School of Computer Science and Information Technology

Department of Computer Science and Information Technology

Computer Vision: Develop a Parking Slots Identification and Tracking Model

Date of Submission: 25/05/25

Submitted by

Name: Haritha N

Reg No./USN No: 23BCAR0321

INDEX

1.Abstract

Parking in crowded cities has become increasingly frustrating due to the unavailability of real-time parking slot information. This project presents a deep learning-based image classification system that identifies and classifies parking slots as "Occupied" or "Free" in a single static parking lot image. The system was built entirely from scratch using a Convolutional Neural Network (CNN), without relying on pretrained models or video streams. After exploring multiple approaches such as automated contour detection and slot extraction via Google image datasets, we settled on a reliable method involving manually annotated slot positions and a CNN trained on full-frame labelled parking lot images. Our system achieved high accuracy and successfully marked over 200 slots, making it a promising solution for scenarios where consistent parking layout images are available.

2. Problem Statement

Project Overview: Smarter Parking with Simple Images

Parking in busy urban areas can be a headache—both for drivers searching for spots and for managers trying to optimize space. While many high-tech solutions rely on live video feeds, we're tackling the problem differently: What if we could determine parking availability just from a single snapshot of a parking lot?

The Goal

We're developing a lightweight, efficient system that can:

- Identify parking slots in a static image (no video required!).

- Classify each spot as free or occupied—quickly and accurately.

- Run on limited resources, making it accessible for smaller lots or budget-conscious setups.

Why It Matters

Not every parking lot needs (or can afford) complex camera systems. By using simple images, we can still provide **real-time insights** without heavy infrastructure. Imagine:

- A shopping mall monitoring spots with occasional snapshots.

- A small business optimizing its private lot.

- Even city planners analysing parking trends from periodic photos.

This approach keeps things simple, cost-effective, and scalable, helping more places manage parking smarter.

3. Methodology

The project was carried out in stages:

Initial Attempts:

- Automatic Contour Detection: Used thresholding and edge detection to identify parking slots. Failed due to misaligned crops and poor accuracy.
- Google Image Dataset: Tried using images from Google with auto-cropping, but classification quality was low.

Final Approach:

- Manual Slot Annotation: Defined precise coordinates for each parking slot in the image.
- Data Collection: Labelled full-frame images manually as "Free" or "Occupied".
- CNN Training: Trained a deep learning model using data augmentation to generalize better.
- Classification: Applied the trained model to cropped slots based on manual annotations.

This approach provided the best balance of accuracy and control.

4. Tools and Technologies Used

- **Python**: Core programming language.
- **OpenCV**: For image processing, thresholding, and visualization.
- **TensorFlow / Keras**: For building and training the CNN.
- **NumPy**: For numerical operations.
- **Scipy**: for scientific and technical computing
- **Matplotlib**: For visualizing training metrics.

5. Dataset Preparation & Preprocessing

- Organized a custom dataset into two classes:
    - Free/
    - Occupied/
- Used full-frame parking lot images labelled manually.
- Applied data augmentation techniques (rotation, flips, brightness adjustment) to improve generalization.

6. Model Architecture

- **Input**: 64x64 RGB images
- **Layers**:
    - Conv2D → ReLU → MaxPooling
    - Conv2D → ReLU → MaxPooling
    - Flatten → Dense → Dropout → Dense (Sigmoid)
- **Loss Function**: Binary Crossentropy
- **Optimizer**: Adam

7. Image Preprocessing

- Converted input image to grayscale

FIG 7.1(GREYSCALE)

