MEKATRONIKA
e-ISSN: 2637-0883
VOL. 4, ISSUE 2, 16 – 23
DOI: https://doi.org/10.15282/mekatronika.v4i2.8850

ORIGINAL ARTICLE

Fine-tuned RetinaNet models for Vision-based Human Presence Detection

Tang Jin Cheng[1], Ahmad Fakhri Ab. Nasir[2], Anwar P.P. Abdul Majeed[3], Thai Li Lim[4], Mohd Azraai Mohd Razman[1], and Ismail Mohd Khairuddin[1]

[1]Faculty of Manufacturing and Mechatronic Engineering Technology, Universiti Malaysia Pahang, 26600 Pekan Pahang, Malaysia.
[2]Faculty of Computing, Universiti Malaysia Pahang, 26600 Pekan Pahang, Malaysia.
[3]School of Robotics, XJTLU Entrepreneur College (Taicang), Xi'an Jiatong-Liverpool University, Suzhou, 215123, P. R. China.
[4]TT Vision Holdings Berhad, 11900 Plot 106, Sungai Hilir Keluang 5, Bayan Lepas, FIZ.4, Bayan Lepas, Pulau Pinang, Malaysia.

ABSTRACT – Moving towards Industry 4.0, the idea of human-robot interaction (HRI) and human-robot collaboration (HRC) has been popularized. To introduce more robots into the industries, risk-correlated issues would be always on the hook as robots are not as flexible as human. In fact, although robots can replace human workers in some of the dangerous tasks, still human safety is always the top priority for all industries. The most common way to safeguard the human was to isolate the working space of human workers and robots. To realize the idea of Industry 4.0, it is postulated to have the robots and cobots out of the cage to maximize productivity and efficiency. Hence, studies have been conducted with the attempts to free the robots from the isolated working space while preserve the safety of human operators. The present study seeks to explore the feasibility of transfer learning strategy — fine-tuning to human presence detection tasks as the base of practicing safe HRI. A custom image dataset with 1463 images was collected and separated into train, validation, and test set with a ratio of 70:20:10. Three RetinaNet object detection models with different backbone networks were fine-tuned with the acquired dataset to transfer the knowledge learned from source domain to the target domain, which is the human presence detection tasks. The result has shown that the RetinaNet_ResNet152-V1-FPN has the highest test AP of 74.4% with an inference speed of 13.09 FPS, suggesting that it is the best fine-tuned RetinaNet models. This study has demonstrated the feasibility of using fine-tuning as the strategy to train the object detection models, which can possibly act as the base for improving HRI applications via a deep learning visual-based method. In summary, the research has signified the uses of deep learning models to perform human presence detections and can be further extended for HRI safety applications.

ARTICLE HISTORY
Received: 12th Oct 2022
Revised: 3rd Nov 2022
Accepted: 20th Nov 2022

KEYWORDS
Human Detection
Deep Learning
Transfer Learning
Fine-tuning
RetinaNet

INTRODUCTION

For several decades, the employment of industrial robots have greatly increased especially for the industries moving towards automated processes to replace human workers in a variety of monotonous, challenging, and hazardous duties [1]. Unlike human, robots are innovated to only receive and execute decisions from the predefined programs, hence they fall short in terms of flexibility and adaptability [2]. Without a proper way to manage the workspace between the robots and the workers, the uncertainty might ultimately cause the life of the human workers [3]. To minimize such risks, the most straightforward way is to keep the robots isolated [4]. Numbers of strategy are innovated by integrating different sensors to improve safe human-robot interactions [5]–[7].

Human-robot interaction is meant to be the study of the interactions between human and robots, often referred as HRI in the research field. As the industries move towards Industry 4.0, the concept of HRI has become well-known as it suits the trend of Industry 4.0 [8]. It has once been described that human-robot collaboration is an ideal combination of both human adaptability and robot efficiency. To realize this wonderful combination, a drastic number of efforts have put in. Still, human-centric approach always comes first where the human is always at the top priority over the robots especially from the aspect of safety. The industries intend to further explore on ways how to execute HRI and HRC concepts as much as possible in a safe manner [9].

Hence, research on methods and strategies to improve human safety with regard to HRI and HRC is crucial. Studies have been conducted in relation to this topic [10]. It has been a fact that over the years, the vision-based systems are considered as the notable approach owing to the richer context provided by this sense [11]. Nevertheless, the major advance of artificial intelligence and deep learning techniques has never forgot to look after every single research domain, including plantation industry such as tomato classification, plantation industry like diagnosis plant disease detection application, as well as the manufacturing industries such as quality inspection [12]–[14]. However, the literature that uses both deep learning techniques and vision-based sensor to help improve human safety in human-robot interactions is rarely

***CORRESPONDING AUTHOR** | Ahmad Fakhri Ab. Nasir | ✉ afakhri@ump.edu.my

found in previous studies. For this reason, this paper intends to explore the feasibility of transfer learning approaches on the deep learning-based object detection models — RetinaNet for the human presence detection tasks.

