International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Accident Detection and Alert System by Using Deep Learning and Edge Computing

Dr. SHARMILA B[1], POORNIMA S[2], PRIYADHARSHINI S [3] PRIYADHARSHINI SD[4] UMASRI S[5]

[1]Assistant Professor, BE Computer Science and Engineering,
Vivekanandha College of Technology for Women
[2,3,4,5]BE Computer science and Engineering, Vivekanandha College of Technology for Women

---***---

Abstract - The use of artificial intelligence and deep learning models to traffic closed-circuit television (CCTV) systems presents a great deal of promise for the detection and tracking of items on roadways. Nevertheless, it is still difficult to reliably identify traffic accidents and extract useful information from identified items. Early identification of vehicle incidents is essential because it improves incident management systems' response times and reduces the number of injuries brought on by traffic accidents. In order to enable a service that can aid in preventing accidents involving other cars and traffic congestion, this study provides a deep learning-based algorithm intended to detect anomalous road occurrences. This application is a good fit for deep learning techniques, which have shown remarkable performance in computer vision applications involving intricate feature connections. To identify traffic incidents in video footage, the suggested automated approach makes use of deep learning. The premise behind it is that certain visual characteristics that develop over time might be used to identify traffic accidents. Consequently, the model architecture has a temporal pattern recognition component after a visual feature extraction phase, enabling efficient accident detection in real-time situations.

Key Words: Image Preprocessing, Data Augmentation, Transfer Learning, Dense Net.

1.INTRODUCTION

The project's goal is to create a sophisticated picture categorization system by combining transfer learning, data augmentation, and image preparation methods. The main goal is to improve the classification model's effectiveness and performance so that it can manage problems like noise, class imbalance, and poor image quality that are frequently encountered in real-world datasets. Normalizing pixel values and enhancing contrast are two ways that image preprocessing makes the input data fit for training. By artificially expanding the dataset, data augmentation techniques help to mitigate class imbalance and improve the model's ability to generalize by learning from a variety of image changes. The project also uses Dense Net, a potent deep learning architecture renowned for its feature reuse and effective handling of vanishing gradients, to implement a refined transfer learning strategy. With applications in a variety of fields that demand trustworthy image analysis and classification, this system seeks to provide robust and accurate picture identification by utilizing pre-trained models and tailoring them for certain classification tasks.

In order to improve the caliber and precision of picture-based machine learning models, image preprocessing is essential. It entails a number of procedures intended to get raw images ready for additional examination while making sure the learning algorithm can use them effectively. Image resizing and conversion to a common colour format, like RGB from BGR, are the first steps in this procedure. After that, scaling techniques are used to divide the pixel values by 255 in order to normalize the pixel intensity levels, which are usually between 0 and 1. This guarantees consistency throughout the dataset and lowers computing cost. Unsharp Masking (UM) is another technique that sharpens details, especially at the edges, to improve image contrast. In order to highlight features, the UM approach subtracts a blurred version of the image from the original. Applying UM to the Lab colour space's lightness channel enhances the image's overall clarity, which is very helpful when working with different-quality photos. The model can efficiently learn from the images and produce precise predictions thanks to these preprocessing procedures.

The process of applying different alterations to the source photos in order to artificially expand the size of a dataset is known as data augmentation. By giving the model more varied data to train from, this procedure helps to overcome problems like class imbalance and avoids overfitting. Flipping is a common data augmentation approach that adds more viewpoints to the model by flipping photos either vertically or horizontally to produce mirror versions. Another technique is rotation, which introduces diversity in the orientation of objects inside photographs by rotating them at various angles. To make sure the model learns to focus on various locations within the image, translation entails moving the image along the x or y-axis. Furthermore, by adding random noise to the images, noise injection improves generalization by allowing the model to learn from marginally modified data. These additions not only increase the model's resilience to changes in the real world but also boost its capacity to identify characteristics in a variety of scenarios.

