Grenze International Journal of Engineering and Technology, June Issue

Road Accident Detection using Deep Learning

P.Ravi Prakash[1],G. Mamatha[2], J.Jhansi[3], K.Madhan[4], M.Jahnavi[5]
[1]Prasad V Potluri Siddhartha Institute Of Technology, Vijayawada, India
[2-5]Prasad V Potluri Siddhartha Institute Of Technology, Vijayawada, India
Email: raviprakashparuchuri@pvpsiddhartha.ac.in, {19501a1231, 19501a1242,19501a1250, 19501a1257}@pvpsiddhartha.ac.in

***Abstract*— TIt is estimated that 1.3 million people die each year from traffic accidents. The response time of paramedics plays an important role in determining survival probability. At the site of an accident, the response time variable depends on the actions of pedestrians and traffic officials. Using a CNN with alternate max-pooling layers, we propose using CCTV camera footage to classify images as accidents or non-accident. Using alternating max-pooling layers helps detect accidents even if the vehicle is not in the centre of the frame. We prepare our dataset from a variety of sources due to the lack of accident datasets in this field.**

***Index Terms*— Accident Detection, Convolutional Neural Network, Image Classification, deep learning.**

I. INTRODUCTION

In developing countries, road traffic accidents are more prevalent than in developed ones. In addition to life loss, they cause considerable economic losses as well. DL-based approaches have demonstrated high performance in computer vision tasks involving complex feature relationships. Therefore, we develop an automated DL-based method for detecting traffic accidents based on deep learning. Deep learning has become increasingly applicable in a variety of fields over the last few years. As part of deep learning, convolutional neural networks are used in visual image analysis for object detection, face recognition, robotic segmentation, pattern recognition, natural language processing, spam detection, regression analysis, and speech recognition and image classification..a CNN is trained to classify images as 'accidents' or 'non-accidents'. A convolutional neural network framework has seven layers. Images were adapted in accident detection based on their pixel values. It has input and output layers. A layer between the input and output layers contains some hidden layers. Layers are not fully connected. Instead, each neuron in a layer is connected to a local neuron. The main aim of this article is to observe the influence of convolutional networks on accident detection. This data set has been processed using tensor flow, which is a Python library for convolutional neural networks. In this paper, we analyse changes to outcomes when using different combinations of convolutional neural networks' hidden layers. There is an accuracy of 94% in detecting accidents, demonstrating a high capacity for detection.

II. RELATED WORK

The proposed system mainly helps to detect accidents. We will be using Deep Learning based on Conventional Neural Networks (CNN) to achieve the outcome of the accident detection proposed model. Here the accident video is captured and can be used to detect the accident. The proposed System covers the following phases

Grenze ID: 01.GIJET.9.2.332_1

- Collection of data for training purposes.
- Preprocessing the data.
- Model building.
- Model Training.
- Model validation.

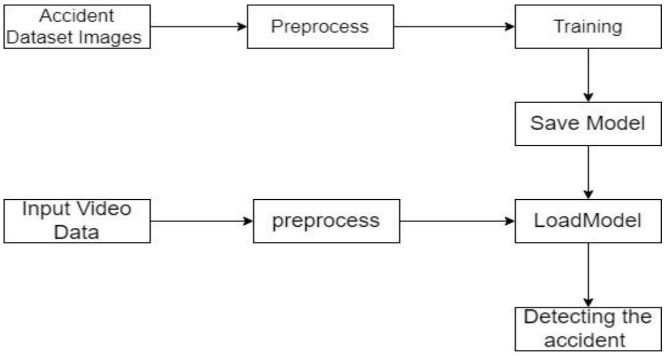


Fig: System Architecture

A. Collection of data:

Pictures from other sources, including other datasets like CADP for accident photographs and DETRAC for non-accident images, were hand-selected and ranged in resolution and quality. The DETRAC vehicle detection data collection is made up of a sizable number of photos that were taken on roads, lanes, etc. A few pictures were also chosen from well-known websites like Google Images. These pictures are either marked as accidental or not. The usage of all three colour channels prevents the image from ever being transformed to grayscale. Both the colour values and the image size are normalised to 250*250. These are the sample CCTV footage photos for accident detection.

Fig: Non-Accident Data

Fig: Accident Data

B. Preprocessing the Data:

Preprocessing data entails altering or coding it to make it simpler for computers to parse.Preprocessing of the data is typically done to ensure reliable results. Data cleaning, data transformation, data reduction, and data integration are involved. Filling in missing numbers, reducing noise in the data, resolving discrepancies, and eliminating outliers are all examples of data cleaning, which is a step in the preprocessing of data. Data from several sources can be combined into a single, larger data storage through the use of data integration. After cleaning the data, we must combine the good data into other formats by altering its value, structure, or format using either normalisation or generalization. Data reduction employs fewer data overall while yet producing analytical conclusions of the same calibre as the original dataset.