The remainder of this paper is organized as follows: Section 2 provides a brief review on the recent works related to the topics of improving human-robot interactions in terms of safety. Section 3 describes about the development of the custom datasets for human presence detection purposes. Section 4 outlines the proposed methods and algorithms used in this study. Section 5 reports and summarizes the results from the proposed approach. Section 6 concludes the paper with some emphasize on the important findings of the research.

RELATED WORK

Mohammed et al. [15] reported an effective online augmented reality-based approach for collision avoidance. To simulate the augmented environment, both the three-dimensional (3D) models of the robots and the 3D models of human operators were included. In case of human operators, depth cameras were used as the sensors and point clouds were captured to simulate the presence of the human operators in the virtual environment. The relative distance between the human operators and the robots in the augmented environment was monitored for the collision detection application. This solution helps improve the HRI safety without having a significant impact on the performance of the robotic system.

Pasinetti et al. [16] presented a vision-based safety system for human-robot collaboration by using Time-Of-Flight (TOF) cameras. A traditional machine learning method known as Histogram of Oriented Gradient (HOG) was used to recognize the human workers, while for robot recognition, the Kanade-Lucas-Tomasi (KLT) algorithm was used. Essentially, two concepts were implemented in this study to practice safe HRI, one is the comfort zones strategies which monitors the distance between the human workers and the robots, another one is the virtual barriers strategy where there is a hard threshold to distinguish whether the area under safe, warning, or danger category. The study has exploit on the use case of TOF cameras to improve safety of the human workers exposing to the robots.

Heo et al. [17] proposed a deep learning-based collision detection framework named CollisionNet. The main idea behind this study is to take in the high-dimensional joint signals from the robots as the input to differentiate between collision or no collision. The algorithm in between the input joint signals and the binary output collision classification is the specially designed deep neural network (DNN) model. It is noted that this method is belongs to the post-collision method as it only detects whether a collision has happened or not. Furthermore, it has been mentioned that this study only applicable to the robots that have cyclical motion. Samples of signals that include collisions had been collected for the models to perform supervised learning. Cycle normalization technique was used to normalize joint signal so that the collision signals are more noticeable. As a result, the proposed approach was tested to be sensitive to collision and robust to false alarms. The authors have mentioned that this specific DNN method could be very insensitive to uncertainties as well as the noises that are not learned by the DNN.

Amin et al. [18] intended to develop a solution based on the combination of visual perception and tactile perception, with the former perception subjected to human actions recognition while the latter cater for physical contact between the human operators and the robots. Thus, two datasets were acquired with respect to the vision and the physical contact data. Two distinct deep learning algorithms were used in the human action recognition and the physical contact detection. For instance, 3D-CNN network was used for the human action recognition, while 1D-CNN was used for the physical contact classification. From the study, it can be analysed that the vision perception is correspond to the pre-collision while the tactile perception is with respect to the post-collision. By combining both perceptions, the authors have enhanced both the productivity and the safety of the HRC applications via deep learning networks.

CUSTOM IMAGE DATASET FOR HUMAN PRESENCE DETECTION

Image acquisition

The entire custom image dataset was built from scratch for the targeted detection task — human presence detection. To be more efficient, the images were acquired from the recorded video footages instead of capturing lively. The recorded video footages within a specific timeframe were obtained from the surveillance database of TT Vision Holdings Berhad. Only a few of the surveillance cameras were selected in the first place. The reason behind this decision is that the human workers in these production areas tend to move around, which can introduce variance to the model during the fine-tuning process. Different pose and location of the human workers enable the model to have better understanding and context of the target detection, subsequently results in a better detection performance.

A total of 1463 useful images were extracted from the provided video footage. The image dataset is then split into three: training, validation, and testing with a ratio of 70:20:10. To further introduce variance, data augmentation techniques were applied to the images before feeding into the fine-tuning process as an input. For instance, horizontal flipping and cropping were implemented.

Figure 1. Example of training images as input

Image annotation

In order for the models to learn and extract the meaningful features about the object of interest, annotation have to be done for every images. In this study, only the human workers appeared in the images were labelled as this study only intended to detect the presence of human workers. Noted that the human workers were annotated with a tight bounding box to have the models capture the appropriate features. As for the annotation tool, LabelIMG [19] was used in this study. The annotation files outputted from the annotation tool was in the format of PASCAL VOC.

