DATA601 Photogrammetry and Semantic Machine Learning for Construction Site Monitoring

Yu Lu (31896413)

Supervised by: Heyang (Thomas) Li, Zachary Todd

Submission date: 11/02/2022

ABSTRACT

Each year, there are around 6000 injuries on the New Zealand construction site that have resulted in more than a week away from work [1]. It is critical for the government to implement a new measurement to detect construction site hazards, thus preventing accidents from happening. This paper focuses on identifying personnel on site and determines whether they are wearing personal protective equipment (PPE), such as hard helmet and high-visibility clothing (Hi-Vis). The process of this project includes data processing, image annotation, image augmentation, dataset split, and training and testing of faster region-based convolutional neural networks (R-CNNs). The final model of this project achieves 75.9% mAP at IoU 0.5 on the testing dataset.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Organisation

1.1.1 Downer

Downer is the leading expert in construction and urban service, including airports, ports, railway, road, power, gas, streetscape, and cycleways. The company's health and safety objectives are to achieve zero harm and ensure the safety of its people, contractors, and community [22].

1.1.2 UC SAIL

University of Canterbury Spatial And Image Learning (UC SAIL) group works on research in collaboration with Waka Kotahi NZ Transport Agency (NZTA), Christchurch City Council (CCC), Christchurch Airport, AgResearch, and Stats NZ Tatauranga Aotearoa. The UC SAIL group research in image processing and machine learning problems and develops prototypes for deployment. UC SAIL group is kindly supported by the UC School of Mathematics and Statistics, UC Research, and Innovation, and KiwiNet [2].

1.2 Goals

The primary goal of this project is to identify any personnel on the construction site and run a simple safety check on them. By using state-of-art image machine learning techniques, any personnel, PPE such as hard helmet and Hi-Vis vest can be located on the image. Additionally, a relationship between the PPE and the on-site personnel can be established: whether they are wearing PPE as required, wearing some of the equipment but not all of them, or not wearing any.

The secondary goal is to define the boundary of the construction site. The construction site is usually surrounded by cones, barriers, fences, or straps. By identifying these items on the image using object detection algorithms, then the area enclosed by them is determined to be the construction site.

The third goal of this project would be checking whether the person in the image is inside the construction site or not. Passengers are not required to wear any PPE. Excluding passengers will prevent this system from making a false alarm. This can be achieved by finding out the positional relationship between the personnel, which is identified in the primary goal, and the construction site boundary, which is identified in the secondary goal.

2. DATA

2.1 Data information

The dataset which was used in this project is provided by Downer. All the data came in video format, which was taken in an underground tunnel construction site under the railway near Wellington. All the videos were taken in April 2021. Three videos were taken by camera one, came as mxf format, altogether 39 seconds long; three videos were taken by camera two, came as mp4 format, altogether 4 minutes and 16 seconds long; and another two videos were taken from a drone, came as mov format, altogether last 4 minutes and 4 seconds. All the videos are in high-definition resolution, which is 1920 pixels wide and 1080 pixels high. Two example frames extracted from the videos are shown in figure 1 and figure 2 in Appendix A

2.2 Constraints

All the data was provided by Downer. Due to the purpose of protecting the privacy of the employees and the company, all the human faces and the company logos, which are shown in the image, need to be covered or blurred in the final product. A non-disclosure agreement (NDA), which is drafted by Downer, is signed by the University of Canterbury. The project team is also required complying with the NDA.

The data was handed over on 17th January 2022, which left four weeks for the project team to do the image annotation, model training and testing. Shortened project period results in an affected and limited project outcome.

The video data was expected to be taken from a stationary monitoring camera on top of a pole. The monitoring camera should record continuous videos for the whole site. The camera should also be placed within 30 meters from the working crews. On the contrary, the videos were taken from either a drone or a handheld camera, thus the shooting angle could not be guaranteed to be perfect. As shown in figures 3 and 4, People and PPE were not always occurring in the image, which made a portion of the data unusable. People, Hi-Vis vest and hard helmets could not be detected by the algorithm if they were too far away from the camera since there were not enough pixels for the object detection model to process. The boundary of the construction site could not be seen in most of the frames, which is led by a camera too closed up or too further away. An example of an image that was taken too far away from the construction site was shown in figure 5. The inadequate camera setup made the secondary and the ultimate goal of this project unachievable. Each video frame was 1920 pixels in height and 1080 pixels in width. The video resolution was also lower than expected. Due to these constraints imposed by the provided dataset, only the primary goal of this project is achieved.

