Currency and Fake Currency Detection Using Machine Learning and Image Processing-An Application for Blind People using Android Studio

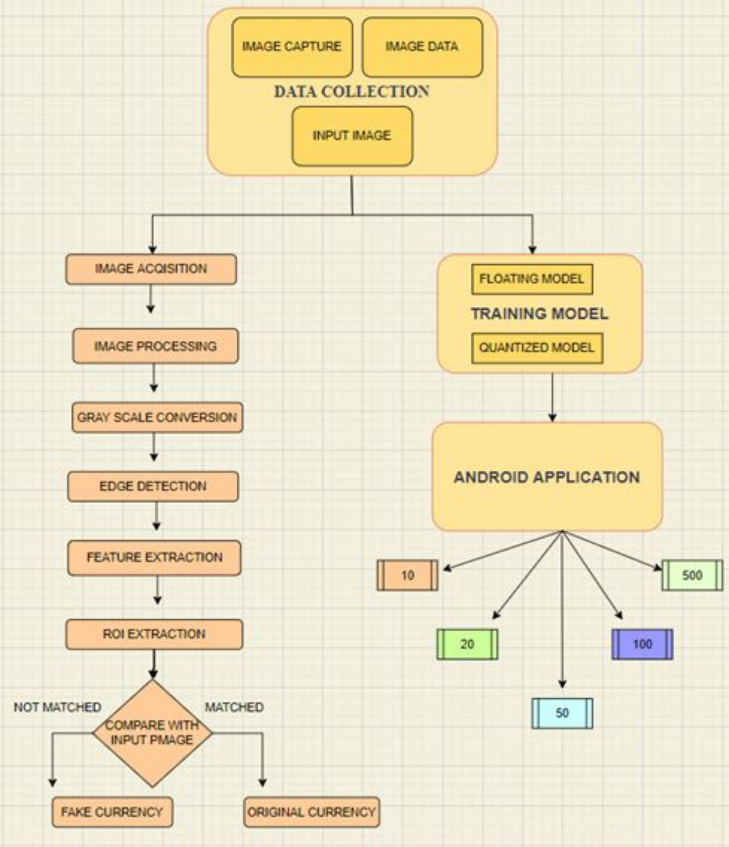
Aims & Objectives

- To identify the fake currencies from the original ones.
- To eliminate the fake currency from the note circulation
- To guide the blind persons to identify the currency using the current technology
- To implement the application of image recognition in eradication of fake currency
- To bring advancement in the field of currency recognition

