International Journal of Autonomous and Adaptive Communications Systems

ISSN online: 1754-8640 - ISSN print: 1754-8632
https://www.inderscience.com/ijaacs

IoT-based vehicular accident detection using a deep learning model

Ishu Rani, Bhushan Thakre, K. Jairam Naik

DOI: 10.1504/IJAACS.2025.10046127

Article History:

Received:	13 August 2021
Accepted:	11 November 2021
Published online:	10 January 2024

IoT-based vehicular accident detection using a deep learning model

Ishu Rani, Bhushan Thakre
and K. Jairam Naik*

Department of Computer Science and Engineering,
National Institute of Technology Raipur,
Raipur – 492010, Chhattisgarh, India
Email: Ishuraniugdia@gmail.com
Email: Bhushanthakre4141@gmail.com
Email: Jnaik.cse@nitrr.ac.in
*Corresponding author

Abstract: With the increase in population and running valuable time, the demand for cars has skyrocketed creating an unprecedented condition in spite of traffic risks and road collisions. The crashes are growing at an unprecedented pace; hence, they cause death. Since Machine Learning has taken over, previously complex problems have become feasible due to the promising real-life applications of these models. A learning model that learns over an image dataset, thereby classifying never-before-seen images and data based on the level of damage, has been proposed in this paper. The artificial neural network is used to train the model and to learn the similarities among images and accident data. The proposed solution is efficient as it was tried to improve the efficiency and accuracy of finding the polarity of images for the same order of dataset as compared to the existing work.

Keywords: vehicles; accident detection; classification; accuracy; deep learning; IoT; Internet of Things; training model; image polarity.

Reference to this paper should be made as follows: Rani, I., Thakre, B. and Naik, K.J. (2024) 'IoT-based vehicular accident detection using a deep learning model', *Int. J. Autonomous and Adaptive Communications Systems*, Vol. 17, No. 1, pp.1–23.

Biographical notes: Ishu Rani received her BTech in Computer Science and Engineering from National Institute of Technology Raipur, India in 2021. Her research interest includes cloud computing, IoT, deep and reinforcement learning.

Bhushan Thakre received his BTech in Computer Science and Engineering from National Institute of Technology Raipur, India in 2021. His research interest includes cloud, IoT, deep learning and reinforcement learning.

K. Jairam Naik is working as an Assistant Professor in the Department of Computer Science and Engineering, National Institute of Technology Raipur, India since 2018. He is having 18 years of teaching and research experience.

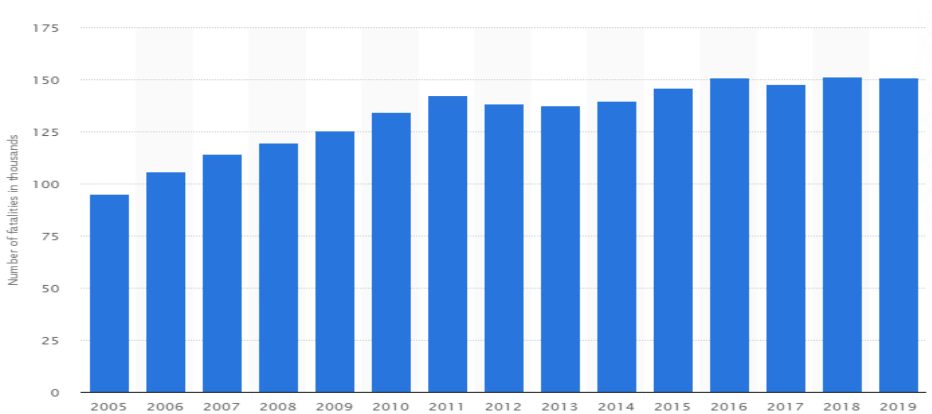
He received PhD degree in Computer Science and Engineering from Jawaharlal Nehru Technological University Hyderabad, India in 2017. He received MTech and BTech degrees in Computer Science and Engineering from JNTU Hyderabad in 2009 and 2003 respectively. His research of interest includes grid and cloud computing, edge/fog and IoT computing.

1 Introduction

1.1 Overview

According to Times of India, approximately 150,000 peoples were losing lives per year in road accidents as shown in Figure 1. Also, this constitutes 30% of total death as represented in Figure 2 that happened in India. We can see that a high percentage of people died because of road accident. The authorities and government could not able to find a proper solution to stop these accidents. Most deaths were from countries with low and middle income with many dangers for on-the-road passengers, bicycles and motor vehicles. With the goal to improve it, Traffic management study has made many efforts Canters for improving prediction and accidents detection technologies in traffic. European study, for example ADVISOR (Webster and Williams, 1992) and (Naylor and Attwood, 2003) involving technical and industrial research institutes detecting anomalies and reducing deaths from traffic, and to improve the condition of roads.

Figure 1 Distribution of death due to road accident across India from 2005 to 2019 (see online version for colours)



Source: Sun (2021)

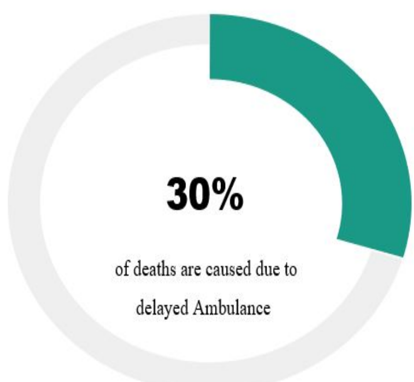
As shown in Figure 1 that after various strategies and steps of various organisation, there is continuous increase in the death caused by road accident from 2005 to 2019. Traffic collisions are causing traffic disturbance, disruption of traffic, and serious urban issues worldwide. At times, serious collisions can result in loss, injury and even death that may be irreparable. The National Highway Traffic Safety Authority (NHTSA), which produces annual traffic safety data reports, states that since 1988 more than 5,000,000 vehicle accidents have occurred in States per year (NHTSA, 2015).

After many years of study, substantial injury impact reductions can be accomplished using reliable detection techniques and appropriate reaction mechanisms have been generally recognised. The precise and rapid identification of road accidents are important as a key component in the management of traffic incidents. Our paper aims to resolve all the issues and limitation which is present in previous solutions and approaches. We aim to develop a Machine Learning Model that identifies the vehicular accidents with greater efficiency and classifies them based on the level of damages sustained through video footage captured by a high resolution, motion-sensitive camera based on Internet of Things (IoT).

1.2 Objective and importance of the work

Millions of people, all over the globe lose their lives annually due to road accidents. Statistics implies that 30% as in Figure 2 of dead could be saved if lifesaving treatment is meted out within the crucial hour. Most collisions result in minor injuries, and victims' lives will be spared if they are rescued quickly. However, due to inadequate coordination mechanisms, it takes an extra delay to manually alert the rescue teams, resulting in patients being left unattended for an extended period of time, resulting in an elevated death risk. There have been profound efforts undertaken by various NGOs and voluntary organisations to mitigate fatality rate by dispensing the necessary in-time services. This model could come in handy to alert the in-time service carriers.