Figure 3. Inadequate video footage due to no presence of people and PPE.

Figure 4. Inadequate video footage due to no presence of people and PPE.

Figure 5. Inadequate video footage due to shot too far away from the construction site.

3. METHODOLOGY

3.1 Environment setup

This project is running on a remote computer which is located at the University of Canterbury. The remote computer uses Ubuntu 18.04.5 system, i9-10900X CPU with 10 cores and 20 threads, and NVIDIA GeForce RTX 3090 GPU with 24 GB of RAM.

This project is based on the faster R-CNN model provided by OpenMMLab and mmdetection. OpenMMLab is a next generation platform for general 3D object detection. Mmdetection, which is an open-source object detection toolbox based on PyTorch, is a project inside OpenMMLab. To run mmdetection on the remote server, Pytorch, torchvision, and CUDA are installed. Pytorch and CUDA are needed to be installed with the correct version to avoid compliance issues. In this case, Pytorch 1.10.1 and CUDA 11.2 are installed as suggested. Additionally, this project uses python packages NumPy, OpenCV, OS, pandas to perform data wrangling on the received dataset.

3.2 Data pre-processing

The data provided by Downer came in video format. Certain procedures must be done to the dataset so that it could be recognised by the object detection model.

3.2.1 Frame extraction

Because videos were shot from different angles, some video footage was determined to be irrelevant to this project. The video shot by the drone was too far away from the construction site, which made people and PPE in the image too small to be detected. An example of drone footage is shown in Figure 5. Since videos taken by drone were not sufficient for the object detection model, they were not used in this project. Furthermore, a video shot by camera one was fixed on a digger with zero people shown in the image, it was excluded from this project. One frame from it is shown in figure 4.

Once relevant videos were decided, frame extraction was done to the videos and obtained the image for the model training. The videos were all 50 frames per second. To balance between not having enough images to train the model and having too many images so that the time between two images was short thus overfitting the model, every tenth frame from the videos were extracted. The python code attached in Appendix A was written to conduct frame extraction.

There are 1323 images extracted in total. Each image is 1920 x 1080 in pixels.

3.2.2 Image annotation

Image annotation aims to label the images inside a dataset to train an image machine learning model. The annotation process consists of four steps, including defining objects' classes, drawing bounding boxes, labelling the bounding boxes, and exporting the annotation with the desired format.

VGG Image Annotator (VIA) was used to label the images. VIA is a simple and standalone manual annotation software which runs in a web browser and does not require any installation or setup [3]. The objects required to be labelled in this project were people, hard helmets, and Hi-Vis vests. These were the three categories that are used during the annotation process. Bounding boxes were drawn around the all the objects which can be distinguished by the human eye. Since the bounding boxes were all rectangles, containing noisy information such as background was unavoidable. However, bounding boxes should be drawn as close to the object as possible while covering every aspect of the object to avoid containing too much noisy data. The left image in figure 7 shows a good annotation because the bounding box boundaries are as close to the object as they can be. The right image in figure 7 shows an example of a bad annotation since it covers too much background.

The annotations were then exported in json format, which contained information such as image file name, image size, shape attribute name, and region. The json file were utilised in the following steps.

Figure 7. Good annotation (left) versus bad annotation (right)

3.2.3 Train-valid-test split

Train, validation, and test data split is essential for all supervised machine learning models. It is used to evaluate the performance of a model. In this project, 1324 images were randomly split into train, valid, and test datasets in the ratio of 4:2:1. This resulted in 756 images in the training dataset, 378 images in the validation dataset, and 189 images in the testing dataset.

3.2.4 Image augmentation

Data augmentation is a widely used method for generating additional data to improve machine learning systems [5]. It randomly generates new images by processing existing images using a certain dedicated method. More images mean there is more data to train the model, which usually gives a better result. However, prior knowledge in the researching field and intensive manual work is often part of the data augmentation requirement. The final product of the image augmentation needs to match what it can be in the real world. For example, a human's photo can be flipped horizontally or tilt to a certain angle but cannot be flipped vertically since the case is rare when a human is seen upside down.

The images in this project were treated with common sense. Since workers could be shot on either left or right side, the images in the training dataset are flipped horizontally. Furthermore, workers can be shot working on a slope, thus the images in the training dataset were randomly rotated with an angle in between 15 degrees clockwise to 15 degrees anti-clockwise. The image augmentation expanded the training dataset three times its original size. Eventually, there were 2268 images in the training dataset.

The train-valid-test split and image augmentation were produced by Roboflow. Roboflow is an online image machine learning aid tool which offers help in every stage of the object detection model development [20].

