Malaysian Journal of Computing, 8 (2): 1589-1601, 2023

eISSN: 2600-8238

SPORTS ITEM DETECTION USING MOBILENETV2 WITH SINGLE SHOT DETECTOR

Aiman Haziq Ab Yazik[1], Siti Nur Kamaliah Kamarudin[2*], Yuzi Mahmud[3] and Azliza Mohd Ali[4]

[1, 2*,3,4]*School of Computing Sciences,*
College of Computing, Informatics, and Mathematics, UiTM Shah Alam
[1]aimanhaziqyazik@gmail.com, [2*]snkamaliah@uitm.edu.my, [3]yuzi@uitm.edu.my,
[4]azliza794@uitm.edu.my

ABSTRACT

Object detection can be applied in various situations and systems. In this research, object detection was specifically applied to build a lending system for tracking the details of sports items in a students' dormitory. The current sports item lending system with a manually recording of data was time consuming, and are more prone to human errors. This inspires the researcher to build a real-time object detection web-based sports item lending system. The system was trained using Single Shot Detector (SSD) with Mobilenet V2 technique to detect the sports item in the warehouse. A total of 960 self-collected sports item image data was applied in four different experiments with the same batch-size configuration and learning rate value of 0.02. From the experiments, several models with different number of training iterations and training data were built to find the best model to be implemented in the sports item lending system. The best model was obtained from the second experiment with a high accuracy of 0.93 mean average precision (mAP), a confidence of 97%, and a total loss of 0.28. For future work, it is recommended to increase the volume of training data, include other variations of objects in order to further improve the results, and apply other object detection techniques for comparison purposes.

Keywords*: Deep Learning, Machine Learning, MobilenetV2, Object Detection, Single Shot Detector.*

Received for review: 19-06-2023; Accepted: 29-09-2023; Published: 10-10-2023
DOI: 10.24191/mjoc.v8i2.22641

1. Introduction

Lending is the process where an individual allows another person or business to borrow something (Barone, 2020). In this research, sports items were used as part of the lending system which records details such as borrowers' details, purpose to borrow, list of the item borrowed and date for the item to be returned. Some organisations or businesses prefer the manual system to keep records because it is simple, low cost and easier to correct entries (Queensland Government, 2016). However, it costs a substantial amount of time and tedious administrative work to keep using the manual system. Furthermore, the manual system is more prone to human error made by distracted, poor training, fatigue, speed working or incompetent team member.

Hence, this research suggests migration to an automated system to avoid human errors and cost-saving (Brick, 2018).

Single Shot Detector (SSD) is a model that can run in real-time without a significant trade-off in accuracy that was developed by Google Research Team in 2016 (Pokhrel, 2020). SSD originally use VGG16 to extract feature maps from the image (Hui, 2018). MobileNet is a neural network architecture, and it is 32 times smaller than SSD's original architecture, the VGG16 (Howard et al., 2017). Even though it is a smaller network architecture in comparison to VGG16, MobileNet has the same accuracy and provides greater speed that is made especially for mobile devices and embedded devices. SSD with MobileNet architecture is used in many real-world applications such as recognition tasks, self-driving cars, and intruder detection in CCTV that need to be carried out in a timely fashion on computationally limited devices (Howard et al., 2017). Hence, the build of the lending system for sports items in this research were by applying the SSD model with the MobileNet technique.

2. Literature Review

Object detection is the process of using computer vision to locate or identify objects of images or videos. Examples of applications for object detection are such as facial and feature recognition, people counting, industrial quality check, self-driving cars, and object detection workflow (Vaishali & Singh, 2019). In general, there are two types of object detection - machine learning-based object detection, and deep learning-based object detection (Patel, 2020). For machine learning-based object detection, the features of the object must be extracted manually. Histogram of Oriented Gradient (HOG), Haar wavelets, and Local Binary Patterns are example of machine-learning based objects. On the other hand, deep learning-based object detection can extract features that are generated from the image like shape and edge automatically (Dhar et al., 2023). One of the commonly used deep learning techniques are Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) (Patel, 2020; Dhar et al., 2023).

