Vrije Universiteit Amsterdam

Bachelor Thesis

A study on emotions and voice: comparing different machine learning algorithms in conjunction with filters

Author: Robin Herlan 2594618

1st supervisor:	Florian Kunneman
daily supervisor:	Daniel Formolo
2nd reader:	Emmeke Veltmeijer

A thesis submitted in fulfillment of the requirements for the VU Bachelor of Science degree in Computer Science April 18, 2021

Abstract

Emotion recognition is becoming more relevant in our society where most services are digitalized and automated and as such there is a need for machine learning algorithms that can accurately infer the emotion of individuals in order to provide a service or to carry out a specific function. This paper presents a study of emotion recognition in noisy environments with focus on using different filters and machine learning algorithms. The research is focused on distinguishing happy from neutral emotion, which can be useful in fields such as voice assistants. Results are compared between Support Vector Machines, Random Forest and K-Nearest Neighbours. Samples were collected from YouTube and peer-reviewed for the two mentioned emotions. The filters that were chosen to be investigated are the lowpass and highpass filter from the ffmpeg software. The experimental results reveal that SVM performed best from the three algorithms at classifying both the happy and neutral emotion with an F1-score of 0.762 and 0.737 respectively.

Background Information

The Basic Emotion Theory [9] puts forth that there are a limited number of basic emotions that humans express in a similar manner. The theory was first started by Charles Darwin in 1872 followed by Paul Ekman and many other psychologists [10]. The six basic emotions in the theory are; anger, joy, disgust, surprise, sadness, and fear. As humans, we exhibit these emotions on a day-to-day basis through non-verbal cues like facial expressions, body language and verbal signals such as voice. It has become widely accepted that emotions are essential for human-to-human interaction and in order to form interpersonal relationships. As such, when Picard introduced affective computing in the 1990s, scientists strived to find ways to measure human emotions through many different sources such as by detecting facial expressions or by analysing voice. Measuring emotions through voice can be challenging due to several reasons. For example, certain emotions have only very subtle prosodic differences which would make it hard for an algorithm to distinguish. Calm and relaxed are an example of emotions which are similar. Additionally, it is important to select the correct features of the voice signals in order to reliably identify the correct emotion and to distinguish between different emotions. Since most languages on earth have different accents and speaking styles, this creates another layer of complexity for an emotion recognition algorithm because such characteristics directly influence the voice signals by changing the intensity and pitch of the signal [11, 12].

Introduction

Emotion recognition algorithms are useful especially in the field of voice assistants where computers are required to interact vocally with a person. Most emotion recognition algorithms will have to work in real environments with background noise and thus it is important to develop and research these systems further. Real environments means that there will be background noise which can negatively impact a voice signal. This would make it harder to detect the emotion in that signal [17]. It is likely that the noise will always be different as real life environments are unpredictable hence the effect on the voice signal cannot be confidently foreseen. An example of different noise can range from music to cars or even other people speaking. It is possible to use filtering techniques to drown out the noise, but this would require removing parts of the signal. This means that a balance is needed between removing noise and still maintaining enough of the signal to confidently and accurately infer the intended emotion. Since the intensity of the noise will differ for different signals, this could have the adverse effect of either removing too much or too little of the noise which would result in the wrong/no emotion being identified.

This gives rise to the research question: To what extent do different machine learning algorithms in conjunction with filters affect the accuracy of classifying emotions in noisy speech?

This work uses different filters to try to reduce the background noise present in voice samples. The voice samples were collected from YouTube and peer-reviewed to make sure the emotions attributed to them are happy and neutral, which have a high difference in arousal making recognition easier for machine learning algorithms. These samples were used to create a model with three different algorithms, namely SVM, RF and KNN. Various metrics such as the confusion matrix and F1-score for both tested emotions are reported for the individual algorithms. The results are analysed and discussed in detail with future research in mind.

The paper is structured as follows. Section 2 is a review of relevant literature. Following this, Section 3 introduces and discusses the methodology for the proposed research in detail. Section 4 contains the experimental setup such as cross-validation results and chosen hyperparameters. Next, in Section 5, the results are reported and in Section 6 these results are discussed in detail. Section 7 summarizes the findings of the research and gives a conclusion based on the research question. Lastly, Section 8 highlights limitations that were encountered during the making of this project and finishes with discussing possible future directions of the research.

Literature Review

Being able to detect the emotion in a person's voice can be very important in certain fields that employ speech recognition like ticket booking stations, the medical field, etc. Such systems will have to be able to detect a person's voice and extract their emotions, often in a noisy environment. There have been many papers and studies about extracting a person's emotion through their voice [1, 7, 11]. These studies use acted audio samples from a controlled environment in order to get more accurate results, meaning there is no background noise that could interfere with an emotion recognition algorithm. However, there have not been too many studies which use real audio samples, meaning that there is noise in the background that could make emotion recognition more difficult.

Some research papers that dive into the topic of emotion recognition in noisy speech are the following. In [5], Schuller et al. presents the results of using SVM on samples from three different databases. These databases contain samples that are acted and recorded in a controlled environment. Schuller at al. adds white noise at different noise levels to the samples in order to receive varying results. The work uses the six basic emotions in addition to the neutral emotion. The study found that the best results were 71.11% on samples with added noise and 87.5% on a clean database. Schuller et al. argues that this difference is due to feature selection and the difficulty to select optimal features with varying noise conditions.

