Intelligent System for Road Accident Detection in India using Deep Learning Models

Nimish Agarwal[1], Aman Jangid[1], Ashutosh Sharma [1], Nikhil Tanwar[1], Mudit Kumar[1], Pankaj Kumar[1]
Pranamesh Chakraborty[2]

***Abstract*—In the age of smart cities, it is more than necessary that a real-time road accident detection system exists that can automatically identify the vehicle, predict its level of damage, and notify emergency services for quick perusal. In this paper, we build an intelligent traffic monitoring system using state-of-the-art deep learning models that aims to detect traffic accidents and the condition of traffic at the site in real-time so that specific measures can be employed to deal with the situation, ranging from deploying police vehicles to handle the amount of congestion to getting the appropriate ambulance depending on the damage level.**

***Index Terms*—Car Crash Detection, Deep Learning, Transfer Learning, CNN, ResNet50, DenseNet, EfficientNet**

I. Introduction

Traffic safety is one of the major transportation concern in India. In 2018, India experienced the highest number of accident related deaths in the world (1,51,417 fatalities), which is approximately 11% of the traffic deaths in the world []. Hence, it is imperative to take crucial steps in reducing crashes and making roads safer to the users. One of the important steps can be taken by traffic management systems for safer road systems is detect accidents quickly such that it can be alerted to the relevant users and necessary steps can be taken to resolve the incidents quickly. Quicker traffic incident detection can help in reducing secondary traffic incident risks, traffic congestion, and reduced capacity due to incidents.

Traditionally, detector-based methods have been used extensively for traffic incident detection using loop based or other static traffic sensors []. However, detector failures and communication errors are perennial problems in such fixed sensors, thereby resulting in unreliable performance. On the other hand, vehicles equipped with sensors such as GPS devices can provide real-time traffic information, which can be utilized to detect anomalies such as traffic incidents. This typically involves understanding the traffic patterns based on historical or recent past traffic information to detect abnormalities from the expected pattern. Traffic information obtained from the mobile GPS devices can overcome the shortcomings of limited spatial coverage from static sensors such as loop detectors. However, both loop detectors or GPS devices based traffic incident detection methods rely on significant change in traffic to detect anomalies. Therefore, crashes on road-sides which do not influence regular traffic flow immediately after the crash are difficult to detect using these traditional sensors. Significant time can be wasted to reflect the impacts of crashes on the traffic, such as speed or density to detect the anomalies using these variables. In these circumstances, cameras possess tremendous potential to detect traffic incidents in a significantly shorter duration of time, thereby leading to quicker recovery and lesser impact on the road users.

Although there has been significant research in using pattern recognition or camera-based techniques for traffic incident detection, majority of these studies were done primarily using traffic data from developed countries such as USA and/or European countries, where traffic characteristics are significantly different from that of developing countries such as India, where mixed non-lane based traffic behavior is observed. Further, most of the research is limited to detecting only traffic crashes and few studies have looked into determining other traffic conditions such as traffic congestion, sparse traffic, etc. from cameras obtained from Indian traffic scenarios. Therefore, the aim of this study to detect different important traffic conditions, such as crashes, burning vehicles, congestion, sparse traffic using Indian traffic cameras.

One of the preliminary challenges in developing such vision-based models for traffic characteristic determination in Indian traffic conditions is the lack of enough data for such conditions, which can be used to train large-scale deep learning models. Although such deep learning usually obtain state-of-the-art performance in camera-based different classification or detection tasks, but are however quite data hungry in nature, requiring significant amount of labelled data. To alleviate this issue, in this study, we adopted two techniques. First, we have used transfer learning approach to fine-tune pre-trained state-of-the-art deep learning models on the traffic dataset, which can handle small scale dataset, even without car crash data which can be hard to obtain. Also, we applied image augmentation to extend the dataset and add variety in the dataset to prevent overfitting of the deep learning models. Secondly, we used web-crawling (using Bing Image Search API) to build a custom dataset containing Indian images that can be used to extend the usability of our model on Indian traffic conditions for detecting road accidents.

