Traffic Sign Recognition System

A Project Report

Submitted in the partial fulfillment of the requirements for

the award of the degree of

BACHELOR OF TECHNOLOGY

In

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

By

Name	Id no
Talapaneni Manogna	180030491
Tejaswi Reddy Kamjula	180030537
Mareedu Meghana	180030547
Chaganti Sai Keerthana	180030769

Under the Esteemed Guidance of

Dr.Pragnyaban Mishra

Associate Professor

(DEEMED TO BE UNIVERSITY)

Koneru Lakshmaiah Education Foundation

(Deemed to be University estd., u/s 3 of UGC Act 1956)

Greenfields, Vaddeswaram, Guntur (Dist.), Andhra Pradesh - 522502

November 2021

KONERU LAKSHMAIAH EDUCATION FOUNDATION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

The Project Report entitled **"Traffic Sign Recognition System"** is a record of bonafide work of Talapaneni Manogna(180030491), Tejaswi Reddy Kamjula(180030537),Mareedu Meghana (180030547),Chaganti Sai Keerthana(180030769) submitted in partial fulfilment for the award of **Bachelor of Technology** in **Computer Science and Engineering** to the K L University during the academic year **2021-22**. The results embodied in this report have not been copied from any other departments or University or Institute.

Talapaneni Manogna	180030491
Tejaswi Reddy Kamjula	180030537
Mareedu Meghana	180030547
Chaganti Sai Keerthana	180030769

KONERU LAKSHMAIAH EDUCATION FOUNDATION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled entitled **"Traffic Sign Recognition System"** is a record of bonafide work of Talapaneni Manogna(180030491), Tejaswi Reddy Kamjula(180030537),Mareedu Meghana (180030547),Chaganti Sai Keerthana(180030769) submitted in partial fulfilment for the award of **Bachelor of Technology** in **Computer Science and Engineering** to the K L University is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute.

Signature of Head of the Department	**Signature of the Supervisor**
Mr. V. Hari Kiran	Dr.Pragnyaban Mishra
Head of the department	Associate Professor

Signature of External Examiner

ACKNOWLEDGMENT

The success in this project would not have been possible but for the timely help and guidance rendered by many people. Our wish to express my sincere thanks to all those who has assisted us in one way or the other for the completion of my project.

Our greatest appreciation to my guide **Dr.Pragnyaban Mishra**, Associate Professor, Department of Computer Science and Engineering which cannot be expressed in words for his tremendous support, encouragement and guidance for this project.

We express our gratitude to **Mr. V. Hari Kiran,** Head of the Department for Computer Science and Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project-based Lab.

We thank all the members of teaching and non-teaching staff members, and also who have assisted me directly or indirectly for successful completion of this project.

Finally, we sincerely thank our friends and classmates for their kind help and co- operation during our work.

ABSTRACT

These days,traffic signs are crucial in means of advancement of auto mobile engineering.There are many proposed systems for traffic sign detection. In this project we have mainly employed CNN- ConvolutionalNeuralNetwork algorithm to classify ,and in training and testing models ,so that this system is more accurate than the already existing works.The fundamental process in this project is to train the data set and predict the signs accurately .So that ,this helps in automation system of vehicles.We have used German Traffic Sign Detection Benchmark -dataset ,which we acquired from Kaggle for implementation in this project.The images mainly comes under forty three different classes . In comparison, the proposed approach, employing spatial and spectral features jointly, will achieve better classification efficiency compared to standard support vector machines (SVM) by demonstrating the enormous potential of the CNN-based approach for accurate traffic sign hyperspectral data classification

TABLE OF CONTENT

INTRODUCTION

Traffic sign recognition is a challenging real-world problem for autonomous driving systems. In theory, traffic signs are designed to be easily detectable and recognizable because they follow standard shapes and colors.There are many factors which have to be taken into account of because every traffic sign is placed outdoors and exposed to different conditons. Generally traffic sign recognition is divided into detection to locate the signs through bounding boxes, and classification to differentiate between each specific traffic class.The existing systems allows mostly upto ten types of classes of traffic signs,but to associate with the new technology coming around we have to enhance this particular feature. The only datasets labelled with text-based signs are the MASTIF ones but in our knowledge, not even the authors addressed these types of classes in their studies. Despite there are many methods for evaluation, there were no common datasets to use unless the German Traffic Sign Recognition Benchmark (GTSRB-2011) and the German Traffic Sign Detection Benchmark (GTSDB-2013) are introduced in later times. Nevertheless, these benchmarks do not include important and challenging signs based on text and with very different aspect ratios; do not take into account the intra variability of traffic sign classes to deal with the cases where symbols or inscriptions vary; methods will only succeed to identify traffic signs of one country (Germany); these two of the bench marks don’t specify the correct accuracy of recognition .So, our proposed system has forty three primary classes for classifications of images using CNN layers.There fore we have to take care of the challenging situations and each layer of our model have to be pre dominantly testes to the worst case scenarios from the german traffic sign data set.This proposed system have shown high accuracy predictions and high performance compared to the existing systems.Some of the important classes of the traffic signs are speed limits,taking turns, oneway , any building ahead, crossing zone, right turn ahead, any certain disturbances on the path and many more.So, employing of this proposed system for any automated vehicles will certainly be helpful. Having more layers to out model help to identify the disorted images and hence the over fitting problem will be minimized to most extent.

CONVOLUTIONAL NEURAL NETWORKS:

CNNs are regular versions of multilayer perceptron. Multilayer perceptron sometimes signifies that absolutely associated networks, that is, every vegetative cell in one layer is connected to any or all neurons within the next layer. The completely-connectedness of those networks arise a risk of overfitting in the models. Typical ways in which of regularization embrace adding some type of magnitude measure of weights to the loss perform. However, CNNs take a distinct approach towards regularization: they make the most of the ranked pattern in knowledge and assemble additional complicated patterns victimization smaller and easier patterns. Hence, on the scale of connectedness and quality, CNNs are a little bit lower comparitively. They are called shift invariant.

Machine Learning:

1. Supervised learning:
These algorithms are trained by giving examples. The rule can receive input as completely different knowledge sets from those data sets the machine are going to be trained
Algorithms: call Tree and Naïve mathematician

2. Unsupervised learning:
These algorithms won't have any historical knowledge. Here the system won't be given the proper answer
Algorithms: Association Rules, K-means cluster

3. Reinforcement learning:
This is primarily utilized in artificial intelligence and navigation wherever trial and error ways are used
Algorithms: Utility learning, Q learning.

4. Semi-supervised learning:
In this rule, the machine uses each tagged and unlabeled knowledge Algorithms: Graph primarily based ways, self-training.

5. Constant learning:
A machine learning rule ought to learn perpetually till most accuracy is reached.

6. Feature learning:
Extract the options even for disordered knowledge sets.

7. Quick processing:
A machine learning rule ought to operate at higher levels.

LITERATURE SURVEY

This section consists of various existing methodologies that are currently being implemented to segment, classify traffic signs which involves image processing techniques. We will be also going to discuss some of the recent CNN models that are used for classification. K-Means clustering algorithm is used traditionally to classify. A research is performed by Amit Joshi et al. [1] to categorize traffic sign using small scale CNN. For training purposes a small scale CNN takes lessthan 10 seconds for one epoch.This is applied to Advanced Driver Assistance System(ADAS) to give more efficiency . The achieved testing accuracy is 97.72%, training accuracy is 99.18% and validation accuracy is 99.62,which are performed on German-Traffic-Sign-Recognition-Benchmark dataset. Alternate to K-Means clustering method, a model is suggested by Pavly Salah Zaki et al. [2] that implements deep learning . The sign recognition is done with multi-object detection systems such as MultiBox Detector combined with various feature extractors such as Mobile Net v1 and Inception v 2, and also Tiny-YOLOv2,with these approaches even when the images are not clear considering real time situatuons ,it got an accuracy of 96% and when tested with GTSRB dataset it achieved an 99.8% accuracy. Many studies in the literature have focused on classification.Traffic sign images based on their classes. The motivation of this study is to find out on traffic sign detection with regionally based persistent neural network which is one of the intensive learning Methods for classifying multiple images at the same time and disorted images such as Overlapping or exclusion of some part of the image.Because of more auto mode vehicles; a difficult issue in auto mobile and autonomous image processing. Many researchers have studied this issue.In 2020 Sunitha et al. [3], suggested a way to divide and classify using deep learning and then Expectation Maximization Algorithm.Neural networks were used.They also introduces suppportvecotrmachine to their models which have shown significant results. Experiments results show that, by deep convolutional and neural networks greatly enhances the potential of detection and still retains its realtime performance.CNN is used for recognition method, SVM is then connected with the final outmost layer of CNN for further classification.But, some traffic signs are miss-recognized when this method is applied on the unmanned ground vehicle.Employing deep learning with also SVM is usually gives some better results in testing the data.

