Final Project For ECE228 Track Number 2

Group Number 12: Sai Praneeth, Potladurthy **Manoj Kumar Reddy, Manchala**

Sesha Sai Rakesh, Jonnavithula

Abstract

Advancements in machine learning algorithms have significantly advanced vision-guided autonomous driving systems. However, ensuring the highest safety standards remains a critical challenge, particularly concerning the misalignment between the onboard camera and the vehicle. Accurate estimation of this misalignment is crucial for precise vehicle control. This research focuses on predicting the vehicle's direction of motion based on camera inputs to address this issue. This estimation of pitch and yaw will facilitate the creation of datasets for enhancing the autonomous navigation models.

The primary technical challenge is achieving precise motion prediction under varying environmental conditions and road scenarios. The limited and often inconsistent data, characterized by dynamic lighting, movement, and noise, further complicates the training of our algorithm.

Our approach involves experimenting with feature extraction techniques from camera frames and designing a weighted loss function to map the pitch and yaw angles. We train a hybrid network composed of convolutions and sequence processing to generate an accurate representation of the vehicle's direction of travel.

The project aims to accurately predict the direction of travel for autonomous vehicles based on camera feeds, aiding in the calibration of onboard camera systems. The implementation is expected to refine camera parameters while reducing computational costs, providing a practical tool for the broader autonomous vehicle community. The code is available at the github link : `https://github.com/potladurthy/ECE_228_Project.git`

I certify that I have filled the evaluation.

1 Introduction

The advent of autonomous driving systems has been one of the most transformative advancements in the automotive industry. Leveraging sophisticated machine learning algorithms, these systems have progressed from conceptual designs to practical implementations, demonstrating the potential to revolutionize transportation by enhancing safety, efficiency, and convenience. Central to the functionality of these autonomous systems is the vision-guided navigation, where onboard cameras play a pivotal role in perceiving the environment and informing control decisions.

Despite the substantial progress, numerous challenges persist in ensuring the highest safety standards for autonomous vehicles. One critical issue is the misalignment between the onboard camera and the vehicle's coordinate system. Such mis-alignments can lead to inaccurate perceptions of the vehicle's surroundings, resulting in erroneous control actions. Therefore, accurately estimating and correcting this misalignment is vital for the reliable operation of autonomous driving systems.

A promising approach to address this challenge involves predicting the vehicle's direction of motion based on the camera input. By understanding the motion dynamics, it is possible to infer the necessary adjustments for camera alignment. However, this task is complicated by various factors, including dynamic environmental conditions, diverse road scenarios, and the inherent noise and inconsistencies in the data captured by the cameras.

The problem of predicting the direction of travel from a single camera input has been the subject of various previous works, such as Huang et al. [2022]. Capturing the impacts of pitch and yaw shifts in the three-dimensional real environment through a two-dimensional data set (image) is the primary issue. The task at hand is twofold: first, to accurately forecast motion and, second, to manage the sparse and noisy data that is currently available. Variability in weather, road architecture, and lighting adds levels of complexity that call for strong, flexible algorithms that can be applied to a variety of scenarios.

Another advantage of estimating pitch and yaw from camera frames is that we can utilize this technique to assist in dataset preparation, which otherwise relies on conventional techniques like in-vehicle sensors (ex: IMU (Inertial Measurement Unit)) to obtain the data, making it cost-inefficient.

To tackle this ill posed problem of estimating pitch and yaw directly, different approaches have been considered.

1. Extracting the relative motion using Optical Flow.
2. Estimating a point in the frame called 'Vanishing Point'.

1.1 Optical Flow

Optical flow is a computational technique used to estimate the motion of objects between consecutive frames in a video sequence. It captures the apparent motion of brightness patterns in the image, representing the velocity of each pixel. We hypothesize that optical flow features, when combined with the original (RGB) images, will supplement the model to focus on the parts of the spatial information where relative motion is observed.