Such computer vision based AI-powered autonomous accident-detection system can be used by the traffic management systems to detect traffic accidents and burning vehicles

[1]Nimish Agarwal, Aman Jangid , Ashutosh Sharma, Nikhil Tanwar, Mudit Kumar, and Pankaj Kumar are undergraduate students in the Department of Civil Engineering, Indian Institute of Technology Kanpur, Kanpur, India
[2]Pranamesh Chakraborty is an Assistant Professor in the Department of Civil Engineering, Indian Institute of Technology Kanpur, Kanpur, India pranames@iitk.ac.in

via live surveillance cameras situated at various locations inside the city and on the national highways/expressways. The system will allow the government to monitor traffic accidents in real-time and notify emergency services for quick perusal. Such information will be automatically geo-tagged so that the concerned authorities can be dispatched quickly and efficiently for response.

The developed automatic camera-based traffic accident and traffic state detection system can be used to streamline the following tasks:

- Taking swift action based on the level of damage (road accident, burning vehicle, fatalities, etc.)
- Analyzing patterns: Analyzing the collected historical data from the model can help to detect repeated accidents at certain places due to blind spots, insufficient lighting, poor road conditions, etc. and the leading cause of accidents can be resolved by taking necessary actions such as constructing foot over bridges, underpasses, etc. wherever possible.
- A comparable metric for safety: Through accidents generated data on a large-scale level, the generated monthly reports can be used to rank cities/highways on "Road safety parameters."
- Manage transportation efficiently: Early detection of an accident can save lives, provides quicker road openings, hence decreasing traffic delays, wasted time and resources, and increasing efficiency.

II. Literature Review

The traffic accident detection and real-time traffic monitoring is a challenging issue and has attracted a lot of attention from researchers. They have proposed and applied various traffic accident detection methods. In general, traffic accident detection methods are mainly divided into the following two kinds: vehicle running condition-based and accident video features-based. [1]

In [2] a computer vision-based accident detection framework is proposed for traffic surveillance which capitalizes on Mask R-CNN for object detection followed by an efficient centroid based object tracking algorithm on surveillance footage. The probability of an accident is determined based on speed and trajectory anomalies in a vehicle after an overlap with other vehicles. Centroid Tracking Algorithm proposed includes 3 major tasks: C1: Determine the overlap of bounding boxes of vehicles, C2: Determining Trajectory and their angle of intersection and C3: Determining Speed and their change in acceleration

A vision-based video crash detection framework for mixed traffic flow environment considering low-visibility condition is proposed in [3], in which first, Retinex image enhancement algorithm was introduced to improve the quality of images, collected under low-visibility conditions. Then, a Yolo v3 model was trained to detect multiple objects from images, including fallen pedestrians/cyclists, vehicle rollover etc. Then, a set of features were developed from the Yolo outputs, based on which a decision model was trained for crash detection. An experiment was conducted to validate the model framework. The results showed that the proposed framework achieved a high detection rate of 92.5%, with relatively low false alarm rate of 7.5%.

In [4] a novel approach to automatic road-accident detection using machine vision is proposed which deals with supervised learning methods comprising three different stages were combined into a single framework in a serial manner to successfully detect damaged cars from static images. The three stages used five SVMs trained with Histogram of gradients (HOG) and Gray level co-occurrence matrix (GLCM) to extract features. Also, normalization was done to reduce the effect of lighting conditions. Models using GLCM performed better. Limitations as only traffic accidents were identified, and it was limited to daytime use only.

Another paper [5] on traffic accident detection by using machine learning methods, aims to analyse collected information from vehicles to detect forward collisions. Drivers will be alerted about collisions and they will have time to take precaution to avoid piled-up collisions. For this SUMO traffic simulator has been used to enable mobility of vehicles and collect position and sleep information. It is a collision free traffic simulator used to simulate accidents. Cars are forced to make a stop in predefined positions. Stops also can be considered important incidents in a road segment. Considered incidents in normal traffic flow as an anomaly. When an accident happens, following cars will slow down or stop, and many cars will be affected by the accident. When location data of vehicles are analysed , it is seen that many cars are collected around accident location. Clustering algorithms are used to group vehicles according to their speed and location in particular road segment. In accident case, algorithms will put vehicles which is affected by accident in one group , other vehicles in other group or groups.

