# Comparative Analysis of automated facial emotion recognition with different levels of blur

Alex Santonastaso
*School of Electronic Engineering and Computer Science*
*Queen Mary University of London*
London, United Kingdom
ec211269@qmul.ac.uk

***Abstract*—Automated facial emotion recognition can play a major role in our daily life as its use cases span across a variety of fields, including health care, employment, or just our daily human-machine interactions. In this paper the author discusses about previous studies of the subjects, particularly the datasets used to solve this problem, as well as the issues encountered by previous researchers. The author lingers in particular on the issue of presence of blur as noise in environments where this kind of system would ideally be deployed, thus he proceeds to design a facial emotion recognition system based on GoogLenet architecture in order to conduct experiments on the issue. The results shows that a model trained only on clear images is robust to a small amount of blur, while a model trained also on blurred data is robust to higher amounts of blur.**

***Index Terms*—facial, emotion, recognition, FER, deep learning, neural networks, database, Aff-Wild2, noise, blur, benchmarking, robustness**

## I. Introduction

Facial Emotion recognition (FER) is a computer vision problem that consists in analysing facial expressions from both images or videos in order to determine a person's emotional state. Reacting to an individual's internal emotional states, social cues, or intents, a face can alter in aspect and color, displaying an expression. Based on the methodology and data employed for the study of this subject, facial emotions can be distinguished in basic emotions such as anger, disgust, fear, joy, sadness and surprise; or compound emotions such as happily surprised, happily disgusted, happily sad, sadly angry, sadly surprised (Vemou and Horvath, 2021).

Due to its multiple possible uses in the real world, this technology is gaining every year more significance as multiple companies are investing in its development. Some examples of real world FER use cases range from healthcare, where it can be employed in detecting autism and neurodegenerative diseases, predicting psychotic disorders or depression thus finding people needing assistance; to employment, where it can be employed in monitoring moods and attention of workers or as a tool for recruiters; or public safety where it can be employed as a tool for lie detectors and smart border control; education, by detecting emotional responses of students and adjusting their learning program accordingly (Vemou and Horvath, 2021). Other uses can be helping blind and autistic people to read facial emotions, helping robots communicate with humans more intelligently, detecting distracted driving in an effort to improve driver safety.

The previous two decades have seen major advancements in the automatic identification of emotions. Known for its capacity to simulate how we learn specific types of information without really being explicitly taught to do so, deep learning has been heavily employed to address this challenge. Because facial expressions are a vast and complex field, one of the main problems to solve was to acquire a vast enough correctly labelled dataset. Early research on the topic has been done on data collected in controlled laboratory settings with frontal faces, perfect illumination and posed expressions, which does not represent real-world conditions. Instead, current research increasingly prioritizes on 'in the wild' data with different environments, backgrounds, illumination, head pose, which can be easily obtained now on the internet from video clips and images (Dhall et al., 2016).

Although most of the data employed on most previous studies of the subject are made under quiet, laboratory conditions, the target applications will sometimes be deployed in environments with cluttered backgrounds and various levels of ambient noise (Banda and Robinson, n.d.). Therefore, it is important to research the effect of noise to determine whether or not noise-reduction strategies should be taken into account when designing facial emotion recognition systems.

In this research a deep learning model will be trained on a subset of a large scale in the wild dataset to learn classifying the six basic emotions. Evaluation will be conducted on a different subset of the same database. The evaluation dataset will be artificially augmented adding blur, thus generating four new evaluation datasets with different level of blur in each. The objective is to measure how blur, as noise, affects automatic emotion detection. A second model will be trained also on some blurred images, which should improve its robustness when classifying blurred data.

The remaining sections of this research paper are the following: Literature Review, consisting of studying how previous related works were conducted; Methodology, consisting of Data Pre-processing of the datasets; Implementation, consisting of the techniques used to implement the deep learning models; Experiments and results; Conclusions and Future Works.

## II. Literature Review

In this section different previously developed FER methodologies are analysed.

In a prior study (Perveen et al., 2016) on FER using machine learning approaches, Support Vector Machine, Artificial Neural Network, and K Nearest Neighbours algorithms were evaluated. The dataset employed by the researchers comprised of images captured from a camera placed four feet away from the participants, as they artificially reproduced the six basic facial emotions. During the pre-processing stage, the pictures have been normalised, scaled to a uniform size, and furtherly, only samples representing just a single emotion having decent lightning and illumination were picked. According to the authors, the head's position in the image also affects the model efficacy, thus it should be uniform to get the best performances. Despite their outcomes were quite close, the k-NN model produced the best performances with a training accuracy rate close to 100%. Nonetheless, due to the constraints imposed by their dataset consisting of images of three people only, taken in laboratory condition, the models of this study would most likely have very poor performances if deployed in target applications.

Deep learning methods are gaining more and more rise in popularity, as we are constantly experiencing an increase in computing power and available data. In the domains of face recognition, object recognition, and several others, state-of-the-art results have recently been attained utilising neural networks. Even in the field of facial emotion recognition, results to date have been encouraging and the majority of competition winners have implemented deep neural networks (Yu and Zhang, 2015). For any deep learning approach, datasets are a key component and have a significant effect on both the training and testing of the algorithms, thus they have to be accurately labelled. Given the complexity of facial expressions, a vast amount of data is needed for a deep learning model to properly comprehend their patterns.

