Computer Vision: Facial Emotion Recognition

Orestis Charalambous

https://drive.google.com/drive/folders/15OfX6-M-v2tTWnQYEkYO18jIAq_c6UsC?usp=sharing

Abstract: In this paper we will implement and evaluate a series of image classification models using different combination of feature descriptors on the RAF-DB [1], to detect Facial Emotion Recognition (FER). To determine the best results for this dataset, we will use the feedforward Multilayer Perceptron (MLP), Support Vector Machines (SVM), and a custom Convolutional Neural Network (CNN), for experimentation with various hyperparameters with grid search and cross validation. The Scale-Invariant Function Transformations (SIFT) and the Histogram from Oriented Gradients (HOG) are two detecting methods used for the MLP and SVM classifier. Furthermore, in addition of the CNN classifier acting on the dataset, we will introduce a video and demonstrate the results live.

1. Introduction

Artificial intelligence has seen a massive development to bridge the divide between human and computer capabilities. Researchers as well as enthusiasts strive to create incredible things across various facets of the industry and Computer Vision domain is one of several such domains. Facial expression recognition is an important direction for computers to understand human emotions, and it is also an important aspect of human-computer interaction. Facial expression recognition refers to selecting the state of expression from a static photo or video sequence to determine the emotional and psychological changes of the character. In the seventies, American psychologists Ekman and Friesen used a wide range of experiments to describe six simple human expressions: happiness, anger, surprise, fear, disgust, and sadness [2]. A neutral expression has been added to the expression classification since then. Recognition of facial expression has broad opportunities of research into interactions between the human machine and emotional computing, including empathy, security, and intelligent medical treatment.

2. Data

The Real-world Affective Faces (RAF) dataset is a private database provided by Pattern Recognition and Intelligent System Laboratory (PRIS Lab) of Beijing University of Posts and Telecommunications (BUPT), that contains nearly 15,000 facial images across different ages and races, including various uncontrolled postures and lighting conditions. There are a total of 7 labelled expressions in the database including anger, disgust, fear, happiness, sadness, surprise and neutral. The video that we used with our best CNN model, consists of a 2012 Elon Musk interview that discusses his company's future, SpaceX, published on YouTube.

3. Models and Feature Descriptors

3.1. Support Vector Machines

A support vector machine is a machine learning algorithm which uses supervised learning to perform data group classification or regression [3]. To be qualified, an SVM, like other supervised learning machines, needs labelled data. SVM training materials are categorised and arranged separately in various points in space. In the context of binary classification, SVM can be expressed as the distance measure that maximises the margin between two data categories. The algorithm would aim to achieve the best data isolation possible by maximising the boundary across the hyperplane and keeping it even on both sides. If the number of features exceeds the number of samples, the SVM performs poorly. In addition, an SVM can be expanded using a single-vs-one approach or one-vs-all to perform multi-class classification. It is also notable that the SVM's greatest advantage is its robustness to high-dimensional data and its convex design that always ensures minimal global convergence.

3.2. Multilayer Perceptron

The Multilayer Perceptron, also known as the Artificial Neural Network, is a parametric classifier that uses hyper-parameters tuning during the training phase. Multilayer Perceptron algorithms are capable of handling multi-class problems by generating probabilities for each class by keeping the of model size fixed in terms of input nodes, hidden layers, and output nodes. When compared to SVMs, Multilayer Perceptron are more vulnerable to being stuck in local minima and sometimes can cause them to overfit if there are not enough training models, which SVMs do not have that problem [4].

3.3. Convolutional Neural Network

In recent years, Convolutional Neural Networks have also been widely used to extract facial expression features. It is a deep learning algorithm that can attach significance weights, and impairments, to different features and subjects of the picture and distinguish them from each other. In comparison with other classification algorithms the pre-processing requirements of the CNN are minimal. Though hand-made filters are used in rudimentary methods and have sufficient experience, such as the HOG and SIFT, CNNs are capable of learning these filters by itself.

A CNN acts different than other machine learning algorithms on images. Through the application of related filters, it can successfully capture spatial and temporal dependence in an image. A CNN can take as an input image a number of colour channels such as grayscale, and HSV. The images in our case are in RGB form so they are divided in 3 colour channels like shown in figure 1. CNN's function is to minimise the images into an easier shape, without missing essential functionality to ensure proper prediction. This is critical when designing an architecture that is good for the learning process.