Object detection differs from image classification where object detection encompasses regression task where it not only predicts the objects but also finds the position of each object present in the image using a bounding box in a single run (Patel, 2020; ArcGIS, n.d.). Object detection algorithms' performance can be measured by using Intersection-Over-Union (IoU) or Jaccard Index (Patel, 2020), where IoU is the area overlapping between predicted segmentation and ground truth divided by the area of the union of ground truth and predicted segmentation. The matric ranged from 0 to 1 or 0 to 100% where 0 means no overlap and 1 signifies perfect overlap segmentation (Patel, 2020). If the value of IoU is more or equal to the threshold value (0.5) then it is a good prediction (Masurekar et al., 2020). Other parameters used for model evaluation are precision, recall, and Mean Average Precision (mAP). Precision measure in percentage on how accurate the model is, recall measure how good our model find all the positive and Mean Average Precision (mAP) is the mean value of the average precisions (Masurekar et al., 2020). The higher the Mean Average Precision (mAP) the better the network performance (Gupta et al., 2023).

2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is part of Deep Learning, and they are best used in object detection for recognizing patterns such as edges, shapes, colours and textures. CNN are currently the state-of-the-art solution for object detection (Naim et al., 2020; Cheng, 2023). The convolutional layer is the hidden layer that acts as a filter (Patel, 2020). First, the convolutional layer receives input and then transforms it by using a specific pattern to the next layer. The

Convolution Layer uses a filter as stride to obtain feature maps. It also can be simplified by mapping negative values to 0 by using the Rectified Linear Unit (ReLU) (Amin et al., 2019). Subsequently, the feature map is reduced into a smaller matrix by 15 and send it to the pooling layer. The more layer the input goes through, the more sophisticated patterns the future ones can detect. Hence, the classification process will occur in a fully connected layer (Patel, 2020).

However, CNN's have a high computational cost in terms of memory and speed in the training phase but can achieve some degree of shift and deformation invariance (Naim et al., 2020). CNN can give a high recognition rate in various image recognition tasks like character recognition, handwritten digit recognition, object detection and facial expression. For example, the accuracy of CNN was satisfactorily at 84.89% when it is trained using the CIFAR-10 dataset (Naim et al., 2020).

2.2 You Only Look Once

The You Only Look Once (YOLO) algorithm utilizes Convolutional Neural Networks (CNN) for object detection purposes, where YOLO is a popular object detection algorithm known for its real-time performance on GPUs (Qaddoura et al., 2021). Unlike two-stage object detectors such as Faster R-CNN, which separates object detection into region proposal and classification stages, YOLO is a single-stage detector that performs detection and classification jointly, hence improving the speed of object detection (Qaddoura et al., 2021). The YOLO algorithm has been successfully applied to various problems, including road surface defects detection (Benallal & Tayeb, 2023), automating vehicle detection on the road (Susanto et al., 2021), and object detection for visually impaired individuals (Chandankhede & Kumar, 2022). Even though R-CNN's accuracy performance are quite high, they exhibit a disadvantage of being quite slow even when running on a GPU (Rosebrock, 2020).

According to Roy et al. (2020), the YOLO algorithm is renowned for its real-time object detection capabilities and is considered the fastest compared to other algorithms such as R-CNN. However, the processing power of the YOLO algorithm were significantly reduced when running the YOLO algorithm on embedded devices. To address this issue, a smaller-sized version (less than 50MB) of YOLO, called Tiny-YOLO, is applied. In terms of speed, Tiny-YOLO outperforms its larger counterpart, achieving up to 442% faster performance and reaching up to 244 frames per second (FPS) on a single GPU when tested with the COCO dataset. Due to its small size and fast architecture, Tiny-YOLO is an ideal choice for running real-time object detection on embedded devices such as Raspberry Pi (Roy et al., 2020). However, it is important to note that YOLO, in general, tends to be less accurate than the two-stage object detectors like R-CNN. Since Tiny-YOLO is a smaller version of YOLO, it can be inferred that Tiny-YOLO is even less accurate than YOLO (Roy et al., 2020).