Outline

Different methodologies for identify the currency

1. Traditional method
2. Digital Image Processing Technique
3. Android Based Detection

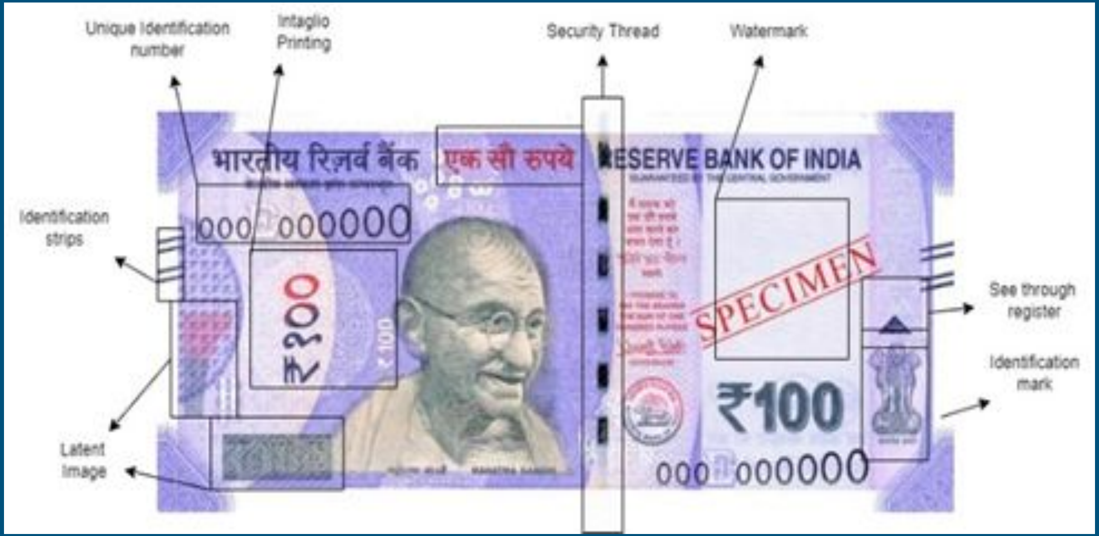


Traditional Method

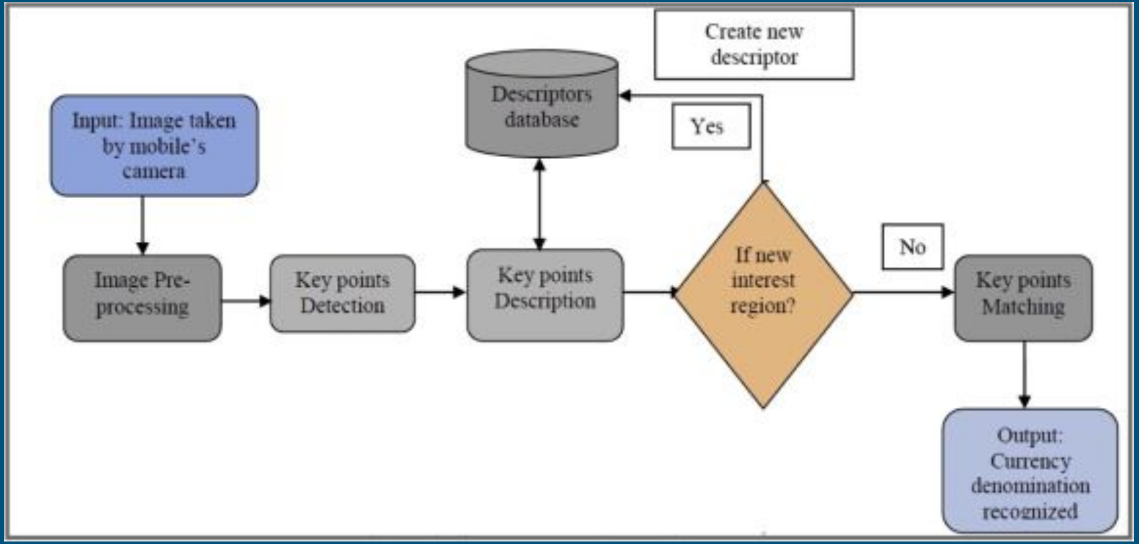
Different Identification marks:

1. Watermarks
2. Fluorescence
3. Latent Images
4. Security Threads
5. Intaglio Printing
6. Optically Variable ink
7. Micro Lettering

