Image classification from video
Project Overview

Business Overview

A field of artificial intelligence which trains computers to interpret and understand the visual world is known as computer vision. Computer vision tries to replicate the functions of the human vision system to identify and process different objects in images or in videos. It is one of the most powerful techniques which has been used extensively to detect and label objects in images or in videos. The recent inventions in the field of artificial intelligence have been able to outshine humans in detecting and labeling objects.
Image transformation, Medical image analysis, Human pose estimation are among the numerous uses of computer vision. Object Classification, Identification, Verification, Detection, Landmark Detection, Segmentation, Recognition are some of the most popular computer vision applications. In recent years Face recognition which tries to identify faces of a person given an image or video and then classifying them has gained popularity. Nowadays a variety of phones come with a face unlocking feature which uses face recognition in the backend. One of the popular social media applications, Facebook is trying to make its algorithm more and more robust where faces will be identified and tagged when multiple people are there in an image. When you enable your camera you see a box surrounding a face, that is a face recognition task. Entering in a company premises also requires a biometric impression which is popular still some are trying to go for face recognition.
The objective of this case study is to understand how to extract frames from a video and train using faces and identify where the classified person is located either in a video or an image. Challenges to overcome are mainly due to inefficient capturing of a face from a person's image and classifying it correctly if located. Our main objective is to extract weights/embeddings from pre-trained weights/models and classify them using a Machine Learning or a Deep Learning Model Technique using images.

Aim

To extract faces out of images and identify/classify a person's face in images and videos.

Data Description

In this project we have used a video from a popular sitcom show Friends. The video can be found on https://www.youtube.com/watch?v=NzOTuh63eVs. The characters Rachel, Chandler, Phoebe, Monica and Ross are taken from the video. By extracting

frames per second and then extracting faces from each frame we get images which are train and test dataset. There are 35 images in total consisting of 7 images per person in the training dataset and 15 images in total for the test dataset.

Tech Stack

➔ Language : Python (Version 3.6.2)
➔ Libraries : OpenCV, scikit-learn, numpy, os, pytube, scikit_image, skimage, keras, tensorflow

- OpenCV for loading images and videos
- Haar Cascade Algorithm for face extraction
- Extracting Embeddings from a pre-trained FaceNet Model
- Convolution Neural Networks for training purpose
- Dense Neural networks to classify images
- VGGFace Architecture
- Matplotlib for plotting images
- Numpy for array and math operations
- Tensorflow and keras for loading and building models
- Skimage for resizing images
- Imutils for video reading
- Os for operating folders

Approach

- Downloading video from youtube using python
- Extracting frames per second from video
- Face extraction using CV
- Modifying every image as per Facenet model requirements
- Extracting embedding and normalizing as per the model requirements
- Fitting SVm model and predictions on test data
- Visualizing normal embedding using TSNE
- Model testing on frames and predicting faces
- Prediction on videos
- Creating VGG Face model architecture
- Data preparation for modelling
- Model training and predictions

Overview

1. Each face from a respective folder is read and resized using the required format for the model requirements.
2. Then pixels extracted are required to go certain pre-processing in order to extract meaningful information from weights or from a model.

3. Once the required embeddings or weights are extracted, these are to go from a ML/DL model depending on the requirement.
4. Faces from a frame or a video are to be extracted using Haar Cascade Object Detection for face extraction.
5. Model training is used to identify faces extracted from Haar Cascade and label if it meets certain thresholds and others.
6. Person's name will be labeled in frames or in a video.

Modular code overview

There are two modular codes in this project. One for facenet model and other for vgg model. Both modular code are described below.

Following image shows modular_code_facenet structure.

```
input
  |_ frames_path
  |_ test_frames
  |_ train
  |_ val
  |_ video

lib
  |_ pre_trained_facenet.ipynb

output

prebuilt_models
  |_ facenet_keras.h5
  |_ haarcascade_frontalface_alt2.XML

src
  |_engine.py
  |_ ML_pipeline
          |_ create_new_folder.py
          |_ download_video.py
          |_ embedding_encoding.py
          |_ extracting_frames.py
          |_ facenet_prediction.py
          |_ image_modification.py

requirements.txt
```

Following image shows the modular_code_vgg structure.

```
input
  |_ frames_path
  |_ test_frames
  |_ train
  |_ val
  |_ video

lib
  |_ deep_learning_vgg_prediction.ipynb
  |_ training_using_google_colab.ipynb

output

prebuilt_models
  |_ character_mapping.pkl
  |_ face_classifier_model.h5
  |_ haarcascade_frontalface_alt2.XML
  |_ saved_model.pb
  |_ vgg_face_weight.h5

src
  |_engine.py
  |_ ML_pipeline
            |_ create_new_folder.py
            |_ data_prep_vgg.py
            |_ download_video.py
            |_ extracting_frames.py
            |_ face_classifier.py
            |_ image_modification.py
            |_ vgg_architecture.py
            |_ vgg_prediction.py

requirements.txt
```

Once you unzip the modular_code.zip file you can find the following folders within it.

1. input
2. src
3. output
4. iib
5. prebuilt_models
6. requirements.txt

1. The input folder contains all the data that we have for analysis. The train and val folders contain training and validation files. The video folder has downloaded video from youtube. The test frames folder contains the frames on which model predictions are generated. The frames_path contains all the frames extracted from the video.

2. The src folder is the heart of the project. This folder contains all the modularized code for all the above steps in a modularized manner. It further contains the following.
    a. ML_pipeline
    b. engine.py

The ML_pipeline is a folder that contains all the functions put into different python files which are appropriately named. These python functions are then called inside the engine.py file

3. The output folder contains the predicted frames by using the test frames from the input folder.

4. The lib folder is a reference folder. It contains the original ipython notebook that we saw in the videos.

5. The prebuilt_models folder contains all the models which are used for training. It also contains a fitted model that we trained for this data. This model can be easily loaded and used for future use and the user need not have to train all the models from the beginning.
6. The requirements.txt file has all the required libraries with respective versions. Kindly install the file by using the command pip **install** -r **requirements**.**txt**

Project Takeaways

1. Downloading video from youtube using python
2. loading a video and extract frames out of a video
3. Iterating through folders and perform the necessary operations
4. Extract faces from images
5. Facenet model
6. Haar cascade model
7. Making use of pre-trained models
8. Modifying images as per Facenet model requirements
9. Training a Machine Learning Model on Embeddings
10. Visualizing normal embedding using TSNE
11. Classify and label persons' faces in images
12. Identifying persons' faces in video
13. Use of Google Colab for training purpose
14. Convolutional Neural Networks
15. Making a popular face recognition deep learning architecture and using pre existing weights of VGG Face model
16. Data preparation for modelling
17. Model training and predictions