**Data Preparation**

The dataset consists of video features extracted using three different deep learning backbones: R(2+1)D, ResNet, and VGG16, categorized into six maneuver classes: Left Lane Change, Right Lane Change, Left Turn, Right Turn, Slow-Stop, and Straight. Custom functions were implemented to handle the loading of VGG16/R(2+1)D and ResNet feature files. All feature files within a specific maneuver class folder were processed, extracting and reshaping features as needed while associating them with corresponding class labels. The iteration process takes place over the dataset paths and maneuver classes, utilizing parallel processing to efficiently load the feature files, aggregating features and labels from all backbones and maneuver classes. Principal Component Analysis (PCA) is applied to the ResNet features to project them into a common feature space with a dimensionality of 512, ensuring consistency in feature dimensions across different backbones. The features from all backbones are combined into a single feature matrix, while the labels are concatenated and encoded. The combined features are then reshaped to match the required input dimensions for the Convolutional Neural Network (CNN) model. The dataset is split into training and validation sets using an 80-20 split ratio, with pixel values normalized to the range [0, 1] to enhance the training performance of the neural network.

**Model Preparation**

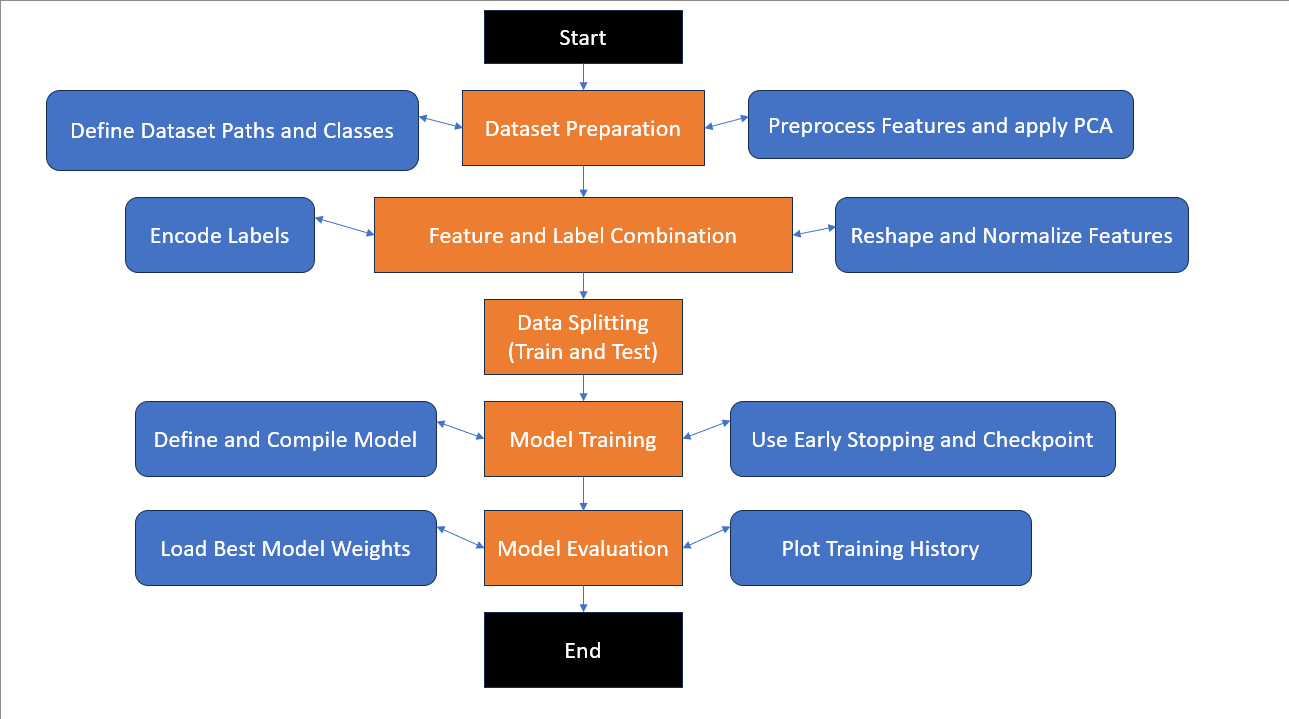
A robust 2D Convolutional Neural Network (CNN) model was designed to classify the maneuver classes. The model architecture consists of several convolutional layers, each followed by a max-pooling layer. The initial convolutional layer has 32 filters with a kernel size of 3x3, followed by a max-pooling layer with a pool size of 1x1. This pattern is repeated with 64 and 128 filters in subsequent convolutional layers. After the convolutional layers, a flatten layer is used to convert the 2D matrix data into a 1D vector, which is then fed into a dense layer with 512 units and ReLU activation. The output layer is a dense layer with a number of units equal to the number of maneuver classes, using a softmax activation function to produce a probability distribution over the classes. The model was compiled with the Adam optimizer and sparse categorical cross-entropy loss function, and the performance was evaluated using accuracy metrics. To prevent overfitting, early stopping and model checkpointing were utilized. Early stopping monitored the validation loss with a patience of 5 epochs, while the best model weights were saved using a model checkpoint callback. The model was trained on the training set and validated on the validation set for up to 50 epochs, with a batch size of 32. After training, the model weights corresponding to the best validation loss were loaded for final evaluation. The training history, including accuracy and loss over epochs, was plotted to visualize the training and validation progress. By leveraging the CNN architecture and effective training strategies, the model was optimized to classify maneuver classes from video features with high accuracy and reliability.

**Performance Measures**

The model's performance was evaluated using accuracy and F1 score metrics, which are critical for assessing the classification capabilities of the model. For the first task, the model achieved an impressive accuracy of 0.9793 and a F1 score of 0.9793. For the second task, the model's performance improved further, with accuracy of 0.9965 and F1 score of 0.9965. These metrics indicate that the model was highly effective in accurately classifying the maneuver classes, demonstrating both precision and reliability in its predictions. The high F1 scores across tasks highlight the model's balanced performance in terms of precision and recall, making it robust for practical applications.

**Methodology Flow Chart**

The entire process is described visually below:

****