There is a vast variety of FER databases created to address this problem (Borreo et al., 2021). Among some of the most popular datasets in this field there is JAFFE (Japanese Female Facial Expression) which consists of 213 images labelled in seven facial expressions; Fer2013, consisting of 35,887 labelled as the six basic emotions with the addition of neutral; CK+ ( Extended Cohn Kanade), consisting of 593 pictures of 123 participants taken in laboratory settings; Raf-DB, consisting of 29,672 pictures collected from the Internet and labelled by 40 individuals into two subsets characterised by basic emotions and compound emotions. Others are the RECOLA database, consisting of 345,000 pictures of 46 subjects made under a laboratory-controlled environment, labelled by 6 annotators (Kollias et al., 2019), and the AFEW (Acted Facial Expression in The Wild) dataset, consisting of 113,355 pictures of more than 300 subjects. Nevertheless, most of the existing datasets are still limited. Their main limitations are caused by a variety of factors such as small number of subjects, for example JAFFE is limited only to 60 subjects which are only Japanese females, as well as having a small amount of total samples(Ko, 2018). Another limitation is caused from strong presence of class imbalance, as seen on the Fer2013 dataset, which although it can be considered to be a dataset of decent size, it is composed of a total of 8,989 'Happiness' samples, while there is only a total of 547 'Disgust' samples (Borreo et al., 2021). These issues led to the ideation and creation of the Aff-Wild database, one of the first large scale FER datasets, with the characteristics of containing over 1 million samples gathered from YouTube videos by querying the keyword 'reaction'. It represents 200 subjects of different ethnicities in different head poses and illumination conditions (Kollias et al., 2019). However, due to the author's current hardware limitations, it is currently not viable to base this research on such a huge dataset, thus only a subset of such a big dataset in the wild will be used, as explained in further details in next sections.

Pranav E. and others (E. et al., 2020) conducted a research on facial emotion recognition using deep learning. The authors used a dataset comprised of 2,550 total samples, grouped into five emotions categories. Using python, they implemented a CNN model architecture with the use of keras library, running on top of TensorFlow. Their developed architecture is composed of two convolution layers: each convolution layer output passes through a ReLU activation function and then is fed into a pooling layer which reduces the input size without losing information. After each convolution layer there is a dropout layer used to reduce overfitting. The dataset has been split to 80/10/10 for training, validation, testing. Lastly their model has been trained for a total of 11 epochs with learning rate of 0.01, Adam as optimizer function, Categorical cross-entropy as loss function. The trained model has been then evaluated on the testing set using accuracy, precision, sensitivity, specificity, F1-score, resulting in an accuracy of 78%.

In another study (Banda and Robinson, n.d.) the authors extended the research of facial emotion recognition by investigating the effect of noise on a FER system. In their study a subset of their used database has been augmented by adding white Gaussian noise with variances going from 0 to 0.005. Their results show that the emotion recognition capacity of their system decreases as the noise increases, as its accuracy ranges from 95.25% accuracy on samples without augmented noise, to 39.6% on samples with the maximum level of noise. Hence, this research will be based on the hypothesis that while modern FER systems are robust to some noise, there will be a cut-off point at which detecting different emotions will automatically drop off. Thus, the objective is to measure how blur, as noise, affects FER.

## III. Methodology

Despite the fact that the concept of facial expressions is complex and varied between cultures, studies have revealed that most of the face expressions used to indicate emotions are common among people from many backgrounds and countries. In his book 'The Expression of the Emotions in Man and

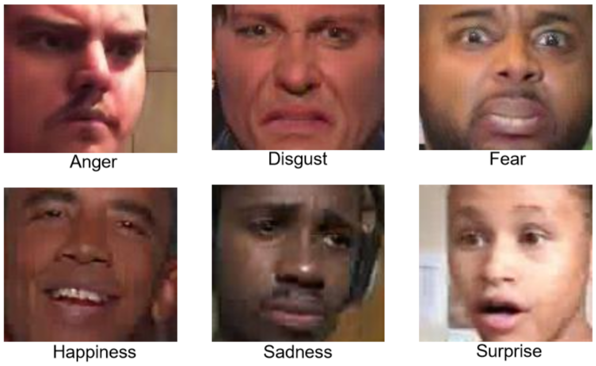
Animals', Darwin asserted that the best way to learn about someone's feelings is by reading their facial expression, and although gestures vary across cultures, facial expressions of emotions are universally manifested in the same way (Ekman, 2009).

Getting a sufficiently large and accurately labelled database to feed the deep learning model for its training is one challenge of this project. For this reason, the dataset chosen for this research is Aff-Wild2, the second version of Aff-Wild which increases its size to a total of 2.5 million pictures containing more than 450 individuals (Kollias and Zafeiriou, 2018). This dataset is ideal for this subject as it contains data representing a decent variety of subjects in different environments with different backgrounds, illumination, head pose. Nevertheless, due to the author's current hardware limitations, this study will be limited to two subsets of said dataset. While a smaller dataset also can accelerate the training process, its limited dimensions may prevent the model from functioning more accurately in an ideal target application. Using the full dataset would ideally result in better performances, which could be furtherly improved by merging different datasets thus increasing variety. Even so, the objective of this research is not to develop a state-of-the-art FER model, but to perform experiments on the effect of blur on its capacities.

### A. *Data Preparation*

The two subsets of Aff-Wild2 used for this study contain different subjects: the largest will be split in training/validation/testing for the deep learning model, the second one will be used as deployment data to perform experiments on data coming from a possible target application; their contents are inspected after importing them in a Jupyter Notebook instance resulting to be 128x128 resolution images in jpg format.

Fig. 1. An example of data in Aff-Wild2 dataset



The plots in Fig. 2 and Fig. 3 show the contents of the datasets and reveal their class imbalance. Because of the ground truths available in this dataset, the study will be limited on single emotion class prediction.