Next, [14] presents a similar approach as [5]. You et al. shows the results of adding white noise and sinusoid noise to a mandarin acted database containing the six basic emotions on SVM. The noise was added at varying signal-to-noise (SNR) ratios using the Lipschitz embedding method. The reported results show that with the above mentioned embedding method, SVM has a recognition accuracy of around 70-75% for varying SNR ratios.

Chenchah et al. shows an approach to reduce noise levels from different samples using different methods of speech enhancements such as wiener filter and classifies these using Hidden Markov Models (HMM) [15]. Chenchah et al. adds 4 different real world noises to an acted database, namely; car, babble, train and airport. These samples are then pre-processed using the previously mentioned speech enhancement methods. The study reports that the speech enhancement methods of spectral subtraction and MMSE significantly increased the recognition rate of samples with the airport and babble noise, but they were not effective at reducing car noise.

Lastly, Sztahó et al. used a database of spontaneous telephone speech and records of different talk shows, which were classified using automatic emotion recognition with SVM [16]. In Sztahó et al's. study, 4 different emotions (angry, happy, sad and neutral) were tested with different specific feature vectors and the recognition results were reported. The best result that SVM received was 66.27 with the angry emotion having the best recognition.

Most of these papers have one main thing in common. They add noise to acted databases. Many papers do this in order to research emotion recognition with noise. There is a significant gap of research on genuine real emotions exhibited in real situations with background noise, which reinforces the need for the research in this paper.

Research Methodology

Data Collection

Different databases were taken into consideration when choosing the best one. For example, the VOiCES database was a strong candidate. It has multiple different speakers with different background noise. However, these samples were recorded in a controlled environment with the noise being played from different speakers in the room. While this could have been used for the study, it was important that the samples are from real situations with the speakers genuinely experiencing the emotions. Due to a lack of available databases that have "real" samples with speakers that exhibit emotion, the data was retrieved from YouTube clips. Attributes that were taken into account when selecting the samples were the amount and type of background noise, the emotion exhibited, the gender of the person, and the length of the samples. Due to time constraint, the scope of the research was limited to two emotions, happy and neutral. The reason for choosing these two emotions in particular is because happy has a high arousal compared to neutral, which means that algorithms may be able to differentiate them better. Additionally, it was difficult to find clips on YouTube with people experiencing other emotions, like the sad or angry emotion with background noise. The type of background noise that was chosen was music. This made the sample collection a little bit easier as samples can be collected from videos taken at music festivals. People are usually happy at festivals and as such this was a good source to take the samples from. Each sample ranged from 3-7 seconds in length as the full sentence of the speaker was usually included. This way 60 samples were retrieved in total, evenly split to the two emotions happiness and neutral and the gender of the speaker. This means that there is an even number of males and females in the samples. 5 samples were used for each speaker resulting in 12 unique speakers. The samples were peer-reviewed by two people who had to agree to the emotion label attributed to a sample for it to be chosen. These people performed the emotion labelling in different environments with different headphones to ensure that the emotion label is the same for these environments. When the two people disagreed with a label on a sample, the sample was discarded and replaced. The samples were then split into 40:20 which were used for training/validation and testing sets respectively.

Sample Pre-processing and Filters

Before any sort of training could take place, the samples had to be pre-processed. This includes converting them into the right format of .wav, and applying audio filters in-order to reduce the noise. The software that was used for both of these steps is ffmpeg. The filters that were used to pre-process the samples were a combination of the lowpass and highpass filters. The filter selection is described in more detail in the experimental setup section. With the lowpass and highpass filter it is possible to cut out certain frequencies from the samples. This means that all frequencies other than human voice frequencies can be removed. The frequency range that human voice is attributed to is usually between 200 to 3000 Hz. However, this does not mean that only human speech is left when applying these filters. This is due to the fact that background noise can have a variety of frequencies that could intersect with the above described frequency range. The filters were tested extensively with different options, alone and together in-order to determine the best accuracy on the validation set.

Classifiers

Support Vector Machine (SVM)

SVM’s are supervised classification algorithms and they can distinguish between at least two groups of data. In order to differentiate two groups, the goal of SVM is to find a hyperplane that distinctly separates the data points. It achieves this by computing the maximum distance between the hyperplane and the closest instances of data. The distance is also known as the margin and the closest data points to the hyperplane are the support vectors. An instance of data that falls on either side of the hyperplane will be categorized as the respective group or class. The process described above is visualized in Fig 1.1.

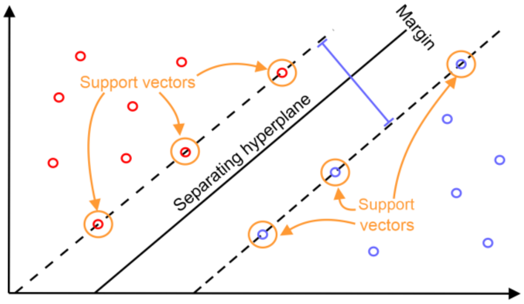

Fig 1.1: A depiction of the SVM algorithm showing the hyperplane, margin and support vectors

SVM was chosen for this study because it is a popular choice in classification problems and widely used in different research including but not limited to emotion recognition due to their performance and generalization capabilities [2, 3, 5]. Additionally, it has been shown to be among the top performing algorithms for different classification problems [4]. In [1], Yacoub et al. shows that when available training data is scarce, SVM outperforms other tested algorithms, which is relevant to this research as there are not many training samples.

