Identifying Parking Lot Occupancy with YOLOv5

Abstract

With the increasing need for efficient parking space management and growing population, the application of artificial intelligence (AI) in occupancy detection has become a topic of significant interest. This paper explores the effectiveness and reliability of the YOLO (You Only Look Once) object detection algorithm in differentiating between occupied and empty parking spots. Moreover, it analyzes the impact of the number of training epochs on the overall accuracy of the AI model. The study utilizes the YOLO algorithm due to its speed and accuracy which makes the training process highly efficient. A custom dataset of 135 images was created and annotated for training purposes. The primary objective of this experiment is to demonstrate the way how AI models can successfully distinguish between occupied and empty parking spaces. By addressing the capabilities of YOLO in occupancy detection, this research aims to contribute to the growing interest in AI applications for efficient parking space management and its implications in tackling real-world challenges.

Introduction

The rapid urbanization and population growth in recent years have led to an increasing demand for efficient parking management systems [1, 2]. Traditional manual monitoring approaches such as ticket booths have proven to be both time-consuming and imperfect [3]. Consequently, there is a growing need for automated system capable of real-time occupancy detection where it can accurately determine the occupancy status of individual parking spaces [4, 5]. This research paper aims to address this need by proposing an AI-based solution that analyzes images of parking lots to identify whether each slot is occupied or vacant [23]. To achieve this, the research utilizes the advanced real-time object detection algorithm, YOLOv5 (You Only Look Once version 5) [22]. YOLOv5 is well-suited for this task due to its high accuracy and efficiency in detecting objects in images and videos in real-time. It performs object detection in a single forward pass of the neural network, making it faster than traditional methods. The primary objective of this research is to develop an AI model capable of binary classification, categorizing each parking slot as either occupied or vacant. The AI model is trained using a supervised learning approach where labeled images of parking spaces are used as training data. These images are captured from various angles and under different lighting conditions, and each parking spot is annotated with its ground truth occupancy status. The output of performing object detection methods based on a trained AI model includes a set of labels indicating the occupancy status of each parking space in the input image accompanied by a confidence score representing the algorithm's certainty in its classification. Additionally, graphs representing the train/obj_loss or metrics are produced to analyze the training performance over each epoch. By effectively determining the occupancy status of parking spaces through computer vision and machine learning techniques, the proposed system has a potential to significantly improve current parking management. This improvement can lead to enhanced efficiency, reduced congestion, and an overall improved user experience. The subsequent sections of this paper will elaborate on the methodology, experimental results, and potential applications of this AI-based parking space occupancy detection system.

Dataset

The dataset utilized in this project was custom-made by capturing pictures of parking spaces in the author's neighborhood. It was specifically designed for parking lot occupancy detection consisting predominantly of images with or without cars parked at parking spaces. The dataset consists of 135 images in JPEG format with a 12.2-megapixel resolution captured using an iPhone. To label the images, the LILIN AI labeling tool was employed [20]. Each image is labeled to indicate whether the parking space is empty or occupied. The images in the dataset were captured from various angles and perspectives, resulting in a diverse collection of parking lot scenarios. The diversity is crucial for training an AI model that can generalize effectively to real-world parking lots. Before training the AI model, several data preprocessing steps were performed. Firstly, all images were resized to a standardized size of 4032x3024 pixels. This uniformity ensures that the model analyzes images of the same dimensions which in turn, enhances its accuracy. Secondly, data augmentation techniques were applied to improve the model's performance. Techniques such as changing angles of the camera and capturing pictures of both occupied and empty

parking spaces at the same location were employed to create additional variations in the dataset. Next, the dataset's pixel values were normalized to a range of 0 to 1 [21]. Normalization helps equalize the importance of each variable in the dataset as it prevents any single variable from disproportionately influencing the model's performance due to large numerical values. To evaluate the AI model's performance, the dataset was split into training and validation subsets without shuffling at the start. The training set contains 90% of the total images, while the remaining 10% forms the validation set [7]. A visual representation of the parking lot scenes and the varying perspectives captured is attached below.

