Machine Learning on Google Cloud Platform

Google Developers

24/03/2023

Nitin Tiwari

Google Developer Expert – Machine Learning

Software Engineer @ LTIMindtree

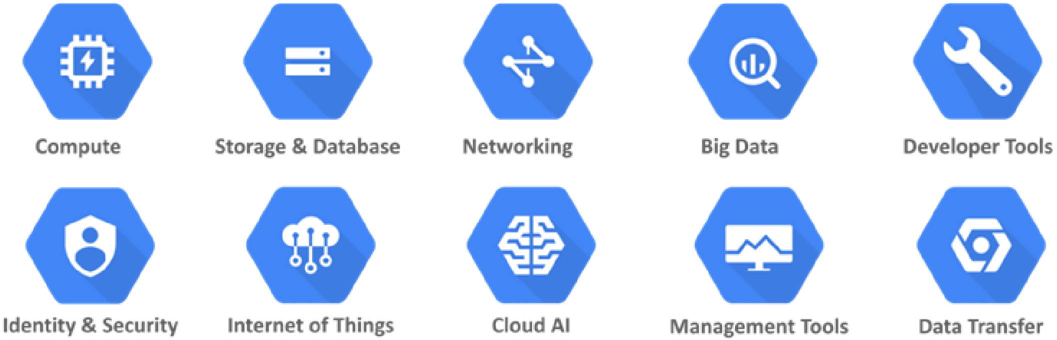
Content

1. Google Cloud Platform & Vertex AI Vision
2. The Building Blocks of an ML application
3. Introduction to Object Detection
4. Building your custom object detector
5. Some interesting examples built with TensorFlow

Google Cloud Platform & Vertex AI Vision

Google Cloud Platform:

Google Cloud Platform (GCP) is a suite of cloud computing services such as data analytics, storage, Machine Learning, API development, etc.

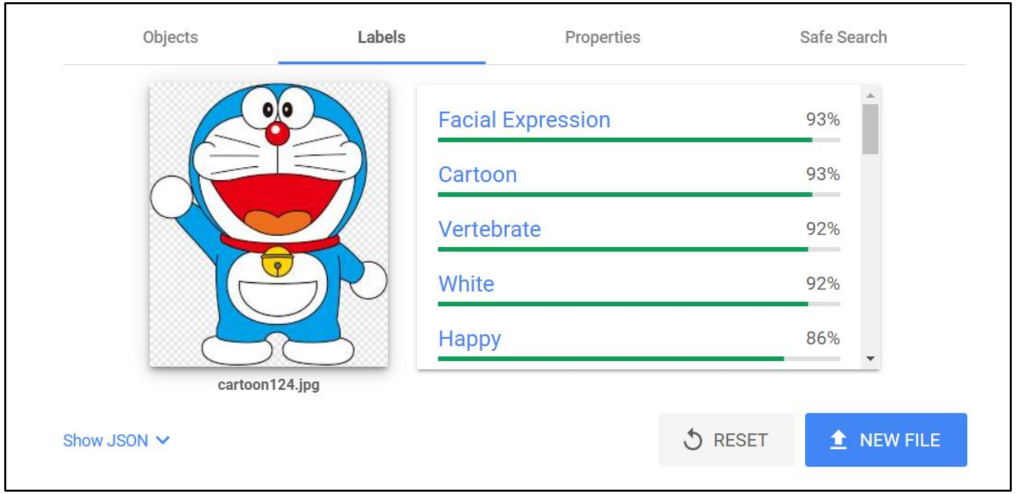


Vertex AI Vision:

A fully managed end-to-end application development environment that lets you easily build, deploy and manage computer vision applications.

Using Vertex AI Vision, you can train your custom ML models for image classification, object detection, image segmentation, text classification, optical character recognition (OCR), etc.

Cloud Vision API:
The Vision API allows developers to easily integrate vision detection features within applications, including image labeling, face, and landmark detection, optical character recognition (OCR), and tagging of explicit content.



