Multi-modal Approaches for Personality Analysis through Videos

July 15, 2016

1 Team details

- Team Name : Pandora
- Team Leader Name : Nishant Rai
- Team Leader Address, Phone Number, Email :Indian Institute of Technology Kanpur, +91 8604293406, nishantr018@gmail.com
- Rest of the team members: NA
- Team website URL (if any) : NA
- Affiliation : Indian Institute of Technology Kanpur

2 Contribution details

- Title of the contribution : Multi-modal Approaches for Personality Analysis through Videos
- Final score : NA
- General method description : We train multiple models which focus on different modalities of the data, namely, visual (facial and background) and audio based cues. We represent a video clip as a collection of the frame wise predictions, this representation is then used to predict the trait scores. Finally, we learn multiple ensembles of these models and use multiple strategies to fuse the predictions of each individual model.
- References :

  One Millisecond Face Alignment with an Ensemble of Regression Trees by Vahid Kazemi and Josephine Sullivan, CVPR 2014

  Kahou, Samira Ebrahimi, et al. "Emonets: Multimodal deep learning approaches for emotion recognition in video." Journal on Multimodal User Interfaces (2015): 1-13.

- Representative image / diagram of the method : NA
- Describe data preprocessing techniques applied (if any) : Not considering extremely close (Time wise) frames to increase diversity. Histogram normalization after detecting the face. Bounding box smoothing (Discussed Later).

3 Visual Analysis

3.1 Face Detection Stage

3.1.1 Features / Data representation

Describe features used or data representation model FOR FACE DETECTION STAGE (if any)

The face detector used is made using Histogram of Oriented Gradients (HOG) features combined with a linear classifier, an image pyramid, and sliding window detection scheme. An implementation of this is available in the Dlib library which was used by us.

3.1.2 Dimensionality reduction

Dimensionality reduction technique applied FOR FACE DETECTION STAGE (if any)

Same as the implementation in Dlib

3.1.3 Compositional model

Compositional model used, i.e. pictorial structure FOR FACE DETECTION STAGE (if any)

Same as the implementation in Dlib

3.1.4 Learning strategy

Learning strategy applied FOR FACE DETECTION STAGE (if any)

Same as the one used by Dlib

3.1.5 Other techniques

Other technique/strategy used not included in previous items FOR FACE DETECTION STAGE (if any)

Bounding Box smoothing: In order to get image sequences where the faces are of roughly the same size (or the size varies gradually), we smooth the sizes of the boxes extracted in each frame. We perform 2 sided averaging (window size 8) to ensure the sizes vary gradually.

In case of frames with no detected faces, we neglect the frame. More on this in later sections.

We also normalize the face image we detect. We perform Histogram equalization in case of the gray scale model (We have two variants, a color model and a gray scale one).

3.1.6 Method complexity

Method complexity FOR FACE DETECTION STAGE
Same as the implementation in Dlib

3.2 Face Landmarks Alignment Stage

Used the Dlib implementation of, One Millisecond Face Alignment with an Ensemble of Regression Trees by Vahid Kazemi and Josephine Sullivan, CVPR 2014 [1]

3.2.1 Features / Data representation

Describe features used or data representation model FOR FACE LANDMARKS ALIGNMENT STAGE (if any)
Same as [1]

3.2.2 Dimensionality reduction

Dimensionality reduction technique applied FOR FACE LANDMARKS ALIGNMENT STAGE (if any)
Same as [1]

3.2.3 Compositional model

Compositional model used, i.e. pictorial structure FOR FACE LANDMARKS ALIGNMENT STAGE (if any)
Same as [1]

3.2.4 Learning strategy

Learning strategy applied FOR FACE LANDMARKS ALIGNMENT STAGE (if any)
Same as [1]

3.2.5 Other techniques

Other technique/strategy used not included in previous items FOR FACE LANDMARKS ALIGNMENT STAGE (if any)
Same as [1]

3.2.6 Method complexity

Method complexity FOR FACE LANDMARKS ALIGNMENT STAGE
Same as [1]

3.3 Facial expression recognition

NA (No direct detection of expressions)

3.3.1 Features / Data representation

Describe features used or data representation model FOR FACIAL EXPRESSION RECOGNITION STAGE (if any)

3.3.2 Dimensionality reduction

Dimensionality reduction technique applied FOR FACIAL EXPRESSION RECOGNITION STAGE (if any)