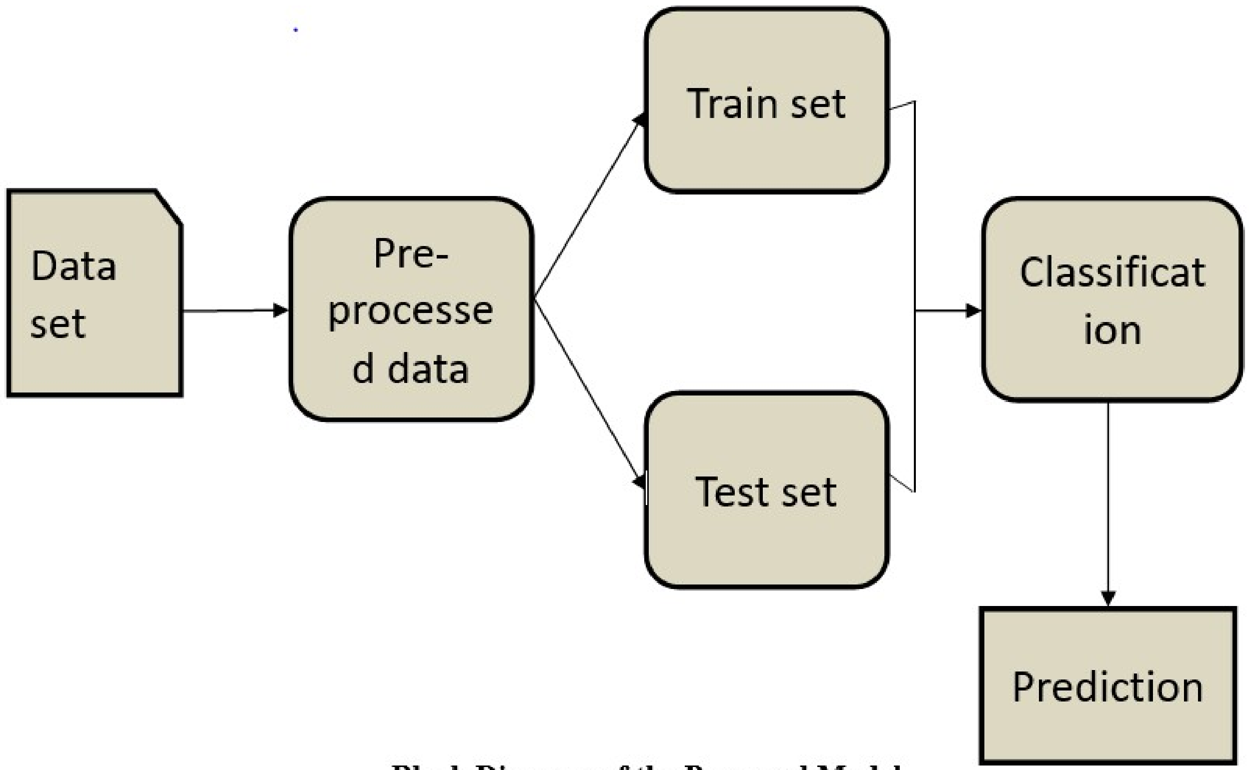
An improved VGG model was proposed by Shuren Zhou et al. [6], in a 2018 study. An inspired IVGG model is used in this paper,which includes 9 layers,compared with the original VGG model.This model helps in augumentations of the image with the real time. So this system is more robust and economical . In the German traffic sign dataset test, the recognition rate is up to 99.00%. (images 29) with precision Rates of 98%, 92%, 95% respectively.
A method was proposed by Zhongyu Wang and Hui Guo It has a significant growth in area of automated and selfassisted driving vehicles. It is of great significance for roadsafety. Regarding this we have used convolutional neuralnetworks (CNN) for traffic sign recognition and image classification. CNN has a high accuracy for object detection of tasks but the time for accurate and précised output CNN requires large amount of hardware. CNN along with YOLO,
darknet 53 Is the improvised version of the CNN model. It is much efficient than the regular CNN model. It has high accuracyand less hardware requirement for system, the sequence of which is as follows Preprocessing, core division, cell division, feature extraction, feature selection and Classification in their 2019 study.A survey was conducted by Ayoub Ellahyani et al. [] in 2020 for Intelligent Transportation Systems in detection of traffic signs Advanced driver assistance systems are developed to enhance the road safety of vehicles and for a better driving experience. These systems mostly consist of advanced assistance tools like in-vehicle navigation, sensors, traffic management and monitoring the surrounding environment. They keep assisting the driver regarding the upcoming hurdles and sign to avoid collisions between vehicles. For device that use ADAS it most relies on traffic sign recognition for realtime scenarios in outsides. In this paper the main goal is to identify the sign and label them accordingly to notify the driver or driving assistant about the upcoming signs, types with 95% success. These systems face a problem in the real-world environment like bad climate,
damaged traffic signs, and bad lighting features .In recent years there is a huge advancement in Image Processing due to the intense study in CNN which has a huge significance. Previous researchers had utilized various CNN architectures for their models. Recent CNN models are proven to be more efficient when they are utilized to improve the classification accuracy explicitly on works such as object segmentation and classification. So a method is employed that a certain image goes through all of the layers in the model. The architecture comprises of a pair of Convolutional layers, a pair of Pooling layers, and a set of fully connected layers. These layers help to improve the CNN architecture as their presence favors the overall efficiency. We also used activation functions in between the layers so that accuracy can be maximized for the given dataset. The input image is passed through the mentioned layers

THEORETICAL ANALYSIS

Convolution Neural Networks play a prominent role in areas of deep learning and machine learning.They help in evolution of modern neural networks. Convolutional Neural Networks are like the kind of similar to standard-Neural Networks.Each of the strand of the nuural networks have their own weight and show some bias based on the parameters, performs a scalar product and optionally follows it with a non-linearity. The final initiated network still expresses one tangible score-value function, from the naive imagepixels on one finish to category scores at the dissimilar, and that they still have a loss operations like SVM,softmax on the last completed layer and every small methods which are applied show some affect we tend to achieve for learning regular Neural Networks are applicable. Conv-Net architectures,kind of assume some parameters from the inputs given as preliminary ones which can be changed automatically. These,then build the forward operate additional convenient measures to implement and then immensely scale back the number of parameters within the network. The network gives higher classification results than the opposite classification methodology. CNN consists of many convolutional and pooling layers. Let's see however the network feels like. We are required to pass an input image to the primary convolutional layer. The output from convulated layer is obtained as an primary thread like a neuron. The filters which are applied within the convolution layer extract relevant options from the input image to even give more. Every filter tries to provide a totally different feature to help the accurate category prediction. Just in case we would like to retain the scale of the image, we tend to use same padding which is very less ,alternative wise valid artefact is employed since it helps to scale back the quantity of options. Pooling layers are then added to further scale back the quantity of parameters. Many convolution and pooling layers are added before the prediction is created.These convolutional layer facilitates in feature extracting options. As we tend to go much deep down with in the network additional specific options are extracted as compared to a narrow network wherever the features extracted are more standard. The output layer during a CNN may be a totally connected layer, wherever the input from the opposite layers is flattened and sent therefore because the re-model the output into the quantity of categories as desired by the network.Then we check the error or any other anomalies with the final layer which is the output layer. A loss operate is given within the totally connected output layer to figure the mean of loss. The gradient of error is then calculated. Then the bias values will be updated.. A CNN consists of an input and an output layer, also as multiple hidden layers. The hidden layers of a CNN generally accommodate convolutional layers, pooling layers, totally connected layers and standardisation layers.Every layer generates a certain type of output ,which may not be accurate all the time ,that output will be passed through the next layer as input or a secondary input.

CNN might embrace native or world pooling layers, that mix the outputs of nerve cell clusters at one layer into one neuron within the next layer. ReLU is that the abbreviation of corrected Linear Units. This layer applies the non-saturating activation operate. The performance of the projected work is compared victimisation Convolutional Neural Network (CNN) to investigate that of those can offer the simplest results. A convolutional neural network (CNN) may be a category of deep, feed-forward artificial neural networks that use a variation of multilayer perceptron designed to need lowest preprocessing. Here, CNN with 4 layers are employed. CNN performs its work accurately exploiting 10,000 images for each epoch. During this work, many pictures are used and accuracy and is found to be quite 98.2%. in line with the study, it's ended that if there's availableness of additional pictures, then CNN is found to supply the simplest results with high accuracy.

