Project- Currency Detection:

Project Description:

The aim of our project is to detect and identify currency notes from various countries. The approach we have chosen is to implement Convolutional Neural Networks to first identify the country, and then identify the denomination.

Approach

- Creating a dataset of images from various countries and denominations.
- Created two CNNs, one to identify the country, and one to identify the denomination.
- Take input from image upload, perform necessary transformations, and run it through the model to identify the country
- If the country is India, run it through another network to identify the denomination of Indian currency
- For any other currency, decide the denomination using the neural network to declare the denomination
- Print the country, and denomination with the highest probability, yet permissible depending on the available denominations from the country

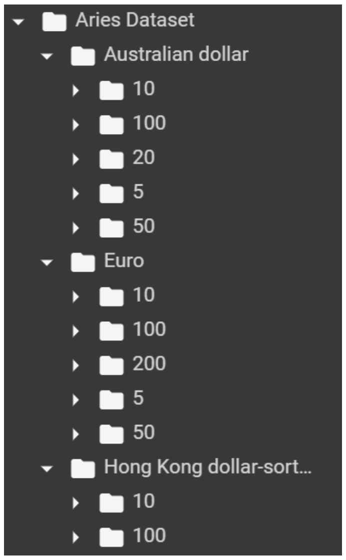
Tech stack

- Google collab
- Python IDLE
- Python
- Keras
- Tensorflow
- Numpy
- Pandas
- Open-CV
- Bing for dataset

Implementation:

Dataset Creation:

Due to the lack of datasets available online, and the unavailability of physical foreign currency banknotes, we used web scraping techniques to create a dataset. We compiled thousands of images to create a dataset of 17 countries, each divided into their respective denominations.



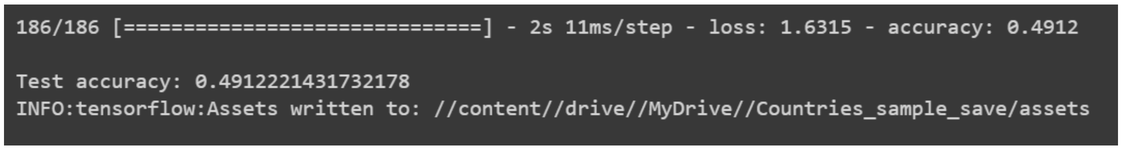
This posed another issue, labeling of data. We decided the labeling of data should be made depending on the folder name.

As we have 2 models(excluding the model to identify Indian currency), we need to compile two datasets. For the country model, we labeled the data according to the upper folder, and for the denomination model, the labels were given based on the inner folder.
As the data was not normalized(for example it ranged from 400 images per class to 1.5k images per class), necessary data augmentation techniques were applied. Images were converted to grayscale using opencv, and they were normalized. Finally, the datasets were split into test and train in a ratio of 7:3.

