Group 8 - AI-Driven Automated Parking Guidance System For Motorbike

Đào Mạnh Dũng[1], Nguyễn Minh Phú[2],
Trương Thiên Phú[3], Tăng Hoàng Phúc[4], Mai Xuân Tuấn[5]
23520325[1],23521184[2],2351190[3],23521219[4],23521714[5]

1 Introduction

Currently, the large number of university students and the frequent movement in and out of the parking lot have caused difficulties in coordinating available spaces for security personnel and students to find parking spots. Therefore, we have proposed a solution to this issue: *AI-Driven Automated Parking Guidance System for*
Motorbike Lots

This method has its advantages and disadvantages. With the support of machine learning technology, universities can minimize labor, reduce traffic congestion caused by confusion, and alleviate overcrowding during peak hours. Students can also save time searching for parking spots in the parking lot. Our top priority is to design a system that is convenient and simple and uses algorithms that can be applied to most parking lots within the National University Group for students.

2 AI-Powered Parking Guidance for Motorbike

2.1 Problem Statement

2.1.1 Constraint

- The parking spaces in the parking lot must have fixed dimensions, with a length of at least 1.6 meters and a width of at least 0.8 meters.
- Pathway: The width must be at least 1.6 meters, and there must be a way to access every parking space. At any intersection, the maximum number of turns allowed is four.

- If there are two rows of parking spaces on either side of the pathway, the parking spaces on one side must be arranged parallel to the spaces on the opposite side to avoid any misalignment.
- The entry and exit gates must be separate. At the entry gate, a camera will be positioned in such a way that it can capture the vehicle license plates (installed above along the entryway, facing directly towards the plates to ensure full visibility, no obstructions, and adequate lighting). The system will guide up to five vehicles at a time (each vehicle will follow a designated path marked by a unique color assigned to it).
- Only motorbikes with license plates will be guided.
- Camera Requirements:
  - The cameras must meet a minimum of 24 FPS and 1080p resolution.
  - Ensure the camera images are well-lit; if any area is dark, additional lighting should be installed.
  - For parking spaces, place cameras directly opposite the parking spaces, with a fixed view aimed at the parking spots. The camera height must be at least 7 meters, and the angle between the camera's line of sight and the vertical axis should be no greater than 40 degrees. Each camera should cover multiple spaces that are visible. If a parking spot is located in a position where other vehicles or objects block the camera's view, that camera should not be responsible for monitoring those spaces.
  - Ensure the number of cameras is sufficient to cover all parking spaces assigned to each camera.
  - For pathways, place cameras with a view aligned along the path, at a height of 7 meters or higher, with the angle between the camera's line of sight and the vertical axis being between 45 and 70 degrees. Each section of the pathway should have two cameras facing each other (to ensure both directions of traffic are visible, including license plates). Each pair of cameras will cover a maximum of 20 meters of the path. Ensure that the cameras are positioned so that the entire length of the path is covered without being obstructed by any objects.
- At each pathway, two electronic signs will be installed: one at the beginning and one at the end of the path, so that vehicles traveling in both directions can see the signs. Additionally, each parking space will be equipped with a signal light (with five colors, each color used to guide one specific vehicle) to help vehicles easily identify their assigned parking spot.

2.1.2 Input:

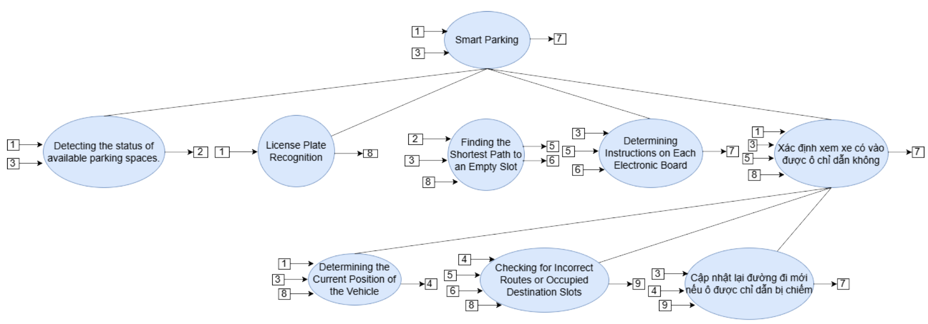
- The parking lot diagram includes the locations of the signage, camera positions, pathways, parking spaces, entry and exit gates, and all the size specifications for the various areas of the parking lot.

- The real-time video feed from all the cameras.

