Detail Project Report

Mask Detection

Revision Number – 1.0

Last Date of Revision : 27– 07 -2022

Balamurali K

Document Version Control

Date	Version	Description	Author
27 – 07 - 2022	1.0	Abstract Introduction General Description	Balamurali
27 – 07 - 2022	1.1	Technical Requirements Data Requirements Data Preprocessing Design Flow	Balamurali
27– 07 - 2022	1.2	Data from User and its validation Rendering the Results Deployment Conclusion	Balamurali

Contents

Abstract

The usage of the mask has become mandatory due to COVID-19 and law also has been enforced for this. Physically monitoring people through cameras in public places is a tedious task for a human for continously staring at the screen. This work discusses the implementation of developing a Object detection model to detect whether a person is wearing a mask or not.

1. Introduction

1.1 Why this DPR Document ?

The main purpose of this DPR documentation is to add the necessary details of the project and provide the description of the machine learning model and the written code. This also provides the detailed description

on how the entire project has been designed end-to-end.

Key points :

- Describes the design flow
- Implementations
- Software requirements Architecture of the project Non-functional attributes like:
- Reusability
- Portability
- Resource utilization

2. General Description

2.1 Problem Perspective

The Mask detection is a object detection model which will detect whether a perosn is wearing a mask or not or a person is wearing incorrectly.

2.2 Problem Statement

Identifying whether a person is wearing a mask or not using a object detection model.

2.3 Proposed Solution

A web app has been developed for this project which takes a image as an input and returns the predictions as a result. The app is dockerized and pushed to dockerhub. Command to pull the image from dockerhub is given below. The object detection model is trained using TFOD 2 (Tensorflow Object detection).

3. Technical Requirements

As technical requirements, we doesn't need any specialized hardware for virtualization of the application. The user should have the device that has the access to the web and the fundamental understanding of providing the input.

3.1 Tools Used

Python programming language and frameworks such as NumPy, Flask used to build the whole model.

- VScode is used as IDE.
- TFOD is used for training the object detection model
- The object detection model is trained in google colab
- Heroku is used for deployment of the model.
- Front end development is done using HTML/CSS.
- Python is used for backend development.
- GitHub is used as version control system.

4. Data Requirements

The dataset is accessible on the Kaggle.

5. Design Flow

5.1 UI Integration

Both CSS and HTML files are being created and are being integrated with the created machine learning model. All the required files are then integrated to the app.py file and tested locally.

5.2 Logging

In logging, at each if an error or an exception is occurred, the event is logged into the system log file with reason and timestamp. These helps the developer to debug the system bugs and rectifying the error.

7. Data from User

The data from the user is retrieved from the created HTML web page.

8. Rendering the Results

The data sent for the prediction is then rendered to the web page.

9. Deployment

The tested model is then deployed to Heroku. So, users can access the project from any internet devices.

10. Conclusion

The Forest Cover Prediction system will help the prediction of type of forest cover based on the observation of catagrophic data.

11. Frequently Asked Questions (FAQs)

Q1) What's the source of data ?

The data for training is taken from kaggle

Q2) What was the type of data ?

The dataset consists of images

Q3) What's the complete flow you followed in this Project ?

Refer Page no 6 for better Understanding.

Q4) How logs are managed ?

We are using different logs as per the steps that we follow in validation and modeling like File validation log, Data Insertion, Model Training log, prediction log etc.

Q5) How Prediction was done ?

We Performed the same life cycle on the provided dataset. Then, on the basis of dataset, model is loaded and prediction is performed. In the end we get the image with predictions

Q6) What are the different stages of deployment?

- First, the scripts are stored on GitHub as a storage interface.

- The model is first tested in the local environment.
- After successful testing, it is deployed on Heroku.