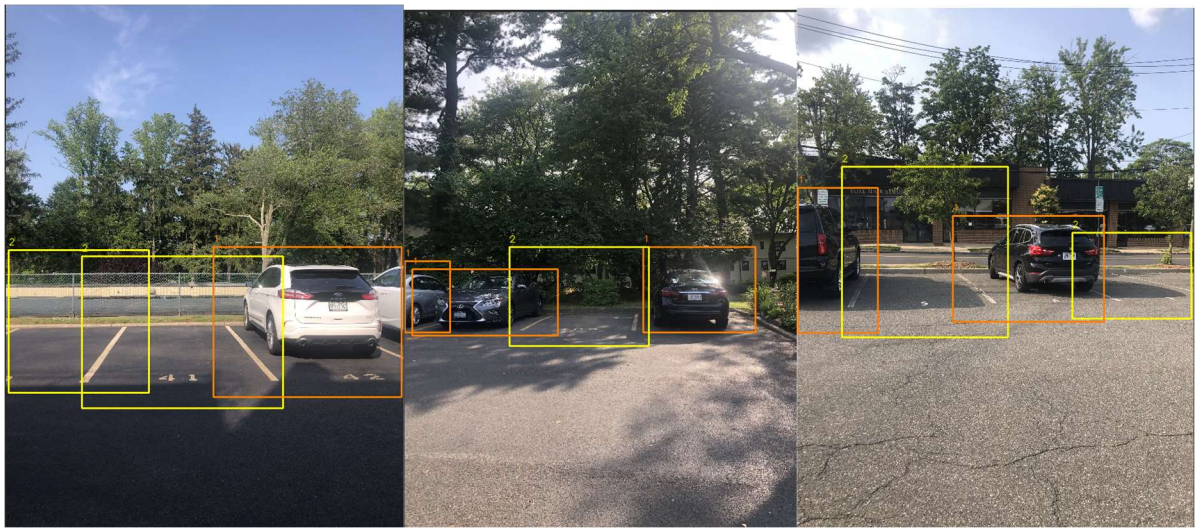


Figure 1. In each image, there are yellow and orange bounding boxes. The yellow boxes represent empty spaces whereas the orange boxes represent occupied spaces.

The dataset used in this research serves as a valuable resource for training the AI model to accurately determine parking space occupancy from images. The subsequent sections will explain details of the training methodology, experimental results, and the model's potential real-world applications.

Methodology/Models

YOLOv5 (You Only Look Once), a popular and effective object detection algorithm, was used for this project [22]. Python code from the YOLOv5 package was executed in a Google Colab notebook to create and train the model from the custom dataset. The YOLO algorithm is a deep learning-based object detection method known for its real-time detection capabilities [8][9][10]. It performs both object localization (identifying the location of one or more objects in an image and drawing a bounding box around it) and classification (naming the bounding boxes) simultaneously. YOLO uses Convolutional Neural Networks (CNN) to detect images. Convolutional Neural Networks (CNN) is a neural network that looks at a whole group of pixels in an image [11][12]. This allows for feature learning where the model can analyze the image thoroughly allowing it to spot any features that will help to make a prediction. The darknet architecture of YOLOv5 supports various types of layers including CNN and pooling layers. CNNs employ convolutional layers to extract relevant features from the images such as edges, textures, and shapes associated with parking spaces. Pooling layers in the CNN reduce the spatial dimensions of the feature map by combining the outputs of neuron clusters at the previous layer into a single neuron in the next layer allowing for the model to intake the most useful information and leave out insignificant features in an image.

During the training process, the YOLO model used loss functions to improve accuracy. YOLOv5's loss function is composed of three parts: box_loss, obj_loss, and cls_loss [13][6]. These loss functions are applied to maximize the objective of mean average precision, a measurement used to measure the performance of computer vision models [14]. After every epoch, the model is tested on the validation dataset and an overall loss value is calculated. The training ends when the number of epochs allotted (300 and 50) is completed. In order to detect objects, YOLO divides the input image into a n x n grid and draws bounding boxes along with probabilities of classified objects for each grid cell. This approach allows for efficient and accurate object detection. Anchor boxes are another important component in the training of the model [15]. Anchor boxes are a set of predefined bounding

boxes of a certain height and width. The size of the anchor boxes is typically based on the object size in the training dataset where bigger objects in a training image would result in bigger anchor boxes. These boxes are tiled across the images and capture the specific object classes to be detected which in this case is empty or occupied parking spaces.

