The development of a rebar-counting model for reinforced concrete columns: Using an unmanned aerial vehicle and deep-learning approach

Wang, S., Kim, M., Hae, H., Cao, M. and Kim, J.

This is an accepted manuscript of an article published in Journal of Construction Engineering and Management, 149 (11), p. 1–13.

The final definitive version is available online:

https://doi.org/10.1061/jcemd4.coeng-13686

The development of a rebar counting model for reinforced concrete columns: Using an unmanned aerial vehicle and deep learning approach

Abstract

Inspecting the number of rebars in each column of a reinforced concrete (RC) structure is a significant task that has to be undertaken during the rebar inspection process. Conventionally, counting the rebars has relied on a manual inspection carried out by visiting inspectors. However, this approach is very time-consuming, labour-intensive, and poses a potential safety risk. Previous studies have focused on the applications of counting the rebars for a production line and/or warehouse, using vision-based methods. Therefore, this study aims to propose an innovative approach incorporating the use of an unmanned aerial vehicle (UAV) on real construction sites to count the rebars automatically. For analysing the images, robust object detection methods based on deep learning (Faster R-CNN, R-FCN, SSD 300, SSD500, YOLOv5, and YOLOv6) were developed. A total of 384 models generated from six different methods were trained and implemented using datasets based on the original and augmented images with adjustments made for the hyperparameters. In a test, the best optimised model based on Faster R-CNN produced an accuracy of 94.61% at AP50. In addition, video testing demonstrated a coverage of up to 32 frames per second in the experimental environment, suggesting that this method has potential for real-time application.

Practical applications

Drones provide an efficient way to monitor the number of rebars in reinforced columns by capturing still images or video footage. However, manually counting the

rebars from this data in the form of images is both time-consuming and laborious. This research therefore develops an AI-driven technique, based on deep learning, designed to automate the process. In the experiment, the approach that was developed achieved an accuracy rate of 94.61% under diverse conditions on real construction sites, including non-uniform illumination and complex backgrounds (e.g., scaffolding and moulding). Nevertheless, there is potential for further improvement in certain scenarios (e.g., where there are shadows in high-illumination images, or similar objects close to the rebars). In addition, video testing demonstrated that the system could process up to 32 frames per second. Despite its limitations, the method developed in this research could be put to practical use on construction sites, except in those scenarios where it showed a lower rate of accuracy. Moreover, as 30 frames per second is often regarded as equivalent to real-time, it would also be feasible to use it for video analytics' applications such as real-time monitoring and progress tracking.

Keywords

Reinforced concrete structure; Rebar counting; Unmanned aerial vehicle; Image augmentation; Deep learning; Faster R-CNN

1. Introduction

Over the past few decades, reinforced concrete (RC) columns have been used mainly in areas of construction involving compression to support loads from ceiling, to floor slab or roof slab, or from beam to floor or foundation. The RC column mainly consists of two materials: concrete and rebar, which is short for reinforcing bar. While the concrete handles compressive and shear stress well, it performs poorly in terms of tensile strength. The rebars are therefore used to compensate for concrete's weakness by increasing its tensile strength. The combination of these materials improves resistance to bending and shear forces in buildings, has a long service life and requires minimal maintenance, making rebars a cost-effective option for many construction projects (Devine et al. 2018).

In designing and constructing RC columns, the number of rebars is regarded as the most important parameter, because it has a significant effect on uneven moment capacity and preventing structural collapse, while over-reinforcing a column could also lead to increased construction costs. In order to guarantee the necessary standards for RC columns, many countries have introduced codes setting out requirements (e.g. BS4449 (Whiteley 1997) in the UK, and ACI 318 (ACI committee: ACI 318-19 2019) in the US), and specifying the minimum, and maximum number of rebars in a column. When rebars are installed in the columns, they are assessed using 2D or 3D drawings. The inspection is generally carried out before the concrete is poured into the mould.

Conventionally, the number of rebars is assessed by a visiting inspector manually counting the rebars in each column with the naked eye. In addition, inspectors stand at specific vantage points, such as on temporary scaffolding or ladders to observe the rebars. However, this in-person approach is time-consuming, tedious, and dangerous for workers. Recently, unmanned aerial vehicles (UAVs) equipped with a single Red-

Green-Blue (RGB) camera have been introduced to resolve the problem on some construction sites. This makes the process faster and safer, by enabling workers to conduct the inspections remotely without entering potentially hazardous areas. Moreover, data in the form of RGB images is sufficient to enable the rebars to be counted accurately, so it is more cost effective than other types of cameras (e.g., a depth camera or laser scanner).

Despite the advantages that drone-based inspection offers, it does have some critical limitations. First, flying drones is a highly regulated activity on construction sites near to certain facilities (e.g. military bases and nuclear plants), due to security and safety concerns. In addition, the photography itself is limited to some extent, for example, images taken at night-time will not be very well illuminated, while blurred images may be caused by strong vibrations due to high winds. In light of aforementioned advantages and limitations, it is argued that UAVs can most effectively be used for counting rebars in RC columns when images of rebars are visible to the naked eye. However, there is one key problem: manually interpretating the images in order to count the rebars is time-consuming, and labour-intensive.

Image processing techniques are used to automatically interpret images by invoking one or more operations. All of the operations performed on images are subject to image processing techniques. The operations are usually carried out by computing devices such as smartphones, and desktop computers so that thousands of images can be processed quickly. There are many different image processing techniques, so the choice of which to use depends on the application, and the information and pattern of images will differ accordingly (Liu et al. 2016b).

Many researchers have investigated the applicability of optimal image processing techniques within the construction industry. It has been reported that automation has achieved an almost human performance level in certain applications, for example, the multi-classification of hand signals for crane operation (e.g. raising the boom and lowering the load) (Mansoor et al. 2023), detection of human intrusion (e.g. transition, and bending) in hazardous areas on construction sites (Mei et al. 2023), and crack classification on the inside of concrete structures (Chow et al. 2021). Accuracy rates of 94.8%, 96.05%, and 99%, respectively, for the aforementioned three applications have been achieved with continuous development. These studies indicate that it is worth undertaking research into image processing techniques in order for automation to be used in other applications. Thus, when suitable image processing techniques for counting the rebars in RC columns via images collected by UAVs can be successfully implemented, the inspection process can be undertaken with a high rate of accuracy and consistency.

