E2E Computer Vision models using TensorFlow

Google Developers

Nitin Tiwari
Google Developer Expert – Machine Learning
Software Engineer @ LTIMindtree

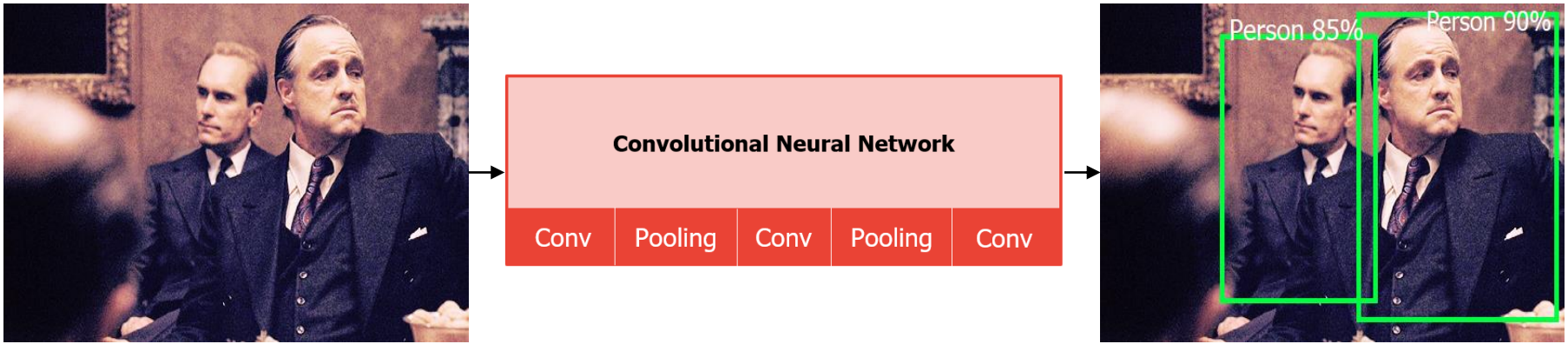
Content

1. Introduction to Object Detection
2. The Building Blocks of an ML application
3. Building your custom object detector
4. Some interesting examples built with TensorFlow

Introduction to Object Detection

Object Detection is a Computer Vision technique to identify and locate objects in images and videos.

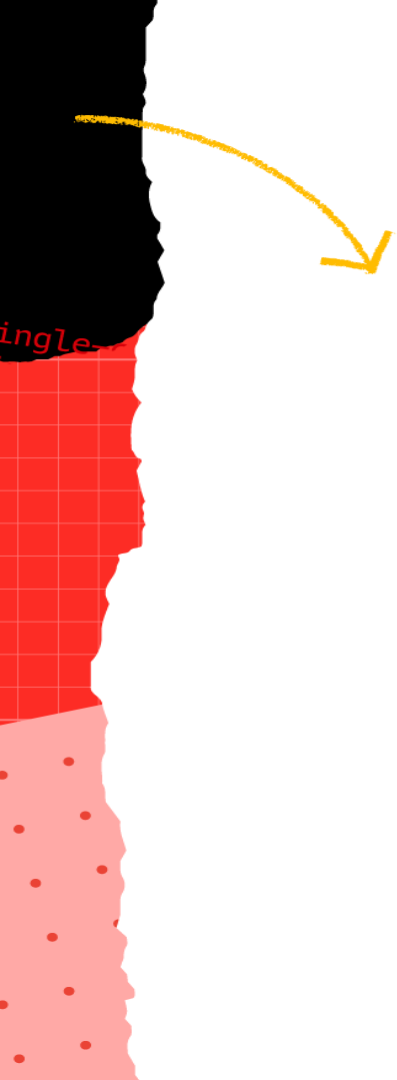
Object Detection = Image Classification + Localization



Original Image (input)

