MainActivity.java

```
package com.example.ece420final.businesscard;

//test
import android.Manifest;
import android.app.Activity;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.hardware.Camera;
import android.net.Uri;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.graphics.BitmapFactory;
import android.widget.Toast;
import android.graphics.Matrix;

import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.io.File;
import android.graphics.Bitmap;

import com.theartofdev.edmodo.cropper.CropImage;
import com.theartofdev.edmodo.cropper.CropImageView;

import org.opencv.android.OpenCVLoader;

public class MainActivity extends AppCompatActivity {
    private static final int REQUEST_CODE = 1;
    private static final String TAG = "MainActivity";
    private Button detectButton;
    private Camera camera;
    private File imgFile;
    private String imgFilePath;
    private String imgFilePathDetect;
    private ImageView imageView;
    private Button cropping;
    private Uri resultUri;
    private Bitmap myBitmap;
    private Bitmap rotatedBitmap;

    static {
        if(!OpenCVLoader.initDebug()){
            Log.d(TAG, "OpenCV not loaded");
        } else {
            Log.d(TAG, "OpenCV loaded");
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //check permissions
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA)
                == PackageManager.PERMISSION_DENIED){
            ActivityCompat.requestPermissions(this, new String[] {Manifest.permission.CAMERA}, 1);
        }

        if (ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE_EXTERNAL_STORAGE)
                == PackageManager.PERMISSION_DENIED){
            ActivityCompat.requestPermissions(this, new String[] {Manifest.permission.WRITE_EXTERNAL_STORAGE}, 1);
        }

        imageView =(ImageView) findViewById(R.id.imageView);
        imageView.setScaleType(ImageView.ScaleType.FIT_CENTER);
        imageView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
                imgFile = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES),timeStamp+".jpg");
                Uri imgUri = Uri.fromFile(imgFile);

                cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT,imgUri);
                cameraIntent.putExtra(MediaStore.EXTRA_VIDEO_QUALITY,1);

                if (cameraIntent.resolveActivity(getPackageManager()) != null){
                    startActivityForResult(cameraIntent,REQUEST_CODE);
```

```
                    if(camera != null){
                        camera.release();
                        camera = null;
                    }
                }

            }
        });


        detectButton = (Button)findViewById(R.id.buttonRecognition);
        detectButton.setText("Find Text");
        detectButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //pass in String
                if(imgFilePathDetect != null){
                    Intent detectIntent = new Intent(getCurrentActivity(),DetectionRecognitionActivity.class);
                    detectIntent.putExtra("imgFilePathDetect",imgFilePathDetect);
                    if(resultUri != null) {detectIntent.putExtra("CroppedUri",resultUri.toString());}
                    startActivity(detectIntent);
                }
            }
        });

        cropping = (Button)findViewById(R.id.buttonCrop);
        cropping.setText("CROP THE CARD");
        cropping.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //Log.d(TAG,"STARTING CROPPING");
                if(imgFilePathDetect != null){
                    //Log.d(TAG,"uri "+imgFilePathDetect);
                    CropImage.activity(Uri.parse("file://"+imgFilePathDetect))
                            .setGuidelines(CropImageView.Guidelines.ON)
                            .start(getCurrentActivity());
                }
            }
        });

    }

    private Activity getCurrentActivity(){
        return MainActivity.this;
    }

    @Override
    public void onResume(){
        super.onResume();
        Log.d(TAG,"Resuming");

        if(imgFilePath != null){
            //Log.d(TAG,imgFilePath);
            myBitmap = BitmapFactory.decodeFile(imgFilePath);
            Matrix matrix = new Matrix();
            matrix.postRotate(90);
            rotatedBitmap = Bitmap.createBitmap(myBitmap, 0, 0, myBitmap.getWidth(), myBitmap.getHeight(), matrix, true);
            imageView.setImageBitmap(rotatedBitmap);

        }

        if(resultUri != null && imgFilePath == null){
            //Log.d(TAG,resultUri.toString());
            try{
                myBitmap = MediaStore.Images.Media.getBitmap(this.getContentResolver(),resultUri);
                imageView.setImageBitmap(myBitmap);


            } catch(IOException e){
                Log.d(TAG,e.getMessage());
            }
        }
    }

    @Override
    protected void onPause() {
        Log.w(TAG, "App paused");
        super.onPause();
        imgFilePath = null;
        resultUri = null;
    }

    @Override
    protected void onDestroy() {
        Log.w(TAG, "App destroyed");
        super.onDestroy();
        if(rotatedBitmap != null) {rotatedBitmap.recycle();}
        if(myBitmap != null) {myBitmap.recycle();}
    }

    protected void onActivityResult(int request,int result,Intent data){
```

```
        if(request == REQUEST_CODE){
            switch (result){
                case Activity.RESULT_OK:
                    if(imgFile.exists()){
                        Toast.makeText(this, "The file was saved at"+imgFile.getAbsolutePath(), Toast.LENGTH_LONG).show();
                        imgFilePath = imgFile.getAbsolutePath();
                        imgFilePathDetect = imgFile.getAbsolutePath();
                    }
                    else{
                        Toast.makeText(this, "Error occurred", Toast.LENGTH_LONG).show();
                    }
                    break;

                case Activity.RESULT_CANCELED:
                    break;
                default:
                    break;
            }
        }

        if (request == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE) {
            CropImage.ActivityResult output = CropImage.getActivityResult(data);
            //Log.d(TAG,"IT MATCHES THE REQUEST CODE");
            if (result == RESULT_OK) {
                //Log.d(TAG,"IT IS OK");
                resultUri = output.getUri();
                Log.d(TAG,resultUri.toString());
            } else if (result == CropImage.CROP_IMAGE_ACTIVITY_RESULT_ERROR_CODE) {
                Exception error = output.getError();
                Log.d(TAG,error.getMessage());
            }
        }
    }
}
```