2. Literature Review

Over the past decade, considerable efforts have been made to extend the use of various image processing techniques in the context of rebar counting. These techniques can be classified as feature-based approaches, which can also be broadly divided into two approaches: the traditional approach, and the deep learning approach. A feature (e.g., colour, or pixel intensity) refers to a characteristic that can be used to differentiate the objects in the given images. A feature extractor, which is implemented by specific operators on local portions of images, is utilised to extract the feature. The transformed feature can be then used together with operators to carry out certain tasks. The following sections explain the traditional approach and the deep learning approach within the limitations of existing research.

2.1 Traditional approach

The traditional approaches utilise manual feature extraction techniques to incorporate prior knowledge in order to undertake specific tasks. The required features in all possible variations for the scenarios are manually defined according to the judgment of human experts - usually domain experts and computer vision technicians.

The earliest piece of research, by Zhang et al. (2008), developed a steel bar counting method for use in a factory production line. First, a template matching algorithm was used to find the rebar area. The difference in pixel intensity between the two elements - rebars, and background objects - was used as a feature, which was extracted by a mutative threshold segmentation algorithm. This method was proved to have a misdetection rate of less than 0.01%. Liu and Ouyang (2018) subsequently developed a rebar counting model designed to be used in factory storage facilities. First, Otsu’s threshold was applied to distinguish between the pixel intensity of the rebars and the background, and then a contour identification method based on Suzuki's algorithm was used to detect the rebars. The results had an accuracy rate of 98%. Xin et al. (2010) proposed a rebar counting method for use on a moving conveyor belt in factories. After segmenting the rebars and the background using Otsu’s threshold, Hough transformation was applied to detect the edges of the rebars. This method was experimentally proven to count the approximate number of steel bars, although no performance quantification was provided. Zhao et al. (2016) proposed a similar approach for counting the number of steel bars passing over moving conveyer belts. Segmented images of the rebars and the background were generated using Otsu’s threshold. A Sobel operator was then applied to detect the edges of the rebars. Finally, the centre point of each bar was located using the Hough transformation. This system achieved an accuracy rate of 96% for a single frame. Su et al. (2010) also developed an automated method for counting moving rebars. The distance transform method was applied to calculate the radius of rebars. The steel bars were localised by combining the Hough circle transform with the estimated radius, and this method was shown to have a failure rate of 0.01%. Wu et al. (2015) also developed a technique for counting steel bars passing along moving conveyer belts. The binary image contour was obtained using Otsu's threshold. Each concave edge point on the rebar contour was extracted using connected area analysis. Finally, an algorithm-based fault tolerance method was applied to detect and count the rebars. This system had an accuracy rate of 99.9% at detecting steel bars with diameters between 8mm and 20mm. Liu et al. (2019) developed a bundled bars counting method to be used in factory storage facility of a factory. The Prewitt operator was applied to extract the oriented gradient as a feature. This feature was then fed into a Support Vector Machine to ascertain the exsitence of rebars. An accuracy rate of 91% on average was attained in their iterative experiments. Another image processing technique was created by Lee and Park (2019). In this study, binary images of rebars and backgrounds were processed by a model based on a random forest method. In this case, the super-pixels - groups of pixels similar in colour - were used as the feature. Another random forest model was used to classify the super-pixels to determine the presence of each rebar. The precision and recall rates were 0.99, and 0.98, respectively.

These provided traditional approaches have been applied to different areas related to rebar counting, most notably carrying out inspections during the manufacturing process in a factory (Liu et al. 2019; Su et al. 2010; Wu et al. 2015; Xiaohu and Jineng 2018; Xin et al. 2010; Zhao et al. 2016) and assisting with the material management process on outdoor construction site (Lee and Park 2019). With regard to the application of such techniques in factories, research has shown that the rebars could be accurately counted through the extracted features. However, in the case of construction sites, the accuracy rate was decreased when variations such as non-uniform lighting and different weather conditions appeared in the images. The resulting performance indicates that the features for counting the rebars accurately were not sufficiently well defined and extracted. This is a well-known disadvantage of the traditional approaches, as it very difficult to design features that are robust enough to cope with all possible variations of complex scenarios. Counting rebars from UAV images on real construction sites can be regarded as a complex scenario due to the complex background textures, varying scales of the rebars, and the irregular illumination. Therefore, a more robust method for dealing with the kind of complex scenarios involving many variations found on construction sites is required.

2.2 Deep learning approach

There is currently a growing trend to use deep learning based methods for various applications, particularly the convolutional neural network (CNN) within the computer vision field, to overcome the drawbacks of conventional methods (Chen et al. 2017). The main difference between traditional and deep learning-based methods is that the feature extractor is replaced by a CNN in the latter. Deep learning is used to automatically extract features by training a high number of trainable parameters using a large amount of images, as data driven methods.

Fan et al. (2019) developed a method that could be used to count rebars in a steel producing factory. Candidate centre points of the rebars were detected using CNN, and Distance Clustering (DC) was then applied to determine the steel bars. The accuracy rate was found to be 99.26%. A method based on deep learning for counting the number of rebars in warehouses on construction sites was proposed by Li et al. (2021). This method, called You Only Look Once (YOLO), produces bounding boxes to localise and classify rebars. Their experimental results produced an average precision and recall rate of 99.7% and 88.3 %, respectively. Zhu et al. (2020) proposed an approach that could be used to count rebars in an outside storage area on construction sites. They used a Receptive Field Block (RFB)-Feature Pyramid Networks (FPN) model to localise and classify the rebars, which produced a maximum F1 score of 98.17%. Hernández-Ruiz et al. (2021) developed a method for counting rebars inside a warehouse. CNN was used to establish whether there were rebars within cropped areas and DC was used to estimate the possible centres of the rebars. This method was able to achieve an average detection accuracy rate of 98.81% for round rebars and 98.57% for square rebars. Li and Chen (2022) applied the YOLO method to a large-scale dataset of steel pipes taking various on-site conditions into account. The experimental results obtained showed an average precision rate of 0.98.

The studies mentioned above applied different deep learning approaches to count rebars on construction sites (Fan et al. 2019; Li et al. 2021; Li and Chen 2022; Zhu et al. 2020) or in factories (Hernández-Ruiz et al. 2021), and have been shown to have a high generalisation ability. The successful application of these methods demonstrates that deep learning approaches have a strong potential for use in complex scenarios on construction sites. Consequently, the application of image processing techniques based on a deep learning approach to investigate RC inspection using UAVs during the construction stage represents a new area of research. The study also discusses and compares other widely accepted deep learning techniques in order to further demonstrate the reliability and accuracy of such methods.