II. RELATED WORK

Ji Sang Park [1], Soe Sandi Htun, and others have suggested this system. In recent years, autonomous driving applications have shown a great deal of interest in the difficult task of identifying traffic collision incidents in driving recordings. It is necessary to create techniques for correctly and effectively detecting traffic accidents from a first-person perspective in order to guarantee safe driving alongside human drivers and anticipate their behaviours. The Tempo Learn network is a unique model that uses spatiotemporal learning to detect traffic accidents, as proposed in this research. The suggested method uses a dilation factor to achieve broad receptive fields and temporal convolutions because of their efficacy in detecting irregularities. The two main parts of the Tempo Learn network are accident classification based on the localisation results and accident localisation, which predicts when the accident occurs in a video. We experiment with a traffic accident dashcam video benchmark dataset, the detection of traffic anomaly (DoTA) dataset, which is currently the biggest and most intricate traffic accident dataset, in order to assess the performance of the suggested network. The accident localisation score, as expressed in terms of AUC, is 16.5% higher than that of the current state-of-the-art model, and the suggested network performs exceptionally well on the DoTA dataset. Additionally, we use experiments on the car collision dataset (CCD), another benchmark dataset, to show the efficacy of the TempoLearn network. Tempo Learn is a fully-supervised accident detection model that is based on a segment-level traffic accident detection methodology, as proposed in this research. The transformer classifier, SPG, and temporal context learning are the three main parts of the Tempo Learn network. As a result, the Tempo Learn network can determine the kind of accident that takes place during the detected segment and predict the temporal positions of traffic accidents in a video together with the confidence scores. Before creating the traffic accident segment proposals, the Tempo Learn network may gather features with temporal context information by integrating the broad receptive fields based on temporal convolutions. These capabilities enable Tempo Learn to detect long-tailed, distributed, and heterogeneous traffic accidents better than the state-of-the-art frameworks. Furthermore, we use the transformer as a classifier to categorise various traffic accident types according to the segments, outperforming the current methods in terms of top-1 accuracy and mAP score.

In this method, Yuta Maruyama and Gosuke Ohashi [2] et al. have presented Recent proposals have included accident prediction models that use deep learning algorithms to forecast the likelihood of traffic accidents. High precision and decision basis visualisation are required for the use of these models. When the motion feature of the risk factor is minimal, current models, which rely on the motion features of objects in the surrounding environment, perform poorly. Meanwhile, by using visual attention functions, drivers can prevent accidents. As the foundation for an accident prediction approach, this study focusses on the divergence between visual attention and focus of expansion (FOE), which are significantly associated in typical driving scenarios. When combined with Dynamic-Spatial-Attention, a deep learning-based accident prediction technique, the suggested model can visualise decision basis with high accuracy, even when the motion characteristic of the risk variables is tiny. In this experiment, we divided data from a popular accident dataset, the Dashcam Accident Dataset, into several accident types. Using the Dashcam Accident Dataset, the suggested approach maintains the same accident prediction performance as the baseline Dynamic-Spatial-Attention method in categories where the motion feature of risk factors tends to be large, while achieving higher accident prediction performance in categories where the motion feature of risk factors tends to be small. Furthermore, in order to give a visual explanation of the decision rationale, the suggested method uses visual attention and FOE to visualise the risk variables. This study suggests a DSA-based accident prediction model that uses motion and object data along with the difference between visual attention and FOE to forecast accidents. Using the visual attention model Y. Maruyama, G. Ohashi: Divergence-Based Accident Prediction Model The accident prediction model incorporates knowledge top-down from the driver gaze data between Visual Attention and FOE. We apply the suggested approach to the DAD dataset and compare it with the DSA and DRIVE. By verifying that it is feasible to predict accidents with high accuracy for all accident scenes carrying risk factors with minor motion features, the results demonstrate the usefulness in the metrics F1, TTA, and FT.

