DEEP LEARNING-BASED DOOR AND WINDOW DETECTION FROM BUILDING FAÇADE

G. Sezen[1], M. Cakir[1], M. E. Atik[1,*], Z. Duran[1]

[1] Department of Geomatics Engineering, Istanbul Technical University (ITU), Maslak, Istanbul, 34469, Turkey, (sezeng17, cakirmer17, atikm, duranza)@itu.edu.tr

KEY WORDS: Deep learning, Building Façade Elements, Object Detection, YOLO, Faster R-CNN.

ABSTRACT:

Detecting building façade elements is a crucial problem in computer vision for image interpretation. In Building Information Modeling (BIM) studies, the detection of building façade elements has an important role. BIM is a tool that allows maintaining a digital representation of all aspects of building information; therefore, it will enable the storage of almost any data related to a given structure, regarding its geometric and non-geometric aspects. Façade segmentation was first studied in the 1970s using hand-crafted expertise. Later, detection and segmentation studies emerged based on shapes of objects and parametric rules. With the developing technology, deep learning approaches in object detection studies have intensified. It is obvious that the desired analyses can be performed faster with deep learning approaches. However, deep learning methods require large training data. Algorithms that consider different situations and are suitable for real-world scenarios continue to be developed. The need in this direction continues in the literature. In this study, door and window detection was carried out with deep learning on an original data set. The algorithms used are YOLOv3, YOLOv4, YOLOv5, and Faster R-CNN. Precision, recall and mean average precision (mAP) are used as evaluation metrics. As a result of the study, precision, recall, and mAP values with YOLOv5 were obtained as 0.85, 0.72, and 0.79, respectively. With Faster R-CNN with the lowest performance, precision, recall, and mAP were obtained as 0.54, 0.63, and 0.54, respectively.

1. INTRODUCTION

With advances in technology, it is now possible to efficiently use and analyze a wide range of data sources and develop new methods for image interpretation, geo-information extraction, and processing (Donmez and Ipbuker, 2018). The development of decision-support systems, especially for crowded urban areas, is directly related to data processing and geo-information. One of the essential structures in urban areas is buildings and building facade elements. The detection of building façade elements is critical for detecting façade faults and reconstructing street scenes for sustainable city development (Zhang et al., 2022). Building information modeling (BIM) technology creates a virtual representation of a building called a building information model. BIM is a tool that allows maintaining a digital representation of all aspects of building information; therefore, it will enable the storage of almost any data related to a given structure regarding both its geometric and non-geometric aspects (Macher et al., 2021). BIM models can be utilized for facility planning, design, construction, operation, and design to help architects, engineers, and builders. (Azhar, 2011). Visualization is an essential step for BIM. For this purpose, the details on the building must be extracted correctly. Nowadays, detail extraction methods are changing from digitizing to image processing techniques with data growth. Thus, much data can be processed more accurately and in less time.

In recent years, deep learning techniques have commonly used in object detection and detail extraction. CNN-based methods have successful results in object detection (Atik and Ipbuker, 2021). CNNs are computer systems that implement the learning ability, which is the fundamental function of the human brain. The visual cortex in biology was the source of inspiration (Cepni et al., 2020). Especially with object detection algorithms, studies on the extraction of façade elements are increasing. Accordingly, the need for a data set arises in addition to the appropriate algorithm. Identifying suitable algorithms and the lack of data sets are still important problems.

In this study, an experiment is presented extraction of windows and doors from the collected building façade images with several deep learning approaches. Additionally, a new data set created from public data is presented. Within the scope of the study, many different neural network libraries and different algorithms using deep convolutional neural networks were used. YOLO v3, YOLOv4, YOLOv5 and Faster R-CNN algorithms were compared in terms of their performance in the extraction of building façade elements using the data set.

2. LITERATURE REVIEW

Façade segmentation was first studied in the 1970s (Ohta et al., 1978), and since then, more attention has been dedicated to this area in order to achieve high accuracy. Several methods make substantial use of hand-crafted expertise, which has proven effective in achieving satisfactory outcomes in façade element detection. In the early times, there were studies on façade element detection or segmentation, which mostly used approaches based on the shapes of objects and parametric rules (Zhang et al., 2022).

