EmoTV

-Reading through emotions

-ITSP Documentation

Team name: VCare

Team members:

1) Rathod Harekrissna Upendra - 17D070001
   rathodhare@gmail.com

2) Adhishree Apte - 170110040
   apteadhishree@gmail.com

3) Parth Milind Shettiwar - 170070021
   parthisultimate@gmail.com

4) Ayushraj Mehrotra - 17D070034
   ayushrajmehrotra.11@gmail.com

Contents

1 EmoTV Overview:

As the name suggests, EmoTv is capable of interacting with a person and determine his current mood. Running on python coded lines, it can efficiently involve in a conversation with a subject. Our TV features playing songs alongside the emotion recognition part. As an added bonus, it shows up a cartoonized image of yours with your name written on bottom right of screen and displays your emotion. We have also added image magick gui alongside the cartoonized image for further editing of the image. The entire project is completely based on python scripts, displaying the vastness of the language. Our project covers topics ranging from machine learning using Tensorflow to Speech recognition and conversion. Our EmoTV Project is a step towards making robots understand human emotion, thus lessening the gap of human machine interaction.

2 Project Details

The following things take place when running EmoTV:

- The system runs with a python script. Firstly it welcomes the person and ask the name of the subject.
- Correctly converting speech to text and then greets it with his name and asks to read his emotions .
- Within few seconds it predicts the person's emotions through an image taken from laptop's webcam using machine learning through Tensorflow.
- It then shows up a cartoonized image of the person, a cute emoji of his emotion and his name at the lower bottom of screen (Which can be taken by the person for his/her memory of EmoTV).
- Now suppose if the subject is sad, EmoTV then plays a song which will refresh the subject altogether.

3 Libraries and Packages used

- **Pyaudio, wave**: For opening and playing a wav file through python
- **OpenCV**: For developing a cartoonized image of the person
- **Tensorflow**: For predicting the emotion of the subject
- **gTTS**: For using the google text to speech api
- **Speech recognition**:For using google api of speech to text
- **Sys**: For manipulating (Transferring from one place to other, saving and deleting the audio files, etc) the audio files while the main program is running
- **Os**: For converting .mp3 files to .wav files
- **Pillow**: For converting text to Image containing the same text
- **math, numpy, random**: For emotion recognition processes.

4 Theory:

The project revolves mainly around correctly predicting the emotion of the subject. This was done using Tensorflow. The machine learning aspect of the project has mainly two parts:

- CNN Network
- Dataset

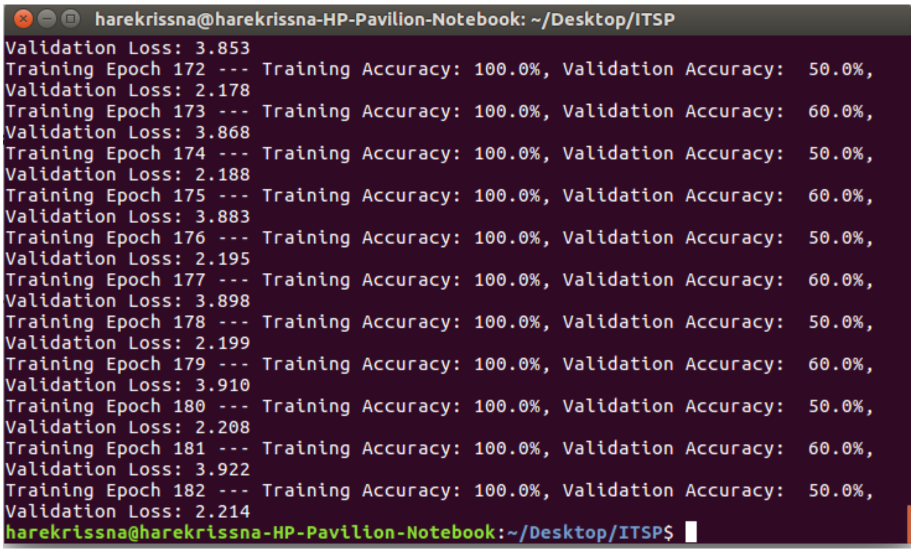
4.1 CNN Network

Following is the basic overview of our neural network:

