Temporary Traffic Control Device Detection for Road Construction Projects using Deep Learning Application

Sungchul SEO[1], Donghui CHEN[2], Kwangcheol KIM[3], Kyubyung KANG[4], Dan KOO[5] Myungjin CHAE[6] And Park, Hyung Keun[7]

[1]Graduate Research Assistant, Dept. of Civil Engineering, Chungbuk National University, Cheongju, South Korea. Email: sungchul92@nate.com
[2]Graduate Research Assistant, Dept. of Computer and Information Science, Indiana University-Purdue University Indianapolis, Indianapolis, IN. Email: dch1@iu.edu
[3]Undergraduate Research Assistant, Dept. of Mechanical Engineering, Indiana University-Purdue University Indianapolis, Indianapolis, IN. Email: kki1@iu.edu
[4]Assistant Professor, School of Construction Management Technology, Purdue University, West Lafayette, IN (corresponding author). ORCID: https://orcid.org/0000-0572-0152-9081. Email: kyukang@purdue.edu
[5]Associate Professor, Dept. of Engineering Technology, Indiana University-Purdue University Indianapolis, Indianapolis, IN. Email: dankoo@iupui.edu
[6]Assistant Professor, Dept. of Manufacturing and Construction Management, New Britain, CT. Email: chae@ccsu.edu
[7]Professor, Dept. of Civil Engineering, Chungbuk National University, Cheongju, South Korea. Email: parkhk@chungbuk.ac.kr

Keywords: Traffic Control Device, Object Detection, YOLO, Deep Learning

ABSTRACT

Traffic control devices in road construction zones play important roles, which (1) provide critical traffic-related information for the drivers, (2) prevent potential crashes near work zones, and (3) protect work crews' safety. Due to the number of devices in each site, transportation agencies have faced challenges in timely and frequently inspecting traffic control devices, including temporary devices. Deep learning applications can support these inspection processes. The first step of the inspection using deep learning is recognizing traffic control devices in the work zone. This study collected road images using vehicle-mounted cameras from various illuminance and weather conditions. Then, the study (1) labeled eight classes of temporary traffic control devices (TTCDs), (2) modified and trained a machine-learning model using the YOLOv3 algorithm, and (3) tested the detection outcomes of various TTCDs. The key finding shows that the proposed model recognized more than 98% of the temporary traffic signs correctly and approximately 81% of temporary traffic control devices correctly. The construction barricade had the lowest mean Average Precision (50%) out of eight classes. The outcomes can be used as the first step of autonomous safety inspections for road construction projects.

INTRODUCTION

Machine learning has been applied for various construction projects to enhance productivity using advanced technologies such as BIM (Bloch and Sacks 2018; Cheng et al., 2020; Kang et al., 2020; Chae et al., 2020) and to solve safety-related problems (Tixier et al., 2016; Pho et al., 2018). Recently, machine learning has been actively used for autonomous vehicle technologies and safety for road construction projects. Several object-detection studies have been conducted to recognize traffic signs, driving lanes, stop signs due to autonomous vehicle technologies. However, few studies have shed light on the recognition for traffic control devices such as signals, signs, and pavement markings.

The study on computer vision for traffic control devices has evolved for about a decade. Stallkamp et al. (2011) organized a challenging competition for traffic sign detection and summarized the results. The objective of this competition was to conduct benchmark tasks and populate data for recognizing traffic control devices in the future. The authors organized twenty teams to participate in various traffic sign recognition in Germany. The highest performed model among the participants presented 98.98% accurate recognition rates, which is better performance than the human recognition rate in the competition. Zhu et al. (2016) investigated Chinese traffic sign recognition. The team has created a new and more realistic traffic sign detection benchmark. And the team has trained two CNNs for detecting China traffic signs. The team network achieved 84% accuracy and 94% recall. And Fast R-CNN has better performance for the larger object. Arcos-Garc´ıa et al. (2018) investigated traffic sign recognition. The author said that Traffic signal detection is a critical study because traffic signal detection systems comprise key components in trendy real-world applications such as autonomous driving and driver safety and support. The study explores the properties of these object detection models modified and specially adapted for the domain of traffic signal detection. Various publicly available object detection models are pre-trained on the COCO dataset, fine-tuned on the German traffic signal detection dataset. Wu and Ranganathan (2012) proposed a practical system for road marking detection and recognition by detecting a set of points of interest (POI). The study demonstrated the recognition results under different lighting conditions, and the algorithm is sensitive to shadows. Kang et al. (2020) investigated pavement markings detection. Pavement marking has an important role in the road. However, pavement marking condition check takes a long time. This study provides an automated state analysis framework for pavement marking using machine learning technology. The proposed study is adequately used, pavement marking is accurately detected, and visibility can be analyzed to quickly identify locations with safety concerns.

