Multimodal Emotion Recognition in Videos

Nishant Rai, Amlan Kar, Atanu Chakrobarty, Sahil Grover

Mentored by: Aditya Modi

Completed under Programming Club, IIT Kanpur in the summers of 2014

Abstract----This project presents a multimodal emotion recognition model which aims at acheiving higher accuracies than the present ones using all the prevalent features in a video, namely text, audio, video. We are classifying the videos into neutral and the 6 basic Ekman emotions [1]. We have decided to use three independent classifiers, one each for text, audio and video. The text classifier gives positive-negative scores showing the extent to which a given statement is positive or negative. There were many implementations, Naive Bayes algorithm, K nearest neighbours and support vector machines. The recognition rate was roughly 74% for all of them, with the Naive Bayes model performing slightly better. The Image classifier used approaches based on Principal Component Anlaysis [2], Eignfaces [3], Haar Cascade detectors [4] and some others. The best accuracy achieved was around 65%. The audio classifier uses Mel frequency cepstrum coefficients [5] as features. Some other concepts using vector quantification are also used. Support Vector Machines and K Nearest Neigbors are used, the best accuracy acheived was 60%. To merge these three classifiers, we used a simple voting method. Then in weights proportional to their precision we added the probabilities, and then decided.

(1) INTRODUCTION :

It is widely accepted from psychological theory that human emotions can be classified into six archetypal emotions: surprise, fear, disgust, anger, happiness, and sadness. Facial expressions, the tone of the speech and the literal meaning of our speech play a major role in expressing these emotions. Our facial expressions can be changed, the features of our speech can be intentionally modified, things we say can be altered to display different feelings. Human beings can recognize these signals even if they are subtly displayed, by simultaneously processing information acquired by ears and eyes. Studies show that some emotions are better identified with audio, and others with video, such as surprise and happiness and yet some others with the text spoken. It's natural to think that models which consider all the features simultaneously will perform better. Although several automatic emotion recognition systems have explored the use of either facial expressions or speech to detect emotions, there are relatively few efforts which studied emotion recognition using all or a combination of them.

The project uses a triad based approach to classify emotions in a better way than the present unimodal systems. The basic inspiration being that one classifier might deliver the right information if the others are confused. Videos are taken as input and are classified into 7 emotions namely : neutral, surprise, fear, disgust, anger, happiness, and sadness.

(2) PROPOSED MODEL FOR EMOTION DETECTION :

The initial model we thought of was a simple voting model, in which the three classifiers voted on what the emotion is according to them. Then we thought of another model in which we compared the probabilities instead of the results of the classifiers. Here, we would listen to classifier which was the most sure (i.e. Has the highest probability for the result. For example : if classifier A says that chances of it being 'x' is 0.7, and classifier B says that chances of it being 'x' is 0.9, we listen to B.)

The next model we thought of was one in which we passed the probabilities of each emotion as features. So in all we have a 7x3 (=21) dimensional feature vector. So this vector could be trained using a svm or neural network to find the appropriate weights for the classifier automatically.

The final model is an improvement over the previous one and is quite extensive in a way, since it considers a lot of things about the classifier. We make variable weights, or more correctly weights which vary with emotion. We decide weights for the classifier on the basis of it's precision for an emotion. Then we multiply the weights one by one with the respective probabilities (one by one for different emotions). Finally we consider the emotion for which making a decision is the easiest i.e. The probability distribution favors that emotion. If there is no such emotion,then we use a backup method with another predefined weights.

(3) ANALYSIS OF THE CLASSIFIERS :

(3.1) TEXT CLASSIFIER :

(3.1.1) IMPLEMENTATION :

Initially we were aiming at classifying text into the 7 emotions mentioned earlier. But after some hours into it, we concluded it's a difficult job, since most sentences are emotionless. They are just said without having any emotion associated with it. Thus we decided to classify the sentences into positive or negative ones, and then take some idea about the overall emotion from it.

The databases we used were [6], [7] and [8].

We tried the following implementations :

1. This was the only classifier which classified into 7 emotions. The approach is based on finding which 'words' correspond to which emotion, and then a simple 'nearest k' like approach gives the respective probabilities of the sentence belonging to that emotion.
2. To improve our approach we also considered POS (part of speech) in a sentence. Mainly nouns, adverbs, adjectives and verbs, modifiers were handled differently, by multiplying their weights with those of the future words. A self implemented POS tagger would require additional data, but thanks to an already implemented POS-Tagger in NLTK we didn't have to search for it.
3. Then we tried the Naive Bayes classifier, it's a general approach consisting of some housekeeping (i.e. Removing stopwords, tokenisation and other stuff) and a standard Bernoulli model.
4. We then thought of a slight improvement in the Naive Bayes classifier. It consists of marking words according to their POS and then using the Naive Bayes approach. Had to make a program to check it's accuracy. Ironically, this reduces the accuracy, it maybe because the dataset contains only review statements, not natural ones which occur in day to day conversations.
5. To improve the second classifier, we merge it with support vector machines with the positive and negative scores as features. We then also experimented with Support Vector Regression. The accuracy changes were very minor.
6. We made another Naive Bayes Classifier with a larger dictionary than the original one. There were slight improvements.