Given two consecutive grayscale frames I_t and I_{t+1}, the optical flow $\mathbf{u} = (u, v)$ at each pixel is computed using the Farneback method, which approximates the neighborhood of each pixel by quadratic polynomials and solves for the displacement that minimizes the intensity difference between the frames. The mathematical formulation is as follows:

$$\min_{\mathbf{u}} \sum_{\mathbf{x}} \left(I_{t+1}(\mathbf{x}+\mathbf{u}) - I_t(\mathbf{x})\right)^2 \tag{1}$$

where $\mathbf{x}$ denotes the pixel coordinates. This method generates a dense optical flow field, providing motion vectors for each pixel, and the flow field has the same dimensions as the input image.

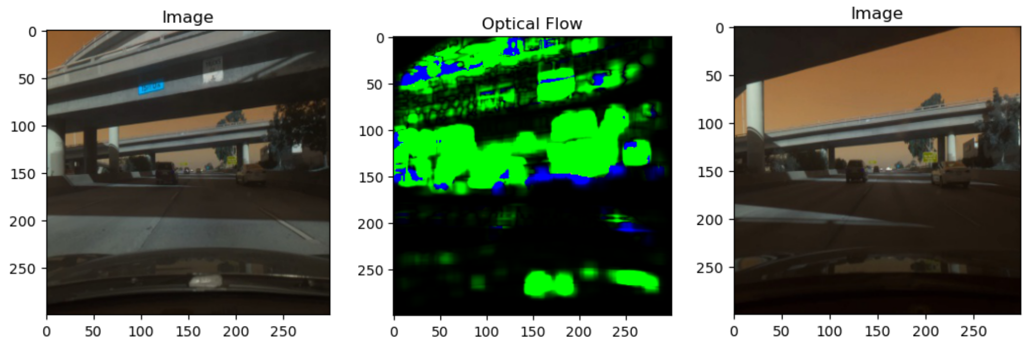

Figure 1: Images and corresponding Optical Flow Features

The centre image of figure 1 depicts the optical flow features extracted between the frames, the left and right images of figure 1. Each colour in the optical flow field represents different motion directions. The darker portion (black) indicates that there is no relative change between the corresponding pixels of the frames considered.

1.2 Vanishing Point

A vanishing point is a point on the horizon within the image where the projection of parallel lines appears to meet in 3D space. In this context, the vanishing point is considered the intersection of the extrapolated lane markings. Our intuition is to build a network to estimate the coordinates of the vanishing point Chang et al. [2018] using specific features from the images, such as lane markings, etc.. Zou et al. [2020]. The projection of the vanishing point into the real world then generates a direction vector, which, when combined with the rotation matrix of the world coordinates, treating it as a regression problem, gives the pitch and yaw of the image. This approach involves identifying specific desirable features, such as lane markings, and also requires multiple networks to predict the pitch and yaw angles. Owing to computational and real-time inference constraints, this turned out to be infeasible and thus this approach is not considered further.

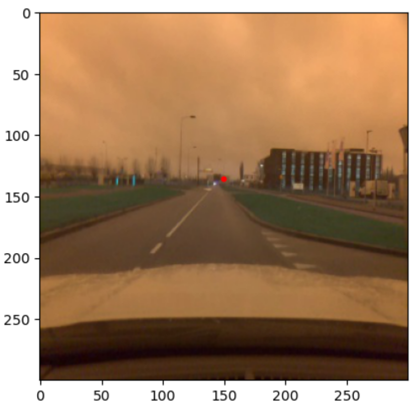

Figure 2: Vanishing Point

2 Dataset

The problem of estimating the pitch & yaw from monocular camera frames of a video is challenging, exclusively in this scenario where there are 5 labeled videos of 1 minute each with 20 fps, leading to 1200 frames per video. With very limited data, additional constraints like missing labels for the timestamps where the speed of the car in the video is less than 4 m/s add an additional overhead of pre-processing and cause errors in estimation. Furthermore, the dataset imbalance is huge; around 80% of times the movement of a car is forward with very low relative change in the pitch & yaw between successive frames, and only 20% of the video contains instances where the direction of movement is not straight, which is the actual instant that needs to be considered to accurately estimate the direction of travel. The dataset is obtained from dat.