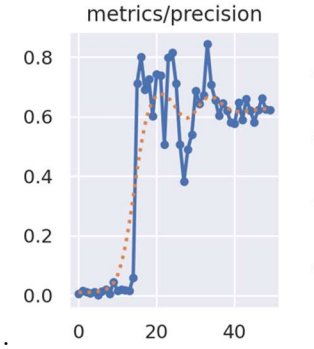
Once the anchor boxes are scattered across the image, a metric such as Intersection Over Union (IoU) is used to determine the probability of a certain object in an image. Intersection Over Union is a number that quantifies the degree of overlap between two boxes [16]. The equation to compute the IoU is the area of overlap of both bounding boxes over the area of union (the ground truth bounding box area + predicted bounding box area - an area of intersection of both bounding boxes). A number between 0 and 1 is calculated where a higher value indicates a greater overlap area and a higher accuracy.

At this point, the model has many bounding boxes shown on the image which makes the image messy and unorganized. Therefore, Non-Maximum Suppression (NMS) method is used to help combine bouncing boxes, but also remove bounding boxes that have an IoU value lower than the NMS threshold [17][18]. NMS selects a single bounding box with the highest confidence score and combines other bounding boxes that have a high overlap with it. NMS helps in reducing duplicate detections and improves the final output by the model.

The model learning procedure consists of the creation of the dataset, preprocessing the data, splitting the dataset, and training the model. As previously stated, I created my own dataset. This way, I ensured that the model was using the most accurate images to train with minimal human error. Once all necessary files were properly uploaded into the Colab notebook, I started the training for 50 and 300 epochs. It took around 2-3 hours for both to finish, and I downloaded all the results produced by the model. Once the training process was completed for both models, I recorded the different predictions produced.

Results and Discussion

Once I trained the model, I went outside and took a video of a car going in and out of a parking space to see if the model could properly detect the empty and occupied parking spaces. The model trained for 300 epochs performed significantly better than the model with 50 epochs when tested on the video and images.



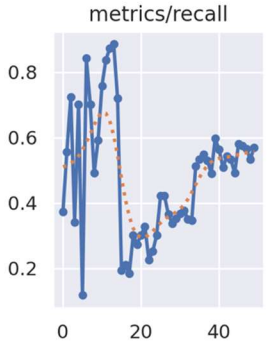
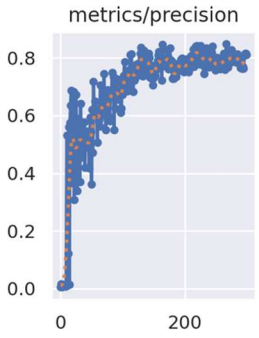


Figure 2. Precision and recall graphs of 50 epoch model



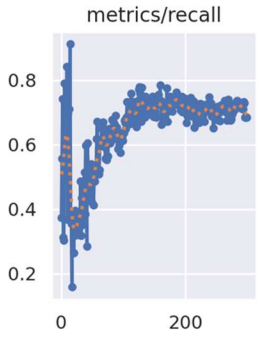


Figure 3. Precision and recall graphs of 300 epoch model

Figures 2 and 3 show the precision and recall metrics of each model. Recall measures the proportion of positive instances (occupied parking spaces) that are correctly identified as positive by the model [19]. In this case, a recall measures a number of occupied parking spaces correctly classified by the model over total number of actual occupied parking spaces. It quantifies the model's ability to avoid false negatives. As shown in the graphs, the model with 300 epochs had an average recall of about 0.7 whereas the model with 50 epochs had an average recall of about 0.55. The higher recall indicated by Figure 3 means that the model was more effective at identifying occupied parking spaces while minimizing the instances where they were incorrectly labeled as vacant. Precision measures the proportion of instances identified as positive by the model that are actually true positives. [19] In this

case, the precision measures number of occupied parking spaces correctly classified over total number of occupied parking spaces classified by the model which may include number of empty parking spaces incorrectly classified. It quantifies the model's ability to avoid false positives. As shown in the graphs, the model with 300 epochs had an average precision of about 0.8 whereas the model with 50 epochs had an average precision of about 0.6. The higher precision indicated by Figure 3 means that the model has a low rate of incorrectly labeling vacant spaces as occupied. Essentially, the graph demonstrates that the model with 300 epochs is more accurate than the model with 50 epochs. In addition to precision and recall measurement, the confusion matrix of both models (Figure 4 and 5) also showed interesting results indicating the model with 300 epochs was more accurate.

