Automatic Accident Detection Using Convolutional Neural Network and Internet of Things

Sathvika Kotha

CISCO, CX Group, Bengaluru, India. sathvikakotha910@gmail.com.

Abstract— Accidents are a major cause of concern anywhere in the world. According to the road transport and highways ministry's report on Road Accidents in India, 2018, accidents are one of India's leading causes of mortality, accounting for around 64.4 percent of all deaths, and this number is steadily rising [1]. Identifying these accidents with the help of technology requires substantial research. This article mainly focuses on finding the best technology that is required for detecting traffic accidents. With the help of a 360 degrees surveillance camera, images such as automobile wrecks, blood, or a person laying on the road with no movement can be detected. This article discusses the technology and algorithm used to recognize and process images, as well as the technology that is utilized to alert hospitals for rapid assistance and to notify the victim's dependents or emergency contacts.

1. INTRODUCTION

An accident is an unexpected event that typically happens suddenly and might result in injury, loss, and harm. Accidents usually involve human beings and lead to chronic disability or injury to a significant percentage of people in the world every year. Accidents may occur in various places, like at home, while traveling, in the workplace, in the hospital, or on the sports field. Many accidents lead to property damage or destruction. Many accidents can be avoided or prevented by taking proper safety precautions and being attentive to one's movements and surroundings. Ambulances and Hospitals play a significant role at the time of accidents. A victim's life is determined by how quickly we bring him or her to the hospital. We can save the victim's life if we get him or her to the hospital promptly; else, saving the victim's life can be extremely difficult.

2. PROBLEM DESCRIPTION

We know that Millions of people die every year in automobile accidents. According to the World Health Organization. (WHO), 1.35 million people die each year in car accidents around the world [2]. Victims of accidents may have a chance to live if they are transported to the hospital in an appropriate timeframe, but the greater part of them die because they are not sent to a medical center in a quick time. According to a BBC news article, after an accident in India, no one comes forward to help because people are afraid of being wrongly accused and of becoming stuck as a witness in a court case in India, where legal proceedings are famously lengthy [3]. Accidents might also happen in isolated or rural regions where no one will be around to contact an ambulance for assistance.

Scenario-1: Kanhaiya Lal begs out for aid, but automobiles swerve right past him. His young son and his wife and infant daughter's sprawled bodies lie next to the damaged motorcycle on which they had all been riding seconds before. Many Indians were disturbed by the extensively publicized CCTV film of this scenario, which showed the pain of a family of hit-and-run victims in northern India and the seeming indifference of passers-by. The family was finally rescued by some bikers and police, but it was too late for Lal's wife and daughter. Their killings triggered a national discussion about bystander involvement [3].

Scenario-2: The surveillance video clip shows a crowd of witnesses encircling Vinay, a 20-year-old boy, and doing nothing when he was hurled from his motorcycle in east Delhi by a fast automobile, and the clip [3].

Scenario-3: A man was driving to a construction site when he was hit by a car; as it was a distant location, no one was able to assist him because hardly anyone passes through that area.

3. PROPOSED SOLUTION

The street poles are equipped with 360-degree cameras. As mentioned in the block diagram in Figure 1, the surveillance camera and GSM/GPS module are connected to Raspberry Pi. When an accident occurs, a camera mounted on a street pole obtains a picture of the victim, which is then analyzed using image processing and CNN. It then delivers the images and notifications to the control room after it has completed its analysis. With the help of an application, the control room authorizes it and sends requests to the nearest hospitals. Once one hospital has accepted the request, the requests for the remaining hospitals are automatically terminated. The ambulance is dispatched to the site by the hospital that accepted the request. Following the victim's admission to the hospital, a call will be made to inform the victim's dependents about the victim's status and provide hospital information as can be seen in Figure 2. This method can be employed not only in the event of an accident but also in the event of an identifiable individual lying on the road due to health issues.



Figure 1. Block Diagram



Figure 2. Flow chart of the proposed solution

4. INTERNET OF THINGS (IoT)

The Internet of Things (IoT) refers to the network of physical objects that are equipped with sensors, computing power, software, and other technologies to connect and exchange data with other devices and systems over the Internet or other communication networks.