Facial Expression	93%
Cartoon	93%
Vertebrate	92%
White	92%
Happy	86%

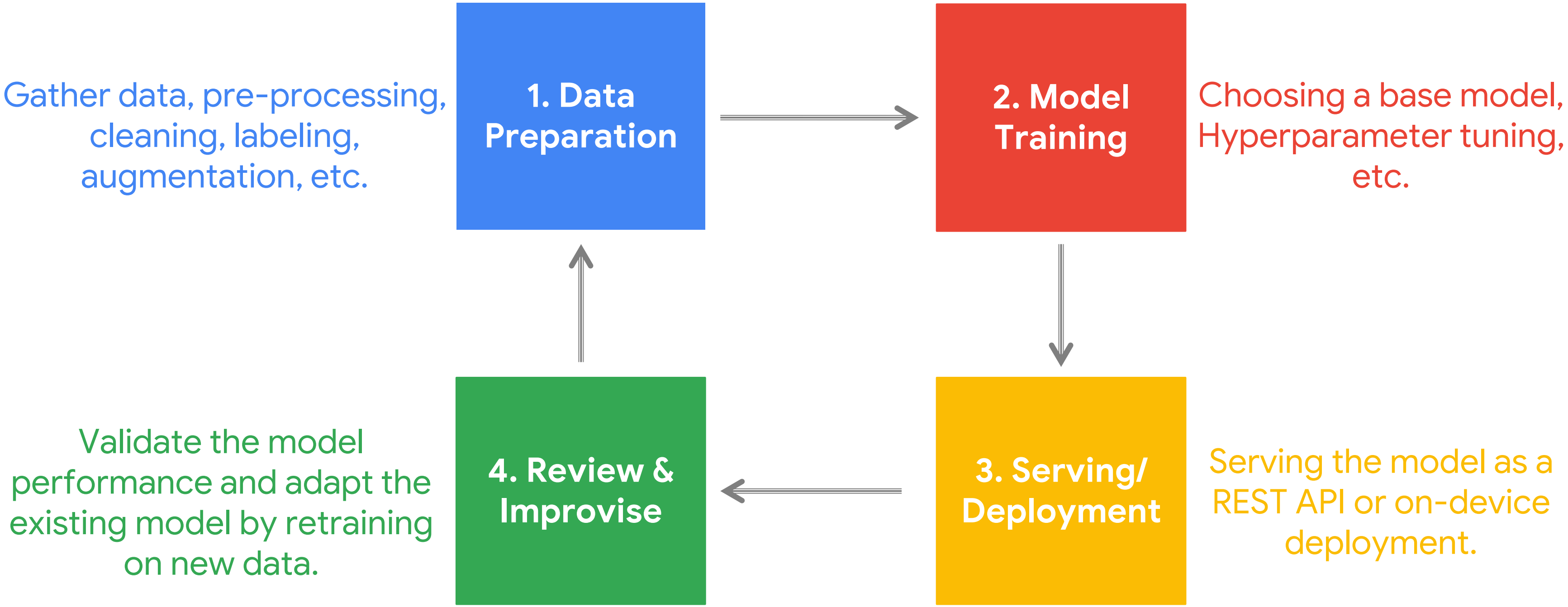
Try it yourself:
https://cloud.google.com/vision/docs/drag-and-drop

Me: feed an image to test
my face recognition model

Model:

The Building Blocks of an ML application

ML Lifecycle



Gather data, pre-processing, cleaning, labeling, augmentation, etc.

1. Data Preparation

2. Model Training

Choosing a base model, Hyperparameter tuning, etc.

3. Serving/ Deployment

Serving the model as a REST API or on-device deployment.