3 Related work

To extract useful information utilizing spatio-temporal data like videos, Zou et al. [2020] implemented a hybrid architecture combining convolutional neural networks (CNN) with recurrent networks (RNN) that allow processing frames as sequences, embedding temporal information, unlike standard convolutions where each frame is processed independently of the other frames in the video. Inspired by this, we adopted the same approach of combining CNN with long-short-term memory (LSTM) to process videos as sequences of frames.

Estimation of pitch & yaw Huang et al. [2022] angles directly from a monocular camera frame is an ill-posed problem, thus approaching it uniquely to first get a point in the frame to estimate the direction of travel called the VANISHING POINT. The Chang et al. [2018] uses a deep learning approach to estimate this vanishing point using a simple multi-class classification problem, but it is not ideal to deal with a high-sparsity multi-dimensional approach; hence, we modified the approach as a regression problem to estimate the vanishing point coordinates directly.

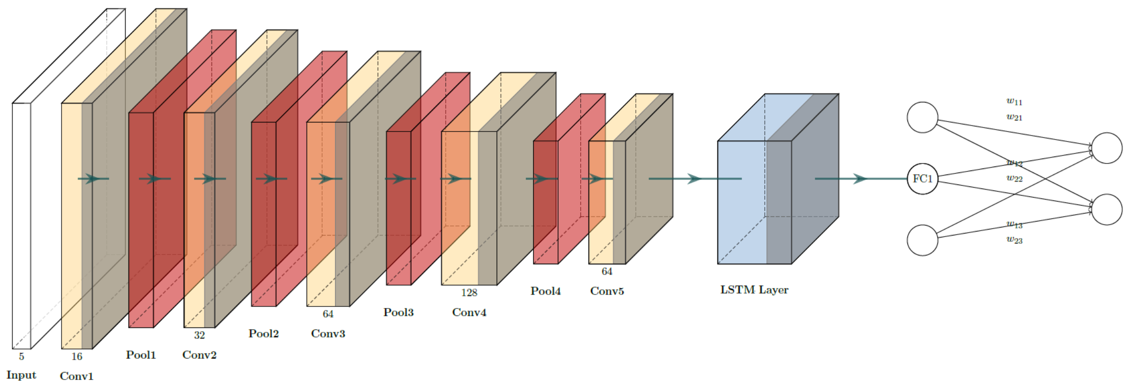

Figure 3: Architecture of Conv-LSTM network

4 Methodology

4.1 Data Preprocessing

This is a preliminary stage where raw data is refined into a format suitable to directly train the model. There are series of steps that are used to refine the raw videos provided.

1. The missing labels, which account for nearly 40% of entire data, are replaced with interpolated labels, this step ensures that each frame has a label associated with it. The choice of interpolating labels is supported by the fact that missing labels are related to car at rest, indicating no movement and can be considered as moving forward.
2. Videos are processed into frames by applying augmentations like resizing, any other augmentation, like random flips, cannot be performed as it completely alters the state of pitch and yaw.
3. As an additional step, optical flow features are calculated using equation 1 and stored. A noteworthy observation is the size of the optical flow features. Ideally, if there are N frames in a video, we end up with N-1 optical flow features of the same size. Thus, to match the input dimensions, the first optical flow feature is repeated.
4. Each image frame (RGB) is then combined with its corresponding optical flow vectors (relative to the current and previous frame) to form a composite input Kiziltepe et al. [2024] for the model, resulting in a 5-channel input (3 for RGB channels and 2 for optical flow components u and v). Note that the optical flow features are not contiguous, and thus the concatenated 5-channel data is made contiguous for smoother convolution operations.

4.2 Architecture

Processing video frames is not a straight-forward approach of treating frames as independent entities and performing spatial feature extraction. Thus, to incorporate the dimension of time, sequential processing is necessary, requiring the combination of both spatial and temporal information. To implement this approach, a hybrid Convolution-LSTM architecture is preferred. The model architecture is shown in figure 3.

The architecture of our model comprises a series of convolution layers, followed by average pooling, an LSTM network, and a feed-forward layer to directly predict the pitch and yaw angles, thus posing this as a regression problem.