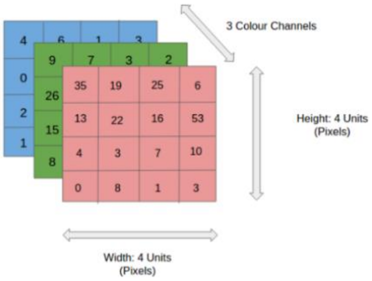
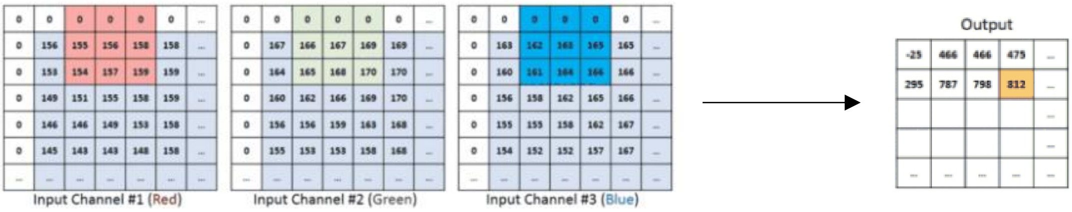


Figure 1. Image pixels divided into 3 colour channels [5].

A kernel is a part of the Convolutional layer which takes a pixel part of the image, for example a 3x3 pixel square, and computes its matrix multiplication. This is done for all the coloured channels of the image and then all findings are added with the bias to and give us as an output a squashed one-depth Convoluted Feature Output channel. The process can be seen more clearly in figure 2. The purpose of the Convolution operation is to remove from the input image the high-level characteristics, such as edges. In addition to the convolutional layer, a Pooling layer is applied that reduces the space size of the display, which further reduces the amount of calculation and weight needed. Then all the neurons in the current layer are fully connected with the activated neurons in the previous layer which is then converted from two-dimensional feature map to a one-dimensional feature map for further feature representation and classification.



0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #1 (Red)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

Output

-25	466	466	475	...
295	787	798	812	...
				...
				...
...	...	...	...	...

Figure 2. Convolutional Layer [5].

3.4. Histogram from Oriented Gradients

The Histogram of Oriented Gradients is a feature descriptor used in the pre-processing process to help classification algorithms detect objects in computer vision and image processing. The technique counts gradient orientation occurrences in located areas of an image and results in a vector containing the locally-normalised histograms as shown in figure 3.

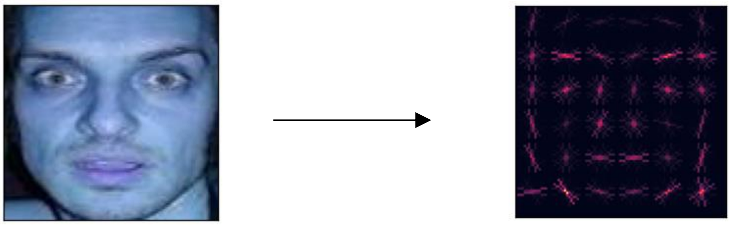


Figure 3. Applied HOG descriptor on image.

3.5. Scale-Invariant Function Transformations

SIFT is a tool for removing vectors that represent local image patches. These characteristic vectors are not only invariant in scale, but are also invariant for translation, rotation, and illumination. These descriptor descriptors are helpful for matching items. We can split a single picture with a homograph when we have certain points in each picture that we know correlate. SIFT helps to find not only the matching points in each graphic, but also easily matched points as demonstrated in figure 4 [6].

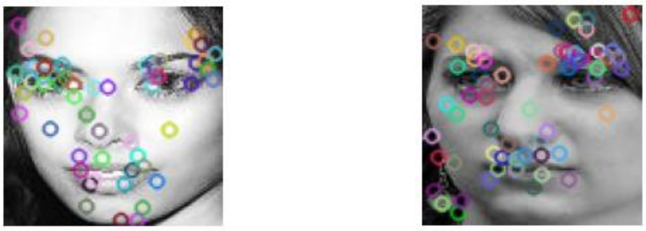


Figure 4. Applied SIFT descriptor on images to find overlaps.

