# Intelligent Surveillance System Using Deep Learning

**Ms.E.L.Jaya Kamachi Elangovan[1], R.Keerthana[2], N.Lakshmiraj[3], R.Nivetha[4], B.Suraj Kumar Sharma[5]**
[1, 2, 3, 4, 5] Coimbatore Institute Of Technology, Coimbatore-641 014

***Abstract-*** *Nowadays the highways accident rate is at its peak and the Indian government is forcefully working to slower the accident rate. Here we have designed a model called Intelligent Surveillance System to work on the live video stream of CCTV footage to detect the occurrence of accident and forward the clipped video footage of accident occurred to the nearest police station and ambulance or emergency service for the earliest facilitation of service to take place with live location provided.*

*The model is initially being fed with the procured dataset of accidents which is then pre-processed. The model is trained using Machine Learning under the domain-Convolution Neural Network. We have designed the model with improved accuracy from the existing model that further optimises the model. This subsequently helps in the real time processing of the live video streams of CCTV footage. Thus, the Intelligent Surveillance System mainly focuses on the swift processing of the live accidents to reach the emergency service, so as to comparatively reduce the rate of death crisis mainly in the accident-prone area.*

## I. INTRODUCTION

### 1.1OVERVIEW

An accident, also known as an unintentional act, is an undesirable, incidental, and an unplanned event that could have been prevented had circumstances leading up to the accident been recognized, and acted upon, prior to its occurrence. Most scientists who study unintentional injury avoid using the term "accident" and focus on factors that increase risk of severe injury and that reduce injury incidence and severity.

Our project is used to recognize accident from surveillance network of CCTV cameras on roads and to identify traffic accidents and report the location and the video clip to the nearest ambulance and police station. We employ convolutional neural network method for image processing and to train model for accident detection. In case of accident occurring, the surrounding people have to call or intimate ambulance service which may take longer time. Or even some people hesitate from being involved in such emergency situation. Hence our project mainly focuses in faster intimation to ambulance service. It will be helpful during occurrence of accident at night time.

### 1.2. DATASET

We took the dataset from **KAGGLE** which contained snapshots from a video captured by a CCTV camera in the highways and remote areas. The training set has **14000** labelled samples with equa distribution among the classes and **4192** unlabelled test samples. An input video is taken and It consists of two categorization of images- accident and non-accident.

### 1.3.DATA PREPROCESSING

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further process.

**1.Data Cleaning**

Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.

**2.Data Integration**

Data with different representations are put together and conflicts within the data are resolved.

**3.Data Transformation**

Data is normalized, aggregated and generalized.

**4.Data Reduction**

This step aims to present a reduced representation of the data in a data warehouse.

**5.Data Discretization**

Involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

### 1.4. FEATURE SELECTION

This becomes even more important when the numbers of features are very large. Feature selection has been the focus of interest for quite some time and much work has been done. With the creation of huge databases and the consequent requirements for good deep learning techniques, new problems arise and novel approaches to feature selection are in demand. Feature selection is the use of specific variables or data points to maximize efficiency in this type of advanced data science. It reduces the complexity of a model and makes it easier to interpret. It improves the accuracy of a model.

Major reasons to use feature selection are:

1. 1.It enables the deep learning algorithm to train faster.
2. It reduces the complexity of a model and makes it easier to interpret.
3. It improves the accuracy of a model if the right subset is chosen.
4. It reduces over-fitting.

### 1.5 DEEP LEARNING

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is a key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers. Deep learning is getting lots of attention lately and for good reason. It's achieving results that were not possible before. In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

#### 1.5.1 CONVOLUTION NEURAL NETWORK

A Convolution Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area.

## II. LITERATURE SURVEY

**2.1 PAPER: Detection and Classification of a Moving Object in a Video Stream**
**Author:** Mahamuni P. D, R. P. Patil, H.S. Thakar

**Description:**

Background subtraction methods are widely exploited for moving object detection in videos in many applications. This approach proves fast, flexible and precise in terms of pixel accuracy. The implementation of the background subtraction algorithm is done in two domains code is written in MATLAB, then using Simulink blocks sets.