4.1 Raspberry Pi

The Raspberry Pi Foundation, a UK nonprofit that aspires to educate people in computing and make computing education more accessible, has created a series of single-board computers known as the Raspberry Pi (see **Figure 3**). The Raspberry Pi was first published in 2012, and since then, various revisions and modifications have been developed. People across the world use Raspberry Pi to code, build hardware projects, home automation, edge computing, and industrial applications. The cost of Raspberry Pi is low and runs on Linux operating system. It has a set of general-purpose input/output ports for connecting electronic modules or components and working on the Internet of Things (IoT). Initially Pi had a singlecore processor 700MHz with 256MB of RAM, whereas the latest model has a quad-core 1.5GHz with 4GB of RAM.



Figure 3. Raspberry Pi

4.2 GPS Module

The tiny processors and antennas that the GPS modules are made of, receive data directly from satellites via specific radio frequencies. It will then get timestamps from all visible satellites, as well as other information. Arduino and Raspberry Pi support GPS modules (see Figure 4).



Figure 4. GPS Module

4.3 GSM Module

A GSM modem is also known as a GSM module. It is a physical device that uses GSM mobile phone technology to connect to a remote network. In the eyes of the mobile phone network, the GSM module is identical to a typical mobile phone. This module has a SIM to identify *itself* to the network. GSM modems provide TTLlevel serial interfaces to their hosts. Embedded systems are the most common application for them. This module helps us in sending messages as well as provides the internet as it has a SIM (see **Figure 5**). If we connect this module to Raspberry Pi, we can write a python code and send messages or mail to the intended user.



Figure 5. GSM Module

5. DEEP LEARNING

Deep learning has had a significant impact on various domains of technology in recent years. One of the most talked-about technologies in the industry is computer vision. With the use of computer vision, computers can understand images and movies on their own. For example, computer vision is used in the automotive industry (autonomous vehicles), public security (facial recognition), transportation (violations detection), and traffic flow analysis. Image processing is the core of computer vision.

Image processing is the process of converting an image to a digital format and then executing specific operations on it to extract relevant information or an improved image. When applying predefined signal processing methods, the image processing system treats every image as a 2D signal.

Image processing broadly follows three steps:

- 1. Import the image with the help of image capturing software.
- 2. Evaluate and manipulate the image.

3. Report the result based on image analysis.

Deep learning is also a technique that mimics the human brain. In the 1950s and 1960s, researchers and scientists came up with a question: Can we make a machine learn and understand the data on its own like how a human learns and understands the data? This led to the invention of neural networks. The first simplest type of neural network is "perceptron". As it is not able to learn the data properly, In the 1980s, Jeoffrey Hinton invented the concept of Back Propagation. Many companies and people developed different efficient applications with the help of backpropagation and neural network architectures.

There are different types of neural network architectures:

- 1. Artificial Neural Network (ANN).
- 2. Convolutional Neural Network (CNN).
- 3. Recurrent Neural Network (RNN).
- 5.1 Artificial Neural network (ANN)

In **Figure 6**, the input is fed into the first layer which yields output to the neurons in the next layer and goes on which provides the result. The output is the result that is predicted either in 0 (non-accident) or 1(Accident). The neuron in each layer computes a function called the activation function. The activation function determines whether to activate the neurons to pass the signal to the neurons in the next layer.

The link between neurons of consecutive layers has a weight parameter associated with it. This weight impacts the output of each layer and the final layer output. Initially, the weights are assigned randomly and updated at each layer iteratively to yield the final output correctly. The main building blocks of neural networks such as a layer, neuron, weight parameters, activation function, and the optimizer will enable the neural network to select the suitable weight and produce the correct outcome.



Figure 6. Artificial Neural Network

5.2 convolutional Neural Network

The convolutional neural network is a type of deep neural network. This kind of neural network is used when the input is in the form of an image. For instance, object detection, face classification. object classification, object recognition, and so on. Let's understand how we as humans understand and identify the image. The human brain is divided into four parts in which the cerebral cortex which is also known as the visual cortex is responsible for seeing the images and identifying them. CNN has multiple layers where each layer is responsible for performing each function. For instance, layer 1 is responsible for finding the edges of an image. Layer 2 is responsible to find if the object in the image is moving. Layer 3 is responsible to find if there are any other objects in the image and so on. In CNN these layers are known as filters. In image processing, black and white and grayscale images are treated as 2D input, and color images are treated as 3D input, convolution operations are applied to those images to extract features using the backpropagation method.