Proposed Model:

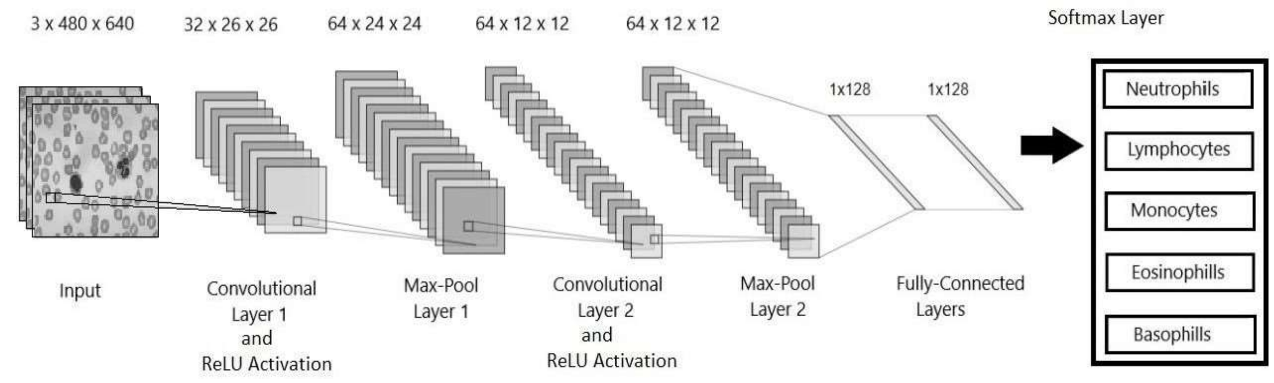


Block Diagram of the Proposed Model

The training and testing data are passed through pre processing stage. During pre-processing we transform color space. After pre-processing data is passed through CNN layers . The training data is used to train the models in the goal of improving accuracy. Testing data is used to perform an unbiased evaluation for final model on the training dataset.

CNN Architecture:

Convolutional Neural Networks is one of the extensively used deep learning networks like image classification. CNN is originated from artificial neural networks. When the hidden layers in artificial neural networks are increased, complexity of the neural network is increased. As the result better results are produced from the network.



Proposed Architecture

i) Convolutional Layer

Convolutional layer is widely used to classify images as it utilizes adjoining pixel information to efficiently sample the image and then work on predictions achieving high accuracy. A filter (feature vector) audits each and every area of input image in search of a line. Every unit is checked as the filter moves one unit to the right until input image end is reached. An array known as feature map records every location and stores them in it.The value of locations will be given on the basis of the number of lines present in them .The more line ,the more value will be taken into consideration.

ii) Max pooling

When the input image is too complex,we need to have more computation power. As a result, computational cost will be significantly higher to implement the method. Maximum pooling layer reduces the incoming input layer resolution. As resolution for input images will be lower, efficiency will be gradually increased in computing the images.

iii) Fully Connected Layers

Located at the end of the neural network, these layers are connected with all the activation location in all the layers preceding it. These layers compile the information extracted from previous layers to generate the final output. Fully connected layers classify the information into various classes after feature extraction.It takes more time than the other layers but significantly less time than the convilational layers.

iv) Rectified Linear Unit (ReLU) Layer

ReLU is the most extensively used activation function in artificial neural networks, specifically in CNNs. ReLU is true for all positive values, and zero for all non-positive values.Here we wont use muh complex math for the methods,so it is easy to implement and takes less hours.So,here it helps in increase the whole project's accountability towards time. This activation function is implemented along with convolution layer in the architecture.

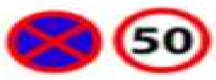
v) Softmax Layer

The soft-max function is commonly applicable on the last layer of CNN which is the final one. The status of a particular class data is shown and a value is generated about which class is closer to it. The probability value of each class is extracted by performing the calculation of probability in network. For these procedures cross entropy is used.

Experimental Investigations

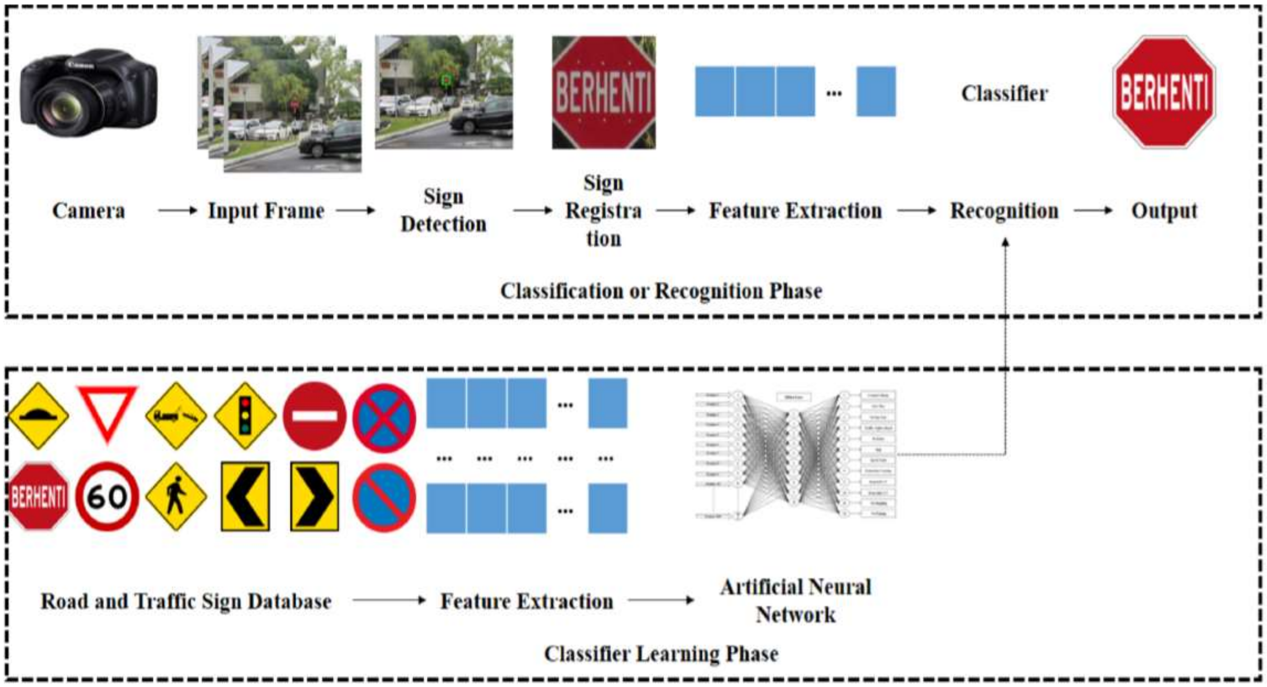
Dataset:

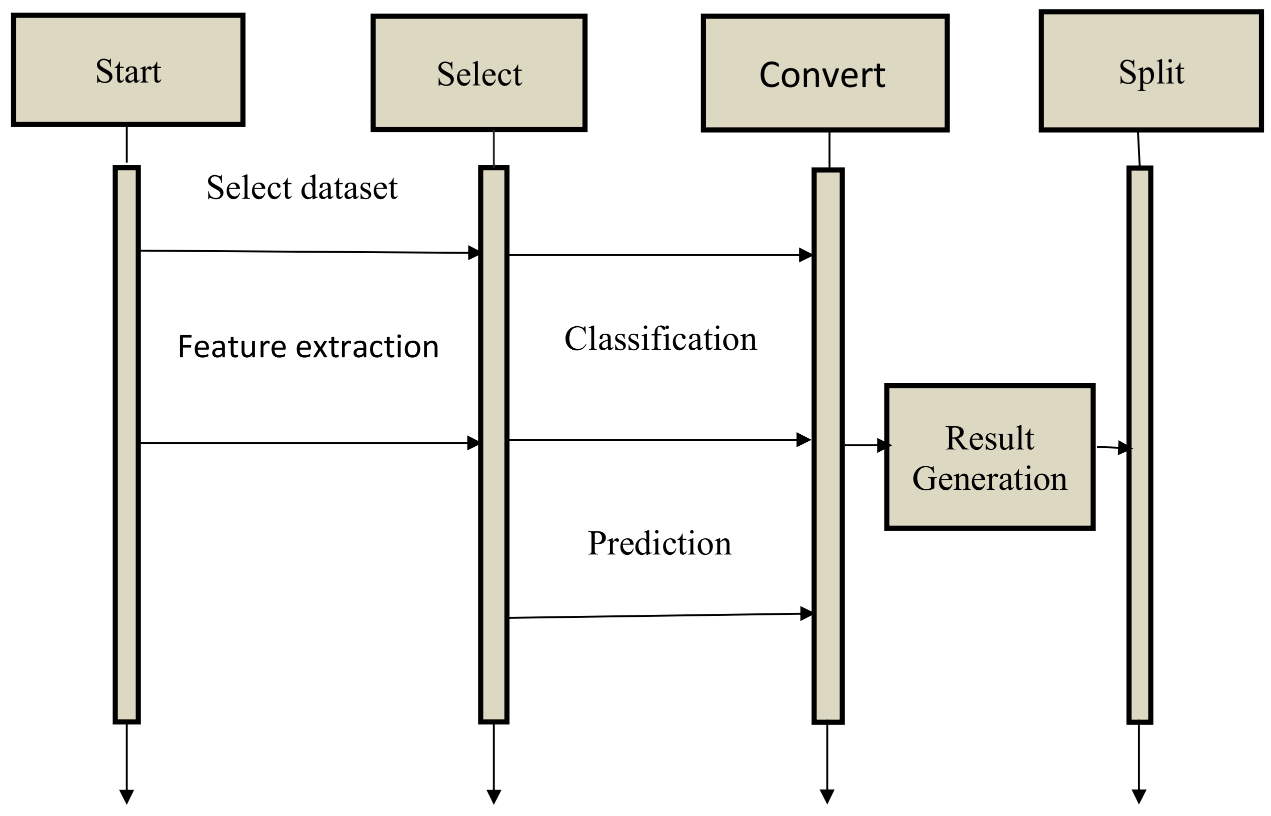
The dataset used in this paper are taken from Kaggle[14] dataset. We have taken real time road sign smear images because our main target area is Classification of Traffic signs. A total of 500000 images of data are examined. These data are traffic sign images which is classified into 43 different classes.