The convolution layer applies convolution operations to the input data, using a set of learnable filters (kernels) to produce feature maps. Each filter slides over the input data spatially (e.g., an image) and performs element-wise multiplications followed by a summation, capturing local spatial patterns such as edges, textures, and shapes. Stacking multiple convolution layers enables the model to learn hierarchical features, progressively capturing more abstract and complex patterns.

Average pooling is used to reduce the spatial dimensions of the feature maps while retaining the most salient information. By taking the average of values within a defined window (pooling kernel), average pooling reduces the size of the feature maps, which helps to decrease computational complexity and mitigate over-fitting. Average pooling is preferred over Max pooling, as it has improved robustness to noise. Also, Max pooling might result in the loss of relevant features such as lane markings due to their inherent low intensity.

An LSTM (Long Short-Term Memory) network captures temporal dependencies in sequential data. Unlike standard RNNs, LSTMs are capable of learning long-term dependencies due to their unique architecture, which includes cell states and gating mechanisms (input, output, and forget gates). These gates regulate the flow of information, allowing the LSTM to remember important information over long periods and forget irrelevant details. In our work, the LSTM network processes the sequence of feature maps extracted by the convolution layers, capturing the temporal dynamics of the video frames.

Finally, a feed-forward layer (a fully connected layer) is used to map the output of the LSTM network to the target variables, pitch and yaw. This layer performs a linear transformation of the input data, followed by a nonlinear activation function (e.g., ReLU). The output of the feed-forward layer represents the predicted values of pitch and yaw.

4.3 Training and Evaluation

Unlike standard data processing, we create a custom dataset using video frames such that predictions are made for the center frame of a sequence (a set of frames). Specifically, a sequence is composed of frames from timestamps $\frac{-i}{2}$ to $\frac{i}{2}$, and the label is predicted for the center frame of the sequence. This approach allows for a holistic representation of direction by aggregating information from a set of frames. Our proposed method concatenates the images and their corresponding optical flow features resulting in a 5x256x256 input. Our hypothesis is that this representation will enhance the feature extraction process as optical flow features encode the relative motion between frames which will guide the convolutions to focus on certain regions where maximum difference is observed. Adding on to it LSTM will process this information across multiple frames to consolidate the features.

The training process involves feeding the model with the concatenated data. To meet the computational limitations a batch size of 10 with a sequence length of 15 is chosen and the proposed Conv-LSTM network is trained using SGD optimizer with a learning rate of 0.0001 using weight decay of 0.001 with dropout of 0.5 to regularize the network preventing weight saturation and over-fitting. In addition to that a learning rate scheduler is employed to prevent model from plateauing for longer duration.

The model is trained using a weighted Mean squared error loss function to account the dataset imbalance into training, the way of calculating loss is shown in equation 2.

$$MSE = \sum_{i} \omega_i (y_i - \tilde{y_i})^2 \tag{2}$$

where, ω_i represent the weight assigned to the label depending on the direction of movement manually observed from the video frames, this weighted label prioritizes the frames during training where turning is observed thereby allowing model to focus on the frames that are under-represented in the dataset.

5 Experiments

To evaluate the accuracy of the proposed model, the dataset is divided into a 4:1 ratio for training and testing. This split ensures a fair assessment of the results, as ground truth labels are available only for the training data. The final proposed model undergoes training for 30 epochs on an Nvidia RTX 4070 with 8 GB of memory.

We investigated several approaches for estimating pitch and yaw values using the Conv-LSTM model. The model trained using only RGB pictures from videos is shown in Figure 4. The fact that the predictions are consistent throughout suggests that the model is unable to identify the underlying patterns in the data by using RGB frames.