Random Forest (RF)

The RF algorithm creates many different decision trees that work together as an ensemble. Each tree in the forest computes a separate class prediction and the class that is predicted by the most trees becomes the chosen prediction. Since RF aggregates the predictions of individual decision trees, it has good generalization [8]. A decision tree in the forest comes to a prediction based on the best feature from a random subset of features. This means that when splitting a node, the random forest only takes into consideration a subset of features instead of all available features. An example of a RF can be seen in Fig 1.2

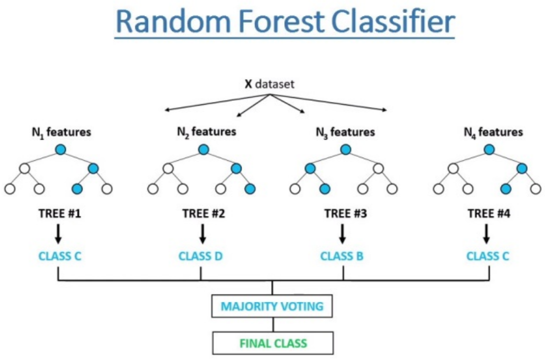

Fig 1.2: A depiction of the RF algorithm showing the ensemble of decision trees and majority prediction

The Random Forest algorithm was picked because it is based on the decision tree classifier. According to different research papers, decision trees and in particular the Random Forest algorithm perform well in the task of emotion recognition [6,7]. In addition, Random Forest is good at avoiding overfitting since it creates trees by using random subsets of features and training data [8].

K-Nearest Neighbours

KNN is a very simplistic but effective supervised learning algorithm. It works under the assumption that similar data is grouped closely together. This means that if there is a data point x, it is reasonable to assume that k closest data points are of the same class as x. When the algorithm needs to classify a new data point, it takes into consideration the value of k and draws a metaphorical circle around itself which encompasses the k nearest data points. It determines which data points are closest through the Euclidean distance also called the straight-line distance. KNN then assigns the data point based on majority voting. An example of KNN can be seen in Fig 1.3.

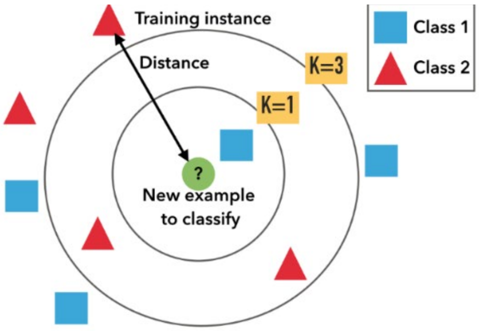

Fig 1.3: A depiction of the KNN algorithm showing the process of assigning a new data point

KNN was chosen because like SVM, it is often used in emotion classification tasks and sometimes in order to compare it to other classifiers [4].

Evaluation

Scoring Metrics

One of the metrics used to interpret the results was the F1-score. The F1-score is a statistical analysis which puts emphasis on the falsely classified samples of a test. It is calculated using the harmonic mean of precision (Equation 2) and recall (Equation 3). The F1-score offers advantages over traditional accuracy because the latter only looks at all correctly identified cases whereas the F1-score also takes into account the incorrectly identified cases. This is especially important in situations where there is an imbalance in class distribution. Equation 1 shows the exact formula of the F1-score. The best score possible is a 1, which means that both the precision and recall have a perfect score and the worst score is a 0, which means that either the precision or recall is 0. While the F1-score will be the prevalent scoring metric, the accuracy, precision and recall will also be calculated and reported in order to have a more in-depth analysis of the results.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$ Equation 1

$$precision = \frac{TP}{TP+FP}$$ Equation 2

$$recall = \frac{TP}{TP+FN}$$ Equation 3

Learning Rate

The learning rate was considered in order to see whether the number of samples played a role in the classifiers accuracy. A plateau in the graph after a certain number of samples indicates that even if more samples were added to the training set, the overall accuracy would not increase. However, if the learning curve does not plateau and there is a positive trend in the graph, then it is reasonable to assume that increasing the number of samples could increase the classifiers accuracy. This learning curve was calculated and graphed for both SVM and RF, one with the default ordering in the local system and one with random ordering. The random ordering was calculated over 10 runs and the average accuracy and standard deviation (SD) were calculated and graphed. For each of the 10 runs, a different model was made with an incrementing number of samples and tested against the entire test set. For example, a model with 1 sample from each class, happy and neutral, was tested against the test set and the results recorded. Then the number of samples was incremented by 1 for each class and this process was repeated again for all samples in the training set. The averages were then calculated and a different run was started from the beginning. The learning curve was not calculated for KNN. The reason for this is due to KNN needing at least the number of samples that the parameter N-neighbours is set to. In this case, hyperparameter search found that the optimal number for this is 12. This means that KNN needs at least 12 samples to create a model and hence the learning curve could not be plotted.

Hyperparameter Selection

In order to find the optimal hyperparameters for each algorithm, grid search with 5-fold cross validation was used on the training set to cover a range of parameters. It is important to have a balance between computation time and the amount of parameters searched, as more parameters means computation time is higher. This is especially true for big datasets with thousands of samples. For SVM and KNN the range of parameters that was searched was quite high since the computation time of these algorithms is low with the amount of training samples. In comparison, Random Forest took the longest to compute because there were many possible options and parameters to consider. Additionally, RF takes considerably longer than the other two algorithms to create a model. The chosen hyperparameters can be found in the Appendix under Table 5-7.

