Accident Detection & Alerting System

Recent Open Source Project Lab (ITL704) project report submitted in partial

Fulfilment of the requirements of the degree of

Information Technology

by

Aashish Shetye (46) [BEIT-2]

Amaan Siddiqui (47) [BEIT-2]

Gaurav Wankhede (58) [BEIT-2]

Manish Yadav (60) [BEIT-2]

Under the guidance of

Prof. Pranoti Nage
(Designation)

Department of Information Technology
Atharva College of Engineering, Malad(W)

University of Mumbai

2022–2023

CERTIFICATE

This is to certify that the B.E. mini-project entitled **"Accident Detection & Alerting System"** is a bonafide work of **"Aashish Shetye" (46) [BEIT-2], "Amaan Siddiqui" (47) [BEIT-2], "Gaurav Wankhede" (58) [BEIT-2] and "Manish Yadav" (60) [BEIT-2] submitted** to University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"Information Technology"** during the academic year 2022-2023.

Prof. Pranoti Nage

Guide Name

Prof. Deepali Maste

Head of Department

Dr. S.P. Kallurkar

Principal

B.E. Mini-Project Report Approval

This mini-project synopsis entitled ***Accident Detection & Alerting System*** by ***Aashish Shetye, Amaan Siddiqui, Gaurav Wankhede & Manish Yadav*** is approved for the degree of ***Information Technology*** from ***University of Mumbai***.

Examiners

1.

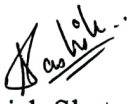
2.

Date:

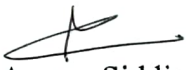
Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will cause disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



Aashish Shetye (46)



Amaan Siddiqui (47)

Gaurav Wankhede (58)

Manish Yadav (60)

Date:

Abstract

As the number of vehicles increases, road accidents are on the rise every day. According to the World Health Organization (WHO) survey, 1.4 million people have died, and 50 million people have been injured worldwide every year. The key cause of death is the unavailability of medical care at the accident site or the high response time in the rescue operation. A cognitive agent-based collision detection smart accident alert and rescue system will help us to minimize delays in a rescue operation that could save many lives. With the growing popularity of smart cities, intelligent transportation systems (ITS) are drawing major interest in academia and business, and are considered as a means to improve road safety in smart cities. This article proposed an intelligent accident detection and rescue system which mimics the cognitive functions of the human mind using the Internet of Things (IoTs) and the Artificial Intelligence system (AI). An IoT kit is developed that detects the accident and collects all accident-related information, such as position, pressure, gravitational force, speed, etc., and sends it to the cloud. In the cloud, once the accident is detected, a deep learning (DL) model is used to validate the output of the IoT module and activate the rescue module. Once the accident is detected by the DL module, all the closest emergency services such as the hospital, police station, mechanics, etc., are notified. Ensemble transfer learning with dynamic weights is used to minimize the false detection rate. Due to the dataset's unavailability, a personalized dataset is generated from the various videos available on the Internet. The proposed method is validated by a comparative analysis. The experiment results show with training, validation, and a test accuracy of 98%, respectively. To measure the performance of the proposed approach in the real world, it is validated on the toy car.

Keywords: AI; intelligent transportation systems (ITS); cognitive science; deep learning; IoT; accident detection; sensors

Table of Contents

Introduction

The demands for vehicles are increasing exponentially as the population increases. The percentage of road accidents has grown tremendously in the last few years, which is an alarming situation for everyone. According to a detailed analysis undertaken by the WHO road accidents claim the lives of millions of people each year and are the world's eighth largest cause of death. The ASIRT reports that each country spends between one and two percent of its annual budget on road accidents. The Transport Research Wing (TRW) of India surveyed road accidents in India and claimed that nearly 1230 accidents and 414 deaths were reported every day in 2019. There are multiple causes of an accident such as high speed, overtaking, using a mobile phone, weather conditions, etc. According to a recent study, over speeding is the main cause of accidents. In the rescue operation, the location of the accident spot is important. In the case of heavy traffic or a city location, emergency assistance will be available shortly, but in lowtraffic areas or highways, it is difficult to provide emergency aid on time. It is observed that significant injuries are converted into death because of delays in medical care. The victim's survival rate depends very much on how long an ambulance takes to enter the accident site and then take the victim to the hospital. Due to the continuous growth in road accidents, road safety gains considerable attention by industries and researchers.

