

Obstacle Detection with Alert System for the Visually Impaired Using Deep Learning and Node-Red

Khattabi Idriss, Boufarhi Ayman

*Department of Computer Engineering, Tanger, Morocco
{Idriss.khattabi, ayman.boufarhi@etu.uae.ac.ma}*

Keywords: Deep Learning, Transfer Learning, Node-Red, MobileNetv2, Object Detection, Alert System, Visually Impaired

Abstract

This paper presents an innovative solution for obstacle detection and classification aimed at enhancing the mobility and independence of visually impaired individuals. The proposed approach focuses on two methodologies: building a Convolutional Neural Network (CNN) from scratch and fine-tuning a pre-trained MobileNetv2 model using a custom dataset composed of curated and scraped images. Additionally, an alert system powered by Node-RED has been integrated, enabling real-time obstacle identification through an API. Upon prediction, an automatic auditory alert system provides immediate feedback to users, enhancing usability. Our results demonstrate the efficacy of the fine-tuned MobileNetv2 model, which outperforms the custom-built CNN in terms of accuracy and generalization. This work underscores the potential of deep learning models and real-time systems in improving accessibility and quality of life for visually impaired individuals

1 Introduction

Navigating environments with obstacles is a significant challenge for visually impaired individuals. Traditional methods, such as canes or guide dogs, offer limited independence. Advances in deep learning now enable the creation of intelligent systems that can detect and classify obstacles in real-time, enhancing accessibility.

This research focuses on a dual-approach system for obstacle classification:

- **Custom-built CNN:** A CNN trained from scratch for classifying obstacles using a tailored dataset.
- **Transfer Learning with MobileNetv2:** Fine-tuning MobileNetv2 for higher accuracy and generalization.

To enhance usability, an alert system powered by Node-RED integrates real-time predictions with auditory feedback. The system uses an API to process images, predict obstacles, and trigger automatic sound alerts to guide users. This streamlined pipeline offers a practical solution for visually impaired individuals navigating complex environments.

2. Methodology

2.1 Dataset Preparation

A custom dataset was created by merging several datasets and scraping images of common obstacles. It contains 10 classes: chair, door, fence, garbage bin, obstacle (e.g., pole, traffic cone, fire hydrant), plant, pothole, stairs, table, and vehicle.

Each image was manually labeled to ensure quality and consistency.

The class distribution for the training and testing datasets is visualized in the following histograms:

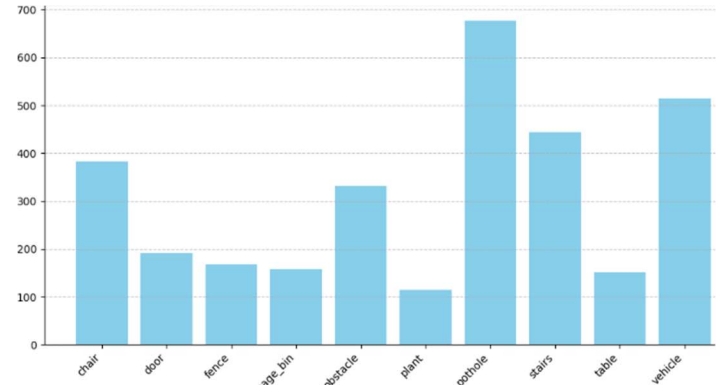


Fig1: Class Distribution in Training Dataset

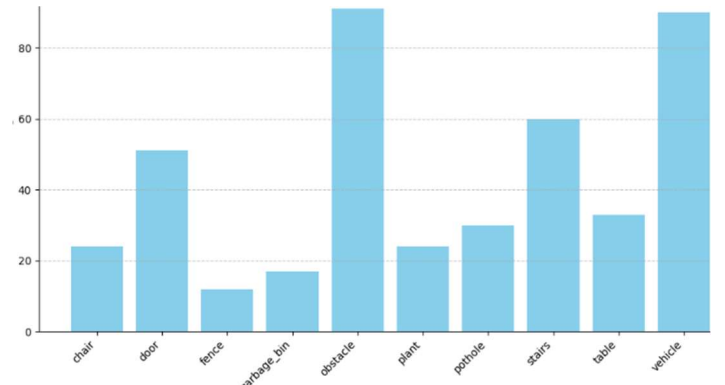


Fig2: Class Distribution in Testing Dataset

To enhance model generalization, a data augmentation pipeline was employed. This pipeline included techniques

such as random resized cropping, horizontal flipping, rotation, and color jittering. These augmentations increased dataset diversity and helped reduce overfitting. All images were normalized using standard ImageNet statistics to ensure compatibility with the model during training.