With the developing technology, the usage of deep learning approaches in object detection studies has intensified. Studies have been published on not only the extraction of building façade elements, but also the detection of all kinds of objects through the image (Cepni et al., 2020; Atik et al., 2022; Atik and Ipbuker, 2020). There are both detection and segmentation studies for building façade elements. The convolutional network (ConvNet) was used by Schmiz and Mayer (2016) to create pixel-wise predictions for semantic façade-segmentation, and it performed well on the eTRIMS dataset. Dai et al. (2021) used bounding

* Corresponding author

boxes produced with U-Net for semantic segmentation and Faster R-CNN to improve object positions. The target façade elements in the study are chimney, door, window, roof, and wall. Martinovic et al. (2012) improve the façade element classification results with a three-layer approach. In the first step, objects are made holistic by merging the hyper partitioned regions with a recursive neural network. These objects are then combined by Adaboost with the channel tree classification results. Finally, objects are aligned horizontally and vertically. Liu et al. (2020) added a new loss function to the architecture of FCN-8s (Fully-Convolutional Neural Networks) to separate windows, doors and balconies in their study. In addition, a region recommendation network was created to create bounding boxes. Zhang et al. (2022) proposed a deep learning approach with a symmetric loss function to automatically detect building facade elements from images. A novel loss function is being developed to incorporate existing engineering knowledge, which can be utilized to compel the identification of highly proportional façade elements. Ma and Ma (2020) presented a study that a reliable and efficient three-stage window detection architecture based on the Faster R-CNN. An object detection branch and a bounding box localization branch are used to detect windows. Nordmark, and Ayenew (2021) presented their approach using a technique that is proposed for performing the segmentation of windows on building façades, which can separate segmentation and classification for recognizing windows in an image, using a bounding box and a partition mask to isolate window samples, especially on building façades. Using sample segmentation, individual windows can be evaluated with detailed and complete information about each pixel and specifically which window it belongs to. A method based on Mask R-CNN architecture is proposed to separate pixels in the same category into various samples. In the study by Ali et al. (2007), a novel window detection algorithm for urban areas is presented. The proposed window detection approach incorporates proper image processing and a multiscale Haar wavelet representation for determining image tiles, which is subsequently fed into a window detection cascaded classifier. The classifier is trained using a Gentle Adaboost-driven cascaded decision tree using masked data from training images and then evaluated against window-based ground truth data, which is publicly available with the original building image datasets. Recky and Leberl (2010) described a modified gradient projection approach that is capable of processing complicated historical building façades. In a single picture scenario, just one image of the inspected structure is analyzed. It is meant to process complicated façades of ancient buildings with various embellishments, arches, patterns, and divisions. They use 5 façades in their trials, which are located in several photos and have associated point clouds. Existing methodologies for window detection in ground view façade photos are summarized by Neuhausen et al. (2016). The proposed methods are assessed in terms of their general applicability to façade photographs in the urban environment. The examined methods are divided into three categories based on their principal strategies: grammar-based, image processing, and machine learning. Because the architectural style of buildings in a given area might change, and windows on a façade can be aligned unevenly, acceptable detection systems must be resistant to such structural problems.

There are many recent studies in the literature for the determination of building façade elements. In this study, a new data set was prepared and presented. In addition, experiments were carried out on the data set prepared with popular object detection methods.

3. MATERIAL AND METHODS

3.1 Dataset

A data set was prepared within the scope of the study. The data set consists of 1000 building images that are detached houses, apartments, and residences. Images were collected by taking screenshots of randomly selected buildings in randomly selected cities and streets in Turkey using the 'View street' feature of the Google Maps application (Figure 1). It is aimed to determine various types of doors and windows By choosing different types of building structures.

After the images were collected, the doors and windows on the building façades were labeled manually. Labels are exported in 'VGG JSON' and 'COCO JSON' formats.

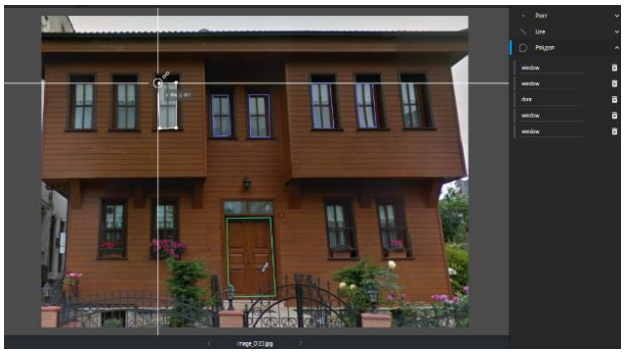

Figure 1. A sample of data labeling.