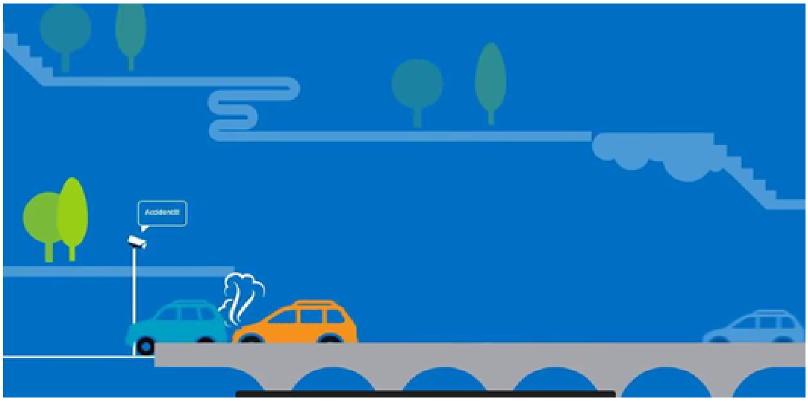
Figure 2 Approximately 150,000 people are killed per year in road accidents (see online version for colours)



Source: Dash (2020)

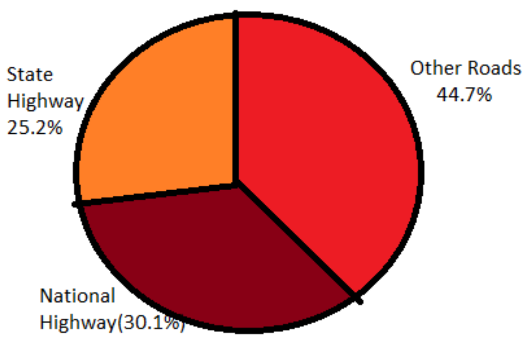
We aim to develop a Machine Learning Model that identifies the vehicular accidents with greater efficiency and classifies them based on the level of damage sustained through video footage captured by a high resolution, motion-sensitive camera based on IoT as shown below in a pictorial view in Figure 3.

Figure 3 Proposed solution in pictorial view (see online version for colours)



Scope of the work: According to "Ministry of Road Transport and Highways", number of accidents in road categories in 2018 is shown in Figure 4. We can see a greater number of accidents are happening in road side area. The reason behind this many deaths are delay of treatment provided to victims so, our aim is to build to a Machine Learning Model that identifies the vehicular accidents with greater efficiency and classifies them based on the level of damage sustained through video footage captured by a high resolution, motion-sensitive camera based on IoT.

Figure 4 Share of road categories in road accident during year 2018 (see online version for colours)



Source: Digital Editor (2019)

Hence, using this project we are trying to provide a solution to problem which is very crucial to save human lives and resources. This type of application can help tremendous of people by saving their life and life of their loved ones. This application has a great importance in future and can be developed further to automatically call the service provider like ambulance, doctors, police etc. based upon level of damage caused by accident as many people who have witnessed the accident also not come forward to help in these scenarios.

The most important features of Convolutional Neural Network led us to this solution are:

Sparse interactions: The kernel or function detector is less than the input image to implement sparse interaction or sparse weights.

Parameter sharing: The parameter or weight number used in CNN is regulated. By sharing CNN parameters, the number of parameters to be learned is cut and computational requirements are reduced.

Equivariant representation: This means that target recognition is invariant to lighting changes, location changes, but the internal representation is equivalent to such changes.

2 Literature review

2.1 Existing work

A number of CNNs have been proposed earlier. As we slowly begin to understand the shortcomings with a particular model, we try to minimise the effect of those short comings, which in turn, gives rise to newer extensions to the already established methods, creating methods that eke out efficiency and complexity. The work in Li et al. (2021) discussed detailed survey of the CNN including applications, classic networks, building blocks, related functions and prospects. The next work in Shah et al. (2018) proposed the dataset CADP – CADP dataset for Traffic Camera based accident forecasting. The advantages and limitations of other datasets are taken up at a greater length.

The work in Wang et al. (2020) and Ijjina et al. (2019) mentions three categories of study:

1 traffic flow modelling

2 modelling of vehicle collisions

3 activity analysis.

Various image enhancing features could mitigate these effects. But, then, the complexity of the image-processing model increases manifold. In subsequent work (Routh et al., 2019) used accelerometer in car alarm application. The main advantages are higher sensitivity and accuracy is indeed achieved using this project, user friendly and reliable. The work in Lee and Shin (2019) used ODTS and deep learning methodology. Its main limitation is its efficiency and further in literature (Alvi et al., 2020) describes and IoT based accident detection as comparative study. However, it is possible that it is less embraced by consumers due to its protection and privacy problems.

In the work (Bhatti et al., 2019) using advanced smart phone criteria, a low-cost approach can be developed and implemented in legacy vehicles for improved transport systems and (Tian et al., 2019) uses latest technology of CVIS and machine vision with model accuracy of 90.02%. The next the work (Ali and Eid, 2015) uses fuzzy logic as a mobile device decision help, analyses incoming sensor data and makes a choice based on the rules. Any kind of accident detected is automatically sent as an alert to the required destination as stated in Madhusan et al. (2016). The next work (Ijjina et al., 2019) proposed framework offers a rigorous approach for achieving a high level of detection and low false alarm on CCTV surveillance footage of general transport.

Today, most mobile phones are equipped with a variety of sensors, including accelerometers, GPS devices, microphones, etc. to provide an opportunity for people to use these devices as both vehicle accident sensors and remote event notifications as discussed in paper (Aldunate et al., 2013; Darwin et al., 2020; Sharma and Sebastian, 2019) using GPS and WIFI, this machine would transmit a short message or Wi-Fi via Internet to WhatsApp when the accident happens.

An approach for minimising total run-time and reducing the deadline misses for a connected car system through efficient load balancing in a fog computing environment was proposed in Naik and Naik (2020). This work motivates the traffic load balancing on the roads through which the frequency of accidents can be minimised. Additionally, management of workflow in Cloud-fog computing environment for tolerating the processing delay proposed by Naik (2020a) and Classification and scheduling the information-centric IoT applications in the same environment proposed by Naik (2020b) was motivated us to incorporate their ideas to monitor and manage the accident-free traffic on the road.

Now, we are proposing our solution as learning Model that learns over an image dataset, thereby classifying never before seen images and data. Our solution performs better because the data used to train the model used in the existing literature (Shah et al., 2018) may have been an imbalanced dataset that may lead to poor results. We took care of the data being balanced and clean so that the model may perform better and produce better results with respect to the results shown in the existing literature. Learning rate also plays a major role when it comes to training a model. It is a widely accepted fact that a model is as good as it is trained. So, if two models having same internal architecture are trained simultaneously on same data and other hyperparameters set to same values and only learning rate is varied between the two. Then the model with the precise learning rate will be trained with less cost function, giving better results overall.

Several methods that are relevant to the domain of study and uses IoT based vehicular accident detection using deep learning model and machine vision are described in Table 1.

