Project Overview

Have you ever been curious about what you're looking at?

And then you ask yourself:

What breed of cat is it, and what is the name of the fruit?

No worries,

Because we have the answers

With our App ImageR, you can classify objects on the move using the camera on your Android phone.

The App uses Machine Learning to classify the data set and label them under different categories.

And provides you with the solutions instantly.

ImageR

Technology Used

Data Gathering

Acquiring all possible images to feed into the dataset

Teachable ML

The Data Gathered was utilised to train a machine learning model using teachable ml, and the model was then exported as a tensorflow file and applied in the app.

Tensor Flow Lite

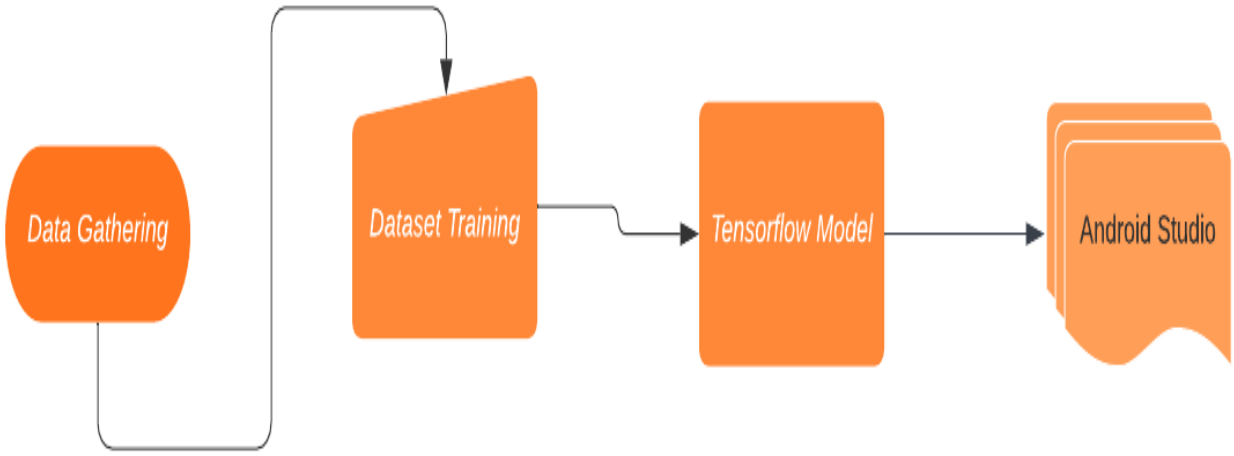
An image classification model is trained to recognize various classes of images. For example, you may train a model to recognize photos representing three different types of animals: rabbits, hamsters, and dogs. TensorFlow Lite provides optimized pre-trained models that you can deploy in your mobile applications

Android Studio (Kotlin)

Android Studio was used to build the app using the tensorflow model and label the data set accordingly.

Project Functionality

- Data Gathering.
- The Dataset used is in the form a tensorflow model.
- The Dataset is then labeled.
- The model it then imported in android studio.



Code

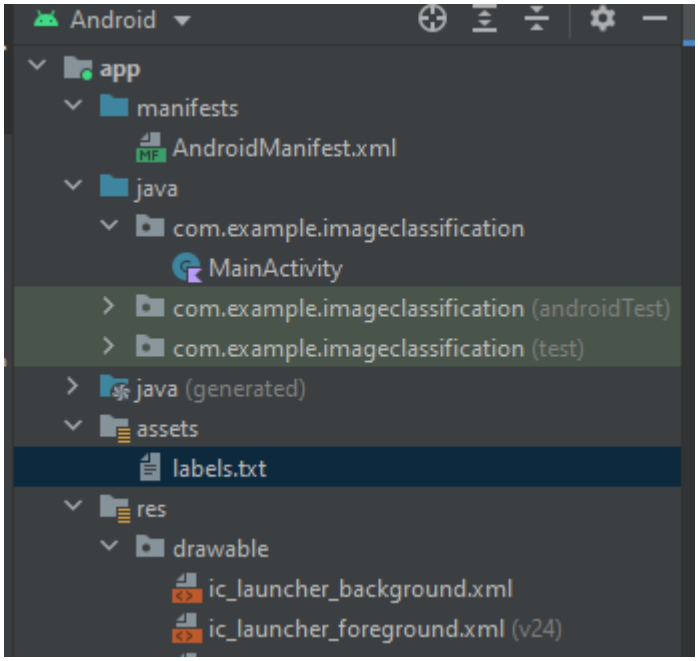
Code for Camera Permission

```
public fun checkandGetpermissions(){
    if(checkSelfPermission(android.Manifest.permission.CAMERA) == PackageManager.PERMISSION_DENIED){
        requestPermissions(arrayOf(android.Manifest.permission.CAMERA), requestCode: 100)
    }
    else{
        Toast.makeText( context: this, text: "Camera permission granted", Toast.LENGTH_SHORT).show()
    }
}

override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<out String>,
    grantResults: IntArray
) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    if(requestCode == 100){
        if(grantResults[0] == PackageManager.PERMISSION_GRANTED)
        {
            Toast.makeText( context: this, text: "Camera permission granted", Toast.LENGTH_SHORT).show()
        }
        else{
            Toast.makeText( context: this, text: "Permission Denied", Toast.LENGTH_SHORT).show()
        }
    }
}
```