2.1.3 Output:

- Display the guidance on each electronic sign and the signal lights at the assigned parking spots.

2.2 Decomposition



Hình 1: The hierarchical diagram based on the proposed idea

2.2.1 Annotation

1. Real-time video feed from the cameras
2. Update the status of parking spaces
3. Parking lot diagram
4. Current location of the vehicle
5. Nearest available parking space
6. Route to the assigned parking space
7. Update the guidance on each electronic sign and the signal lights at the assigned parking spot
8. License plate of the vehicle entering the gate
9. Boolean value indicating whether the assigned parking space is occupied

2.2.2 Description of the Decomposition Tree

Smart Parking

- This is the overall system responsible for managing and operating the smart parking lot. It receives input data from sensors and cameras, then processes it to provide parking guidance.

Detecting the status of available parking spaces.

- Analyze images to determine the status of each parking space (vacant or occupied).

License Plate Recognition

- Use real-time video to recognize the license plate of incoming vehicles.

Finding the Shortest Path to an Empty Slot

- Utilize the license plate of the incoming vehicle, the parking lot layout, and the vehicle's current position to compute the optimal route.
- Output includes the destination slot and the path to it.

Determining Instructions on Each Electronic Board

- Receive information about the destination slot and the route.
- Update instructions on each electronic board and the signal light at the destination slot.

Checking If the Vehicle Can Reach the Assigned Slot

- Verify if the vehicle is following the assigned route or if the destination slot is occupied by another vehicle.
- If necessary, recalculate the route and assign a new slot.

Determining the Current Position of the Vehicle

- Input includes the parking lot layout and real-time video.
- The system determines the vehicle's current position.

Checking for Incorrect Routes or Occupied Destination Slots

- Use the vehicle's license plate information and parking slot status to check for occupancy.
- Use the vehicle's current position and assigned route to determine if it is following the instructions.

Updating the New Route

- The system recalculates the path to an alternative empty slot.

2.3 Evaluation

To evaluate the efficiency of the intelligent motorcycle parking system, we compare two cases:

- Guided parking with automated instructions.
- Unguided parking (drivers search for slots themselves).

Time measurement starts from card scanning (t_{start}) until the vehicle reaches a parking slot (t_{park}).

2.3.1 Calculation Formulas

$$T_{new} = \frac{1}{n} \sum_{i=1}^{n} \left(t_{park_i} - t_{start_i} \right) \ (s)$$

$$\Delta H = \frac{T_{old} - T_{new}}{T_{old}} \cdot 100 \, (\%)$$

T_{new} and T_{old} are calculated similarly, but T_{new} applies automated guidance, while T_{old} does not.

Symbols:

- n: Number of vehicles entering the parking lot.
- t_{park_i}: Time when vehicle i parks in a slot.
- t_{start_i}: Time from card scanning at moment i.
- T_{new}: Average parking time with automated guidance.
- T_{old}: Average parking time without automated guidance.
- t_{screen_i}: Display time of instructions on the first screen for different vehicles.

2.3.2 Objective:

$$\Delta H > 20 \ (\%)$$

- Demonstrates improved efficiency with the new system.

2.3.3 Detailed Evaluation

1. **System Latency**
   - The formula considers only the first screen, as all screens have the same response time.

$$T_{delay} = \sum_{i=1}^{n} (t_{screeni} - t_{starti}) \ (s)$$

Symbols:

- T_{delay}: Total system delay time.
- $t_{screeni}$: Display time of instructions on the first screen for each vehicle.
- t_{starti}: Time from card scanning at moment i.

Objective:

$$T_{delay} < 5\,(s)$$

- The system must ensure a delay of less than 5 seconds for user experience.

2. **Video Processing Speed**
   - Real-time processing with an error margin:

$$\Delta t < 0.01\,(s)$$

2.3.4 Conclusion

The intelligent parking system is evaluated based on:

- Average parking time (T_{new}) with automated guidance.
- Efficiency improvement rate ΔH exceeding 20
- System latency below 5 seconds.

2.4 Algorithms

- Step 1:
Draw a bounding box for each parking slot in the image frame of the camera responsible for it.
Assign an ID to each parking slot.

- Step 2:
For each road:
If the road is adjacent to a row of parking slots:
Divide the road into smaller segments so that each segment is adjacent to two parking slots on both sides (if the road has parking slots on both sides; otherwise, each segment is adjacent to only one parking slot).
Each segment is responsible for the parking slots adjacent to it.
If the road is at an intersection:
The section at the junction is assigned as a single segment.
If the road is a straight path with no intersections and no adjacent parking slots:
This segment is assigned as a single section.
If the road is an entrance segment:
This segment is assigned as a single section.
Draw bounding boxes for each segmented road section within the image frame of the responsible camera.
Assign an ID to each segmented road section.
Create a dictionary `responsible_road_section`, where the key is the ID of the segmented road section and the value is a list of parking slot IDs that the

section is responsible for.