2.2 Model Architecture (CNN)

Two methods were utilized and compared for the classification approach:

- **Custom-built CNN:** A CNN model was trained from scratch using the custom dataset. The architecture was designed to balance performance and computational efficiency:

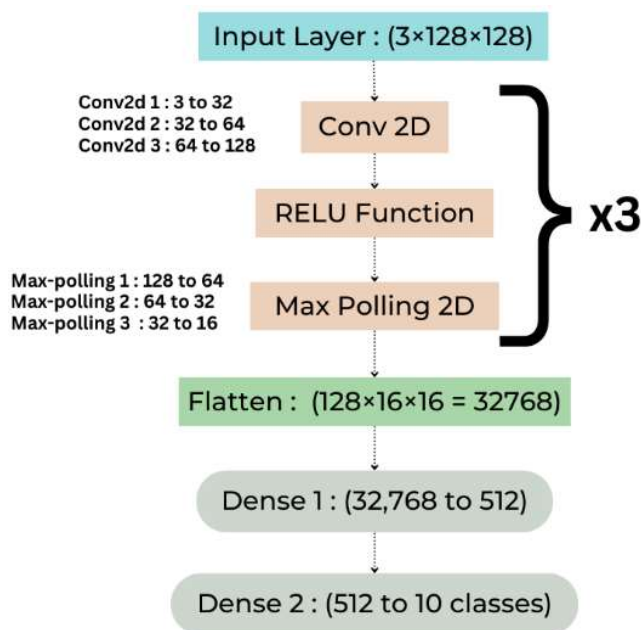


Fig3: CNN Architecture

- **MobileNetV2** is a lightweight convolutional neural network architecture designed for mobile and embedded vision applications. Developed by Google researchers [1], it improves upon the original MobileNet by introducing an inverted residual structure with linear bottlenecks. This design allows the network to be more efficient in terms of computational complexity and model size while maintaining good accuracy.

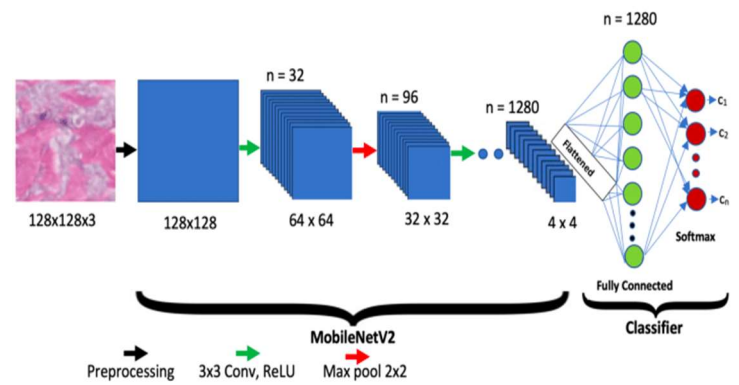


Fig4 : MobileNetV2 Architecture [5]

2.3 Alert System Architecture (Node-Red) [6]

Node-RED is an open-source, flow-based development tool for visual programming. It allows users to wire together devices, APIs, and online services by creating flows using a browser-based interface. With its drag-and-drop functionality, Node-RED enables rapid development and deployment of applications, especially for Internet of Things (IoT) projects, automation, and data integration tasks.

To integrate the trained model into a user-friendly system, an alert mechanism was developed using Node-RED. The alert system provides real-time auditory feedback to visually impaired users by performing the following steps:

- 1- **API Integration:** The trained classification model is deployed as an API endpoint. Images are sent to this endpoint for obstacle prediction.
- 2- **Alert System:** The alert system uses both image and video inputs for real-time obstacle detection. Upon detecting an obstacle, it triggers a response that includes:
 - **Command Line Feedback:** Display the detected class and obstacle type directly in the terminal or console where the application is running.
 - **Audio Alerts:** A corresponding audio warning is played, providing an additional layer of notification.
- 3- **Image and Video Prediction:** The prediction system processes both image and video inputs to detect obstacles:

2.1 Image Prediction:

- **Loading the Image:** The image is retrieved from the file system using a specified path.
- **Image Conversion:** The image is transformed into a format suitable for the prediction model (e.g., base64 encoding).
- **Prediction Request:** The processed image is sent to the backend model via an HTTP request. The model predicts the class of the object, such as a potential obstacle.
- **Response Handling:** If an obstacle is detected with high confidence (probability > 0.8), an alert is triggered, showing the predicted class to the user. If no obstacle is

found or the confidence is low, the user is notified accordingly.

