Multimodal Emotion Recognition -EMOSENSE Using Deep Learning

Angel Reddy Nakkala
University of Central Missouri
Nueral network and Deep learning
Report
AXN82170@UCMO.EDU

***Abstract*—In this paper, we are exploring state of the art models in multimodal emotion recognition. We have chosen to explore textual, sound and video inputs and develop an ensemble model that gathers the information from all these sources and displays it in a clear and interpretable way. Multimodal Emotion Recognition is a relatively new discipline that aims to include text inputs, as well as sound and video. This field has been rising with the development of social networks that gave researchers access to a vast amount of data. Recent studies have been exploring potential metrics to measure the coherence between emotions from the different channels.**

Keywords—Emotion recognition · Text · Sound · Video · Affective Computing

I. Introduction

Affective computing is a field of Deep Learning and Computer Science that studies the recognition and the processing of human affects. Multimodal Emotion Recognition is a relatively new discipline that aims to include text inputs, as well as sound and video. This field has been rising with the development of social network that gave researchers access to a vast amount of data. We have chosen to diversify the data sources we used depending on the type of data considered. All data sets used are free of charge and can be directly downloaded.

For the text input, we are using the **Stream-of-consciousness** dataset that was gathered in a study by Pennebaker and King [1999]. It consists of a total of 2,468 daily writing submissions from 34 psychology students (29 women and 5 men whose ages ranged from 18 to 67 with a mean of 26.4). The writing submissions were in the form of a course unrated assignment. For each assignment, students were expected to write a minimum of 20 minutes per day about a specific topic. The data was collected during a 2-week summer course between 1993 to 1996. Each student completed their daily writing for 10 consecutive days. Students' personality scores were assessed by answering the Big Five Inventory (BFI) [John et al., 1991]. The BFI is a 44-item self-report questionnaire that provides a score for each of the five personality traits. Each item consists of short phrases and is rated using a 5-point scale that ranges from 1 (disagree strongly) to 5 (agree strongly). An instance in the data source consists of an ID, the actual essay, and five classification labels of the Big Five personality traits. Labels were originally in the form of either yes ('y') or no ('n') to indicate scoring high or low for a given trait.

For audio data sets, we are using the **Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)**. This database contains 7356 files (total size: 24.8 GB). The database contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions. Each expression is produced at two levels of emotional intensity(normal, strong), with an additional neutral expression. All conditions are avail-able in three modality formats: Audio-only (16bit, 48kHz .wav), Audio-Video(720p H.264, AAC 48kHz, .mp4), and Video-only (no sound)."

For the video data sets, we are using the popular **FER2013** Kaggle Challenge data set. The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The data set remains quite challenging to use, since there are empty pictures, or wrongly classified images.

II. Motivation

We are trying to provide definitions of affective computing and multimodal sentiment analysis in the context of our research. Those definitions may vary depending on the context. Our aim is to develop a model able to provide real time sentiment analysis with a visual user interface using Tensorflow.js technology.

Therefore, we have decided to separate two types of inputs :
1. Textual input, such as answers to questions that would be asked to a person from the platform.

2. Video input from a live webcam or stored from an MP4 or WAV file, from which we split the audio and the images.

Recent studies have been exploring potential metrics to measure the coherence between emotions from the different channels. We are going to explore several categorical targets depending on the input considered. Table 1 gives a summary of all the categorical targets we are evaluating depending on the data type.

Table 1: Categorical target depending on the input data type.

Data Type	Categorical target
Textual	Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism
Sound	Happy, Sad, Angry, Fearful, Surprise, Neutral and Disgust
Video	Happy, Sad, Angry, Fearful, Surprise, Neutral and Disgust

For the text inputs, we are going to focus on the so-called Big Five, widely used in personality surveys.

The research will explore state of the art multimodal sentiment analysis, but will also focus on compliance in the context of General Data Protection Regulation (GDPR). The aim of this project is to provide candidates seeking for a job a platform that analyses their answers to a set of pre-defined questions, as well as the non-verbal part of a job interview through sound and video processing

https://github.com/angelreddy09/DeepLearning-Project/tree/main/library

III. Related Work

A. Text mining for personality trait classification

Emotion recognition through text is a challenging task that goes beyond conventional sentiment analysis : instead of simply detecting neutral, positive or negative feelings from text, the goal is to identify a set of emotions characterized by a higher granularity. For instance, feelings like anger or happiness could be included in the classification. As recognizing such emotions can turn out to be complex even for the human eye, machine learning algorithms are likely to obtain mixed performances. It is important to note that nowadays, emotion recognition from facial expression tends to perform better than from textual expression. Indeed, many subtleties should be taken into account in order to perform an accurate detection of human emotions through text, context-dependency being one of the most crucial. This is the reason why using advanced natural language processing is required to obtain the best performance possible. There exists different ways to tackle natural language processing problems, the two main ones being rule-based and learning-based approaches. While rule-based approaches tend to focus on pattern-matching and are largely based on grammar and regular expressions, learning-based approaches put the emphasis on probabilistic modeling and likelihood maximization. Here, we will mainly focus on learning based methods, and review some of the central methods, from "traditional" classifiers to more advanced neural network architectures.

