TRAFFIC SIGNS DETECTOR

Group 02

Chamath Amarsinghe E/16/022
Diwanga Amasith E/16/025
Wishva Madushanka E/16/222
Shamra Marzook E/16/232

PROBLEM STATEMENT AND MOTIVATION

- Traffic signs - convey, guide, restrict, warn, or instruct information
- Identification - important to ensures road safety
- Insufficient illumination, partial occlusion and serious deformation are the factors that cause.
- Traffic signal detection is a challenging problem.
  - driving systems
  - automatic driving systems

SOLUTION

- Train and classify Traffic Signs using Convolutional neural networks.
- This will be done using TensorFlow, Keras and OPENCV in real time using a simple webcam.
- Vision based solution
    - Can use smartphone cameras or webcams mounted on vehicles to detect traffic signals
    - Minimum cost of technical devices is an advantage

Dataset used

- German Traffic Sign Recognition Benchmark (GTSRB)
  public archive with ID daaeac0d7ce1152aea9b61d9f1e19370
- 43 classes of signals ~ 34,000 images

DATA PREPROCESSING

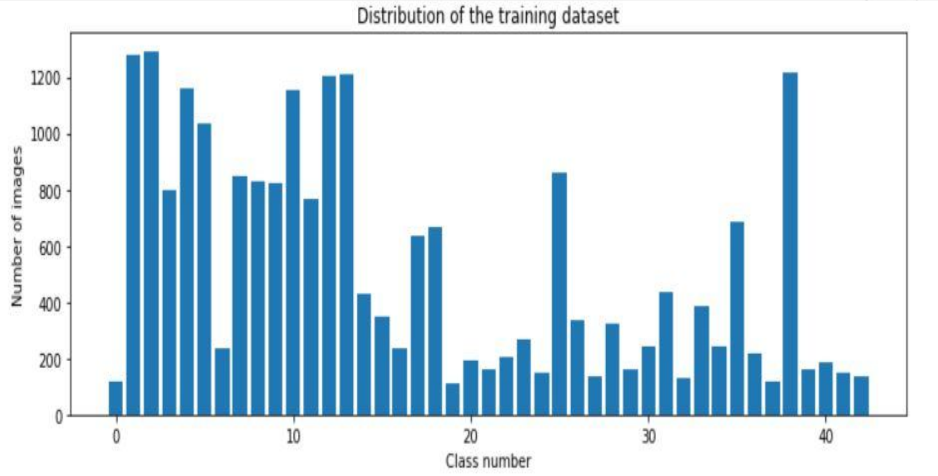
PROCEDURE

- Importing of images
- Splitting data into train, test and validation sets

```
testRatio = 0.2    # if 1000 images split will 200 for testing
validationRatio = 0.2 # if 1000 images 20% of remaining 800 will be 160 for validation
```

- Dataset consists of 43 classes

Sample images of classes

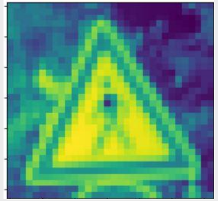


Distribution of training dataset

PROCEDURE

- Data preprocessing
  - Convert to grayscale
  - Standardize the lighting in an image
  - Normalize values between 0 and 1 instead of 0 and 255

Before preprocessing



After Preprocessing

- Image augmentation
  - To make images more generic
    - Shifting width and height
    - Magnitude of the shear angle and zoom level

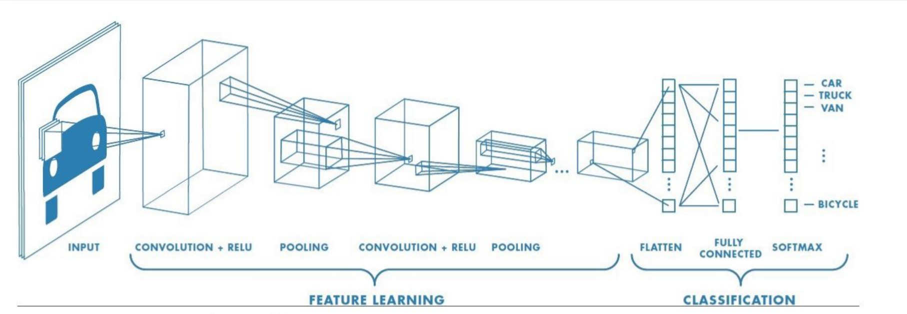
Image augmentation

MODEL DESIGNING AND IMPLEMENTATION

PROCEDURE

- Developing convolution neural network model
    - Defining number of filters and size of a filter
        - Kernel size 5x5
        - Remove 2 pixels from each border (when 32x32 images used)
    - More convolution layers added
        - For increase accuracy
    - In Pooling layers Scale down the feature map to generalize the solution
    - Dropout regularization
        - Better generalization
        - Reduce overfitting
    - 500 ,43 nodes in hidden layers in Dense Layer

