Emotion recognition, from speech to code, models for classification of emotions

Federico Fiorio, Student, Università degli Studi di Milano, Via Celoria 18, Milano, Italy,
federico.fiorio@studenti.unimi.it

Abstract—All humans have the ability to recognize others people's feelings, this ability is called empathy, some humans are better than others in doing this, but this also happens in the world of machine learning.
With the help of good features extraction and a good dataset to rely on, this task it's not impossible even for machines. Some models can be considered better than others at being empathetic.
In this paper I will discuss the differences between knn, decision tree, random forest and CNN models at being empathetic.

Index Terms—Machine learning, Knn, Decision tree, Random forest, Cnn, empathetic, Models.

1 Introduction

The ability to recognize others people's feeling can be underestimated, if anyone could deal perfectly with emotions of a human being, that person will have an advantage over that person, in this case he would have a better understanding of the needs of that person, a better understanding of why the other person is saying certain things, and so in general would understand better the message that the other person will try to tell us; In fact a normal conversation have different factors to take into account, for example: a rich vocabulary, how the person is behaving, how he/she is moving, and also what the person feels when he/she talks to us.
Detecting emotions is one of the most important marketing strategies in today's world. You could personalize different things for an individual specifically to suit their interest.
Emotions play an important role, what if a business tries to detects emotions of their customers on phone calls to better understand how to treat them, what are their needs.
This kind of idea can be simply applied to chat bots that responds to phone calls, or also to try to understand if the customer needs really hard to speak with another person.
Speaking of businesses, this is the era of internet and the era where video games are everywhere, not only on our pc but also in virtual reality.
A good emotion recognition system can be implemented in the behaviour of the various NPCs of the game so they can better interact with our player, that, of course, in a virtual reality environment will speak to them directly.
This ability will make the game more immersive and also more enjoyable to play.
The possibilities are endless, and this is why the possibility to have a good classifier can help businesses to grow and to open them to knew possibilities, and also to help the customers by treating them better.
The approach adopted in this paper consists in using 4 different classifiers and the RAVDESS dataset [3] to try to gain the best result.
All the 4 classifiers are well known in the machine learning field, but the one that deserves most attention is the CNN, in fact it's the model with the highest accuracy in prediction among the 8 possible emotions in the RAVDESS dataset [3].

2 Related work

The audio recognition is becoming really relevant in recent years, many different models had been tried out to achieve the best results. One of the work to be cited is the one by Iqbal, A., and Barua [1], in this work the main models adopted are SVM and KNN.
The work done by Iqbal was performed on the RAVDESS (male), RAVDESS (female), RAVDESS (combined) and SAVEE; However in this work we will only take care of the results on the RAVDESS dataset.
In RAVDESS (male) SVM and KNN have 100% accuracy in both anger and neutral. But in happiness and sadness Gradient Boosting performs better than SVM and KNN. In RAVDESS (female) SVM achieves 100% accuracy in anger as same as male part. SVM has overall good performance except in sadness. Performance of KNN is also good in anger and neutral like 87% and 100% respectively . In anger and neutral, Gradient Boosting performs poorly. KNN performance is very poor in happiness and sadness comparing with other classifiers. In male and female combined dataset, performances of SVM and KNN are really good in anger and neutral rather than Gradient Boosting. KNN's performance is really poor in happiness and sadness. Average performances of classifiers in male dataset are better than female dataset except SVM. In combined database, SVM get high accuracy than gender based datasets.
Another model that keeps getting more attention is the neural network CNN, used for various tasks like image classification, Object detection for self-driving cars, and also for audio speech recognition.
The proposed work by Ansari [2], have results with an accuracy of 79.33% with the use of a CNN on the RAVDESS dataset [3], but only on five of the eight possible emotions.
Even if the results with the CNN are not impressive in Ansari work, the model have high potential and possibility to perform better, in this work the CNN adopted on the same dataset could achieve an accuracy over 88%.