In **Figure 7**, the input is given to the hidden layer. In each hidden layer, we perform convolution and activation functions on the image and send it to the next hidden layer, and so on. Once the convolution part is completed then the output is flattened which is the output of convolution layers gets converted into a feature vector for connecting it to the final layer.



Figure 7. Convolutional Neural Network

5.3 Recurrent Neural Network

RNN is another type of neural network in which the output from the previous step is used as input in the next step. In general, all the inputs and outputs in standard neural networks are independent of one another. However, in some circumstances, such as when predicting the next word of a phrase, the prior word is necessary. Sentences are sequences of words, the previous words must be remembered. RNN was created for this purpose. It uses a hidden layer to remember certain information about a sequence, which is the most essential element of RNN.

6. ACTIVATION FUNCTION

The activation function is used in the process of converting the input signal to the output signal. It can decide whether to activate the specific neuron or not. The neural network could not learn complex network mappings without activation function as it is the one that converts the model to non-linear. The linear model has only limited capability and can't understand complicated images, speech, and so on. The activation functions are of two types: Linear activation function and Non-Linear activation function.

6.1 Linear Activation Function

This function resembles a linear line which doesn't work for complex parameters that are fed into neural networks (see Figure 8).



Figure 8. Linear Activation Function

6.2 Non-Linear Activation Function

Non-Linear activation functions are suitable for learning and generating complex network mappings.

6.2.1 Sigmoid Activation Function

The sigmoid function is used in the logistic regression model of machine learning, and it ranges between (0,1). It is also used in neural networks, and it ranges between (0,1) Sigmoid functions possess low convergence and vanishing gradient problems.

The sigmoid function is given by,

$$\phi(x) = \frac{1}{1 + e^{-x}}$$

Figure 9 shows the graph of the sigmoid function.



Figure 9. Sigmoid Activation Function

6.2.3 Rectified Linear Unit Activation Function

The most widely used activation function is Rectified Linear unit (ReLU). This function enables the mode to train with high performance. ReLU function provides better performance than the sigmoid function as it overcomes vanishing gradient problems (see **Figure 10**). This function does not activate all the neurons at the same time. The neurons get activated only if the output is greater than 0. The ReLU function is given by,





Figure 10. ReLU Activation Function

6.2.3 Softmax Activation Function

The Softmax function is used on the output layers of neural network models. This function predicts the probability distribution.

The softmax function is given by,

$$S(y)_{i} = \frac{\exp(y_{i})}{\sum_{j=1}^{n} \exp(y_{j})}$$

7. OPTIMIZER

The loss function is a function that helps in measuring if the model is working properly and providing accurate output. This function computes the difference between the value predicted by the model and the actual value. It is also known as the cost function or error function.

A common loss function is given by,

$$Loss = (y^{-}y)^2$$

In every epoch, we try to reduce the loss value. To reduce the loss value, we use optimizers.

7.1 Gradient Descent Optimizer

Gradient Descent will update all the weights in such a way to make the value of y^{\wedge} similar to y. The learning rate is small so it takes much time to reach the global minima but if we take a higher learning rate value then the value will jump from one side to the other side and will never reach the global minima point. Here, we give all n data points once at a time as input to the neural network.

7.2 Stochastic Gradient Descent Optimizer

Stochastic gradient descent is similar to gradient descent but here we give only one record as an input to the neural network. If we give K number of records as an input, then it is known as mini-batch stochastic gradient descent. Gradient descent follows a straight path to reach the global minima, but SGD goes in a zig-zag way which takes more time to reach the global minima. There might be a chance of noisy data getting added while reaching the minimum point. SGD with momentum concept is used to alter the noisy data. An exponential moving average helps in removing noisy data.

7.3 Adaptive Gradient Descent Optimizer

ADAM is the mostly used optimizer in a neural network. The learning rate differs for each hidden layer and neuron. This optimization technique is more efficient and requires less memory.

