 3 – DEEP LEARNING APPLICATIONS DETAILS

In this chapter, "5" different applications will be executed to demonstrate the benefits of deploying deep learning in construction sites for getting an unprecedent insight of information.

3.1 Algorithm Application Framework

Every object detection problem shown in this research will be developed using Python language on Google Collaboratory Pro platform and will follow a pre-defined structured framework from preprocessing stage to results evaluation in order to ensure consistency in execution. The framework is divided to 4 stages as shown below:

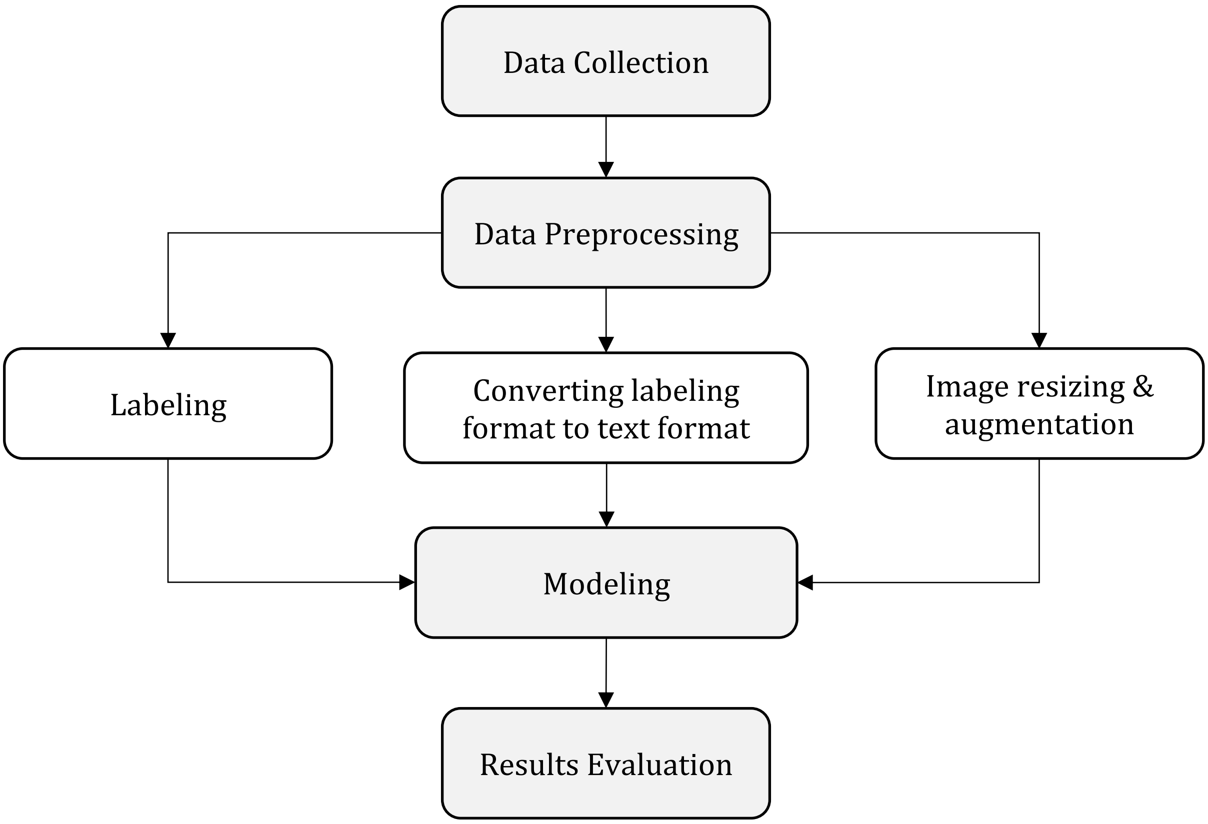

Figure 10: Modeling Framework

3.2 Object Detection Applications

This section will illustrate the applications details that will be developed in this project. The projects that will be developed are:

1. Development of construction vehicles detection algorithm.
2. Development vehicle license plate detection algorithm.
3. Development Helmet detection algorithms.
4. Development of Face Mask Detection algorithm.
5. Development of concrete crack classification algorithm.

3.2.1 Construction Vehicles Detection Algorithm

An object detection algorithm will be trained using YOLOv4 algorithm to detect "3" types of construction vehicles. The dataset used to train the algorithm contains 2,825 labeled images with XML format which is not applicable for YOLO algorithms, and approximately 6,000 object of construction vehicles which are: Concrete mixer truck, dump truck, and excavator. The dataset is divided to 1,836 training images and 989 testing images. The dataset was retrieved from: https://cutt.ly/KntMHLO

Table 3: Construction vehicles dataset details

Dataset	Class	Training (Objects)	Testing (Objects)
1,836 Training 989 Testing	Concrete Mixer	550	285
	Dump Truck	1531	1012
	Excavator	1789	828

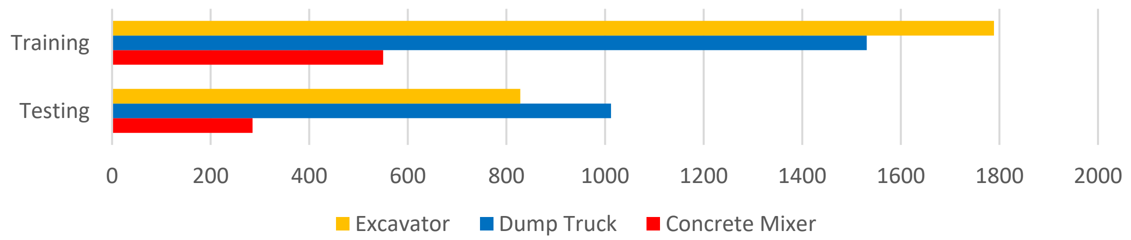

Figure 11: Construction vehicles images

3.2.2 Vehicles License Plate Dataset

An object detection algorithm will be trained using YOLOv4 algorithm to detect vehicles license plate. The dataset used to train the algorithm contains 2,386 labeled images with text format, and more than 3,400 objects. The dataset is divided to 2,000 training images and 386 testing images. The images original size varies. The dataset was retrieved from: https://cutt.ly/onsSQnE

Table 4: License plates dataset details

Dataset	Class	Training (Objects)	Testing (Objects)
2,000 Training 386 Testing	License Plate	2,934	512

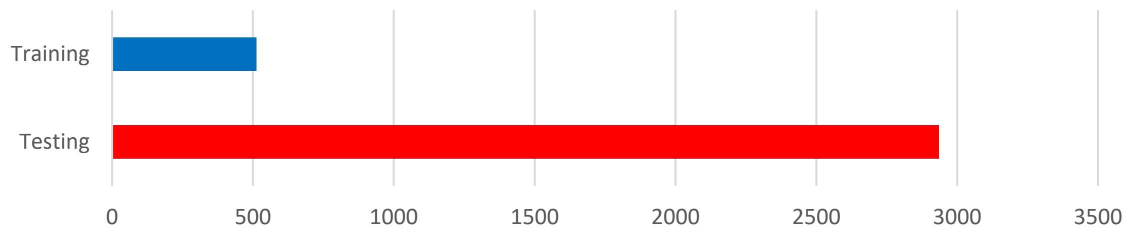

Figure 12: License plate images

3.2.3 Helmet Dataset

An object detection algorithm will be trained using YOLOv4 algorithm to detect “2” objects. The dataset used to train the algorithm contains 7,035 labeled images with text format, and more than 25,000 objects of: No helmet, helmet. The dataset is divided to 5,269 training images and 1,766 testing images. The dataset was retrieved from: https://cutt.ly/tnt1COC

Table 5: Helmet dataset details

Image Size	Class	Training (Objects)	Testing (Objects)
5,269 Training 1,766 Testing	No Helmet	14,810	4,937
	Helmet	5,008	1,669

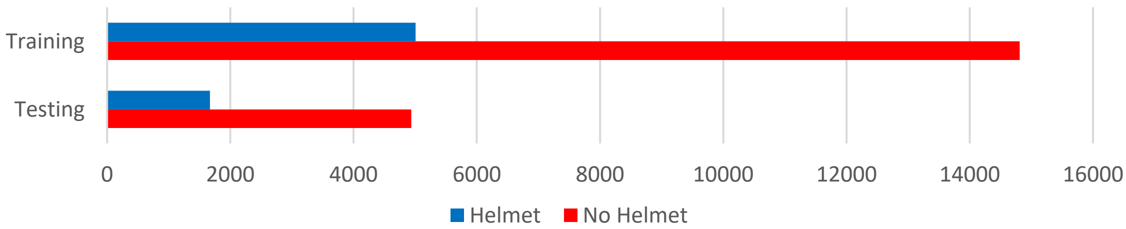

Figure 13: Helmet images

3.2.4 Face Mask Dataset

An object detection algorithm will be trained using YOLOv4 algorithm to detect "2" objects. The dataset used to train the algorithm contains 827 labeled images with text format, and 3,900 objects of: Mask, No Mask. The dataset is divided to 630 training images and 197 testing images. The dataset was retrieved from: https://cutt.ly/Snd8Y2m

Table 6: Face mask dataset details

Image Size	Class	Training (Objects)	Testing (Objects)
630 Training 197 Testing	Mask	2386	798
	No Mask	542	174

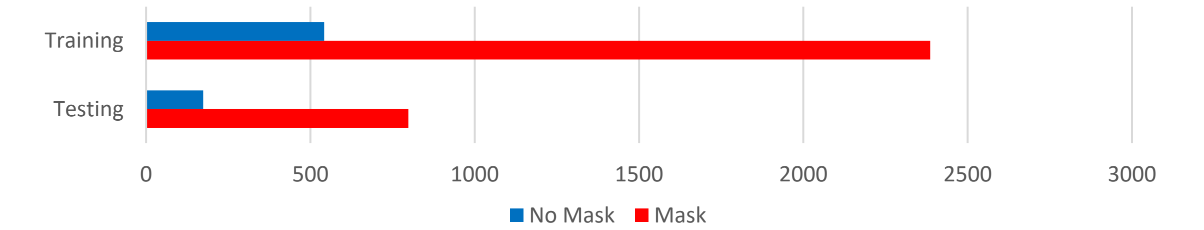

Figure 14: Face mask images

3.2.5 Object Detection Algorithm Configuration

The object detection algorithms will be developed in this project based on Convolutional Neural Network. Our proposed algorithms will utilize transfer learning technique to use YOLOv4 architecture along with Microsoft COCO dataset pretrained weights which was trained on over 300,000 images and 2 million objects in 80 classes. In addition to removing its classifier and part of the feature extraction [20]. A new part of feature extraction and a whole classifier will be trained to detect construction vehicles (Project No.1), detect license plate (Project No.2), detect helmets (Problem No.3), detect mask (Problem No.4). The input image size is fixed at 960 x 960 x 3 pixels for all object detection algorithms.