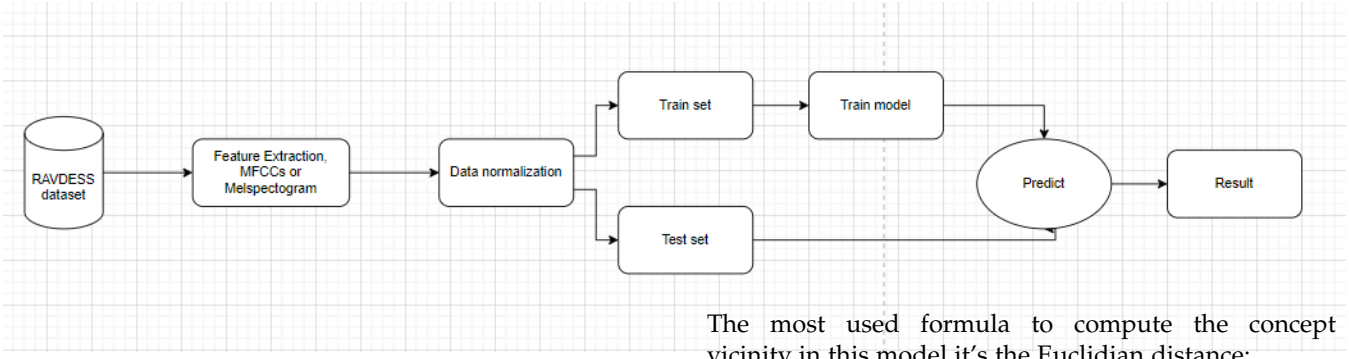

Fig. 1. The flow adopted for the models proposed

3 The method and models adopted

3.1 method description

The method adopted can be visualized in the Fig. 1, as we can see the features extracted from the RAVDESS dataset [3] are mainly the MFCCs, but also some experiments were tried with the melspectogram but they were disappointing, so the reported work does refer only to MFCCs. MFCC also known as Mel-Frequency Cepstral Coefficients, are actually a type of cepstral representation of the signal, where the frequency bands are distributed according to the mel-scale, instead of the linearly spaced approach.
After we got the features on which we will train the models, there is and intermediate step called data normalization, this step makes features equally weighted, it is calculated by subtracting mean (u) from each feature and divided by standard deviation (s):
$z = (x - u)/s$
Standardization of a features is a common requirement for many machine learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data.
The data is divided in two main sets, the training set and the test set.
The test set size is 33% of the total data, meanwhile the training set is 67% of the total data, these dimensions keep enough data in the training set so the model can learn enough without overfitting too much and then the test set can verify that the model have learned to generalize well enought.

The models of classification of emotions here proposed are: Knn, decision tree, random forest, and last one, based on a deep learning strategy named convolutional neural network (CNN).

3.2 Knn

The first model also known as K-nearest neighbour, utilises K "closest" points for performing classification, these "closest" points are referred to as nearest neighbours.
The main idea behind this model is that, if the points (datas) are near each others than they will be considered similar.
The most used formula to compute the concept of vicinity in this model it's the Euclidian distance:
$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$
This concept of distance can be generalized by the Minkowski distance.
In the Knn adopted in this paper, the distance chosen it's the Euclidian, with a k = 5, the accuracy that I was able to achieve it's around 70%.

3.3 Decision tree

The second model described it's the decision tree.
The model presented utilises Gini as the function to measure the quality of a split (quality of impurity), it means that the possible gain depends on the difference of the Gini functions before the split and after the split.
$Gain = P–M$
Where P it's the impurity measured before the split and M it's the impurity measure after the split. A node will be split if this split induces a decrease of the impurity.
The decision tree model was able to achieve a 81% accuracy on the test set.

3.4 Random forest

The random forest model is an example of ensemble technique. From the training set more than one classifiers are built and the result on the training set is gained by combining the results of the classifiers created previously.
Decision trees are highly sensitive to training data, so the model might fail to generalize, here comes into help the random forest algorithm. In fact, it is a collection of random decision trees and it is much less sensitive to training data.
In order to create a random forest the dataset must be boostrapped into multiple datasets, after this phase, some trees are created and trained on different features of the datasets. After the training phase the prediction will depend on the predicted class that received the highest number of votes by the previously created random trees (if we use the majority voting to aggregate the prediction).
The boostrapping phase consists into creating multiple dataset from the original one by choosing random samples from it that can be chosen more than one time, so every time a new sample is chosen it can be taken from all the possible samples of the original dataset. Boostrapping is essential to train the trees on different data so the model will be less sensitive to the training data; also the random features selection keeps less correlation between the trees.
he random forest adopted in this paper consists in:

- 22000 trees;

- the measure for impurity and so for the splitting of nodes in trees it's still the Gini function;
- the trees have a maximum depths of 10;
- The minimum number of samples required to be a leaf node it's 3;
- The number of features to consider when looking for the best split is $log2(n_features)$;
- The minimum number of samples to split a node is 20;

This type of random forest was able to achieve an accuracy of around 82% after the hyper parameters tuning performed via a GridSearch approach.