Fig. 2. Class distribution of the original dataset used for training the models of this study, before pre-processing

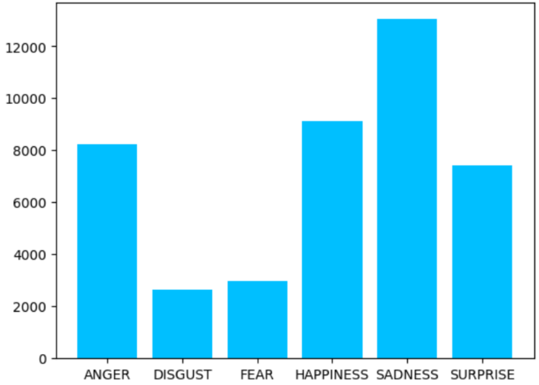
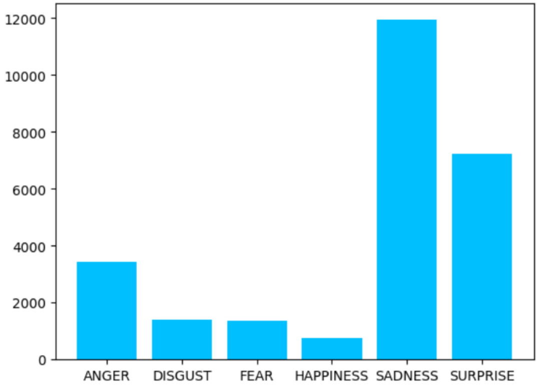


Fig. 3. Class distribution of the original dataset used for evaluating the models of this study, before pre-processing



The contents of the datasets are loaded in the Jupyter Notebook as a numpy array list containing the images, and a list containing the labels of the samples, obtained from the names of the folders containing them. The labels are consequently encoded using the LabelEncoder method from scikit-learn python library, which encodes them with values between zero and five.

Before proceeding with any other pre-processing step, both subsets have been examined for duplicates samples. Using the library hashlib in python, it is possible to calculate a sample's hash using different algorithms such as sha256, sha384, sha512, md5, etc. Each sample's hash has been calculated and compared with each other, then duplicates found have been removed. A total of 196 duplicate samples have been removed from the larger dataset, while only 11 duplicates have been removed from the second one.

### B. *Data Pre-processing*

The majority of FER systems include a face detection component and a face alignment component, which are important for the system's effectiveness as it has been proved that they improve the system's accuracy to a small degree (Pise et al., 2022).

There are multiple face detection python libraries available online. Among these MTCNN is highly reliable and accurate. Although it operates a bit slowly than other face detectors, even with the presence of differences in size, lightning conditions or severe rotations, it correctly recognises faces (Esler, 2019).

Another popular face detection system is DeepFace, developed by a research group at Facebook. It is a lightweight face detection and attribute analysis framework for python, known for having a higher accuracy than humans in such tasks (Serengil and Ozpinar, 2020) (Serengil and Ozpinar, 2021). It also includes a face alignment component that aligns an image based on left and right eye coordinates, which is the one used in this study to align both the images, as shown in Fig. 4 and Fig. 5

Fig. 4. A random image contained in the training dataset

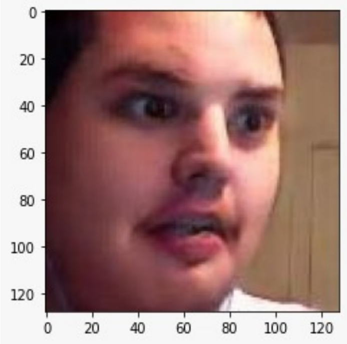
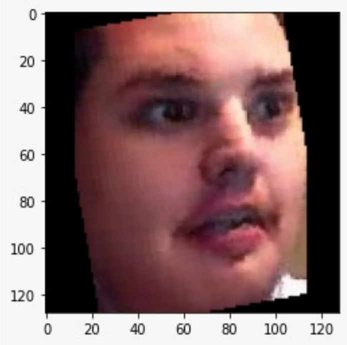


Fig. 5. A random image contained in the training dataset after having been aligned by DeepFace's (Serengil and Ozpinar, 2020) face alignment component



After face detection and face alignment the larger dataset has been augmented to balance the class distribution, which could lead to a biased model. Minority classes have been augmented with horizontal flipping and by slightly increasing brightness, in such a way that their number of samples matches the number of samples belonging to the majority class. The data has also been normalised, which makes all its values lie between zero and one.

The augmented dataset, which now has a total of 78,240 images divided in six balanced classes, is then split into training/validation/testing sets with a 70/20/10 ratio, which will be used to train and evaluate our model.

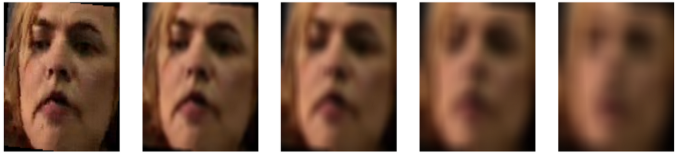
### *C. Augmenting deployment dataset with different levels of blur*

The second dataset also presents class imbalance. As this will only be used for experiments and not for the training of the model, it doesn't have to be a large scale dataset thus it has been balanced by downsampling each class to 750 samples, which is close to the amount of samples contained in the minority class. The resulting 4,500 images have then been augmented with blur using the BoxBlur method of the ImageFilter module from the PIL python imaging library, which blurs an image by changing every pixel to the mean value of the pixels in a square box which extends in each direction to a specified radius value. A total of four augmented datasets has been generated, containing images with low (radius 2), medium (radius 4), high (radius 8), very high (radius 14) amount of blur. Fig. 6 depicts a random sample of the dataset and its four blur levels augmented versions. The first image on the left hand side represents the original sample, while the first on the right hand side represents the same sample augmented with very high level of blur.

| | Baseline Model | Second Model |
|---|---|---|
| Training | 54,768 | 67,368 |
| Validation | 15,648 | 19,248 |
| Testing | 7,824 | 9,624 |
| **Total** | 78,240 | 96,240 |

TABLE I
70/20/10 TRAINING/VALIDATION/TESTING SPLIT OF DATASETS USED TO TRAIN THE MODELS OF THIS STUDY

Fig. 6. The five different levels of blur of this study: none on the left hand side, very high on the right hand side



### *D. Generating a second training dataset containing blurred images*

After the dataset that will be used for training/testing/validation of the model and the subsets for experiments containing different subjects are prepared, another training dataset is required so that a second model can be trained on data containing augmented noise as blur. This second model will also be experimented on the deployment data, and it is expected to be more robust when classifying blurred data.