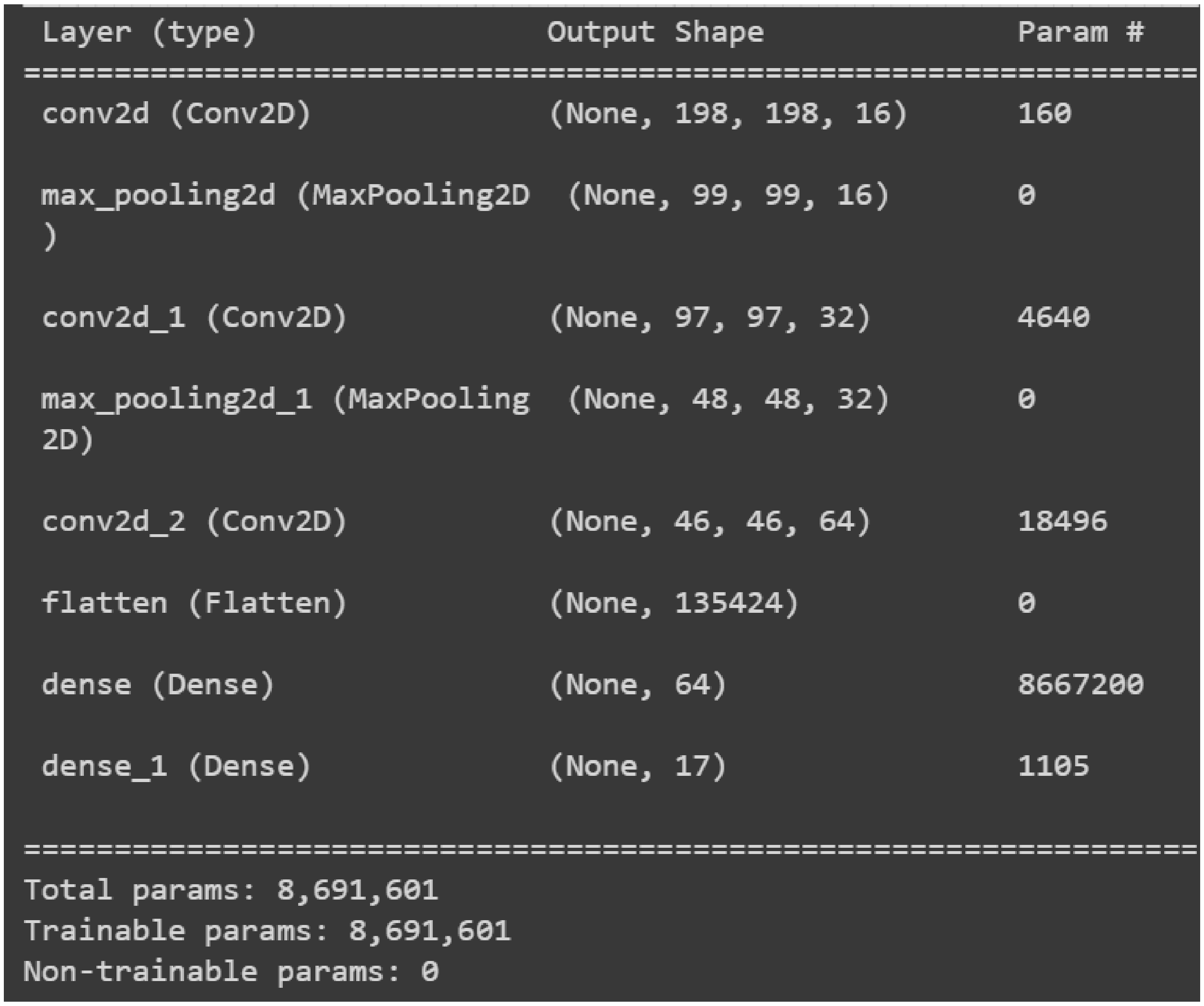
Creating the Models:

Three models were created for this project. One for the prediction of countries, one for the prediction of denomination, and one specifically for the prediction of Indian currencies. For optimum results, we used early stopping with patience as 3. We used adam optimizer and the loss function chosen was Sparse Categorical Crossentropy. The activation function of each layer was Relu.

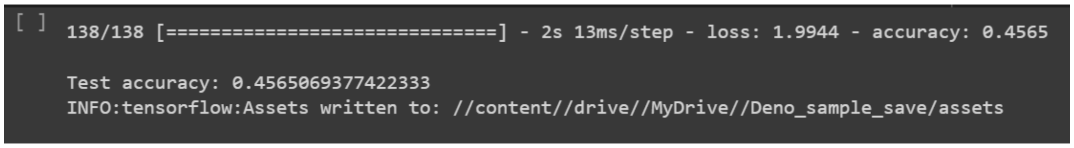
1. **Countries:** We used Keras to implement this model. The input was a 200x200x1 (as it was grayscale), and the network was 4 layers deep. The output layer had 17 nodes. The accuracy came out to be 49%.