2.2. Video Prediction:

- Video Input: A video file is provided for frame-by-frame processing.
- Frame Extraction: Using FFmpeg, each frame is extracted (in JPEG format) and sent for obstacle detection.
- Frame Processing: Each valid frame is converted and sent to the backend model for prediction.
- Prediction and Alert: Similar to image prediction, obstacles in frames are detected, and alerts are triggered based on the predicted class.

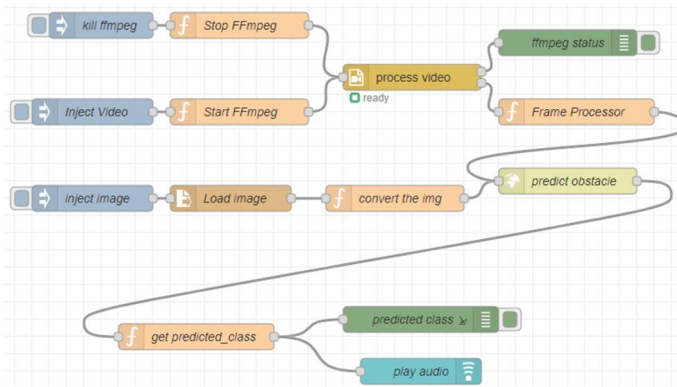


Fig5: Node-Red Architecture

4- System Components:

- Input Device: An image or video frame is sent to the API for classification.
- Processing Unit: Node-RED coordinates the interaction between the prediction API and the alert system, ensuring real-time responsiveness.
- Output Device: Speakers or headphones deliver the auditory feedback, ensuring effective communication of obstacle information to the user.

The system integrates image and video prediction models with an alert system to detect obstacles in real-time. By processing image and video inputs, the system detects obstacles, triggers appropriate alerts, and provides feedback to the user through visual and audio notifications.

3. Obstacle Classification Results

The classification models were evaluated using key metrics such as accuracy, loss, and confusion matrices to assess their performance on the custom dataset.

- **CNN Model from Scratch:**
The custom-built CNN achieved moderate accuracy, demonstrating the feasibility of training models from scratch when sufficient and diverse training data is available. However, its performance was limited compared to pre-trained models, especially for complex obstacle categories.
- **Transfer Learning with MobileNetv2:**
The fine-tuned MobileNetv2 significantly outperformed the CNN model. By leveraging pre-trained weights, it

demonstrated strong generalization capabilities, particularly for intricate obstacle categories, making it more suitable for real-world applications.

3.1 Classification Metrics Overview

- **Training Loss:**
Both models exhibited a gradual decrease in training loss, indicating consistent learning.
- **Validation Loss:**
While the CNN model showed minor fluctuations, the MobileNetv2 model maintained more stable validation loss, reflecting better generalization and resistance to overfitting.

3.2 Model Performance Summary

The classification metrics for both models are summarized below:

- **CNN Model:**
 - Accuracy: 0.8171
 - Loss: 0.7721

	precision	recall	f1-score	support
chair	0.60	0.75	0.67	24
door	0.89	0.76	0.82	51
fence	0.50	0.83	0.62	12
garbage_bin	0.60	0.88	0.71	17
obstacle	0.92	0.84	0.87	91
plant	0.92	0.96	0.94	24
pothole	0.76	0.93	0.84	30
stairs	0.79	0.82	0.80	60
table	0.83	0.88	0.85	33
vehicle	0.93	0.73	0.82	90
accuracy			0.82	432
macro avg	0.77	0.84	0.80	432
weighted avg	0.84	0.82	0.82	432

Actual \ Predicted	chair	door	fence	garbage_bin	obstacle	plant	pothole	stairs	table	vehicle
chair	18	0	0	0	0	1	1	2	0	2
door	0	39	0	6	2	0	1	1	1	1
fence	1	1	10	0	0	0	0	0	0	0
garbage_bin	1	0	0	15	0	0	0	0	1	0
obstacle	4	1	0	1	76	0	3	3	2	1
plant	0	0	0	0	0	23	0	1	0	0
pothole	0	0	0	0	1	0	28	1	0	0
stairs	2	3	4	0	0	0	1	49	0	1
table	1	0	0	0	1	0	1	1	29	0
vehicle	3	0	6	3	3	1	2	4	2	66