**Advantages:**

- The proposed classification function gave an accurate result in most of the frames.
- Simple approach to classify moving objects from a video stream into vehicles and human beings was proposed.
- The main advantage of the classification function is that it is implemented after background subtraction function is implemented which makes detection of moving object easier.

**Disadvantages:**

- The bounding boxes do not contain all moving objects.
- The classification worked only for few cases with a smaller number of objects in the frame.

**2.2 PAPER: Moving object detection using background subtraction algorithm using Simulink**
**Author:** Asim R. Aldhaheri and Eran A. Edirisinghe

**Description:**

Moving objects are firstly detected from the background using a background subtraction technique which is implemented in a MATLAB environment. Edge detection of moving objects was performed using Canny or Prewitt operations, while bounding boxes were implemented over the moving objects. Classifying the moving object into humans and vehicles was accomplished by finding the height-width ratio of the bounding box around the moving object in each frame and estimating the speed of the moving object from two consecutive frames in the video stream.

**Advantages**:

- Improved segmentation results.
- Being able to extract additional information such as frame difference.
- Provides better information of the moving object in a video.

**Disadvantage**:

- Works well only for a single object in a frame.
- Requires more time to compute results.

**2.3 PAPER: A Study on Video Surveillance System for Object Detection and Tracking**
**Author:** Pawan Mishra

**Description:**

Visual surveillance System is basically used for analysis and explanation of object behaviours. It consists of static and moving object detection, video tracking to understand the events that occur in scene. Different methods such as background subtraction, statistical method, and temporal frame differencing are used for the detection of moving objects. Point tracking, Silhouette Tracking and Kernel tracking had been used for the object tracking purpose

**Advantages:**

- The survey describes various object detection, tracking techniques and algorithms.

**Challenges described in the survey:**

1. Illumination changes.
2. Dynamic Background.
3. Occlusion.
4. Presence of Shadow

**Limitations of The Survey:**

- Information provided by the survey is not useful where higher key frames are required for object detection.
- It is not suitable to track moving object for denser environment such as crowds of moving people.

**2.4 PAPER:Smart Surveillance: Applications, Technologies and Implications**
**Author:** Arun Hampapur, Lisa Brown, Jonathan Connell, Sharat Pankanti, Andrew Senior, Yingli Tian.

**Description:**

Today's video surveillance systems while providing the basic functionality fall short of providing the level of information need to change the security paradigm from "investigation to pre-emption". Automatic visual analysis technologies can move today's video surveillance systems from the investigative to preventive paradigm. The outputs of video cameras are recorded digitally and simultaneously analysed by the smart surveillance server, which produces real time alerts and a rich video index. The types and parameters of the alerts are user configurable.

**Advantages:**

- The ability to pre-empt incidents - through real time alarms for suspicious behaviours.
- Enhanced forensic capabilities -- through content-based video retrieval.
- situational awareness – through joint awarenessof location, identity and activity of objects in the monitored space.

**Disadvantages:**

- Technical Challenge include challenges in robust object detection, tracking objects in crowded environments, challenges in tracking articulated bodies for activity understanding, combining biometric technologies like face recognition with surveillance to achieve situation awareness.
- Evaluating performance of video analysis systems requires significant amounts of annotated data. Typically, annotation is a very expensive and tedious process. Additionally, there can be significant errors in annotation. All of these issues make performance evaluation a significant challenge.

## III. SYSTEM ANALYSIS

### 3.1 EXISTING SYSTEM

Accidents occurring in highways and remote areas are highly risky. As it is difficult to contact the emergency service and security at the hectic time, the goal of the project is to develop a intelligent surveillance system. The existing system has a stable CCTV coverage that does not focus on the real time intimation service. Moreover, the predictions of the accident are not to the level accurate. The existing system primarily employs background subtraction and temporal frame differencing for object detection which has comparatively slow computation result. It also employs point tracking and kernel tracking methods for object tracking. The classification implemented here could only identity object that has very few objects in the frame. But the actual sequence of accidents contains a plenty of objects moving around.