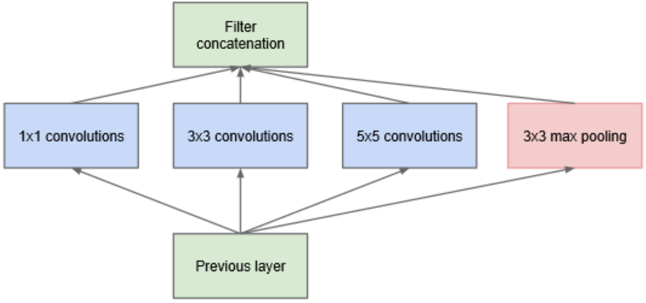
Because of the constraints due to the author's available RAM and GPU, the dataset has been augmented with only 1,000 low, 1,000 medium, 1,000 high blur samples per emotion, which were picked up randomly from the previously preprocessed data. The result is a dataset containing a total of 78,240 clear samples, with the addition of 18,000 samples augmented with blur, resulting in 96,240 samples. Again, this second dataset is split into training/validation/testing with a 70/20/10 ratio.

# IV. Implementation

The models developed for this study will be based on the deep convolutional neural network architecture named Inception ideated by Google researchers for image classification in the ILSVRC14 challenge. The model proposed by the researchers based on this architecture, which also won the competition, was named GoogLeNet (Szegedy et al., 2014). Their model, trained on ImageNet dataset, classifies images in 1,000 categories and notably outperformed state of the art models at the time.

GoogLeNet's innovativity in computer vision was mainly due to the introduction of the Inception module (Fig. 7) which allows to run multiple operations such as convolution and pooling with different filter sizes simultaneously in one single layer, in an efficient way. Known the fact that the more layers a model has, the higher the chance of overfitting, as it becomes easier for the model to memorize the training set; the researchers conceived this architecture with the idea of being wider instead of deeper. Thanks to this idea, although its total number of layers is over 100, the architecture results to be only 22 layers deep (27 if also counting pooling).

Fig. 7. Naive version of Inception module (Szegedy et al., 2014)



GoogLeNet's implementation for this study was made possible using the python Torchvision package, which is part of Pytorch, and comprises a variety of models architectures ideated for classification. In an effort to try reducing possible issues due to the small size of the datasets used for training, a pre-trained model has been used. The classifier has been changed so that it classifies into the six categories of this study, instead of the default 1000.

Using the same architecture two models have been trained for this study. The baseline model has been implemented using the original class balanced dataset, the other one has been trained on the second balanced dataset augmented with the different levels of blur of the study. Both models have been trained for 15 epochs with a batch size of 32 and a learning rate of 0.0001, using CrossEntropyLoss and Adam optimizer. The training process took around five hours using the original balanced dataset, and seven hours with the dataset augmented with blur. Figures 10 and 11 shows the training and loss curves of the two models (baseline model on the left hand side). Both models have been initially evaluated on their testing set using accuracy and F1-Score, as shown in the Tab. II.

Fig. 8. Graph representing GoogLeNet architecture including all its building blocks (Szegedy et al., 2014)