Image annotation formats conversion

TensorFlow was selected as the deep learning framework in correspondence to the development of the deep learning object detection models as well as the implementation of the fine-tuning strategy. Hence, the annotation files were required in the format of TFRecord as this is the format that can be understandable by the TensorFlow library to load the datasets. In this case, TFRecord files can be described as a serialized representation of the image dataset in sequence of binary strings. A script was utilized to transform all the images and the PASCAL VOC annotation files into the required TFRecord files. Considering that the TFRecord only store binary strings, the annotation label will translate to have only the class IDs. To define a mapping between the class IDs and the class name, a label map file was generated with only the "person" class involved.

FINE-TUNED RETINANET MODELS

Transfer Learning via fine-tuning

Throughout this study, a ready-to-use toolkit known as TensorFlow Object Detection API [20] was leveraged to implement the transfer learning. The employed deep learning models were the variants of RetinaNet, which is considered as a one-stage detector. In general, the detectors can be partitioned into three components, with ResNet as the backbone network, followed by the feature pyramid network as the neck, and lastly SSD as the detector head [21]–[23]. Before the transfer learning technique is applied, the RetinaNet models were pre-trained by the COCO 2017 dataset [24]. Thereafter, the feature extractor of the pre-trained RetinaNet model were extracted and fit into the fine-tuned RetinaNet model.

In this context, rather than having the deep learning model to learn from the custom dataset which is smaller in size and could be less semantics, the model generalizes over the source dataset followed by the target dataset with such fine-tuning strategy. By putting fine-tuning into practice are advantageous as it does not require an enormous amount of data for training. By this means, only the detector head was subject to the fine-tuning process, while the feature extractor was remained as the same. Three RetinaNet models were employed in this study, with a difference in the backbone network — ResNet50, ResNet101 and ResNet152. All three RetinaNet models were having the same configuration, with a 2.5k

of warmup steps paired with a learning rate of 0.01. As a whole, three RetinaNet models underwent 100k training steps paired with a learning rate of 0.001.

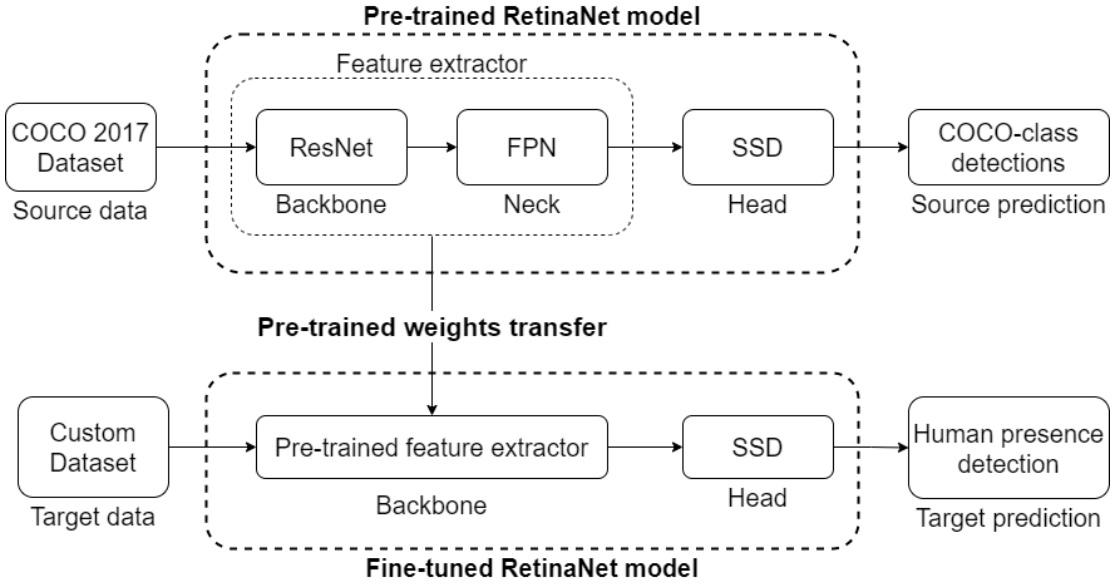

Figure 2. Transfer learning of RetinaNet models via fine-tuning strategy

Learning rate and loss inspection

A number of hyperparameters can be modified to improve the learning of the fine-tuning process. Learning rate is one of the crucial hyperparameters among the others as it can affects the sensitivity of learning behaviour as well as time taken to complete the fine-tuning process. In order to carefully handle the learning rate, the learning rate is scheduled via the technique of cosine annealed with warm restart learning schedule [25]. The idea behind this scheduling technique is that the learning rate is started with a relatively high learning rate that will drop in a cosine manner to a minimum value. The warm restart increases the learning rate when the learning rate is near the minimum value, which simulated as a reset of the learning process with reusing the good weights as the starting point.