Table 7: Pavement crack detection algorithm configuration & head (detector) details

Input Size	Channels	Batch	Mini batch	Filters
960 x 960	3	64	48	(Classes + 5) x 3 = 24 (Problem 1) (Classes + 5) x 3 = 18 (Problem 2) (Classes + 5) x 3 = 21 (Problem 3,4)

Learning Rate	Momentum	Decay	Max Batches
0.001	0.949	0.0005	6000

	Type	Filters	Size	Output
	Convolutional	32	3 x 3	640 x 640
	Convolutional	64	3 x 3 / 2	320 x 320
1x	Convolutional	32	1 x 1	
	Convolutional	64	3 x 3	
	Residual			160 x 160
	Convolutional	128	3 x 3 / 2	80 x 80
2x	Convolutional	64	1 x 1	
	Convolutional	128	3 x 3	
	Residual			80 x 80
	Convolutional	256	3 x 3 / 2	40 x 40
8x	Convolutional	128	1 x 1	
	Convolutional	256	3 x 3	
	Residual			40 x 40
	Convolutional	512	3 x 3 /2	20 x 20
8x	Convolutional	256	1 x 1	
	Convolutional	512	3 x 3	
	Residual			20 x 20
	Convolutional	1024	3 x 3 / 2	10 x 10
4x	Convolutional	512	1 x 1	
	Convolutional	1024	3 x 3	
	Residual			10 x 10
	Average Pooling		Global	
	Fully Connected		1000	
	SoftMax		2	

The YOLOv4 code been retrieved from "2" links which will be used for all detection algorithms in this project. The first link is YOLOv4 owner Github page that contains YOLOv4 model. The other link is to activate it on TensorFlow:

1. https://cutt.ly/4bLDgxz
2. https://cutt.ly/PbLDjp7

3.2.6 Data Augmentation

The images augmentation was applied to increase the robustness of the algorithm to detect objects in various conditions. Augmentation techniques applied include zooming in and out by 10%, hue, exposure, saturation, rotation, and mosaic (merging of images).

3.2.7 Images labeling

The construction vehicles dataset was partially labeled manually using LabelImg software till the label format conversion method was discovered which will be discussed in subsection 3.2.8 LabelImg is a free opensource tool written in Python for images labeling. The labeling tool can save the annotated images as XML files in PASCAL VOC dataset format that is used by ImageNet or saving it as text format that is compatible with YOLO format. The labeling can be installed by following the instructions in the link and accessed from Command Prompt by typing LabelImg: https://cutt.ly/MnqcaMA

Figure 15: LabelImg tool graphical user interface

The annotation created will be named automatically with similar name to the annotated image for each image separately. The text will start with class ID, bounding box center coordinates, height, and width of the bounding box [19]. Each numeric value will be separated by space to differentiate them. Below is example of image label in text format:

1 0.567891 0.345678 0.067891 0.014596

0 0.633355 0.527252 0.078616 0.091214

3.2.8 Labeling Format Conversion

During the dataset search, the construction vehicles images dataset was found with label in XML format which is not compatible with YOLO. Therefore, in order to convert the labeling to text format, Roboflow website been used. Roboflow website provides image processing services including labeling, augmentation, and annotation format conversion with different pricing plans. To prepare the collected datasets, the image datasets with their XML annotation been merged in single file and uploaded to "roboflow" website for annotation conversion. It should be noted that the website charges 10$ for every 1,000 converted images. It can be accessed through this link: https://cutt.ly/YnqmN1e

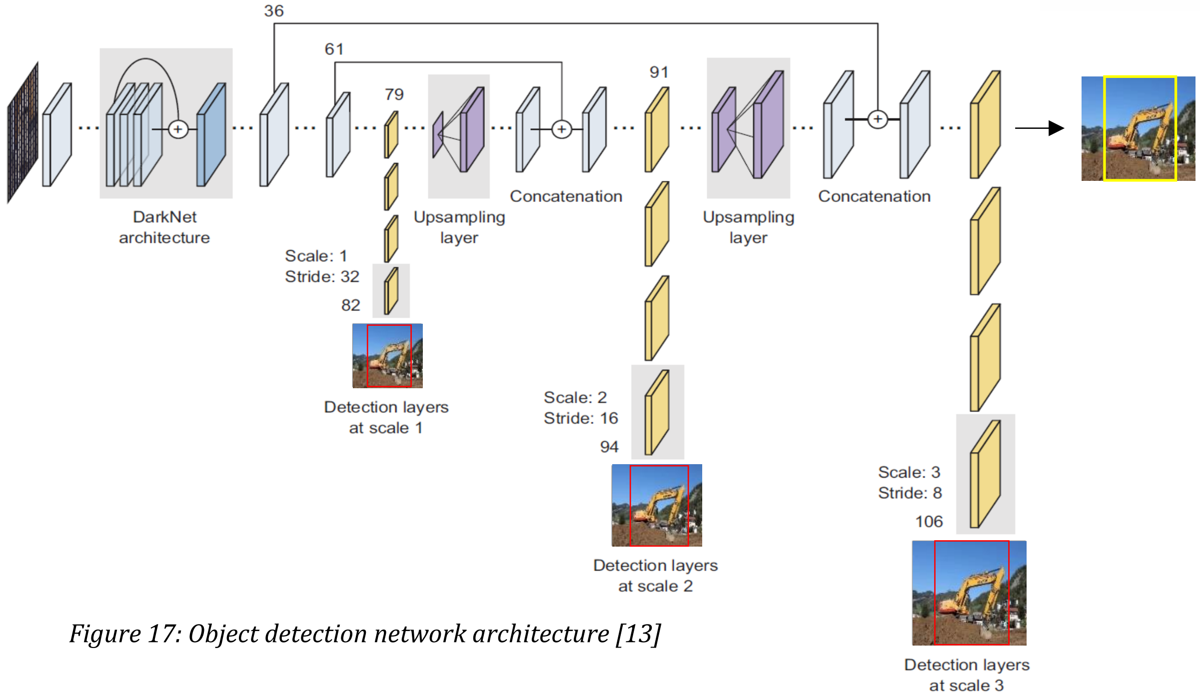

Figure 17: Object detection network architecture [13]

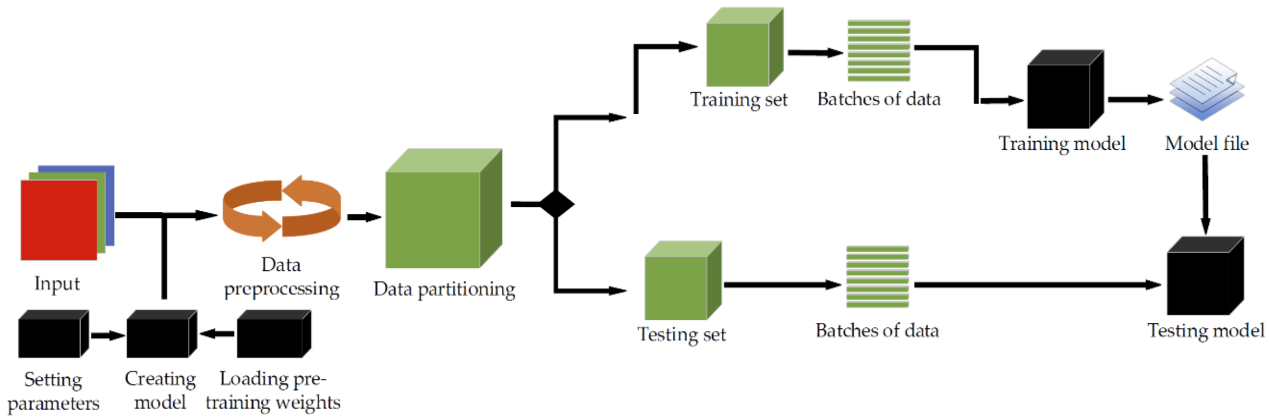

Figure 17: Algorithm's framework summary [12]

3.3 Concrete Crack Dataset

A crack classification algorithm will be trained based on several model architectures to determine the best combination of convolutional layers, dropout, batch normalization, and weight decay. The dataset used to train the algorithm contains of 40,000 images divided to 20,000 crack images and 20,000 uncracked images. The dataset was divided to 24,000 training images (12,000 each class), 8,000 validation images (4,000 each class), and 8,000 testing images (4,000 each class). The images original size is 227 x 227 pixels. The dataset was retrieved from: https://cutt.ly/cnt3Mmo

Table 8: Crack dataset details

Image Size	Class	Training	Validation	Testing
227 x 227 x 3	Cracked	12000	4000	4000
	Uncracked	12000	4000	4000