3.3 Model

3.3.1 R-CNN

Before the invention of the R-CNN method, CNN fell off the object detection stage after the rise of support vector machines in the 1990s. The reason why CNN lost its popularity was that the objects of interest might have different spatial locations and different aspect ratios, which led to an increasing number of regions that need to be selected and demanded huge computation resources [7]. In 2014, Ross Girshick and his team solved the CNN localization problems by extracting 2000 region proposals, using large CNN to compute features for each region, and classifying each region using linear support vector machines (SVMs) [8].

The advantage of the R-CNN algorithm was it was the most accurate object detection method at its time, and it achieved a 30% relative improvement towards the previous best result. However, as in the first stage of the development of the R-CNN model, it required a lot of computational resources and it was very slow to run both in training and testing stages: it was 9 times slower than the OverFeat, which was the second-best object detection method in accuracy in 2014 [8]. A ConvNet forward pass was performed individually for every object proposal without sharing computation was the main reason making R-CNN slow.

3.3.2 Fast R-CNN

In 2015, Ross Girshick came up with an idea to speed up the process of R-CNN and it was called Fast R-CNN. The entire image was input to the CNN to generate a conv feature map instead of feeding it region proposals. For each proposal, a region of interest (ROI) pooling layer extracted a fixed size feature vector and established the feature map [9]. Then, a softmax layer was used to estimate probabilities of all classes and another layer offset values for bounding boxes.

Fast R-CNN was significantly faster than the previous R-CNN method because 2000 region proposals did not need to be input into the CNN every time. Using VGG 16 dataset, the training time for Fast R-CNN is 9.5 hours, which was 9 times faster than using the old R-CNN method [9]. The testing rate for Fast R-CNN was 0.32 seconds per image, which was also much faster than using R-CNN which is 47 seconds per image [9]. The accuracy of the Fast R-CNN method did not drop because of the short training and testing time. On contrary, on the VOC 2012 test, the mean average precision (mAP) for the Fast R-CNN model is 65.7%, which is higher than the 62.4% for R-CNN model.

3.3.3 Faster R-CNN

Both R-CNN and Fast R-CNN used selective search, which was slow and time-consuming, to find the region proposals [7]. Shaoqing Ren et al. replaced the old selective search with the Region Proposal Network (RPN), which shared full-image convolutional features with the detection network and reduced the proposal computing time to 10 milliseconds per image [6]. The structure of RPN is shown in figure 8. RPN worked on the feature map generated by the last convolutional layer. An n×n size window was sliding across the feature map and generating multiple region proposals [10]. An anchor was located at the centre of the sliding window, and it was associated with a scale and aspect ratio [6]. The purpose of using anchors was to classify and regress bounding boxes with reference to anchor boxes with different scales and aspect ratios to achieve multi-scale predictions [6]. The proposed regions then went through a Fast R-CNN detector. The structure of the Faster R-CNN object detection model is shown in the figure 9.

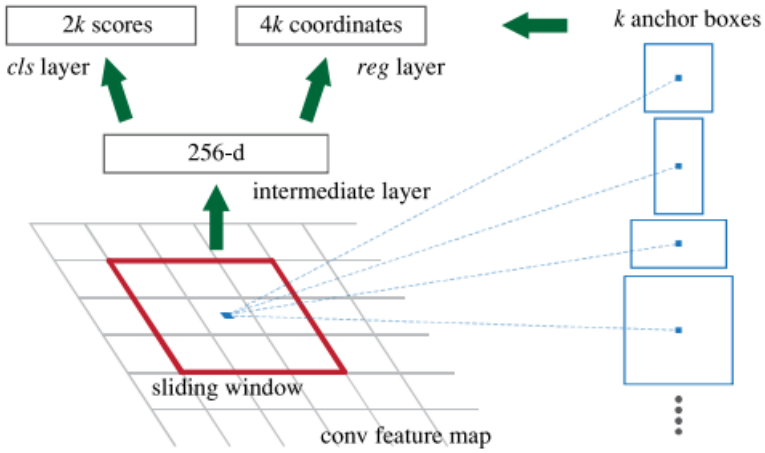

Figure 8. Region Proposal Network [6].

