A Performance Study of Different Deep Learning Architectures

For Detecting Construction Equipment in Sites

by

Mohamed Mamdooh Sabek

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved March 2022 by the
Graduate Supervisory Committee:

Kristen Parrish, Chair
Thomas Czerniawski
Steven K. Ayer

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

There are relatively few available construction equipment detectors models that use deep learning architectures; many of these use old object detection architectures like CNN (Convolutional Neural Networks), RCNN (Region-Based Convolutional Neural Network), and early versions of You Only Look Once (YOLO) V1. It can be challenging to deploy these models in practice for tracking construction equipment while working on site.

This thesis aims to provide a clear guide on how to train and evaluate the performance of different deep learning architecture models to detect different kinds of construction equipment on-site using two You Only Look Once (YOLO) architectures-YOLO v5s and YOLO R to detect three classes of different construction equipment on-site, including Excavators, Dump Trucks, and Loaders. The thesis also provides a simple solution to deploy the trained models. Additionally, this thesis describes a specialized, high-quality dataset with three thousand pictures created to train these models on real data by considering a typical worksite scene, various motions, varying perspectives, and angles of construction equipment on the site.

The results presented herein show that after 150 epochs of training, the YOLOR-P6 has the best mAP at 0.981, while the YOLO v5s mAP is 0.936. However, YOLO v5s had the fastest and the shortest training time on Tesla P100 GPU as a processing unit on the Google Colab notebook. The YOLOv5s needed 4 hours and 52 minutes, but the YOLOR-P6 needed 14 hours and 35 minutes to finish the training.

The final findings of this study show that the YOLOv5s model is the most efficient model to use when building an artificial intelligence model to detect construction equipment because of the size of its weights file relative to other versions of YOLO models- 14.4 MB for YOLOV5s vs. 288 MB for YOLOR-P6.

This hugely impacts the processing unit's performance, which is used to predict the construction equipment on site. In addition, the constructed database is published on a public dataset on the Roboflow platform, which can be used later as a foundation for future research and improvement for the newer deep learning architectures.

ACKNOWLEDGMENTS

This thesis would not be a reality without the kind support of my professor, Dr. Kristen Parrish, who supported me all the way to reach this point. Further, I am thankful to have been part of Dr. Thomas Czerniawski's internship program, which introduced me to the world of artificial intelligence applications and construction. Both of them gave me a chance, and they pushed me forward to explore the wonderful world of artificial intelligence and how its application can impact our daily lives. Because of that, I was able to deliver this thesis, which means a lot to me as a person and as an engineer at the same time. I am also indebted to Dr. Steven Ayer for his service on my thesis committee.

Furthermore, I would like to thank my beloved family for the huge support that I got from them. They were the most prominent supporters for me and those that helped me accomplish many goals in my life.

Finally, thank you very much to my loved one, Doaa, who was the most supportive person on this journey, and I thank God for having her in my life.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST of FIGURES

GLOSSARY

Artificial intelligence (AI): is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by animals, including humans. ((Legg, and Hutter.2007)

Deep learning: is a class of machine learning algorithms thatuses multiple layers to extract higher-level features from the raw input progressively. (Deng, and Yu.2014)

Computer vision: is an interdisciplinary field that deals with how computers can be made to gain high-level understanding from digital images or videos to automate tasks that the human visual system can do.

Tensor Flow: is a free and open-source software library for machine learning and artificial intelligence. (Janane and Jeyanthi., 2019).

YOLO: You Only Look Once is an algorithm that utilizes a single convolutional network for object detection. (YOLO. Medium, 2022).

CNN: convolutional neural network (CNN/ConvNet) is a class of deep neural networks that uses a unique technique called Convolution (Mandal,2022).

RCNN: Region-Based Convolutional Neural Network (R-CNN) is a machine learning model for computer vision applications and expressly object detection, which combines rectangular region proposals with convolutional neural network features (Mathworks,2022).

Object detection: is a computer vision technology related to computer vision and image processing that deals with detecting instances of semantic objects of a particular class in digital images and videos (Dasiopoulou et al., 2005)

GPU: A graphics processing unit is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. ("What Is A GPU", Intel).

FPS: is the total number of consecutive full-screen images displayed on the screen for each second; it is an important parameter as a performance metric for computer visions models.

SVM: Support Vector Machines is a machine learning algorithm that analyzes data for classification and regression analysis (What Is SVM, Techopedia,2022).

mAP: (mean Average precision) is a popular metric in measuring the accuracy of object detectors like Faster R-CNN, YOLO which computes the average precision value for recall value over 0 to 1. (Hui, 2022).

Precision: is a measure of how precise a model is at prediction time, which reflects the true positives divided by all positives that have been detected. ("A Guide for Model Production - Roboflow")

Recall: A measure of performance for a prediction system. The recall is used to assess whether a prediction system is guessing enough and representing the true positives, divided by all possible true positives ("A Guide for Model Production - Roboflow").

API: An application programming interface (API) is a connection between computers or between computer programs. It is a type of software interface offering a service to other pieces of software, and in deep learning applications, it provides the detections from the model neural network weights file (Reddy,2011).

Keywords: Construction Equipment, Deep Sort, Deep Learning, You Only Look Once

(YOLO), Image Dataset, Real-Time Detection

1. INTRODUCTION

Construction sites can be dangerous places because of their noisy and often congested nature. Indeed, such sites are characterized by various kinds of construction equipment and construction workers all over the place, increasing the risk that an accident could happen without anyone taking immediate notice.

Some of the most significant accidents that can happen on construction sites are construction equipment colliding with each other or with construction personnel on-site, leading in extreme cases to casualties and financial loss.

1.1. Safety in construction sites

According to research published in the Journal of Safety Research by Michael McCann, backhoe and significant truck accidents accounted for half of all deaths in construction fatal accidents by heavy equipment. Construction trucks clashed with other vehicles or objects, flipped over, or went off the road in these types of incidents. According to the international labor organization, it is estimated that around 60,000 fatal accidents happened in construction sites all over the globe in 2005, which accounted for approximately one-sixth of fatality rates globally in that year ("Facts on Safety at Work"). Unfortunately, even nowadays, fatality rates continue to plague the construction industry. According to OSHA, in 2020, the United States of America had more than 5,333 on-the-job worker deaths. Around 21% of these deaths in private industry were in the construction industry with a total of 1,008 deaths; this is why one of the highest fatality rates across sectors.

For example, the manufacturing industry only had 340 deaths in 2020 all over the United States of America, a much lower fatality rate than the construction industry, which has four times the number of fatalities between construction workers. ("Worker Fatalities, OSHA").

Figure 1 shows that the construction industry has the highest number of fatality rates among all the different industries all over the United States of America in 2020, which was and still is a constant scenario for the last decade in the construction industry, which should be improved to avoid such significant fatalities.

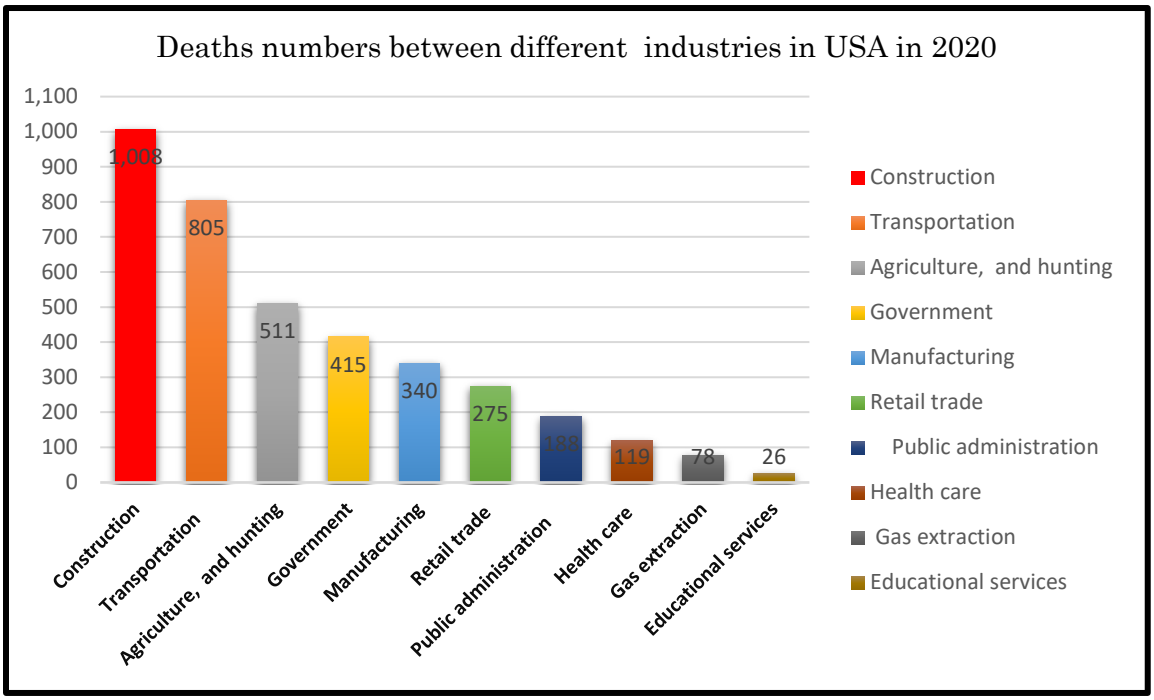

Figure 1-Deaths numbers between different industries in the USA in 2020- OSHA

1.2. Safety applications of object detection in the construction industry

Accidents and fatalities in construction sites could be reduced or eliminated if intelligent monitoring systems were implemented, which can, in turn, reduce the fatality rates in the industry. Indeed, by implementing an early alerting system that can monitor the whole site, including different people and different construction equipment across the site over time, with limited, if any, need for human supervision or interaction.