- The network contains 3 convo layers taking inputs from preceding layers
- A flattening layer taking input from convo 3.
- 2 fully connected layers of which first doesn't have Relu while the second one has .
- Finally a softmax layer which correctly outputs one of the three emotions(happy ,sad or surprised).
- Each conv layer has the basic three layers of conv, pool and a Relu layer.
- Adam optimizer was used for training on dataset.

4.2 Dataset

A two day long search was made to find a proper dataset either labelled or unlabelled. Unfortunately the search failed. About 60 datasets from all over the world were tried but either they were licensed or did not satisfy our requirements. So we had to resort to manually collecting the dataset. To get to as close to the real test image, we tried to include Indian images which all were available on Net. With some time taken ,a 250 image dataset was collected and fed to the network. Although the dataset is not enough, we tried our best to modify our training parameters to accurately predict the subjects emotion. To avoid high variance in training ,we kept our network small. The training set is divided into two parts - 1)Train set and 2)Validation set. With 80:20 ratio respectively. Since the dataset was small, although train accuracy was 100% , validation accuracy turns out to be mere 50%.


harekrissna@harekrissna-HP-Pavilion-Notebook: ~/Desktop/ITSP

```
Validation Loss: 3.853
Training Epoch 172 --- Training Accuracy: 100.0%, Validation Accuracy:  50.0%,
Validation Loss: 2.178
Training Epoch 173 --- Training Accuracy: 100.0%, Validation Accuracy:  60.0%,
Validation Loss: 3.868
Training Epoch 174 --- Training Accuracy: 100.0%, Validation Accuracy:  50.0%,
Validation Loss: 2.188
Training Epoch 175 --- Training Accuracy: 100.0%, Validation Accuracy:  60.0%,
Validation Loss: 3.883
Training Epoch 176 --- Training Accuracy: 100.0%, Validation Accuracy:  50.0%,
Validation Loss: 2.195
Training Epoch 177 --- Training Accuracy: 100.0%, Validation Accuracy:  60.0%,
Validation Loss: 3.898
Training Epoch 178 --- Training Accuracy: 100.0%, Validation Accuracy:  50.0%,
Validation Loss: 2.199
Training Epoch 179 --- Training Accuracy: 100.0%, Validation Accuracy:  60.0%,
Validation Loss: 3.910
Training Epoch 180 --- Training Accuracy: 100.0%, Validation Accuracy:  50.0%,
Validation Loss: 2.208
Training Epoch 181 --- Training Accuracy: 100.0%, Validation Accuracy:  60.0%,
Validation Loss: 3.922
Training Epoch 182 --- Training Accuracy: 100.0%, Validation Accuracy:  50.0%,
Validation Loss: 2.214
harekrissna@harekrissna-HP-Pavilion-Notebook:~/Desktop/ITSP$
```