In the context of our study, we chose to use text mining in order not to detect regular emotions such as disgust or surprise, but to recognize personality traits based on the "Big Five" model in psychology. Even though emotion recognition and personality traits classification are two separate fields of studies based on different theoretical underpinnings, they use similar learning-based methods and literature from both areas can be interesting. The main motivation behind this choice is to offer a broader assessment to the user : as emotions can only be understood in the light of a person's own characteristics, we thought that analyzing personality traits would provide a new key to understanding emotional fluctuations. Our final goal is to enrich the user experience and improve the quality of our analysis : any appropriate and complementary information deepening our understanding of the user's idiosyncrasies is welcome Many psychology researchers (starting with D. W. Fiske [1949], then Norman [1963] and Goldberg [1981]), believe that it is possible to exhibit five categories, or core factors, that determine one's personality. The acronym OCEAN (for openness, conscientiousness, extraversion, agreeableness, and neuroticism) is often used to refer to this model. We chose to use this precise model as it is nowadays the most popular in psychology : while the five dimensions don't capture the peculiarity of everyone's personality, it is the theoretical framework most recognized by 8 researchers and practitioners in this field. Many linguistic-oriented tools can be used to derive a person's personality traits, for instance the individual's linguistic markers (obtained using text analysis, psycholinguistic databases and lexicons for instance). Since one of the earliest studies in this particular field [Mairesse et al., 2007], researchers have introduced multiple linguistic features and have shown correlations between them and the Big Five. These features could therefore have a non-negligible impact on classification performances, but as we stated before, we will mainly focus machine learning methods and leave out the linguistic modeling as it does not fit into the spectrum of our study. Our main goal is to leverage on the use of statistical learning methods in order to build a tool capable of recognizing the personality traits of an individual given a text containing his answers to pre-established personal questions. Our first idea was to record a user's interview and convert the file from audio to text : in this way we would have been able to work with similar data for text, audio and video. Nevertheless, the good transcription of audio files to text requires the use of expensive APIs, and the tools available for free in the market don't provide sufficient quality. This is the reason why we chose to apply our personality traits detection model to short texts directly written by users : in this way we can easily target particular themes or questions and provide indications of the language level to use. As a result of this, we can make sure that the text data we use to perform the personality traits detection is consistent with the data used for training, and therefore ensure the highest possible quality of results. In the following sections, we will go through some of the learning techniques that are commonly used in order to perform personality traits recognition. These methods are usually applied in a sequential way through a pipeline including preprocessing steps in order to standardize the data, embedding in order to represent it as a numerical vector, then the classification algorithm in order to predict labels. Let's focus on a few technical aspects.

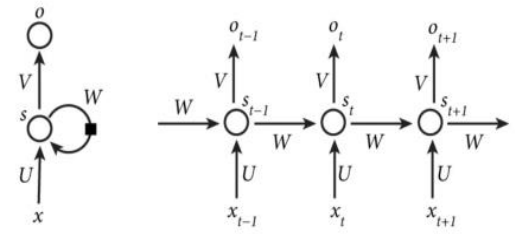
The preprocessing is the first step of our NLP pipeline : this is where we convert raw text document to cleaned lists of words. In order to complete this process, we first need to tokenize the corpus. This means that sentences are split into a list of single words, also called tokens. Other preprocessing steps include the use of regular expressions in order to delete unwanted characters or reformat words. For instance, it is common to lowercase tokens, and delete some punctuation characters that are not crucial to the understanding of the text. The removing of stopwords in order to retain only words with meaning is also an important step : it allows to get rid of words that are too common like 'a', 'the' or 'an'. Finally, there are methods available in order to replace words by their grammatical root : the goal of both stemming and lemmatization is to reduce derivationally related forms of a word to a common base form. Families of derivationally related words 9 with similar meanings, such as 'am', 'are', 'is' would then be replace by the word 'be'. Finally, in the context of word sense disambiguation, part-of-speech tagging is used in order to markup words in a corpus as corresponding to a particular part of speech, based on both its definition and its context. This can be used to improve the accuracy of the lemmatization process, or just to have a better understanding of the meaning of a sentence.

Classification algorithms used Multinomial Na¨ıve Bayes and Support Vector Machines, multinomial naive bayes and support vector machines. The multinomial naive bayes algorithm applies Bayes theorem : it is based on the rather strong assumption that, in the context of classification, every feature is independent of the others. This classifier will always output the category with the highest a priori probability using Bayes theorem. This algorithm has a simple

and intuitive design and is a good benchmark for classification purposes.

Recurrent Neural Networks and LSTM Recurrent neural networks leverage on the sequential nature of information : unlike regular neural network where inputs are assumed to be independent of each other, these architectures progressively accumulate and capture information through the sequences.

gradient signal might cause learning to diverge (exploding gradient phenomenon).



(a) RNN cell (b) Corresponding unfolded network

This is one of the essential reasons why Long Short Term Memory architectures, introduced by Hochreiter Schmidhuber [1997], have an edge over conventional feed-forward neural networks and RNN. Indeed, LSTMs have the property of selectively remembering patterns for long durations of time. This is made possible by what is called a memory cell. Its unique structure is composed of four main elements : an input gate, a neuron with a self-recurrent connection, a forget gate and an output gate. The self-recurrent connection ensures that the state of a memory cell can remain constant from one timestep to another. The role of the 18 gates is to fine-tune the interactions between the memory cell and its environment using a sigmoid layer and a point-wise multiplication operation. The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!”

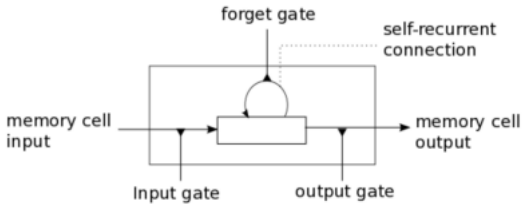


Fig. 1: LSTM memory cell

Let’s now simply provide a list of the different steps of our pipeline in order to sum up the information given above and give more details about our final choices. This precise pipeline is the one that obtained the highest accuracy among all the combinations we have tried.

