Facial Emotion Recognition Using Deep Learning

Csaba Bokányi (FUWQK1), Tamás Mészégető (H6LGMQ)

***Abstract* – In this project, we attempted to create, train and test a facial emotion recognition (FER) solution using deep learning. We use the public FER2013 dataset, and build models described in the literature using Keras. Based on the reviwed literature, we create models of three different architectures with huge differences in the size of the parameter space. The models are trained using different hyperparameters and optimization approaches, resulting in a total 9 trained models achieving human-like 61-67% accuracy. As this is a mainstream solution in state-of-the-art systems, we go on to create different decision-level ensembles to improve the overall accuracy of our solution. A final accuracy of 69% is achieved.**

Impact Statement – This study reconstructs deep learning-based FER solutions described in the literature and combines them into ensembles. We test a weighting approach which considers the confidence in the different decisions of the different models, but the best results are provided by simple accuracy-weighted averaging, just above 69% on the official test sample of the FER2013 dataset.

***Index Terms*—Deep Learning, FER, Machine Vision**

I. Introduction

Deep learning has conquered numerous areas of data processing, and is playing a key role in the image processing field. One of the main reasons, among others, is the high level of abstraction used by humans when understanding and analyzing scenes, which is difficult to model with hard-coding approaches. That is why we choose an image processing task for our project, which, despite being a traditional classification task, is challenging for humans and machines alike. Faces representing any kind of sentiment have many common features, which may be the reason for the relatively low average human accuracy of 65%. On the other hand, this performance leaves room for algorithmic solutions not only to replace human activity, but also to surpass it.

II. Literature review

A. FER overview

The survey from Li and Deng [1] summarizes the latest developments in the field: with the appearance of sufficiently large and diverse datasets, such as the FER2013 [2] used in this project, it became possible to train deep neural networks for emotion recognition tasks, surpassing the accuracy of earlier solutions. While dynamic approaches consider multiple subsequent images of the subject thus may utilize this additional information, static solutions solve the problem using a single frame. The possible additional variation in the data and such the needed greater data quantity and model complexity are considerable challenges in the case of the dynamic approach and lead us to the decision to deal with the static case.

In the intensively researched field of static Facial Emotion Recognition we orientated with the help of surveys ([1],[3]) and also reviewed standalone studies that consider the FER2013 dataset ([4], [5], [6], [7]) and several other works ([8], [9], [10], [11]) to capture the diversity of approaches in the field.

The referenced surveys summarize the mainstream workflow of a FER solution. The pre-processing approaches mainly try to compensate for the lack of sufficiently large datasets by either eliminating certain unwanted variations or by generating additional training samples. Face alignment using feature detectors or convolutional neural networks, and illumination and pose normalization are among the most widely used approaches to eliminate certain variances. Data augmentation efforts on the other hand, strive to solve the same problem by generating more samples. In this case the usual approaches used in many image classification solutions are applicable, such as the addition of certain types of noise, horizontal flipping and random cropping, modifications to the saturation, contrast or other image parameters depending on the representation and other image transforms which preserve the semantical contents of the image.

In the terms of network design, a great diversity is observable. The main approaches used in image processing tasks, such as convolutional neural networks (CNN), Deep Belief Networks (DBN), Deep Autoencoders, Recurrent Neural Networks (RNN) and General Adversarial Networks (GAN) have all been used to solve the emotion recognition task The work of Pamerdorfer et al. [3] mentions, that many neural networks that were developed for such purposes are significantly shallower than in related fields, still realizing near state-of-the art results. However, common deep convolutional architectures, such as VGG, Inception or ResNet, among others, have been utilized successfully for the emotion recognition tasks, thanks to the evolution of databases and pre-processing solutions. The authors also mention that deep networks provide the possibility of further improvements over earlier, shallower solutions, given that the databases also improve in the future.

Although going deeper is one of the main developments in the field, some studies show that models containing almost 2 magnitudes fewer parameters can also deliver good results [4] . Such solutions may prove robust as fewer parameters limit the dangers of overfitting and thus force a higher level of abstraction.

Fernandez et al. [8] build a network in their recent work with an integrated attention net. This approach is another way to eliminate variance by focusing only on the face and allows accuracy levels over 80% on different FER datasets.

