Getting Started with

ML Kit

SIERRA OBRYAN (SHE/HER)

@_sierraOBryan

What is it??

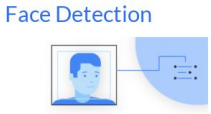
“ML Kit brings Google’s machine learning expertise to mobile developers in a powerful and easy-to-use package.”

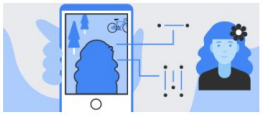
Built By Google

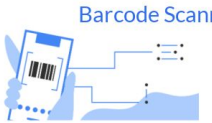
Optimized for Mobile

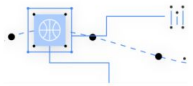
Easy to Use

Okay.. But what can it do?

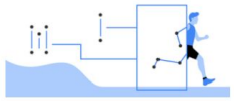


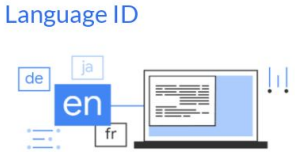


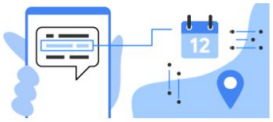




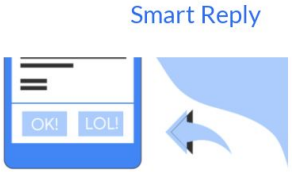
Vision APIs







Natural Language APIs



Who can use it??

Everyone!

ML Kit is available for both Android and iOS!

Meet our Client: Steve

Steve is my Dad.

He really likes gardening.

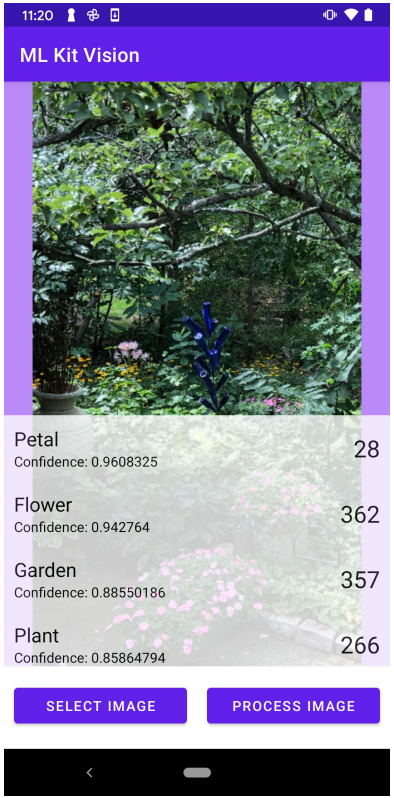
He plants a lot of things and can’t always remember what everything is.

He finally knows what I do for work.

He ask if I could write an app to help.

With ML Kit the answer is YES

Let’s get our app started!



How do we use it?

We need to add the Image Labeling dependency to our gradle

```
dependencies {
  // ...
  // Use this dependency to bundle the model with your app
  implementation 'com.google.mlkit:image-labeling:17.0.2'
}
```

Since we're building this for Steve to use in the yard, I'm going to bundle it with the app

```
dependencies {
  // ...
  // Use this dependency to use dynamically downloaded model in Google Play Service
  implementation 'com.google.android.gms:play-services-mlkit-image-labeling:16.0.2'
}
```

Clipper	Bonfire	Tuxedo	Beach
Nail	Comics	Mouth	Rainbow
Cola	Himalayan	Desert	Branch
Cutlery	Iceberg	Dinosaur	Moustache
Menu	Bento	Mufti	Garden
Sari	Sink	Fire	Gown
Plush	Toy	Bedroom	Field
Pocket	Statue	Goggles	Dog
Neon	Cheeseburger	Dragon	Superhero
Icicle	Tractor	Couch	Flower
Pasteles	Sled	Sledding	Placemat
Chain	Aquarium	Cap	Subwoofer
Dance	Circus	Whiteboard	Cathedral
Dune	Sitting	Hat	Building
Santa claus	Beard	Gelato	Airplane
Thanksgiving	Bridge	Cavalier	Fur
Tuxedo	Tights	Beanie	Bull
Mouth	Bird	Jersey	Bench
Desert	Rafting	Scarf	Temple
Dinosaur	Park	Vacation	Butterfly
Mufti	Factory	Pitch	Model
	Graduation	Blackboard	

We’re just going to use the Base Model

Step One
Prepare the Input Image

First, we need to let Steve pick an Image from his photos.