1. Preprocessing
   – Tokenization of the document
   – Standardization of formulations using regular expressions (for instance replacing ”can’t” by ”cannot”, ”’ve” by ” have”) – Deletion of the punctuation
   – Lowercasing the tokens – Removal of predefined stopwords (such as ’a’, ’an’ etc.)
   – Application of part-of-speech tags remaining tokens
   – Lemmatization of tokens using part-of-speech tags for more accuracy.
   -Padding the sequences of tokens of each document to constrain the shape of the input vectors. The input size has been fixed to 300 : all tokens beyond this index are deleted. If the input vector has less than 300 tokens, zeros are added at the beginning of the vector in order to normalize the shape. The dimension of the padded sequence has been determine using the characteristics of our training data. The average number of words in each essay was 652 before any preprocessing. After the standardization of formulations, and the removal of punctuation characters and stopwords, the average number of words dropped to with a standard deviation of . In order to make sure we incorporate in our classification the right number of words without discarding too much information, we set the padding dimension to 300, which is roughly equal to the average length plus two times the standard deviation.
2. Embedding
   – Each token is replaced by its embedding vector using Google’s pre-trained Word2Vec vectors in 300 dimensions (which is the largest dimension available and therefore incorporates the most information), and this embedding is set to be trainable (our training corpus is to small to train our own embedding).
3. Classifier – Neural network architecture based on both one-dimensional convolutional neural networks and recurrent neural networks. The one-dimensional convolution layer plays a role comparable to feature extraction : it allows finding patterns in text data. The Long-Short Term Memory cell is then used in order to leverage on the sequential nature of natural language : unlike regular neural network where inputs are assumed to be inde pendent of each other, these architectures progressively accumulate and capture information through the sequences. LSTMs have the property of selectively remembering patterns for long durations of time. Our final model first includes 3 consecutive blocks consisting of the following four layers : one-dimensional convolution layer - max pooling - spatial dropout 20 - batch normalization. The numbers of convolution filters are respectively 128, 256 and 512 for each block, kernel size is 8, max pooling size is 2 and dropout rate is 0.3. Following the three blocks, we chose to stack LSTM cells with 180 outputs each. Finally, a fully connected layer of 128 nodes is added before the last classification layer.

B. Speech Signal processing for emotion recognition

Speech emotion recognition purpose is to automatically identify the emotional or physical state of a human being from his voice. The emotional state of a person hidden in his speech is an important factor of human communication and interaction as it provides feedbacks in communication while

not altering linguistic contents. Speech emotion recognition is based on discrete emotion classification system. In most cases, literature focus only on 6 emotions labels introduced by Ekman, including happy, sad, angry, disgusted, fear and surprise. Although the emotion categories are more abundant and complex in real life. The usual process for speech emotion recognition consists of three parts: signal processing, feature extraction and classification. Signal processing applies acoustic filter on original audio signals and splits it into meaningful units. The feature extraction is the sensitive point in speech emotion recognition because features need to both efficiently characterize the emotional content of a human speech and not depend on the lexical content or even the speaker. Finally, emotion classification will map feature matrix to emotion labels.

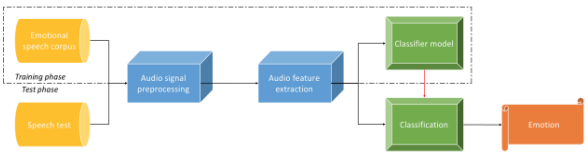


Fig. 2: Speech emotion recognition pipeline

Convolutional Neural Networks (CNNs) show remarkable recognition performance for computer vision tasks while Recurrent Neural Networks (RNNs) show impressive achievement in many sequential data processing tasks. The concept of time distributed convolutional neural network is to combine a deep hierarchical CNNs feature extraction architecture with a recurrent neural network model that can learn to recognize sequential dynamics in a speech signal. Unlike the SVM approach, we will no longer work on global statistics generated on features from time and frequency domain. This network only takes the log mel-spectrogram (presented in previous section) as input. The main idea of time distributed convolutional neural network is to apply a rolling window (fixed size and time-step) all along the log-mel-spectrogram. Each of these windows will be the entry of a convolutional neural network, composed by four Local Feature Learning Blocks (LFLBs) and the output of each of these convolutional networks will be fed into a recurrent neural network composed by 2 cells LSTM (Long Short Term Memory) to learn the long-term contextual dependencies. Finally, a fully connected layer with softmax activation is used to predict the emotion detected in the voice.

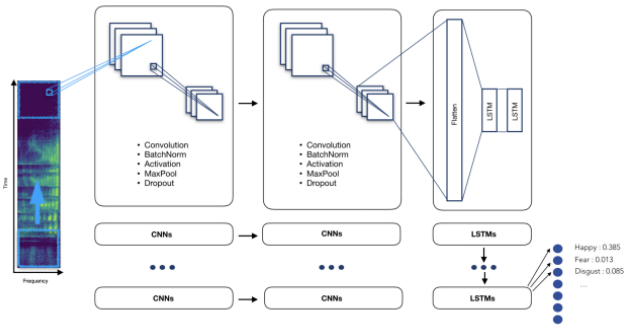


Figure: Time distributed Convolutional Neural Network schema

SVM :We first implemented SVM classifiers based on different kernel functions (linear, polynomial and RBF), without dimensionality reduction and gender differentiation. Speech emotion recognition accuracies shown in next table were relatively low. However, the SVM with RBF kernel functions seems to be the best performer with an accuracy rate of 56.51%. Then we applied both feature selection (1%-Chi-squared test removed 75 features) and feature transformation (PCA) to reduce the dimension of the features. For PCA, three levels of explained variance were tested (90%, 95% and 98%) respectively leading to the following features dimensions : 100, 120 and 140. Our performances were still very low but the accuracy of polynomial and RBF increased respectively by 6% and 3% with the 140 feature dimension corresponding to the 98% contribution. RBF kernel still remains the best classifier.

The first major improvement was observed with the implementation of gender differentiation as suggested in previous section. accuracy scores of almost all classifier (except for 3-degree polynomial) increased by almost 5%. The next figure illustrates accuracy rates obtained by crossvalidation and the confusion matrix of the classifier with the highest accuracy score: RBF Kernel and PCA 180 features dimension (corresponding to 98% contribution).