A smart accident detection and warning system is necessary to reduce the number of deaths from road accidents. It will alert all emergency services like hospitals, police stations, etc., once the system detects an accident. A number of accident detection and alert systems have been introduced in the last few years. Most of the existing approaches use the Global Positioning System (GPS) to find the vehicles' locations. Some vehicles are even equipped with GPS, sensing the location, and sending it to the cloud. e-Notify is one of the popular accident alert systems requiring an on-board unit (OBU) in every vehicle. Because of cloud computing and Big Data, it is now possible to process the massive amounts of data that we receive from satellites, radar, and GPS systems. Even in today's world, it is possible to follow the movement of each individual object using a GPS. In the past few years, Internet of Things (IoT) and deep learning-based solutions have been extensively used to develop road safety solutions for smart cities.

Problem Statement

In emergency conditions, every second is important in saving human's life. Also, people around the event, do not take any initiative whatsoever. So to overcome this issues, we have proposed a system to detect the accident and alert nearby higher authority & hospital to receive the victim.

Objective

The desire to help and save people from unwanted incidents is the main push to drive this project towards a social welfare. It just takes seconds for disasters to happen and this project here might just help people stay safe and make it to the other end. It takes a bit of knowledge on how to connect all the modules which can make a great impact on small things.

Scope of project:

The purpose behind our project is to use each second efficiently to save a persons life and communication for timely help.

The Scope of this Project are:

- Detecting accidents efficiently.
- Finding the nearest hospital & police station.
- Sending the notifications to the hospital and police to arrive to accident spot a.s.a.p. and perform their duties..

Review of Literature

1. Accident detection using deep learning:
   The idea of an alert system which, after sensing the probability of accident, will alert the respective guardians and to medical aids by sending emergency notification or text message to nearby medical facility.

2. Automatic Traffic Accident Detection System Using ResNet and SVM:
   In order to address this problem, we propose a Machine Learning and Deep Learning based model on the concepts of Clustering and Classification that can be used to detect accidents from the traffic surveillance cameras.

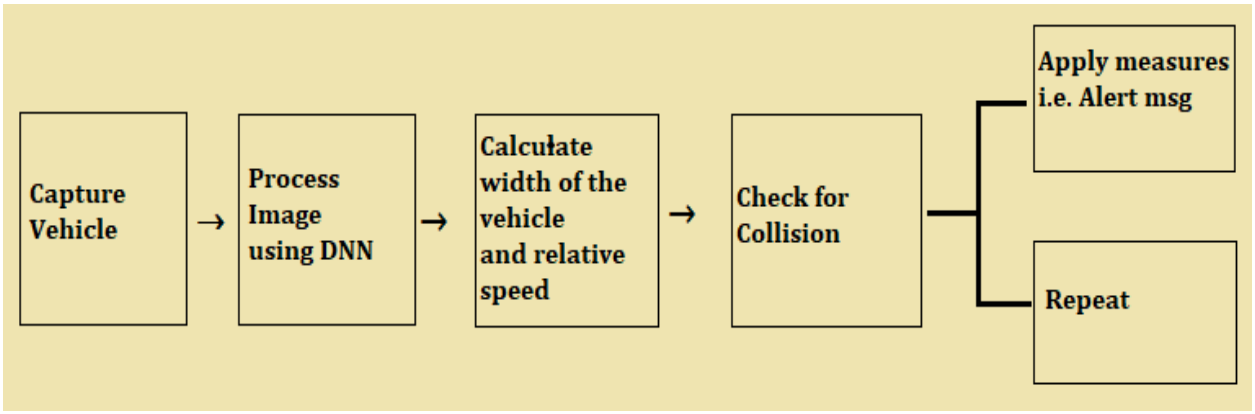
3. Automatic Traffic Accident Detection System Using ResNet and SVM
   The results show that while deep learning models are performing well in detecting crashes, combination of three deep learning models (i.e. LSTM, GRU & DNN) using Multilayer Perceptron (MLP) and Random Forest Classifier (RFC) can improve detection performance. Interestingly, combination of these models achieved the highest detection rate and lowest false alarm rate values, respectively, among all the studied models.