These monitoring systems can be successfully implemented thanks to newly developed deep learning architectures and artificial intelligence algorithms that can be used to train deep learning models to detect various kinds of construction equipment and construction workers on-site, predict accidents that could happen, and alert the site supervisors about the potential accidents before they happen. By doing so, the fatality rates could be reduced dramatically theoretically, if it was adopted by the actual workers working on the sites themselves.

The systems can be implemented using modern artificial intelligence algorithms and various kinds of deep learning architectures. Deep learning architectures significantly improved in recent years. With that being said, one of the main pillars of modern computer vision applications is Object Detection /Recognition (Ker et al. 2018), which was very technically complicated and demanded processing resources that required quite of a bit of time to complete (Voulodimos et al.2018); even today, Object Detection/Recognition is still a very demanding task for any graphics processing unit or GPU.

However, with the advancement of neural networks in the last decade, Object Detection has become available to anyone in any sector, which is possible because of the newly developed deep learning architectures like Tensor Flow Lite and yolov5s tiny (Gothane 2021). Moreover, advancements have reduced model sizes which, in turn, requires less processing power; thus, the newer deep learning algorithms make it suitable for small processing units on portable devices (e.g., smartphones).

Finally, this learning of the deep learning model sizes has resulted in new technological developments, i.e., the newly developed and released OAK-D, an intelligent 3D camera with neural inference and depth processing capabilities that is capable of Spatial data processing through different AI trained models (Rojas-Perez, and Martinez-Carranza,2021).

Object Recognition is one of the most critical sectors of artificial intelligence that has become pervasive in modern life; it is responsible for products ranging from Snapchat filters to MRI analysis to identify cancerous tumors (Noronha et al., 2021).

Despite the ubiquity of Object Detection and Recognition, the construction industry has yet to fully adapt to computer vision applications, especially Object Detection/Recognition. Perhaps this is due to the lack of available trained artificial intelligence models that can be implemented into the daily functions of construction personnel (Oprach et al., 2019). Further, construction equipment is an object that has insufficient representation in modern artificial intelligence detection models and databases that use deep learning architectures, which may also have limited the application of computer vision in construction.

1.3. Construction Datasets for Object Detection and Object Recognition

General models and datasets like COCO (Lin TY. et al.2014) are sophisticated enough to identify different things and items in our daily lives, including people, animals, fruit, vegetables, clothing, and blood components; this dataset keeps expanding.

Conversely, when it comes to construction equipment, there are very few publicly available deep learning models and datasets that are accurate (i.e., have strong detecting performance per frame) and available for research and practical uses. However, as the construction industry continues to move in the direction of adopting artificial intelligence tools and techniques, these datasets will likely continue to expand (Abioye et al., 2021).

1.4. Purpose of Research

The purpose of this thesis is to build an artificial intelligence model to detect three different kinds of construction equipment on-site and provide a clear performance comparison between two different state-of-the-art deep learning architectures models, YOLO V5s and YOLOR-P6, both of which are available publicly. This thesis explicitly explores how these architectures detect excavators, wheel loaders, and dump trucks and compare two different YOLO deep learning architecture iterations.

1.5. Research Objectives/Goals

This thesis seeks to achieve the following objectives:

- Provide clear steps for building a deep-learning artificial intelligence model, starting from preparing the training data, annotating that data, training the model on the data, and finally deploying the model online and getting live predictions from the trained model using APIs to detect specific construction equipment on site.
- Create a well-functioning model to detect excavators, wheel loaders, and dump trucks quickly and with high accuracy with an efficient model size to minimize the processing power requirement.
- Create a publicly available training dataset with 3,000 pictures of excavators, wheel loaders, and dump trucks, with various angles and scenarios as a training base for various deep learning models.
- Compare the performance of YOLOV5s and YOLOR-P6 computer vision architecture when addressing the created training datasets in terms of how each platform architecture performs with respect to the accuracy, processing time, and file size to elucidate the relative strengths and weaknesses of each architecture.

2. Research Methodology

This section describes the complete development and assessment process for the training models considered and alternative approaches for completing these tasks so that this thesis can serve as a guide for future researchers. This describes how the author gathered data, annotated it, applied prepossessing algorithms to these data, and the different augmentation processes applied to the data to make the data more diverse for the training purpose of the model. This section also describes how the author expanded the training data size without acquiring a new dataset by implementing augmentations of the existing training data, training the model, evaluating the training results and predictions from the trained model, and preparing the model for deployment in various environments.

Figure 2 presents the development process used in this thesis. While specific elements of this development process can be different according to the workflow of the model's training, the basic structure presented is "typical" for building a computer vision model.

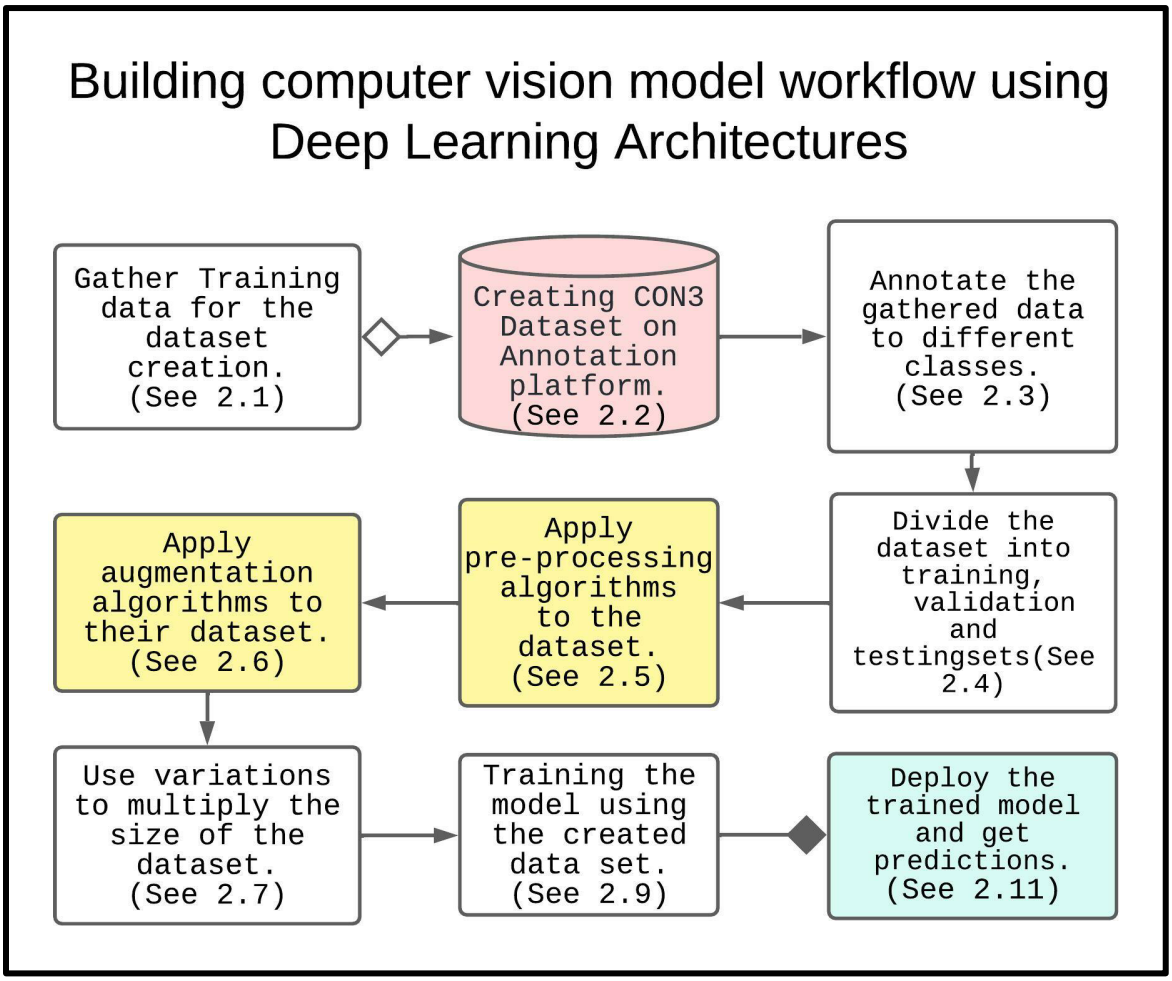

Figure 2-Building computer vision model workflow using deep learning architectures.

2.1. Gather Training data for the dataset creation

In order to train deep learning artificial intelligence models to detect various kinds of construction equipment with high accuracy and precision, it is essential to have a good database of images for these kinds of equipment in different scenarios, locations, lighting conditions, and functional states (i.e., idle, active, start-up).

Multiple resources can be used to gather the required training data for the model. In the case of construction equipment, a few databases are available that contain images of construction equipment that can be used to train a model. Three common tactics to gather the training data include:

1- Use large-scale public databases that contain classified and labelled data ready to train artificial intelligence models, like COCO (Microsoft Common Objects in Context), the Open Image Dataset (Kuznetsova et al., 2020), or the Alberta Construction Image Dataset (ACID) (Xiao, and Kang,2021). ACID is one of the most extensive datasets for detecting construction machines; the University of Alberta's AIRCon-Lab created it. ACID was created to aid in using and developing deep learning applications in construction automation. The dataset is available for download by anyone who wants to use it. It features photos of up to ten different classes of construction equipment gathered from construction sites worldwide (Xiao, and Kang,2021).

