Deep Learning Models for Content-Based Retrieval of Construction Visual Data

Nipun D. Nath[1] and Amir H. Behzadan, Ph.D.[2]

[1]Zachry Dept. of Civil Engineering, Texas A&M Univ., College Station, TX 77843. E-mail: nipundebnath@tamu.edu
[2]Dept. of Construction Science, Texas A&M Univ., College Station, TX 77843. E-mail: abehzadan@tamu.edu

ABSTRACT

Deep learning (DL) algorithms such as convolutional neural networks (CNNs) can assist in tasks such as content search and retrieval, image tagging and captioning, scene description, motion prediction, and language processing. This paper presents research that aims at designing and validating DL models for automated content-based retrieval of daily construction images and videos. Information retrieval from visual data is key to labor-intensive tasks such as safety inspection, crew activity logging, and work progress documentation. In order to train deep neural networks (DNNs), large repositories of high-quality annotated visual data are needed. However, generating such labeled datasets in construction is non-trivial and resource intensive, and requires specific skillset. To overcome this challenge, we present a methodology for fast object detection and tagging in visual data using DNNs trained with a relatively small dataset. Two state-of-the-art object detection algorithms, i.e., you-only-look-once (YOLO) and mask region-based CNN (a.k.a., Mask R-CNN) are investigated. Training data is obtained via web mining (the Internet) and crowdsourcing. Results show that training on data from both sources yields the best classification accuracy. Testing the model on new data reveals that the fully-tuned model can achieve a minimum mean average precision (mAP) of 79% when tested on different image subsets.

INTRODUCTION

In recent years, machine learning has become a major area of research in many disciplines that deal with data-intensive applications including but not limited to computer vision, robotics, speech and handwriting recognition, human-computer interaction (HCI), and biomedicine (Witten 2016, Nasrabadi 2007). Among several approaches to machine learning, deep learning (DL) has gained much popularity due to its ability to handle high volume of data while yielding high accuracy. A DL model trained on a large image dataset can be used for visual or semantic content search and retrieval, image tagging, and captioning, scene understanding and description, motion prediction, and language processing (Deng and Yu 2014, Schmidhuber 2015). In particular to the construction domain, analyzing jobsite imagery and videos with a DL model can assist in generating various key project documents such as progress and safety reports, request for information (RFI), and change orders with increased accuracy and timeliness.

With these motivations, in *Project PICTOR*, the authors aim at designing and validating DL models for automated content-based retrieval of daily construction images and videos. A key facilitator of this research is the exponentially increasing number of images and videos captured on a daily basis from construction sites using digital cameras, drones, and smartphones. However, a review of existing methodologies and practices reveals that in most cases, such visual data are manually sorted, analyzed for content, and ultimately archived. The majority of digital images and videos are stored only with date/time and/or geolocation tags, making the manual retrieval of a particular subset of digital media based on specific visual contents a non-trivial and extremely time-consuming task. A potential remedy to this challenge is to automatically analyze the captured scene for content followed by identifying specific objects of interest (a.k.a. object detection) using a computer algorithm. For example, the progress of steel frame erection can be quantified by comparing multiple images (or successive video frames) captured over a period of time from a building under construction. Similarly, identifying heavy equipment and workers and their spatial relationship would be very useful for monitoring productivity, safety, and planning resource allocation. When performed at a high processing rate, this approach can also cater to specific project needs such as creating and/or updating as-built information in (near) real-time, marking deviations from plans, identifying risky behaviors, and detecting impending accidents.