This study collected images using the vehicle-mounted camera. It determined eight classes (construction cone, barrel, barricade, looper cone, end construction sign, road construction ahead sign, right lane reduction sign, and right lane closed ahead sign) to label them appropriately. Then, this research trained a model using YOLO-v3 algorithm and tested the model performance. This study shared the total loss function of the model during the training process and the mean Average Precision (mAP) rate for each class of TTCDs.

BENCHMARK

This study collected 2,050 data for temporary traffic devices such as construction cones, barrels, barricades, and various signs for traffic controls. The collected data have been annotated using the soft VoTT. This section also shows the data statistics.

Data Collection

This research has used two devices to collect data for TTCDs labeling. The first device that the team has used is a commercial dashboard camera which can be mounted to any vehicle. The device can record videos with 30 frames per second (fps) in Full High Definition (FHD). This device is also capable of recording GPS coordination and timestamps for the collected videos. It saves time and geospatial information in text file formats, and our team saved this information in the database corresponding to the videos. The second device that the team has used is a commercial camera used in the situation. This device is small enough to be mounted on any vehicle. The device can record videos with 60 fps in 4K Ultra High Definition (UHD). The device is capable of recording timestamps for the collected videos.

This study focuses on eight objects, which are a construction barrel (also known as a construction drum), a construction cone, a looper cone (typically thinner than a construction cone), a barricade, and four types of signs (end construction, road construction ahead, right lane reduction and right lane closed ahead). These objects are commonly used for many road construction projects in the United States. Although they are critical to controlling existing traffic for construction projects, there are no publicly available TTCDs training data sets for machine-learning detection. The team has collected images and videos of these objects in the State of Indiana for over a year. This research has not specified time and weather conditions to collect data, which means the data were collected from daytime and nighttime. Furthermore, the collected data set involves various weather conditions, illumination intensity, and light and shade.

Data Annotation

The team has performed two pre-processing for data labeling, which are (1) converting the extract image frames from the collected video files and (2) developing a systematic rule for data labeling. To get consistent quality of labeled data, the team converted the collected video files to image files. The team has decided to use images per 10 consecutive frames to give enough variations for angles and lights of objects and have a reasonable trade-off between the number of labeled data and labeling costs.

The second pre-processing for data labeling that the team has conducted is developing a systematic rule for labeling, consisting of three cases. First, the team collected perfect objects that were not hidden. The team has excluded a partially hidden object from another. It is important in the data annotation part. Second, the team excluded TTCDs in the opposite lane. The object in the opposite lane had severe distortions. Figure 1 presents distorted objects in an image. For example, the electric pole on the right-hand side of the image looks longer than it actually is, and it looks bent. Third, the team has only conducted labeling for TTCDs, ranging between 10 and 15 meters away from the data collection vehicle to provide clear legibility. Texts of construction signs farther than 10-15 meters were not legible by the available optical character recognition (OCR) algorithm. Data annotation of objects that are not clear reduces the accuracy of machine learning detection.

Figure 1. Examples of Distorted Objects in the Collected Image

The team used MS-VoTT (Visual Object Tagging Tool), an open-source image annotation provided by Microsoft. The team used a rectangular shape mode to make bounding boxes of eight TTCD types during data labeling. The team took three labeling steps: (1) creating a new project, resource file, video data, result file, and tags of TTCDs, (2) playing videos for data annotation, pausing every ten frames, and (3) creating bounding boxes when the object is clearly visible and legible (e.g., range between approximately 10 and 15 meters). The team had to make sure that each bounding box involves entire shapes of targeted objects and excludes untargeted objects as much as possible, which can be in the backgrounds of the targeted objects. This criterion is critical to minimize any potential noises for a good outcome of object recognition. Figure 2 presents an example of the eight object tags and labeling process.