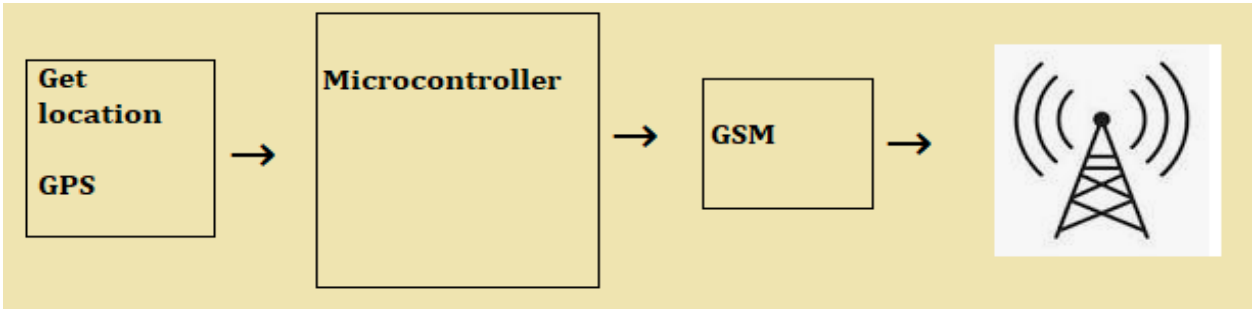
Proposed System

- The proposed model contains CNN layers for the classification of continuous video captured from the camera.
- The video stream from CCTV is directly sent to a preprocessing part. This preprocessing is carried out by openCV library, which converts the video into frames of images. These images will be formatted in order to be compatible with the CNN model.
- This model is trained using the collected datasets. The training is conducted in a supervised learning fashion where the example data sets along with its class value is fed into the model for its learning. Example, dataset categorized as accidents.
- For alerting system, we will be using Web API Services.
- One of the main causes of road accidents is the speed of vehicles on the road. Most of the existing accident detection system mainly identifies an accident based on the speed sensor.
- If the vehicle's speed is changed suddenly and exceeds the predefined threshold, then the system detects an accident. So, the false detection rate of these systems can be high, because in various circumstances, the vehicle's speed can be changed, such as a speed breaker, an obstacle on the road, a technical problem of the vehicle, etc.
- The proposed accident detection system uses the deep learning technique to minimize this misidentification rate, which takes the input video from the dashboard's camera.

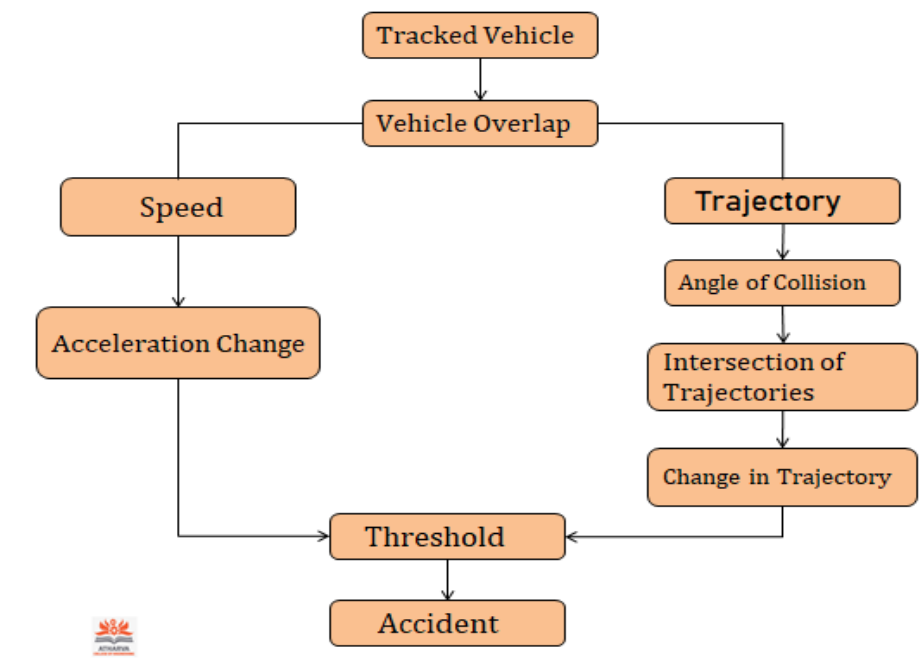
Block Diagram

Existing System:





Proposed System:



Implementation

Two steps are involved in the deep learning module of the proposed Accident Detection System. These steps are:

1. Dataset:
   The quality and amount of the dataset have the greatest influence on the performance of the model. We converted video into frames using the CV2 Python library. The entire dataset is split into two parts: training and testing. Training data are used to train the model, while test data are used to measure the model's performance at test time. Training and test data are balanced for both the classes to avoid overfitting. The number of images in our dataset is between 500-600. Image augmentation techniques are frequently used to increase the size of a dataset, which generates more numbers of images by applying various operations on the images such as rotation, zoom out, zoom in, sharping, etc.
2. Customization and Training of the Pre-Trained Model:
   We are employing a pre-trained model due to the limited amount of data available. In addition to learning hidden patterns from the huge dataset, the pre-trained models have already been trained on a large dataset. In this work, we have trained two well-known pre-trained models called ResNet and InceptionResnetV2. Both the pre-trained models have been trained on large datasets which include 1.2 million images and are capable of categorizing an image into 1000 different categories. This dataset is being made available as part of a competition named the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). All pre-trained models are composed of a large number of poling layers (PL) and convolution layers (CL).

Pseudo code

Capturing the video from given path and extracting frames for pre processing.

In [4]:
```
count = 0
videoFile = "Accidents.mp4"
cap = cv2.VideoCapture(videoFile)   # capturing the video from the given path
frameRate = cap.get(5) #frame rate
x=1
while(cap.isOpened()):
    frameId = cap.get(1) #current frame number
    ret, frame = cap.read()
    if (ret != True):
        break
    if (frameId % math.floor(frameRate) == 0):
        filename ="%d.jpg" % count;count+=1
        cv2.imwrite(filename, frame)
cap.release()
print ("Done!")
```

```
Done!
```

Pre processing & training the model

In [10]:
```
from keras.applications.vgg16 import preprocess_input
X = preprocess_input(X,data_format=None)
```

In [11]:
```
from sklearn.model_selection import train_test_split
X_train, X_valid, y_train, y_valid = train_test_split(X, dummy_y, test_size=0.3, random_state=42)
```

Epochs indicates the no of passes of entire training dataset the ML algo has completed

In [19]:
```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [20]:
```
model.fit(train, y_train, epochs=100, validation_data=(X_valid, y_valid))
```

```
Epoch 1/100
5/5 [==============================] - 1s 143ms/step - loss: 0.9461 - accuracy: 0.5355 - val_loss: 0.8127 - val_accuracy: 0.6
269
Epoch 2/100
5/5 [==============================] - 1s 107ms/step - loss: 0.4157 - accuracy: 0.8065 - val_loss: 0.7974 - val_accuracy: 0.7
015
Epoch 3/100
5/5 [==============================] - 1s 106ms/step - loss: 0.2559 - accuracy: 0.9032 - val_loss: 0.8554 - val_accuracy: 0.6
866
Epoch 4/100
5/5 [==============================] - 1s 106ms/step - loss: 0.1436 - accuracy: 0.9548 - val_loss: 0.8230 - val_accuracy: 0.7
015
Epoch 5/100
5/5 [==============================] - 1s 104ms/step - loss: 0.1165 - accuracy: 0.9677 - val_loss: 0.7933 - val_accuracy: 0.7
313
Epoch 6/100
5/5 [==============================] - 1s 103ms/step - loss: 0.0746 - accuracy: 1.0000 - val_loss: 0.7947 - val_accuracy: 0.7
612
Epoch 7/100
```

In [22]:
```
count = 0
videoFile = "Accident-1.mp4"
cap = cv2.VideoCapture(videoFile)
frameRate = cap.get(5) #frame rate
x=1
while(cap.isOpened()):
    frameId = cap.get(1) #current frame number
    ret, frame = cap.read()
    if (ret != True):
        break
    if (frameId % math.floor(frameRate) == 0):
        filename ="test%d.jpg" % count;count+=1
        cv2.imwrite(filename, frame)
cap.release()
print ("Done!")
```