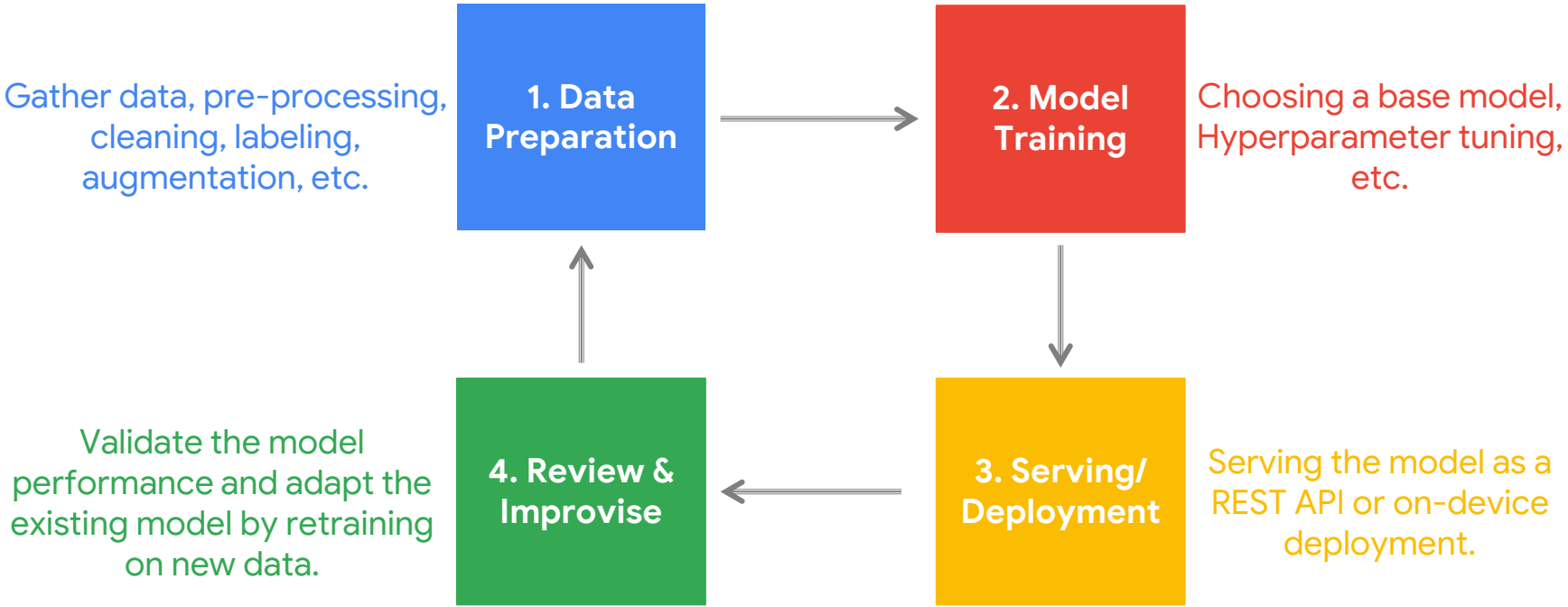
Image with detection (output)

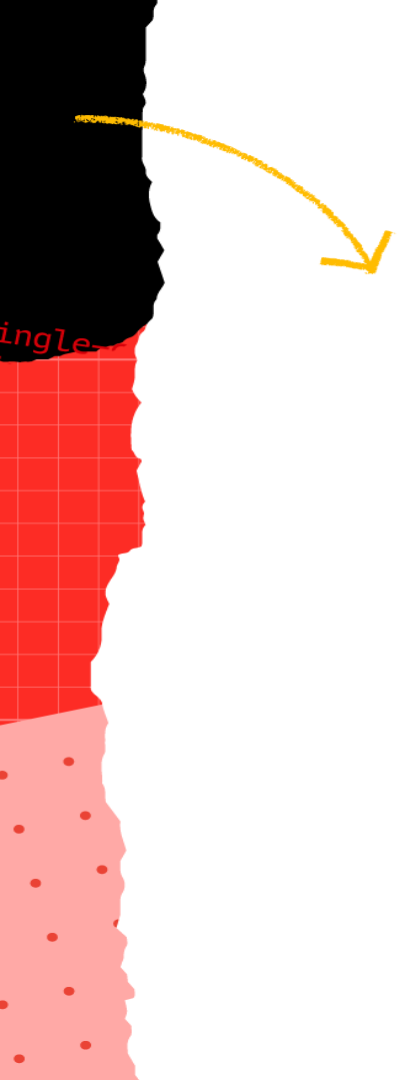
- **Convolution Layer:** Detects the important features in the input image.
- **Pooling layer:** Reduces the size of the image while retaining the features.
- There are multiple convolutional and pooling layers in a CNN.
- Input tensor: [n x n x 3] image pixels
- Output tensor: [(labels), (bounding box coordinates), (confidence), (no. of detections)]



