COMPARISON BUS PASSENGER COUNTING AND GENDER DETECTION USING YOLOV8, FASTER R-CNN, AND MASK R-CNN ALGORITHM

[1]Arvel Adelar Budiono, [2]Yulianto Tejo Putranto
[1]Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Katolik Soegijapranata
[1]21k10002@student.unika.ac.id, [2]yulianto@unika.ac.id

ABSTRACT

Bus passengers' data are crucial for the bus agents, especially in Indonesia. With this data, bus agents could identify the traffic for each route of the bus. To handle this problem, many researchers have made a system to count and detect the public transportation passengers with different algorithms. Many researchers defined that You Look Only Once (YOLO) has best performance to overcome the object detection problem that has similarity in this research. The Convolutional Neural Network algorithm is also not inferior in implementing object detection either. In this research, it will investigate these three algorithms, You Only Look Once version 8 (YOLOv8), Faster Region Convolutional Neural Network (Faster R-CNN), and Mask Region Convolutional Neural Network (Mask R-CNN), in counting bus passengers and detecting the bus passenger's gender. To find the best performance of these two algorithms, they will use a dataset that contains 408 photos of bus passengers. This research aims to analyze the result of the bus passengers data that could reduce the misalignment and determine the best algorithm to use in this case.

Keywords: YOLOv8, Faster R-CNN, Mask R-CNN, Object Detection, Bus Passenger Counting, Gender Detection

INTRODUCTION

Background

Public transportation plays an important role in facilitating mobility for people in Indonesia, especially with large vehicles like buses. Numerous bus agencies provide this service to improve mobility and reduce traffic congestion. Many Indonesians still rely on buses for their daily activities. However, discrepancies often arise between the recorded passenger data and the actual number of passengers due to factors such as inaccurate counting, fare evasion, and sudden changes in passenger load during special events. In some cases, stowaways also contribute to this misalignment. People counting is a critical factor for the transportation industry and can be effectively addressed using machine learning systems capable of processing vast image datasets. This research utilizes You Only Look Once version 8 (YOLOv8), Faster R-CNN, and Mask R-CNN algorithms for passenger counting and gender detection based on labeled head images. Identifying the gender of passengers allows bus agencies to collect more accurate data and understand their target market better. The focus of this research is to assist bus agencies in accurately counting passengers and identifying their gender while the bus is in transit, enabling drivers and agencies to check the current passenger count and reduce data inaccuracies.

Problem Formulation

There are a number of this research focuses on to make this research on point, such as:

1. Which algorithm has the highest accuracy in counting the bus passengers counting and gender detection?
2. How to implement an effective method and technology to make an accurate count and gender detection of the bus passengers?
3. How effective are these methods in reducing the mismatch of the bus passengers data?

Hypothesis

This research has an expected outcome, which is that using the You Only Look Once version 8 (YOLOv8), Faster R-CNN, and Mask R-CNN could reduce or remove the passenger data mismatch between bus agent and the number of passengers in the bus.

Research Objective

The primary objective of current research is to avoid the mismatch between bus agent passenger data with the actual number of passengers in the bus. This research considers the best method or algorithm of counting the number of bus passengers and detecting the passenger's gender.

Scope

Dataset of this research will take from the existing passengers photos that were captured by the IBos GPS camera and it will be labeled first. Because of that, this research implementation only works on passenger bus photos, not videos that include when the counting does. It also does not support counting the passengers in real time. The result is the number of passengers and, man and woman presentation; events, speed, condition in the bus will not be shown.

LITERATURE REVIEW

To make this research possible, the writer has reviewed a number of literature that provide different methods for the people counting strategy. [1] Ya-Wen Hsu et al. stated that currently, so many related applications exist for managing bus information services, such as website pages or mobile applications that can be used by the people to detect the location and arrival time of buses. They proposed the use of deep learning object detection and Convolutional Autoencoder (CAE) method to evaluate the number of passengers in a bus by summing the result of these methods. This literature selected the You Only Look Once version 3 (YOLOv3) to count based on object detection for areas where the head of passengers are more visible, and CAE architecture for the scenarios of the crowded areas on bus. Videos were used as the dataset of this research to support the real-time detection. This research uses the YOLOv8 for better performance than the YOLOv3 that is used in this literature because of its high efficiency as the detection model.