4.3 Files

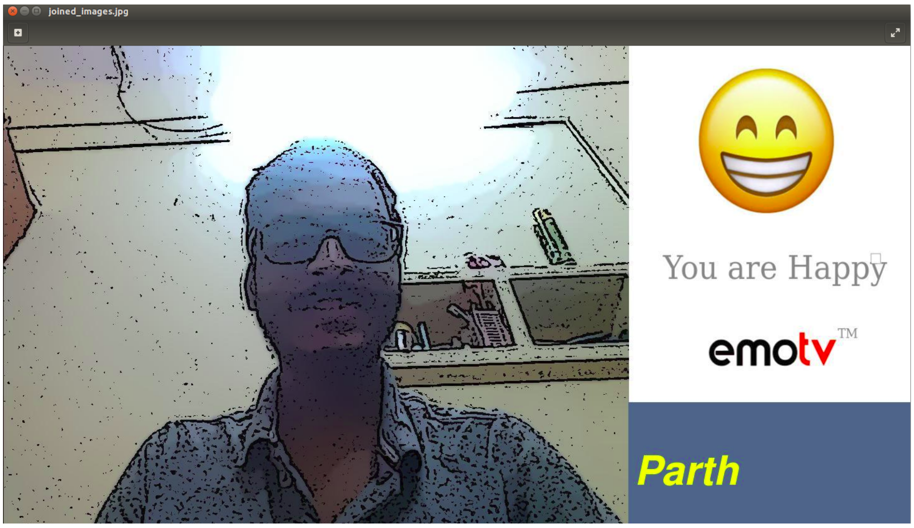
The whole machine learning part of the project was written down in three python scripts -namely dataset.py, train.py and predict.py. The train.py takes data from dataset.py which reads through all images. The network then trains on the images and stores its weights in file called checkpoint .The predict.py then restores the weights from checkpoint and reads an image which is captured from the laptop webcam. It then computes the ‘result’ which outputs the probability of each emotion. If probability of some emotion is more than 50% and that emotion is then taken as the expected emotion and outputted to the screen.

5 Other important aspects of the project

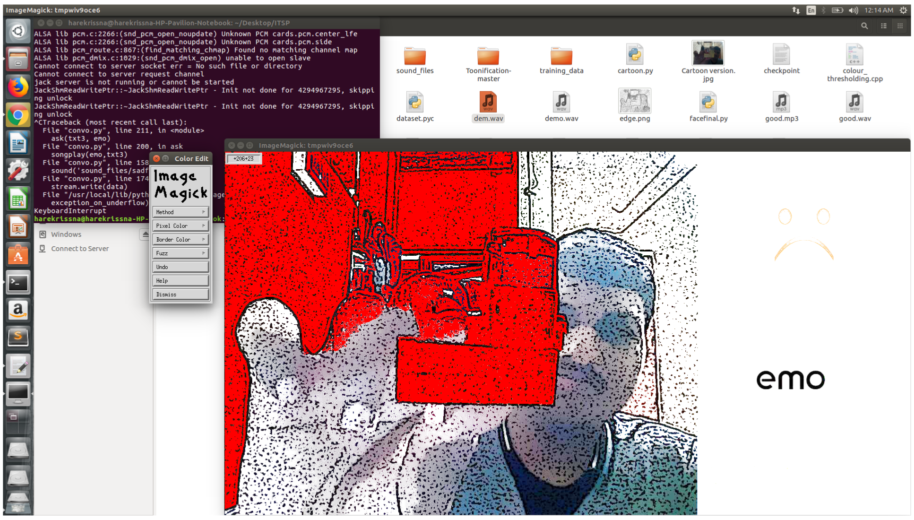
Other important aspect of the project is learning and using of various python libraries. Our project involves conversion from speech to text and vice versa several times, for which we rely on google. The major task involved in our project are:

- Welcome the person through pre recorded sound files, asking the name of the person which will be used in further processing- sound files are played using the pyaudio, wave libraries.

- Record the name of the person, automatically detect the ending time of recording through silence detection using pyaudio.
- Then we process the recorded file to remove unnecessary noises, through various functions for reducing ambient noise.
- Then we use the processed speech for google speech to text recognition, getting the name of the person.
- We need to convert the .mp3 files to .wav files for easier use, for which we use python os library.
- Then the TV asks the person for letting it read his/her emotions, asking him to face the webcam of the laptop.
- One other python script, simultaneously makes the webcam ready for capturing the image.
- Automatically capturing the image, another python file starts working on the captured picture for emotion recognition.
- Based on the emotion predicted, feedback is sent back to the main python script, which proceeds to further tasks.
- The captured image is sent for cartoonization to another python file, which does colour thresholding to the image using openCV and makes it vectorized.
- The main python file till then starts to give results to the person, specifying his name and emotions.
- It then asks the person if he wants his cartoonized image and if the person gives a positive reply, his/her cartoonized photo is shown along with his/her name and emotions - stitching of the various images is done through normal OpenCV commands and displayed through Pillow library.