### 3.2.PROPOSED SYSTEM

The main objective of the project is to predict the accident that happen in the remote highways using CNN. Our attempt is to develop an accurate and robust system for detecting accident and reach the emergency services. Since video processing is a lengthy process, here videos are converted into frames for faster detection and improved accuracy. Data is pre-processed by converting the original image into grayscale image and resize it. The next step is to load the images. The images are segregated as training set and testing set. The next step is to develop CNN modelwith four activation layers,two dense layers**,** two convolution 2D layers and two max pooling layers. The developed CNN model is used to classify the input images as accident and non-accident as specified in features.With this,the further intimation is provided to emergency service to reach site only if it is detected as an accident. The intimation involves the sending of a clipped image of the accident and the auto detected location to the nearest service. Our CNN model has faster execution in the real time and enhanced accuracy of 93%.

## IV. SYSTEM IMPLEMENTATION

### 4.1. OVERVIEW

Detecting the occurrence of accident is useful in the current scenario of the world in minimising the abnormalities of injury in the accident .Thus the Intelligent Surveillance System mainly focuses on the swift processing of the live accidents to reach the emergency service, so as to comparatively reduce the rate of death crisis mainly in the accident prone area.

### 4.2. MODULES

The modules used in the system are,

- Creating and pre-processing the dataset
- Training the customised CNN model and Testing the model
- Location collector using selenium web driver
- sending to the customized area using message sending API

#### 4.2.1. Creating and pre-processing the dataset

For training our customized CNN model we develop dataset by collecting the videos related to roadside accidents from the video streaming sites such as YouTube, then to create the image dataset from that videos by using the OpenCV tool we converted the video sources to the frames by frame image dataset. Pre-processing of dataset includes Feature Selection and Discretization. Feature selection and dimension reduction are common data mining approaches in large datasets. Here the high data dimensionality of the dataset due to its large feature set poses a significant challenges these helps to improve the accuracy of the classifier.

#### 4.2.2. Training the customised cnn model and Testing the model

**Convolutional Neural Networks (CNNs)** is the most popular neural network model being used for image classification problem. Choosing the CNN model because pre-processing required is much lower as compared to other classification algorithm. thus we created our CNN model for intelligent surveillance system for accident detection and trained our model with our created dataset and increasing the accuracy by analysing with visualizing the accuracy graph using Matplotlib library .Then testing our created model with our testing dataset to check whether it detect the occurrence of accident events.

#### 4.2.3. Location collecting by pining in google map

The respective place location is collected by pinning the location in the google map and to be embedded in the CCTV camera system.

#### 4.2.4. Sending to the customized area using message sending API

For our intelligent surveillance system for accident detection we using the certain short message service API such as SMS4INDIA to sending the collected information on the location to the customized centre near by the zone which is built in the system.

#### 4.2.5. One Block VGG model

The one-block VGG model has a single convolutional layer with 32 filters followed by a max pooling layer.

#### 4.2.6. Two Block VGG model

The two-block VGG model extends the one block model and adds a second block with 64 filters.

#### 4.2.7. One Block VGG model with Dropout regularisation

This model is one block VGG model with Dropout regularization. Dropout regularization is a computationally cheap way to regularize a deep neural network. Dropout works by probabilistically removing, or "dropping out," inputs to a layer, which may be input variables in the data sample or activations from a previous layer. It has the effect of simulating a large number of networks with very different network structures and, in turn, making nodes in the network generally more robust to the input

## V. RESULT

Converting the collected video dataset into frames dataset

```
In [1]: import cv2
vidcap = cv2.VideoCapture('Music/video.avi')
def getFrame(sec):
    vidcap.set(cv2.CAP_PROP_POS_MSEC,sec*1000)
    hasFrames,image = vidcap.read()
    if hasFrames:
        cv2.imwrite("Music/image"+str(count)+".jpg", image)     # save frame as JPG file
    return hasFrames
sec = 0
frameRate = 0.5 #//it will capture image in each 0.5 second
count=1
success = getFrame(sec)
while success:
    count = count + 1
    sec = sec + frameRate
    sec = round(sec, 2)
    success = getFrame(sec)
print(count)

123
```

