**FACE RECOGNITION WEB APP**

**Web App Description:**

This web app is built in Django Framework using Python Programming Language. It contains 4 different web pages named as – “Home”, “Info”, “Face Recognition” and “Contact Us”.

Home – This is welcome page to greet our website visitors.

Info – It provides an overview on “how to use the web app properly”.

Face Recognition – This page lets the use to Face Recognition Security Access.

Contact Us – This page can be used to send message.

**DJANGO ARCHITECTURE**

Django is based on MVT (Model-View-Template) architecture. MVT is a software design pattern for developing a web application.

MVT Structure has the following three parts –

Model: Model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySql, Postgres).

View: The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/Javascript and Jinja files.

Template: A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

**How Django Web App is built?**

**Step 1) Prepare Virtual Environment**

Virtual Environment is used to isolate the python environment to execute the whole project with custom modules required for that specific Project. It help to avoid collision with version requirement for different projects. And also helps to verify which modules specifically being installed for the project. To create Virtual Environment you need to first install “virtualenv” module using command:

* *pip install virtualenv*

Create Virtual Environment and activate it using this command:

* *virtualenv venv*
* *cd venv*
* *scripts\activate*

Make note that the slash symbol should be ‘\’ and not ‘/’. Then Install Dependencies using pip install.

**Step 2) Create Django Project**

Create Django Project using this command:

* *(venv) django-admin startproject Face-recognition-webapp*

On the running the above command user define face\_recognition\_app Django Project directory will be generated like this:

├── face\_recognition\_app/

| ├── \_\_init\_\_.py

| ├── settings.py

| ├── urls.py

| └── wsgi.py

└── manage.py

**Step 3) Add Django App inside Django Project**

Django is famous for its unique and fully managed app structure. For every functionality, an app can be created like a completely independent module. To create a basic app in Django project, go to directory containing manage.py and from there enter the command:

* *(venv) python manage.py startapp recognition*

**Step 4) Configure the App Setting**

To consider the app in your project you need to specify your project name in INSTALLED\_APPS list as follows in settings.py:

# Application definition

*INSTALLED\_APPS = [*

*'django.contrib.admin',*

*'django.contrib.auth',*

*'django.contrib.contenttypes',*

*'django.contrib.sessions',*

*'django.contrib.messages',*

*'django.contrib.staticfiles',*

*'recognition'*

*]*

Move to face-recognition-webapp -> main -> urls.py and add below code in the header.

*from django.urls import include*

Now in the list of URL patterns, you need to specify app name for including your app urls. Here is the code for it –

*from django.contrib import admin*

*from django.urls import path, include*

*urlpatterns = [*

*path('admin/', admin.site.urls),*

*# Enter the app name in following syntax for this to work*

*path('', include("recognition.urls")),*

*]*

**Step 5) Add Template and Static Files**

Templates are the third and most important part of Django’s MVT Structure. A template in Django is basically written in HTML, CSS and JavaScript in an .html file. Django framework efficiently handles and generates dynamically HTML web pages that are visible to end-user. Django mainly functions with a backend so, in order to provide frontend and provide a layout to our website, we use templates. There are two methods of adding the template to our website depending on our needs.

We can use a single template directory which will be spread over the entire project.

For each app of our project, we can create a different template directory.

For our current project, we will create a single template directory which will be spread over the entire project for simplicity. App-level templates are generally used in big projects or in case we want to provide a different layout to each component of our webpage.

Configuration

Django Templates can be configured in app\_name/settings.py,

*TEMPLATES = [*

*{*

*# Template backend to be used, For example Jinja*

*'BACKEND': 'django.template.backends.django.DjangoTemplates',*

*# Directories for templates*

*'DIRS': ['template'],*

*'APP\_DIRS': True,*

*# options to configure*

*'OPTIONS': {*

*'context\_processors': [*

*'django.template.context\_processors.debug',*

*'django.template.context\_processors.request',*

*'django.contrib.auth.context\_processors.auth',*

*'django.contrib.messages.context\_processors.messages',*

*],*

*},*

*},*

*]*

Then create a template in templates folder using Django Format style.

**Step 6) Create Views in Django App**

Views are used to render the templates added in template folder. Similar to these:

*def info(request):*

*return render(request,"info.html")*

*def base(request):*

*return render(request,"base.html")*

*def contactus(request):*

*return render(request,"contactus.html")*

*def upload\_image(request):*

*context = {*

*"image": "static/image/empty-image.png",*

*}*

*return render(request,"uploadimage.html",context=context)*

**Step 7) Create URLs in Django App**

Urls are used to manage the url paths like which link will show which url and so on. Similar to these:

*urlpatterns = [*

*path('', base,name="base"),*

*path('info', info, name="info"),*

*path('contact', contactus,name="contactus"),*

*path('upload-image', upload\_image,name="upload-image"),*

*path('access-grant', recognition,name="access-grant"),*

*]*

**How to Run the Django App?**

To run the App you just need to execute command:

* *python manage.py runserver*

Then visit <http://localhost:8000/>

**Face Recognition Model Working:**

Web app is using recognition function (defined inside recognition/views.py) to execute the Face Recognition Models in backend.

Face Recognition Architecture codes are inside recognition/Face\_Recognition\_Model/ folder. It contains inception\_resnet\_v1.py and mtcnn.py Architecture.

MTCNN Model is used to perform Face Detection part by detecting the faces in user uploaded image.

Inception Resnet is being used to perform Face Recognition Part by measuring the Euclidean Distance been uploaded face image and database known faces.

Pipeline.py

This code include the Face Recognition task. It used Both MTCNN and Inception ResNet to build pipeline to build the working model.

**ALGORITHM:**

1. Defining MTCNN Model
2. Defining Inception Resnet Model
3. Using MTCNN model to detect face from all known faces inside the Web app.
4. Embedding the Known Image dataset by using ResNet Model
5. Detecting face in User Uploaded Unknown Image file.
6. Embedding the unknown image using ResNet.
7. Now applying Face Matching using Tolerance Threshold of 0.75

*If the Distance is smaller than 0.75 Euclidean Distance then the face Matches with database known Faces.*

*Else Access Denied.*

READ THE PIPELINE.PY CODE FILE TO UNDERSTAND IT IN MORE DETAIL.