VEHICLE MOVEMENT ANALYSIS & INSIGHT GENERATION IN A COLLEGE CAMPUS USING EDGE AI

Problem Statement - 13

JULY 15, 2024

SENTINELS
KIT’s College of Engineering, Kolhapur

Team Member 1

- **Name:** Ankita Tanaji Yadav

- **Specific tasks and responsibilities handled:**

  a. Integration of the SQL database for efficient data management and storage.

  b. Implementation of CRUD operations using Node.js to handle database interactions.

- **Code sections written or major contributions to the project**

1. **Developed the database schema and tables for storing vehicle detection and recognition data:**
   - Created three tables: ‘readings’, ‘tags’, and ‘anpr’, each with specific roles in the system.
     - The ‘readings’ table stores detailed information about each vehicle detection, including various versions of the recognized license plate and their corresponding confidence scores.
     - The ‘tags’ table contains tags associated with each recognized license plate, such as "Staff" if the entry exists in the database, or "Visitor" if it does not.
     - The ‘anpr’ table logs ANPR events, recording license plate details, state information, date, day of the week, month, time, and associated tags.

2. **Wrote the logic for inserting, updating, and retrieving data in the SQL database:**
   - Implemented data insertion logic to manage the storage of new vehicle detection results and update entries based on confidence scores.
   - Maintained data integrity by ensuring that only recognition results with higher confidence scores replace existing entries for the same vehicle.

- Developed validation and commitment processes to ensure data accuracy and reliability.

❖ <u>**Challenges faced and how they were overcome:**</u>

a. **Challenge:** Ensuring real-time performance while managing large volumes of data in the SQL database.
   Solution: Optimized SQL queries and used indexing to speed up data retrieval operations.

b. **Challenge:** Managing data integrity by avoiding duplicate and inconsistent records.
   Solution: Developed a logic to check for existing records belonging to the same vehicle (i.e., records with the same vehicle ID) and replace them only if the new record had higher confidence scores than the existing one.

c. **Challenge:** Managing asynchronous operations and error handling in Node.js.
   Solution: Implemented comprehensive error handling to catch and log errors, ensuring smoother debugging and maintenance.

❖ <u>**Collaboration with other team members:**</u>

- Collaboration with the interface team for integrating frontend and backend functionalities.
- Assistance in annotating license plate images for recognition to improve model accuracy.

Team Member 2

- ❖ **Name:** Bhakti Prakash Ayarekar

- ❖ **Specific tasks and responsibilities handled**

  a. **Front Page Design and Development of Web Interface:** Developed and designed the front pages, focusing on layout, responsive design, interactive elements integration, performance optimization, and user testing to ensure usability and functionality.

  b. **Created a ResNet-50 OCR model:** Developed a tailored Optical Character Recognition model to accurately read and interpret vehicle number plates.

  c. **Implemented the ResNet-50 OCR model:** Successfully integrated the ResNet-50 OCR model, to work along with the rest of the model involved in the License Plate Recognition system.

- ❖ **Code sections written or major contributions to the project**

1. **Development of ResNet-50 OCR model for Efficient Character Recognition:**
    - Mounted Google Drive for dataset access and installed Python packages like Keras and TensorFlow.
    - Defined a CNN model using Keras, set up data generators with real-time augmentation, compiled, and trained the model with the Adam optimizer, saving it to disk.
    - Tested the model on new images, visualizing predictions and original images.
    - Developed an OCR pipeline to preprocess, segment, and recognize characters in license plate images.
    - Visualized the original images and the prediction results.

2. **Designing and Layout of Frontend User Interface:**
    - Created a responsive navigation bar using HTML and CSS to ensure easy access to essential pages within the ANPR project.

- Implemented a modal login form to enhance user interaction and streamline access to secure areas of the application.
- Designed a feedback form utilizing HTML forms and CSS styling to gather user input and improve user engagement.
- Developed a team section that showcases member details, including links to their social profiles, fostering transparency and team visibility.
- Integrated a footer with relevant links and contact information, providing users with easy access to additional resources and support.

