Machine Learning in Android App Development

Fajar Ulin Nuha -
Principal Engineer Android

tokopedia

About Me

- Ilmu Komputer UGM 2014
- OmahTI Mobile Apps
- Lab Sistem Cerdas
- Skripsi: Training Dataset Reduction on GAN
- Research Assistant: Comet assay classification
- Tokopedia Android Engineer (2018-now)

tokopedia

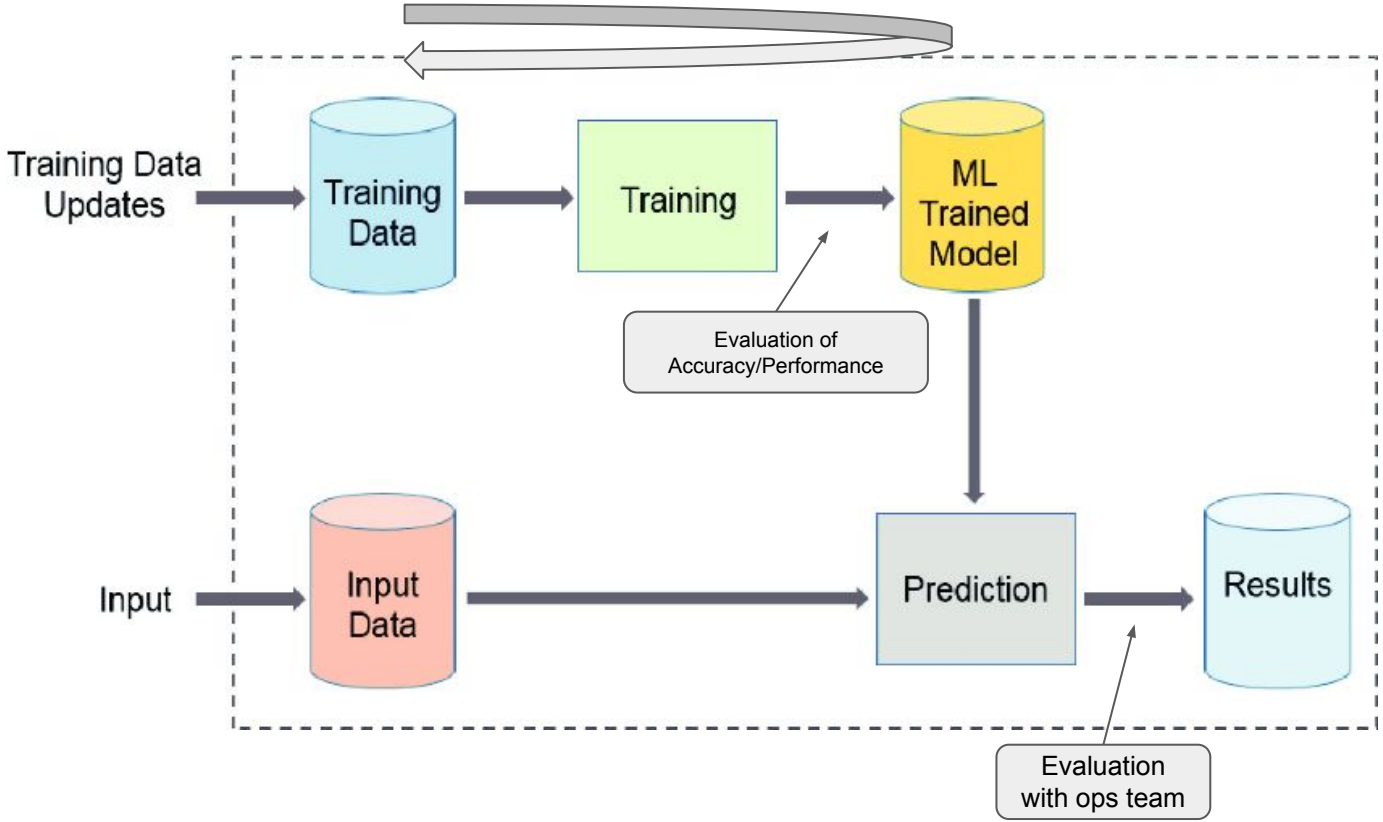
Overview

1. Machine Learning Deployment Overview
2. Android ML: ML Kit & Tensorflow Lite
3. Mobile Apps: Implementation & Challenge
4. Careers at Tokopedia