(3.1.2) ACCURACY :

The best accuracy achieved was 80% by the Naive Bayes Classifier, owing to its compatibility with textual data. But since it was trained on the movie review dataset, it isn't very robust. It falters on real life data, which is natural in a way much different than movie reviews.

So considering robustness and other things in mind, the best classifier is the one which uses the sum of positive and negative scores (i.e. The second classifier). It gave an accuracy of around 65% with the movie review set, and works quite well on natural data.

(3.2) IMAGE CLASSIFIER :

(3.2.1) IMPLEMENTATION :

Initially we had two options wich were:

1. Classifying on the basis of motion of the face: The model is based on FACS (Facial Action Coding System). This can only classify emotions in a video.
2. Classify on the basis of still images: This model classifies only on the basis of the facial expression at the instant, this can classify emotions in both a video and an image.

Both of them required some special libraries in C++, which were extremely difficult to install. Thus we moved on to the following other approaches :

(1) Convert the image to Eigenfaces and then compare them.
(2) Resize the image and construct a feature vector taking all the pixels as features and use SVM, kNN or other methods to classify.
(3) Convert the images to a feature vector, and then reduce the feature size by Principal Component Analysis. Then use classification methods on it.
(4) Instead of considering the whole face as features, use Haar Cascade detectors to find the nose, eyes and mouth, consider only these as features. Then use classification methods on it.

(4) was just an improvement which was an addon to all the other methods, it greatly helped the classifiers increase the accuracy.

The databases used were [9] and [10].

The classifier with the best accuracy had the following implementation. First, we removed noise using normalised box filtering. Then we normalised the intensity using Histogram equalistaion (this was an important step). This was then followed by the algorithms mentioned above.

(3.2.2) ACCURACY :

The best accuracy acheived was around 72% for similar faces. For completely different faces the best accuracy achieved was aorund 55%. This is significantly better than the random 14.28%. Support Vector machines were the best performing classifiers followed by random forest classifiers.

It also performed well for the web cam input. Because of the histogram normalisation, it performed well in different lighting conditions.

(3.3) AUDIO CLASSIFIER :

(3.3.1) IMPLEMENTATION :

The main features of speech which can help in determining emotions are as follows:

1. Pitch features
2. Energy contours
3. MFCC and related stuff (it's median, mean, maxima, etc)
4. LFPC and related data

The database used is [11].

The classifiers used pyaudio and scikits.audiolab for audio input-output. The basic structure composed of calculating MFCC features for all the frames of an audio clip. For each frame, the MFCC was a 13 dimensional vector.

Since we couldn't make all of them features (the dimension of the feature vector would be overwhelming). We decided to use vector clustering or quantization. The scipy.cluster library is used for vector quantization. So we had codes with values from 0-63, then each set of mfcc vector (13 values) was assigned a code using the scipy.cluster algorithm.

Thus we had a sequence of numbers ranging from 0-63. This step brought down the initial problem to a problem which is well suited for Markov Models.

Due to some problems we weren't able to use HMM and GMM. So we turned to SVM and nearest K algorithm using the original mfcc vectors as features.

We also used the mean, variances and extremes as features which brought down the whole feature size to 17. This worked really well and made the classifier a bit robust.

(3.3.2) ACCURACY :

The best accuracy ahieved was around 64% using a SVM. It should be noted that accuracy for human recognition on the same database is also around 67%. Thus our classifier works reasonably well.

(4) MERGING THE CLASSIFIERS :

The merging was done according to the algorithms discussed in (2). There are no formal values available since the data we had to use was made by us, and was very less in number.

(5) CONCLUSION :

This sure was a hell of a project, it was a mixture of 5 'major' areas of computer science : Machine Learning, Computer Vision, Image Processing, Audio Processing, Natural Language Processing and some more minor ones.