Figure 2. Tool (VoTT) & process for data annotation.

Dataset Statistics

After discarding some of the images containing appropriate backgrounds, our new benchmark has 2,050 labeled data (objects of eight classes). 1,300 were the images collected during the daytime, and 750 images were collected during the nighttime. Out of 2,050 images, this study labeled 549 images of construction cones, 66 images of looper cones, 902 images of construction barrels, 49 images of construction barricades, 70 images of end construction signs, 49 images of road construction ahead signs, 57 images of right lane reduction signs, and 308 images of right lane

closed ahead signs. Figure 3 presents labeled construction cones, barrels, barricades, and various signs.

Figure 3. Labeled Objects (e.g., Cones, Barrels, Barricades, and Various Signs)

MODEL & EXPERIMENT

Model Demonstration of YOLOv3 Network

Object detection requires localization and classification to be performed. Object detection is divided into One-stage object detection and Two-stage object detection. Briefly introduce one-stage object detection and two-stage object detection to give pros and cons between one-state object detection and two-stage object detection. The difference between one-stage object detection and two-stage object detection is that one-stage object detection performs localization and classification simultaneously. In contrast, two-stage object detection performs localization and classification sequentially. Two-stage object detection is more accurate than one-stage object detection, and one-stage object detection is faster than two-stage object detection. Yolo is balanced in object detection speed and accuracy. YOLO has a rapid object detection of 45 frames per second and accuracy like two-stage object detection. Therefore, YOLOv3 was chosen as the TTCDs detection model in this study.

YOLO divides the input image into an SxS grid for object detection. Each grid cell has a B bounding box and a confidence score for each bounding box. Each grid cell has C conditional class probabilities. C is B*(x, y, w, h, confidence) + categories. YOLO builds a CNN network that outputs tensors with size (S, S, C). YOLOv3 consists of 24 convolutional layers and two fully connected layers where the convolutional layers extract the feature maps. The fully connected layers predict the output probability and coordinates—composed of DarkNet-53 and multi-scale prediction module. The DarkNet-53 consists of DBL blocks and residual blocks. YOLOv3 extracts three feature maps of different scales to get predictions of different scales.

Objective Function for the Training Process

YOLO should select one bounding box that best contains the detected object to predict the bounding box for each grid cell and find the loss for true positive. YOLO uses sum-squared error between prediction and ground truth to obtain three loss functions. The three loss functions are classification loss, localization loss, and confidence loss. When an object is detected, classification loss in each cell is a squared error in class conditional probabilities of each class. If the object is not detected, classification loss will be zero. During training, the YOLO optimizes the following multipart loss functions (Redmon, et al. 2016).

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Experiment Settings

Regarding the proposed TTCDs detection model, the authors trained and tested the proposed TTCDs detection model with 2,050 data points using the following computation device: Xeon W-2245 @ 3.9GHz, Nvidia GeForce RTX 2070 Super, and total memory of 128GB. The employed deep learning framework is TensorFlow. TensorFlow is one of the most popular software libraries used for machine learning tasks.

As mentioned earlier, the proposed model used 2,050 labeled object data from 742 collected images. Each image can contain multiple objects that are suitable for labeling. To enhance the testing result of the model, this study has pre-trained the model with Common Objects in Context (COCO) dataset, published by Lin et al. COCO provides a total of 81 categories of multi-object labeling, segmentation mask annotations, image captions, key-point detection, and panoptic segmentation annotations. The pre-training process with COCO does not enhance the true prediction rate of the model but minimizes potential false predictions at the end of the testing process. The study used 90% of randomly selected 2,050 object data (from 742 images) for training and 10% for true/false prediction testing, which are 205 object data.

In terms of the configuration of the training parameters, this study has set the initial learning rate as 0.0001. In general, higher learning rates allow the model to learn faster instead of reaching the final set of weights. The lower the learning rate, the more optimal or overall, the model can learn the optimal weights, but it can take much longer. Since we do not have much object data to

train and test the model, this study set a relatively low initial learning rate. The hyperparameter Epochs of gradient descent, which controls the total number of passes through the training dataset, consisted of 50. This study used Adam Optimizer as a training optimizer.