Category	Shape	Color	Example
Prohibitory	Circular	Red, Blue, White & Black	
Mandatory	Rectangular & Circular	Blue, White & Black	
Danger	Triangular	Red, White & Black	

Segmentation:

Segmentation subdivides an image into constituent regions or objects which means partitioning the original image into multiple segments or different regions. The goal of segmentation is to extract more features and information from the image and analyze it in a more meaningful and easier manner. It takes the Pre-processed digital image as an input and gives the attributes of an image as output which is used for feature extraction and object recognition.We have to assign each value for each pixel.At the end an observation can be made upon that these distinguished pixels share some common characteristics. There are many approaches for image segmentation, but we have used the Edge-based segmentation method which is more suitable for white blood cell detection.Data can be extracted by segmenting the images ,which is a feasible process.An image partition have to made in the basis of its pixels only. The main purpose of interest in an image which helps in an image that helps in the annotation of the object scene. There are three main approaches of image segmentation which are region-based approach, boundary approach, and edge approach that depends on two properties discontinuity and similarity. Edge- based segmentation also known as edge detection is a process of locating the edges of an image. Edges usually correspond to points in the image where the grey value changes from one pixel to the other. discovered that the most important information lies in the edges of an image. Canny Edge Detection is used to get optimal results.





Feature Extraction:

In this paper, the Harris Corner Detection algorithm is used as the feature extraction technique. This technique is used to extract the important features from the image. It was first published in 1988 by Chris Harris and Mike Stephens. It is an operator for corner detection which is widely used in machine learning algorithms to identify corners. This algorithm can detect the edges, flat area, and corners present in the image which helps in identifying whether the cell is White blood cell or not. A corner is a point that can be viewed as a two-edge intersection, where an edge reflects a sudden change in the brightness of the image. Corners are where a slight location shift will lead to a significant change in intensity both vertical and horizontal axes.

Packages:

For the implementation of the project we have used a number of packages ,some of them are given .

Keras:
Keras is a kind of neural-networks library.It is very user friendly.Day by day ,many contributions are being made to keras.It means many people are helping to improve the Keras library.There is a chance of raise of concerns while using keras like version control,but it is quite easy to handle them.Also,keras can run on many reliable platforms which is a lot of help of many deep learning

models.The tensorflow is considered as backend for keras,like keras helps in implementing tensorflow in certain situations.It runs above of tensorflow.Keras is implemented in python language only.A large number of neural networks are implemented with the help of keras. A keras model is employed with many components which are an architecture, a set of bias values termed as weighs,an optimization model, a set of negative parameters and metrics .

Tensorflow:

TensorFlow is an open source platform for data science . It has vast number of flexible ecosystem of tools, libraries and community resources that helps people to push the most efficient researchs in ML and developers can build and power machine learing applications.Deep neural networks are given most priority in TensorFlow.The core of tensorflow is not developed in Python but it also employs cpp and nivida's CUDA.Large scale numerical computations can are much feasible in TensorFlow. Apart from high level application interfaces it also provides many low level API's too.

OpenCV:

OpenCV is termed as computervision is an opensource machine learning software library. It is employed for the purpose of processing of any media like images or videos.It can even process the real hand writing of people .It is highly reliable .This package is very safe to use because there is a chance of happening of security breaches when is employ this type of packages in out project .We can observe that there are almost two thousand and five hundred algoritms are included in an optimized state.OpenCV can also used to find the similarities between two media objects.In this certain library,the inputs which we give will be converted into numpy arrays.
We can also use 3-D inputs.Many start ups also use Open CV because it is economically feasible and reliable for large projects too.

Pandas:

Pandas is one of the widely used data science package.It is in python only,offers us a wide range of models or methods to convert our dataframe to our requirements. We can work with labeled data or data which can be relatable on a certain basis.We can also combine number of data sets using Pandas. It is a python package which is available on Conda.We need to have cython to install pandas in our developer tool.Labeling of hierarchal data is also possible in pandas.To explore data science using pandas is always a starting step,but pandas offers much more

Matplotlib:

Matplotlib is an open source plotting library in python which acts like an extension of numpy.It can also implement Matlab functions.There are vast numbers of methods in it.It contains many exhaustive and complex plot options that help us to easily understand and analyse our data.

tKinter:

It is used as interface .It is one of the standard python user-interfaces which are feasible to work with for any type of projects. The name comes from tk interface.tkinter is emplpyed in desktop based applications mostly.

Anaconda:

Anaconda is the best platform for data science .It has many applications in it which are easy to process when one wants to use two or more tools for the projects.There are many number of packages,apart from them if you want to install other than those in anaconda package manager you can install them too easily.It is a data science Technology.It is suitable for the today’s fastly evolving and changing data-science driven world.A lot of useful and functioning tools are available.

IMPLEMENTATION

main.py

```
import os

import pandas as pd

import numpy as np


import matplotlib.pyplot as plt

from matplotlib.image import imread

import seaborn as sns

dir_path = r'Dataset'

os.listdir(dir_path)


train_path = dir_path +'\Train'

test_path = dir_path + '\Test'

print(sorted(os.listdir(train_path)))

sorted(os.listdir(test_path))

dim1 = []
dim2 = []

for i in range(0,10):
    labels = train_path + '/{0}'.format(i)

    image_path = os.listdir(labels)
    for x in image_path:

        img = imread(labels + '/' + x)
        dim1.append(img.shape[0])
        dim2.append(img.shape[1])

sns.jointplot(dim1,dim2)


plt.show()
#the first plot
```

```
import random

images_path = os.listdir(test_path)

plt.figure(figsize=(25,25))


for i in range(1,26):

    plt.subplot(5,5,i-1)

    random_i_path = testpath +'/'+ random.choice(images__path)

    rand_image = imread(random_i__path)

    plt.imshow(rand_img)

    plt.xlabel(rand_img.shape[1], fontsize = 20)#width of image

    plt.ylabel(rand_img.shape[0], fontsize = 20)#height of image


np.mean(dim1)

np.mean(dim2)

image_shape = (50,50)

#we can re shape the image here



#Data Preprocessing
#Importing the images

from PIL import Image

images = []
label_id = []

for i in range(10):

    labels = train_path + '/{0}'.format(i)
    image_path = os.listdir(labels)

    for x in image_path:

        img = Image.open(labels + '/' + x)
        img = img.resize(image_shape)
        img = np.array(img)

        images.append(img)

        label_id.append(i)



images = np.array(images)
images = images/255

label_id = np.array(label_id)

label_id.shape
images.shape


plt.figure(figsize=(15,5))
sns.countplot(label_num)


plt.title('classes of distribution for images', fontsize = 15)
plt.xlabel('Label_num', fontsize=12)

plt.show()


np.save('Trained Set', images)
np.save('Label_num', label_num)


import numpy as np

import pandas as pd

images = np.load('Training_set.npy')

label_id = np.load('Label_Id.npy')

#Splitting the data
from sklearn.model_selection import  train_test_split

x_train, x_val, y_train, y_val = train_test_split(images, label_id , test_size = 0.2, random_state = 42)


#keras has a built-in function for one-hot encoding.


from tensorflow.keras.utils import to_categorical
```

```
y_train_cat = to_categorical(y_train)
```

```
y_val_cat = to_categorical(y_val)
```

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Dropout, Conv2D, MaxPool2D
model = Sequential()
```

```
#1st layer
model.add(Conv2D(filters = 64, kernel_size = (4,4), input_shape = x_train.shape[1:], activation = 'relu',
padding = 'same'))

model.add(MaxPool2D(pool_size=(2,2)))

model.add(Dropout(0.5))