tokopedia

Machine Learning Deployment Overview

Generic Machine Learning Workflow



On-Device Inference

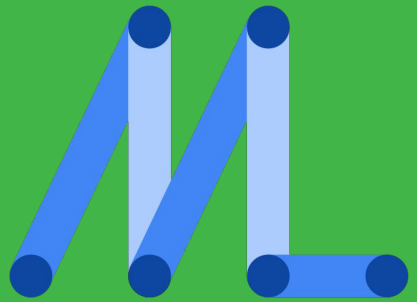
Cloud-based Inference

Inference

Issue	On-device inference	Cloud-based inference
Latency	Lower latency enhances the realtime experience	Asynchronous communication and available bandwidth can affect latency
Resources	The particular device's resources, like processing power and storage, can limit performance	Cloud-based resources are more powerful and storage is more plentiful
Offline/Online	The ability to operate offline is a plus for running with poor or non-existing network infrastructure	A network connection is required
Cost	Battery usage, model download time for end users	Bandwidth for data transfer for end users, computing charges for developers
Privacy	User data never leaves the device	Data may leave the device, additional precautions may be necessary

source

ML Kit



- Available in Android and iOS
- On-device inference
- Powerful yet easy to use Vision and Natural Language APIs
- Powered by Google's best-in-class ML models and offered at no cost
- Google product

Key ML Development Areas

Design: https://pair.withgoogle.com/guidebook (ux)
Explore pre-trained models: https://tfhub.dev/s?deployment-format=lite
Training: local or cloud

ML Kit

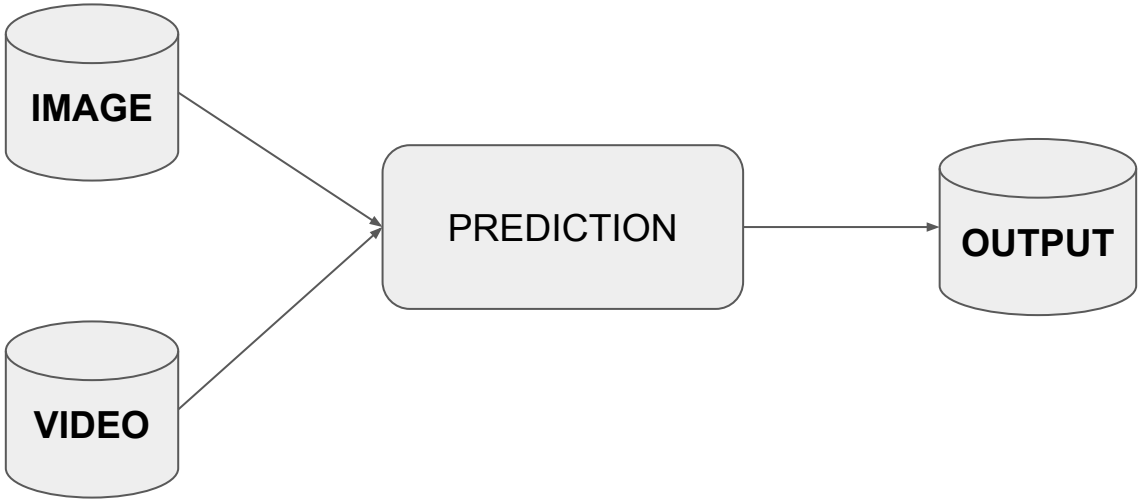
Vision

- Text Recognition
- Face Detection
- Pose Detection (beta)
- Selfie Segmentation (beta)
- Barcode Scanning
- Image Labelling
- Object Detection & Tracking
- Digital Ink Recognition
- Custom Models

Natural Language

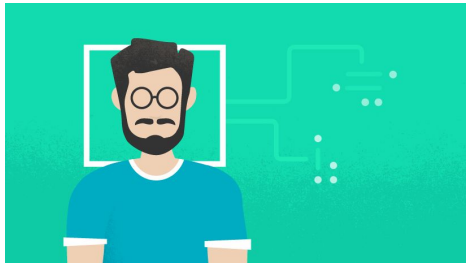
- Identify Languages
- Translate Text
- Smart Replies
- Entity Extraction (beta)

ML Vision



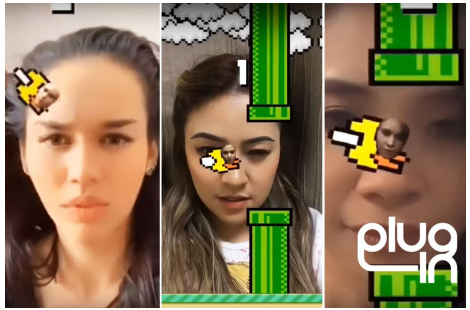
Face Detection

- Identify key facial features
- Get the contours of detected faces
- Detects faces, it does not recognize people