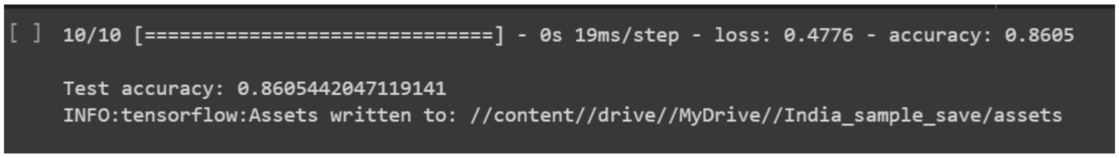
```
186/186 [==============================] - 2s 11ms/step - loss: 1.6315 - accuracy: 0.4912

Test accuracy: 0.4912221431732178
INFO:tensorflow:Assets written to: //content//drive//MyDrive//Countries_sample_save/assets
```


```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 198, 198, 16)      160

 max_pooling2d (MaxPooling2D  (None, 99, 99, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 97, 97, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 48, 48, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 46, 46, 64)        18496

 flatten (Flatten)           (None, 135424)            0

 dense (Dense)               (None, 64)                8667200

 dense_1 (Dense)             (None, 17)                1105

=================================================================
Total params: 8,691,601
Trainable params: 8,691,601
Non-trainable params: 0
```

2. **Denominations:** Again, Keras was used to implement a 4 layer-deep model. The accuracy obtained was 45%.


```
[ ]  138/138 [==============================] - 2s 13ms/step - loss: 1.9944 - accuracy: 0.4565

     Test accuracy: 0.4565069377422333
     INFO:tensorflow:Assets written to: //content//drive//MyDrive//Deno_sample_save/assets
```

3. **India:** For this model, the input layer had 3 channels, instead of 1. The model also included a layer for further data augmentation. Finally, a 4 layer-deep model was used.


```
[ ]  10/10 [==============================] - 0s 19ms/step - loss: 0.4776 - accuracy: 0.8605

     Test accuracy: 0.8605442047119141
     INFO:tensorflow:Assets written to: //content//drive//MyDrive//India_sample_save/assets
```

Future:

The future of this project is to create a well-functioning network hosted on a website, where we can input an image into the website or take a picture from a web camera, it will run it through the country prediction network to predict the country, and then the denomination prediction network to finally return the prediction. We can apply this for a real-world application as well, where we can use this website and it could narrate the output prediction, which could be used for currency identification for the blind.

Alternate approaches/Issues Faced

An alternate approach to denomination detection was based on reimagining the aim of the model. The aim of the model was to identify the numbers on the image or the digits on the image. The aim was just to return the denomination. We thought of various methods to approach this, one was using pytesseract, which didn't yield any good results. Another approach was some pre-processing of the dataset to highlight the number on the digits and reduce unnecessary noise, using some preprocessing techniques offered by open-cv. Either of these approaches wasn't fruitful

Some other issues faced were related to data augmentation and the size of the dataset. When the same algorithms and models were used on smaller datasets, containing only 6 countries, the output was much better, and the accuracy achieved was about 82% for countries and 69% for denominations. However, this accuracy was lowered on more countries.

Resources:

https://phdservices.org/currency-recognition-using-opencv-python/
https://www.pantechelearning.com/product/currency-detection-using-machine-learning-opencv-and-python/
https://www.phddirection.com/currency-recognition-using-image-processing-python/
https://www.journals.resaim.com/ijresm/article/view/108
https://www.irjet.net/archives/V9/i1/IRJET-V9I136.pdf
https://orapp.aut.ac.nz/bitstream/handle/10292/12032/ZhangQ.pdf?sequence=3&isAllowed=y
https://www.ijrte.org/wp-content/uploads/papers/v9i2/B3955079220.pdf
https://ieeexplore.ieee.org/document/9432274
https://link.springer.com/chapter/10.1007/978-981-15-6648-6_6