#2nd layer
model.add(Conv2D(filters = 64, kernel_size = (4,4), activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2)))

model.add(Dropout(0.5))

#3rd layer
model.add(Conv2D(filters = 64, kernel_size = (4,4), activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2)))

model.add(Dropout(0.5))

model.add(Flatten())

#Dense layer
model.add(Dense(128, activation = 'relu'))
```

```
    model.add(Dropout(0.5))

    #Output layer

    model.add(Dense(43, activation = 'softmax'))


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



    early_stopping = tf.keras.callbacks.EarlyStopping(monitor = 'val_loss', patience = 2)



model.fit(

   x_train, y_train,
   epochs = 10,
   batch_size = 32,
   validation_data = (x_val, y_val),
   callbacks = [early_stopping],
   verbose = 1

)
model.save(r'traffic_recognition.h5')



cnn=(model.evaluate(x_val,y_val,verbose=1)[1])*100


print('Convolutional Neural Network Accuracy is:',cnn,'%')



history=model.history.history



#Plotting the accuracy
train_loss = history['loss']
val_loss = history['val_loss']
train_acc = history['accuracy']
val_acc = history['val_accuracy']


history=model.history.history
#Plotting the accuracy
```

```
train_loss = history['loss']
val_loss = history['val_loss']
train_acc = history['accuracy']
val_acc = history['val_accuracy']



# Loss
plt.figure()
plt.plot(train_loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.title('Loss')
plt.legend()
plt.show()



# Accuracy
plt.figure()
plt.plot(train_acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.title('Accuracy')
plt.legend()
plt.show()




import tkinter as tk

import tensorflow as tf

from tkinter import filedialog

from tkinter import *

from PIL import ImageTk, Image
import numpy

#load the trained model to classify sign

import keras

from keras.models import load_model


model = tf.keras.models.load_model(r"C:\Users\manogna\Desktop\source code\traffic_recognition.h5")
```

```
classes = { 1:'Speed limit (20km/h)',
            2:'Speed limit (30km/h)',
            3:'Speed limit (50km/h)',
            4:'Speed limit (60km/h)',
            5:'Speed limit (70km/h)',
            6:'Speed limit (80km/h)',
            7:'End of speed limit (80km/h)',
            8:'Speed limit (100km/h)',
            9:'Speed limit (120km/h)',
            10:'No passing',
            11:'No passing veh over 3.5 tons',
            12:'Right-of-way at intersection',
            13:'Priority road',
            14:'Yield',
            15:'Stop',
            16:'No vehicles',
            17:'Veh > 3.5 tons prohibited',
            18:'No entry',
            19:'General caution',

            20:'Dangerous curve left',
            21:'Dangerous curve right',
            22:'Double curve',
            23:'Bumpy road',
            24:'Slippery road',
            25:'Road narrows on the right',
            26:'Road work',
            27:'Traffic signals',

            28:'Pedestrians',

            29:'Children crossing',

            30:'Bicycles crossing',

            31:'Beware of ice/snow',

            32:'Wild animals crossing',

            33:'End speed + passing limits',

            34:'Turn right ahead',

            35:'Turn left ahead',

            36:'Ahead only',

            37:'Go straight or right',

            38:'Go straight or left',

            39:'Keep right',

            40:'Keep left',

            41:'Roundabout mandatory',
            42:'End of no passing',

            43:'End no passing veh > 3.5 tons'
}
```

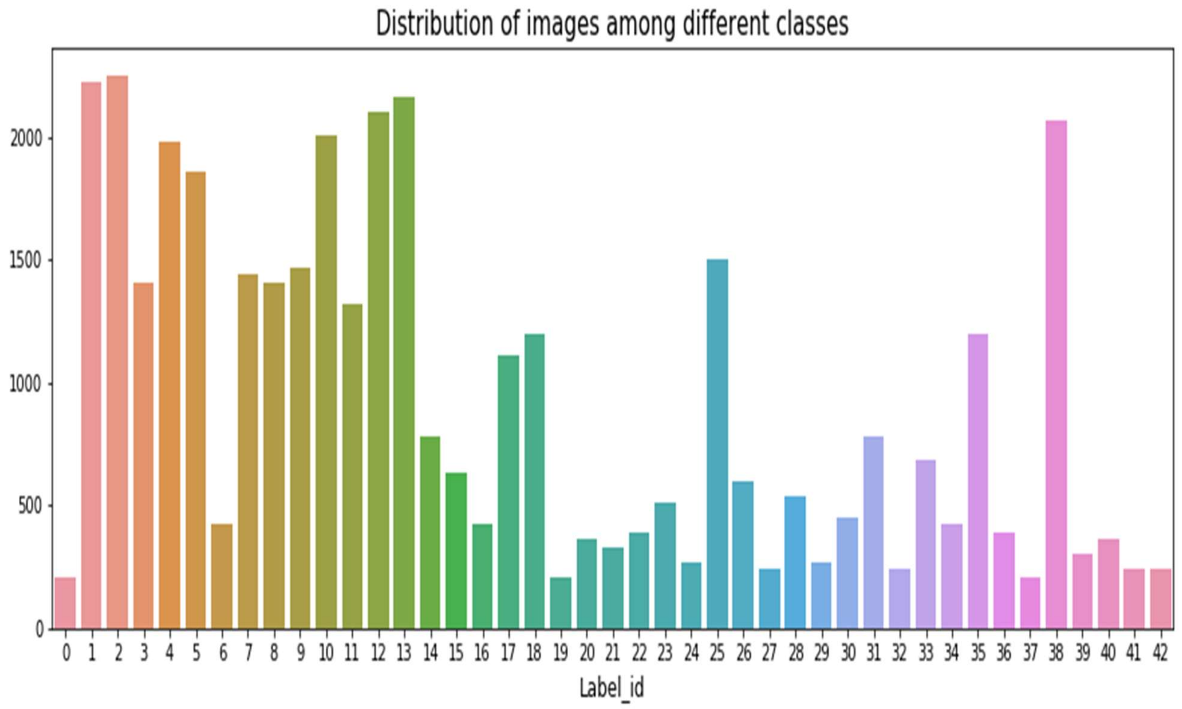
```
top=tk.Tk()
top.geometry('800x600')
top.title('Traffic sign classification')


top.configure(background='#CDCDCD')
label=Label(top,background='#CDCDCD', font=('arial',15,'bold'))


sign_image = Label(top)


def classify(file_path):

    global label_packed

    image = Image.open(file_path)
    image = image.resize(((50,50)))

    image = numpy.expand_dims(image, axis=0)
    image = numpy.array(image)

    pred = model.predict_classes([image])[1]

    sign = classes[pred]

    print(sign)

    label.configure(foreground='#011638', text=sign)
```

```
def show_classify_button(file_path):

    classify_b=Button(top,text="Classify Image",command=lambda: classify(file_path),padx=10,pady=5
    classify_b.configure(background='#364156', foreground='white',font=('arial',10,'bold'))
    classify_b.place(relx=0.79,rely=0.46)

def upload_image():
    try:
        file_path=filedialog.askopenfilename()
        uploaded=Image.open(file_path)
        uploaded.thumbnail(((top.winfo_width()/2.25),(top.winfo_height()/2.25)))
        im=ImageTk.PhotoImage(uploaded)
        sign_image.configure(image=im)
        sign_image.image=im

        label.configure(text='')
        show_classify_button(file_path)
    except:
        pass

upload=Button(top,text="Upload an image",command=upload_image,padx=10,pady=5)
upload.configure(background='#364156', foreground='white',font=('arial',10,'bold'))


upload.pack(side=BOTTOM,pady=50)


sign_image.pack(side=BOTTOM,expand=True)
```

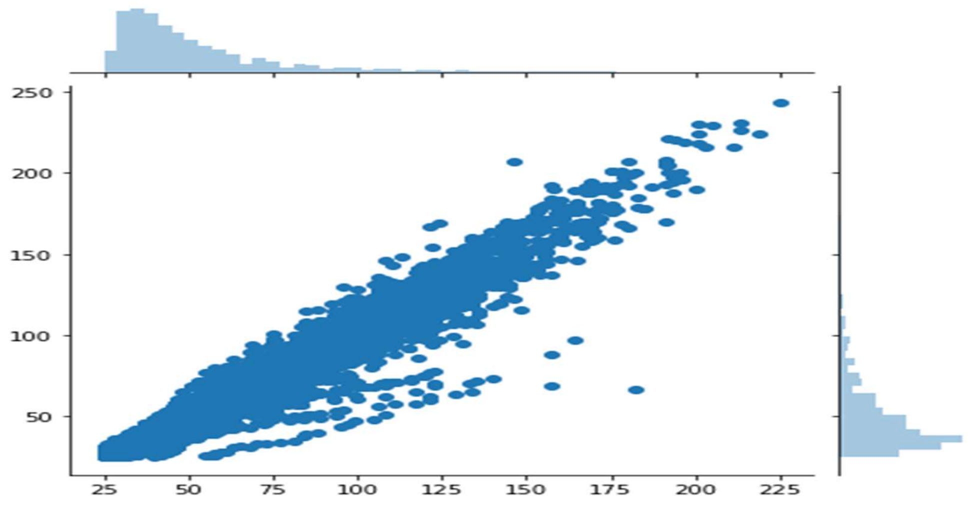
```
label.pack(side=BOTTOM,expand=True)