2.3 Single Shot Detector

The Single Shot Detectors (SSD) algorithm is specifically designed for real-time object detection, similar to the YOLO model (Deng & Li, 2023). While Faster R-CNN utilizes a region proposal network to create boundary boxes and classify objects, it is slower in achieving real-time performance (Deng & Li, 2023). To address this limitation, SSD incorporates changes such as using low-resolution images, eliminating the need for a region-proposal network, and implementing multi-scale features and default boxes (Deng & Li, 2023). These modifications enable SSD to achieve real-time speed while maintaining accuracy. Based on the research conducted by Deng & Li. (2023), the Single Shot Detectors (SSD) algorithm achieves a mean average precision (mAP) of 74.3, surpassing the performance of Fast R-CNN (70.0) and Faster R-CNN (73.2). However, it falls slightly behind their proposed SSD framework (78.1 mAP) and DSSD (78.6 mAP). Despite this, SSD demonstrates superior speed compared to all those methods achieving a frame rate of 46 FPS, while Faster R-CNN operates at 7 FPS, DSSD at 9.5 FPS, and their proposed method at 36 FPS (Deng & Li, 2023). This agrees with Gupta et al. (2023)'s study on driver drowsiness detection through facial expression using CNN, where SSD MobileNet V2 exhibited a performance that was almost on par with that of their proposed model of SSD ResNet50 V1. Both models outperform EfficientDet D0 (Gupta et al. 2023).

3. Methodology

The following subsections described the general phases used to complete this research. The first phase described how data was manually collected and prepared, the second phase described on how the model was designed and developed, applying the SSD with MobileNet V2 algorithm, and lastly how the sports item lending system was designed and developed.

3.1 Data Collection and Preparation

The data are manually collected from the internet and self-captured using the mobile phone. However, the data collected are not uniform in terms of definition, size and quality of the image. Then the data are named first to avoid confusion during the preparation process.

Next, the data undergo the preparation process, so it can be used for training and testing. This phase's key aspect is removing redundancies and noises in the data. The corrupted image must first be removed by analysing the captured image. These preparation phases include cleaning the data, labelling the data, standardising the data format, and removing any outliers and errors. The presence of redundancies and noises in the dataset will influence the training process and result to produce an inaccurate system in the future. In order to ensure the high quality of the data used for training, thorough preparation was conducted. To reduce its size and standardize its type, a node package manager (NPM) called browser-image-compression was used with Node.js. Each image was processed and reduced the size to below 100kb and each of them was compressed to a maximum height or width of 1080px.

Then, each of the data was labelled for training and testing purposes. This process was done manually by using *LabelImg* application. Finally, the data were split into the standard split ratio - 80% training set, and 20% testing set to be used later to train with Single Shot Detector with Mobilenet V2 model.

3.2 Model Design and Development

In model design and development phase, the Single Shot Detector with MobileNet V2 was chosen as the technique to perform the object detection. The reason why SSD was chosen was that the particular model had high processing performance and was able to outperform Faster R-CNN in terms of accuracy (Deng & Li, 2023). In addition, SSD and MobileNet were suitable to be implemented into a web-based application.

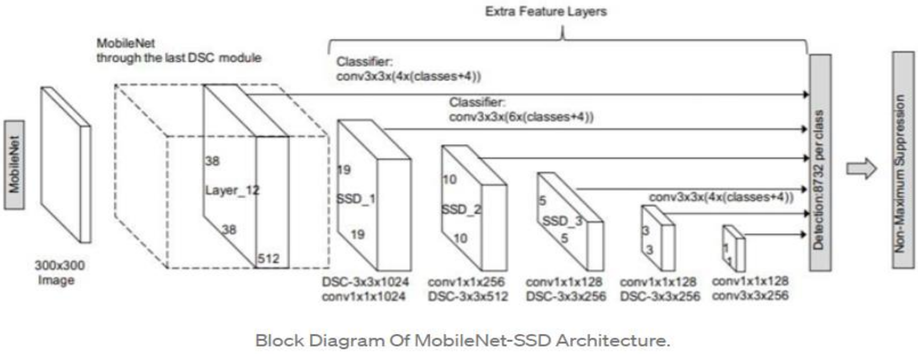

Figure 1. Block diagram of MobileNet-SSD architecture (Narayan, 2019).

Figure 1 shows the model architecture of MobileNet-SSD. This architecture starts with the MobileNet network. First, the image will become the input to the MobileNet network. The purpose of the MobileNet network was to extract feature maps, while SSD make many predictions for every single class. There are six-layer which is the convolutional layer that performs the classification of object detection tasks. Single Shot Detector will make 8732 predictions for every single object. Then, the Non-max Suppression was applied to remove duplicate prediction. Subsequently, SSD will check the confidence score of each box and pick the top 200 predictions per image.

