

Project Report

Ghassen Chaabouni

02/08//2019

Plan

- 1-Preparing a dataset
 - 2-The model
 - 3-The app
-

Money classification app for the blinds

1-Preparing a dataset

I prepared more than 2000 pictures consisting of different Tunisian money bills.

2-The model

I used a fully connected convolutional neural network consisting of one hidden layer. After training the model I got more than 90% accuracy on both training and validation datasets.

3-The app

- To run classification on android I used [tensorflow-for-poets-2](#)
- These are the steps I took to modify the app
 - 1-I replaced the existing model and labels.
 - 2-I added a few lines of code to add a voice for each bill.

ImageClassifier.java

I created two new functions called “money()” and “classifyMoney(Bitmap bitmap)” to return the value of a bill when the probability is greater than 0.9

```
private String money() {
    for (int i = 0; i < labelList.size(); ++i) {
        sortedLabels.add(
            new AbstractMap.SimpleEntry<>(labelList.get(i), labelProbArray[0][i]));
        if (sortedLabels.size() > RESULTS_TO_SHOW) {
            sortedLabels.poll();
        }
    }
    String money="";
    int _10=0, _20=0, _50=0, _5=0;
    boolean B_10=false, B_20=false, B_50=false, B_5=false;
    final int size = sortedLabels.size();
    for (int i = 0; i < size; ++i) {
        Map.Entry<String, Float> label = sortedLabels.poll();
        B_10 = (Objects.equals(String.format("%s", label.getKey()), b: "10"));
        B_20 = (Objects.equals(String.format("%s", label.getKey()), b: "20"));
        B_50 = (Objects.equals(String.format("%s", label.getKey()), b: "50"));
        B_5 = (Objects.equals(String.format("%s", label.getKey()), b: "5"));
        money="";
        if ((B_10 == true) && _10 == 0 && label.getValue() > 0.90) {
            Log.d(TAG, msg: "---10---\n");
            money="10";
            _10++;
        }
        if ((B_20 == true) && _20 == 0 && label.getValue() > 0.90 ) {
            Log.d(TAG, msg: "---20---\n");
            money="20";
            _20++;
        }
        if ((B_50 == true) && _50 == 0 && label.getValue() > 0.90) {
```

```

    if ((B_50 == true) && _50 == 0 && label.getValue() > 0.90) {
        Log.d(TAG, "msg: \"---50---\\n\");
        money = "50";
        _50++;
    }
    if ((B_5 == true) && _5 == 0 && label.getValue() > 0.90) {
        Log.d(TAG, "msg: \"---5---\\n\");
        money = "5";
        _5++;
    }
}
return money;
}

String classifyMoney(Bitmap bitmap) {
    if (tflite == null) {
        Log.e(TAG, "Image classifier has not been initialized; Skipped.");
        return "Uninitialized Classifier.";
    }
    convertBitmapToByteBuffer(bitmap);
    // Here's where the magic happens!!!
    long startTime = SystemClock.uptimeMillis();
    tflite.run(imgData, labelProbArray);
    long endTime = SystemClock.uptimeMillis();
    // smooth the results
    applyFilter();

    // print the results
    String money = money();
    return money;
}

```

Camera2BasicFragment.java

• I created new variables. Each new variable is an array because I need to modify their value later in a function.

```

private int[] _10 = new int[1];
private int[] _20 = new int[1];
private int[] _50 = new int[1];
private int[] _5 = new int[1];

```

- I modified these values to get the best performance.

```
/** Number of results to show in the UI. */
private static final int RESULTS_TO_SHOW = 1;

/** Dimensions of inputs. */
private static final int DIM_BATCH_SIZE = 1;

private static final int DIM_PIXEL_SIZE = 3;

static final int DIM_IMG_SIZE_X = 150;
static final int DIM_IMG_SIZE_Y = 150;

private static final int IMAGE_MEAN = 30;
private static final float IMAGE_STD = 300.0f;
```

- I initialised these variables every time a bill is classified to avoid repetitive sounds which is an issue I encountered. I found out the best way to do that is in “onViewCreated” function.

```
public void onViewCreated(final View view, Bundle savedInstanceState) {
    textureView = (AutoFitTextureView) view.findViewById(R.id.texture);
    textView = (TextView) view.findViewById(R.id.text);
    _10[0] = 0;
    _20[0] = 0;
    _50[0] = 0;
    _5[0] = 0;
}
```

- I made these variables an input for the “classifyFrame” function.

```
private Runnable periodicClassify =
    new Runnable() {
        @Override

        public void run() {
            synchronized (lock) {
                if (runClassifier) {
                    classifyFrame(_10, _20, _50, _5);
                }
            }
            backgroundHandler.post(periodicClassify);
        }
    };
```

- I modified the “classifyFrame” function to play a sound for each classification.

- To get the value of a bill I added this line:

“String money=classifier.classifyMoney(bitmap)”

- I added these conditions to avoid repetition: if(money==”<bill>” &&
<variable>[0]==0){...}

```
private void classifyFrame(int[] _10, int[] _20, int[] _50, int[] _5) {
    if (classifier == null || getActivity() == null || cameraDevice == null) {
        showToast( text: "Uninitialized Classifier or invalid context.");
        return;
    }
    Bitmap bitmap =
        textureView.getBitmap(ImageClassifier.DIM_IMG_SIZE_X, ImageClassifier.DIM_IMG_SIZE_Y);
    String textToShow = classifier.classifyFrame(bitmap);
    String money=classifier.classifyMoney(bitmap);
    bitmap.recycle();
    showToast(textToShow);
    Log.d(TAG, msg: "----ok---\n");
    MediaPlayer mediaPlayer=MediaPlayer.create(getContext(), R.raw._10);
    if (money=="10" && _10[0]==0) {
        mediaPlayer.start();
        _10[0]=1;
        _20[0]=0;
        _50[0]=0;
        _5[0]=0;
    }else if (money=="20" && _20[0]==0) {
        mediaPlayer=MediaPlayer.create(getContext(), R.raw._20);
        mediaPlayer.start();
        _10[0]=0;
        _20[0]=1;
        _50[0]=0;
        _5[0]=0;
    }else if (money=="50" && _50[0]==0) {
        mediaPlayer=MediaPlayer.create(getContext(), R.raw._50);
        mediaPlayer.start();
        _10[0]=0;
        _20[0]=0;
        _50[0]=1;
        _5[0]=0;
    }else if (money=="5" && _5[0]==0) {
        mediaPlayer=MediaPlayer.create(getContext(), R.raw._5);
        mediaPlayer.start();
        _10[0]=0;
        _20[0]=0;
        _50[0]=0;
        _5[0]=1;
    }
}
```