[2] Object detection algorithms by using deep learning have been a huge success until right now. If these algorithms are used in real-time, it should be prepared with a graphics processing unit (GPU), so Hyunduk Kim et al. proposed a real-time Automatic Passenger Counting system with Tiny-YOLOv3 head detection. They proposed this research because transport systems used information like the number of passengers to plan transportation routes and schedules. The transport companies could utilize their resources, improve their service quality, and make the transport cost more effective. But, these proposed algorithms have a high accuracy result if it is implemented in a less crowded area. Two steps were implemented, first is head detection and second is the passenger direction recognition. By using Tiny-YOLOv3 in head detection, the result of this implementation showed 99% accuracy and a 0.041 second recognition speed that indicates Tiny-YOLOv3 has a huge impact (used seven convolutional layers and six max-pooling layers).

[3] Daniel Baumann et al. explained that counting passengers in public transportation is still done manually, so they proposed a modern method to count passengers using RetinaNet as the model. In this article, there is a statement that Convolutional Neural Network (CNN) has proven to be highly effective for people counting in image recognition and CNN has a large number of architectures released in recent years. At the end of the RetinaNet model, ReLU activation function was used because it suitables to transform the number of people to value range and it would be able to output every number of people. [4] As a comparison, Region based Convolutional Neural Networks (R-CNN) algorithm was used by Sultan Daud Khan et al. They proposed a deep learning model method that worked as a head detector in sports video. It also takes into consideration the scale of head variations in the video. Their framework could detect heads in low and high density crowd situations, count the number of people, and localize people's heads simultaneously. In a bus situation, it is included in a low density crowd situation because the passengers sit on their own seats. So, [5] Joseph Redmon et al. stated that YOLO is better than R-CNN for people counting, especially in low density crowds. R-CNN uses region proposal methods to generate bounding boxes in an image and then run a classifier on the bounding boxes. These methods are slow and also hard to be optimized, because each individual component of images must be trained separately. Joseph Redmon et al. compared YOLO with the other real-time object detection on the same dataset. Their research results show that YOLO learns very general representations of objects, and YOLO outperforms the other object detection methods, including R-CNN.

[6] Some questions about classification were raised in this research. And one of the applications that is implemented using support vector machine (SVM) is application in face detection. It uses a kernel function and database of 361 pixel patterns about face and non-face. It runs 30 times faster and it revealed that SVM is effective to use for face detection. [7] SVM also was used to classify the age and gender of a person in this article. The proposed work of the system was evaluated by using classification rate, precision, and recall. Preprocessing is the critical factor in this research, because it improved the result accuracy at the end for the age and gender detection.

Then, Deep Convolutional Neural Network (DCNN) was used to extract the unique features from the preprocessed input. The SVM method will be used as a comparison to detect age and gender of the bus passengers. [8] CNN also used in this proposed research by Karahan, M. et al. to make the gender and age classification based on the facial features with YOLOv3. The CNN algorithm used three convolutional layers, they are two connected layers and a final output layer. The first layer contains 96 nodes with 7 kernel size, the second layer contains 256 nodes with 5 kernel size, and the third layer contains 384 nodes with 3 kernel size. The output layer of the gender classification are two classes, and the output of the age classification are eight classes type of age range. This could be compared with the R-CNN algorithm method, but Karahan, M. et al. stated that the result from the YOLOv3 age and gender classification is faster and works with higher accuracy rate above eighty percent. Because YOLOv3 has the high accuracy rate of age and gender classification, this proposed work will be used as this research guideline that needs to count and classify the passengers of a bus.

[9] Real-time detection for age and gender detection has been stated by Musaddik Moulavi et al. They want to beat the disadvantages of human victimization in surveillance by a much automated visual surveillance system. They proposed a visual surveillance which includes setting modeling, motion segmentation, object classification, and many more. OpenCV was used in this work that supports Fast R-CNN and YOLO algorithm methods. YOLO's design of this work allows for end-to-end training and speeds while the system is maintaining a high average accuracy. And also YOLO could be used as an evaluator of the fast R-CNN detections to reduce the false positive background errors, then result in significant high performance. This paper will be used as a comparison because they combined the two methods, Fast R-CNN and evaluated it with YOLO, which will not be explained in this research.

[10] Palak Kakani and Shreya Vyas stated in their research that they integrate Roboflow for the image of groceries annotation, and they used YOLOv7 for the object detection algorithm. Their research involved annotating a diverse grocery item images dataset from various sources using Roboflow, and then the annotated images processed and trained using YOLOv7. They result in a successful implementation of accurate and efficient object detection. Their research paper will be used as a guideline for current research because it has the same method which is using Roboflow for the dataset of images annotation. And also there will be a comparison between YOLOv7 in their research and YOLOv8 in current research.

