# Object Detection Model Training Framework for Very Small Datasets Applied to Outdoor Industrial Structures

M. Z. Baharuddin[1](✉), D. N. T. How[1], K. S. M. Sahari[1], A. Z. Abas[2], and M. K. Ramlee[2]

[1] Department of Electrical and Electronics Engineering, Universiti Tenaga Nasional, Kajang, Malaysia
{zafri,dickson,khairuls}@uniten.edu.my
[2] Tenaga Nasional Berhad, Kuala Lumpur, Malaysia

**Abstract.** Visual inspection of electrical utility assets is crucial in ensuring the continuous operation of a system or plant. With the advent of digital imagery using mobile devices, it has become easy to collect a vast amount of asset pictures from sites. To further enhance inspection efficiency, we propose RetinaNet, a deep learning-based object detection model that can be trained to automatically detect specific objects and features from images of outdoor industrial structures. The model is capable of detecting features such as intrusions, tree or bushes in the vicinity of the lattice towers. We also introduce a model training framework for use with very small datasets which consists of rigorous data augmentation, image pre-sizing, focal loss function, progressive resizing, learning rate finder, and the Ranger optimizer. Experiment results show that the proposed model used in conjunction with the aforementioned training framework results in the lowest validation loss and highest mean average precision of 31.36

**Keywords:** Object detection · Deep learning · Convolutional neural network · RetinaNet · Small dataset · Industrial inspection

## 1 Introduction

Monitoring the condition of electrical utility assets is essential for a stable electricity network. Mobile smart devices with built in cameras have enabled the collection of a vast amount of on-site asset imagery. It is desired to create a system that can extract meaningful information from these images to further enhance the inspection efficiency in terms of accuracy and speed.

Recent advances in the area of deep learning has allowed computers to match or even surpass the accuracy of human experts on visual inspection tasks [1,2]. Given the rapid development pace of deep learning, there are a plethora of deep

Supported by Uniten R&D Seed Grant U-TS-RD-19-31.

learning-based object detector models that can be used depending on the nature of the task and data the availability of labeled dataset. In most modern deep learning object detectors, the models are usually composed of three components: backbone, neck and head. These three components are stacked atop one another to form the overall model. The backbone functions acts as a feature extractor, whereas the head is trained to predict the bounding boxes and object classes. Meanwhile the neck is an intermediate component that facilitates the head to better process the features extracted from the backbone [3]. Advances in deep learning object detectors are being made in all three components. To date, most prominent works on the backbone structure are residual networks (ResNet) [4], VGG16 [5], EfficientNet [6], SpineNet [7], CSPResNeXt50 and CSPDarknet53 [8]. Recent works on the neck includes feature pyramid network (FPN) [9], bi-directional feature pyramid network (BiFPN) [10] and path aggregation network (PAN) [11]. For the head, the two most common form is the one-stage and two-stage detection heads, each with its own pros and cons. In each form there are also anchor-based and anchor free approaches. One-stage detection heads are known to be faster and less accurate compared to its counterpart. Notable works include region proposal networks (RPN) [12], You Only Look Once (YOLO) [13], Single-shot multibox detector (SSD) [14] and RetinaNet [15]. Anchor free one-stage detectors include CornerNet CenterNet and Fully convolutional one-stage (FCOS) object detectors [16]. Two-stage detection heads are known to be more accurate at the expense of heavier computation. Anchor based versions include Faster Region Based Convolutional Neural Networks (Faster-RCNN) [17], Mask RCNN [18], and R-FCN [19]. An example anchor free version is Point Set Representation (RepPoint) [20]. Figure 1 succinctly illustrates the related works on components of a deep learning object detector.

In this study we propose a deep learning model training framework based on RetinaNet to assist in the detection of activities or features in outdoor industrial images. The main contributions of this paper are as follows:

i. We propose the RetinaNet model to detect specific activities or features in outdoor industrial images.
ii. We introduce a model training framework that enables the proposed model to be trained with a limited number of labeled images.
iii. We evaluate the performance of the proposed framework against several modern object detection models such as RetinaNet, Faster-RCNN and FCOS.