3.2 You Only Look Once (YOLO)

YOLO architecture (Redmon et al., 2016) is inspired by the GoogLeNet model and uses the Darknet framework trained on the ImageNet-1000 dataset. The model consists of 24 convolutional layers and 2 fully connected layers. Features are extracted from the image by estimating the output probabilities and coordinates of the fully connected layers. The YOLO algorithm handles the object detection problem as a regression from image pixels to bounding box coordinates and class probabilities. It is also based on only one CNN network, unlike the others. It is sufficient to look at the image once to detect an object in the relevant architecture. (Redmon and Farhadi, 2017). The general structure of YOLO architecture is presented in Figure 2.

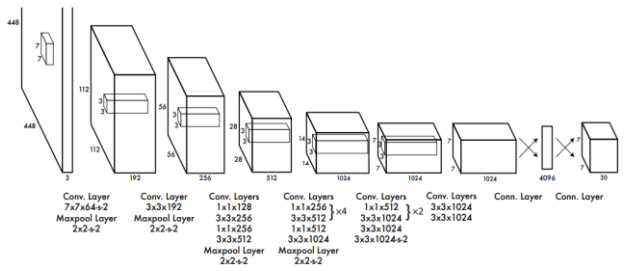

Figure 2. The general structure of YOLO architecture (Redmon et al., 2016).

3.2.1 YOLOv3: YOLOv3 estimates 4 coordinates for bounding boxes tx, ty, tw, th using dimension sets as junction boxes. The 'cx, cy' cell is the shift amounts from the top left corner of the image, and the 'pw and ph' amount is the width and height of the previous bounding box. The formulation of the estimates is as follows:

$$b_x = \sigma(t_x) + c_x \quad (1)$$

$$b_y = \sigma(t_y) + c_y \quad (2)$$

$$b_w = p_w e^{t_w} \quad (3)$$

$$b_h = p_h e^{t_h} \quad (4)$$

Logistic classifiers are used for each class in class predictions in YOLO v3. Thus, it estimates an objectivity score for each bounding box. YOLOv3 estimates each box at three different scales, and the feature pyramid network (FPN) extracts features with a similar concept (Redmon and Farhadi, 2018).

3.2.2 YOLOv4: YOLOv4 (Bochkovskiy et al., 2020) uses the CSPDarknet53 backbone. The fact that the backbone has been developed in some ways distinguishes it from Darknet50. One of these aspects is the replacement of the GPU. The YOLOv4 backbone uses the same head as CSPDarknet53, the neck uses SPP (spatial pyramid pooling) and PAN, and the head uses the same head as YOLOv3. One of the two significant changes made is the addition of an SPP block on the spine to distinguish context features. In the second change, PANet is used instead of FPN. The accuracy of the classifier in the new model obtained was tested with the ImageNet (ILSVRC 2012 val) dataset, and the detector accuracy was tested with the MS COCO (test-dev 2017) dataset. As a result, YOLOv4's single-stage anchor-based detector has faster and more accurate technology than any alternative detector available. Widely used, the detector can be trained and used on a conventional GPU with 8-16 GB-VRAM.

3.2.3 YOLOv5: YOLOv5 (ultralytics, 2022) emerged very shortly after YOLOv4. In addition, even the fact that it is named YOLOv5 is a matter of debate since neither an official article has been published nor has it not been developed by the original founders. However, the neck and head structures used in the model are the same as the YOLOv4 model. The YOLOv5 is the fastest and most successful model among the YOLO models and is unofficially published by author Glenn Jocher. There is no official release for YOLOv5 and all code is in the repository of Ultralytics LLC, of which Gleen is the founder and CEO. There are 4 different models in the warehouse: YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. The accuracy of the models increases, respectively. The most important difference of YOLOv5 from other YOLOs is the first local version was written in Pytorch instead of Darknet. Compared to the last previous version, YOLOv4, the YOLOv5 Ultralytics environment is much easier to install in colab than the YOLOv4 Darknet environment. Moreover, they have similar formats in terms of data setup formats. The main difference is seen in the education period. In YOLOv4 Darknet, the training length takes around 14 hours. However, it reached the maximum validation evaluation at 1300 iterations in approximately 3.5 hours. Training the YOLOv5 model in 200 epochs takes 14.46 minutes.

