Automated Railway Station Management System Using Open CV

Project Name

AutoVisionary Rail Nexus

Group Name

ByteBuds

Team Members:

Rajat Mahamalla

Siddharth Parihar

Shrishti Mourya

Aarsi Ansari

INDEX

1. Introduction
2. Objective
3. Literature survey
4. Proposed Work
   1. Methodology
   2. Workflow
   3. Hardware Requirement
   4. Software Requirement
   5. Working structure of automated light by using OpenCV
   6. Working structure of automated light by using OpenCV
   7. Working structure of automated light by using OpenCV
   8. Working structure of automated light by using OpenCV

Future Scope

Conclusion

Reference

1. **Introduction**

Railway stations are crucial hubs in the transportation network, serving as gateways for millions of commuters daily. Ensuring a comfortable and efficient environment at these stations is essential for passenger satisfaction. One aspect of achieving this is the effective management of energy resources, such as controlling the operation of fans and lights based on the actual presence of people and also aimed to detect fire by using the OpenCv that will alert people by early detection of fire. This project also aims to utilization of data analysis in storing visitor data at a railway station is a crucial aspect of modern transportation management systems. Railway stations witness a constant influx of passengers, making it essential to efficiently manage and analyze visitor data for various purposes, including security, service optimization, and resource planning.

By employing data analysis techniques, we can extract valuable insights from the collected data, leading to improved operational efficiency and enhanced passenger experiences. The integration of computer vision technology, specifically OpenCV (Open-Source Computer Vision Library), provides an intelligent solution to automate the control of fans and lights at railway stations. OpenCV enables the development of real-time image processing applications, allowing for the detection and analysis of human presence in each area. The proposed system aims to enhance the energy efficiency of railway stations by automating the activation and deactivation of fans and lights based on the occupancy of specific zones. This not only contributes to energy conservation but also promotes a greener and more sustainable approach to railway station management.

The project also aims to develop a fire detection system using OpenCV, a popular computer vision library in Python. Fire detection is crucial for ensuring the safety of both people and property, and leveraging computer vision techniques can enhance the speed and accuracy of detecting fires in various environments. In this Project we also implement the Data analysis to store the data of visitors in a railway station, data analysis can be employed to gather, process, and manage the information effectively.

1. **Objective**

The primary objective of the Railway Station Management System using OpenCV is to create an intelligent and automated system that addresses various aspects of station management. This includes temperature-controlled automated fan systems and intelligent lighting control.

* **Lighting Control System**: The lights are programmed to automatically turn off during the day to ensure well-lit and safe surroundings for passengers. As night falls, the system switches on necessary lights to conserve energy while maintaining adequate illumination for security and navigation.
* **Fire Detection**: This project is to create a robust and real-time fire detection system using computer vision techniques. OpenCV will be utilized to process video streams or static images, identifying regions with characteristics indicative of fire.
* **Data Analysis:** Data analysis enables the monitoring and prediction of passenger traffic patterns. This information can be used to optimize station layouts, staffing levels, and other resources to manage crowds more effectively, reducing congestion and improving the overall passenger experience.
* **Automated Fan System**: One of the key features of the system is the implementation of an automated fan system. The system monitors the ambient temperature and activates or deactivates fans accordingly. This not only ensures a comfortable environment for passengers but also contributes to energy efficiency by avoiding unnecessary fan operation during cooler temperatures.
* **OpenCV Integration**: The integration of OpenCV into the Railway Station Management System adds a layer of intelligence through computer vision. OpenCV allows for real-time monitoring of various parameters, such as crowd density, security surveillance, and facial recognition for enhanced security measures. The system utilizes OpenCV algorithms to analyze video feeds from strategically placed cameras, providing valuable insights to station authorities.

1. **Literature Survey**

[1] Reconstructing a three-dimensional (3D) model of a scene from photographs taken from different view-points, i.e. Structure from Motion (SFM), starts with detecting features in each photograph and matching the detected features in a subsequent photograph or photographs

[2] There exists a wide range of feature detectors, e.g. ScaleInvariant Feature Transform (SIFT) The SIFT (Scale-Invariant Feature Transform) algorithm is a computer vision technique used for feature detection and description. It detects distinctive key points or features in an image that are robust to changes in scale, rotation, and affine transformations

[3] Speeded-Up Robust Features (SURF) is a patented local feature detector and descriptor.It can be used for tasks such as object recognition, image registration, classification, or 3D reconstruction. It is partly inspired by the scale-invariant feature transform (SIFT) descriptor.