Time distributed Convolutional Neural Network In this part, we present the results obtained with the deep learning model whose architecture has been presented in the previous sections. To limit overfitting during training phase, we split our data set into train (80%), validation (15%) and test set (5%). We also added early stopping to stop the training when the validation accuracy starts to decrease while the training accuracy steadily increases. We chose Stochastic Gradient Descent with decay and momentum as optimizer and a batch size of 64. Following graphics present loss (categorical cross-entropy) and accuracy for both train and validation set:

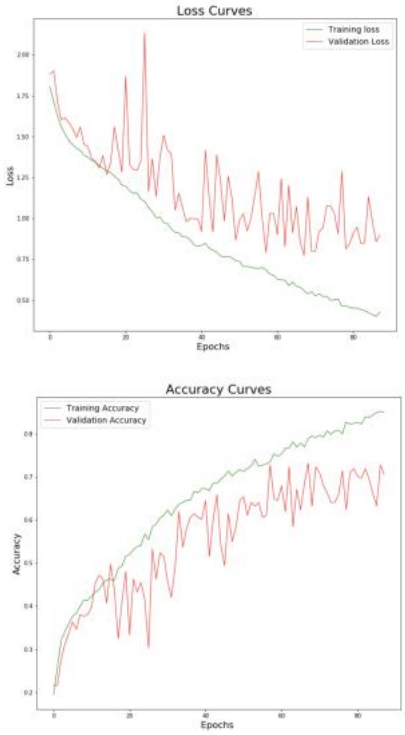


Figure: Loss and accuracy on training and validation set

Our model allows us to obtain a maximum score of 74% on the validation set. Thanks to a Keras functionality called

ModelCheckpoint we have been able to save the weights associated to the best model, allowing us to obtain a score of 72% on the test set. The use of deep learning and time distributed convolutional neural network allow us to achieve a 10% higher performance compared to the traditional approach using SVM.

C. Computer vision(Video) for emotion recognition

In the field of facial emotion recognition, most recent research papers focus on deep learning techniques, and more specifically on Convolution Neural Network (CNN). The aim of the following section is to develop the basis of CNNs.
Convolution Neural Network:
CNNs are special types of neural networks for processing data with grid-like topology.
In more traditional approaches, a great part of the work was to select the filters (e.g Gabor filters) and the architecture of the filters in order to extract as much information from the image as possible. With the rise of deep learning and greater computation capacities, this work can now be automated. The name of the CNNs comes from the fact that we convolve the initial image input with a set of filters. The parameter to choose remains the number of filters to apply, the dimension of the filters, and the stride length. The stride length is the step by which we convolve the filter on the image. Typical values for the stride length lie between 2 and 5. In some sense, we are building a convolved output that has a volume. It's no longer a 2 dimensional picture. The filters are hardly humanly understandable, especially when we use a lot of them. Some are used to find curves, other edges, other textures... Once this volume has been extracted, it can be flattened and passed into a dense Neural Network.

Data exploration and visualization First of all, when exploring the data of the FER2013 data set, we observe that there is an imbalance in the number of images by class (emotion).

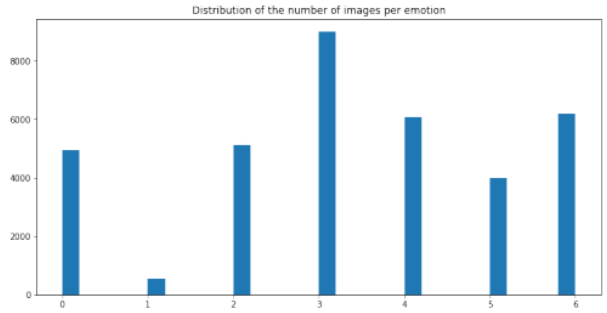
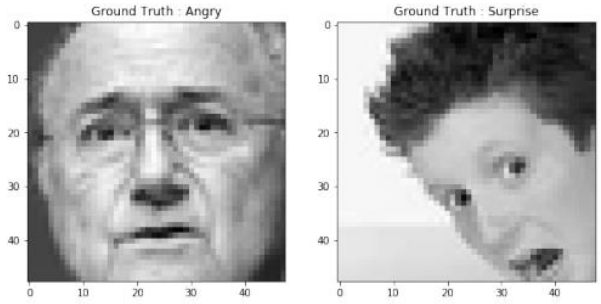
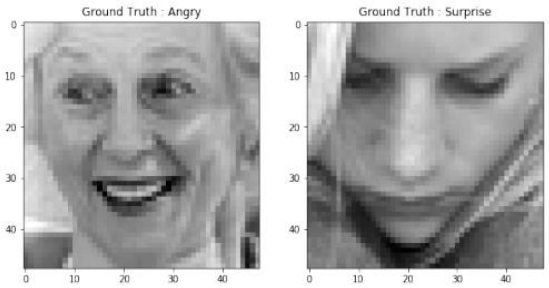


Figure: Emotion distribution: Angry (0), Disgust (1), Fear (2), Happy (3), Sad (4), Surprise (5) and Neutral (6)

The train set has 28709 images, the test set has 3589 images. For each image, the data set contains the grayscale color of 2304 pixels (48x48), as well as the emotion associated. The challenge is that some pictures are miss-classified, while other images only show part of a face. For example, the two images of Figure 13 seem rather clearly miss-classified.



The challenge is that some pictures are miss-classified, while other images are only show part of the face. For example, the two images below seem rather clearly miss-classified.



Above figure is the example of miss-qualified images.

It might be interesting to look at the average face per emotion with our data. Thanks to this representation, we can understand the way an emotion is being interpreted within our data. We notice how the axis of the eyes for the anger is different from the happiness for example.