## 2 Proposed Framework

A new training framework is introduced to enable training of the RetinaNet model using a small dataset. The framework consists of augmentation and pre-sizing of the dataset images. The training optimization uses the focal loss function and progressive resizing. To select the learning rate, $\alpha$ hyperparameter, we used the learning rate finder and Ranger optimizer.

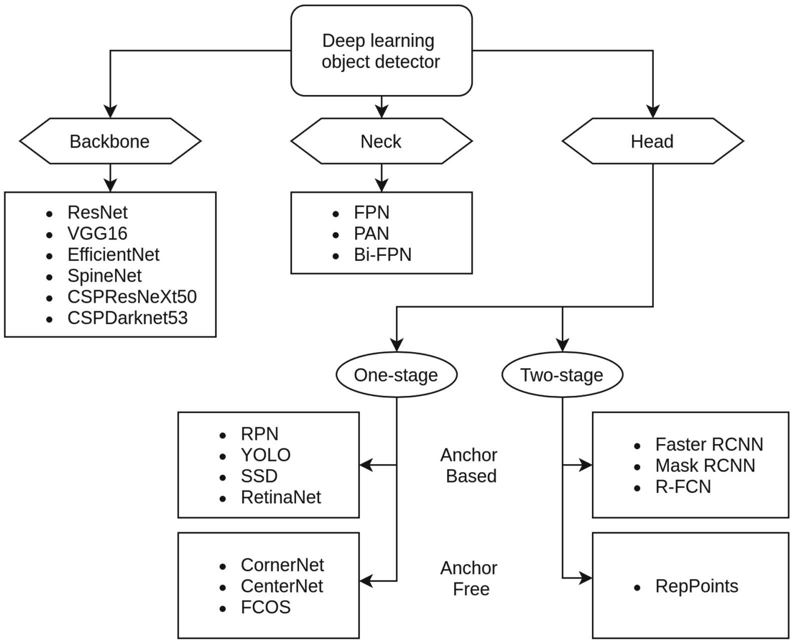


**Fig. 1.** Components of a modern deep learning object detectors and prominent architectures.

## 2.1 Dataset

In this work, we utilized proprietary images from the Malaysian national electricity utility. The images were sampled by employees as part of their routine to perform periodic inspections on transmission towers. The raw images are manually annotated and curated into three separate sets namely train, validation, and test dataset. The number of samples forming each dataset are 66, 31, and 15 images respectively. Images were resized to a $512 \times 512$ pixels image resolution. Figure 2 illustrates a few sample images from the train dataset. Due to the limited number of images, a data augmentation pipeline was used to artificially increase the number of samples for training. The images from the training dataset was subjected to the random combination of the following augmentation techniques:

i. Rotation - Image is subjected to a random rotation of 15° clockwise or counter-clockwise.
ii. Brightness - Image is subjected to a random change in brightness.
iii. Horizontal Flip - Image is subjected to a random chance of horizontal flip.
iv. Translation - Image is subjected to a random translation in the x and/or y-axis.

To boost the performance of the proposed model, a technique known as pre-sizing was used in the augmentation pipeline. Pre-sizing is a technique used in [21] whereby images were resized to a larger size than the original image and augmented before feeding them into the model. This reduces the number of lossy operations and computations so that they can be more efficiently processed on the graphical processing unit (GPU). A more in-depth explanation on pre-sizing is detailed in [22]. In this work, all images were presized to a resolution of $640 \times 640$ pixels.