8. IMPLEMENTATION METHODS

8.1 Using the CNN Model

A basic CNN model is used for implementation with specific batch size, image height, and width. Firstly, the data is divided into three: training, validation, and test dataset. Secondly, the images in the data set are preprocessed and then used these data to train the model. Finally, check the accuracy of the train data and validation data.

Table I shows us the amount of loss and accuracy of train data as well as validation data's loss and accuracy in each number of epochs. 5 epochs, ADAM as an optimizer, sparse categorical entropy as a loss function, and ReLU

as an activation function are used at the time of implementation.

Table 1. Loss and Accuracy of Train andValidation Data

Epoch	Loss	Accuracy	Val Loss	Val Accuracy
epoch 1/5	5.4577	0.5436	1.6529	0.4694
epoch 2/5	0.6433	0.6523	0.9371	0.5000
epoch 3/5	0.5999	0.6802	0.9210	0.4796
Epoch 4/5	0.5447	0.7143	0.7910	0.5510
Epoch 5/5	0.4574	0.7901	0.8555	0.5102

Figure 11 shows the accuracy of train data and validation data for a specific number of epochs and Figure 12 shows us the loss of train data and validation data for a specific number of epochs.



Figure 11. Train and Validation Accuracy



Figure 12. Train and Validation Loss

From Table 1, we can see that the accuracy rate of validation data is 51% but for better accuracy, we have totrain our model more properly so let's see the second implementation method which CNN is using the VGG16 model.

8.2 Implementation of CNN using VGG16 model and softmax function

In this method for the output layer, the softmax activation function is used and trained the model is with the training data set.

Table 2 shows the amount of loss and accuracy of train data as well as validation data's loss and accuracy in each number of epochs.

Table	2.	Loss	and	Accuracy	of	Train	and
Valida	tio	n Data					

Epoch	Loss	Accuracy	Val Loss	Val Accuracy
Epoch 1/ 5	0.9948	0.5588	0.6791	0.6837
Epoch 2/ 5	0.5580	0.7143	0.3739	0.8061
Epoch 3/ 5	0.4254	0.8028	0.3343	0.8367
Epoch 4/ 5	0.4148	0.8129	0.2926	0.8776
Epoch 5/5	0.3346	0.8496	0.3051	0.8571

Table 3 describes the detailed VGG16 model summary using CNN. For hidden layers, ReLU and Max pooling was performed and SoftMax on the dense layer to get a better output.

Table 3. Model Summary

Layer(type)	Output Shape	Param#
input 1 (Input Layer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1 pool (MaxPooling2D)	(None, 112, 112, 64)	0
Block2_conv1 (Conv2D)	(None, 112, 112, 64)	73856
Block2 conv2 (Conv2D)	(None, 112, 112, 64)	147584
Block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3 conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3 pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4 conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5 conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense 1 (Dense)	(None, 2)	50178
Total params: 14,764,866		
Trainable params: 50,178		
Non-trainable params: 14,714,		
688		

Figure 13 shows the accuracy of train data and validation data for a specific number of epochs.



Figure 13. Accuracy of train and validation data

Figure 14 shows us the loss of train data and validation data for a specific number of epochs.





9. LIBRARIES IMPORTED

9.1 Tensorflow

Tensorflow is an open-source platform for Machine Learning, Deep Learning, and Artificial Intelligence. It is designed in python programming language, so people feel easy to understand. It is a software library developed by google brain so the machine learning and deep learning concepts can be implemented easily [4].

9.2 Matplotlib

Matplotlib is an open-source library and was created by John D. Hunter. Matplotlib is a python visualization toolkit with a low-level graph plotting library [5]. Matplotlib is largely written in python for platform compatibility, with a few segments written in C, Objective-C, and JavaScript.

9.3 Keras

Keras was developed by Francois Chollet, and it is a deep learning framework for python. It supports different backends and platforms. It provides a python interface for artificial neural networks [6].

9.4 Pandas

Pandas is a Python package that allows you to work with large datasets. It offers tools for data analysis, cleansing, exploration, and manipulation. It was Wes McKinney who came up with the name "Pandas" in 2008. The word refers to both "Panel Data" and "Python Data Analysis."

