Facial Expression Recognition Model: Development Report

LaTeX template adapted from:
European Conference on Artificial Intelligence

Rares Neagu[1]

Other group members:
Abdullah Al-Qattan[2], **Mohammad A Khan**[3]

Abstract. Non-verbal communication is one of the most important means of communication. Most of the information conveyed by humans is done by non-verbal means consisting of body language and facial expression. Over time, humans evolved to accurately interpret non-verbal signals making it essential in human interaction. While it is completely natural for a person to detect and accurately predict another person's emotions, it is hard to translate and implement the means necessary for such a task to be accomplished by a computer. The difficulties come from the different facial features human use in expressing emotions and from the context clues humans are able to asses in order to further increase the accuracy of their prediction. All these challenges make the development of an algorithm based on artificial intelligence difficult. Having technologies such as Convoluted Neuronal Networks allow scientist in the field to create a model that could predict to a satisfactory degree the human expressions. The main topic of this report is the process of developing such an algorithm and how all these problems could be overcame.

1 Introduction

Creating an algorithm capable to correctly classify human emotions is difficult. The virtual infinite ways human express their emotions excludes the possibility of a *one fits all* solution. The ability to combine different movements of the facial features allows for a greater range of emotions to be transmitted, but it immensely reduce the capability of a computer, with no prior understanding or concept of emotions, to recognize these emotions. It should be taken into consideration that, even if it was possible to train a computer to detect human emotions through expression, other factors such as lighting, angle and how much of the face is obscured are important and it is necessary account for them while developing such an algorithm.

The need for classifier algorithms that are able to classify an *object*, accurately and more importantly quick, arose in multiple fields. Therefore, a more general solution was crucial. Convoluted Neuronal Networks (CNN) are one of the best methods of developing a classifier algorithm. Convoluted Neuronal Networks allow to be trained with a different dataset, making the *objects* that need to be classified almost irrelevant. A CNN is a great solution when it comes to *learning* and classifying human emotions.

2 Background

Deep learning architectures such as CNN are a breakthrough in artificial intelligence and it is relevant in different areas of expertise such as medical field, audible or visual signal analysis, automated language processing and computer vision, including facial expressions. [5]

Multiple Convoluted Neuronal Networks architectures are available and each and every one of them has its own advantages and drawbacks. Analyzing this architectures is an important part of preparation for a project like this. Making sure the architecture used is capable to create an accurate model will save a lot of time and resources down the development line.

For this application we decided it was best to use the VGG-16 architecture. VGG-16 is considered a state of the art architecture when it comes to computer vision. This architecture stands out by having convolution layers of 3x3 filter with a stride of 1 and it uses the same padding followed by a max pool layer of 2x2 filter of stride 2. This arrangement of convolution and max pool layers is consistent throughout the architecture. It ends with 2 FC (fully connected layers) using softmax for output. Having 16 weighted layers gave the name VGG-16. This network is on the larger side, having approximately 138 million parameters. Such a large number of parameters allow the VGG-16 architecture to classify objects to a larger number of classes compared to other CNN architectures, but training it is resource expensive.

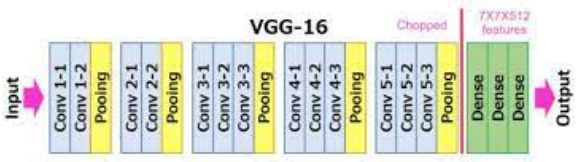

Figure 1. VGG-16 architecture

Concerning the training of our model we used a popular dataset known as FER2013. It was introduced in 2013 at the International

[1] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: rn8171n@gre.ac.uk
[2] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: aa9328h@gre.ac.uk
[3] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: mk3366t@gre.ac.uk

