**Emotion recognition in hospital patients and identification of the most appropriate machine learning?**

**A Final Report for Major Project, B.S.C Computing at The University of Bolton.**

**Lee Disley - 1802905.**

**BSC Computing Top-Up**

**Supervisor: Andrew Parker**

Contents

[Abstract 4](#_Toc8211312)

[Acknowledgements 4](#_Toc8211313)

[Introduction 4](#_Toc8211314)

[Section 1. Methodology 6](#_Toc8211315)

[Packages Used 7](#_Toc8211316)

[Further Research 7](#_Toc8211317)

[Material 7](#_Toc8211318)

[1.Biovid Dataset 7](#_Toc8211319)

[2.Cohn-Kanade Dataset (CK+) 8](#_Toc8211320)

[3.Paul Van Gent Images 8](#_Toc8211321)

[Data Preparation 8](#_Toc8211322)

[First Support Vector Machine 9](#_Toc8211323)

[Second Support Vector Machine 10](#_Toc8211324)

[Third Support Vector Machine 11](#_Toc8211325)

[Convolutional Neural Network 12](#_Toc8211326)

[Performance analysis and results 14](#_Toc8211327)

[Support Vector Machines 14](#_Toc8211328)

[Polynomial Kernel 14](#_Toc8211329)

[Fisher Face 15](#_Toc8211330)

[Random Forest Classifier 15](#_Toc8211331)

[Convoluted Neural Network 17](#_Toc8211332)

[Conclusion and future work 20](#_Toc8211333)

[References 21](#_Toc8211334)

[Appendices 23](#_Toc8211335)

[Appendix a - Logbook 24](#_Toc8211336)

[Figure 1 9](#_Toc8202418)

[Figure 2 9](#_Toc8202419)

[Figure 3 10](#_Toc8202420)

[Figure 4 11](#_Toc8202421)

[Figure 5 11](#_Toc8202422)

[Figure 6 12](#_Toc8202423)

[Figure 7 13](#_Toc8202424)

[Figure 8 14](#_Toc8202425)

[Figure 9 18](#_Toc8202426)

# Abstract

Recent research has concentrated on the development of Convoluted Neural Networks (CNN) on emotion recognition. To see if this is an improvement on older techniques using Support Vector Machines (SVM) this work aims to identify the optimum system for emotion recognition in patients, focusing on pain recognition and potential future application and research. Datasets were obtained from Kohn-Kanade, Bio-vid and Paul Van Gent to train three SVMs and a CNN. The machine learning techniques were scrutinised, the SVMs were found to perform around the 70% mark. The CNN around the 50% mark. Ultimately the CNN performed faster and was easier to adjust than the SVM, allowing for fine tuning and finding a more suitable algorithm for emotion recognition.

# Acknowledgements

I would like to acknowledge the support I received from my project supervisor who helped guide me through some difficult aspects of this report. I also would like to recognise work by Paul Van Gent and Katrina Weitz which helped me to focus and complete some of my work on Support Vector Machines and pain recognition. And finally, I would like to thank my partner and my two children without who’s love and support this report would not have been finished.

# Introduction

The original aim of this project was to look at how well a neural network can recognise human emotions from a series of images and from there to investigate the implications and applications from the research.

The research was then focused on pain identification providing a potentially usable system in healthcare to help with identifying pain rather than relying on human recognition, opening up to human error. The research is explained more indepth in the literature review in appendix a. Several datasets were analysed to suggest the viability of studying pain by itself.

After downloading the Cohn-Kanade dataset, a previously unfound report was downloaded with the dataset which highlighted a problem area for emotion recognition (Kanade, 2000). The report stated; using facial landmarks to identify emotions was not an accurate way to identify an emotion, because it only took in to consideration the movement of the face, and more is needed to identify specific emotion, including body movement, heart rate and other biological signals. As such a system to monitor and identify an emotion cannot be based purely on facial signals (Kanade, 2000).

This brought an important point to my mind, because there are not only neural networks to identify an emotion, but also Support Vector Machines which use algorithms to classify an image from its general features (Gent, 2016) and also from landmarks, depending on what kernel is used, rather than just facial landmarks (or Action Units) to be taken and fed into it like a neural network would (such as a Feed Forward neural network). Which would be better? Would a newer kind of neural network be better, such as a convoluted neural network (CNN), which is better at handling visual data, be better suited to the task? Before a system can be developed to identify a single emotion such as pain, the algorithms used must be looked at first, and then how to link biological signals from the subject to the facial movements to identify the correct emotion and its level can be decided on.

