Distracted Driver Detection using Multiple Action Detection

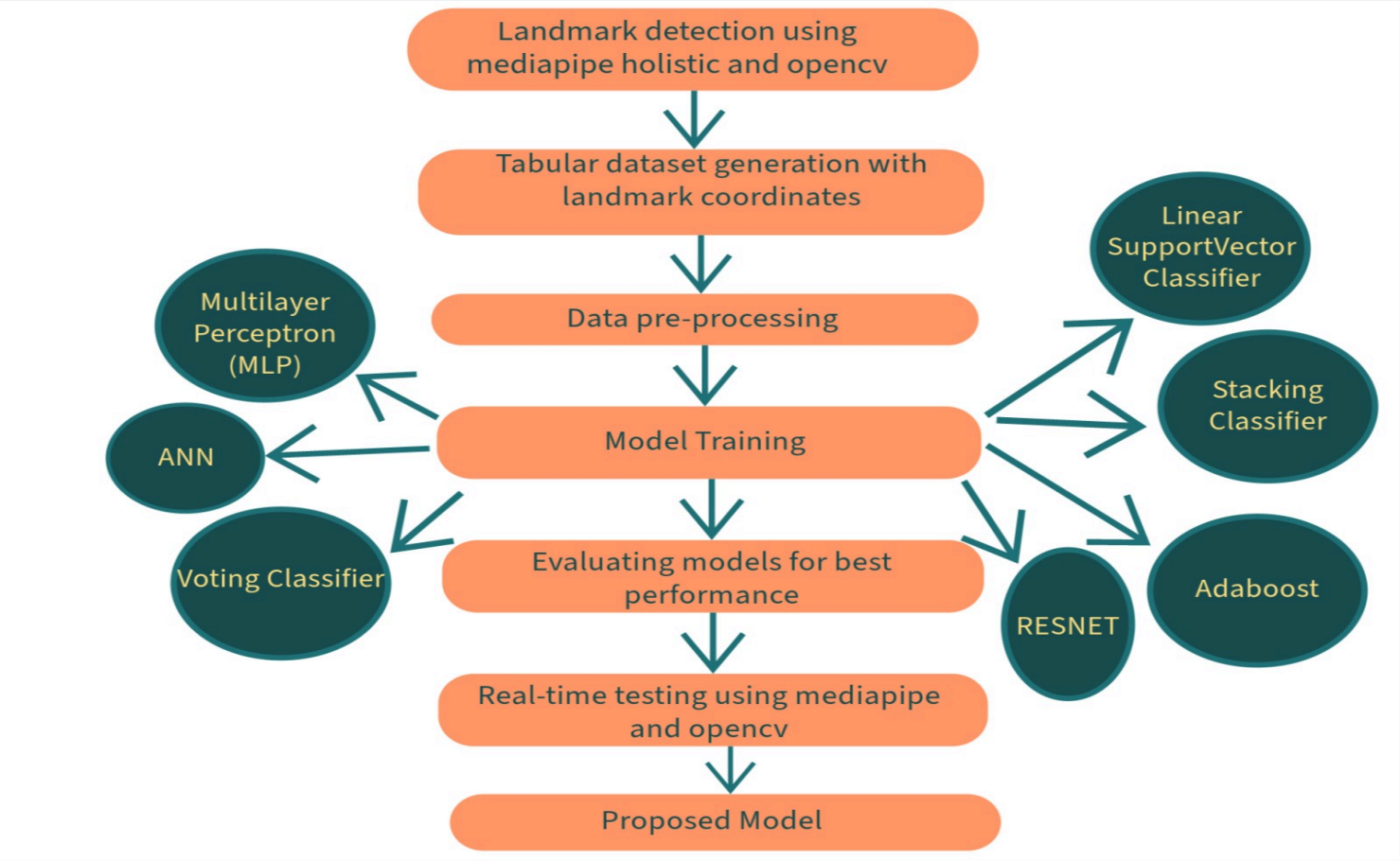
Problem

Driver distraction is a foremost cause for motor vehicle accidents and incidents. Driving requires an intensive amount of concentration otherwise the results can be fatal. Yet, most motor vehicles have no system in place to assist the driver when he is feeling drowsy, fatigued or distracted.The primary goal of distracted driving detection using computer vision is to identify and mitigate instances of distracted driving in real-time. This is achieved by analyzing visual cues from within the vehicle to recognize behaviors indicative of distraction, such as smartphone use, drowsiness, or other non-driving-related activities.The problem at hand is to develop a robust and accurate system for detecting instances of distracted driving in real-time. This system must be capable of analyzing driver behavior, identifying potential distractions, and providing timely warnings or interventions to prevent accidents.Developing an effective distracted driving detection system requires a multidisciplinary approach, incorporating machine learning, computer vision, and real-time processing techniques.