Table 1 Table for existing work

Author	*Paper title*	*Method used*	*Description*
Li et al. (2021)	A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects	CNN including applications, classic networks, building blocks,	It ranks approximation, dimension, reduction, and faster computing speed. But, poor crowded scene result, lack of equivariance, low generalisation, which makes CNN hard to handle
Shah et al. (2018)	Accident Forecasting in CCTV Traffic Camera Videos	Faster R-CNN and context mining	CADP dataset right from the development to the results and its future scopes. But score obtained is not improved with the increased context mining
Wang et al. (2020)	A Vision-Based Video Crash Detection Framework for Mixed Traffic Flow Environment Considering Low-Visibility Condition	Computer Vision	Image enhancing features could mitigate extra Effects. But, High complexity partly due to the numerical complexity of the latency model

Table 1 Table for existing work (continued)

Author	*Paper title*	*Method used*	*Description*
Shah et al. (2018)	A Novel Approach to Automatic Road-Accident Detection using Machine Vision Techniques	Five support vector machines trained with HOG and GLCM features	The overall system accuracy for DCD–1 was 81.83%. But the detection was done based on various assumptions
Ijjina et al. (2019)	Computer Vision-based Accident Detection in Traffic Surveillance. (2019)	Neoteric framework	Robust method for fast identification and low falsifying alarm. But, after overlap with other cars, the likelihood of a crash is dependent on speed and direction errors in a car
Lee and Shin (2019)	Automatic Vehicle Accident Detection and Messaging System	The dangerous driving can be detected using accelerometer in car alarm application	Higher sensitivity and accuracy are indeed achieved using this project, user friendly and reliable. But, less efficient, high operating cost, hard to maintain
Routh et al. (2019)	An Application of a Deep Learning Algorithm for Automatic Detection of Unexpected Accidents Under Bad CCTV Monitoring Conditions in Tunnels	Deep Learning technology and Object Detection and Tracking System	This method enables a move object to be tracked in time, which in CNN is not common. But, less efficient, less accurate
Alvi et al. (2020)	A Comprehensive Study on IoT Based Accident Detection Systems for Smart Vehicles	Various ML techniques and GPS/GSM	It mitigates traffic accidents, determines accurate crash sites and facilitates all relief activities. But, Privacy and security issues, so less adopted by users
Bhatti et al. (2019)	A Novel Internet of Things-Enabled Accident Detection and Reporting System for Smart City Environments	Advanced specifications of Smartphones	Low-cost model customised Android application. But, Privacy and security issues, so less adopted by users
Tian et al. (2019)	An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems	Machine Vision and CVIS	Detection time is 0.0461 seconds with 90.02% average precision. But low vehicle smart system penetration and vehicle shielding effects

2.2 Summarisation of literature

Based on the literature described in Table 1, there are several issues that are to be addressed were identified.

When it comes to accident detection application, it is crucial to have the below properties:

i *Good accuracy*: The application must give results accurately and that's possible when you have trained the model with more and more dataset.

ii *Must be user friendly*: The system must be less complex and easier to use and it should have low maintenance cost.

iii *Privacy*: While two approaches can produce identical results, it must be emphasised that, the approach with more secure and private, lower cost of installation, development, operation, and maintenance must be used.

iv *Reliable and secure*: It should implement appropriate security measures to ensure that the system performs nominally under extreme circumstances. The ability to hack the system should be reduced.

In the next section we propose a learning model that learns over an image dataset, thereby classifying never before seen images and data. We aim to classify these real-time accidents based on the level of damage. We will make use of the artificial neural network to train the model to learn the similarities among images and accident data.

3 Proposed methodology

3.1 System architecture

As you can see in Figure 5, firstly we prepare a dataset for training our model which can find the polarity of the input image. We take Positive dataset in the form of the video from CADP dataset and negative dataset from DETRAC dataset. CADP contains videos containing road accidents and DETRAC contains normal traffic videos. These videos are then converted into vectors of frames by capturing video instances in specified intervals. Then to reduce the dimensionality and complexity, we further convert those frames from 3D RGB colour scale to 2D greyscale. Then we train our model by feeding the dataset with faster RCNN where region proposal is extracted where the probability of finding required objects like cars, buses bikes are more and then we compute CNN features and classify the scenario of the frames.

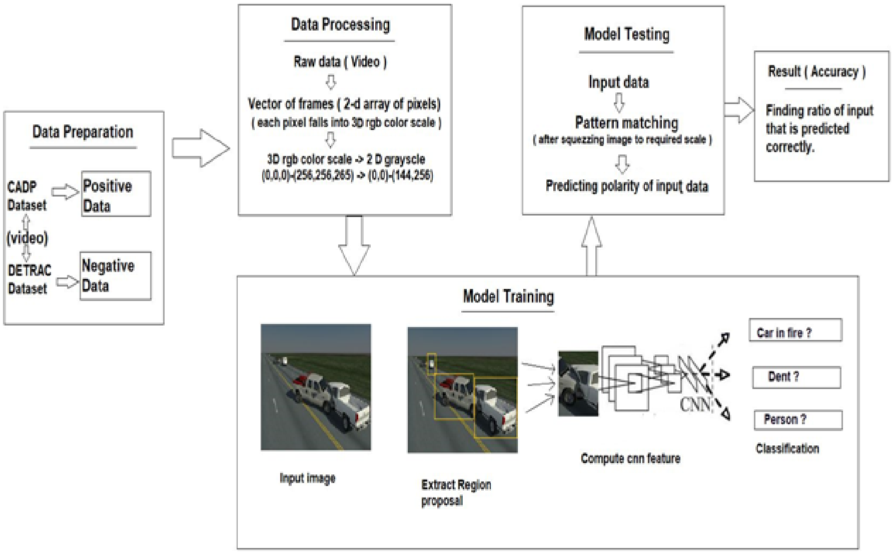
When model training is done, we test our model by feeding an unknown dataset to it. dataset can be a video or a vector of frames depending upon the type of input available. If the dataset is video, then we first convert our video into a vector of frames. After squeezing the images into required complexity, we predict the polarity of the dataset. For finding the accuracy of our model, we either test our dataset against some best existing model which is already trained with very huge dataset or we can pick the dataset such that whose outcome is known to us, so that we can compare the calculated polarity of the dataset.

Using the multi-task loss in Faster R-CNN, we try to minimise an objective function (Girshick, 2015). Our loss function for an image is defined as:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum L_{cls}\left(p_i, p_i^*\right) + \lambda \frac{1}{N_{reg}} \sum p_i^* L_{reg}\left(t_i, t_i^*\right) \tag{1}$$

Here, i = index of an anchor in a mini-batch, P_i = predicted probability of anchor i being an object.

Figure 5 Proposed system architecture (see online version for colours)



The ground-truth label $p * i$ is 0 if the anchor is found negative, and is 1 if the anchor is found positive. t_i is a vector representing the 4 parameterised coordinates of the predicted bounding box, and $t * i$ is that of the ground-truth box associated with a positive anchor (Ren et al., 2017).

