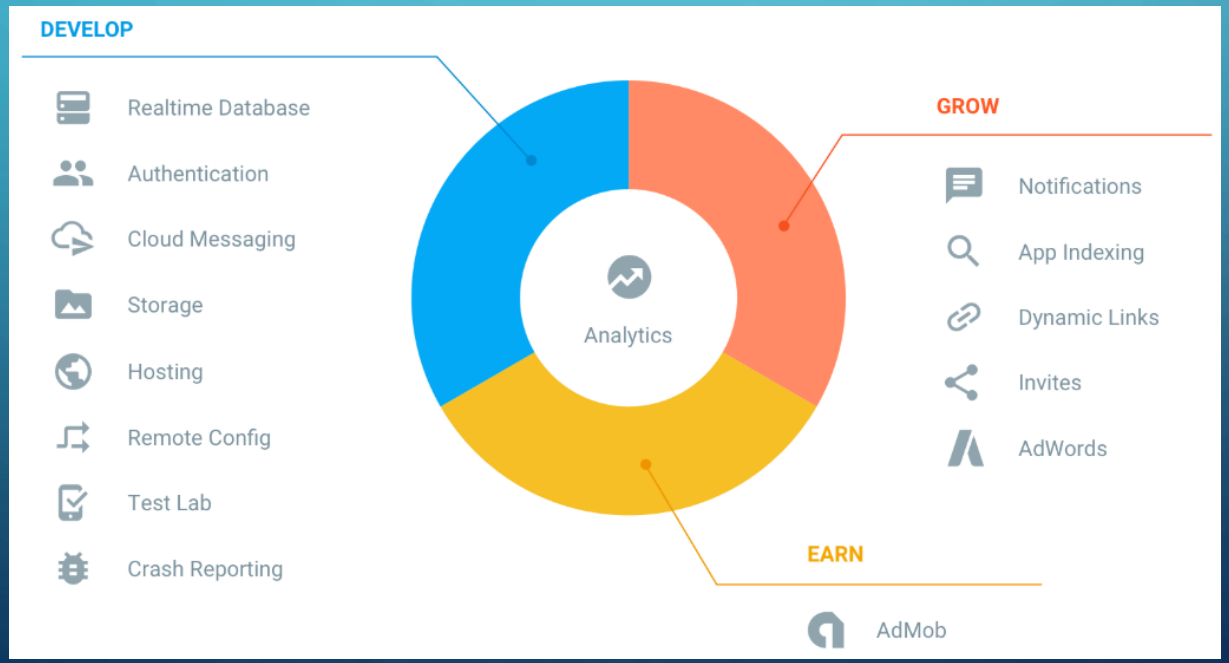
LAB4:
CREATE AI ON YOUR MOBLIE DEVICE

FIREBASE

- Firebase 是一個同時支援 Android、iOS 及網頁的 app 雲端開發平台，協助 app 開發者在雲端快速建置後端服務，提供即時資料庫，有效縮短 app開發時間，並幫助開發者更專注在前端的優化
- 2014年10月Google收購了Firebase，並擴大了Firebase的應用。