The ACID dataset provides many futures that can be downloaded and used to train the model after submitting a request on the dataset website. (Xiao, and Kang,2021). ACID Construction Dataset pro features:

- Ten categories of construction machines.
- 10,000 labelled images.
- 15,767 construction machine objects.

2- Another approach to gathering training data for the artificial intelligence model is to capture the data directly on-site for various kinds of construction equipment while they are functioning on-site; this can be time-consuming, but with appropriate quality management, high-quality training data can be created using this method.

3- The last approach to gathering different training data is to use automated web-crawling techniques, which use a specific type of crawling software that can work independently as BOT. (A BOT is a computer program configured to do specified activities to replicate human actions). Bots are meant to automate operations without human participation, removing the need for time-consuming manual processes. These jobs are frequently repetitious, and they can be completed significantly faster, more reliably, and correctly than a human can, like searching the whole Internet for specific kinds of data photos for various kinds of construction equipment. (Nath et al.).

The author opted to use the third technique, web crawling, and video frame splitting techniques, to build the dataset for this thesis so he could customize the images included in the project.

2.2. Creation of CON3 dataset

For the sake of this thesis, a new dataset was generated, and it is called (CON3). The CON3 dataset is generated through web crawling and video frame splitting techniques.

The dataset was gathered by searching public imagery datasets and archives for high-quality pictures for the three classes that the model is trained on: excavators, wheel loaders, and dump trucks. In particular, the author searched digital libraries and videos on the Internet containing different kinds of construction equipment used on construction sites. The author extracted images from videos using splitting software that takes a shot of the videos every five seconds and converts it to an image, which provided the database with realistic images of the construction equipment in use on-site. Images with various angles, positions, and lighting conditions are critical, as this allows users to train the model to identify equipment in various conditions and situations.

After compiling all images, images were filtered to exclude any unwanted images that would result in false-positive identifications, which, in turn, would reduce the accuracy of the model. To do so, only images that contained the primary classes of this project – excavators, wheel loaders, and dump trucks – were selected. Blurred images were avoided to avoid false positives in the model later.

At the end of this phase, the total number of images inside the training dataset was 2,680.

2.3. Dataset annotation

After getting all the data needed for the training, the data needed to be annotated; each image was annotated by drawing boxes for selected objects inside each picture for each class in this project. These boxes are called bounding boxes. They highlight the location of each element inside the picture; each bounding box has unique coordinates on the X&Y axis of each picture, which can be represented by different formats like YOLO, COCO, and Albumentations. Annotation is critical to the success of the training of the artificial intelligence model to avoid any missed detection due to poorly annotated training data.

The different annotation formats identify and annotate the object the model will identify.

Each annotation format has a unique way of addressing bounding box coordinates. For example, albumentations is like pascal_voc because it uses four values [x_min, y_min, x_max, y_max] to represent a bounding box. However, unlike pascal_voc, albumentations use normalized values. To normalize values, users divide coordinates in pixels for the x- and y-axis by the width and the height of the image (width and height are also measured in pixels). Albumentations uses this format internally to work with bounding boxes and augment them.

COCO is a format used by the dataset of the same name. In COCO, a bounding box is defined by four values in pixels [x_min, y_min, width, height]. They are coordinates of the top-left corner and the width and height of the bounding box.

YOLO V5 annotation format is considered one of the most used annotation formats in object detection. It is popular due to its excellent detection performance and lightweight neural network model weights files, making it one of the most used deep learning architectures in object detection applications (Zhang, Shizhao, et al.,2021).

Figure 3 shows the YOLO annotation format; a bounding box is represented by four coordinates [x_center, y_center, width, height]. x_center and y_center are the normalized coordinates of the center of the bounding box.

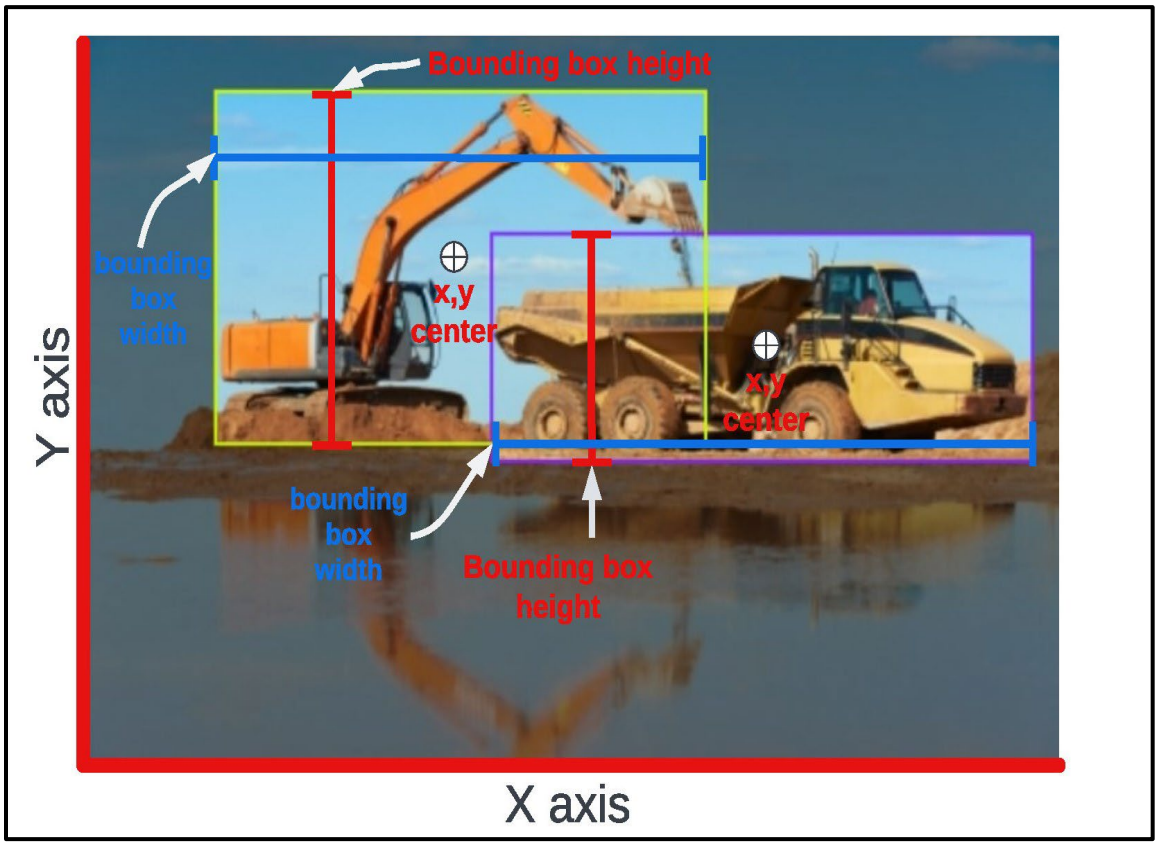

Figure 3-Bounding boxes on different construction machinery in YOLO annotation format

2.3.1. Data annotation tools

Many available data annotation tools online can be used to annotate different training data and make the data suitable for training different artificial intelligence models to detect custom objects. It is crucial to know what type of data will be used to train the model because not all annotation tools can deal with all formats of training data. For example, not all annotation tools would handle such data if the model needs to annotate 3D LIDAR data. That can also be said about DICOM data, which is a digital output format for X-rays imagery data. DICOM imagery data can be used to train artificial intelligence models to detect various kinds of tumors in X-rays and CT scans (Wang et al., 2020)

2.3.2. Annotation of the dataset on the Roboflow platform

The author used the Roboflow platform to annotate images in the CON3 dataset. Roboflow is a free-to-use and publicly available platform, making it an excellent tool for this thesis project. The Roboflow platform provides a variety of annotation assistance tools and techniques to improve the efficiency of the annotation process.

This is especially important in large datasets, which can be time-consuming to annotate accurately.

The Annotation process in the Roboflow platform starts with uploading the image dataset to the platform. The data can consist of raw unannotated imagery only or an annotation label format file containing some annotation data for imagery in the dataset. Roboflow recognizes twenty-six distinct kinds of computer vision annotation formats; the platform supports all of them.

The uploaded imagery data is then annotated using Roboflow's annotation toolbox. Each image needs to be annotated, meaning every object inside the image will be identified by drawing a bounding box around it and choosing the suitable class for it (Figure 4).