The following report is split into several sections. Section one will address the methodology I utilised for this project, identifying the packages and the materials used and their source, data preparation and the methods used in the Neural Networks and Support Vector Machines.

Section two identifies the results of running the Neural Network and Support Vector Machines. This will be followed by a conclusion summarising all relevant points identified.

Finally, the appendices hold any relevant documentation for this report, including my Literature review.

# Section 1. Methodology

To identify an emotion, a Support Vector Machine (SVM) classifier will be used, and then I will and then compare to the results from a Neural Network.

An SVM takes labelled training data (known as supervised learning) and it categorises new examples based on that training data. The SVM is used in conjunction with kernel functions to produce a multi-layer neural network . There several types of kernel that can be used with the SVM, Polynomial, Linear and Random Forest model (Ray, 2015). There are others but testing all of them, although more thorough, would be very time consuming on this time limited project. Another SVM uses the Fisher Face technique to recognise images using a non-landmark technique. Fisher face is explained in in detail in the second vector support machine section.

In simple terms a neural network acts like a human brain to solve a prediction or classification problem (Jeeva, 2018). Before neural networks became popular from 2010 onwards, SVMs performed most of the classification work that machine learning required (Jeeva, 2018). Neural networks work better on a linear basis, an SVM works on a non-linear system. When working with neural networks, a stack of multiple layers of neural networks are used to solve a problem, getting a yes/no result.

Rather than using labels to learn and classify an image, convolutional neural network(CNN) uses an images pixels to derive a difference/similarity between images. A CNN has layers like a normal neural network but has a convolutional layer in between those that use a set of learnable filters to detect features etc and produces an activation map which is passed to the next layer in the neural network (Udofia, 2018). Finally, a neural network uses an activation function to decide whether to ‘fire’ a neutron, i.e. make a decision.

This project only compares a CNN to an SVM. This decision to compare the SVMs to the CNN was made to identify and compare the old way of classifying images with the new way, ultimately identifying the most appropriate for emotion recognition in a live pain recognition system.

## Packages Used

|  |
| --- |
| Package |
| OpenFace |
| Free Video to JPG Converter |
| Python 3.7 |
| DLib |
| OpenCV |
| TensorFlow |
|  |

## Further Research

To ensure that the images I used for the neural networks were a *‘perceptually accurate representation*’ (Christopher Kanan, 2012) I researched whether using colour images versus grayscale images were best. In cases where colour is a key component of identifying the image then obviously colour images must be used, but this brings problems with the illumination of the image, having to compensate for the time of day, season and lighting angles. If colour is not needed for image recognition then it just becomes ‘noise’ of the image, unnecessary information that does not need to be processed for the task. Also finding the edges of images whilst working in grayscale is easier as the luminance can interfere and makes coding for the images more complex (Rethunk, 2012). Using grayscale images negates all these problems, and cuts processing time of an image by three or four times (Rethunk, 2012) compared to colour images.

## Material

In creating a large enough dataset for this report, three separate datasets were chosen and randomly merged. These were the Biovid, Cohn-Kanade and Paul Van Gent datasets.

### 1.Biovid Dataset

It contains data from ninety participants from three age groups compromising thirty images of men and women, in the age categories of 18-35, 36-50 and 52-65.

The dataset has five parts:

* Part A – Pain simulation (Videos)
* Part B – Pain simulation with partially occluded faces (Short windows)
* Part C – Pain simulation (Long videos)
* Part D – Posed pain and different emotions
* Part E – Emotion elicitation (Video)

Pain elicitation for this dataset was done by inducing heat in a thermode on the right arm of the participant (Walter et al, 2013). Emotion was elicitated using the International Affective Picture System (IAPS) (Walter et al, 2013). This dataset also includes four pain intensities from individual pain thresholds.

### 2.Cohn-Kanade Dataset (CK+)

Contained data from 210 participants in the age range of 18-50 years, 69% of whom were female. It was split into emotion videos and emotion labels. The participants were asked to create twenty-three facial displays for the dataset (Lucey, et al., 2010). All images were taken on a Panasonic AG-7500 camera.

### 3.Paul Van Gent Images

These were extracted from a Google search within a python script and using Zig lite Chrome extension to batch download them. It contains a small range of emotion pictures (Gent, 2016).

## Data Preparation

The data from the Biovid Dataset was initially run through Free Video to JPG converter to extract stills from the videos. The stills were then manually sorted to pull out the images that were the best fit for the neural network using work completed by Katharina Blandina Weitz as a reference point (Weitz, 2018).