3. Proposed approach

3.1 Faster R-CNN

While there may be numerous alternatives to the traditional methods, integrating object detection with deep learning techniques presents a promising avenue for counting rebars. Object detection, also referred to as object recognition, involves performing two sub-tasks simultaneously: determining the Regions of Interest (ROI) for target objects; and categorising the localised objects within an image. The Faster Region-based Convolutional Neural Network (R-CNN) method (He et al. 2016) is a type of object detection approach, which is able to localise and classify the target objects in each image through the use of bounding boxes. The original Faster R-CNN method was not developed with a specific task in mind, but rather to function as a versatile object detection algorithm. This adaptable and efficient method can be customised for different tasks by training it on appropriate datasets. Faster R-CNN has been extensively employed in numerous applications requiring the localisation of target objects and the identification of their categories.

Fang et al. (2018a) developed an automatic detection approach designed to detect construction workers' safety helmets using Faster R-CNN. The proposed method had a precision rate of 95.7% under a variety of conditions. Fang et al. (2018b) proposed a novel framework based on Faster R-CNN for detecting workers and equipment (e.g. an excavator) on construction sites. The results revealed a high level of average precision (workers: 91%; equipment: 95%) in detecting the target objects. Li et al. (2022b) applied Faster R-CNN to recognising which tasks workers were performing (e.g. whether they were straightening or transferring steel bars). The Faster R-CNN detector performed well, with an average accuracy rate of 96.54%.

The aforementioned research confirmed that Faster R-CNN is a promising method that can achieve an excellent rate of accuracy for different tasks on construction sites in complex scenarios involving objects of different scales and changes in illumination throughout the day. Each rebar can be represented as a predicted bounding box by a well-trained Faster R-CNN model, and the number of rebars can then be calculated by counting the bounding boxes. Therefore, Faster R-CNN was selected as the main method to be used for rebar counting from the UAV images in this study.

3.2 Comparative methods

A comparison with other popular methods of detection was undertaken to further demonstrate the performance of Faster R-CNN. Within the computer vision field, object detection can be broadly categorised into two approaches: one-stage detection, which uses a single deep neural network, and is known to have better detection speed, while two-stage detection, which uses two deep neural networks, including Faster R-CNN, is known to have better accuracy (Bu et al. 2022). In this research, the performance of Faster R-CNN was compared with both single-stage and two-stage methods.

The following methods were tested, as representative of one-stage detection: YOLO: YOLOv5, and YOLOv6 (Li et al. 2022a); Single Shot MultiBox Detector (SSD): SSD300, and SSD500 (Liu et al. 2016a). The primary distinction between YOLOv5 and YOLOv6 lies in the latter's adoption of a more intricate network architecture and an additional scale of anchor boxes. This enhancement enables YOLOv6 to identify smaller objects with greater precision, positioning it as a superior alternative to its predecessor in terms of both accuracy and small object detection.

Nevertheless, YOLOv6's higher computational demands may lead to slower processing speeds. SSD300 and SSD500, on the other hand, are specifically designed to process images with dimensions of 300 x 300 pixels and 500 x 500 pixels, respectively. Because of its ability to analyse higher resolution images, SSD500 outperforms SSD300 in terms of accuracy. However, its more extensive architecture necessitates greater computational resources and memory, which may result in slower performance and compatibility issues on certain devices. In summary, SSD500 surpasses SSD300 in terms of accuracy and small object detection capabilities, albeit at the cost of increased computational intensity and potential speed reduction.

The Region-based Fully Convolutional Networks (R-FCN) method was selected as being representative of two-stage methods. The main feature of this method is that it employs FCN, which use position-sensitive score maps to predict object locations and class probabilities. With position-sensitive score maps, the algorithm is potentially less accurate than Faster R-CNN, because it eliminates the region-based feature extraction step, which is considered to be a key stage in improving the accuracy of object detection (Xiao and Kang 2021).

Although the aforementioned methods typically exhibit marginally lower accuracy and faster inference times compared to Faster R-CNN, numerous studies have demonstrated their strong generalisation capabilities and reasonable inference times across various tasks (Xiao and Kang 2021). The accuracy and inference time of these methods may vary when applied to specific tasks, primarily due to the differing complexity of features associated with each task. Consequently, it is imperative to evaluate a model's effectiveness with respect to its designated purpose. Less computationally demanding approaches may prove more practical and suitable for implementation if they demonstrate adequate accuracy in tasks such as counting rebars from UAV images.

3.3 Proposed Faster R-CNN

The aim of this research is to develop a method using Faster R-CNN for automatically counting the number of longitudinal rebars in images of RC columns taken by a UAV. As shown in Figure 1, the Faster R-CNN model is a compound operation that includes two interrelated processes: the Region Proposal Network (RPN) and Fast R-CNN. These two methods perform distinct yet complementary tasks: the RPN's function is to identify prospective ROIs where the target object might be located. Fast R-CNN takes the ROIs generated by the RPN and refines them to achieve greater accuracy. It also classifies these ROIs to identify the objects they contain. In this research, some of the configurations such as the architecture and hyperparameters were modified to achieve the best performance of Faster R-CNN in terms of detecting the rebars. Moreover, its accuracy at detecting rebars was investigated using different metrics.

3.3.1 Training Faster R-CNN

The schematic architecture of RPN, is shown in Figure 2a. RPN is a key component of the Faster R-CNN architecture, responsible for generating region proposals that are likely to contain objects. The concept of anchors is used to describe the initial rectangular region proposals generated that have a variety of defined aspect ratios and scales. They are created at the centre of a spatial window, which runs over the extracted feature maps. The primary function of the RPN is to fine-tune the anchors based on ground truth information obtained through the RPN classifier and regressor.

The Fast R-CNN’s schematic architecture is depicted in Figure 2b. The Fast R-CNN consists of a feature extractor, classifier and regressor. When the RPN produces possible region proposals ranked by their abjectness scores, these proposals are sent to the Fast R-CNN model for further refinement. The first stage in this process involves pooling the features related to each region proposal. The ROI pooling layer achieves this by resizing each proposal to a fixed dimension, ensuring that they are compatible with the following fully connected layers. The classifier and regressor are utilised to classify the class probability of the detected proposal and localise each object with the comparison of ground truth annotation. As the RPN and Fast R-CNN analyse the given images by sharing the same convolutional layers, they are unified into a single network.