3.3.3 Compositional model

Compositional model used, i.e. pictorial structure FOR FACIAL EXPRESSION RECOGNITION STAGE (if any)

3.3.4 Learning strategy

Learning strategy applied FOR FACIAL EXPRESSION RECOGNITION STAGE (if any)

3.3.5 Other techniques

Other technique/strategy used not included in previous items FOR FACIAL EXPRESSION RECOGNITION (if any)

3.3.6 Method complexity

Method complexity FOR FACIAL EXPRESSION RECOGNITION STAGE
NA (Not performed)

4 Personality Trait recognition from Visual data

4.1 Features / Data representation

Describe features used or data representation model FOR VISUAL TRAIT RECOGNITION STAGE (if any)

We experiment with the following models using different features,

- Face based model: This model only uses the facial image to predict the scores. We take each frame one at a time and extract the facial bounding box (After smoothing). We use this image (after resizing appropriately) as our input to a CNN. The CNN is further trained to predict the personality scores for that frame. The frame wise predictions are later merged using other models.
- Background based model: It's reasonable to expect first impressions to be affected by the background information too. For example, a person (gamer) with a lot of posters and other games in the background would probably give a different impression than a person (cook) with a kitchen in the background. We use the popular VGG-net to get representations of the background by sampling random crops and feeding it to the network. We then try avg and max pooling the features from the crops.
- Merging the Face model with the VGG model: Merged model which takes the separately trained models and creates another feature by concatenating the mid layers of both the models.
- Face Landmark based model: A model which uses the facial landmarks as features was also trained, but the results were very poor. The features were normalized by centering the points to a central point and scaling it.

4.2 Dimensionality reduction

Dimensionality reduction technique applied FOR VISUAL TRAIT RECOGNITION STAGE (if any)

We learn smaller sized representations (128D) of the VGGnet features (4096D) during the training of the BG based model. This smaller representation is further fed into an MLP which then predicts the trait scores. A similar representation (256D) is learned in case of the face based model.

4.3 Compositional model

Compositional model used, i.e. pictorial structure FOR VISUAL TRAIT RECOGNITION STAGE (if any)

4.4 Learning strategy

Learning strategy applied FOR VISUAL TRAIT RECOGNITION STAGE (if any)

We use the following methods for the above mentioned methods,

- Face model: Simple training of a CNN with aggressive dropout and data augmentation. Each frame gives us a facial image which is used for training. But the amount of training samples is extremely less, this to ensure

that the model trained is robust and accurate we select random crops of the images along with a bit of resizing. This helps us to expand the dataset size and allow training a better network.

- Background based model: We randomly take multiple (15-20) crops of each frame and get the VGG representations of the same. We only consider a low number of frames (Around 2-3 for each clip). To counter the low number of training samples in this instance, during pooling we randomly choose 7-8 crops and pool them. This gives rise to a lot more training samples (Though they are slightly redundant, so the effective gain might be a bit low which was the case since a larger architecture would over-fit very quickly).
- Merging the models: For each frame we take the face (256D) and BG model (128D) representation. After concatenation, we use the formed feature vector (384D) to predict the scores. Unfortunately, the result is still not better than only the face based model (Probably needs more parameter refinement).

4.5 Other techniques

Other technique/strategy used not included in previous items FOR VISUAL TRAIT RECOGNITION (if any)

We represent each video as a collection of frame predictions. Our CNN model is able to give us the predictions for each frame, to learn a representation of a video and then use it to predict the final scores. We concatenate the predictions for each frame and train another predictor for this concatenated representation. There were significant improvements when we considered the aggregated scores of all the frames to predict the final one.

Note that to train a model on top of these frame wise predictions, we need a constant size representation (Unless using LSTMs and variable length models). To achieve this target, we select a constant number of frames (15 in our case) using expansion and averaging (Inspired from [2]). This gives us a good way to represent the video clip.

We experiment with the following strategies for merging the scores,

- For each trait, take the corresponding scores for each frame and feed only these as input.
- Take all the corresponding trait scores as done in the previous method, but now sort the scores and provide this as input. The motivation being that the position of the frames is not that relevant in such a score based model. But using method 1. puts emphasis on the position of the frame, which isn't desirable. Thus we sort the predictions and feed this as input. There was a decent improvement in the results compared to method 1.

- Instead of taking only one trait as input, feed all the traits as input. This method allows the model to use the inter trait relationships for predictions. This performs better than the previous ones.