RESULT

Training Process

The training process of the neural network calculates the losses through a forward inference and then updating related parameters based on the derivative of losses to make the predictions as accurate as possible. Hence, the design and monitoring of loss functions are critical to getting the model's good training and testing results. The YOLO-v3 algorithm employs mainly three loss functions, and then it combines as one total loss function value. Three components of the total loss value are the sum of the square errors (giou_loss), confidence score (conf_loss), and binary cross-entropy loss (prob_loss). The total loss function of the model is the summation of these three loss functions. TensorFlow, the platform that this study used for training and testing, allows the users to check the loss functions of the model during training. Using this default setting, the users can see trends of specific parameters or defined metrics. Figure 4 presents the trends of the total loss function estimated based on three components during the training process. The figure shows the loss function is declined rapidly in the beginning and converges after approximately 6,000 iterations.

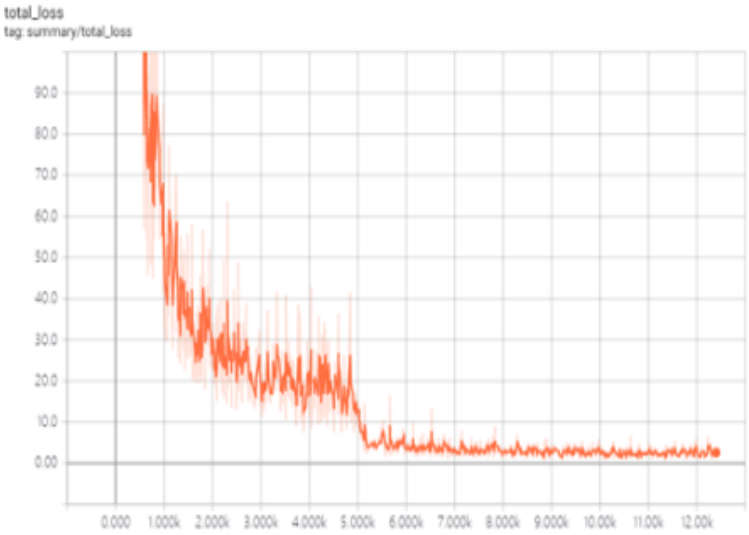

Figure 4. The trends of loss functions during the training process monitored

Evaluation Results

This study has used 94 images to recognize 205 TTCD objects and visually checked the recognized objects in the images. The test sample image is fed directly to the model as input, and the machine automatically detects and finds the object in the image. Figure 5 shows Visual results were evaluated on the testing samples.

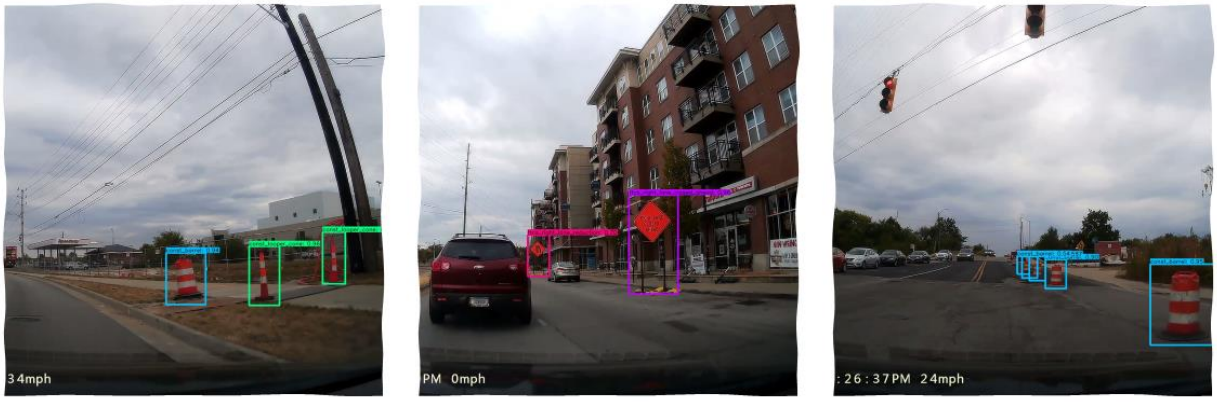

Figure 5. Visual results were evaluated on the testing samples.