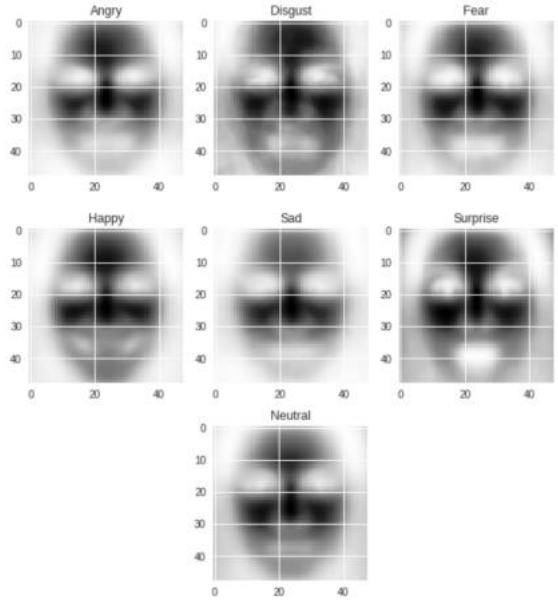
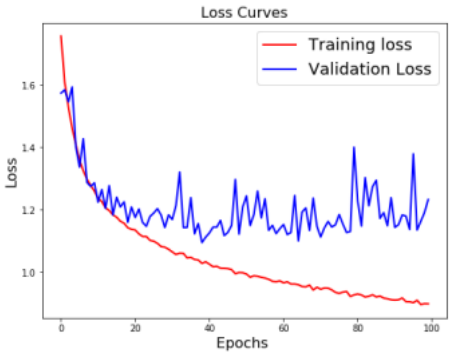
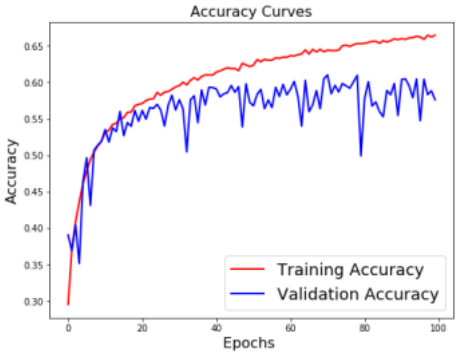


Figure: Average Face per emotion

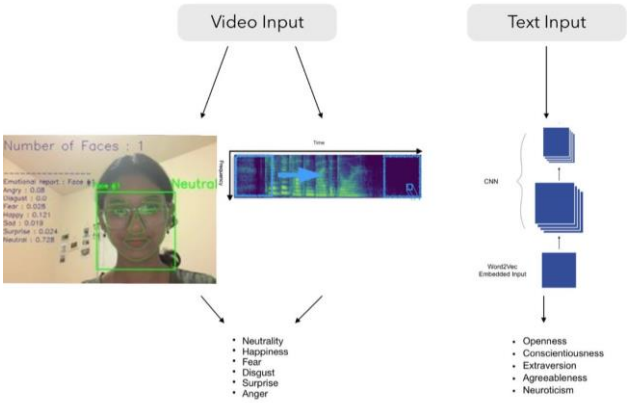
This simple architecture produces over 440'000 parameters to estimate. The computation time is around 8 hours on local machine. In order to prevent overfitting, we also apply Keras built-in data generation module, and add batch normalization. The optimizer we chose is RMSprop, an optimizer that divides the learning rate by an exponentially decaying average of squared gradients. The loss we use is the categorical cross-entropy, since we face a classification

problem. Finally, the metric we use is the accuracy. When plotting the accuracy and the loss in terms of the epochs, on both the training and the validation set, we observe a rather clear overfitting that occurs after approximately 20 epochs. Solutions will be addressed in the next section





We seem to face an issue of overfitting. For this reason, the next section will detail the approach that has been selected and the results that have been reached with such approach. This simple CNN architecture allows for an easy interpretability. We can indeed plot class activation maps, which display the pixels that have been activated by the last convolution layer. We notice how the pixels are being activated differently depending on the emotion being labeled. The happiness seems to depend on the pixels linked to the eyes and mouth, whereas the sadness or the anger seem for example to be more related to the eyebrows



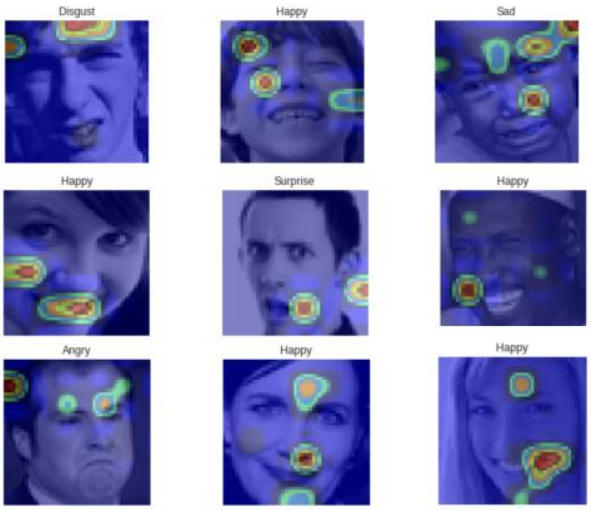


Figure: Class Activation Map

One of the main challenges in this task is to limit overfitting. Indeed, due to the large class imbalance and the number of parameters that need to be learned, the models can easily overfit. For this reason, several approaches and techniques have been developed and implemented. In the next sections, we will cover the main techniques, and the outcomes of these different techniques, as well as the final model that has been implemented.

In the context of multimodal sentiment analysis, a key component is to be able to understand emotions from a video input, not only from pictures.

The methodology to deploy our trained model on a webcam stream is the following :

• analyze the video image by image
• apply a grayscale filter to work with fewer inputs
• identify the face and zoom on it
• manage multiple faces
• reduce pixel density to same pixel density than the train set
• transform the input image to a model readable input
• predict the emotion of the input

The image processing is done with OpenCV on Python. The face detection is done with a Cascade Classifier. Cascade classifiers is based on the concatenation of several classifiers, using and uses all the information from the output from the previous classifier as additional information for the next classifier in the cascade. Cascade Classifiers are typically applied to regions of interest of an image. The classifier will return 0 or 1 depending on whether the region is likely to show the object tested. The classifiers typically are Adaboost, Real Adaboost, Gentle Adaboost and Logitboost. OpenCV offers pre-trained cascade models for face detection. The webcam facial emotion algorithm presented above can be illustrated with a quick example. The classification seems to work well in practice. There are however still many sources of improvements.