Model Summary

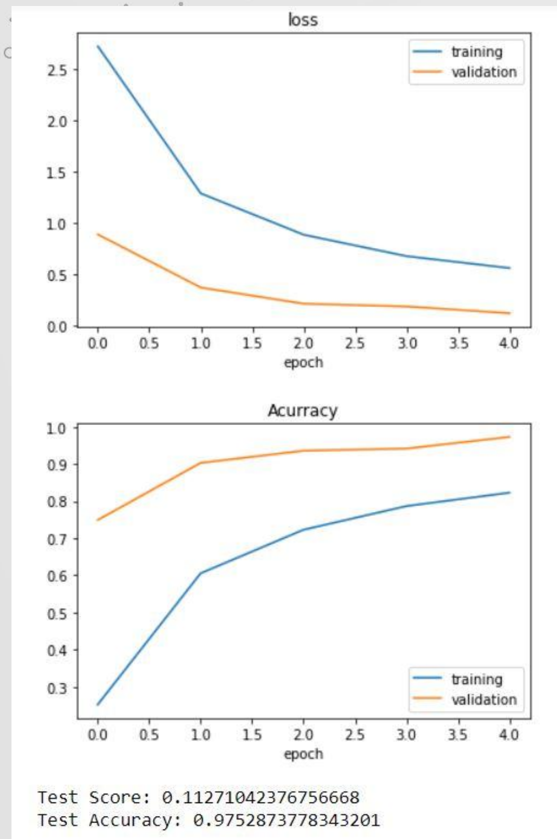


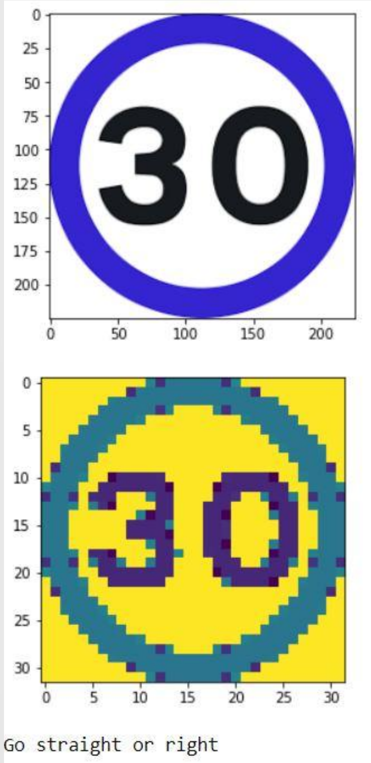

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 28, 28, 60)        1560
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 24, 24, 60)        90060
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 12, 12, 60)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 10, 10, 30)        16230
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 8, 8, 30)          8130
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 4, 4, 30)          0
_________________________________________________________________
dropout (Dropout)            (None, 4, 4, 30)          0
_________________________________________________________________
flatten (Flatten)            (None, 480)               0
_________________________________________________________________
dense (Dense)                (None, 500)               240500
_________________________________________________________________
dropout_1 (Dropout)          (None, 500)               0
_________________________________________________________________
dense_1 (Dense)              (None, 43)                21543
=================================================================
Total params: 378,023
Trainable params: 378,023
Non-trainable params: 0
```

