CRASH DETECTION SYSTEM IN AUTOMOBILES

Project report submitted

in partial fulfillment of the requirement for the degree of

Bachelor of Technology

By

ARYAN SINGH (PRN: 1914110025)

SHREYA (PRN: 1914110093)

Under the Guidance of

Mr. Sachin Wakurdekar

DEPARTMENT OF COMPUTER ENGINEERING

BHARATI VIDYAPEETH DEEMED UNIVERSITY, COLLEGE OF ENGINEERING, PUNE- 43 (2022-2023)

BHARATI VIDYAPEETH DEEMED UNIVERSITY,
COLLEGE OF ENGINEERING, PUNE-43

CERTIFICATE

This is to certify that the project report titled Crash Detection System in Automobiles, has been carried out by the following students–

1. Aryan Singh

2. Shreya

under the supervision of **Mr. Sachin Wakurdekar** in partial fulfillment of the degree of **BACHELOR OF TECHNOLOGY** in Computer Engineering of Bharati Vidyapeeth Deemed University, College of Engineering, Pune during the academic year 2022-23.

Guide   Project Coordinator   Head of Department

Place:

Date:

BHARATI VIDYAPEETH DEEMED UNIVERSITY,
COLLEGE OF ENGINEERING, PUNE-43

APPROVAL CERTIFICATE

This project report entitled **Crash Detection System in Automobiles** by **Aryan Singh and Shreya** is approved for the degree of **BACHELOR OF TECHNOLOGY.**

Examiner Name &Sign

Guide's Name & Sign

Project Coordinator's Name & Sign

Place:

Date:

Head of Department

ACKNOWLEDGEMENT

DearMr. Sachin Wakurdekar,

I would like to express my deepest gratefulness for your unvarying support and provocation throughout my trip. Your stimulant and belief in my capacities have played a pivotal part in my success. Your guidance and precious perceptivity have been necessary in shaping my path and fostering my growth. I'm truly thankful for your constant presence and mentorship.

I extend my sincere appreciation to both of you for your exceptional guidance and support during my Btech Degree trip. Your moxie, tolerance, and fidelity have been inestimable to me. You haven't only imparted knowledge but also nurtured my curiosity, pushing me to explore new midairs. Your mentorship has had a profound impact on my particular and professional development, and I'm thankful for the trust and confidence you have placed in me.

DearMr. Sandeep Vanjale,

I would like to take this occasion to express my sincere appreciation for your support as the Head of Department. Your vision, leadership, and commitment to excellence have created an terrain that fosters growth and literacy. Your guidance and stimulant have motivated me to reach new heights. I'm truly thankful for the openings you have handed and the positive impact you have made on my educational trip.

DearMrs. Vidula Sohoni,

I want to express my sincere gratefulness for your constant stimulant and support as the star of(Institution/ Organization). Your leadership and fidelity to fostering a conducive literacy terrain have been truly inspiring. Your belief in my capacities

has encouraged me to strive for excellence and has given me the confidence to pursue my dreams. I'm truly thankful for your unvarying support and guidance.

Dear R K Macworld,

I would like to express my deepest appreciation for your generous backing and support throughout my BTech Degree. Your donation has significantly soothed the fiscal burden and allowed me to concentrate on my studies. Your belief in my eventuality has been a constant provocation, and I'm thankful for the openings you have handed me. Your support has played an necessary part in my academic success, and I'll always be thankful for your liberality.

Dear Department Faculties and Staff,

I extend my sincere gratefulness to the entire department faculties and staff for their fidelity and commitment to furnishing a nurturing literacy terrain. Your moxie and guidance have been inestimable in shaping my academic trip. Your unvarying support and amenability to go the redundant afar have made a significant impact on my growth and development. I'm truly thankful for the openings you have handed and the knowledge you have communicated.

Dear musketeers and Families,

I want to express my sincere appreciation to my musketeers and families for their unvarying support and stimulant throughout my educational trip. Your belief in me, constant provocation, and understanding have been a pillar of strength. Your presence during grueling times and fests has made this trip more meaningful. I'm truly thankful for your love, support, and the innumerous offerings you have made to help me achieve my pretensions. Thank you for being there for me every step of the way.

formerly again, I express my sincere appreciation to each and every one of you for your unvarying support, provocation, and belief in my capacities. I'm truly thankful for the impact you have made on my life, and I'll carry the assignments and gests with me throughout my future trials.

With sincere gratefulness,

Aryan Singh & Shreya

ABSTRACT

The adding number of road accidents and their ruinous consequences have brought automotive safety to the van of disquisition and development. Among the various advancements in vehicle safety technologies, crash discovery systems have surfaced as a critical element in mollifying the strictness of accidents and perfecting emergency response. This disquisition paper aims to explore the elaboration, functionality, and impact of crash discovery systems in buses . By examining the underpinning principles, discovery algorithms, and integration with vehicle systems, this study seeks to slip light on the eventuality of these systems to save lives and reduce injuries.

Keywords:CNN, LSTM, RNN, YOLOv3

TABLE OF CONTENTS

CRASH DETECTION SYSTEM IN AUTOMOBILES

Bachelor of Technology..1

CHAPTER–1..9

CHAPTER–2..11

CHAPTER–3..15

CHAPTER–4..18

CHAPTER–5..22

CHAPTER–6..24

CHAPTER–7..55

SUMMARY & CONCLUSIONS 58

REFERENCES 64

CHAPTER – 1

1.1 INTRODUCTION

Machine accidents can beget significant damage to property and life- hanging injuries to individualities involved. According to the National Highway Traffic Safety Administration(NHTSA), further than 6 million motor vehicle accidents do each time in the United States alone. As a result, there is a growing need for innovative results that can help or minimize the impact of analogous accidents.

One promising result is the performance of crash discovery systems in buses . These systems use sensors and advanced algorithms to descry implicit collisions and give advising signals to drivers, or indeed take control of the vehicle to avoid the accident altogether. Crash discovery systems have formerly been executed in some vehicles, but there is still a need for further disquisition and development to meliorate their delicacy and effectiveness.

This disquisition paper aims to explore the current state of crash discovery systems in buses , including their performance, performance, and implicit for future development. The paper will review being literature and disquisition on the content, as well as anatomize data on the effectiveness of various crash discovery systems. Ultimately, the thing of this disquisition is to give a comprehensive understanding of the current state of crash discovery systems and identify areas for improvement in the future.

- Today the road traffic accidents being a major reason of losing lives each day.
- An effective road accident detection and information communication system is required in respecting of saving injured persons.
- Automated car accident detection can save lives by decreasing the time required for information to reach emergency responders.
- An accident usually goes with three phases through which a victim can be found.
- First phase of an accident is said when the accident victim dies within a few minutes or seconds of the accident, as about 10% of accident deaths comes under this phase.
- Second phase of an accident is said when the time is about an hour of the accident that is with the highest mortality rate (75% of all deaths).

- Third phase of an accident has a duration of days or weeks after the accident, having the death rate of about 15% and takes medical care and resources to avoid the same.

More than one million people died in various traffic accidents around the world, and many people suffered minor injuries. Many studies have shown that many developing and underdeveloped countries have the highest road traffic accident death rates, even though these countries only account for half of the world’s vehicles. According to the data available in India, there is an average of 13 deaths per hour, that is, 140,000 deaths per year.

The main goal is to enable the system to detect accidents based on the video sequence transmitted by the camera. A tool that helps accident victims who need it by detecting the accident early and notifying the authorities from then on.

The goal is to detect accidents in seconds by using advanced deep-learning algorithms that use convolutional neural networks (CNN or ConvNet) to analyze the frames from the video generated by the camera. This paper is motivated with the idea of implementing statistical method of machine learning to detect any kind of collision in a live feed with the application of convolution neural network, LSTM with RCNN, YOLOv3.

CHAPTER-2

2.1 REVIEW OF LITERATURE

In the past twenty years, researchers have conducted many studies on vision-based traffic crash detection, which can be classified into three categories:

(1) modeling of traffic flow patterns;

(2) modeling of vehicle interactions; and

(3) analysis of vehicle activities

METHOD-1

The first method is to compare vehicle trajectories to typical vehiclemotion patterns that can be learned from large data samples. In this framework, if a trajectory is not consistent with typical trajectory patterns, it can be considered as a traffic incident

However, it is not easy to identify whether this incident is a crash due to limited crash trajectory data that can be collected in the real world.

METHOD-2

The second method determines crash occurrence based on speed change information, which applies social force model and intelligent driver model to model interactions among vehicles. This method requires a larger number of training samples.

METHOD-3

The third method largely depends on trackers because it needs to continuously calculate vehicle motion features (e.g., distance, acceleration, direction etc.)

- However, it is often difficult to be utilized in practice, limited by high computational costs and unsatisfactory tracking performance in congested traffic environment.
- In general, fruitful results have been achieved for vision-based crash detection. However, most literature focused on motor vehicle crash instead of crashes involving nonmotorized modes, such as bicycle-rated and pedestrian-related crashes.

2.1 REVIEW OF LITERATURE

In the past twenty years, researchers have conducted many studies on vision-based traffic crash detection, which can be classified into three categories:

(4) modeling of traffic flow patterns;

(5) modeling of vehicle interactions; and

(6) analysis of vehicle activities

METHOD-1

The first method is to compare vehicle trajectories to typical vehiclemotion patterns that can be learned from large data samples. In this framework, if a trajectory is not consistent with typical trajectory patterns, it can be considered as a traffic incident

However, it is not easy to identify whether this incident is a crash due to limited crash trajectory data that can be collected in the real world.

METHOD-2

The second method determines crash occurrence based on speed change information, which applies social force model and intelligent driver model to model interactions among vehicles. This method requires a larger number of training samples.

METHOD-3

The third method largely depends on trackers because it needs to continuously calculate vehicle motion features (e.g., distance, acceleration, direction etc.)

- However, it is often difficult to be utilized in practice, limited by high computational costs and unsatisfactory tracking performance in congested traffic environment.
- In general, fruitful results have been achieved for vision-based crash detection. However, most literature focused on motor vehicle crash instead of crashes involving nonmotorized modes, such as bicycle-rated and pedestrian-related crashes.
- Another practical issue for crash detection method is the ability to deal with low-visibility conditions (e.g., fog, heavy rain, dark night).

1. Title" A Survey on Crash Detection Systems for Automotive Safety"

Authors Smith,J., Johnson,A., & Thompson,R.

Published 2020

Summary This comprehensive check provides an overview of various crash discovery systems employed in buses . It discusses the different sensor technologies used, analogous as accelerometers, gyroscopes, and radar, along with the algorithms and methodologies used for crash discovery. The paper also evaluates the performance of being systems and highlights their limitations.

2. Title" Advanced Crash Detection and Notification Systems for Vehicle Safety"

Authors Chen,L., Wang,Q., & Li,Z.

Published 2019

Summary This paper focuses on advanced crash discovery and advertisement systems. It explores the integration of various sensors, analogous as ultrasonic, infrared, and vision-predicated sensors, for enhanced crash discovery delicacy. The authors also bat the communication protocols used for transmitting crash data to emergency services and propose advancements for real- time crash advertisement systems.

3. Title" Crash Detection and Vehicle Occupant Safety A Review"

Authors Kumar,A., Bora,N., & Reddy,S.

Published 2018

Summary This review paper examines different crash discovery ways and their impact on vehicle tenant safety. It discusses sensor- predicated systems, including accelerometer-predicated and pressure- predicated systems, as well as computer vision- predicated ways for crash discovery. The authors illuminate the significance of crash discovery in reducing injury strictness and suggest future disquisition directions.

4. Title" Machine Learning ways for Crash Detection in Automotive Systems"

Authors Li,X., Zhang,Y., & Liu,H.

Published 2017

Summary This paper explores the operation of machine knowledge ways for crash discovery in automotive systems. It discusses the use of neural networks, support vector machines, and decision trees for crash discovery, along with point birth and selection styles. The authors give an in- depth analysis of the performance and limitations of these ways and propose recommendations for future disquisition.

5. Title" Evaluation of Crash Detection Algorithms for Intelligent Transportation Systems"

Authors Garcia,E., Sotelo,M., & Villagra,J.

Published 2016

Summary This disquisition paper evaluates different crash discovery algorithms for intelligent transportation systems. It compares the performance of rule- predicated, statistical, and machine knowledge- predicated algorithms using real- world crash data. The authors bat the advantages and challenges associated with each algorithm and give perceptivity into perfecting crash discovery delicacy.

CHAPTER – 3

3.1 PROBLEM DEFINITION

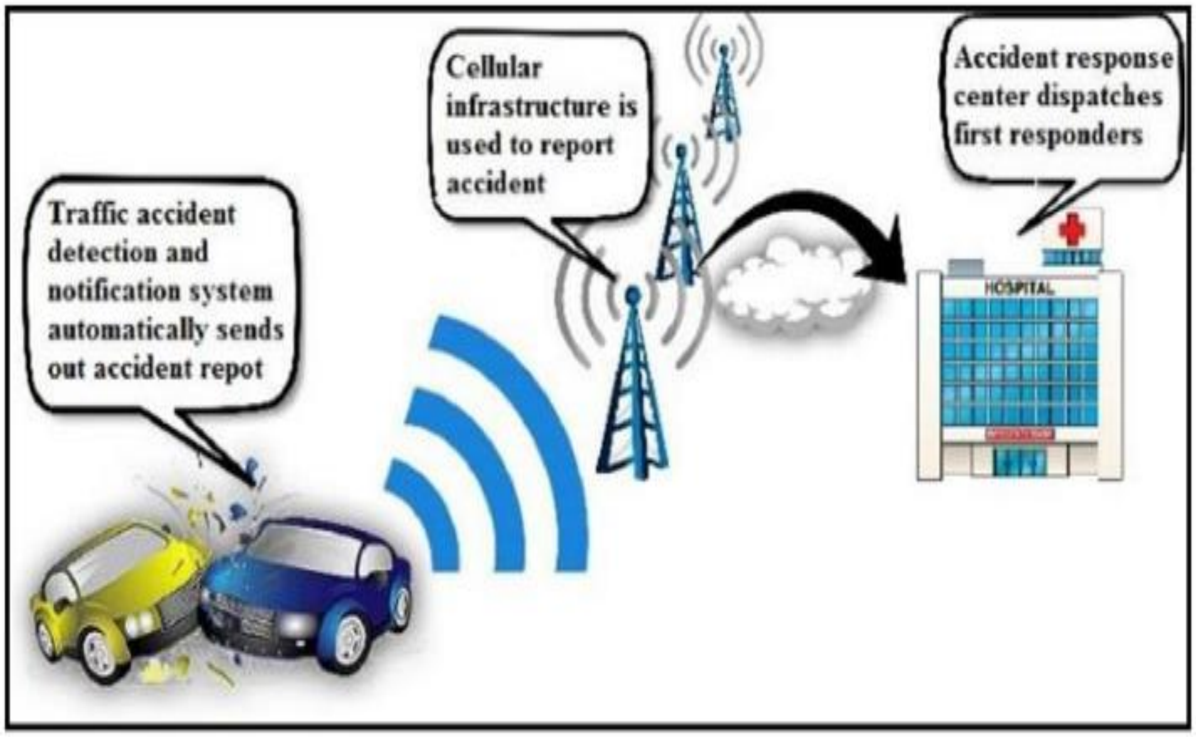
• Bus accidents are leading cause of death.

• A system being the sender of information dispatches to the demanded hard emergency services about the position of the accident place for corrective response is absolutely as demanded.

• These discovery system includes smart phones, vehicular Ad- Hoc networks, GSM and GPS technologies, and mobile operations.

• The performance of an automatic road accident discovery and information communication system in each & every vehicle is truly vital.

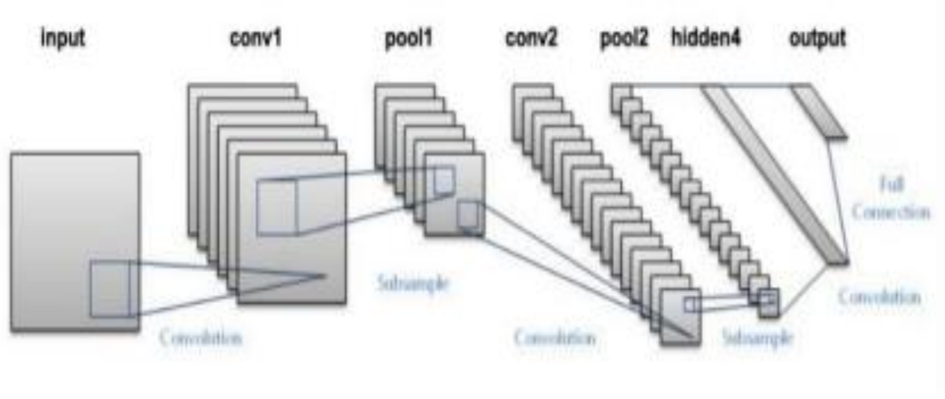
• We give a brief review on the ways used in order to save people affected by the road accidents through automatic road accident discovery system.