- Applied Gaussian Blur and adaptive thresholding

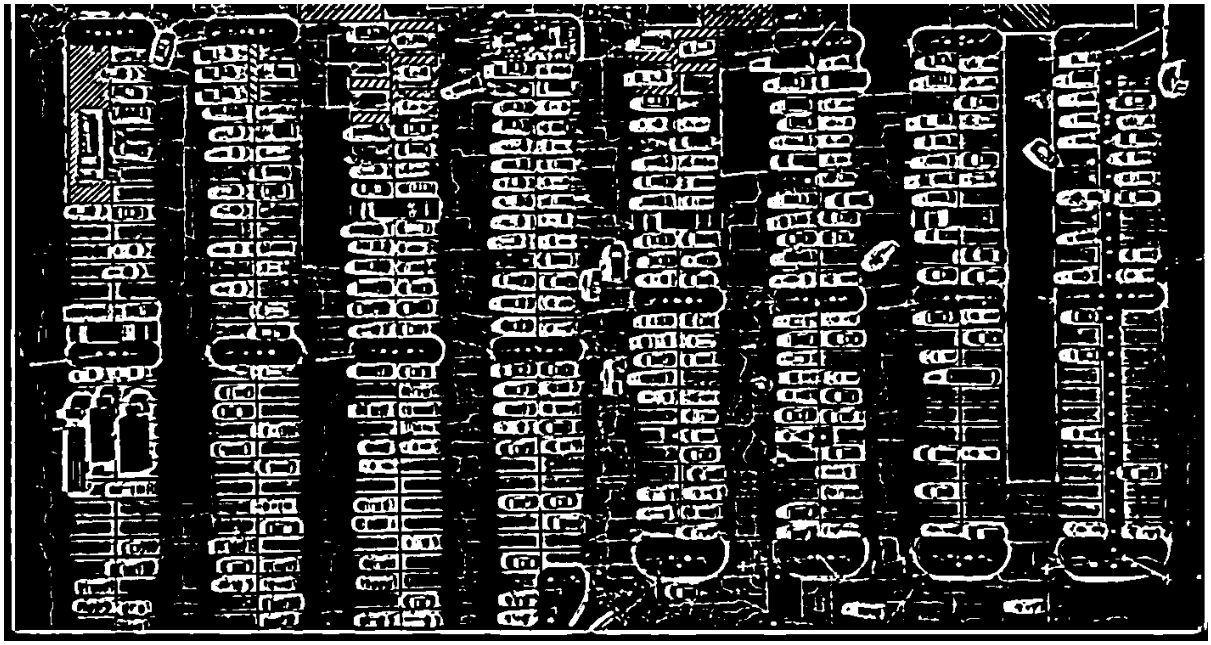

FIG 7.2(THRESHOLD)

FIG 7.3(GAUSSIAN BLUR)

- Median Blur and Dilation were used in earlier experiments

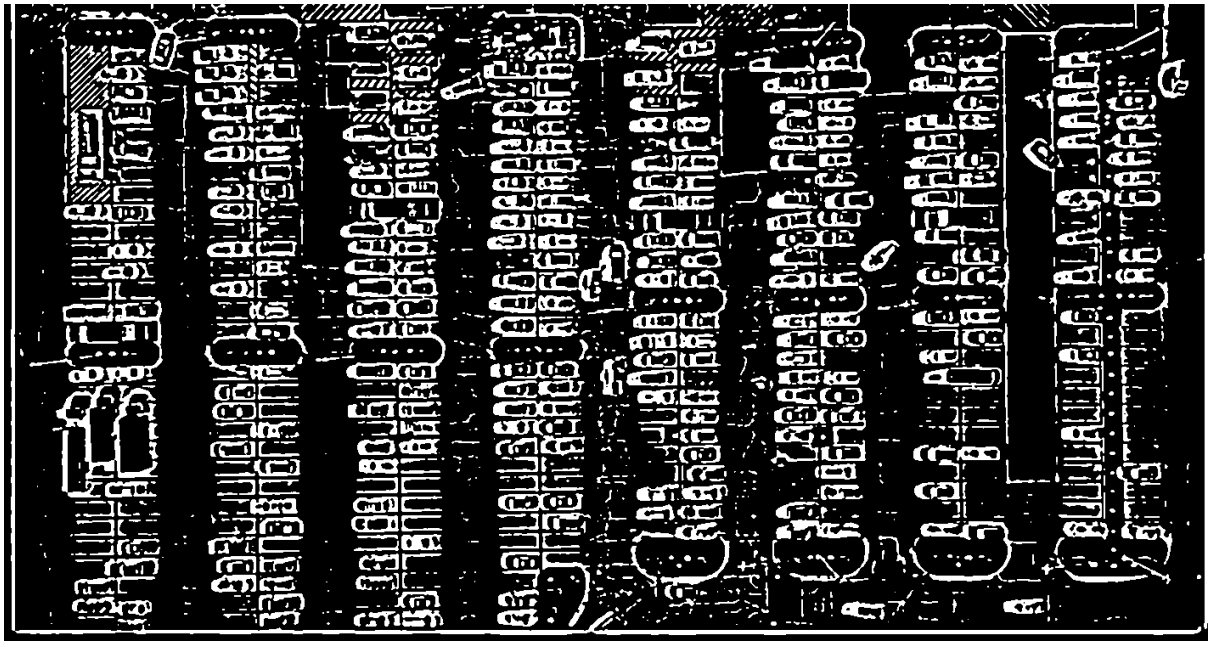

FIG 7.4(MEDIAN BLUR)

- In final implementation, preprocessing was minimal due to manual slot cropping

8. Manual Slot Mapping

- Annotated a list of bounding box coordinates: (x, y, width, height)
- Each bounding box region was classified using the trained CNN
- Color-coded bounding boxes were drawn for each slot:
    - Green: Free
    - Red: Occupied