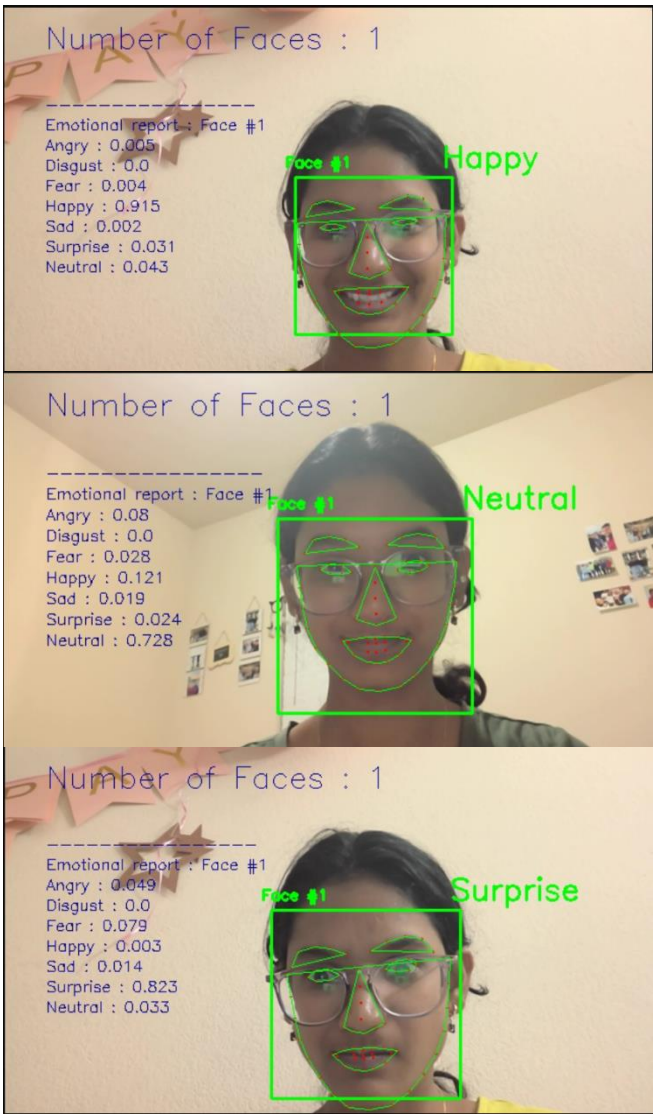


Figure: Live facial emotion recognition

Our model works well and the accuracy reaches 64%. We can illustrate the accuracy of the algorithm with a concrete example. Additional work has also been made to allow for multi-faces real time emotion recognition.

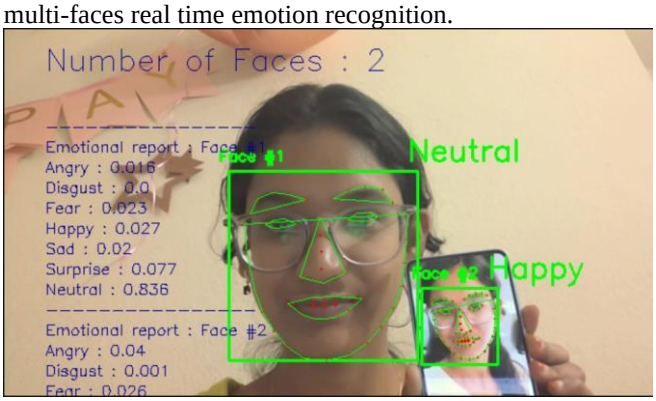


Figure: Multi-Faces emotion recognition

IV. Results and Potential Improvements

Text Results:

We tested different combinations of embeddings and classifiers in order to compare results. As explained at the end of the part on Word2Vec embeddings, we tried a hybrid model using averaged vector representations using TF-IDF weights : there is a loss of accuracy compared to the complete Word2Vec embedding, but results are better than the regular TF-IDF embedding. Let's provide the details of the accuracy obtained with each combination that we tested in our pipeline:

Model	EXT	NEU	AGR	CON	OPN
TF-IDF + MNB	45.34	45.11	45.24	45.31	45.12
TF-IDF + SVM	45.78	45.91	45.41	45.54	45.56
Word2Vec + MNB	45.02	46.01	46.34	46.38	45.97
Word2Vec + SVM	46.18	48.21	49.65	49.97	50.07
Word2Vec (TF-IDF averaging) + MNB	45.87	44.99	45.38	44.21	44.84
Word2Vec (TF-IDF averaging) + SVM	46.01	46.19	47.56	48.11	48.89
Word2Vec + NN (LSTM)	51.98	50.01	51.57	51.11	50.51
Word2Vec + NN (CONV + LSTM)	**55.07**	**50.17**	**54.57**	**53.23**	**53.84**

Possible improvements :In order to improve our accuracy, we could use hybrid models including both learning-based methods and linguistic features : adding lexicon-based features, using psycholinguistic databases, or even adding home-made features based on psychological research should have a positive impact on our accuracy scores.

Speech Results:

We first implemented SVM classifiers based on different kernel functions (linear, polynomial and RBF), without dimensionality reduction and gender differentia tion. Speech emotion recognition accuracies shown in next table were relatively low. However, the SVM with RBF kernel functions seems to be the best performer with an accuracy rate of 56.51%. Then we applied both feature selection (1%-Chi-squared test removed 75 features) and feature transformation (PCA) to reduce the dimension of the features. For PCA, three levels of explained variance were tested (90%, 95% and 98%) respectively leading to the following features dimensions : 100, 120 and 140. Our performances were still very low but the accuracy of polynomial and RBF increased respectively by 6% and 3% with the 140 feature dimension corresponding to the 98% contribution. RBF kernel still remains the best classifier.

PCA dimension	linear	poly (2)	poly (3)	rbf
None	*51.67%*	*54.28%*	*52.79%*	*56.51%*
140	*53.53%*	*50.19%*	*52.79%*	***59.48%***
120	*55.02%*	*50.56%*	*52.79%*	*58.74%*
100	*52.79%*	*48.33%*	*52.79%*	*58.36%*

The first major improvement was observed with the implementation of gender differentiation as suggested in previous section. As shown in the following table, accuracy scores of almost all classifier (except for 3-degree polynomial) increased by almost 5%. The next figure illustrates accuracy rates obtained by cross validation and the confusion matrix of the classifier with the highest accuracy score: RBF Kernel and PCA 180 features dimension (corresponding to 98% contribution).