In [6] a real-time autonomous highway accident detection model is proposed based on big data processing and computational intelligence. They chose 7 RTMS (Real-Time Monitoring System) sensors data located on a highway. There are no traffic lights, stop signs, sharp curves, etc. on that section of the highway. So, they can assume if any slowdown or stoppage is seen in the traffic, it should be due to a disruption on the road. Data is collected every 2min through these sensors which consist (i) Number of cars passing every 2 minutes. (ii) Average speed of the vehicles in the last 2 minutes. (iii)Average occupancy ratio of the lane in the last 2 minutes. (iv) Date/Time information. They took the data of accidents/disruptions from the Traffic Department Database. 72 incidents were observed on that highway. They select three models: KNN, regression tree, and feed-forward neural networks.

III. Study Data

Training a deep convolution neural network based computer vision model for detecting road accidents, car crashes, and different traffic conditions requires a large-scale image dataset.

However, there are very few road accident image datasets currently available for Indian traffic conditions. So two different datasets were used for image classification, one benchmark dataset and the other collected by web-crawling.

A. Traffic-Net Dataset

Traffic-Net [7] is a dataset containing traffic images, collected in order to ensure that vision-based models can be trained to detect traffic conditions and provide real-time monitoring, analytics, and alerts. This is a part of the DeepQuest AI's to train machine learning systems to perceive, understand, and act accordingly in solving problems in any environment they are deployed.

The Traffic-Net dataset comprise of 4,400 traffic images (in good resolution) that span over 4 classes, with approximately 1,100 images for each category. The classes included in this dataset are:

- Accident
- Fire
- Dense Traffic
- Sparse Traffic

B. Custom Indian Traffic Dataset

The goal to create this custom dataset is to gather a diverse database of Indian traffic images that can be used to extend the usability of our model in Indian traffic conditions for detecting road accidents. Google Chrome Extension [8] and Bing Image Search API [9] was used to crawl the web for accident and non-accident related images. Queries (such as Delhi road accident, traffic jam in Mumbai) were used to obtain a diverse set of images for different traffic conditions. Figure 1 shows sample images in the dataset. The dataset comprises of the same set 4 classes of images as in the Traffic-net dataset. The number of images collected for each class are:

- Accident - 167
- Fire - 219
- Dense Traffic - 219
- Sparse Traffic - 156

Manual inspection of the collected images were done to remove noisy and non-Indian images. The final dataset consists of approximately 150 images belonging to each class. These images were used to analyse our model extensively to see how it will behave on ground level Indian traffic conditions.

IV. Methodology

Building our deep learning based image classification model on the Traffic-Net dataset to detect road accidents/car crashes, burning vehicles, and traffic conditions (sparse and congested) in real-time comprises of three distinct steps: (a) Setup (b) Image Processing, and (c) Image Classification. Each task is described next, with our primary focus geared towards image classification using state-of-the-art deep learning models.

A. Setup

We split the Traffic-Net dataset in 4:1 ratio with approximately 900 images for training and 200 images for testing for each class. This constitutes training our model on a total 3600 images and testing our model on a total 800 images. Then we followed progressive training approach for training our deep learning models by gradually increasing the number of images in training dataset and number of epochs. We then progressively applied different advanced deep learning models and compared their performance on the custom built Indian dataset. All models were trained and tested on open-source GPU resource provided by Google Colaboratory [10].

B. Image Processing

Noisy/mislabelled images removal: First, we manually removed noisy/mislabelled images from Indian dataset gathered from the search API by bifurcating images from each class into positive and negative samples. That way we can ensure only Indian images in our testing data and test performance of our model on true positives as well as false negatives.

Image Augmentation: We then applied image augmentation to artificially create training images through different ways of processing or combination of multiple processing, such as random rotation, shifts, shear and flips, etc. Since deep networks require large amounts of labelled training data to achieve good performance, this data augmentation step helps to build a powerful image classifier using limited training data and boost the performance of deep networks.