3.3 System Design and Development

The system architecture for the lending system was developed to define how the object detection model can perform in a real-world application. The system architecture was designed to define the structure, behaviour, and the overall view of the system.

The system architecture was divided into five core sections - the user, front-end, back-end, object detection and database.

- User Section:

  From the user section, the user will give the information such as the item borrowed, borrower information and the return date and then submit the information. The system will display the borrowing result whether it is approved or not.

- Front End

  The front end is the graphical user interface part of the website that interacts with the user. Reactjs library is used for the system front end that will perform posts and get requests based on activity.

- Backend

  Backend is server-side rendering that processes business logic and accesses the other resources such as the MongoDB database and cloud services. Nodejs with Strapi is used as a backend for this system.

- Database

  A database is an organized collection of data where the system can store the information and retrieve the information. MongoDB Atlas database is used in this system.

- Object Detection

  Object detection is used as the component in the system which help to detect tools by using real-time video. The object detection use is SSD MobileNet model and hosted on IBM cloud.

The software of lending system using object detection were then developed based on the architecture designed as in Figure 2. The system development starts with developing a graphical user interface, encoding process, database deployment, system process and lastly setup of the hardware.

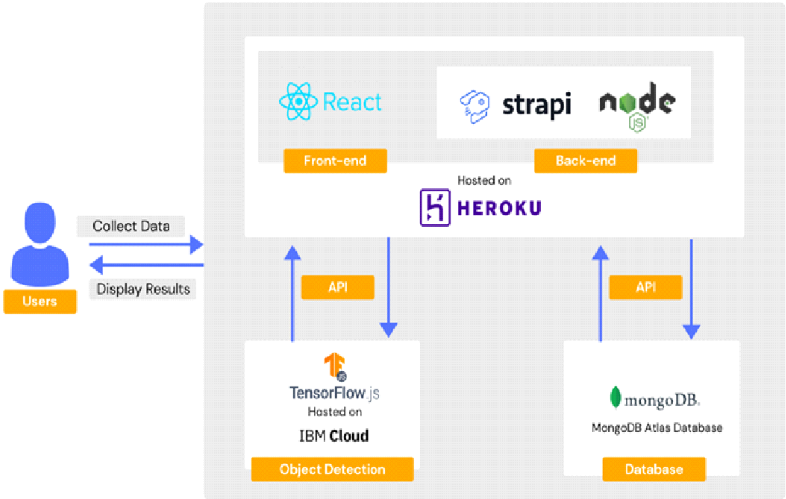

Figure 2. Lending System architecture.

The system was developed by using *Reactjs* as the front end and Nodejs with Strapi as its backend. The database to store the information was deployed on MongoDB Atlas. The model that was trained with Single Shot Detector with MobileNet V2 were then converted into *TensorFlowjs* model and hosted on IBM cloud. Figure 3 shows the example of the system output detection interface and the bounding box around the targetted object.

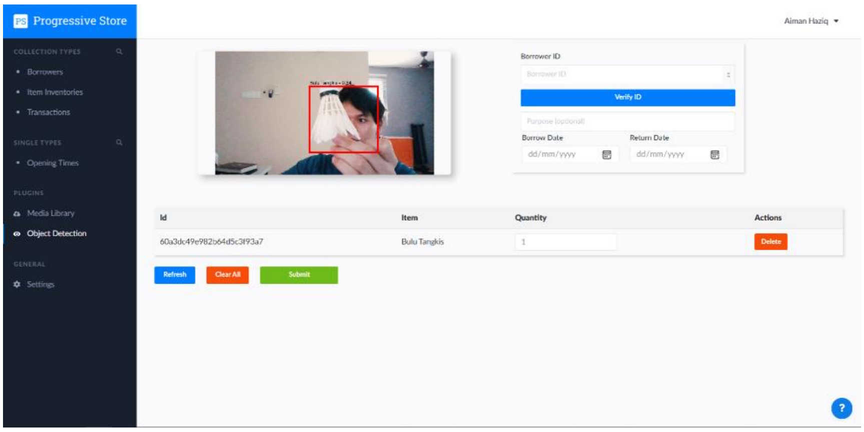

Figure 3. Output of the object detection in the Lending System.