- **Step 3:**
Set the entrance section as the coordinate origin (0,0).
Define the direction from the entrance to the parking lot as upward.
Define `assign_coordinates(current, continue)`: (where `current` is the ID of the current road section and `continue` is the ID of the adjacent section)

    Assign coordinates to `current` as (a, b).
    If the direction from `current` to `continue` is upward:
        Assign coordinates to `continue` as (a, b+1).
    If the direction is turning left:
        Assign coordinates to `continue` as (a+1, b).
    If the direction is turning right:
        Assign coordinates to `continue` as (a-1, b).
    If the direction is downward:
        Assign coordinates to `continue` as (a, b-1).
    For each adjacent section `continue_of_continue` (excluding `current`):
        Call `assign_coordinates(continue, continue_of_continue)`.

Assuming the entrance section has ID 1 and its adjacent section has ID 2: `assign_coordinates(1, 2)`.

- **Step 4:**
Assign IDs to electronic boards.
Assign IDs to each camera responsible for monitoring roads.
Create a dictionary `electronic_board_positions`, where the key is the electronic board ID, and the value is a list containing:
- The coordinate ID of the road section in front of the electronic board.
- The coordinate ID of the road section where the electronic board is located.
- A list of camera IDs responsible for monitoring the roads facing the board.

- **Step 5:**
Consider the segmented road sections as nodes.
For each pair of adjacent road sections, create a bidirectional path between their corresponding nodes.

- **Step 6:**
If a vehicle scans its card to enter:

    Use Yolo and EasyOCR to recognize the license plate.
    Update the license plate in the database.
    Update the vehicle's current position as the entrance node in the database.
    Update the ID of the camera that detected the license plate in the database.

- **Step 7:**
Use all real-time video frames from all cameras:

    Apply thresholding to create binary images from cameras monitoring parking slots.

For each parking slot in the list:
If the number of white pixels < Threshold, the slot is "Empty"; otherwise, it is "Occupied".
Update the status of parking slots by ID.
If all slots are "Occupied", deny entry to the parking lot.
For each key, value in `responsible_road_section.items()`:
If at least one parking slot in `value` is "Empty":
Mark the road section as a target node.
Else:
Mark the road section as a normal node.
Use BFS to find the nearest target node from the current node.
Select an "Empty"slot within `responsible_road_section[target node]` and mark it as "Wait".
Update the path and destination slot ID in the database for the vehicle's license plate.
Let `path` be the list of coordinates along the route.
Let `s` be the ID of the camera detecting the license plate.
`v = path[1] - path[0]`.
For each key, value in `electronic_board_positions.items()`:
If `s` is in `value[2]`:
`u = value[0] - value[1]`.
If `u` and `v` are in the same direction:
Display a U-turn instruction on electronic board `key`.
For `i` in range `len(1, path - 1)`:
For each key, value in `electronic_board_positions.items()`:
If `value[1] == path[i]` and `value[0] == path[i-1]`:
Compute directional change and display turn instructions.
Display signal at the destination slot.

- Step 8:
For each frame in real-time video:
For each license plate in the database:
Use Yolo and EasyOCR to find the camera ID and road section ID of the detected plate.
If the plate is no longer visible in any camera:
Remove it from the database.
Else if the road section is not in the planned path or the destination slot is "Occupied":
Update the vehicle's current position and detected camera in the database.
Repeat Step 7.
Else:
Continue.

3 Computational Thinking Application

3.1 Decomposition

The problem is broken down into the following components and tasks:

- **Detecting vacant parking spots:**
  Determine the status of each parking spot using images from cameras.
- **License plate recognition:**
  Identify vehicles by their license plates to associate them with their target parking spots.
- **Finding the shortest path:**
  Use graph algorithms to optimize the vehicle's route.
- **Updating guidance:**
  – Display directions on electronic signs at intersections and activate signal lights at destination parking spots to guide vehicles correctly.
- **Monitoring and updating status:**
  – Track real-time video feeds to update the current position of vehicles.
  – Detect cases where vehicles deviate from the assigned route or when the destination parking spot becomes occupied.