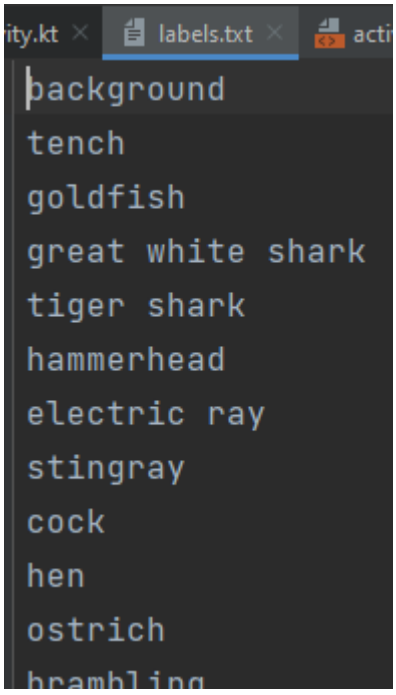
Code for acquiring the labels from a text file and then matching it up with the trained dataset.

```
    val labels = application.assets.open( fileName: "labels.txt").bufferedReader().use { it.readText() }.split( ...delimiters: "\n")

    select_image_button.setOnClickListener(View.OnClickListener {  it: View!
        Log.d( tag: "mssg",  msg: "button pressed")
        var intent : Intent = Intent(Intent.ACTION_GET_CONTENT)
        intent.type = "image/*"

        startActivityForResult(intent,  requestCode: 250)
    })

    make_prediction.setOnClickListener(View.OnClickListener {  it: View!
        var resized = Bitmap.createScaledBitmap(bitmap,  dstWidth: 224,  dstHeight: 224,  filter: true)
        val model = MobilenetV110224Quant.newInstance( context: this)

        var tbuffer = TensorImage.fromBitmap(resized)
        var byteBuffer = tbuffer.buffer

' Creates inputs for reference.
        val inputFeature0 = TensorBuffer.createFixedSize(intArrayOf(1, 224, 224, 3), DataType.UINT8)
        inputFeature0.loadBuffer(byteBuffer)
```