[4] Binary Robust Invariant Scalable Key points is Decomposing an image into local regions of interest or ‘features’ is a widely applied technique in Computer Vision used to alleviate complexity while exploiting local appearance properties

[5] Oriented FAST and Rotated BRIEF (ORB) ORB is basically a fusion of FAST keypoint detector and BRIEF descriptor with many modifications to enhance the performance. First it use FAST to find keypoints, then apply Harris corner measure to find top N points among them. It also use pyramid to produce multiscale-features

[6] AKAZE The AKAZE algorithm is a typical image registration algorithm that has the advantage of high computational efficiency based on non-linear diffusion. However, it is weaker than the scale-invariant feature transformation (SIFT) algorithm in terms of robustness and stability

**4 Proposed Work:**

4.1Methodology

1. Automated Presence Detection: OpenCV will be utilized to analyze live video feeds from strategically placed cameras in different zones of the railway station. The system will detect the presence of passengers in these areas.
2. Zone-Based Control: The railway station will be divided into zones, each equipped with its own set of fans and lights. OpenCV algorithms will determine the occupancy of each zone, triggering the activation or deactivation of fans and lights accordingly.
3. Real-time Monitoring: The system will provide real-time monitoring of passenger movements, ensuring that fans and lights respond promptly to changes in occupancy. This dynamic control mechanism adapts to the ever-changing foot traffic in different areas of the station.

4.2Workflow:

1. Install OpenCV: Install OpenCV on the computer.
2. Set up Cameras: Install and configure cameras at strategic locations in the railway station.
3. Write Python Scripts for Object Detection and Tracking: Develop Python scripts using OpenCV to detect and track objects (people, vehicles, etc.) in the camera feed.
4. Implement Security Features: Integrate security features such as fire detection.
5. Database Integration: Set up a database to store information about detected objects, their locations, and timestamps.
6. Real-time Monitoring: Implement real-time monitoring capabilities to observe the railway station activities.
7. Alert System: Integrate an alert system to notify authorities in case of suspicious activities or security breaches.
8. User Interface (Optional): Develop a user interface for station staff or security personnel to monitor and control the system.
9. Testing and Optimization: Test the system under various scenarios and optimize the algorithms for better performance.
10. Deployment: Deploy the system at the railway station and ensure its proper functioning.

4.3 Hardware Requirements:

|  |  |  |
| --- | --- | --- |
| Components | Image | Functionality |
| Esp32 cam | ESP32-CAM, Camera Module Based On ESP32, OV2640 Camera and ... | The ESP32-CAM can be widely used in intelligent IoT applications such as wireless video monitoring, Wi-Fi image upload, QR identification, and so on |
| ESP8266 | Introduction to the Internet of Things (IoT): ESP8266 architecture and  Arduino GUI | The ESP8266 is a low-cost Wi-Fi microcontroller chip that allows microcontrollers to connect to 2.4 GHz Wi-Fi. It can be used to enable the Internet of Things (IoT) and help exchange information between connected objects |
| DHT11 (temperature sensor) | DHT 11 Temperature Humidity Sensor Module, 5.5v Dc, Serial(single-wire  Two-way) at Rs 60/piece in Mumbai | The DHT11 is a low-cost digital sensor that measures temperature and humidity |
| LDR sensor | LDR Sensor Module (Light Dependent Resistor), LDR-MOD Sensor ... | Light Dependent Resistors are often used as light sensors. They are usually utilized when it is required to detect the presence and absence of light or to measure the light intensity |
| DC motor | Benefits of DC Motors for Robotics | automate.org | The term 'DC motor' is used to refer to any rotary electrical machine that converts direct current electrical energy into mechanical energy |
| LED | Buy 8mm DIP LED Red Online at Best Price in India | Robu.in | The high efficiency and directional nature of LEDs makes them ideal for many industrial uses. LEDs are increasingly common in street lights, parking garage lighting, walkway and other outdoor area lighting, refrigerated case lighting, modular lighting, and task lighting. |

4.4Software Requirement

1. OpenCV library for Python
2. Python scripts for automation
3. Roboflow (to democratize computer vision by streamlining the computer vision process from end to end)
4. YOLO (You Only Look Once) is a popular object detection algorithm

4.5 Working structure of automated light by using OpenCv

Here's a general working structure for implementing this system:

1. Capture Video Feed:

* Use a webcam or any suitable camera to capture a live video feed.
* OpenCV provides functions to interface with cameras and capture frames.

1. Image Processing:
   * + Capture frames continuously from the video feed.
     + Convert the captured frame to grayscale to simplify processing.
     + Apply image processing techniques to determine the average brightness or luminance of the scene.

3. Thresholding:

* Establish a threshold value to classify the scene as either bright (daytime) or dark (nighttime).
* Compare the average brightness of the frame with the threshold to make the decision.

4. Light Control Logic:

* If the average brightness is above the threshold, consider it daytime.
* Turn off the lights if they are currently on.