The Cohn-Kanade dataset is a collection of images already sorted into folders of emotion types, a small python program developed by Gent (2016) was used to extract neutral faces and other emotion faces and sort them into respective folders.

The Gent dataset was already sorted into folders of emotion types, these were copied to the folders created by the python program during the Cohn-Kanade dataset preparation. The images from the Biovid Dataset (all emotion types) stills extracted were then added to the same folders.

Images for both the classifiers and the neural network where enhanced by cropping and highlighting just the face area and turning to grayscale. This was to ensure that all images were easy to work with and negated some of the problems that using colour pictures can create. Grayscale images simplify the algorithm and reduce computational requirements (Christopher Kanan, 2012). Grayscale images are also less sensitive to lighting conditions and show better performance when lighting is variable compared to colour images (Christopher Kanan, 2012).

The images are then split 80 – 20 using computer random selection, into training data (80) and testing data (20).

## First Support Vector Machine

The first Support Vector Machine (SVM) was a classifier with a Polynomial kernel. The polynomial kernel is used to “*turn non-linear training data into a linear equation with higher number of dimensions*” (Crowley, 2016) and is shown notated in the below image.



Figure 1

Image from: <http://www-prima.imag.fr/jlc/Courses/2015/ENSI2.SIRR/ENSI2.SIRR.S5.pdf>

A Support Vector Machine separates classes of an image in a multidimensional plane (x, z) so that the kernel (K) can perform its calculations on the image using a regularisation parameter (c) (Crowley, 2016).

This form of algorithm is known as a classification algorithm and is used to collect the landmarks and describe the data, classifying it as a certain state. The landmarks describe the moving parts of the face (the emotion shown) in the image, this feeds into the classifier for training and then testing.

The is a problem of images of different emotions being alike in morphology may confuse the classifier, even when the same emotion is being looked at, the numerical values extracted from the landmarks may be different. To help overcome this the values were normalised between 0 and 1, this was done in the program, the code for which is shown below.



Figure 2

This normalisation removes some of the variation between images that would allow them to be differentiated between. To overcome this, the position of the landmark points were calculated from a central point, a kind of centre of gravity (Gent, 2016). This creates a line between landmarks to the centre, providing a magnitude (distance between points) and direction (angle relative to image) creating a vector. The vector is then used to identify an emotion rather than just the landmarks, providing a more unique but easier to process emotion (Gent, 2016).

To compensate for rotation of faces in images where faces were tilted, I used the bridge of the nose as being generally straight and calculated the offset angle, the code for which is shown below:



After running the classifier, pickle was used to save the trained data. The classifier was then run again with the pre-trained data and more training to provide the result on the next pass.

The classifier was then run with the other two kernel functions to compare results and to see which would be best for the final neural network.

## Second Support Vector Machine

The second SVM utilised a different approach than the first, using a face recognition class known as Fisher Face. Fisher Face (also known as Linear Discriminant Analysis (LDA)) is derived from an idea by R.A Fisher in 1936, which ‘*finds the subspace representation of a set of face images*’ (Martinez, 2011). The Fisher Face class extracts features from a face that distinguish it from another and create vectors for the SVM to use. Although this is an older technique for emotion recognition, it is thought that it is more accurate and reliable than some others (Arapu, 2018).

The same pre-processed data was used for this emotion recognition as the previous SVM. So, the images used in Fisher Face were the same for a fair comparison, but also the pre-processing of image cropping and changing the image to grayscale which help to maximise the image for the Fisher Face (Arapu, 2018) had already been done.

The Fisher Face class for the SVM perform two actions on the images. First it finds the with-in class differences using a within-class scatter matrix, notated as:



Figure 3

Image from: <http://www.scholarpedia.org/article/Fisherfaces>

“Where xij is the ith sample of class j , μj is the mean of class j , and nj the number of samples in class” j (Martinez, 2011).

The Fisher Face class then finds the between class differences using:



Figure 4

Image from: <http://www.scholarpedia.org/article/Fisherfaces>

where μ represents the mean of all classes.

These are then used to find the basis vectors of the image where Sw is minimised and Sb is maximised, giving an image that looks like those in figure 1 (Martinez, 2011).



Figure 5

As most of the work on the images is performed by the Fisher Face class, the program to train and test the SVM is the same as the first one. Using the Fisher faces to compare and classify against the training and test sets.

The program to classify the images first ensured the images where correct, by performing any cropping and applying grayscale that was needed. Landmarks where then applied to the images. The images where then passed through the Fisher Face classifier where further image processing and the image classification was performed.