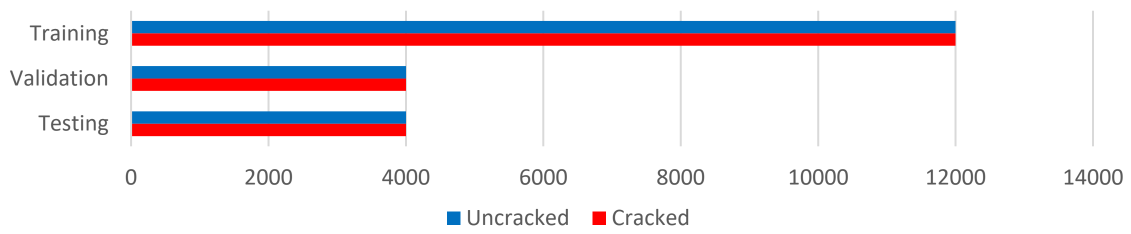

Figure 18: Concrete crack images dataset

3.4 Hardware & Software Requirements

Developing object detection algorithms require high-end hardware equipment and software installations to perform the computations. Below are the details:

3.4.1 Hardware Specifications

The device used to develop the algorithms is DELL XPS15 equipped with Intel Core i7-10750H CPU (10th Generation) @ 2.60 GHz, 16 GB RAM DDR4, 1 TB SSD, and NVIDIA GeForce GTX 1650 Ti GPU. However, due to the extreme requirement needed to train the algorithm, a specialized graphics card was rented from Google Collaboratory with a cost of 9.99 $ / month to develop this project. In off-peak hours, access to T4 or P100 GPU will be available for usage. T4 GPU has 320 Turing Tensor cores, 2560 NVIDA CUDA cores able to perform single precision performance of 8.1 TFLOPS and mixed precision performance of 65 TFLOPS. On the other hand, P100 has 3584 NVIDIA CUDA cores able to perform single precision performance of 9.3 TFLOPS. In addition, with this subscription, the user gets access to 25.46 GB RAM which is double the speed of the free Google Colab version.

3.4.2 Software Requirements

To develop an object detection algorithm, 10 software's are required to be installed which are: Python installation, GIT installation, Cmake Installation, Visual Studio Installation, GPU drivers latest update, CUDA installation, CuDNN installation, OpenCV installation, Cmake Open CV configuration, and building OpenCV in visual Studio. The steps to install the above-mentioned requirements can be found on the internet.

3.5 Algorithms Evaluation Metrics

Several parameters will be used to evaluate the algorithms performance such as: Precision, recall, accuracy, confusion matrix, F-score, IoU, and mAP metrics.

A. True Positive (TP): Model correctly identified and classified positive case [7], [13].
B. True Negative (TN): Model correctly identified and classified negative case [7], [13].
C. False Positive (FP): Model misclassified negative case to be positive case [7], [13].
D. False Negative (FN): Model misclassified positive case to be negative [7], [13].

$$Precision\ (P) = \frac{TP}{TP + FP} \quad , \quad Recall\ (R) = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad , \quad F - score = 2 \times \frac{P \times R}{P + R}$$

Confusion Matrix	
TP	FN
FP	TN

$$IoU = \frac{Intersection\ of\ areas\ detected\ with\ ground\ truth}{Union\ of\ areas\ detected\ with\ ground\ truth}$$

CHAPTER 4 – RESULTS & DISCUSSION

This chapter will demonstrate and discuss the developed algorithms results. The details are shown in the following pages and the summary of results is shown below:

Table 9: Summary of algorithms results

Project No.	Task	Object	Dataset Size	No. of Classes	Precision	Recall	mAP	FPS
1	Detection	Construction Vehicles	2,825 Images 5,995 Objects	3	0.87	0.91	93.95 %	16.5
2	Detection	License Plates	2,000 Training 386 Testing	1	0.92	0.89	92.30%	16.7
3	Detection	Helmet	7,035 Images 26,424 Objects	2	0.91	0.93	94.90%	16.7 34.5
4	Detection	Face Mask	827 Images 3,900 Object	2	0.87	0.91	89.53%	16.7

Project No.	Task	Object	Dataset Size	No. of Classes	Precision	Recall	Accuracy
5	Classification	Concrete Cracks	40,000	2	0.99	0.99	93.95 %

4.1 Construction Vehicles Detection Algorithm

A YOLOv4 algorithm been developed to detect construction vehicles through training classifier and part of the feature extraction. The training duration took 15 hours 18 minutes till reaching 4,300 iterations with 6 hours left to complete the training. However, to avoid any training crashes or exceeding GPU limit usage in Google Collaboratory, the training was stopped. Results are demonstrated below:

Table 10: Construction vehicles detection algorithm testing results

Precision	Recall	F1-Score	IoU	mAP @ 0.5
0.87	0.91	0.89	72.69%	93.95%

Table 11: Algorithms average precision in detecting construction vehicles

Ref.	Algorithm	Concrete Mixer (AP)	Dump Truck (AP)	Excavator (AP)	mAP @ 0.5	FPS
[8]	Imp. F-RCNN	-----	-----	95.00 %	93.00 %	9.90
[9]	YOLOv3	94.90 %	83.30 %	93.50 %	90.57 %	**26.3**
[9]	SSD	90.80 %	71.20 %	85.40 %	82.47 %	20.8
[9]	Faster R-CNN	92.60 %	81.50 %	92.50 %	88.87 %	8.3
[9]	R-FCN	94.30 %	82.40 %	90.80 %	89.17 %	11.5
Ours	**YOLOv4**	**95.63 %**	**89.43 %**	**96.79 %**	**93.95 %**	16.5

Detection results @ 0.5 IoU		
Class	**TP**	**FP**
Concrete Mixer	260	16
Dump Truck	872	178
Excavator	792	85
Total FN		201

Scan the QR code to view a video of construction vehicles detection in Wadi Al Sail & East Sitra Housing Projects

SCAN ME

SCAN ME

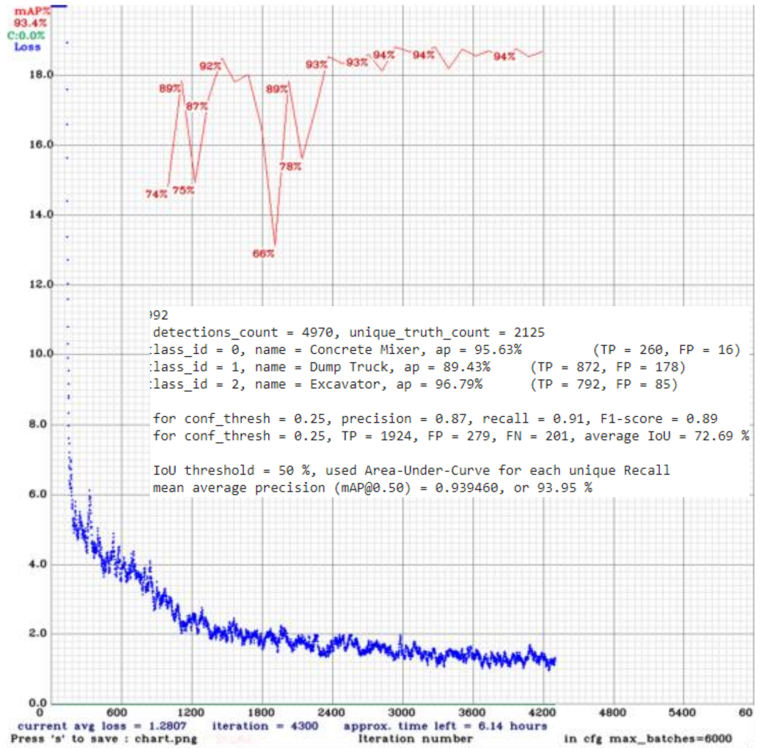

Figure 19: Learning curves of training loss and accuracy vs iteration

A. Concrete mixers images

Figure 20: Concrete mixer detection images

B. Dump trucks images

Figure 21: Dump trucks detection images (Above picture classified twice)