If average brightness > threshold

turn\_off\_lights ()

* If the average brightness is below the threshold, consider it nighttime.
* Turn on the lights if they are currently off.

else:

turn\_on\_lights ()

5. Light Control Functions:

* Define functions to control the lights.
* This may involve interfacing with smart home systems, IoT devices, or directly controlling lighting circuits.

def turn\_on\_lights ():

Pass # Code to turn on lights

def turn\_off\_lights ():

Pass # Code to turn off lights

Pass

1. Continuous Operation:

* Implement a loop to continuously capture frames, process them, and control the lights based on the determined lighting conditions.

while True:

# Capture frame from the video feed

frame = capture\_frame ()

# Process the frame and calculate average brightness

average\_brightness = process\_frame (frame)

# Control the lights based on the average brightness

light\_control\_logic (average\_brightness)

7. Optimization and Calibration:

* Fine-tune the threshold values and any other parameters based on the specific lighting conditions of the environment.
* Test the system under various lighting scenarios to ensure accurate and reliable operation.

1. Integration with External Systems:

* If applicable, integrate the system with external sensors or smart home platforms to enhance functionality.
* Ensure proper error handling and logging for system reliability.

4.6Working structure of Fire Detection by using OpenCv

Here's a general working structure for implementing this system:

1. Data Collection:
   * Gather a dataset containing diverse images and videos of fires, as well as non-fire scenarios.
   * Ensure that the dataset includes various lighting conditions, perspectives, and environments to enhance the robustness of the model.
2. Data Preprocessing:
   * Resize images to a consistent format.
   * Normalize pixel values to a common scale.
   * Augment the dataset by applying transformations like rotation, flipping, or changes in brightness to improve model generalization.
3. Training Data Split:

* Divide the dataset into training and testing sets to evaluate the model's performance accurately.

1. Fire Detection Model Selection:
   * + Choose a suitable deep learning architecture for fire detection. Common choices include Convolutional Neural Networks (CNNs) due to their effectiveness in image classification tasks.
     + Fine-tune a pre-trained model (like a ResNet or MobileNet) on your fire detection dataset or train a model from scratch if you have a sufficiently large dataset.
2. Model Training:
   * + Train the selected model using the training dataset.
     + Utilize transfer learning if using a pre-trained model.
     + Monitor the model's performance on the validation set to prevent over fitting.
3. Model Evaluation:

* Evaluate the trained model on the testing dataset to assess its performance in terms of accuracy, precision, recall, and F1 score.
* Tweak hyperparameters or the model architecture if necessary.

1. Integration with OpenCV:
   * Implement an interface with OpenCV for real-time image or video stream processing.
   * Utilize OpenCV functions to capture frames from a camera or video file.
2. Image Preprocessing in Real-Time:

* Preprocess each frame captured by OpenCV similarly to the preprocessing steps applied during training.

1. Fire Detection Inference:

* Apply the trained model to each preprocessed frame to perform inference.
* Set a threshold for classification confidence to determine if the frame contains fire.

1. Alert System:

* Implement an alert system to notify relevant authorities or users in case of a detected fire.
* This could include visual indicators on a screen, sounding alarms, or sending notifications.

4.7Working structure of Data Analysis by using OpenCV

1. Data Collection:

* Use surveillance cameras or video feeds to capture footage of the railway station area.

1. Preprocessing:
   * + - Preprocess the video frames to enhance quality and reduce noise. This might involve tasks like resizing, denoising, and converting to grayscale.
2. Object Detection:

* Utilize OpenCV's object detection algorithms, such as Haar cascades or more advanced deep learning-based models like YOLO (You Only Look Once), to detect people in each frame of the video.

1. Counting Logic:

* Develop counting logic to track the movement of individuals across frames. This could involve techniques like centroid tracking or background subtraction to determine when individuals enter or exit the scene.

1. Visualization and Analysis:

* Once you've processed the video and counted the visitors, you can visualize the data using graphs or charts to analyze trends over time. This analysis can provide insights into peak hours, crowd densities, or other relevant metrics.

1. Optimization and Validation:

* Fine-tune your algorithms and validate their accuracy by comparing the automated counts with manual counts or ground truth data.

1. Deployment:

* Deploy the solution to run in real-time or batch processing mode, depending on your requirements.
  1. Working structure of Automated Fan by using OpenCV

1. Temperature Sensors:

* Sensors are strategically placed across the railway station to continuously monitor temperature levels in various zones.

1. OpenCV Integration:

* OpenCV is utilized to process live video feeds from cameras installed at different locations within the station. Image processing techniques are applied to detect and analyze temperature variations in the captured images.

1. Machine Learning Algorithms:

* Machine learning algorithms are trained to recognize patterns and identify regions of interest corresponding to elevated temperature levels.