```
Epoch 1/10
446/446 [==============================] - 74s 165ms/step - loss: 2.6546 - accuracy: 0.2611 - val_loss: 1.0425 - val_accuracy:
0.6649
Epoch 2/10
446/446 [==============================] - 74s 165ms/step - loss: 1.3224 - accuracy: 0.5922 - val_loss: 0.3956 - val_accuracy:
0.8664
Epoch 3/10
446/446 [==============================] - 69s 154ms/step - loss: 0.9045 - accuracy: 0.7143 - val_loss: 0.2252 - val_accuracy:
0.9339
Epoch 4/10
446/446 [==============================] - 69s 154ms/step - loss: 0.6974 - accuracy: 0.7817 - val_loss: 0.1787 - val_accuracy:
0.9456
Epoch 5/10
446/446 [==============================] - 66s 147ms/step - loss: 0.5742 - accuracy: 0.8184 - val_loss: 0.1277 - val_accuracy:
0.9652
Epoch 6/10
446/446 [==============================] - 67s 150ms/step - loss: 0.4951 - accuracy: 0.8413 - val_loss: 0.0863 - val_accuracy:
0.9777
Epoch 7/10
446/446 [==============================] - 65s 145ms/step - loss: 0.4433 - accuracy: 0.8600 - val_loss: 0.0854 - val_accuracy:
0.9772
Epoch 8/10
446/446 [==============================] - 70s 158ms/step - loss: 0.4038 - accuracy: 0.8727 - val_loss: 0.0680 - val_accuracy:
0.9826
Epoch 9/10
446/446 [==============================] - 70s 156ms/step - loss: 0.3639 - accuracy: 0.8832 - val_loss: 0.0630 - val_accuracy:
0.9813
Epoch 10/10
446/446 [==============================] - 70s 157ms/step - loss: 0.3520 - accuracy: 0.8890 - val_loss: 0.0488 - val_accuracy:
0.9878
```

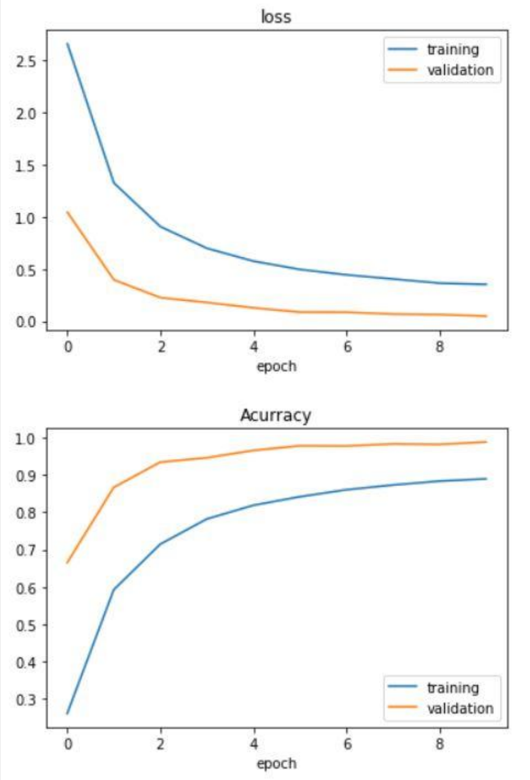
MODEL TESTING AND EVALUATION

- For 5 epochs


loss
training
validation
epoch
Acurracy
training
validation
epoch
Test Score: 0.11271042376756668
Test Accuracy: 0.9752873778343201

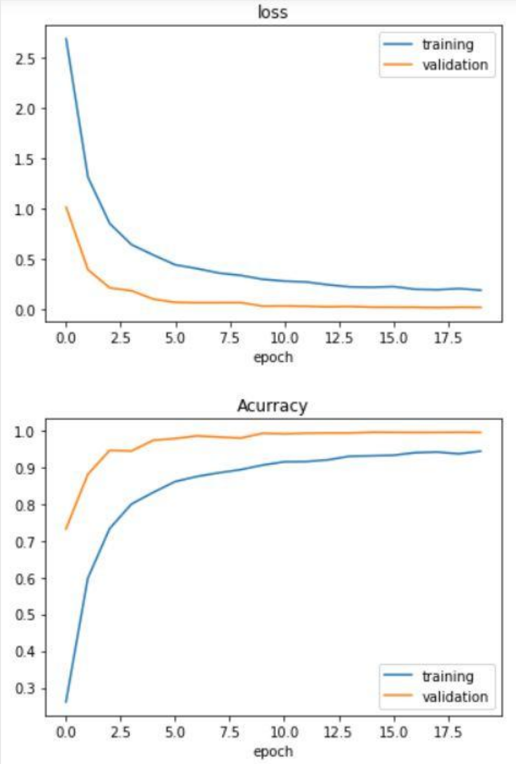

30
Go straight or right

- For 10 epochs



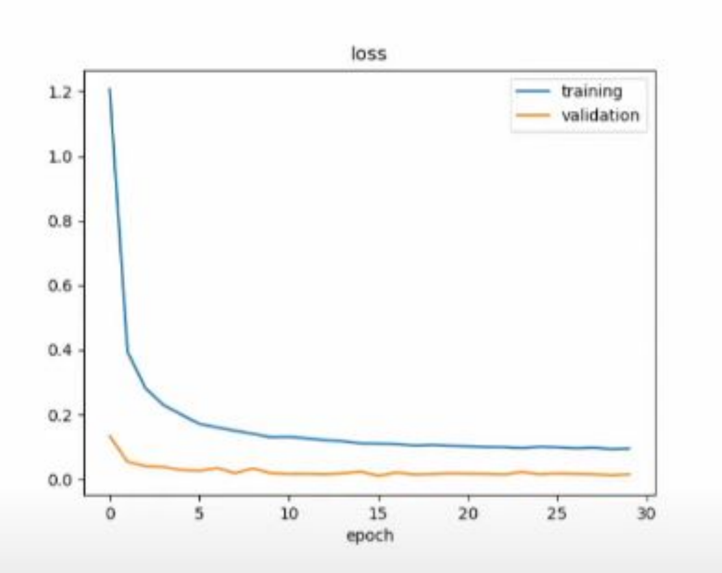