3.5 CNN

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 40, 128)           768

 activation (Activation)     (None, 40, 128)           0

 dropout (Dropout)           (None, 40, 128)           0

 max_pooling1d (MaxPooling1D  (None, 5, 128)           0
 )

 conv1d_1 (Conv1D)           (None, 5, 70)             44870

 activation_1 (Activation)   (None, 5, 70)             0

 dropout_1 (Dropout)         (None, 5, 70)             0

 flatten (Flatten)           (None, 350)               0

 dense (Dense)               (None, 10)                3510

 activation_2 (Activation)   (None, 10)                0

=================================================================
Total params: 49,148
Trainable params: 49,148
Non-trainable params: 0
_________________________________________________________________
```

Fig. 2. Architecture of the CNN

The last model it's the convolutional neural network or CNN.
A CNN is able to capture patterns across frequency and time for given input spectograms.
This type of network it's feed forward and similar to the multilayer perceptron, the main difference it's in the hidden layers or also called convolutional layers, these layers perform a convolution operation utilizing some filters on the data in input, we can think of these filters as patterns detectors. Typically filters are matrices of random initialized numbers that will change during training phase.
Another fundamental layer in the CNN architecture it's the pooling, this layer serves to progressively reduce the spatial size of the representation, to reduce the number of parameters, amount of computation in the network, and hence to also control overfitting. This is known as downsampling.
In the proposed model the CNN is composed like the Fig.2, The network is able to work on vectors of 40 features for each audio file provided as input.
The architecture of the CNN is a combination of convolutional operations, maxpooling and dropout. The dropout it's mainly utilized for the training part, to avoid overfitting, in fact it deactivates some neurons meanwhile the training phase is in progress, it is essential to avoid that the model learns too much on the training data and can't really generalize well.
The activation functions for the input and hidden layer are the Relu functions.
The Relu function can be defined as:
$f(x) = max(0, x)$
It's the one used in this model so it is able to converge quickly.
The hidden layer it's composed by 70 neurons to try to satisfy the empirical rule:
$hiddenlayer = (inputneurons + outputneurons)/2$
It couldn't be demonstrated theoretically but the results over experiments seems to be better with an hidden layer of that specified size.
For the output layer we got a dense layer (fully connected layer) with an activation function called softmax, it is the most used for classification problems as it used to normalize the output of a network to a probability distribution over predicted output classes; It gives a probability for each class and they sum up totally to 1. The model will make it's prediction based on the class with highest probability.
The flatten part connects the convolutional parts of the layer with the Dense parts. Flatten simply takes all dimensions and concatenates them after each other.
This model was able to achieve an accuracy of 86%.

4 Results and Dataset description

The dataset on which the training and tests were performed it's the RAVDESS dataset [3].
The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) contains 7356 files. 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. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.
The dataset contains videos, songs, and speech, for the purpose of this research, only the speech audio files were used.
Each of the 7356 RAVDESS files has a unique filename. The filename consists of a 7-part numerical identifier (e.g., 02-01-06-01-02-01-12.mp4). These identifiers define the stimulus characteristics:

- Modality (01 = full-AV, 02 = video-only, 03 = audio-only);
- Vocal channel (01 = speech, 02 = song);
- Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised);
- Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion;

- Statement (01 = “Kids are talking by the door”, 02 = “Dogs are sitting by the door”);