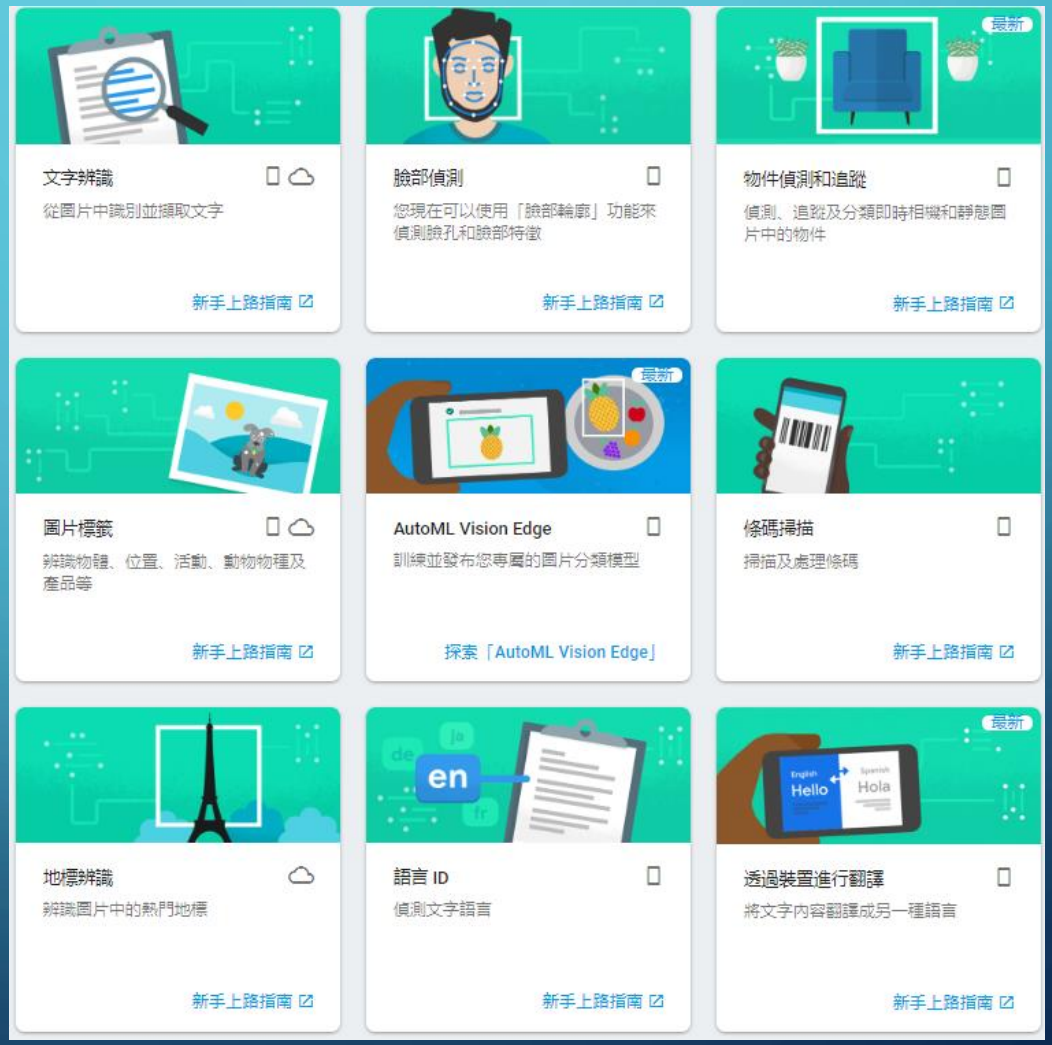
FIREBASE MACHINE LEARNING KIT

- 在 2018 年的 Google I/O 大會上，Google 宣布 Firebase 釋出了最新功能 Firebase ML Kit，一個專為行動裝置開發的 AI SDK。

- Firebase ML Kit 是一個能簡易把 Google Machine Learning 應用到一般行動 App 產品中的封裝（iOS + Android）。無論是否對於 Machine Learning 是否有經驗，都能夠透過簡單的步驟，把強大的功能加入您的產品中。

ML KIT 中與影像處理的 VISION API

- Text Recognition
- Face Detection
- Barcode Scanning
- Image Labeling
- Landmark Recognition



ML KIT 提供了通用的介面給CLIENT/CLOUD

- CLIENT端的 API 能夠快速的在本地處理數據。
- CLOUD端的 API 則使用 Google Cloud Platform 機器學習技術，提供更細緻、更高準確度的資料(但須要使用付費方案)。

Feature	On-device	Cloud
Text recognition	✓	✓
Face detection	✓	
Barcode scanning	✓	
Image labeling	✓	✓
Landmark recognition		✓
Custom model inference	✓	

利用ML KIT部屬自定義的模型

- 支援客製化的 TensorFlow Lite 模型。
- 將自己的模型上傳到 Firebase，Firebase 便會託管該模型，ML Kit 會提供您自定義模型的 API 層，使用上非常方便。

- 因為時間有限，今天用的模型是Firebase官方內建的，不會教到如何訓練自己的TF Lite模型並放到手機上，有興趣放自己模型的同學可以參考以下資源：
  - https://firebase.google.com/docs/ml-kit/android/use-custom-models
  - https://firebase.google.com/docs/ml-kit/android/use-custom-tflite

PART1: ANDROID CAMERA-KIT

- 負責對手機上的鏡頭所捕捉到的影像作傳輸媒介
- 非官方原生的package，但支援gradle script，使用起來非常方便

Get The Most From CameraKit

There are currently two versions of CameraKit that we support, `v1.0.0-beta3.X` and `v0.13.X`.

If photo is your only need, try out the latest and greatest CameraKit features with `v1.0.0-beta3.11`. Our `beta3.11` release does not yet support video, but that feature is coming!

In the meantime, if your application requires video we recommend sticking with `v0.13.4`; the latest stable release with video implementation.