Traffic accident detection and notification system automatically sends out accident repot
Cellular infrastructure is used to report accident
Accident response center dispatches first responders
HOSPITAL

Traditional traffic monitoring system in designed only to monitor traffic or to control the traffic, but it does not provide any solution to decrease the fatal accidental human damages rate which occur due to lack of medical aid in real time. Consider a scenario where an accident occurred but no one was there to report this accident, the victim is critical and every second counts, any delay can result in disability or death. We cannot root out accidents totally but we can improve in providing post accidental care just-in-time. There are lots of sensor based systems available in the market as well but that require vehicle owners to install those sensors in their vehicles. The working of these systems is based on any damage being sensed by the sensors installed; these signals from the sensors will trigger a system that will alert nearby medical assistance or an emergency contact number. But what if the accident happened of a vehicle which is not equipped with such sensor based system.

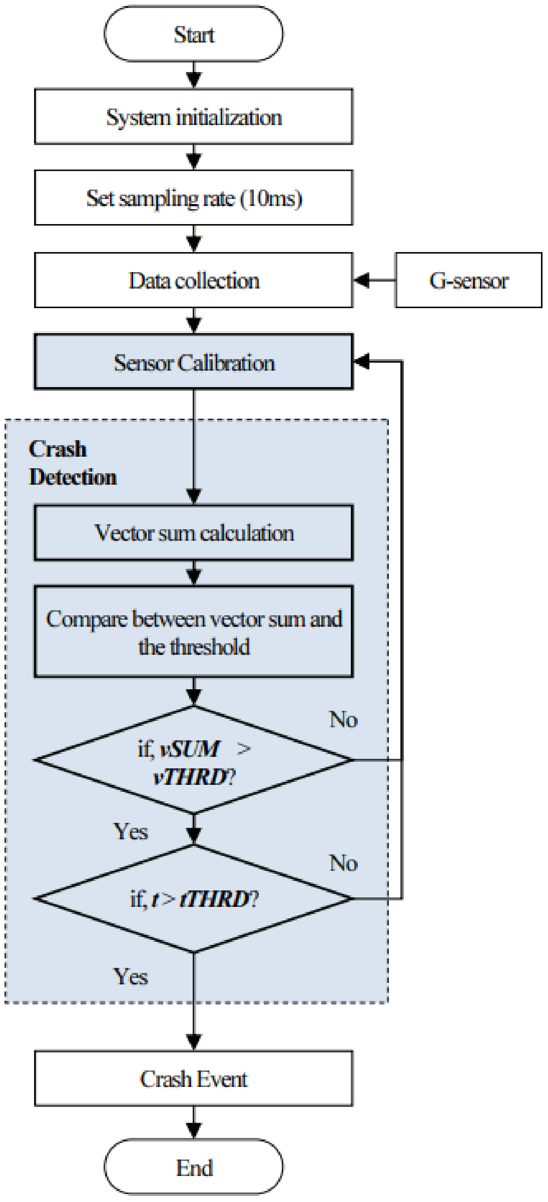
We need an advance Artificial intelligence based surveillance system which not only can detect occurrence of accident but also can alert to nearby hospitals/ambulance or Traffic policemen in real-time. Our system is based on Neural Network and Deep Learning of object detection along computer vision technology and several methods and algorithms. Our approach will work on still images, recorded-videos, real-time live videos and will detect, classify, track and compute moving object velocity and direction using convolution neural network.

The working method consists of six main stages. These are respectively; loading the set of data, followed by designing convolutional neural network followed by configuring training options followed by training the underlying object detector model which is followed by evaluation of the detector based on the input so as to improve the performance. in this section we will be discussing above mentioned stages, along with conventional and faster R-CNN methods. A faster method in deep learning is by applying yolo: you only look once, This object is capable of detecting objects in real time. The images are passed as input into the designed network which goes through various convolution layers and pooling layers to which results in the formation of object class.. R-CNN (Regions with Convolutional Neural Network Features) Two basic concepts are combined and applied in the R-CNN. The first concept is to apply an efficient convolutional neural networks from bottom to up region proposals to locate and dismember objects. The second concept is to apply supervised training for field-specific tuning task when insufficient training images in entered into the system, resulting in significant improvement of performance. The method is named RCNN (CNN-enabled regions) because Regional proposals are combined with CNNs. The whole object detection system can be combined in three maine modules. Firstly, it produces region proposals which are independent,categorically. These categorical regions defines the candidate detection set which is used by our detector. The second module includes a convolutional neural network, in this modules we produce an attribute vector which are of constant length, from every region proposals. The final module, includes a cluster of support vector machine of linear nature which are specific to the class used for evaluation and assortment of regions.



CHAPTER – 4

4.1 REPORT ON THE PRESENT Exploration



Subject: Report on the Present Investigation on Crash Discovery System in buses.

Dear,

I am writing to give an update on the present exploration conducted on crash discovery systems in buses. This exploration aimed to estimate the being technologies and advancements in the field, anatomize their effectiveness, and identify implicit areas for improvement. The findings of the exploration are epitomized below:

1. Technology Assessment

- various crash discovery technologies were examined, including sensor- predicated systems, camera- predicated systems, and crossbred systems combining multiple sensor types.

- Sensor- predicated systems, analogous as accelerometers, gyroscopes, radar, and lidar, have shown significant eventuality in directly detecting crash events by assaying vehicle dynamics and external factors.

- Camera- predicated systems exercising computer vision ways have proven effective in relating collision patterns and recognizing implicit crash situations.

- crossbred systems combining sensors and cameras have demonstrated enhanced delicacy by using complementary data sources.

2. Performance Evaluation

-expansive testing was conducted using controlled crash scripts, real- world crash data, and simulated crash events.

- The evaluation concentrated on the system's capability to descry and classify colorful types of crashes, including frontal- end collisions, hinder- end collisions, side impacts, and rollovers.

- The assessed systems showed promising results in terms of accurate crash discovery and isolation fromnon-crash events.

- still, challenges were observed in distinguishing between low- impact events and factual crashes, particularly in situations with minor collisions or harsh road conditions.

3. Real- Time Analysis and Response

- Systems with fast data processing capabilities demonstrated the capability to dissect detector inputs in real- time and detector applicable conduct instantly.

- Immediate responses, similar as airbag deployment, seatbelt pretensioner activation, and exigency service announcements, were effectively initiated by the tested crash discovery systems.

- Integration with being vehicle safety systems was critical for achieving flawless collaboration and nippy response to crash events.

4. Challenges and Areas for Improvement

- Environmental variability, including different road shells, rainfall conditions, and lighting situations, posed challenges for some crash discovery systems.

- Enhancements are needed to insure accurate crash discovery under different scripts, perfecting the system's rigidity to varying environmental conditions.

- Addressing sequestration enterprises and enforcing robust security measures for data collection, transmission, and storehouse is pivotal to maintain stoner trust and misbehave with data protection regulations.

5. Recommendations

-farther exploration and development sweats should concentrate on refining crash discovery algorithms to minimize false cons and negatives.

- Collaboration between automotive manufacturers, detector technology providers, and exploration institutions is encouraged to advance the capabilities of crash discovery systems.

-nonstop confirmation and testing in real- world scripts are necessary to enhance the trustability and delicacy of the systems.

- The integration of machine literacy and artificial intelligence ways should be explored to ameliorate the system's capability to learn and acclimatize to new crash scripts.

In conclusion, the present disquisition on crash discovery systems in motorcars has handed precious perceptivity into the current state of the technology. The assessed systems show pledge in directly detecting crashes and initiating timely responses. still, challenges related to environmental variability and data sequestration need to be addressed for farther advancements. The recommendations outlined in this report serve as a companion for unborn exploration and development sweats in this field.

Should you bear any farther information or explanation, please don't vacillate to reach out.

Thank you for your attention to this report.

Unfeignedly,

Shreya

Aryan Singh

CHAPTER – 5

5.1 SOFTWARE AND HARDWARE SPECIFICATION

Crash discovery systems in motorcars generally involve a combination of software and tackle factors to directly descry and respond to collisions. Then are some of the key software and tackle rudiments generally used in crash discovery systems

1. Accelerometers and Gyroscopes These tackle detectors are used to measure the vehicle's acceleration and angular haste. They give vital information about the vehicle's movement and exposure, which can be used to descry unforeseen changes indicating a collision.

2. Impact and propinquity Detectors These detectors are designed to descry impacts or objects in the vicinity of the vehicle. They can include radar, lidar, ultrasonic detectors, or cameras that help measure the distance to objects and identify implicit collision pitfalls.

3. Electronic Control Unit(ECU) The ECU is a devoted tackle element responsible for monitoring and controlling colorful systems in the vehicle. It receives input from the detectors and processes the data using technical software algorithms to determine if a crash has passed or is imminent.

4. Crash Detection Algorithms Software algorithms are employed to dissect the detector data and descry collision events. These algorithms frequently use advanced signal processing ways and pattern recognition algorithms to identify abrupt changes in vehicle dynamics that indicate a crash.

5. Decision- Making sense Once a crash is detected, the system needs to make opinions regarding applicable conduct. This involves software factors that assess the inflexibility of the collision, identify implicit pitfalls to inhabitants, and spark applicable safety measures like planting airbags or driving seatbelt pretensioners.

6. Communication Interfaces Crash discovery systems may also incorporate communication interfaces to bear information about the crash to external realities, similar as exigency services or telematics systems. These interfaces can use technologies like cellular networks or devoted short- range communication(DSRC) to transmit data.

7. Data Storage and Event Logging Some crash discovery systems include storehouse capabilities to record applicable detector data leading up to and during a crash event. This

data can be used forpost-incident analysis, forensic purposes, or enhancement of crash discovery algorithms.

It's important to note that the specific tackle and software factors used in crash discovery systems can vary depending on the vehicle make and model, as well as the position of complication asked . Advanced motorist- backing systems(ADAS) and independent vehicles may incorporate fresh detectors and complex software infrastructures to enhance crash discovery and forestallment capabilities.

CHAPTER – 6

6.1 PROPOSED SYSTEM MODEL

- *In this section, comparing the different works & researchers on the topic accident detection techniques.*
- *A framework was proposed for mixed traffic flow environment, techniques included in this were:-*
- *CNN*
- *CNN + LSTM + RNN*
- *YOLO*

System Model Offer Crash Detection System in motorcars
The proposed system model for a crash discovery system in motorcars encompasses the integration of colorful factors and technologies to directly descry and respond to crash events. The system model consists of the following crucial rudiments:

1. Detectors The system incorporates a combination of detectors, including accelerometers, gyroscopes, radar, lidar, and cameras. These detectors collect data regarding vehicle dynamics, acceleration, retardation, rotational forces, propinquity to obstacles, and visual information of the girding terrain.

2. Data Acquisition The detector data is acquired and reused in real- time by the system. The data accession module is responsible for collecting detector inputs and transmitting them to the processing unit for analysis.

3. Processing Unit This unit comprises important processors and algorithms designed to dissect the detector data. The processing unit is responsible for performing complex calculations, relating patterns, and determining whether a crash event has passed.

4. Crash Detection Algorithms The system employs sophisticated crash discovery algorithms to interpret the detector data and identify crash events directly. These algorithms are designed to dissect the collected data in real- time, considering factors similar as unforeseen changes in vehicle dynamics, collision forces, and patterns reflective of crash events.

5. Crash Bracket Once a crash event is detected, the system categorizes the type of crash, similar as frontal- end collision, hinder- end collision, side impact, or rollover.

The bracket module utilizes the detector data to separate between colorful crash scripts.

6. Decision- Making Grounded on the analysis and bracket of the crash event, the system makes opinions regarding the inflexibility of the crash and the applicable response. This may include planting airbags, cranking seatbelt pretensioners, cutting off energy force, or initiating exigency announcements.

7. Integration with Vehicle Safety Systems The crash discovery system seamlessly integrates with the vehicle's being safety systems. This integration enables immediate response conduct, similar as cranking pre-existing safety mechanisms like airbags and seatbelt pretensioners, to alleviate the implicit impact of the crash and cover the inhabitants.

8. Communication and waking The system is designed to communicate with external realities similar as exigency services or connected bias. It can shoot real- time crash announcements, including crash position, inflexibility, and inhabitant information, to exigency services for prompt backing.

9. Stoner Interface The system includes a stoner interface, generally located on the vehicle's dashboard, to give applicable information to the motorist and inhabitants. This interface can display crash cautions, safety recommendations, and exigency contact details to grease effective communication and decision- timber.

10. Data Storage and Analysis The system can store crash data forpost-analysis and disquisition purposes. This data can be employed to assess the effectiveness of the system, identify areas for enhancement, and grease accident reconstruction if demanded.

It's important to note that the proposed system model is a abstract frame, and the perpetration details may vary grounded on specific technologies, vehicle models, and nonsupervisory conditions. The system model serves as a foundation for designing an effective and accurate crash discovery system, icing inhabitant safety and mollifying the implicit pitfalls associated with crashes in motorcars.

6.2 ALGORITHM & PERPETRATION

CNN:

The main goal is to incorporate a system that can recognize an accident from video footage captured by a camera. By promptly detecting an accident and immediately notifying the authorities, the system is intended to assist accident victims in need. Utilizing cutting-edge Deep Learning Algorithms that make use of Convolutional Neural Networks (CNNs) to analyze frames taken from the camera's video, the goal is to identify an accident within seconds of its occurrence. We have concentrated on establishing this system on highways where there is less traffic and assistance rarely reaches accident victims in a timely manner. On parkways we can arrangement CCTV camera's put at distance of around 500 meters which go about as a vehicle for observation, on this camera we can set up the proposed framework which takes the recording from the CCTV camera's and runs it on the proposed mishap recognition model to recognize mishaps.

In an machine crash discovery system, a Convolutional Neural Network(CNN) can be employed to dissect detector data and images captured by onboard cameras or other detectors to identify implicit crash situations. Then is an overview of how a CNN algorithm could be applied in such a system.

1. Data collection Collect a labeled dataset that includes images or detector data from a variety of scripts, encompassing both crash andnon-crash cases. These cases should cover different types of collisions, road conditions, and environmental factors.

2. Data preprocessing Prepare the collected data for training by performing necessary preprocessing way. This might involve resizing and normalizing images, converting data formats, and applying addition ways to enhance the diversity and robustness of the dataset.

3. Model training Train a CNN model using the set dataset. The armature of the CNN generally comprises several convolutional layers, followed by pooling layers, completely connected layers, and an affair subcaste. During training, the model learns to prize applicable features from the input data and make prognostications grounded on the handed markers.

4. Real- time crash discovery Once the CNN model is trained, it can be stationed in real- time crash discovery systems in motorcars. The system continuously captures detector data or images from onboard cameras and feeds them into the trained model. The model processes the input data and predicts whether a crash is

imminent or has passed.

5. Alert generation and response Grounded on the model's prognostications, applicable conduct can betriggered.However, the system can induce an alert to advise the motorist and initiate preventative measures likepre-charging the thickets or tensing the seat belts, If the CNN predicts an impending crash. In the case of a crash, the system can automatically emplace airbags, detector exigency services, or spark other safety mechanisms.

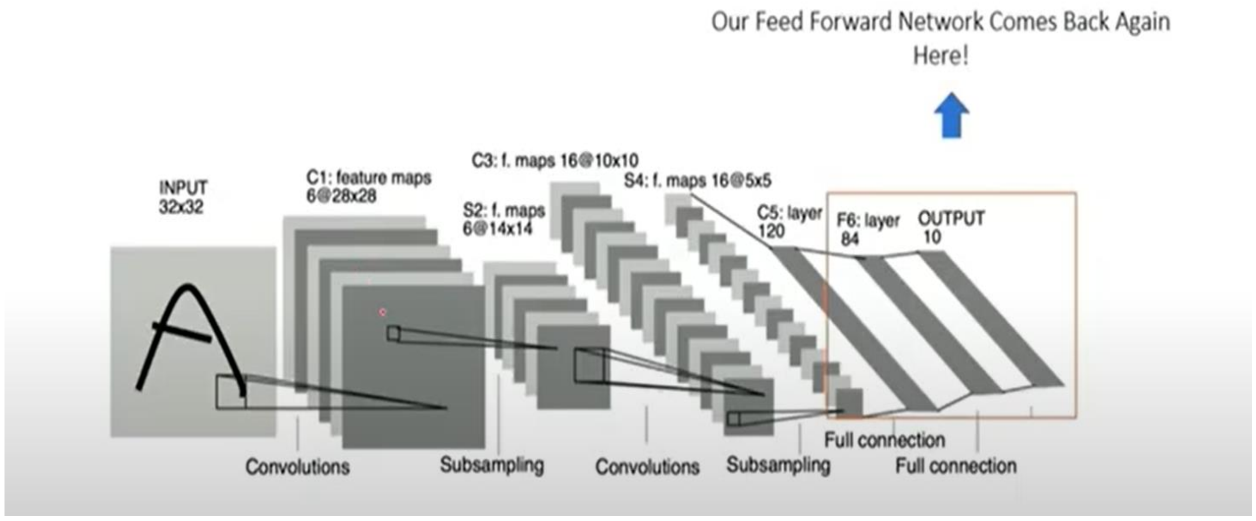
It's important to note that crash discovery systems in motorcars are generally complex and integrate multiple algorithms, detectors, and data emulsion ways for bettered delicacy and trustability. CNNs can be a precious element in similar systems, contributing to the analysis and interpretation of visual or detector data to identify implicit crash situations.