Resource Folders



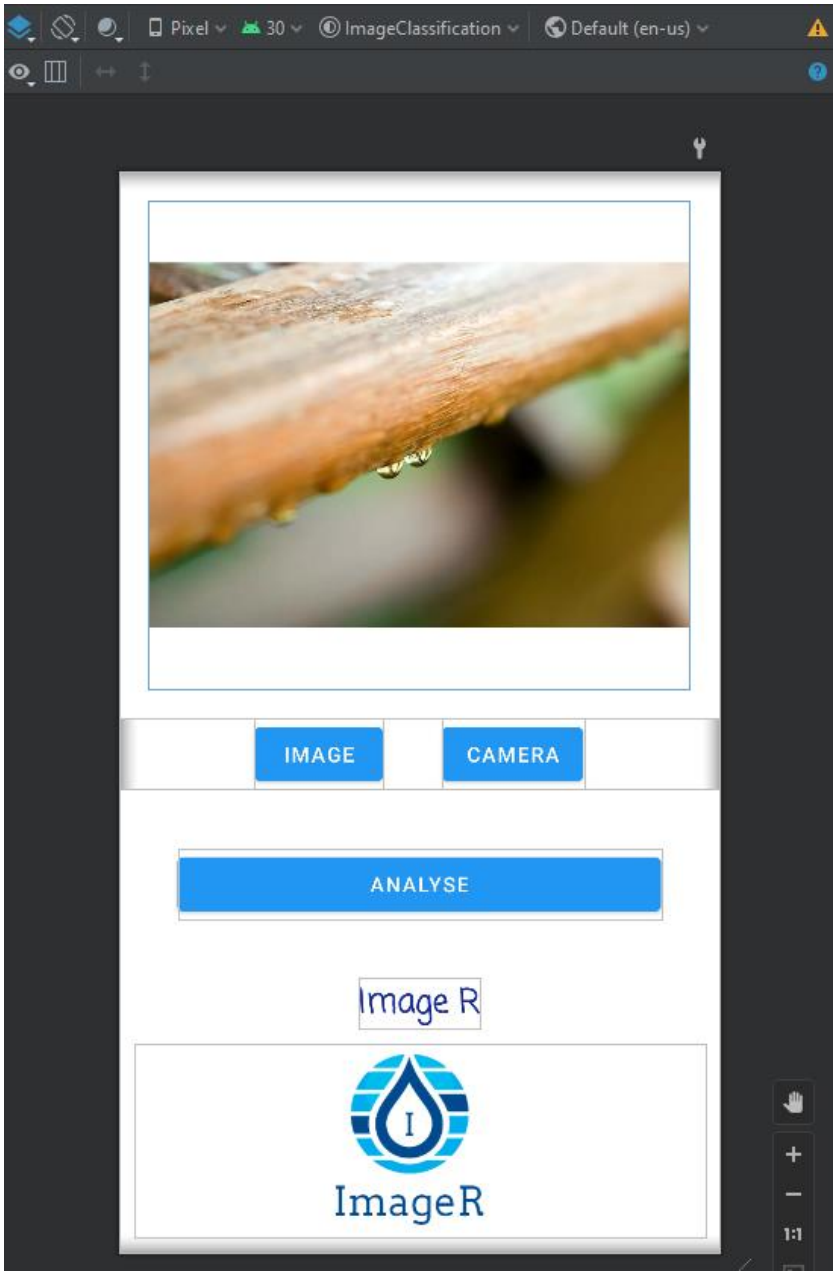
Labels File(txt)