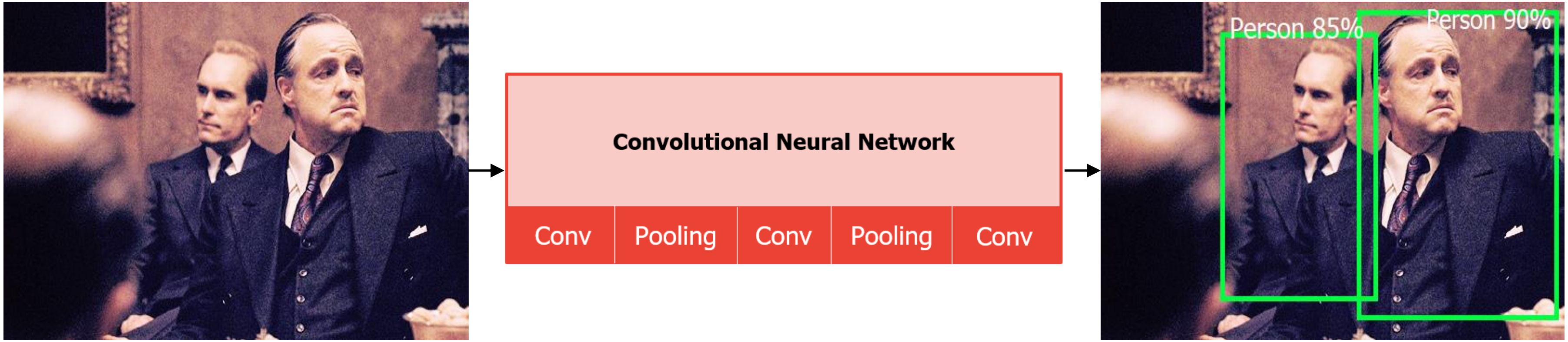
4. Review & Improvise

Validate the model performance and adapt the existing model by retraining on new data.

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



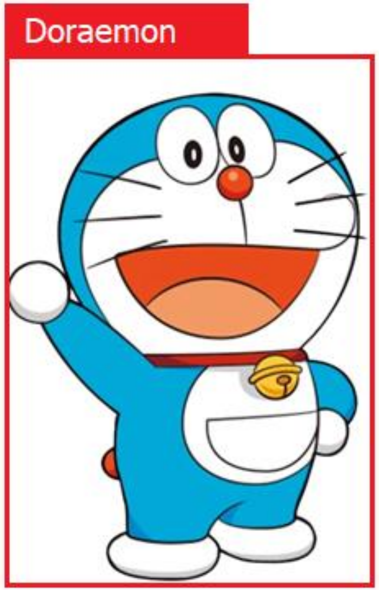
Original Image (input)

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)]

Building your custom object detector

Custom Cartoon Detector



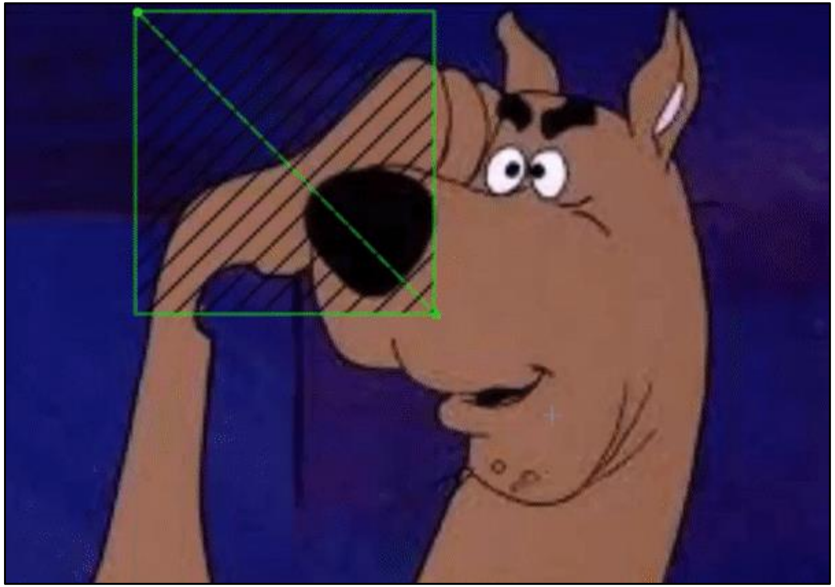
Data Preparation:

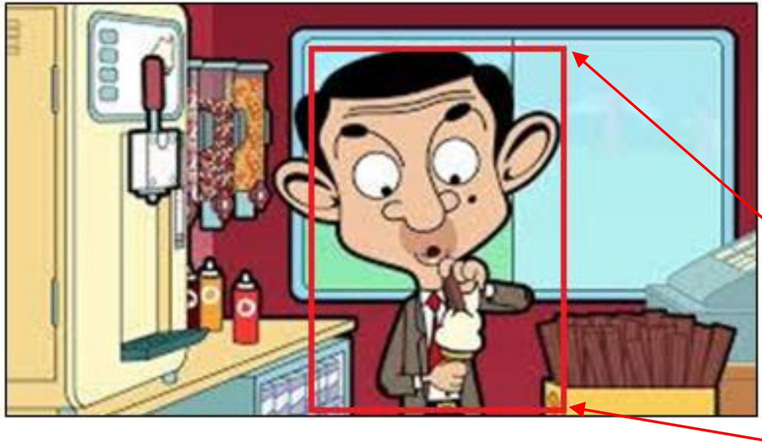
1. **Collection:** Collect data from different sources such as Google Images, web scraping, etc.

2. **Pre-processing:** Cleaning, transformation

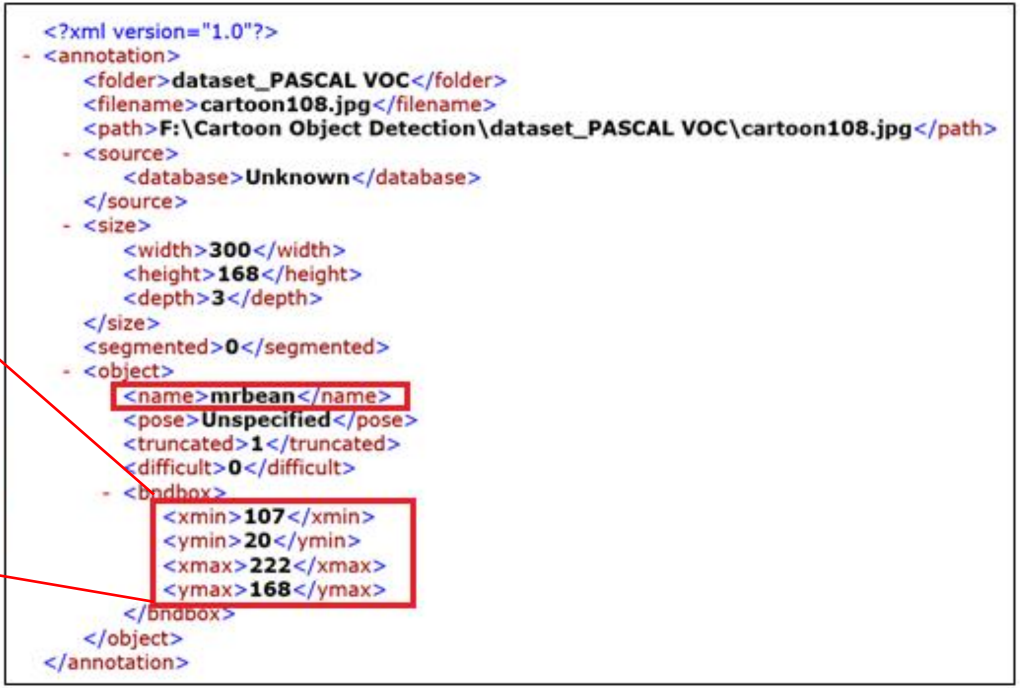
3. **Augmentation:** Generate new data by rotating, scaling, adding/removing noise, etc.