etc.,



Digital Image Processing Technique



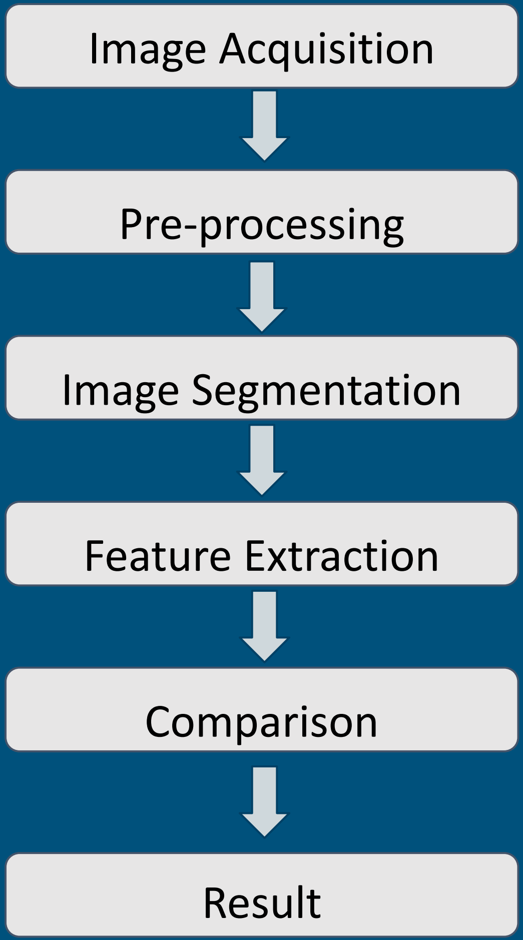


Image Acquisition

Pre-processing

Image Segmentation

Feature Extraction

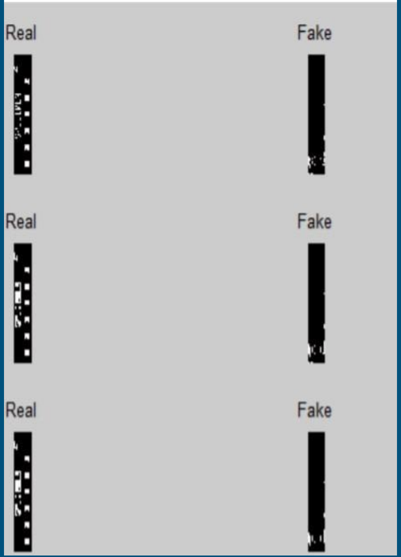
Comparison

Result

Digital Image Processing Technique -

Experimental Results

Gray Scale Converted Image



Grayscale image cropped and converted to binary image

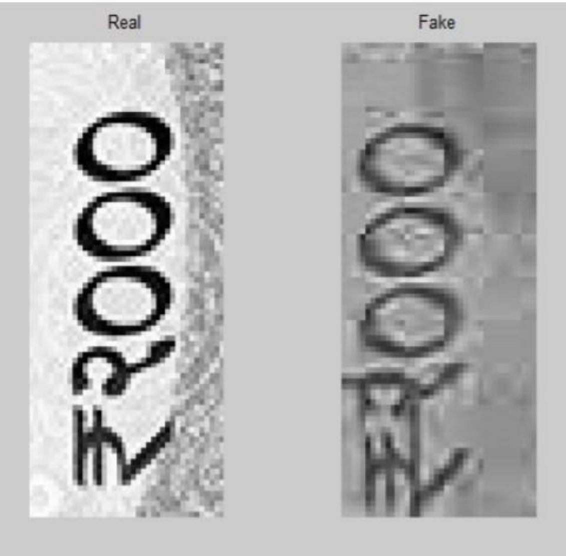


Image cropped and enlarged for comparing

Code for Simulation

```
clc;
close all
%_______Image Reading_______________
[FILENAME,PATHNAME]=uigetfile("*.jfif","Select the
Image");
FilePath=strcat(PATHNAME,FILENAME);
disp('The Image file location is:');
disp(FilePath);
[DataArray,map]=imresize(imread(Filepath),[300,650]);
figure,imshow(DataArray,map);
title("Input Image");
%________Seperate Channel__________
r_channel=DataArray(:,:,1);
b_channel=DataArray(:,:,2);
g_channel=DataArray(:,:,3);
%______Noise Removal____
r_channel=medfilt2(r_channel);
g_channel=medfilt2(g_channel);
b_Channel=medfilt2(b_channel);
%_________Restore channels________
rgbim(:,:,1)=r_channel;
rgbim(:,:,2)=g_channel;
rgbim(:,:,3)=b_channel;
figure,imshow(uint8(rgbim));
title('Denoised Image');
% RGB to Gray
Igray = 0.30*r_channel + 0.59*g_channel +
0.11*b_channel;
figure,imshow(uint8(Igray));
title('Gray Image');
```

```
%______Edge Detection________
y=double(Igray);
f1=zeros(3,3,5);
f1(:,:,1)=[1 2 1;0 0 0;-1 -2 -1];
f1(:,:,2)=[-1 0 1;-2 0 2;-1 0 1];
f1(:,:,3)=[2 2 -1;2 -1 -1;-1 -1 -1];
f1(:,:,4)=[-1 2 2;-1 -1 2;-1 -1 -1];
f1(:,:,5)=[-1 0 1;0 0 0;1 0 -1];
for i=1:5
   g.im(:,:,i)=filter2(f1(:,:,i),y);
end
[m,p]=max(g.im,[],3);
edim=edge(y,'canny');
im2=(p.*edim);
edhist=im2;
figure,imshow(edhist)
title("Edge Detection");
Avg=mean2(Igray);
if(Avg>202 && Avg<207)
   load 100.mat
elseif(Avg>175 && Avg<180)
   load 200.mat
elseif(Avg>190 && Avg<195)
   load 500.mat
elseif(Avg>209 && Avg<214)
   load 2000.mat
end
```