For this experiment, the SIFT features were calculated by extracting a visual vocabulary using K-means clustering from the training set which grayscales the images.

4. Face Emotion Recognition

The model mentioned above will be evaluated with various function extractors and metrics in this section. We would first compare the MLP and SVM algorithms with each other on the same extractor and then compare the CNN on the RAF dataset.

4.1. Evaluation on dataset

For the evaluation of results, we use the overall sample accuracy which is the most common and straightforward metric. The effects of each classifier with its respective function descriptor are shown in the table below. Although SIFT is a great function descriptor, it is safe to say that it does not suit our models performing on this classification task.

All the models shown above where trained and experimented with different hyperparameters to derive these results. For the SVM and MLP algorithms we used a grid search with a 10 k-fold cross validation method for the findings. This resulted in an increase of +0.06 in accuracy for the SVM model and the best parameters that was found were a linear kernel and a value of 0.001 for the gamma parameter. On the other hand, the grid search on MLP performed for over 2 hours and resulted in no good hyperparameters findings. This is partly because MLP takes more parameters into account than the SVM and therefore takes exponentially more time to locate hyperparameters in each validation process. We did not realize that the grid search function could not manage these parameters from the start, and we will take this into consideration for future work.

Model	Feature Descriptor	Accuracy
SVM	SIFT	0.38
	HOG	0.64
MLP	HOG	0.61
CNN	--	0.81

In order to prevent the network from overfitting too quickly, some image transformations can be artificially done, such as flipping, rotating, cutting. For our model we introduced a randomised horizontal flip of images in the training process. We found that the CNN performed the better with the following hyperparameters: 4 batch size, 50 epochs and 0.001 learning time. We both tested Adam and SGD optimizers to find out that Adam was much better than SGD overall with 0.20 accuracy differences and the loss converge significantly faster with Adam. Also, we noticed that by increasing the batch size resulted in less overall accuracy. Figure 5 provides the test images in the trained CNN model and the probabilities of various expressions are obtained.

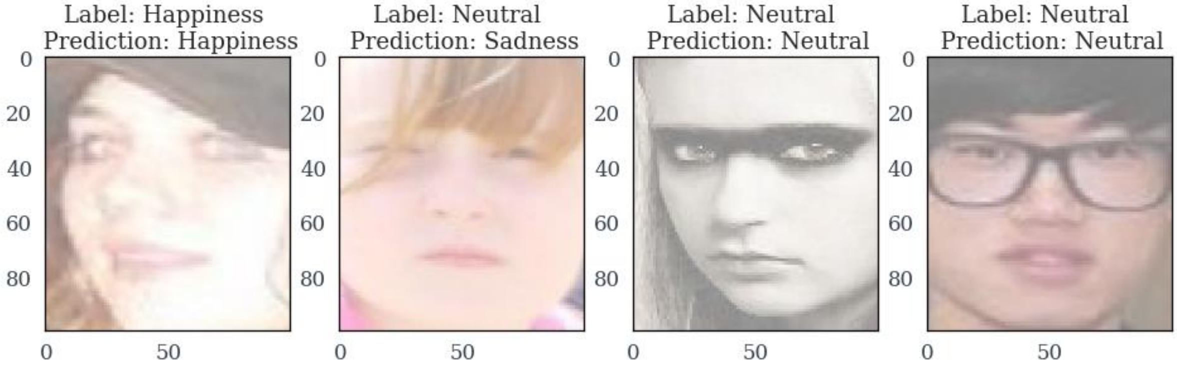


Figure 5. Prediction results using CNN.

4.2. Evaluation on video

For the video we used our best trained CNN model for the prediction of the emotions. The Facenet library developed by Google was also used to help us detect the faces in the video. Although our model performs exceptionally on the video, we had to lower the quality for the video to run faster. This resulted from problems that arose when using the face emotion recognition algorithm on the video in colab, so we had to use our local GPU. In figure 6 we can see some of the results of the video. We did not encounter any major problems while implementing the algorithm. However, changing with transformation of images during pre-processing for our CNN model resulted in better accuracy predictions in the video.