C. Excavator’s images

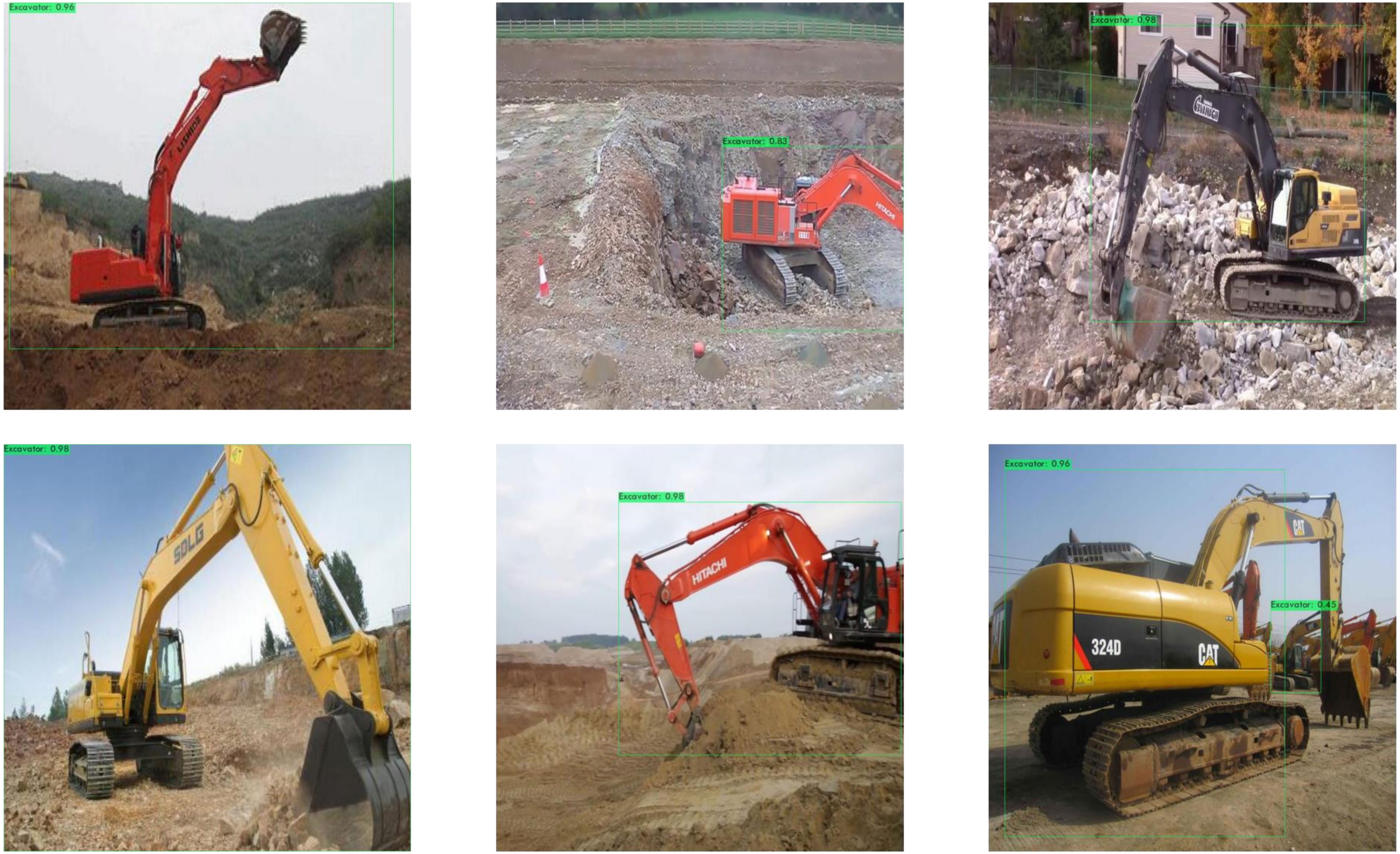

Figure 22: Excavator detection images

D. Multiple class images

Figure 23: Multiple class detection images

4.1.1 Construction Vehicles Detection Results Discussion

The YOLOv4 construction vehicles detection algorithm obtained high results scoring 93.95% mAP. The algorithm detection speed been tested on "2" different video captured in Wadi Al Sail & East Sitra housing projects. Both achieved 16.5 FPS. Although improved Faster R-CNN proposed by Fang, W. *et al.* [8] achieved 93% mAP result, almost similar to our algorithm, yet it suffers in FPS by scoring 9.9 FPS which is 40% less than our algorithm detection speed. This also shows that R-CNN algorithms suffer from low FPS against YOLO algorithm.

Our results been compared against several other algorithms results retrieved from literature which showed the superiority of our algorithm results in terms of mAP. However, in terms of FPS, our algorithm was ranked third. It should be highlighted that every author tested the FPS on different video.

Below is comparison of difference in percentage of results:

Table 12: Construction vehicles algorithms mAP performance difference comparison

Ref.	Algorithm	mAP @ 0.5	Percentage Improvement
[8]	Imp. F-RCNN	93.00 %	+ 1.02 %
[9]	YOLOv3	90.57 %	+ 3.73 %
[9]	SSD	82.47 %	+ 13.92 %
[9]	Faster R-CNN	88.87 %	+ 5.72 %
[9]	R-FCN	89.17 %	+ 5.36 %
Ours	**YOLOv4**	**93.95 %**	0 %

Table 13: Construction vehicles algorithms FPS performance difference comparison

Ref.	Algorithm	FPS	Percentage Difference
[8]	Imp. F-RCNN	9.90	+ 66.67 %
[9]	YOLOv3	**26.3**	-37.26 %
[9]	SSD	20.8	-20.67 %
[9]	Faster R-CNN	8.3	+ 98.80 %
[9]	R-FCN	11.5	+ 43.48 %
Ours	**YOLOv4**	16.5	0 %

4.2 License Plate Detection Algorithm

The license plate detection algorithm been developed on YOLOv4 through training a new classifier and part of the feature extraction. The training duration took 10 hours till reaching 2,060 iterations with 23 hours left to complete the training. However, to avoid any unforeseen crashes or exceeding GPU limit usage, the training was stopped. Results are demonstrated below:

Table 14: License plate detection algorithm testing results

Precision	Recall	F1-Score	IoU
0.92	0.89	0.90	74.13%

Table 15: Algorithms average precision in detecting license plate

Ref.	Algorithm	License Plate AP @ 0.5	FPS
Ours	YOLOv4	92.30%	16.7

Confusion Matrix

True Label	Class	TP	FP
	License Plate	454	41
	FN	58	

Scan the QR code to view a video of license plate detection captured in Nuwaidrat

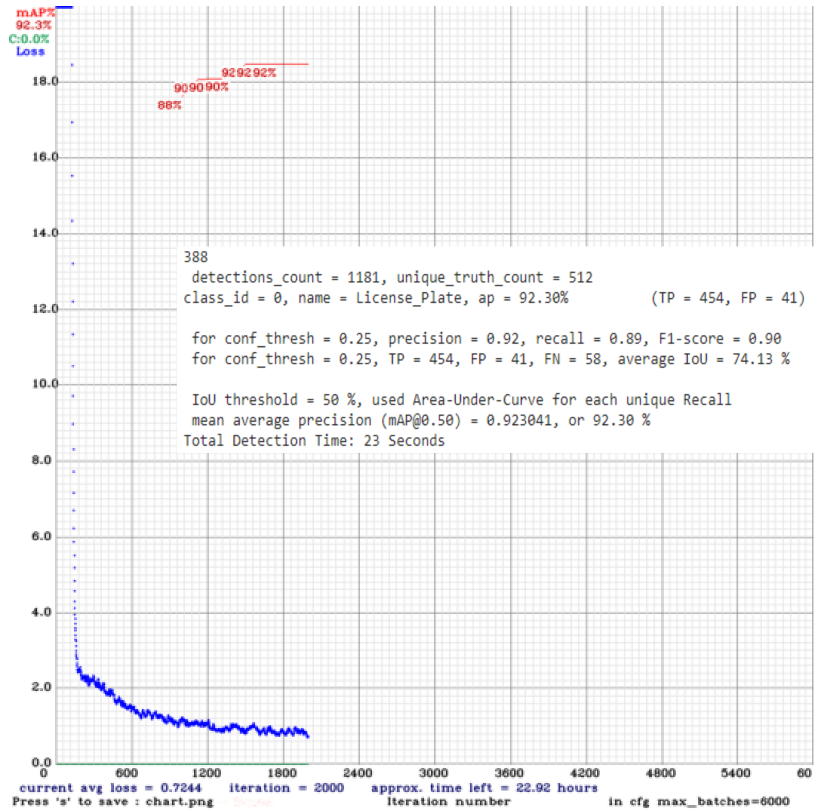

Figure 24: Learning curves of training loss and accuracy vs iteration

Figure 25: License plate detection images

4.3 Helmet Detection Algorithm

The helmet detection algorithm been developed on YOLOv4 through training a new classifier and part of the feature extraction. The training duration took 12 hours 16 minutes till reaching 2,372 iterations with 19 hours left to complete the training. However, to avoid any unforeseen crashes or exceeding GPU limit usage, the training was stopped. Results are demonstrated below:

Table 16: helmet detection algorithm testing results

Precision	Recall	F1-Score	IoU	mAP @ 0.5
0.91	0.93	0.92	73.13%	94.90%

Table 17: Algorithms average precision in detecting helmets

Ref.	Algorithm	Helmet (AP)	No Helmet (AP)	mAP @ 0.5	FPS
[10]	YOLO	82.00 %	89.10 %	85.60 %	-----
[10]	Faster R-CNN	85.30 %	88.10 %	86.20 %	-----
[10]	Imp. F-R-CNN	94.70 %	94.00 %	94.30 %	11.6
[4]	Faster R-CNN	-----	**94.90 %**	-----	4.95
Ours	YOLOv4	**95.61 %**	94.12 %	**94.90 %**	**34.5** **16.7**

Helmet Confusion Matrix

True Label	Helmet	No Helmet
Helmet	6010	640
No Helmet	751	2040

Predicted Label

SCAN ME

SCAN ME

Scan the QR code to view a video of helmet detection in Qalali & Tubli Housing Projects