❖ Any challenges faced and how they were overcome

1. **Custom OCR:**
   a. **Data Preprocessing:** Ensuring uniformity in image dimensions and quality was crucial. This was achieved by resizing images to a consistent size (64x64 pixels) and applying normalization techniques. Additionally, data augmentation techniques such as random flipping, rotation, and contrast adjustment were used to enhance the robustness and generalization of the model.

   b. **Model Selection:** Choosing an appropriate deep learning architecture required experimentation. In this implementation, the ResNet-50v2 model was selected for its balance between model complexity and performance. The model was initialized with pre-trained weights from ImageNet, and the top layers were replaced to adapt to the OCR task.

   c. **Training Optimization**: Addressing overfitting and underfitting challenges during model training involved several techniques:
      - Freezing the pre-trained layers initially and then unfreezing them for fine-tuning.
      - Adding a global average pooling layer, batch normalization, and dropout regularization (with a rate of 0.25) to enhance model generalization.
      - Using a categorical cross-entropy loss function and the Adam optimizer with a learning rate of 1e-5.
      - Incorporating the WandbCallback for logging and tracking experiments with Weights and Biases (wandb).

   d. **Integration Logic**: Encountered challenges while integrating the ResNet-50 OCR model with the License Plate detection model. Specifically, the issues arose in aligning the outputs of the detection model with the OCR model's input requirements. Additionally, there was a need to implement custom image processing operations to enhance the character recognition accuracy of the OCR model.

2. **User Interface:**
   a. **User Interaction:** Addressed user interaction challenges by refining the modal login form and feedback mechanisms based on usability testing and user feedback.

   b. **Navigation Design:** Managed complexities in navigation design by conducting user surveys and testing to optimize the navigation bar for intuitive access to essential project features.

   c. **Visual Appeal:** Enhanced visual appeal by iterating on the interface design, focusing on color schemes, typography, and UI elements to create a cohesive and aesthetically pleasing user experience.

   d. **Team Section Integration:** Successfully integrated the team section with social profile links, overcoming challenges related to content alignment and responsive display on various devices.

❖ **<u>Collaboration with other team members</u>**

- **Number Plate Annotation:** Collaborated closely with team members to annotate a diverse dataset of number plate images, ensuring accurate labeling of regions of interest (ROIs) containing alphanumeric characters. This involved using tools like CVAT and maintaining consistency in annotation standards.
- **Integration of ResNet-50 OCR:** Worked collaboratively to integrate ResNet-50 OCR for feature extraction within the pipeline. This included performing image processing operations on the number plate images to extract relevant features before passing them through the OCR model, ensuring efficient character recognition.

Team Member 3

- ❖ **Name:** Chinmay Anand Kulkarni

- ❖ **Specific tasks and responsibilities handled**

  a. **Data Generation for Visualization:** Generated a structured dataset tailored for visualization purposes to enhance data analysis and insight generation.

  b. **Development of Web Interface:** Utilized Bootstrap classes for a structured and responsive layout, enhanced CSS for readability and visual appeal, and applied consistent styling to table elements. Integrated Chart.js placeholders for data visualization and leveraged Bootstrap for overall responsive design, ensuring a seamless and user-friendly interface.

- ❖ **Code sections written or major contributions to the project**

1. **Insights Generation:**

- **Data Loading and Preprocessing**: Loading a structured dataset containing information on vehicle entries. The initial preprocessing steps include handling missing values, parsing date columns, and filtering out irrelevant data to ensure the dataset is clean and ready for analysis.

- **Exploratory Data Analysis (EDA)**: A thorough exploratory data analysis is conducted, focusing on key aspects such as:
  - **State-wise Distribution**: Analysis of the number of vehicle entries per state, identifying Maharashtra and Karnataka as the states with the highest entries.
  - **Date-wise Analysis**: Identification of peak entry dates, which may correlate with special events or patterns.
  - **Day-wise and Hour-wise Analysis**: Investigation into the distribution of entries across different days of the week and hours of the day, revealing patterns such as peak entry hours and the busiest days.

- **Category Distribution**: Breakdown of the entries into different categories, showing that visitors constitute the majority of entries, with a small percentage accounted for by staff.
- **Monthly Trends**: Analysis of the number of entries per month, highlighting the months with the highest and lowest entries.
- **Weekday Time Distribution**: Calculate the median entry hour for each weekday to understand daily entry patterns.
- **Heatmap Visualization**: Generation of heatmaps to visualize the geographic distribution of entries, emphasizing regions with consistent daily entries.
- **Growth Rate Calculation**: Computation of the daily average growth rate in license plate entries, providing insights into the trend over time.
- **Insights Generation**: Compilation of insights derived from the analysis, summarizing key findings such as the states with the highest entries, peak entry dates and times, and the overall distribution patterns.

❖ **<u>Designing and Layout of Frontend User Interface:</u>**

1. **Structured Layout:**
   - Utilized Bootstrap classes to create a structured and responsive layout, ensuring the application adapts well to various screen sizes and devices.

2. **Enhanced Styling:**
   - Enhanced the CSS to improve readability and visual appeal, ensuring a cohesive and attractive design throughout the application.

3. **Table Styling:**
   - Designed a table to display number plate data, including columns for Licence, StateCode, State, Date, Day, Month, Time, and Tag.
   - Applied consistent styling with alternating row colors, borders, and padding to enhance readability.

- Used CSS to set a uniform font size and text alignment for the table elements.