heading = Label(top, text="Know Your Traffic Sign",pady=20, font=('arial',20,'bold'))

heading.configure(background='#CDCDCD',foreground='#364156')

heading.pack()

top.mainloop()
```

EXPERIMENTAL RESULTS

After the execution our model succeeded in classifying the traffic data set present in each category.



The proposed method is implemented on a system with 8 processors of Intel Core i7-8550U CPU with 2.0 GHz and 8 GB RAM. The CNN model was developed using “Keras” [15] and “Tensorflow” [16] libraries in python. We used “Anaconda Distribution” as platform to implement our model. The other libraries include Pandas, Matplotlib, Scikit Learn [17], and OpenCV [18].



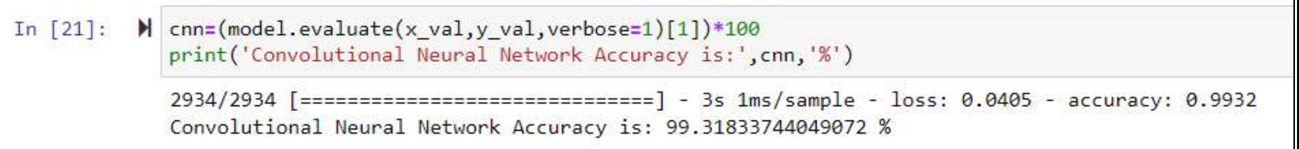
```
    plt.xlabel(rand_img.shape[1], fontsize = 20)#width of image
    plt.ylabel(rand_img.shape[0], fontsize = 20)#height of image
```

Accuracy is defined as the ratio of sample images that sample images.

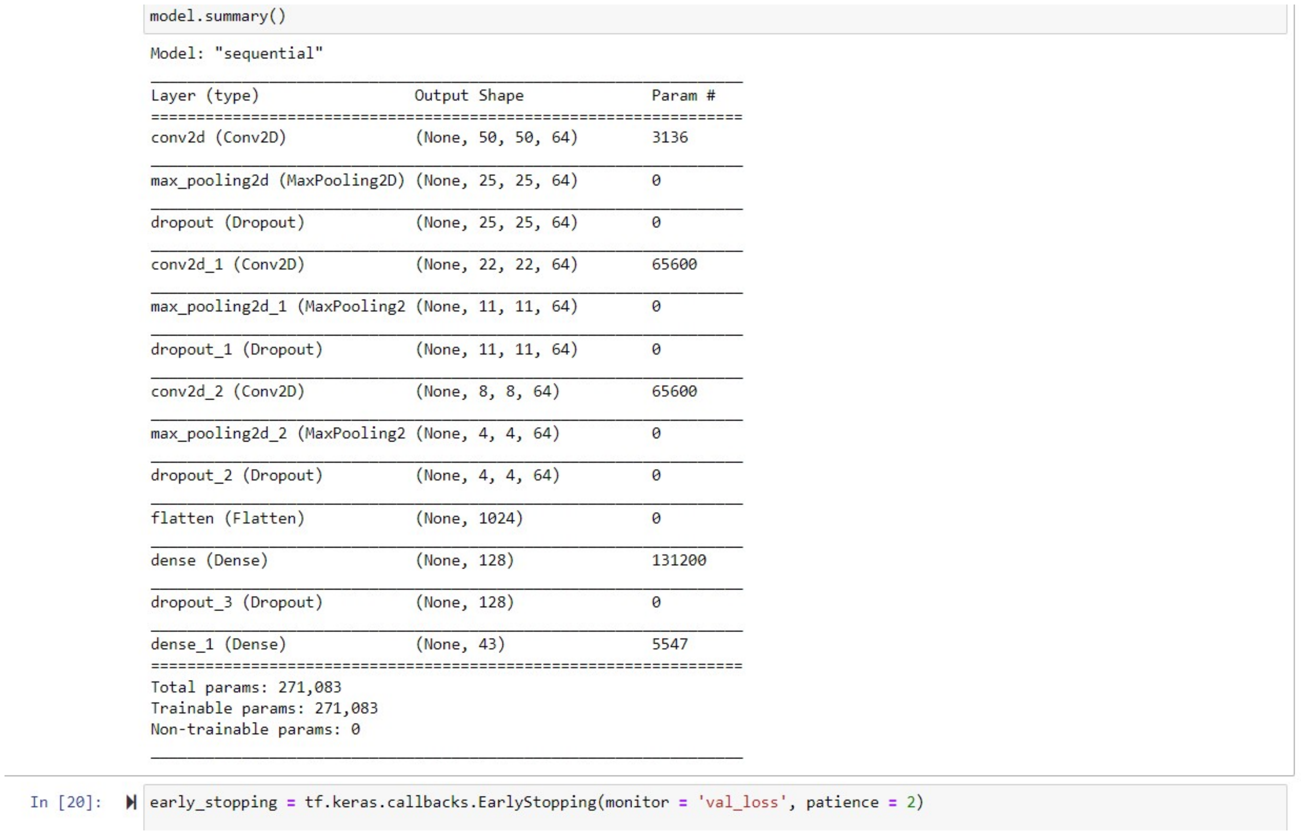
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

We achieved an accuracy of 99.31% through our proposed model. We implemented different combinations of Activation functions, Optimizers, and Loss functions to compare the achieved accuracy with other models.

The activation map was displayed in the below output.



```
In [21]:  cnn=(model.evaluate(x_val,y_val,verbose=1)[1])*100
          print('Convolutional Neural Network Accuracy is:',cnn,'%')

          2934/2934 [==============================] - 3s 1ms/sample - loss: 0.0405 - accuracy: 0.9932
          Convolutional Neural Network Accuracy is: 99.31833744049072 %
```

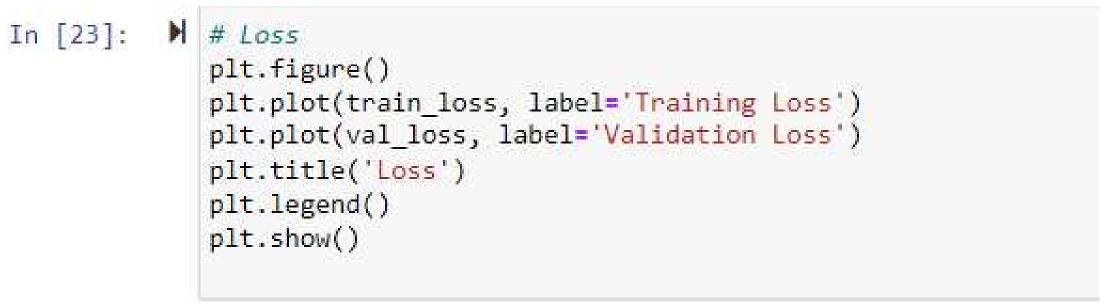


```
model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 50, 50, 64)        3136
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 25, 25, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 25, 25, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 22, 22, 64)        65600
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 11, 11, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 11, 11, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 8, 8, 64)          65600
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 64)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 4, 4, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 1024)              0
_________________________________________________________________
dense (Dense)                (None, 128)               131200
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 43)                5547
=================================================================
Total params: 271,083
Trainable params: 271,083
Non-trainable params: 0
_________________________________________________________________