		Predicted labels						
		Happy	Sad	Angry	Scared	Neutral	Dis-gusted	Sur-prised
Actual labels	Happy	***65.9%***	*4.9%*	*7.3%*	*0.0%*	*7.3%*	*14.6%*	*0.0%*
	Sad	*17.9%*	***61.5%***	*7.7%*	*7.7%*	*0.0%*	*0.0%*	*5.1%*
	Angry	*7.9%*	*5.3%*	***63.2%***	*2.6%*	*0.0%*	*5.3%*	*15.8%*
	Scared	*5.3%*	*5.3%*	*0.0%*	***76.3%***	*7.9%*	*2.6%*	*2.6%*
	Neutral	*10.3%*	*5.1%*	*7.7%*	*5.1%*	***53.8%***	*10.3%*	*7.7%*
	Disgusted	*4.5%*	*0.0%*	*4.5%*	*4.5%*	*6.8%*	***72.7%***	*6.8%*
	Surprised	*3.3%*	*20.0%*	*3.3%*	*6.7%*	*6.7%*	*3.3%*	***56.7%***

Figure: Confusion Matrix of best classifier

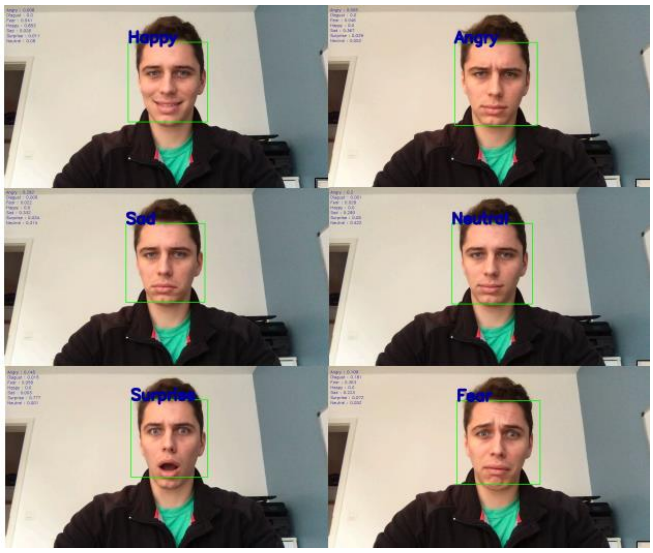
As can be seen above, Surprise and Neutral emotions were classified with the poorest accuracy compared to other emotions such as Scared and Disgust who achieved the highest results (respectively 76% and 73%). RAVDESS database contains speeches for 7 different emotions but we decided to remove Surprise, as our classifier had trouble differentiating it from other emotions. Final results have been quite satisfying. We have succeeded to obtain an accuracy score of almost 75% as shown in following table.

		Predicted labels					
		Happy	Sad	Angry	Scared	Neutral	Dis-gusted
Actual labels	Happy	***80.0%***	*0.0%*	*5.7%*	*5.7%*	*5.7%*	*2.9%*
	Sad	*8.1%*	***81.1%***	*0.0%*	*0.0%*	*2.7%*	*8.1%*
	Angry	*6.3%*	*6.3%*	***75%***	*0.0%*	*6.3%*	*6.3%*
	Scared	*6.7%*	*0.0%*	*4.4%*	***71.1%***	*8.9%*	*8.9%*
	Neutral	*11.1%*	*5.6%*	*2.8%*	*8.3%*	***66.7%***	*5.6%*
	Disgusted	*0.0%*	*8.7%*	*0.0%*	*4.3%*	*2.2%*	***84.8%***

Potential improvements: Our model presents reasonably satisfying results. Our prediction recognition rate is around 65% for 7-way (happy, sad, angry, scared, disgust, surprised, neutral) emotions and 75% for 6-way emotions (surprised removed). In order to improve our results and to try to get closer to the state of the art, we will try to implement more sophisticated classifiers in second period of this project. For example, Hidden Markov Model (HMM) and Convolutional Neural Networks (CNN) seem to be potential good candidates for speech emotion recognition. Unlike SVM classifiers, those classifiers are train on short-term features and not on global statistics features. HMM and CNN are considered to be advantageous for better capturing the temporal dynamic incorporated in speech. To implement a multimodal model for emotion recognition we will also need to set up the removal of silence and probably build a speaker identifier to not bias our emotion predictions in the speech domain.

VIDEO RESULTS:

The webcam facial emotion algorithm presented above can be illustrated with a quick example. As shown on figure , the classification seems to work well in practice. There are however still many sources of improvements.



Potential improvements: Our model seems to work reasonably well. The accuracy reached 61%, which is still 14% behind state-of-the-art models. We seem to face an issue of overfitting. For this reason, we will include drop out layers in future model architectures. Moreover, it is very hard to obtain a prediction of ”disgust”. Our initial model is imbalanced, and this seems to have an effect on this class. We need targeted data augmentation on this specific class for example. 46 Another way to improve the model and approach state of the art solutions is to include extracted facial features in the design matrix. The extraction in state of the art models is usually made using histograms of oriented gradients (Hog) on sliding windows, but also take into account face landmarks. Finally, in terms of graphical display, a potential improvement, instead of displaying probabilities on the top corner, would be to display graphs instead.

V. CONCLUSION

To conclude, it is possible to construct rather accurate classifiers for both personality traits and emotions recognition for different input types considered separately, each modality requiring its own set of features and hyper-parameters. The following steps for our project will be to design an ensemble model capable of combining the insights gained from both personality traits detection and emotions recognition in order to provide a broader assessment of a user’s interview. Our final model would include a type of coherence measure expressing the similarity between a specific user’s emotional profile and the average characteristics of people in the same psychological category according to the Big Five model. This would typically imply unsupervised clustering techniques.