We proposed accident detection using CNN Algorithm. The intent is to create a system which would detect an accident based on the live feed of video from a CCTV camera installed on a highway. The idea is to take each frame of a video and run it through a deep learning convolution neural network model which has been trained to classify frames of a video into accident or non-accident.
Convolutional Neural Networks has proven to be a fast and accurate approach to classify images. CNN based image classifiers have given accuracy’s of more than 95% for comparatively smaller datasets and require less pre-processing as compared to other image classifying algorithms.

System Analysis A. Existing System The available accident detection systems in automobiles are only post-accident response systems such as air bags. They fail to track collision and pre-damage status. Also monitoring the physical condition.

of the driver is missing. There is no automatic means of reaching out for immediate help or first aid in the occurrence of an accident. There is no such accident prevention technologies in use. B. Proposed System The proposed accident prevention system alerts the driver when he is about to fall asleep.
This system also continuously monitors the heart rate and alcohol consumption levels of the driver and intimates the emergency contacts in case of any abnormalities such as a heart attack and provides them his location details.

ARCHITECTURE of CNN



Limitations of CNN:-

- A vanishing gradient problem is dealed under the Convolutional Neural Network, which might cause problems at the time of feature extractions affecting the accuracy of the overall system.

- When there are more layers in the network, the value of the product of derivative decreases until at some point the partial derivative of the loss function approaches a value close to zero, and the partial derivative vanishes. We call this the vanishing gradient problem.

METHODS TO OVERCOME:-

The simplest solution to the problem is to replace the activation function of the network. Instead of sigmoid, use an activation function such as ReLU.
Rectified Linear Units (ReLU) are activation functions that generate a positive linear output when they are applied to positive input values. If the input is negative, the function will return zero.

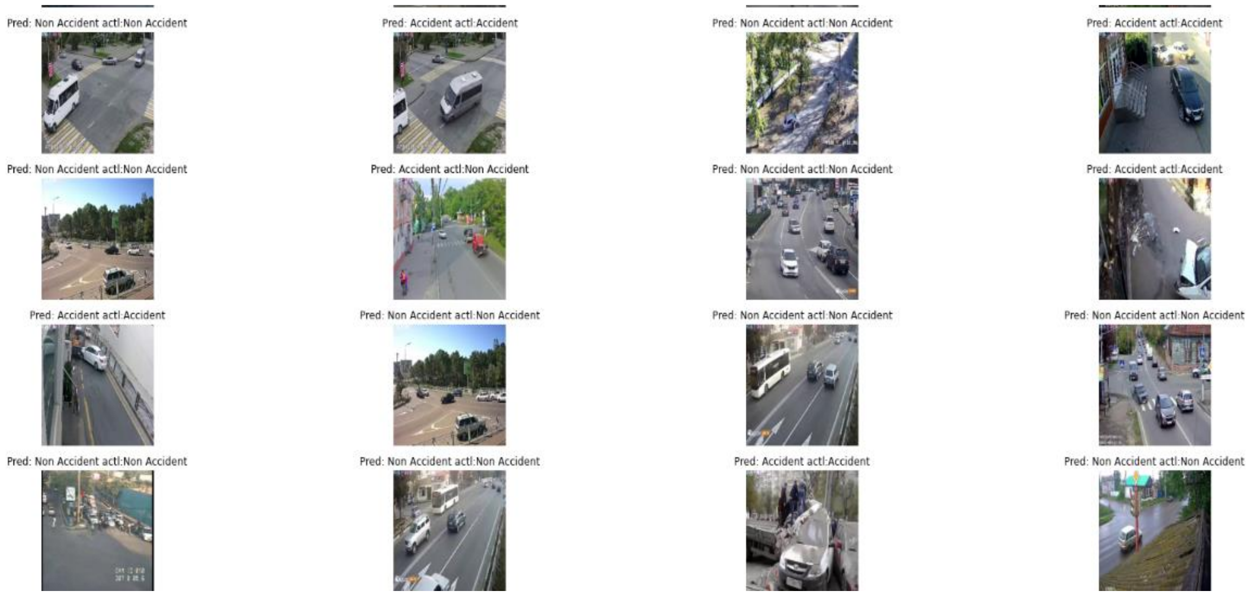
The problem with the use of ReLU is when the gradient has a value of 0. In such cases, the node is considered as a dead node since the old and new values of the weights remain the same. This situation can be avoided by the use of a leaky ReLU function which prevents the gradient from falling to the zero value.

CNN IMPLEMENTATION CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import layers
from time import perf_counter
import os
batch_size = 100
img_height = 250
img_width = 250
training_ds = tf.keras.preprocessing.image_dataset_from_directory(
    '/kaggle/input/accident-detection-from-cctv-footage/data/train',
    seed=42,
    image_size= (img_height, img_width),
    batch_size=batch_size

)
validation_ds =  tf.keras.preprocessing.image_dataset_from_directory(
    '/kaggle/input/accident-detection-from-cctv-footage/data/val',
    seed=42,
    image_size= (img_height, img_width),
    batch_size=batch_size)
testing_ds = tf.keras.preprocessing.image_dataset_from_directory(
    '/kaggle/input/accident-detection-from-cctv-footage/data/test',
    seed=42,
    image_size= (img_height, img_width),
    batch_size=batch_size)

class_names = training_ds.class_names

## Configuring dataset for performance
AUTOTUNE = tf.data.experimental.AUTOTUNE
training_ds = training_ds.cache().prefetch(buffer_size=AUTOTUNE)
testing_ds = testing_ds.cache().prefetch(buffer_size=AUTOTUNE)

MyCnn = tf.keras.models.Sequential([
  layers.BatchNormalization(),
  layers.Conv2D(32, 3, activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(128, 3, activation='relu'),
  layers.MaxPooling2D(),
  layers.Flatten(),
  layers.Dense(256, activation='relu'),
  layers.Dense(len(class_names), activation= 'softmax')
])



MyCnn.compile(optimizer='adam',loss='sparse_categorical_crossentropy', metrics=['accuracy'])
retVal = MyCnn.fit(training_ds, validation_data= validation_ds, epochs = 10)
plt.plot(retVal.history['loss'], label = 'training loss')
plt.plot(retVal.history['accuracy'], label = 'training accuracy')
plt.grid(True)
plt.legend()
plt.plot(retVal.history['val_loss'], label = 'validation loss')
plt.plot(retVal.history['val_accuracy'], label = 'validation accuracy')
plt.grid(True)
plt.legend()
AccuracyVector = []
plt.figure(figsize=(30, 30))
for images, labels in testing_ds.take(1):
    predictions = MyCnn.predict(images)
    predlabel = []
    prdlbl = []


    for mem in predictions:
        predlabel.append(class_names[np.argmax(mem)])
        prdlbl.append(np.argmax(mem))


    AccuracyVector = np.array(prdlbl) == labels
for i in range(40):
        ax = plt.subplot(10, 4, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title('Pred: '+ predlabel[i]+' actl:'+class_names[labels[i]] )
        plt.axis('off')
        plt.grid(True)


MyCnn.save('Model.h5')


from keras.utils.vis_utils import plot_model
plot_model(MyCnn, to_file='model_plot.png', show_shapes=True, show_layer_names=True)
```

TESTING DATASET

Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Non Accident
Pred: Accident actl:Non Accident
Pred: Non Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Non Accident


Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Non Accident actl:Non Accident
Pred: Accident actl:Accident
Pred: Non Accident actl:Non Accident

CNN LSTM RNN:

Combining CNN(Convolutional Neural Network) with LSTM(Long Short- Term Memory) and RNN(intermittent Neural Network) infrastructures can produce a important frame for crash discovery in motorcars. This mongrel approach can work the strengths of each network type to effectively dissect both image and successional detector data. Then is an overview of how CNN, LSTM, and RNN can be combined in a crash discovery system.

- A Convolutional LSTM (Long Short Term Memory) Auto-Encoder is used and this method fails to detect an accident if participants are connected externally.
- Due to lack of proper visibility in the tunnels the low object detection is found. The system reliability also needs improvement.
-

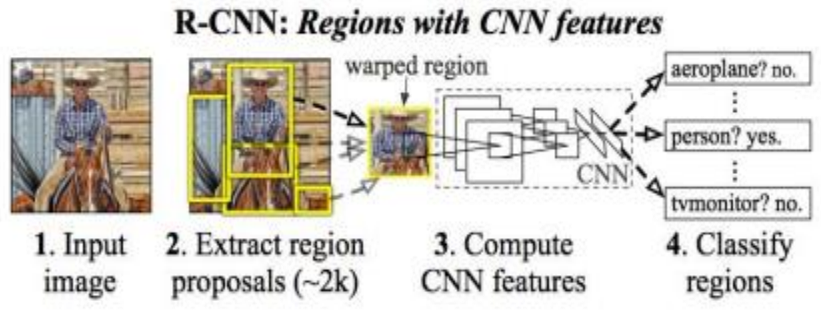
Region Proposal Various recent studies have provided methods to produce categorical independent zone recommendations. These methods have examples such as the image window's objectness , Recognition of object by selective Search , object proposals based on category independence, segmentation of object by using min-cut parametric constraints, grouping of Multiscale combinatorial and so on.
The above methods implements convolution neural network with square cuts resulting in establishment of cells. Although R-CNN is not based on the specific zone proposal method, selective search methods are used for applying R-CNN to provide comparison with the predetermined work.
The RCNN is used to extract region by applying selective search. These extracted regions extracted by selective search in turn forms the objects. These regions are based on texture, colours, varying scales and enclosures. The selective search identifies these regions in the images. The step that is followed is, first it takes input in the form of an image, and then to have multiple regions sub-segmentations are generated from the input image, next smaller similar regions are combines to form a larger region based on texture, size and shape.

Leveraging the Power of RNN-CNN Hybrids:

While RNNs and CNNs have several differences, they are not completely mutually exclusive. It is actually possible for you to use them together for increased effectiveness. This can especially be helpful when the input has to be classified as visually complex with temporal characteristics.
When these two networks are combined, the resultant network is also known as RCNN.

ARCHITECTURE of RCNN with LSTM

- A system is been introduced that is made able to detect accidents in the videos. In this accident detection technique, the various deep learning concepts are used. Also, the Convolutional Neural Network with LSTM units has been used in order to train a model that could detect accidents in videos as required.
- In the architecture, two convolutional layers being used for featuring of extraction and then 2 layers of LSTM units, and each layer consisting of around 256 hidden LSTM units, followed by the other dense layers.



VISUALIZATON OF TRAINING AND TESTING DATA

LSTM with RCNN IMPLEMENTATION CODE

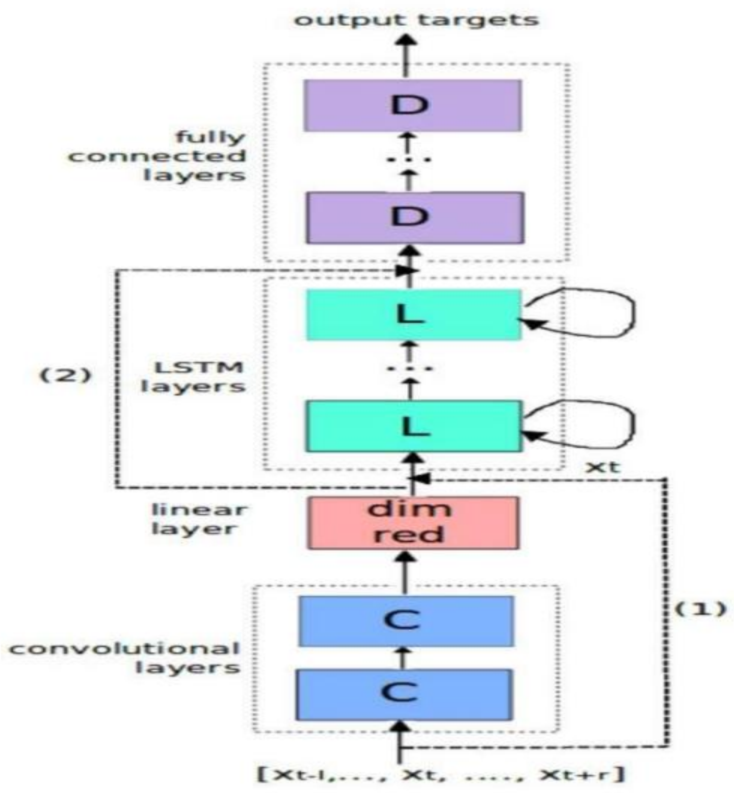
```
 import os
 import copy
 import cv2
 import glob
 import random
 import tqdm
 import numpy as np
 import pandas as pd
 import matplotlib.pyplot as plt
 import seaborn as sns
 from PIL import Image
 from sklearn.model_selection import train_test_split
 from sklearn.metrics import *
 import torch
 from torch import nn
 from torch.utils.data import Dataset, DataLoader
 from torchvision import models
 import torchvision.transforms as transforms
 from torch import optim
 from torch.optim.lr_scheduler import ReduceLROnPlateau
from torchvision.transforms.functional import to_pil_image

 EXTENSION = ".mp4"
 FRAMES = 16
 BATCH_SIZE = 4
 DEVICE = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
print(f'Using {DEVICE} device')

 def get_vids(path2dataset):
     listOfCats = os.listdir(path2dataset)
     ids = []
     labels = []
     for catg in listOfCats:
         path2catg = os.path.join(path2dataset, catg)
         listOfSubCats = os.listdir(path2catg)
         path2subCats= [os.path.join(path2catg, vid) for vid in listOfSubCats]
         ids.extend(path2subCats)
         if catg == 'negative_samples':
             catg = 'No_Accident_Detected'
         else:
             catg = 'Accident_Detected'
```

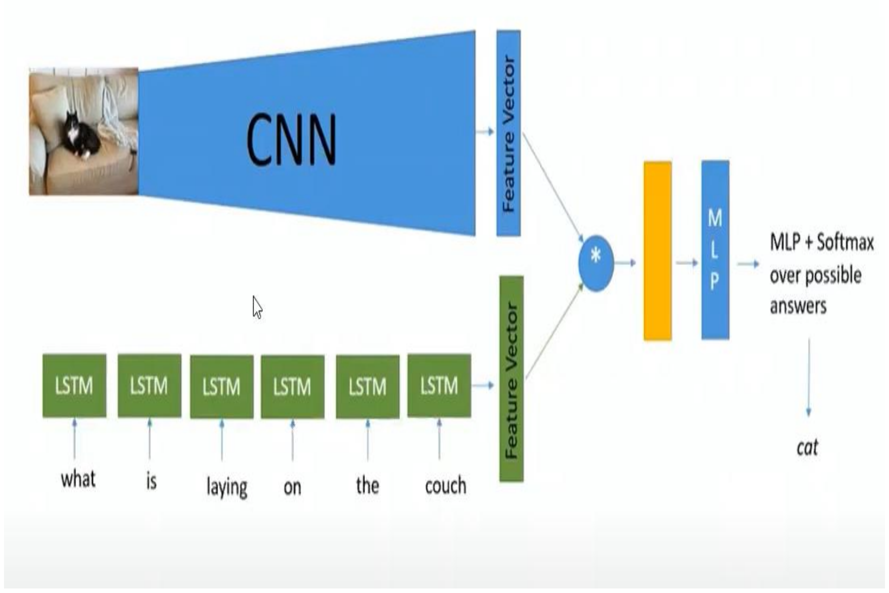
```
        labels.extend([catg]*len(listOfSubCats))
  return ids, labels

def denormalize(x_, mean, std):
    x = x_.clone()
    for i in range(3):
        x[i] = x[i]*std[i]+mean[i]
    x = to_pil_image(x)
  return x