Against various approaches to create additional feature vectors as inputs, the work of Alizadeh and Fazel [5] shows that well designed networks are capable of learning the sufficient features and such efforts are not necessarily needed.

In the terms of training approaches, diversity prevails: there are numerous end-to-end design and training solutions, for example [4],[9]. Still, to counter the data quantity and quality issues, human face related pretrained networks (such as face detectors or FER models created for a wider variety of emotions [11] are often utilized as a base for FER solutions. Alternatively, pre-trained networks are not only complemented, but the whole network is trained, and such the pre-training functions as an initialization approach. Meanwhile, Boughrara et al. [6] uses an on-the-fly hyperparameter optimization: while training, if certain predefined accuracy levels are not reached, the number of neurons is changed (increased) in the hidden layers.

B. Neural Network ensembles

Beyond pre-processing, architecture design, initialization, hyperparameter optimization and training strategies, there remains on important step to improve the overall performance of FER solutions: the interpretation of the output of the network.

Although many FER solutions are not particularly deep, and there are many approaches to understand the inner functioning of artificial neural networks, deep learning is still a bit of a black box. There is no guaranteed reaction to any new data, and all the above-mentioned steps (pre-process, design, initialization, optimization) are going to influence this reaction. This sometimes results in huge output-variance: small changes to the input result in huge changes in the output, potentially resulting in false classification.

Network ensembles counter this sensitivity and uncertainty by utilizing multiple networks and a decision-making step based on the multiple outputs. It is important to note, that there are also feature-level ensembles, but in FER ensembles are more widely used in the decision-making step.

To create an ensemble, one needs to gather different models: different architectures, data and data augmentation techniques, initialization and optimization strategies all yield different solutions. Provided that all models yield the output in the same format, there are multiple decision-making strategies: majority voting and averaging are simple solutions, that assign the same weight to all networks. However, one may want to consider the confidence in each solution. Many approaches are used in the literature, such as the composition of weighted averages based on accuracy or loss, or the exponentially-weighted decision function and hierarchical committees described by Kim et al. 10.

Using ensembles, significant improvements have been reached on many datasets, such as in the case of Pramerdorfer and Kampel [3], which motivates the further integration of newer and better solutions.

III. CONCEPT OF THE PROJECT

When building and training neural networks, the ultimate goal is robustness: the net should be able to handle new data, so that it can be used in real-world applications, along with hard-coded software solutions. This is what motivated our team to experient with small ensembles, using slightly modified models described in the literature. By reconstructing working models described by other scholars, we have a good basis to experiment with different ensemble architectures.

IV. MODELS AND TRAINING

As mentioned, we were using the FER213 dataset and thus were aiming for models that work well with the dataset but have a variety of architectures. The dataset consists of 35887 images depicting 7 different emotions: anger, disgust, fear, happiness, sadness, surprise and there are also face classified as neutral. the dataset is by default split so that 1/10 of all pictures belongs to the validation, and an equal part to the test partition, thus only 80% of the images is used for training purposes. Further data augmentation was applied, performing horizontal flips, and random rotations and shifts.

First we considered the models of Alizadeh [5]. The first, deeper architecture has the following structure:

[Conv-(BN)-ReLU-(Dropout)-(Max-pool)]M -v[Affine-(BN)-ReLU-(Dropout)]N - Affine - Softmax.

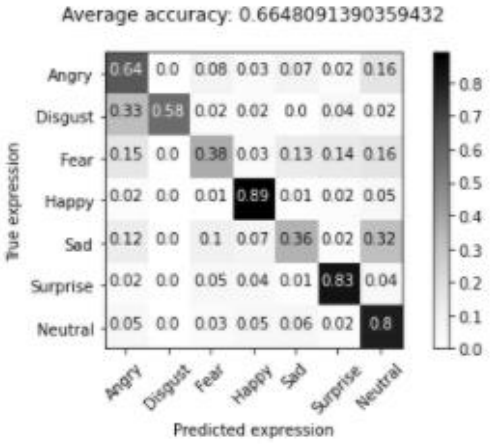
The first part of the network refers to M convolutional layers that can possess spatial batch normalization (BN), dropout, and max-pooling in addition to the convolution layer and ReLU nonlinearity, which always exists in these layers. After M convolution layers, the network is led to N fully connected layers that always have Affine operation and ReLU nonlinearity, and can include batch normalization (BN) and dropout. Finally, the network is followed by the affine layer that computes the scores and Categorical cross-entropy loss function. The developed model gives the user the freedom to decide about the number of convolutional and fully connected layers, and contains batch normalization, dropout and max-pooling layers. Along with dropout and batch normalization techniques, we included L2 regularization in our implementation. This model has roughly 4.5 million trainable parameters.