Use Case	Version	Notes	Documentation Link
Photo only	`v1.0.0-beta3.11`	The latest and greatest CameraKit has to offer. Video support coming soon!	camerakit.io/docs/beta3.11
Photo and Video	`v0.13.4`	Stable build with full photo and video support	camerakit.io/docs/0.13.4

Reference: https://github.com/CameraKit/camerakit-android

開始建立專案

- 1. 建立Android Studio的empty project
- 2. 在build.gradle(Module: app)中implement CameraKit:

```
dependencies {
    …….
    implementation 'com.camerakit:camerakit:1.0.0-beta3.11'
    implementation 'com.camerakit:jpegkit:0.1.0'
    implementation 'org.jetbrains.kotlin:kotlin-stdlib-jdk7:1.3.0'
    implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.0.0‘
    …..
}
```

若你的Android版本是8.0以上
這行請把版本改成3.10

將CAMERAKITVIEW放到LAYOUT之中顯示

- 建議先將原生的Constraint Layout改成Relative Layout(右鍵->Convert view)
- 將以下程式碼加到activity_main.xml

```
<com.camerakit.CameraKitView
    android:id="@+id/camera"
    android:layout_width="match_parent"
    android:layout_height="250dp"
    android:layout_marginTop="15dp"
    android:adjustViewBounds="true"
    android:keepScreenOn="true"
    android:scaleType="centerInside"
    app:camera_facing="back"
    app:camera_focus="continuous"
    app:camera_permissions="camera"/>
```

- 鏡頭所捕捉的畫面會在你所拉的CameraKitView裡面顯示

在ACTIVITY宣告CAMERAKITVIEW

- 1.先在MainActivity宣告變數

```
private CameraKitView cameraKitView;
```

- 2.在onCreate中把連結剛剛所拉好的Layout

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    cameraKitView = findViewById(R.id.camera);
```

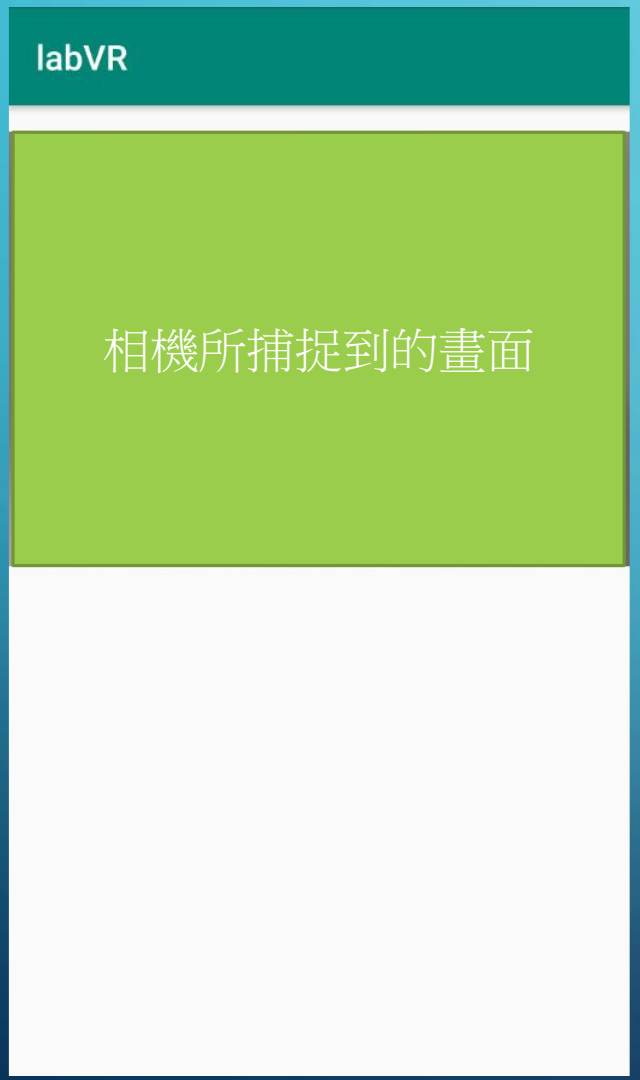
- 3.在Activity加入APP每個生命週期CAMERAKITVIEW所對應的動作

這些週期包括: onStart(), onResume, onPause, onStop (見下頁)