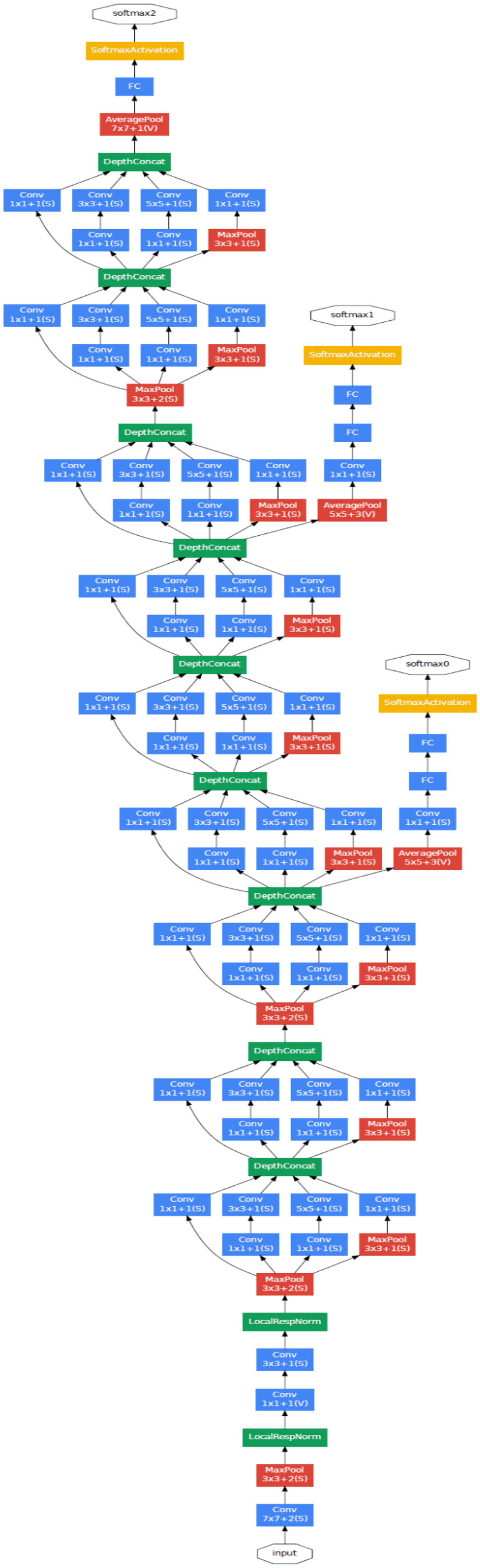
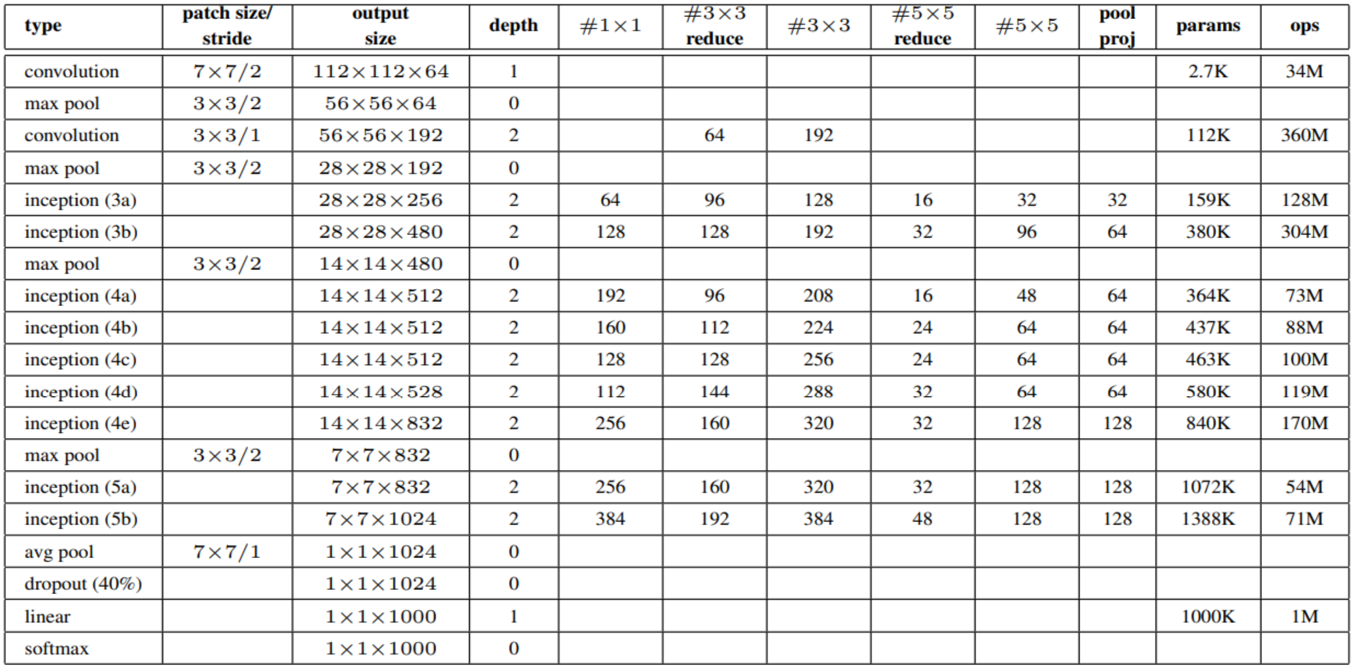
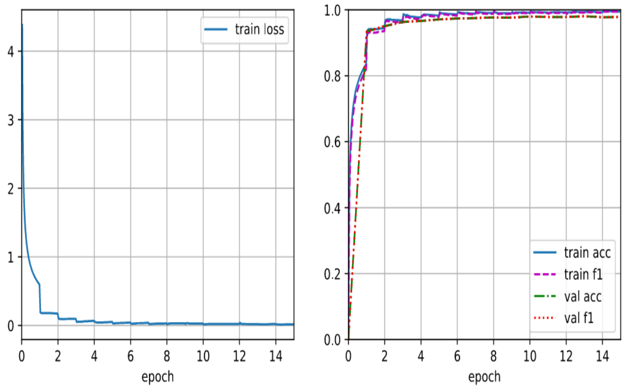


Fig. 9. Description of all the 22 layers of the GoogLeNet architecture (Szegedy et al., 2014)



| type | patch size/ stride | output size | depth | #1×1 | #3×3 reduce | #3×3 | #5×5 reduce | #5×5 | pool proj | params | ops |
|---|---|---|---|---|---|---|---|---|---|---|---|
| convolution | 7×7/2 | 112×112×64 | 1 | | | | | | | 2.7K | 34M |
| max pool | 3×3/2 | 56×56×64 | 0 | | | | | | | | |
| convolution | 3×3/1 | 56×56×192 | 2 | | 64 | 192 | | | | 112K | 360M |
| max pool | 3×3/2 | 28×28×192 | 0 | | | | | | | | |
| inception (3a) | | 28×28×256 | 2 | 64 | 96 | 128 | 16 | 32 | 32 | 159K | 128M |
| inception (3b) | | 28×28×480 | 2 | 128 | 128 | 192 | 32 | 96 | 64 | 380K | 304M |
| max pool | 3×3/2 | 14×14×480 | 0 | | | | | | | | |
| inception (4a) | | 14×14×512 | 2 | 192 | 96 | 208 | 16 | 48 | 64 | 364K | 73M |
| inception (4b) | | 14×14×512 | 2 | 160 | 112 | 224 | 24 | 64 | 64 | 437K | 88M |
| inception (4c) | | 14×14×512 | 2 | 128 | 128 | 256 | 24 | 64 | 64 | 463K | 100M |
| inception (4d) | | 14×14×528 | 2 | 112 | 144 | 288 | 32 | 64 | 64 | 580K | 119M |
| inception (4e) | | 14×14×832 | 2 | 256 | 160 | 320 | 32 | 128 | 128 | 840K | 170M |
| max pool | 3×3/2 | 7×7×832 | 0 | | | | | | | | |
| inception (5a) | | 7×7×832 | 2 | 256 | 160 | 320 | 32 | 128 | 128 | 1072K | 54M |
| inception (5b) | | 7×7×1024 | 2 | 384 | 192 | 384 | 48 | 128 | 128 | 1388K | 71M |
| avg pool | 7×7/1 | 1×1×1024 | 0 | | | | | | | | |
| dropout (40%) | | 1×1×1024 | 0 | | | | | | | | |
| linear | | 1×1×1000 | 1 | | | | | | | 1000K | 1M |
| softmax | | 1×1×1000 | 0 | | | | | | | | |

