

A Deep Learning Based Framework for Blind Road Occupancy Detection

Yiteng Liu*

Shanghai Pinghe School, Shanghai, China

*Corresponding author: liuyiteng@shphschool.com

Abstract. This research paper addresses the pressing concern of road safety for the visually impaired in densely populated regions, especially in China. While tactile paving exists to guide blind individuals, unexpected obstacles pose serious hazards. The study proposes an Artificial Intelligence (AI) system utilizing advanced machine learning techniques to identify obstacles on blind roads, providing real-time feedback for navigation. To overcome data limitations, a virtual environment using Pybullet is created for data generation, combining synthetic and real-world images for training. This study introduces a Convolutional Neural Network (CNN)-based model and integrates a Vision Transformer (ViT) model, comparing their efficacies. The progressive training approach yields a highly effective CNN model, outperforming ViT models. The practical application of the CNN model in real-world scenarios has proven its efficacy in detecting obstacles, underscoring its reliability and significant contribution to improving road safety for visually impaired individuals. This research extends beyond providing a safety enhancement measure; it also sheds light on the wider applicability of such methods in addressing issues like the scarcity of real-world data.

Keywords: Machine learning, computer vision, blind road, PyBullet, convolutional neural network, vision transformer.

1. Introduction

Artificial Intelligence (AI) has dramatically transformed a myriad of sectors, with computer vision emerging as one of its most significant breakthroughs. As the technological world progresses, AI's potential for serving humanity grows, aligning with global efforts to build a more inclusive society. One such pressing concern, especially in densely populated regions like China, is road safety. For the visually impaired, navigating urban landscapes can be particularly challenging. In China, 'blind road' (i.e. tactile paving) is prevalent, designed specifically to guide blind individuals safely. However, these pathways sometimes harbor unexpected obstacles that can pose serious hazards.

While guide dogs have been instrumental in assisting the blind, they come with inherent limitations – they require extensive training, can be expensive, and are not a feasible solution for everyone. Tactile canes, on the other hand, not only offer limited information about the environment, potentially missing certain obstacles, but also require continuous waving by the user. This continuous motion can be physically exhausting over longer durations, further emphasizing the need for a more autonomous solution.

AI research within the domain of transportation encompasses an extensive array of subjects and practical implementations. Traditional categorizations of AI literature in transportation span topics e.g. traffic management [1], obstacle detection [2], and path planning [3], representing areas that have received extensive and concentrated academic attention. For example, The Faster R-CNN [4] introduced a novel architecture for object detection that integrates Region Proposal Networks (RPN) to generate potential object regions in an image, and He et al. [5] proposed Mask R-CNN and achieved state-of-the-art results in various benchmark datasets and is widely recognized for its effectiveness in simultaneously addressing object detection and segmentation tasks. While these areas related to computer vision have received substantial attention, the application of AI on blind roads remains notably unexplored. Although there are scattered few studies that focus on blind road detection, object detection on blind roads is a rarely studied topic. Hence, this research endeavors to bridge the existing void by crafting an AI system that employs advanced machine learning techniques. The primary goal

is to precisely identify obstacles on blind roads, addressing the current dearth of comprehensive studies in this domain. The system can provide real-time feedback to users, allowing them to adjust their paths and make decisions during navigation.

This work offers a notable contribution through its innovative approach to data acquisition. Acknowledging the practical limitations in gathering extensive real-world data, especially considering the complexities of human labeling, this study introduces a pragmatic solution. A realistic virtual environment using the open-sourced simulation Pybullet was crafted to generate data. These synthetically rendered images serve as a robust source for training data. When strategically combined with a sparse set of real-world images, this approach facilitates the efficient training of the AI models. This use of a virtual environment successfully navigates the issues related to data limitations, establishing a practical and effective method for training AI models with notable precision and effectiveness.

2. Method

2.1. Data Generation

As few datasets about blind tracks can be found, this study used Pybullet to generate data to train the proposed AI model as shown in Fig. 1. In this way, this study can make full use of the limited real-world data resource, as any number of simulation data can be generated.