Code for Simulation

```
A= [];
figure,imshow(uint8(DataArray));
title('ROI Extracted Texture & Statstical Features');
hold on
for n=1:size(A,1)
   rectangle('Position',A(n,1),'EdgeColor','r','Linewidth',4)
end
pause(1)
figure,imshow(edhist);
title("ROI Extracted Edge and shape Features:");
hold on
for n=1:size(A,1)
   rectangle('Position',A(n,1),'EdgColor','r,','LineWidth',4)
end
pause(1)
SFL_Data=zeros(size(A,1),6);
SSL_Data=zeros(size(A,1),12);
for n=1:size(A,1)
   imcropgray = imcrop(Igray,A(n,1));
   Img_data=imcropgray;
   % STATISTICAL FEATURES
   % FIRST LEVEL FEATURES
   Mean = mean2(Img_data);
   Variance = mean2(var(double(Img_data)));
   Kurtosis = kurtosis(double(Img_data(:)));
   stats = graycoprops(Img_data,'Contrast Correlation Energy
Homogeneity');
   Energy = stats.Energy;
   Constrast = stats.Constrast;
   Entropy = entropy(Img_Data);
```

```
FL_Feat=[Mean Variance Kurtosis Energy Contrast Entropy];
   FL_Feat(isnan(FL_Feat))=0;
   % disp('First Level Features');
   % disp(FL_Feat);
   SFL_Data(n,:)=FL_Feat;
   % Second Level Features
   offsets = [0 1;-1 1;-1 0;-1 -1];
   GLCM1 =
graycomatrix(Img_data,'NumLevels',8,'offset',offsets);
   GLCM2 =
graycomatrix(Img_data,'NumLevels',32,'offset',offsets);
   stats = graycoprops(GLCM1,'Constrast Correlation Energy
Homogeneity');
   stats1 = graycoprops(GLCM2,'Constrast Correlation Energy
Homogeneity');
   Correlation=[mean(stats.Correlation)
mean(stats1.Correlation)];
   ASM=[mean(stats.Energy) mean(stats1.Energy)];
   Homogeneity=[mean(stats.Homogeneity)
mean(stats1.Homogeneity)];
   IDM=[Inverse_Diff(GLCM1) Inverse_Diff(GLCM2)];
   Max_prob=[Maximum_Prob(GLCM1) Maximum_Prob(GLCM2)];
   Entropy = [entropy(GLCM1) entropy(GLCM2)];
   SL_Feat=[ASM Correlation Homogoneity IDM Max_Prob Entropy];
   SL_Feat(isnan(SL_Feat))=0;
   %    disp('Second Level Features')
   %    disp(SL_Feat)
   SSL_Data(n,:)=SL_Feat;
end
ST_feat=[mean(SFL_Data) mean(SFL_Data)];
disp('Statistical Features')
disp(ST_feat);
EF_Data=zeros(size(A,1),7);
for n=1:size(A,1)
   imcropedge = imcrop(edhist,A(n,1));
```

Code for Simulation

```
% EDGE FEATURES

results=regionprops(imcropedge,'Area','EulerNumber','Orientati
on','BoundingBox','Extent',...'Perimeter','Centroid','Extrema'
,'PixelIdxList','ConvexArea',...'FilledArea','PixelList','Conv
exHull','FilledImage','Solidity',...'ConvexImage','Image','Sub
arrayIdx','Eccentricity','MajorAxisLength',...
 'EquivDiameter','MinorAxisLength','EulerNumber');
   NR=vertcat(results.BoundingBox);
   Circularity=zeros(size(NR,1));
   Eccentrcicty=zeros(size(NR,1));
   Convexity=zeros(size(NR,1));
   Area=zeros(size(NR,1));
   Compactness=zeros(size(NR,1));
   Extent=zeros(size(NR,1));
   Solidity=zeros(size(NR,1));
   for ii=1:size(NR,1)

Circularity(ii)=((results(ii).Perimeter).^2)./(4*(pi*(results(
ii).Area)));
       Circularity(isnan(Circularity))=0;
       Circularity(isinf(Circularity))=0;

Compactness(ii)=(4*results(ii).Area*pi)/(results(ii).Perimeter
).^2;
       Compactness(isnan(Compactness))=0;
       Compactness(isinf(Compactness))=0;
       Convexity(ii)=results(ii).ConvexArea;
       Convexity(isnan(Convexity))=0;
       Convexity(isinf(Convexity))=0;
       Area(ii)=results(ii).Eccentricity;
       Area(isnan(Area))=0;
       Area(isinf(Area))=0;
```