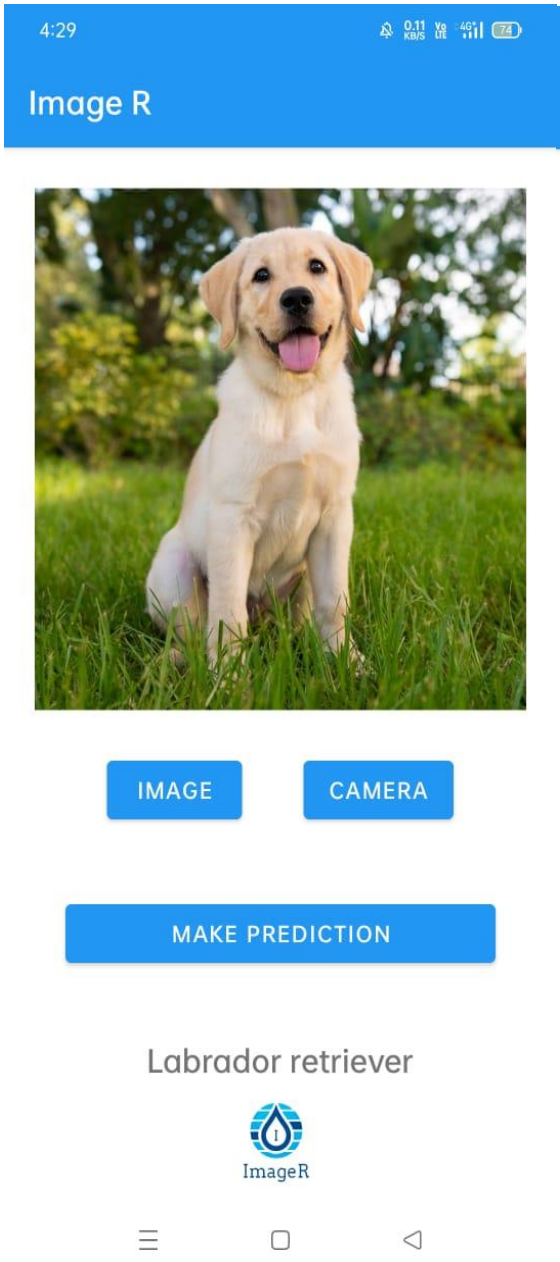
UI Code

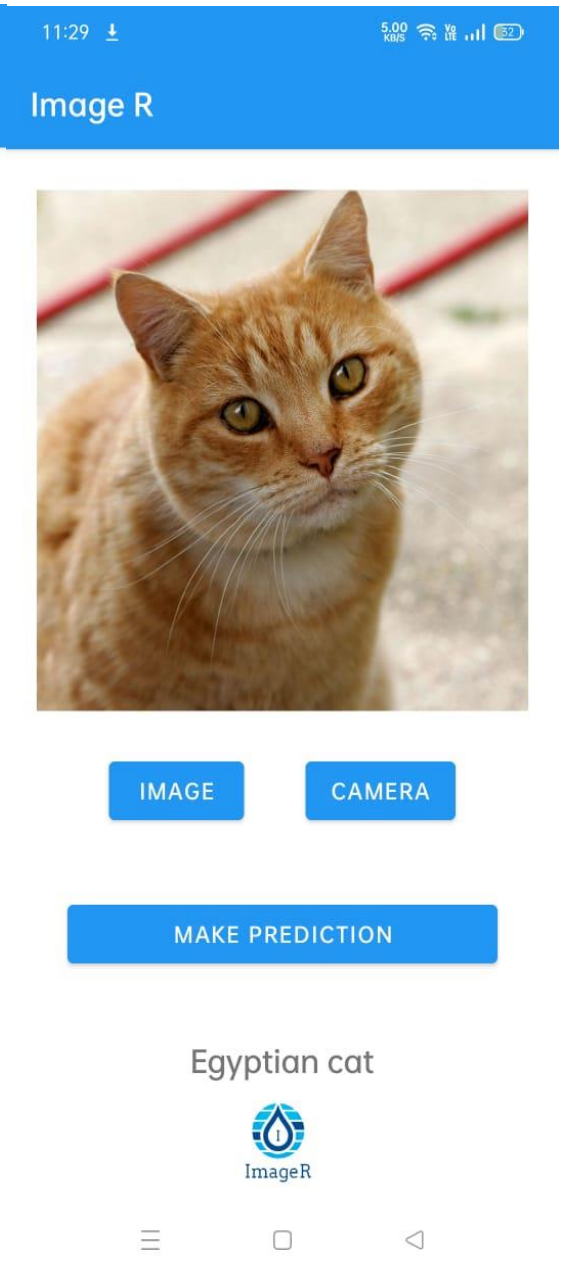
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="match_parent"
        android:layout_height="330dp"
        android:layout_margin="20dp"
        tools:srcCompat="@tools:sample/backgrounds/scenic" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
```

App Ui

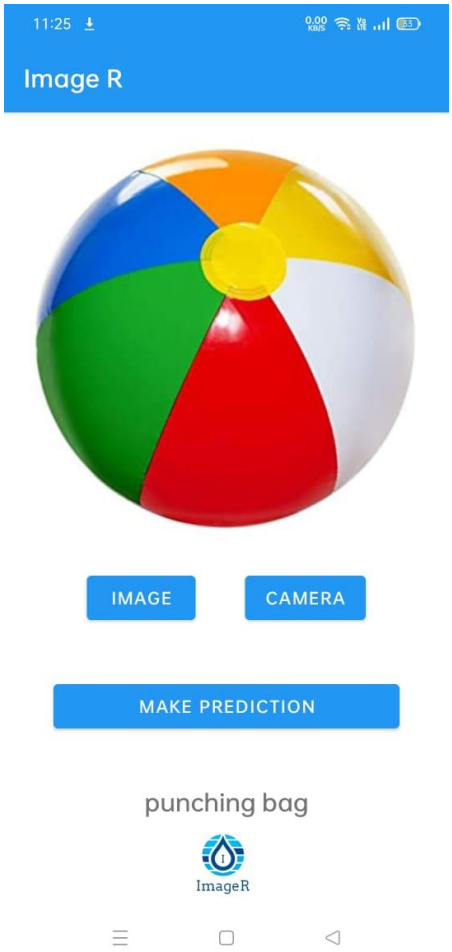

Pixel
30
ImageClassification
Default (en-us)
IMAGE
CAMERA
ANALYSE
Image R
ImageR
1:1

Results


4:29
Image R
IMAGE
CAMERA
MAKE PREDICTION
Labrador retriever
ImageR


11:29
Image R
IMAGE
CAMERA
MAKE PREDICTION
Egyptian cat
ImageR

Inaccuracy



Limitations

- Error Rate can vary as no model can get close to perfection
- Difficulty in classifying low pixel or low poly image
- The image classification can go as far as the limit of the dataset.

Future Scope

In Addition to image classification, we can also have an option for audio classification.

Video classification can also be implemented in the future using hardware accelerated machine learning.

References

https://developer.android.com/docs

https://www.tensorflow.org/lite/android/play_services

https://www.termsfeed.com/blog/android-sensitive-data-collection/

https://github.com/tensorflow/tensorflow