**Research Aim: Investigating the detection capabilities and efficacy of OpenCV's computer vision technology in identifying pedestrians under various image quality conditions during driving. The goal is to assess the robustness and limitations of pedestrian detection across diverse scenarios.**

**Research Hypothesis: Testing the effectiveness of OpenCV's computer vision in detecting pedestrians while driving with varying different image quality scenarios, with higher-quality images yielding more accurate detection rates compared to lower-quality images.**

**Research Questions:**

1. How will lighting effect the detection process such as daytime and night?
2. Can compression and resolution effect the quality and ability of the detection process?
3. Will lower quality videos effect the detection process compared to higher quality?

**The development of pedestrian detectors serves a critical role in safety especially when it comes to urban areas with multiple pedestrians with crowded sidewalks and intersections. In these environments the development of such detection systems aid multiple drivers avoiding accidents with high traffic volumes, complex road networks and constant danger of pedestrians crossing streets. Testing pedestrian detection systems in multiple scenarios is important in ensuring their reliability and effectiveness across multiple environments and conditions The prototype script is written in Python and consist of using OpenCV, Histogram of Oriented Gradients (HOG) to detect pedestrians within a specified video file. OpenCV provides comprehensive support for computer vision, offering functionalities and algorithms for image and video processing. HOG is used to capture the distinctive features that make up a pedestrian such as the edges and gradients. All of this would allow for real time processing detection which is used for each frame in the prototype. After initialization, the designated video file is opened with a OpenCVs function and frames from the video file is extracted and resized to a width of 400 pixels for better performance and a lack of drop in frames while viewing the video in the results. After the implemented video is processed the detection process starts via HOG with each frame of the video using a function in the HOG library “detectMultiScale()”. This does multiple scans on each resized frame to identify potential pedestrian locations based on learned patterns. Detected pedestrian are then visually indicated by drawn rectangle around the general area were with the edges of the pedestrian is seen, marking were the OpenCV function “Rectangle()” has spotted and marked a pedestrian. Lastly, the prototype will display the processed video to the user, showcasing the results of the pedestrian detection algorithm. A set of experiments are conducted on the program to record results of different quality-based tests to examine, and stress test the detection process. The experiments will consist** of multiple videos and will be captured by a **camera that will be positioned on the front dashboard, ensuring that the captured footage closely resembles the visual input experienced during actual driving scenarios.** The first set of videos will be captured during daylight, simulating normal driving conditions to assess the model's detection functionalities. The second set of videos will be recorded at night to examine the model's ability to detect pedestrians under low-light conditions. Within the imutils package is a function “resize()” that is used to resize images while keeping their aspect ratio. This function will be used in the prototype and will be set to 400 on initial usage for the previously mentioned tests. Following these initial experiments, an additional test will be conducted to evaluate the performance of the pedestrian detection model on less compressed videos with higher resolutions. **To simulate this, the resize() function will be adjusted to a width of 700. By increasing the width amount, the images will become less compressed, providing higher resolution and potentially more detailed information for the pedestrian detection model to analyse. This change aims to test the model's accuracy when presented with higher-quality input data.**

Pipeline/plan

**Phase 1 Initial Research and Setup**

|  |  |  |
| --- | --- | --- |
| **Pedestrian detection via computer vision, uses OpenCV and the impact of image quality and hardware constraints on algorithm efficacy.** | **High-level Semantic Feature Detection: ANewPerspective for Pedestrian Detection**  **CSANet: Channel and Spatial Mixed Attention CNN for Pedestrian Detection** | **Operating system: Windows 10**  **Mobile device**  **Libraries Required**  **OpenCV**  **Imutils** |

**Phase 2 Data Acquisition**

|  |  |  |
| --- | --- | --- |
| **Gather Data with multiple videos using the camera positioned in the front dashboard, footage vary with different scenarios, including daylight, nighttime, and reduced-quality daytime conditions** | **Analyse the data to discern the features and patterns related to image quality and temporal dynamics.** | **Organization of the dataset by adjusting image formats, extracting key features** |

**Phase 3 Experimentation**

|  |  |  |
| --- | --- | --- |
| **Development on the pedestrian detection algorithm using the OpenCV model** | **Analyse the videos to identify patterns and key features in the results of the experiment** | **Investigate the effectiveness of OpenCV's pedestrian detection algorithms under various image quality scenarios** |

**Phase 4 Evaluation**

|  |  |  |
| --- | --- | --- |
| **Assessing the accuracy of experiments and data quality to determine where the pedestrian detection algorithm performs well and encounters challenges.** | **Testing the prototype performance in detecting pedestrians across various scenarios to determine strengths and weaknesses** | **Acknowledging constraints, such as hardware dependencies, and enhancing the algorithm to be more effective and adaptable in different conditions.** |