3.3 Faster R-CNN

Faster R-CNN (Ren et al., 2015) combines the proposed region network Region Proposal Network (RPN) and Fast R-CNN models. Within the scope of Fast R- CNN, first CNN is applied to the image and then the feature map created is divided into suggestion regions. The RPN, on the other hand, acts as an attention navigator, identifying the most appropriate bounding boxes among the wide variety of scales and aspect ratios to be evaluated for object classification. In short, it tells the classifier where to look. (Fan et al., 2016). In addition, RPN brings innovation to the method by connecting Fast R-CNN directly to the sampling layer. In this method, a feature map is produced by first applying CNN to the image. After this point, the difference with Fast R-CNN emerges. After the ESA is applied, the RPN comes into play. With the RPN, the suggested regions are extracted and the estimation accuracies of the regions are calculated. Then, the bounding box suggestions from the RPN are combined with the features in the backbone feature map using the RoI pooling layer. The resulting classifier and score prediction layers are finally combined in the Fast R-CNN network. With the use of RPN in the Faster R- CNN model, the estimation time is very short. The architecture of Faster R-CNN is shown in Figure 3.

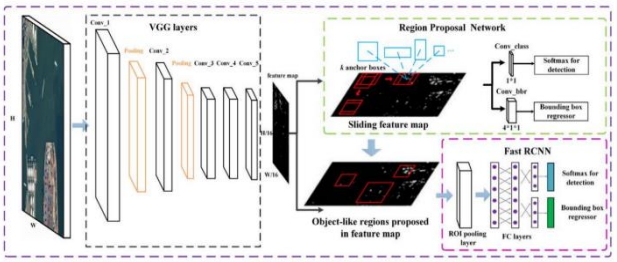

Figure 3. The architecture of Faster R-CNN (Deng et al., 2018).

3.4 Evaluation Metrics

Precision, recall, and mean average precision (mAP) are used as evaluation metrics. The fraction of points categorized as positives is measured by precision. The recall measures the fraction of true positives in a set of positives. The mean Average Precision, or mAP score, is the mean precision over all classes and/or overall IoU thresholds, depending on the numerous detection challenges that occur.

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

The number of points in predicted and ground truth that have the same label is known as true positive (TP). The term "false positive" (FP) refers to the number of points that are predicted to be positive but have a negative classification. The term false negative (FN) refers to the number of points that are predicted to be negative but have a positive label (Atik et al., 2021).

4. EXPERIMENTS

All models used within the scope of the study were carried out on the Google Colaboratory platform. Libraries such as Keras, Tensorflow, Pytorch are used during applications. The data set was entered into training as 80% training, 10% validation and 10% test data in each model. The PyTorch 1.10.0 library CUDA 11.1 was installed during training in YOLO versions. While the batch size was entered as 16, the epochs parameter was set to 100. The training period lasted approximately 5 hours. Along with Darknet, all necessary libraries, especially OPENCV 3.2.0, CUDA 11.1 and CUDNN 7.6.5, have been installed and related

parameters have been adjusted. The images were included in the training with their original dimensions. After the training, each model was tested separately on the test data. Thus, accuracy performances were evaluated under the same conditions. The training parameters were determined experimentally in an optimum way for each algorithm. The parameters are used in the study are presented in Table 1.

Parameter	YOLOv3	YOLOv4	YOLOv5	Faster R-CNN
Image size	416	416	416	416
Batch size	16	64	16	128
Iteration	6000	6000	2600	2000
Training time	5 h	8 h	1 h 21 m	2 h 56 m

Table 1. The training parameters.

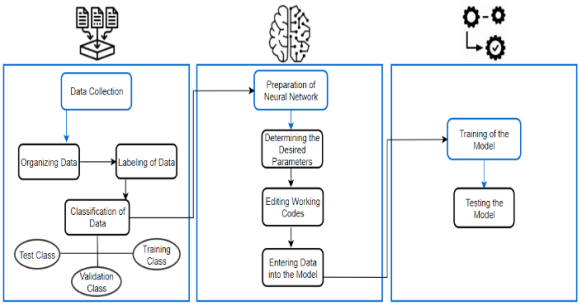

Figure 4. The workflow of the study.