Fixed Image Resolution: Images in the image-net dataset vary in size, therefore, we establish a base size for all images fed into our AI algorithms and reduced resolution to decrease training and prediction time since these models need to be tested for video classification.

C. Image Classification using Deep Learning Models

Implemented state-of-the-art deep learning models including DenseNet [11], ResNet50 [12] and finally EfficientNet-B1 [13] for image classification purposes using custom training method of ImageAI Library. ImageAI provides very powerful yet easy to use classes to train state-of-the-art deep learning algorithms on our own image datasets. [14]

1) DenseNet: ConvNets can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. Based on this observation, DenseNet connect each layer to every other layer in a feed-forward fashion. Whereas traditional convolutional networks with L layers have L connections - one between each layer and its subsequent layer - DenseNet network has $L(L+1)/2$ direct connections. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. DenseNets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. When tested on four highly competitive object recognition benchmark

Fig. 1. Custom Indian traffic dataset

tasks (CIFAR-10 [], CIFAR-100[], SVHN[], and ImageNet[]), DenseNets obtain significant improvements over the state-of-the-art on most of them, whilst requiring less computation to achieve high performance [11]. We trained this model using transfer learning on only 2 classes (accident and fire).

2) ResNet50: Residual Networks or ResNets [12] reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning un-referenced functions. These residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset [], these residual nets with a depth of up to 152 layers—8x deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. Deep residual nets won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation. ResNet50 has 48 Convolution layers along with 1 Max Pool and 1 Average Pool layer. Architecture of ResNet50 is shown in Figure 2. It has $3.8X10^9$ Floating points operations. We trained this state-of-the-art Deep Learning model on the complete Traffic-Net dataset (4 classes) for image classification of different traffic conditions.

3) EfficientNet: Convolution Neural Networks (ConvNets) are commonly developed at a fixed resource budget, and then scaled up for better accuracy if more resources are available. After studying model scaling systematically, it is identified that carefully balancing network depth, width, and resolution can lead to better performance. Based on this observation, a new scaling method was developed that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient.

Going further, researchers at Google used neural architecture search to design a new baseline network and scale it up to obtain a family of models, called EfficientNets, which achieve much better accuracy and efficiency than previous ConvNets. In particular, the EfficientNet-B7 model achieved state-of-the-art 84.3% top-1 accuracy on ImageNet, while being 8.4x smaller and 6.1x faster on inference than the best existing ConvNet [13].

V. RESULTS AND DISCUSSION

We implemented the state-of-the-art deep learning models DenseNet, ResNet50 and EfficientNet-B1 for image classification of four different classes. The models are trained on Google Colab GPU [10] achieving high image classification accuracy. Our models are trained on image-net dataset containing 4400 images (1100 per class). The efficacy of the models are validated using the validation dataset (subset of image-net dataset) and the final accuracy is reported for the test dataset which is custom build Indian dataset containing a total of 600 images (150 images for each class). The accuracy of the model (ACC) is given by the accuracy of correctly positive classified images (TPR) and negative images (TNR), as shown in Equations 3, 1, and 2. TP and TN refer to the number of correctly identified images for each class while P and N refer to total number of images for each class (150 each). Table I shows the validation accuracy (VAL), testing accuracy (ACC), frames-per-second a model can classify (FPS), precision, recall and F1-score for each model.

TABLE I
PERFORMACE COMPARISON OF DIFFERENT MODELS TO DETECT ROAD ACCIDENT AND TRAFFIC CONDITION

Model	VAL	ACC	FPS	Precision	Recall	F1-Score
DenseNet	0.94	0.89	15	0.86	0.97	0.91
ResNet50	0.91	0.94	20	0.88	0.88	0.88
EfficientNet-B1	0.93	0.88	0.71	0.88	0.88	0.88

$$TPR = \frac{TP}{P} \tag{1}$$

$$TNR = \frac{TN}{N} \tag{2}$$

$$ACC = \frac{TP + TN}{P + N} \tag{3}$$

It shows that

After achieving high validation accuracy on traffic net dataset, we tested our models on custom built dataset containing images from Indian settings gathered from the Internet. When tested over 400 Indian images (100 per class), the model classified 344 images correctly giving a classification accuracy of 84%. Sample results on Custom Indian dataset can be found in Figure 3 [15] [16]. This shows that our model has successfully classified all the four classes correctly with high confidence levels in Indian setting as well.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 2. ResNet50 architecture