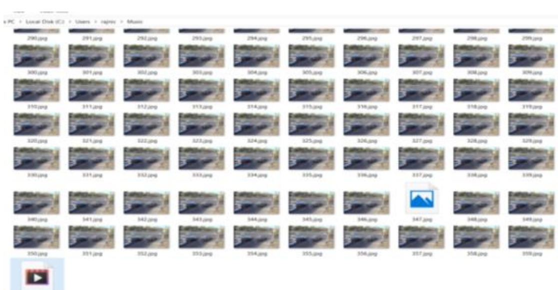

**Fig 5.1: converting the video to frames**

```
In [1]: import tensorflow as tf
import numpy as np
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D

from tensorflow.keras.callbacks import ModelCheckpoint
import pickle
import time

In [2]: import numpy as np
import matplotlib.pyplot as plt
import os
import cv2
from tqdm import tqdm

In [1]: from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.support.ui import WebDriverWait
import time

In [6]: import requests
import json
```

**Fig no:5.2 Importing the libraries**

```
DATADIR = "F:\dataset"

CATEGORIES = ["accident", "notaccident"]

for category in CATEGORIES:  # do ACCIDENT  and NOT ACCIDENT
    path = os.path.join(DATADIR,category)  # create path to ACCIDENT  and NOT ACCIDENT
    for img in os.listdir(path):  # iterate over each image per ACCIDENT  and NOT ACCIDENT
        img_array = cv2.imread(os.path.join(path,img) ,cv2.IMREAD_GRAYSCALE)  # convert to array
        plt.imshow(img_array, cmap='gray')  # graph it
        plt.show()  # display!

        break
    break
```

**Fig no.:5.3 converting the original image into Gray scale image and displaying**

```
In [7]: IMG_SIZE = 150

new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
plt.imshow(new_array, cmap='gray')
plt.show()
```

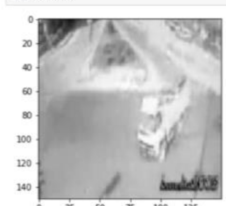

**Fig no:5.4 resizing a particular image**

```
In [9]: training_data = []

def create_training_data():
    for category in CATEGORIES:  # do ACCIDENT  and NOT ACCIDENT

        path = os.path.join(DATADIR,category)  # create path to ACCIDENT  and NOT ACCIDENT
        class_num = CATEGORIES.index(category)  # get the classification  (0 or a 1). 0=ACCIDENT 1=ACCIDENT

        for img in tqdm(os.listdir(path)):  # iterate over each image per ACCIDENT  and NOT ACCIDENT
            try:
                img_array = cv2.imread(os.path.join(path,img) ,cv2.IMREAD_GRAYSCALE)  # convert to array
                new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))  # resize to normalize data size
                training_data.append([new_array, class_num])  # add this to our training_data
            except Exception as e:  # in the interest in keeping the output clean...
                pass
            #except OSError as e:
            #    print("OSErrroBad img most likely", e, os.path.join(path,img))
            #except Exception as e:
            #    print("general exception", e, os.path.join(path,img))

create_training_data()

print(len(training_data))

100%|██████████| 2287/2287 [00:12<00:00, 180.22it/s]
100%|██████████| 6326/6326 [00:32<00:00, 193.33it/s]

8569
```

**Fig no:5.5 pre-processing and adding features to all the images in the dataset**

```
In [11]: import random

         random.shuffle(training_data)

In [12]: for sample in training_data[:20]:
             print(sample[1])

1
1
1
0
1
0
0
1
0
1
0
0
1
0
1
1
1
1
1
1
```

**Fig no.:5.6 shuffling all the images in the dataset**

```
In [15]: import pickle

pickle_out = open("X.pickle","wb")
pickle.dump(X, pickle_out)
pickle_out.close()

pickle_out = open("y.pickle","wb")
pickle.dump(y, pickle_out)
pickle_out.close()

In [19]: pickle_in = open("X.pickle","rb")
X = pickle.load(pickle_in)

pickle_in = open("y.pickle","rb")
y = pickle.load(pickle_in)
```