C. Model Building:

CNN is used to build the accident detection model by using libraries like Keras and Tensor flow. The tensor flow is used to describe how data operations on a tensor or multidimensional array flow. A tensor can be created from a calculation's input or output data. Tensor Flow enables automated data gathering, model tracking, performance monitoring, and model retraining in accordance with industry best practices. A high-level Python package called Keras utilises the Tensor Flow framework. It is employed to comprehend deep learning methods like building layers for neural networks. Data visualisation and graphical plot representation with Matplotlib are cross-platform.

D. Model Training:

Training a model is the primary step in any learning model, resulting in a working model that can be validated, tested, and deployed. Performance during training determines how well the model will perform.

E. Model validation:

Model validation occurs after Model Training when a testing dataset is used to evaluate the trained model. In this phase, users can understand whether the predicted results are obtained or not.

III. METHODOLOGY

A. Convolutional Neural Network

The convolutional neural network enables machines to think and act like humans. A convolutional neural network has seven layers with an input layer consisting of a 28 by 28 pixel image with 784 neurons as input data. Whenever an input image is provided, it has to go through two phases: feature learning and classification. In feature learning, convolution, and RELU are used as activation Functions that improve model performance. The second layer is the pooling layer. Its function is to reduce the output information from a convolutional layer. The third layer is the flattened layer. The task is to transform a 2D feature map into a 1D feature vector, allowing management of the data in fully connected layers. The fourth layer is the fully connected layer, also known as the dense layer. The output layer of the network consists of 10 neurons and determines the numbers 0-9 to improve the performance of the model. The output layer uses the Soft Max activation function that sorts the output.

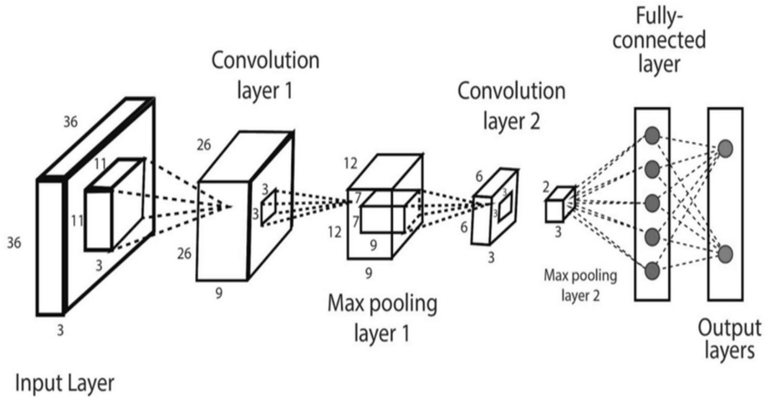


Fig : A Seven layered convolutional neural network.

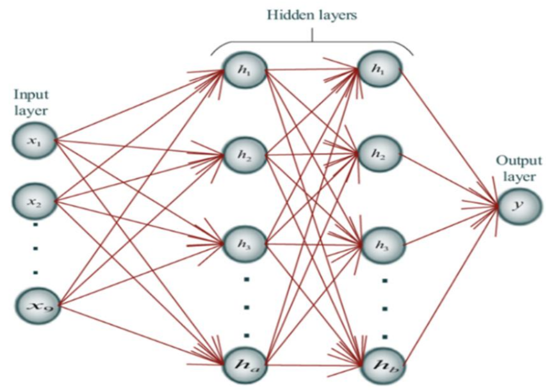


Fig : Hidden layers

B. Convolution layer

CNN uses images as inputs and assigns weight and bias to different parts of the images, thus differentiating between them. It is possible to capture the overlay portions of images using a ConvNet, and the image is processed by taking a three-dimensional or five-dimensional convolution of each pixel of the image, and the convolutions are then used to create new features by processing the entire image. The weights and biases of each convolution are applied to the images after each convolution.