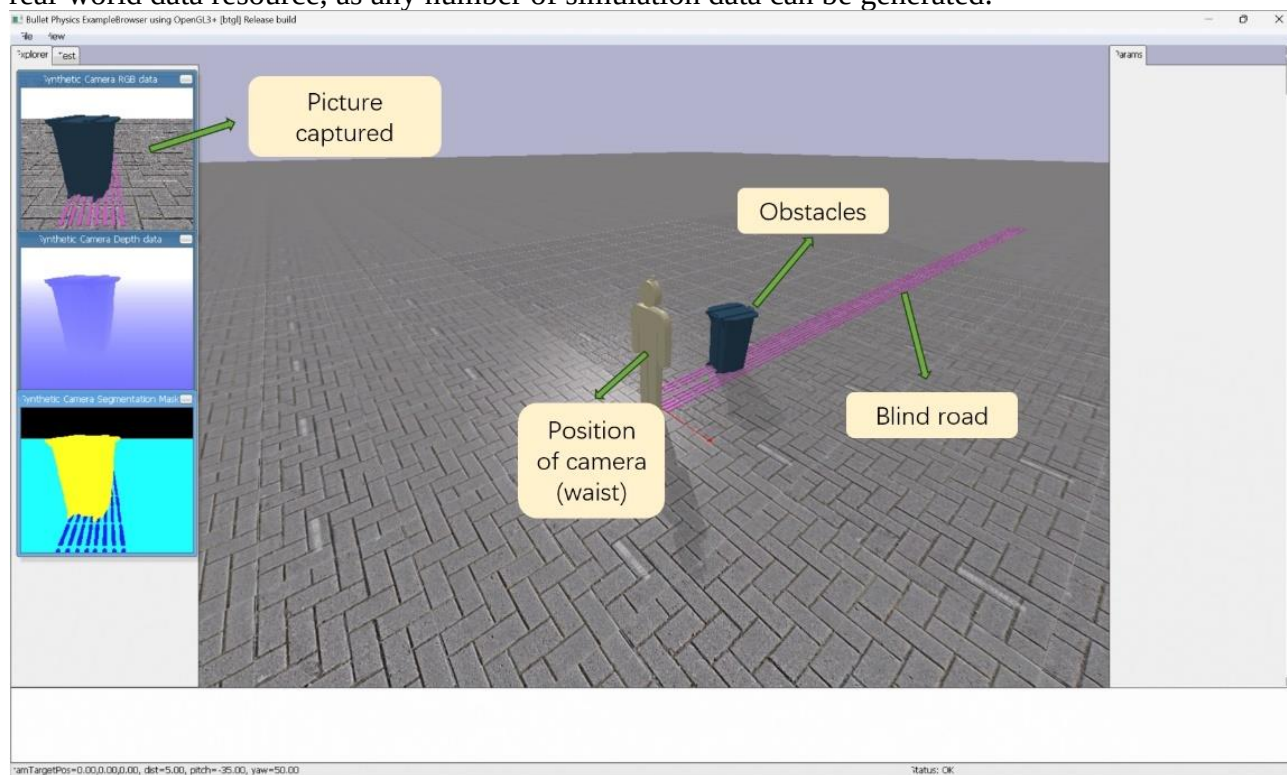


Fig. 1 The Pybullet simulator and the models, the pictures captured by the camera are shown on the left side of the picture (Photo credit : Original).

Pybullet stands out as a rapid and user-friendly Python module designed for simulating robotics and supporting machine learning applications, particularly emphasizing seamless transitions from simulation to real-world scenarios. Through Pybullet, users can effortlessly import articulated bodies from various file formats e.g. URDF, SDF, and MJCF. This versatile module encompasses functionalities ranging from forward dynamics simulation, inverse dynamics computation, and forward and inverse kinematics to collision detection and ray intersection queries [6]. 错误!未找到引用源。 The simulation allows this study to generate plentiful data that will be utilized in the training process. Fig. 2 below presents the process of generating simulation data.

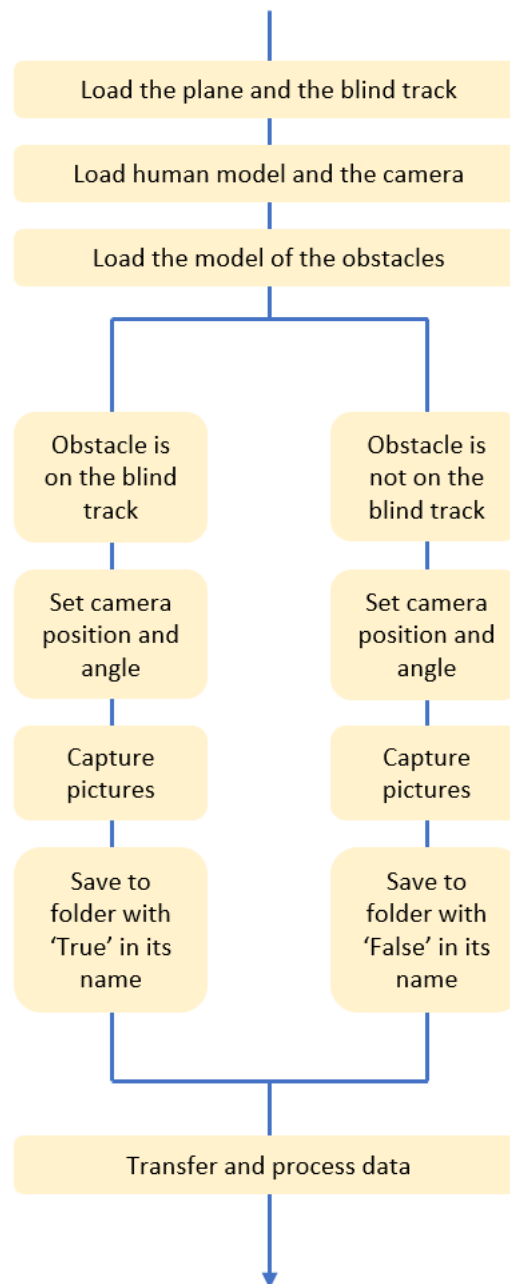


Fig. 2 The general process of generation simulation data (Photo credit : Original).

In terms of the URDF files that are loaded into the simulator, this study used SOLIDWORKS (which is a CAD software) to draw and export them. The URDF files used for data generation include the road model (which is the blind track, see Fig. 3), the human model (see Fig. 4) and the other URDF models.