Prediction Label: Accident
Confidence: 99.99

Prediction Label: Fire
Confidence: 99.99

Prediction Label: Dense_Traffic
Confidence: 99.97

Prediction Label: Sparse_Traffic
Confidence: 99.98

Fig. 3. Sample Results on Custom Indian dataset

Prediction Label: Dense_Traffic
Confidence: 90.50

Actual Label: Sparse_Traffic
Confidence: 100

Fig. 4. An example of a False Positive

Looking at some examples of false positive cases as shown in Figure 4 shows that our model needs some more fine-tuning on the Indian dataset.

Achieving high classification accuracy on the image dataset, we moved one step forward and applied this model on the two sample videos of different graphical settings using rolling averaging method. The accident detection model runs at around 30 frames per second (fps) on the google colab GPU, making it suitable for real-time performance.

VI. Future Scope

Using LSTMs and RNNs for time-series data (since subsequent frames in a video are correlated with respect to their semantic contents)

VII. Conclusion

This paper presents a method to detect four different kinds of traffic situations namely dense traffic, sparse traffic, road accidents, and burning vehicles using state-of-the-art deep learning models. Here, we use transfer learning to fine tune the deep learning models of ResNet50, DenseNet and EfficientNet-B1 to train on traffic visuals from multiple sources of traffic data. Considering the fact that images gathered from the internet have some noisy images left even after manual pre-processing, we have achieved a decent accuracy on Indian dataset.

References

[1] D. Tian, C. Zhang, X. Duan, and X. Wang, "An automatic car accident detection method based on cooperative vehicle infrastructure systems," *IEEE Access*, vol. 7, pp. 127 453–127 463, 2019.

[2] E. P. Ijjina, D. Chand, S. Gupta, and K. Goutham, "Computer vision-based accident detection in traffic surveillance," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2019, pp. 1–6.

[3] C. Wang, Y. Dai, W. Zhou, and Y. Geng, "A vision-based video crash detection framework for mixed traffic flow environment considering low-visibility condition," *Journal of advanced transportation*, vol. 2020, 2020.

[4] V. M. Vishnu, M. Rajalakshmi, and R. Nedunchezhian, "Intelligent traffic video surveillance and accident detection system with dynamic traffic signal control," *Cluster Computing*, vol. 21, no. 1, pp. 135–147, 2018.

Fig. 5. Sample images of car accident detection in NFS game across 3 frames taken at 1-second interval

[5] N. Dogru and A. Subasi, "Traffic accident detection by using machine learning methods," in *Third International Symposium on Sustainable Development (ISSD'12)*, 2012, p. 467.
[6] A. B. Parsa, H. Taghipour, S. Derrible, and A. K. Mohammadian, "Real-time accident detection: coping with imbalanced data," *Accident Analysis & Prevention*, vol. 129, pp. 202–210, 2019.
[7] "Traffic net 2.0," http://traffic-net.org/.
[8] "Download all images," https://chrome.google.com/webstore/detail/download-all-images/ifipmflagepipjokmbdecpmjbibjnakm.
[9] "Bing image search api," https://www.microsoft.com/en-us/bing/apis/bing-image-search-api.
[10] "Google colaboraory," https://colab.research.google.com/.
[11] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks. arxiv 2016," *arXiv preprint arXiv:1608.06993*, vol. 1608, 2018.
[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[13] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.
[14] "Image ai," https://imageai.readthedocs.io/en/latest/prediction/index.html.
[15] "Traffic wikipedia," https://en.wikipedia.org/wiki/Traffic_collisions_in_India.
[16] "Telegraph india," https://www.telegraphindia.com/bihar/zero-mile-to-masaurhi-4-lane-relief/cid/1671511.