```
Done!
```

Capturing the testing video for prediction purpose

In [26]:
```
# preprocessing the images
test_image = preprocess_input(test_image, data_format=None)

# extracting features from the images using pretrained model
test_image = base_model.predict(test_image)
test_image.shape
```

Out[26]:
```
(9, 7, 7, 512)
```

Prediction testing

In [30]:
```
for i in range (0,9):
    if predictions[i][0]<predictions[i][1]:
        print("No Accident")
    else:
        print("Accident")
```

```
No Accident
No Accident
No Accident
No Accident
No Accident
No Accident
Accident
No Accident
Accident
```

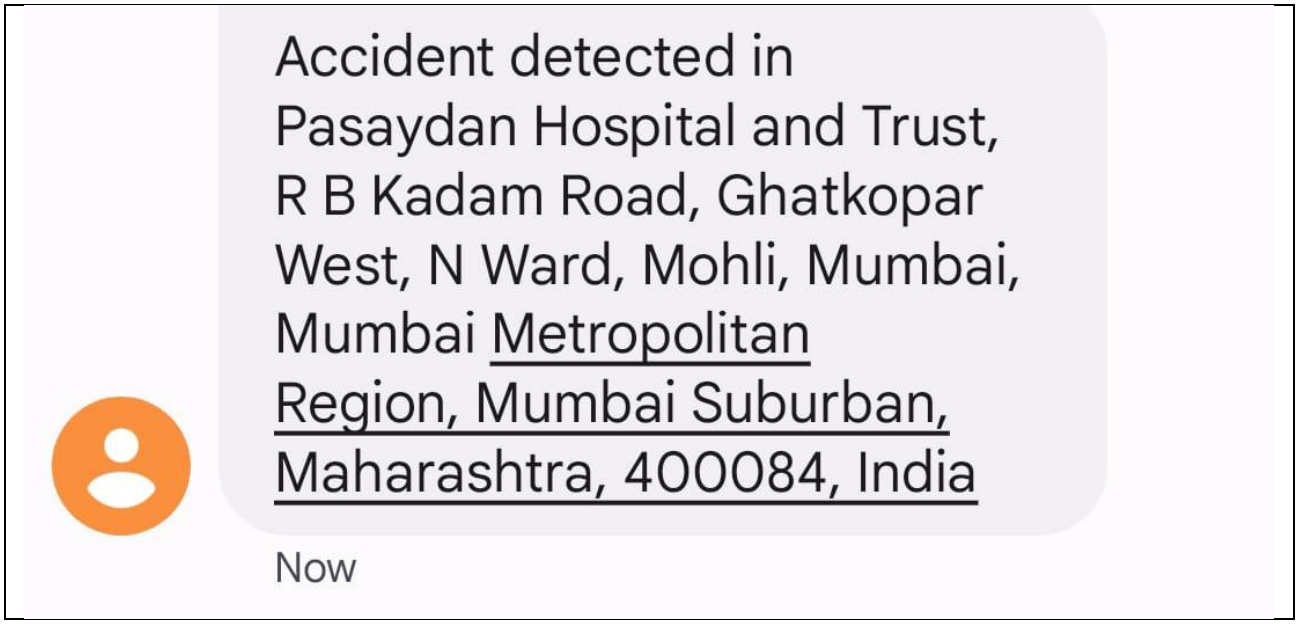
Real time testing of video & predicting the results. Also, on accident detection an alert message is send to nearest hospital and police authority.

In [31]:
```
geoLoc = Nominatim(user_agent="GetLoc")
g = geocoder.ip('me')
locname = geoLoc.reverse(g.latlng)
account_sid = "ACf8662654f30704c3ba0a616d34900496"
auth_token = "76745f038b5544252538496300387 7b5"
client = Client(account_sid, auth_token)
```

In [50]:
```
cap = cv2.VideoCapture(r'C:/Users/Admin/ACCIDENT-DETECTION-WITH-A-REPORTING-SYSTEM/Accident-1.mp4')
i=0
flag=0
while(cap.isOpened()):
    ret,frame=cap.read()
    if ret==True:
        if predictions[int(i/15)%9][0]<predictions[int(i/15)%9][1]:
            predict="No Accident"
        else:
            predict="Accident"
            flag=1
        font = cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(frame,
                predict,
                (50, 50),
                font, 1,
                (0, 255, 255),
                3,
                cv2.LINE_4)
        cv2.imshow('Frame', frame)
        i=i+1
        if cv2.waitKey(50) & 0xFF == ord('q'):
            break
#         cv2.imshow('Frame', frame)
    else:
        break
if flag==1:
    client.messages.create(
                 body="Accident detected in "+locname.address,
                 from_= +14783752797,
                 to= +917506266617 )