The Building Blocks of an ML application

ML Lifecycle



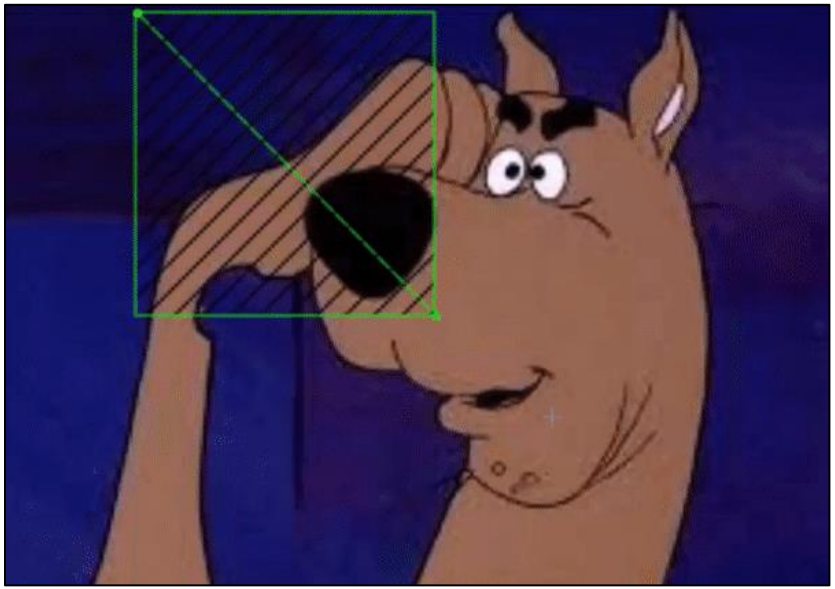


Building your custom object detector

Custom Cartoon Detector

Data Preparation:

1. **Collection:** Collect data from different sources such as Google Images, web scraping, etc.
2. **Pre-processing:** Cleaning, transformation
3. **Augmentation:** Generate new data by rotating, scaling, adding/removing noise, etc.
4. **Annotation:** Labelling the images with relevant tags/classes.



Image

```
<?xml version="1.0"?>
- <annotation>
    <folder>dataset_PASCAL VOC</folder>
    <filename>cartoon108.jpg</filename>
    <path>F:\Cartoon Object Detection\dataset_PASCAL VOC\cartoon108.jpg</path>
  - <source>
      <database>Unknown</database>
    </source>
  - <size>
      <width>300</width>
      <height>168</height>
      <depth>3</depth>
    </size>
    <segmented>0</segmented>
  - <object>
      <name>mrbean</name>
      <pose>Unspecified</pose>
      <truncated>1</truncated>
      <difficult>0</difficult>
    - <bndbox>
        <xmin>107</xmin>
        <ymin>20</ymin>
        <xmax>222</xmax>
        <ymax>168</ymax>
      </bndbox>
    </object>
</annotation>
```

Annotation file (XML)

Model Training:

Choose a base model (EfficientDet-Lite2 is an object detection model for Android/IoT devices)

```
spec = model_spec.get('efficientdet_lite2')
```