Although the technique can help improve the learning behaviour, chance is still there for exploding gradient and vanishing gradient to occur when the error gradients accumulate. Overseeing these phenomena, monitoring of the learning losses was take place. The total loss was monitored throughout the fine-tuning process to further guarantee the learning of the RetinaNet models goes well.

Performance evaluations

Within the domain of deep learning-based object detection, Average Precision (AP) is the most common evaluation metrics used to estimate the performance of an object detector. In relation to the computation of AP, precision and recall have to be calculated at first. To define the classification is either true or false, the threshold value of Intersection over Union (IoU) is required to set beforehand as the calculation of precision and recall is based on the concept of IoU. In specific, IoU is defined as the ratio of overlapping area between the prediction box and the ground-truth box to union area of two. The calculation of IoU is visualized in Figure 3.

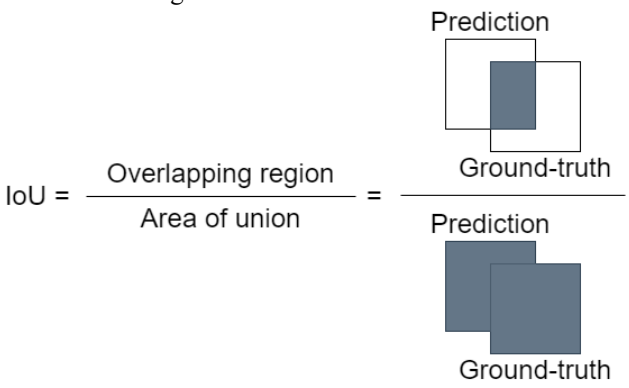

Figure 3. Visualization of Intersection over Union (IoU)

In this study, the threshold value of IoU is set to 0.5, indicates that as long as the IoU of the prediction box is above the IoU threshold value, it is classified as True. Subsequently, the precision and recall of the predictions can be calculated, thus the precision-recall curve can be computed by using the formula below:

$$Average\ Precision, AP = \sum_{n} (Recall_n Recall_n) Precision_n \quad (1)$$

EXPERIMENTAL RESULTS

Monitoring of the learning losses was conducted to ensure that the learning of the models throughout the fine-tuning process goes well. As such, the training curve as well as the validation curve were monitored. With respect to three RetinaNet models underwent the fine-tuning process, three learning curves were illustrated as shown in Figure 4 to Figure 6. Each of the graph includes both training and validation losses of that specific RetinaNet fine-tuned model, with the orange line as the training loss and the grey dots as the validation loss.