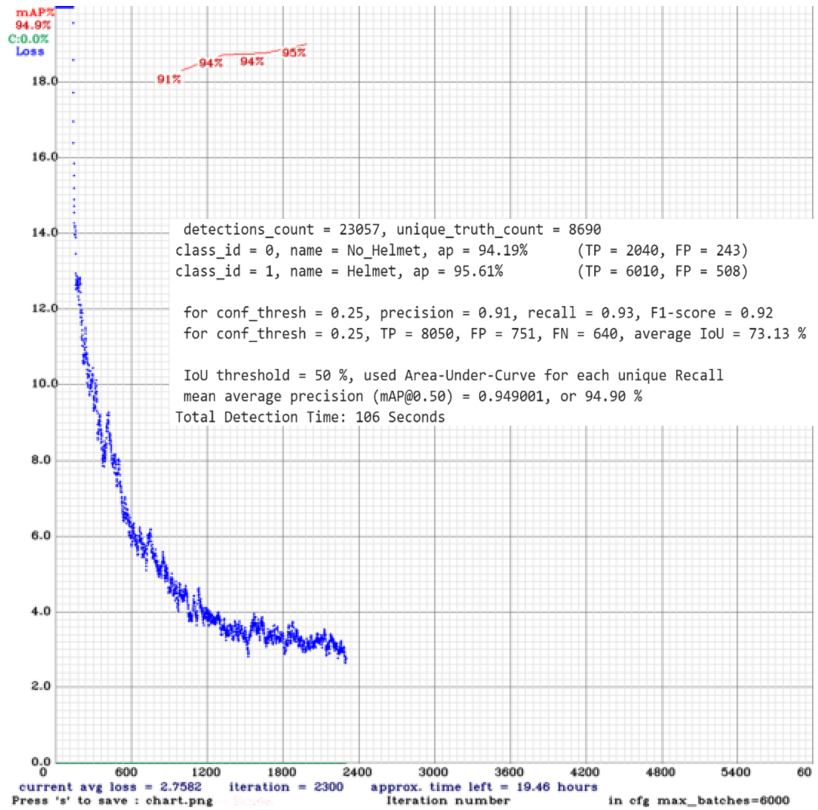

Figure 26: Learning curves of training loss and accuracy vs iteration

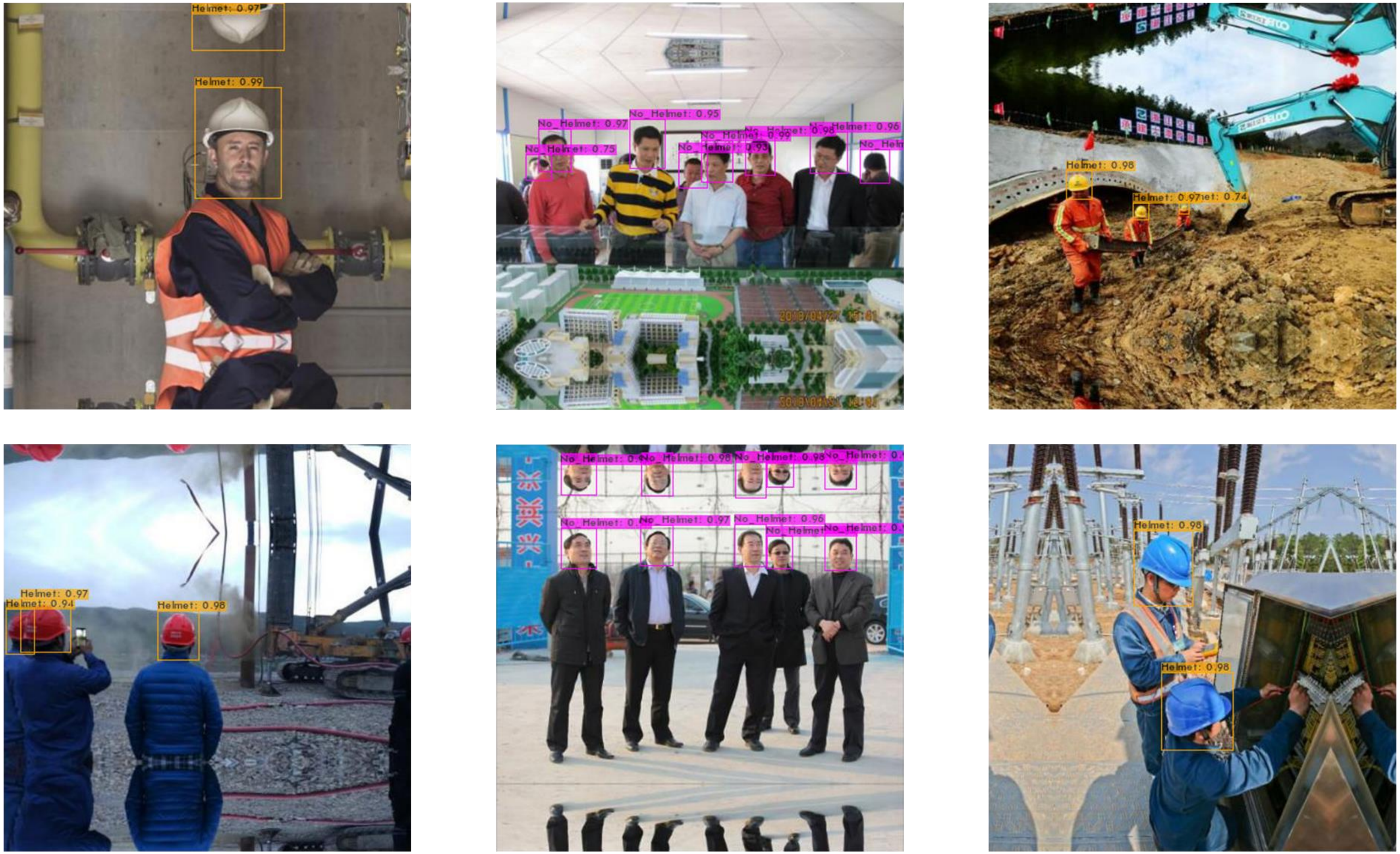

Figure 27: Helmet detection images

Figure 28: Site inspection (Housing Foundation) in Wadi Al Sail Housing Project

4.3.1 Helmet Detection Results Discussion

The YOLOv4 helmet detection algorithm obtained high results scoring 94.90% mAP. The algorithm detection speed been tested on "2" different video captured in Qalali & Tubli housing projects. It achieved 34.5 FPS and 16.7 FPS, respectively. It should be highlighted that it was noticed that not only the algorithm affects the detection speed, but also the video format and resolution. Our results been compared against several other algorithms results retrieved from literature which showed the superiority of our algorithm results in terms of both mAP and FPS. The reason behind achieving high results is the large dataset used as it consists of more than 7,000 images and more than 26,000 objects captured in different background and environments.

Furthermore, it should be highlighted that the algorithm was not able to detect "No Helmet" class if the person is wearing mask because it was trained to detect visible faces not wearing helmet and not masked faces not wearing helmet. Hence, it can be stated that the algorithm is relatively biased towards visible faces to determine the detection class. The reason behind this type of bias can be the unbalanced class images as the "No Helmet" class images amount was higher by 3 times than the "Helmet" class.

Below is comparison of difference in percentage of results:

Table 18: Helmet algorithms mAP performance difference comparison

Ref.	Algorithm	mAP @ 0.5	Percentage Improvement
[10]	YOLO	85.60 %	+ 10.86 %
[10]	Faster R-CNN	86.20 %	+ 10.09 %
[10]	Imp. F-R-CNN	94.30 %	+ 0.64 %
[4]	Faster R-CNN	-----	-----
Ours	YOLOv4	**94.90 %**	0 %

Table 19: Helmet algorithms FPS performance difference comparison

Ref.	Algorithm	FPS	Percentage Improvement
[10]	YOLO	-----	-----
[10]	Faster R-CNN	-----	-----
[10]	Imp. F-R-CNN	11.6	+ 197.41 %
[4]	Faster R-CNN	4.95	+ 596.97 %
Ours	YOLOv4	**34.5**	0 %

4.4 Face Mask Detection Algorithm

The face mask detection algorithm developed was done on YOLOv4 through training a new classifier and part of the feature extraction. The training duration took 11 hours 43 minutes till reaching 2,200 iterations with 25 hours left to complete the training. However, to avoid any unforeseen crashes or exceeding GPU limit usage, the training was stopped. Results are demonstrated below:

Table 20: Face mask detection algorithm testing results

Precision	Recall	F1-Score	IoU	mAP @ 0.5
0.87	0.91	0.89	70.22%	89.53%

Table 21: Algorithms average precision in detecting face masks

Ref.	Algorithm	Mask (AP)	No Mask (AP)	mAP @ 0.5	FPS
[11]	SSD	-----	-----	92.60%	15.71
[12]	SSD	98.60%	94.10%	97.20%	34.69
[12]	Faster R-CNN	97.40%	94.30%	95.60%	2.44
[12]	YOLOv3	98.10%	92.10%	94.80%	21.39
[12]	YOLOv4	96.90%	94.30%	95.20%	23.83
[12]	Imp. YOLOv4	**99.50%**	**97.90%**	**98.30%**	**54.57**
Ours	YOLOv4	93.38%	85.67%	89.53%	16.7

Face Mask Confusion Matrix

True Label	Mask (Predicted Label)	No Mask (Predicted Label)
Mask	735	85
No Mask	131	152

SCAN ME

Scan the QR code to view a video of face mask detection in Qalali Housing Project