```
@Override
  protected void onStart() {
      super.onStart();
      cameraKitView.onStart();
  }
  @Override
  protected void onResume() {
      super.onResume();
      cameraKitView.onResume();
  }
  @Override
  protected void onPause() {
      cameraKitView.onPause();
      super.onPause();
  }
  @Override
  protected void onStop() {
      cameraKitView.onStop();
      super.onStop();
  }
@Override
  public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
      super.onRequestPermissionsResult(requestCode, permissions, grantResults);
      cameraKitView.onRequestPermissionsResult(requestCode, permissions, grantResults);
  }
```

每個生命週期在做什麼LAB1的講義都有寫，若忘記的同學建議可以找出來複習。

SELF CHECKPOINT1：CAMERAKITVIEW是否顯示

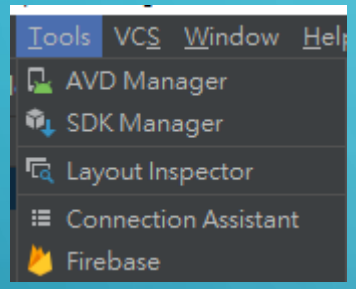


寫到這邊先Run APP, 確認是否沒有Bug且成功顯示畫面（不需要DEMO)

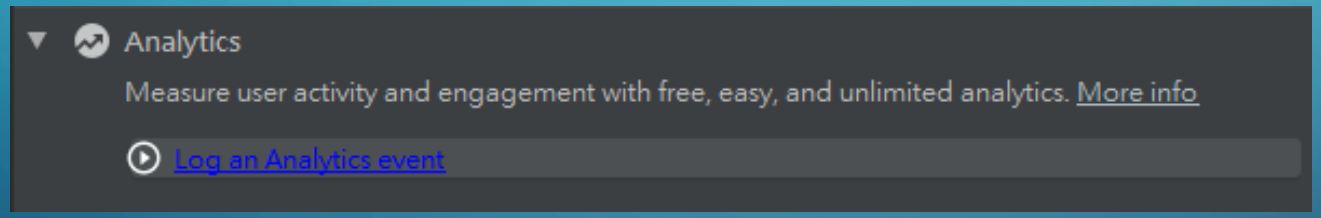
NOTICE:記得開APP的相機權限

PART2: IMPLEMENTATION OF FIREBASE

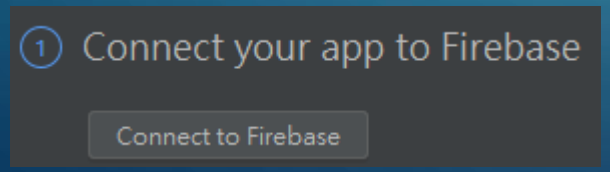
- 1.Tools -> Firebase



- 2.右側的Firebase中，點開Analytics的標籤 -> Log an Analytics events

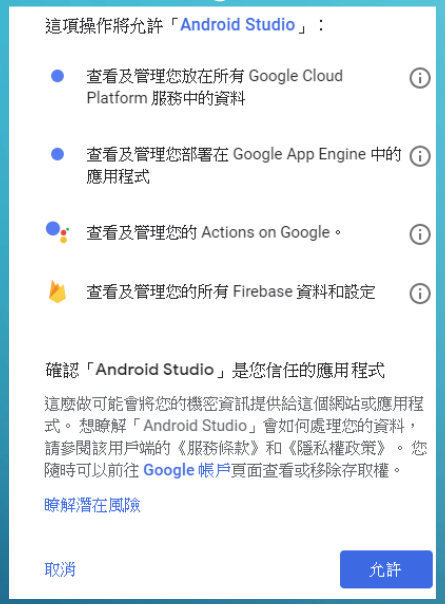


- 3.Connect to Firebase，開始連結專案



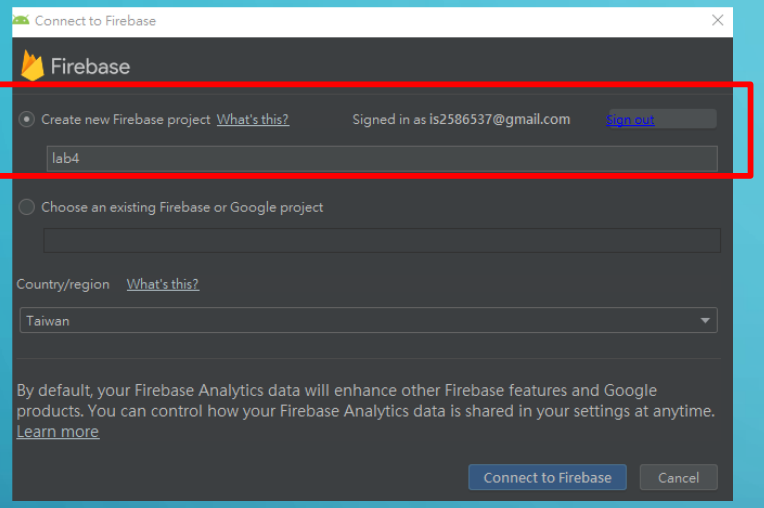
開始使用GOOGLE FIREBASE ML KIT

- 用Google帳號登入，允許權限。

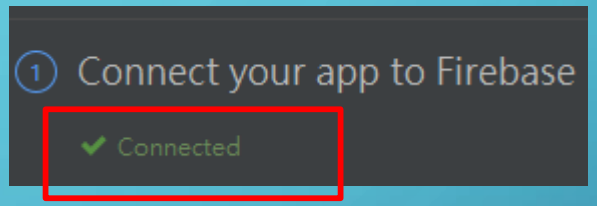


- 回到Android Studio繼續設定

繼續新增FIREBASE專案



確認Firebase連接(有打綠色勾勾)



之後不需要繼續照著他的步驟來Add Analytics(會引入較舊版本)，
這邊我們照官方最新文件引入（下頁接續）。

ADD RELATED GRADLE SCRIPT

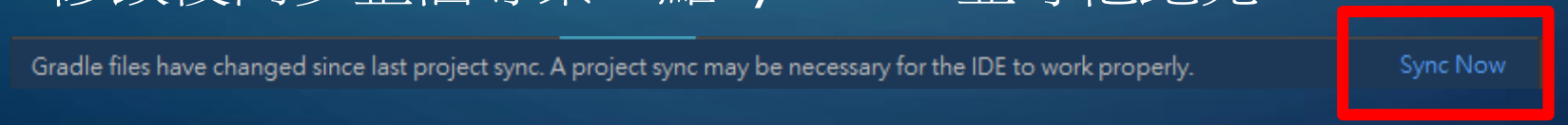
- 在Build.gradle(Module:app)引入ML kit libraries