```
Test Score: 0.05001476779580116
Test Accuracy: 0.986494243144989
```

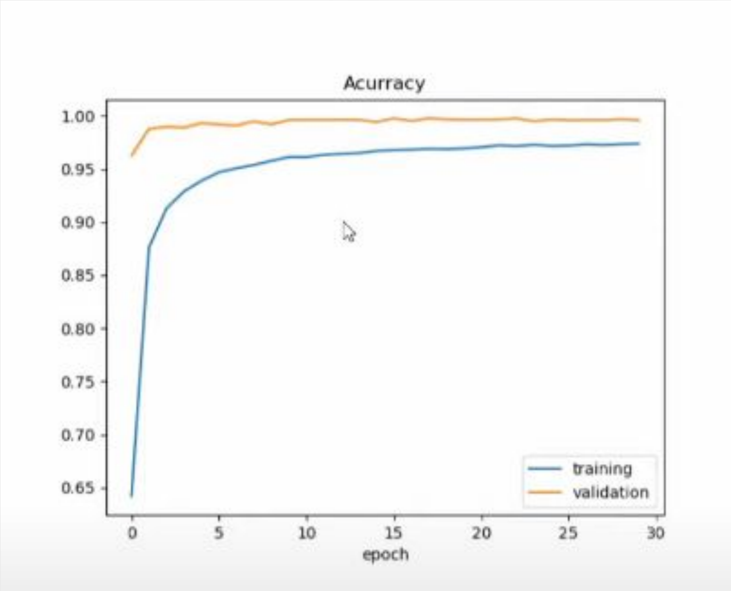
- For 20 epochs

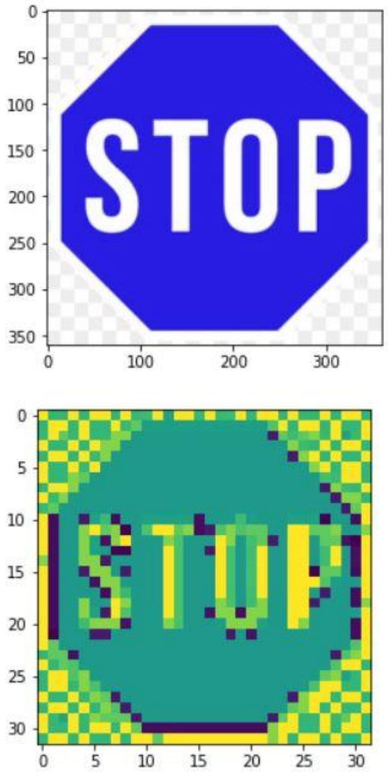



```
Test Score: 0.026479333639144897
Test Accuracy: 0.9933907985687256
```

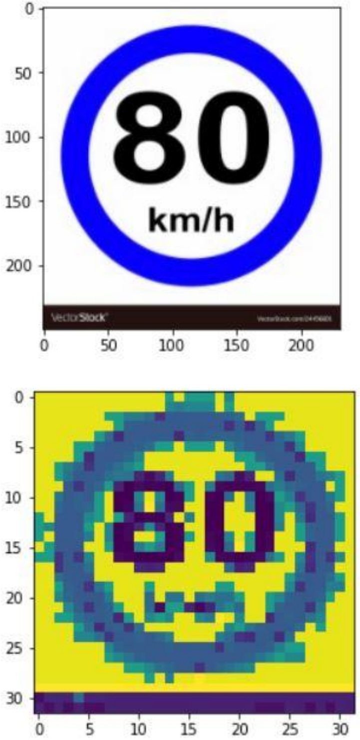
- For 30 epochs







Predicted sign : Stop
Predicted accuracy : 0.9303546



Speed limit (80km/h)
Predicted sign : Speed limit (80km/h)
Predicted accuracy : 0.9247637


Q
&
A

Thank you!