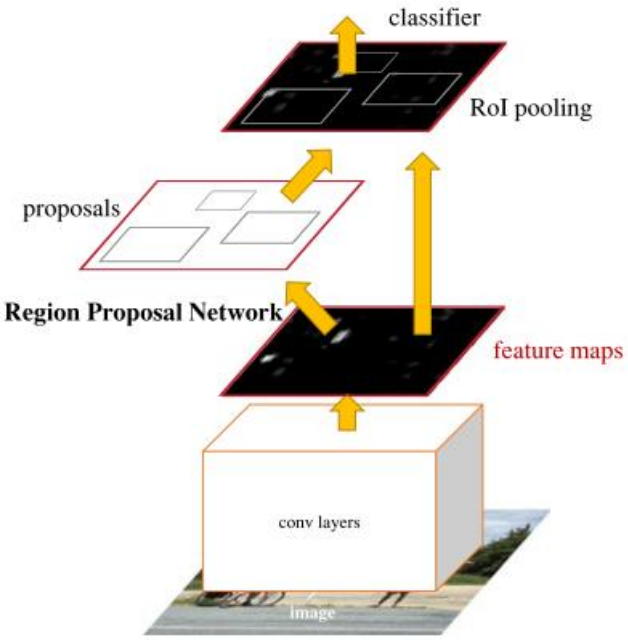

Figure 9. Faster R-CNN object detection model structure [6].

The use of Faster R-CNN on the VGG-16 dataset showed a significant improvement over the previous model. When Fast R-CNN took 320 milliseconds on the VGG-16 dataset using selective search proposals, Faster R-CNN only took 198 milliseconds on both proposal and detection [6]. Additionally, with shared convolutional features, it only took 10 milliseconds for RPN to compute additional layers. Meanwhile, the Faster R-CNN model also achieved 67% mAP on PSACAL VOC 2012 test set, while the Fast R-CNN model, which used the selective search method, only achieved 65.7% mAP.

3.4 Backbone

The backbone of the model used in this project was a 101-layer ResNeXt network. The ResNeXt network was built by repeating a building block that aggregates a set of transformations with the same topology [21]. A new dimension called "cardinality" was added to the previous ResNet network which only had two dimensions, depth and width. Cardinality not only could improve classification accuracy under restricted conditions but was also more effective than going deeper or wider when capacity increased [21].

3.5 Transfer learning

In machine learning and data mining, an important assumption is that the training data and future data will be in the same format. However, many limitations will take place and restrict the usage of the data. In this case, transfer learning can be utilised to avoid the heavy labelling cost by utilising the previous model's result [14]. Moreover, in object detection, more labelled data

always yields better results. Since there is not always enough data for researchers to conduct a new study, transfer learning in object detection enables researchers to share their model result and enrich their own studies using others' outcome. This will significantly boost the model accuracy by simply enlarging the database [13]. In this project, A pre-trained model "faster_rcnn_x101_64x4d_fpn_2x_coco" was utilised to improve model accuracy [6].

3.6 Evaluation measurements

3.6.1 Intersection over Union (IoU)

IoU was used to evaluate the performance of an object detector on a dataset. IoU was calculated using the predicted bounding boxes provided by the model, and the labelled ground-truth bounding boxes. IoU was the overlapped area of two bounding boxes divided by the union of the area of two bounding boxes. A larger IoU value meant the prediction is better.

3.6.2 Loss function

Faster R-CNN used the loss function to calculate losses at different stages of the model. loss_rpn_cls was the number of improper classifications of anchor boxes proposed by RPN. Loss_rpn_bbox was the localisation accuracy of RPN. Loss_cls was a loss that measures how "tight" the predicted bounding boxes are to the ground truth object. Loss_bbox was a loss that measures the correctness of the classification of each predicted bounding box. In this project, the loss would be simply calculated by the sum of loss_rpn_cls, loss_rpn_bbox, loss_cls, and loss_bbox.

3.6.3 Precision and recall

Precision was calculated by true positive predictions divided by all positive predictions, which included true positive and false positive. Recall was the ratio of true positive predictions to the sum of true positive predictions and false negative predictions. In general, the precision increased when recall decreased, and the precision decreased when recall increased.

3.6.4 Average precision (AP) and mean average precision (mAP)

The formula to calculate AP is shown in figure 10. The definition of AP was finding the area under the precision-recall curve. The r_n represents the n^{th} recall value, and p_{interp} represents the according precision value.

$$\mathrm{AP} = \Sigma \left(r_{n+1} - r_n \right) p_{interp}(r_{n+1})$$

Figure 10. Average precision formula.

The AP is usually calculated for one class. The mAP was calculating the mean of AP of all classes. The mAP was used to find the portion of the correct predictions in the model. Higher the mAP, better was the prediction results. The mAP was evaluated on the final set of bounding boxes at two IoU thresholds of 0.5 and 0.75. The upper bound of the IOU threshold (0.75) reflects the localization accuracy of the network while the lower bound of the IOU threshold (0.5) reflects the detection accuracy [13].