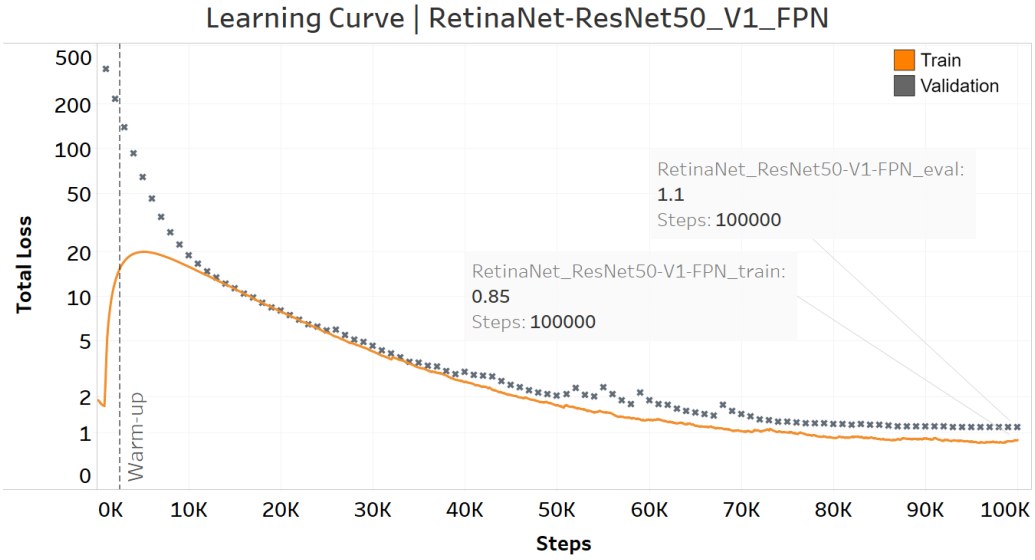

Figure 4. Learning curves of fine-tuned RetinaNet with ResNetV1-50

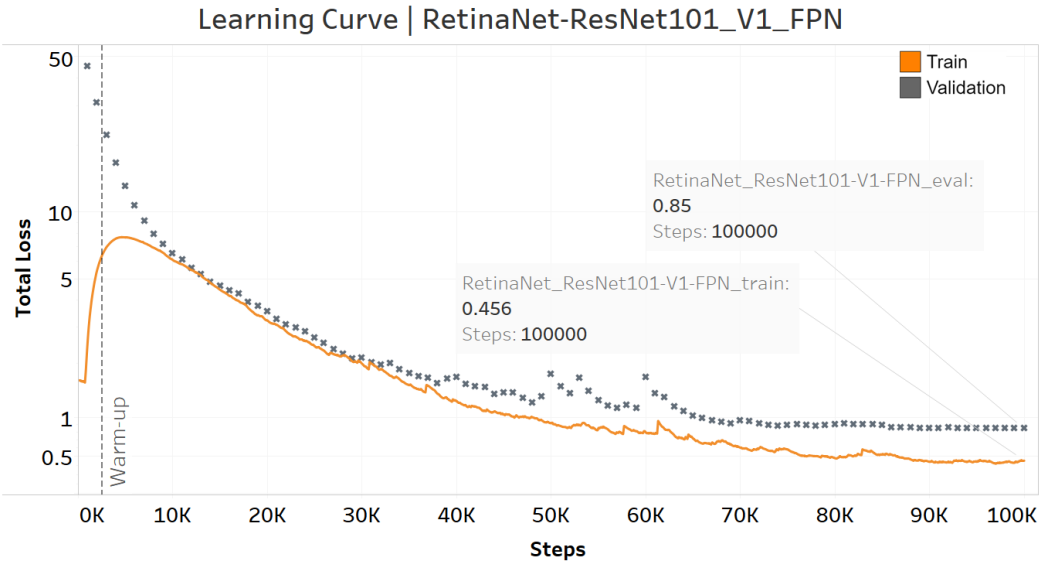

Figure 5. Learning curves of fine-tuned RetinaNet with ResNetV1-101

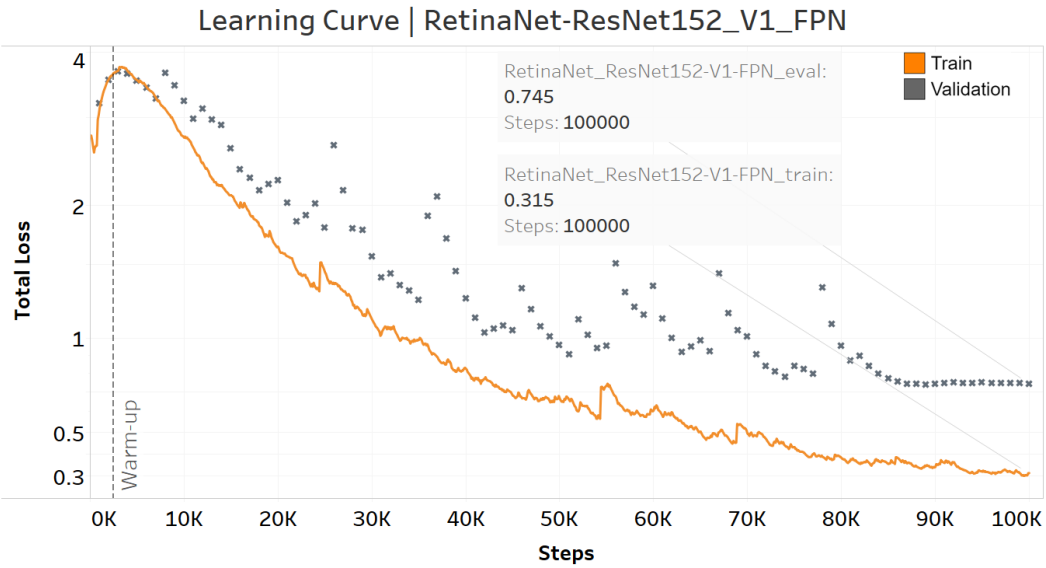

Figure 6. Learning curves of fine-tuned RetinaNet with ResNetV1-152

From all the total loss curves of the RetinaNet fine-tuned models, it can be observed that before the warm-up steps, the total loss is either having a high start-off point or increasing in a drastic manner. After the warm-up steps, the total loss starts to behave differently, either the slope of the curve is becoming less steep, or the increasing trend starts to slow down. It is reasonable to have such phenomena at the beginning of the fine-tuning stage as the knowledge and the features learned from the source domain are not specific to the target domain. By providing the custom human detection dataset to the models, the models are able to adapt the knowledge that have been learned from the previous domain to the current human detection tasks and learn the important feature to specifically detect the presence of human.