Load the train and test dataset

```
train_data = object_detector.DataLoader.from_pascal_voc(path_to_train_images,
path_to_train_annotations, labels)

test_data = object_detector.DataLoader.from_pascal_voc(path_to_test_images,
path_to_test_annotations, labels)
```

Define the hyperparameters

```
model = object_detector.create(train_data, model_spec=spec, batch_size=4,
train_whole_model=True, validation_data=test_data)
```

Evaluate the model

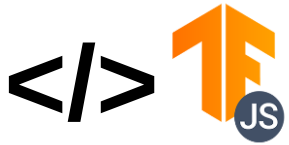
```
model.evaluate(validation_data)
```

Complete Colab notebook:
https://github.com/NSTiwari/Custom-Object-Detection-on-Android-using-TF-Lite

Serving/Depoyment:

- Any ML workflow is incomplete without the deployment of the model.
- TensorFlow models can be of the following formats depending upon where they need to be deployed.

Standalone Desktop Application
(TensorFlow)



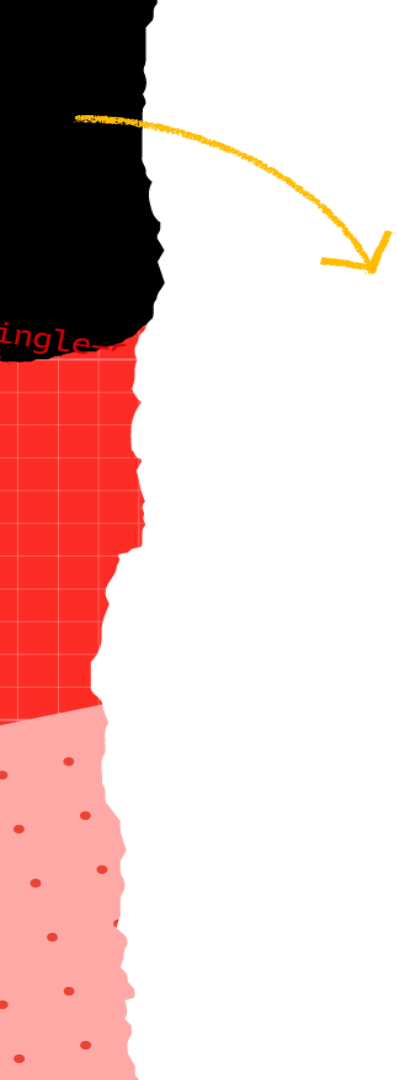
Browser/Web Application
(TensorFlow.js)

Mobile/IoT Devices
(TensorFlow Lite)

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_float_model = converter.convert()

with open('model.tflite', 'wb') as f:
f.write(tflite_float_model)
```

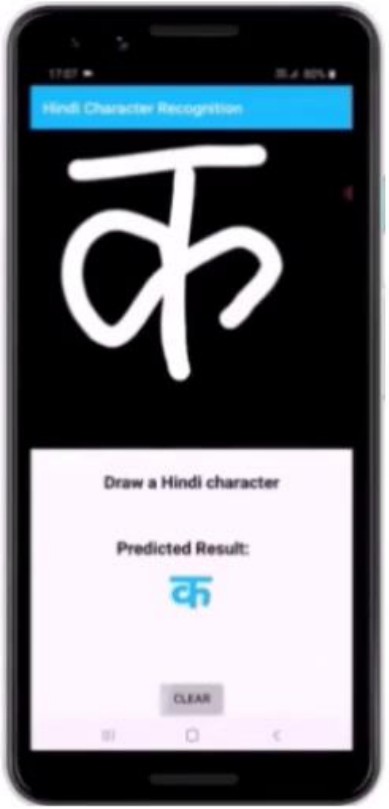
E2E Flow: The final TF Lite model is deployed on an Android application.



examples

Pose Classifier-based Video Game Controller – Ashes Cricket 2009

Real-time object detection – GTA Vice City



Hindi Character Recognizer


SO YA
38
GOT ANY QUESTIONS?

Thank you.

github.com/NSTiwari

medium.com/@tiwarinitin1999

Experts

Machine Learning