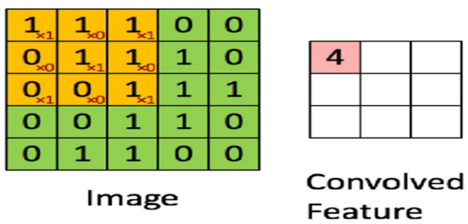
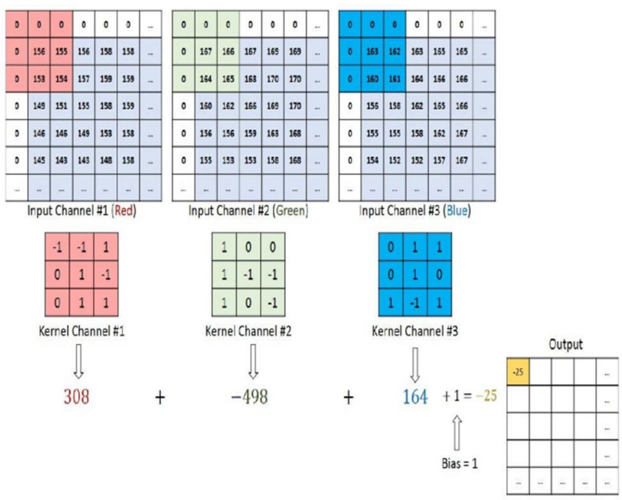


Fig : Convolution layer of Kernel

C. Activation Map and Filter :

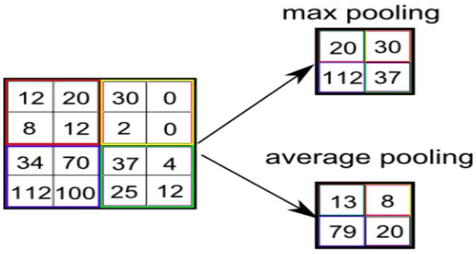
An activation map represents how various linear algebraic operations progress through the various layers of the network as images progress through. For creating activation maps, User need to decide the size of an convolution filter and then first activation map results from shifting the kernel from the top and bottom to the entire image. Process another filter and another image through the entire image to predict the outputs. Those series of outputs are called as Convolution activation maps.



Using convolutional filters, the RGB images have been processed and they have been processed in a multichannel manner. Images were sided over the convolutional process and sided over the image to create the multiple feature maps. This filter is a standard one that we use whenever we need to process the convolutional images and to create features.

D. Pooling layer :

Pooling process is done after completion of the activation layer. Images will be shrinked to smaller size.



E. Activation Function :

The activation functions also allow the models to predict the outcome of a variable by analyzing the images through the final outputs. We use RELU and sigmoid activation functions in these accident detection models. Rectified linear units are used for predicting continuous variables from 0 to infinity. By using activation function, we can predict the output based on the input variables.

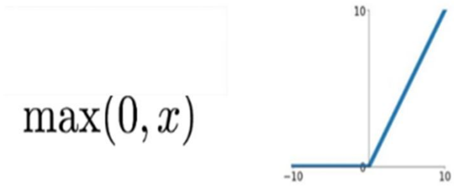


Fig : RELU function

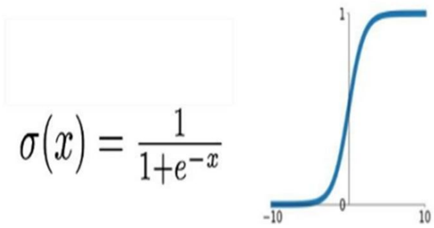


Fig : Sigmoid Function

Sigmoid activation function is commonly used to predict the probability as an output. Since probability exists only between 0 and 1.

F. Fully Connected Layer :

The fully connected layer is the final layer where classification actually takes place. A single list is created by taking our filtered and shrunk images.