For bounding box regression, we adopt the parameterisations of the 4 coordinates following (Girshick et al., 2014):

$$\left.\begin{aligned} t_x &= \frac{(x-x_a)}{w_a}, t_x = \frac{(y-y_a)}{h_a} \\ t_w &= log\left(\frac{w}{w_a}\right), t_h = log\left(\frac{h}{h_a}\right) \\ t_x^* &= \frac{(x^*-x_a)}{w_a}, t_y^* = \frac{(y^*-y_a)}{h_a} \\ t_w^* &= log\left(\frac{w^*}{w_a}\right), t_h^* = log\left(\frac{h^*}{h_a}\right) \end{aligned}\right\} \tag{2}$$

where x, y, w, and h are the box's canter coordinates and its width and height. Variables x, x_a, and x^* are for the predicted box, anchor box, and ground truth box respectively (likewise for y, w, h). This is considered a regression of the binding box from an anchor box to the nearest box.

3.2 Key improvements over existing literatures

Our proposed architecture has enhanced performance and prediction accuracy as compared to the traditional method given in the existing literature because of the following two reasons:

1. The data used to train the model in the existing research has an imbalanced dataset that may lead to poor results. Hence, in this work, it has been taken care of the data being balanced and clean so that the model may perform better and produce improved results with respect to the results shown in the existing literature.
2. Learning rate also plays a major protagonist when it comes to training a model. It is a widely accepted fact that a model is as good as it is trained. So, if two models having same internal architecture are trained simultaneously on same data and other hyperparameters set to same values and only learning rate is varied between the two. Then the model with the precise learning rate will be trained with less cost function by giving better results overall.

3.3 Algorithm and flow of information

The entire procedure of the work can be broken down broadly into three components, namely, creating dataset, training model and testing model.

In Algorithm 1, we are creating a dataset for our model. We are taking a video file as an input and converting it into the vector of images. We take Positive dataset in the form of the video from CADP dataset and negative dataset from DETRAC dataset. CADP contains videos containing road accidents and DETRAC contains normal traffic videos.

Algorithm 1: Creating dataset

```
1:  Creating_dataset (Input: Video file) {
2:  Define temp_dir="path/to/directory"
3:  For video_file in temp_dir:
            {
    a.  Define path=temp_dir+"/"+video_file
    b.  Assign cap=cv2.VideoCapture(path)
          Assign property_id = int(cv2.CAP_PROP_FRAME_COUNT)
          Assign length = int(cv2.VideoCapture.get(cap, property_id))
    c.  Initialize count = 0
          Initialize success = 1
          Initialize cut = length-99
    d.  Define _dir=cut = "path/to/frames"+folder
    e.  Make directory "_dir"
    f.  While (success):
              {
        i.    success,image=cap.read()
        ii.   if count >= cut:
                    {
                    assign n=count-cut
                    Write image with cv2 in _dir
                    }// End of If
        iii.  count=count+1
        iv.   }// End of while
    g.  }// End of Loop
4:  } // End of algorithm
```

In Algorithm 2, we are training our model by providing the dataset whose polarity we already know. We are using faster RCNN in which region proposal network is the bottleneck of the architecture. We built a convolutional neural network for image classification with keras. We created a sequential model which linearly stacks all the layers, using keras models. We implemented different keras layers like Conv2D-convolutional layer, MaxPooling2D- max pooling layer, Dropout, Dense- add neurons, Flatten- convert the output to 1D vector, using keras layers along with 'ReLU' and 'SoftMax' as activation functions.

Algorithm 2: Model Training

```
1:  Model_training (Input: dataset_with_preknown_polarity) {
2:  Initialize X=train_features.npy
    Initialize Y=train_labels.npy
    X=X/255
3:  Reshape X ( X.reshape(X.shape)+[1])
4:  X = np.load("train_features.npy")
5:  Create model (model=Sequential)
6:  Add first layer (i.e, convolutional layer)
    Pass input_shape only in first layer
    Adding pooling layer (MaxPooling2D(pool_size = (2 , 2) )
    Add another convolutional layer (Conv2D(32, (5 , 5), activation="relu"))
    Add another pooling layer (MaxPooling2D(pool_size = (2 , 2)))
    Add flatteing layer i.e. fully connected layer (model.add(Flatten()))
7:  Embede nuerons using dense layer(model.add(Dense(1000, activation="relu")))
8:  Add a dropout with 50% droupout rate (model.add(Dropout(0.5)))
9:  Embed nuerons using dense layer (model.add(Dense(500, activation="relu")))
10: Add a dropout with 50% droupout rate(model.add(Dropout(0.5)))
11: Embed nuerons using dense layer(model.add(Dense(250, activation="relu")))
    Embed nuerons using dense layer(model.add(Dense(2, activation="softmax")))
12: Compile the model
13: (loss = "sparse_categorical_crossentropy", optimizer = "adam", metrics = ["accuracy"]))
14: Train the model (X, Y, batch_size = 256, epochs = 10, validation_split=0.2, shuffle=True)
15: Save model
16: } // End of algorithm
```

Algorithm 3 shows the pseudo code of testing the model. While compiling the model we used 'categorical cross entropy' as our loss function, ''NAdam' as our optimiser and 'accuracy metrics' as metrics. Then in model fit we ran the model for '30' epochs with '0.2' validation split. Using this model, we are able to predict whether given video contains accident or not.

Algorithm 3: Testing model

```
1:  Model_testing (Input: dataset_for_testing) {
2:  Initialize img_filepath = "path/to/directory"
3:  Define pos = glob.glob(img_filepath + '99frames/*.mp4')
    Define neg = glob.glob(img_filepath + 'negative/*.mp4')
4:  Concatenate all_files ( np.concatenate((pos, neg)) )
5:  Verify (len(neg),len(pos))
6:  Assign labels = np.concatenate(([1]*len(pos), [0]*len(neg[0:len(pos)])))
7:  define n_values=np.max(labels)+1
8:  labels = np.eye(n_values)[values]
9:  Varify (len(labels))
10: Assign batch_size = 15
    Assign num_classes = 2
    Assign epochs = 30
    Assign row_hidden = 128
    Assign col_hidden = 128
    Assign frame , row, col =(99,144,256)
11: Initialize x =Input(shape=(frame, row, col))
12: Define encoded_rows = TimeDistributed(LSTM(row_hidden))(x)
    Define encoded_columns =LSTM(col_hidden)(encoded_rows)
13: Do prediction = Dense(num_classes, activation='softmax')(encoded_columns)
14: Calculate prediction (model = Model(x, prediction))
15: Compile model (loss='categorical_crossentropy', optimizer='NAdam', metrics=['accuracy'])
16: } // End of algorithm
```