3.3.2 Architecture

Deep neural networks have the following limitation: as the depth of the network increases, the accuracy declines rapidly, which is known as the gradient vanishing problem. During the training process, the gradient of the earlier layers is computed by multiplying the gradient of the later layers. When the gradient of the later layers is less than one, the gradient of the earlier layers will be close to zero, resulting in the previous layer's gradient information becoming very small. To address this issue, a popular type of architecture, ResNet-101, is often employed. It maintains information about previous layers' gradients by forwarding the input to subsequent layers without alteration, making it easier for the network to retain information about earlier layer gradients. In addition, the architecture of the RPN is the same as that configured in the original paper. In the last layer of the RPN, bounding boxes consisting of four coordinates with expectation values indicating the probability that an object exists, are estimated. To detect the existence of each rebar, the number of neurons in the final layer of the classifier was altered to indicate one of two classes: rebar or background.

Generally, it takes a long time and requires a lot of data to train deep learning architecture from scratch. A pre-trained model that has already been developed in advance can be used as an alternative. After being pre-trained with one dataset, the model can utilise the features it has learned to perform various other tasks. In this research, ImageNet (Krizhevsky et al. 2012), which is designed for academic computer vision research, was used as one of the datasets for the pre-trained model. ImageNet contains over 14 million images, covering many general categories such as buildings, and people. Thus, in this research, the ResNet-101 feature extractor was initialised by pre-training it on ImageNet.

3.2.2 Optimisation of the hyperparameters

There are many tuneable hyperparameters that can be optimised to train the model. As the model performance may be altered by the use of different hyperparameters, combinations of hyperparameters are carefully optimised through experiments. However, it is almost impossible to explore all the possible configurations exhaustively, due to the constraints of time and computation resources. In this research, the following hyper parameters were chosen to optimise the deep learning model. Stochastic Gradient Descent (SGD) was used as an optimiser, which randomly sampled the batch size to update the weight of the model. In SGD, the current gradient is combined with the previous gradients multiplied by a momentum term, which is a user-defined coefficient value. The momentum values were set to 0.7

and 0.9. When the model was trained on Faster R-CNN, weight decay was used, which is a regularisation technique designed to prevent overfitting. Weight decay was applied by adding the L2 norm to the loss function for all the model weights. The L2 norm was multiplied by a factor of the weight decay parameter and the values of weight decay assigned were 0.0005 and 0.001. The learning rate refers to the step size of the weight used for updating the previous weight. The values assigned to the basic learning rate were 0.00025 and 0.0001. The batch size is a hyperparameter that controls the number of sample data in one iteration. The possible values for the batch size are limited by the amount of GPU memory available. The values were set to 1 and 2. The number of iterations has a significant impact on the precision and training time of the models. With fewer iterations, the training time may be shorter, but it may not achieve optimal accuracy, resulting in low precision. With more iterations, the accuracy can be optimised and stabilised, but it will require more time, resulting in greater use of resources. In this research, 20,000 and 30,000 were deemed appropriate values for training the model. In summary, 32 combinations of five different hyper-parameters, namely: batch size, learning rate, weight decay, momentum, and iteration, were used, as listed in Table 1.

3.3 Evaluation of model performance

In addition to the Root Mean Square Error (RMSE), which is used as the loss function for the Faster R-CNN training process, there are various other metrics that can be used to evaluate the performance of the model in object detection tasks. In this study, two aspects: average precision (AP), and detection speed, were considered.

(1) AP

In object detection, AP is widely used as a numerical metric for the evaluation of accuracy. AP is the average precision across all recall values between 0 and 1 at different thresholds of Intersection over Union (IOU), referring to the proportion of the area that is common to both the predicted and ground truth bounding boxes, relative to the total area covered by both boxes. AP is calculated using the formula shown in Equation (1), where n is the total number of bounding boxes detected, i is the rank of a specific detected bounding box, p(i) is the precision value between the first, and ith detection, and *Δr(i)* is the change in recall values between the (*i*-1)th and *i*th detection.

$$\mathrm{AP} = \sum_{i=1}^{n} p(i)\Delta r(i) \tag{1}$$

In this study, the following two IOU thresholds were set: AP50, and AP50:AP95. AP50:AP95 denotes that the mean AP ranges from 0.5 to 0.95 with an incremental step size of 0.05.

(2) Detection speed

The detection speed is the time computed by a model for a single frame. The speed is measured in FPS (Frames Per Second). One objective of measuring FPS was to ascertain how quickly the Faster R-CNN model can detect the given images.

4. Experiment

4.1 Dataset preparation

4.1.1 Original dataset

A high-precision DJI Phantom 4 Pro drone was commissioned at five unique construction sites in South Korea during peak productivity hours. Construction supervisors manually controlled the drone, thus ensuring the capture of clearly visible images of the rebars. Representative samples from the original dataset are displayed in Figure 3. The drone's path was methodically guided in a vertical trajectory above each column, positioning it directly above the rebars at an estimated altitude of 1 to 2 meters. At each position, still images were captured with the columns nearly centred in each frame. To underscore the pragmatic viability of the proposed method, rebar images were sourced under authentic operational conditions, thus encapsulating the complexities and challenges that typify bustling construction sites. This dataset contained a diverse array of variations in factors such as illumination, scale, and perspective. Moreover, other construction equipment such as scaffolding, timber and moulding were also observed in the images.

In total, 728 images of rebars with a resolution of 1,500 × 900 pixels were captured. The original dataset was divided into the training set; the validation set, which was used for selecting the best trained model; and the test set for testing the performance of the chosen model. The images were split into training (60%), validation (20%), and test (20%) subsets (436, 146, and 146 images, respectively) through random selection to ensure that each subset was representative of the original dataset.

4.1.2 Augmenting the dataset using training image augmentation methods

Because constructing the datasets can be time-consuming and tedious, image augmentation, which involves artificially expanding the training datasets was introduced. Although there are various different augmentation techniques available, the following five augmentation techniques were applied to real-world images captured by UAVs in this research. The purpose of each augmentation technique is described briefly with the parameter used. A detailed explanation with the relevant mathematical operations is described in previous research (Shorten and Khoshgoftaar 2019).