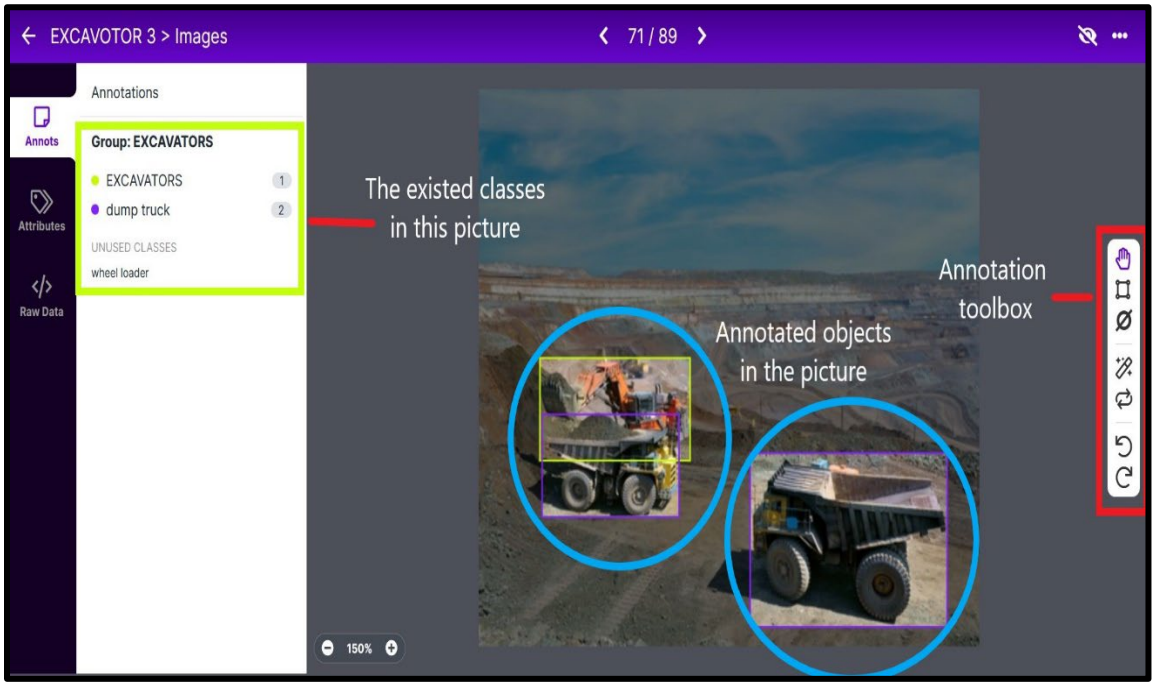

Figure 4-Overview of Roboflow annotation tool window-Roboflow 2022.

Roboflow's annotation tool supports the annotation of custom objects inside the images. To annotate a custom object in an image, the class of that object must be identified, and it must have its unique name as an ID.

Each unique object needs a unique class name. Having a unique class ID is particularly important to avoid false detection, which compromises the model's accuracy and is difficult to fix later. The dataset also included 3,910 annotations for all three training classes.

Figure 5 shows the number of annotations for each class; excavators had most of the annotations (1561 annotations) followed by dump trucks (1,268 annotations) and wheel loaders (1,081 annotations).

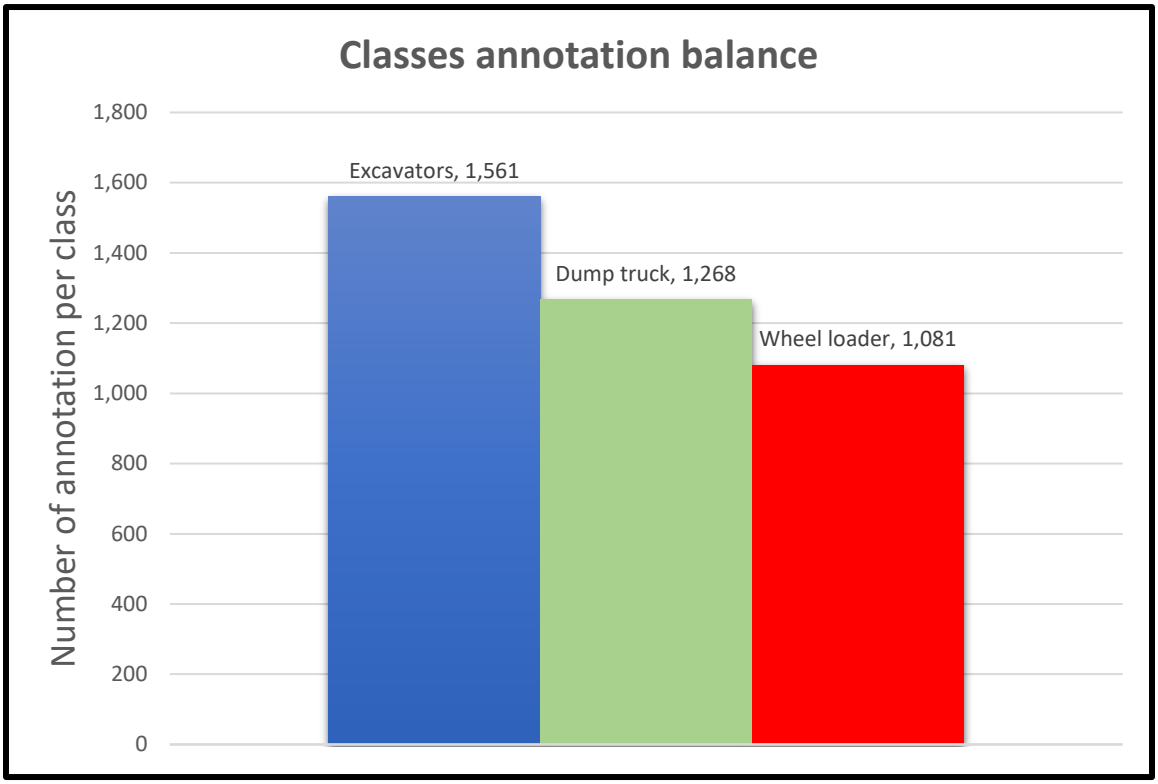

Figure 5-Classes annotation number per class

Figure 5 shows that not all the classes need to have the same number of images or annotations. However, to train the model with high accuracy, it is recommended that each class will have at least 1500 images per class to train the model ("Tips for Best Training Results · Ultralytics/Yolov5 Wiki").

2.4. Dataset split

The image dataset should be split into three main categories: (1) training set, (2) validation set, and (3) testing set, to be used at different stages of model training. To train the model for this thesis, the split was as follows:

- Training set: 85% of the total annotated images. It is critical to devote the majority of the dataset data to training; this provides the model with enough data to recognize the custom objects it is trained to recognize.
- Validation set = 10 % of the total annotated images. The validation dataset is distinct from the testing data. Images in this dataset are withheld from the model's training dataset and are not used to "teach" the model; instead, they are utilized to provide an unbiased evaluation of the final adjusted model's performance for comparing or selecting amongst models (James et al., 2013). In other words, the validation dataset is used to check whether or not the model learned to identify objects in each class correctly.
- Testing set = 5% of the total annotated images. The testing set is used to test the model's performance after training. It is essential to ensure that the testing data has never been used in the training process to have an unbiased and accurate performance overview of the model.

Figure 6 shows the partitioning of the data for each set described above.

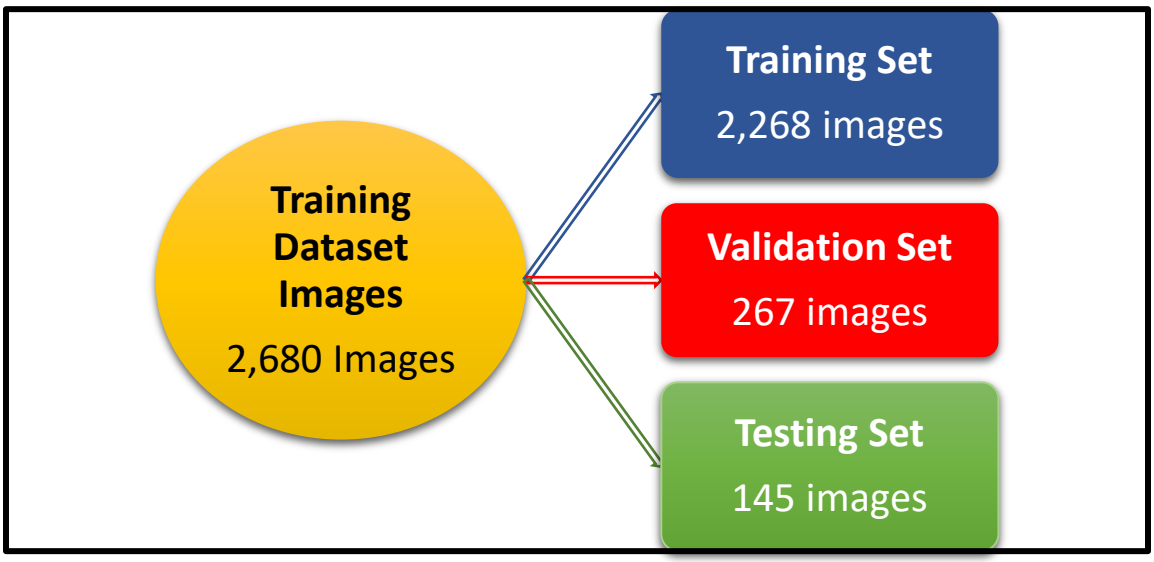

Figure 6-Dataset split configuration for each category

2.5. Dataset pre-processing

Image data used to train the model needs to be pre-processed to ensure it is suitable for training. Pre-processing supports optimal training performance and avoids unnecessary energy and time losses in the training process that cost time and money, especially in larger models that can take days to train.

The following pre-processing steps could be applied to the dataset before starting the training phase of the model:

- **Auto-Orientation:** this is important to ensure that all the training data is oriented in the same direction and on the most natural orientation possible, which will help the model learn the typical orientation of the custom objects the model will be trained to detect.
- **Resizing training imagery:** downsizing images results in smaller file sizes and faster training. Moreover, this allows for consistency in the training data, which is essential to the function of deep learning architecture models.
- **Auto-Adjust Contrast:** this boosts images contrast based on the image's histogram to improve normalization and line detection in varying lighting conditions. This supports consistency among the training data.
- **Apply Null Filter:** it is important to maintain a certain ratio of annotated null images, i.e., ensure that the training dataset includes images that do not include any objects. This allows the model to learn that not all images include objects, which can reduce the false-positive detections of the model.
- **Modify classes:** class modification allows users to correct any mistakes that may have happened in choosing class names during the annotation process. The class modification also supports turning off certain classes and combining other classes under one class name before training the model.

Figure 7 shows some of the available pre-processing tools and algorithms available in the Roboflow platform.