Figure 6 shows the results of the quantitative validation of the detection model on the testing dataset. As shown in the top-left subfigure, 94 sample images and 205 TTCD objects are included in the evaluation dataset. The bottom-left subfigure demonstrates the number of true and false predictions upon the testing samples for each category. The red portion represents the false predictions, and the green potion refers to the true predictions. The model recognized all of the construction barrels correctly as expected. However, the model recognized 37 objects which are not construction barrels.

Similarly, the model recognized 17 objects which are not construction cones. Furthermore, the model was able to recognize only 50 construction cones correctly out of 57 cones. The model only recognized only half of the labeled construction barricades and misrecognized six objects as construction barricades.

The figure on the right-hand side presents the mean Average Precision (mAP) for each class. The overall mAP for eight classes is 90.82% - all the traffic signs except "right lane reduction" show a 100% precision rate. The right lane reduction sign consists of symbols, whereas other signs consist of texts. This study presumed that the mAP of the right lane reduction sign is lower than other signs. The mAP of the construction cone is 86% which is lower than the average. One of the reasons it has lower mAP than other objects is that many cones are recognized in the condition of overlapped with other construction cones. One other interesting feature is the construction barricade. It only has 50% of mAP. The model used a lower number of labeled data to train the model for construction barricades.

Furthermore, most barricades are laid along with driving directions. It caused large distortions in the shapes of barricades in the images. Overall, traffic control signs show higher mAP than traffic control devices such as construction cones and barricades.

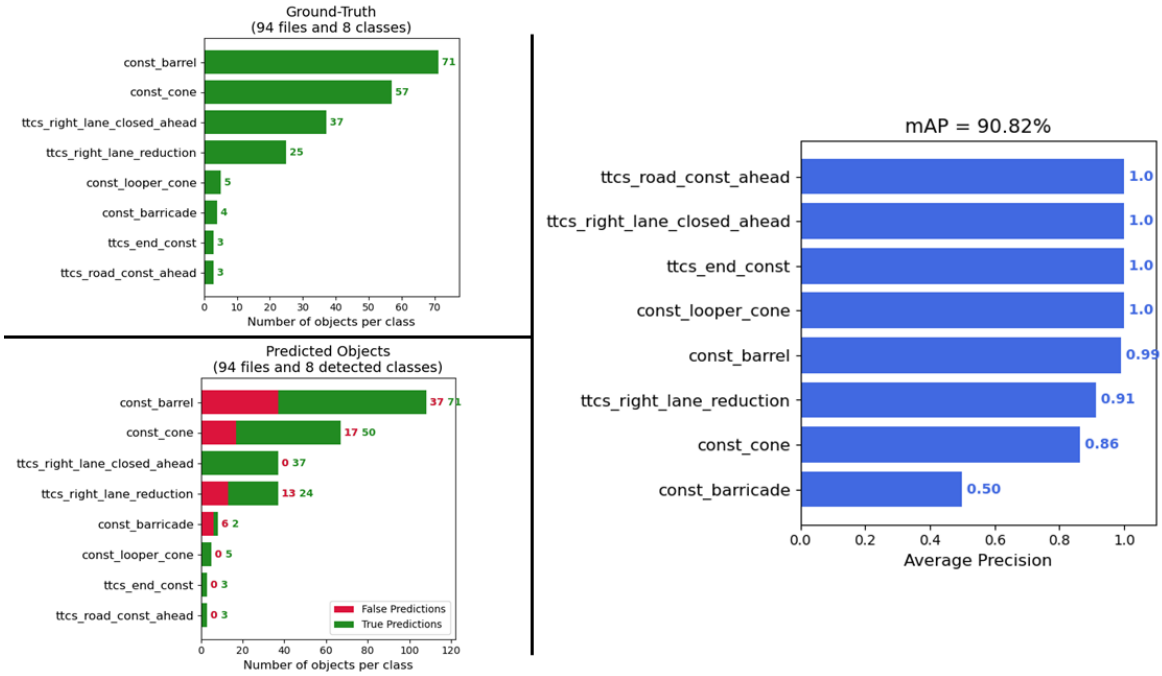

Figure 6. The quantitative evaluation information on the testing dataset of the trained model.