In this system, Abdelkader Dairi, Fouzi Harrou, and Ying Sun [3] et al. This study offers a successful data-driven anomaly detection method for detecting drunk driving. In particular, the suggested anomaly detection method combines the Isolation Forest (if) scheme with the desirable features of the t-distributed stochastic neighbour embedding (t-SNE) as a feature extractor to identify drivers' levels of intoxication. In order to achieve good detection, we exploited the t-SNE model's ability to reduce the dimensionality of nonlinear data while maintaining the input data's local and global structures in the feature space. Simultaneously, the iF scheme is a successful unsupervised tree-based method for detecting anomalies in multivariate data. This method is more appealing for identifying intoxicated drivers in real-world situations because it only uses data from typical occurrences to train the detection model. We utilised publicly available data gathered using a digital camera, a temperature sensor, and a gas sensor to confirm the suggested t-SNE-if approach's detection ability in accurately identifying drivers with excessive alcohol. With an AUC of about 95%, the detection system as a whole demonstrated a good detection performance, proving the durability and dependability of the suggested methods. Additionally, the suggested t-SNE-based iF scheme provides better drunk driver status detection performance than the Principal Component Analysis (PCA), Incremental PCA (IPCA), Independent Component Analysis (ICA), Kernel PCA (kPCA), and multi-dimensional scaling (MDS)-based forest, EE, and LOF detection schemes. Reducing traffic accidents and enhancing road safety clearly depend on accurate drunk driving detection. This study introduces a data-driven mechanism for identifying intoxicated drivers. Crucially, this combines the discrimination power of the if in anomaly detection with the increased capability of the t-SNE nonlinear dimensionality reduction as a features extractor to improve drunk driving detection. The properties of the gathered multivariate data are extracted using the t-SNE once the input data has been normalised. In order to identify possible intoxicated driving, the if detector then applies t-SNE features. This method's main benefits are that it does not make any assumptions about the distribution of data and does not require labelled data in order to find anomalies. The efficacy of the detection is evaluated using real-world data gathered by sensors and a digital camera.

According to Xianglun Mo, Chuanpeng Sun [4] et al., this system's ability to detect motorway traffic occurrences at night is essential for boosting rescue efficiency and averting secondary collisions. Most motorways in China have fully functional video monitoring systems. At night, however, the motorway traffic event detection system still relies on manual detection, which is inefficient. The analysis of the nighttime motorway traffic event detection method is the first step in this dissertation. Based on this, this study uses SpyNet deep learning and the Mask technique to build a deep learning network for nighttime highway vehicle recognition. A dense optical flow consisting of night vehicle light flow is the detecting object. Finally, we use the Deep sort method to track and quantify the velocity of the found object. Using the observed data, this study compares the classical optical flow method, YOLOv3, the background difference method, and the proposed method. The results show that the recommended method offers the advantages of high detection accuracy and fast detection speed. advantageous daylighting conditions and a somewhat uncomplicated background. It cannot employ its straightforward and efficient features for vehicle detection at night. The popular deep learning system can identify and categorise cars in nighttime settings by using data set training. However, it cannot identify and detect common nighttime driving conditions, such as opposing lamp interference, which affects the recognition of traffic events and subsequent speed tracking. This work uses the dense optical ow vehicle detection technique based on Mask-SpyNet, which eliminates opposing lamp light interference and improves the accuracy of nocturnal vehicle detection. The Deep Sort algorithm can accurately detect traffic events at night and has a wide range of possible uses in the current low-intelligent highway environment. There are still gaps in detecting technology and hardware facilities, even though this study relies on widely used monitoring equipment on motorways, which has a high detection rate, cheap cost, and easy installation.