Experimental Setup

The purpose of the experiment is to determine the accuracies of SVM, RF and KNN on noisy speech in conjunction with filters. The filters are applied to the speech samples, which are then used to create a model for the respective machine learning algorithms. These will then be evaluated against the test set to receive the results.

Filter Selection

For the filter selection various different filters were taken into account and tested. These filters were primarily selected from the ffmpeg program, but Audacity was also taken into consideration.

For Audacity the main filter that was tested was the noise reduction effect. However, for this filter to work a sample of only background noise has to be supplied to it, which was not feasible as the samples were selected to have human voice in the majority of it. Additionally, while Audacity does have a command line interface, the noise reduction effect is not supported at the time of writing. Due to this all samples would have to be manually processed without automation, which is not viable.

Ffmpeg has multiple filters which can apply the noise reduction effect. The two primary filters of this category that were considered and tested were the afftdn and arnndn filter. The main parameter of afftdn is “nr” which takes as input a number between 0.01 and 97 which represents a decibel amount. However, applying this filter to the samples made no difference in any of the algorithms accuracy even with the highest dB. The arnndn filter is used to reduce noise from speech using recurrent neural networks. It takes as input a model m and then applies this to the sample in order try and isolate the voice. With SVM and KNN, this filter did not make a difference in the accuracy. However, for RF the accuracy decreased when this filter was applied. Hence, neither of these filters was selected to be used.

	SVM	RF	KNN
No Filters/Baseline	0.62	0.75	0.68
Highpass=200, Lowpass=3000	0.68	0.85	0.68
Highpass=300, Lowpass=5000	0.72	0.88	0.75

Table 1: 5-fold cross validation accuracy of SVM, RF and KNN for different filters

Ffmpeg also has the lowpass and highpass filters. These filters make it possible to remove certain frequencies from the samples. Since human voice frequencies are limited to a certain range, usually between 200-3000 hertz, these filters were a good choice to remove some of the background noise. For all of the algorithms the filters showed a significant increase in accuracy. Different frequency ranges were used in order to determine which range has the best accuracy for the algorithms. Table 1 shows the accuracies of the different algorithms for two different frequency ranges and for no filters. No other frequency range was comparable to the accuracy of the 300-5000 Hz range. Other frequencies were tested, but they were either worse or the same as the 200-3000 Hz range. Based on these results, the lowpass and highpass filters were chosen to be applied to the samples with the 300-5000 Hz range. Note that all previously mentioned filters were tested in conjunction with one another, but the resulting accuracies never increased.

Cross-Validation Scores

After the hyperparameters were selected, 5-fold cross validation was used on the training set in order to assess the results for the filters in conjunction with the hyperparameters. This was done to receive some insights into what the final results on the test set may look like. The accuracy and standard deviation (SD) were calculated for this and reported in Table 2. Much like in [4], the 5-fold cross validation was repeated 10 times to account for possible variance. For Random Forest a different random seed was used for each computation to ensure that the results are different for each run.

	Accuracy Score	Standard Deviation
SVM	0.72	0.24
KNN	0.72	0.15
RF	0.733	0.164

Table 2: Average 5-fold cross validation scores of SVM, KNN and RF with chosen hyperparameters

For both SVM and KNN the results did not change for any of the 10 runs meaning that every computation returned the same accuracy and SD. Even though these 10 runs returned the same results, the SD is not 0. This is because the accuracies of the 5 folds during cross validation are different, but this difference is always the same, hence the average accuracy and SD for the 10 runs is also the same. However, for RF the results were different at each run, which was expected since a different random seed was employed. From these preliminary results it can be observed that the algorithms perform similar to one another in distinguishing happy from neutral.

Results

Similarly to the cross validation scores, SVM and KNN did not see a change in results no matter how often they were executed. For RF, ten different random states were used in order to make sure the results are stable. For all of these states the results were the same. As such, the results reported in Table 3-4 for all algorithms are constant and do not change based on the number of times they are run.

	Accuracy Score	Precision	Recall	F1-Score
SVM	0.75	0.727	0.8	0.762
KNN	0.65	0.636	0.7	0.667
RF	0.65	0.714	0.5	0.588

Table 3: Accuracy, recall, precision and F1-score results for class 0(happy) of SVM, KNN and RF

	Accuracy Score	Precision	Recall	F1-Score
SVM	0.75	0.778	0.7	0.737
KNN	0.65	0.666	0.6	0.632
RF	0.65	0.615	0.8	0.696

Table 4: Accuracy, recall, precision and F1-score results for class 1(neutral) of SVM, KNN and RF

As can be seen in Table 3-4, SVM performs best from these algorithms with an accuracy score of 0.75 and F1-score of 0.762 and 0.737 for the happy and neutral emotion respectively. SVM's precision of 0.727 in table 3 indicates that when it predicts a sample to be happy, it has a 73% chance to be correct. SVM has a recall of 0.8 in table 3 which means that is successfully identified 80% of all happy samples. Alone these metrics are not particularly useful, as a perfect score in one does not indicate a good model. However, when used in conjunction like in the F1-score, these metrics give a good indication of the algorithms performance. KNN performs better than RF for class 0 and vice versa for class 1. KNN performs well when identifying happy samples with an F1-score of 0.667 and a recall of 0.7. This is not true for RF as it cannot differentiate happy samples. This can be seen in table 3 where its recall is 0.5, which means that it got half of the identified happy emotions correct. However, RF excels at identifying the neutral emotion. Table 4 shows this with its recall of 0.8.