## Third Support Vector Machine

The third SVM used a technique known as a Random Forest classifier. Random forest is an ensemble algorithm, which means it uses more than one algorithm (of the same type or different ones). Random forest creates a set of decision trees that provide a ‘vote’ on the data. The average of these votes is then the final class of the test objects (Patel, 2017).

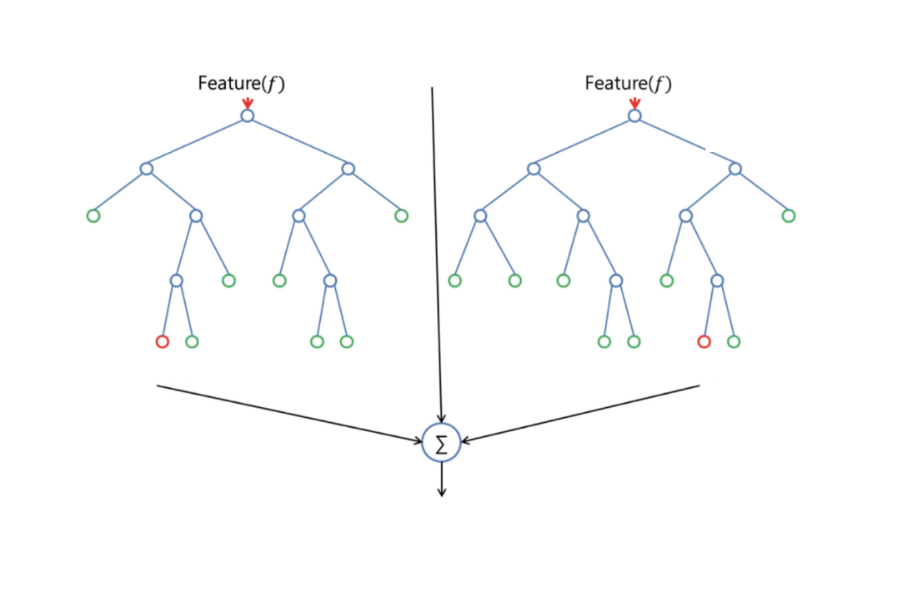


Figure 6

Figure two shows how two decision trees work to come to a mean answer. Each decision tree has nodes known as leaves, and when deciding, the random forest classifier splits the nodes to get a random threshold for each feature (Donges, 2018).

When using the Random Tree classifier there are several parameters that can be changed to improve the result, unlike with other SVM classifiers that use static algorithms. You can tell the random forest classifier how many trees to use, telling it how many processors to use, what cross validation method to use and setting to make the random forests output replicable.

I used the same program as the first two SVM and kept the images used the same. The program applied landmarks to the images as before, and the random forest classifier used these to make its decisions. I then used several different settings for the random forest classifier to see what effect if any they had on the outcome.

## Convolutional Neural Network

As I stated before, a neural network attempts to mimic how a human brain works. The aim is to provide a system that can identify or classify a problem in a more accurate and human like way. I wanted to see if a neural network was more accurate than a classifier and if it was quicker, speed being a main factor to consider when creating an application to identify pain in a patient.

A neural networks design is known as an architecture (akin to computer network design). The neural network I used was based on work completed by (Hassner, 2019). The architecture of the convolutional neural network includes twenty-two layers, these layers are split into input layer, convolutional layers, pooling layers, dropout layers, inner product layers (fully connected layers), and output layers. A graphic representation of a similar neural network is shown below (fig 3). When making the neural network, libraries from Google called TensorFlow were used to provide pre-coded neural network layers, a copy of the neural network coding is provided in appendix b.

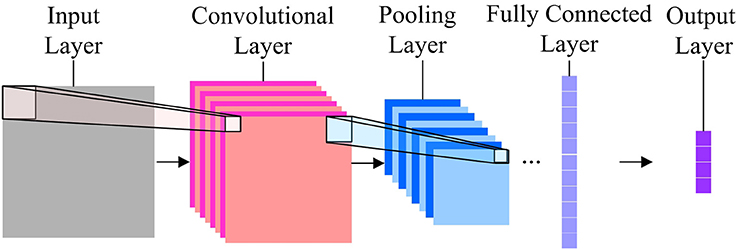


Figure 7

Image from: www.frontiersin.org

The images used where the same as the ones used for the classifiers to keep the comparison fair.