[11] Juan Du tried to implement the CNN Family on their research. They used a general dataset called VOC07 to implement it. They stated the urgent factor that impacts the object detection is the speed of the algorithm. So when the CNN series developed to the Faster Region of CNN (Convolutional Neural Network), the Mean Average Precision has increased to 76.4, they make this research to compare the each of CNN Family algorithms which are faster than the general CNN with the YOLO (You Only Look Once) v2 algorithm. In conclusion, YOLO has more advanced settings to implement it on object detection. YOLO is also better in generalizing representations of objects that rely on fast object detection.

[12] Yohan Marvel Anggawijaya, Tien Hsuing-Wen, and Rosita Herawati implemented YOLO to detect the parking lot in this research. They stated that searching a parking lot is a time-consuming task. By using YOLO and some algorithms, it could save energy and power consumption of CPU and GPU by 97 percent. So, this research will be used as a comparison to see the consumption of CPU and GPU while implementing YOLO and CNN Family in counting and detecting the bus passenger.

RESEARCH METHODOLOGY

Data Collection

The main dataset of this research is taken from PT Buana Online Sejahtera that contains the bus passengers photo without label. The main dataset contains 408 passenger photos The dataset will be used to train the model and some of the main dataset will be used to test the machine learning model.

Data Pre-processing

Before training the dataset, the main dataset containing bus passenger photos is labeled to indicate whether the photos contain a person and to specify their gender as male or female. After labeling, the dataset undergoes pre-processing to optimize, multiply, and improve the data quality. The pre-processing involves three steps: Auto-orientation, which automatically adjusts the image's orientation based on the embedded EXIF (Exchangeable Image File Format) data; Resizing to a fixed dimension of 640x640 pixels, which stretches the images without losing original data; and Auto-adjust contrast, which enhances the difference between light and dark areas by expanding the pixel value range for better visibility. To further enhance the model's performance, the dataset is augmented by generating three additional versions of each original image with random modifications, including hue adjustment (shifting the image’s hue within -15 to +15 degrees to simulate different lighting conditions), random saturation adjustment (modifying saturation between -15% and +15% to improve detection across color variations), and random brightness adjustment (altering brightness between -15% and +15% to simulate varying lighting conditions). After pre-processing and augmentation, the dataset is used to train two machine learning models: You Only Look Once version 8 (YOLOv8) and the Convolutional Neural Network Family (CNN Family).

Compare Algorithms

The dataset has been labeled and can be inserted into selected machine learning algorithms, namely YOLOv8 and the Convolutional Neural Network Family (CNN Family). YOLOv8 is a state-of-the-art real-time object detection model based on deep learning and computer vision, with multiple variants including YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. CNN Family, on the other hand, is a deep learning approach for image processing and recognition, with multiple layers improving detection accuracy. It includes four models: Region-based Convolutional Neural Network (R-CNN), Fast R-CNN, Faster R-CNN, and Mask R-CNN. R-

CNN uses a two-stage process, segmenting regions of interest (ROI) via selective search before passing them through CNNs for feature extraction. Fast R-CNN improves on R-CNN by combining feature extraction and classification steps using ROI pooling for faster results. Faster R-CNN further enhances speed and accuracy by introducing a Region Proposal Network (RPN) that generates region proposals from the feature map. Mask R-CNN extends Faster R-CNN with ROI Align to improve feature map alignment and includes a Feature Pyramid Network (FPN) for multi-scale feature detection and enhanced segmentation capabilities. After implementing these algorithms, their performance will be validated by measuring accuracy from the training process and the speed of detecting passengers in new images.

IMPLEMENTATION AND RESULTS

Research Setup

The researcher compared three machine learning models: YOLOv8, Faster R-CNN, and Mask R-CNN for bus passenger counting and gender detection. Google Colab with an NVIDIA T4 GPU was used to accelerate model training. Faster R-CNN was implemented using PyTorch, while Mask R-CNN was implemented with Detectron2, both enabling fine-tuning of model hyperparameters for enhanced object detection performance.

Implementation

To determine the best approach for the bus passenger detection, the researcher compares three different state of art algorithms: YOLOv8, Faster R-CNN, and Mask R-CNN. Before diving into these three algorithms, the first step of the implementation is about the dataset.