In this method, Burcu Kir Sava and Yaar Becerikli [5] et al. Information technology advancements and changes have been crucial to the creation of intelligent car systems in recent years. One significant contributing element to auto accidents is driver weariness. For this reason, experts have been tracking road accidents that involve drivers who are fatigued or irresponsible. This article suggests a Multi-tasking Convulational Neural Network (ConNN*) model to identify driver weariness or drowsiness. The driver behaviour model makes use of the eye and mouth characteristics. Driver weariness is tracked by changes in these attributes. Unlike the research in the literature, the proposed Multi-task ConNN model simultaneously classifies both mouth and eye information into a single model. The calculation of yawning frequency/frequency of mouth (FOM) and eyes closure duration/percentage of eye closure (PERCLOS) helps assess driver weariness. There are three classes based on the driver's level of weariness in this study. Using the YawdDD and NthuDDD datasets, the suggested model detected fatigue with 98.81% accuracy. A comparative presentation of the model's success follows. In order to identify driver weariness in real time, this research uses multi-task convolutional neural network models. To precisely identify the driver's eye and mouth information, the Dlib algorithm is employed. After that, Multi-task ConNN models are used to train the system in order to determine the fatigue parameters. The study maintains a steady frequency range and a fixed number of frames. The final classification of exhaustion is very weary, less fatigued, and not tired, based on fatigue parameters. Additionally, these scenarios undergo dynamic testing and coding at specific times to ensure their continuity. When tested in real time, the system's accuracy performance is incredibly reliable. The suggested system is able to simulate the interaction between the mouth, eye, and sub-states. exhaustion at a specific time is seen as a contributing cause to exhaustion at the current time, and time fluctuates depending on an individual's behaviour. The system functions properly. One of the study's most potent aspects is that it uses a single model instead of building two distinct ConNN models with two different architectures, making it a faster and more potent system.

The advent of edge computing represents a major achievement in the rapidly evolving field of technology. This invention has transformed a number of fields, and accident detection is one of its most important uses. By facilitating the real-time processing and analysis of sensory data from onboard sensors, cameras, and other connected devices, edge computing can improve road safety and emergency response. The inventors of the current technology can get beyond the drawbacks of traditional centralized cloud-based approaches and build a safer transportation network by incorporating edge computing into accident detection systems. The authors of this study describe an accident detection framework that uses Deep Learning (DL) in an edge cloud environment. The authors of the current system have employed a DL model based on Convolutional Neural Networks (CNNs) for accident detection. The edge node close to the data source is where the DL model finds the accident. When compared to solely cloud-based deployment, the current architecture offers low latency, minimum network utilization, and shorter execution times. Furthermore, in the cloud-edge environment, the current accident detection model has an accuracy of up to 95.91% with precision 0.9574, recall 0.9574, and F1 score 0.9574. Investigating how edge cloud computing might transform accident detection systems is the focus of this study. The goal of this project was to create a rapid accident-detection system by combining edge and cloud computing. The study's findings demonstrate that the edgeenabled framework improves execution time, network utilization, and latency compared to cloud deployment. According to the findings from this study, edge-based applications require less bandwidth to send data to the edge node and finish tasks faster. With the use of the current DL model, the study advances our knowledge of edge-based accident detection performance. The DL model in this study uses three CNN layers, max-pooling, and dense layers to determine whether or not the photos are accidental. In the cloud-edge context, the current accident detection model has an accuracy of up to 95.91% with precision 0.9574, recall 0.9574, and F1 score 0.9574. Its simulation-based study was its primary flaw. Nevertheless, the creators of the current system employed a well-known paper by S. Banerjee et al.: DL-Based Car Accident Detection Framework Using Edge and Cloud Computing.

