Code Structure

The main Code Structure tree is shown below:

```
    - FaceReco
      | - FaceRecoApi
      |   | - __init__.py
      |   | - api.py
      |   | - process_unknown_face.py
      | - faces.py
      | - FaceRecoMain.py
      | - run.sh
      | - setup.sh
```

setup.sh

```bash
#! /bin/bash

install_dependencies="pip install --no-cache-dir -r requirements.txt"

read -p "Enter name of the virtual enviornment to create : " dir

if [ -d "$dir" ]; then
    echo "The directory named '$dir' exists in this current folder."

    sleep 1

    echo -e "Attempting to activate detected virtual enviornments...\n\n"

    source "$dir"/bin/activate

    output=$(pip freeze)

    # Check if the output variable is empty or not
    echo -e "Validating dependencies..."
    if [ -z "$output" ]; then
    sleep 1
    echo -e "Missing dependenciess\nInstalling Dependencies... \n\n"

    sleep 1

    eval $install_dependencies
    else
        echo -e "Dependencies already installed!\nQuitting now...\n\n"
    fi

else
    echo -e "Creating venv with name $dir\n\n"
    python -m venv "$dir"

    echo -e "Activating the venv\n\n"

    sleep 1

    source venv/bin/activate

    echo -e "Now installign required packages, please wait...\n\n"

    sleep 1

    eval $install_dependencies

    sleep 1

fi


#Installing OPENBLAS

if [ -d "OpenBLAS" ]; then

    echo -e "OPENBLAS ALREADY INSTALLED\nSkipping..."
else
    echo -e "Installing OpenBLAS....\n"

    git clone https://github.com/xianyi/OpenBLAS.git

    cd OpenBLAS

    make

    sudo make PREFIX=/usr/local/lib install

    cd ..
fi
mkdir Unknown_Faces
# Adding OPENBLAS to path
line_to_check='export LD_LIBRARY_PATH="/usr/local/lib:PATH"'

bashrc_file="$HOME/.bashrc"
if grep -qF "$line_to_check" "$bashrc_file"; then
    # echo "The line already exists in $bashrc_file."
    echo -e "OPENBLAS path already exist in ~/.bashrc\nSkipping"
else
    echo -e "The OPENBLAS path does not exist in $bashrc_file.\nAdding...."
    echo -e "\nexport LD_LIBRARY_PATH=\"/usr/local/lib:PATH\"" >> ~/.bashrc
fi



echo ",,,Exiting,,,"
```

run.sh

```
#! /bin/bash

source venv/bin/./activate

# Decalring all global variables
count=0
run_loop=true
start_time=$(date +%s)      # Storing starting timestamp

ran_face_recognition=0
ran_add_face=0
ran_list_all_faces=0
ran_delete_face=0


# Calculate run time of the program
run_time() {
    end_time=$(date +%s)
    duration=$((end_time - start_time))

    if (( duration < 60 )); then
        echo "Script ran for $duration seconds."
    elif (( duration < 3600 )); then
        minutes=$(( duration / 60 ))
        seconds=$(( duration % 60 ))
        echo "Script ran for $minutes minutes and $seconds seconds."
    else
        hours=$(( duration / 3600 ))
        minutes=$(( (duration % 3600) / 60 ))
        seconds=$(( duration % 60 ))
        echo "Program ran for $hours hours, $minutes minutes, and $seconds
seconds."
    fi
}

run_status()
{
    run_time
    echo "Ran face recogmition tests : $ran_face_recognition times"
    echo "Ran add face : $ran_add_face times"
    echo "Ran list all faces : $ran_list_all_faces times"
    echo "Ran delete face : $ran_delete_face times"
}
# Function to run face recognition
run_face_recognition() {
    ((ran_face_recognition += 1))
    echo_instructions
    echo "Running Face Recognition..."
    python FaceRecoMain.py
    clear
    echo_instructions
```

```
}

# Function to add a new face
add_new_face() {
    ((ran_add_face += 1))
    echo_instructions

    echo -e "USAGE :\n\tTo add a new known face : python faces.py -a
<image_path>\n \tOR"
    echo -e "\tpython faces.py -ak <image_path>\n"
    echo -e "\tTo add a new unknown face :  python faces.py -au <image_path>\n"
    custom_command="python faces.py -a "
    echo -e "\tpress tab to search for the file"
    echo -e "\n|---------------------------------------------|\n"
    read -e -i "$custom_command" modified_command
    eval "$modified_command"
}

# Function to see all listed faces
see_all_faces() {
    ((ran_list_all_faces += 1))
    echo_instructions
    echo "Listing All Faces..."
    python faces.py -l
    # Add your code to display all listed faces here
}

# Function to delete a face
delete_face() {
    ((ran_delete_face++))
    echo_instructions
    custom_command="python faces.py -d "
    echo "write the name of the face you want to delete at the end"
    echo -e "\n|---------------------------------------------|\n"
    read -e -i "$custom_command" modified_command
    eval "$modified_command"
    # Add your code to delete a face here
}

echo_instructions(){
    echo "███████╗ █████╗  ██████╗███████╗██████╗ ███████╗ ██████╗ ██████╗  █████╗ 
██████╗ ██╗"
    echo
"██╔════╝██╔══██╗██╔════╝██╔════╝██╔══██╗██╔════╝██╔════╝██╔═══██╗██╔══██╗██╔══██╗
██║"
    echo "█████╗  ███████║██║     █████╗  ██████╔╝█████╗  ██║     ██║   
██║███████║██████╔╝██║"
    echo "██╔══╝  ██╔══██║██║     ██╔══╝  ██╔══██╗██╔══╝  ██║     ██║   
██║██╔══██║██╔═══╝ ██║"
    echo "██║     ██║  ██║╚██████╗███████╗██║  ██║███████╗╚██████╗╚██████╔╝██║  
██║██║     ██║"
    echo "╚═╝     ╚═╝  ╚═╝ ╚═════╝╚══════╝╚═╝  ╚═╝╚══════╝ ╚═════╝ ╚═════╝ ╚═╝  
╚═╝╚═╝     ╚═╝"


    echo -e "\n\n"
    echo "Please select an option:"
    echo "1. Run Face Recognition"
    echo "2. Add a new Face"
    echo "3. See All known faces"
    echo "4. Delete a Known Face"
    echo "5. Clear the current screen"
    echo "q. Quit the program"
    echo -e "\n\t"
}

# Main function to present the options and call corresponding functions
main() {
    if [ "$count" -eq 0 ]; then
        clear
        echo_instructions
    fi


    read -p "Enter your choice (1-5): " choice

    case $choice in
        1)
            clear
            run_face_recognition
            ;;
        2)
            clear
            add_new_face
            ;;
        3)
            clear
            see_all_faces
            ;;
        4)
            clear
            delete_face
            ;;
        5)
            clear
            echo_instructions
            ;;
        q)
            run_loop=false
            clear
            deactivate
            echo_instructions
            echo "Program terminated..."
            run_status
            ;;
        *)
            clear
            echo_instructions
            echo "Invalid choice. Please enter a number between 1 and 5."
            ;;
    esac
}

# Call the main function to start the program
while $run_loop
do
    main
    ((count++))
done
```