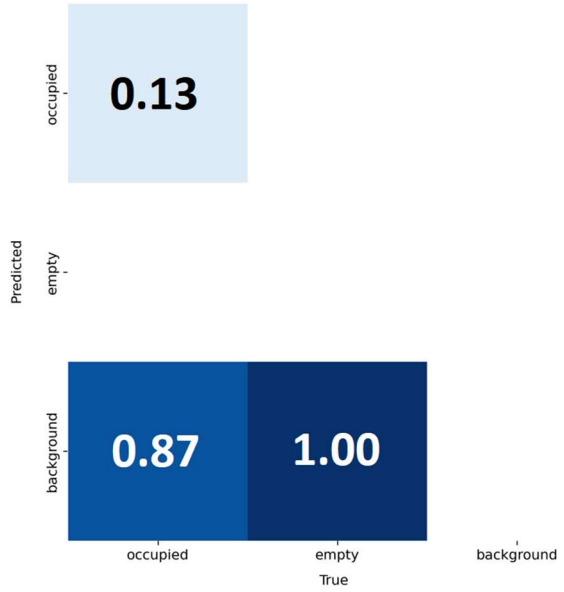


Figure 4. Confusion matrix of 50 epoch model

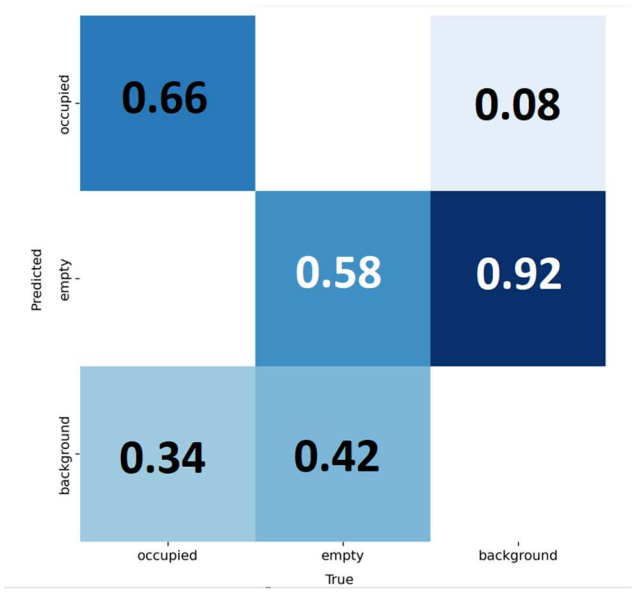


Figure 5. Confusion matrix of 300 epoch model

The figures above are the confusion matrices of each model. Confusion matrices use True Positives (TP; Instances when predicted as occupied and is actually occupied), False Positives (FP; Instances when predicted as occupied although they are actually vacant, instances when predicted as background although they are actually vacant, or instances when predicted as empty although they are actually background), False Negatives (FN; Instances when predicted as vacant although they are actually occupied, instances when predicted as background although they are actually occupied, or instances when predicted as occupied although they are actually background), and True Negatives (TN; Instances when predicted as vacant and they are actually vacant) to represent how well the model performed.

The numbers in the confusion matrix represent the model's accuracy of its prediction. For example, in Figure 5, 0.66 represents the model predicted correctly occupied for 66% of time, whereas 0.34 represents the model predicted background 34% of time although it was occupied. Figure 4 shows the 50 epochs model correctly predicted occupied spaces for 13%, whereas the 300 epochs model correctly predicted occupied for 66% and empty for 58%. In addition, either model did not classify empty as occupied or vice versa. Both models worked to a certain extent that they were able to recognize properly if parking spaces were occupied or not. However, in Figure 4, it was understood that the model with 50 epochs barely recognized the parking space itself. As shown, the model, for the most part, was not able to identify the parking space based on the number in the graph as sum of the values where the model incorrectly predicted background (1.87 = 0.87+1) over total value (2 = 0.13 +0.87+1.0) is very high whereas it is other way around for a situation where the model correctly predicted either an empty or occupied (0.13 over 2). This indicates that the model had a low capability to recognize parking spaces where in most of the predictions being classified as "background".