Furthermore, we looked into the idea of determining whether optical flow features alone are adequate for the same estimation. The predictions are shown in Figure 5 when the model is provided with

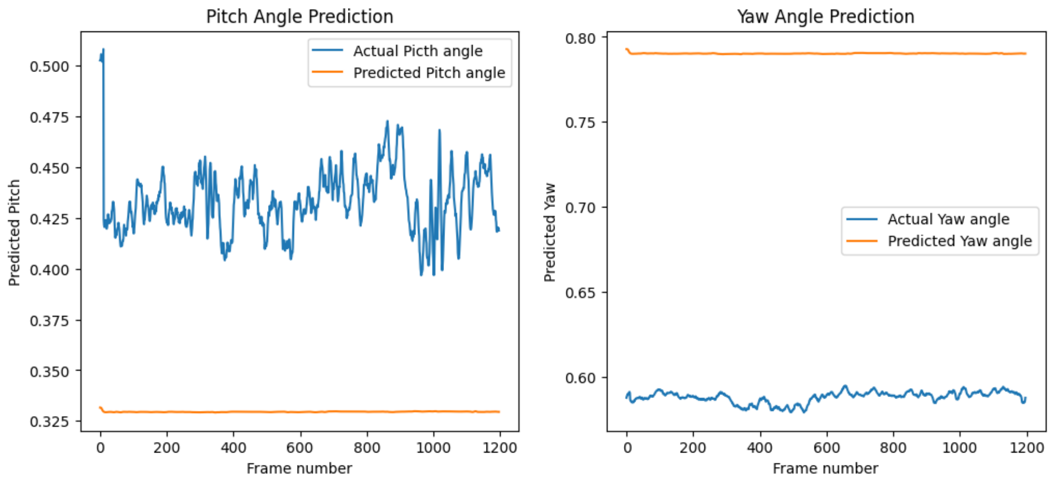

Figure 4: Predicted Pitch and Yaw using RGB images

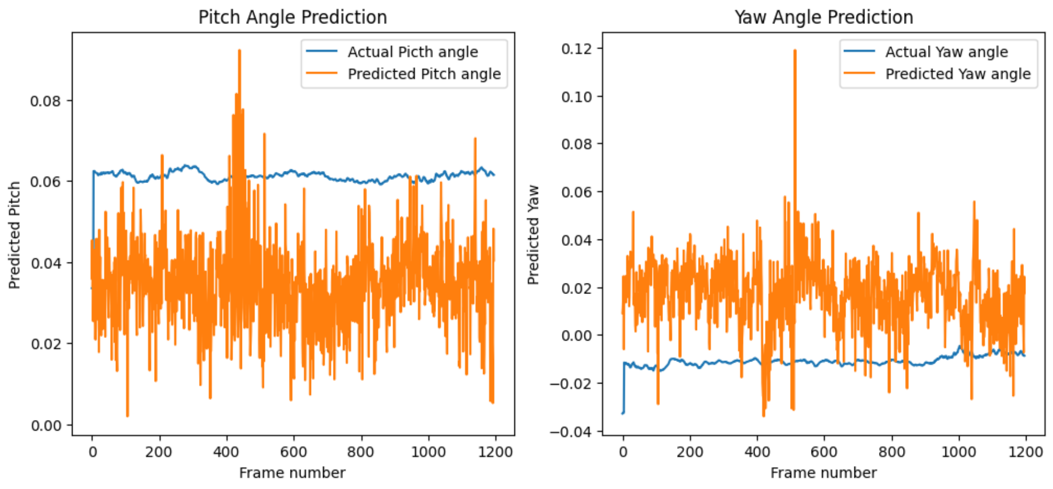

Figure 5: Predicted Pitch and Yaw using Optical flow features

optical flow features alone rather than RGB images. Difference between figure 4 and figure 5 emphasizes that optical flow can enhance the representation of complex motion patterns thereby supporting our hypothesis of fusing or concatenating these with RGB images Kiziltepe et al. [2024] to obtain better results. However, just the optical flow features are not effective in estimating the pitch and yaw as evident from 5 where the predictions are highly noisy.