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)

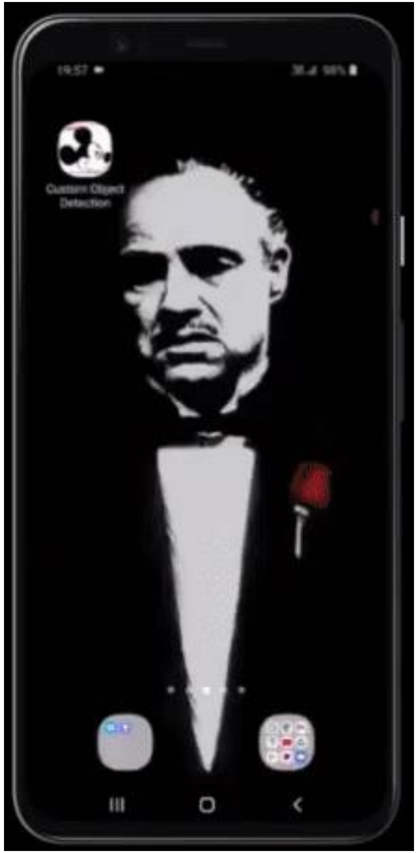
Browser/Web Application
(TensorFlow.js)

Mobile/IoT Devices
(TensorFlow Lite)