However, the model with 300 epochs performed much better. Firstly, the TP and TN values were 0.66 and 0.58 respectively. This means that the model predicted 1.24 (0.66+0.58; sum of the values where the model correctly predicted the parking space occupancy) out of 3 (0.66+0.34+0.58+0.42+0.08+0.92; sum of the TP (0.66), TN (0.58), FP (empty(0.42), background(0.92)), and FN (occupied(0.34), background(0.08)) values) which is a significant

improvement compared to the model with 50 epochs. However, interestingly, the AI model predicted 0.92 (value where the model predicted an empty parking space when in reality it was a background) out of 1 (0.92 + 0.08: total value for background). A possible explanation is that the model interprets any image of a car as an occupied space, and anything else as empty. Since the bounding boxes setup in the training data were large, the model might have incorrectly learned that a frame without a car is an empty space. This means that the model may not have spotted parking space features such as two white lines or the numbers that indicate a parking spot. In addition, 0.08 out of 1 false positive which means the model predicted occupied when in reality it was background, could be explained by cars being in the background that were picked up by the model, but were not annotated.

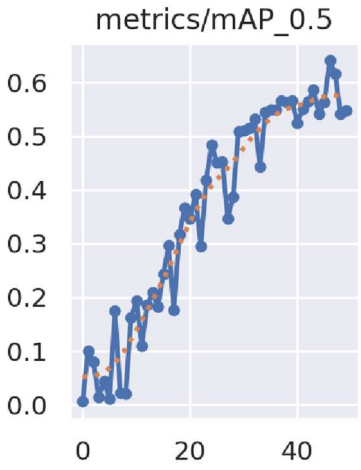