III. PROPOSED SYSTEM

In order to detect traffic accidents in real time, the suggested method combines deep learning with Internet of Things technologies. An IoT kit is made to gather real-time accident data and send it to the cloud for analysis because deep learning requires a lot of data. The method uses pre-trained deep learning models that are adjusted for particular tasks using smaller datasets because balanced and labelled datasets are hard to come by. The process of preparing photographs involves scaling them to 224×224 pixels and converting them to JPG format in order to guarantee consistency. Resize, crop, rotate, flip, and colour jitter are examples of data augmentation techniques used to improve generalization and rectify dataset imbalance. Additionally, images undergo normalization and greyscale conversion. Feature extraction and fine-tuning are two methods of transfer learning that capitalize on the advantages of previously trained models to increase classification accuracy. To maximize learning and minimize overfitting, the model design takes into account important factors including depth, width, kernel size, skip connections, and channel selection. By modifying factors including optimizer type, learning rate, activation function, batch size, and number of epochs, Bayesian optimization further improves model performance through hyperparameter adjustment. This all-inclusive system effectively integrates preprocessing, augmentation, IoT-based data collecting, and optimized transfer learning to provide reliable and accurate traffic accident detection.

A. Image Preprocessing

Before supplying input data to a learning algorithm, image preparation is essential for enhancing its quality. In order to improve training and classification, preprocessing was done in this module to improve image quality and normalize intensity values. Before being turned into NumPy arrays, the photographs were first scaled and translated from BGR to RGB format. All image data was scaled to the [0,1] range by dividing by 255 in order to standardize the pixel intensity values and lower computing cost. By subtracting a fuzzy version of the image from the original, the Unsharp Masking (UM) approach sharpens the images and improves contrast and highlights edges. Sharpening the brightness channel, switching RGB photos to Lab colour space, and then converting back to RGB allowed for more refinement. In order to balance clarity without adding noise, the sharpening process was controlled by three parameters: radius, amount, and threshold, which were set at [2, 2, 0.1]. All things considered, this module makes sure the photos are crisp, clean, and formatted consistently for the best feature extraction.

B. Data Augmentation Techniques

This module used data augmentation strategies to improve model generalization and address class imbalance. Through the creation of altered replicas of the original photos, these changes artificially expand the training dataset. This improved spatial robustness by simulating various viewing angles through horizontal and vertical flipping. The model was able to learn rotational invariance by applying rotation in a range of 0 to 360 degrees. To avoid spatial bias and guarantee that the model learns features throughout the full image, translation—or moving images along the x and y axes—was carried out. Additionally, to create a variety of image instances, Gaussian noise was introduced, which made the model extract reliable characteristics rather than learning particular patterns. By eliminating overfitting and offering a more varied and balanced dataset, these augmentation techniques help the neural network function better under a range of real-world circumstances.

C. Fine-Tuned Transfer Learning Network

With the help of pre-trained knowledge, this module's refined Dense Net-based transfer learning technique enhances classification performance. Because it solves the vanishing gradient problem efficiently and encourages feature reuse by connecting all layers directly, Dense Net (Densely Connected Convolutional Networks) is the preferred method. The convolutional layers of a pre-trained Dense Net model—which was first trained on ImageNet—are frozen to preserve the previously acquired features. For the current binary classification challenge, only the last classification layer has been trained and customized. The model can now focus on fruit categorization while maintaining its broad feature extraction capabilities thanks to the replacement of the previous output layer with the new fully linked layer. The development rate (number of new feature maps per layer), transition layers, and dense blocks provide a deep, compact architecture that can capture intricate patterns. Incorporating batch normalization, ReLU activation, and 3×3 convolution into every layer improves model performance even further. By utilizing the power of deep representations acquired from extensive image datasets, this fine-tuning technique enables effective training on smaller datasets.

