Introduzione

Questa fase può essere svolta senza problemi all’intero di un qualunque sistema operativo, unico vincolo è la necessità di disporre del modello (frozen\_inference\_graph.pb) precendentemente ottenuto.

Viene mostrata l’installazione del software di sviluppo Android Studio e l’apertura dell’applicazione mobile fornita da Tensor Flow.

L’importazione del modello addestrato nelle fasi precedenti all’interno della suddetta applicazione

1. [Installazione Android Studio …………………………….…………………………….…………………………….……………… 2](#InstallazioneAndroid)
2. [Installazione App …………………………….…………………………….…………………………….…………………………….… 3](#InstallazioneApp)

Installazione Android Studio

Per installare Android Studio ed importare il progetto eseguire i seguenti step:

In ambiente Linux è possibile recarsi nello store di Ubuntu dal menù laterale, cercare Android Studio ed eseguire l’installazione.

Alternativamente per tutti i sistemi operativi recarsi alla seguente pagina:

<https://developer.android.com/studio/>

scaricare il file ed eseguire l’installazione.

Una volta eseguita l’installazione sarà necessario aprire il progetto Android fornito da TensorFlow.

La repository si trova sulla pagina github di tensorflow:

<https://github.com/tensorflow/tensorflow>

Essa contiene anche il progetto per Android Studio dell’applcazione.

1. Aprire Android Studio ed installare tutte le componenti che vengono richieste.
2. Scegliere l’opzione Open an existing Android Studio project e selezionare la seguente cartella:

tensorflow/tensorflow/examples/**android**

1. Attendere che venga eseguito il build ed installare tutti gli eventuali elementi richiesti in messaggi di errore.
2. Verrà richiesto di aggiornare l’Android Gradle plugin alla versione 4.4, questa operazione non è necessaria, se si decidesse di eseguirla comparirà un warning che indica il fatto che la configurazione ‘compile’ è obsoleta; ignorare il warning in quando non inficerà in alcun modo il corretto funzionamento dell’applicazione.

Installazione App

Una volta eseguita l’installazione e l’importazione del progetto sarà necessario importare il proprio modello addestrato all’interno del progetto Android Studio stesso.

Come spiegato nella documentazione della demo Android fornita da tensorflow: <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android>,

è necessario nel file Gradle Scripts/build.gradle, settare alla riga 43:

def nativeBuildSystem = 'none'

Dopo la modifica eseguire Sync Now in alto a destra.

Sarà quindi necessaria la connessione internet per eseguire il passo successivo, in assenza di connessione non sarà possibile eseguirlo.

Quindi eseguire Build -> Rebuild Project

1. Dal menù laterale aprire la cartella assets. All’interno saranno presenti i modelli di default forniti da TensorFlow e i relativi files.txt
2. Aprire l’Activity all’interno della cartella java: Detector Activity.

Alle righe 65-66-67sarà presente il seguente codice:

private static final String *TF\_OD\_API\_MODEL\_FILE* =  
 "file:///android\_asset/ssd\_mobilenet\_v1\_android\_export.pb";  
private static final String *TF\_OD\_API\_LABELS\_FILE* = "file:///android\_asset/coco\_labels\_list.txt";

Per non eseguire modifiche al codice dell’applicazione sarà necessario rinominare il proprio modello ottenuto dalla fase di train come: **ssd\_mobilenet\_v1\_android\_export.pb**

1. Creare quindi un file di testo attraverso il proprio editor di testo nella prima riga inserire tre punti interrogativi e nella seconda il nome dell’oggetto da riconoscere utilizzato nelle fasi precedenti (sia in LabelImg che nel file .pbtxt).

Per non eseguire modifiche del codice sarà necessario chiamare questo file: **coco\_labels\_list.txt.**

In alternativa si può modificare quello presente strutturandolo come spiegato sopra.

1. Eliminare il/i file presente/i facendo tasto destro su di esso/i, delete e dalla finestra che si apre deselezionare l’opzione “safe delete”.

Inserire il proprio modello (ed eventualmente il file.txt) opportunamente rinominato. Per fare questo sarà necessario eseguire copia e successivamente incolla nella cartella assets del menù leterale.

(Sarà possibile inserire il modello da me addestrato nel modo appena spiegato, esso è già rinominato in maniera corretta, sarà necessario realizzare il file.txt con i tre punti interrogativi nella prima riga e “san\_bernardo” nella seconda riga, avendo quindi questa struttura.