def get_frames(filename, n_frames= 1):
    v_cap = cv2.VideoCapture(filename)
    v_len = int(v_cap.get(cv2.CAP_PROP_FRAME_COUNT))
    height = v_cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
    if height != 720:
        return [], v_len, False
    fps = v_cap.get(cv2.CAP_PROP_FPS)
    duration = v_len/fps
    if duration<3 or duration>16:
        return [], v_len, False
    frames = []
    frame_list= np.linspace(0, v_len-1, n_frames+1, dtype=np.int16)
    for fn in range(v_len):
        success, frame = v_cap.read()
        if success is False:
            continue
        if (fn in frame_list):
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            frames.append(frame)
    v_cap.release()
  return frames, v_len, True

def transform_frames(frames, model_type="rnn"):
    if model_type == "rnn":
        h, w = 224, 224
        mean = [0.485, 0.456, 0.406]
        std = [0.229, 0.224, 0.225]
    else:
        h, w = 112, 112
        mean = [0.43216, 0.394666, 0.37645]
        std = [0.22803, 0.22145, 0.216989]

    test_transformer = transforms.Compose([
            transforms.Resize((h,w)),
            transforms.ToTensor(),
            transforms.Normalize(mean, std)])

    frames_tr = []
    for frame in frames:
        frame = Image.fromarray(frame)
        frame_tr = test_transformer(frame)
        frames_tr.append(frame_tr)
    imgs_tensor = torch.stack(frames_tr)

    if model_type=="3dcnn":
        imgs_tensor = torch.transpose(imgs_tensor, 1, 0)
    imgs_tensor = imgs_tensor.unsqueeze(0)
   return imgs_tensor

def store_frames(frames, path2store):
    for ii, frame in enumerate(frames):
        frame = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)
        path2img = os.path.join(path2store, "frame"+str(ii)+".jpg")
      cv2.imwrite(path2img, frame)

path2data     =     "/kaggle/input/hwid12-highway-incidents-detection-dataset/Video-
Accident-Dataset"
path2dataset = "/kaggle/working/Accident-Dataset-Keyframes"
path2Cats = path2data
Categories = os.listdir(path2Cats)
Categories

for cat in Categories:
    print("Category:", cat)
    path2acat = os.path.join(path2Cats, cat)
    listOfSubs = os.listdir(path2acat)
    print("No of Videos:", len(listOfSubs))
   print("-"*50)

for root, dirs, files in os.walk(path2Cats, topdown=True):
    if len(dirs) == 0:
        cat = root.split('/')[-1]
        cnt = 0
        for name in files:
            if EXTENSION not in name:
                continue
            path2vid = os.path.join(root, name)
            frames, vlen, chk = get_frames(path2vid, n_frames= FRAMES)
            if not chk:
                continue
            cnt += 1
            path2store = path2vid.replace(path2data, path2dataset)
            path2store = path2store.replace(name, name.split("_")[-1].split(".")[0])
            #print(path2store)
            os.makedirs(path2store, exist_ok= True)
            store_frames(frames, path2store)
        print(f'Category: {cat}\nVideos converted into Frames: {cnt}')
      print("-"*50)

  all_vids, all_labels = get_vids(path2dataset)
len(all_vids), len(all_labels)

  labels_dict = {'Accident_Detected': 1, 'No_Accident_Detected': 0}
labels_dict

  from sklearn.model_selection import train_test_split

  x_train, x_test, y_train, y_test = train_test_split(all_vids, all_labels, test_size=0.25,
  random_state=42, stratify=all_labels)
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size=0.15,
random_state=42, stratify=y_train)

np.unique(y_train, return_counts=True)

np.unique(y_val, return_counts=True)

np.unique(y_test, return_counts=True)

x_train[:5], y_train[:5]

x_val[:5], y_val[:5]

x_test[:5], y_test[:5]

  np.random.seed(42)
  random.seed(42)
  torch.manual_seed(42)

  class VideoDataset(Dataset):
    def __init__(self, ids, labels, transform):
        self.transform = transform
        self.ids = ids
        self.labels = labels

    def __len__(self):
        return len(self.ids)

    def __getitem__(self, idx):
        path2imgs = glob.glob(self.ids[idx]+"/*.jpg")
        path2imgs = path2imgs[:FRAMES]
        label = labels_dict[self.labels[idx]]
        frames = []
        for p2i in path2imgs:
            frame = Image.open(p2i)
            frames.append(frame)

        seed = np.random.randint(1e9)
        frames_tr = []
        for frame in frames:
            random.seed(seed)
            np.random.seed(seed)
            frame = self.transform(frame)
            frames_tr.append(frame)
        if len(frames_tr)>0:
            frames_tr = torch.stack(frames_tr)
      return frames_tr, label

model_type = "rnn"

if model_type == "rnn":
    h, w = 224, 224
    mean = [0.485, 0.456, 0.406]
    std = [0.229, 0.224, 0.225]
else:
    h, w = 112, 112
    mean = [0.43216, 0.394666, 0.37645]
  std = [0.22803, 0.22145, 0.216989]

trainTransform = transforms.Compose([
            transforms.Resize((h,w)),
            transforms.RandomHorizontalFlip(p=0.5),
            transforms.RandomAffine(degrees=0, translate=(0.1,0.1)),
            transforms.ToTensor(),
            transforms.Normalize(mean, std),
])

  trainDs = VideoDataset(ids=x_train, labels=y_train, transform=trainTransform)
len(trainDs)

  imgs, label = trainDs[9]
imgs.shape, label, torch.min(imgs), torch.max(imgs)

  plt.figure(figsize=(20,20))
  plt.subplots_adjust(wspace=0, hspace=0)
  for ii, img in enumerate(imgs[::3]):
      plt.subplot(1,10,ii+1)
      plt.imshow(denormalize(img, mean, std))
    plt.axis("off")

  testTransform = transforms.Compose([
            transforms.Resize((h,w)),
            transforms.ToTensor(),
            transforms.Normalize(mean, std),
  ])
  valDs = VideoDataset(ids=x_val, labels=y_val, transform=testTransform)
  testDs = VideoDataset(ids=x_test, labels=y_test, transform=testTransform)
len(valDs), len(testDs)

  imgs, label = testDs[9]
  plt.figure(figsize=(20,20))
  plt.subplots_adjust(wspace=0, hspace=0)
  for ii, img in enumerate(imgs[::3]):
      plt.subplot(1,10,ii+1)
      plt.imshow(denormalize(img, mean, std))
    plt.axis("off")

  def collateRNN(batch):
      imgs_batch, label_batch = list(zip(*batch))
      imgs_batch = [imgs for imgs in imgs_batch if len(imgs)>0]
      label_batch = [torch.tensor(l) for l, imgs in zip(label_batch, imgs_batch) if
  len(imgs)>0]
      imgs_tensor = torch.stack(imgs_batch)
      labels_tensor = torch.stack(label_batch)
    return imgs_tensor, labels_tensor
```

```
trainDL = DataLoader(trainDs, batch_size=BATCH_SIZE, shuffle=True,
num_workers=2, collate_fn=collateRNN)
valDL = DataLoader(valDs, batch_size=BATCH_SIZE, shuffle=True, num_workers=2,
collate_fn=collateRNN)
testDL = DataLoader(testDs, batch_size=2*BATCH_SIZE, shuffle=False,
collate_fn=collateRNN)

for xb, yb in trainDL:
    print(xb.shape, yb.shape)
    break
for xb, yb in valDL:
    print(xb.shape, yb.shape)
    break
for xb, yb in testDL:
    print(xb.shape, yb.shape)
  break

from torchvision.models import ResNet18_Weights
class Resnt18Rnn(nn.Module):
    def __init__(self, params_model):
        super(Resnt18Rnn, self).__init__()
        num_classes = params_model["num_classes"]
        dr_rate= params_model["dr_rate"]
        pretrained = params_model["pretrained"]
        rnn_hidden_size = params_model["rnn_hidden_size"]
        rnn_num_layers = params_model["rnn_num_layers"]

        baseModel = models.resnet18(weights=ResNet18_Weights.DEFAULT)
        num_features = baseModel.fc.in_features
        baseModel.fc = Identity()
        self.baseModel = baseModel
        self.dropout= nn.Dropout(dr_rate)
        self.rnn = nn.LSTM(num_features, rnn_hidden_size, rnn_num_layers)
        self.fc1 = nn.Linear(rnn_hidden_size, num_classes)
        self.relu = nn.ReLU()

    def forward(self, x):
        b_z, ts, c, h, w = x.shape
        ii = 0
        y = self.baseModel((x[:,ii]))
        output, (hn, cn) = self.rnn(y.unsqueeze(1))
        for ii in range(1, ts):
            y = self.baseModel((x[:,ii]))
            out, (hn, cn) = self.rnn(y.unsqueeze(1), (hn, cn))
        out = self.dropout(out[:,-1])
        out = self.fc1(out)
        out = self.relu(out)
        return out

class Identity(nn.Module):
    def __init__(self):
        super(Identity, self).__init__()
    def forward(self, x):
      return x

params_model={
    "num_classes": 2,
    "dr_rate": 0.1,
    "pretrained" : True,
    "rnn_num_layers": 1,
    "rnn_hidden_size": 100,}

model = Resnt18Rnn(params_model).to(DEVICE)
print(model)

with torch.no_grad():
    x = torch.zeros(1, 16, 3, h, w).to(DEVICE)
    y = model(x)
    print(y.shape) path2weights = "/kaggle/working/weights.pt"
torch.save(model.state_dict(), path2weights)

def get_lr(opt):
    for param_group in opt.param_groups:
        return param_group['lr']

def metrics_batch(output, target):
    pred = output.argmax(dim=1, keepdim=True)
    corrects = pred.eq(target.view_as(pred)).sum().item()
    return corrects

def loss_batch(loss_func, output, target, opt=None):
    loss = loss_func(output, target)
    with torch.no_grad():
        metric_b = metrics_batch(output,target)
    if opt is not None:
        opt.zero_grad()
        loss.backward()
        opt.step()
    return loss.item(), metric_b

def loss_epoch(model,loss_func,dataset_dl, sanity_check=False, opt=None):
    running_loss=0.0
    running_metric=0.0
    len_data = len(dataset_dl.dataset)
    for xb, yb in tqdm.notebook.tqdm(dataset_dl):
        xb=xb.to(DEVICE)
        yb=yb.to(DEVICE)
        output=model(xb)
        loss_b,metric_b=loss_batch(loss_func, output, yb, opt)
        running_loss+=loss_b
        if metric_b is not None:
            running_metric+=metric_b
        if sanity_check is True:
            break
    loss=running_loss/float(len_data)
    metric=running_metric/float(len_data)
    return loss, metric

def train_val(model, params):
    num_epochs=params["num_epochs"]
    loss_func=params["loss_func"]
    opt=params["optimizer"]
    train_dl=params["train_dl"]
    val_dl=params["val_dl"]
    sanity_check=params["sanity_check"]
    lr_scheduler=params["lr_scheduler"]
    path2weights=params["path2weights"]

    loss_history={
        "train": [],
        "val": [],
    }

    metric_history={
        "train": [],
        "val": [],
    }

    best_model_wts = copy.deepcopy(model.state_dict())
    best_loss = float('inf')

    for epoch in range(num_epochs):
        current_lr = get_lr(opt)
        print('Epoch {}/{}, Current Learing Rate={}'.format(epoch, num_epochs - 1,
current_lr))
        model.train()
        train_loss, train_metric = loss_epoch(model,loss_func,train_dl,sanity_check,opt)
        loss_history["train"].append(train_loss)
        metric_history["train"].append(train_metric)
        model.eval()
        with torch.no_grad():
            val_loss, val_metric=loss_epoch(model,loss_func,val_dl,sanity_check)
        if val_loss < best_loss:
            best_loss = val_loss
            best_model_wts = copy.deepcopy(model.state_dict())
            torch.save(model.state_dict(), path2weights)
            print("Copied best model weights!")

        loss_history["val"].append(val_loss)
        metric_history["val"].append(val_metric)

        lr_scheduler.step(val_loss)
        if current_lr != get_lr(opt):
            print("Loading best model weights!")
            model.load_state_dict(best_model_wts)

        print("Train loss: %.6f, Validation loss: %.6f, Accuracy: %.2f" %(train_loss,
val_loss, 100*val_metric))
        print("-"*50)
    model.load_state_dict(best_model_wts)

  return model, loss_history, metric_history

def plot_loss(loss_hist, metric_hist):
    num_epochs= len(loss_hist["train"])
    plt.title("Train-Val Loss")
    plt.plot(range(1,num_epochs+1),loss_hist["train"], label="Train")
    plt.plot(range(1,num_epochs+1),loss_hist["val"], label="Val")
    plt.ylabel("Loss")
    plt.xlabel("Training Epochs")
    plt.legend()
    plt.show()
```

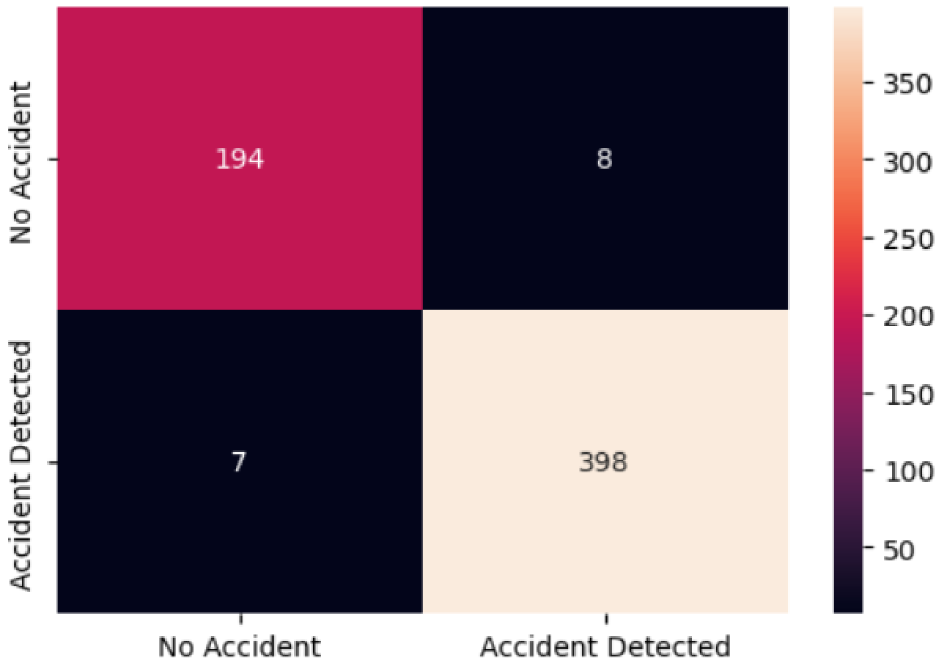
```
      plt.title("Train-Val Accuracy")
      plt.plot(range(1,num_epochs+1), metric_hist["train"],label="Train")
      plt.plot(range(1,num_epochs+1), metric_hist["val"],label="Val")
      plt.ylabel("Accuracy")
      plt.xlabel("Training Epochs")
      plt.legend()
    plt.show()

  loss_func = nn.CrossEntropyLoss(reduction="sum")
  opt = optim.Adam(model.parameters(), lr=0.00085)
  lr_scheduler = ReduceLROnPlateau(opt, mode='min',factor=0.5, patience=5,verbose=1)
  os.makedirs("/kaggle/working/models", exist_ok=True)

  params_train={
      "num_epochs": 50,
      "optimizer": opt,
      "loss_func": loss_func,
      "train_dl": trainDL,
      "val_dl": valDL,
      "sanity_check": False,
      "lr_scheduler": lr_scheduler,
      "path2weights": "/kaggle/working/models/weights_"+model_type+".pt",
  }

model, loss_hist, metric_hist = train_val(model, params_train)

plot_loss(loss_hist, metric_hist)

  with torch.no_grad():
      testLoss, testAcc = loss_epoch(model, loss_func, testDL)
    print("Test loss: %.6f, Accuracy: %.2f" %(testLoss, 100*testAcc))

  y_pred = []
  y_true = []
  model.eval()
  with torch.no_grad():
      for video_batch, labels  in tqdm.notebook.tqdm(testDL):
          video_batch, labels = video_batch.to(DEVICE), labels.to(DEVICE)
          output = model(video_batch)
          output = (torch.max(torch.exp(output), 1)[1]).data.cpu().numpy()
          labels = labels.cpu().numpy()
          y_pred.append(output)
```