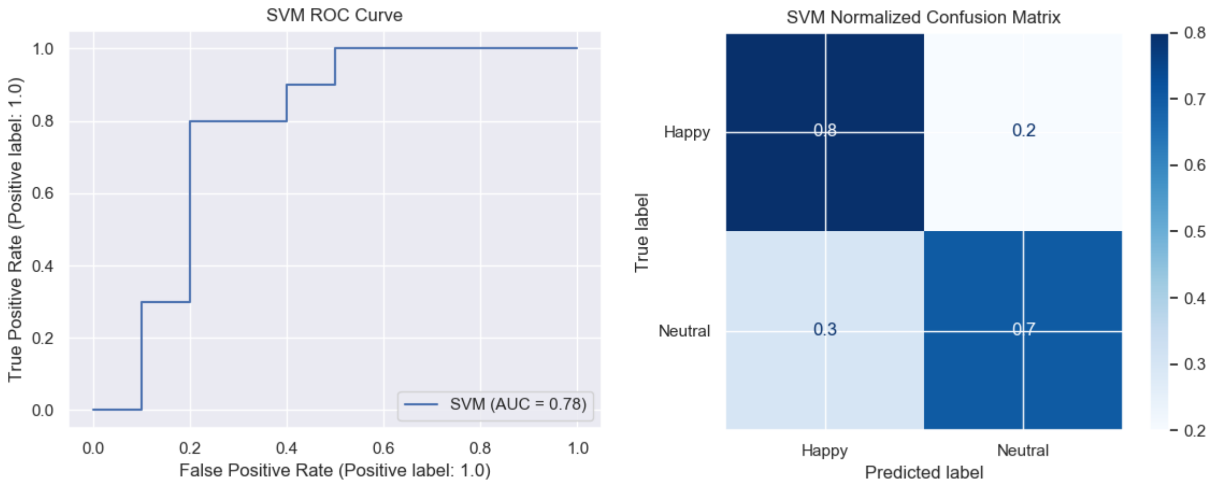

Fig 2.1: SVM ROC Curve and Normalized Confusion Matrix

SVM's ROC Curve in Fig 2.1 shows the best separability of all classifiers with an AUC at 0.78. This indicates that SVM has a 78% chance of correctly identifying the class. From SVM's confusion matrix it can be seen that it performs best when classifying the happy emotion and decently well when classifying the neutral emotion. This shows that SVM can discriminate well between the two emotions. This is not the case for RF, which according to the confusion matrix shows that it misclassified half of the happy samples.

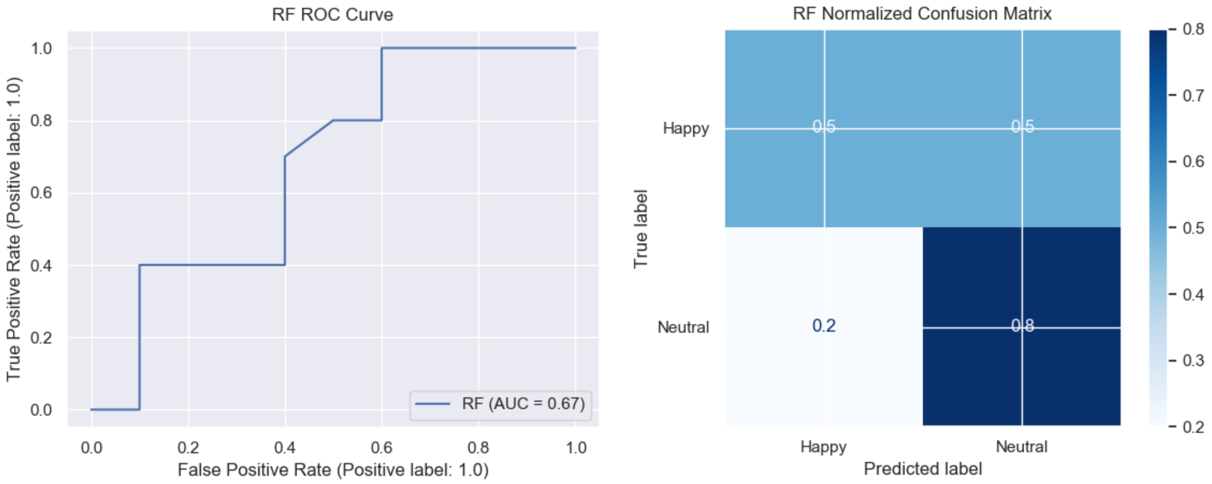

Fig 2.2: Random Forest ROC Curve and Normalized Confusion Matrix

The ROC curve of RF in Fig 2.2 shows that its separability is subpar and it is somewhat close to approaching a 45 degree diagonal, which would indicate that it cannot separate the two classes at all. An AUC of 0.67 can be interpreted as a 67% chance that the model will correctly distinguish the two classes. Based on this, RF's AUC is the worst of the three classifiers. The RF confusion matrix shows that Random Forest performs better in classifying the neutral emotion compared to both SVM and KNN. However, it is not able to accurately classify the happy emotion and is the worst algorithm in doing so.