Conference on Machine Learning. This dataset consists of images collected from Google images servers and labeled by hand. FER2013 consists of seven classes: anger, disgust, fear, happiness, sadness, surprise and neutral. Unfortunately FER2013 contains a variable number of objects for each class. Since the images contained by this dataset is largely composed of pictures people post on internet, there is a big difference between the number of elements in each class. This dataset contains 4953 elements labeled ANGER, 547 elements labeled DISGUST, 5121 elements labeled FEAR, 8989 elements labeled HAPPINESS, 6077 elements labeled SADNESS, 4002 elements labeled SURPRISE and 6198 elements labeled NEUTRAL. After its introduction, FER2013 became the benchmark in comparing facial expression recognition model. During the years the best accuracy obtained used FER2013 was, according to our research, 75,25% [3]

Figure 2. FER2013 examples

3 Experiments and results

In order to achieve our goal, a model that is capable to detect human emotions through facial expressions, it is necessary to prepare our dataset in order for the VGG16 architecture to be able to pick on the different expression portrayed in the images. The next step would be to implement the actual model and the last stage would be to optimize the model for an increase the accuracy as much as possible.

3.1 Pre-processing

In the interest of training the model it is essential to convert the dataset into a format that the CNN can train on. In this case the pre-processing consists of reshaping the images such as their size is 48x48 pixels and converting it from a coloured format to a gray-scale one. Fortunately, the dataset used in this project is already in a gray-scale format. After resizing, it is important that we normalize the pixel values. Normalizing is done by dividing the pixel values to their maximum value, in this case 255, this will give every pixel a value between 0 and 1.

Another important step, that could admittedly be omitted, is creating different but similar copies of the images. Rotating, re-scaling and shifting the images along the X and Y axis will artificially create new images that represent the same emotion under different angles. Again, FER2013 does contain such images and this step is omitted in our implementation, but it is important to mention it.

To properly test the model it has to be presented with similar but different images. To create a well defined testing method and aiming to remove as much variance between the training set and testing set, 10% of the whole FER2013 dataset was used solely for testing. This ratio between training and testing sets combined with the great number of images found in the FER2013 dataset made us confident that the number of images used for training and testing will reflect the actual detection capability of the model. To test our line of thinking we increased the number of testing images from 10% to 20%, but no difference in accuracy was noticed.

3.2 Implementation

To achieve our goal of creating an artificial intelligence model capable of classifying human facial expressions we considered three CNN architectures: Xception, VGG-16 and VGG-19. Considering that VGG-19 is a more complex version of VGG-16, it was deemed redundant to implement two architectures as similar as these two.

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	0.790	0.945	22,910,480	126	109.42	8.06
VGG16	528	0.713	0.901	138,357,544	23	69.50	4.16
VGG19	549	0.713	0.900	143,667,240	26	84.75	4.38

Figure 3. Comparison between models

As you can see in *figure 3* [2] the *Xception* model seemed the best choice for our purpose. The small size and the rather low number of parameters, but the high accuracy makes *Xception* a model that needs to be taken into conssideration. In practice we observed that the VGG-16 architecture outperforms the Xception architecture. Our findings were backed by Ayan et al. [1], they observed that VGG-16 was a better choice for a model destined to detect pneumonia based on X-ray images. Their study finds that the VGG model has a better accuracy while the Xception model presents a better sensitivity. In another study from the medical field [4], conducted by Sivaramakrishnan et al in 2019, that aims to train a model able to detect a malaria parasite in blood smears it is revealed that VGG is a better choice than Xception. In the same study the authors concluded that ResNet-50, a model that was not considered during the conception of our project, had the best performance.

Considering our own observations, the conclusion of the studies referenced above and our knowledge in artificial intelligence, we decided to implement the VGG-16 network.

For the optimizer we considered that *Adam* was the most fitting for our model, considering is a grate middle-way and excellent starting point.

3.3 Training

Our first training cycle was composed of 10 epochs with a batch size of 32. The first results yielded by this model were an accuracy of 94.34% and a validation accuracy of 66.54%. This preliminary training cycle revealed that our model presents an overfitting problem.