Fig. 10. Training evaluation of baseline model trained only on clear images



| | Accuracy | F1-Score |
|---|---|---|
| Baseline Model | **0.983** | **0.983** |
| Second Model | 0.98 | 0.98 |

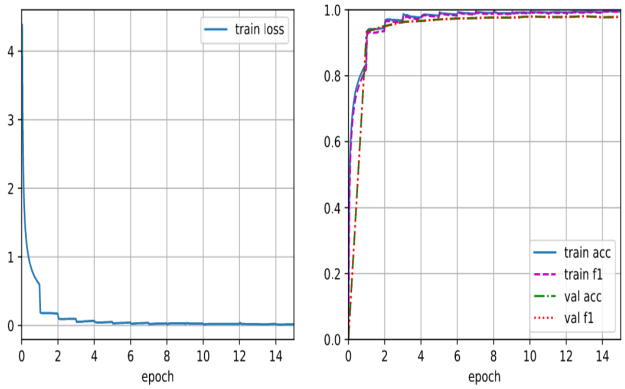
TABLE II
EVALUATION OF THE MODELS ON THEIR TESTING SETS

Both models have similar performances on their validation and test sets. Baseline model is only 0.3% more accurate than the second one trained also on some blurred images.

## V. Results

In this section the two models trained in previous section are experimented with the five pre-processed subsets of the second dataset obtained as defined in subsection C of Methodology section, including a dataset without augmented blur and four datasets containing different levels of blur each.

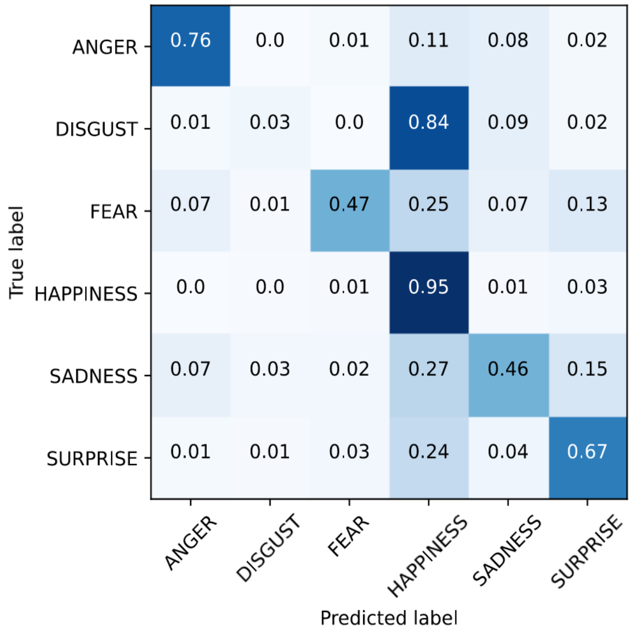
Fig. 11. Training evaluation of second model trained also on some blurred data



As specified before, the subset of Aff-Wild2 used for this section contains different subjects than the ones contained in the dataset used for training, thus a decline in performances is expected compared to the evaluation conducted on testing sets.

The baseline model achieved an accuracy of 0.56 and a F1-Score of 0.53 on the non blurred experimental data. Figure 12 represents its confusion matrix. Although the model performed decently on fear and sadness classifications; good on surprise and anger; very good on happiness; its performances on disgust samples classification are very poor, with only 3% of

Fig. 12. Evaluation of the baseline model on deployment non blurred data using confusion matrix



disgust samples classified correctly. This issue is probably due to the original class imbalance of the training dataset, since its disgust class is counting the lowest amount of samples among all classes. Regardless of that this model is good enough for this research, consisting in experiments on the effect of blur in a FER system, not in developing a state of the art model.

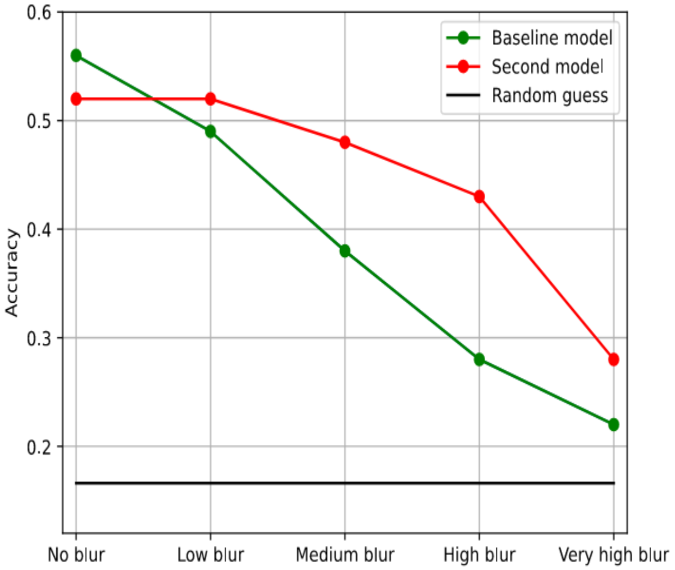
From Table III it can be observed that as the level of blur increases, the baseline model's performances decrease. Starting with an accuracy of 56% on non blurred samples, it gradually decreases until reaching 22% on samples with very high level of blur, which is just around 5% more accurate then what it would have been if just random guessing among the 6 classes.