Within the general area of computer vision and digital data mining, and thanks in part to the availability of large labeled image datasets, recent work such as Krizhevsky et al. (2012), and Simonyan and Zisserman (2014) has made it possible to identify everyday objects/animals in digital imagery using DL and computational methods. Despite this, to the authors' best knowledge, there is a dearth of publicly available datasets containing labeled construction site imagery. In particular to the construction domain, a limited number of studies designed and validated methodologies for recognizing construction equipment and materials (Zou and Kim 2007; Brilakis and Soibelman 2008) using traditional machine learning. A major limitation of such studies is the need for careful extraction and selection of hand-crafted features that best represent the content of a particular dataset, and therefore the trained models may not necessarily be generalizable to a new dataset. In contrast, DL technique can achieve significantly better results by streamlining the feature learning process (Ding et al. 2018; Kolar et al. 2018; Siddula et al. 2016). However, most DL frameworks require large computational time for both training and testing due to the deep neural network (DNN) architecture, which could impede their applicability to real-time prediction or detection problems. In light of this, the research presented in this paper describes a methodology for building and testing a DL model trained on a relatively small dataset but capable of detecting construction objects (e.g., building, equipment, worker) in real-time and with high fidelity.

LITERATURE REVIEW

Object detection using hand-crafted features generally utilizes color information, e.g., HSV (hue, saturation, and value) color space, and/or geometric information (e.g., shapes, edges). Zou and Kim (2007) used saturation threshold to detect excavators. Brilakis and Soibelman (2008) proposed a methodology to identify shapes and detect corresponding material types (e.g., steel, concrete). Wu et al. (2009) applied Canny edge detection (Canny 1986) and watershed transformation (Beucher and Meyer 1992) to detect column edges in an image. Recent studies also applied traditional machine learning algorithms to recognize objects from hand-crafted features. Examples include but are not limited to Chi and Caldas (2011) who used Naïve Bayes (NB) and neural network (NN) classifiers to detect workers, loaders, and backhoes, as well as Han and Golparvar-Fard (2015) who applied Support Vector Machine (SVM) to classify ~20 types of construction materials. However, these methodologies require careful extraction and selection of features relevant to specific classes that might not be directly applicable to other cases (Kolar et al. 2018). DL-based methods such as convolutional neural network (CNN), on the other hand, offer a more convenient solution due to their ability to self-learn relevant and useful features in a large number of annotated images (Kolar et al. 2018). In particular, CNN has proven to achieve significantly better results in image classification as well as reduce computational time compared to traditional NN (LeCun et al. 1998). Applications of CNN model are plentiful, including handwritten digit recognition (LeCun et al. 1998), and identification of everyday objects from among a total of 1,000 different classes (Krizhevsky et al. 2012; Simonyan and Zisserman 2014). In the construction domain, Kolar et al. (2018) used CNN to identify safety guardrails, Siddula et al. (2016) used CNN coupled with the Gaussian mixture model (GMM) (Reynolds 2015) to detect objects related to roof construction, and Ding et al. (2018) combined CNN with the long short-term memory (LSTM) to recognize unsafe behaviors of construction workers (e.g., climbing a ladder) in video frames. More recently, the authors have proposed a method to classify construction images using deep and transfer learning techniques (Nath et al. 2018).

For object detection problems, Region-based CNN (R-CNN) is one of the most powerful methods that can achieve excellent results (Girshick 2015). R-CNN performs object detection in two steps. First, it finds candidate regions (a.k.a. regions of interest) for objects, and then, it applies CNN to classify objects inside the proposed regions (Girshick 2015). To overcome the enormous computational burden of R-CNN, however, more efficient variants of the algorithm, e.g., Fast R-CNN (Girshick 2015) and Faster R-CNN (Ren et al. 2015), have been proposed. Another variant, namely Mask R-CNN, extends Faster R-CNN by adding an extra branch which outputs segmentation masks on top of the existing branches that output classification labels and bounding boxes (He et al. 2017). Unlike these approaches, the YOLO (You-Only-Look-Once) algorithm unifies classification and localization (i.e., prediction bounding boxes) tasks in a single neural network which allows extremely fast detection of objects (Redmon et al. 2016). Therefore, it is possible to detect objects in real-time from a live video stream by sequentially and individually processing successive frames at a rate of ~30 frames per second (FPS).

Real-time (or near real-time) detection is vital, particularly for ensuring safety and mitigating risks. For example, contact collisions can be prevented by real-time detection and localization of workers in close proximity to equipment. Therefore, the main focus of this paper is on algorithms that can perform (near) real-time object detection (i.e., Mask R-CNN and YOLO).