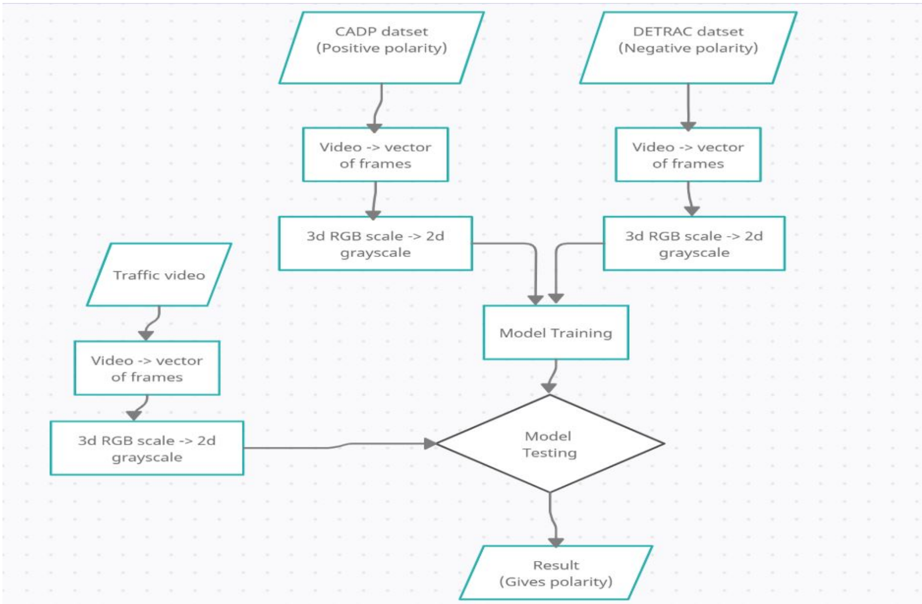
3.4 Flow of information

In the flow chart diagram shown in Figure 6, the data start flowing from the traffic video recorded in IoT based camera, dataset can be a video or a vector of frames depending upon the type of input available, if input type is in video form, the video is converted into vector of frames and to reduce the dimensionality and complexity, we further convert those frames from 3D RGB colour scale to 2D greyscale.

Parallelly we will train our model from CADP video frames which will also be converted into vector of frames containing positive images and DETRAC video frames into vector of frames containing negative images. Then we train our model by feeding the dataset with faster RCNN where region proposal is extracted where the probability of finding required objects like cars, buses bikes are more and then we compute CNN features and classify the scenario of the frames.

When model training is done, we test our model by feeding an unknown dataset to it. After squeezing the images into required complexity, we predict the polarity of the dataset and at last step the result of unknown data will be positive or negative depending on if accident really happened or not.

Figure 6 Flow chart of proposed solution (see online version for colours)



4 Result analysis and performance evaluation

4.1 Experimental setup

We built a simulator for the prototype of our work. The configurations used while running the simulation of the accidental detection system through ML is shown in Table 2. PyCharm Community Edition is python based integrated development environment which we used for testing our code. OpenCV is used for all kinds of image and video processing such as facial recognition and identification, photo editing, automated robotic viewing, Optical Recognition of Character.

Table 2 Specifications while experimental execution

IDE/Dependencies	*Version/Type*
PyCharm Community Edition	2021.1
Opencv-python	3.2.0.8
Keras	3.7, 2
Matplotlib	3.4.1
Scikit-learn	0.22
Scikit-image	0.18, 1

4.2 Pre-processing data input

A *CADP – Accident Detection and Prediction dataset*

The collection consists of 1416 YouTube video segments with 205 video segments containing complete time-space annotations. Compared to other contemporary datasets, this dataset is the highest in terms of number of casualties. CADP features YouTube videos shot in different styles and characteristics of cameras, environmental environments and edited or re-sampled videos.

B *DETRAC dataset*

The UA-DETRAC dataset consists of 100 daunting images from real-life traffic scenes. The dataset consists of more than 140,000 frames of high metadata, including lights, types of vehicles, occlusions, truncation ratio, car bounding boxes.

C *DATASET preparation*

We use two datasets, namely CADP and DETRAC. CADP has accident images from various CCTV camera footages on YouTube, and all accumulated at a single location. DETRAC, on the other hand, comprises non-accident, normal images that are derived from traffic cameras. We use 28,000 accident images from CADP and 23,000 non-accident images from DETRAC dataset. The images are divided into directories 'Accident' and 'Non-Accident', and each image is subsequently scaled to 144–256 and then converted to greyscale. The scaled vector with the image matrix and label is appended into a NumPy array.

After loading the complete set of images, we shuffle the contents of the array (shuffling inserts images with two different labels at random locations, which bring randomness into the data).

4.3 Training faster RCNN model

4.3.1 Loading training data

The previously loaded image matrices and labels are retrieved for the learning procedure. We load the previously stored data from the disk. For each image in the loaded set, we normalise the intensity values at spatial locations. Normalisation removes bias from dataset by downsizing the large contribution from some features and upsizing small contribution from some features.

4.3.2 Network training

We have sequentially stacked all the layers – keras Conv2D, MaxPooling2D, Activation, Flatten, Dense, Dropout layer. The convolutional layer extracts the feature map from image volume. Activation layer adds the activation function for adding non-linearity. Flatten layer converts all output layers to 1D vectors. 'ReLU' and 'Sigmoid' are used as activation functions, and 'binary cross entropy' as our loss function and 'Adam' optimiser and for accuracy matrix calculation.

4.3.3 Testing

For Testing, from the three above-mentioned instances, we take 10% instances as the validation set. We observe that for low-size training samples, the accuracy is low. On increasing, the number of instances, the accuracy somewhat increases, but, on further addition to the sample size, the accuracy hardly increases.

4.4 Performance evaluation and comparison with existing systems

So far, the performance is modest. The issue in object detection may be related to the limited similarities between accident images and the absence of non-linear layers that learn non-linear characteristics.

With an increase in dataset, the accuracy for categorising positive and negative images increases as like shown in Figure 7, the model training becomes more efficient because it gets more data for analysing the greyscale pattern for positive images.

Figure 7 Accuracy of model with different training data size (see online version for colours)

The simulation was conducted on a different number of dataset. Here, number of dataset refers to the amount of data provided for training the model. With increase in dataset provided for training model, model prediction become more efficient i.e., for the size of dataset of 500 images, the accuracy comes out to be 75% which increases to 80% when size of dataset increases to 1500 images. This is because greyscale pattern for positive images can be analysed for larger scale and more variation can be found which will ultimately be able to predict the test dataset more precisely.

We can say change in accuracy on changing dataset amount depends on the type of data provide. But in general, we can say change in accuracy is logarithmic with respect to the dataset size.

Accuracy for a general dataset follows the trend as shown in Table 3. Change in accuracy is logarithmic in accordance to the change in dataset size used for model training because chance of finding new patterns on 2D greyscale map reduces as we keep training the model because many datasets follow the similar 2D greyscale pattern for same polarity.