1. Fan Control Mechanism:

* Based on the temperature readings obtained from the sensors and the analysis conducted by OpenCV, the system automatically adjusts the speed of fans installed in the respective zones to maintain comfortable temperature levels.

3.9 Block Diagram

ESP 8266

ESP 32 Cam

LDR Sensor

Laptop

DHT11

OpenCV

Light

Thing Worx

FAN

1. Advantages of this project
2. Energy Efficiency: By automating the operation of fans and lights based on actual passenger presence, the system optimizes energy usage. Unoccupied zones will experience reduced energy consumption, contributing to cost savings and environmental conservation.
3. Informative & User-Friendly Interface: A user-friendly interface will be developed for administrators to configure and monitor the system. It may include features such as setting threshold occupancy levels, adjusting fan and light settings, and viewing historical usage data.
4. Real-time Monitoring: Continuous monitoring of passenger movement using OpenCV for instant adaptation to be changing conditions.
5. Automation: Automated fire detection systems reduce the reliance on human surveillance. By employing computer vision algorithms, we can continuously monitor areas for signs of fire without the need for constant human attention.
6. Security Enhancement: Analyzing visitor data allows railway authorities to identify and track individuals, enhancing security measures. Patterns and anomalies in passenger movements can be detected, aiding in the prevention of security breaches and ensuring the safety of both passengers and staff.
7. Future Scope
8. Weapon Detection: Using OpenCV for weapon detection in railway stations has promising future scope. With advancements in computer vision algorithms and hardware capabilities, it's feasible to develop real-time surveillance systems that can detect weapons and enhance security measures. By leveraging techniques such as object detection, deep learning, and image processing, OpenCV can be utilized to identify weapons in video streams from railway station cameras, enabling authorities to respond quickly to potential threats. Additionally, integrating such systems with alert mechanisms can further enhance safety measures and help prevent security breaches.
9. Garbage Detection: Using OpenCV for garbage detection in railway detection systems offers promising prospects for enhancing railway safety and cleanliness. By employing computer vision algorithms, such as object detection and image segmentation, railway authorities can identify and classify various types of garbage and debris on tracks or within station premises. This can help automate the process of detecting and removing garbage, thereby reducing the risk of accidents and ensuring smoother railway operations. Additionally, integrating machine learning techniques with OpenCV can improve the accuracy and efficiency of garbage detection systems over time. With further research and development, this technology could significantly contribute to maintaining cleaner and safer railway environments.
10. Cross-platform line detection: Cross-platform line detection in railway detection using OpenCV could indeed have significant future scope. By developing such a system, you could potentially improve railway safety, automate inspections, and enhance maintenance processes. It could also aid in monitoring track conditions and detecting anomalies early on, ultimately contributing to more efficient and safer railway operations.
11. Conclusion

In conclusion, the implementation of an automated railway station equipped with systems to control lighting, fans, fire detection, and visitor counting offers numerous benefits. By integrating these technologies, railway stations can enhance safety, energy efficiency, and operational efficiency. Automated lighting and fan controls ensure optimal energy usage and comfort for passengers and staff. Fire detection systems provide early warning and rapid response to prevent disasters. Visitor counting technology enables better management of crowd flow and resource allocation. Together, these automated systems contribute to a more secure, convenient, and efficient railway station experience for all stakeholders involved.

8. Reference

[1] R. Hartley and A. Zisserman, Multiple view geometry in computer vision. Cambridge university press, 2003

[2] D. G. Lowe, “Distinctive image features from scale-invariant key points,” International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004

[3] H. Bay, T. Tutelars, and L. Vavanool, “Surf: Speeded up robust features,” in Computer vision–ECCV 2006. Springer, 2006, pp. 404– 417.

[4] S. Leute egger, M. Chli, and R. Y. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” in Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 2548–2555.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: an efficient alternative to sift or surf,” in Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 2564–2571.

[6] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, “Kaze features,” in Computer Vision–ECCV 2012. Springer, 2012, pp. 214–227

[7] P. F. Alcantarilla and T. Solutions, “Fast explicit diffusion for accelerated features in nonlinear scale spaces,” IEEE Trans. Patt. Anal. Mach. Intell, vol. 34, no. 7, pp. 1281–1298, 2011.

[8] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration.” VISAPP (1), vol. 2, pp. 331–340, 2009.

[9] G. Bradski and A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library. ” O’Reilly Media, Inc.”, 2008.

[10] Zahraa Shihab Al Hakeem, Hawaraa H.Abass, Haider Ismael Shahadi, “An Automatic System for Detection of Fires in Outdoor Areas” in IEEE Access. DOI: 10.1109/ICECET55527.2022.9872883