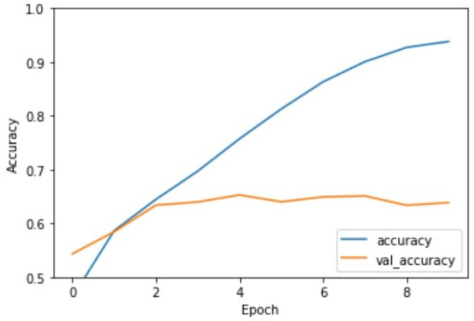

Figure 4. First results graph

Overfitting is a common pitfall in deep learning algorithms that represents a model that is learning to much details about the training data, including noise and irrelevant data in the model's training. Overfitting prevents a model from correctly categorizing new inputs, drastically decreasing the validation accuracy. This is usually a result of a model too complex for the dataset. In order to prevent this problem we considered a few methods of reducing the complexity of our model.

First method we tried was bending the VGG architecture by reducing the number of convoluted layers from 5 to just 2. Training our model for 10 epochs with a batch size of 32 in this manner resulted in a negligible. At this stage we decided to change the training parameters of the model. Training for a larger number of epochs resulted in similar or even worse results.

Another step we took towards a better fit was to remove some of the fully connected layers. This proved to be counterproductive, reducing the validation accuracy and increasing the loss. This method combined with the removal of some of the convoluted layers gave similar results.

Changing the optimizer from *Adam* to *SGD* was considered in combating overfitting in our model. In order to better compare the performance of SGD against the Adam optimizer, the model was restored to the initial VGG-16 architecture and the new optimizer was loaded with the same learning rate of 0.001.

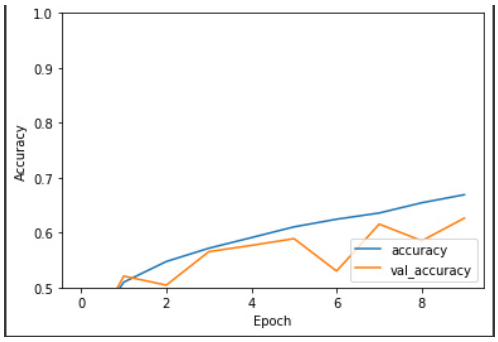

Figure 5. Results after chaining the optimizer

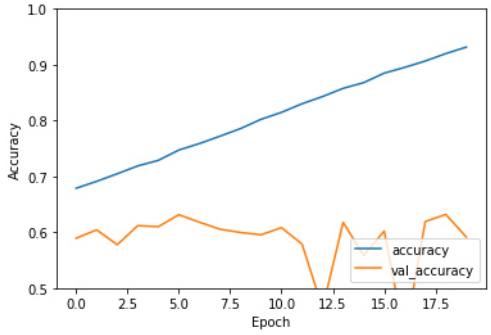

Figure 6. Results after 30 epochs with SGD optimizer

Considering that the accuracy and the validation accuracy had much closer values after a cycle of training, as you can see in *figure 5*, overfitting seemed to be less of an issue in this setup. Continuing to train the model for another two cycles, 30 epochs in total, it became apparent that not only overfitting is still present but the validation accuracy also decreased by a considerable margin, the final value being 59.13%, as can be observed in *figure 6*. After analyzing this results, we came to the conclusion that Adam is still the right choice for this project.

The last method used in combating overfitting was *early stopping*. Early stopping is a method of training that will stop the training cycle when a metric stops improving. In this case, the metric that needs improvement is the validation accuracy. For this test we used the VGG-16 architecture with the Adam optimizer, but with a callback that will stop the testing when the validation accuracy stops increasing. Unfortunately this method was not a good solution to the overfitting problem. The difference between the accuracy and validation accuracy is still substantial. As you can see in *figure 7* it could not be said that the overfitting problem is solved in anyway.