5. RESULTS

Precision, recall, and mAP were used as evaluation metrics. A metric value was calculated for all window and door classes. According to the precision values, the highest value belongs to YOLOv5 with 0.85. In other words, YOLOv5 correctly detected the existing doors and windows in general. Then YOLOv4 and YOLOv3 have the precision of 0.75 and 0.68, respectively. Faster R-CNN has the lowest precision value of 0.54. In Recall, the highest values belong to YOLOv4 and YOLOv3, with 0.88 and 0.82, respectively. Faster R-CNN has the lowest recall. Considering the 0.5 confidence threshold, the mAP value is highest in YOLOv4 with 0.84. Then YOLOv5 has 0.79 mAP. The results are presented in Table 2. Examples of the results are presented in Figure 5.

Algorithm	Precision	Recall	mAP@.5
YOLOv3	0.68	0.82	0.77
YOLOv4	0.75	0.88	0.84
YOLOv5	0.85	0.72	0.79
Faster R-CNN	0.54	0.63	0.54

Table 2. The results of each model. The metrics are normalized between 0 and 1.

(a) Faster R-CNN

(b) YOLOv5

(c) YOLOv4

(d) YOLOv3

Figure 5. Result image for each model.

6. DISCUSSION

The methods generally find the same doors and windows. The difference occurs in the accuracy of the detected doors and windows. YOLOv4 appears to yield higher results than YOLOv5. The main reason for this is that YOLOv5 is trained with fewer epochs. Within the scope of the study, the best method is YOLOv5. In general, although YOLOv5 detects doors and windows, the false detection rate is higher than YOLOv4. It has been determined that YOLOv4 detects façade elements with higher reliability and less error. The lowest metrics belong to Faster R-CNN.

Considering the training times, YOLOv4 has the longest training time. YOLOv5 stands out with its speed. Considering the accuracy and speed together, it was concluded that YOLOv5 could be the appropriate algorithm for the detection of building façade elements. Although Faster R-CNN is fast, it has low accuracy metrics in this study.

While other methods detected the doors well, it was seen that they could not find the windows well. It was observed that the windows of the cars in front of the buildings were marked as building windows. It has also been frequently seen that glass doors are marked as windows. For this reason, data of building surfaces with different windows and doors should be collected

and more than one image should be obtained for each different window and door.

Apart from this, better results could be obtained by increasing the training epoch/iteration. Google Colab was used free of charge because the hardware's capacities were not sufficient. Since Google Colab stopped working after 12 hours, the number of epochs/iterations could not be increased due to the slowness of the other methods except for Yolov5. While these reasons are the limitations in front of the training, it has been seen that outstanding results will be obtained when the number of epochs is increased due to the speed of YOLOv5.

7. CONCLUSIONS

Within the scope of the study, YOLOv3/4/5 and Faster R- CNN models were trained and tested in an original window-door detection dataset. When the test data obtained as a result of the study were compared, it was observed that the fastest model was YOLOv5 and the model that gave the most optimum result for the purpose of the study was YOLOv4. For this reason, it is recommended that researchers who prioritize time in their studies should choose YOLOv5. Parameters for the appropriate version should be arranged according to the data set and purpose. Although YOLOv4 gives high accuracy, it is thought that YOLOv5 can reach accuracy of YOLOv4 as the iteration number is increased. To obtain better results, the data of the train can be increased or different façade elements can be added. From this point of view, it is predicted that in the future, deep learning and object detection studies and fields will expand by using different data sources. Images can be integrated with LiDAR to obtained 3D information. In addition, significant contributions can be made to the literature by developing appropriate deep learning approaches.

REFERENCES

Ali, H., Seifert, C., Jindal, N., Paletta, L., Paar, G. 2007: Window detection in façades. In *14th International Conference on Image Analysis and Processing (ICIAP 2007)* (pp. 837-842). IEEE.

Atik, M. E., Duran, Z., Özgünlük, R. 2022. Comparison of YOLO Versions for Object Detection from Aerial Images. *International Journal of Environment and Geoinformatics*, *9*(2), 87-93.