4. Results

Four experiments were conducted to develop the object detection component with better accuracy and performance for the object detection - lending system object. The details for the experiment's configuration are as in Table 1.

Table 1. Parameter Configurations for each Experiment.

Experiment Number	Number of Datasets	Number of Iterations
Experiment 1	480	20000
Experiment 2	960	20000
Experiment 3	480	50000
Experiment 4	960	50000

The first experiment was conducted using 5,000 iterations with 480 datasets or using half of the prepared data to see how well it performed. The learning rate for all the experiment training processes was fixed at 0.02 with a batch size of 4. Then the model was evaluated and converted into *Tensorflow.js*, so the model performance can also be measured on the web. The final step taken was to see and measure how well the model performs in real-world applications. Next, a few experiments were completed by combining different iterations and different numbers of datasets to analyse and compare the results with the previous experiment. Finally, the best system with high accuracy and the lowest detection error was selected for an object detection system parameter for the lending system.

All the experiments were evaluated where the best model among the various experiments conducted will be applied for the object detection component in the lending system. The comparison of the experiment was divided into two parts which are model evaluation and system performance. For the model evaluation, the best model was the model that performs with the highest mAP(precision), highest recall, lowest classification loss, lowest localization loss, and lowest total loss. However, the best model in model evaluation was not chosen to be implemented into the system if it did not satisfy the system evaluation in a real-time environment. In system evaluation, the best accuracy and detection errors were also recorded.

5. Discussion

The main purpose of model evaluation is to evaluate the generalized performance of the model in terms of its precision and recall. The overall model evaluation for each experiment which include the generalized classification loss, localization loss, and total loss were recorded.

Table 2 shows the model evaluation for all the experiment conducted. Experiment 1 showed the worst mAP (0.90090), recall (0.8515), localisation loss (0.05666) and total loss (0.2858). However, experiment 1 had a better classification loss (0.1349) as compared to experiment 3 (0.1693).

Meanwhile, experiment 2 demonstrated a mean average precision (mAP) of 0.93414, on par with experiments 3 and 4. However, it outperformed experiments 1 and 3 in terms of classification loss, achieving a value of 0.1215. Experiment 2 performed better than experiment 1 in terms of localization loss and slightly better total loss than experiment 1. In addition, experiment 2 also has the highest recall compared to the other three experiments.
Next, it can be observed that experiment 3 achieved the highest mAP (0.93820) and localization loss (0.04326) than the other experiments. However, experiment 3 did not perform in terms of classification loss, localisation loss, and total loss as compared to experiment 4.

Table 2. Model Evaluation.

Experiment	**mAP (Precision)**	**Recall**	**Classification Loss**	**Localisation Loss**	**Total Loss**
Experiment 1	0.90090	0.8515	0.1349	0.05666	0.2858
Experiment 2	0.93414	0.8776	0.1215	0.05404	0.2825
Experiment 3	**0.93820**	0.8596	0.1693	0.04326	0.2784
Experiment 4	0.93736	0.8682	**0.1124**	**0.4323**	0.2288

In the case of experiment 4, the classification loss was recorded as 0.1124, the localization loss as 0.04323, and the total loss as 0.2288. It is also on par in terms of precision and recall with the best result in experiment 2 with less than 0.01 differences.

System evaluation was done after completing the model evaluation. The main purpose of system evaluation is to assess the complete system performance on how likely the system performs in real-world conditions. In this research, for the system evaluation purpose, the threshold limit was lowered from 0.8 (80%) to 0.1 (10%) to display the system detection accuracy on sports items and detection error accuracy on other non-sports object that has the same shape.

Figure 4 shows the accuracy of the system of each of the experiments. For this system evaluation, the sports equipment was placed at least 50 cm from the camera to detect. Experiment 4 displayed the best result which is 100% followed by experiment 3 (99%) and experiment 2 (97%) and experiment 1 (86%). Experiment 4 able to detect every object with high accuracy up to 100%.