We learnt a lot, from methods in dimensionality reduction to algorithms in machine learning. We tried hundreds of models day in day out to see which one worked the best. We understood that when you do machine learning, there is no right or wrong, there's only an optimum. Even if you write tonnes of code, you aren't guaranteed satisfactory results. Unlike other areas, where correct code means it has to work, here the things are different. Nuisances like noise, some poor data points make life difficult even for the best models. We went so deep in search of libraries to meet our needs, that we couldn't find any, we had to use code out of people's personal repos. Code which was meant to be a fun project, and thus we had to push ourselves to understand what was written.

Come and meet any of us and we can blabber like that for another 30 minutes. So if you, the reader feels like doing a project involving machine learning. I suggest you go for it, this area adds another dimension to your computer science knowledge and we are sure you will enjoy it.

(6) FUTURE WORK :

This is a project where you can never be statisfied with what you have achieved. The improvements which we suggest are as follows:

1. The first one is something you must expect, which is improving the accuracy of the classifiers.
2. The image classifiers could use many other models such as Active Shape Models, Active Appearance Models. Somethings we had plannes but couldn't do because of absence of libraries. Presently the best classifiers use the above mentioned models.
3. Classification on the basis of videos can also be tried, in which the previous frames are also considered. Facial Coding System can be used.
4. More features can be extracted for the audio classifier such as pitch, frequency, energy, etc. This was again not done due to the absence of libraries.
5. Different ways of merging could be tried.

(7) REFERENCES :

[1] Ekman Emotions : http://www.nbb.cornell.edu/neurobio/land/oldstudentprojects/cs490-95to96/hjkim/emotions.html

[2] Principal Component Analysis : https://www.ce.yildiz.edu.tr/personal/songul/file/1097/principal_components.pdf

[3] Eigenfaces : http://en.wikipedia.org/wiki/Eigenface

[4] Haar Cascade algorithm : http://docs.opencv.org/trunk/doc/py_tutorials/py_objdetect/py_face_detection/

[5] Mel Frequency Cepstrum coefficients :
http://en.wikipedia.org/wiki/Mel-frequency_cepstrum

[6] Database for affective text :
http://www.cse.unt.edu/~rada/affectivetext/

[7] Sentiword corpus :
http://sentiwordnet.isti.cnr.it/

[8] Movie review datbase :
http://www.cs.cornell.edu/People/pabo/movie-review-data/

[9] Japanese models image database :
http://www.kasrl.org/jaffe.html

[10] 2D face database :
http://pics.psych.stir.ac.uk/2D_face_sets.htm

[11] Berlin Emotional Speech Database :
http://www.expressive-speech.net/

[12] Text Semantics Analysis :
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4427100

[13] A wonderful paper on emotion recognition in sentences :
http://www.scielo.org.mx/pdf/poli/n45/n45a7.pdf

[14] Recognizing Emotions in Text :
http://www-scf.usc.edu/~saman/pubs/2007-MS-Thesis.pdf

[15] A good source of corpora :
http://emotion-research.net/wiki/Databases

[16] Emotional Prosody Speech and Transcripts :
https://catalog.ldc.upenn.edu/LDC2002S28

[17] Part of Speech Tagging :
http://www.monlp.com/2011/11/08/part-of-speech-tags/
http://www.nltk.org/book/ch05.html

[18] Alchemy API :
http://www.alchemyapi.com/products/features/sentiment-analysis/

[19] FLANDMARK :
http://cmp.felk.cvut.cz/~uricamic/flandmark/

[20] ASMLIB :
https://code.google.com/p/asmlib-opencv/

[21] STASM :
http://www.milbo.users.sonic.net/stasm/

[22] Classifiers using PCA algorithm:
http://www.ijsce.org/attachments/File/v3i4/D1824093413.pdf

[23] Classifiers using AAM models:
http://people.uncw.edu/pattersone/research/publications/RatliffPatterson_HCI2008.pdf

[24] Classifiers using PCA method and then kNN:
http://www.ijcaonline.org/volume9/number12/pxc3871933.pdf

[25] Classifiers using motion of the face and then SVM:
http://www.cs.cmu.edu/~pmichel/publications/Michel-FacExpRecSVMAbstract.pdf

[26] Classifiers using PCA and neural networks:
http://uav.ro/stiinte_exacte/journal/index.php/TAMCS/article/viewFile/2/11

[27] Survey on speech emotion recognition: Features, classification schemes, and databases :
http://www.sciencedirect.com/science/article/pii/S0031320310004619

[28] Speech Emotion Recognition Using SVM :
http://www.sersc.org/journals/IJSH/vol6_no2_2012/15.pdf

[29] OPENSMILE :
http://opensmile.sourceforge.net/

[30] SoX :
http://sox.sourceforge.net/sox.html