We use the Sequential api of keras library to build the cnn model. A Sequential model API creates deep learning models by creating instances of the Sequential class and adding layers to them. Adam optimizer is also used to improve neural network accuracy by adjusting learnable parameters in neural networks.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
batch_normalization (BatchNo (None, 250, 250, 3)       12
_________________________________________________________________
conv2d (Conv2D)              (None, 248, 248, 32)      896
```

max_pooling2d (MaxPooling2D)	(None, 124, 124, 32)	0
conv2d_1 (Conv2D)	(None, 122, 122, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 61, 61, 64)	0
conv2d_2 (Conv2D)	(None, 59, 59, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 29, 29, 128)	0
conv2d_3 (Conv2D)	(None, 27, 27, 256)	295168
max_pooling2d_3 (MaxPooling2	(None, 13, 13, 256)	0
flatten (Flatten)	(None, 43264)	0
dense (Dense)	(None, 512)	22151680
dense_1 (Dense)	(None, 2)	1026

===

Total params: 22,541,134
Trainable params: 22,541,128
Non-trainable params: 6

Above diagram shows the output after defining the CNN model by using the keras and tensor flow.

IV RESULTS

This model gives the results for training and validation during the training. As shown in Fig, the CNN models give the resulting training accuracy of 94.94% and a validation accuracy of 87.76% and Fig 3 shows the loss for the training and validation. In this CNN model, we have used 20 epochs for the training as the available dataset has more images.

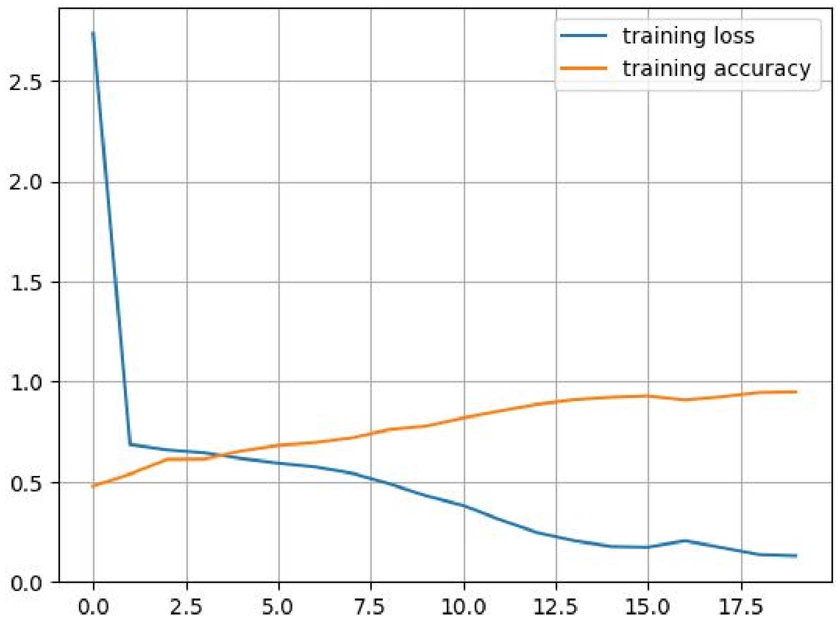


Fig : Training loss and Training accuracy

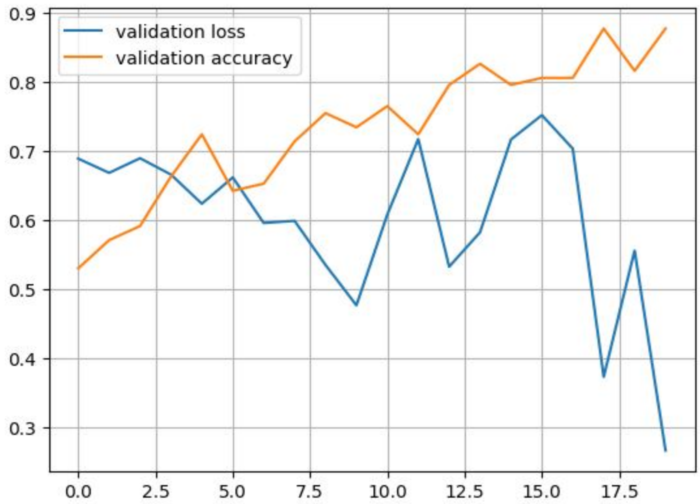


Fig : Validation loss and Validation accuracy

A randomised batch of test data was formed to check the test accuracy. Since the batch size is 64, 5 randomised batches were created. Below figure shows the results of testing.

V. CONCLUSION

Computer vision is a rapidly growing field, and there are several methods for solving classification tasks. At this stage, the model alone cannot be used in the real world without a human operator to mitigate the errors, as the problem requires over 99% accuracy with near zero false positives. Max-pooling levels seem to have mitigated the need for object localisation to some extent, as evidenced by the fact that the same accuracy is maintained even if the accident is in a different part of the image. With a more rigorous and diverse dataset, the accuracy of the CNN and the environment in which it provides satisfactory results could improve. Using spatio-temporally annotated data with an RNN could lead to a more natural solution to the problem. One way to improve the current method would be to use a probability heat map with the CNN to represent the area of interest, i.e. the accident.

Fig : Results of testing

REFERENCES

[1] Lee, K. B., & 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).

[2] Gokul Rajesh, Amitha Rossy Benny, Harikrishnan A, James Jacob Abraham and Nithin Prince John "A Deep Learning based Accident Detection System " 978-1-7281-4988-2/20/$31.00 ©2020 IEEE.

[3] Sreyan Ghosh; Sherwin Joseph Sunny; Rohan Roney "Accident Detection using Convolutional Neural Networks" 2019 International Conference on Data Science and Communication.

[4] https://www.kaggle.com/datasets/ckay16/accident-detection-from-cctv-footage

[5] https://www.geeksforgeeks.org/introduction-convolution-neural-network/

[6] https://www.researchgate.net/publication/364325254_Deep_Learning_Applied_to_Road_Accident_Detection_with_Transfer_Learning_and_Synthetic_Images.