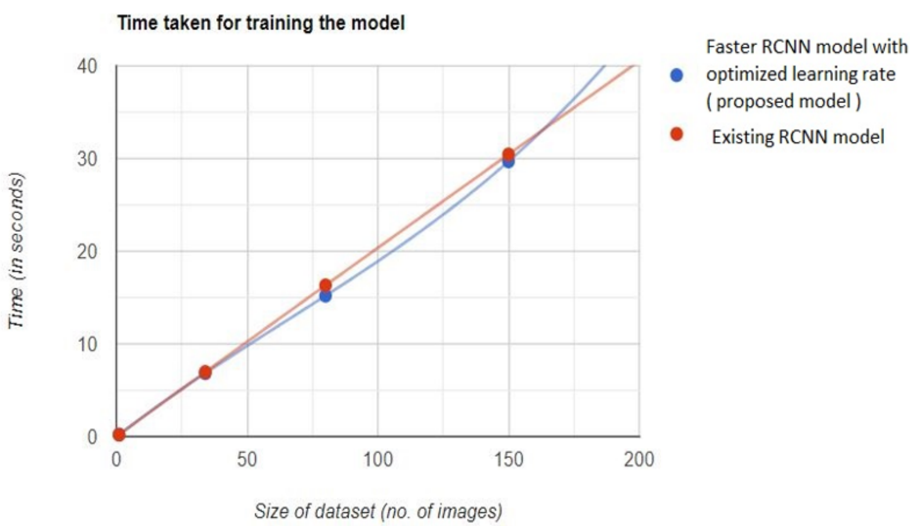
Table 3 Accuracy of proposed and existing model against dataset size

Size of dataset (*No. of images*)	*Accuracy* (*Proposed model*)	*Accuracy* (*Existing model*) (*Shah et al., 2018*)
50	64	61
200	71	67
600	76	73
1500	81	78

For our research, we use Faster RCNN, which greatly reduces time to train the model. The updated version of RCNN is the faster RCNN version. The key difference is that RCNN (Shah et al., 2018) selects regions of interest for generation, while faster uses 'regional network proposals' RCNN (RPN). The regions of interest are the most significant distinction. The entry consists of image maps and a number of object proposals, each with an entity value as an output are generated.

Time taken to train the model for proposed model and existing model for different dataset size is shown in Figure 8. Time needed to process images increases linearly with increase in number of images because all images are processed individually and are independent of each other's.

Figure 8 Accuracy of model with different training data size (see online version for colours)



Data metrics: We have used accuracy and precision to test the results obtained. They are as formulated in equations (1)–(3)

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} foraccident - related \tag{4}$$

$$Precision = \frac{TN}{TN + FN} \, fornon - accident - related \tag{5}$$

where TP is true positive, TN is true negative, FP is False positive, FN is false negative.

In Table 4, time taken for training the model for different dataset size is shown. Graph of time needed to train model against the number of images feeded are linear due to individual processing of images. Existing method (Shah et al., 2018) gives result like this, for the size of dataset of 75 images, the time taken to train comes out to be 15.21 s which increases to 30.01 s when size of dataset increases to 150, while our proposed method takes 14.98 s for 75 images and 29.67 s for 150 images.

Table 4 Time required for training the model against dataset size

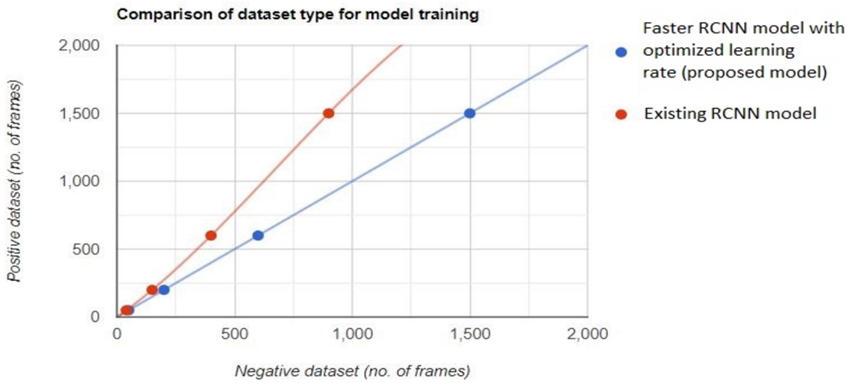
Size of dataset (no. of images)	*Time (in seconds) Proposed model*	*Time (in seconds) Existing model (Shah et al., 2018)*
1	0.202	0.207
34	6.839	6.97
80	15.2	16.32
150	29.7	30.45

In Figure 9, accuracy of positive and negative dataset is shown for different data size. We can observe that accuracy of negative dataset is low as compared to positive dataset, this is because in our model training, we try to find the pattern for accident-related incident like collision, dent in vehicle, certain colours like plasma colour etc. and negative image can also contain some area where similar patterns can be found which act as a false pattern because of similarity with dent or colour pattern.

Figure 9 Accuracy of positive and negative dataset against size of dataset (see online version for colours)

In Figure 10, comparison is shown between the type of dataset i.e., positive and negative dataset. The data used to train the model in existing work was imbalanced dataset that may lead to poor results. It was biased more towards training positive dataset. We took care of the data being balanced and clean so that the model may perform better and produce better results with respect to the results shown in the existing work.

Figure 10 Comparison between positive and negative dataset used for model training against size of dataset (see online version for colours)



We have found most appropriate matrix to judge our classification model is precision, recall and F1-score. Precision tells us about the possibility of predicting data positive which actually turns out to be positive whereas recall tells us about the accuracy of predicting positive data. In our model, recall needs to be more accurate because miss-prediction accident scene as non-accident is costly as compared to labelling non-accident scene as accident because lives are at stakes. F1 scores shows how does the precision and recall are balanced with respect to each other. Figure 11 shows precision, recall and F1-score of our classification model. Figure 11 describes data values against negative class using Faster RCNN.

Figure 11 Data values against negative class using faster RCNN (see online version for colours)

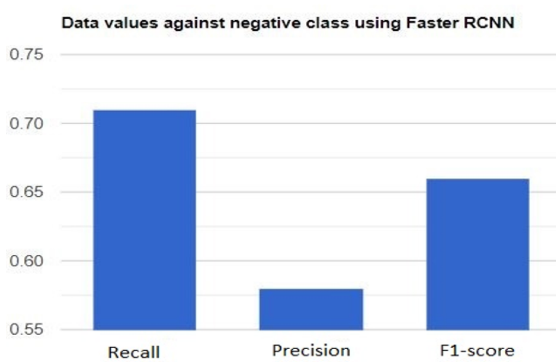


Figure 12 describes data values against negative class using Faster RCNN and Figure 13 Precision of model with different training data size. Figure 14 shows our simulation of work. Average time taken for a general dataset to get trained while using faster RCNN is 0.2 s approximately. Our local training model took around 6.839 s for the dataset of size 34 which is approximately 0.201 s per image of the dataset which is quite a decent time complexity.

