ANU Tech Launcher Project Cinema

User Manual

Overview:

This project developed an algorithm for Cinefly to use machine learning technology to extract key information from videos and build user profiles.

This project adopts agile development management method and is divided into two development stages. This user manual provides an explanation of the output of the second phase of the project.

Code structure:

In the second stage of development, the team produced a total of 5 demos. Among them, demo 1 and demo 2 are both early iterative versions of demo 3. Demo 4 is a technical route exploration that proved to be a failure. Demo 5 is a successful alternative to demo 4 and can provide inspiration for the follow-up work of this project.

The files of the project code package include:

Demo 1: DemoV1.py

Demo 2: DemoV2_part1.py, DemoV2_part2.py

Demo 3: DemoV3_part1.py, DemoV3_part2.py

Demo 4: DemoV4.py

Demo 5: DemoV5.py

Demo 3's CV detection module (from the first stage of this project): age_deploy.prototxt, age_gender_detect.py, age_net.caffemodel, gender_deploy.prototxt, gender_net.caffemodel, opencv_face_detector.pbtxt, opencv_face_detector_uint8.pb, MIT License for age & gender part

Other files: mask_rcnn_coco.h5 (pre-trained model used in Demo 4), README.md

Output example:

The Google key used by each demo was not included in the final delivery. You need to download the service key of Google Cloud, rename it to "GCKey.jason", and place it in the project directory.

Functions and usage description:

Demo 1 extracts the voice in a single input video and converts it into text. Demo 1 has been encapsulated as a python function and used in demo 3. So you don't have to run demo 1 separately.

Demo 2 is an early iterative version of demo 3. Run DemoV2_part2.py to run this demo. It can parse the voice information of all videos in two given folders and save it in PersonInfo.csv. You can adjust the location of the two input directories in DemoV2_part2.py.

PersonInfo.csv is a file that includes file name, user name, age (CV detection), gender (CV detection) and voice text.

After getting PersonInfo.csv, you need to manually log in to Amazon cloud service and select the custom model entity detection service in Amazon "comprehend" to extract the tags you are interested in. For specific operations, please refer to the Amazon comprehend documentation.

Demo 3 is the main delivery object of this project. It can process videos from different users and stored in different directories, extract information from different videos of the same user, and implement information integration. Before running this demo, please make sure that you save the original video in the correct format: first you should create a folder named "video_cut" in the project directory, and in this folder, please press the "shot_#" format , Create folders for eight different shots in the storyboard, such as "shot_1". Finally, please store the corresponding shot video in the corresponding directory.

After saving the original video correctly, you should run DemoV3_part1.py to get All_Txt.csv first. This file contains a user's voice information in all shots. Then please follow the similar steps as in demo2, log in to Amazon "comprehend", extract the entities you are interested in, and get Amazon's return file "output".

Next, please modify the two input directories in DemoV2_part2.py to "shot_1" and "shot_7" in "video_cut" to obtain PersonInfo.csv. At this time, three files, PersonInfo.csv, output and All_Txt.csv, should exist in your directory at the same time. At this time, please run DemoV3_part2.py. This script will parse the output into a readable file and combine the above three files to generate a Final.csv file which will contain all the information you are interested in.

Demo 4 and demo 5 are both attempts to further improve the accuracy of demo 3. Demo1-3 all use NLP technology. And demo 4 and demo 5 try to use CV technology to identify and classify all objects in the video. Among them, the Mask_RCNN technology used in demo 4 was proved by the team to require extremely high computing power, so this technical route was abandoned.

Demo 5 uses Google Vision API. The video source storage method it requires is the same as that of demo 3. The final output uid_shots.csv file will contain all the detected objects in the video. In Demo 5, the parameters you can adjust are the probability threshold in runDemoV5() and the number of frames uploaded for each video interception in processSingleVideo(). (Due to cost and necessity considerations, we cannot upload all the frames in the video)

If you need technical support, please contact our team! Thank you very much!