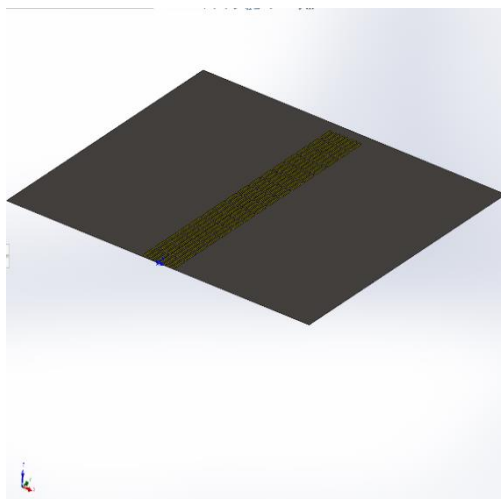


Fig. 3 The model for blind track (Photo credit : Original).

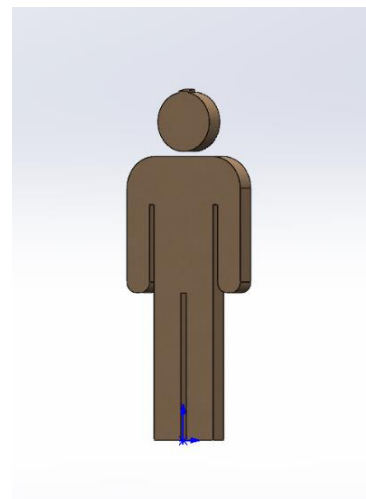


Fig. 4 The human model (Photo credit : Original).

For the blind track model, it was drawn according to the blind track brick standards as illustrated in Fig. 5.

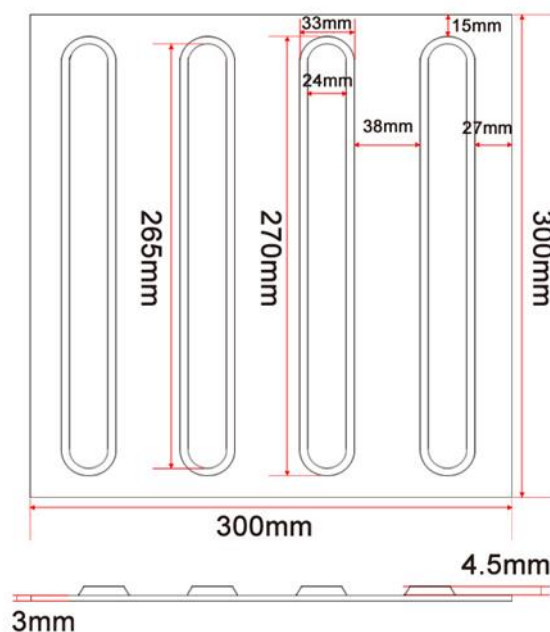


Fig. 5 the standard for blind road bricks [7].

When configuring the camera within the simulation environment, this study employed a strategy to enhance the diversity and robustness of the generated training data. Firstly, this research introduced randomness to the camera's position, allowing it to vary within a predefined range of 0 to 0.15 units. Additionally, the camera's orientation was subjected to random rotation, spanning an angle range from -15 degrees to 15 degrees. These deliberate variations in both position and rotation contribute significantly to the realism and adaptability of the training data. As a result of these adjustments, the camera's perspective is situated at the waist level of the simulated human model, replicating a practical viewpoint for blind road obstacle detection.

Within the simulation, this research incorporated various URDF models, including objects such as trash bins and traffic cones. These models play a crucial role in simulating real-world obstacles that may be encountered on blind roads. Fig. 6 demonstrates the wide variety of obstacle models.

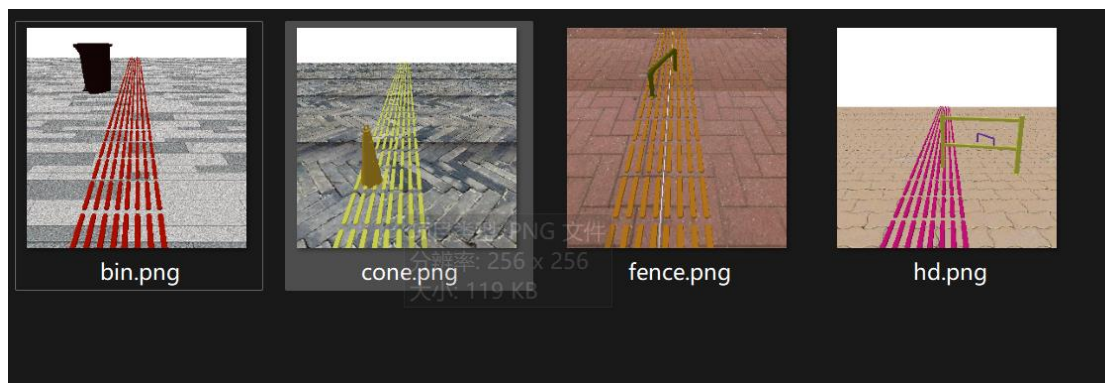


Fig. 6 The captured pictures of different kinds of obstacles (Photo credit : Original).

When loading these models into the simulator, this research employed a multi-step approach. Initially, each model is assigned a random initial position within the environment. Notably, this research exercise precise control over the x-axis positioning to determine whether the object resides on the blind track or not. Concurrently, the y-axis positioning is managed to ensure that the object falls within the camera's field of view, guaranteeing it will be captured in the generated images.

Subsequently, this research meticulously organizes and categorizes the images into corresponding folders, utilizing binary labels such as 'True' or 'False.' These labels serve the essential purpose of indicating whether the blind track is occupied by an obstacle in each captured image. This structured data management simplifies the subsequent data processing stages and facilitates efficient label assignment for the training dataset. A visual representation is shown in Fig. 7 and Fig. 8. This systematic approach enables this study to create a well-annotated dataset that is indispensable for training and evaluating the blind road obstacle detection model.