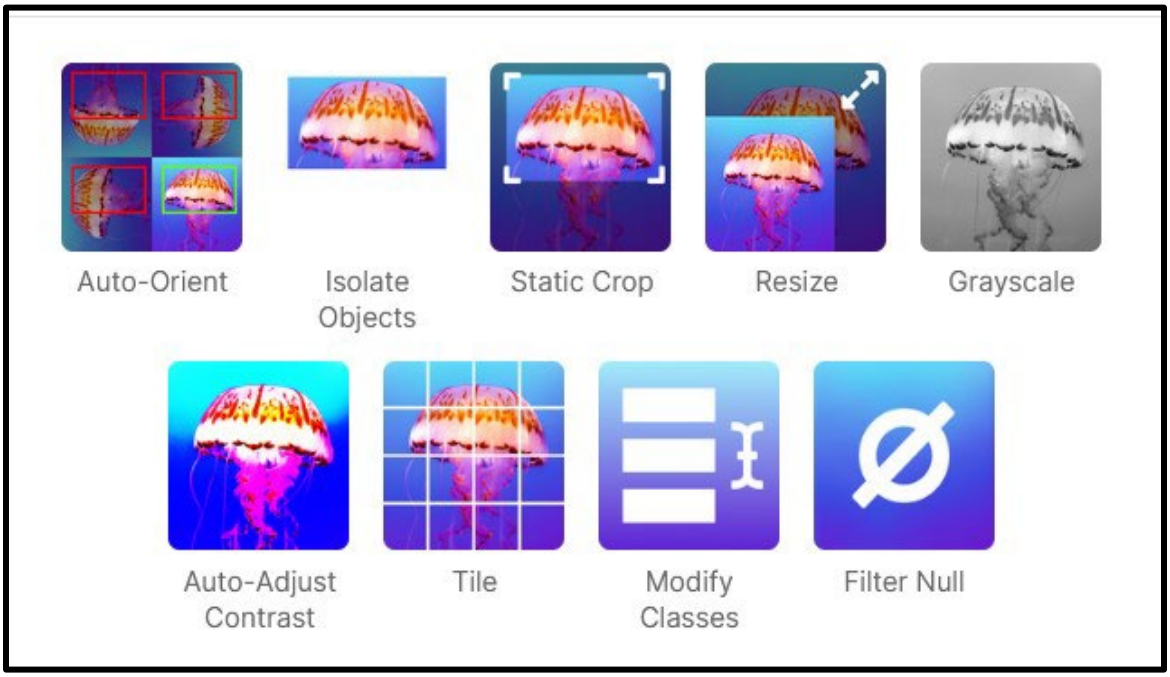

Figure 7-Preprocessing tools and algorithms in the Roboflow platform

For the author's work, only Auto-Orientation, Auto-Adjust Contrast, resizing training imagery data, and Null Filter was required during pre-processing. Indeed, pre-processing should only be done as required for the quality and type of the training data. For example, images captured by professional photographers may be less likely to require auto-adjust contrast pre-processing. The contrast would likely already be strong by virtue of the photographer's skill when setting up and capturing the photo.

2.6. Dataset augmentation:

Data augmentation allows the creation of training data in forms other than its original form. Developing new training examples from existing ones is known as image augmentation. This involves modifying training images slightly to create a new training dataset. For example, augmentation could produce a new image that is a bit brighter, cut a section from the original image, change the situation of the original image, and so on (Albumentations -image augmentation, 2022).

Image augmentation can: (1) boost the detection rate of the model without increasing the training dataset size and (2) increase the diversity of the training set without acquiring any new data.

Training dataset image augmentation has several key benefits:

- The model's repeatability is enhanced, and it gives researchers the ability to discover new things about how the model performs.

  For example, a researcher could discover that the model performs better on bright photographs than on dark images, in which case the researcher should gather more low-light training data. (Image Augmentation - Roboflow, 2022).
- Augmentation reduces the training time. Augmentation usually requires the CPU to process the training images, which will create a bottleneck for the GPU while training the model.

- Augmentation can save costs as a powerful GPU is required to train models. The relationship between image diversity and training time and power is generally linear. A more diverse set of images in the dataset will require more training time and power than an augmented dataset that comprises fewer raw images. In turn, an augmented dataset can be trained in less time, thereby reducing the cost of GPU time and/or GPU power.

Figure 8 shows the augmentation processes applied to the CON3 dataset to train the model for this thesis: Image horizontal and vertical flip, image clockwise 90-degree rotation, image cropping, image forty-five rotation, cut-out boxes, hue change to 25%, adding noise, and finally, image blur.

Figure 8-The augmentation processes that were applied to the training dataset

2.7. Final dataset export with different variations

The final version of the training dataset was exported with eight different variations, one for each pre-processing and augmentation transformation; this resulted in 18,556 images in the training dataset. Each image has different variations according to the pre-processing and augmentation conversions applied to the image. This step allows the training dataset to expand, so the model has a more extensive training dataset without increasing the number of original images. The final training dataset was exported to the model training Colab notebook through Roboflow's API. This API provided the Colab notebook with the necessary training dataset, including labels for each image, without manually uploading all the images and the labels to the Colab storage space.

2.8. Object detection deep learning architecture

For this thesis, it was essential to train the model on the state-of-the-art, available, and publicly accessible deep learning architecture that anyone can use.

In the last decade, object detection algorithms have improved significantly in two main categories: (1) the detection performance of the models in terms of FPS and (2) producing neural network model weights files that have smaller file sizes. Understanding the history of object detection algorithms on architectures is essential to understand why the author selected the computer detection architecture he did for this thesis.

2.8.1. RCNN (Region-Based Convolutional Neural Networks)

The modern object detection algorithms can be traced to early 2014, when the RCNN was introduced. It opened the path for newer deep learning architectures to be improved and developed in the deep learning era. RCNN begins with a selective search to extract a set of object suggestions, "object bounding boxes." (Ren et al., 2018).

Each suggestion is then resampled to a fixed-size image and put into a CNN model that has already been trained to extract features. Finally, linear SVM classifiers are utilized to predict the existence of an item and distinguish object types inside each region. (Tetko et al.).

However, RCNN was not a perfect solution to the object detection problems and performance bottlenecks. Although RCNN outperforms older approaches, it has several flaws. The detection performance suffers due to redundant feature calculations on many overlapping predictions for object detection, which can reach up to two thousand bounding boxes for each image. A faster RCNN detector was proposed in 2015, and it was the first end-to-end deep learning object detector and the first near-real-time deep learning object detector. (Ren et al., 2015).

Even though Faster RCNN overcomes RCNN and Fast RCNN's performance limitations, there was still computation duplication at the later detection phases, which was a time-consuming operation that degraded network performance.

Figure 9 shows the Faster RCNN diagram; it illustrates the various parts of the Faster RCNN architecture and how it paved the ground for modern object detection algorithms.

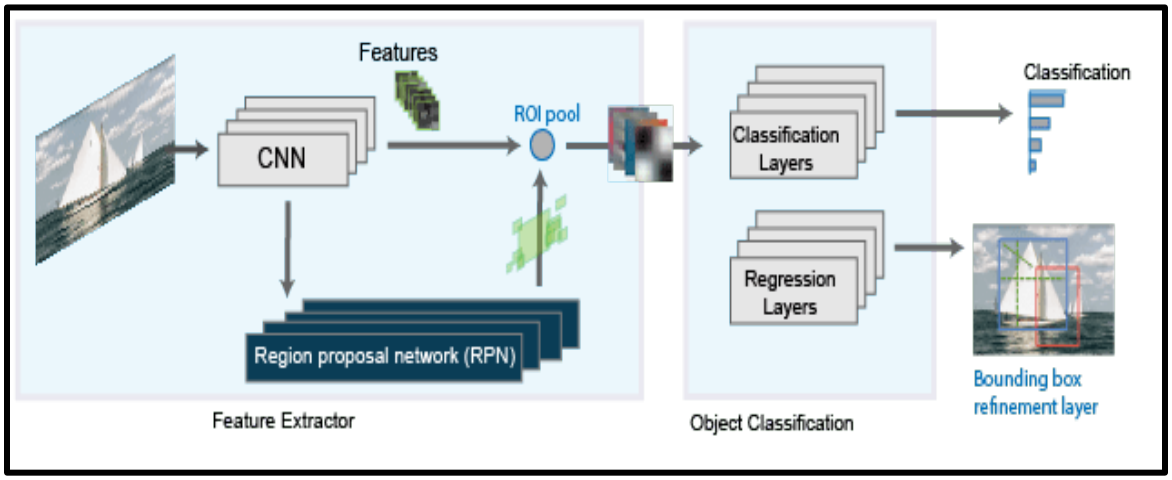

Figure 9-Faster R-CNN diagram (The MathWorks, Inc.)

2.8.2. YOLO (You Only Look Once)

The prior object detection techniques used areas to locate an item inside the image. Instead, YOLO examines the entire image and sub-areas within the image that have a high probability of including the object class of interest. YOLO improves detection performance by training on entire photos using a single CNN that predicts multiple bounding boxes determines probabilities that each box contains a given class using YOLO. It also estimates all bounding boxes for an image throughout all classes simultaneously. The YOLO v5s and YOLO R deep learning architectures are most commonly used to build the construction equipment detector described in this thesis. YOLO R is one of the latest editions of the YOLO series of deep learning architectures that is available at the time of this publication.