The input layer has fixed dimensions that fitted the pre-processed images used for the classifiers. The input layer then ‘pipelined’ the image into the neural network as a NumPy array.

Once passed to the convolutional layer, the NumPy array was passed through filters (the filters have randomly generated weights) and then the filters pass along each image, with shared weights, to create a feature map. The image was passed through several filters, creating several image maps.

The pooling layer was used to downsize the feature maps using maxpooling 2D layer option. This passed along the feature maps keeping maximum pixel values, the pooled values produce an image reduced by 4. This helped to keep computational requirements down and is necessary as the amount of feature maps increase.

The fully connected layer is the layer that works like the human brain. It took the feature maps and passes them through weighted neurons that are trained. The weights were trained by forward propagation of the training data and backward propagation of the errors, that calculated the level of weight adjustment. (To prevent overfitting of the neural network, dropout was performed, this is a random reduction of less than 50% of the neurons to zero. Dropout is provided for in TensorFlow by maxpooling which was used in the pooling layer.)

The output layer was where the activation function was applied to the image, this provided the probability of which emotion the image is showing. I used the SoftMax function for this layer.

# Performance analysis and results

## Support Vector Machines

### Polynomial Kernel

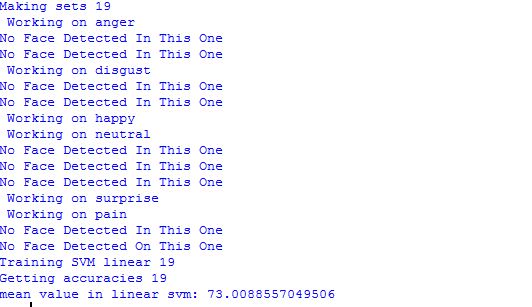


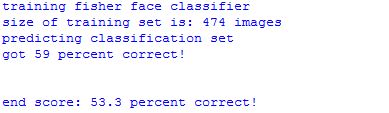
Figure 8

This result was gained from the second pass of the classifier. The classifier was set to run twenty times on each pass (0-19). The dataset numbered 480 images. This classifier achieved a result of 73% success. The classifier took 10 minutes to run each pass, so 20 minutes in total to come to the result.

To see where the classifier was going wrong I used predict\_proba() function to create a matrix of the whole dataset with a prediction against each category for an image, I created a table to show this, shown in appendix c. The terms and conditions for the datasets I used stated that I cannot show any images so used the file names as reference.

Working through the prediction data and comparing to the original image, I could see where the classifier had gone wrong. For instance, with several of the images, the classifier got the wrong emotion, but the correct emotion was its second choice. This happened especially with happiness and disgust, and contempt and pain. Looking at the corresponding images, the mistakes can be understood, more training data from a larger training set may correct this.

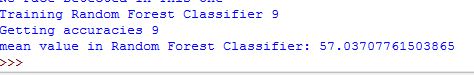
### Fisher Face



This result was also gained from the second pass of the classifier. The result is the average of 20 passes of the classifier each time. Again, the dataset was 480 images, with some of the images unable to be read by the classifier. The final mean result was 53.3% correct. The classifier took 15 minutes per pass, so 30 minutes in total.

Images that the classifier got incorrect were passed to a directory and tagged with the emotion it decided on. Many of the errors came from the fisher face classifier mis-identifying surprise as happiness and sadness as pain. Both are quite understandable, and the errors could be solved by running larger datasets

### Random Forest Classifier



This result of 57% was from the first try of the random forest classifier with the following settings:



On the next double-pass I changed the number of n\_estimators, or number of trees in the forest to 200 and had the classifier run 20 times per pass rather than the 10 for the first try (only just noticed I ran the first trial 10 times twice). This increase in n\_estimators provided a mean percentage of 87% which is much better. I ran this again, to make gain a better average over the two settings and got 79%. This gave an average score of 74.3% for the random forest classifier.

So together the classifiers provided results of:

Polynomial – 73%

Fisher Face – 53%

Random Forest – 74.3%

I compared my results for the SVMs to results achieved by Paul Van Gent (Gent, 2016):

Polynomial – 83%

Fisher Face – 82%

Random Forest – 88%

The results by Paul Van Gent where achieved from a dataset that included only seven emotions, excluding pain images, and he trimmed the dataset by removing categories of emotion that had a small number of images.

If I had trimmed the dataset similarly or found larger datasets I believe my classifiers would achieve similar results. But the range of differences between results show that classifiers are not a reliable way to detect emotion, one time they may produce an excellent guess at an emotion, another time they may not.