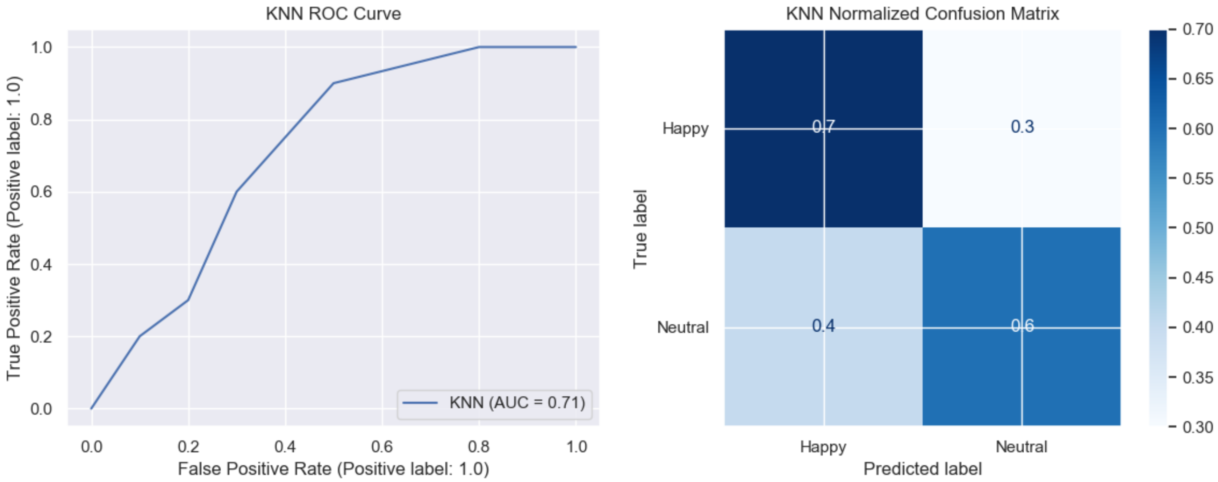

Fig 2.3: KNN ROC Curve and Normalized Confusion Matrix

KNN's ROC curve in Fig 2.3 is similar to that of RF in regards to having subpar separability. However, it performs worse than the RF ROC curve both in the beginning until 0.2 FP and in the end from 0.6 FP. KNN's confusion matrix shows that unlike SVM or RF, it does not have the best performance in any of the fields that matter. It does perform worse in categorizing the neutral emotion compared to both other algorithms.

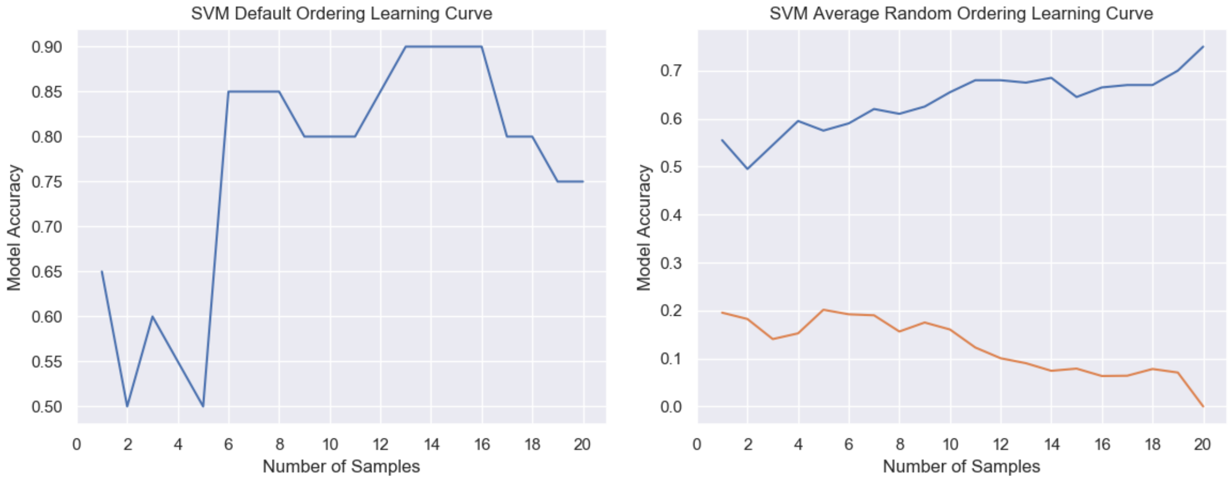

Fig 3.1: Learning Curve using default ordering and average of random ordering for SVM

Note that in the learning curves for Fig 3.1 & Fig 3.2 the number of samples correspond to 1 sample of each class. This means that 1 sample on the graph is 1 sample for the happy emotion and 1 for the neutral emotion since machine learning algorithms require at least 1 sample of each class in order to make a predicition.

SVM's default ordering learning curve shows that after six samples, the accuracy spikes to 0.85 and does not decrease significantly from this point until the very end. At the beginning of the learning curve the algorithm is having trouble to classify the samples correctly, which is to be expected as it is hard for it to learn anything from very few samples.

The random ordering learning curve shows a positive trend and a steady increase in accuracy as the number of samples increases. Additionally, the SD decreases as the number of samples increase. This decrease in SD is to be expected as the algorithm is able to learn more about the samples the more samples are provided to it. Something to note is that no matter what ordering the samples had, SVM always had the same results of 0.75 when all samples were taken into consideration. This can be seen in Fig 3.1 under the random ordering learning curve at 20 samples. The accuracy is 0.75 and the SD is 0. Based on how SVM works, this makes sense and is nothing unusual. The same is true for RF with an accuracy of 0.65 as can be seen in Fig 3.2 for 20 samples.