YOLO v5s and YOLO R are similar. However, their structure has some key differences; YOLOR-P6 is an object identification machine learning method that differs from YOLOv1 to YOLOv5 in terms of origin, design, and model infrastructure. YOLOR-P6 stands for "You Only Learn One Representation," not "You Only Look Once," as in older YOLO versions from 1 through (Wang, C.-Y 2021).

YOLOR-P6 is described as a "unified network that encodes implicit and explicit information simultaneously," according to Wang (2021).

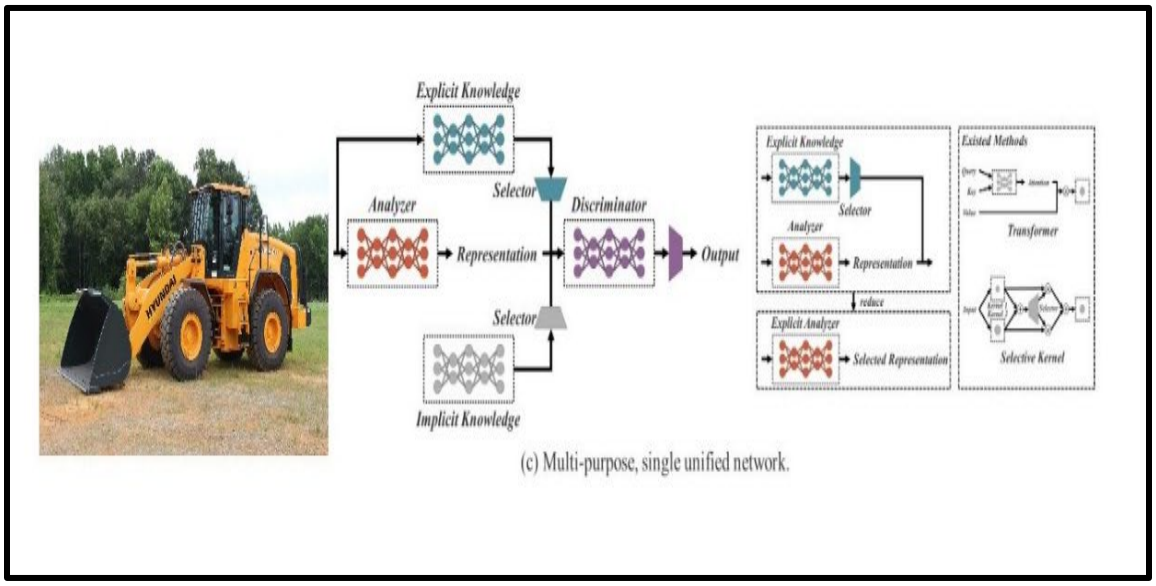

Figure 10-YOLOR-P6 neural network diagram (Wang, C.-Y (2021).

YOLO v5 is a single-stage object detector that contains three key components: (1) the backbone of a model, (2) the neck of a model, and (3) the head of a model; the primary purpose of Model Backbone is to extract significant characteristics from an image. The CSP - Cross Stage Partial Networks backbone is utilized in YOLO v5 to extract valuable features from an image that the user inputs (Ultralytics/Yolov5 Wiki, 2021).

2.9. Training the model

The training of this study's model is done using Google Colab notebook because it is the most commonly available online tool for public use that can be utilized to train artificial intelligence models without the need for expensive and potentially difficult-to-acquire hardware, e.g., a GPU.

Google Colab notebooks provide various GPUs that can be used to train and process deep learning algorithms; these GPUs are used for this thesis.

The author leveraged Google Colab plus to train the YOLO v5 and YOLOR-P6 models to ensure that the training time would not exceed the training time limit included with the standard Colab notebook subscription. Two Google Colab notebooks were used to train each model.

After the two notebooks finished the model's training, a neural network model weights file was saved to save the training outcomes. The neural network model weights file is used to deploy the model on a construction site; that is, the author would utilize the weights file to produce predictions to detect the three different classes considered in this study.

2.10. Neural Network Model Weights

The output of the model's training will be one file called neural network model weights; this file will present the learned knowledge and parameters that the model figured out through repetitive training by trial and error. These specific parameters are used to reach the required function of the model – in this case, to recognize the three different kinds of construction equipment.

A neural network comprises a sequence of nodes, also known as neurons. A collection of inputs, a weight, and a bias value are all contained within each node. When an input is fed into a node, it is multiplied by a weight value, and the result is either seen or sent to the next layer of the neural network.

The weights of a neural network are frequently stored in one file with an extension of (.pt); this is the case in YOLO architecture (Alammar,2021).

Figure 11 shows the different components of a neural network with one input and one output. The input can be imagery data, and the weights and bias represent the parameters identified by the training process.

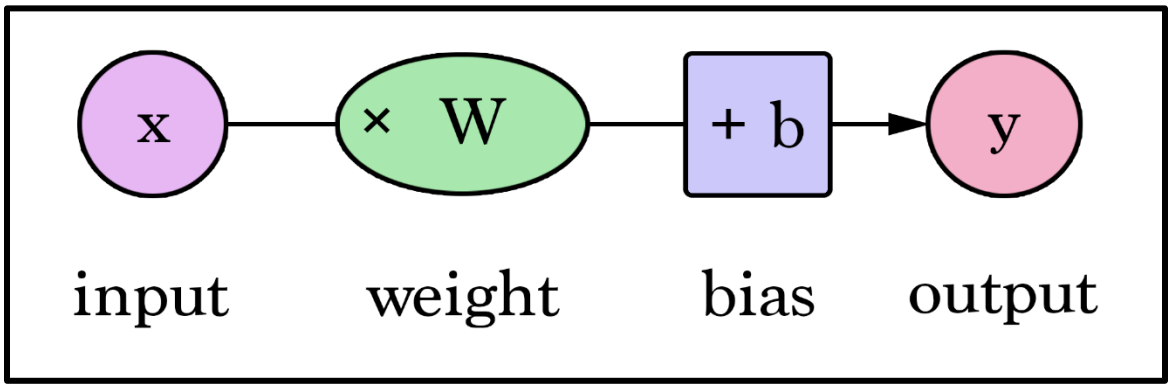

Figure 11-Alammar, Jay. Neural Network with One Input and One Output. 2022

2.11. Deploy the trained model and get predictions

The Roboflow platform API was used to get live predictions from the YOLO v5 and YOLOR-P6 models using the Neural Network Model Weight file(Figure 12).

This approach is straightforward; the author deployed a model to get live predictions from the trained model, and he was then able to check whether the predictions indeed detected the various objects in actual image data. Models can be deployed in a variety of ways such that they yield lifetime predictions; however, these require hosting the model on separate online container spaces, e.g., Docker.

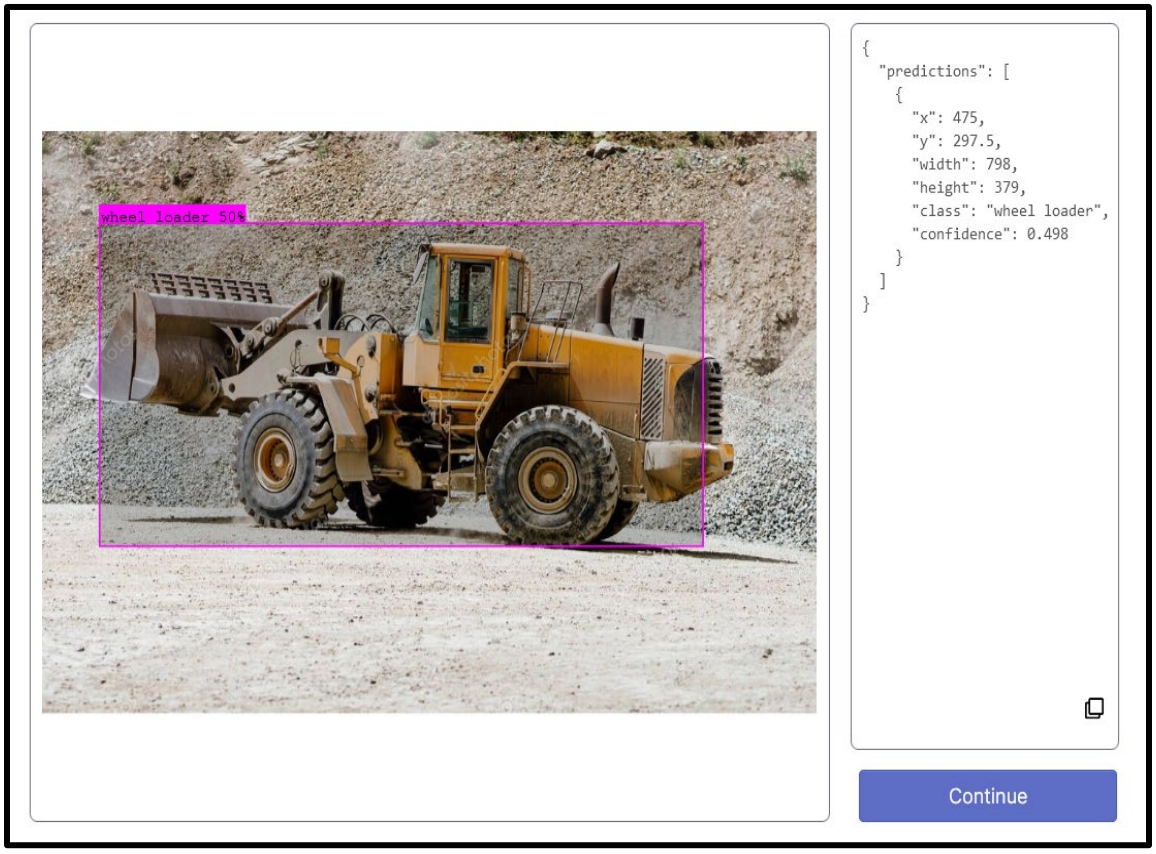

Figure 12-Model deployment prediction window on Roboflow platform

3. Results and Discussion

After finishing the training using the YOLO v5 and YOLOR-P6 architectures, there was no clear "winner" in terms of performance. Indeed, each architecture has strengths and weaknesses that will affect its use. Thus, users will have to select an architecture that is best suited to their situation, considering training time, the size of the weights file, and the accuracy of the model predictions.