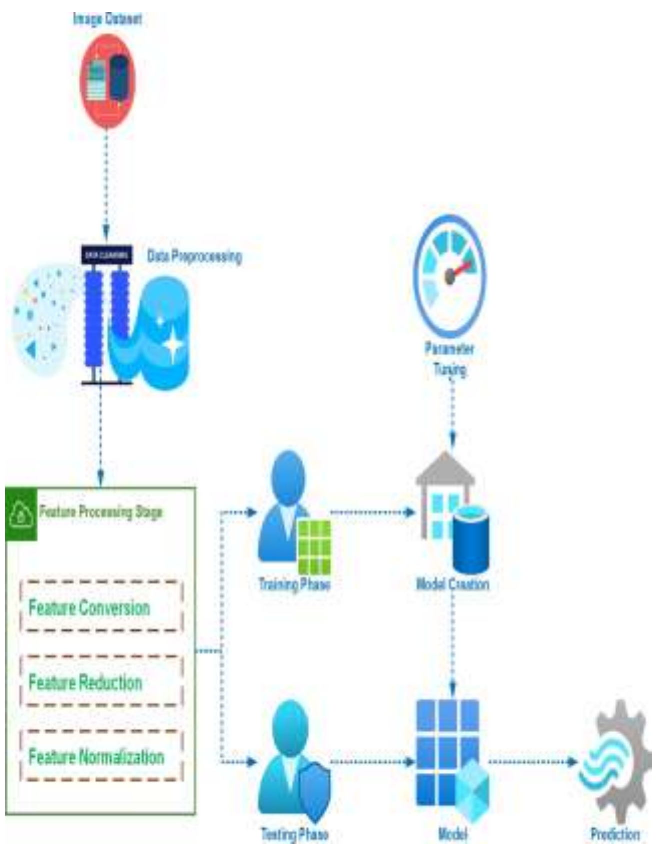

Image Dataset:

The classification system is based on the picture dataset, which serves as the machine learning model's raw input. A wide variety of photos depicting various fruit groups make up this dataset. To guarantee diversity and improve model resilience, the photos are taken in a range of lighting conditions and perspectives. To make sure the model can effectively generalize to new, unseen data, the dataset is usually divided into training, validation, and testing sets. By assigning a class to each image, the model is able to establish connections between visual characteristics and their labels during the training phase.

Data Pre-processing:

One of the most important steps in getting the images ready for the learning algorithm is data pre-processing. This step uses a number of methods to enhance the photographs' quality and fit the model. The resizing of images to a consistent size guarantees consistency throughout the dataset. By normalizing the pixel values between 0 and 1, the model may train more effectively and with less computing complexity. Additionally, to improve image clarity and eliminate extraneous information, noise reduction techniques like Gaussian blurring or sharpening are used. By highlighting only the images' important elements, this phase makes the images better suited for model training.

Feature Processing Stage:

The goal of the feature processing stage is to extract pertinent information from the previously processed images. In order to help the model concentrate on the most crucial aspects of the image, this step may involve feature extraction methods like edge detection, texture analysis, or colour histograms. To make the images compatible with machine learning algorithms, they are usually converted into numerical feature representations, like matrices or vectors. To ensure that only the most discriminative characteristics are given into the learning model, these features are utilized to capture patterns that can aid the model in differentiating between various fruit classes.

Parameter Tuning:

The practice of modifying the machine learning model's hyperparameters to maximize performance is known as parameter tuning. The learning rate, batch size, number of layers, activation functions, and other hyperparameters can all have a big impact on how well the model learns and generalizes. This step involves experimenting with alternative hyperparameter combinations using a variety of techniques, such as grid search and random search. Finding the ideal combination of values that produce the best results on the validation set is the aim. Optimizing the model's parameters guarantees that it can effectively learn from the data and get the maximum level of accuracy.

Training Phase:

The machine learning model learns to map inputs (pictures) to their appropriate labels (fruit categories) during the training phase. The model is trained using an appropriate learning technique, such as a convolutional neural network (CNN) or deep neural network, utilizing the pre-processed images and extracted features. Based on the discrepancy between the true labels and the anticipated outputs, the model modifies its internal weights and biases throughout training. Backpropagation and optimization methods like gradient descent are used to accomplish this. In order to reduce the loss function and enhance its capacity to accurately predict the label for every image, the model repeatedly runs through the training dataset (epochs).