4.6 Method complexity

Method complexity FOR VISUAL TRAIT RECOGNITION STAGE
TODO

5 Personality Trait recognition from Audio data

5.1 Features / Data representation

Describe features used or data representation model FOR AUDIO TRAIT RECOGNITION STAGE (if any)

Features extracted using the openSMILE framework. The features we used are the same as the one used in INTERSPEECH 2010 Paralinguistic Challenge. The set contains 1582 features which result from a base of 34 low-level descriptors (LLD) with 34 corresponding delta coefficients appended, and 21 functionals applied to each of these 68 LLD contours (1428 features). In addition, 19 functionals are applied to the 4 pitch-based LLD and their four delta coefficient contours (152 features). Finally the number of pitch onsets (pseudo syllables) and the total duration of the input are appended (2 features).

We extract the audio from each video clip. The average size of such an audio clip is around 15 seconds. Thus, at this stage we have around 6000 audio samples with their respective scores. We create more training samples based on the following belief, If we are to predict the personality trait of people based on their voices; then 3-4 seconds worth of audio would be sufficient to do it. There are studies which support the claim that first impressions are formed in a very short time (Around 10 seconds) and are somewhat accurate. So, we take each 15 second audio clip and extract overlapping segments of size 4(s). This gives us a huge increase in the number of training samples. Also, it allows us to concatenate the results for each segment and use them to predict the final scores instead. This leads to a very significant improvement.

5.2 Dimensionality reduction

Dimensionality reduction technique applied FOR AUDIO TRAIT RECOGNITION STAGE (if any)

Nothing noteworthy (Principal Component analysis, learning smaller representations using NNs, poor results)

5.3 Learning strategy

Learning strategy applied FOR AUDIO TRAIT RECOGNITION STAGE (if any)

The first step involves training regressors on the audio segments. We try many models such as Support Vector Regressors, Bagged Decision Tree Regressors, Extra Trees, Neural Networks, Random Forests and Linear Regressors. But it was seen that ensemble models and models using multiple weak regressors performed the best. We finally used Bagged Regressors as they performed the best.

5.4 Other techniques

Other technique/strategy used not included in previous items FOR AUDIO TRAIT RECOGNITION (if any)

We create ensembles where each individual model looks at different features, thus providing complementary views to the data. We provide different features to each component by varying the segment length and pooling strategy used in constructing the final features. We mainly experiment with Avg and Max pooling to cluster the k consecutive segments (where k varies between 3 and 5).

To create diverse models, we also normalize the features in some cases (Min-max normalizations), and train models using the normalized features as features.

5.5 Method complexity

Method complexity FOR AUDIO TRAIT RECOGNITION STAGE

6 Multimodal Personality Trait recognition

6.1 Data Fusion Strategies

List data fusion strategies (how different feature descriptions are combined) for learning the model / network: Single frame, early, slow, late. (if any)

There are two strategies we experimented with,

- Single frame fusion: We use the two visual models (The face based model and the VGG BG based one) to create a fused model which takes the learned representations from both the separately trained models and tries to predict the scores. The predictions are combined with the other model predictions at the end using another model.
- Late combination of features: The predictions of each model are combined at the end using another model.

6.2 Global Method Description

- Total method complexity: all stages
- Which pre-trained or external methods have been used (for any stage, if any) : The famous VGG convNet was used for getting the representations of the frame crops (As discussed earlier).
- Which additional data has been used in addition to the provided ChaLearn training and validation data (at any stage, if any) : Only the pre-trained VGG model.
- Qualitative advantages of the proposed solution
- Results of the comparison to other approaches (if any)
- Novelty degree of the solution and if is has been previously published

7 Other details

- Language and implementation details (including platform, memory, parallelization requirements) : Python (Theano, Keras, skLearn), i7 Processor (8 cores), 4GB RAM, No GPU used (thus restricted architecture of deep models).
- Human effort required for implementation, training and validation? : 8 hours a day, Three weeks, One person
- Training/testing expended time? : Training time : 5 days, testing time : 1 day
- General comments and impressions of the challenge? what do you expect from a new challenge in face and looking at people analysis? It was a nice challenge overall except for the few glitches in between. The lack of validation labels and the messed up server at the end almost made me lose interest and I didn't do anything for a week. One point I would like to emphasize is the scoring metric, it seems a bit too weak given the fact that just predicting the average score (always!) gives a score of 0.878. I agree that Absolute distance is a popular metric for such challenges but if possible, a bit more aggressive/penalizing metric might makes things more exciting!