**Fig no.:5.7 storing the pre-processed images using pickle library**

```
X = X/255.0

model = Sequential()

model.add(Conv2D(256, (3, 3), input_shape=X.shape[1:]))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())  # this converts our 3D feature maps to 1D feature vectors

model.add(Dense(64))
model.add(Activation('relu'))

model.add(Dense(1))
model.add(Activation('sigmoid'))


model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

model.fit(X, y, batch_size=32, epochs=3, validation_split=0.3)

model.save('64x3-CNN.model')

Train on 5998 samples, validate on 2571 samples
Epoch 1/3
5998/5998 [==============================] - 1192s 199ms/sample - loss: 0.5725 - accuracy: 0.7513 - val_loss: 0.4452 - val_accu
racy: 0.8012
Epoch 2/3
5998/5998 [==============================] - 1067s 178ms/sample - loss: 0.2914 - accuracy: 0.8838 - val_loss: 0.2298 - val_accu
racy: 0.9168
Epoch 3/3
5998/5998 [==============================] - 1104s 184ms/sample - loss: 0.1574 - accuracy: 0.9416 - val_loss: 0.1438 - val_accu
racy: 0.9432
```

**Fig no.:5.8 Training CNN model**

```
In [2]: import cv2
import tensorflow as tf

CATEGORIES = ["accident", "notaccident"]


def prepare(filepath):
    IMG_SIZE = 150
    img_array = cv2.imread(filepath, cv2.IMREAD_GRAYSCALE)
    new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
    return new_array.reshape(-1, IMG_SIZE, IMG_SIZE, 1)


model = tf.keras.models.load_model("64x3-CNN.model")
prediction = model.predict([prepare('testing sample.jpg')])
print(prediction)

[[1.]]
```

**Fig no.:5.9 Testing CNN model**

```
import requests
import json

URL = 'https://www.sms4india.com/api/v1/sendCampaign'

# get request
def sendPostRequest(reqUrl, apiKey, secretKey, useType, phoneNo, senderId, textMessage):
  req_params = {
  'apikey':apiKey,
  'secret':secretKey,
  'usetype':useType,
  'phone': phoneNo,
  'message':textMessage,
  'senderid':senderId
  }
  return requests.post(reqUrl, req_params)

# get response
response = sendPostRequest(URL, '47308RPYV4W6ZCX8X5MD11ZF6D211SRS', 'F42PKZ60MQ4FMP6U', 'stage', '9791789124', 'Su911', 'There is an accident at https://goo.g
"""
  Note:-
    you must provide apikey, secretkey, usetype, mobile, senderid and message values
    and then requst to api
"""
# print response if you want
print (response.text)

{"code":"200 OK","total-messages-sent":1,"req-type":"post","remaining-sms":24,"message":"Campaign sent successfully.","usetype":"stage","balacre":"2","status":"su
```

**Fig no.:5.10 Sending the location to the respective number using message sending API**

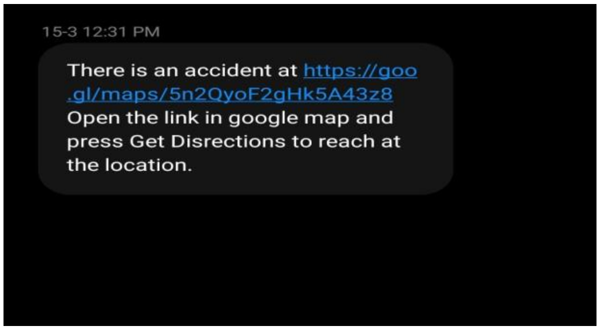


**Fig no.:5.11 intimation message at the time of accident to the number**

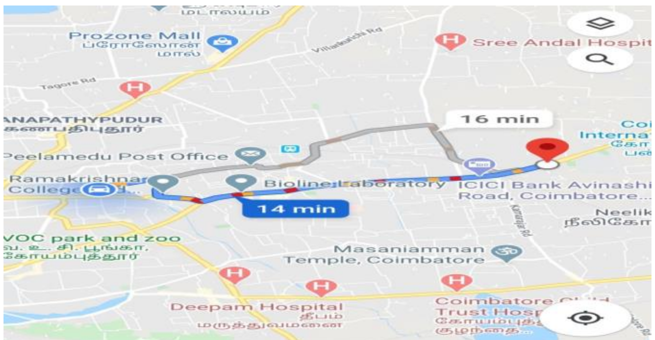


**Fig no.: 5.12 Location tracked from the received link via message.**

```
def define_model():
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(200, 200, 3)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Flatten())
    model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
    model.add(Dense(1, activation='sigmoid'))
    # compile model
    opt = SGD(lr=0.001, momentum=0.9)
    model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])
    return model
```