# release the cap object
cap.release()
# close all windows
cv2.destroyAllWindows()
```

Result


Accident detected in Pasaydan Hospital and Trust, R B Kadam Road, Ghatkopar West, N Ward, Mohli, Mumbai, Mumbai Metropolitan Region, Mumbai Suburban, Maharashtra, 400084, India
Now

Conclusion

In the smart city, due to the good quality of the roads, drivers run their vehicles at high speeds, resulting in an increase in road accidents. Although a range of accident detection and prevention systems are being launched on the market, many fatalities still occur. At least some of the issue is exacerbated by an insufficient automatic identification of accidents, ineffective warning, and emergency service responses. This work is carried out in which a deep learning-based model is used to validate the training model and perform the rescue operation. Once an accident has been identified which measures the impact on the car and a GPS module for the vehicle speed, it transfers all useful information to the cloud. In the second phase, pre-trained models, are used to minimize the false detection rate and activate the rescue module. If an accident is detected by the deep learning module, a rescue module is activated and the details are sent to the nearest police station, hospitals, and relatives. Experiment results show that ResNet-50 is a more suitable model for accident detection than the InceptionResNetV2, as it has a higher test accuracy and recall rate of the accident class. Although an in-vehicle accident detection system provides emergency responders with essential information as fast as possible but unavailability of this system are restricted by their non-portability and costs. The proposed vehicle accident detection system can track an accident at its moment of occurrence. Compared with other deployment systems composed of expensive sensors and unnecessary hardware, the proposed system is more economical, more reliable, and more accurate than similar systems, mainly due to the model-based approach.

FUTUREWORK

In the future, we can combine both supervised and unsupervised methods together to improve the system . We can use supervised learning models to identify the accidents from the frames which are flagged anomalously by unsupervised models. More sophisticated methods for tracking accident happened area. Algorithms like kelman can be used to predict the potential of a vehicle passing through the accident happened place. This will help the driver to avoid incident happen area and evade a potential traffic jam.

The proposed model can help us to minimize the death rate due to the unavailability of emergency services at the accident location. The model has zero false positives during the training time, and extremely low false positives at the test time. The proposed model does not consider the security aspect, so we intend to address this issue in future work. In addition, some driver alert systems such as the drowsiness detection module can also be added in the proposed model.

Reference

1. D. K. Yadav, Renu, Ankita and I. Anjum, "Accident Detection Using Deep Learning," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020, pp. 232-235, doi: 10.1109/ICACCCN51052.2020.9362808.
2. A. K. Agrawal et al., "Automatic Traffic Accident Detection System Using ResNet and SVM," 2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), 2020, pp. 71-76, doi: 10.1109/ICRCICN50933.2020.9296156.
3. Homa Taghipour, Amir Bahador Parsa, Rishabh Singh Chauhan, Sybil Derrible, Abolfazl (Kouros) Mohammadian,
   A novel deep ensemble based approach to detect crashes using sequential traffic data,
   IATSS Research,
   Volume 46, Issue 1,
   2022,
   Pages 122-129,
   ISSN 0386-1112,
   https://doi.org/10.1016/j.iatssr.2021.10.004.
4. Pathik N, Gupta RK, Sahu Y, Sharma A, Masud M, Baz M. AI Enabled Accident Detection and Alert System Using IoT and Deep Learning for Smart Cities. *Sustainability*. 2022; 14(13):7701. https://doi.org/10.3390/su14137701