METHODOLOGY

The schematic diagram of the overall methodology is shown in Figure 1. As shown in this Figure, images are sourced (from the Internet via web mining and crowdsourcing). In this work, Google image search engine is used for web mining (Deng et al. 2009; Fergus et al. 2005). Images are collected from this database by searching the following keywords: 1) building under construction, 2) construction equipment, and 3) construction worker. Also, crowdsourced images are collected from a number of construction projects in China. Collectively, these images constitute PICTOR v.1 dataset. Next, Labelbox (a web-based labelling toolbox) is used to label the images within the dataset. Particularly, three type of objects, i.e., *building* (buildings under construction), *equipment* (construction equipment, e.g. trucks, excavators, and loaders), and *worker* (construction workers), are marked by drawing polygonal shapes around the boundaries of these objects (a.k.a. semantic segmentation). The average time required to annotate an image from the Internet and crowdsourced dataset is ~99 seconds and ~163 seconds, respectively. Of note, there are more instances of each class per image in the images obtained through web mining, compared to crowdsourced images. For instance, there are 1,816 instances of "worker" in 667 Internet images (2.72 workers/image), versus 1,906 instances of "worker" in 832 crowdsourced images (2.29 workers/image). Therefore, the time required for annotating Internet images is generally higher. Table 1 shows the breakdown of PICTOR v.1 dataset by the number of images containing each object type. In this Table, a single image may contain multiple object classes.

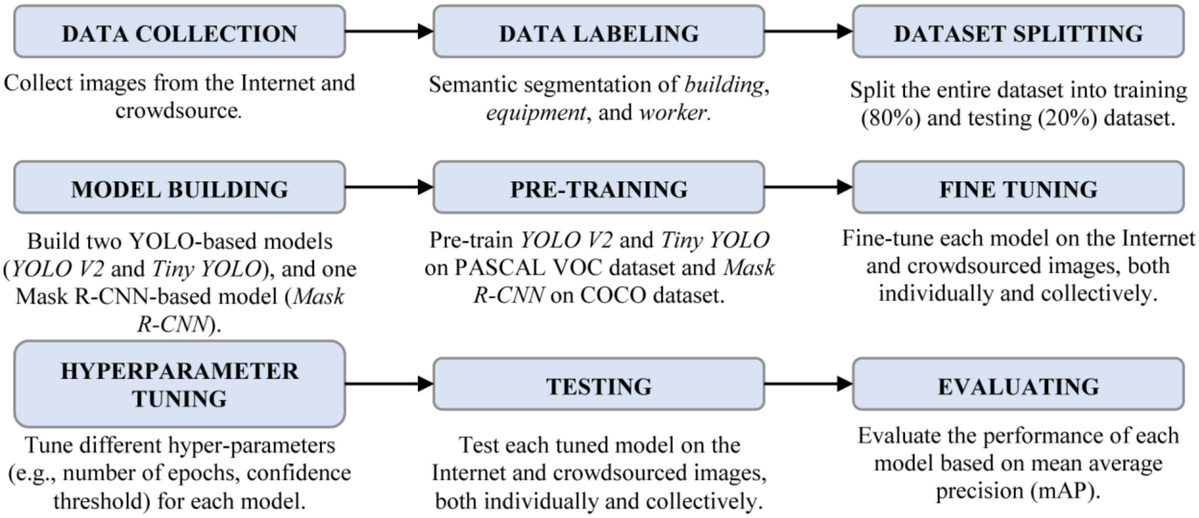
DATA COLLECTION
Collect images from the Internet and crowdsource.
DATA LABELING
Semantic segmentation of *building*, *equipment*, and *worker*.
DATASET SPLITTING
Split the entire dataset into training (80%) and testing (20%) dataset.
MODEL BUILDING
Build two YOLO-based models (*YOLO V2* and *Tiny YOLO*), and one Mask R-CNN-based model (*Mask R-CNN*).
PRE-TRAINING
Pre-train *YOLO V2* and *Tiny YOLO* on PASCAL VOC dataset and *Mask R-CNN* on COCO dataset.
FINE TUNING
Fine-tune each model on the Internet and crowdsourced images, both individually and collectively.
HYPERPARAMETER TUNING
Tune different hyper-parameters (e.g., number of epochs, confidence threshold) for each model.
TESTING
Test each tuned model on the Internet and crowdsourced images, both individually and collectively.
EVALUATING
Evaluate the performance of each model based on mean average precision (mAP).