4. RESULTS

4.1 Epochs and loss convergence

An epoch in object detection means one training cycle through the whole dataset. When training a model, as the number of epochs increase, the loss is expected to be lower and converge. The model was trained with learning rate of 0.02 with epoch of 24. The loss versus epochs plot is shown in figure 11. The overall trend is that the loss was gradually decreasing as the epoch increased. After 18 epochs, the loss became stable and converged, thus the training stage was decided finished.

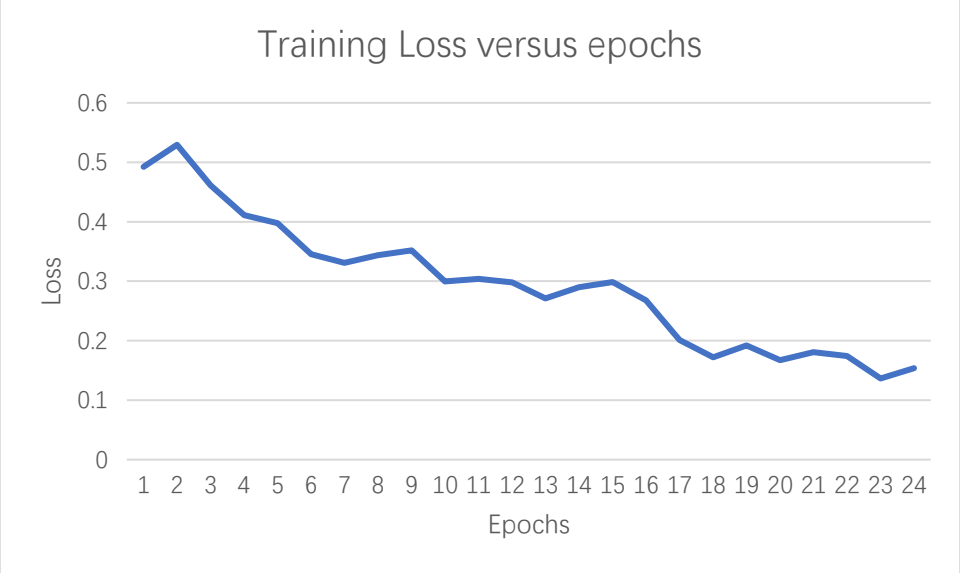

Figure 11. Training loss versus epochs.

4.2 mAP

The mAP of the prediction is shown in table 1. When IoU threshold was set to be 0.5, mAP was determined to be 0.759. When IoU threshold was set to be 0.75, mAP was determined to be 0.345. The Object was small when it had less than 1024 pixels. Medium objects had less than 9126 pixels but had more than 1024 pixels. Large objects had more than 9126 pixels. As shown in table 1, model prediction was getting worse when the object got smaller.

Table 1. mAP for various conditions

Intersection Of Union	Object Size	Maximum Detections	Mean Average Precision (mPA)
0.5:0.95	All	100	0.371
0.5	All	100	0.759
0.75	All	100	0.345
0.5:0.95	Small	100	0.322
0.5:0.95	Medium	100	0.412
0.5:0.95	Large	100	0.529

4.3 Confusion matrix

Figure 12 shows the model prediction's confusion matrix. The precision for the hard helmet is 55% and the recall is also 55%. The precision for the Hi-Vis vest is 75% and the recall is 81%. The precision for human is 75% and the recall for human is 97%. The detection for human and Hi-Vis vests performed well, and the detection for hard helmets performed poorly. Hard helmet often contain fewer pixels than human and Hi-Vis vest does, so it is hard for the model to detect.