Figure 6. mAP graph of 50 epoch model

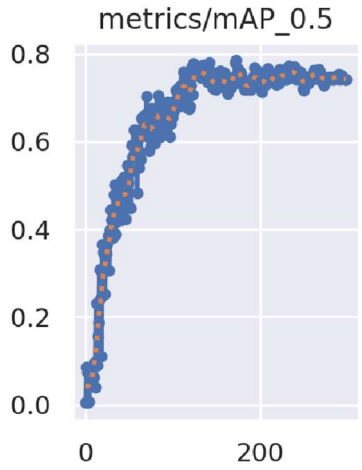


Figure 7. mAP graph of 300 epoch model

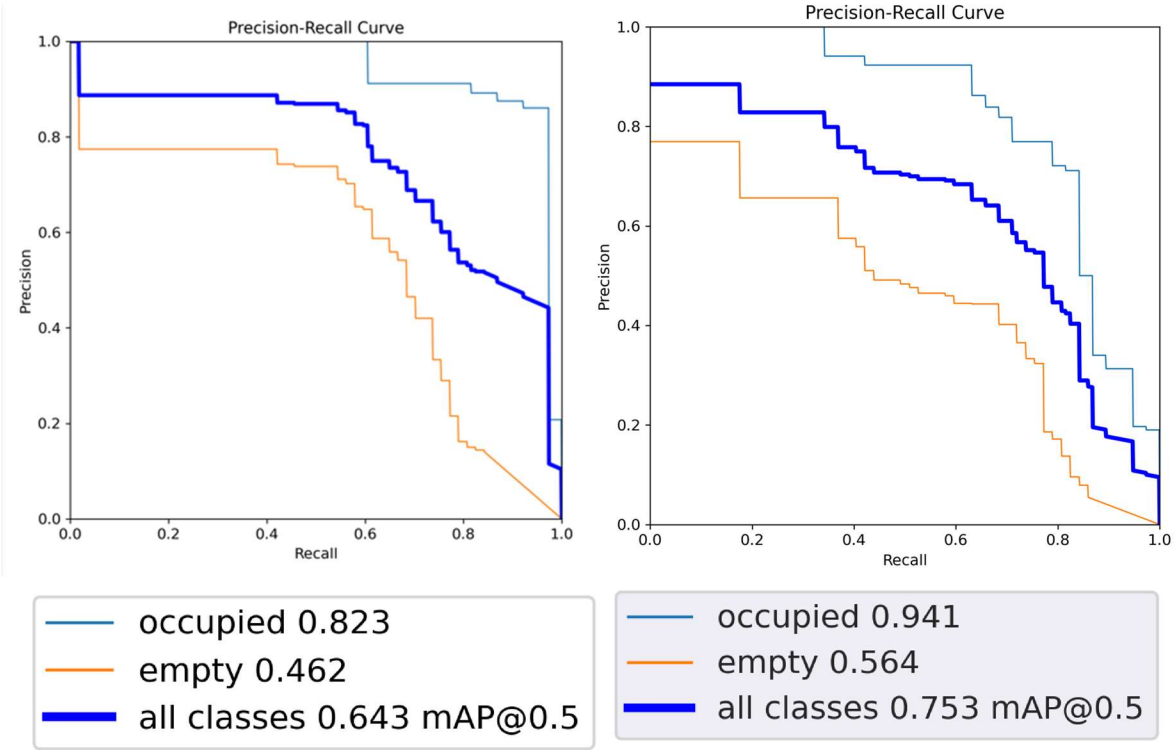
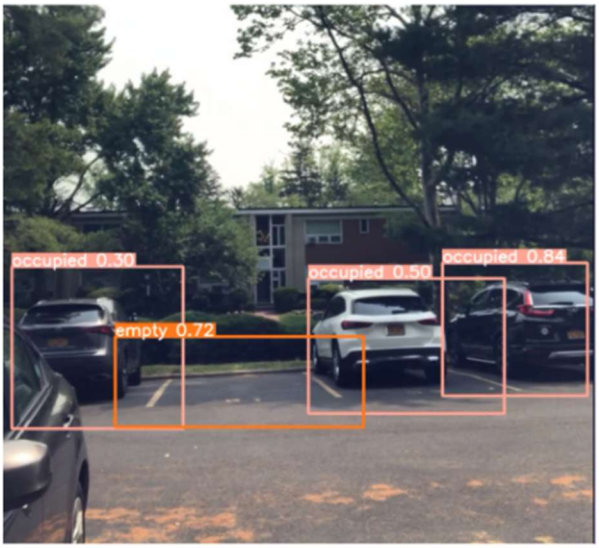


Figure 8. Precision-Recall graph of 50 epoch model

Figure 9. Precision-Recall graph of 300 epoch model

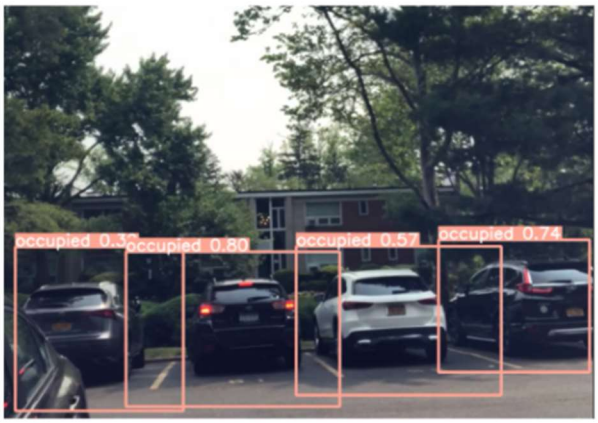
The graphs in Figures 8 and 9 represent the Mean Average Precision (mAP) of each model. Mean average precision is a commonly used evaluation metric in object detection tasks [14]. It measures the accuracy of an object detection model by considering both precision and recall. The Precision-Recall curve is used to compute the mean Average Precision. As shown in Figures 6 and 7, both models have fluctuations, but overall mAP increases towards the end. The mAP for the 50 epochs model is about 0.6 at the end of the training, whereas the mAP for the 300 epochs model is about 0.75. A higher mAP indicates better performance, meaning the 300 epochs model is capable of more accurately identifying parking spots within the parking lot image or video. This statement corresponds to the data represented in Figures 2 and 3, as the 50 epochs model had a lower precision and recall value than the 300 epochs model, as well as Figures 4 and 5 where the matrices showed improved accuracy of model with the 300 epochs compared to the 50 epochs.