```
        y_true.append(labels)

  y_true = np.hstack(y_true)
y_pred = np.hstack(y_pred)

  df = pd.DataFrame((y_true, y_pred), index = ['Label', 'Prediction'])
df

  classes = ('No Accident', 'Accident Detected')
  cf_matrix = confusion_matrix(y_true, y_pred)
  df_cm = pd.DataFrame(cf_matrix, index = [i for i in classes],
                      columns = [i for i in classes])
  plt.figure(figsize = (6, 4))
sns.heatmap(df_cm, annot=True, fmt="d")

  print("Precision:\t"+str(precision_score(y_true, y_pred)))
  print("Recall:\t"+str(recall_score(y_true, y_pred)))
print("F1-Score:\t"+str(f1_score(y_true, y_pred)))
```





CONFUSION MATRIX & PERFORMANCE ANALYSIS



ACCURACY : 0.9752883031
PRECISION: 0.9802955665024631
RECALL: 0.9827160493827161
F1-SCORE: 0.9815043156596794

1. Data collection Gather a labeled dataset that includes both image data captured by cameras and successional detector data from colorful driving scripts, encompassing crash andnon-crash cases.

2. Data preprocessing Preprocess the image data by resizing, homogenizing, and accelerating it as demanded. For successional detector data, organize it into suitable time- series sequences and apply any necessary normalization or scaling.

3. CNN for image analysis use a CNN to reuse the image data and excerpt applicable features. The CNN's convolutional layers learn spatial patterns and hierarchical representations, while the posterior pooling and completely connected layers capture high- position features. The affair of the CNN is a fixed- length vector representing the image's features.

4. LSTM/ RNN for successional data analysis Feed the successional detector data, organized as time- series sequences, into an LSTM or RNN network. These networks exceed at landing temporal dependences and modeling successional patterns. The LSTM layers, with their memory cells, can retain important information over longer time way.

5. Fusion of CNN and LSTM/ RNN Concatenate or combine the labors from the CNN(image features) and LSTM/ RNN(successional information) into a unified representation. This emulsion can be achieved by concatenating the vectors or applying other emulsion ways, similar aselement-wise addition or addition.

6. Bracket and crash discovery Connect the fused representation to completely connected layers, followed by an affair subcaste for crash discovery. Train the combined model using the labeled dataset, allowing it to learn to make prognostications grounded on the fused features.

7. Real- time crash discovery In the stationed system, continuously prisoner and process new image data from cameras and successional detector data from onboard detectors. Pass the images through thepre-trained CNN to prize image features. contemporaneously, feed the successional detector data into the LSTM/ RNN layers to capture temporal patterns. Combine the labors of both networks and pass them through the trained bracket layers to descry implicit crash events in real- time.

8. Alert generation and response Grounded on the crash discovery results, induce applicable cautions and initiate response conduct, similar as cranking safety mechanisms, waking the motorist, or notifying exigency services.

By combining the strengths of CNN, LSTM, and RNN, the crash discovery system can effectively dissect both visual and successional detector data, enabling accurate and timely discovery of implicit crash events in motorcars.

YOLOv3

Yolo algorithm achieves its result by applying a neural network on an image..The image is divided in SxS grid and comes up with bounding box[21].This algorithm has 24 convolutional layers which in turn has two fully connected layers. The reduction in feature space is done by Alternating 1x1 convolutional layers from preceding layers. The object identification problem is considered to be a regression problem with the objective of spatially bounding box separation along with the probability of associated classes in the bounding boxes. A single neural network can predict the bounded boxes and class probabilities directly from the input images in just one evaluation which can be optimized end-toend.

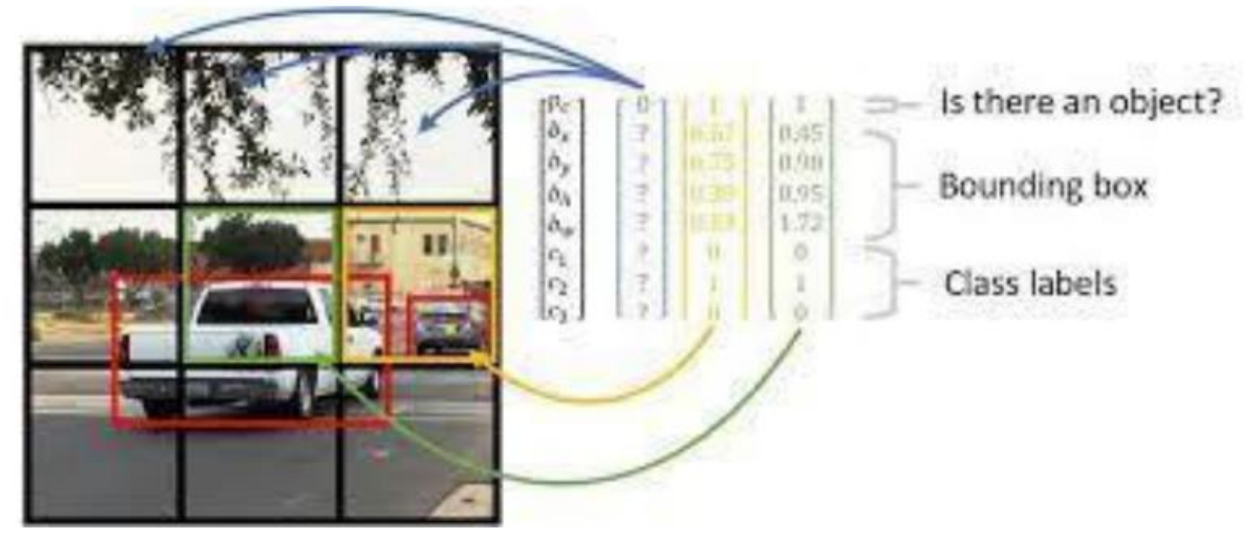


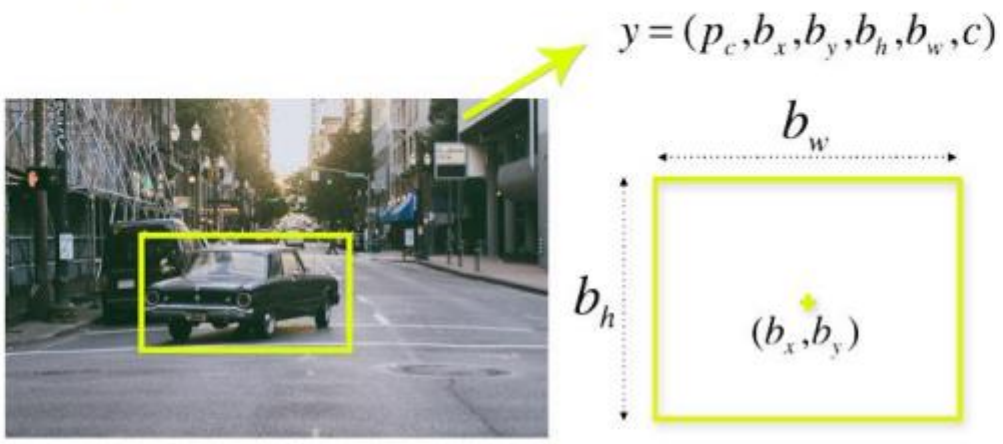
FIGURE 1: YOLO ALGORITHM

In this study, we are going to apply yolo algorithm for detection of objects through a live feed or an image. The working of yolo is quite simple as yolo is based on regression. Unlike CNN which selects interesting parts in an image, yolo on the other hand predicts the class and bounding boxes for the whole image in one run of the algorithm.
To apply this algorithm we need to know what we are going to predict i.e. the objects we are likely to be interested in so that we can train our algorithm to look for classes of the objects and the bounding box specifying the object location.

The bounding box are described using these four descriptions

- ● Center of bounding box (bx, by)
- ● Width (bw)
- ● Height (bh)
- ● C: class name of the identified object

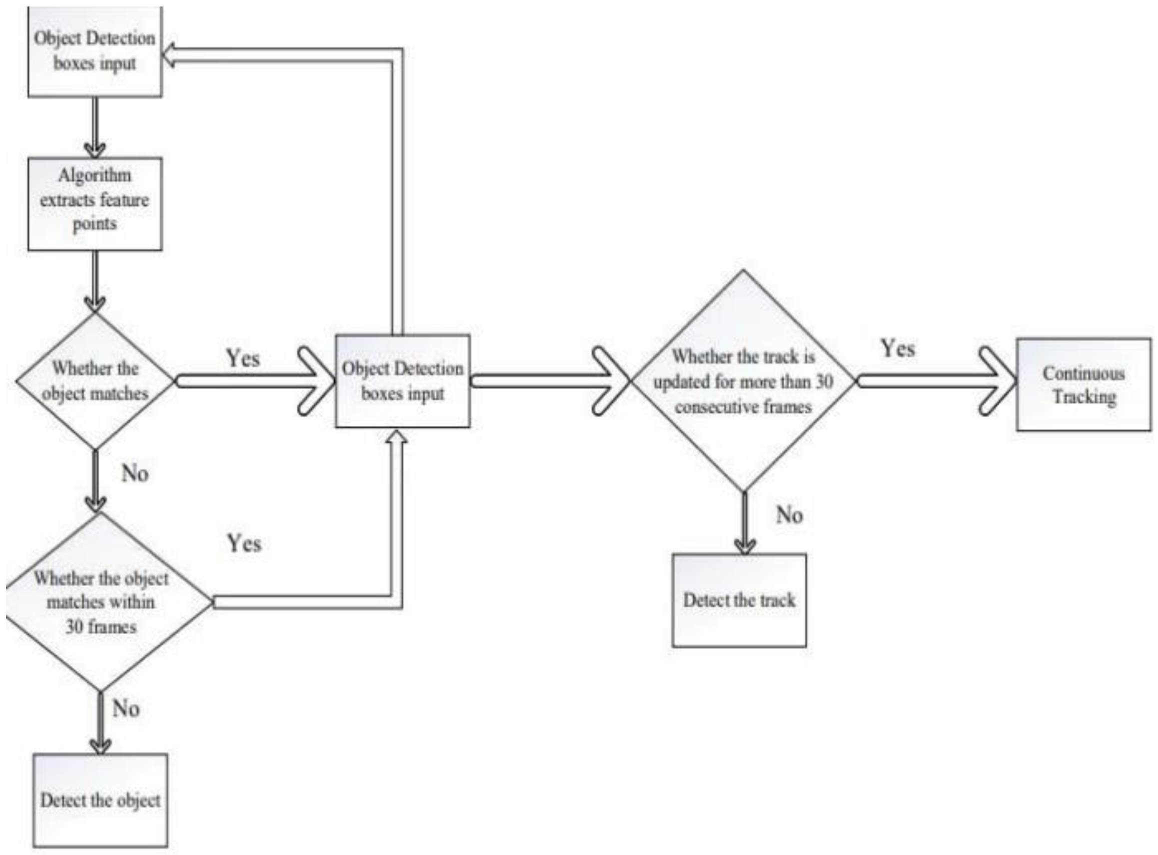
Pc is the probability of objects in the bounding box.



OBJECT IDENTIFICATION

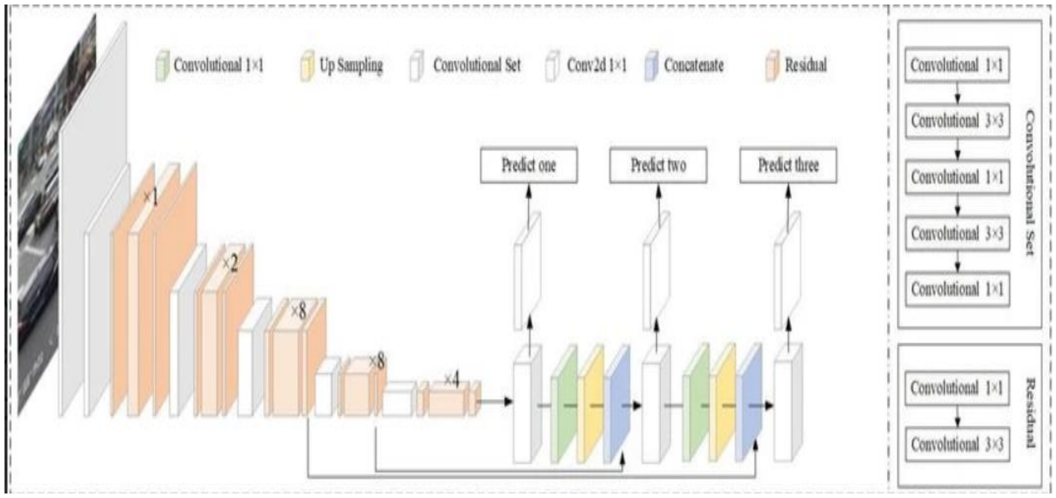
Car Detection:

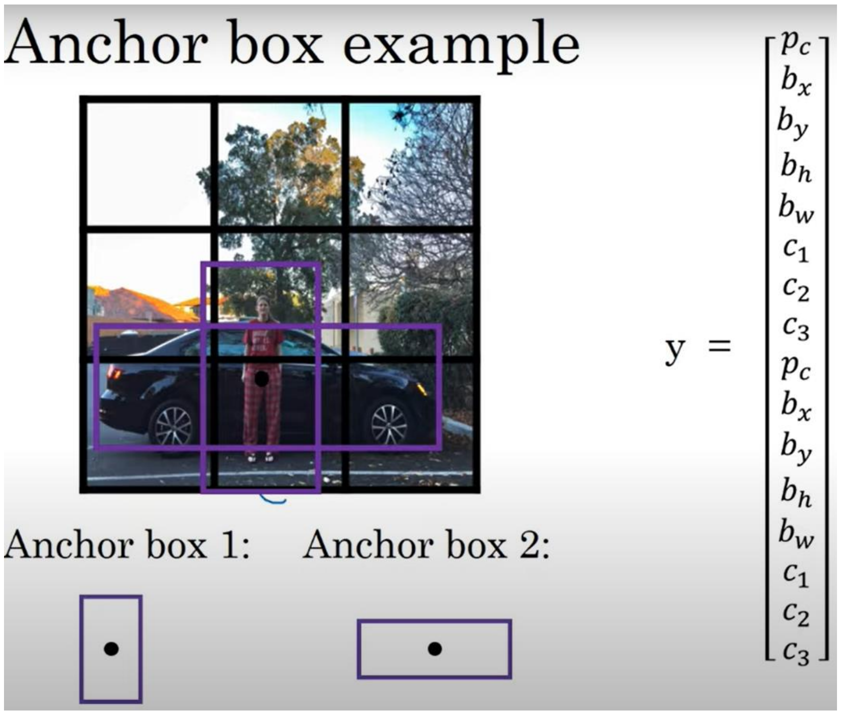
For car detection we used the YOLO net, it was proposed by, improved in and more recently, the last version was presented in. Despite other convolutional networks, it detects objects with a single pass creating a grid of SxS boxes, where each box has a logistic regression and a classification method, the regression method predict each box with five values: x, y, w, h and the confidence of the object being there.
The classifier predicts C conditional class probabilities.
At the final stage multiple bounding boxes appear around a single object, so nonmaximum suppression is applied in order to keep the strongest detection around a single object. The architecture of YOLO for this research project is showed in the table I, obtained from Darknet library.
The result in this step are the bounding boxes for each car, the YOLO performance is showed, here, this network has a very high accuracy and more important have a very low time processing against the rest.


Object Detection boxes input
Algorithm extracts feature points
Whether the object matches
Yes
Object Detection boxes input
Whether the track is updated for more than 30 consecutive frames
Yes
Continuous Tracking
No
Yes
Whether the object matches within 30 frames
No
Detect the track
No
Detect the object

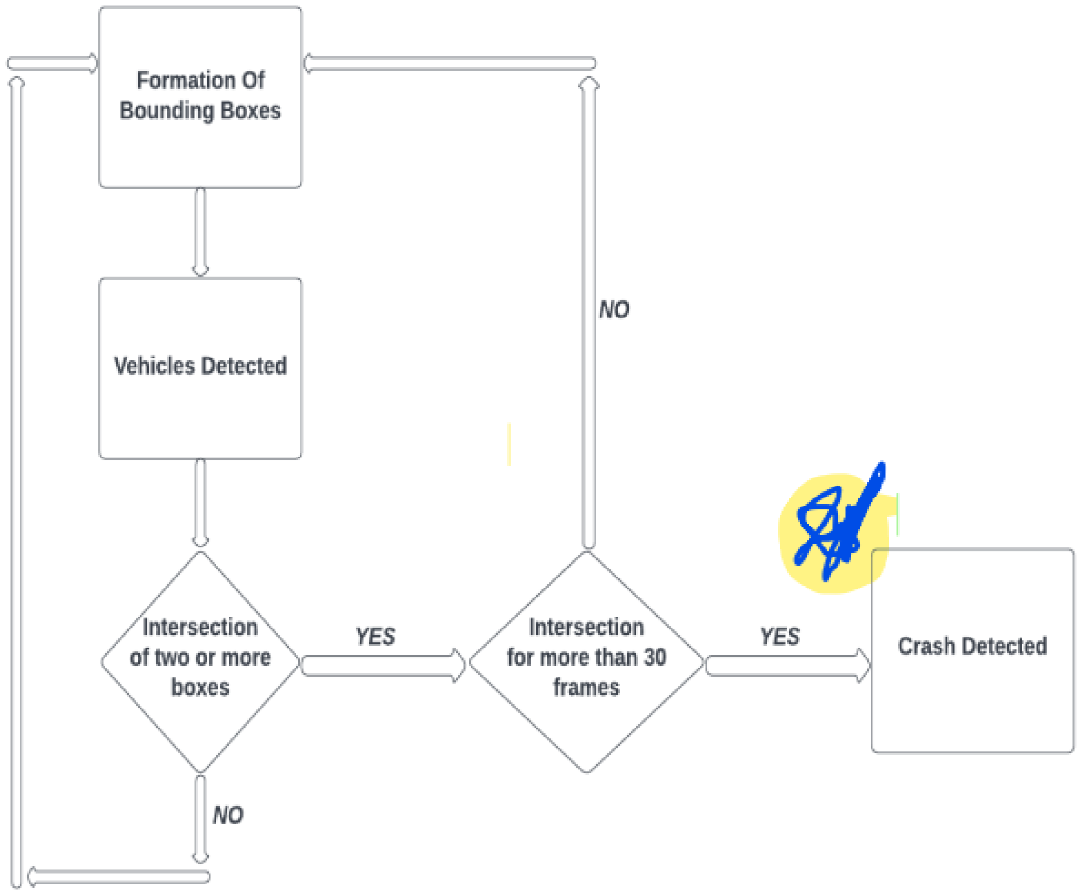
Car tracking:
We apply a car tracking algorithm each 30 frames, generating one tracker per car and saving the tracker position. Then we extract small videos, one per car. In the right of the Figure 5 we show an example, where three small videos have been extracted because three cars were detected in the first stage of the proposed model. The car tracker used in our model is based in a visual. Objects algorithm with correlation filters, proposed by Danelljan et al. The algorithm consists in the translation and scale estimation of the object through a series of correlations over the Fourier domain, where each correlation filter is submitted to an online learning process.
The procedure of correlation filters since the position of the visual object, according to Chen et al, can be summarized as follows.
In each frame the region whose position was predicted in the previous frame is extracted for detection. Then the features HOG are extracted since the intensity level of each pixel of the image region, a cosine window is applied for smoothing a boundary effects. Next, a series of correlation operations are performed with element-wise multiplications using the Discrete Fourier Transform (DFT), it is computated with an efficient Fast Fourier Transform (FFT) algorithm, the answer is a spatial confidence map obtained with the inverse of FFT (IFFT).
The position with the maximum value located on the map is predicted as the new position of the target.
Then, the features of the estimated position region are extracted again for the training and

updating of the correlation filter, being part of the online learning, the IFFT is applied again to obtain the response map and another prediction, this process continuous for each frame.





Car Crash Detection:
In order to detect a car crash scene, we are going to
use the ViF descriptor because of the very low cost and acceptable accuracy. The ViF descriptor regards the statistics of magnitude changes of flow vectors over time. In order to get these vectors, used the optical flow algorithm proposed by named Iterative Reweighted Least Squares (IRLS), but in this context, we used the ViF descriptor with Horn-Schunck as optical flow algorithm proposed by.


Formation Of Bounding Boxes
Vehicles Detected
Intersection of two or more boxes
YES
Intersection for more than 30 frames
YES
Crash Detected
NO
NO

YOLOv3(You Only Look Once v3) is an object discovery algorithm that can be employed in a crash discovery system in motorcars. YOLOv3 is known for its real-time object discovery capabilities, making it suitable for recycling videotape aqueducts or images captured by onboard cameras. Then is how YOLOv3 can be applied in a crash discovery system.

1. Data collection Collect a labeled dataset that includes images or videotape frames captured by cameras in colorful driving scripts, covering both crash andnon-crash cases. The dataset should contain annotated bounding boxes around the objects of interest, including vehicles, climbers, and other applicable objects.

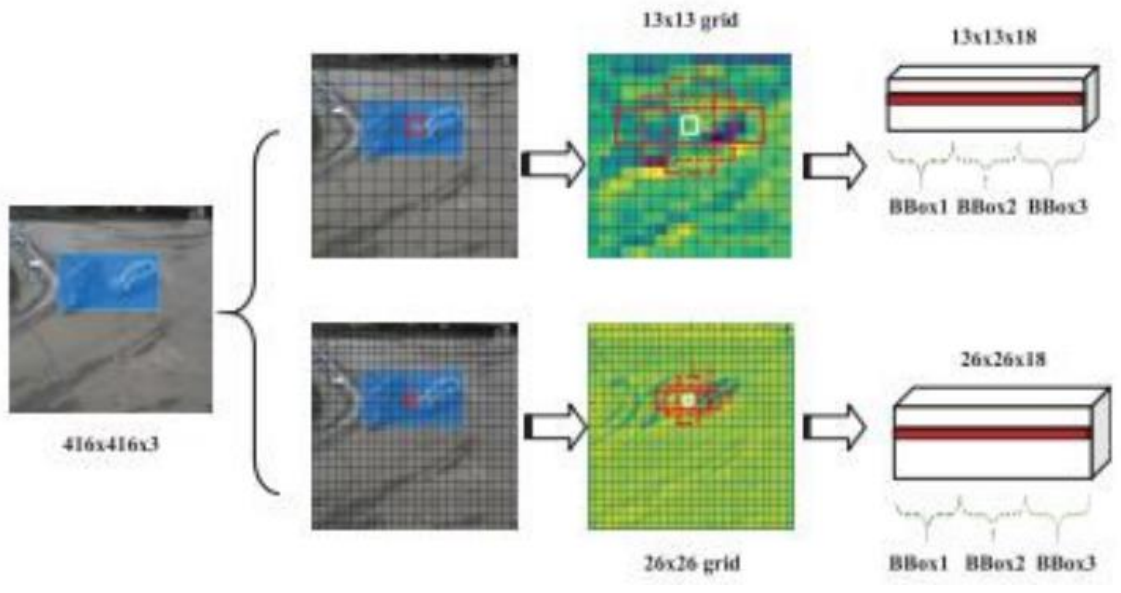
2. Model training Train a YOLOv3 model using the labeled dataset. This involves configuring the network armature and optimizing it using a large- scale dataset. Training generally involves dividing the input images into a grid, prognosticating bounding boxes and class chances within each grid cell, and enriching the prognostications using multiple layers.

3. Real- time crash discovery Emplace the trained YOLOv3 model in a real- time crash discovery system. The system continuously captures videotape aqueducts or images from onboard cameras and feeds them into the model. YOLOv3 processes the input data and detects and localizes objects of interest, including vehicles and climbers.

4. Crash event determination dissect the detected objects and their positions to determine implicit crash events. This may involve fresh sense, similar as assessing the relative rapidity, distances, and circles of objects, to infer the liability of a crash.

5. Alert generation and response Grounded on the crash event determination, induce applicable cautions or detector responseactions.However, the system can spark safety mechanisms, similar as planting airbags or driving exigency retardation systems, If a crash is detected or prognosticated. also, cautions can be generated to notify the motorist or exigency services.

It's important to note that while YOLOv3 can descry and localize objects, crash discovery systems frequently employ fresh algorithms and detector inputs for further comprehensive analysis. The integration of YOLOv3 with other algorithms, similar as line analysis or emulsion with successional detector data, can enhance the delicacy and trustability of the crash discovery system in motorcars.



THE DETECTION PRINCIPLE OF YOLOV3

HOW YOLO WORKS ?

- The YOLOv3 algorithm first separates an image into a grid.
- Each grid cell predicts some number of boundary boxes (sometimes referred to as anchor boxes) around objects that score highly with the aforementioned predefined classes.