Over the training steps, it is obvious that both the losses of all the models decrease, whereby it indicates that the models have slowly adapted the knowledge from source domain towards the target domain human presence detection tasks. As the fine-tuning process move towards the end, all the losses are becoming more stable. The total loss converges approximately at 75k, 80k, and 90k for three fine-tuned models respectively. In general, as the depth of the backbone network is greater, the training steps taken for the loss to converge is greater. Additionally, it is observed that the validation loss is higher than the training loss for most of the time towards the end of the fine-tuning process. This is another signal to validate that the models are not underfitting and overfitting after the fine-tuning process.

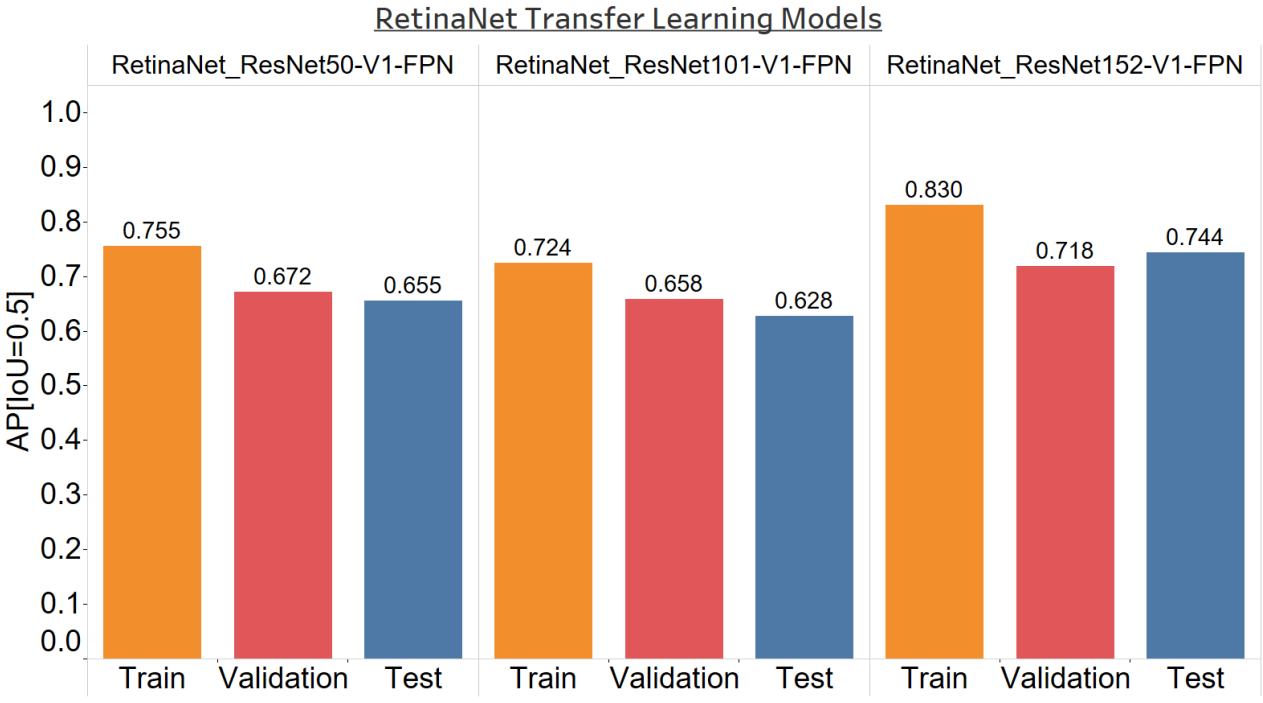

Figure 7. Train, validation, and test AP of RetinaNet fine-tuned models with 0.5 IoU

Table 1. Performance details of RetinaNet fine-tuned models

RetinaNet fine-tuned models	Model size	Number of parameters	Inference speed (FPS)
RetinaNet_ResNet50-V1-FPN	121.11 MB	31.685 M	20.92
RetinaNet_ResNet101-V1-FPN	193.89 MB	50.730 M	16.34
RetinaNet_ResNet152-V1-FPN	253.88 MB	66.419 M	13.09