10A. Results of video input for 50 epoch model without car



10B. Results of video input for 300 epoch model without car

10C. Results of video input for 50 epoch model with car



10D. Results of video input for 300 epoch model without car

Figure 10

Lastly, the images shown in Figure 10 are the predictions made by both models from a video of a car entering and leaving a parking space. As previously explained, the 300 epochs model had a significantly higher accuracy than the 50 epochs model. Figures 10A and 10C represent the predictions made by the 50 epochs model, and it is clear that the model was not able to accurately predict anything which corresponds to the confusion matrix which predicted mostly everything as “background”. However, Figures 10B and 10D showed a totally different result. In both images, it was clear that the model with 300 epochs was able to predict all the parking spots perfectly with most having a relatively high confidence value. This result also corresponds to the metrics collected where prediction for parking lot occupancies using a model with 300 epochs is significantly better than the model with 50 epochs.

Conclusion

In this research paper, the objective was to analyze the effectiveness of the YOLO model in accurately predicting the occupancy of parking spaces based on a custom-made dataset containing images of empty and occupied parking spaces so that it can be used for automated system capable of real-time parking occupancy detection. The model trained with 50 epochs achieved an accuracy of approximately 6.5%, while the model trained with 300 epochs achieved an accuracy of about 41.3%. Although both models exhibited some level of accuracy, their performance fell short of being suitable for integration into automated parking systems.

The outcome of this study gives us hints for several factors that may have caused limited success of the models. One possibility is that the models were not trained for a sufficient number of epochs as evidence showed significant improvement observed by increasing the number of epochs from 50 to 300. Additionally, the large image size in the dataset might have hindered the models' learning capabilities, prompting consideration for using resized images in future experiments. The dataset's lack of variety, particularly in terms of different angles and perspectives of parking spaces, could have limited the models' ability to generalize effectively. Future research in this area could benefit from addressing these limitations. In addition, by exploring different CNN architectures and variations, researchers may discover improved models with enhanced accuracy. Acquiring a more extensive and diverse dataset that encompasses various parking environments could also enhance the models' ability to generalize across different scenarios. Furthermore, the investigation of alternative data preprocessing techniques such as image augmentation and feature extraction may lead to performance gains. While the models used in this research did not reach the desired level of accuracy for integration into automated parking systems, this research highlights the positive impact of increasing the number of epochs on the AI model's accuracy in determining parking space occupancy from images. Despite the limitations encountered, the study opens the door to more efficient parking management solutions. The insights gained from this research can inform future endeavors to develop more robust and accurate AI-based parking occupancy detection systems, potentially revolutionizing the way parking spaces are managed and improving user experiences.

Acknowledgements

I would like to thank the staff at Eastchester High School for helping me throughout the process and guiding me through the process of writing this paper. I would also like to thank my parents for supporting me throughout the entire process.

References

[1] "Urban Development." *World Bank*, www.worldbank.org/en/topic/urbandevelopment/overview. Accessed 28 June 2023.

[2] "68% of the World Population Projected to Live in Urban Areas by 2050, Says Un | UN Desa Department of Economic and Social Affairs." *United Nations*, www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html. Accessed 28 June 2023.

[3] Noyes, Lexie. "Stop Wasting Time Searching for Parking. Park Smarter." *SpotHero Blog*, 12 Dec. 2017, blog.spothero.com/park-smarter-parking-search-time.

[4] "Automated Parking System Market Projected to Achieve a Valuation of US$ 5.2 Billion by 2032, with Europe Leading the Market Share at 42%." GlobeNewswire News Room, 11 Apr. 2023, www.globenewswire.com/en/news-release/2023/04/11/2644888/0/en/Automated-Parking-System-Market-projected-to-achieve-a-valuation-of-US-5-2-billion-by-2032-with-Europe-leading-the-market-share-at-42.html.

[5] "Automated Parking System Market Size, Share & Covid-19 Impact Analysis, by Application Type (Commercial Parking, Residential Parking), by Automation Level (Fully Automated, Semi-Automated), by Component Type (Hardware, Software), and Regional Forecasts, 2021-2028." *Automated Parking System Market Size, Growth | Forecast, 2028*, www.fortunebusinessinsights.com/automated-parking-system-market-105486. Accessed 30 June 2023.