We also reconstructed the shallower network with only 2 convolutional and one fully connected layers. In the first convolutional layer, we had 64 3x3 filters, with the stride of size 1, along with batch normalization, dropout, and maxpooling with a filter size 2x2. In the second convolutional layer, we had 128 3x3 filters, with the stride of size 1, along with batch

normalization and dropout and also max-pooling with a filter size 2x2. In the FC layer, we had a hidden layer with 512 neurons and categorical cross-entropy as the loss function. Also in all the layers, we used ReLU as the activation function. This model consists of more than 9.5 million trainable parameters, most of which are located in the fully connected layer.
To observe the effect of adding convolutional layers and FC layers to the network, we trained a deeper CNN with 4 convolutional layers and two FC layers. The first convolutional layer had 64 3x3 filters, the second one had 128 5x5 filters, the third one had 512 3x3 filters and the last one had 512 3x3 filters. In all the convolutional layers, we have a stride of size 1, batch normalization, dropout, max-pooling and ReLU as the activation function. The hidden layer in the first FC layers had 256 neurons and the second FC layer had 512 neurons. In both FC layers, same as in the convolutional layers, we used batch normalization, dropout and ReLU. Also we used Categorical cross-entropy as our loss function.
We implemented these networks using Keras. We used Adam optimizer, early stopping and reduced the learing rate if the training was stuck. Training was done tuning the learning rate, weight decay and dropout parameters to find the optimal hyperparameters. For each of these parameters, we investigated a range of 2 magnitudes (base value, 1/10*base value and 10*base value). We rached the best outputs by applying the following parameters:

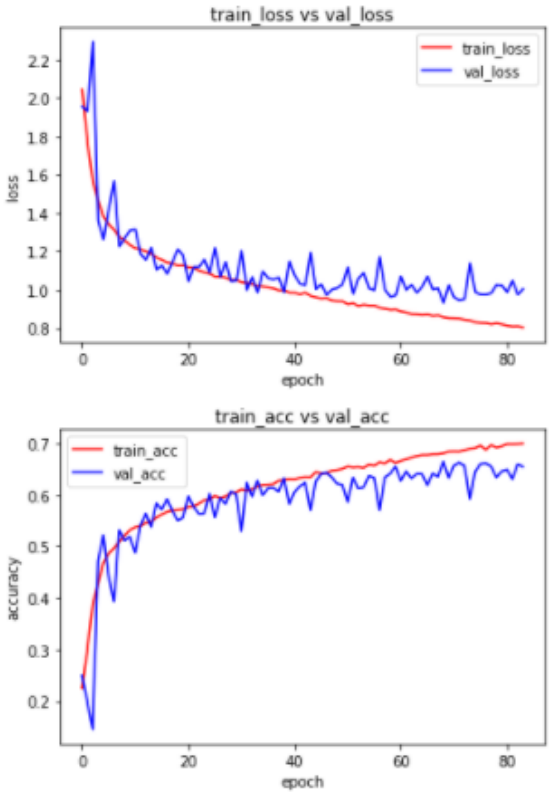
learning rate: 0.01
weight decay: 1e-7
dropout: 0.3

On the following figures one of the tyipical learning curves and the obtained confusion matrices are depicted (Figures 1. and 2.)