After training the model the accuracy will return as
Found 3362 images belonging to 2 classes.
Found 1138 images belonging to 2 classes.
> 94.124

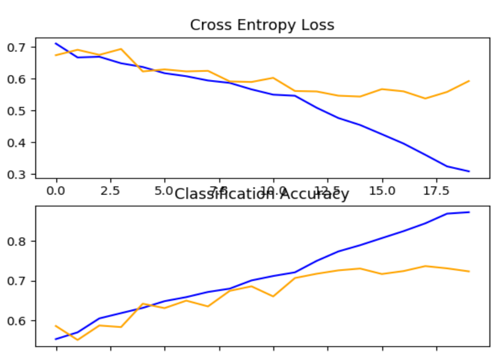


**Fig no.: 5.13 One Block VGG model**

```
In [99]: # define cnn model
def define_model():
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(200, 200, 3)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Flatten())
    model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
    model.add(Dense(1, activation='sigmoid'))
    # compile model
    opt = SGD(lr=0.001, momentum=0.9)
    model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])
    return model
```

Found 3362 images belonging to 2 classes.
Found 1138 images belonging to 2 classes.
> 99.736

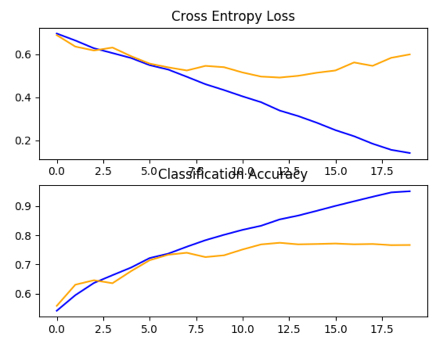


**Fig no.: 5.14 Two Block VGG model**

```
def define_model():
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(200, 200, 3)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Dropout(0.2))
    model.add(Flatten())
    model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
    model.add(Dense(1, activation='sigmoid'))
    # compile model
    opt = SGD(lr=0.001, momentum=0.9)
    model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])
    return model
```

Found 3362 images belonging to 2 classes.
Found 1138 images belonging to 2 classes.
> 98.594

**Fig no.: 5.15 One Block VGG model with Dropout regularisation.**

## VI. CONCLUSION AND FUTURE WORK

Accident detection became very important topic in traffic management systems. Detection of accident will avoid future accidents and will help authorities to make road segment available for traffic again This work will reduce the number of deaths in accidents by reporting the incident to the nearest emergency unit at the earliest. This work will also be used to help police to find the crimes and hit-run cases. In our project, the exact location name will be detected and transmitted using Map. The criticality of the accident will be surely reduced using Intelligent Surveillance System. In Future, this work will be extended to indicate the driver about the possibilities of accident occurring by using machine learning and ensure the rate of speed of intimating the accident to nearby emergency unit will be increased along with victim's details.

**REFERENCES**


[1] Mahamuni P. D, R. P. Patil, H.S. ThakarDetection and Classification of a Moving Object in a Video Stream

[2] Asim R. Aldhaheri and Eran A. EdirisingheMoving object detection using background subtraction algorithm using Simulink

[3] Pawan MishraA Study on Video Surveillance System for Object Detection and Tracking

[4] Smart Surveillance: Applications, Technologies and Implications by Arun Hampapur, Lisa Brown, Jonathan Connell, Sharat Pankanti, Andrew Senior, Yingli Tian.

[5] Mritunjay Rai, Agha Asim Husain, Tanmoy Maity and Ravindra Kumar Yadav Advance Intelligent Video Surveillance System (AIVSS)

[6] Anthony C. Davie,Sergio Velastin A Progress Review of Intelligent CCTV Surveillance Systems(2005)

[7] Ms.Shamshad Shirageri ,Dr.G.R.Udupi, Prof.G.A.BidkarDesign and development of Optical flow based Moving Object Detection and Tracking (OMODT) System.