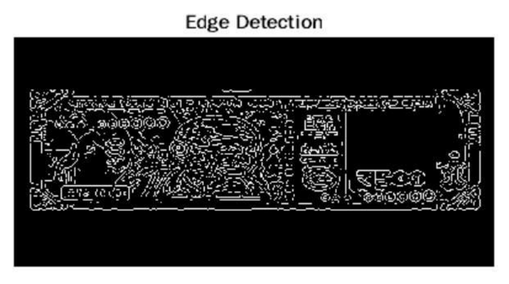
```
Eccentricity(ii)=results(ii).Eccentricity;
       Eccentricity(isnan(Eccentricity))=0;
       Eccentricity(isinf(Eccentricity))=0;
       Extent(ii)=results(ii).Extent;
       Extent(isnan(Extent))=0;
       Extent(isinf(Extent))=0;
       Solidity(ii)=results(ii).Solidity;
       Solidity(isnan(Solidity))=0;
       Solidity(isinf(Solidity))=0;
   end
   SF=[mean2(Area) mean2(Solidity) mean2(Convexity)
mean2(Circularity) mean2(Eccentricity) mean2(Compactness)
mean2(Extent)];
   EF_Data(n,:)=SF;
end
EDF_feat=mean(EF_Data);
disp('Edge Features')
Tfeat=[ST_feat EDF_feat];
load Pdata.mat
load Ndata.mat
xdata = [Train_dataP;Train_dataN];
group = [Train_LabP;Train_LabN];
svmTrain = fitcsvm(xdata,group,'Kernel_function','rbf');
Classify_Result = ClassificationSVM(svmTrain,Tfeat);
if(Classify_Result==1)
   figure,imshow(DataArray,map);
   title("Currency Type: REAL");
   msgbox('Currency Type: REAL');
else
   figure,imshow(DataArray,map);
   title("Currency Type: FAKE");
   msgbox("Currency Type: FAKE");
end
```

Simulation output

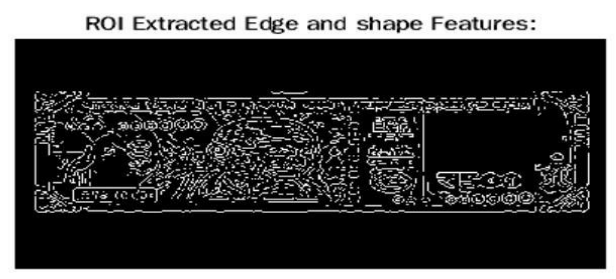
Input Image

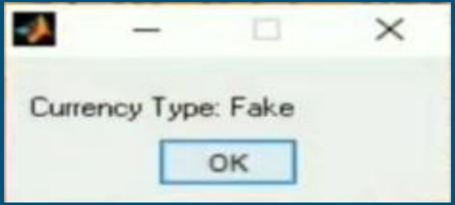
Denoised Image

Gray Image


Edge Detection

ROI Extracted Texture & Statstical Features

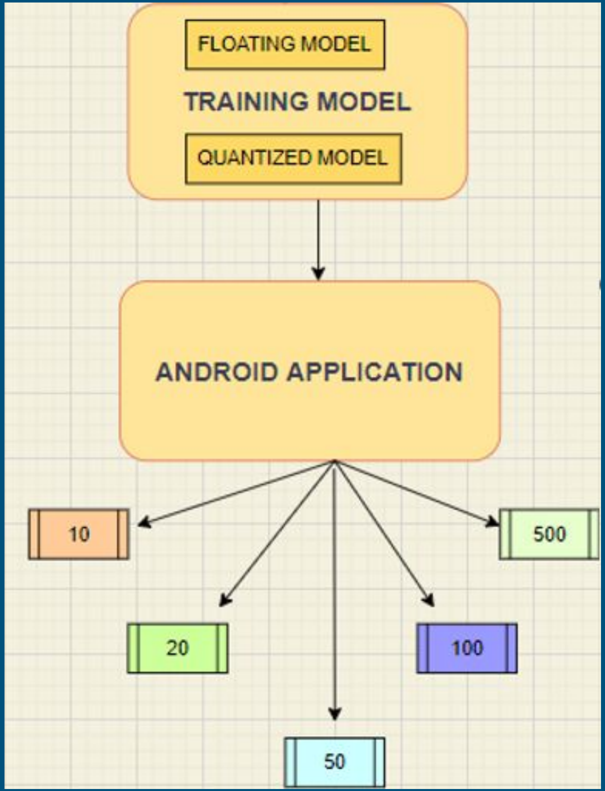

ROI Extracted Edge and shape Features:


Currency Type: Fake

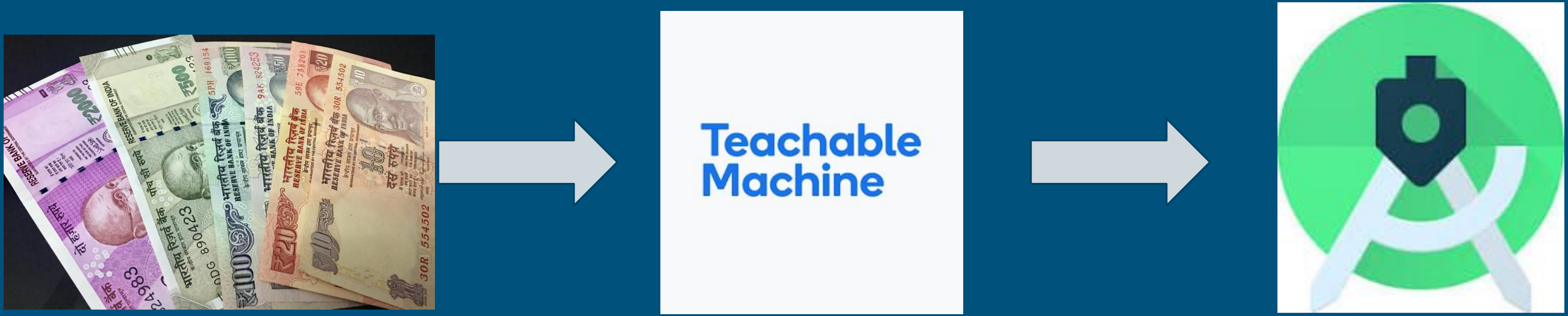
Android Based Currency Detection

It involves 3 steps:

1. Data Collection
2. Training the model
3. Building a software application



Android Based Currency Detection



Data Collection

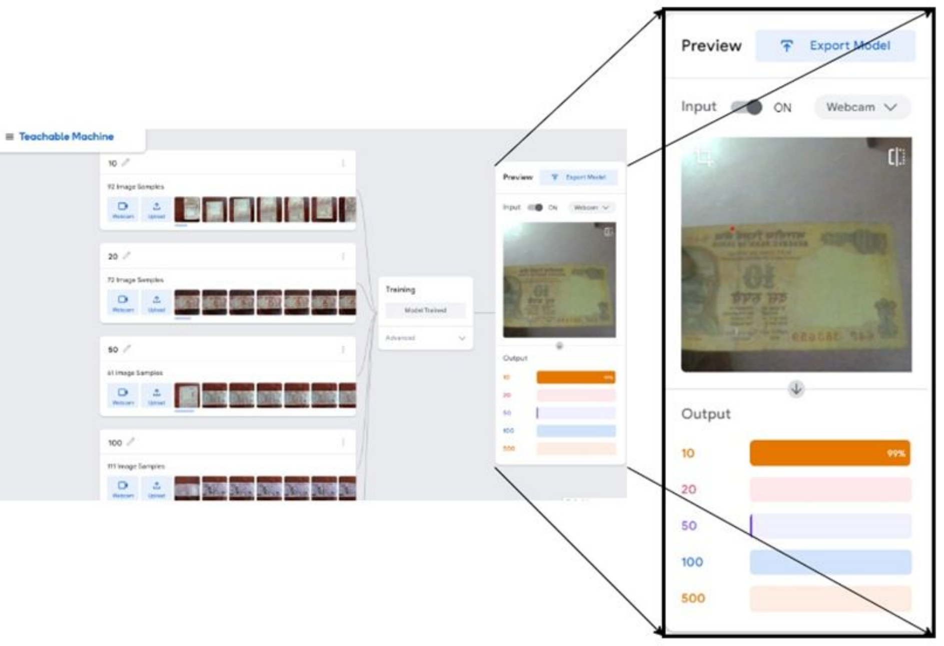
Teaching the model/machine

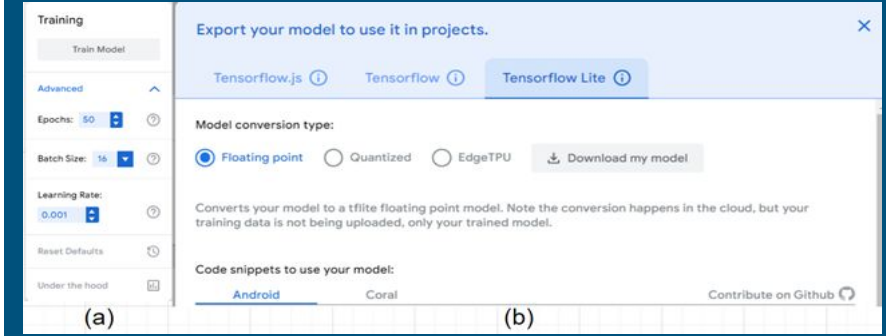
Application Creation

Android Based Currency Detection

Step 1: Collection the data by means of android devices like mobile / camera ..

Step 2 : Teaching the model





Epochs: 50

Batch Size: 16

Learning Rate: 0.001

Export your model to use it in projects.

Model conversion type: ● Floating point ○ Quantized ○ EdgeTPU

Converts your model to a tflite floating point model. Note the conversion happens in the cloud, but your training data is not being uploaded, only your trained model.

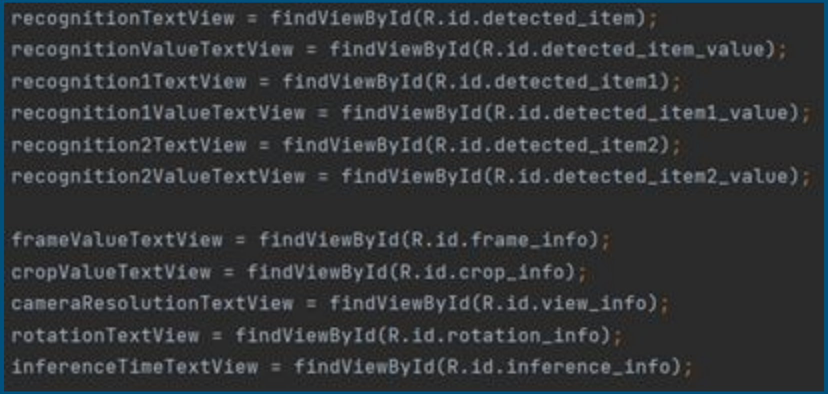
Code snippets to use your model:

Teaching the model/machine

Importing Files

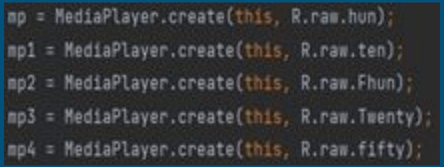
Android Based Currency Detection

Step 3 : Building a software application using Android Studio

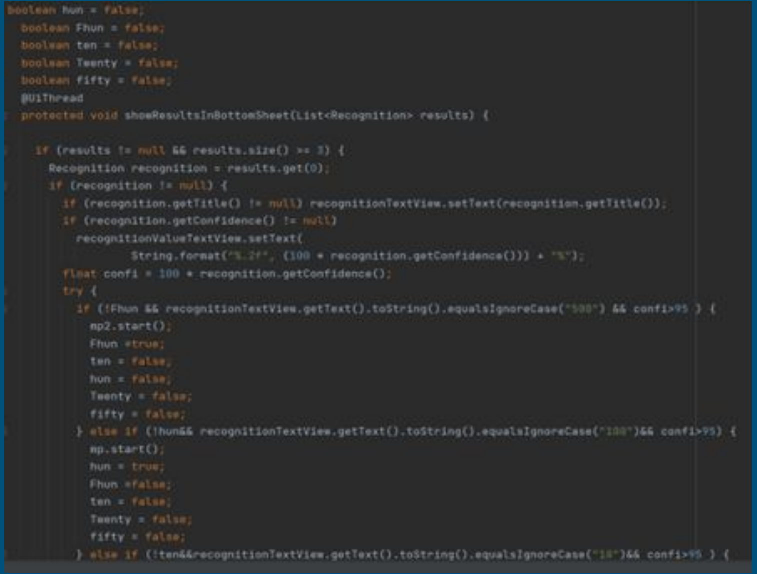


```
recognitionTextView = findViewById(R.id.detected_item);
recognitionValueTextView = findViewById(R.id.detected_item_value);
recognition1TextView = findViewById(R.id.detected_item1);
recognition1ValueTextView = findViewById(R.id.detected_item1_value);
recognition2TextView = findViewById(R.id.detected_item2);
recognition2ValueTextView = findViewById(R.id.detected_item2_value);

frameValueTextView = findViewById(R.id.frame_info);
cropValueTextView = findViewById(R.id.crop_info);
cameraResolutionTextView = findViewById(R.id.view_info);
rotationTextView = findViewById(R.id.rotation_info);
inferenceTimeTextView = findViewById(R.id.inference_info);
```



```
mp = MediaPlayer.create(this, R.raw.hun);
mp1 = MediaPlayer.create(this, R.raw.ten);
mp2 = MediaPlayer.create(this, R.raw.Fhun);
mp3 = MediaPlayer.create(this, R.raw.Twenty);
mp4 = MediaPlayer.create(this, R.raw.fifty);
```