3.1. Training time

YOLOv5s took only **4 hours and 52 minutes** for **150 training epochs s**. However, the YOLOR-P6 needed **14 hours and 35 minutes** to finish 150 training epochs s. Both architectures completed the training on Tesla P100 GPU as a processing unit on the Google Colab notebook. The training time for the YOLOv5s was approximately a third of the training time required for the YOLOR-P6 model. In cases where the training dataset is extensive and has much more training data than the CON3 dataset, the training time could become prohibitive.

3.2. Neural Network Model Weights file size

This study finds that the YOLOv5s model is the most efficient for building an artificial intelligence model to detect construction equipment when it will be deployed on low power processing devices like DVRs (CCTV camera recorders) or other IoT devices. The small size of its weight's files relative to other versions of YOLO models- **14.4 MB** for YOLOV5s vs. **288 MB** for YOLOR-P6 is essential because low power CPUs would not be able to process more extensive neural networks with larger weights files. Thus, if the intent is to leverage the model for

real-time object detection on-site with a smartphone or similar, rather than a large computer, YOLOv5s will be most efficient.

3.3. The precision and the accuracy of the model detection performance

The results between the two models show that after 150 epochs of training, the **YOLOR-P6** has the best **mAP at 0.98**, while the **YOLOv5s mAP is 0.93.** However, both results are good results according to the industry standards and considering that the model was only trained for 150 epochs s, which is much lower than the recommended numbers of training epochs by the YOLOv5s wiki. Note that the author could not achieve the recommended threshold for the number of training epochs because the YOLOR-P6 architecture exceeded the training time limit on all Google Colab notebooks (the training platform used for this thesis).

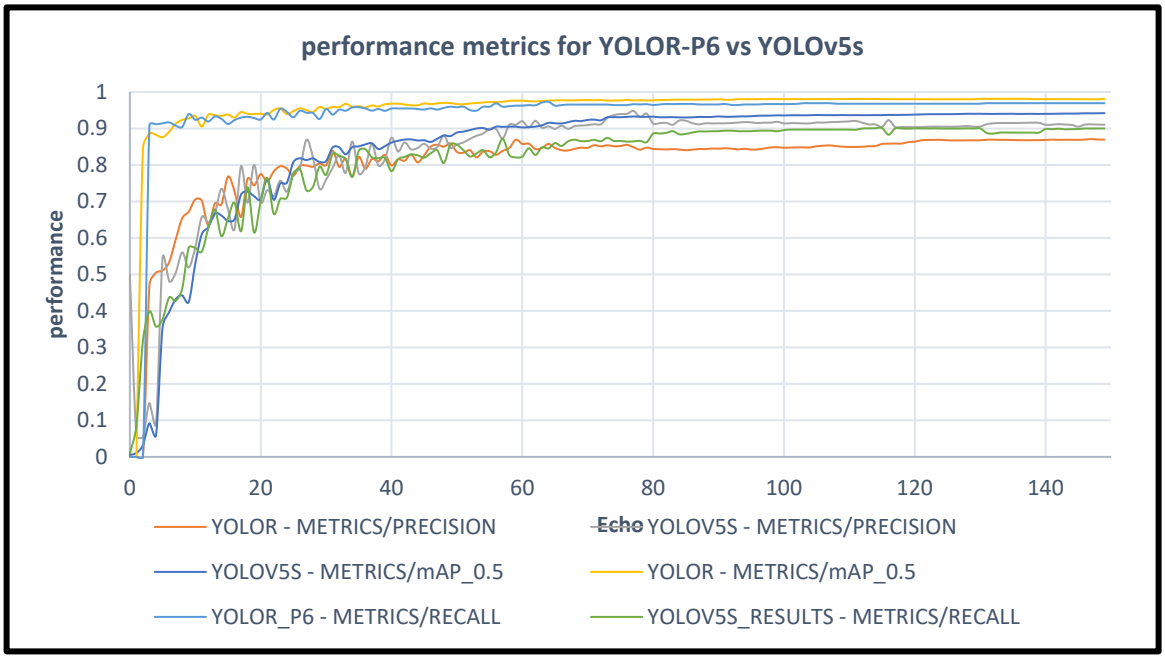

Figure 13-performance metrics for YOLOR-P6 vs. YOLOv5s

Figure 14 shows the detection output of the YOLOv5s model after being trained for **150 epochs** of training; the figure shows that the model was able to detect all the three classes without any problem and with high accuracy with tight bounding boxes, which is a good indicator of accuracy performance.

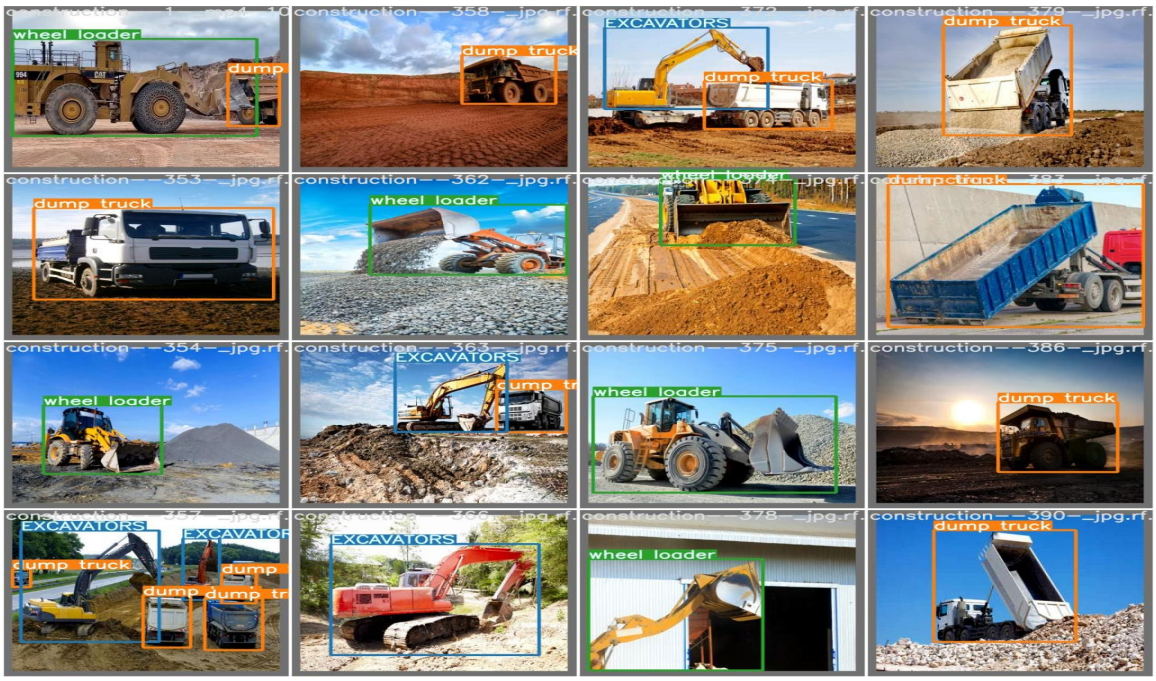

Figure 14-YOLOv5s model detection output for different classes

It is fair to say that YOLOV5s had some advantage over YOLOR-P6 in this study because the YOLOV5s is built to have the best detection performance across all other versions of YOLOV5 and provide the tiniest weight file possible at the same time, but this comes at the cost of small precision and accuracy performance hit and much lower number of parameters in the neural network.

Also, a very few available performances benchmarks as a reference, especially for YOLOR, because it is still a newly developed architecture. The community still does not fully implement and recognize it as the YOLOV5 is now.

3.4. Limitations

As with all studies, this work is subject to limitations, including:

- The results of this study can be impacted by any updates in the proposed neural networks architecture structure; this is very common, especially with YOLOV5s. YOLOV5 is constantly being updated and changed by multiple companies and associations (e.g., Ultralytics) to improve its performance and adaptivity to different use cases. As such, the results of this study are only valid for the time of its publishing but give clear aspect how each of these deep learning architectures behaves generally.
- The quality and diversity of the training dataset can also impact the accuracy performance results of this study for the models studied. However, the diversity and quality of the training dataset would not impact the training time results, nor the results that address the size of the output weight file.

4. Conclusion

4.1. Impacts of this research

This research can impact the construction industry in a variety of ways, including:

- **Autonomous driving in construction equipment:** is perhaps the most important output of this research. A trained model with the highest possible accuracy and detection performance can recognize different kinds of construction equipment. This is critical as the construction industry moves towards autonomously driven equipment, as accurate object detection will reduce the risk of autonomous vehicles colliding with each other, let alone other hazards on-site.
- **Increase safety construction sites**: using the trained model to detect different kinds of construction equipment can benefit automatic safety systems on construction sites. The trained model could allow these monitoring systems to detect and observe construction equipment automatically 24/7 and issue early warnings if an accident seems imminent or if construction personnel is deemed too close to the equipment. This could reduce the fatality rate in the construction industry.