Project overview

The idea of distracted driving detection is based on detecting the actions that comes under driver distractions like sleeping, using smartphone for talking and texting, drinking water from bottle while driving etc.so in order to detect these actions we propose a model that uses mediapipe holistic landmark detection approach for detection of various actions using mediapipe face ,hand and pose landmark detection.the landmarks are consist of total 501 landmarks including 468 face landmarks and 33 pose landmarks with each landmark correspond to 4 coordinates each which are x,y,z and visibility in which x,y,z represents height width and depth and visibility coordinate represent whether the landmark is visible or not then after detecting all landmarks we then save the landmarks in csv file in order to generate dataset then we use opencv and mediapipe to detect various actions and generate dataset by detecting different landmarks corresponding to various classes of action of different people in order of getting more diverse dataset .After generating the dataset we do data preprocessing by removing null values and using dimensionality

reduction technique pca in order to reduce the dimension from 501x4 to only 500 after data preprocessing we are down to model training and for that we train our dataset on different models which are ensemble technique voting classifier(using logistic regression,ridge classifier,random forest,gradient boosting,svm,gaussian naive bias),MLP, ensemble stacking classifier(using svm,mlp,gradient boosting,random forest) ,ANN,Adaboost,Linear support vector classifier,Resnet and then doing hyperparameter tuning on these models and out of these we choose the model that give us best possible accuracy by evaluating the model using suitable evaluation metrics then we do realtime testing of the model using mediapipe and opencv and checking whether the model is giving correct prediction for classes ‘active’ and ‘distracted’ then after realtime testing our model is ready for deployment.



DATASET

The dataset that we are using for our project is generated by detecting landmarks using mediapipe holistic and opencv and storing it in csv file the dataset contains 2004 columns containing coordinates of landmarks and 1950 instances containing classes “active” and “distracted” the dataset is generated using landmarks of multiple persons in order to get diverse dataset for more generalized model.

DATA PREPROCESSING

Handling Missing data - In our dataset we first checked the presence of null values in the dataset. Given our dataframe df we can check for null values by using df.isnull()which will return 1 for true if null value is present in our dataset and 0 for false.. If missing values have been found, there are particularly two ways to resolve this issue: Either Remove the entire row that contains a missing value. However, removing the entire row can generate a possibility of losing some important data. This approach is useful if the dataset is very largeOr Estimate the value by taking the mean, median or mode.
In our dataset no null values were present.

StandardScaling - Standardization is a useful technique to transform attributes with a Gaussian distribution and differing means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.

We can standardize data using scikit-learn with the StandardScaler class.It works well when the features have a normal distribution or when the algorithm being used is not sensitive to the scale of the features.

In our model we have performed standard scaling to normalize the dataset.

PCA - Principal Component Analysis (PCA) is a powerful technique used in data analysis, particularly for reducing the dimensionality of datasets while preserving crucial information. It does this by transforming the original variables into a set of new, uncorrelated variables called principal components. Here’s a breakdown of PCA’s key aspects:
Dimensionality Reduction: PCA helps manage high-dimensional datasets by extracting essential information and discarding less relevant features, simplifying analysis.

Data Exploration and Visualization: It plays a significant role in data exploration and visualization, aiding in uncovering hidden patterns and insights.

Linear Transformation: PCA performs a linear transformation of data, seeking directions of maximum variance.

Feature Selection: Principal components are ranked by the variance they explain, allowing for effective feature selection.

In our project we have performed PCA to reduce the dimension of the dataset from 2004 features to 500 features.

LabelEncoding - Label Encoding is a technique that is used to convert categorical columns into numerical ones so that they can be fitted by machine learning models which only take numerical data. It is an important preprocessing step in a machine-learning project.
We have label encoded our output feature class in case of models like Resnet and ANN.

MODELS

Voting Classifier

A Voting Classifier is an ensemble learning method that combines the predictions of multiple base estimators (machine learning models) and predicts the class label by taking a vote. It's applicable to both classification and regression problems.

Hard Voting:

In hard voting, each base estimator (model) predicts the class label, and the final prediction is determined by a majority vote. The class label with the most votes becomes the final prediction.In the context of classification, the class that receives the most votes is chosen as the final predicted class label.

Soft Voting:

In soft voting, each base estimator predicts the probability distribution over all the classes. The final prediction is determined by averaging these probabilities and then selecting the class with the highest average probability.The final prediction is then determined by averaging these probabilities across all the base estimators and selecting the class with the highest average probability.

In our model we have used hard voting with models comprising Logistic Regression, Ridge Classifier, RandomForest Classifier, Gradient Boosting Classifier, Support Vector machine and Gaussian Naive Bayesian(GaussianNB).