First, various images of rebars with different lighting conditions were collected. To gather a dataset that includes variations in lighting conditions, brightness augmentation can be used to artificially increase or decrease the pixel intensity of the original images to make them brighter or darker. To set the parameters for this, a pixel intensity of between -30 and 30 was randomly added. Second, the flight altitude of the UAV can cause variations in the scale of the rebars. Hence, scale augmentation can be employed to scale the images up or down by altering the original images along the coordinate axis. The original images were randomly transformed by a value of between 80% and 150% in the x and y-coordinates of their original size. Third, as blurred images caused by the strong winds were included in a dataset, the model needed to be trained to accurately recognise objects in blurred images too. To produce a blurred effect, a Gaussian filter can be applied to the original image, resulting in a more pixelated image. This process smooths the image by giving more weight to nearby pixels and less weight to more distant ones, effectively averaging the pixel values in a weighted manner. A parameter of θ for the Gaussian filter was applied randomly, ranging from 0 to 1. Furthermore, as the UAV navigates its flight path to capture images of vertical columns, its orientation relative to the columns inevitably changes. This leads to the images being taken from various angles, causing variations in the orientation of the rebars within the images. Rotation augmentation addresses this issue by rotating the original images by a specific number of degrees. In this research, the rotation angle was randomly assigned to between 5 and 90 degrees to generate a more diverse set of rotated images. In addition, UAVs capture images from different positions and altitudes as they progress along their flight path. This results in images being taken from varying viewpoints, leading to perspective distortions. Perspective transformation augmentation was applied to the images to simulate different viewpoint changes by multiplying the homography matrix with the original image's pixel coordinates. The magnitude of the perspective distortion was controlled for within the range of 0.01 to 0.15.

Based on the 436 original training images, each technique was applied separately once, and then a combination of five techniques were applied at the same time. In total, 2,180 augmented images were newly generated. Figure 4 shows examples of the augmented images produced using brightness augmentation, image smoothing, and scale augmentation.

4.1.3 Synthesis of final dataset

To demonstrate the effectiveness of the augmentation methods, four different datasets with different purposes were prepared for the experiments: training data, training data and augmented data, validation data, and test data. Next, the rebars were annotated for counting by assigning the rectangular ground-truth bounding boxes to the area of each rebar. In this case, the rebars were annotated as ‘rebar’ and the remaining areas were designated as ‘background’. Table 2 shows the detailed distribution of annotated information for each dataset.

4.3 Experimental settings

All the experiments were conducted using a Windows 10 system with an Intel Core i7-7700HQ @ 2.80 GHz×8, a NVIDIA GeForce GTX 3080ti GPU and 32G RAM. The Faster R-CNN was run with Detectron2 (Wu et al. 2019). This is a

Facebook AI Research (FAIR) software system that uses state-of-the-art deep learning algorithms. Other configurations retained the default settings of Detectron2, except for the hyperparameters, as shown in Table 1.

5. Results

5.1 Results of training and validation

Monitoring the changes in training and validation loss during the training can provide a useful indication of whether the model is converging. In addition, a model with a lower loss but a lower AP score might have a smaller overall error but could be failing to detect some objects or producing more false positives. On the other hand, a model with a higher AP score is likely to be more effective at detecting objects accurately and with fewer false positives, which is more desirable in rebar detection tasks. Therefore, when choosing the best model, the AP accuracy was prioritised over other aspects.

The training and validation loss graphs for all 64 Faster R-CNN models were plotted; however, due to space constraints, they are not displayed here, but can be accessed via a link located in the "Data Availability Statement". As representative examples, the training and validation loss graphs of the models are displayed in Figure 5. For all the models, the loss curves displayed greater fluctuation in the early stages, which then progressively smoothed out with subsequent iterations. In the original and augmented datasets with 20,000 iterations, neither the training nor the validation loss converged, meaning that they showed relatively higher fluctuations. By contrast, at 30,000 iterations, the models trained on both the original and augmented datasets exhibited convergence. However, it is noteworthy that the specific iteration point at which convergence was achieved varied between the models. A general trend for a lower validation loss was observed among the models that were trained using the augmented dataset. This pattern suggests that data augmentation effectively boosted the models' performance on the validation data.

The performances of the APs (AP50, and AP50:95) in each model are displayed in Table 3, and the graphical representation are shown in Figure 6. Specific hyperparameters do not consistently yield the best performance across all datasets. With 30,000 iterations, the models trained using augmentation techniques outperformed those trained on the original dataset in terms of AP. However, when the number of iterations was reduced, the accuracy of the models trained with augmentation techniques fell below those trained on the original dataset. This suggests that the proposed augmentation techniques can enhance the models' ability to generalise, provided the appropriate hyperparameters are used.

Regarding the overall AP performance, although all the Faster R-CNN models performed well at AP50, their accuracy dropped significantly when a stricter IOU metric of AP50:90 was applied. This indicates that most of the predicted bounding boxes were unable to accurately localise rebars with the corresponding ground truth boxes. Out of all the models generated, the highest performing model was case19 (batch size:1, learning rate: 0.00025, weight decay: 0.001, momentum: 0.7, iteration: 30,000), with the augmented dataset, which showed accuracy of 94.69%, 54.34%, and a detection speed of 0.032, at AP50 and AP50:95, respectively. As a result of the experiments, case19 was ultimately chosen to assess whether a similar level of accuracy could be achieved on real construction sites.

5.2 Model evaluation in test images

The model trained by case 19 was applied to unseen images from the test dataset. The accuracy level obtained at AP50 was 94.61%, while at AP50:95, it was 54.52%, with a detection speed of 0.032. These results were closely aligned with the performance measures achieved during validation. Figure 7 and Figure 8 show examples of visually detected images that had a relatively high level and a relatively low level of accuracy, respectively.

First, the images that were detected with a higher level of accuracy are shown in Figure 7. In Figure 7a, 24 rebars are depicted within a close-range view. This image also presents a relatively simplistic scenario, with good lighting and background including elements such as debris and equipment. All the rebars within the image were successfully detected by the model. Figure 7b features a more complex setting with dimmer illumination and a more cluttered background inclusive of elements such as scaffolding, moulding, and various other items. Nevertheless, the model adeptly detected all 32 rebars, unhampered by these more challenging conditions.