REFERENCES

[1] B.Kratzwalda, S.Ilie´, M.Kraus, S.Feuerriegel, H.Prendinger. Deep learning for affective computing: text-based emotion recognition in decision support, Sep. 2018. https://arxiv.org/pdf/1803.06397.pdf

[2] N.Majumder, S.Poria, A.Gelbukh, E.Cambria. Deep Learning-Based Document Modeling for Personality Detection from Text, 2107. http://sentic.net/deep-learning-based-personality-detection.pdf

[3] The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), https://zenodo.org/record/1188976/?f=3.XAcEs5NKhQK

[4] B.Basharirad, and M.Moradhaseli. Speech emotion recognition methods: A literature review. AIP Conference Proceedings 2017. https://aip.scitation.org/doi/pdf/10.1063/1.5005438

[5] L.Chen, M.Mao, Y.Xue and L.L.Cheng. Speech emotion recognition: Features and classification models. Digit. Signal Process, vol 22 Dec. 2012.

[6] T.Vogt, E.Andre´ and J.Wagner. Automatic Recognition of Emotions from Speech: A Review of the Literature and Recommendations for Practical Realisation. Affect and Emotion in Human-Computer Interaction, 2008.

[7] T.Vogt and E.Andre´. Improving Automatic Emotion Recognition from Speech via Gender Differentiation. Language Resources and Evaluation Conference, 2006.

[8] T.Giannakopoulos. pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. Dec. 2015https://doi.org/10.1371/journal.pone.0144610

[9] T.Giannakopoulos. pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. Dec. 2015https://doi.org/10.1371/journal.pone.0144610

[10] The Facial Emotion Recognition Challenge from Kaggle, https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression recognition-challenge/data

[11] C.Pramerdorfer, and M.Kampel. Facial Expression Recognition using Con volutional Neural Networks: State of the Art. Computer Vision Lab, TU Wien. https://arxiv.org/pdf/1612.02903.pdf

[12] OpenCV open source library for image feature extraction, https://opencv.org/ 13. End-to-End Multimodal Emotion Recognition using Deep Neural Networks, https://arxiv.org/pdf/1704.08619.pd

[13] E. Cambria, "Affective Computing and Sentiment Analysis," IEEE Intelligent Sys tems, vol. 31, no. 2, 2016, pp. 102–107.

[14] E. Cambria et al., "SenticNet 4: A Se mantic Resource for Sentiment Analysis based on Conceptual Primitives," Proc. 26th Int'l Conf. Computational Lin guistics, 2016, pp. 2666–2677.

[15] J. Digman, "Personality Structure: Emer gence of the Five-Factor Model," Ann. Rev. Psychology, vol. 41, 1990, pp. 417–440.

[16] F. Mairesse et al., "Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text," J. Artificial Intelligence Research, vol. 30, 2007, pp. 457–500.

[17] T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," Computing Research Repository (CoRR), 2013; http://arxiv.org/abs/1301.3781.

[18] J.W. Pennebaker and L.A. King, "Linguistic Styles: Language Use as an Individual Difference," J. Personality and Social Psychology, vol. 77, no. 6, 1999, pp. 1296–1312.

[19] M. Coltheart, "The MRC Psycholinguis tic Database," Quarterly J. Experimental Psychology, vol. 33A, 1981, pp. 497–505.

[20] M.D. Zeiler, "ADADELTA: An Adap tive Learning Rate Method," Computing Research Repository (CoRR), 2012; https://arxiv.org/abs/1212.5701.

[21] S.M. Mohammad and P.D. Turney, "Crowdsourcing a Word-Emotion Association Lexicon," Computational Intelligence, vol. 29, no. 3, 2013, pp. 436–465.

[22] S. Mohammad and P. Turney, "Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon," Proc. NAACL-HLT Workshop Computational Approaches to Analysis and Generation of Emotion in Text, 2010, pp. 26–34.

[23] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Machine Learning Research, vol. 12, Oct. 2011, pp. 2825–2830. 12. S.M. Mohammad and S. Kiritchenko, "Using Hashtags to Capture Fine Emotion Categories from Tweets," Com putational Intelligence, vol. 31, no. 2, 2015, pp. 301–326.

[24] B.G. Patra, D. Das, and S. Bandyopad hyay, "Multimodal Mood Classifica tion Framework for Hindi Songs," Computación y Sistemas, vol. 20, no. 3, 2016, pp. 515–526

[25] C.-H. Wu and W.-B. Liang, "Emotion Recognition of Affective Speech Based on Multiple Classifiers Using Acoustic-Prosodic Information and Semantic Labels," IEEE Trans. Affect. Comput., vol. 2, no. 1, pp. 10–21, Jan. 2011.

[26] S. S. Narayanan, "Toward detecting emotions in spoken dialogs," IEEE Trans. Speech Audio Process., vol. 13, no. 2, pp. 293–303, Mar. 2005.

[27] B. Yang and M. Lugger, "Emotion recognition from speech signals using new harmony features," Signal Processing, vol. 90, no. 5, pp. 1415–1423, May 2010.

[28] C.-C. Lee, E. Mower, C. Busso, S. Lee, and S. Narayanan, "Emotion recognition using a hierarchical binary decision tree approach," Interspeech, vol. 53, pp. 320–323, 2009.

[29] S. Bjorn, S. Steidl, and A. Batliner, "The INTERSPEECH 2009 Emotion Challenge," 2009.

[30] J. P. Arias, C. Busso, and N. B. Yoma, "Shape-based modeling of the fundamental frequency contour for emotion detection in speech," Comput. Speech Lang., vol. 28, no. 1, pp. 278–294, Jan. 2014.

[31] M. Grimm, K. Kroschel, E. Mower, and S. Narayanan, "Primitives-based evaluation and estimation of emotions in speech," Speech Commun., vol. 49, no. 10–11, pp. 787–800, Oct. 2007.