Adaboost

AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances.The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. Usually, AdaBoost is presented for binary classification, although it can be generalized to multiple classes or bounded intervals on the real line.

Linear SupportVector Classifier

Linear SupportVector Classifier is a type of Support Vector Machine (SVM) used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Mulitlayer Perceptron

Multi-layer perception is also known as MLP. It is fully connected dense layers, which transform any input dimension to the desired dimension. A multi-layer perception is a neural network that has multiple layers. To create a neural network we combine neurons together so that the outputs of some neurons are inputs of other neurons.A multi-layer perceptron has one input layer and for each input, there is one neuron(or node), it has one output layer with a single node for each output and it can have any number of hidden layers and each hidden layer can have any number of nodes

Stacking Classifier

Stacking is a way of ensembling classification or regression models. It consists of two-layer estimators. The first layer consists of all the baseline models that are used to predict the outputs on the test datasets. The second layer consists of Meta-Classifier or Regressor which takes all the predictions of baseline models as an input and generates new predictions.They can improve the existing accuracy that is shown by individual models. We can get most of the Stacked models by choosing diverse algorithms in the first layer of architecture as different algorithms capture different trends in training data by combining both of the models can give better and accurate results. In our project we have used RandomForest Classifier, Gradient Boosting Classifier, Support Vector Machine(SVM) and Multilayer Perceptron for the various classification models for the first layer of Stacking architecture .

Resnet

Residual Network (ResNet) is a deep learning model used for computer vision applications. It is a Convolutional Neural Network (CNN) architecture designed to support hundreds or thousands of convolutional layers. Previous CNN architectures were not able to scale to a large number of layers, which resulted in limited performance. However, when adding more layers, researchers faced the “vanishing gradient” problem.ResNet provides an innovative solution to the vanishing gradient problem, known as “skip connections”. ResNet stacks multiple identity mappings (convolutional layers that do nothing at first), skips those layers, and reuses the activations of the previous layer. Skipping speeds up initial training by compressing the network into fewer layers. Then, when the network is retrained, all layers are expanded and the remaining parts of the network—known as the residual parts—are allowed to explore more of the feature space of the input image.Most ResNet models skip two or three layers at a time with nonlinearity and batch normalization in between. More advanced ResNet architectures, known as HighwayNets, can learn “skip weights”, which dynamically determine the number of layers to skip.

ANN

Neural networks (also known as artificial neural networks or neural nets, abbreviated ANN or NN) are a branch of machine learning models inspired by the neuronal organization found in the biological neural networks in animal brains.

An ANN is made of connected units or nodes called *artificial neurons*, which loosely model the neurons in a brain. These are connected by *edges*, which model the synapses in a brain. An artificial neuron receives signals from connected neurons, then processes them and sends a signal to other connected neurons. The "signal" is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs, called the *activation function*. Neurons and edges typically have a *weight* that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection.

Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the *input layer*) to the last layer (the *output layer*), possibly passing through multiple intermediate layers (*hidden layers*). A network is typically called a deep neural network if it has at least 2 hidden layers.

EVALUATION

The performance of each model is evaluated using standard metrics such as accuracy, precision, recall, and F1 score, micro average ,macro average,confusion matrix and Cross-validation techniques are employed to ensure robustness and generalization of the models.

SYSTEM REQUIREMENTS

- Operating System: Windows, macOS, or Linux.
- Processor: Intel Core i5 or AMD Ryzen 5 (or equivalent) processor
- RAM: 8GB or higher
- Storage: At least 50GB of available disk space for storing datasets, models, and libraries
- Graphics Card: Integrated graphics card (Intel HD Graphics or AMD Radeon Graphics) or dedicated GPU (NVIDIA GeForce GTX 1050 or higher) for GPU acceleration (optional but recommended for faster training)
- Software:
- Python 3.x installed
- Required Python packages installed (NumPy, scikit-learn,opencv, TensorFlow, Keras, MediaPipe, etc.)

USAGE

1) Install the required dependencies.
2) Prepare the dataset and ensure it is properly formatted.
3) Run the preprocessing steps, including PCA dimensionality reduction.
4) Train the models using the preprocessed dataset.
5) Evaluate the models using appropriate metrics
6) Realtime testing and hyperparameter tuning.
7) Select the best-performing model for deployment.

FUTURE WORK

1) Fine-tune hyperparameters for improved model performance.
2) Exploring other deep learning and machine learning models.
3) Extending and generalizing the dataset by capturing more data of different individuals corresponding to various actions

PROJECT MEMBERS

SPARSH SINGH(2021UCA1844)
YASH KUMAR(2021UCA1855)