However, there were some inaccuracies as shown in Figure 8a, which displays a near-field view of 24 rebars under medium-intensity brightness. As a result of the moderately complex background that includes elements such as a hoop and a blue mark, the model encountered difficulties in recognising objects. Pipe rings situated on the floor, in close proximity to one of the rebars, caused confusion for the model, leading it to erroneously identify one of these pipe rings as a rebar. Figure 8b exhibits a scene with 18 rebars, high illumination, a shadow cast by another object, and rain-induced stains, all within a near-field view. In this image, the model failed to detect three rebars that were obscured within the shadow cast by adjacent columns. Additionally, it incorrectly interpreted a shadow of a rebar as an actual rebar. These examples underline the challenges that can arise with object detection in variable environmental conditions.

5.3 Comparison with other detectors

5.3.1 Experimental configurations

To ensure a fair comparison, the same adjustable configurations as for the Faster R-CNN model, including the hyperparameters (e.g. batch size, and iteration), and backbone architecture based on ResNet-101 were used with different datasets (original and augmented datasets) for other detectors: YOLOv5, YOLOv6, SSD300, SSD500, and R-FCN. The other features used were the same as in the corresponding original version. To accommodate the customised configurations, the codes were adjusted accordingly using the MMDetection platform (Chen et al. 2019) to run the models with all of these detectors.

5.3.2 Comparison of performance

In this experiment, 64 different models were run across six different detectors, giving a total of 384 unique models. Their respective training and validation losses, along with their performance metrics, are documented and can be accessed via a link located in the "Data Availability Statement". In summary, the loss patterns observed were akin to those detailed in Figure 5 for the Faster R-CNN model. All the models exhibited considerable volatility in their loss curves during the initial stages, gradually becoming more stable as the iterations increased. Nonetheless, none of the models reached a point of complete convergence, either with the original or augmented datasets, and showed a marked instability around the 20,000-iteration point.

Convergence was only achieved with the original dataset and the augmented version at 30,000 iterations.

A comparison of the best models using six different methods with APs, and different detection speeds is presented in Table 4. While all the models performed well in terms of detecting rebars at AP50 after being trained on a dataset of UAV images, their performance dropped significantly when a stricter IOU metric of AP50:90, was applied. This result indicates that most of the predicted bounding boxes had a lower overlap with the corresponding ground-truth boxes. Out of all the models, the proposed Faster R-CNN model had advantages in terms of accuracy. Compared to the one-stage detectors, the two-stage detectors (Faster R-CNN and R-FCN) achieved higher AP rates at all the IOU levels. Although they take longer to detect objects due to including the additional step of region proposal generation, the estimated detection speed was still reasonably good. Of the one-stage detectors, YOLOv6 performed best for the selected metrics.

5.3.2 Video testing

Video analytics offers significant benefits for applications such as real-time monitoring, and progress tracking. The best model based on Faster R-CNN was demonstrated via video testing. This model was able to process an image in a remarkable 0.032 seconds, meaning that it can effectively cover up to 32FPS. Thus, a video captured at 30FPS by a superintendent could be analysed in real time. Figure 9 shows a representative example of sequential frames processed by the Faster R-CNN model. The model correctly counted 25 rebars when the columns in the image were nearly centred, as exemplified in frames #246, #247, and #248. By contrast, when the columns were not centred (as seen in frames #348, #349, and #350), there was a noticeable decrease in accuracy. This variation in accuracy might be attributable to the characteristics of the training dataset, which was mostly captured in a vertical trajectory, with columns nearly centred in each frame. Despite this limitation, the model could still prove highly effective for counting rebars in real time, especially when the columns are well-centred within the frame.

6. Conclusions

In this paper, a novel strategy was proposed for automatically counting the rebars in an RC structure using UAVs and Faster R-CNN. In order to implement the model with a high generalisation ability, 64 models, which comprised various combinations of two different datasets and 32 hyper parameters were trained and validated. After selecting the most optimised model, its real performance using the unseen data was evaluated. In addition, the other widely accepted detectors used in many different areas of application were compared to test the performance of the proposed Faster R-CNN model. From the experimental results, the following conclusions were drawn:

(1) One of the objectives of this research was to rapidly generate synthetic data with diverse variations to enhance the accuracy of rebar counting. Utilising five augmentation techniques - brightness, scale, blurring, perspective, and rotation - the models trained on the augmented datasets demonstrated superior accuracy compared to those using the original dataset, given sufficient iterations. This methodology could also prove beneficial for similar applications within construction settings, such as deducing rebar spacing and estimating rebar diameters.

(2) Based on the most optimised model (i.e. case 19), the test results showed a 94.61%, and 54.52% level of accuracy at AP50 and AP50:95, respectively, and a detection speed of 0.032 seconds. The resulting performance produced a reliable level

of accuracy in counting the rebars in complex scenarios but still had potential for improvement in certain other scenarios (e.g. shadow in high illumination images, and similar objects close to the rebars). The most optimised model that was generated could be used for practical purposes on construction sites, except in those scenarios in which it had a low rate of accuracy.

(3) A comparison with other widely used detectors was carried out to evaluate the performance of the Faster R-CNN model. The results showed that the proposed Faster R-CNN model outperformed the following popular methods: YOLOv5, YOLOv6, SSD300, SSD500, and R-FCN, in terms of rebar counting accuracy. In addition, video testing of the Faster R-CNN model demonstrated that it had coverage of up to 32 frames per second in the experimental environment, meaning that it has considerable potential for real-time investigations.

The architecture used in this research showed promising results, but it is still a long way from achieving near-perfect accuracy. Thus, advanced architectures that incorporates additional convolutional or attention-based layers may be better at capturing the intricate patterns of rebars, leading to more accurate detection, which could be investigated in future research. In addition, the current model was shown to have limitations when the columns are not centred in the frames. Therefore, to enhance the model's robustness, the datasets should be enriched with more diverse video footage, captured in real time from a variety of construction sites and under different environmental conditions.

Data Availability Statement

All data, models, and code generated or used during the study appear in the submitted article. The dataset obtained, trained model, and test results, video output can be downloaded from Figshare data repository (Wang 2023).

References

ACI committee: ACI 318-19. 2019. *Building Code Requirements for Structural Concrete and Commentary, American Concrete Institute*. *ACI 318-19 Build. Code Requir. Struct. Concr. Comment.*

Bu, T., J. Zhu, and T. Ma. 2022. "A UAV Photography–Based Detection Method for Defective Road Marking." *J. Perform. Constr. Facil.* https://doi.org/10.1061/(asce)cf.1943-5509.0001748.