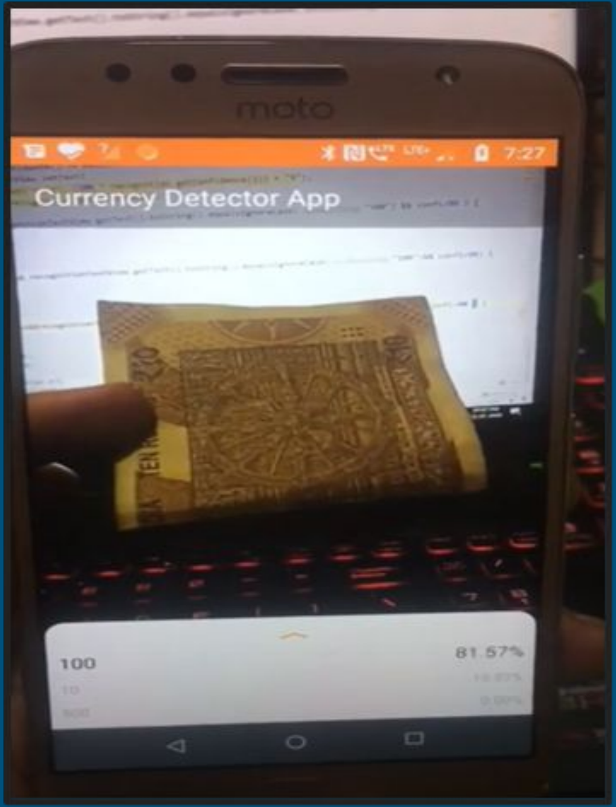


```
boolean hun = false;
  boolean Fhun = false;
  boolean ten = false;
  boolean Twenty = false;
  boolean fifty = false;
  @UiThread
  protected void showResultsInBottomSheet(List<Recognition> results) {

    if (results != null && results.size() >= 3) {
      Recognition recognition = results.get(0);
      if (recognition != null) {
        if (recognition.getTitle() != null) recognitionTextView.setText(recognition.getTitle());
        if (recognition.getConfidence() != null)
          recognitionValueTextView.setText(
                  String.format("%.2f", (100 * recognition.getConfidence())) + "%");
        float confi = 100 * recognition.getConfidence();
        try {
          if (!Fhun && recognitionTextView.getText().toString().equalsIgnoreCase("500") && confi>95 ) {
            mp2.start();
            Fhun =true;
            ten = false;
            hun = false;
            Twenty = false;
            fifty = false;
          } else if (!hun&& recognitionTextView.getText().toString().equalsIgnoreCase("100")&& confi>95) {
            mp.start();
            hun = true;
            Fhun =false;
            ten = false;
            Twenty = false;
            fifty = false;
          } else if (!ten&&recognitionTextView.getText().toString().equalsIgnoreCase("10")&& confi>95 ) {
```

Android Based Currency Detection -

Experimental Results



Currency Detector App

```
100   81.57%
10    [illegible]
500   0.00%
```

The output is showing that the currency detected is an Rs 10 indian domination with a accuracy of 81.57%

Summary

The efficiency of the system can be increased by proper and the number count in data collection step.

Thank You