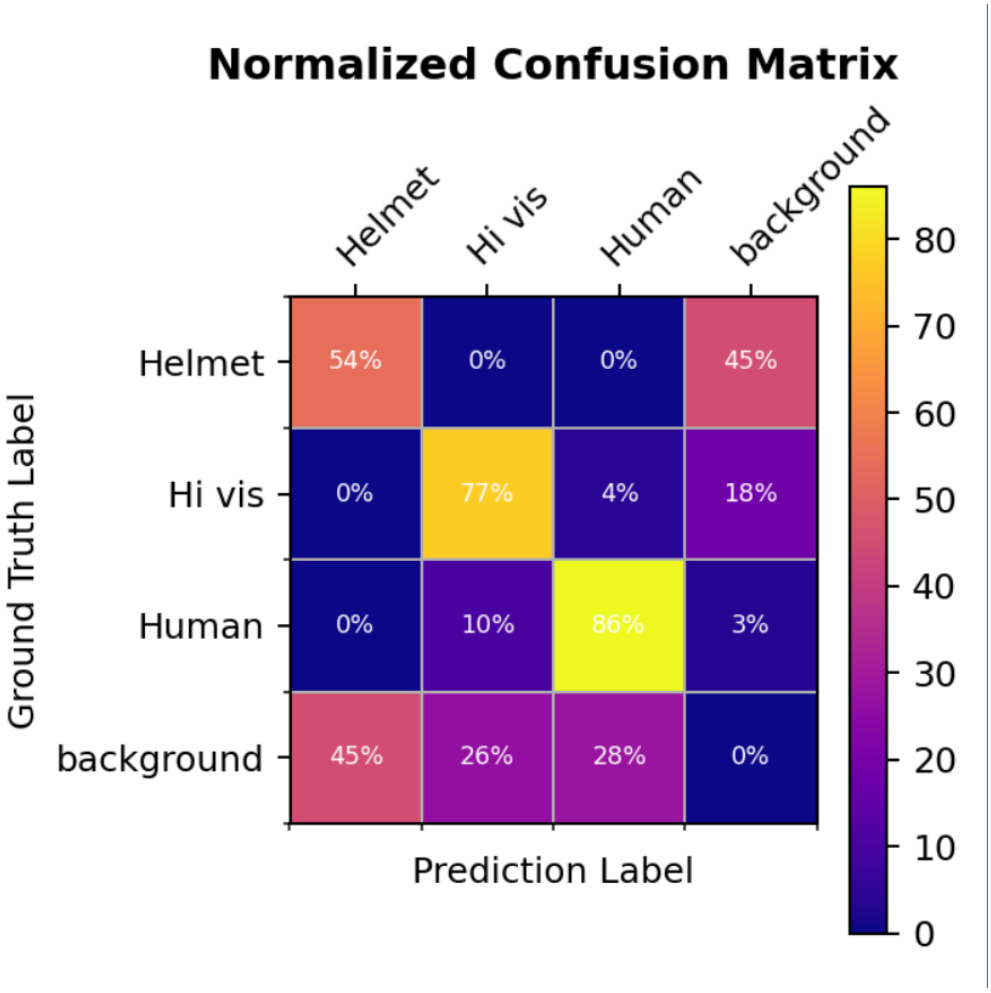

Figure 12. Confusion matrix for hard helmet, Hi-Vis vest, and human.

4.4 Prediction

Figure 13. Predictions (upper) versus ground truth (lower)

Figure 13 shows the difference between predictions and the ground truth. Overall, the model did a great job on predicting human and high vis vest. There was one hard helmet label missing in the prediction. The missing hard helmet was far away from the camera. It was small on the image and did not contain too many pixels, so it was hard for the model the detect. This explained why this model performed poorly on detecting hard helmets.

4.5 Object detection on a new image

Figure 14. Object detection on a new image [19]

Figure 14 shows the object detection model prediction on a new image. The person on the left and the person on the right were successfully detected, but the person in the middle was not. The reason could be people in the Downer dataset were all wearing orange Hi-Vis vests, which led to the situation that the model cannot recognise people who are not wearing orange Hi-Vis vests. This also explained why the orange Hi-Vis vests were detected in the image while green Hi-Vis vests were not found by the model. Lack of colour variation on hard helmets in the Downer dataset also led to no hard helmet is detected by the model.

5. SIMILAR STUDIES

Similar to this project, Nipun Nath, Amir Behzadan, and Stephanie Paal also conducted a study on using an object detection algorithm to find people and PPE on the image.

Different from using the Faster R-CNN model, Nipun Nath et al. built a deep learning model based on You-Only-Look-Once v3 (YOLOv3) architecture [3]. The models prior to YOLO purposed classifiers to detect objects on the image. These models took a classifier for the object and evaluated it at various locations and scales in a test image [15]. On contrary, YOLO purposed a simple model structure: a single convolutional layer predicted several bounding boxes and their

class probabilities. The consequence of having a simple structure was a short training and testing time. The disadvantages of using YOLO include slightly less accurate, hard to detect objects in unfamiliar aspect ratios, and incorrect localizations [16]. By using YOLOv3, Nipun Nath et al. achieved 72.3% mAP on detecting PPE and it only takes 87.7 milliseconds for processing one test image on average [3].