Also, the images used for my SVMs were produced in controlled environments with very little variation, except for the google images created by Paul Van Gent which were less standardised . Paul did not use these google images in his results noted above, when he did he achieved results of 61% mean across the three classifiers (Gent, 2016).

When SVMs are trained and used in the real world on classification problems, they can be quick, one such classifier ran at 0.1 secs per image to classify in waste management (Sakr, et al., 2016). So, although the classifiers I used ran slowly, once they are trained on bigger datasets and included in a system to identify pain I believe they would have similar performance.

## Convoluted Neural Network

The number of steps taken by the CNN was 3000, this took over 6 hours on a pc with 4GB of RAM and Intel i-5 processor 3.20 Ghz. Using the CUDA cores on a NVidia graphics card halved this to just under 3 hours running time. I used the CUDA cores setup to run the CNN for the results achieved.

The dataset used for the neural network was the same one used for the classifiers and was already pre-processed.

The results achieved by the CNN on the first run was 67% accuracy. On the second run the CNN achieved 55%.

As with the classifiers, where the neural network got the wrong result, the second choice was usually the correct one, shown in a table produced in appendix c.

A comparable CNN created by Jostine Ho which achieved 58% accuracy, shows how their neural network identified human emotions:

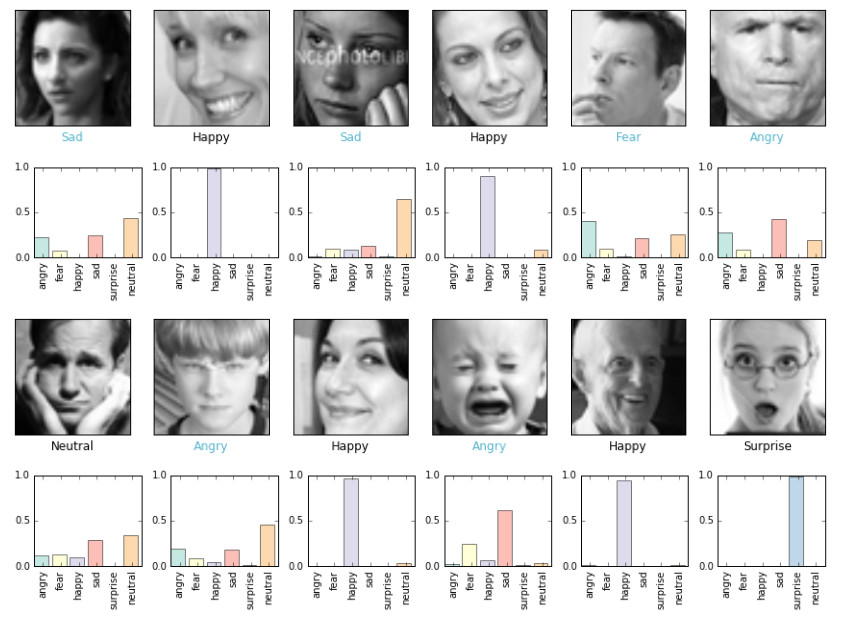


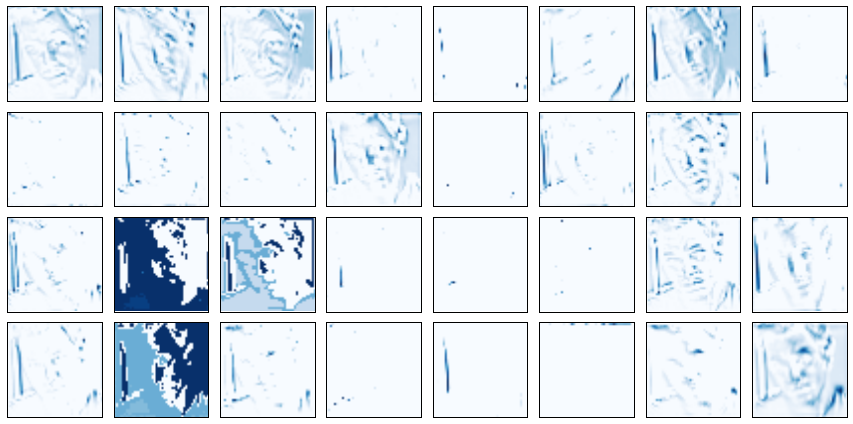
Figure 9

Image from: <https://github.com/JostineHo/mememoji/blob/master/data_visualization.ipynb>

I could not work out how to extract this information fully from my CNN, but with Jostines producing a similar result to mine, I used their images to understand where my neural network could be going wrong.