Fig. 7 The example of images saved (Photo credit : Original).

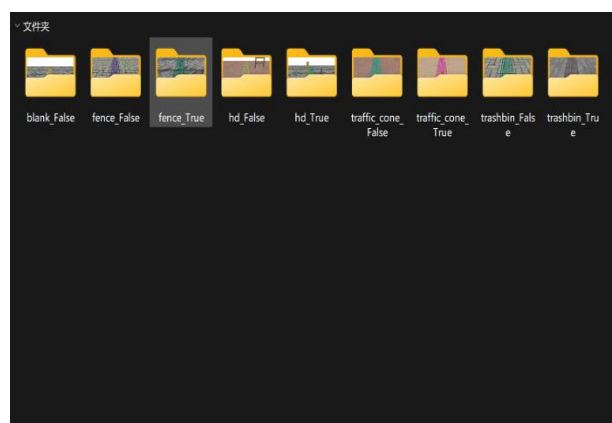


Fig. 8 The example of folders (Photo credit : Original).

2.2. Data Processing

After generating the raw data, the labels were assigned to them. Since they have been grouped based on whether the blind track is occupied, this study assigned 1 to all images in folders with 'True' in its name and 0 to all images in folders with 'False' in its name.

After labelling the images, this study split them into train dataset and validation dataset. The proportion of these datasets are set to be 80% and 20%. Fig. 9 shows the general process of transferring and processing simulation data.

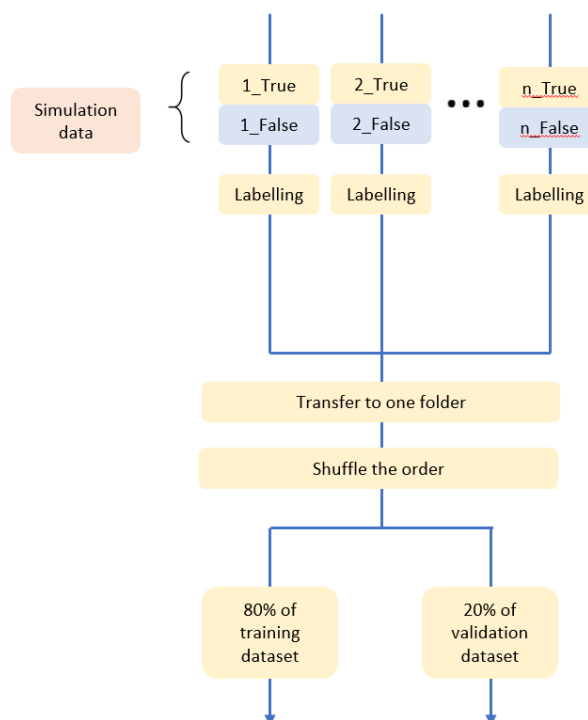


Fig. 9 The general process of transferring and processing simulation data (Photo credit : Original).

In summary, PyBullet serves as a versatile tool for creating synthetic data that closely mimics real-world blind road scenarios. By utilizing these simulated data in the training pipeline, this study bootstrapped the models' learning process and enhance their ability to detect obstacles and navigate complex environments. The generated simulation data is totally based on real world scenes. Fig. 10 and Fig. 11 show that they are highly similar.

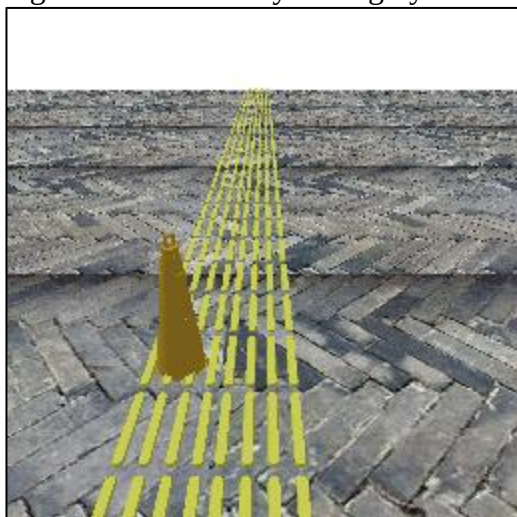


Fig. 10 The simulation data of blind road that is generated with PyBullet (Photo credit : Original).



Fig. 11 The real-world picture that is captured in Shanghai (Photo credit : Original).

2.3. Convolutional Neural Network Model

In the heart of this research's proposed system leverages the Convolutional Neural Network (CNN) [8-12], which is known for its prowess in image recognition tasks. This research designed a CNN-based model, aiming to tap into its inherent strengths to facilitate the visually impaired in their interactions with their surroundings, especially on blind roads. The details of the CNN-based model can be viewed in the picture Fig. 12 below:

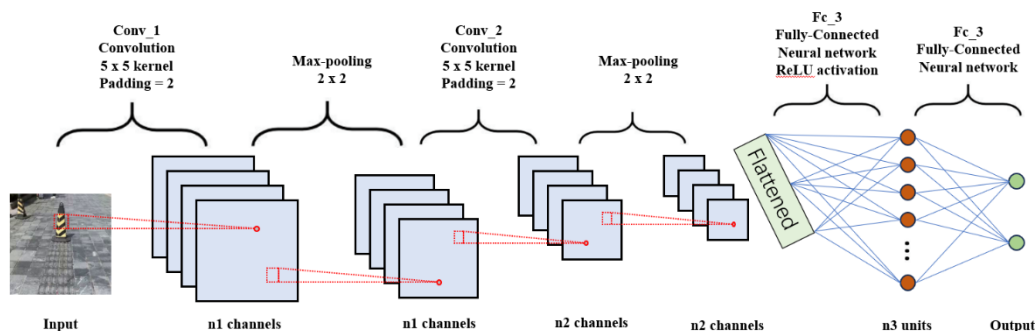


Fig. 12 The CNN framework adopted in the research (Photo credit : Original).