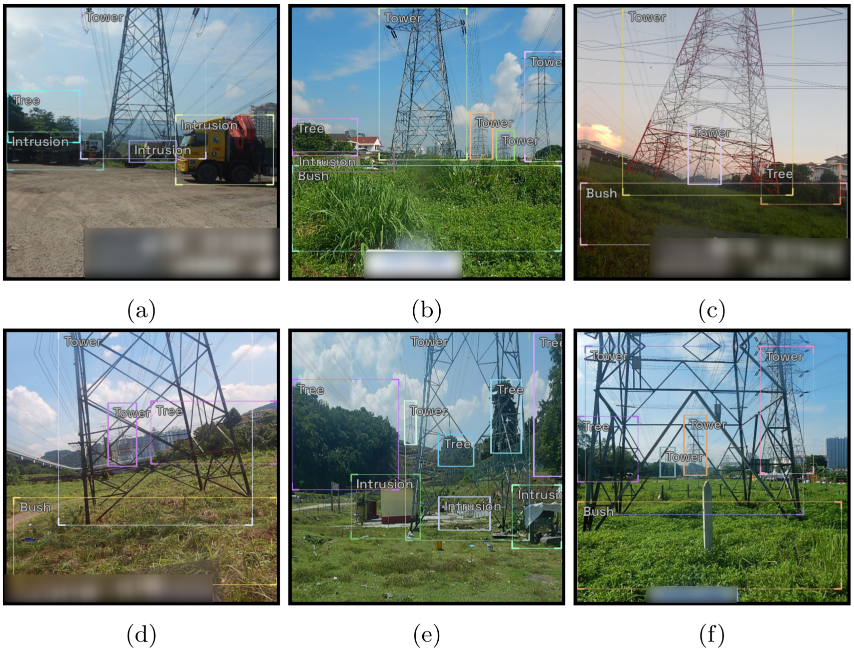


**Fig. 2.** Sample images and ground truth bounding boxes from the training dataset, augmented with (a) horizontal flip, (b) brightness increase, (c) rotation with reduced brightness, (d) rotation, (e) translation, and (f) translation with brightness increase.

## 2.2 Model

The proposed model is the RetinaNet architecture, first introduced in [15] consisting of a backbone, neck and head as shown in Fig. 3. The backbone uses ResNet, an artificial neural network first introduced in [4] that has become a standard backbone for many object detection models.

The proposed model utilizes feature pyramids as the neck which allows the model to robustly detect objects at varying scales. The various types of feature pyramids is illustrated in Fig. 4. In Fig. 4(a) is a feature pyramid mostly used for hand-engineered features such as HOG and SIFT. Figure 4(b) is commonly found in one-stage detectors such as YOLO. Figure 4(c) is commonly used in one-stage detectors such as the SSD. Figure 4(d) was proposed by [15] to be used in RetinaNet and is also known as a feature pyramid network (FPN).

Additionally, the proposed model also includes the use of a specific loss function known as the Focal Loss (FL) instead of the conventionally used Cross-Entropy (CE) loss. The FL is designed to alleviate the class imbalance problem for one-stage object detectors and has shown to improve the effectiveness of the model. FL and CE loss is given in Eqs. 1 and 2.

$$CE(p, y) = -\log(p_t) \tag{1}$$

$$FL = -\alpha_t(1 - p_t)^\gamma \log(p_t) \tag{2}$$

The term $(1 - p_t)^\gamma$ in known as the modulating factor and is a key addition to the original CE loss that improved the performance of RetinaNet over other one-stage object detectors. $\alpha_t$ is the weighting factor. The notation $p_t$ is defined as

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases} \tag{3}$$

where $y \in \{\pm 1\}$ is the ground truth class, and $p \in [0, 1]$ is the model's estimated probability of the class.

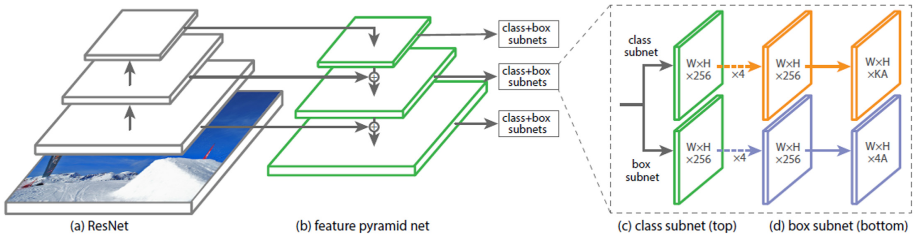


**Fig. 3.** Illustration of the model architecture [15]. (a) Backbone of the model, (b) Neck of the model, (c) and (d) Head of the model.