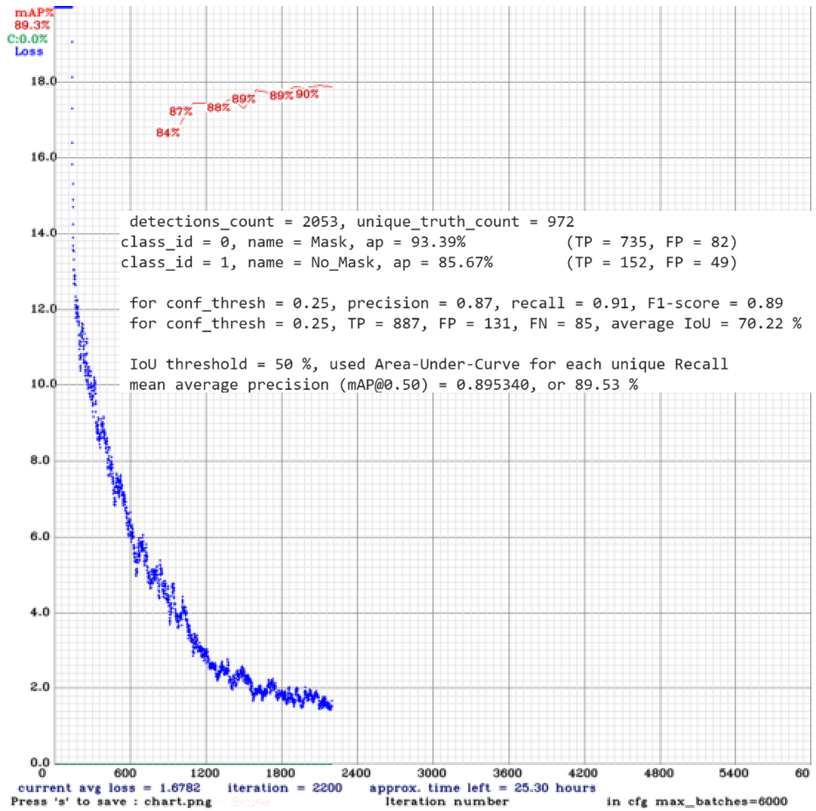

Figure 29: Learning curves of training loss and accuracy vs iteration

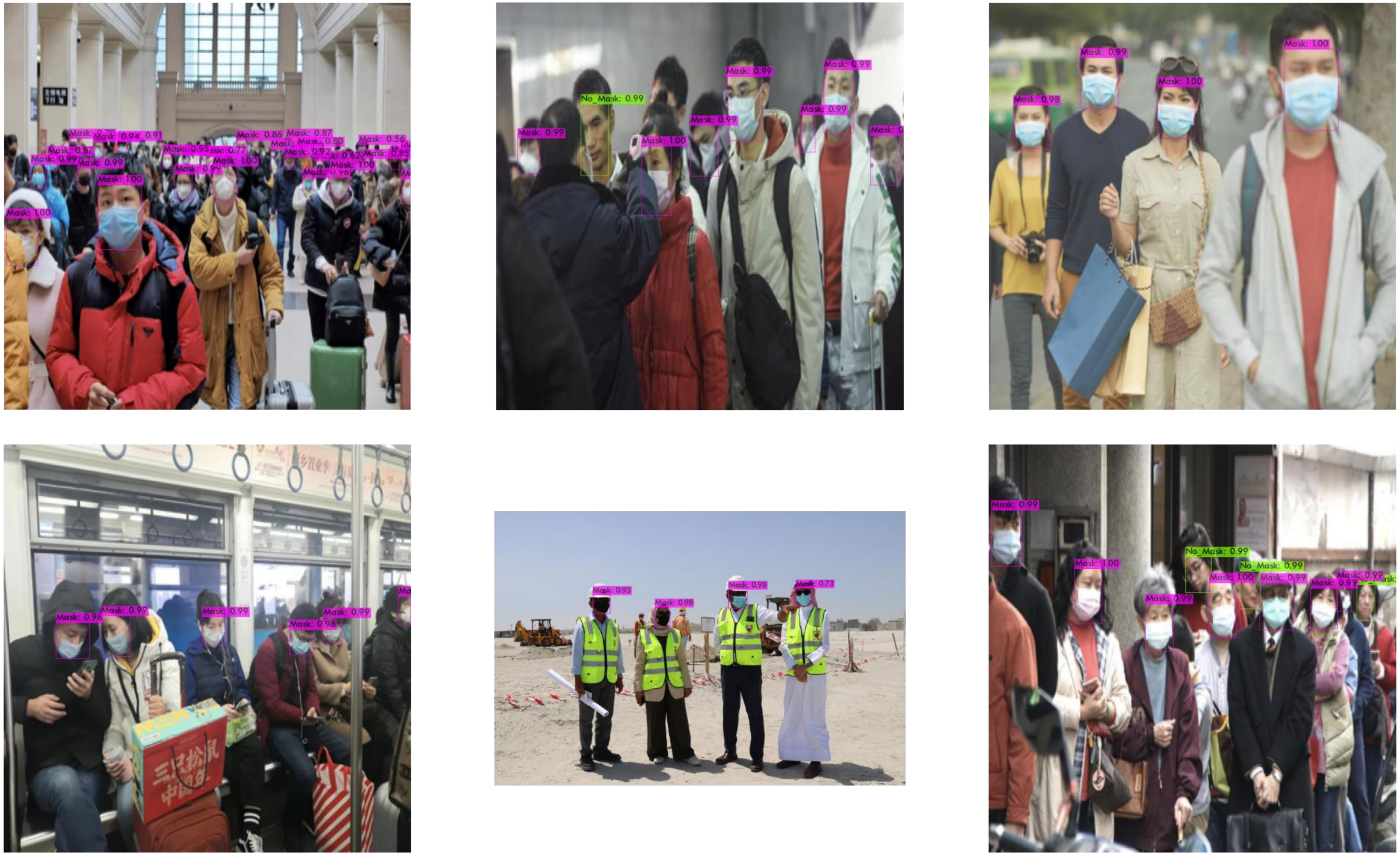

Figure 30: Face mask detection images

Figure 31: H.E. Minister of Housing visit to East Sitra Housing Project

4.4.1 Face Mask Detection Results Discussion

The YOLOv4 face mask detection algorithm obtained relatively high results scoring 89.53% mAP at 16.7 FPS. Our results were the lowest against several other algorithms results retrieved from literature due to the following reasons:

1. The literature algorithms been trained on different datasets that are larger than our dataset reaching to 13 times larger as shown in chapter 2.
2. Our dataset images are more limited to similar environment. i.e., they are less variant compared to other datasets that were collected from various environments and backgrounds.

Yet, the difference is not major. By comparing the mean average precision results of our algorithm, SSD algorithm of Nagrath, P. *et al.* [11], and Faster R-CNN algorithm of Yu, J. *et al.* [12], the high mAP results the authors have obtained were on the price of lower FPS, as our algorithm achieved higher FPS with lower mAP. In fact, this finding proves that Faster R-CNN achieves high mAP and low FPS results when compared to YOLO algorithm. Below is comparison of difference in percentage of results:

Table 22: Face mask algorithms mAP performance difference comparison

Ref.	Algorithm	mAP @ 0.5	Percentage Difference
[11]	SSD	92.60%	- 3.32 %
[12]	SSD	97.20%	- 7.89 %
[12]	Faster R-CNN	95.60%	- 6.35 %
[12]	YOLOv3	94.80%	- 5.56 %
[12]	YOLOv4	95.20%	- 5.96 %
[12]	Improved YOLOv4	**98.30%**	- 8.92 %
Ours	YOLOv4	89.53%	0 %

Table 23: Face mask algorithms FPS performance difference comparison

Ref.	Algorithm	FPS	Percentage Difference
[11]	SSD	15.71	+ 6.30 %
[12]	SSD	34.69	- 51.86 %
[12]	Faster R-CNN	2.44	+ 584.43 %
[12]	YOLOv3	21.39	- 21.93 %
[12]	YOLOv4	23.83	- 29.92 %
[12]	Improved YOLOv4	**54.57**	- 69.40 %
Ours	YOLOv4	16.7	0 %

4.5 Concrete Cracks Classification

Crack classification results were obtained from 8 model configurations. Each model developed had an additional feature that the previous model did not have. The model configurations and results obtained are as follows:

Table 24: Concrete cracks classification algorithm results

Model No.	Convolutional Layers	Max Pooling	FCN	Optimizer	Dropout	Batch Normalization	Decay	Transfer Learning	Backbone	Training Accuracy	Testing Accuracy
1	1	1	1	Adam						97.35%	92.56%
2	2	2	2		✓					98.65%	95.72%
3	2	2	2		✓	✓				98.20%	55.16%
4	2	2	2		✓	✓	✓			97.90%	82.81%
5	4	4	2							98.95%	98.21%
6	**✓**	**✓**	**2**					**✓**	**VGG16**	**100.00%**	**98.90%**
7	✓	✓	2					✓	ResNet50	99.80%	94.95%
8	✓	✓	2					✓	ResNet101	99.75%	95.88%

Crack Images Results				
Model No.	**Precision**	**Recall**	**F-Score**	**Accuracy**
1	0.90	0.96	0.93	0.93
2	1.00	0.92	0.96	0.96
3	1.00	0.10	0.19	0.55
4	1.00	0.66	0.79	0.83
5	1.00	0.97	0.98	0.98
6	**0.98**	**1.00**	**0.99**	**0.99**
7	0.91	1.00	0.95	0.95
8	0.92	1.00	0.96	0.96

Uncracked Images Results				
Model No.	**Precision**	**Recall**	**F-Score**	**Accuracy**
1	0.95	0.90	0.92	0.93
2	0.92	1.00	0.96	0.96
3	0.53	1.00	0.69	0.55
4	0.74	1.00	0.85	0.83
5	0.97	1.00	0.98	0.98
6	**1.00**	**0.98**	**0.99**	**0.99**
7	1.00	0.90	0.95	0.95
8	1.00	0.92	0.96	0.96