```
dependencies {
    .....
    implementation 'com.google.firebase:firebase-ml-vision:20.0.0'
    implementation 'com.google.firebase:firebase-core:16.0.9'
    .....
}
apply plugin: 'com.google.gms.google-services'
```

- 在build.gradle(Project: xxx)引入gms4.2.0

```
dependencies {
        classpath 'com.android.tools.build:gradle:3.4.1'
        classpath 'com.google.gms:google-services:4.2.0'
        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
```

- 修改後同步整個專案，點Sync Now並等他跑完



給予APP權限

- 因為我們這次LAB要寫ON-DEVICE的API，需要給予APP自動下載Firebase ML model的權限，在增加AndroidManifest.xml新增一筆meta-data:

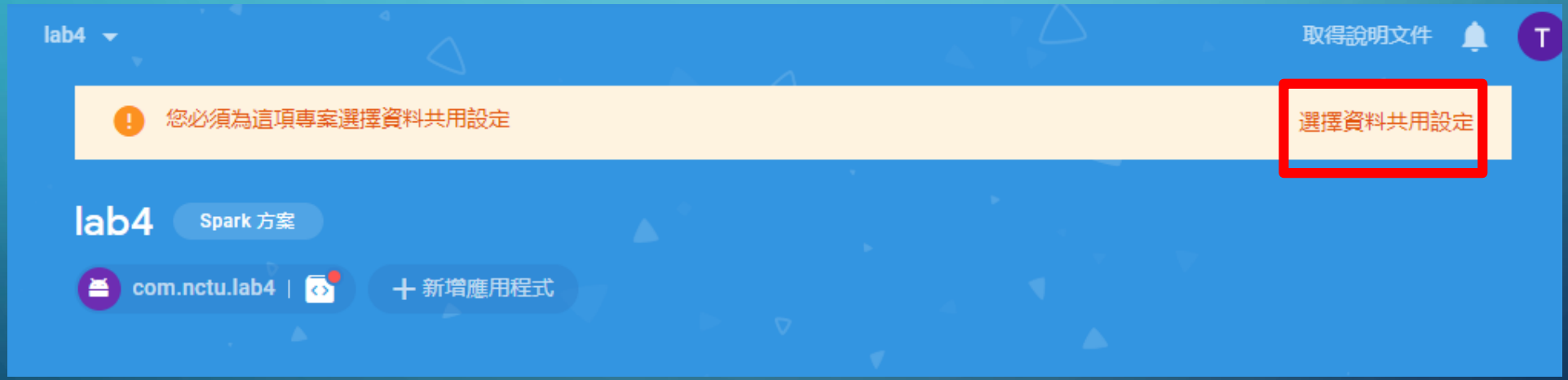
```
<application ...>
  ...
  <meta-data
      android:name="com.google.firebase.ml.vision.DEPENDENCIES"
      android:value="ocr" />
</application>
```

- 這樣就引入完成了，為了驗證，我們必須回到Firebase後端設定與連線

官方文件: https://firebase.google.com/docs/ml-kit/android/recognize-text

設定FIREBASE後端