- Each boundary box has a respective confidence score of how accurate it assumes that prediction should be and detects only one object per bounding box.
- The boundary boxes are generated by clustering the dimensions of the ground truth boxes from the original dataset to find the most common shapes and sizes.
- To classify different kinds of objections, independent logistic classifiers are used instead of a SoftMax.

Decision tree model was considered for crash classification, based on features obtained from Yolo v3.

CHAPTER – 7

8.1 RESULT ANALYSIS

CNN RESULT ANALYSIS:

Convolutional Neural Networks (CNNs) have shown promising results in crash detection systems for automobiles. These systems utilize CNNs to analyze sensor data from various sources such as accelerometers, gyroscopes, and cameras, to accurately identify and detect crash events.

By training CNNs on large datasets of labeled crash data, these networks can learn complex patterns and features that distinguish normal driving behavior from crash events. The input sensor data is typically preprocessed and fed into the CNN, which consists of multiple convolutional layers, pooling layers, and fully connected layers.

The convolutional layers in the CNN help extract spatial and temporal features from the sensor data, allowing the network to capture important patterns related to crash events. The pooling layers downsample the feature maps, reducing computational complexity and aiding in generalization. The fully connected layers at the end of the network perform classification, determining whether a crash has occurred or not.

Through the training process, the CNN learns to recognize distinctive patterns associated with crash events, such as sudden deceleration, significant changes in vehicle orientation, or visual cues indicative of a collision. Once trained, the CNN can be deployed in real-time crash detection systems, continuously analyzing incoming sensor data and making timely decisions.

While specific results may vary depending on the architecture, dataset, and evaluation metrics used, CNNs have demonstrated promising performance in crash detection tasks. They can achieve high accuracy rates, low false-positive and false-negative rates, and quick response times, making them valuable tools in improving vehicle safety.
However, it's worth noting that advancements and further research may have been made since my knowledge cutoff in September 2021.

LSTM with RCNN RESULTS:

LSTM with RCNN (Recurrent Convolutional Neural Network) is a hybrid model that combines the strengths of Long Short-Term Memory (LSTM) and CNN architectures. While LSTM is known for its ability to capture temporal dependencies in sequential data, RCNN enhances the model's capability to extract spatial features from sensor data. Although I don't have access to the latest developments or specific results, I can provide you with a general understanding of how LSTM with RCNN can be applied in crash detection systems.

In crash detection systems, LSTM with RCNN can be employed to analyze sequential sensor data, such as accelerometer and gyroscope readings, and simultaneously capture both temporal and spatial patterns related to crash events. The model typically consists of two main components: an LSTM module and an RCNN module.

The LSTM module processes the sequential input data, learning the temporal dependencies and patterns over time. It takes into account the historical context of the sensor data, enabling the model to capture changes and anomalies indicative of a crash event. The LSTM module is responsible for encoding the sequential information into a fixed-length representation.

The RCNN module, on the other hand, operates on the encoded sequences from the LSTM module and focuses on extracting spatial features from the data. By applying convolutional layers, the RCNN module identifies spatial patterns and local dependencies within the encoded sequences. This allows the model to capture important spatial cues, such as sudden changes in vehicle orientation or acceleration, which can be crucial in crash detection.

By combining the strengths of LSTM and RCNN, the hybrid model can effectively capture both temporal and spatial characteristics of crash events, enhancing the accuracy and performance of the crash detection system. It enables the system to analyze sensor data in a comprehensive manner, leveraging both the sequential nature of the data and the spatial relationships between sensor readings.

While the specific results of LSTM with RCNN on crash detection systems may vary depending on the dataset, model architecture, and evaluation metrics used, this hybrid approach has shown promise in improving the accuracy and reliability of crash detection. However, it's important to note that the field of deep learning is rapidly evolving, and newer developments may have occurred since my knowledge cutoff in September 2021.
YOLOv3 RESULT:

YOLOv3 (You Only Look Once v3) is an object detection algorithm known for its real-time performance and accuracy. While I don't have access to specific results or the latest developments, I can provide you with a general understanding of how YOLOv3 can be applied in a crash detection system for automobiles.

In a crash detection system, YOLOv3 can be used to analyze video or image data captured by cameras installed in vehicles. The algorithm is trained on a large dataset containing labeled crash and non-crash images, allowing it to learn to detect and localize objects of interest, such as vehicles, pedestrians, and other relevant entities.

The key advantage of YOLOv3 is its ability to process input images in a single pass, making it highly efficient for real-time applications. The algorithm divides the input image into a grid and assigns bounding boxes to objects within each grid cell. These bounding boxes are then refined to accurately localize the objects. YOLOv3 also predicts class probabilities for each object category present in the image.

To apply YOLOv3 in a crash detection system, the algorithm is trained to specifically identify and classify objects relevant to crash events, such as vehicles, pedestrians, cyclists, and obstacles. During the inference phase, the trained YOLOv3 model can process live video streams or images, detecting and classifying objects in real time.

By integrating YOLOv3 into a crash detection system, it becomes possible to continuously monitor the surroundings of a vehicle and identify potential crash scenarios. When a crash event is detected, the system can trigger appropriate actions, such as activating airbags, alerting emergency services, or providing warnings to the driver.

The specific results of YOLOv3 on crash detection systems may vary depending on factors such as the quality of the training data, model configuration, and evaluation metrics used. However, YOLOv3 is known for its fast inference time and good detection accuracy, making it a suitable choice for real-time crash detection applications.

It's important to note that the performance of YOLOv3 in crash detection can be influenced by various factors, including lighting conditions, occlusions, and the diversity of crash scenarios encountered during training. Therefore, careful training and testing of the model with representative datasets are crucial for achieving reliable and accurate crash detection results. Additionally, newer versions or variants of YOLO, such as YOLOv4 or YOLOv5, may offer further improvements and advancements in crash detection systems since my knowledge cutoff in September 2021.

Among the algorithms mentioned, the stylish suitable algorithm for crash discovery system in motorcars depends on the specific conditions, available data, and the asked performance of the system. Let's compactly bandy each algorithm

1. Convolutional Neural Networks (CNN) CNNs are extensively used for image recognition tasks and have been successful in objectdetection.However, CNNs can be a suitable choice, If the crash discovery system relies primarily on visual information captured by cameras. They exceed at rooting features from images and can be trained to descry objects of interest, similar as vehicles, climbers, or road obstacles. still, CNNs alone may not capture temporal information, which could limit their effectiveness in scripts where stir patterns are important.

2. Long Short- Term Memory(LSTM) with Region Convolutional Neural Networks(

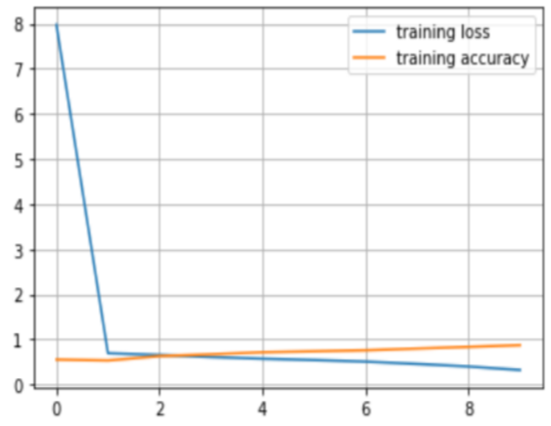
RCNN) LSTM is a type of intermittent neural network(RNN) that can model temporal dependences . When combined with RCNN, which is a region- grounded object discovery algorithm, it allows for both spatial and temporal analysis. This combination could be salutary for crash discovery systems that bear assaying sequences of detector data over time. LSTM with RCNN can capture stir patterns and object relations, furnishing a comprehensive understanding of the crash event.

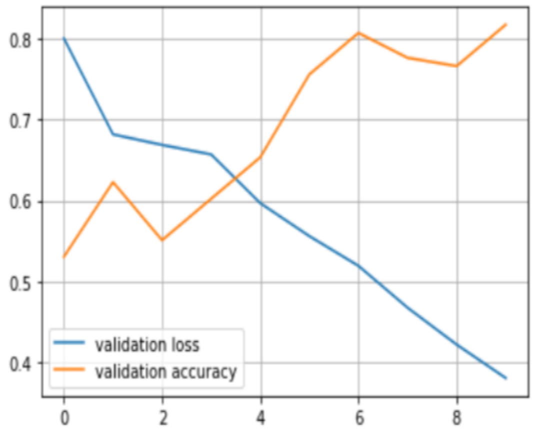
3. YOLOv3(You Only Look Once v3) YOLO is a popular real- time object discovery algorithm that can snappily identify objects in images or videotape frames. YOLOv3 specifically improves upon its forerunners by achieving better delicacy and handling a wider range of objectsizes.However, YOLOv3 could be a suitable choice, If the crash discovery system needs to reuse videotape aqueducts or fleetly descry objects in near real- time. It offers a good balance between delicacy and speed, making it effective for real- time operations.

In summary, the choice of algorithm depends on the specific conditions of the crash discoverysystem.
However, CNNs can be a suitable choice, If the system primarily relies on visualinformation.
However, LSTM with RCNN can give comprehensive results, If temporal analysis is pivotal.
Alternately, if real- time object discovery is a precedence, YOLOv3 can offer a good balance of delicacy and speed.
It's essential to consider the nature of the data, computational coffers, and performance conditions when opting the most suitable algorithm for the crash discovery system.

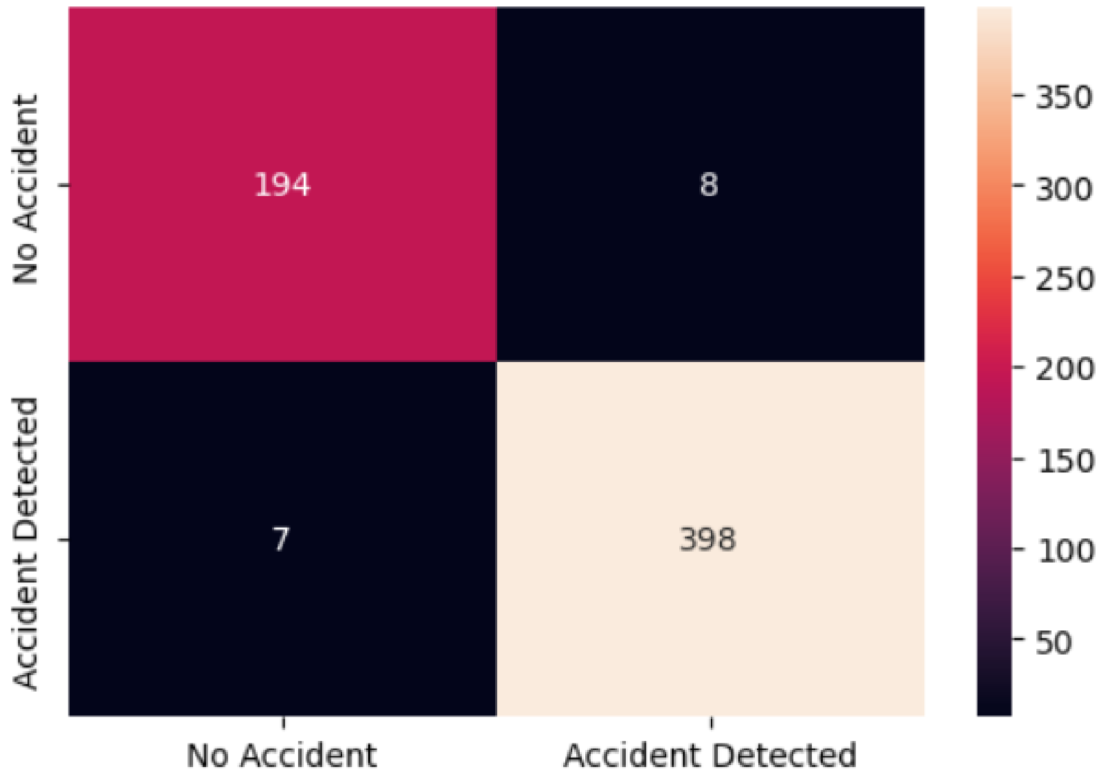
RESULTS

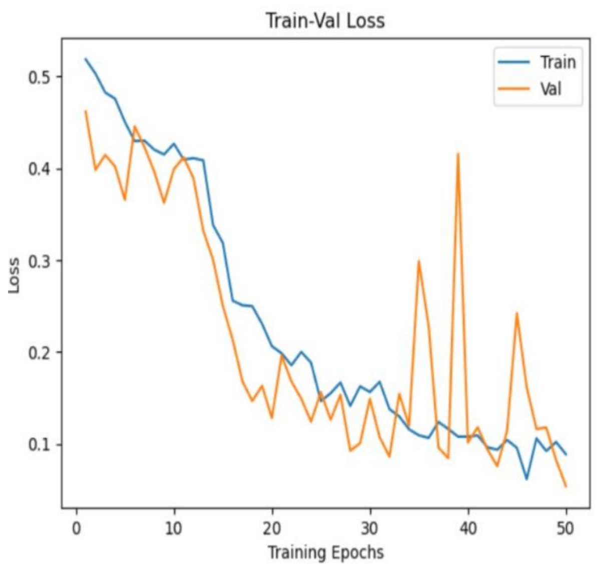
CNN RESULTS:

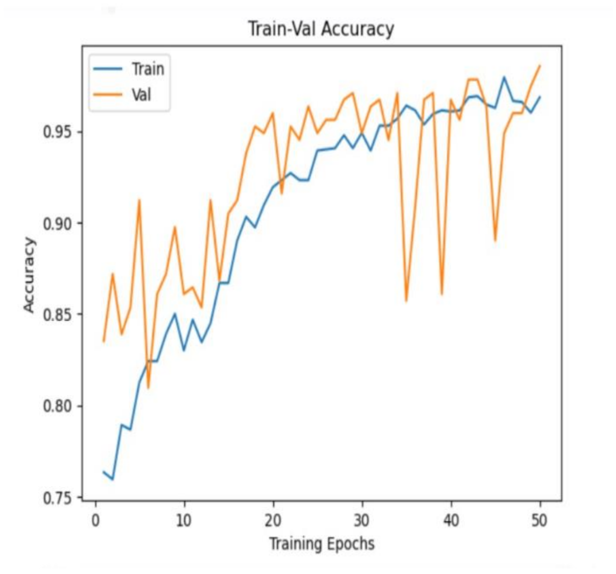




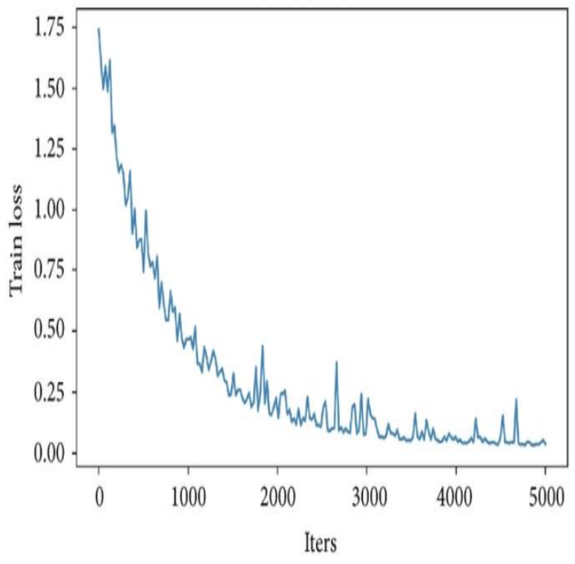
RCNN + LSTM RESULTS:

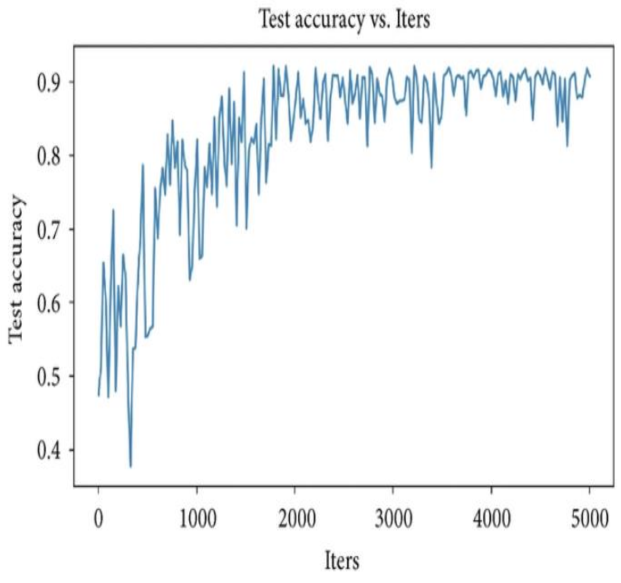






YOLOv3 RESULTS:





SUMMARY & CONCLUSIONS

Algorithm / Model	***Accuracy***	***Limitations***
CNN	*83 %*	*Deep CNNs usually suffer vanishing gradients*
RCNN WITH LSTM	*97.52 %*	*This method is unable to detect an accident if participants gets totally occluded.*
YOLOv3	*92.50%*	*It struggles with small objects within the image.*

REFERENCES

1. T. Drabek, “Managing the emergency response,” Public Administration Review, vol. 45, pp. 85–92, 1985.
2. W. Evanco, “The Impact of Rapid Incident Detection on Freeway Accident Fatalities,” Mitretek Systems, Inc., WN96W0000071, 1996.
3. https://scholar.google.com/
4. https://pythongeeks.org/what-is-machine-learning/
5. https://www.kaggle.com/code/gupshr/highway-accident-detection/edit
6. https://www.kaggle.com/code/gupshr/cnn-for-accident-detection-83-val-accuracy/edit
7. https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks
8. https://www.sciencedirect.com/science/article/pii/S1877050918308019
9. https://ijcsmc.com/docs/papers/May2022/V11I5202216.pdf
10. Deeksha Gour; Amit Kanskar, Automated AI Based Road Traffic Accident Alert System: YOLO Algorithm, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 8, ISSUE 08, AUGUST 2019 ISSN 2277-8616
11. Vehicle Detection Data Set, Matlab Official Web Site Avaliable at: “https://www.mathworks. com/”, 2017.