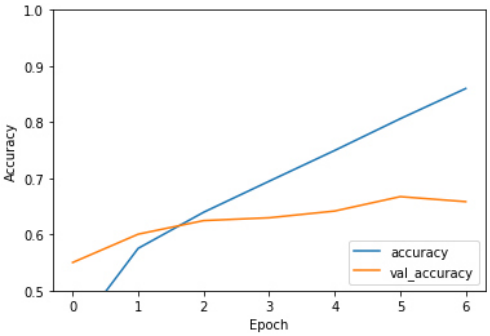

Figure 7. Results with Early Stopping

3.4 Testing

As expected, the testing results were in line with results observed in the training phase. Facing the model with new pictures from the dataset we observed that the model correctly identifies two out of three images. The model was also tested with a computer vision script based on OpenCV. During the live test the model predicted correctly an acceptable number of expressions. The hardest expression for this models are the ones with a limited number of images in the dataset. Expressions like disgust and sadness were detected much less than happiness or surprise. This could be a result of multiple factors, first being our ability to mimic expressions on demand, the facial part that is used in expressing a certain emotion, subtle eyebrows movements were easier to detect compared to subtle mouth movements for example. After the testing phase we concluded that the model did learned the facial expressions to a certain degree and it is able to correctly identify them approximately 66% of the time.

4 Discussion

An aspect worth discussing about this project is its purpose and stakeholders. An artificial intelligence algorithm capable of interpreting human expressions has a multitude of applications in the online and even offline mediums. Agencies and organizations from private and public sectors would benefit from a model like this. Online agencies and social media platforms would be able to estimate the emotional reaction an user presents when faced with an advertising or piece of content. This would allow the aforementioned agencies to better calibrate their recommendation algorithms at the expense of their users privacy. This phenomenon is mitigated by legislation, such as General Data Protection Regulation (GDPR) in the European Union. In the public sector, the capabilities to estimate a population's

general feelings to new laws and regulation. Along side other surveillance methods, expression recognition could turn into a powerful oppressive tool.

For this reasons we consider that the development of a model with as many malicious application as human expression recognition should be considered only for academical reasons. The novelty and complexity associated with a problem like makes it attractive to researchers in the artificial expression field. Its significance should be reduced to the methods and conclusions it leads during the development stages and not the final results. Conclusions that could be applied to problems with the same or even higher complexity levels, but with less malevolent functions. associated

5 Conclusion and future work

This problem is great in presenting the power Convoluted Neuronal Network posses and the impressive results that could be achieved by this type of architectures. Given that in this day and age the hardware is not a problem anymore, developing a model capable of detecting human expressions poses obstacles of theoretical nature. We studied the overfitting problem present in our project and the methods used in order to minimize it as much as possible. From changing the training methods, adding stopping callbacks and even modifying the VGG-16 architecture, all suitable methods of decreasing overfitting, proved to be insufficient. This report reflects the challenges of optimizing and fine tuning a model and the results yield by each modification or addition to the source code. In this report it was also discussed the ethical implication of developing model with this capabilities and its less beneficial application to the everyday person. Data collection by big tech corporation used in optimizing their own algorithms to state surveillance and everything in between is a real concern and could be considered a threat to our online and offline privacy.

ACKNOWLEDGEMENTS

I would like to acknowledge that the most significant hurdle we had to overcome was our own lack of knowledge and understanding of the subject. The basic understanding of the mechanisms behind a model as complex as this one led us to foreseeable results that could have been avoided, slowing down our development process.

REFERENCES

[1] Enes Ayan and Halil Murat Ünver, 'Diagnosis of pneumonia from chest x-ray images using deep learning', in *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, pp. 1–5. Ieee, (2019).

[2] Keras Api Documentation. Keras applications.

[3] Christopher Pramerdorfer and Martin Kampel, 'Facial expression recognition using convolutional neural networks: state of the art', *arXiv preprint arXiv:1612.02903*, (2016).

[4] Sivaramakrishnan Rajaraman, Sameer K Antani, Mahdieh Poostchi, Kamolrat Silamut, Md A Hossain, Richard J Maude, Stefan Jaeger, and George R Thoma, 'Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images', *PeerJ*, **6**, e4568, (2018).

[5] Boukaye Boubacar Traore, Bernard Kamsu-Foguem, and Fana Tangara, 'Deep convolution neural network for image recognition', *Ecological Informatics*, **48**, 257–268, (2018).