Atik, M. E., Duran, Z., Seker, D. Z. 2021: Machine learning-based supervised classification of point clouds using multiscale geometric features. *ISPRS International Journal of Geo-Information*, *10*(3), 187.

Atik, S. O., Ipbuker, C. 2020. Instance Segmentation Of Crowd Detection In The Camera Images. In *Proceeding of 41th Asian Conference on Remote Sensing (ACRS 2020).*

Atik, S. O., Ipbuker, C. 2021. Integrating Convolutional Neural Network and Multiresolution Segmentation for Land Cover and Land Use Mapping Using Satellite Imagery. *Applied Sciences*, *11*(12), 5551.

Azhar, S. 2011. Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry. *Leadership and Management in Engineering*, 11, 241-252.

Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. 2020: Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Cepni, S., Atik, M. E., Duran, Z. 2020. Vehicle detection using different deep learning algorithms from image sequence. *Baltic Journal of Modern Computing*, *8*(2), 347-358.

Dai, M., Ward, W. O., Meyers, G., Tingley, D. D., Mayfield, M. 2021. Residential building façade segmentation in the urban environment. *Building and Environment*, *199*, 107921.

Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., Zou, H. 2018. Multiscale object detection in remote sensing imagery with convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, *145*, 3-22.

Donmez, S.O., Ipbuker, C. 2018. Investigation on agent based models for image classification of land use and land cover maps. in *Proceedings - 39th Asian Conference on Remote Sensing: Remote Sensing Enabling Prosperity, ACRS 2018*. vol. 4, pp. 2005-2208.

Fan, Q., Brown, L.M., Smith, J.R. 2016. A closer look at Faster R-ESA for vehicle detection, *2016 IEEE Intelligent Vehicles Symposium (IV)*, DOI:10.1109/IVS.2016.7535375.

Liu, H., Xu, Y., Zhang, J., Zhu, J., Li, Y., Hoi, S. C. 2020. DeepFaçade: A deep learning approach to façade parsing with symmetric loss. *IEEE Transactions on Multimedia*, *22*(12), 3153-3165.

Ma, W., Ma, W. 2020. Deep window detection in street scenes. *KSII Transactions on Internet and Information Systems (TIIS)*, *14*(2), 855-870.

Macher, H., Roy, L., Landes, T. 2021. Automation of windows detection from geometric and radiometric information of point clouds in a scan-to-BIM process. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*., 5-9 June 2021, Nice, 43. doi.org/10.5194/isprs-archives-XLIII-B2-2021-193-2021.

Martinović, A., Mathias, M., Weissenberg, J., Gool, L. V. 2012. A three-layered approach to façade parsing. In *European conference on computer vision* (pp. 416-429). Springer, Berlin, Heidelberg.

Neuhausen M., Koch, C., König, M. 2016: Image-based Window Detection — An Overview. *Proceedings of Workshop of the European Group for Intelligent Computing in Engineering*, EG-ICE (2016), pp. 217-225.

Nordmark, N., Ayenew, M. 2021. Window Detection In Façade Imagery: A Deep Learning Approach Using Mask R-CNN. *arXiv preprint arXiv:2107.10006*.

Ohta, Y. I., Kanade, T., Sakai, T. 1978. An analysis system for scenes containing objects with substructures. In *Proceedings of the Fourth International Joint Conference on Pattern Recognitions* (pp. 752-754).

Recky, M., Leberl, F. 2010: Window detection in complex façades. In *2010 2nd European Workshop on Visual Information Processing (EUVIP)* (pp. 220-225). IEEE.

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. doi: 10.1109/CVPR.2016.91.

Redmon, J., Farhadi, A. 2017: YOLO9000: better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517-6525, DOI:10.1109/CVPR.2017.690.

Redmon, J., Farhadi, A. 2018: Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

Schmitz, M., Mayer, H. 2016. A convolutional network for semantic façade segmentation and interpretation. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *41*, 709.

ultralytics. yolov5. Available online: https://github.com/ultralytics/yolov5 (accessed on 3 April 2022).

Zhang, G., Pan, Y., Zhang, L. 2022. Deep learning for detecting building façade elements from images considering prior knowledge. *Automation in Construction*, *133*, 104016.