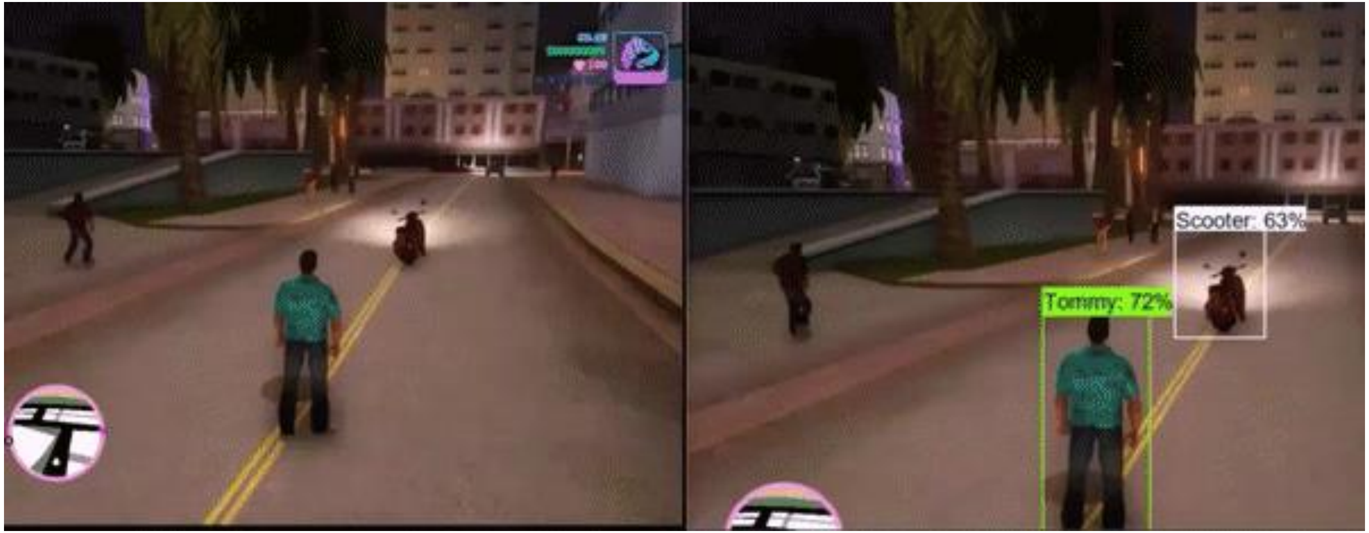
```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_float_model = converter.convert()

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

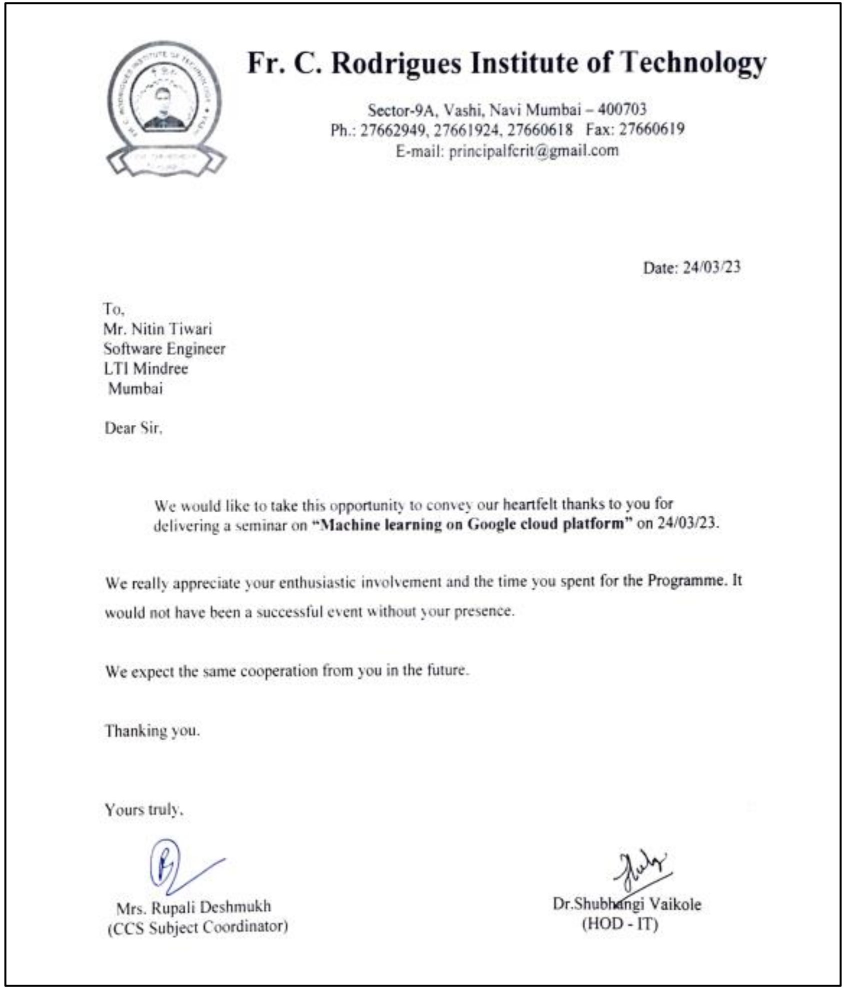


Real-time object detection – GTA Vice City

Hindi Character Recognizer

SO YA
38
GOT ANY QUESTIONS?

Event Highlights



Fr. C. Rodrigues Institute of Technology

Sector-9A, Vashi, Navi Mumbai – 400703
Ph.: 27662949, 27661924, 27660618 Fax: 27660619
E-mail: principalfcrit@gmail.com

Date: 24/03/23

To,
Mr. Nitin Tiwari
Software Engineer
LTI Mindree
Mumbai

Dear Sir.

We would like to take this opportunity to convey our heartfelt thanks to you for delivering a seminar on **"Machine learning on Google cloud platform"** on 24/03/23.

We really appreciate your enthusiastic involvement and the time you spent for the Programme. It would not have been a successful event without your presence.

We expect the same cooperation from you in the future.

Thanking you.

Yours truly,

Mrs. Rupali Deshmukh
(CCS Subject Coordinator)

Dr.Shubhangi Vaikole
(HOD - IT)

```
aspopup=''menu'' hidden='' fixed
avigation'' data-title=''Hide side navigation''
xpanded=''true''><span class=''material-icons
```

Thank you.

github.com/NSTiwari

medium.com/@tiwarinitin1999

Experts

Machine Learning