With regards to the performance of the fine-tuned RetinaNet models, all three train, validation, and test APs are computed and visualized in Figure 7. Discussing about the performance of the fine-tuned models, RetinaNet_ResNet50-V1-FPN has yielded 75.5% of train AP, 67.2% validation AP and 65.5% test AP, RetinaNet_ResNet101-V1-FPN has yielded 72.4% train AP, 65.8% validation AP and 62.8% test AP, while RetinaNet152_ResNet152-V1_FPN has achieved 83.0% train AP, 71.8% validation AP and 74.4% test AP. It is reported that the fine-tuned RetinaNet_ResNet152-V1-FPN has the best performance, followed by RetinaNet_ResNet50-V1-FPN and RetinaNet_ResNet101-V1-FPN. The only difference between these fine-tuned models is the backbone network, which are ResNet50-V1, ResNet101-V1 and ResNet152-V1. From here, it can be deduced that the concept of "deeper network is better" is not necessarily true as the performance of fine-tuned RetinaNet model with ResNet50-V1 is better than the fine-tuned RetinaNet model with ResNet101-V1.

Subsequently, the inference speed on RTX 3070 GPU as well as other details of the fine-tuned RetinaNet models are tabulated in Table 1. Taking the details into consideration, as the backbone network grow deeper, the model size and the number of parameters is getting greater, the inference speed is getting slower. By comparing both RetinaNet_ResNet50-V1-FPN and RetinaNet_ResNet152-V1-FPN, the fine-tuned RetinaNet_ResNet152-V1-FPN model achieves higher test AP of 8.9% with a sacrifice of 7.83 FPS. Hence, this can be attributed to the depth of the ResNet network. If inference speed is considered as a key metric for the evaluation, RetinaNet_ResNet50-V1-FPN is recommended. For this study, AP is considered the most important metrics to evaluate the performance of the fine-tuned models, thus the RetinaNet_ResNet152-V1-FPN is suggested as the best. In short, the fine-tuned RetinaNet_ResNet152-V1-FPN model is proposed as the best fine-tuned model with correspond to the human presence detection task.

CONCLUSION

In the present research, the cameras with the surveillance systems were utilized to obtain the custom image dataset for human presence detection purposes. Human workers appeared in the scene of surveillance cameras were annotated with bounding box and appropriate settings via related tools. TensorFlow was used as the main framework in this study, therefore the dataset was converted into relevant format that is readable by the framework. Throughout the model training, a transfer learning strategy known as fine-tuning was used to fit the model with the intention of reducing time taken to train the models. Weights of the pre-trained models were transferred to the fine-tuned model and the prediction heads were trained again by feeding in the custom image dataset that is meant for human presence detection task. As a result, the fine-tuned RetinaNet_ResNet152-V1-FPN model has achieved the highest AP among all three fine-tuned RetinaNet models.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to TT Vision Holdings Berhad for providing the image dataset to make this evaluation possible as well as for supporting the study in collaboration with Universiti Malaysia Pahang through UIC200816 and RDU202405.

REFERENCES

[1] L. Wang, S. Liu, H. Liu, and X. V. Wang, "Overview of Human-Robot Collaboration in Manufacturing," in *Lecture Notes in Mechanical Engineering*, 2020, pp. 15–58.

[2] G. O. Hoskins, J. Padayachee, and G. Bright, "Human-Robot Interaction: The Safety Challenge (An inegrated frame work for human safety)," in *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, Jan. 2019, pp. 74–79, doi: 10.1109/RoboMech.2019.8704744.

[3] L. D. Evjemo, T. Gjerstad, E. I. Grøtli, and G. Sziebig, "Trends in Smart Manufacturing: Role of Humans and Industrial Robots in Smart Factories," *Curr. Robot. Reports*, vol. 1, no. 2, pp. 35–41, Jun. 2020, doi: 10.1007/s43154-020-00006-5.

[4] A. Kolbeinsson, E. Lagerstedt, and J. Lindblom, "Foundation for a classification of collaboration levels for human-robot cooperation in manufacturing," *Prod. Manuf. Res.*, vol. 7, no. 1, pp. 448–471, 2019, doi: 10.1080/21693277.2019.1645628.

[5] T. Malm, T. Salmi, I. Marstio, and J. Montonen, "Dynamic safety system for collaboration of operators and industrial robots," *Open Eng.*, vol. 9, pp. 61–71, Mar. 2019, doi: 10.1515/eng-2019-0011.