## 2.3 Training

To train the proposed model, we utilized a technique known as progressive resizing, introduced in [21]. In this technique, an initial model was trained using a small image size of $64 \times 64$ pixels as input. Next we used the trained weights as a starting point and re-train the model again with a larger image resolution of $128 \times 128$ pixels. With each successive training the input resolution of the images were increased until the desired resolution ($512 \times 512$ pixels). In this study the image resolution was increased in the order: $64 \times 64 > 128 \times 128 > 256 \times 256 > 384 \times 384 > 512 \times 512$ pixels. The benefit of progressive resizing is that it allows the model to be trained quickly, and also generalizes better by mitigating the

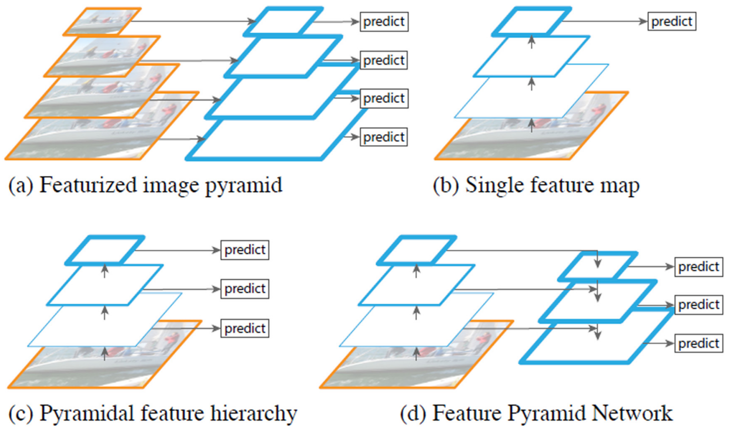

predict
(a) Featurized image pyramid
(b) Single feature map
(c) Pyramidal feature hierarchy
(d) Feature Pyramid Network

**Fig. 4.** Different feature pyramid architectures [9].

overfitting tendency due to data scarcity [23]. For consistency, all models used in this study utilized backbones pretrained on the ImageNet dataset.

Learning rate, is arguably one of the most important hyperparameter that significantly influences model performance [24]. Instead of empirically choosing the learning rate, the proposed model was trained using a learning rate finder as a guide. The learning rate finder was first introduced in [25]. In combination with the Ranger optimizer [26], this allowed the proposed model to be trained quickly with less overfit on the training dataset. Figure 5 illustrates the learning rate finder plot showing the optimal learning rate range of values. According to [25] optimal learning rate values lie within the minimum and valley point in the plot where the loss descents most rapidly. The proposed model was trained on a learning rate, $\alpha = 2e - 4$.

All models in this study were trained of a Ubuntu 20.04 LTS with Intel core Intel Core i7-4790K CPU at 4.00GHz, 32GB RAM and a single Nvidia RTX3090 GPU. We utilized the open source PyTorch 1.8.0 [27], IceVision 0.8.0 [28] and Fastai 2.4.1 [21].

## 3 Results and Discussions

### 3.1 Performance Metrics

In this section we present the performance of the proposed model against other models (RetinaNet and Faster-RCNN) with different pretrained ResNet backbones. The RetinaNet and Faster-RCNN model is based on the implementation from [15] and [17] respectively. The ResNet architecture is based on the implementation from [4]. The numbers following the model name indicate the number of layers in the ResNet model and the length of pretrained epoch on the ImageNet dataset. For example, ResNet50_2x indicates a 50-layer deep residual

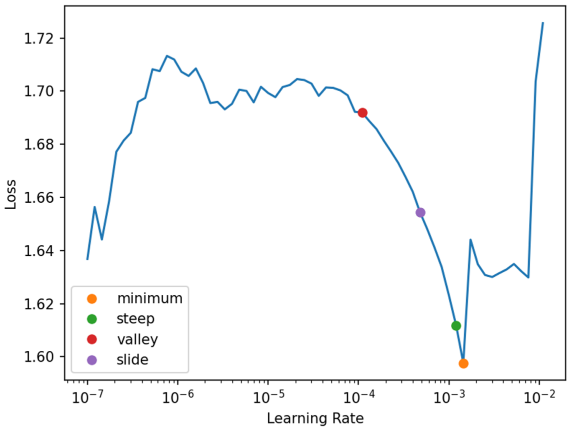


**Fig. 5.** Learning rate finder algorithm estimates the optimal range of values that results in the rapid decrease in loss function. Show in this figures are four suggested learning rate values: minimum, steep, valley and slide.eps

network trained for 24 epochs (twice the normal training epoch of 12) on the ImageNet dataset.