[6] Solawetz, Jacob. "What Is YOLOv5? A Guide for Beginners." *What Is YOLOv5? A Guide for Beginners*, Roboslow, 29 June 2020, https://blog.roboflow.com/yolov5-improvements-and-evaluation/. Accessed 6 July 2023.

[7] Kathuria, Ayoosh. "How to Train Yolo V5 on a Custom Dataset." *Paperspace Blog*, 10 Apr. 2023, blog.paperspace.com/train-yolov5-custom-data/.

[8] Jiang, Peiyuan, et al. *A Review of Yolo Algorithm Developments*, 2021 (link).

[9] "Yolo Algorithm for Object Detection Explained [+examples]." YOLO Algorithm for Object Detection Explained [+Examples], www.v7labs.com/blog/yolo-object-detection. Accessed 30 June 2023.

[10] "Introduction to Yolo Algorithm for Object Detection." *Section*, www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/. Accessed 30 June 2023.

[11] "What Is a Convolutional Neural Network?: 3 Things You Need to Know." What Is a Convolutional Neural Network? | 3 Things You Need to Know - MATLAB & Simulink, www.mathworks.com/discovery/convolutional-neural-network-matlab.html. Accessed 3 July 2023.

[12] Awati, Rahul. "What Are Convolutional Neural Networks?: Definition from TechTarget." Enterprise AI, 24 Apr. 2023, www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network.

[13] Lihi Gur Arie, PhD. "The Practical Guide for Object Detection with Yolov5 Algorithm." *Medium*, 14 Feb. 2023, towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843.

[14] Solawetz, Jacob. “Mean Average Precision (MAP) in Object Detection.” *Roboflow Blog*, 25 Nov. 2022, blog.roboflow.com/mean-average-precision/.

[15] Anchor Boxes for Object Detection - MATLAB & Simulink, www.mathworks.com/help/vision/ug/anchor-boxes-for-object-detection.html. Accessed 6 July 2023..

[16] Kukil. “Intersection over Union IOU in Object Detection Segmentation.” LearnOpenCV, 13 Feb. 2023, learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/#:~:text=Intersection%20Over%20Union%20(IoU)%20is,Ground%20Truth%20and%20Prediction%20region.

[17] Hosang, Jan, et al. “A Convnet for Non-Maximum Suppression.” *arXiv.Org*, 8 Jan. 2016, https://arxiv.org/pdf/1511.06437.pdf.

[18] “Papers with Code - Non Maximum Suppression Explained.” *Explained | Papers With Code*, paperswithcode.com/method/non-maximum-suppression#:~:text=Non%20Maximum%20Suppression%20is%20a,below%20a%20given%20probability%20bound. Accessed 6 July 2023.

[19] Fessel, Kimberly, director. Never Forget Again! // Precision vs Recall with a Clear Example of Precision and Recall. YouTube, YouTube, 1 Mar. 2021, https://www.youtube.com/watch?v=qWfzIYCvBqo&t=5s&ab_channel=KimberlyFessel. Accessed 8 July 2023.

[20] LILINOpenGitHub. “LILINOpenGitHub/Labeling-Tool: Yolo Ai Labeling Tool.” *GitHub*, github.com/LILINOpenGitHub/Labeling-Tool. Accessed 20 July 2023.

[21] Alam, Mahbubul. “Data Normalization in Machine Learning.” *Medium*, 14 Dec. 2020, towardsdatascience.com/data-normalization-in-machine-learning-395fdec69d02.

[22] Ding, Xiangwu, and Ruidi Yang. *Vehicle and Parking Space Detection Based on Improved YOLO ... - Iopscience*, 2019, https://iopscience.iop.org/article/10.1088/1742-6596/1325/1/012084/pdf.

[23] Siddiqui, Shahan Yamin, et al. “Smart Occupancy Detection for Road Traffic Parking Using Deep Extreme Learning Machine.” *Journal of King Saud University - Computer and Information Sciences*, 6 Feb. 2020, www.sciencedirect.com/science/article/pii/S1319157819313928.