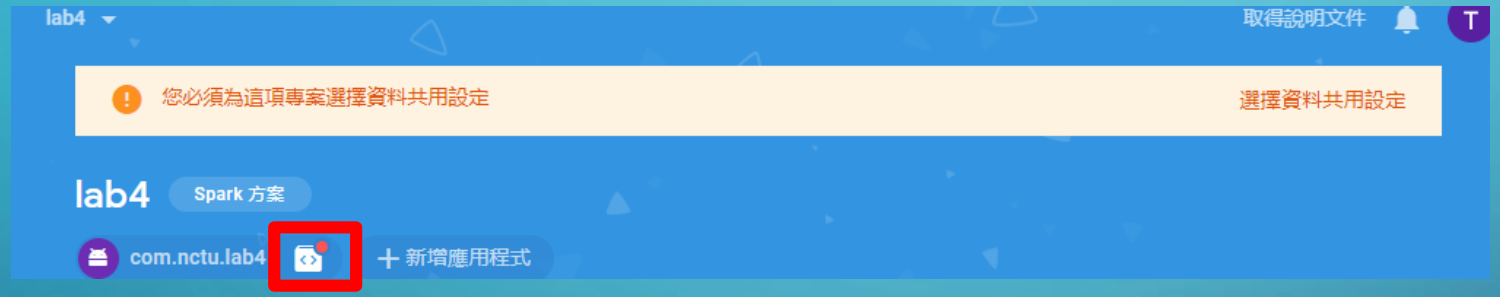
- 回到Firebase後端並登入:https://console.firebase.google.com
- 點擊你剛剛創立的專案，進入後選擇資料共用設定，兩個都打勾即可。



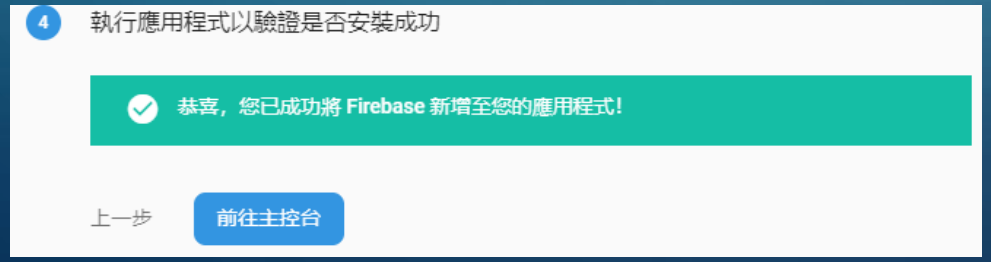
- 這樣就成功讓專案可以獲得後端資料的權限了，再回到ANDROID STUDIO

APPENDIX:驗證APP是否成功安裝FIREBASE

- 回到Firebase後端並登入:https://console.firebase.google.com
- 點擊並設定SDK



- 2.的下載設定檔可以跳過，3的部分講義前面已經新增過了，所以直接點擊4.執行應用程式以驗證是否安裝成功，
- 在Firebase後端等待連線之後run APP，再回到網頁查看是否驗證成功



LAYOUT設定

- 在我們的文字識別的APP中，除了CameraKitView之外還需要兩個元件:
  - 執行拍照->文字辨識的Button
  - 印出文字辨識結果的TextView
- 首先在原本的Relative Layout裡面放一個TextView在CameraKitView下方

```
<TextView
        android:layout_width="match_parent"
        android:layout_height="320dp"
        android:layout_marginTop="300dp"
        android:id="@+id/result" />
```

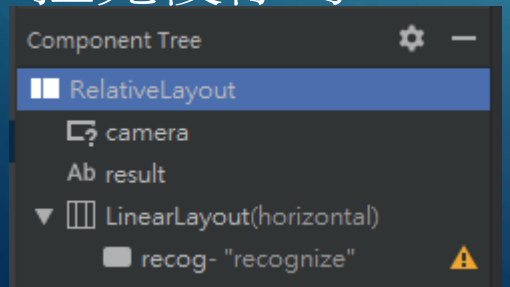
LAYOUT設定

- 然後在Relative Layout裡面拉一個Linear Layout(horizontal)，並放入Button:

```
<LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_alignParentBottom="true">