- We have also added a GUI of Image Magick editor for further editing of the photo, this GUI can be activated by clicking on the photo, it has many good editing features.



```
ALSA lib pcm.c:2266:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2266:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_route.c:867:(find_matching_chmap) Found no matching channel map
ALSA lib pcm_dmix.c:1029:(snd_pcm_dmix_open) unable to open slave
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for 4294967295, skippi
ng unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for 4294967295, skippi
ng unlock
^CTraceback (most recent call last):
  File "convo.py", line 211, in <module>
    ask(txt3, emo)
  File "convo.py", line 200, in ask
    songplay(emo,txt3)
  File "convo.py", line 158
    sound('sound_files/sadf
  File "convo.py", line 174
    stream.write(data)
  File "/usr/local/lib/pyth
    exception_on_underflow)
KeyboardInterrupt
harekrissna@harekrissna-HP-
```

- Finally, the program takes its leave by playing a song according to the mood of the person, again using the wave library.

6 The files in project

- **convo.py:** This is the main python script and backbone of the programme. Its major component is the audio recording and processing part. It sends and receives various feedbacks, textes, sounds, photos, etc from other python files in the project. All the major components of our project have found themselves a place in one of the functions of this python script.

- **predict.py:** It runs the emotion recognition part of the project, after receiving the picture from webcam of the pc through facefinal.py. Finally, it prints the recognized emotions on the terminal, sends the results back to convo.py

- **cartoon.py:** It is the python script which is used to convert the webcam image to cartoon and saves them.

- **ttI.py:** This is used to convert the name of the person to Image containing the name.

- **combine.py:** This is the final file, which is used for stitching the cartoonized image, emoji image and name containing images together

7 Major problems faced while doing the project:

There were many problems we faced during our ITSP project- to make it completely working from end to end. Apart from management of the teams, these were the major problems faced:

- We purchased the Kinect v1.8 in hope for using its various features for the benefit of the project - but soon we started to realize our problems. The online support and documentation for older version of kinect is already and completely removed and hence, we weren't able to change any aspect of its coding. We gave at least 15 days for trying to get through the programming of Kinect, but we got almost no help from anyone or any site as hardly anyone uses kinect. Finally we had to leave the Kinect and had to continue using the laptop webcam and microphone.

- We were going to use Kinect for gesture recognition, person tagging as an additional feature to the project but due to Kinect issues, we had to leave those features. We still tried gesture recognition and we are almost through using webcam, python and finding the largest contour in the body, then finding its centroid to recognize the gestures.

- Another major problem we had was for collection of dataset. We needed the dataset of Indian faces for our recognition as most of our subjects were going to be Indians. But wherever we searched on internet, we got either very bad or licensed dataset. So we had to resort to manual dataset collection for which we clicked our and our friends' photos. But still, we don't have sufficient dataset and hence our accuracy of emotion recognition ranges only around 50-60% for three major emotions - happy, sad, surprised.

8 Conclusion

Overall the Team VCare couldn't achieve all the features of EmoTV project.But during the process there was lot of learning which will help us somehow or other in future. Exposure to problems and working as a team were the plus points during the project. We understood what all problems are faced if proper planning is not done in advance. Also a Never Give up attitude was later developed in us to complete the project. Due to constraint of time, we learnt the lesson of time management. We feel EmoTV can revolutionize how the world works. If properly extended, it can automize any tom dick and harry gadget in future just based on person's emotions.

Our Team’s working YouTube video and project github link:

YouTube link:
https://www.youtube.com/watch?v=HD_Xbej_fns

Github link:
https://github.com/parthisultimate/EmoTV-ITSP-

Hope you like our project!