Figure 1. Schematic diagram of the designed methodology.

Table 1. Number of sourced images in PICTOR v.1 dataset containing each object type.

	Internet		Crowdsourced	
Class	Train	Test	Train	Test
Building	319	299	324	147
Equipment	577	468	375	166
Worker	361	306	589	243
Total	1,000	862	1,000	417

Table 2. Three models and their descriptions.

Model	Algorithm	CNN Architecture	FLOP [a]	Pre-trained Dataset
YOLO V2	YOLO	23 Conv. Layers	62.94 Billion	VOC
Tiny YOLO	YOLO	9 Conv. Layers	5.41 Billion	VOC
Mask RCNN	Mask RCNN	ResNet-50	3.8 Billion	COCO

[a] Floating point operations which refers to the number of mathematical operations (e.g., addition, subtraction, multiplication, and division) performed on floating-point (real) numbers.

For a fair comparison between each source (Internet and crowdsourced subsets), 1,000 images from each subset are randomly selected for training and the rest is used for testing. For consistent performance, all models utilize transfer learning, i.e., learning general features via pre-training with a different but related (and larger) dataset, and learning more specific features related to the target classes via re-training some parts of the model (a.k.a. fine-tuning) with the desired (and smaller) dataset (Oquab et al. 2014; Shin et al. 2016). Particularly, three different models are created using three different architectures, namely, *YOLO V2*, *Tiny YOLO*, and *Mask R-CNN* (ResNet-50). A summary of the description of these models is given in Table 2.

The *YOLO V2* and *Tiny YOLO* models both use the YOLO algorithm for detection and are pre-trained on the Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL) Visual Object Class (VOC) dataset (Everingham et al. 2015). On the other hand, the

Mask RCNN model utilizes the Mask R-CNN algorithm and is pre-trained on Microsoft's Common Objects in Context (COCO) dataset (Lin et al. 2014). Each model is trained and tested on PICTOR v.1 subsets, both individually and collectively, and the mean average precision (mAP) is calculated to compare the performance of the models.

RESULTS AND DISCUSSION

The performance of all three models in terms of mAP is shown in Figure 2. From this Figure, it can be seen that overall, the performance of *YOLO V2* is better than the other two models. Particularly, the *YOLO V2* model trained on the web mining subset or both subsets and tested on the web mining subset can achieve 86% mAP. Of note, for comparison, the *YOLO V2* model trained on VOC-2007 and VOC-2012 training datasets has achieved 78.6% mAP when tested on the VOC-2007 validation dataset (Everingham et al. 2015). In general, with Intel Core i7-8850H CPU @ 2.60Hz (six cores), 16 GB RAM, and NVIDIA Quadro P2000 4GB GPU, the inference speed of *YOLO V2*, *Tiny YOLO*, and *Mask R-CNN* models is ~16, ~41, and ~2.5 FPS, respectively.