Model Creation:

The process of creating a model entail putting the neural network's layers together and choosing an architecture that is suitable for the given issue. Because convolutional neural networks (CNNs) can learn hierarchical features from photos, they are frequently utilized for fruit classification. Multiple convolutional layers for feature extraction, pooling layers for down sampling, and fully linked layers for classification are possible components of the architecture. To take advantage of prior knowledge from related tasks, advanced models may use strategies like transfer learning or fine-tuning pre-trained models (e.g., Dense Net, ResNet). Determining the optimization technique and loss function that will direct the training process is another aspect of this step.

Testing Phase:

The testing phase involves assessing the trained model's performance on a different test set that it has never encountered before. In this stage, the model generates predictions based on the test data, and the accuracy of the model is evaluate by comparing the predictions with the true labels. Measures of the model's performance include confusion matrix, precision, recall, and F1 score. This stage is essential for figuring out whether the model is overfitting to the training data or has generalized well to new data. The testing phase's outcomes shed light on how well the model will function in practical settings.

Model Prediction:

The term "model prediction" describes the system's ultimate output following training and testing of the model. During this stage, the model uses the patterns it has learnt during training to identify fresh, unseen images. Upon receiving a fresh image, the model uses its network to process it and generate a projected class label. Whichever fruit group the image most likely depicts, the anticipated label matches. The system's primary role is the prediction stage, which allows it to classify data in real time and produce valuable results for tasks like inventory management, sorting, and fruit identification.

IV. RESULT AND ANALYSIS

A deep learning model's performance is assessed using a number of metrics, including accuracy, precision, recall, and F1-score. Although it does not differentiate between different kinds of errors, accuracy is the total number of accurate predictions. Since precision indicates the proportion of projected positive cases that were actually positive, it is especially crucial when minimizing false positives. Conversely, recall highlights the model's capacity to catch positive occurrences by quantifying the proportion of real positive cases that were accurately detected. The F1-score is particularly helpful when working with imbalanced datasets since it strikes a balance between precision and recall. When there are several classes in the dataset, these metrics are calculated for each class, and the weighted average or macro approaches can be used to average the results. The weighted average considers the number of instances in each class, whereas the macro average assigns each class equal weight. This makes it possible to guarantee that the performance evaluation takes into account both the effectiveness of the model as a whole and the performance of each individual class. These assessments highlight the model's advantages and disadvantages, especially when it comes to pinpointing areas that might require enhancements, including correcting class imbalance or enhancing precision and recall for particular classes. By concentrating on these measures, the model can be improved for more generalization and robustness in practical applications, for example, by adopting methods like data augmentation or fine-tuning transfer learning models.

Precision

Precision=TP+FP/TP

TP (True Positives): Correctly predicted positive instances

FP (False Positives): Incorrectly predicted positive instances

Recall

Recall=TP+FN/TP

- **FN** (False Negatives): Actual positives that were incorrectly predicted as negative

Recall measures the proportion of actual positives that were correctly identified.

F1-Score

F1-Score=2×Precision+Recall/ Precision× Recall

F1-Score is the harmonic mean of precision and recall, providing a balance between the two.

Accuracy

Accuracy=TP+TN+FP+FN/TP+TN

- **TN** (True Negatives): Correctly predicted negative instances

Accuracy measures the overall correctness of the model.