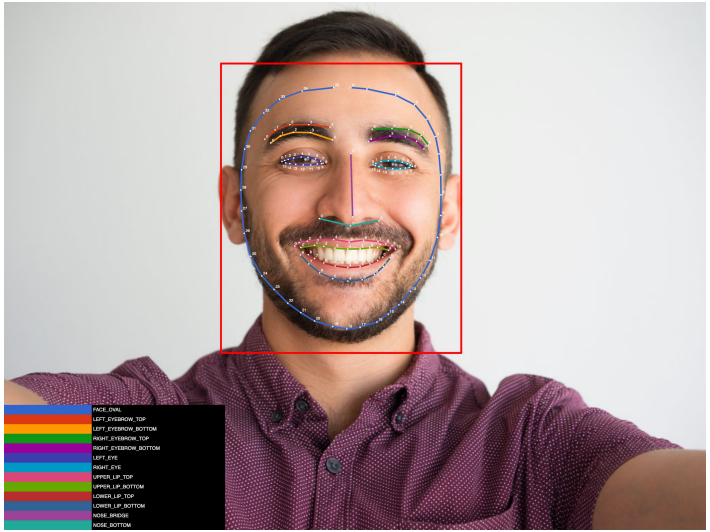


Application:

- Perform tasks like embellishing selfies and portraits
- Generating avatars from a user's photo
- Applications like video chat or games that respond to the player's expressions



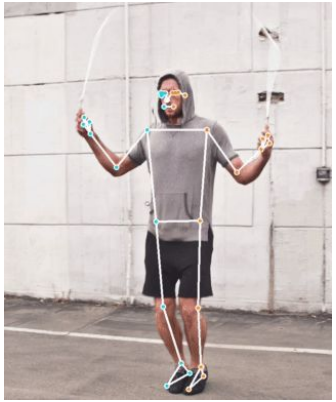
Face 1 of 3		
Bounding polygon	(884.880004882812, 149.546676635742), (1030.77197265625, 149.546676635742), (1030.77197265625, 329.660278320312), (884.880004882812, 329.660278320312)	
Angles of rotation	Y: -14.054030418395996, Z: -55.007488250732422	
Tracking ID	2	
Facial landmarks	Left eye	(945.869323730469, 211.867126464844)
	Right eye	(971.579467773438, 247.257247924805)
	Bottom of mouth	(907.756591796875, 259.714477539062)
	... etc.	
Feature probabilities	Smiling	0.88979166746139526
	Left eye open	0.98635888937860727
	Right eye open	0.99258323386311531



Facial feature contours	
Nose bridge	(505.149811, 221.201797), (506.987122, 313.285919)
Left eye	(404.642029, 232.854431), (408.527283, 231.366623), (413.565796, 229.427856), (421.378296, 226.967682), (432.598755, 225.434143), (442.953064, 226.089508), (453.899811, 228.594818), (461.516418, 232.650467), (465.069580, 235.600845), (462.170410, 236.316147), (456.233643, 236.891602), (446.363922, 237.966888), (435.698914, 238.149323), (424.320740, 237.235168), (416.037720, 236.012115), (409.983459, 234.870300)
Top of upper lip	(421.662048, 354.520813), (428.103882, 349.694061), (440.847595, 348.048737), (456.549988, 346.295532), (480.526489, 346.089294), (503.375702, 349.470459), (525.624634, 347.352783), (547.371155, 349.091980), (560.082031, 351.693268), (570.226685, 354.210175), (575.305420, 359.257751)
(etc.)	