4. **Form for Updating Number Plate Information:**
    - Built a form that allows users to update number plate details, including fields for Licence, StateCode, State, Date, Day, Month, Time, and Tag.
    - Styled the form to be visually appealing with appropriate padding, borders, and input field styles.
    - A submit button with hover effects was included to improve the user interface and make interactions more intuitive.

5. **Chart.js Integration:**
    - Added placeholders for Chart.js to visualize data through two canvas elements.
    - Styled the chart containers with borders and padding to distinguish them from other content.
    - Implemented JavaScript to initialize and configure the charts, setting up data and options as needed to ensure accurate and clear data representation.

❖ **<u>Any challenges faced and how they were overcome</u>**

1. **Challenges Faced in Insight Generation**
    a. **Challenge:** Generating artificial/dummy data that closely resembles real-time data.
    Solution: To generate insights, I created dummy data that closely resembles actual real-time data. I added logic to assign "Staff" tags to some entries and "Visitor" tags to the rest. Additionally, as our college has holidays every Monday and specific Sundays, I implemented a function to identify these days as off days, preventing data generation on these dates.

    b. **Challenge:** Discovering relevant patterns and insights within the data.
    Solution: I used various visualization techniques, such as histograms, heatmaps, and scatter plots, to uncover patterns. I also

employed statistical analysis to validate the observed trends and ensure they were not due to random chance.

2. **Challenges Faced while developing User Interface:**
   a. **Table Readability:**
   Challenge: Ensuring the table displaying number plate data is readable and visually appealing.
   Solution: Applied alternating row colors, adequate padding, and borders. Conducted user testing to gather feedback and adjusted the design accordingly.

   b. **Form Usability:**
   Challenge: Designing a user-friendly form for updating number plate information.
   Solution: Included clear labels, adequate spacing, and intuitive input field styles. Added hover effects on the submit button for better user experience.

   c. Chart.js Integration:
   Challenge: Implementing dynamic charts that update with changing data.
   Solution: Due to errors and time constraints, I implemented static charts to demonstrate the benefits of visualization.

❖ **<u>Collaboration with other team members</u>**

- Assistance in annotating license plate images using an online annotation tool called CAVT, ensuring accurate labeling of regions of interest (ROIs).

- Assisted in implementing front-end CRUD functionality, with the backend logic and database queries, ensuring data integrity and security.

Team Member 4

- **<u>Name:</u>** Rohan Sachidanand Chopade

- **<u>Specific tasks and responsibilities handled:</u>**
  a. Implementing the YOLOv8 Models: The custom-trained model was specifically optimized for detecting license plates, while the pre-trained model was used for general vehicle detection.
  b. Integration of SORT Algorithm for tracking vehicles across different frames.
  c. Development of 'main.py'.

- **<u>Code Sections Written or Major Contributions:</u>**
  a. **YOLOv8 model:**
     - Worked on training, testing, and fine-tuning the custom YOLOv8 model for License Plate Detection.
     - Worked on Integrating the YOLOv8 models; one pre-trained model for vehicle type identification and the second a custom-trained model for License plate detection.
     - Integrated the models with the SORT algorithm to track vehicles across frames.

  b. **SORT Algorithm:**
     - Implemented the SORT algorithm for vehicle tracking across different frames, in case the input is a video file.
     - Configured SORT to handle bounding boxes generated by pre-trained YOLOv8 model for vehicle detection.
     - For this I also changed the sequential ID to hash ID in the SORT algorithm, which made the algorithm work efficiently with image and video input
     - To reduce errors in the database and to create robust tracking I changed the type of ID which was a sequential ID (1,2, 3, …) used as a tracking ID in the Original algorithm to a hash ID. Thereby,

making the tracking part work well for both types of inputs, i.e. image as well as video file.

c. **Developed the 'main.py':**
   - Worked on developing the main.py script, integrating all modules, and ensuring seamless workflow.
   - Coordinated the data flow between YOLOv8 models, SORT algorithm, and OCR models.

❖ <u>**Any challenges faced and how they were overcome**</u>

a. **Lack of vehicle-specific identifier:**
   - The initial prototype used a custom-trained YOLOv8 model to detect license plates and then send them to OCR for character recognition. This method worked well for images but encountered issues with videos, as multiple readings for the same vehicle were appended to the database. We couldn't determine which license plate belonged to which vehicle just by looking at the table.
   - To maintain database integrity and reduce redundancy, we needed to keep only the records with the highest confidence scores. We implemented a solution to check if the license plate already existed in the database: if it did, we compared the old and new confidence scores, retaining the highest one; if it didn't, we added the new reading.
   - However, OCR errors complicated this process. For example, if a license plate MH09BM8187 was initially stored with a confidence score of 0.8 and then misread as NH09BM8187 with a score of 0.7, it was treated as a new entry due to the lack of a vehicle-specific identifier, causing inconsistencies.
   - So, I implemented a pre-trained YOLOv8 model which would detect and classify vehicles, along with that I also used the SORT algorithm, for tracking the vehicles across different frames, by assigning a unique sequential ID (1, 2, 3, …).
   - This ID played the role of a vehicle-specific identifier and resolved the issue.