TABLE III
EVALUATION OF THE MODELS ON DEPLOYMENT SETS CONTAINING DIFFERENT LEVELS OF BLUR

| Blur level | Baseline Model | | Second Model | |
|---|---|---|---|---|
| | Accuracy | F1-Score | Accuracy | F1-Score |
| None | **0.56** | **0.53** | 0.52 | 0.50 |
| Low | 0.49 | 0.45 | **0.52** | **0.48** |
| Medium | 0.38 | 0.31 | **0.48** | **0.45** |
| High | 0.28 | 0.22 | **0.43** | **0.40** |
| Very high | 0.22 | 0.14 | **0.28** | **0.22** |

The same tests are conducted on the second model trained with some blurred data, and can be observed from Table III. Despite the fact of being 4% less accurate than the baseline model on non blurred data, its expectations on higher robustness on blurred data are confirmed. In fact, it resulted to be 3% more accurate on low blur level samples, 10% more on medium blur level samples, 15% more on high blur level samples, 6% more on very high blur level samples.

Fig. 13. Accuracy of both models on different level of blur



## VI. Conclusion and Future Works

This research's objective was to measure how blur, as noise, affects a modern FER system. As expected from the hypothesis, a FER system is robust to a small amount of blur, but there is a cut-off point at which detecting different emotions will automatically drop off.

The results showed that as the level of blur increase, the FER system performances decrease, and that a model trained also on some blurred samples proves to be more robust when classifying blurred images. With that in mind, future developers of FER systems shall take in consideration including samples with some blur for the training of their model.

Future projects on FER systems shall take in consideration the option of adding a component that detects blur. Consequently, techniques to deblur an image such as Wiener filter, or just reduce the level of blur, shall be experimented with in an effort to measure how robust a model is on artificially de-blurred data.

## Acknowledgment

I would like to express my deepest gratitude to my supervisor Dr. Helen L. Bear, for providing guidance and feedback throughout this project.

Thanks to my parents Amelia and Domenico for making it possible for me to study in one of the best universities in London. Thanks also to my brothers Giampiero, Riccardo and Marco who always supported me during my studies.

# REFERENCES

Banda, N. and Robinson, P. (n.d.), 'Noise analysis in audio-visual emotion recognition'.

Borreo, R., Naga, P. and Marri, S. D. (2021), 'Facial emotion recognition methods, datasets and technologies: A literature survey'.

Dhall, A., Goecke, R., Gedeon, T. and Sebe, N. (2016), 'Emotion recognition in the wild - journal on multimodal user interfaces'.

E., P., Kamal, S., C., S. C. and M.H., S. (2020), 'Facial emotion recognition using deep convolutional neural network'.

Ekman, P. (2009), 'Darwin's contributions to our understanding of emotional expressions'.

Esler, T. (2019), 'facenet-pytorch: Pretrained pytorch face detection (mtcnn) and facial recognition (inceptionresnet) models'.
**URL:** *https://github.com/timesler/facenet-pytorch*

Ko, B. C. (2018), 'A brief review of facial emotion recognition based on visual information'.

Kollias, D., Tzirakis, P., Nicolaou, M. A., Papaioannou, A., Zhao, G., Schuller, B., Kotsia, I. and Zafeiriou, S. (2019), 'Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond - international journal of computer vision'.

Kollias, D. and Zafeiriou, S. (2018), 'Aff-wild2: Extending the aff-wild database for affect recognition'.

Perveen, N., Ahmad, N., Khan, M. A. Q. B., Khalid, R. and Qadri, S. (2016), 'Facial expression recognition through machine learning - ijstr'.

Pise, A. A., Alqahtani, M. A., Verma, P., K, P., Karras, D. A., S, P. and Halifa, A. (2022), 'Methods for facial expression recognition with applications in challenging situations'.

Serengil, S. I. and Ozpinar, A. (2020), Lightface: A hybrid deep face recognition framework, *in* '2020 Innovations in Intelligent Systems and Applications Conference (ASYU)', IEEE, pp. 23–27.

Serengil, S. I. and Ozpinar, A. (2021), Hyperextended lightface: A facial attribute analysis framework, *in* '2021 International Conference on Engineering and Emerging Technologies (ICEET)', IEEE, pp. 1–4.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2014), 'Going deeper with convolutions'.

Vemou, K. and Horvath, A. (2021), 'Techdispatch 1/2021 - facial emotion recognition'.

Yu, Z. and Zhang, C. (2015), 'Image based static facial expression recognition with multiple deep network learning: Proceedings of the 2015 acm on international conference on multimodal interaction'.

# MSc Project - Reflective Essay

| | |
|---|---|
| **Project Title:** | Comparative Analysis of automated facial emotion recognition with different levels of blur |
| **Student Name:** | **Alex Santonastaso** |
| **Student Number:** | **210991068** |
| **Supervisor Name:** | Helen L. Bear |
| **Programme of Study:** | Big Data Science with Machine Learning Systems MSc |

In this study I have developed a FER system and investigated on its behaviour when presented to data artificially augmented with blur. The baseline model's performances have then been compared to a second model based on the same architecture, with the difference of including some blurred images in the dataset used for training.