        <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="0.33"
        android:id="@+id/recog"
        android:text="recognize"/>
    </LinearLayout>
```

- 拉完後你的LAYOUT的Component Tree應該長這樣:



INITIALIZE MAINACTIVITY

```
public class MainActivity extends AppCompatActivity {
    private CameraKitView cameraKitView;
    private FirebaseAnalytics mFirebaseAnalytics;
    Button btn_REC;
    TextView tv_result;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mFirebaseAnalytics = FirebaseAnalytics.getInstance(this);
        btn_REC = findViewById(R.id.recog);
        tv_result = findViewById(R.id.result);
        cameraKitView = findViewById(R.id.camera);
        runTextRecognition();
    }
    …
    private void runTextRecognition(){
      //先留空 待會會教
    }
```

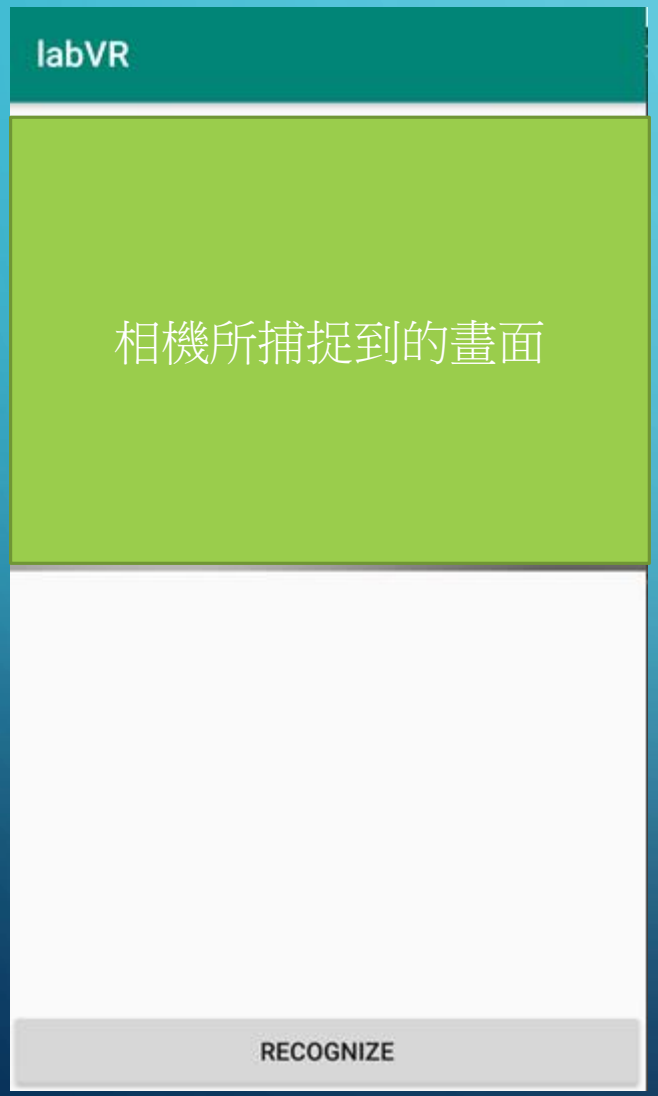
初始化Firebase

待會要用到的function

新增按下按鈕與相機抓到圖片的LISTENER

```
private void runTextRecognition(){
    btn_REC.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View v) {
            Log.d("notice", "Shot");
            cameraKitView.captureImage(new CameraKitView.ImageCallback(){
                @Override
                public void onImage(final CameraKitView cameraKitView, final byte[] capturedImage) {
                  Bitmap bitmap = BitmapFactory.decodeByteArray(capturedImage, 0, capturedImage.length);
                //這邊待會要引入Firebase的model，先留空
                }
            });
        }
    });
}
```

SELF CHECKPOINT2: APP是否成功連接FIREBASE



這邊需要確認:
1.鏡頭依然正常運作
2.按鈕位置正確，且按下之後console會印出‘shot’
3.FIREBASE連接成功的狀態下APP不會閃退

加入其他FIREBASE的元件

- 首先把decode完的圖片轉成FirebaseVisionImage的格式

```
FirebaseVisionImage image = FirebaseVisionImage.fromBitmap(bitmap);
```

- 再來引入On-device的文字辨識的detector