[6] S. Robla-Gomez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria, "Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments," *IEEE Access*, vol. 5, pp. 26754–26773, 2017,

doi: 10.1109/ACCESS.2017.2773127.
[7] M. Bdiwi, M. Pfeifer, and A. Sterzing, "A new strategy for ensuring human safety during various levels of interaction with industrial robots," *CIRP Ann.*, vol. 66, no. 1, pp. 453–456, 2017, doi: 10.1016/j.cirp.2017.04.009.
[8] A. Rojko, "Industry 4.0 concept: Background and overview," *Int. J. Interact. Mob. Technol.*, vol. 11, no. 5, pp. 77–90, 2017, doi: 10.3991/ijim.v11i5.7072.
[9] A. Khalid, P. Kirisci, Z. Ghrairi, K. Thoben, and J. Pannek, "Implementing Safety and Security Concepts for Human-Robot Collaboration in the context of Industry 4 . 0 Towards Implementing Safety and Security Concepts for Human-Robot-Collaboration in the context of Industry 4 . 0," *Int. MATADOR Conf. Adv. Manuf.*, no. July, pp. 1–7, 2017.
[10] E. Matheson, R. Minto, E. G. G. Zampieri, M. Faccio, and G. Rosati, "Human-robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, pp. 1–25, 2019, doi: 10.3390/robotics8040100.
[11] R.-J. Halme, M. Lanz, J. Kämäräinen, R. Pieters, J. Latokartano, and A. Hietanen, "Review of vision-based safety systems for human-robot collaboration," *Procedia CIRP*, vol. 72, pp. 111–116, 2018, doi: 10.1016/j.procir.2018.03.043.
[12] O. P. Toon, M. A. Zakaria, A. F. Ab. Nasir, A. P.P. Abdul Majeed, C. Y. Tan, and L. C. Y. Ng, "Autonomous Tomato Harvesting Robotic System in Greenhouses: Deep Learning Classification," *MEKATRONIKA*, vol. 1, no. 1, pp. 80–86, Jan. 2019, doi: 10.15282/mekatronika.v1i1.1148.
[13] B. S. Bari, M. N. Islam, M. Rashid, M. J. Hasan, M. A. M. Razman, R. M. Musa, A. F. Ab Nasir, and A. P.P. Abdul Majeed., "A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework, " *PeerJ Computer Science,* 7:e432, 2021, doi: 10.7717/peerj-cs.432
[14] J. A.Mat Jizat, A.P.P. Abdul Majeed, A. F.Ab. Nasir, Z.Taha, and E.Yuen,"Evaluation of the machine learning classifier in wafer defects classification", *ICT Express,* vol. 7, pp. 535-539, 2021, doi: 10.1016/j.icte.2021.04.007.
[15] A. Mohammed, B. Schmidt, and L. Wang, "Active collision avoidance for human–robot collaboration driven by vision sensors," *Int. J. Comput. Integr. Manuf.*, vol. 30, no. 9, pp. 970–980, Sep. 2017, doi: 10.1080/0951192X.2016.1268269.
[16] S. Pasinetti, C. Nuzzi, M. Lancini, G. Sansoni, F. Docchio, and A. Fornaser, "Development and Characterization of a Safety System for Robotic Cells Based on Multiple Time of Flight (TOF) Cameras and Point Cloud Analysis," in *2018 Workshop on Metrology for Industry 4.0 and IoT*, Apr. 2018, pp. 1–6, doi: 10.1109/METROI4.2018.8439037.
[17] Y. J. Heo, D. Kim, W. Lee, H. Kim, J. Park, and W. K. Chung, "Collision Detection for Industrial Collaborative Robots: A Deep Learning Approach," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 740–746, Apr. 2019, doi: 10.1109/LRA.2019.2893400.
[18] F. M. Amin, M. Rezayati, H. W. van de Venn, and H. Karimpour, "A Mixed-Perception Approach for Safe Human–Robot Collaboration in Industrial Automation," *Sensors*, vol. 20, no. 21, p. 6347, Nov. 2020, doi: 10.3390/s20216347.
[19] L. Tzu Ta, "LabelImg. Git code." 2015, Accessed: Jun. 28, 2021. [Online]. Available: https://github.com/tzutalin/labelImg.
[20] J. Huang, C. Sun, K. Murphy, and S. Guadarrama, "Speed / accuracy trade-offs for modern convolutional object detectors : Supplementary Materials," *Cvpr*, pp. 7310–7319, 2017.
[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, Dec. 2015, doi: 10.1109/CVPR.2016.90.
[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," *Proc. - 2019 IEEE Intl Conf Parallel Distrib. Process. with Appl. Big Data Cloud Comput. Sustain. Comput. Commun. Soc. Comput. Networking, ISPA/BDCloud/SustainCom/SocialCom 2019*, pp. 1500–1504, Dec. 2016, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00217.
[23] W. Liu *et al.*, "SSD: Single shot multibox detector," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.
[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, 2018, doi: 10.1109/TPAMI.2018.2858826.
[25] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," Aug. 2016, doi: 10.48550/arxiv.1608.03983.