2.4. Vision Transformer Model

Further enriching the proposed approach is the integration of the Vision Transformer (ViT) model. Originating from the foundational Transformer architecture [13] – which has dramatically advanced the fields of natural language processing – the Vision Transformer variant has been pivotal in image processing tasks. This study has harnessed the ViT model not only to bolster the model's performance but also to draw a comparative analysis between the efficacies of CNN and ViT, particularly when dealing with limited datasets. The architecture of the adopted ViT-based model is depicted in the subsequent image Fig. 13:

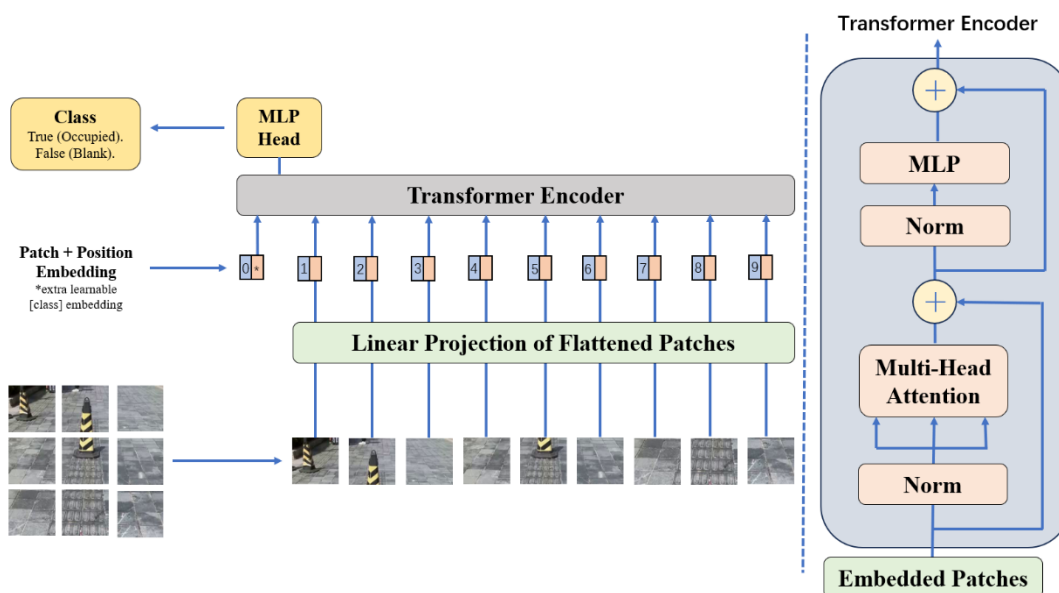


Fig. 13 The ViT framework adopted in the research (Photo credit : Original).

3. Results and Discussion

3.1. Model Training

For both the CNN and the ViT model, this study trained three separate models for each. The first set of models are trained with all simulation photos. The learning curve for the models are presented below in Fig. 14 to Fig. 17:

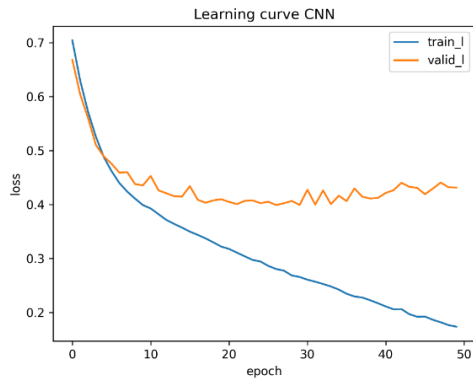


Fig. 14 The loss curves for CNN model with simulation data (Photo credit : Original).

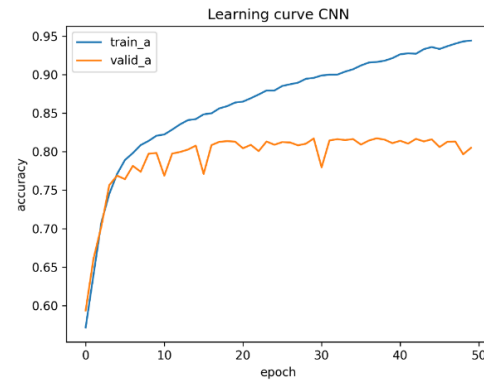


Fig. 15 The accuracy curves for CNN model with simulation data (Photo credit : Original).

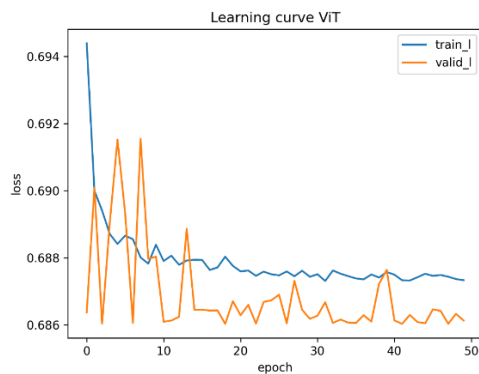


Fig. 16 The loss curves for ViT model with simulation data (Photo credit : Original).