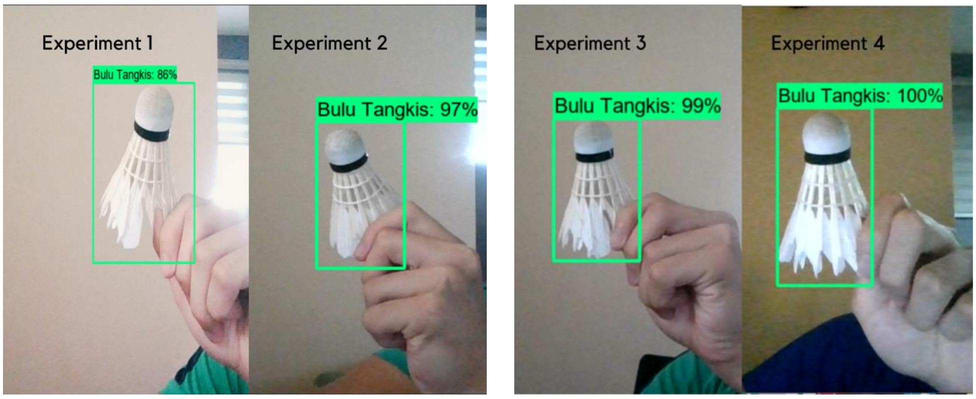

Figure 4. Object detection accuracy results.

However, experiment 4 and experiment 3 have a bigger error where it detects other objects such as the clock as takraw or shuttlecock with higher than 80% in terms of accuracy, as depicted in Figure 5. Experiment 1 and experiment 2 also make detection errors but they detect not more than 60% accuracy throughout the system evaluation.

This shows that experiment 3 and experiment 4 were not suitable to be implemented into the system as the system built will detect input if the accuracy is more than the 0.8(80%) threshold. It can be concluded that experiment 2's model was the best model to be implemented into the system.

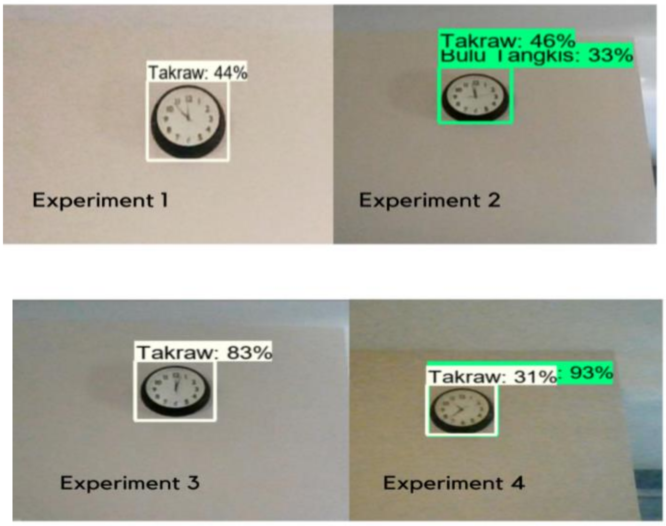

Figure 5. Object Detection accuracy results other than sports items.

5.1 Lending System Evaluation

Next, the testing phase on the completed system was done to evaluate how well the system works and whether it is able to detect all four sports items scoped in this research. The system also shows the performance of the chosen experiment 2's model to detect sports items in real-time.

Figure 6 shows a summarized results of the object detection lending system successfully predicting all four sports items. Users also need to specify the quantity of the item in the table before submitting the form. The other actions such as refreshing the object detection, clearing all items in the table or deleting an item are also built-in in the object detection lending page to provide a better user experience. After the form is submitted, it will be recorded in the transactions database and look at the detail in transaction collection types.