Figure 12 Data values against positive class using faster RCNN (see online version for colours)

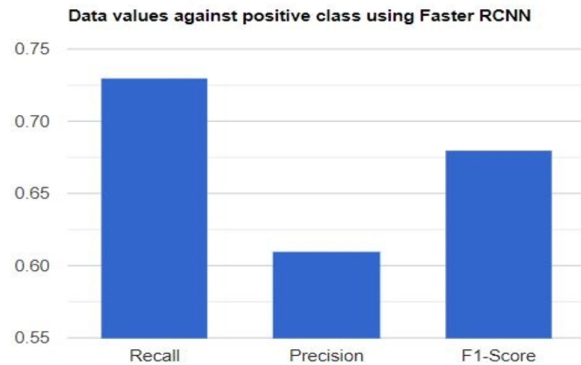
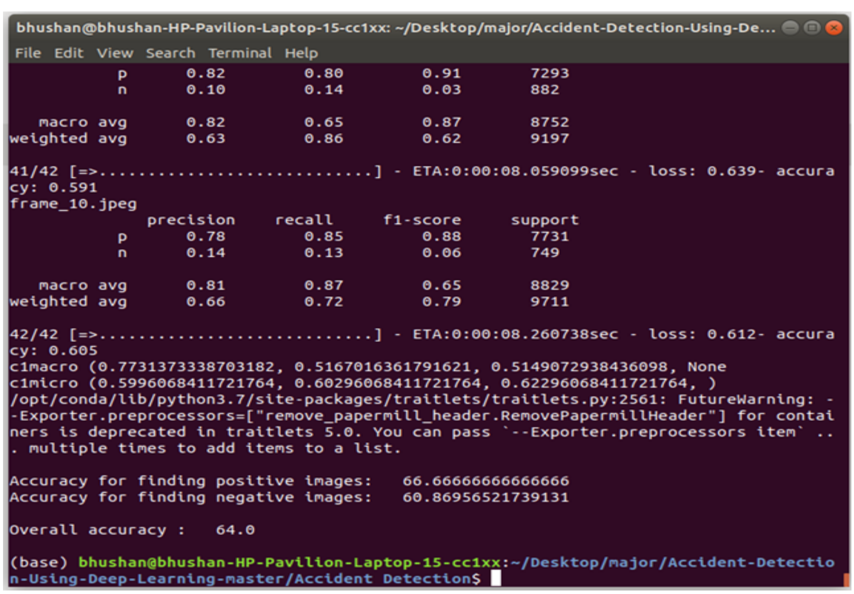


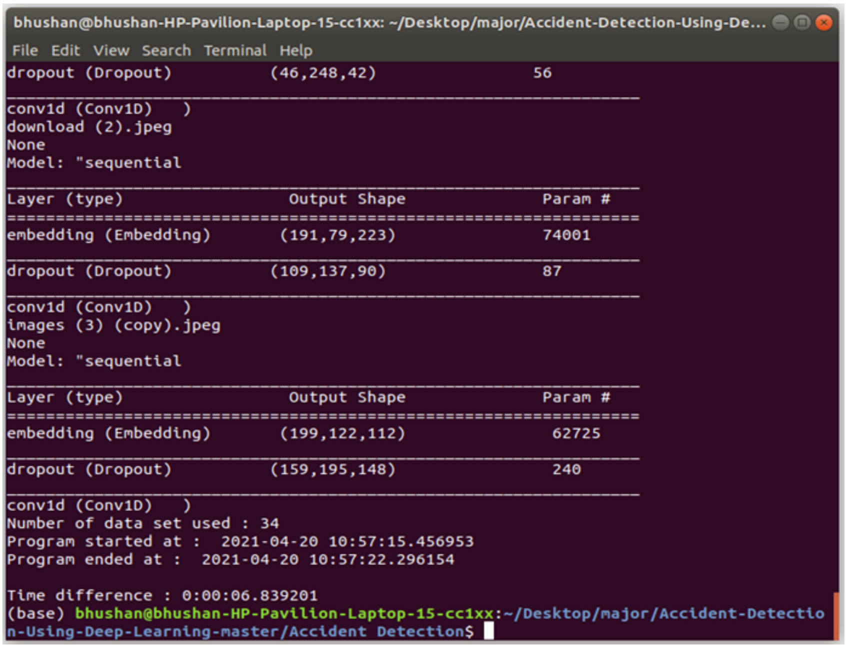
Figure 13 Precision of model with different training data size (see online version for colours)

Figure 14 Average time taken for training small dataset (see online version for colours)



Accuracy obtained by testing a general dataset after training the model is around 64% as shown in Figure 15, which is improvement over the existing work (Shah et al., 2018) which gives an accuracy of around 61% when same amount of dataset is used for training the model. Accuracy for finding the polarity of the image dataset increases in both the model with an increase in dataset size used for model training.

Figure 15 Testing dataset for finding polarity (see online version for colours)



Further improvements can be done in the work by using a better system with faster processing power and faster storage access which will improve the result of the project further. You look only once (YOLO) Algorithm can be added to classify the various type of objects in the accident and non-accident images. Context augmentation with faster RCNN could further improve the accuracy.

Because, it makes sure that a data subject is not affected (e.g., not harmed) by their entry or participation in a database, while maximising utility/data accuracy (as opposed to random/empty outputs) for the queries, the *Differential Privacy* (*DP*) can be used to measure the privacy of this model. The key advantages of DP are that the raw data will not be viewed and does not need to be modified ant the DP is Resilience to post-processing.

5 Conclusion and future scope

With an increase in dataset, the accuracy for categorising positive and negative images increases as model training becomes more efficient because it gets more data for analysing the greyscale pattern for positive images. The issue in object detection may be related to the limited similarities between accident images and the absence of non-linear layers that learn non-linear characteristics. The performance obtained is quite decent. In future, the implementation could be improved by stacking more layers in the model. Though we are training on images, the actual implementation envisages a video frame to image converting utility that sits in between Camera feed and Testing Model. This, in turn, would convert Real-time traffic camera feeds into video frames and subsequently to greyscale images (RGB images take up exceedingly long time to train and test). After making the above improvements, we aim to achieve better accuracy through the enhanced model.

References

Aldunate, R.G., Herrera, O.A. and Cordero, J.P. (2013) 'Early vehicle accident detection and notification based on smartphone technology', *Ubiquitous Computing and Ambient Intelligence. Context-Awareness and Context-Driven Interaction, Lecture Notes in Computer Sci.*, Vol. 8276, Springer, Cham, https://doi.org/10.1007/978-3-319-03176-7_46

Ali, A. and Eid, M. (2015) 'An automated system for accident detection', *IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 11–14 May, 2015, Italy, pp.1608–1612, doi: 10.1109/I2MTC.2015.7151519.

Alvi, U., Khattak, M.A.K., Shabir, B., Malik, A.W. and Muhammad, S.R. (2020) 'A comprehensive study on IoT based accident detection systems for smart vehicles', *IEEE Access*, Vol. 8, pp.122480–122497, doi: 10.1109/ACCESS.2020.300688.

Bhatti, F., Shah, M.A., Maple, C. and Islam, S.U. (2019) 'A novel internet of things-enabled accident detection and reporting system for smart city environments', *Sensors* (*Basel, Switzerland*), Vol. 19, No. 9, p.2071, doi: 10.3390/s19092071.