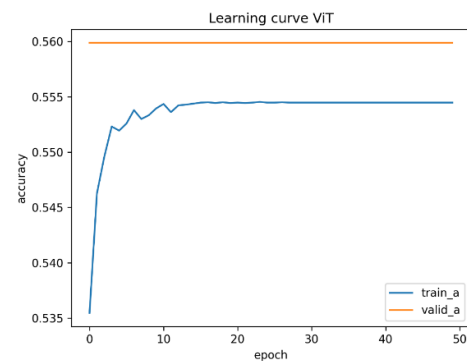


Fig. 17 The accuracy curves for ViT model with simulation data (Photo credit : Original).

The second set of models is trained only with real-world data. The learning curve for the models are presented below in Fig. 18 to Fig. 21:

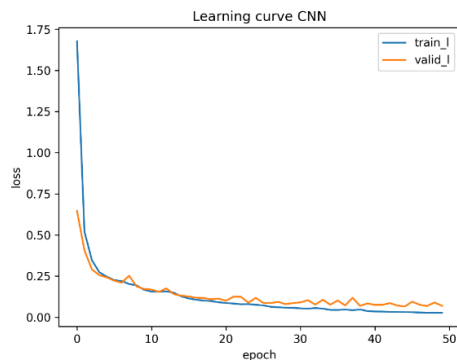


Fig. 18 The loss curves for CNN model with real-world data (Photo credit : Original)

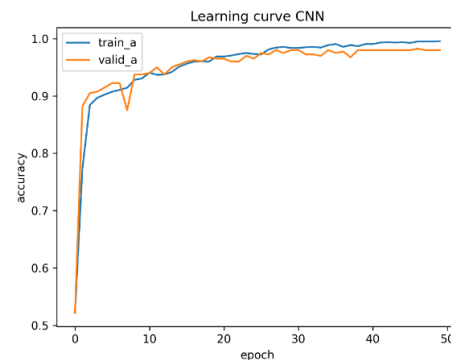


Fig. 19 accuracy curves for CNN model with real-world data (Photo credit : Original)

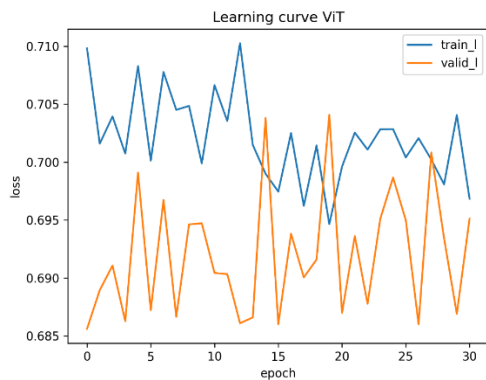


Fig. 20 The loss curves for ViT model with real-world data (Photo credit : Original)

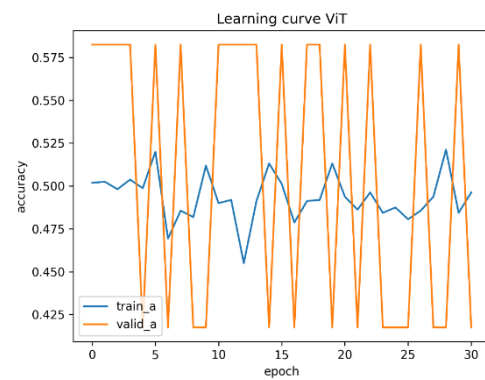


Fig. 21 The accuracy curves for ViT model with real-world data (Photo credit : Original)

A third set of models is formed by continuing to train models trained using simulation data of real-world data. The learning curve for the models are presented below in Fig. 22 to Fig. 25:

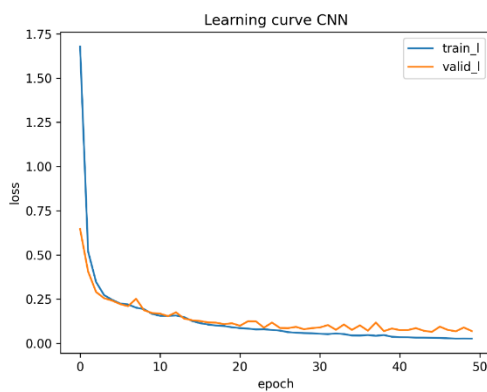


Fig. 22 loss curves for the final version of the CNN model (Photo credit : Original)

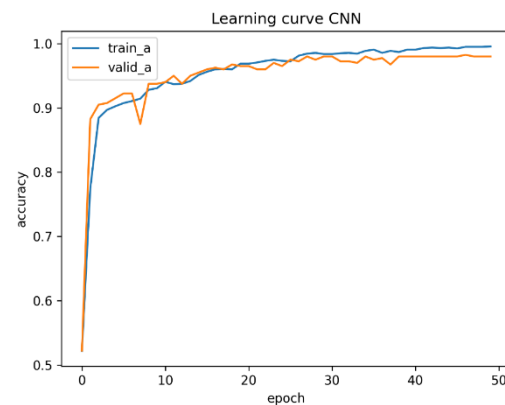


Fig. 23 accuracy curves for the final version of the CNN model (Photo credit : Original)