Pose Detection

- Detect the pose of a subject's body in real time from a continuous video or static image
- Full body tracking The model returns 33 key skeletal landmark points, including the positions of the hands and feet
- ***InFrameLikelihood*** score For each landmark, a measure that indicates the probability that the landmark is within the image frame. The score has a range of 0.0 to 1.0, where 1.0 indicates high confidence
- Z Coordinate for depth analysis This value can help determine whether parts of the users body are in front or behind the users' hips



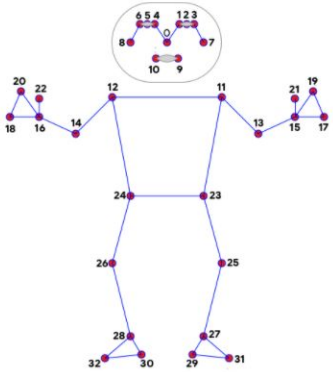
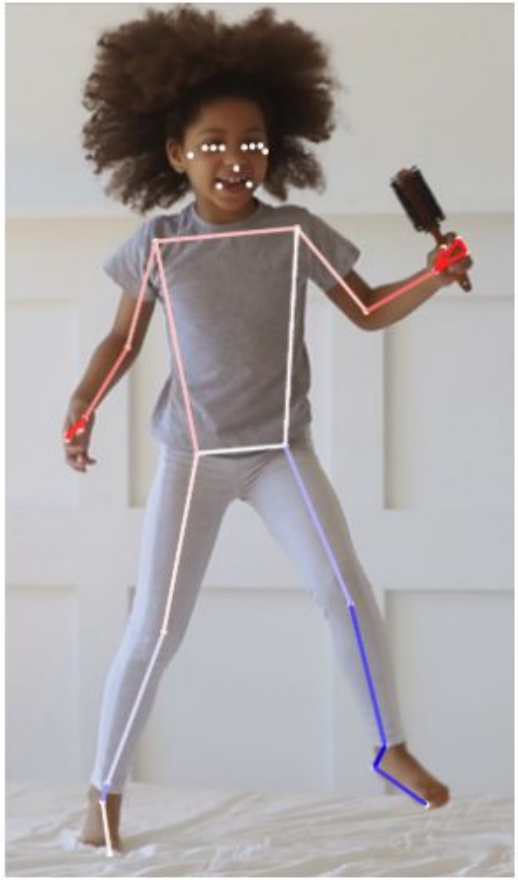


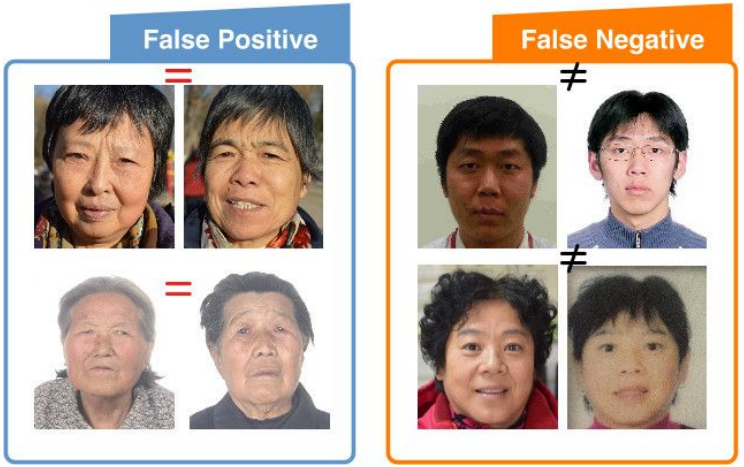
Figure 1. Landmarks

Landmark	Type	Position	InFrameLikelihood
11	LEFT_SHOULDER	(734.9671, 550.7924, -118.11934)	0.9999038
12	RIGHT_SHOULDER	(391.27032, 583.2485, -321.15836)	0.9999894
13	LEFT_ELBOW	(903.83704, 754.676, -219.67009)	0.9836427
14	RIGHT_ELBOW	(322.18152, 842.5973, -179.28519)	0.99970156
15	LEFT_WRIST	(1073.8956, 654.9725, -820.93463)	0.9737737
16	RIGHT_WRIST	(218.27956, 1015.70435, -683.6567)	0.995568
17	LEFT_PINKY	(1146.1635, 609.6432, -956.9976)	0.95273364
18	RIGHT_PINKY	(176.17755, 1065.838, -776.5006)	0.9785348