Data Collecting

This research uses a quantitative approach to gather a dataset for machine learning models. The data is collected from bus agents of PT Buana Online Sejahtera through the iBos ticketing system. Bus passenger photos, captured by a real-time GPS camera on buses and stored in a MongoDB database, are downloaded for offline modeling and analysis. The dataset contains training, testing, validation, and preprocessed images. This dataset serves as a foundation for result analysis, algorithm comparison, and machine learning modeling.

Data Labeling

To estimate the number of passengers and their gender, the researcher used Roboflow to annotate the bus passengers dataset, divided into two classes: man and woman. Square bounding boxes were applied for YOLOv8 and Faster R-CNN, while polygon annotations were used for Mask R-CNN to accurately detect complex segmentation.

Data Pre-processing and Augmentation

To enhance the machine learning models, data preprocessing and augmentation were applied using Roboflow. Preprocessing techniques included:

1. Auto-Orient: Ensures photos are correctly oriented based on metadata.
2. Resize to 640x640: Standardizes photo size for deep learning models, maintaining proportional annotations.
3. Auto-Adjust Contrast: Enhances image visibility and feature distinguishability by adjusting pixel intensity.

For data augmentation, three techniques were applied to each image to increase the training dataset size:

1. Hue Adjustment: Alters hue between -15° and +15° to simulate varying lighting conditions.
2. Saturation Adjustment: Changes saturation by -25% to +25% to account for color intensity variations.
3. Brightness Adjustment: Adjusts brightness between -15% and +15% to simulate different exposure levels and improve model robustness.

Algorithms Implementation for Object Detection

The researcher conducted a comparative analysis of three state-of-the-art object detection algorithms: YOLOv8, Faster R-CNN, and Mask R-CNN, explaining each algorithm's implementation for detecting bus passengers and their gender.

1. YOLOv8

Before training, the researcher set the runtime to GPU T4 on Google Colab for efficient deep learning workloads. The !nvidia-smi command checks the GPU's performance and availability. To set up YOLOv8, the researcher installed the Ultralytics library, imported the YOLO class, and verified the environment, ensuring compatibility and GPU availability. For dataset management, the researcher used the Roboflow to download the annotated dataset, accessing the 'passenger-head-detection' workspace and the YOLOv8 format dataset. During training, the YOLOv8 model was fine-tuned using a pre-trained model (yolov8n.pt) with a 10-epoch iteration over a 640x640 image dataset. The batch size was set to 16, and the model was saved after training. For testing, the trained model was used to predict on the test dataset, with a 50% confidence threshold. Bounding boxes were drawn on detected objects, and the final accuracy was assessed using YOLOv8's confusion matrix. The confusion matrix evaluated the model's performance, identifying true positives, false positives, true negatives, false negatives, and background detections, which were used for further analysis.

2. Faster R-CNN

Before implementing Faster R-CNN, the researcher installs essential libraries such as 'torchvision' and 'pycocotools.' 'Torchvision' supports computer vision tasks like object detection, while 'pycocotools' offers tools for evaluating models on the COCO dataset. Next, a custom dataset class, ‘Detection,’ is defined for loading images, annotations, and applying transformations. The class methods handle dataset initialization, image loading, target loading, and transforming bounding boxes into the necessary format for PyTorch. The dataset undergoes transformations, such as resizing and tensor conversion, using the ‘get_transforms’ function. For training, a function defines model parameters, sets the optimizer, and tracks losses during each epoch. A learning rate scheduler is applied to control the training process. For dataset management, Roboflow’s API is used to download the COCO-formatted dataset and load annotations. The Faster R-CNN model is prepared by adjusting the classifier to accommodate the custom dataset’s classes. The training process involves setting up a data loader, optimizer, and learning rate scheduler, followed by iterating through the epochs. After training, the model is saved. Prediction on the test dataset includes loading images, running inference, and visualizing predicted bounding boxes. Model evaluation is done by calculating true positives, false positives, background detections, and double detections using a confusion matrix. Finally, results are calculated by comparing real and predicted annotations and summarizing the performance in a table format.

3. Mask R-CNN

To start the Mask R-CNN training process, the dataset is first downloaded from Roboflow in COCO format, which includes polygon annotations suitable for Mask R-CNN. The necessary dependencies are installed, including the Roboflow package to fetch the dataset and Detectron2, a library developed by Facebook AI for object detection. After installing Detectron2, the dataset is registered using the register_coco_instances function, which registers the training and test datasets with COCO format annotations (bounding boxes, segmentation masks, and labels). Dataset metadata is retrieved from DatasetCatalog and MetadataCatalog for training and testing.

