Computer Vision

Mini Project Report

Indian Denomination Classification

Submitted By: Aryan Tyagi (101816022)

Batch: COSE 3rd Year

Group: 3CS1

Submitted to: Dr. Shailendra Tiwari

Introduction

My aim was to build a system that can classify the different denominations in an image, so I created a denomination classifier that can identify the different currency note in a given image.

This classifier can classify amongst the currency notes of Rs.10, Rs.20, Rs.100 & Rs.200. This classification can further be expanded and trained for other different kind of currency notes and for any currency, for demonstration purposes we took only 4 classes.

My main problem was to device a system that can differentiate between different currency denominations, to minimise the human efforts, and to automate & digitalize the identification of currency notes, to help differently abled people to handle money transactions in a better way

Speaking about the blind people, our Indian currency notes are not made in that way that a blind person can identify them, by adding an audio feedback, it can successfully help in identifying the note.

At post offices where, the manpower is not enough, this system can assist in identification of the denomination, similar kind of scenario can be solved at the banks also.

This Entire project is implemented in Python 3.8

TensorFlow, Keras are used for core implementation along with other modules and Google Collaboratory for faster computations is used.

Data Collection

Data is collected from www.data.mendeley.com (source 1) which is having total different images of Rs.10, Rs.20, Rs.100 & Rs.200 notes in different orientations & with variable backgrounds. In some pictures even hands are there while clicking the images.

Due to less number of images, we also captured videos (source 2), featuring currency notes (shot in HD, for clearer images) & then screenshots were extracted from them, adding to our dataset.

Data was then segregated manually into different folders, for training purposes & simultaneously data cleaning & pre-processing, like removing blurred, duplicate, distorted images, was done.

Data Pre-processing

- All the images were mixed by default. Images were then segregated into different folders

- Duplicate images were removed from the dataset, as they could result in low accuracy of the model.

- Some images were having water marks on them, some notes were having something written on them, all that kind of images were dropped off from the dataset.

- **Some images were having different extensions, jpeg & jpg. They all were converted into jpg format with the help of pathlib module.**

Method and Models

Model:

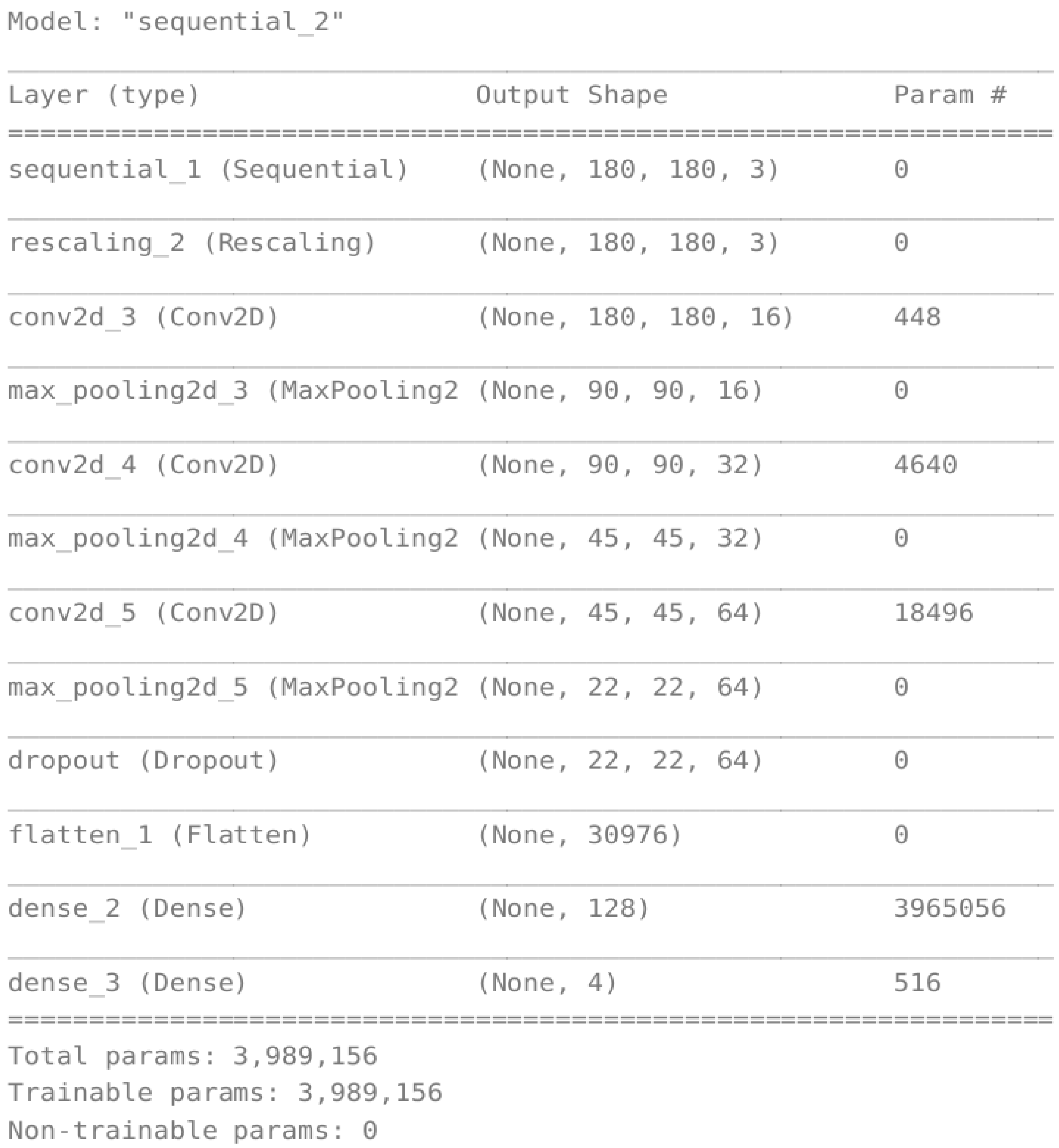
The model has three convolution blocks with a max pool layer in each layer.