- **Automatic counting and clerking:** the model can be used to track the movement of construction equipment and automatically log utilization and productivity parameters. In turn, these parameters can help inform decisions about whether various pieces of construction equipment should be bought or rented for each project. As utilization increases, construction companies may determine that a piece of equipment is worth purchasing.
- **Understanding how construction equipment moves on the site:** this will allow construction engineers to plan the construction sites better to give the construction equipment the fastest routes to travel inside the site, improving site logistics, and ultimately, saving both time and money on projects.

4.2. Recommendations for future research – unresolved questions

One of the most critical questions that should be answered in future research is the performance of other artificial intelligence architectures other than the YOLO architecture family of models. For instance, future researchers may assess the performance of Tensor Flow from Google or Detectron2 from Facebook.

4.3. Summary

This thesis finds that YOLOV5s is one of the most beneficial computer vision architectures due to its fast-training time, fast detection rate, and relatively small size of its neural network weight output files. This makes it an ideal solution for deploying artificial intelligence models on devices with low processing power like CCVT camera decoders or autonomous driving systems.

Moreover, YOLOv5s has an easier workflow than YOLOR-P6 due to the different available training platforms that he's being offered by the developers of the deep learning architecture, so anyone can use it to build a precise computer vision model to detect custom objects. In this case, it is construction equipment; the construction industry has one of the lowest productivity rates compared to other industries globally (Jonathan Woetzel, 2022), and implementation of a precise computer vision model could feasibly reduce the number of workers required to complete capital projects, in turn, increases the efficiency of the construction industry.

REFERENCES

1- "A Guide for Model Production - Roboflow". *Docs.Roboflow.Com*, 2022, https://docs.roboflow.com/model-tips.

2- "What Is SVM, Techopedia". *Techopedia.Com*, 2022, https://www.techopedia.com/definition/30364/support-vector-machine-svm.

3- "YOLO". *Medium*, 2022, https://towardsdatascience.com/yolo-you-only-look-once-17f9280a47b0.

4- "Facts On Safety at Work". *Ilo.Org*, 2005, https://www.ilo.org/wcmsp5/groups/public/---dgreports/---dcomm/documents/publication/wcms_067574.pdf.

5- "Tips For Best Training Results · Ultralytics/Yolov5 Wiki". *Github-Yolov5*, 2022, https://github.com/ultralytics/yolov5/wiki/Tips-for-Best-Training-Results.

6- "Worker Fatalities, OSHA". *Occupational Safety and Health Administration*, 2022, https://www.osha.gov/data/commonstats. Accessed 15 Dec 2021.

7- Abioye, Sofiat O. et al. "Artificial Intelligence in The Construction Industry: A Review of Present Status, Opportunities, and Future Challenges". *Journal of Building Engineering*, vol 44, 2021, p. 103299. *Elsevier BV*, https://doi.org/10.1016/j.jobe.2021.103299.

8- Alammar, Jay. "A Visual and Interactive Guide to The Basics of Neural Networks". *Jalammar.Github.Io*, 2022, http://jalammar.github.io/visual-interactive-guide-basics-neural-networks/.

9- Albumentations.ai. 2022. *Albumentations -image augmentation*. [online] Available at: <https://albumentations.ai/docs/introduction/image_augmentation/> [Accessed 18 February 2022].

10- Dasiopoulou, S. et al. "Knowledge-Assisted Semantic Video Object Detection". *IEEE Transactions on Circuits and Systems for Video Technology*, vol 15, no. 10, 2005, pp. 1210-1224. *Institute Of Electrical and Electronics Engineers (IEEE)*, https://doi.org/10.1109/tcsvt.2005.854238.

11- Deng, Li, and Dong Yu. *Deep Learning: Methods and Applications*. 1st ed., Microsoft, 2014.

12- Docs.roboflow.com. 2022. *Image Augmentation - Roboflow*. [online] Available at: <https://docs.roboflow.com/image-transformations/image-augmentation> [Accessed 18 February 2022].

13- Gothane, Dr. Suwarna. "A Practice for Object Detection Using YOLO Algorithm". *International Journal of Scientific Research in Computer Science, Engineering, and Information Technology*, 2021, pp. 268-272. *Technoscience Academy*, https://doi.org/10.32628/cseit217249. Accessed 6 Feb 2022.

14- Hui, Jonathan. "Map For Object Detection". *Medium*, 2022, https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173.

15- Janane, G. and K. Meena Alias Jeyanthi. "Deep Learning with Images using Tensorflow." International journal of engineering research and technology 8.9 (2019). 10 3 2022. <https://ijert.org/deep-learning-with-images-using-tensorflow>.

16- Jonathan Woetzel, Jan Mischke. "The Construction Industry Has a Productivity Problem". *Marketwatch*, 2022, https://www.marketwatch.com/story/the-construction-industry-has-a-productivity-problem-and-heres-how-to-solve-it-2017-03-04.

17- Ker, Justin et al. "Deep Learning Applications in Medical Image Analysis". *IEEE Access*, vol 6, 2018, pp. 9375-9389. *Institute Of Electrical and Electronics Engineers (IEEE)*, https://doi.org/10.1109/access.2017.2788044. Accessed 2 Feb 2022.

18- Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T. and Ferrari, V., 2020. The Open Images Dataset V4. *International Journal of Computer Vision*, 128(7), pp.1956-1981.

19- Legg, Shane, and Marcus Hutter. "Universal Intelligence: A Definition of Machine Intelligence". *Minds and Machines*, vol 17, no. 4, 2007. *Springer Science and Business Media LLC*, https://doi.org/10.1007/s11023-007-9079-x.

20- Lin, Tsung-Yi, et al. "Microsoft Coco: Common Objects in Context." *Computer Vision – ECCV 2014*, 2014, pp. 740–755., https://doi.org/10.1007/978-3-319-10602-1_48.

21- Mandal, Manav. "CNN for Deep Learning | Convolutional Neural Networks". *Analytics -Vidhya*, 2022, https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/.

22- McCann, Michael. "Heavy Equipment and Truck-Related Deaths on Excavation Work Sites". *Journal of Safety Research*, vol 37, no. 5, 2006, pp. 511-517. *Elsevier BV*, https://doi.org/10.1016/j.jsr.2006.08.005. Accessed 2 Jan 2022.

23- Nath, Asoke et al. "Designing and Implementing Conversational Intelligent Chat-Bot Using Natural Language Processing". *International Journal of Scientific Research in Computer Science, Engineering, and Information Technology*, 2021, pp. 262-266. *Technoscience Academy*, https://doi.org/10.32628/cseit217351.

24- Noronha, Elroy et al. "BRAIN TUMOR DETECTION USING DEEP LEARNING". *International Journal of Innovative Research in Engineering & Management*, vol 8, no. 4, 2021. *Marwah Infotech*, https://doi.org/10.21276/ijirem.2021.8.4.4.

25- Oprach, Svenja et al. "Building the Future of the Construction Industry Through Artificial Intelligence and Platform Thinking". *Digitale Welt*, vol 3, no. 4, 2019, pp. 40-44. *Springer Science and Business Media LLC*, https://doi.org/10.1007/s42354-019-0211-x.

26- Reddy, Martin. *API Design for C++*. Elsevier, 2011.

27- Ren, Shaoqing et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 39, no. 6, 2015, pp. 1137-1149. *Institute Of Electrical and Electronics Engineers (IEEE)*, https://doi.org/10.1109/tpami.2016.2577031.

28- Ren, Yun et al. "Object Detection Based on Fast/Faster RCNN Employing Fully Convolutional Architectures". *Mathematical Problems in Engineering*, vol 2018, 2018, pp. 1-7. *Hindawi Limited*, https://doi.org/10.1155/2018/3598316.

29- Rojas-Perez, Leticia Oyuki, and Jose Martinez-Carranza. "Towards Autonomous Drone Racing Without GPU Using An OAK-D Smart Camera". *Sensors*, vol 21, no. 22, 2021, p. 7436. *MDPI AG*, https://doi.org/10.3390/s21227436.

30- Tetko, Igor V et al. *Artificial Neural Networks, and Machine Learning - ICANN 2019: Workshop and Special Sessions*. 2019.

31- The MathWorks, Inc. *Faster R-CNN Diagram*. 2022, https://au.mathworks.com/help/vision/ug/getting-started-with-r-cnn-fast-r-cnn-and-faster-r-cnn.html#mw_a9cdd2b3-b910-4d3d-90db-b485b415fd9b. Accessed 19 Feb 2022.

32- Voulodimos, Athanasios et al. “Deep Learning for Computer Vision: A Brief Review”. *Computational Intelligence and Neuroscience*, vol 2018, 2018, pp. 1-13. *Hindawi Limited*, https://doi.org/10.1155/2018/7068349. Accessed 2 Feb 2022.

33- Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2021). *You Only Learn One Representation: Unified Network for Multiple Tasks*. https://arxiv.org/abs/2105.04206.

34- Wang, Sophia Y. et al. “Big Data Requirements for Artificial Intelligence”. *Current Opinion in Ophthalmology*, vol 31, no. 5, 2020, pp. 318-323. *Ovid Technologies (Wolters Kluwer Health)*, https://doi.org/10.1097/icu.0000000000000676.

35- Xiao, Bo, and Shih-Chung Kang. “Development of an Image Dataset of Construction Machines for Deep Learning Object Detection”. *Journal of Computing in Civil Engineering*, vol 35, no. 2, 2021, p. 05020005. *American Society of Civil Engineers (ASCE)*, https://doi.org/10.1061/(ASCE)cp.1943-5487.0000945. Accessed 2 Feb 2022.

36- Zhang, Shizhao, et al. “Domain Adaptive YOLO for One-Stage Cross-Domain Detection.” *Asian Conference on Machine Learning*. PMLR, 2021.