For training, the Mask R-CNN model is configured with a ResNet-50 backbone and FPN, and hyperparameters such as learning rate, batch size, and epochs are set. The model is then trained on the dataset. After training, the model is set up for predictions by loading the trained weights (maskmodel.pth) and configuring the prediction threshold. A DefaultPredictor is used to run the trained model on test images. To evaluate the model’s performance, bounding boxes in COCO format are converted to tensor format, and an Intersection over Union (IoU) function is implemented to calculate prediction accuracy. A confusion matrix function calculates true positives, false positives, background detections, and double detections by comparing ground truth annotations with predicted bounding boxes. Finally, predictions are made on the test images, visualized with bounding boxes, and the results are compared with real annotations. The performance is summarized in a table, showing the counts of true positives, false positives, background detections, and double detections.

Results

In the current research, three algorithms—YOLOv8, Faster R-CNN, and Mask R-CNN—were tested for bus passenger detection. The researcher compared the detection results with the actual data, which included the total number of male and female passengers. To evaluate the models, the following metrics were calculated:

$$\mathit{Counting\ accuracy}\ (\%) = \frac{\mathit{total\ detection} - \mathit{total\ background\ detection}}{\mathit{total\ number\ of\ passenger}} \times 100\% \qquad (1)$$

$$\mathit{Gender\ accuracy}\ (\%) = \frac{\mathit{true\ positive\ man} + \mathit{true\ positive\ woman}}{\mathit{total\ passenger} - \mathit{total\ background\ detection}} \times 100\% \qquad (2)$$

Additionally, Precision and Recall were calculated to evaluate the model's performance.

Precision measures the percentage of correct positive predictions:

$$\mathit{Model's\ precision}\ (\%) = \frac{[\text{illegible}]}{[\text{illegible}] + [\text{illegible}]} \times 100\% \qquad (3)$$

Recall measures the percentage of correct predictions compared to actual positives:

$$\mathit{Model's\ recall}\ (\%) = \frac{[\text{illegible}]}{[\text{illegible}] + [\text{illegible}]} \times 100\% \qquad (4)$$

1. YOLOv8

The researcher evaluated the YOLOv8 model's performance for bus passenger counting and gender detection by tuning various hyperparameters, including pre-trained models (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x), epoch (5-20), batch size (8-20), learning rate (0.001-0.01), and optimizer (Adam, AdamW, SGD). The best model for counting bus passengers was YOLOv8x with an Adam optimizer, a batch size of 16, a learning rate of 0.001, and 10 epochs, achieving the highest accuracy of 60.31% with precision of 51.94% and recall of 87.01%. For gender detection, the best model was YOLOv8x with an AdamW optimizer, batch size of 8, learning rate of 0.001, and 10 epochs, reaching an accuracy of 97.58%, with precision of 46.9% and recall of 89.63%. These results demonstrate YOLOv8's potential for accurately counting passengers and detecting their gender in real-world bus travel scenarios.

2. Faster R-CNN

The researcher evaluated the Faster R-CNN model's performance for bus passenger counting and gender detection by tuning key hyperparameters, including optimizer, batch size, learning rate, and epochs. The best model for bus passenger counting achieved 44.75% accuracy with 67.83% precision and 35.29% recall, using the Adam optimizer, a batch size of 32, learning rate of 0.00025, and 10 epochs. The best model for gender detection achieved 77.46% accuracy with 78.57% precision and 22.63% recall, using the SGD optimizer, a batch size of 16, learning rate of 0.01, and 20 epochs. The prediction results for both counting and gender detection were then presented for various bus routes.

3. Mask R-CNN

The researcher evaluated the performance of the Mask R-CNN model by tuning several hyperparameters, including learning rate, epochs, batch size, images per batch, and workers. The model's best performance in counting bus passengers was achieved with a learning rate of 0.01, 100 epochs, a batch size of 64, 2 images per batch, and 8 workers, reaching an accuracy of 70.43%, with 64.57% precision and 57.95% recall. The model accurately predicted passenger counts across various routes, such as Klepu - Terminal Kebumen and Klepu - Gombong. For gender detection, the best Mask R-CNN model achieved an accuracy of 81.82%, with 72.73% precision but a low recall of 3.15%. While the model was successful at predicting gender in some routes, it misclassified a few female passengers, particularly in cases like Jogja - PondokUngu.