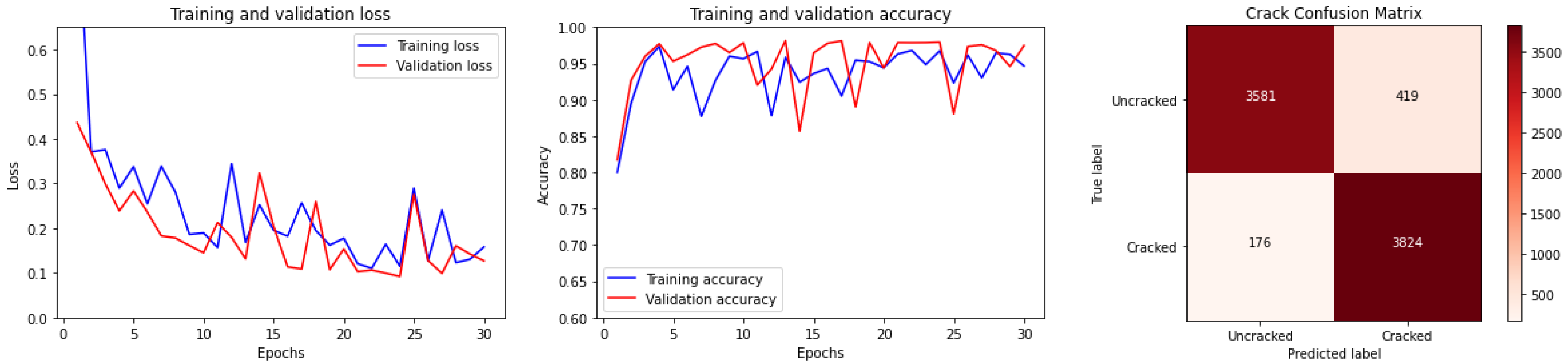

Figure 32: Model No. 1 learning curves and confusion matrix

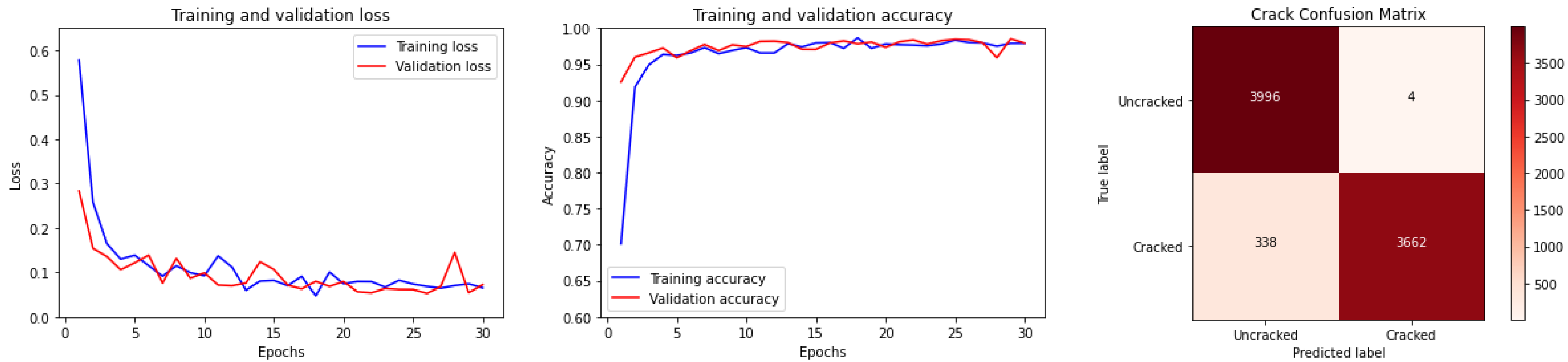

Figure 33: Model No. 2 learning curves and confusion matrix

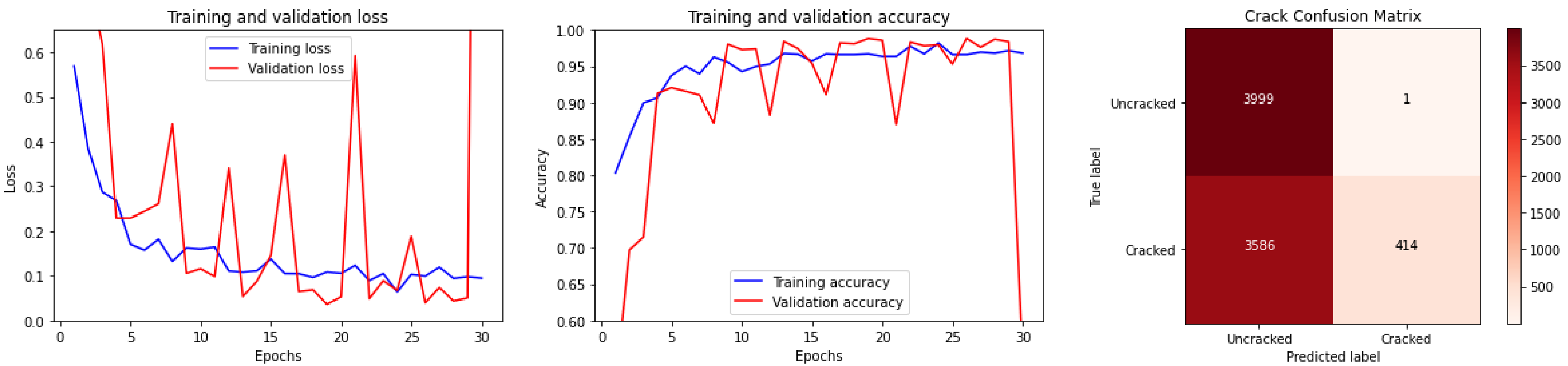

Figure 35: Model No. 3 learning curves and confusion matrix (Worst model)

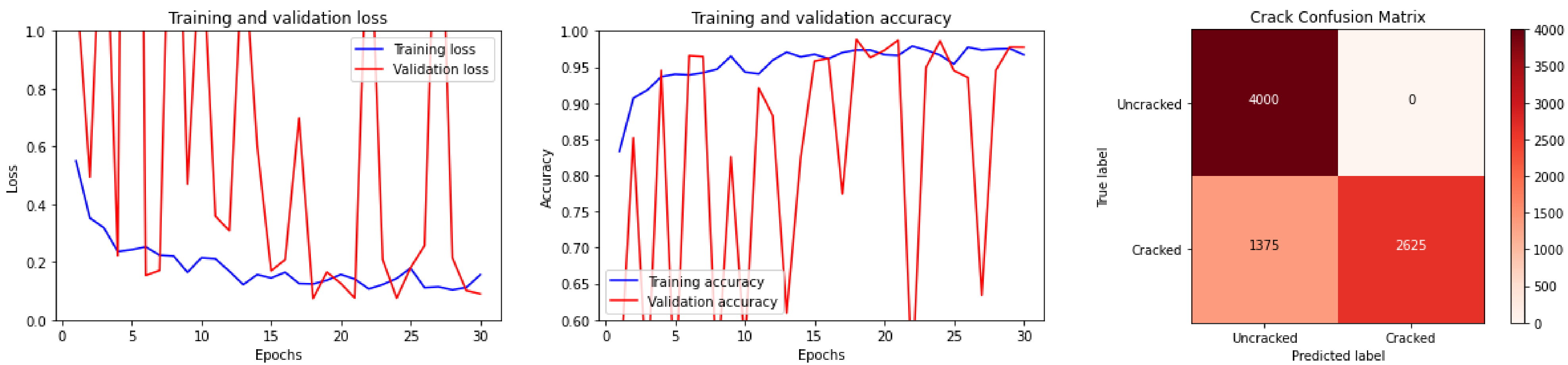

Figure 35: Model No. 4 learning curves and confusion matrix

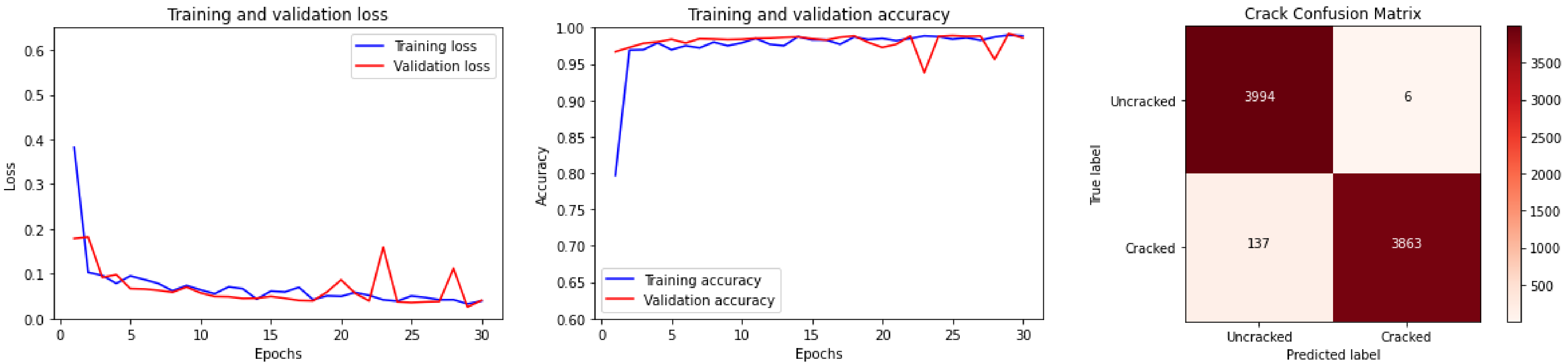

Figure 37: Model No. 5 learning curves and confusion matrix

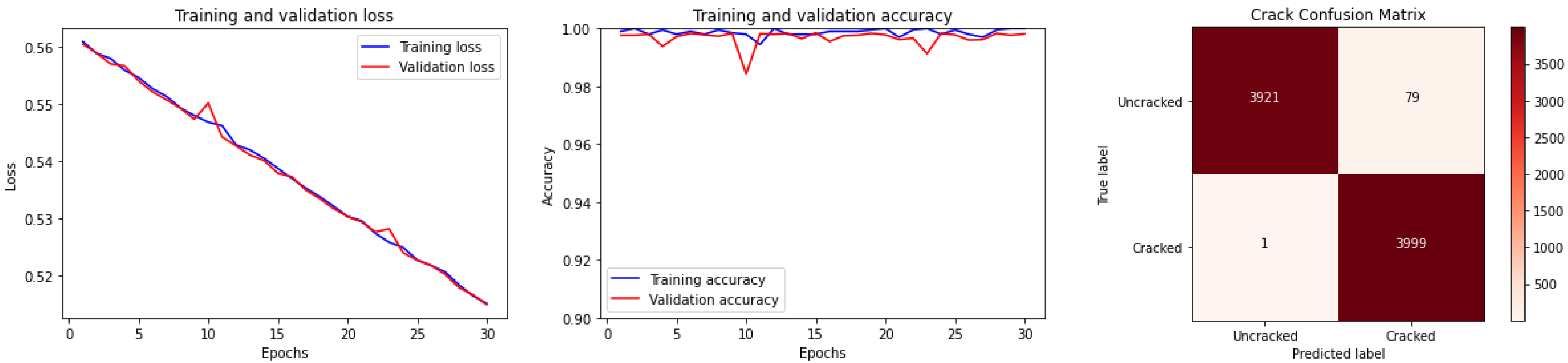

Figure 37: Model No. 6 learning curves and confusion matrix (Best model)

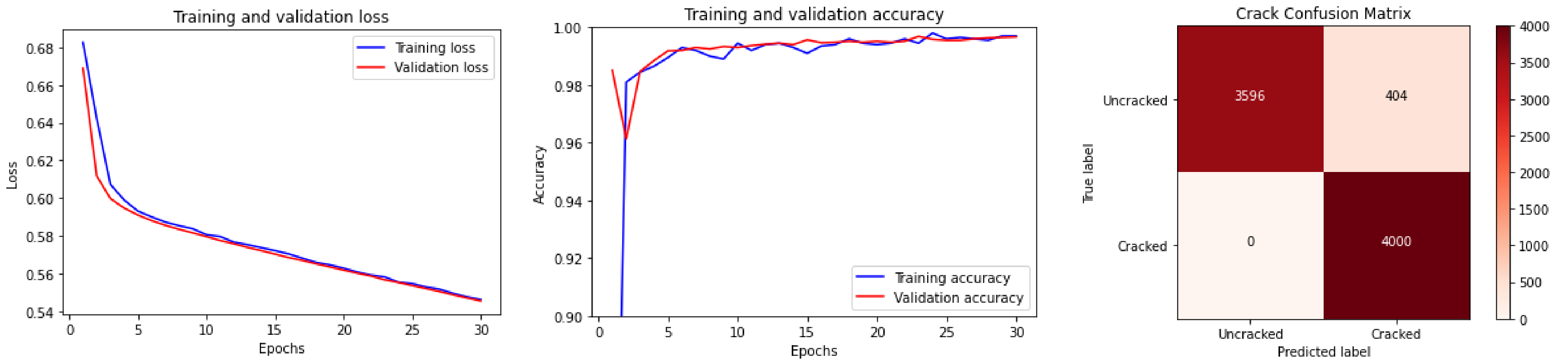

Figure 38: Model No. 7 learning curves and confusion matrix

Figure 40: Model No. 8 learning curves and confusion matrix

The best model achieved 98.90% accuracy in testing and 100% accuracy in training. It utilized the transfer learning from a VGG16 model as shown in figure 38. The only change done is that the classifier part (2 FCN) and 3 feature extraction layers been removed and replaced with 2 FCN layers having 512 neurons & 2 neurons, respectively. On the other hand, the worst model surprisingly achieved 55.16% in testing and 98.20% in training. The low results cause seems to be due to introducing normalization. As this trend was also noticed in model no.4. However, the other models did not face this issue as normalization was not included.