9.5 NumPy

NumPy is a Python module for interacting with arrays. Travis Oliphant invented it in 2005. It also provides functions for working with matrices, Fourier transforms, and linear algebra. It is an open-source project, so is free to use. Numerical Python is referred to as NumPy.

10. DATASET USED

For the implementation, we used an accidentdetection dataset from Kaggle [7]. As is commonly done, the Dataset is split into three parts: Training dataset, Validation dataset, and Test dataset. Each dataset comprises two folders they are accident and non-accident (see **Table 4**).

Table 4. Dataset

	Accident	Non-Accident	Total
Training Dataset	369	422	791
Validation Dataset	46	52	98
Test Dataset	47	54	101
Total	462	528	990

11. ALGORITHM

1. Import required libraries like pandas, NumPy, Keras, Tensorflow, and so on.

2. Define batch specifications like batch size, image height, and width.

3. Load train data set.

4. Load validation data set.

5. Load test data set.

6. Define CNN using the VGG16 model.

7. Train model using train and validation data set. 8. Check accuracy.

9. Compare the loss and accuracy of train data and validation data using matplotlib graphs.

CONCLUSION

In this article, we could see that two implementation methods are followed. CNN using VGG16 gives better accuracy when compared to CNN. The loss in the second implementation method is gradually decreased. The accuracy of 85% is decent but if we would like to get more accuracy, then we can take big data set and work on it. Making model overfit is also not recommendable.



- [1] Nancy. P, Dilli Rao. D, Babuaravind. G, Bhanushree. S, "HIGHWAY ACCIDENT DETECTION AND NOTIFICATION USING MACHINE LEARNING', Vol. 9, Issue. 3, March 2020.
- [2] "Road traffic Injuries" <u>https://www.who.int/news-room/fact-heets/detail/road-traffic-injuries</u>
- [3] "If no one helps you after a car crash in India, this is why", BBC News, <u>https://www.bbc.com/news/magazine-36446652</u>
- [4] <u>"Tensorflow Documentation"</u>, <u>https://www.tensorflow.org/</u>
- [5] <u>"</u>Matplotlib Documentation", <u>https://matplotlib.org/</u>
- [6] "Keras Documentation", https://keras.io/
- [7] "Accident Detection data set" <u>https://www.kaggle.com/code/vanvalkenber</u> <u>g/cnn-for-accident-detection-83-val-</u> accuracy/data

AUTHOR BIOGRAPHY



Sathvika Kotha received B.Tech degree in Computer Science and Engineering from Mallareddy Engineering College for Women. She is currently working at CISCO Systems Pvt Ltd. A network consulting

engineer specializing in troubleshooting in Data Center storage and switching devices and have hands-on experience in automating networking applications using python, ML, web technologies, and virtualization. She is a certified Cisco Certified Network Associate (CCNA) and Cisco Certified Devnet Associate. Her research interests include automation using Machine Learning, Deep Learning, Data Science, and IoT.



FEEDFORWARD

AMA (Ask Me Anything) Session Björn Schuller, Imperial College London

Held on Tue, March 22nd, 2022, 12 noon PT

Moderator: Vishnu S. Pendyala, San Jose State University

Topics Covered include:

Artificial Intelligence, Emotional AI, Artificial Emotional Intelligence, Machine Learning, Affective Computing, Deep Learning, Audio AI, Reinforcement Learning, Digital Health, CoVID, Corona Virus, MFCC, Mel-frequency Cepstral Coefficients, Coronavirus Mutation, Imbalanced Dataset, Speech Emotion Recognition, Audio Emotion Detection, Unsupervised Learning, Semi-Supervised Learning, Active Learning, Lie Detector, Asymptomatic Depression, Federated Learning, Quantum Computing, Multitasking Models, Support Vector Machines, SVM, Loss Function, Institutional Review Board, IRB, Double-Blind Reviews, arxiv, Career In Machine Learning

Recording: https://ieeetv.ieee.org/video/ama-with-bjornschuller-artificial-intelligence

VOLUME 1, NUMBER 1

IEEE CS SCV Chapter's

NFIC 2022

New Frontiers in Computing Half-Day Virtual Seminar Saturday, May 21, 2022 4:00 PM - 8:00 PM (PDT) Stay tuned for more details

JANUARY - MARCH 2022