In [20]:  early_stopping = tf.keras.callbacks.EarlyStopping(monitor = 'val_loss', patience = 2)
```

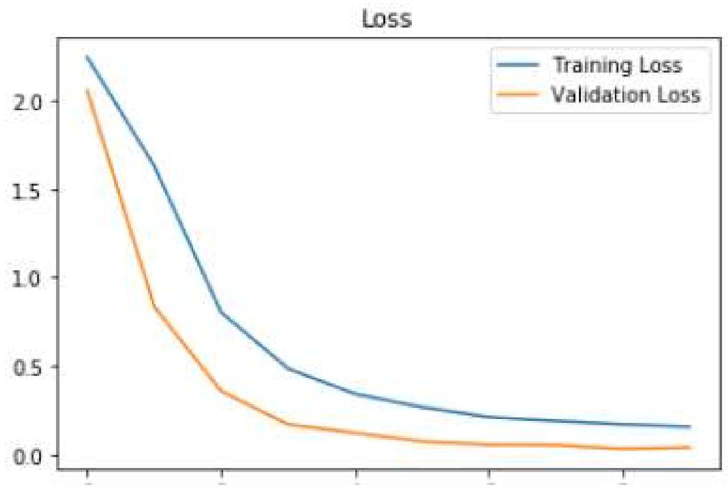
DISCUSSION OF RESULTS

We have done the classification on “German Traffic sign dataset ” which was collected from Kaggle. All the models are trained with that dataset. The model trained is then utilized to test the data on other datasets.

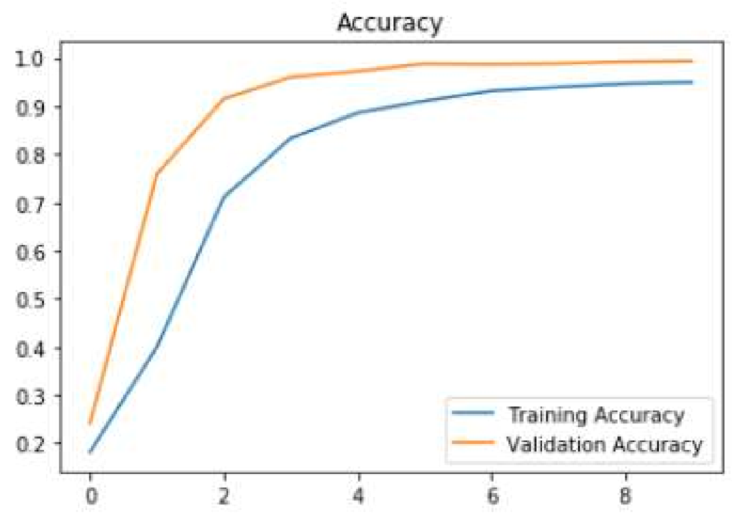
Our proposed model has achieved an accuracy of 99.31%. In our model we used “Softmax” function as Activation function, Optimizer, and “Categorical_Crossentropy” as Loss function. We implemented various combinations of Activation functions, Optimizers, and Loss functions to compare the proposed model with other models

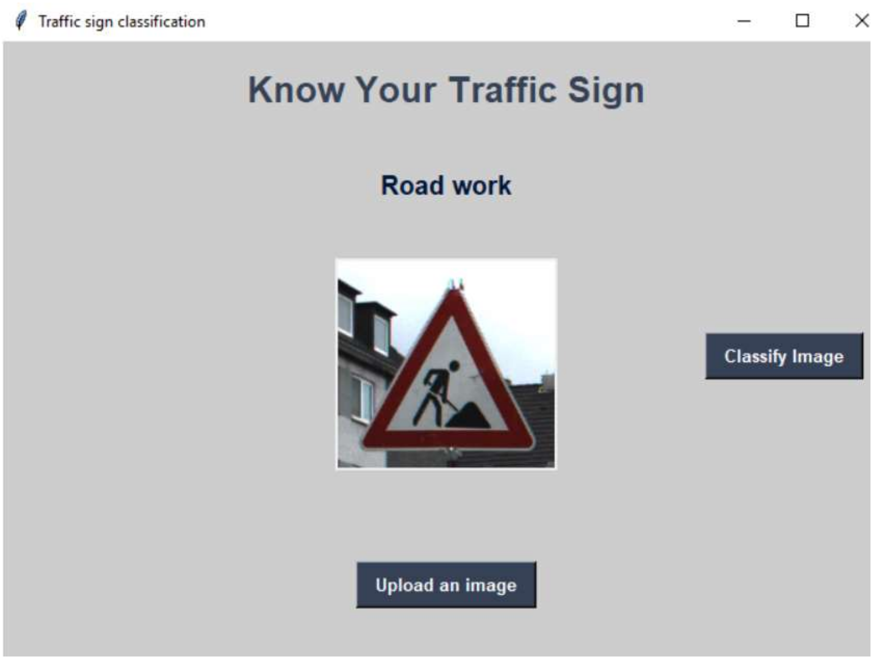


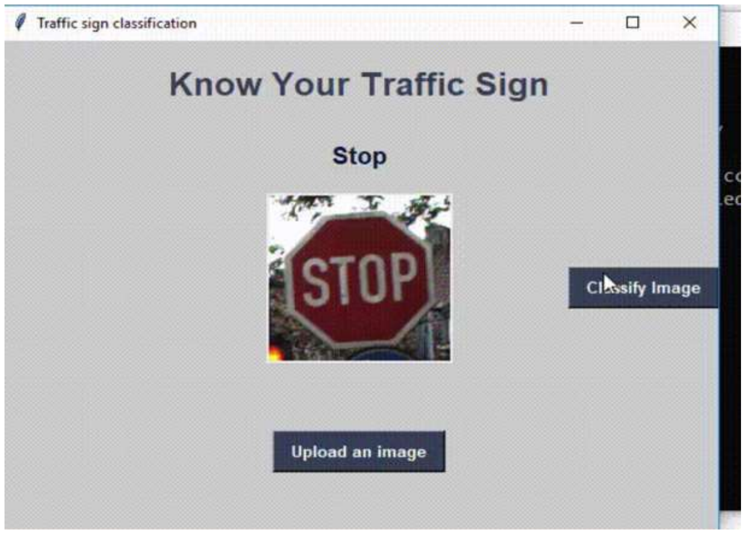
```
In [23]:  # Loss
          plt.figure()
          plt.plot(train_loss, label='Training Loss')
          plt.plot(val_loss, label='Validation Loss')
          plt.title('Loss')
          plt.legend()
          plt.show()
```

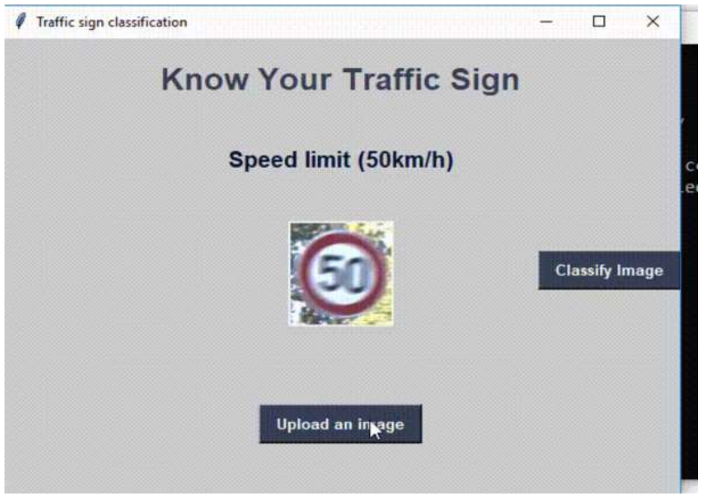


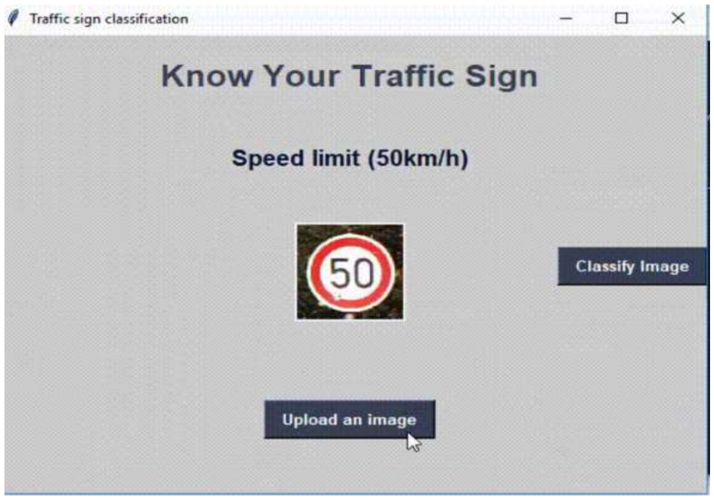
```
In [24]:  # Accuracy
          plt.figure()
          plt.plot(train_acc, label='Training Accuracy')
          plt.plot(val_acc, label='Validation Accuracy')
          plt.title('Accuracy')
          plt.legend()
          plt.show()
```






Traffic sign classification
Know Your Traffic Sign
Stop
Classify Image
Upload an image


Traffic sign classification
Know Your Traffic Sign
Speed limit (50km/h)
Classify Image
Upload an image


Traffic sign classification
Know Your Traffic Sign
Speed limit (50km/h)
Classify Image
Upload an image

CONCLUSION

In this project,we had make use of convolutional neural networks to implement traffic sign recognition system.Also we have taken the dataset from Kaggle which is benchmarks dataset for all the traffic sign related projects and trained that data with our models.The results which we got have shown the accuracy of 99.31%in the cases.The outcomes of our second module are promising,because in most cases when we have submmting the image they have shown the accurate class names of traffic sign.When in comparison with other models of out model in same the same area,this has shown better state in terms of accuracy and performance . This method holds a capability to be applied to a vast number of applications because of its adaptableness and simplicity.

REFERENCES

[1] Tejas Chaudhari1 , Ashish Wale, Amit Joshi, Suraj Sawant, " Traffic Sign Recognition Using Small-Scale Convolutional Neural Network," in International Conference on Communication and Information Processing (ICCIP-2020) Available on: SSRN .

[2] Pavly Salah Zaki, Marco Magdy William, Bolis Karam Soliman, Kerolos Gamal Alexsan " Traffic Signs Detection and Recognition System using Deep Learning," in the Nect Generation Information Technology Summit (Confluence), 2018.

[3] Sunitha.A , Shanthi.S "TRAFFIC SIGN CLASSIFICATION AND DETECTION USING DEEP LEARNING." IJIRTEXPLORE 2021.

[4] S. . Zhou, W. . Liang, J. . Li and J. . Kim, "Improved vgg model for road traffic sign recognition," Computers, Materials & Continua, vol. 57, no.1, pp. 11–24, 2018.

[5]. C. G. Serna and Y. Ruichek, "Classification of traffic signs: The European dataset," IEEE Access, vol. 6, pp. 78136–78148, 2018.

[6] H. S. Lee and K. Kim, "Simultaneous traffic sign detection and boundary estimation using convolutional neural network," IEEE Trans. Intell. Transp. Syst., vol. 19, no. 5, pp. 1652–1663, May 2018.

[7]"GTSRBCompetition,"[Online]Available: http://benchmark.ini.rub.de/?section=gtsrb&subsection=news.

[8] Wu, Yiqiang, et al. "Real-time traffic sign detection and classification towards real traffic scene." Multimedia Tools and Applications (2020): pp.1-20.

[9] Greenhalgh, J.; Mirmehdi, M. Real-time detection and recognition of road traffic signs. IEEE Trans. Intell. Transp. Syst. 2012, 13, 1498–1506. [CrossRef].

[10] Bechtel, Michael G., et al. "Deeppicar: A low-cost deep neural network-based autonomous car." 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). IEEE, 2018..

[11] A. Shustanov and P. Yakimov, "CNN Design for Real-Time Traffic Sign Recognition", 3rd International Conference "Information Technology and Nanotechnology, 25-27 April 2017.