One problem that may have occurred is that the feature maps that the neural networks created could be becoming more abstract with the more pooling layers applied to the maps to combat over fitting of the data, as identified by Jostine in their GitHub article. The image below shows the problem, extracted with this code from my neural network.





Once again, I feel that if my dataset had been bigger then the emotion recognition would have been more accurate. And if I did the neural network again I would address the pooling problem by removing a pooling layer between one of the convoluted layers so reducing the feature map abstraction.

Once the neural network is trained and the data saved, the neural network can process a new image quicker than the classifiers and managed to identify an emotion within 2 minutes. With more training on bigger datasets I believe this can be shortened to a more usable time frame for live emotion recognition from a video feed.

# Conclusion and future work

In this work I used SVMs and a CNN to perform emotion recognition on images from several datasets.

An answer to the question posed as the title for the project is not an easy one to come by. Image processing by machine learning, whether it be a SVM of a CNN depends on several factors. The main factor affecting the results of either type of machine learning is the size of the dataset used for training either of the machine learning types. Once either has been trained and the training saved, then the time taken to identify an emotion is quick. The accuracy shown by either type of machine learning I believe is good, considering the difficulty of the task, but a system that is wrong 30-40% of the time is not a viable one to use on patients in a healthcare setting. Although, if I had more time to work with the neural network I believe the improved handling of images and processing of the images is better with a Convoluted Neural Network and has a much wider scope for improvement. The SVMs in the main are static algorithms with very little in the way of the capability for enhancements. A CNN allows for much better ‘tweaking’ of images and of the neural networks algorithms to find a better way.

This work has shown that emotion recognition is a difficult task for machine learning to undertake. The subtle nuances of the human face do not always portray the emotion that the person is feeling. In future work, to get a truer picture of emotion recognition, the use of other biological signals, such as heart rate viability (HRV), galvanic skin response (GSR) and electrocardiography (ECG) must be used in conjunction with emotion recognition to predict a true emotion. Much as a human does automatically. Once this is done, the recognition of pain in patients would be easier for a neural network and crucially more accurate, providing better pain management and confidence in the neural network that would allow for the monitoring of patients to become more automated. Also in future work, I would suggest more research into peoples reactions to a machine performing this sort of work, and their feelings towards any actions the system could take. Not only covering if this can be done, but also that people would be comfortable with it.

Thus, it can be concluded that the best machine learning to detect emotions is a CNN. To move this along to pain recognition in a usable system, not only would the above points need to be addressed, but also peoples feelings towards a ‘computer’ being sole observer of a patient and being allowed to decide their outcomes.

# References

aionlinecourse.com, 2018. *Kernel SVM | Machine Learning.* [Online]   
Available at: http://www.aionlinecourse.com/tutorial/machine-learning/kernel-svm  
[Accessed 3 4 19].

Arapu, M. A. a. L., 2018. *Face Recognition Using Fisherface Method.* s.l., IOP Publishing, pp. 1-9.

Christopher Kanan, G. W. C., 2012. Color-to-Grayscale: Does the Method Matter in Image Recognition?. *PLoS One,* 7(1).

Crowley, J. L., 2016. *Intelligent Systems: Reasoning and Recognition.* [Online]   
Available at: http://www-prima.imag.fr/jlc/Courses/2015/ENSI2.SIRR/ENSI2.SIRR.S5.pdf  
[Accessed 3 4 2019].

Donges, N., 2018. *The Random Forest Algorithm.* [Online]   
Available at: https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd  
[Accessed 14 04 2019].

Gent, P. V., 2016. *emotion recognition with python opencv and a face dataset.* [Online]   
Available at: http://www.paulvangent.com/2016/04/01/emotion-recognition-with-python-opencv-and-a-face-dataset/  
[Accessed 23 02 2019].

Hassner, G. L. a. T., 2019. *Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns - Demo.* [Online]   
Available at: https://github.com/GilLevi/AgeGenderDeepLearning/blob/master/EmotiW\_Demo.ipynb  
[Accessed 29 03 2019].

Ho, J., 2016. *mememoji.* [Online]   
Available at: https://github.com/JostineHo/mememoji  
[Accessed 19 04 2019].

Jeeva, M., 2018. *The Scuffle Between Two Algorithms -Neural Network vs. Support Vector Machine.* [Online]   
Available at: https://medium.com/analytics-vidhya/the-scuffle-between-two-algorithms-neural-network-vs-support-vector-machine-16abe0eb4181  
[Accessed 14 04 2019].