Darwin, N.A., Suresh, T., Aarthi, M., Gomathi, K., Aarthi, G. and Mugilan, P. (2020) 'Accident detection, alert and tracking system based on IoT', *European Journal of Molecular and Clinical Medicine*, Vol. 7, No. 4, pp.1370–1374.

Dash, D.K. (2020) *This is What Makes Indian Roads Deadliest*, The article in TOI, Presented on 29 October, 2020, https://timesofindia.indiatimes.com/india/this-is-what-makes-indian-roads-deadliest/articleshow/78870630.cms (Visited 30 March, 2021).

Dashora, C., Sudhagar, P.E. and Marietta, J. (2020) 'IoT based framework for the detection of vehicle accident', *Journal: Cluster Computing*, Issue 2/2020.

Digital Editor (2019) *Road Accidents in 2018 Rose 2.4% To 151,000; 70% Deaths In 18-45 Age Group*, Road accidents | National Highways Press Trust of India, https://www.business-standard.com/article/pti-stories/road-accident-deaths-record-2-37-pc-rise-to-1-51-lakh-in-2018-119111901227_1.html (19 November, 2019) (visited on 30 March, 2021).

Girshick, R. (2015) 'Fast R-CNN', *IEEE International Conference on Computer Vision (ICCV)*, 7–13 December, 2015, Washington DC, USA, pp.1440–1448, doi: 10.1109/Iccv.2015.169.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) 'Rich feature hierarchies for accurate object detection and semantic segmentation', *IEEE Conference on Computer Vision and Pattern Recognition*, 23–28 June, 2014, Washington DC, USA, pp.580–587, doi: 10.1109/CVPR.2014.81.

Ijjina, E.P., Chand D., Gupta, S. and Goutham, K. (2019) 'Computer vision-based accident detection in traffic surveillance', *10th International Conference on Computing, Communication and Networking Technologies* (*ICCCNT*), Kanpur, India, pp.1–6, doi: 10. 1109/ICCCNT45670.2019.8944469.

Lee, K.B. and Shin, H.S. (2019) 'An application of a deep learning algorithm for automatic detection of unexpected accidents under bad CCTV monitoring conditions in tunnels', *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, Istanbul, Turkey, pp.7–11, doi: 10.1109/Deep-ML.2019.00010.

Li, Z., Yang, W., Peng, S. and Liu, F. (2021) 'A survey of convolutional neural networks: analysis, applications, and prospects', *IEEE Transactions on Neural Networks and Learning Systems*, pp.1–21, doi: 10.1109/TNNLS.2021.3084827.

Madhusan, A., Viswanathan, L., Ravindran, V. and Rangaswamy, S. (2016) 'A survey on road accident detection and reporting', *International Journal of Scientific and Engineering Research*, Vol. 7, No. 4, April, pp.71–81.

Naik, K.J. (2020a) 'Classification and scheduling of information-centric IoT applications in cloud-fog computing architecture (CS_IcIoTA)', *14th International Conference on Innovations in Information Technology (IIT)*, 17–18 November, United Arab Emirates University (UAEU), doi: 10.1109/IIT50501.2020.9299006.

Naik, K.J. (2020b) 'A processing delay tolerant workflow management in cloud-fog computing environment (DTWM_CfS)', *International Conference on Decision Aid Sciences and Application (DASA 20)*, 8–9 November, Kingdom of Bahrain, doi: 10.1109/DASA51403.2020.9317201, IEEE Explore, pp.908–913..

Naik, K.J. and Naik, D.H. (2020) 'Minimizing deadline misses and total run-time with load balancing for a connected car systems in fog computing', *International Journal of Scalable Computing: Practice and Experience, SCPE*, Vol. 21, No. 1, pp.73–84, ISSN 1895-1767, doi: 10.12694:/scpe.v21i1.1616.

Naylor, M. and Attwood, C.I. (2003) *ADVISOR: Annotated Digital Video for Intelligent Surveillance and Optimised Retrieval*, Classification: Public, ©The ADVISOR Consortium 2000–2003, Issue 1.

NHTSA (2015) National highway traffic safety administration (NHTSA), United States Department of Transportation, Visited 6 April, 2021, https://cdan.nhtsa.gov/tsftables/tsfar.htm

Ren, S., He, K., Girshick, R. and Sun, J. (2017) 'Faster R-CNN: towards real-time object detection with region proposal networks', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, pp.1137–1149.

Routh, J., Das, A., Kundu, P. and Thakur, M. (2019) 'Automatic vehicle accident detection and messaging system using GPS and GSM module', *International Journal of Engineering Trends and Technology*, Vol. 67, No. 8, August, pp.69–72, doi: 10.14445/22315381/IJETT-V67I8P.211.

Shah, A., Lamare, J.B., Anh, T.N. and Hauptmann, A. (2018) 'Accident forecasting in CCTV traffic camera videos', *IEEE International Workshop on Traffic and Street Surveillance for Safety and Security,* arXiv preprint arXiv: 1809.05782.

Shah, A.P., Lamare, J-B., Nguyen-Anh, T. and Hauptmann, A. (2018) 'Cadp: A novel dataset for cctv traffic camera-based accident analysis', *15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 27–30 November, 2018, IEEE, Poland, pp.1–9.

Sharma, S. and Sebastian, S. (2019) 'IoT based car accident detection and notification algorithm for general road accident', *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 9, No. 5, October, pp.4020–4026.

Sharma, S. and Sebastian, S. (2019) 'IoT based car accident detection and notification algorithm for general road accidents', *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 9, No. 5, October, pp.4020–4026.

Sun, S. (2021) *Number of Deaths Due to Road Accidents Across India From 2005 to 2019*, Statista Research Department (on 15 March, 2021) (Visited on 30 April, 2021), https://www.statista.com/statistics/746887/india-number-of-fatalities-in-road-accidents/

Tian, D., Zhang, C., Duan, X. and Wang, X. (2019) 'An automatic car accident detection method based on cooperative vehicle infrastructure systems', *IEEE Access*, Vol. 7, pp.127453–127463, doi: 10.1109/ACCESS.2019.2939532.

Wang, C., Dai, Y., Zhou, W. and Geng, Y. (2020) 'A vision-based video crash detection framework for mixed traffic flow environment considering low-visibility condition', *Journal of Advanced Transportation,* Vol. 2020, 11 pages, https://doi.org/10.1155/2020/9194028

Webster, W.H. and Williams, H. (1992) *The City in Crisis: A Report by the Special Advisor to the Board of Police Commissioners on the Civil disorder in Los Angeles*, Local government publication, Office of the Special Advisor to the Board of Police Commissioners.

Websites

CADP: A Novel Dataset for CCTV Camera based Accident Analysis (Visited on 5 April, 2021) https://ankitshah009.github.io/accident_forecasting_traffic_camera

UA-DETRAC, Welcome to the UA-DETRAC Benchmark Suite, DETRACT Dataset: https://detrac-db.rit.albany.edu/ (Visited on 10 April, 2021).