Table 1 shows cross-model comparison on mAP metric performance on the train, validation and test dataset. We observe that for all models, the mAP on the training dataset surpasses the mAP for the validation and test dataset significantly. The difference in performance could indicate overfitting on the training dataset which is expected given the small amount of available training images. The performance of all models on the validation dataset is close to the test dataset indicating the validation data is representative of the test dataset. The proposed model achieves a mAP of 31.36% on the test dataset, outperforming others.

We observe that ResNet101 backbones tend to perform poorer than ResNet50 backbones. This is also consistent with a plausible overfitting of the training data since ResNet101 twice the number of layers compared to ResNet50 and hence contains significantly more parameters. As for the ResNet pretraining duration, we observe no significant trend in performance on the test dataset. The proposed model outperformed all other models in the mAP metric for the test dataset. Despite similar model architecture, the performance of the proposed model is slightly better compared to RetinaNet-ResNet50_2x on the test dataset. We attribute the performance increase to the training method of the proposed model.

The proposed model was trained with a more recent Ranger optimizer in combination with pre-sizing and progressive resizing while all other models were trained on the Adam optimizer with no pre-sizing and progressive resizing. The combination of Ranger optimizer, pre-sizing and progressive resizing seems to slightly mitigate the overfitting issue encountered across all models. This can also be observed in the validation loss graph of the proposed model as show in

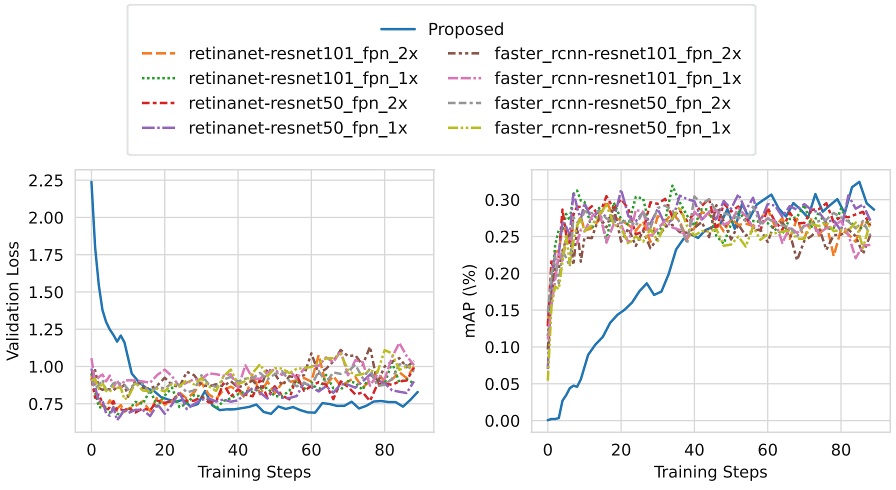
Fig. 6(a). Compared to all other models, the validation loss of the proposed model dived below all the others approximately halfway through training. Note also that the validation loss of all models tend to increase with prolonged duration of training. This shows that increasing training duration contributes to more overfitting. However, the validation loss curve of the proposed model is not only lower than the others, but it does not progress in an upward trend resulting in better performance as shown in Fig. 6(b). For all other models, the validation mAP generally stops improving about halfway through training. However, for the proposed model the validation mAP continue to increase until the end of the training.

**Table 1.** Mean average precision performance metric on the train, validation and test dataset across all models and backbones.

| Model Type | Backbone | mAP (%) | | |
|---|---|---|---|---|
| | | Train | Valid. | Test |
| RetinaNet(Proposed) | ResNet50_2x | 52.75 | 30.01 | 31.36 |
| RetinaNet | ResNet50_1x | 49.63 | 31.01 | 29.82 |
| RetinaNet | ResNet50_2x | 56.30 | 29.10 | 30.27 |
| RetinaNet | ResNet101_1x | 53.12 | 31.24 | 27.39 |
| RetinaNet | ResNet101_2x | 54.27 | 27.80 | 28.11 |
| Faster RCNN | ResNet50_1x | 48.35 | 27.42 | 30.74 |
| Faster RCNN | ResNet50_2x | 48.71 | 27.45 | 30.37 |
| Faster RCNN | ResNet101_1x | 44.82 | 24.10 | 24.81 |
| Faster RCNN | ResNet101_2x | 47.56 | 24.85 | 27.11 |
| FCOS | ResNet50_1x | 47.22 | 22.82 | 23.16 |
| FCOS | ResNet101_1x | 54.65 | 24.55 | 24.08 |