Figure 6 compares the predicted pitch and yaw with the ground truth labels when both RGB and optical flow features are combined. The comparison reveals that while the model effectively captures the general trend in the direction of movement using combined input representation, the predicted values exhibit high variance. This variance might stem from the pre-processing step involving interpolated labels. Also close inspection of plots in figure 6 reveals that prediction of yaw is more accurate when compared to pitch. This can be attributed to that fact that

Figure 7 presents the train vs test loss curves. From figure 7, it is interesting to see that the loss decay is more smooth in case of only RGB input and only Optical flow features as input. The combination of both might have lead to a different loss landscape that resulted in model to over-fit just after 15-20 epochs. One possible reason is that the model might have stuck in local minima paving a path for the choice of different optimization techniques such as simulated annealing etc..

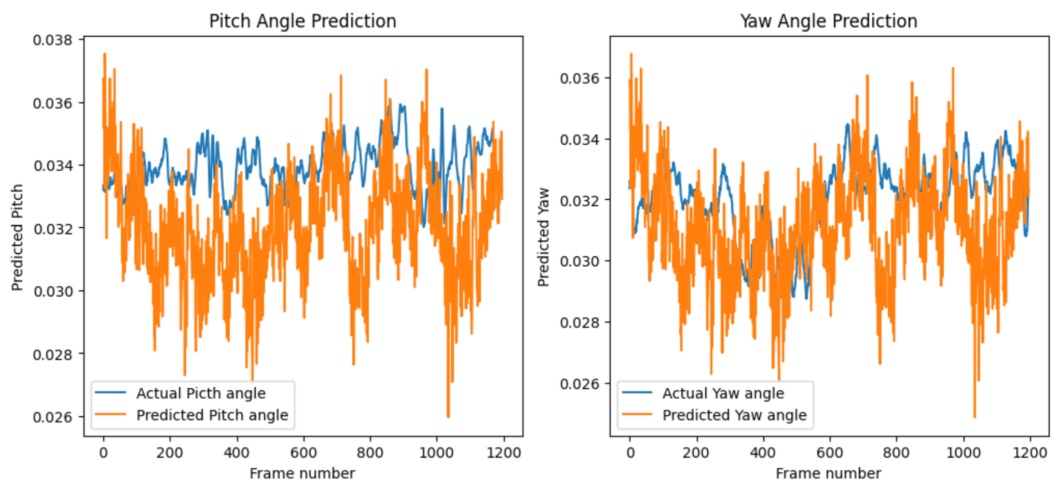

Figure 6: Predicted pitch and yaw values using combination of optical flow and RGB images