Case: KYC

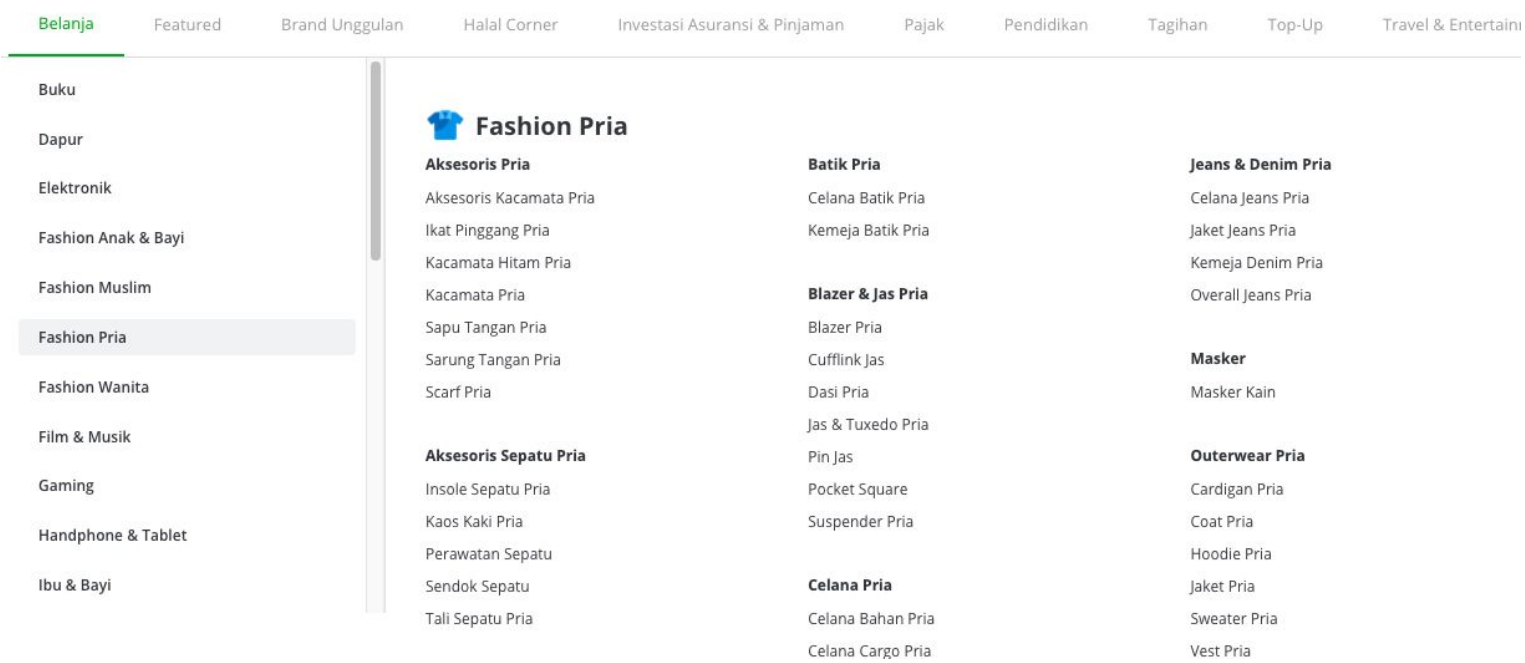
- Know Your Customer, ID Verification
- Typical flow: KTP & Selfie
- Many problems with false positive case
- Can we improve using Machine Learning?
- Liveness: Action Video
- Face Detection: Face feature probability distribution
- Face Comparison



source

tokopedia

Case: Category Recommendation



Custom Models

- Before going further: Use a model published on TensorFlow Hub

Options to train your own image classification model	
AutoML Vision Edge	• Offered through Google Cloud AI • Create state-of the art image classification models • Easily evaluate between performance and size
TensorFlow Lite Model Maker	• Re-train a model (transfer learning), takes less time and requires less data than training a model from scratch
Convert a TensorFlow model to TensorFlow Lite	• Train a model with TensorFlow and then convert it to TensorFlow Lite

Custom Models

FTF: Try to explore models published on TensorFlow Hub

TensorFlow Lite Model Maker

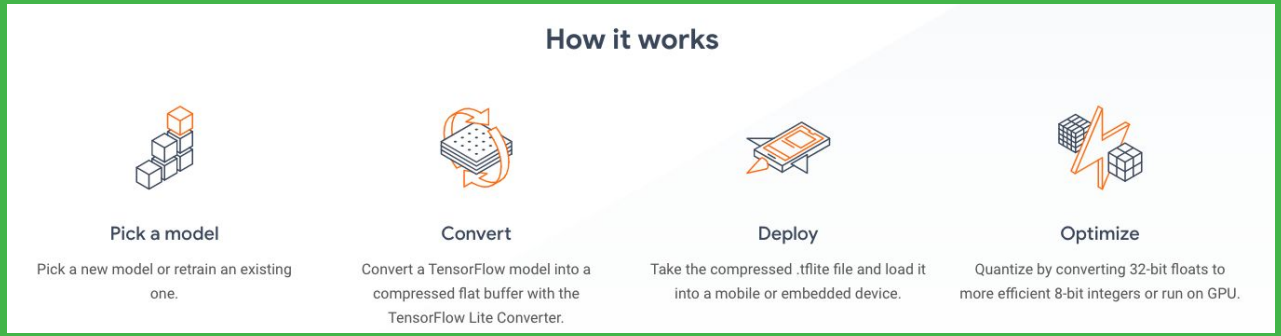
Re-train a model (transfer learning), takes less time and requires less data than training a model from scratch