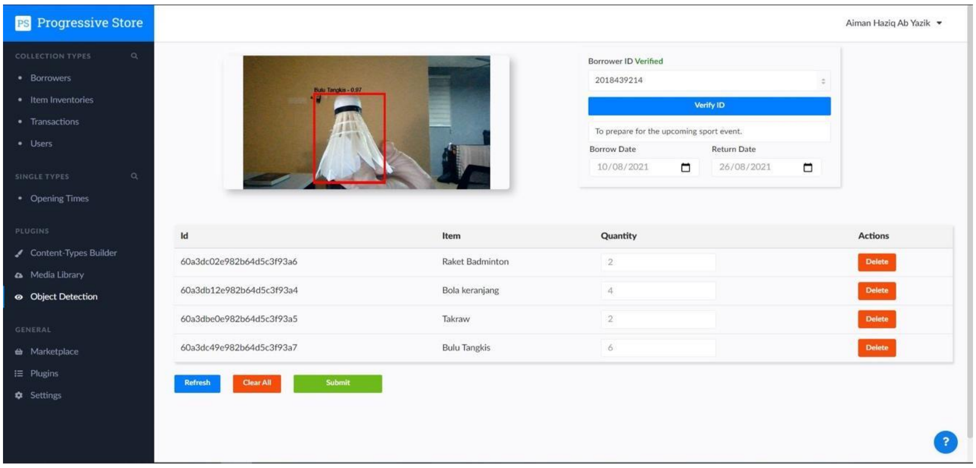

Figure 6. Tool Lending System interface and in use.

6. Conclusion and Future Works

The Single Shot Detector model were able to be implemented into the system and the system run successfully in real time. Nevertheless, the system built has its own limitation and weakness, and it can be improved in many ways in the future. Some of the recommendations for future works were to increase the number of datasets for each of the items for the training process, so the system can ensure the stability of detection, increase the number of tools and categories, classify the tools at any angle of the tools and lastly make a progressive web application for the tool lending system.

Acknowledgements

The authors would like to acknowledge the contribution and guidance provided by the School of Computing Sciences, College of Computing, Informatics, and Mathematics, UiTM Shah Alam in facilitating the completion of this research. The support and resources provided by the institution have been instrumental in allowing us to conduct this study, and we are grateful for the opportunities that have shaped our intellectual development and research capabilities.

Funding

The author (s) received no specific funding for this work.

Author Contribution

Author 1 conducted the experiments and the drafted the paper. Author 2 formatted and corrected the paper in general, and author 3 and author 4 proofread the paper for clarity.

Conflict of Interest

The authors have no conflicts of interest to declare.

References

Admin. (2017). *The Cost of Human Error*. https://www.gammasolutions.com/gamma-chat/cost-human-error/

Amin, P., S., A. B., Shetty, B. B., K., K., & Shetty, L. (2019). Object Detection using Machine Learning Technique. *International Research Journal of Engineering and Technology (IRJET)*, 6(5), 4. https://irjet.org/volume-6-issue-05

ArcGIS. (n.d.). *How single-shot detector (SSD) works? https://developers.arcgis.com/python/guide/how-ssd-works/*

Barone, A. (2020, 19 August 2020). *Lender. https://www.investopedia.com/terms/l/lender.asp*

Bhavsar, P. (2019). *Object Detection with Convolutional Neural Networks*. https://medium.com/datadriveninvestor/object-detection-with-convolutional-neural-networks-dde190eb7180

Breitmeyer, R. (2015, 21 July 2015*). What are the 7 disadvantages to a manual system ?* https://www.linkedin.com/pulse/what-7-disadvantages-manual-system-richard-breitmeyer

Brick, B. (2018, 21 August 2018). *The Disadvantages of a Manual Operating System in a Library*. https://bizfluent.com/12746087/the-disadvantages-of-a-manual-operating-system-in-a-library.

Cheng, C., "Real-Time Mask Detection Based on SSD-MobileNetV2," *2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, Shenyang, China, 2022, pp. 761-767, doi: 10.1109/AUTEEE56487.2022.9994442.

Dhar, M., Jena, A. R., & Saha, P. (2023, September). Caption an object using the pre-trained deep learning model for single shot multi-box detector. In *AIP Conference Proceedings* (Vol. 2876, No. 1). AIP Publishing

Frankenfield, J. (2020). *Artificial Neural Network (ANN).https://www.investopedia.com/terms/a/artificial-neural-networks-ann.asp*

Gupta, N. D., Rajoo, R., & Jacob, P. J. (2023). Driver drowsiness detection system through facial expression using Convolutional Neural Networks (CNN). *Malaysian Journal of Computing (MJoC)*, *8*(1), 1375-1387.

Howard, A. G., Wang, W., Zhu, M., Weyand, T., Andreetto, M., Adam, H., Chen, B., & Kalenichenko, D. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

Hui, J. (2018). SSD object detection: Single Shot MultiBox Detector for real-time processing. *Jonathan Hui.–URL: https://jonathan-hui. medium. com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06 (date of application: 10.10. 2021)*.