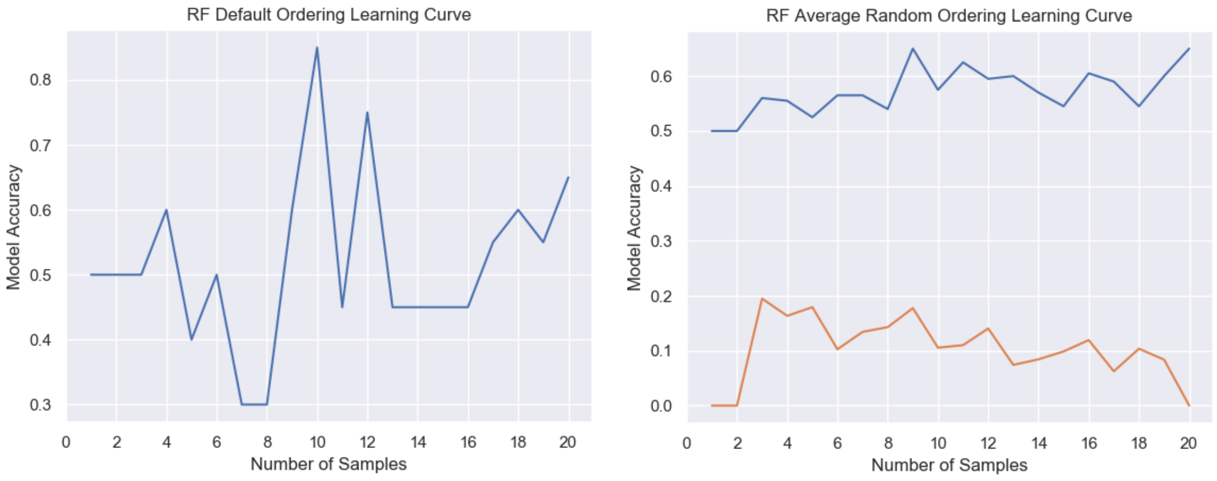

Fig 3.2: Learning Curve using default ordering and average of random ordering for RF

For RF, the default ordering learning curve in Fig 3.2 is inconsistent and quite disordely. In the beginning, it seems to wrongly classify a majority of the samples when the accuracy drops to 0.3 after which it sharply spikes to 0.85. What follows is a seemingly random decrease and increase in accuracy for the next 3-4 samples. At 16 samples, the accuracy slowly begins to increase. This may mean that the algorithm has started to learn based on the most relevant features.

The RF random ordering learning curve in Fig 3.2 shows a similar trend as the SVM random ordering learning curve showed. There is a slight positive trend in increasing accuracy as the number of samples increases. Similarly, the SD decreases as the number of samples increases. This is due to the same fact as described above, namely that with more samples the algorithm is able to learn and distinguish the classes much better.

Discussion

A number of points can be discussed based on the results reported. Firstly, from the results mentioned above, it can be seen that SVM performed better than the RF and KNN algorithms. This can be attributed to SVM's ability to generalize since "it is based on the statistical learning theory of structural risk management" [4, 13]. However, this does not mean that SVM is always the better algorithm to choose for emotion classification tasks. For example, if more or less samples were used, one of the other algorithms may perform better.

Secondly, when comparing the obtained results to other research, it can be seen that similar results were received as in [4]. In the specified study, the reported SVM accuracy was 78.75% and for KNN it was 70.84%, which is similar to this research. However, it has to be kept in mind that the study by Zhao et al. was slightly different compared to this study since it used acted databases with no background noise and the six basic emotions. Additionally, the study in [1] also found a similar SVM accuracy of 78.4% and a 3-NN accuracy of 78.4%. While the KNN accuracy is not similar, the SVM accuracy is again comparable to this research. However, Yacoub et al. used anger vs. neutral instead of happiness. In addition, they used an acted database. Even though their study used anger instead of happiness, it is still comparable to this research as anger has a high arousal just like happiness, which means that it is likely that similar results would be achieved on anger vs. neutral.

Next, even though RF's AUC is the lowest, this does not mean that RF is the worst classifier. This is because even though the AUC of SVM and KNN is better, it is possible for these to score worse in a specific region. This can be seen when comparing RF's and KNN's confusion matrix. While KNN outperforms RF when classifying the happy emotion, RF performs better when classifying the neutral emotion. This is further backed up when looking at the F1-score of KNN and RF in Table 4. This shows that RF's F1-score is significantly better than that of KNN at 0.696 and 0.632 respectively. Additionally, RF's ROC curve outperforms KNN's ROC curve in two positions, as was discussed under Fig 2.3.

From the random ordering learning curve for SVM in Fig 3.1 it can be seen that the accuracy and hence the learning rate for SVM is steadily increasing based on the number of samples. The graph shows that the learning curve did not plateau. Fig 3.2 shows the random ordering learning curve for RF and much like SVM there is a positive trend. This could mean that if more samples were used in training, both SVM's and RF's accuracy could increase.

Lastly, Table 1 shows that the lowpass and highpass filter have a greater effect with a higher frequency range. This could be because high arousal emotions like happiness may result in people being louder and having higher frequencies compared to other emotions that are less intense. As a result, if the filter cuts off anything above the 3000 Hz range, it may cut off part of the voice or emotion for some people. This is most likely what happened since the higher frequency range of 300-5000 Hz performed better.