Image Labeling API or the Object Detection and Tracking API

Convert Tensorflow model to Tensorflow Lite

If you have an existing TensorFlow image classification model, you can convert it using the TensorFlow Lite converter.

TensorFlow Lite



Vision

- Image Classification
- Object Detection
- Pose Estimation
- Segmentation
- Style Transfer
- Super resolution

Text

- BERT Question and Answer
- Smart Reply
- Text classification
- Recommendation

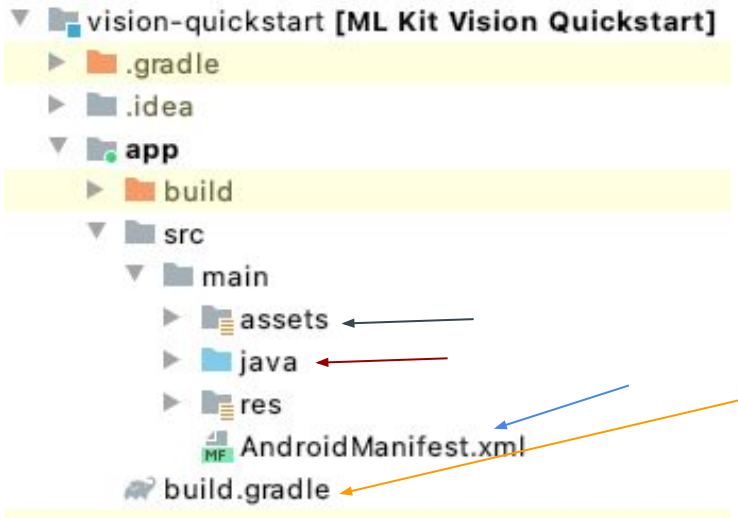
Mobile Apps: Implementation & Challenge

Let’s Code

- Object Detection and Tracking
- Base model & Custom model
- It detects up to five objects in the image along with the position of each object in the image
- When detecting objects in video streams, each object has a unique ID that you can use to track the object from frame to frame
- You can also optionally enable coarse object classification, which labels objects with broad category descriptions

Object 0	
Bounds	(1, 97), (332, 97), (332, 332), (1, 332)
Category	FASHION_GOOD
Classification confidence	0.95703125
Object 1	
Bounds	(186, 80), (337, 80), (337, 226), (186, 226)
Category	FASHION_GOOD
Classification confidence	0.84375
Object 2	
Bounds	(296, 80), (472, 80), (472, 388), (296, 388)
Category	FASHION_GOOD
Classification confidence	0.94921875
Object 3	
Bounds	(439, 83), (615, 83), (615, 306), (439, 306)
Category	FASHION_GOOD
Classification confidence	0.9375

Let's Code



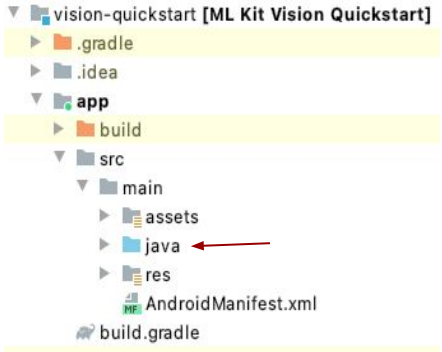

```
dependencies {
  // ...

  implementation 'com.google.mlkit:object-detection:16.2.4'

}
```

tokopedia

Let's Code



Kotlin Java

```kotlin
// Live detection and tracking
val options = ObjectDetectorOptions.Builder()
        .setDetectorMode(ObjectDetectorOptions.STREAM_MODE)
        .enableClassification()  // Optional
        .build()

// Multiple object detection in static images
val options = ObjectDetectorOptions.Builder()
        .setDetectorMode(ObjectDetectorOptions.SINGLE_IMAGE_MODE)
        .enableMultipleObjects()
        .enableClassification()  // Optional
        .build()
```