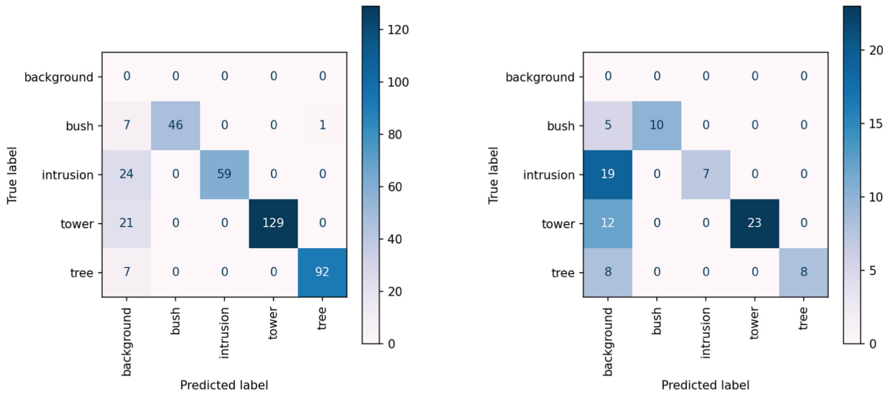
### 3.2 Inference

We ran the images from the test dataset to visualize the inference bounding boxes in comparison to the ground truth boxes. In Fig. 8, the inference output on a portion of the test dataset images. The top row shows the ground truth images and bounding boxes, while the bottom row shows the inference output from the proposed model. Observe that in Fig. 8(a) and (d) interestingly, the model was able to correctly detect objects that were not labeled in the ground truth image. For example, Fig. 8(d), the model detects "tree" and "intrusion" near the bottom left corner which were absent in the ground truth image. Comparing Fig. 8(b) and (e), the ground truth image was incompletely labeled where the labels for tree and tower were mistakenly left out. However, the bounding boxes on the output image clearly shows that the model was able to detect objects that were not labeled as ground truth. This shows that the model has sufficiently learned good representations of the objects from the training dataset. Comparing to Fig. 8(c)



(a) (b)

**Fig. 6.** Validation loss and mean average precision of all models on the validation set during training.



| True label \ Predicted label | background | bush | intrusion | tower | tree |
|---|---|---|---|---|---|
| background | 0 | 0 | 0 | 0 | 0 |
| bush | 7 | 46 | 0 | 0 | 1 |
| intrusion | 24 | 0 | 59 | 0 | 0 |
| tower | 21 | 0 | 0 | 129 | 0 |
| tree | 7 | 0 | 0 | 0 | 92 |

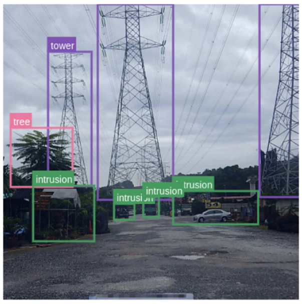
(a) Evaluation on the training dataset.

| True label \ Predicted label | background | bush | intrusion | tower | tree |
|---|---|---|---|---|---|
| background | 0 | 0 | 0 | 0 | 0 |
| bush | 5 | 10 | 0 | 0 | 0 |
| intrusion | 19 | 0 | 7 | 0 | 0 |
| tower | 12 | 0 | 0 | 23 | 0 |
| tree | 8 | 0 | 0 | 0 | 8 |

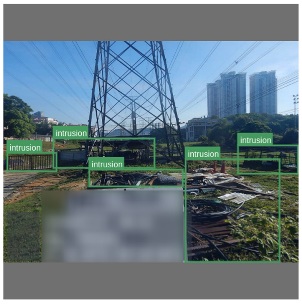
(b) Evaluation on the test dataset.