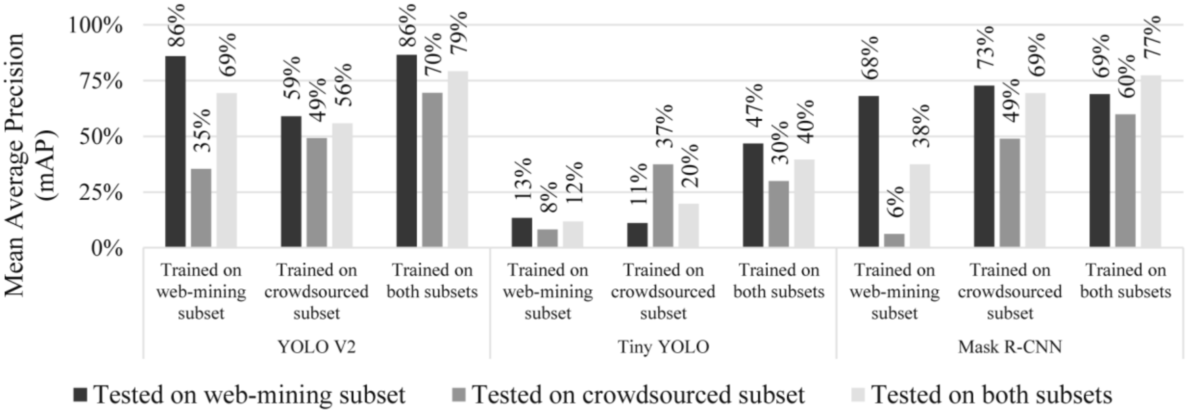

Figure 2. Performance of three models trained and tested on different datasets.

In can be observed in Figure 2 that models tested on the web mining subset generally perform better than those tested on the crowdsourced subset. This can be attributed to the fact that in general, Internet images have better quality (taken with good cameras, in well-lit conditions, without obstacles) preserving image details. On the contrary, crowdsourced images are of lower quality (taken in diverse lighting and weather conditions, containing visual obstacles). However, interestingly, models perform better when trained on images from both subsets (web mining and crowdsourced). This can be best described considering that models learn relevant and useful features from the well-structured Internet images, while diverse and challenging crowdsourced images help prevent overfitting and assist in selecting features that are more general. In short, this counteracting balance makes the models more robust. In this work, a model that can perform well on crowdsourced images is of interest since these images resemble the real-world conditions. Therefore, the *YOLO V2* model is selected for real-time construction object detection. As shown in Figure 3, this model takes a 416×416 image input and outputs rectangular bounding boxes for detected objects, each associated with a confidence level indicating the probability of the detected object inside the box. A threshold value is used to discard detections with low confidence. A lower threshold allows the model to output more detections with lower confidence, increasing false positives, and lowering detection precision.

On the contrary, a high threshold filters out potentially correct detections, increasing false negatives, and lowering detection recall. This inherent precision-recall tradeoff can be better understood from Figure 4. It is found that for a confidence threshold of 0.1, the harmonic mean of precision and recall (a.k.a. F1 score) is optimal at 28.75%.

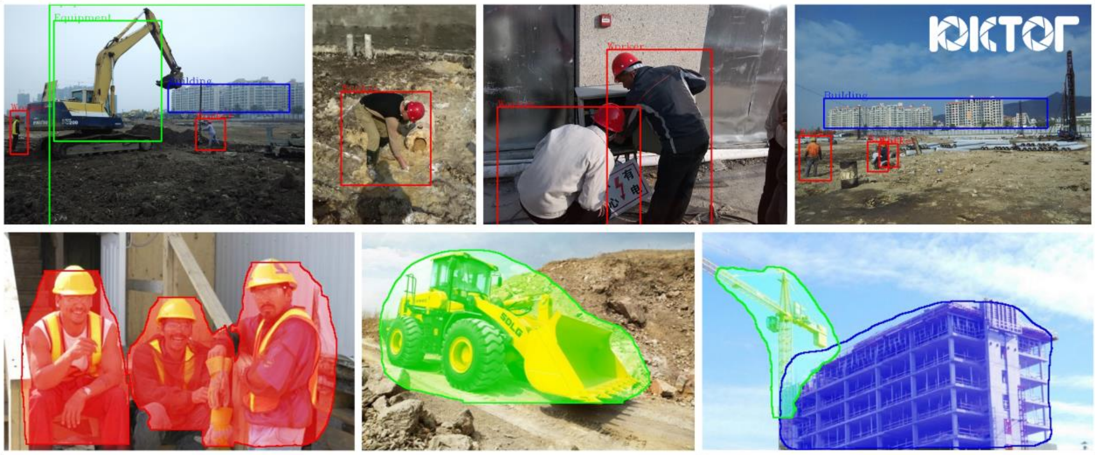

Figure 3. Detection of building (blue), equipment (green), and worker (red) by YOLO (top row) and Mask R-CNN (bottom row) algorithms.

