Pre-trained Model (Place this is same place as app.py)-
https://drive.google.com/file/d/1637IFFuLmoKjuaxNRNqh4uB70PVxmuCH/view?usp=sharing
Place this in color folder -
https://drive.google.com/file/d/123oQ6jLAHUhy0PVSvJ6pXDLgLNdEwKNM/view?usp=sharing
Place this in gray folder -
https://drive.google.com/file/d/1h_CyPwR0e5MJdDEMe0-grMa0aicQWLsc/view?usp=sharing
Place this in lstm folder -
https://drive.google.com/file/d/1mTGQB9uZcMsyIaVAU2PeT_Cq5-dZMA84/view?usp=sharing

Steps to Run in Linux or Mac

1. Clone/Download this repository.
2. Download the model and store in the current folder.(Same place where app.py is there)
3. Open terminal in the downloaded folder, perform pip install -r requirements.txt
4. Execute app.py in terminal
5. You will get a link - http://127.0.0.1:4001/
6. Go to that link, each api is performed using buttons.

Documentation for APIs

1. http://127.0.0.1:4001/extractfaces

Input - `request.files.getlist['file'] & request.form.get("personID")`
Output -{“imagelist”: path for each image, “array” : extracted faces} ***For images***
{“imagelist”: path for each image, “array” : extracted faces, “heading”:details about the faces (multiple_face/one_face/no_face)} ***For videos***

Example - a. Upload required file(photo/video)
b. Route to the above link.
c. Returns a JSON as the above output format.

2. http://127.0.0.1:4001/countfaces

Input - `request.files.getlist['file'] & request.form.get("personID")`

Output -{“number of faces”: count of faces, “path” :path to each face, “array”: displays the faces}

Example - a. Upload required file(s) [Images]
b. Route to the above link.
c. Returns a JSON as the above output format.

3. http://127.0.0.1:4001/addface

Input - `request.files['file'] & request.form.get("personID")`

Output - {“extracted faces”:faces extracted, “heading”: encoding added/not, “encodings”: encoding of the face found} ***For images***

“extracted faces”:faces extracted, “heading”: encoding added/not, “encodings”: encoding of the face found} ***For video***

Example - a. Upload required file(photo/video)
b. Route to the above link.
c. Returns a JSON as the above output format.

4. http://127.0.0.1:4001/recognize

Input - `request.files['file']`

Output - {“imagepath”: path of all recognized face,“Predicted”: Name of the person, “probability”: the probability of recognized face} ***For images***

{“imagepath”: path of all recognized face,“Predicted”: Name of recognized face, “probability”: the probability of identified face} ***For videos***

Example - a. Upload required file(photo/video)
b. Route to the above link.
c. Returns a JSON as the above output format.

5. http://127.0.0.1:4001/removeencodings

   Input - `request.files['file']`

   Output - {“Removed encodings”:Number of encodings removed}

   Example - a. Upload required photo(s)

   b. Route to the above link.

   c. Returns a JSON as the above output format.

6. http://127.0.0.1:4001/webcam

   Input - `Webcam gets activated`

   Output - {“imagepath”: path of all recognized face,“Predicted”: Name of recognized face, “probability”: the probability of identified face}

   Example - a. Route to the above link.

   b. Returns a Response type in the above output format.

7. http://127.0.0.1:4001/removePersonID

   Input - `request.form.get("personID")`

   Output - {“ID”: ID of the removed person}

   Example - a. Enter the person ID

   b. Route to the above link.

   c. Returns a JSON as the above output format and deletes the corresponding encodings.

8. http://127.0.0.1:4001/getAllFaceEncodings

   Input - `request.form.get("personID")`

   Output - {“encodingslist”: list of all the encodings of the person}

   Example - a. Enter the person ID

b. Route to the above link.

c. Returns a JSON as the above output format