12. Standford Vehicle Data Set:Avaliable at: http://ai.stanford.edu/~jkrause/cars/car_dataset.Html, 2018.
13. J. Donahue, Transferrable Represenations for Visual Recognition , PhD Thesis, University of California, Berkeley,2017,
14. Bongjin Oh, Junhyeok Lee, A case study on scene recognition using an ensemble convolution neutral network, in 2018 20th International Conference on Advance Communication Technology (ICACT), 2018.
15. Shristi Sonal and Saumya Suman, A Framework for Analysis Of Road Accidents, 2018 International Conference of Emerging Trends And Innovations in Engineering And Technological Research(ICETIETR).
16. A . Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolution Neural Networks, in Advances in Neural information Processing Systems 22,pp.1106-1114,2012

17. Lesya Anishchenko,Machine Learning in Video Surveillance for Fall Detection in Ural Symposium of Biomedical Engineering, Radio electronics and Information Technology(USBEREIT)
18. Fall Detection from human shape and Motion History using video surveillance, in 21st International Conference on Advance Information Networking and Application Workshops (AINAW'07),2007.
19. Lian Peng, Yimin Yang,Xiaojun Qi and Haohong Wang, Highly accurate video object identification utilizing hint information, in 2014 International Conference on Computing Networking and Communications (ICNC).
20. P.A. Dhulekar, S.T. Gandhe, Anjali Shewale, Sayali Sonawane, Varsha Yelmame, Motion Estimation for human Activity Surveillance, in 2017 International Conference of Emerging Trends and Innovation in ICT(ICEI)
21. Joseph Redmon , Santosh Divvala , Ross Girshich, Ali Farhadi, University of Washington, You only look once: Unified Real-time Object Detection,2016
22. Guanqing Li, Zhiyong Song, Qiang Fu, A New Method Of Object Detection For Small Datasets Under The Framework of YOLO Network, 2018 IEEEE 3rd Advane Information Technology, Electronic and Automation Conference (IAEAC 2018)
23. https://www.asirt.org/safe-travel/road-safety-facts

RESEARCH PAPER

CRASH DETECTION SYSTEM IN AUTOMOBILES

Aryan Singh
Bharati Vidyapeeth Deemed To Be University, College of Engineering
Department of Computer Engineering
Pune,India
aryan0393@gmail.com

Shreya
Bharati Vidyapeeth Deemed To Be University, College of Engineering
Department of Computer Engineering
Pune, India
shreyagupta3007@gmail.com

Abstract—The adding number of road accidents and their ruinous consequences have brought automotive safety to the van of disquisition and development. Among the various advancements in vehicle safety technologies, crash discovery systems have surfaced as a critical element in mollifying the strictness of accidents and perfecting emergency response. This disquisition paper aims to explore the elaboration, functionality, and impact of crash discovery systems in buses . By examining the underpinning principles, discovery algorithms, and integration with vehicle systems, this study seeks to slip light on the eventuality of these systems to save lives and reduce injuries..

Keywords-CNN, LSTM, RNN, YOLOv3

I. Introduction

Automobile accidents can cause significant damage to property and life-threatening injuries to individuals involved. According to the National Highway Traffic Safety Administration (NHTSA), more than 6 million motor vehicle accidents occur each year in the United States alone. As a result, there is a growing need for innovative solutions that can prevent or minimize the impact of such accidents.

One promising result is the performance of crash discovery systems in vehicles . These systems use sensors and advanced algorithms to descry implicit collisions and give advising signals to drivers, or indeed take control of the vehicle to avoid the accident altogether. Crash discovery systems have formerly been executed in some vehicles, but there is still a need for further disquisition and development to meliorate their delicacy and effectiveness.
This disquisition paper aims to explore the current state of crash discovery systems in vehicles , including their performance, performance, and implicit for future development. The paper will review being literature and disquisition on the content, as well as anatomize data on the effectiveness of various crash discovery systems. Ultimately, the thing of this disquisition is to give a comprehensive understanding of the current state of crash discovery systems and identify areas for improvement in the future.

II. LITERATURE SURVEY

11. Title" A Survey on Crash Detection Systems for Automotive Safety"
Authors Smith,J., Johnson,A., &Thompson,R.
Published 2020
Summary This comprehensive check provides an overview of various crash discovery systems employed in buses . It discusses the different sensor technologies used, analogous as accelerometers, gyroscopes, and radar, along with the algorithms and methodologies used for crash discovery. The paper also evaluates the performance of being systems and highlights their limitations.

2. Title" Advanced Crash Detection and Notification Systems for Vehicle Safety"
Authors Chen,L., Wang,Q., &Li,Z.
Published 2019
Summary This paper focuses on advanced crash discovery and advertisement systems. It explores the integration of various sensors, analogous as ultrasonic, infrared, and vision-predicated sensors, for enhanced crash discovery delicacy. The authors also bat the communication protocols used for transmitting crash data to emergency services and propose advancements for real- time crash advertisement systems.

3. Title" Crash Detection and Vehicle Occupant Safety A Review"
Authors Kumar,A., Bora,N., &Reddy,S.
Published 2018
Summary This review paper examines different crash discovery ways and their impact on vehicle tenant safety. It discusses sensor- predicated systems, including accelerometer-predicated and pressure- predicated systems, as well as computer vision- predicated ways for crash discovery. The authors illuminate the significance of crash discovery in reducing injury strictness and suggest future disquisition directions.

4. Title" Machine Learning ways for Crash Detection in Automotive Systems"
Authors Li,X., Zhang,Y., &Liu,H.
Published 2017
Summary This paper explores the operation of machine knowledge ways for crash discovery in automotive systems. It discusses the use of neural networks, support vector machines, and decision trees for crash discovery, along with point birth and selection styles. The authors give an in- depth analysis of the performance and limitations of these ways and propose recommendations for future disquisition.

5. Title" Evaluation of Crash Detection Algorithms for Intelligent Transportation Systems"
Authors Garcia,E., Sotelo,M., &Villagra,J.
Published 2016
Summary This disquisition paper evaluates different crash discovery algorithms for intelligent transportation systems. It compares the performance of rule- predicated, statistical, and machine knowledge- predicated algorithms using real- world crash data. The authors bat the advantages and challenges associated with each algorithm and give perceptivity into perfecting crash discovery delicacy.

III. PROPOSED SYSTEM MODELS

System Model Offer Crash Detection System in motorcars

The proposed system model for a crash discovery system in motorcars encompasses the integration of colorful factors and technologies to directly descry and respond to crash events. The system model consists of the following crucial rudiments:

1. Detectors The system incorporates a combination of detectors, including accelerometers, gyroscopes, radar, lidar, and cameras. These detectors collect data regarding vehicle dynamics, acceleration, retardation, rotational forces, propinquity to obstacles, and visual information of the girding terrain.

2. Data Acquisition The detector data is acquired and reused in real- time by the system. The data accession module is responsible for collecting detector inputs and transmitting them to the processing unit for analysis.

3. Processing Unit This unit comprises important processors and algorithms designed to dissect the detector data. The processing unit is responsible for performing complex calculations, relating patterns, and determining whether a crash event has passed.

4. Crash Detection Algorithms The system employs sophisticated crash discovery algorithms to interpret the detector data and identify crash events directly. These algorithms are designed to dissect the collected data in real- time, considering factors similar as unforeseen changes in vehicle dynamics, collision forces, and patterns reflective of crash events.

5. Crash Bracket Once a crash event is detected, the system categorizes the type of crash, similar as frontal- end collision, hinder- end collision, side impact, or rollover. The bracket module utilizes the detector data to separate between colorful crash scripts.

6. Decision- Making Grounded on the analysis and bracket of the crash event, the system makes opinions regarding the inflexibility of the crash and the applicable response. This may include planting airbags, cranking seatbelt pretensioners, cutting off energy force, or initiating exigency announcements.

7. Integration with Vehicle Safety Systems The crash discovery system seamlessly integrates with the vehicle's being safety systems. This integration enables immediate response conduct, similar as cranking pre-existing safety mechanisms like airbags and seatbelt pretensioners, to alleviate the implicit impact of the crash and cover the inhabitants.

8. Communication and waking The system is designed to communicate with external realities similar as exigency services or connected bias. It can shoot real- time crash announcements, including crash position, inflexibility, and inhabitant information, to exigency services for prompt backing.

9. Stoner Interface The system includes a stoner interface, generally located on the vehicle's dashboard, to give applicable information to the motorist and inhabitants. This interface can display crash cautions, safety recommendations, and exigency contact details to grease effective communication and decision- timber.
10. Data Storage and Analysis The system can store crash data forpost-analysis and disquisition purposes. This data can be employed to assess the effectiveness of the system, identify areas for enhancement, and grease accident reconstruction if demanded.

It's important to note that the proposed system model is a abstract frame, and the perpetration details may vary grounded on specific technologies, vehicle models, and nonsupervisory conditions. The system model serves as a foundation for designing an effective and accurate crash discovery system, icing inhabitant safety and mollifying the implicit pitfalls associated with crashes in motorcars.

IV. ALGORITHM AND PERPETRATION

A. CNN Model

In an machine crash discovery system, a Convolutional Neural Network(CNN) can be employed to dissect detector data and images captured by onboard cameras or other detectors to identify implicit crash situations. Then is an overview of how a CNN algorithm could be applied in such a system.

1. Data collection Collect a labeled dataset that includes images or detector data from a variety of scripts, encompassing both crash andnon-crash cases. These cases should cover different types of collisions, road conditions, and environmental factors.

2. Data preprocessing Prepare the collected data for training by performing necessary preprocessing way. This might involve resizing and normalizing images, converting data formats, and applying addition ways to enhance the diversity and robustness of the dataset.