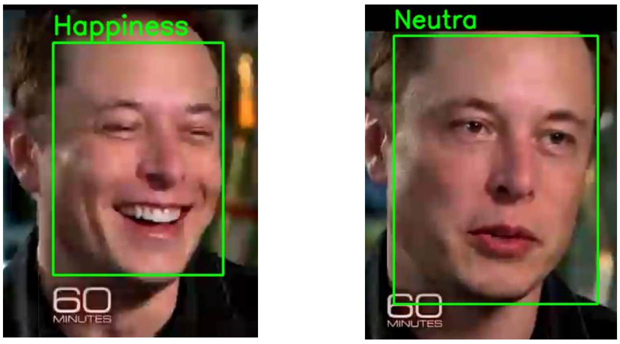


Figure 6. Prediction results using CNN on a video.

5. Conclusion

This project presents an approach to apply facial emotion recognition algorithms on a customised dataset. We observed that our feature descriptors have played a major role in our training phase and should be carefully selected for each one of our machine learning algorithms. SVM was found to be a superior image recognition algorithm than the MLP. The CNN, however, surpass all the previous algorithms.

Facial emotion recognition is evolving day by day and future work can be done to ensure better results. For example, we could introduce in our CNN model the dropout method that can effectively reduce over-fitting and improve accuracy. The dropout method is equivalent to randomly deactivating some connections during training, and supplementing these connections during testing, which is equivalent to integrating multiple good models to make comprehensive predictions. Additionally, although PyTorch is a powerful framework, we will consider using other deep learning APIs such as Keras in Tensorflow and see differences in results.

References

[1] Shan Li, Weihong Deng, and JunPing Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2852–2861, 2017.

[2] Ekman, P. & Friesen, W. V. (1971). Constants Across Cultures in the Face and Emotion. Journal of Personality and Social Psychology, 17(2) , 124-129.

[3] Scikit-learn.org. 2021. *1.4. Support Vector Machines — scikit-learn 0.24.1 documentation*. [online] Available at: <https://scikit-learn.org/stable/modules/svm.html> [Accessed 10 Apr 2021].

[4] Pico. 2021. What are the advantages of Artificial Neural Networks over Support Vector Machines? [online] Available at: <https://www.pico.net/kb/advantages-of-artificial-neural-networks-over-support-vector-machines.> [Accessed 11 Apr 2021].

[5] Medium. 2021. A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way. [online] Available at: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

[6] Medium. 2021. Implementing SIFT in Python. [online] Available at: <https://lerner98.medium.com/implementing-sift-in-python-36c619df7945>

Appendix:

Stochastic Gradient Descent	A gradient descent algorithm in which the batch size is one. In other words, SGD relies on a single example chosen uniformly at random from a dataset to calculate an estimate of the gradient at each step.
Hyperparameters	A hyperparameter is a parameter whose value is tweaked and used to control the learning process of a machine learning algorithm.
Convolutional Layer	A layer of a deep neural network in which a convolutional filter passes along an input matrix. For example, consider the following 3x3 convolutional filter:
Grid Search	It is a library function of sklearn's model selection package that helps to loop through predefined hyperparameters and fit your estimator (model) on your training set.
Adam Optimisation	An algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments.
Pooling	Pooling usually involves taking either the maximum or average value across the pooled area. For example, suppose we have the following 3x3 matrix:
Feature	A feature represents an attribute and value combination with respect to a dataset.
Overfitting	Occurs when your model learns the training data fast and efficient and incorporates details and noise specific to your dataset.
Neural Networks	A machine learning algorithm derived from the neural connection in the brain.
Classification	Predicting categorical outputs
Model	A data structure that holds information about the algorithm, parameters and results performed on a dataset.
Feedforward	Feedforward is the provision of context of what one wants to communicate prior to that communication.
Keras	A popular Python machine learning API. Keras runs on several deep learning frameworks, including TensorFlow, where it is made available as tf.keras.
API	API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other.
TensorFlow	A large-scale, distributed, machine learning platform. The term also refers to the base API layer in the TensorFlow stack, which supports general computation on dataflow graphs.

References: https://developers.google.com/machine-learning/glossary

http://www.datascienceglossary.org/