- Repetition (01 = 1st repetition, 02 = 2nd repetition);
- Actor (01 to 24. Odd numbered actors are male, even numbered actors are female);

All the models except for the Knn were able to achieve an accuracy equal or higher than 80%.
The main model that is taken mostly in consideration for the results is the CNN, in Table 1 we can see how the model performed on the various classes of the dataset, in Table 2 we can see the comparison of results on the models talked about in this paper, in Table 3 we can see the comparison between the same CNN on different features for the training phase, in Table 4 we can see a comparison over some others models.

The CNN overall seems to be the model with higher results, it outclasses others models on Calm, Sad, Disgust and Surprised features.(Table 2)
It also performs good in comparison to other state-of-the-art models.(Table 4)
The CNN seems to be working better with MFCCs than melspectograms, the same CNN trained on melspectograms have almost half of the precision. (Table 3)
Since the CNN is considered a deep neural network model, having the possibility to access more data for the training can bring an increase of results on this model.
A possible solution it’s to use also the RAVDESS dataset [3] video files, by extracting the audio files from them, or by using a technique called data augmentation.
Data augmentation can bring us more possible data to use, it consists into creating more data by modify the original one with some little random noise.

	precision	recall	f1-score	support
Neutral	0.84	0.86	0.85	56
Calm	0.90	0.92	0.91	130
Happy	0.81	0.90	0.85	137
Sad	0.82	0.81	0.82	121
Angry	0.88	0.84	0.86	125
Fearful	0.82	0.83	0.83	120
Disgust	0.93	0.77	0.84	136
Surprised	0.86	0.91	0.88	137
Accuracy			0.86	962
M. avg	0.86	0.86	0.86	962
W. avg	0.86	0.86	0.86	962

TABLE 1
Performance of CNN over the test set

	Knn	Decision Tree	Random Forest	CNN
Neutral	0.42	0.72	**0.90**	0.84
Calm	0.73	0.83	0.71	**0.90**
Happy	0.67	0.85	**0.87**	0.81
Sad	0.68	0.76	0.79	**0.82**
Angry	0.71	**0.88**	**0.88**	**0.88**
Fearful	0.70	**0.88**	0.80	0.82
Disgust	0.87	0.73	0.86	**0.93**
Surprised	0.72	0.76	0.77	**0.86**

TABLE 2
Comparison of models based on precision

	CNN melspectograms	CNN MFCCs
Neutral	0.07	**0.84**
Calm	0.42	**0.90**
Happy	0.51	**0.81**
Sad	0.31	**0.82**
Angry	0.76	**0.88**
Fearful	0.48	**0.82**
Disgust	0.32	**0.93**
Surprised	0.47	**0.86**

TABLE 3
Comparison of the same CNN architecture but with different features as input (using precision to compare)

	SVM [1]	SVM [4]	My CNN
Neutral	**0.93**	0.76	0.85
Calm	’	0.88	**0.91**
Happy	0.66	**0.92**	0.85
Sad	0.67	0.71	**0.82**
Angry	**1.0**	0.86	0.86
Fearful	’	**0.86**	0.83
Disgust	’	’	**0.84**
Surprised	’	0.71	**0.88**

TABLE 4
Comparison with other papers based on f1-score and on the RAVDESS dataset

5 Conclusion

In this paper we have discussed 4 different models: Knn, Decision tree, Random forest, CNN, all trained and tested on the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [3]. We saw the comparison on performances on these models and the classes in which they are the best.
The deep learning approach seems to work out better than the others, the are also possibilities to gain higher accuracy with the inclusion of more data.
The dominant feature used for the training is the MFCC.
The other models different from the CNN can be used in applications where there is little or no time for training, the overall results of decition trees, and random forest aren’t bad either and they might be taken into consideration for future use.
The CNN achieved best results and it’s the advised model to start for further experimentations and analysis.

References

[1] IQBAL, A., AND BARUA, K. A real-time emotion recognition from speech using gradient boosting. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) (2019), IEEE, pp. 1–5.

[2] Abdul Ajij Ansari, Ayush Kumar Singh, Ashutosh Singh, Speech Emotion Recognition using CNN.

[3] Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391. https://doi.org/10.1371/journal.pone.0196391.

[4] ZHANG, B., ESSL, G., AND PROVOST, E. M. Recognizing emotion from singing and speaking using shared models. In 2015 International Conference on Affective Computing and Intelligent Interaction (ACII) (2015), IEEE, pp. 139–145.