```
FirebaseVisionTextRecognizer detector = FirebaseVision.getInstance().getOnDeviceTextRecognizer();
```

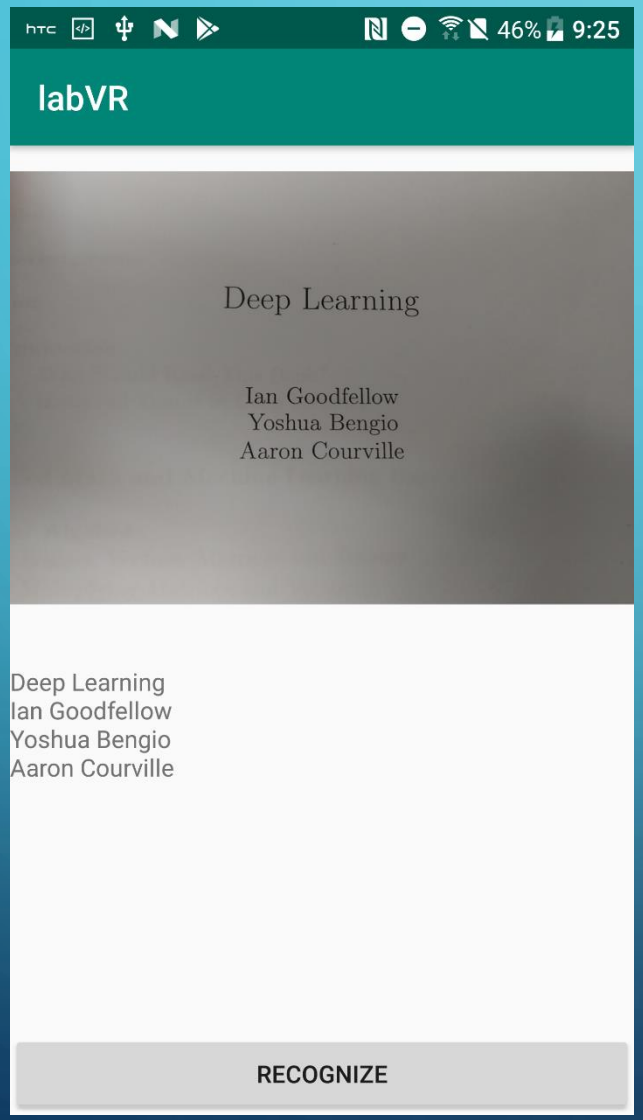
這邊的程式碼都寫在剛剛public void onImage()裡面

- 新增一個影像辨識的Task,並增加辨識success/fail事件的Listener:

```
Task<FirebaseVisionText> result =
                    detector.processImage(image).addOnSuccessListener(new OnSuccessListener<FirebaseVisionText>() {
                        @Override
                        public void onSuccess(FirebaseVisionText firebaseVisionText) {
                            //辨識成功則顯示success在銀幕上
                            Toast.makeText(MainActivity.this, "success", Toast.LENGTH_SHORT).show();
                           //用TextView來接影像辨識的結果
                            tv_result.setText(firebaseVisionText.getText());
                        }
                    })
                    .addOnFailureListener(
                            new OnFailureListener() {
                                @Override
                                public void onFailure(@NonNull Exception e) {
                                    //辨識失敗則顯示fail在銀幕上
                                    Toast.makeText(MainActivity.this, "fail", Toast.LENGTH_SHORT).show();
                                }
                            });
```

這邊的程式碼一樣寫在剛剛public void onImage()裡面

CHECK POINT: TEXT RECOGNITION



隨便找一段文字, 並測試辨識結果, 這邊要注意鏡頭實際上抓到的畫面會比你設定的CameraKitView大,所以會辨識到更大範圍的文字

Note:剛剛所寫的元件都需要另外import，善用ALT+ENTER來自動import

MORE THINGS YOU NEED TO TRY

- 用FirebaseVisionText.TextBlock來分句輸出結果

```
for (FirebaseVisionText.TextBlock block: firebaseVisionText.getTextBlocks()) {
    String blockText = block.getText();
    for (FirebaseVisionText.Line line: block.getLines()) {
        String lineText = line.getText();
        …
```

- Firebase ON-CLOUD模型如何引入與使用?

```
FirebaseVisionCloudTextRecognizerOptions …
```

- TENSORFLOW-LITE模型的訓練與使用(前面有放教學連結)

延伸學習:

- Recognize Text in Images with ML Kit on Android (Firecasts):

https://www.youtube.com/watch?v=_qrl1JUCMjI

- ML Kit for Firebase Quickstart:

https://github.com/firebase/quickstart-android/tree/master/mlkit

- Recognize text, facial features, and objects in images with ML Kit for Firebase:

https://codelabs.developers.google.com/codelabs/mlkit-android/#0

- Getting Started With Firebase ML Kit for Android

https://code.tutsplus.com/tutorials/getting-started-with-firebase-ml-kit-for-android--cms-31305