Hutt, D. (2016). Man V machine : Overcoming human error in the workplace. *https://southeastasiaglobe.com/human-error/*

Khandelwal, R. (2019). SSD : Single Shot Detector for object detection using MultiBox. Towards data science. Retrieved 3/1/2021 from *https://towardsdatascience.com/ssd-single-shot-detector-for-object-detection-using-multibox-1818603644ca*

Masurekar, O., Jadhav, O., Kulkarni, P., & Patil, S. (2020). Real Time Object Detection Using YOLOv3. *International Research Journal of Engineering and Technology (IRJET)*, 7(3).

Naim, M. M. H., Amrin, R., Ali, M. R., Zubaer, A. A., & Islam, M. A. (2020). Object Detection from Images using Convolutional Neural Network based on Deep Learning. *International Journal Of Engineering Research & Technology (IJERT)*, Volume 09(Issue 09).

Narayan, T. (2019). Object Detection : A Comparison of performance of Deep learning Models on Edge Using Intel Movidius Neural Compute Stick and Raspberry PI3. *https://medium.com/intel-student-ambassadors/object-detection-a-comparison-of-performance-of-deep-learning-models-on-edge-using-intel-f66eb7f45b17*

Patel, A. (2020, 11 Jun 2020). What is Object Detection? *https://medium.com/ml-research-lab/what-is-object-detection-51f9d872ece7*

Peterson, D. (2018, November 2018). Maximize Efficiency: How Automation Can Improve Your Loan Origination Process. *https://www.moodysanalytics.com/articles/2018/maximize-efficiency-how-automation-can-improve-your-loan-origination-process*

Pokhrel, S. (2020). Real-Time Vehicle Detection with MobileNet SSD and Xailient. *https://www.xailient.com/post/real-time-vehicle-detection-with-mobilenet-ssd-and-xailient*

Queensland Government. (2016, 29 June). *Electronic and manual record keeping. https://www.business.qld.gov.au/running-business/finances-cash-flow/records/electronic-manual*

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

Rosebrock, A. (2020). YOLO and Tiny-YOLO object detection on the Raspberry Pi and Movidius NCS.

Rouse, M. (2020). Machine Learning. *https://searchenterpriseai.techtarget.com/definition/machine-learning-ML (SearchEnterpriseAI)*

Söderholm, J. (2018). *Borrowing and lending tools: The materiality of x-lending libraries* (Doctoral dissertation, Högskolan i Borås).

Tonioni, A., Serra, E., & Di Stefano, L. (2018, December). A deep learning pipeline for product recognition on store shelves. In *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)* (pp. 25-31). IEEE.

Vaishali, S. S. (2019). Real-time object detection system using caffe model. *Int Res J Eng Technol*, *6*(5), 5727-5732.

Younis, A., Shixin, L., Jn, S., & Hai, Z. (2020, January). Real-time object detection using pre-trained deep learning models MobileNet-SSD. In *Proceedings of 2020 the 6th international conference on computing and data engineering* (pp. 44-48).

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.

Benallal, M. A., & Tayeb, M. S. (2023). An image-based convolutional neural network system for road defects detection. *IAES International Journal of Artificial Intelligence*, *12*(2), 577.

Chandankhede, P., Kumar, A. (2022). Guiding Aid For Visually Impaired. IJEM, 2(12), 34-40. https://doi.org/10.5815/ijem.2022.02.04

Deng, X., & Li, S. (2023, March). An Improved SSD Object Detection Algorithm Based on Attention Mechanism and Feature Fusion. In *Journal of Physics: Conference Series* (Vol. 2450, No. 1, p. 012088). IOP Publishing.

Qaddoura, R., M. Al-Zoubi, A., Faris, H., & Almomani, I. (2021). A multi-layer classification approach for intrusion detection in iot networks based on deep learning. *Sensors*, *21*(9), 2987.

Roy, B., Nandy, S., Ghosh, D., Dutta, D., Biswas, P., & Das, T. (2020). MOXA: A deep learning based unmanned approach for real-time monitoring of people wearing medical masks. *Transactions of the Indian National Academy of Engineering*, *5*, 509-518.

Susanto, S., Budiarjo, D. D., Hendrawan, A., & Pungkasanti, P. T. (2021). The implementation of intelligent systems in automating vehicle detection on the road. *IAES International Journal of Artificial Intelligence*, *10*(3), 571.