9. Results and Output

- Achieved ~97% accuracy on validation set

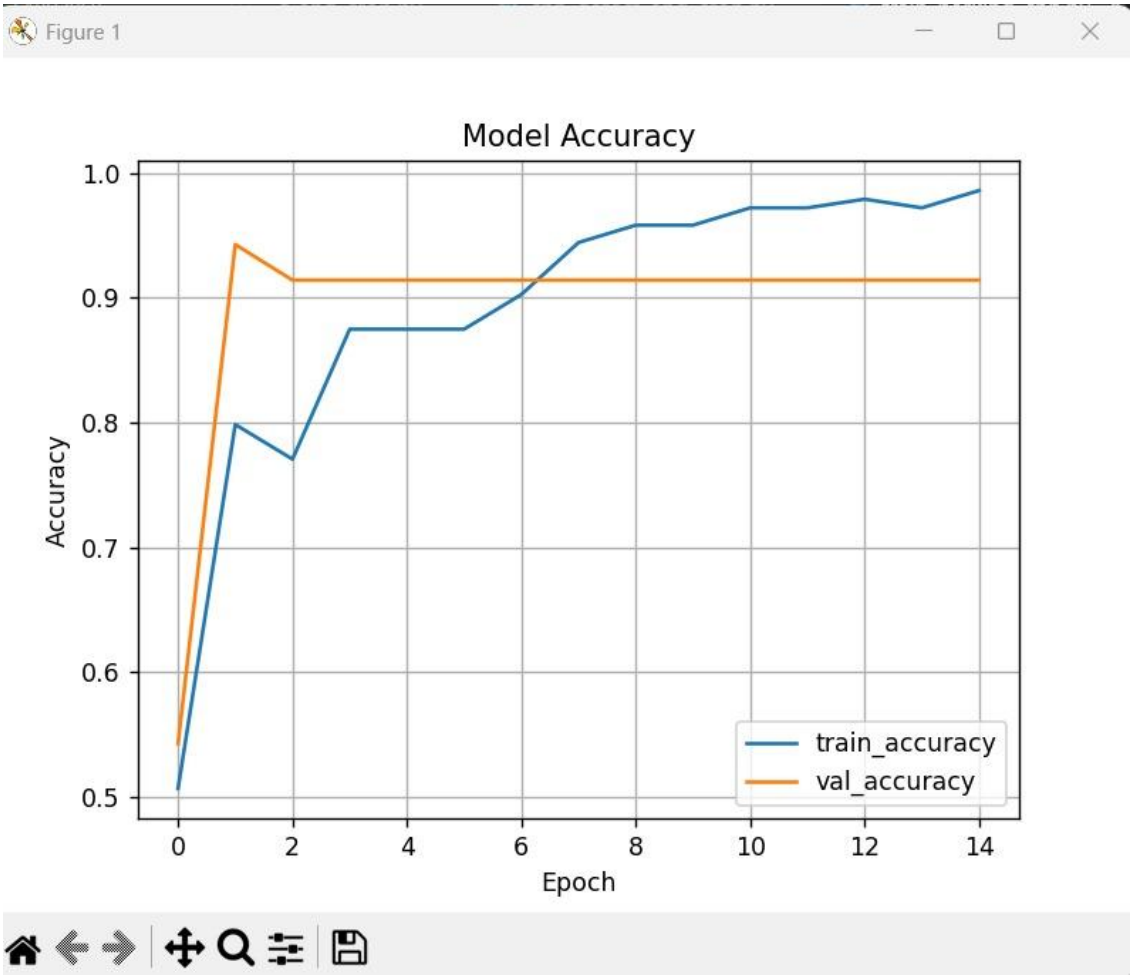

Fig.9.1

- The final output image correctly labelled over 200 parking slots

FIG 9.2.1(parking_image.png[input image])

Fig.9. 2.2(output_detected_simple.jpg[output image])

- Output image saved as output_detected_simple.jpg
- Displayed counts of Free vs Occupied slots on image

10. Conclusion

This project successfully demonstrates the implementation of a CNN-based static parking slot detection and classification system, tailored for fixed-layout parking lot images. Through multiple trials, challenges, and revisions, we explored different strategies — from automated contour-based detection to fully manual slot mapping — ultimately arriving at a reliable hybrid method combining manual bounding box definitions and a CNN classifier trained on full-frame labeled parking images.

Despite early setbacks using automatic detection methods (e.g., thresholding, contour filtering), the project evolved into a more dependable solution by shifting focus toward a robust dataset, data augmentation, and slot-level classification. The final model achieved high validation accuracy (~97%) and performed well on real-world test images, accurately classifying over 200 individual slots as Occupied or Free.

This solution is especially useful in environments where camera positions and layouts are fixed, such as parking lot surveillance systems, smart parking apps, or automated parking management systems.