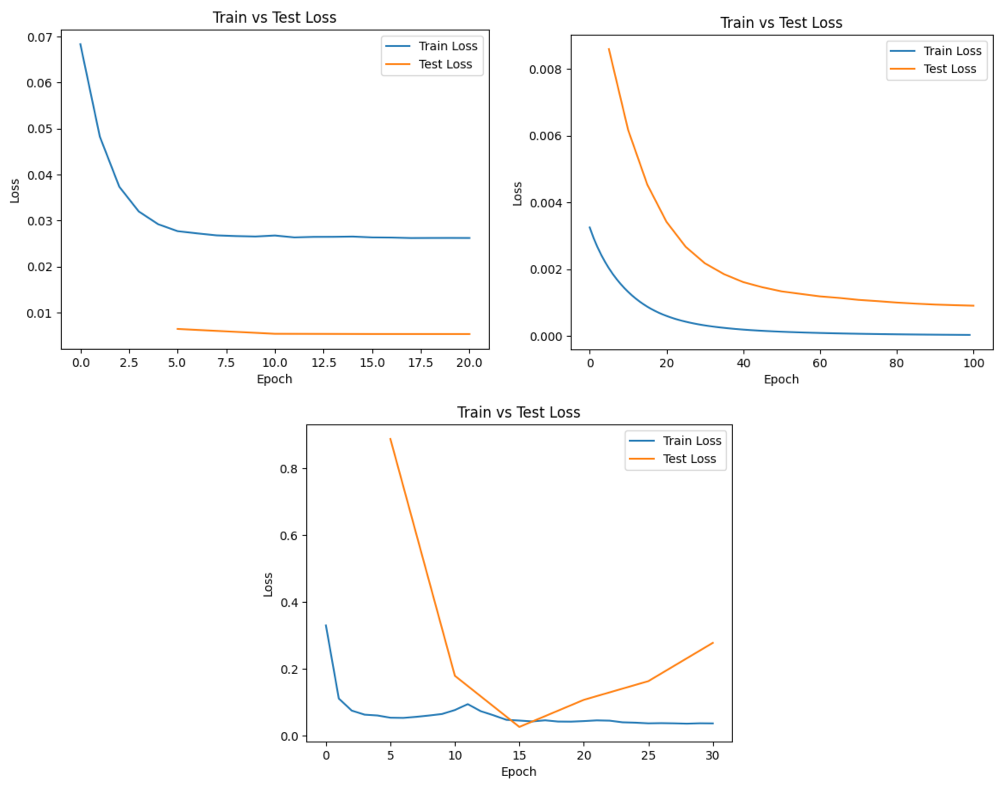

Figure 7: Loss curves : The left curve represents the Train vs Test loss for a model using RGB pictures, the right curve for a model using only optical flow features, and the bottom curve for a model that combines RGB and optical flow features.

The optimal metric for comparison is the estimation of mean squared error on the dataset. According to related studies van in this field, the baseline estimate for a reasonable approximation of pitch and yaw is less than or equal to 25%. Our experimental results show an approximation error of 21.72% for pitch and yaw over test set.

6 Conclusion

This project has demonstrated the potential of using deep learning techniques for accurate prediction of vehicle pitch and yaw angles based on monocular camera inputs. The methodology involved the integration of convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) networks to process video frames as sequences, effectively capturing the temporal dependencies crucial for motion prediction.

Our results indicate that the proposed hybrid Conv-LSTM model can effectively predict the general trend in movement direction, achieving a mean squared error approximation error of 21.72% for pitch and yaw, which is within the acceptable range based on related studies. However, the model's performance is hindered by high variance in predicted values, likely due to inconsistencies and noise in the preprocessed data, particularly from interpolated labels and limited data.

The reliance on pre-computed optical flow features has been identified as a limiting factor for real-time applications.

In conclusion, the project successfully illustrates the feasibility of using deep learning models for vehicle direction prediction, but also highlights the need for further refinement in data preprocessing and model architecture to enhance accuracy and real-time performance. These improvements are crucial for the broader adoption of this approach in autonomous driving systems, ensuring higher safety standards and more precise vehicle control.

7 Future scope

The current method depends on pre-computed optical flow features, which restricts its real-time application. Implementing advanced deep learning-based optical flow techniques could help overcome this limitation.

Rather than using standard LSTM networks, which are effective for shorter video sequences, we can utilize Vision-Transformers or 3D-convolutions. These are better suited for handling long videos where temporal dependencies span more than a few dozen frames.

References

AI calibration challenge. `https://github.com/commaai/calib_challenge`.

OpenCV based vanishing point detection. `https://github.com/gengrill/cnn-camera-calibration`.

C.-K. Chang, J. Zhao, and L. Itti. Deepvp: Deep learning for vanishing point detection on 1 million street view images. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018. doi: 10.1109/icra.2018.8460499. URL `http://dx.doi.org/10.1109/ICRA.2018.8460499`.

W. Huang, W. Li, L. Tang, X. Zhu, and B. Zou. A deep learning framework for accurate vehicle yaw angle estimation from a monocular camera based on part arrangement. *Sensors*, 22(20):8027, Oct. 2022. ISSN 1424-8220. doi: 10.3390/s22208027. URL `http://dx.doi.org/10.3390/s22208027`.

R. S. Kiziltepe, J. Q. Gan, and J. J. Escobar. Integration of feature and decision fusion with deep learning architectures for video classification. *IEEE Access*, 12:19432–19446, 2024. ISSN 2169-3536. doi: 10.1109/access.2024.3360929. URL `http://dx.doi.org/10.1109/ACCESS.2024.3360929`.

Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang. Robust lane detection from continuous driving scenes using deep neural networks. *IEEE Transactions on Vehicular Technology*, 69(1): 41–54, 2020.