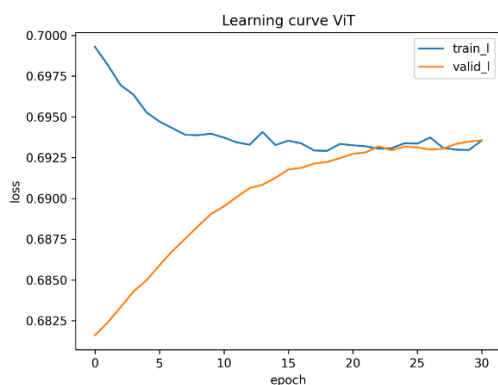


Fig. 24 loss curves for the final version of the ViT model (Photo credit : Original)

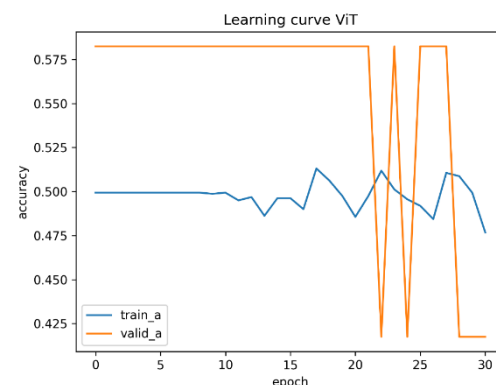


Fig. 25 accuracy curves for the final version of the ViT model (Photo credit : Original)

3.2. Model Testing

This research tested the trained models with two real-life models. The first video contains few elements in it and the obstacles (if exist) are relatively clear in every frame of the video. For the second video, there are far more elements in the videos, which will make it more difficult for the AI models to identify them.

For the first video, the examples of frames extracted from it are presented below in Fig. 26 and Fig. 27:



Fig. 26 the blank blind road in the first video
(Photo credit : Original)



Fig. 27 the blind road that is occupied by a tricycle in the first video (Photo credit : Original)

The test results of the first video are shown below in Fig. 28 and Fig. 29:

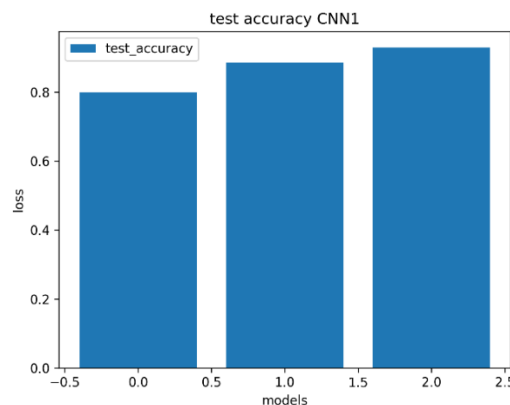


Fig. 28 The test accuracy for the three CNN models (simulation, real-world only, and final version from left to right), the accuracy being 0.8, 0.886 and 0.93 (Photo credit : Original)

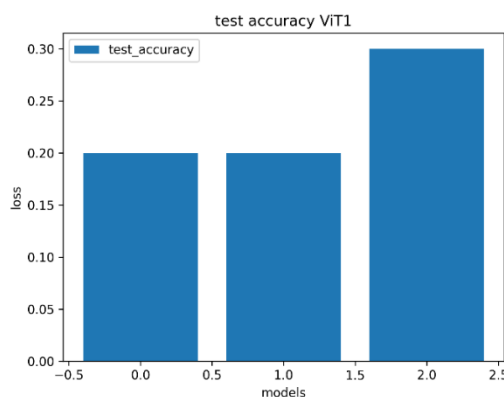


Fig. 29 The test accuracy for the three ViT models (simulation, real-world only, and final version from left to right), the accuracy being 0.2, 0.2 and 0.3 (Photo credit : Original)

For the second video, the example frames extracted are shown below in Fig. 30 and Fig. 31:



Fig. 30 The blank blind road in the second video (Photo credit : Original)



Fig. 31 The blind road that is blocked by bicycles in the second video (Photo credit : Original)

The test results of the second video are shown below in Fig. 32 and Fig. 33:

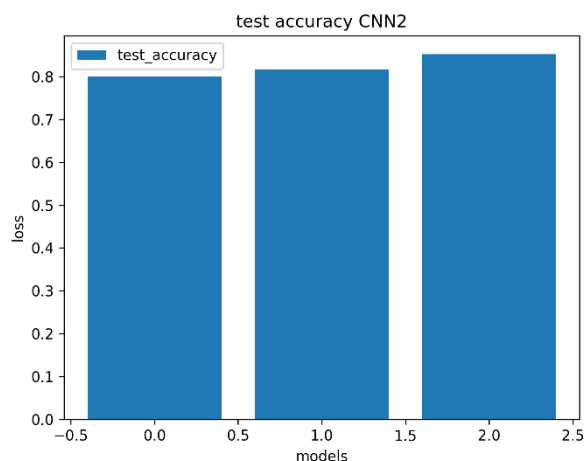


Fig. 32 The test accuracy for the three CNN models (simulation, real-world only, and final version from left to right), the accuracy being 0.8, 0.816 and 0.853 (Photo credit : Original)