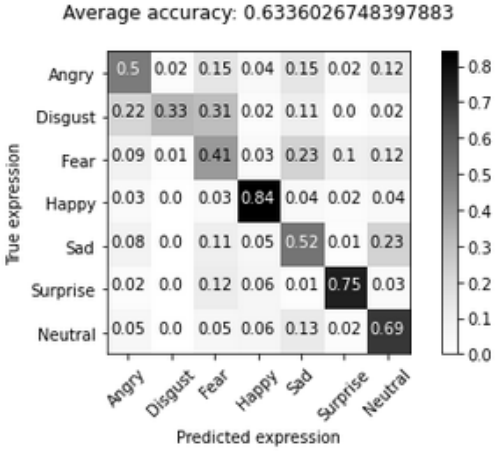
1. Figure - Confusion Matrix of the deeper network, using the best found hyperparameters

The curves run as expected, a separation between the validation and training accuracies is observable around 80 epochs, which was the typical number of epochs needed to train these networks.



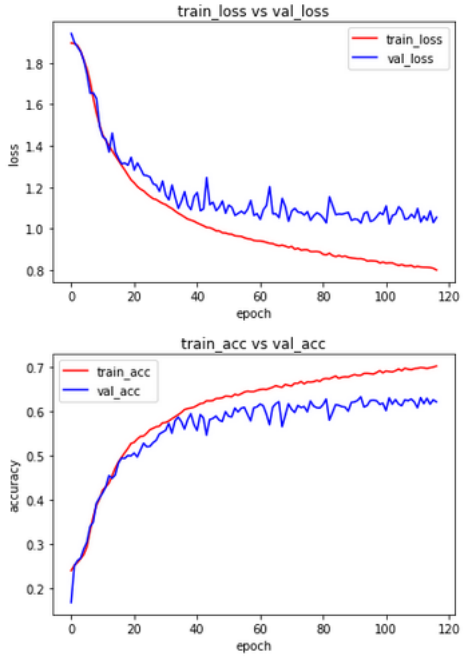
2. Figure - Losses and accuracies training the deeper network, using the optimal hyperparameters

The deeper model yielded around 66%, the shallower 60% accuracy, which is close to the typical human level of 65%. We continued with the implementation of a rather different approach, originating from Agrawal et al. [4].
This network is deep, but has a quite unique structure. There are no fully connected layers, nor do they use drop-outs. As the authors describe, there is a constituent layer, consisting of two convolutional layers, batch normalization and ReLu activation. Their work inspects many hyperparameters, and concludes that the use of a constant filter size of 8 yields the best hyperparameters, and provide two models consisting of 8 and 7 constituent layers, respectively.



3. Figure - confusion matrix of the fully concolutional model

We choose the latter one to articulate the differences even more: this model has only roughly 464 000 trainable parameters. That's a magnitude less than the other two models had, which, as we hoped, will yield a different behaviour that could be beneficial when combining these networks. These models yielded accuracies between the former two, around 63%. Figures 3 and 4 depict the confusion matrix and the obtained learning curves.

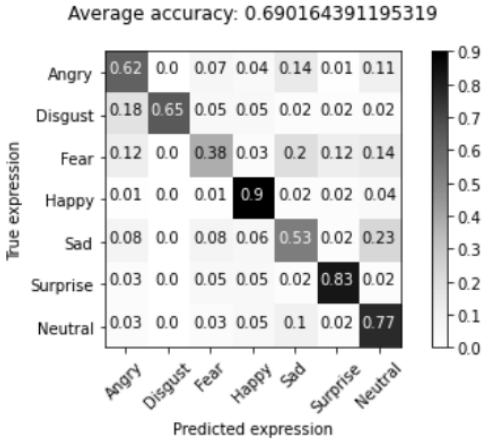


4. Figure - teching the fully convolutional net

After the teaching and optimization procedures, we obtained a total of 9 models, of which 8 yielded accuracy levels over 55% and the best could achieve 66%.

V. BUILDING ENSEMBLES

As described in the literature, state of the art FER approaches – and solutions in other fields too – often build on the predictions of multiple networks. However, these often cosists of a few hundred models such as in the case of [10]. With our ensembles we ought to test, whether a small number of models which are still considerably diverse are able to yield better results when combined.



5. Figure - confusion matrix of the simple averaging ensemble

As it is observable in our code, we experimented with different solutions. Simple averaging, overall- and categorical accuracy based weighted-averaging and exponential weighting. With the exception of the simple averaging, we faced a dilemma: whether to determine the weights based on the validation dataset or a part of the test dataset. Potentially, tuning these on the test data yields better results, as more pictures are used in the whole training part. However, in order to be able to compare our results to the results other scholars reported, we decided to do the weighting based on the validation data.

