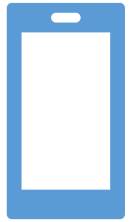
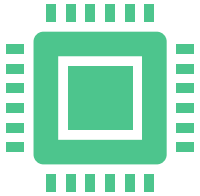
# Currency Detection

# Technology stack:



**Frontend Technology**

Java, Android

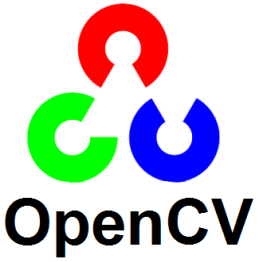


**Backend Technology**

Flask, Python


**Machine Learning Algorithm**

Image processing traditional computer vision brute force matching



# Flask Server

- Endpoint: ‘/image’
- Sending image of the currency note from android to the backend server
- Image is provided to the ML algorithm to predict
- Response of the detected currency is handled and modified through regular expressions
- Response is sent to the app by jsonifying the detected currency

```python
@app.route('/image', methods=['POST'])
def handle_request():
    files_ids = list(flask.request.files)
    image_num = 1
    file_name = ""
    for file_id in files_ids:
        imagefile = flask.request.files[file_id]
        filename = werkzeug.utils.secure_filename(imagefile.filename)
        print("Image Filename : " + imagefile.filename)
        imagefile.save(filename)
        file_name = filename
        image_num = image_num + 1
    from detect import helper
    note = helper(file_name)
    note += ".jpg"
    currency = ""
    if(re.findall(".*[2][0][0][0].*", note)):
        currency = "2000"
    elif(re.findall(".*[2][0][0][^0].*", note)):
        currency = "200"
    elif(re.findall(".*[2][0][^0].*", note)):
        currency = "20"
    elif(re.findall(".*[1][0][0][^0].*", note)):
        currency = "100"
    elif(re.findall(".*[1][0][^0].*", note)):
        currency = "10"
    elif(re.findall(".*[5][0][0].*", note)):
        currency = "500"
    elif(re.findall(".*[5][0][^0].*", note)):
        currency = "50"
    else:
        currency = "-1"

    if currency != "-1":
        return jsonify({
            "note": currency
        })
    else:
        return jsonify({
            "note": -1
        })
```

# Feature Matching Open CV

- This project uses bff.knn matcher
- It is a brute force technique for matching two images
- The bff.knn function extracts the features from image and this features are used for matching similarity between two images.

# BFF KNN Matcher

- It is a traditional Computer Vision approch that selects keypoints within an image.
- The number of keypoints you specify it select that amount.
- BFF KNN matcher works well even if two similar images have different dimensions and alignments.
- The knn algorithm finds the closest possible match.

# Android Code

Select Image from Device

```
public void selectImage(View v) {
    mVib.vibrate( milliseconds: 50);
    Intent intent = new Intent();
    intent.setType("*/*");
    intent.putExtra(Intent.EXTRA_ALLOW_MULTIPLE, value: true);
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(Intent.createChooser(intent, title: "Select Picture"), GALLERY_CODE);
}
```

OkHttpClient request to connect to server

```
OkHttpClient client = new OkHttpClient.Builder()
        .connectTimeout( timeout: 30, TimeUnit.SECONDS)
        .writeTimeout( timeout: 30, TimeUnit.SECONDS)
        .readTimeout( timeout: 30, TimeUnit.SECONDS)
        .build();
Request request = new Request.Builder()
        .url(postUrl)
        .post(postBody)
        .build();

client.newCall(request).enqueue(new Callback() {
    @Override
    public void onFailure(Call call, IOException e) {
        // Cancel the post on failure.
        call.cancel();
        // In order to access the TextView inside the UI thread, the code is executed inside runOnUiThread()
        runOnUiThread(() -> {
                Toast.makeText( context: TakePictureActivity.this,
                        text: "Failed to Connect to Server. Please Try Again", Toast.LENGTH_SHORT).show();
        });
    }
```

# Android Code

Handling the response came from the server and display it on app screen

```
@Override
public void onResponse(Call call, final Response response) throws IOException {
    // In order to access the TextView inside the UI thread, the code is executed inside
    runOnUiThread(() → {
            try {
                Intent intent = new Intent( packageContext: TakePictureActivity.this, Result
                String res = response.body().string();
                JSONObject Jobject = new JSONObject(res);
                String note = Jobject.getString( name: "note");
                intent.putExtra( name: "note_value", note);
                if (imageStoragePath != null)
                    intent.putExtra( name: "note_image", imageStoragePath);
                else
                    intent.putExtra( name: "note_image", selectedImagesPaths.get(0));
                startActivity(intent);
                finish();
            } catch (IOException e) {
                Toast.makeText( context: TakePictureActivity.this, text: "error1", Toast.LEN
                e.printStackTrace();
            } catch (JSONException e) {
                Toast.makeText( context: TakePictureActivity.this, text: "error2", Toast.LEN
                e.printStackTrace();
            }
    });
}
```

Asking for permission

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.INTERNET" />
```