10/10/2018

Unsupervised Face Clustering Pipeline

Christ Prateek Prasanna Xaxa

Xaxa, Prateek

Problem Statement

Live face-recognition is a problem that automated security division still faces. With the advancements in Convolutions Neural Networks and specifically creative ways of Region-CNN, it's already confirmed that with our current technologies, we can opt for supervised learning options such as FaceNet, YOLO for fast and live face-recognition in a real-world environment.

To train a supervised model, we need to get datasets of our target labels which is still a tedious task. We need an efficient and automated solution for the dataset generation with minimal labeling effort by user intervention.

Proposed Solution

Introduction

We are proposing a dataset generation pipeline which takes a video clip as a source and extracts all the faces and clusters them to limited and accurate sets of images representing a different person. Each set can easily be labeled by human input with ease.

Technical Details

We are going to use opencv lib for per second frames extraction from input video clip.

We will use face_recognition library (backed by dlib) for extracting the faces from the frames and align them for feature extractions.

We will extract the human observable features and cluster them using DBSCAN clustering provided by scikit-learn.

For the solution, we will crop out all the faces, create labels and group them in folders for users to adapt them as a dataset for their training use-cases.

Challenges in implementation

For a larger audience, we plan to implement the solution for execution in CPU rather than an NVIDIA GPU. Using an NVIDIA GPU may increase the efficiency of the pipeline.

CPU implementation of facial embedding extraction is very slow (30+ sec per images). To cope up with the problem, we implement them with parallel pipeline executions (resulting in ~13sec per image) and later merge their results for further clustering tasks.

We introduce tqdm along with PyPiper for progress updates.

We introduce the resizing of frames extracted from input video for smooth execution of pipeline.

Troubleshooting

1. The whole pc freezes when extracting facial embedding

The solution is to decrease the values in frame resize function when extracting frames from input video clip. Remember, decreasing the values too much will result improper face clustering. Instead of resizing frame, we can introduce some frontal face detection and clip out the frontal faces only for improved accuracy.

2. The pc becomes slow while running the pipeline

The cpu will be used at maximum level. To cap the usage, you can decrease the number of threads specified at pipeline constructor.

3. The output clustering is too much inaccurate

The only reason for the case can be the frames extracted from the input video clip will have very faces with a very small resolution. Kindly get a video clip with bright and clear images of faces in it.

References

1. Adrian blog post – face clustering
2. PyPiper guide
3. OpenCV manual
4. StackOverflow

Github link

https://github.com/cppxaxa/FaceRecognitionPipeline_GeeksForGeeks