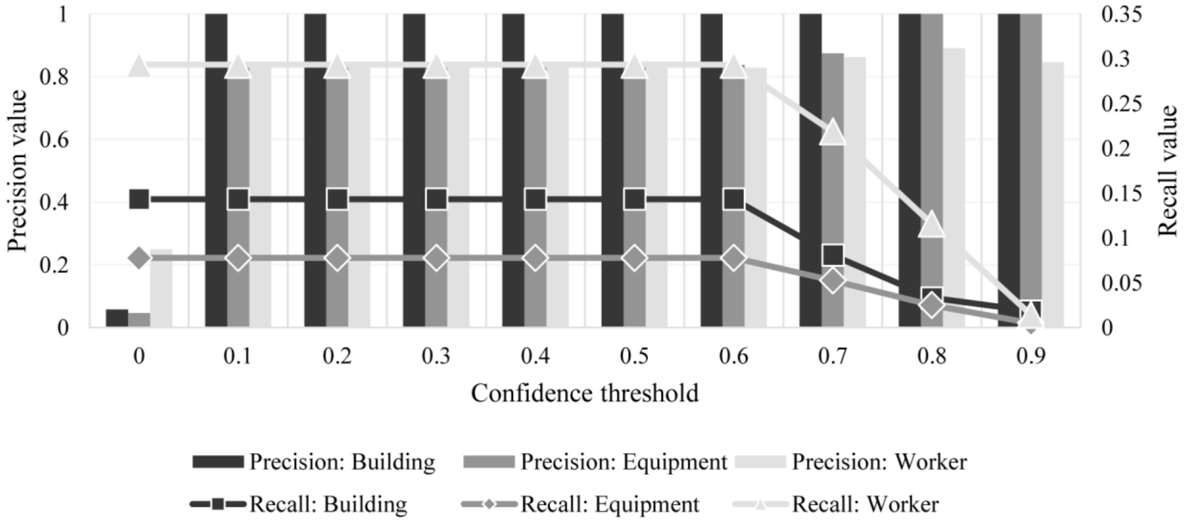

Figure 4. Precision and recall of the *YOLO V2* model.

SUMMARY AND CONCLUSION

DL algorithms can learn features and classify objects with high accuracy. However, two major challenges still exist in applying DL particularly to construction domain: (1) scarcity of large labeled datasets, and (2) high computational time. In this paper, the authors presented a DL technique to overcome these challenges by detecting construction objects with high precision in real-time. First, a large number of images were collected from the Internet (using web mining) and crowdsourcing to construct a large dataset named PICTOR v.1. Transfer learning was used to utilize the features learned from larger and more relevant datasets. Next, two algorithms, namely YOLO and Mask R-CNN were used to train and test models to detect objects in real-time. Three different models (i.e., *YOLO V2*, *Tiny YOLO*, and *Mask R-CNN*) were trained and tested on three different combinations of the PICTOR v.1 dataset (i.e., web mining and crowdsourced subsets, and the entire dataset). It was found that the *YOLO V2* model performed better, particularly, when trained on the entire dataset (images from the Internet and crowdsourcing), and was able to detect buildings, equipment, and workers in crowdsourced and web mining image subsets with 79% and 86% mAP, respectively. Therefore, this model can be used for analyzing construction images/videos and retrieving specific contents of interest.

ACKNOWLEDGMENTS

Authors gratefully acknowledge U.S. National Science Foundation (NSF) for supporting this project through grant CMMI 1800957, and Mr. Yalong Pi for assisting in data preparation. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily represent the views of the NSF or the individual named above.

REFERENCES

Beucher, S., and Meyer, F. (1992). “The morphological approach to segmentation: the watershed transformation.” *Optical Engineering-New York-Marcel Dekker Incorporated*, 34, 433-433.