Figure 5 depicts the best obtained results, which was around 69%, using simple averaging. All the used weighted averages were always close to the output of the simple averaging, but always performed a bit worse, the difference was usually less than 1 percent. This means an overall over 2,5% increase compared to the standalone networks.

The categorical weighting, which combines the outputs of the networks assigning different weights to their output channels, consistently underperformed the other solutions.

Interestingly, additional models almost always enhanced the accuracy, even when the added model performed under the average of the already considered models. The best performance was achieved when considering a total of 8 models, of which three performed not better than 60.2%. This confirms again, that ensembles are able to provide robustness by the redundancy of a completely different architecture, optimization or training approach.

VI. CONCLUSION

During our project we considered the problem of static FER: the classification of human faces based on the depicted emotion. We used the widely known FER2013 dataset and built models based on the literature. After achieving the expected accuracy levels with slightly modified models, we built ensembles using different strategies. Although weighting based on different confidence measures, such as overall and categorical accuracy were implemented, the simple averaging yielded slightly better results. An over 2.5% increase was reached using such ensembles.

However, this performance gain does not come free. Naturally, using more models for the same task will require more resources, which may not be available in the case of a real-world application. Also, when observing the depicted confusion matrices, one may notice that the performance of certain networks may be better in the case of certain categories (see neutral) as the combined result – which, depending on the application, may lead overall worse performance. (Note that the two best standalone confusion matrices are depicted which were both built into the best ensemble.)

Our project was only an entry into the field of FER. State-of-the-art solutions work with even deeper networks utilizing pre-training and transfer learning approaches, to counter the lack of data. Other approaches build ensembles too but consider a much higher number of standalone networks. Combining both approaches – going deeper and working with many parallel networks – could yield the most accurate and most costly solutions yet.

REFERENCES

[1] Li, Shan; Deng, Weihong (2020): Deep Facial Expression Recognition: A Survey. In *IEEE Trans. Affective Comput.*, p. 1. DOI: 10.1109/TAFFC.2020.2981446
[2] Challenges in representation learning: Facial expression recognition challenge: http://www.kaggle.com/c/challengesin-representation-learning-facial-expression-recognitionchallenge
[3] Pramerdorfer, Christopher; Kampel, Martin (2016. 12. 09): Facial Expression Recognition using Convolutional Neural State of the Art.
Available online at http://arxiv.org/pdf/1612.02903v1.
[4] Agrawal, Abhinav; Mittal, Namita (2020): Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy. In Vis Comput 36 (2), pp. 405–412. DOI: 10.1007/s00371-019-01630-9.
[5] Alizadeh, Shima; Fazel, Azar (2017. 04. 22): Convolutional Neural Networks for Facial Expression Recognition.
Available online at http://arxiv.org/pdf/1704.06756v1.
[6] Boughrara, Hayet; Chtourou, Mohamed; Ben Amar, Chokri; Chen, Liming (2016): Facial expression recognition based on a mlp neural network using constructive training algorithm. In Multimed Tools Appl 75 (2), pp. 709–731. DOI: 10.1007/s11042-014-2322-6
[7] Arushi Raghuvanshi and Vivek Choksi, "Facial Expression Recognition with Convolutional Neural Networks", CS231n Course Projects, Winter 2016
[8] Fernandez, Pedro D. Marrero; Pena, Fidel A. Guerrero; Ren, Tsang Ing; Cunha, Alexandre (2019. 06. 16. - 2019. 06. 17): FERAtt: Facial Expression Recognition With Attention Net. In : 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2019 IEEE/CVF
[9] Zhang, Hongli; Jolfaei, Alireza; Alazab, Mamoun (2019): A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing. In IEEE Access 7, pp. 159081–159089. DOI: 10.1109/ACCESS.2019.2949741.
[10] Kim, Bo-Kyeong; Roh, Jihyeon; Dong, Suh-Yeon; Lee, Soo-Young (2016): Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. In J Multimodal User Interfaces 10 (2), pp. 173–189. DOI: 10.1007/s12193-015-0209-0
[11] Chieh-En James Li, Lanqing Zhao: Emotion Recognition using Convolutional Neural Networks, 2019 Purdue Undergraduate Research Conference