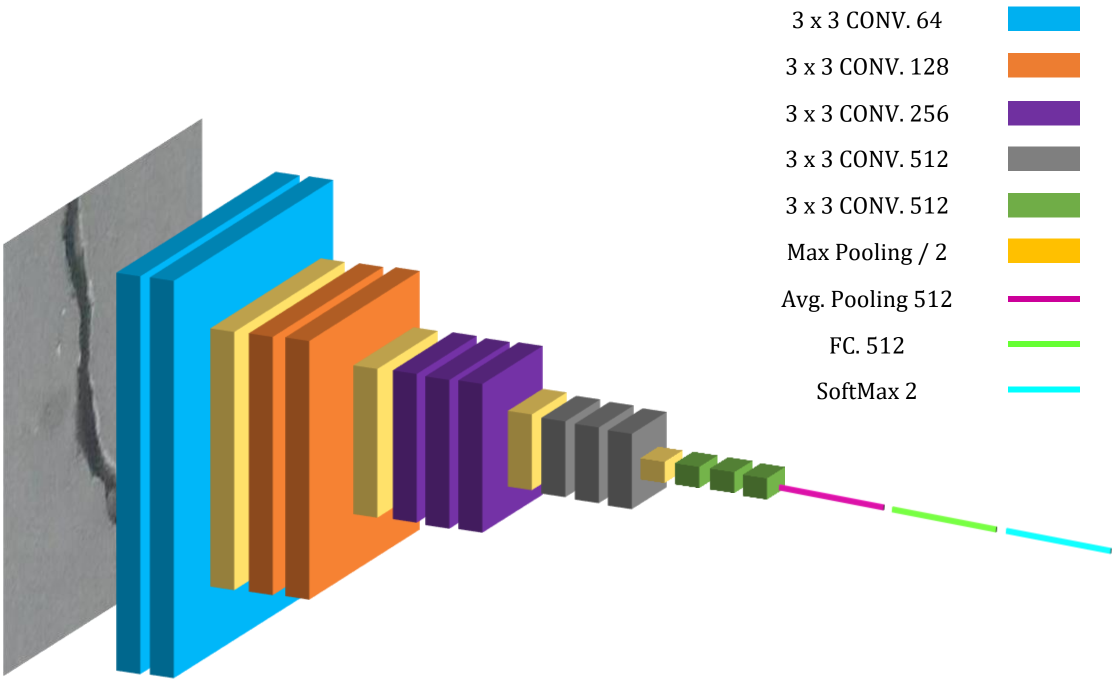

Figure 39: Model 6 Classification algorithm architecture

CHAPTER 5 - CONCLUSION

The project briefly presented the construction industry status in Bahrain and demonstrated the benefits that can be realized by integrating deep learning in construction sites. It also, discussed the importance of maintaining high level of health & safety precautions in construction sites. On the other hand, the project illustrated the current practice and procedures being followed in Ministry of Housing for construction sites access control, health, and safety quality control. The project proposed object detection YOLO based algorithms to automate these tasks instead of deploying human H&S officer, in addition to CNN based algorithm for crack classification.

YOLOv4 object detection algorithms been developed to exploit deep learning capabilities and demonstrate the possibilities that can be realized of integrating intelligence in construction industry. The first algorithms been developed for construction vehicle detection achieved 93.95% mAP at 16.5 FPS. The second algorithm developed for license plate detection achieved 92.30% mAP at 16.7 FPS. The third algorithm developed for helmet detection achieved 94.90% mAP at 16.7 – 34.5 FPS. The fourth algorithm developed for face mask detection achieved 89.53% mAP at 16.7 FPS. While the fifth algorithm been developed for classifying concrete crack images. It achieved 98.90% accuracy and F-score results.

The developed applications proved that utilization of deep learning in construction sites can enhance site management and reduce efforts of day-to-day activities. The labors adherence to health and safety protocol can be evaluated intelligently and simultaneously monitored on a wide scale area covering the whole project without the need to deploy health & safety officers on daily bases at every construction site. Similarly, the security personal can be replaced with intelligent license plate detection algorithm to automate the gate access control. All these merits will be realized with savings in cost, time, efforts, reducing manpower requirement, improving safety, and productivity.

The future research can be done in integrating and deploying the developed algorithms on camera devices distributed over construction sites. In addition to increase the classes detection to include other construction vehicles along with integrating it to a report generation system to produce daily reports automatically showing the number of manpower and machinery attended site.

REFERENCES

[1] Ministry of Finance & National Economy, "Bahrain Economic Quarterly Report," *Ministry of Finance & National Economy*, April 2021. [Online]. Available: https://cutt.ly/xnsJvpX [Accessed 02 May 2021].

[2] A. A. Saeed, "3 Billion Dollars King Hamad Causeway Construction Cost," *Al Ayam Newspaper*, 14 September 2018. [Online]. Available: https://cutt.ly/xnsH6Tl [Accessed 01 May 2021].

[3] Dji Enterprise, "Aerial Surveing and AEC," *Dji Enterprise*, 2021. [Online]. Available: https://enterprise.dji.com/surveying [Accessed 10 May 2021].

[4] Fang, Q., Li, H., Luo, X., Ding, L., Luo, H., Rose, T. M., & An, W., "Detecting non-hardhat-use by a deep learning method from far-field surveillance videos," *Elsevier Automation in Construction,* no. 85, pp. 1-8, 2018.

[5] Fang, W., Ding, L., Luo, H., & Love, P. E., "Falls from heights: A computer vision-based approach for safety harness detection," *Elsevier Automation in Construction,* vol. 91, pp. 1-8, 2018.

[6] Social Insurance Organization, "Statistical Report - 2020 Quarter 4," *Social Insurance Organization*, Manama, 2020. [Online]. Available: https://cutt.ly/MnsJOp6 [Accessed 02 May 2021].

[7] Xiang, X., Lv, N., Guo, X., Wang, S., & El Saddik, A., "Engineering vehicles detection based on modified faster R-CNN for power grid surveillance," *MDPI Sensors,* vol. 18, no. 7, pp. 1-13, 2018.

[8] Fang, W., Ding, L., Zhong, B., Love, P. E., & Luo, H., "Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach," *Elsevier Advanced Engineering Informatics,* vol. 37, pp. 1-4, p. 6, pp. 8-10, 2018.

[9] Xiao, B., & Kang, S. C., "Development of an Image Data Set of Construction Machines for Deep Learning Object Detection.," *ASCE Journal of Computing in Civil Engineering,* vol. 35, no. 2, pp. 1-16, 2021.

[10] Chen, S., Tang, W., Ji, T., Zhu, H., Ouyang, Y., & Wang, W., "Detection of Safety Helmet Wearing Based on Improved Faster R-CNN," in *IEEE 2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-2, pp. 5-6, 2020.

[11] Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J., "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Elsevier Sustainable cities and society,* no. 66, pp. 1-5, pp.7-10, 2021.

[12] Yu, J., & Zhang, W., "Face mask wearing detection algorithm based on improved YOLO-v4," *MDPI Sensors,* vol. 21, no. 9, pp. 1-2, p. 11, pp. 13-14, pp. 16-19, 2021.

[13] M. Elgendy, CNN architecture, New York: Manning Publications Co., pp, 240-337 2020.

[14] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A., "You only look once: Unified, real-time object detection," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.1-8, 2016.

[15] Redmon, J., & Farhadi, A., "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.1-8, 2017.

[16] Redmon, J., & Farhadi, A., "Yolov3: An incremental improvement," *arXiv,* pp. 1-5, 2018.

[17] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M., "YOLOv4: Optimal Speed and Accuracy of Object Detection," *Computer Vision and Pattern Recognition,* pp. 1-11, 2020.

[18] Du, Y., Pan, N., Xu, Z., Deng, F., Shen, Y., & Kang, H., "Pavement distress detection and classification based on YOLO network," *International Journal of Pavement Engineering,* pp. 1-7, pp. 11-13, 2020.

[19] Silva, L. A., Sanchez San Blas, H., Peral García, D. P., Sales Mendes, A. S., & Villarubia González, G. V., "An Architectural Multi-Agent System for a Pavement Monitoring System with Pothole Recognition in UAV Images," *MDPI Sensors,* vol. 20, no. 21, pp. 1-6, pp. 9-18, 2020.

[20] Majidifard, H., Jin, P., Adu-Gyamfi, Y., & Buttlar, W. G., "Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses.," *Transportation Research Record*, vol. 2674, no. 2, pp. 1-9, 2020.

[21] Nie, M., & Wang, C., "Pavement Crack Detection based on yolo v3," in *IEEE 2nd International Conference on Safety Produce Informatization (IICSPI)*, pp. 1-3, 2019.