Chen, K., J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. 2019. "MMDetection: Open MMLab Detection Toolbox and Benchmark."

Chen, M., X. Shi, Y. Zhang, D. Wu, and M. Guizani. 2017. "Deep Feature Learning for Medical Image Analysis with Convolutional Autoencoder Neural Network." *IEEE Trans. Big Data*. https://doi.org/10.1109/tbdata.2017.2717439.

Chow, J. K., K. fu Liu, P. S. Tan, Z. Su, J. Wu, Z. Li, and Y. H. Wang. 2021. "Automated defect inspection of concrete structures." *Autom. Constr.* https://doi.org/10.1016/j.autcon.2021.103959.

Devine, R. D., S. M. Barbachyn, A. P. Thrall, and Y. C. Kurama. 2018. "Effect of Tripping Prefabricated Rebar Assemblies on Bar Spacing." *J. Constr. Eng. Manag.* https://doi.org/10.1061/(asce)co.1943-7862.0001559.

Fan, Z., J. Lu, B. Qiu, T. Jiang, K. An, A. N. Josephraj, and C. Wei. 2019. "Automated Steel Bar Counting and Center Localization with Convolutional Neural Networks." 1–9.

Fang, Q., H. Li, X. Luo, L. Ding, H. Luo, T. M. Rose, and W. An. 2018a. "Detecting non-hardhat-use by a deep learning method from far-field surveillance videos." *Autom. Constr.*, 85 (September 2017): 1–9. Elsevier. https://doi.org/10.1016/j.autcon.2017.09.018.

Fang, W., L. Ding, B. Zhong, P. E. D. Love, and H. Luo. 2018b. "Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach." *Adv. Eng. Informatics*. https://doi.org/10.1016/j.aei.2018.05.003.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep residual learning for image recognition." *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*

Hernández-Ruiz, A. C., J. A. Martínez-Nieto, and J. D. Buldain-Pérez. 2021. “Steel bar counting from images with machine learning.” *Electron.* https://doi.org/10.3390/electronics10040402.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. “ImageNet Classification with Deep Convolutional Neural Networks.” *Adv. Neural Inf. Process. Syst.*, 1–9. https://doi.org/http://dx.doi.org/10.1016/j.protcy.2014.09.007.

Lee, J. H., and S. O. Park. 2019. “Machine learning-based automatic reinforcing bar image analysis system in the internet of things.” *Multimed. Tools Appl.* https://doi.org/10.1007/s11042-018-5984-7.

Li, C., L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei. 2022a. “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications.”

Li, J., G. Zhou, D. Li, M. Zhang, and X. Zhao. 2022b. “Recognizing workers’ construction activities on a reinforcement processing area through the position relationship of objects detected by faster R-CNN.” *Eng. Constr. Archit. Manag.* https://doi.org/10.1108/ECAM-04-2021-0312.

Li, Y., and J. Chen. 2022. “Computer Vision–Based Counting Model for Dense Steel Pipe on Construction Sites.” *J. Constr. Eng. Manag.* https://doi.org/10.1061/(asce)co.1943-7862.0002217.

Li, Y., Y. Lu, and J. Chen. 2021. “A deep learning approach for real-time rebar counting on the construction site based on YOLOv3 detector.” *Autom. Constr.* https://doi.org/10.1016/j.autcon.2021.103602.

Liu, C., L. Zhu, and X. Zhang. 2019. “Bundled Round Bars Counting Based on Iteratively Trained SVM.” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*.

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. 2016a. “SSD: Single shot multibox detector.” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*.

Liu, Y.-F., S. Cho, B. F. Spencer, and J.-S. Fan. 2016b. “Concrete Crack Assessment Using Digital Image Processing and 3D Scene Reconstruction.” *J. Comput. Civ. Eng.* https://doi.org/10.1061/(asce)cp.1943-5487.0000446.

Mansoor, A., S. Liu, G. M. Ali, A. Bouferguene, and M. Al-Hussein. 2023. “A Deep-Learning Classification Framework for Reducing Communication Errors in Dynamic Hand Signaling for Crane Operation.” *J. Constr. Eng. Manag.* https://doi.org/10.1061/jcemd4.coeng-12811.

Mei, X., X. Zhou, F. Xu, and Z. Zhang. 2023. “Human Intrusion Detection in Static Hazardous Areas at Construction Sites: Deep Learning–Based Method.” *J. Constr. Eng. Manag.* https://doi.org/10.1061/(asce)co.1943-7862.0002409.

Shorten, C., and T. M. Khoshgoftaar. 2019. “A survey on Image Data Augmentation for Deep Learning.” *J. Big Data*. https://doi.org/10.1186/s40537-019-0197-0.

Su, Z., K. Fang, Z. Peng, and Z. Feng. 2010. “Rebar automatically counting on the product line." *Proc. 2010 IEEE Int. Conf. Prog. Informatics Comput. PIC 2010*.

Wang, S. 2023. "Rebar counting detection using object detection based on deep learning." *Figshare Data Repos.* https://doi.org/10.6084/m9.figshare.23633703.

Whiteley, J. 1997. "BS 4449 and European standards." *Concr.*

Wu, Y., A. Kirillov, F. Massa, W.-Y. Lo, R. Girshick, U. Wu, A. Kirillov, F. Massa, Wan-YenLo, and R. Girshick. 2019. "Detectron2." *https://github.com/facebookresearch/detectron2*.

Wu, Y., X. Zhou, and Y. Zhang. 2015. "Steel bars counting and splitting method based on machine vision." *2015 IEEE Int. Conf. Cyber Technol. Autom. Control Intell. Syst. IEEE-CYBER 2015*.

Xiao, B., and S.-C. Kang. 2021. "Development of an Image Data Set of Construction Machines for Deep Learning Object Detection." *J. Comput. Civ. Eng.* https://doi.org/10.1061/(asce)cp.1943-5487.0000945.

Xiaohu, L., and O. Jineng. 2018. "Research on steel bar detection and counting method based on contours." *2018 Int. Conf. Electron. Technol.*, 294–297. IEEE.

Xin, Y., W. Xue, P. X. Yuan, Q. Da Han, and C. H. Chen. 2010. "Research on an automatic counting method for steel bars' image." *Proc. - Int. Conf. Electr. Control Eng. ICECE 2010*.