???

san\_bernardo

1. Fatto ciò sarà necessario impedire all’applicazione di eseguire il download dei modelli che andrebbero a sovrascrivere quelli da noi inseriti. Per fare ciò recarsi nuovamente nel file build.gradle; alla riga 185 vi è un commento fornito da TensorFlow che spiega come impedire il download dei modelli commentando la riga sottostante. Commentare quindi la riga 187.

Eseguire quindi nuovamente Sync Now in alto a destra.

Fatto ciò potrebbe essere necessario rieseguire il build cliccando sul simbolo del martello nella barra degli strumenti.

1. Per installare l’applicazione sul proprio dispositivo vi sono quindi due metodologie differenti:

La prima consiste nel collegare il dispositivo tramite cavo USB, quindi abilitare l’opzione debug USB dal dispositivo stesso (sotto una guida per come fare):

<https://www.androidpit.it/come-attivare-debug-usb>

Collegato il dispositivo e forniti i permessi per il computer che si sta utilizzando premere il pulsante play nella barra degli strumenti e selezionare il dispositivo connesso come device di installazione.

La seconda metodologia consiste invece nell’esportazione del file di installazione APK. Per fare ciò cliccare su Build e quindi Build APK(s); il file verrà generato e sarà possibile localizzarlo cliccando su locate nella finestra che si apre in basso a destra ad operazione conclusa.

In ogni caso si troverà nella cartella:

tensorflow/tensorflow/examples/**android/gradleBuild/outputs/apk/debug**

Quindi caricare sul proprio device l’apk ed eseguire l’installazione dello stesso.

(Io ad esempio utilizzo l’applicazione di messaggistica Telegram per inviarmi il file da desktop e tenerlo salvato sul dispositivo ma ogni metodo utilizzato va benissimo).

1. Una volta installata l’applicazione compariranno 4 icone differenti di TensorFlow:

TF Classify, TF Stylize, TF Speech e TF Detect.

L’unica applicazione alla quale siamo interessati è TF Detect, infatti obbiettivo del progetto è quello di addestrare un modello di machine learning e verificare che attraverso esso sia possibile identificare un oggetto all’interno di un ambiente.

Poiché l’obbiettivo dell’azienda che ha commissionato il progetto è quello di implementare tale riconoscitore su google glasses o dispositivi simili, l’applicazione ha lo scopo di verificare la bontà di addestramento del modello e capire come è possibile interfacciarsi con esso.

Tale applicazione una volta eseguita mostrerà un messaggio di errore che come spiegato nella documentazione fornita da TensorFlow: <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/examples/android>

Indica che il tracking dell’oggetto non è supportato settando **def nativeBuildSystem = 'none'** che però serve per poter eseguire il build del progetto su Android Studio. Ad ogni modo ciò non inficia in alcun modo il funzionamento dell’app ed il riconoscimento di un oggetto.

Con l’applicazione sarà possibile inquadrare l’ambiente circostante, nel momento in cui viene inquadrato l’oggetto per il quale è stato eseguito il training, comparirà attorno ad esso un riquadro colorato, una percentuale di affidabilità ed il label indicante il nome dell’oggetto.

Le altre applicazioni invece utilizzeranno i modelli di default forniti da tensorflow in quanto le funzionalità da esse proposte non sono necessarie per quanto richiesto dal progetto.

La repository scaricata da Github di tensorflow è la stessa presente in AndroidFolder nella quale le modifiche appena spiegate sono già state eseguite; se si decidesse di utilizzare questa sarà necessario solamente importare il progetto come spiegato [qui](#Import) selezionando però la cartella AndroidFolder/tensorflow/examples/android ed installare l’applicazione.

Sia che si utilizzi la repository ed il progetto da me fornito, sia che si inserisca il mio modello all’interno della repository scaricata da GitHub di tensorflow l’oggetto che verrà riconosciuto da TF Detect sarà la bottiglia di acqua San Bernardo.

L’APK presente nella cartella consegna è stato ottenuto dalla repository AndroidFolder seguendo i procedimenti spiegati sopra. Quindi l’applicazione TF Detect avrà il funzionamento da me voluto riconoscendo solo la bottiglia di San Bernardo mentre le altre applicazioni utilizzeranno i modelli di default di TensorFlow e avranno le funzionalità spiegate nella documentazione di TensorFlow.