Brilakis, I., and Soibelman, L. (2008). “Shape-based retrieval of construction site photographs.” J. Comput. Civ. Eng., 22, 14–20.

Canny, J. (1986). “A computational approach to edge detection.” *IEEE Transactions on pattern analysis and machine intelligence*, 6, 679-698.

Chi, S., and Caldas, C. H. (2011). “Automated object identification using optical video cameras on construction sites.” Comput.-Aided Civ. Infrastruct. Eng., 26, 368–380.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). “ImageNet: A large-scale hierarchical image database.” Proc., IEEE Conference on Computer Vision and Pattern Recognition, 248–255.

Deng, L. and Yu, D. (2014). “Deep learning: methods and applications.” *Foundations and Trends in Signal Processing*, 7(3–4), 197-387.

Ding, L., Fang, W., Luo, H., Love, P. E., Zhong, B., and Ouyang, X. (2018). “A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory.” Autom. Constr., 86, 118–124.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). “The pascal visual object classes challenge: A retrospective.” Int. J. Computer Vision, 111(1), 98-136.

Fergus, R., Fei-Fei, L., Perona, P., and Zisserman, A. (2005). “Learning object categories from Google’s image search.” Proc., 10th IEEE International Conference on Computer Vision, 1816–1823.

Girshick, R. (2015). “Fast R-CNN.” *Proc., IEEE International Conference on Computer Vision*, 1440-1448.

Han, K. K., and Golparvar-Fard, M. (2015). “Appearance-based material classification for monitoring of operation-level construction progress using 4D BIM and site photologs.” Autom. Constr., 53, 44–57.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). “Mask R-CNN.” *Proc., IEEE International Conference on Computer Vision*, 2980-2988.

Kolar, Z., Chen, H., and Luo, X. (2018). “Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images.” Autom. Constr., 89, 58–70.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). “ImageNet classification with deep convolutional neural networks." Proc., Advances in Neural Information Processing Systems, 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). "Gradient-based learning applied to document recognition." Proc., IEEE, 2278–2324.

Lin, T-Y, Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., (2014). "Microsoft COCO: Common objects in context." *Proc., European Conference on Computer Vision*, 740-755.

Nasrabadi, N. M. (2007). "Pattern recognition and machine learning." *Journal of electronic imaging*, 16(4), 049901.

Nath, N., Chaspari, T., Behzadan, A. H. (2018). "A Transfer Learning Method for Deep Neural Network Annotation of Construction Site Imagery." *Proc., 18th Int. Conf. on Construction Applications of Virtual Reality*, Auckland, New Zealand, 1-10.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). "Learning and transferring mid-level image representations using convolutional neural networks." Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 1717–1724.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: Unified, real-time object detection." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 779-788.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). "Faster R-CNN: Towards real-time object detection with region proposal networks." *Proc., Advances in Neural Information Processing Systems*, 91-99.

Reynolds, D. (2015). "Gaussian mixture models." *Encyclopedia of biometrics*, 827-832.

Schmidhuber, J. (2015). "Deep learning in neural networks: An overview." *Neural networks*, 61, 85-117.

Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., and Summers, R. M. (2016). "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning." IEEE Trans. Med. Imaging, 35, 1285–1298.

Siddula, M., Dai, F., Ye, Y., and Fan, J. (2016). "Unsupervised feature learning for objects of interest detection in cluttered construction roof site images." Procedia Eng., 145, 428–435.

Simonyan, K., and Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition." ArXiv Prepr. ArXiv:1409.1556.

Witten, I.H., Frank, E., Hall, M.A. and Pal, C.J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Wu, Y., Kim, H., Kim, C., and Han, S. H. (2009). "Object recognition in construction-site images using 3D CAD-based filtering." J. Comput. Civ. Eng., 24, 56–64.

Zou, J., and Kim, H. (2007). "Using hue, saturation, and value color space for hydraulic excavator idle time analysis." J. Comput. Civ. Eng., 21, 238–246.