Other than using a different object detection model, Nipun Nath et al also used three distinct approaches to find and locate PPE and construction stuff. Figure 15 illustrates what these three approaches are. Approach one used an object detection model to find the location of the worker, Hi-Vis vests, and hard hats first. Then this location information was input to a machine learning (ML) classifier to verify PPE compliance for each work. Approach two gave the semantic definition of the classes (W, WH, WV, and WHV) and one worker was assigned to one of the classes. Approach three was similar to approach one but involved additional training of the classifier models.

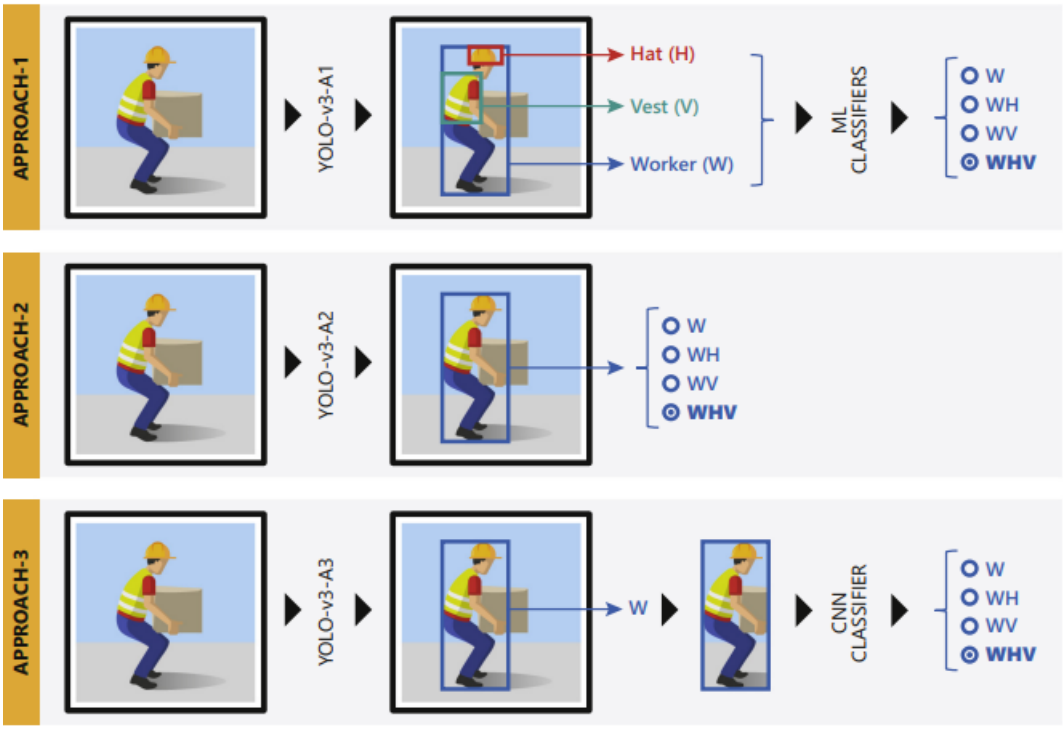

Figure 15. Three distinct approaches used by Nipun Nath et al [3].

6. FUTURE WORKS

Many future works can be done to this project on increasing the object detection model accuracy and shorten training and testing time. These improvements can be implemented through two channels, improved data quality, and object detection model improvement.

6.1 Improved data quality

The dataset given by Downer in this project was considered inadequate in many aspects: the video quality was lower than expected; the length of the video was shorter than expected; the methods used to record the video were not ideal; the video was given to the project team quite late.

Improved data quality can be easily achieved and will have a great impact on the outcome model. first of all, a better camera can be used to record the video, so the project team will receive video with better quality. The camera should be installed on top of a stationary pole, locates ideally less than 30 meters away from the working personnel, point at an angle which can monitor the whole site, and includes most of the cones or barriers in the image. The stationary monitoring camera can also record much longer than a man holding a camera, so the project team can acquire much more data to process.

6.2 Object detection model improvement

In this project, due to the time constraint, only one object detection model is trained and tested. There are many other models which have their unique advantages. Mask R-CNN can generate a segmentation mask for each object and achieves higher accuracy in predictions [17]. Cascade R-CNN composes a sequence of detectors trained with increasing IoU and results in even higher accuracy than using Mask R-CNN [18]. These state of art models are also worth experiencing to obtain a better object detection model.