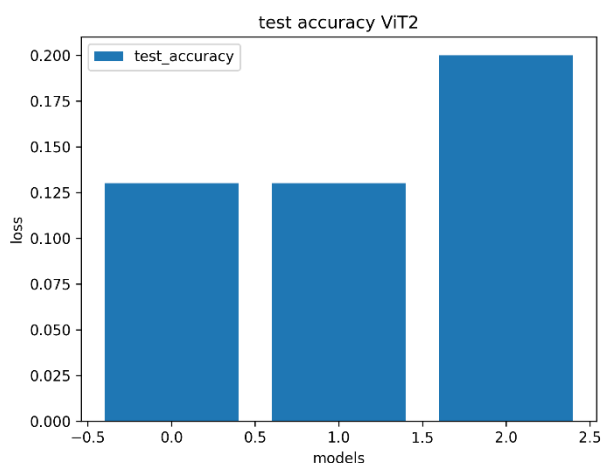


Fig. 33 The test accuracy for the three ViT models (simulation, real-world only, and final version from left to right), the accuracy being 0.13, 0.13 and 0.2 (Photo credit : Original)

According to the results, while the CNN model demonstrated a remarkable performance in the case, the ViT model struggled to achieve comparable accuracy levels. One of the primary factors influencing the results of this study is the limited amount of data available for training. The ViT model,

being a large architecture, demands a substantial volume of data to generalize effectively. The observations suggest that the small dataset provided may not have been sufficient to fully harness the potential of the ViT model. This underscores the critical importance of data quantity in training deep learning models, especially those with extensive parameters. The performance issue aligns with the insights presented in the essay titled "Understanding the Difficulty of Training Transformers" [14], which identifies strong dependencies on residual branches as a key contributor to training instability. Admin, as proposed in the paper, introduces an adaptive initialization approach to mitigate these dependencies and stabilize Transformer training. Admin's method involves controlling dependencies at the outset of training while maintaining flexibility to capture them as training stabilizes, which could possibly reduce the adverse effects of strong dependencies and, consequently, improve the model's training stability.

Despite the success of the CNN model, it is important to acknowledge its limitations. In complex environments where there are numerous irrelevant elements in the images, the model faces challenges in making accurate decisions. The model will be hardly able to make precise decisions facing a picture complex like Fig. 34, which is a picture taken in Shenzhen. This limitation highlights the need for continued research and improvements, particularly in developing attention mechanisms or filtering techniques to focus on relevant information while ignoring distractions in the images.



Fig. 34 The blind road in Shenzhen (Photo credit : Original).

4. Conclusion

In conclusion, this AI research project aimed to develop a model for detecting obstructions and obstacles on blind roads, such as bicycles, cars, and roadblocks. This research pursued a multi-step training approach, beginning with simulated data and transitioning to real-world data, which ultimately resulted in the creation of a highly effective CNN model. The findings and outcomes of this study can be summarized as follows:

Progressive Training Approach: This research demonstrated the effectiveness of a two-stage training process, starting with simulated data and then transitioning to real-world data. This approach led to significant improvements in model performance, highlighting the importance of gradually introducing real-world complexities into the training process.

Model Performance: The final version of the CNN model achieved impressive test accuracy scores, with values of 0.93 and 0.853 in two separate tests. This high level of accuracy indicates the model's proficiency in detecting obstacles on blind roads.

Comparison with ViT Model: This study's experiments revealed that CNN models outperformed ViT models significantly. The CNN model consistently achieved higher accuracy scores, reinforcing the suitability of CNNs for image-based tasks like obstacle detection in blind road scenarios.

Real-World Generalization: The ability of the final CNN model to perform well on real-world data underscores its robustness and capacity to adapt to the challenges of detecting obstacles in actual blind road situations.

In summary, this research demonstrated the success of a multi-step training strategy that combines simulated and real-world data, ultimately resulting in a highly accurate CNN model for blind road obstacle detection. These findings emphasized the importance of choosing the right architecture for the task at hand, as CNNs proved superior to ViT models in this context. This study's work contributed to the advancement of AI-based solutions for enhancing road safety and could have valuable applications in assisting visually impaired individuals in navigating their environments.

References

- [1] Ma X, Dai Z, He Z, et al. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 2017, 17(4): 818.
- [2] Redmon J, Farhadi A. YOLO9000: better, faster, stronger. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 7263-7271.
- [3] Kavraki L E, Svestka P, Latombe J C, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 1996, 12(4): 566-580.
- [4] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 2015, 28.
- [5] He K, Gkioxari G, Dollár P, et al. Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*. 2017: 2961-2969.
- [6] Pybullet quickstart guide [EB/OL]. Google Docs, Google, 2023, <https://docs.google.com/document/d/10sXEhzFRSnvFcl3XxNGhnD4N2SedqwdAvK3dsihxVUA/edit#heading=h.2ye70wns7io3>.
- [7] Sgss8. Standard Size of Tactile Paving Bricks, Dimensions of Tactile Paving Bricks [EB/OL]. Sad Sayings Bar, 2023, <https://www.sgss8.net/tpdq/7454697/1.htm>.
- [8] Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1988, 1(2): 119-130.
- [9] LeCun Y. Generalization and network design strategies. *Connectionism in perspective*, 1989, 19(143-155): 18.
- [10] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989, 1(4): 541-551.
- [11] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, 86(11): 2278-2324.
- [12] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012, 25.
- [13] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. *Advances in neural information processing systems*, 2017, 30.
- [14] Liu L, Liu X, Gao J, et al. Understanding the difficulty of training transformers. *arXiv preprint arXiv:2004.08249*, 2020.