Kanade, T. C. J. a. T. Y., 2000. *Comprehensive database for facial expression analysis..* s.l., IEE, pp. 46-53.

Lucey, P. et al., 2010. *The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression.* [Online]   
Available at: http://pitt.edu/~jeffcohn/cvpr2010\_ck+2.pdf  
[Accessed 29 3 2019].

Martinez, A., 2011. Fisherfaces. *Scholarpedia,* 6(2), p. 4282.

Patel, S., 2017. *Chapter 5: Random Forest Classifier.* [Online]   
Available at: https://medium.com/machine-learning-101/chapter-5-random-forest-classifier-56dc7425c3e1  
[Accessed 14 04 2019].

Ray, S., 2015. *Powerful Guide to learn Random Forest.* [Online]   
Available at: https://www.analyticsvidhya.com/blog/2015/09/random-forest-algorithm-multiple-challenges/?utm\_source=blog&utm\_medium=understandingsupportvectormachinearticle  
[Accessed 8 4 2019].

Rethunk, 2012. *why we should use gray scale for image processing.* [Online]   
Available at: https://stackoverflow.com/questions/12752168/why-we-should-use-gray-scale-for-image-processing/12775620  
[Accessed 08 04 2019].

Sakr, G. E. et al., 2016. *Comparing deep learning and support vector machines for autonomous waste sorting.* [Online]   
Available at: https://ieeexplore.ieee.org/document/7777453  
[Accessed 19 4 2019].

Udofia, U., 2018. *Basic Overview of Convolutional Neural Network (CNN).* [Online]   
Available at: https://medium.com/@udemeudofia01/basic-overview-of-convolutional-neural-network-cnn-4fcc7dbb4f17  
[Accessed 14 04 2019].

Walter, S. G. S. ,. E. H. T. J. T. H. C. W. P. .. d. S. G. M., 2013. *The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system.* s.l., s.n., p. 128–131.

Weitz, K., 2018. *Applying Explainable Artificial Intelligence for Deep Learning Networks to Decode Facial Expressions of Pain and Emotions,* Bamberg: Otto-Friedrich-University.

# Appendices

Appendix a – Logbook

## Appendix a - Logbook

Major Project Logbook

Lee Disley – 180 2905

Supervisor – Andrew Parker

Date: Tuesday 23rd October 2018

First Meeting, brought my preliminary research into emotion recognition

Points for next meeting:

1) Look at huckster.io

2)List hardware needed

3) Academic challenge of project – NHS

4) Look for images of emotions

5) Look up genetic algorithms

6) Create project plan

Date: Tuesday 13th November 2018

Showed Andrew research produced from previous weeks meeting. Concentrated on the academic challenge of my project and decided to research into pain recognition. I still had not made my mind up about what equipment to use. I could not find stock images of emotions but looking for datasets to download.

Points for next meeting:

1) Get stuff ready for literature review

2) Research into other algorithms

3) Start literature review

Date: Monday 26th November 2018

Update meeting to keep Andrew appraised of my progress. Concentrating on my literature review and finding usable datasets.

Points for next meeting:

1)Complete literature review

2)Find datasets

Date: Tuesday 8th Janurary 2019

Meeting to show Andrew my literature review previously emailed to him. Reported some conflict between completing literature review and database assignment and professional issues presentation and assignment but managed to get them all in on time.

Points for next meeting:

1) Get literature review proof read

2) Hand in literature review

Date: Wednesday 30th Janurary 2019

No notes kept from meeting but did hand in paperwork that I needed filling in by the University so I could access a dataset of images of people in pain (Biovid dataset) to Andrew. Was told it would be forwarded to Louise for signing.

Date: Monday 18th Feburary 2019

Emailed Andrew my request for equipment in room C2-05 for a workstation with a NVIDIA graphics card for neural network processing using CUDA cores.

Date: 28th Feburary 2019

Emailed Louise about the paperwork for the dataset, no reply. Went in day after and saw Louise who said they would be ready next week.

Date: 18th March 2019

Received email from Andrew that Dataset forms are ready. Went into see Andrew, received my ethics forms back, no luck with the dataset forms.

Date: 19th March 2019

Met Andrew to receive Biovid forms. Sent request for dataset to dataset holder. Had a quick meeting about progress so far, but not much has been done whilst waiting for the dataset forms to be returned to me.

Started Major project write up using research that I had been doing into SVMs whilst waiting for Biovid forms. Informed Andrew that I would not have enough time to properly look at CNN, so I changed my project title to reflect the work I had already done.