Conclusion

This paper presents a study of emotion recognition in noisy environments using different filters and machine learning algorithms. To this end, different filters and algorithms were investigated to find an optimal set. The best filters found were the lowpass and highpass filter as presented in the ffmpeg software. These filters were used to pre-process samples collected from YouTube exhibiting happy and neutral emotions. These samples were tested with three different algorithms; SVM, RF and KNN. The best algorithm that was found was SVM with an F1-score of 0.762 for class 0 (happy) and 0.737 for class 1 (neutral). This is followed by RF with F1-scores of 0.588, 0.696 and KNN with F1-scores 0.667, 0.632 for class 0 and class 1 respectively. Keeping in mind that samples with background noise were used in this research, the results of this study are quite promising since they are comparable to other studies without background noise.

These results also refer back to the research question: To what extend do different machine learning algorithms in conjunction with filters affect the accuracy of classifying emotions in noisy speech? Table 1 shows that when applying filters, the accuracy of all the machine learning algorithms increased by almost 10%. This may be significant enough to claim that the effect of filters with machine learning algorithms increase the accuracy of classifying emotions to a certain extent, but more research is needed to have a more definitive answer.

Limitations/Future Works

Limitations

A limitation of this research was that the amount of samples available to train were most likely not sufficient to reach the best possible accuracy. This can be seen in the learning curves for SVM and RF in Fig 3.1 & Fig 3.2. The learning curve did not plateau at any point, which could mean that if more samples were available, the algorithms may have been able to categorize the emotions better.

Due to time constraints and the lack of a suitable database, the study had to be limited to two emotions instead of the six basic emotions. This meant that the results could not be analysed as in depth as previously hoped.

During grid search, the RF algorithm took up a majority of the computation time compared to SVM and KNN. In order to reduce the running time of grid search on RF, the parameter range was decreased, which has the possibility of having found a suboptimal set of hyperparameters. There may be a different set of hyperparameters that could increase the accuracy of the RF algorithm.

Future Works

Since RF was the best at categorizing the neutral emotion as was seen in Fig 2.2 and SVM was best in categorizing the happy emotion, in future research a hybrid could be made between SVM and RF where SVM categorizes the happy emotion and RF categorizes the neutral emotion. This way it may be possible to receive better results overall.

The learning curves for both SVM and RF in Fig 3.1 & Fig 3.2 showed that there was no plateau. Hence it is reasonable to assume that if more samples were added to the training set, the accuracies for the classifiers could increase, which would be a good test for future papers.

Lastly, testing a new dataset with different background noise or different emotions could give some insight whether the results obtained in this research are comparable.

References

[1] Yacoub, Sherif, et al. "Recognition of emotions in interactive voice response systems." *Eighth European conference on speech communication and technology*. 2003.

[2] Pan, Yixiong, Peipei Shen, and Liping Shen. "Speech emotion recognition using support vector machine." *International Journal of Smart Home* 6.2 (2012): 101-108.

[3] Ke, Xianxin, et al. "Speech emotion recognition based on SVM and ANN." *International Journal of Machine Learning and Computing* 8.3 (2018): 198-202.

[4] Zhao, Xiaoming, Shiqing Zhang, and Bicheng Lei. "Robust emotion recognition in noisy speech via sparse representation." *Neural Computing and Applications* 24.7 (2014): 1539-1553.

[5] Schuller, Björn, et al. "Emotion recognition in the noise applying large acoustic feature sets." *Proc. Speech Prosody 2006, Dresden*. 2006.

[6] Butt, Ammar Mohsin, Yusra Khalid Bhatti, and Fawad Hussain. "Emotional Speech Recognition Using SMILE Features and Random Forest Tree." *Proceedings of SAI Intelligent Systems Conference*. Springer, Cham, 2019.

[7] Noroozi, Fatemeh, et al. "Vocal-based emotion recognition using random forests and decision tree." *International Journal of Speech Technology* 20.2 (2017): 239-246.

[8] Hao Li et al. "Chapter 9 - 4.1.4 Random Forest (RF) Classifier." *Machine Learning for Subsurface Characterization* (2020): 243–287.

[9] Ekman, Paul. "Basic emotions." *Handbook of cognition and emotion* 98.45-60 (1999): 16.

[10] Gu, Simeng, et al. "A model for basic emotions using observations of behavior in Drosophila." *Frontiers in psychology* 10 (2019): 781.

[11] Basharirad, Babak, and Mohammadreza Moradhaseli. "Speech emotion recognition methods: A literature review." *AIP Conference Proceedings*. Vol. 1891. No. 1. AIP Publishing LLC, 2017.

[12] Banse, Rainer, and Klaus R. Scherer. "Acoustic profiles in vocal emotion expression." *Journal of personality and social psychology* 70.3 (1996): 614.

[13] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *Machine learning* 20.3 (1995): 273-297.

[14] You, Mingyu, et al. "Emotion recognition from noisy speech." *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 2006.

[15] Chenchah, Farah, and Zied Lachiri. "Speech emotion recognition in noisy environment." *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE, 2016.

[16] Sztahó, Dávid, Viktor Imre, and Klára Vicsi. "Automatic classification of emotions in spontaneous speech." Analysis of verbal and nonverbal communication and enactment. The Processing Issues (2011): 229-239.

[17] Song, Judy H., et al. "Perception of speech in noise: neural correlates." Journal of cognitive neuroscience 23.9 (2011): 2268-2279.

Appendix

	SVM
C	0.001
Kernel	Linear
Tol	1e-13

Table 5: SVM chosen hyperparameters

	KNN
Algorithm	Brute
N-neighbours	12
P	1
Weights	Uniform

Table 6: KNN chosen hyperparameters

	RF
N estimators	700
Max depth	40
Max features	None
Min samples split	5
Bootstrap	False

Table 7: RF chosen hyperparameters