7. CONCULSION

In conclusion, this project trained a Faster R-CNN object detection model on a video dataset provided by Downer. 1323 images were extracted from the given videos, and then workers and PPE were annotated on the images using VGG IA. Image augmentation was used to enrich the dataset. Dataset was then split into train, validation, and test for a Faster R-CNN model to be trained. After 24 epochs of training, the loss converges, and the model achieves a 75.9% mAP at IoU 0.5 on the testing dataset. However, the model does have some flaws, such as cannot detect people not wearing an orange Hi-Vis vest and can only detect black and white hard helmets. Future works can be done to improve the outcome model, such as improving data quality and improving the object detection model.

8. REFERENCES

[1] *Accident Compensation Corporation data*. Work Safe. *https://data.worksafe.govt.nz/focus-/construction*

[2] *Spatial and Image Learning Group.* University of Canterbury. *https://www.canterbury.ac.nz/engineering/schools/mathematics-statistics/our-research/sail/*

[3] Nath, N. D., Behzadan, A. H., & Paal, S. G. (2020). Deep learning for site safety: Real-time detection of personal protective equipment. *Automation in Construction*, 112, 103085. https://doi.org/10.1016/j.autcon.2020.103085

[4] Abhishek Dutta and Andrew Zisserman. 2019. *The VIA Annotation Software for Images, Audio and Video.* In Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3343031.3350535.

[5] Cubuk, E. D., Zoph, B., Shlens, J., & Le, Q. V. (2019). RandAugment: Practical automated data augmentation with a reduced search space.

[6] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[7] Rohith Gandhi. *R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms.* Towards Data Science. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.

[8] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation.

[9] Girshick, R. (2015). Fast R-CNN. Paper presented at the 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[10] Ahmed Gad. *Faster R-CNN Explained for Object Detection Tasks.* PaperspaceBlog. https://blog.paperspace.com/faster-r-cnn-explained-object-detection.

[11] Sik-Ho Tsang. *Review: Faster R-CNN (Object Detection).* Towards Data Science. https://towardsdatascience.com/review-faster-r-cnn-object-detection-f5685cb30202.

[12] Guo, J., Han, K., Wang, Y., Zhang, C., Yang, Z., Wu, H., Chen, X., & Xu, C. (2020). Hit-detector: Hierarchical trinity architecture search for object detection.

[13] Talukdar, J., Gupta, S., Rajpura, P. S., & Hegde, R. S. (2018). Transfer learning for object detection using state-of-the-art deep neural networks. Paper presented at the 78-83. https://doi.org/10.1109/SPIN.2018.8474198

[14] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345-1359. https://doi.org/10.1109/TKDE.2009.191

[15] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Paper presented at the 779-788. https://doi.org/10.1109/CVPR.2016.91

[16] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement.

[17] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2020). Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2), 386-397. https://doi.org/10.1109/TPAMI.2018.2844175

[18] Cai, Z., & Vasconcelos, N. (2021). Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(5), 1483-1498. https://doi.org/10.1109/TPAMI.2019.2956516

[19] *Construction*. Unsplash. https://unsplash.com/s/photos/construction.

[20] Roboflow. https://roboflow.com/

[21] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2016). Aggregated residual transformations for deep neural networks.

[22] Downer Group. https://people.downergroup.co.nz/

APPENDIX A.

Figure 1. Extracted frame

Figure 2. Extracted frame

APPENDIX B.

In [1]:
```
import numpy as np
import cv2 as cv
import os
from glob import glob
```

In [5]:
```
os.getcwd()
```

Out[5]:
```
'/home/ucsail2/safety/Images'
```

In [2]:
```
def create_dir(path):
    try:
        if not os.path.exists(path):
            os.makedirs(path)
    except OSError:
        print(f"ERROR: creating directory with name {path}")
```

In [6]:
```
def save_frame(video_path, save_dir):
    name = video_path.split("/")[-1].split(".")[0]
    save_path = os.path.join(save_dir, name)
    create_dir(save_path)

    cap = cv.VideoCapture(video_path)
    idx = 0
    skip = 10

    while True:
        ret, frame = cap.read()
        frameId = int(round(cap.get(1)))
        if ret == False:
            cap.release()
            break
        if frameId % skip == 1:
            cv.imwrite(f"{save_path}/{name}v{idx}.png", frame)
        idx += 1
```

In [7]:
```
if __name__ == "__main__":
    video_paths = glob("drone/*")
    save_dir = "drone image"

    for path in video_paths:
        save_frame(path, save_dir)
```

In [8]:
```
cap = cv.VideoCapture("videos/C0019.mp4")
ret, frame = cap.read()
```

Figure 16. Frame extraction python notebook