CONCLUSION

This study demonstrated an application of object detection for temporary traffic control devices (TTCDs) by creating training data sets and modifying a state-of-the-art object detection algorithm. The performance of the model has been validated based on the collected and labeled data for eight classes of TTCDs in the United States. In the quantitative results of the experimental section, the precision of the TTCDs detection module was significantly higher, and the YOLO-v3 detection effect was sufficiently verified. Meanwhile, visual results show that more than 90% of targetted TTCDs are correctly detected in bounding boxes and classified as attached text in road scene images. The construction barricade mAP from the test samples is relatively low. The reason is seen as the shape characteristics of the barricades. The construction cones, looper cones, and barrels have no change in shape depending on the angle compared to the barricades. And the signs are always front-facing by vehicle standards. The sign has no change in shape depending on the angle. For this reason, the barricade has a significantly lower mAP than the seven objects, except for the right lane reduction sign. The other three signs (end construction, road construction ahead, and right lane closed ahead) have a high mAP of 100%. The right lane reduction sign has a higher mAP than the construction cone and barricade, but a lower mAP than the other three signs, and the right lane reduction sign is showing 13 false predictions. For this reason, the other three signs are text-type, and the right lane reduction sign is symbol-type, and the object in text-type has a high recognition rate as a unified symbol. TTCDs play important roles, which are but are not limited to (1) provide critical traffic-related information for the drivers, (2) prevent potential crashes near work zones, and (3) protect work crews' safety. Proper and frequent inspection of TTCDs is the key to the success to play these roles. This study demonstrated object detection of TTCDs, which is the first step of the site inspection of the road construction zone. Based on the detected TTCDs,

transportation agencies can create 3D models of objects using consecutive images for visual inspection of each TTCD in the work zones and construct the digital twin of the road construction zone for analyzing the as-is condition of the work zone.

REFERENCES

Arcos-Garcia, A., Alvarez-Garcia, J. A., & Soria-Morillo, L. M. (2018). "Evaluation of deep neural networks for traffic sign detection systems." *Neurocomputing*, *316*, 332-344.

Bloch, T., & Sacks, R. (2018). Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. *Automation in Construction*, *91*, 256-272.

Cheng, J. C., Chen, W., Chen, K., & Wang, Q. (2020). "Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms." *Automation in Construction*, *112*, 103087.

Chae, M., Kang, K., Koo, D., Oh, S., & Chun, J. Y. (2020). "Fuzzy Controller Algorithm for Automated HVAC Control." In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (Vol. 37, pp. 566-570). IAARC Publications.

Kang, K., Chen, D., Peng, C., Koo, D., Kang, T., & Kim, J. (2020). "Development of an Automated Visibility Analysis Framework for Pavement Markings Based on the Deep Learning Approach." *Remote Sensing*, *12*(22), 3837.

Kang, T., Patil, S., Kang, K., Koo, D., & Kim, J. (2020). "Rule-based scan-to-BIM mapping pipeline in the plumbing system." *Applied Sciences (Switzerland)*, *10*(21).

Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In European Conference on Computer Vision; Springer: Cham, Switzerland,2014; 8693 LNCS (PART 5); pp. 740–755.

Poh, C. Q., Ubeynarayana, C. U., & Goh, Y. M. (2018). "Safety leading indicators for construction sites: A machine learning approach." *Automation in construction*, *93*, 375-386.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2011, July). "The German traffic sign recognition benchmark: a multi-class classification competition." *In The 2011 international joint conference on neural networks* (pp. 1453-1460). IEEE.

Tixier, A. J. P., Hallowell, M. R., Rajagopalan, B., & Bowman, D. (2016). "Application of machine learning to construction injury prediction." *Automation in construction*, *69*, 102-114.

Wu, T., & Ranganathan, A. (2012, June). "A practical system for road marking detection and recognition." In *2012 IEEE Intelligent Vehicles Symposium* (pp. 25-30). IEEE.

Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., & Hu, S. (2016). "Traffic-sign detection and classification in the wild." In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2110-2118).