[12] Rezki Wulandari Arief, Ingrid Nurtanio, Faizal Arya Samman, "Traffic Signs Detection and Recognition System Using the YOLOv4 Algorithm", Artificial Intelligence and Mechatronics Systems (AIMS) 2021 International Conference on, pp. 1-6, 2021..

[13] "keras api documentation", [online] Available: https://keras.io/..

[14] Kaggle, https://www.kaggle.com/datasets, accessed on 21st August, 2021.

[15] Chollet, Francois. "Building powerful image classification models using very little data." *Keras Blog* (2016).

[16] Ertam, Fatih, and Galip Aydın. "Data classification with deep learning using Tensorflow." In *2017* International Conference on Computer Science and Engineering (UBMK), pp. 755-758. IEEE, 2017.

[17] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *the Journal of machine Learning research* 12 (2011): 2825-2830.

[18] Bradski, Gary, and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[19] Liu, Weiyang, et al. "Large-margin softmax loss for convolutional neural networks." *ICML*. Vol. 2. No. 3. 2016.

[20] Bengio, Yoshua. "Rmsprop and equilibrated adaptive learning rates for nonconvex optimization." *corr abs/1502.04390* (2015).

[21] Zhang, Zhilu, and Mert Sabuncu. "Generalized cross entropy loss for training deep neural networks with noisy labels." *Advances in neural information processing systems*. 2018.

DrillBit
Anti Plagiarism Software

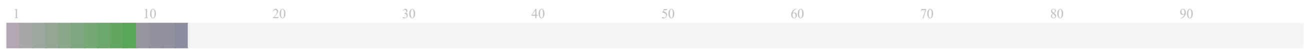
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

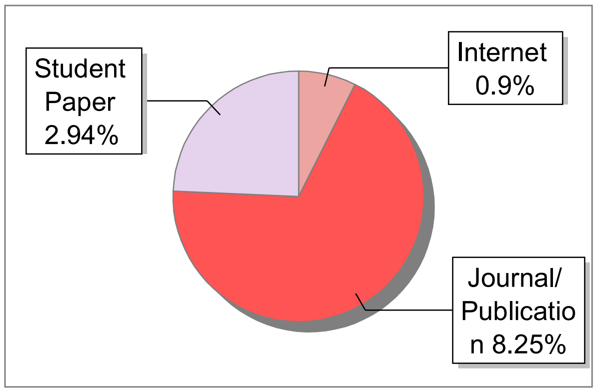
Author Name	Pragnyaban Mishra
Title	M.Tech Report
Paper/Submission ID	413786
Submission Date	2021 11-22 16:09:33
Total Pages	42
Document type	Dissertation

Result Information

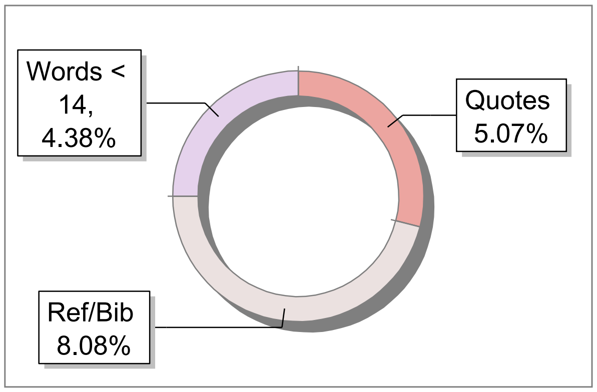
Similarity **14 %**



Sources Type



Report Content



Alert..! Improper usage of 'Quotes' work in a report

Exclude Information

References/Bibliography	Not Excluded
Quotes	Not Excluded
Sources: Less than 14 Words Similarity	Not Excluded
Excluded Source	**0 %**

A Unique QR Code use to View/Download/Share Pdf File

DrillBit
Anti Plagiarism Software

DrillBit Similarity Report

14	30	B	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	AN INTELLIGENT SYSTEM FOR SURVIVAL PREDICTION OF PATIENTS BY BHAVANA YR-2020	3	Student Paper
2	Traffic Signs Detection and Classification for European Urban Environm by Serna-2019	2	Publication
3	Thesis submitted to shodhganga - shodhganga.inflibnet.ac.in	1	Publication
4	www.researchgate.net	1	Publication
5	Thesis submitted to shodhganga - shodhganga.inflibnet.ac.in	1	Publication
6	fr.scribd.com	1	Internet Data
7	Thread-based Probabilistic Models for Expert Finding in Enterprise Microblogs by Xu-2015	1	Publication
8	A Weighted Combining Algorithm for Spatial Multiplexing by Kangli Zhang - ieeeexplore.org	<1	Publication
9	The Impact of Governance on Foreign Direct Investment Performance in Amhara Regi by Daddi-2013	<1	Publication
10	www.ncbi.nlm.nih.gov	<1	Internet Data
11	www.jipbs.com	<1	Publication
12	One Possible Model of Secure em-Government System by Markovi-2010	<1	Publication

13	Assessment of glycemic status and expenditure incurred in diabetic subjects in ur By Rekha, S 2012- RGUHS	<1	Publication
14	CLOCK DRIVEN MULTI BIT FLIPFLOP BASED ON PROBABILITY BY 15S51D5702 - 2019, JNTUH	<1	Student Paper
15	www.researchgate.net	<1	Internet Data
16	Corrigendum to Self-consistent modelling of Mercurys exosphere b by P- 2010	<1	Publication
17	IEEE 2019 International Conference on Economic Management and Model	<1	Publication
18	www.arxiv.org	<1	Publication
19	Data Mining Model for Money Laundering Detection in Financial Dom- www.ijcaonline.org	<1	Publication
20	Automated 3-D Lung Tumor Detection and Classification by an Active Contour Model by Kasinathan-2019	<1	Publication
21	www.futuremedicine.com	<1	Publication
22	www.epw.in	<1	Publication
23	www.doaj.org	<1	Publication
24	www.amity.edu	<1	Publication
25	Introduction to the Special Issue on Pioneers in Synchrotron Radiation by Williams-2015	<1	Publication
26	Geometry of Matrix Polynomial Spaces by Dmytryshyn-2019	<1	Publication
27	Formation of Azaspiracids-3, -4, -6, and -9 via Decarboxylation of Carboxyazaspi by McCarron-2009	<1	Publication

28	FARMERS PERCEPTION TOWARDS AGRICULTURE MARKETING COMMITTEES A A CAS BY KUMAR, VINEET 2010 -KRISHIKOSH	<1	Publication
29	Effects of spatial variability in the groundwater isotopic composition on hydrog by Kiewiet-2020	<1	Publication
30	A hybrid brain-computer interface for closed-loop position control of a robot ar by Rakshit-2020	<1	Publication