b. **Problem with Using Sequential ID assigned by the SORT algorithm:**

- The problem with using sequential IDs assigned by the SORT algorithm was identified when the database already contained records with the same ID.
- For example, when an image of a vehicle was provided, the pre-trained model would first identify the vehicle type, and the SORT algorithm would assign an ID to it, say 1. After the OCR process, the reading would be stored in the database without any issues.
- However, when the next image of a vehicle was provided, the SORT algorithm would again assign the same sequential ID, 1, to the vehicle.
- This created a problem when inserting the license plate reading into the database because the ID was already present.
- If the associated confidence score was less than that of the existing record, the new record was not inserted; if the confidence score was greater, the old record was overwritten. Consequently, license plates from different vehicle images were compared due to having the same ID.
- To resolve this, I used a hash ID instead of a sequential ID in the SORT algorithm, which eliminated the error.

❖ <u>**Collaboration with other team members**</u>

a. **Image Annotation:**

- Collaborated with team members to accurately annotate a diverse dataset of vehicle images, focusing on the region of interest (ROI), which was the license plate. Utilized online tools like CVAT to ensure precision and maintained consistent annotation standards throughout the process.

b. **Integrating the Database:**

- Assisted in implementing the database by aligning it with the code and using the hash ID assigned by the SORT algorithm for each vehicle. This ensured efficient storage and retrieval of the corresponding license plate data.

Team Member 5

❖ **Name:** Mohammadsaad Hidayat Jamadar

❖ **Specific Tasks and Responsibilities Handled:**

a. Developed and implemented functions for license plate recognition and correction.
b. Integrated OCR tools (ResNet-50 OCR, EasyOCR, and Tesseract) for text extraction from images.
c. Created functions for image preprocessing and enhancement to improve Custom OCR accuracy.
d. Implemented the logic to identify and correct misread characters in license plates.

❖ **Code Sections Written or Major Contributions to the Project:**

- I implemented a function called **correct_license_plate(ocr_result)**. This function ensures that any characters misread by the OCR (Optical Character Recognition) tools are correct. It uses a predefined list of common OCR mistakes and corrects them to make sure the license plate text is accurate.

- Next, there's the **write_csv(results, output_path)** function. This function takes the processed results and saves them into a CSV file. By doing this, we can easily store and analyze the data later in a structured format.

- The **get_car(license_plate, vehicle_track_ids)** function finds the specific area of the car in an image using vehicle tracking IDs. It matches the license plate to the vehicle's tracking ID to locate and return the precise bounding box around the car.

- For image processing, I implemented the **img_op(image)** function. This function preprocesses the image to improve OCR accuracy. It applies techniques like enhancing contrast, sharpening, and blurring to make the license plate text clearer for the OCR tools to read.

- The **predict(image, model)** function uses a trained machine learning model to identify and predict the characters on the license plate. It takes an image as input and returns the predicted alphanumeric characters.

- The **license_plate_reader(image, model, weights)** function is a comprehensive solution for reading license plates. It uses multiple OCR tools, including EasyOCR, Tesseract, and a custom-trained model, to achieve high accuracy in reading the text on the license plates.

- The **get_state_and_state_name(license_plate_easyocr)** function extracts the state code and name from the license plate text recognized by EasyOCR. This provides additional information about the vehicle's origin, which can be useful for various applications.

❖ Any Challenges Faced and How They Were Overcome:

a. **Challenge**: Improving OCR accuracy with low-quality images.
   Solution: Implemented image preprocessing techniques such as contrast enhancement, sharpening, and Gaussian blurring to improve OCR accuracy.

b. **Challenge**: Correcting misread characters in license plates.
   Solution: Developed dictionaries to map commonly misread characters to their correct counterparts and applied these mappings in the correct_license_plate function.

c. **Challenge**: Integrating multiple OCR tools and models for license plate recognition.
   Solution: Combined results from EasyOCR, Tesseract, and a custom-trained model to improve the overall accuracy and robustness of the license plate recognition system.

❖ Collaboration with Other Team Members:

- Worked closely with team members to integrate the license plate recognition system into the larger project.

- Assisted team members in debugging and resolving issues related to OCR and image processing.

- Collaborated on annotating number plate images using CVAT an online annotation tool, ensuring accurate and consistent labeling.