Fig6: classification report and Confusion Matrix for CNN

- **MobileNet2 (Fine-tuning):**
 - Accuracy: 0.919
 - Loss: 0.2867

		precision	recall	f1-score	support
	chair	1.00	0.71	0.83	24
	door	0.87	0.92	0.90	51
	fence	0.67	0.83	0.74	12
	garbage_bin	0.93	0.82	0.87	17
	obstacle	0.97	0.97	0.97	91
	plant	0.96	1.00	0.98	24
	pothole	0.87	0.87	0.87	30
	stairs	0.90	0.93	0.92	60
	table	0.79	0.91	0.85	33
	vehicle	1.00	0.94	0.97	90
	accuracy			0.92	432
	macro avg	0.90	0.89	0.89	432
	weighted avg	0.92	0.92	0.92	432

Actual \ Predicted	chair	door	fence	garbage_bin	obstacle	plant	pothole	stairs	table	vehicle
chair	17	0	0	0	0	0	0	1	6	0
door	0	47	0	0	1	0	1	1	1	0
fence	0	2	10	0	0	0	0	0	0	0
garbage_bin	0	3	0	14	0	0	0	0	0	0
obstacle	0	0	1	0	88	0	2	0	0	0
plant	0	0	0	0	0	24	0	0	0	0
pothole	0	0	1	0	1	0	26	2	0	0
stairs	0	1	1	0	0	1	1	56	0	0
table	0	1	1	1	0	0	0	0	30	0
vehicle	0	0	1	0	1	0	0	2	1	85

Fig7: classification report and Confusion Matrix for MobileNetV2

3.3 Model Testing

There are some results of Model Testing :



Fig 8 : classification Testing Results

These results underscore the effectiveness of transfer learning with MobileNet2, which consistently outperformed the custom-built CNN in all key metrics, making it the preferred approach for obstacle classification in this system.

4. Conclusion

In this research, we developed an intelligent system for obstacle classification and alerting, designed to assist visually impaired individuals in navigating their environments. The system focused on two classification approaches: a CNN model trained from scratch and a MobileNetV2-based model leveraging Transfer Learning. Additionally, an alert system using Node-Red was integrated to enhance practical usability by providing real-time audio feedback for detected obstacles.

The MobileNetV2 model significantly outperformed the CNN model, achieving higher accuracy (91.9%) and demonstrating superior generalization across obstacle categories. The integration of the alert system further improved the system's functionality by offering immediate auditory cues, enhancing the user's awareness of nearby obstacles.

Despite these promising results, challenges remain. The limited diversity of the dataset impacted the generalization of both models in specific obstacle classes, and additional improvements in data augmentation and collection are necessary. Furthermore, expanding the model's deployment to include low-power embedded devices will increase its accessibility and scalability.

This work demonstrates the potential of deep learning and Internet of Things (IoT) technologies to bridge the gap between cutting-edge advancements and societal needs, offering a practical solution to improve the independence and quality of life of visually impaired individuals. Future research will focus on refining the system's efficiency, expanding the dataset, and further integrating IoT capabilities to create a robust and scalable navigation aid.

5. References

- [1] Mark Sandler Andrew Howard Menglong Zhu Andrey Zhmoginov Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Google Inc
- [2] Yahia Said, Mohamed Atri, Marwan Ali Albahar, Ahmed Ben Atitallah and Yazan Ahmad Alsariera: Obstacle Detection System for Navigation Assistance of Visually Impaired People Based on Deep Learning Techniques
- [3] O'Shea, Keiron, and Ryan Nash. *An Introduction to Convolutional Neural Networks*. arXiv preprint arXiv:1511.08458, 2015.
- [4] Simonyan, Karen, and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556, 2014.
- [5] What is MobileNetV2?, Analytics Vidhya <https://www.analyticsvidhya.com/blog/2023/12/what-is-mobilenetv2>
- [6] Node-RED Documentation. <https://nodered.org/docs/>
- [7] Albawi, Saad, et al. Understanding of a Convolutional Neural Network. Proceedings of 2017 International Conference on Engineering and Technology (ICET), IEEE, 2017.
- [8] Deng, Jia, et al. ImageNet: A Large-Scale Hierarchical Image Database. Proceedings of the 2009 IEEE Conference

on Computer Vision and Pattern Recognition (CVPR), IEEE, 2009.

[9] Shorten, Connor, and Taghi M. Khoshgoftaar. A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 2019.