ALGORITHM	ACCURACY
CNN	80
Fine-tuned transfer learning	88

Figure 1 comparison table

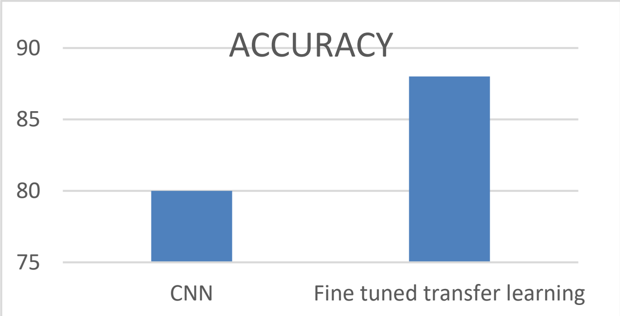

Figure 2 comparison graph

V. CONCLUSION

In summary, with a strong foundation based on methods like picture preprocessing, data augmentation, and fine-tuned transfer learning, the suggested deep learning model shows great promise in tackling the problem at hand. Accuracy, precision, recall, and F1-score are among the evaluation measures that show the model's strengths and possible weaknesses. The model achieves better feature representation and efficient parameter utilization by utilizing the advantages of transfer learning with architectures such as Xception and depth wise separable convolutions. There is still opportunity for improvement, though, especially in resolving class disparities and adjusting the model to further improve recall and accuracy for under-represented classes. All things considered, the model has promise for practical uses and, with additional fine-tuning and improvement, could produce reliable, high-performing outcomes on a variety of datasets.

VI. FUTURE WORK

To further improve the model's performance and flexibility, there are a number of directions to pursue in subsequent research. To alleviate class imbalances and increase the model's generalization over a range of real-world settings, one possible enhancement is the incorporation of more sophisticated data augmentation approaches. Improved accuracy and robustness may also result from experimenting with more intricate designs, such as integrating ensemble approaches or merging Xception with other models. Enhancing the training dataset, adding more varied data sources, and adjusting hyperparameters would all aid in improving the model and lowering overfitting. Additionally, putting the model into production settings and adding real-time inference capabilities

may present new difficulties that call for more speed and efficiency optimization. In the end, improving the model's interpretability and explainability would also be essential for learning more about how it makes decisions and fostering confidence in its use.

VII. REFERENCES

1. Waleed S. Alnumay, Moumita Roy, ManaSh Kumar Mondal, Sourav Banerjee, and Uttar BiSwaS An Edge and Cloud Computing-Based Deep Learning-Based Car Accident Detection Framework IEEE AcceSS, 2024

2. An Automatic Car Accident Detection Method Based on Cooperative Vehicle InfraStructure Systems by Daxin Tian, Chuang Zhang, Xuting Duan, and Xixian Wang IEEE AcceSS, 2019

3. Ke-Yu Su, Liang-Bi Chen, and Wann-Jung Chang DeepCraSh is an Internet of Vehicles system based on deep learning that detects head-on and single-vehicle accidents and provides emergency notifications. AcceSS, IEEE, 2019

4. TempoLearn Network: Using Spatio-Temporal Learning for Traffic Accident Detection by Soe Sandi Htun, Ji Sang Park, Kang-Woo Lee, and Ji-Hyeong Han IEEE AcceSS, 2023

5. Using the Difference Between Visual Attention and Focus of Expansion in Vehicle-Mounted Camera Images, Yuta Maruyama and GoSuke OhaShi Accident Prediction Model IEEE AcceSS, 2023

6. SenSorS: A Multi-Learning-Based Anomaly Detector for Effective Driver Drunk Detection by Abdelkader Dairi, Fouzi Harrou, and Ying Sun IEEE AcceSS, 2022

7. Real-Time Driver Fatigue Detection System Based on Multi-TaSk ConNN IEEE AcceSS, 2020 by Burcu Kir SavaS and YaSar Becerikli

8. DharmeSh J. Shah and Sachin UmeSh Sharma A Useful Computer ViSion Technique for Animal Detection and Collision Avoidance IEEE AcceSS (2016)

9. The Temporal ImpulSe of Traffic Accidents in South Korea by Hyunkyung Shin and Jaeho Lee IEEE AcceSS, 2020

10. Using Deep Audio Representation and a BLSTM Network for Road Audio Surveillance Sound Detection Yanxiong Li, Xianku Li, Yuhan Zhang, Mingle Liu, and Wucheng Wang Anomalou IEEE AcceSS, 2018