The layers are activated by the relu activation function

The RGB channels were also standardized to fit in the [0, 1] range from the [0, 255] as neural networks prefer that range. This was done by using a Rescaling layer.

As for the loss function, since we have multiple classes, we used the SparseCategoricalCrossentropy loss function and the optimizer used was Adam.

Model Summary:



```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential_1 (Sequential)    (None, 180, 180, 3)       0
_________________________________________________________________
rescaling_2 (Rescaling)      (None, 180, 180, 3)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 180, 180, 16)      448
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 90, 90, 16)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 90, 90, 32)        4640
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 45, 45, 32)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 45, 45, 64)        18496
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 22, 22, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 22, 22, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 30976)             0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               3965056
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 516
=================================================================
Total params: 3,989,156
Trainable params: 3,989,156
Non-trainable params: 0
```

After creating the dataset, I trained a multi class classifier on my dataset that could differentiate between 10, 20, 100 and 200 notes.

The trained model would be fed an image of a note and it would then tell us (according to the model) the class the note belongs to.

The model was trained over 100 epochs of the dataset. At the end of the training an accuracy of 92 percent was achieved over the validation set.

ML Techniques used

- Deep Learning (CNN)
- Classification
- Dropout
- Data Augmentation

Limitations

- This system can't read multiple notes in a single image. The image should only contain a single note of currency.

- Till now, it cannot detect the location of the note in image. It only predicts amount of the note

- It cannot detect multiple notes of same currency, of same denomination.

Conclusion

We trained our model using TensorFlow and we were able to get an accuracy of 92 percent.

Previously, you have to manually enter the image path in the script but then we modified the script to get the input from the web cam. Just run the script, it will prompt you to input the image through the web cam, and after clicking the picture (like holding the note with your hand(s)), it will predict the denomination from the image.

Future Scope

- We can make it self-learning, so that whenever anyone gives an image to our model, it only classifies the note, but also add that image into the dataset and get trained on it. So, it will keep on adapting and improving itself as we use it to predict.

- We can extend it to any kind of currency with any number of classifications for denomination.

- We can also install support with audio feedback, or turn it into a fully fledged software, to make it more interactive with the outer world. Audio feedback can help blind people to use this system.

- We can also train this software detect any kind of cracks, or is the note is in whole shape or not.

- With the help of object detection, we can also detect multiple notes in a single image, having same notes of single/multiple currency.

References

The documentations and references that were helpful in completion of this project

[1] www.keras.io

[2] www.kaggle.com

[3] www.tensorflow.org

[4] www.data.mendeley.com