3.2 Abstraction

In this problem, we focus on essential elements and eliminate unnecessary details:

- **Key elements to focus on:**
  – Parking lot layout (locations of parking spots, pathways, guidance signs, and cameras).
  – Real-time video data from cameras monitoring parking spots to determine availability and cameras monitoring pathways to track vehicle positions.
  – License plate recognition for vehicle identification.
  – Displaying guidance instructions on electronic signs at specific locations.
- **Details to disregard:**
  – Vehicle color or design, as they do not impact the solution.
  – Irrelevant information from video feeds, such as other objects in the frame.

3.3 Pattern Recognition

Identifying the patterns of the problem is as follows:

- **Parking Spot Status Recognition:**
  – Apply the thresholding method on images to determine the number of black/white pixels in each parking spot, thereby identifying the spot's status (vacant or occupied).

- **License Plate Recognition:**
  - Detect license plates in images at the entrance gate.
  - Extract characters from the license plate to identify the vehicle.
- **Vehicle Location Identification:**
  - Determine the bounding box of the license plate to locate the vehicle within the parking lanes and identify its current position.
- **Shortest Path Finding:**
  - Use a graph of nodes (corresponding to divided road segments) to determine the optimal path to an available parking spot, with the starting node being the vehicle's current position.
- **Guidance Update:**
  - Display turning commands at intersections via signboards and light signals at the destination spot.

3.4 Algorithms

The main algorithms in the system include:

- **License Plate Recognition Algorithm:**
  - **YOLO (You Only Look Once):** Detect vehicles in the frame.
  - **EasyOCR:** Extract and recognize characters from license plates.
- **Parking Spot Status Determination Algorithm:**
  - Apply the thresholding method to convert images to binary format, determining the spot's status based on the number of white pixels.
- **Shortest Path Finding Algorithm:**
  - **BFS (Breadth-First Search):** Find the fastest route from the entrance to the target spot.
  - Based on a graph of nodes (road segments) connected in the parking lot layout.
- **Guidance Update Algorithm:**
  - Calculate turning directions at each intersection based on the cross product of path vectors.
  - Display left turn, right turn, or straight-ahead commands on electronic boards at each guidance sign.
- **Error Handling Algorithm:**
  - Detect vehicles deviating from the planned route or occupied destination spots.
  - Recalculate a new path and update guidance accordingly.

4 Ethics and Social

The application of modern technology in parking systems has brought significant advancements but also raises important ethical and social issues, which considerably impact the realization of such initiatives. These issues can be divided into two main aspects:

4.1 Positive Impacts

- **Job Creation:** The design, construction, and maintenance of the system generate new job opportunities, meeting labor demands in the technology sector.
- **Encouraging Innovation:** Contributes to raising technological awareness, fostering innovation, and supporting national digital transformation while keeping up with modern trends.
- **Enhancing Operational Efficiency:** Takes over tasks such as vehicle management and security, saving manpower and increasing productivity and operational efficiency.
- **Ensuring Fairness:** The system ensures no discrimination between vehicle types, whether luxury or standard, promoting transparency and fairness.
- **Modernizing Management:** Helps create a more modern working environment, reduces congestion, saves time in the parking process, and improves management efficiency.

4.2 Negative Impacts

- **Job Displacement:** Manual labor is replaced by knowledge-based roles. Many unskilled workers may lose jobs due to the inability to meet new technological requirements.
- **Privacy Risks:** Users may feel monitored as the system collects and stores license plate information. In case of cyberattacks, this sensitive information could be leaked and misused.
- **Inconsistency Issues:** Ensuring consistency between vehicle entry and exit can be challenging, leading to potential errors in management.
- **System Dependence:** Users may become overly dependent on the system, and in case of errors, it could lead to wasted time, financial loss, unnecessary disputes, or even loss of trust in the system.
- **Accessibility Barriers:** Elderly individuals or those less familiar with technology may struggle to adapt to the system's modern interface.
- **Reduced Social Interaction:** Decreased direct communication may cause some individuals to feel isolated or unsupported in situations requiring personal attention.

- **Lack of Empathy:** Unlike human staff, AI-based systems often lack flexibility and empathy, making it difficult to address customers' specific needs.

5 Conclusion

The design and implementation of a smart parking system based on the specified requirements and constraints help optimize space management and usage, enhance guidance and monitoring efficiency, and ensure operational standards. This not only provides a better user experience but also contributes to improving the overall efficiency of parking operations.

Although there are still some limitations in system processing and further solutions are needed for detailed optimization, this approach remains feasible and offers significant benefits when applied in practice.

THE END