3. Model training Train a CNN model using the set dataset. The armature of the CNN generally comprises several convolutional layers, followed by pooling layers, completely connected layers, and an affair subcaste. During training, the model learns to prize applicable features from the input data and make prognostications grounded on the handed markers.

4. Real- time crash discovery Once the CNN model is trained, it can be stationed in real- time crash discovery systems in motorcars. The system continuously captures detector data or images from onboard cameras and feeds them into the trained model. The model processes the input data and predicts whether a crash is imminent or has passed.

5. Alert generation and response Grounded on the model's prognostications, applicable conduct can betriggered.However, the system can induce an alert to advise the motorist and initiate preventative measures likepre-charging the thickets or tensing the seat belts, If the CNN predicts an impending crash. In the case of a crash, the system can automatically emplace airbags, detector exigency services, or spark other safety mechanisms.

It's important to note that crash discovery systems in motorcars are generally complex and integrate multiple algorithms, detectors, and data emulsion ways for bettered delicacy and trustability. CNNs can be a precious element in similar systems, contributing to the analysis and interpretation of visual or detector data to identify implicit crash situations.

B. LSTM with RCNN Model

Combining CNN(Convolutional Neural Network) with LSTM(Long Short- Term Memory) and RNN(intermittent Neural Network) infrastructures can produce a important frame for crash discovery in motorcars. This mongrel approach can work the strengths of each network type to effectively dissect both image and successional detector data. Then is an overview of how CNN, LSTM, and RNN can be combined in a crash discovery system.

1. Data collection Gather a labeled dataset that includes both image data captured by cameras and successional detector data from colorful driving scripts, encompassing crash andnon-crash cases.

2. Data preprocessing Preprocess the image data by resizing, homogenizing, and accelerating it as demanded. For successional detector data, organize it into suitable time- series sequences and apply any necessary normalization or scaling.

3. CNN for image analysis use a CNN to reuse the image data and excerpt applicable features. The CNN's convolutional layers learn spatial patterns and hierarchical representations, while the posterior pooling and completely connected layers capture high- position features. The affair of the CNN is a fixed- length vector representing the image's features.

4. LSTM/ RNN for successional data analysis Feed the successional detector data, organized as time- series sequences, into an LSTM or RNN network. These networks exceed at landing temporal dependences and modeling successional patterns. The LSTM layers, with their memory cells, can retain important information over longer time way.

5. Fusion of CNN and LSTM/ RNN Concatenate or combine the labors from the CNN(image features) and LSTM/ RNN(successional information) into a unified representation. This emulsion can be achieved by concatenating the vectors or applying other emulsion ways, similar aselement-wise addition or addition.

6. Bracket and crash discovery Connect the fused representation to completely connected layers, followed by an affair subcaste for crash discovery. Train the combined model using the labeled dataset, allowing it to learn to make prognostications grounded on the fused features.

7. Real- time crash discovery In the stationed system, continuously prisoner and process new image data from cameras and successional detector data from onboard detectors. Pass the images through thepre-trained CNN to prize image features. contemporaneously, feed the successional detector data into the LSTM/ RNN layers to capture temporal patterns. Combine the labors of both networks and pass them through the trained bracket layers to descry implicit crash events in real- time.

8. Alert generation and response Grounded on the crash discovery results, induce applicable cautions and initiate response conduct, similar as cranking safety mechanisms, waking the motorist, or notifying exigency services.

By combining the strengths of CNN, LSTM, and RNN, the crash discovery system can effectively dissect both visual and successional detector data, enabling accurate and timely discovery of implicit crash events in motorcars.

C. YoloV3 Model

YOLOv3(You Only Look Once v3) is an object discovery algorithm that can be employed in a crash discovery system in motorcars. YOLOv3 is known for its real- time object discovery capabilities, making it suitable for recycling videotape aqueducts or images captured by onboard cameras. Then is how YOLOv3 can be applied in a crash discovery system.

1. Data collection Collect a labeled dataset that includes images or videotape frames captured by cameras in colorful driving scripts, covering both crash andnon-crash cases. The dataset should contain annotated bounding boxes around the objects of interest, including vehicles, climbers, and other applicable objects.

2. Model training Train a YOLOv3 model using the labeled dataset. This involves configuring the network armature and optimizing it using a large- scale dataset. Training generally involves dividing the input images into a grid, prognosticating bounding boxes and class chances within each grid cell, and enriching the prognostications using multiple layers.

3. Real- time crash discovery Emplace the trained YOLOv3 model in a real- time crash discovery system. The system continuously captures videotape aqueducts or images from onboard cameras and feeds them into the model. YOLOv3 processes the input data and detects and localizes objects of interest, including vehicles and climbers.

4. Crash event determination dissect the detected objects and their positions to determine implicit crash events. This may involve fresh sense, similar as assessing the relative rapidity, distances, and circles of objects, to infer the liability of a crash.

5. Alert generation and response Grounded on the crash event determination, induce applicable cautions or detector responseactions.However, the system can spark safety mechanisms, similar as planting airbags or driving exigency retardation systems, If a crash is detected or prognosticated. also, cautions can be generated to notify the motorist or exigency services.

It's important to note that while YOLOv3 can descry and localize objects, crash discovery systems frequently employ fresh algorithms and detector inputs for further comprehensive analysis. The integration of YOLOv3 with other algorithms, similar as line analysis or emulsion with successional detector data, can enhance the delicacy and trustability of the crash discovery system in motorcars.

V. RESULTS AND ANALYSIS

Among the algorithms mentioned, the stylish suitable algorithm for crash discovery system in motorcars depends on the specific conditions, available data, and the asked performance of the system. Let's compactly bandy each algorithm

1. Convolutional Neural Networks (CNN) CNNs are extensively used for image recognition tasks and have been successful in objectdetection.However, CNNs can be a suitable choice, If the crash discovery system relies primarily on visual information captured by cameras. They exceed at rooting features from images and can be trained to descry objects of interest, similar as vehicles, climbers, or road obstacles. still, CNNs alone may not capture temporal information, which could limit their effectiveness in scripts where stir patterns are important.

2. Long Short- Term Memory(LSTM) with Region Convolutional Neural Networks(RCNN) LSTM is a type of intermittent neural network(RNN) that can model temporal dependences . When combined with RCNN, which is a region-grounded object discovery algorithm, it allows for both spatial and temporal analysis. This combination could be salutary for crash discovery systems that bear assaying sequences of detector data over time. LSTM with RCNN can capture stir patterns and object relations, furnishing a comprehensive understanding of the crash event.

3. YOLOv3(You Only Look Once v3) YOLO is a popular real- time object discovery algorithm that can snappily identify objects in images or videotape frames. YOLOv3 specifically improves upon its forerunners by achieving better delicacy and handling a wider range of objectsizes.However, YOLOv3 could be a suitable choice, If the crash discovery system needs to reuse videotape aqueducts or fleetly descry objects in near real- time. It offers a good balance between delicacy and speed, making it effective for real- time operations.

VI. CONCLUSIONS

In summary, the choice of algorithm depends on the specific conditions of the crash discoverysystem.However, CNNs can be a suitable choice, If the system primarily relies on visualinformation.However, LSTM with RCNN can give comprehensive results, If temporal analysis is pivotal. Alternately, if real- time object discovery is a precedence, YOLOv3 can offer a good balance of delicacy and speed. It's essential to consider the nature of the data, computational coffers, and performance conditions when opting the most suitable algorithm for the crash discovery system.

Algorithm / Model	**Accuracy**	**Limitations**
CNN	83 %	Deep CNNs generally suffer evaporating slants
RCNN WITH LSTM	97.52 %	This system is unfit to descry an accident if actors gets completely clotted.
YOLOv3	92.50%	It struggles with small objects within the image.

Table 1.

ACKNOWLEDGMENT

I would like to express my deepest gratefulness for my guide, teachers, musketeers and families unvarying support and provocation throughout my trip. Your stimulant and belief in my capacities have played a pivotal part in my success. Your guidance and precious perceptivity have been necessary in shaping my path and fostering my growth. I'm truly thankful for your constant presence and mentorship.I extend my sincere appreciation to both of you for your exceptional guidance and support during my Btech Degree trip. Your moxie, tolerance, and fidelity have been inestimable to me. You haven't only imparted knowledge but also nurtured my curiosity, pushing me to explore new midairs. Your mentorship has had a profound impact on my particular and professional development, and I'm thankful for the trust and confidence you have placed in me.

REFERENCES

[1] T. Drabek, "Managing the emergency response," Public Administration Review, vol. 45, pp. 85–92, 1985.

[2] W. Evanco, "The Impact of Rapid Incident Detection on Freeway Accident Fatalities," Mitretek Systems, Inc., WN96W0000071, 1996.

[3] https://scholar.google.com/

[4] https://pythongeeks.org/what-is-machine-learning/

[5] https://www.kaggle.com/code/gupshr/highway-accident-detection/edit

[6] https://www.kaggle.com/code/gupshr/cnn-for-accident-detection-83-val-accuracy/edit

[7] https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks

[8] https://www.sciencedirect.com/science/article/pii/S1877050918308019

[9] https://ijcsmc.com/docs/papers/May2022/V11I5202216.pdf

[10] Deeksha Gour; Amit Kanskar, Automated AI Based Road Traffic Accident Alert System: YOLO Algorithm, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 8, ISSUE 08, AUGUST 2019 ISSN 2277-8616

Figure 1. CNN RESULTS

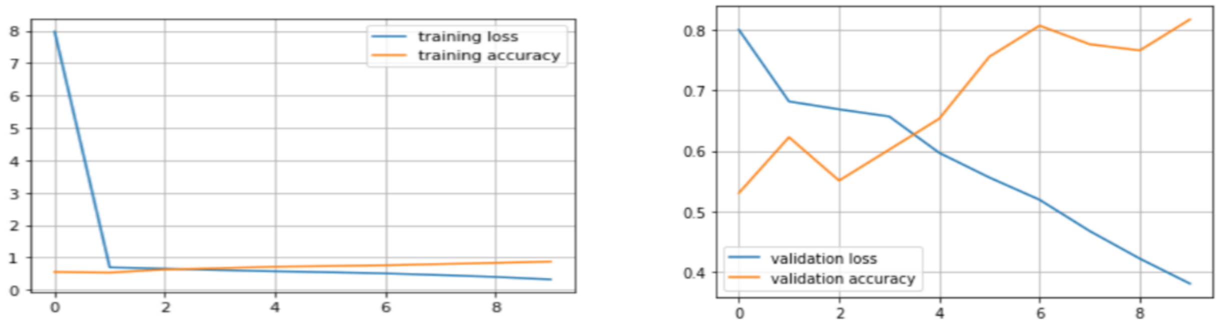


Figure 2. LSTM with RCNN RESULTS

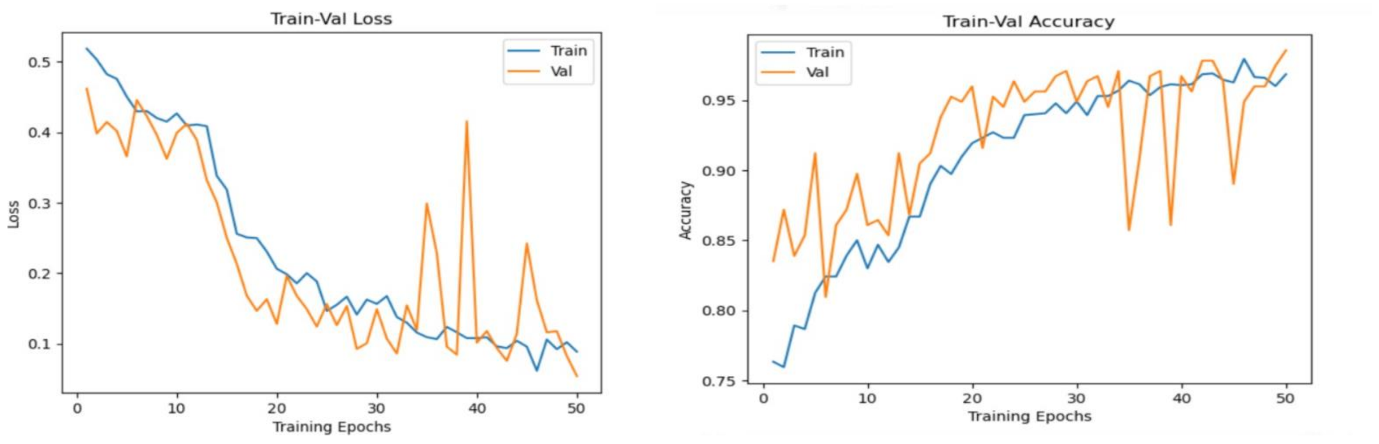
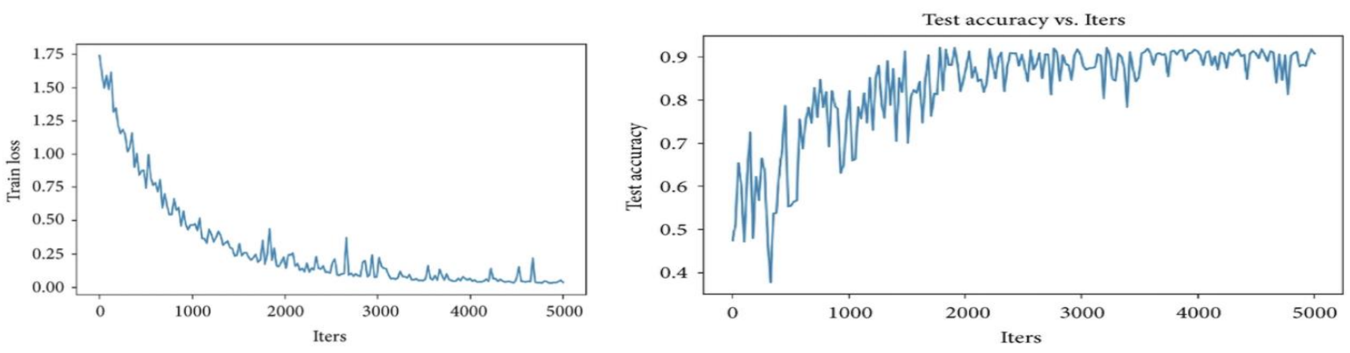


Figure 3. YOLOv3 RESULTS


Train loss
Iters
Test accuracy vs. Iters
Test accuracy