Zhang, D., Z. Xie, and C. Wang. 2008. "Bar section image enhancement and positioning method in on-line steel bar counting and automatic separating system." *Proc. - 1st Int. Congr. Image Signal Process. CISP 2008*.

Zhao, J., X. Xia, H. Wang, and S. Kong. 2016. "Design of real-time steel bars recognition system based on machine vision." *Proc. - 2016 8th Int. Conf. Intell. Human-Machine Syst. Cybern. IHMSC 2016*.

Zhu, Y., C. Tang, H. Liu, and P. Huang. 2020. "End-Face Localization and Segmentation of Steel Bar Based on Convolution Neural Network." *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.2989300.

Tables

Table 1 Combinations of hyperparameters used in this research

Case	Hyperparameters				
	Batch size	Learning rate	Weight decay	Momentum	Iteration
1	1	0.00025	0.0005	0.7	20,000
2	1	0.00025	0.0005	0.9	20,000
3	1	0.00025	0.0001	0.9	20,000
4	1	0.00025	0.0001	0.7	20,000
5	1	0.001	0.0005	0.7	20,000
6	1	0.001	0.0005	0.9	20,000
7	1	0.001	0.0001	0.7	20,000
8	1	0.001	0.0001	0.9	20,000
9	2	0.00025	0.0005	0.7	20,000
10	2	0.00025	0.0005	0.9	20,000
11	2	0.00025	0.0001	0.7	20,000
12	2	0.001	0.0001	0.7	20,000
13	2	0.001	0.0005	0.7	20,000
14	2	0.001	0.0005	0.9	20,000
15	2	0.00025	0.0001	0.9	20,000
16	2	0.001	0.0001	0.9	20,000
17	1	0.00025	0.0005	0.7	30,000
18	1	0.00025	0.0005	0.9	30,000
19	1	0.00025	0.0001	0.7	30,000
20	1	0.00025	0.0001	0.9	30,000
21	1	0.001	0.0005	0.7	30,000
22	1	0.001	0.0005	0.9	30,000
23	1	0.001	0.0001	0.7	30,000
24	1	0.001	0.0001	0.9	30,000
25	2	0.00025	0.0005	0.7	30,000
26	2	0.00025	0.0005	0.9	30,000
27	2	0.00025	0.0001	0.7	30,000
28	2	0.00025	0.0001	0.9	30,000
29	2	0.001	0.0005	0.7	30,000
30	2	0.001	0.0005	0.9	30,000
31	2	0.001	0.0001	0.7	30,000
32	2	0.001	0.0001	0.9	30,000

Table 2 Detailed distribution of constructed datasets

Purpose	Number of images	Rebars
Training	436	10,734
Training, and augmentation	2,616	64,404
Validation	146	3,842
Test	146	3,700

Total	3,344	82,680

Table 3 Rankings of trained models with different hyperparameters and datasets

Rank	**Data**	**Case**	**AP50**	**AP (50:95)**	**Rank**	**Data**	**Case**	**AP50**	**AP (50:95)**
1	Aug.	case19	94.74	54.47	33	Aug.	case5	89.21	47.66
2	Aug.	case30	94.64	53.94	34	Ori.	case31	89.08	47.57
3	Aug.	case18	94.59	52.84	35	Aug.	case7	89.05	49.01
4	Aug.	case29	94.58	53.62	36	Aug.	case8	89.02	48.52
5	Aug.	case17	94.52	53.75	37	Ori.	case20	88.88	47.86
6	Aug.	case24	94.04	52.98	38	Ori.	case29	88.86	46.63
7	Aug.	case32	93.86	52.66	39	Ori.	case27	88.71	46.83
8	Aug.	case26	93.73	53.32	40	Aug.	case2	88.54	48.38
9	Aug.	case27	93.57	54.38	41	Ori.	case24	88.53	47.11
10	Aug.	case25	93.28	52.78	42	Aug.	case16	88.48	47.59
11	Aug.	case23	93.19	52.79	43	Aug.	case10	88.26	47.43
12	Aug.	case20	93.08	53.67	44	Aug.	case1	88.25	48.20
13	Aug.	case21	92.82	53.55	45	Aug.	case9	87.97	47.91
14	Aug.	case28	92.80	52.33	46	Aug.	case12	87.81	47.62
15	Aug.	case22	92.73	52.37	47	Aug.	case3	87.78	48.03
16	Aug.	case31	92.64	52.57	48	Aug.	case4	87.77	47.48
17	Ori.	case21	90.31	48.02	49	Ori.	case11	87.28	43.04
18	Ori.	case25	90.23	47.45	50	Ori.	case14	87.24	42.35
19	Ori.	case26	90.22	47.25	51	Ori.	case1	87.19	43.06
20	Ori.	case28	90.16	46.90	52	Ori.	case9	87.05	43.55
21	Ori.	case30	90.06	47.01	53	Ori.	case3	87.02	43.95
22	Ori.	case19	89.91	46.64	54	Ori.	case4	86.93	42.41
23	Ori.	case23	89.83	47.92	55	Ori.	case12	86.92	44.12
24	Aug.	case6	89.76	48.95	56	Ori.	case10	86.74	43.35
25	Ori.	case22	89.73	46.33	57	Ori.	case13	86.63	43.15
26	Aug.	case11	89.73	47.38	58	Ori.	case7	86.62	42.57
27	Aug.	case14	89.71	49.35	59	Ori.	case5	86.50	42.66
28	Aug.	case13	89.58	48.49	60	Ori.	case2	86.35	42.95
29	Aug.	case15	89.55	48.72	61	Ori.	case16	86.18	42.90
30	Ori.	case17	89.43	46.99	62	Ori.	case8	85.80	43.44
31	Ori.	case18	89.37	47.97	63	Ori.	case15	85.71	43.34
32	Ori.	case32	89.31	47.88	64	Ori.	case6	85.70	42.73

Table 4 Comparison of results produced by different detectors

	Validation		Test		
Model	AP50	AP(50:95)	AP50	AP(50:95)	Detection speed
SSD300	89.19	46.01	88.27	48.27	0.018
SSD500	89.78	46.62	89.32	49.13	0.019
YOLOv5	91.36	50.29	90.73	50.21	0.017

YOLOv6	92.34	50.76	91.95	50.59	0.016
R-FCN	93.13	51.38	93.02	51.32	0.024
Faster R-CNN	94.69	54.34	94.61	54.47	0.032