Kotlin Java

```kotlin
val objectDetector = ObjectDetection.getClient(options)
```

Kotlin Java

```kotlin
val image = InputImage.fromBitmap(bitmap, 0)
```

Kotlin Java

```kotlin
objectDetector.process(image)
    .addOnSuccessListener { detectedObjects ->
        // Task completed successfully
        // ...
    }
    .addOnFailureListener { e ->
        // Task failed with an exception
        // ...
    }
```

tokopedia

App Size Challenge

- App size is very important to a user
- ML model tends to fill great amount of size in the apk
- Each model will take MBs. Face Detection model can take up to ~16 MB

Base Model: Bundled vs Unbundled

- A bundled model which is part of your app and an unbundled model that depends on Google Play Services. The two models are the same. If you select the unbundled model your app will be smaller.

Feature	Unbundled	Bundled
Implementation	Model is dynamically downloaded via Google Play Services.	Model is statically linked to your app at build time.
App size	No impact.	About 16MB model size.
Initialization time	Might have to wait for model to download before first use.	Model is available immediately

```
dependencies {
  // ...
  // Use this dependency to use dynamically downloaded model in Google Play Service
  implementation 'com.google.android.gms:play-services-mlkit-face-detection:16.1.5'
}
```

To opt-in:
Edit build.gradle

```
<application ...>
  ...
<meta-data
    android:name="com.google.mlkit.vision.DEPENDENCIES"
    android:value="face" />
<!-- To use multiple models: android:value="face,model2,model3" -->
</application>
```

Install-time download:
Edit AndroidManifest.xml

tokopedia

Custom Model: Bundled vs Hosted

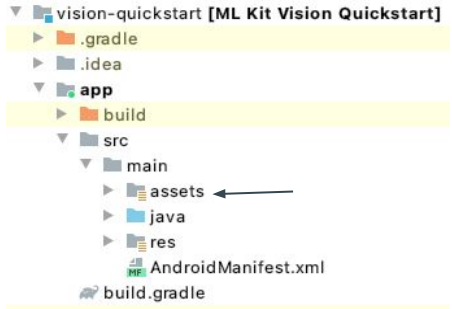
Bundled Model	Hosted Model
The model is part of your app's APK, which increases its size.	The model is not part your APK. It is hosted by uploading to Firebase Machine Learning.
The model is available immediately, even when the Android device is offline	The model is downloaded on demand
No need for a Firebase project	Requires a Firebase project
You must republish your app to update the model	Push model updates without republishing your app
No built-in A/B testing	Easy A/B testing with Firebase Remote Config

- There are two ways to integrate a custom model. You can bundle the model by putting it inside your app's asset folder, or you can dynamically download it from Firebase

tokopedia

Custom Model

```
dependencies {
  // ...
  // Object detection & tracking feature with custom bundled model
  implementation 'com.google.mlkit:object-detection-custom:16.3.1'
}
```



1. Copy the model file (usually ending in .tflite or .lite) to your app's assets/ folder
2. Then, add the following to your app's build.gradle

```
android {
    // ...
    aaptOptions {
        noCompress "tflite"
        // or noCompress "lite"
    }
}
```

Kotlin Java

```
val localModel = LocalModel.Builder()
        .setAssetFilePath("model.tflite")
        // or .setAbsoluteFilePath(absolute file path to model file)
        // or .setUri(URI to model file)
        .build()
```

tokopedia

Custom Model

Kotlin Java

```kotlin
val localModel = LocalModel.Builder()
        .setAssetFilePath("model.tflite")
        // or .setAbsoluteFilePath(absolute file path to model file)
        // or .setUri(URI to model file)
        .build()
```

```kotlin
// Multiple object detection in static images
val customObjectDetectorOptions =
        CustomObjectDetectorOptions.Builder(localModel)
        .setDetectorMode(CustomObjectDetectorOptions.SINGLE_IMAGE_MODE)
        .enableMultipleObjects()
        .enableClassification()
        .setClassificationConfidenceThreshold(0.5f)
        .setMaxPerObjectLabelCount(3)
        .build()

val objectDetector =
        ObjectDetection.getClient(customObjectDetectorOptions)
```

tokopedia

Explore more

- ML Kit : https://developers.google.com/ml-kit/guides
- ML Kit Samples: https://github.com/googlesamples/mlkit
- Tensorflow Lite : https://www.tensorflow.org/lite
- Tensorflow Lite Samples:
  https://github.com/tensorflow/examples/tree/master/lite/examples
- Tensorflow Lite Model Maker: https://www.tensorflow.org/lite/guide/model_maker