The code has been written and tested on my personal Windows desktop machine in Python, using Anaconda Distribution, which allowed me to easily setup different environments in the initial phase and run different versions of packages. It includes Jupyter notebook, which similarly to Google Colab, allowed me to arrange the code and its outputs in a step-by-step manner.

## Approach

In order to perform experiments on the effect of blur on facial emotion recognition, the first step is to develop a working FER model, which I evaluated using a different subset of Aff-Wild2 representing frames containing different subjects, which was artificially augmented generating a total of 5 evaluation datasets containing different levels of blur, ranging from none to very high. After observing that the model's performance decreases as the blur increases, I trained a second model on data containing the clean images and blurred images, which proved to perform better on blurred data compared to the baseline model.

## Analysis of strength and weaknesses

### Strengths

- The developed models successfully worked in measuring how blur affects FER performances.
- The dataset used was not made under laboratory settings, thus accurately represents data used in an ideal target application.
- Class imbalance is addressed with data augmentation.
- Like most recent FER systems, the methodology of this study includes face detection and alignment.
- Using a pre-trained model resulted in decent performances despite the fact of having used a small dataset for training.
- Models have been evaluated with different evaluation metrics such as accuracy, F1-Score, confusion matrix.

Weaknesses

- Given the complexity of the facial emotion subject, the dimensions of the dataset used for this study is not sufficient to accurately extract patterns from facial emotions, resulting in not having the best performances when deploying the model.
- Balancing classes using data augmentation caused poor performances when classifying minority classes. Using an originally balanced dataset would have led to better performances.
- Although MTCNN face detector and DeepFace facial alignment are robust in performing these tasks even with the presence of different conditions, their algorithms are slow performing and it took 7-8 hours on average to perform such operations.
- The second model's training data does not include samples containing very high level of blur, due to RAM limitations. Even though it managed to classify such images more accurately than the baseline model.
- Both models have been trained for 15 epochs only, which took 5 to 7 hours on average.

Limitations and practical challenges

Due to the dimensions of the dataset, I could not use the resources provided by the University, as I've found myself unable to upload the data on the university's Jupyter Hub instance. Furthermore, multiple operations such as face alignment and training of the models took multiple hours, so a constant uptime was needed. This study has been conducted on my personal desktop machine which has only 16GB of RAM and a GTX 1060 6GB as a GPU, thus its methodology has been limited by the available computational power.

Even though the size of each image sample is 3-4kb, their size hugely increases when they are converted to numpy arrays during the pre-processing steps, forcing me to use only a small subset of the entire dataset which barely fitted my 16GB available RAM, especially after data augmentation. This issue has been addressed by changing the data type of numpy arrays containing images to 'uint8' during the pre-processing steps, being able to store values from 0 to 255, which is exactly what we need for RGB images. After normalising the images to a range from 0 to 1, the pre-processed data has been saved as float16 numpy arrays, which drastically reduced size in memory compared to the default data type. The 6GBs of VRAM available from my GPU have been enough to allow me training of the models, as data was loaded in small batches, hence there has been no need for the entire training and validation sets to be loaded in the GPU RAM at the same time.

Another limitation was due to the strong class imbalance present in the training dataset which had to be balanced using data augmentation. Although showing decent performances in classifying the emotion categories of this study, the models performed very poorly in classifying images in the disgust category, which was the minority class in the original subset used for training.

Available computational resources caused a limitation in time, as the face alignment process using MTCNN took more than 7 hours on average, while the training process of baseline and second model took 5 and 7 hours on average, forcing me to train both models only for 15 epochs.

Without these limitations I could have added more data to the study, as well as train many different models with different hyper-parameters and for a higher amount of epochs. Limitations in data quantity have been addressed by importing a pre-trained model from PyTorch library, which resulted in decent deployment performances after training on FER data. Nevertheless, the trained model has been proved to be good enough for the objective of this study, which was not to develop a state of the art FER model, but to measure the impact of blur on its performances.

## Awareness of Legal, Social Ethical Issues

The ethical concerns related to facial emotion recognition include racial bias, racial discrimination, privacy, lack of informed consent and transparency. The tools used for this study have been used ethically, as the model only read and learn from images converted to numbers, so they have not been used to determine things such as someone's race, skin color, religion, gender, disability, as the categories of the study only refer to the six basic human emotions.

Privacy is not an issue of this study, as the data used for this study is anonymised. Furtherly, the data comes from Youtube videos thus it has been previously uploaded under copyright licenses that allow for free reuse. As the deployment has been performed on the same type of data, no further data from real world has been collected, thus there is no opportunity in this study to collect data which can be then misused for other purposes. The dataset used for this study is not openly shared to the public, hence it can't be misused for other purposes without the authors' authorisation.

## Possibilities for future projects

I am very happy with the results of this project as it made me more confident in my machine learning and python skills, as well as make me learn about some of the many issues in the FER subject which I was not aware of. It was particularly motivating to work on this topic as I learned more about it and started imagining about many of its possible use cases in the real world, coming to the realisation that it will be a daily part of our lives soon.

The achieved results showed us that a FER system trained only on clean data has a rapid performance decline on blurred data, while a system trained also on some blurred data is much more robust than that. To overcome this limitation caused by blurred data, future developers of FER system should take in consideration adding noised data to the training dataset to make the system more robust in these circumstances. Another option to address this issue is to perform experiments on the effect of blur detection and then artificial deblurring of data on a FER system performance. Assuming this option furtherly improves the robustness of FER systems on blurred data, future FER developers should add these components in their architecture.