CONCLUSION

The researcher has done experiments to determine the best algorithm to make the bus passenger counting and bus passenger gender detection. Based on the researcher’s implementation, Mask R-CNN with learning rate of 0.01, batch size of 64, 2 images for each batch, 8 number of workers, and trained for 100 epochs is the best model for counting bus passengers with 70.43% of accuracy. Then, for the bus passenger gender detection, YOLOv8 with YOLOv8x pre-trained model, AdamW optimizer, batch size of 8, learning rate of 0.001, trained for 10 epochs is the best model with 97.57% of accuracy.

Then, based on the researcher's experiments, to make an accurate and effective method of bus passengers detection, it needs a huge amount of image data with a good quality. After that, the dataset needs to be preprocessed and last give the best suite hyperparameters to the machine learning model to get the highest accuracy, like the number of epochs, the learning rate, the optimizer, and the batch size. The current research is not very effective to reduce the mismatch of the bus passengers data because the accuracy of the model needs to be improved more.

For further research, this research’s researcher suggests having a better quality of image data, improving the preprocessing step, and finding the better method to detect the bus passengers at night or in difficult lighting conditions.

REFERENCES

[1] Ya-Wen Hsu, Yen-Wei Chen, Jau-Woei Perng. Estimation of the Number of Passengers in a Bus Using Deep Learning. https://www.mdpi.com/1424-8220/20/8/2178 (2020).

[2] Hyunduk Kim, Myoung-Kyu Sohn, Sang-Heon Lee. Development of a Real-Time Automatic Passenger Counting System using Head Detection Based on Deep Learning. https://xml.jips-k.org/full-text/view?doi=10.3745/JIPS.04.0246 (2022).

[3] Daniel Baumann, Eric Sax, Martin Sommer. Use of Deep Learning Methods for People Counting in Public Transport. https://www.researchgate.net/profile/Martin-Sommer-3/publication/359614575_Use_of_Deep_Learning_Methods_for_People_Counting_in_Public_Transport/links/64ccfb81d394182ab3a21363/Use-of-Deep-Learning-Methods-for-People-Counting-in-Public-Transport.pdf (2022).

[4] Sultan Daud Khan, Mohib Ullah, et al. Person Head Detection Based Deep Model for People Counting in Sports Videos. https://www.researchgate.net/profile/Sultan-Khan-12/publication/337538151_Person_Head_Detection_Based_Deep_Model_for_People_Counting_in_Sports_Videos/links/5de4f25c299bf10bc337bb69/Person-Head-Detection-Based-Deep-Model-for-People-Counting-in-Sports-Videos.pdf (2019).

[5] Joseph Redmon, Santosh Divvala, et al. You Only Look Once: Unified, Real-Time Object Detection. https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf (2016).

[6] Zhao Jun. The Development and Application of Support Vector Machine. https://iopscience.iop.org/article/10.1088/1742-6596/1748/5/052006/pdf (2021).

[7] S. Haseena, S. Saroja, et al. Prediction of the Age and Gender Based on Human Face Images Based on Deep Learning Algorithm. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9433232/pdf/CMMM2022-1413597.pdf (2022).

[8] Karahan M, Lacinkaya F, et al. Age and Gender Classification from Facial Features and Object Detection with Machine Learning. https://www.journal-fea.com/article_148293_a1f7e806aa7be5598296ee35059cdbaf.pdf (2022).

[9] Musaddik Moulavi, Anuprita Barge, et al. Object Detection and Recognition and Age-Gender Prediction using YOLO and OpenCV. https://www.researchgate.net/profile/A-Ospanova-3/publication/360611945_Object_Detection_and_Recognition_and_Age-Gender_Prediction_using_YOLO_and_OpenCV/links/6280f8124f1d90417d6e2751/Object-Detection-and-Recognition-and-Age-Gender-Prediction-using-YOLO-and-OpenCV.pdf (2022).

[10] Palak Kakani, Shreya Vyas. Automated Catalog Generation Using Deep Leaning. https://www.irjmets.com/uploadedfiles/paper/issue_8_august_2023/44010/final/fin_irjmets1692089966.pdf (2023).

[11] Juan Du. Understanding of Object Detection Based on CNN Family and YOLO. https://iopscience.iop.org/article/10.1088/1742-6596/1004/1/012029/pdf (2018).

[12] Yohan Marvel Anggawijaya, Tien Hsuing-Wen, Rosita Herawati. Energy Aware Parking Lot Availability Detection Using YOLO on TX2. https://ieeexplore.ieee.org/document/8982448 (2019).