Careers at Tokopedia

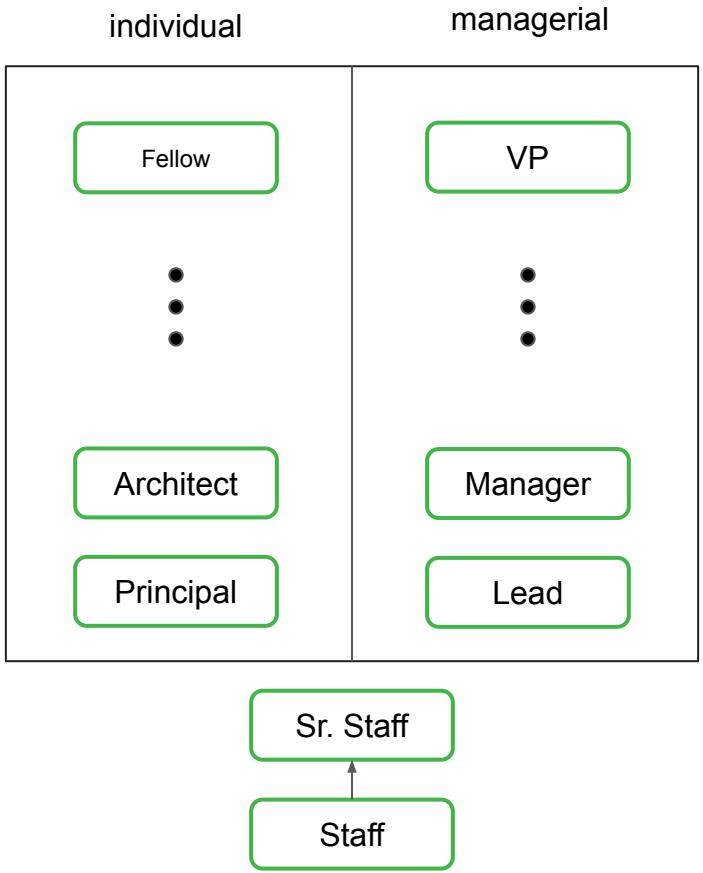
3 DNA of Tokopedia - our culture, our belief

Working @ Tokopedia

1. My Career
2. Environment
3. Perks
4. Data office

tokopedia

Careers @ Tokopedia



tokopedia


Data Office
Tokopedia

Data Office

Data Scientist

- AI solution
- research/scientist
- model

Data Engineer

- build data pipeline from source to data warehouse

Data Analyst

- analyze data for business team
- business data visualization and analysis

Business Intelligence

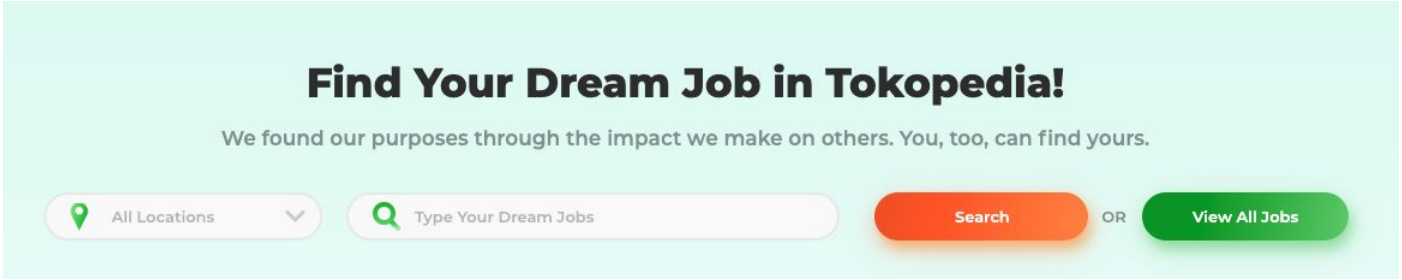
- public data


tokopedia
Advanced Analytics
Academy

What makes you a good candidate?

1. AI/ML General Knowledge
2. Coding, Software Engineering -> python, pandas, seldoncore, git, linux, ssh, etc
3. Problem Solving & Analytical Thinking
4. Be practical and have a business mindset
5. 3 DNA

tokopedia



https://www.tokopedia.com/careers/

video

Thank You

Contact me fajar.nuha@tokopedia.com, @fajarnuha