**Fig. 7.** Confusion matrix plot of the proposed model on the train and test dataset.

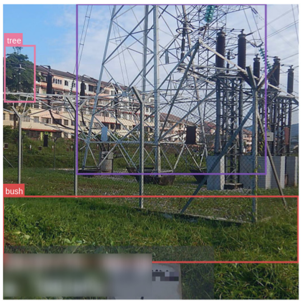
and (f), the model mistakenly detected the fencing around the transmission tower as an intrusion with low confidence level but correctly identified a patch of bushes not labeled in the ground truth. Figure 7 shows the confusion matrix on the training and test dataset. In both datasets "intrusion" and "tower" were the most frequently mis-detected classes.



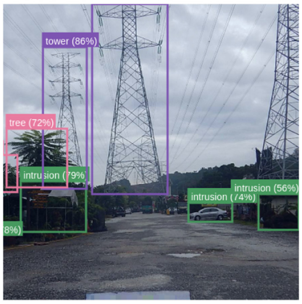
(a)



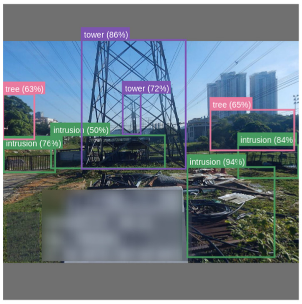
(b)



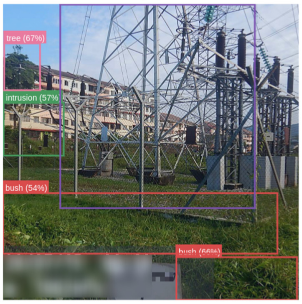
(c)



(d)



(e)



(f)

**Fig. 8.** Ground truth with manual annotations (top row) and automatic inference bounding boxes performed by the proposed model (bottom row).

## 4 Conclusion

In this study we proposed the use of a RetinaNet deep learning-based object detection model for automatic detection of activities and features in outdoor industrial images. We also introduced a model training framework for use with very small datasets. This new model training framework was benchmarked with various deep learning based object detectors using the available dataset. The comparison showed that the introduced training framework resulted in a lower validation loss and a higher mAP metric, even with scarce data availability. Our model achieves the highest mAP of 31.36% on the test dataset, outperforming others.

## References

1. Zhou, W., Yang, Y., Yu, C., Liu, J., Duan, X., Weng, Z., Chen, D., Liang, Q., Fang, Q., Zhou, J., et al.: Ensembled deep learning model outperforms human experts in diagnosing biliary atresia from sonographic gallbladder images. Nature Commun. **12**(1), 1–14 (2021)
2. Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X.: Object detection with deep learning: a review. IEEE Trans. Neural Networks Learn. Syst. **30**(11), 3212–3232 (2019)

3. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
5. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
6. Tan, M., Le, Q.: Efficientnet: rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR, pp. 6105–6114 (2019)
7. Du, X., et al.: Spinenet: learning scale-permuted backbone for recognition and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11 592–11 601 (2020)
8. Wang, C.-Y., Liao, H.-Y.M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., Yeh, I.-H.: Cspnet: a new backbone that can enhance learning capability of cnn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 390–391 (2020)
9. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
10. Tan, M., Pang, R., Le, Q.V.: Efficientdet: scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10 781–10 790 (2020)
11. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018)
12. Faster, R.: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 9199 (2015)
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
15. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
16. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems 28, pp. 91–99 (2015)
18. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
19. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379–387 (2016)

20. Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: Reppoints: point set representation for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9657–9666 (2019)
21. Howard, J., Gugger, S.: Fastai: a layered api for deep learning. Information **11**(2), 108 (2020)
22. Howard, J.: Deep Learning for Coders with fastai and PyTorch. O'Reilly Media (2020)
23. Bhatt, A., Ganatra, A., Kotecha, K.: Covid-19 pulmonary consolidations detection in chest x-ray using progressive resizing and transfer learning techniques. Heliyon, p. e07211 (2021)
24. Van Rijn, J.N., Hutter, F.: Hyperparameter importance across datasets. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2367–2376 (2018)
25. Smith, L.N.: A disciplined approach to neural network hyper-parameters: Part 1-learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820 (2018)
26. Wright, L., Demeure, N.: Ranger21: a synergistic deep learning optimizer. arXiv preprint arXiv:2106.13731 (2021)
27. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. Advances in neural information processing systems **32**, 8026–8037 (2019)
28. Vazquez, L., Hassainia, F.: Icevision: an agnostic computer vision framework (2020)