Then, we need to take that URI and transform it into a Bitmap.

```
private fun startChooseImageIntentForResult() {
    val intent = Intent()
    intent.type = "image/*"
    intent.action = Intent.ACTION_GET_CONTENT
    startActivityForResult(
        Intent.createChooser(intent, "Select
Picture"),
        REQUEST_CHOOSE_IMAGE
    )
}

override fun onActivityResult(
    requestCode: Int,
    resultCode: Int,
    data: Intent?
) {
    onSelectImageResult(data?.data != null)
    if (requestCode == REQUEST_CHOOSE_IMAGE &&
        resultCode == Activity.RESULT_OK
    ) {
        val imageUri = data!!.data
        setPreview(imageUri)
    } else {
        super.onActivityResult(
            requestCode, resultCode, data
        )
    }
}
```

Step One
Prepare the Input Image

First, we need to let Steve pick an Image from his photos

Then, we need to take that URI and transform it into a Bitmap

```
private fun setPreview(imageUri: Uri?) {
    try {
        if (imageUri == null) return

        val preview = findViewById<ImageView>(R.id.preview)

        val imageBitmap = getBitmapFromUri(imageUri) ?: return

        this.imageBitmap = imageBitmap

        preview.setImageBitmap(imageBitmap)

    } catch (e: IOException) {
        Toast.makeText(this,
            getString(R.string.something_went_wrong),
            Toast.LENGTH_SHORT
        ).show()
    }
}

@Throws(IOException::class)
private fun getBitmapFromUri(uri: Uri): Bitmap? {
    val parcelFileDescriptor =
        contentResolver.openFileDescriptor(uri, "r")
    val fileDescriptor = parcelFileDescriptor?.fileDescriptor
    val image = BitmapFactory
       .decodeFileDescriptor(fileDescriptor)
    parcelFileDescriptor?.close()
    return image
}
```

Process the image

Now that we have our bitmap, we can convert that to an imageInput.

Then we can create our labeler. In this case, we're just using the default options.

```
if (imageBitmap != null) {
    val imageInput = InputImage.fromBitmap(imageBitmap!!, 0)

    val labeler = ImageLabeling.getClient(ImageLabelerOptions.DEFAULT_OPTIONS)

    labeler.process(imageInput).addOnSuccessListener { labels ->
```

Finally, we can can process our imageInput with our labeler

```
        // do something with our labels

    }.addOnFailureListener {
        Toast.makeText(this, getString(R.string.nothing_found), Toast.LENGTH_SHORT).show()
    }
}
```

We add an onSuccessListener for when it works YAY!

And a onFailureListener for when it doesn't):

Send our labels to our view

```
if (imageBitmap != null) {
    val imageInput = InputImage.fromBitmap(imageBitmap!!, 0)

    val labeler = ImageLabeling.getClient(ImageLabelerOptions.DEFAULT_OPTIONS)

    labeler.process(imageInput).addOnSuccessListener { labels ->

        val recyclerView = findViewById<RecyclerView>(R.id.labels)
        recyclerView.layoutManager = LinearLayoutManager(this)
        recyclerView.adapter = LabelAdapter(labels)
        recyclerView.visibility = View.VISIBLE

    }.addOnFailureListener {
        Toast.makeText(this, getString(R.string.nothing_found), Toast.LENGTH_SHORT).show()
    }
}
```

When we successfully process our image, we get back a list of labels

For our simple app, we're going to display our list of labels in a recyclerView so we pass them into an adapter

Display the Labels

Each ImageLabel
(in our list of ImageLabels)
has a

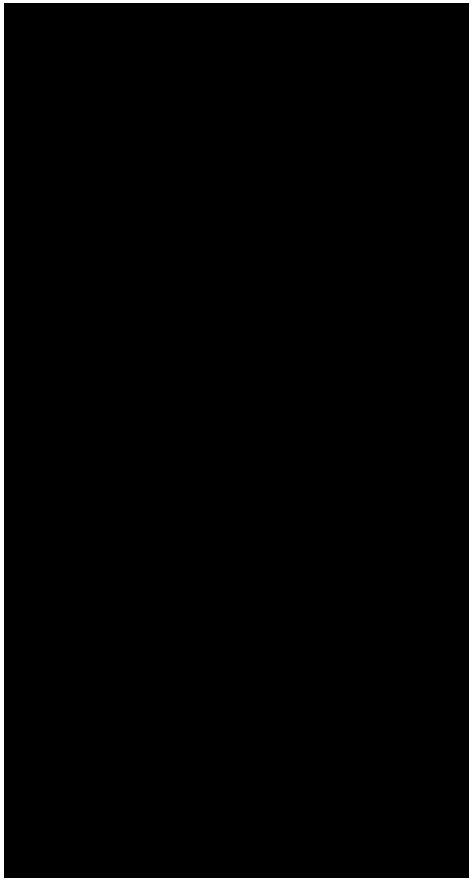
Text (String)
Confidence (Float)
Index (Integer)

```
fun bind(imageLabel: ImageLabel) {
    label.text = imageLabel.text
    confidence.text = String.format(
        itemView.resources.getString(R.string.confidence_format),
        imageLabel.confidence.toString()
    )
    index.text = imageLabel.index.toString()
}
```

In our simple example, we bind those to our row view and we're good to go!

And with that we have an app!

Let’s try it out!



Where to go from here

Maybe make a custom data model to label plants

Maybe use the camera so that I don’t have to load an image

Maybe overlay the image with the labels

Rewrite in Jetpack Compose!

Convince my dad to use an Android Phone so he can use it):

With great power comes great responsibility

Thank you!

Where can you find this code?

https://github.com/sierraobryan/examples/tree/main/MLKitVision

Where can you find me?

@_sierraOBryan

https://sierraobryan.com/

SIERRA OBRYAN