Ukrainian Catholic University

Faculty of Applied Sciences

Data Science Master Programme

Facial Expression Recognition in Natural Images

Machine Learning final project progress report

Authors:
Anastasiia Khaburska
Andrii Yurkiv

27 April 2019

1 Introduction

Facial expression is one or more motions or positions of the muscles beneath the skin of the face. These movements convey the emotional state of an individual and become the primal form of nonverbal communication.

It is assumed that certain facial expressions and gestures correspond to specific emotions (for instance, happiness is associated with laughter and smiling, sadness with tears, anger with clenched jaw, fear with grimace, surprise with raised eyebrows and wide eyes and disgust with wrinkled nose and squinted eyes) which are recognized by humans regardless of culture, language or time. But in general, this hypothesis has not been scientifically verified and received both critical and supportive reviews.

Both sides of this scientific debate agree that the face expresses emotion. The controversy surrounds the uncertainty about what specific emotional information is read from a facial expression [2].

2 Motivation

It has been proved multiple times that the majority of information human perceives (up to 83%) during message absorption is obtained through eyesight. For this purpose body language in general, and facial expression, in particular, are the things which provide the most essential and specific information about the intentions and emotional state of the message source.

It's not hard to conclude that accurate perception of facial expressions is a key to effective face-to-face communication.

Humans have a great ability to perform this kind of tasks. And since main facial expressions are universal and do not vary with culture or environment, we usually recognize others emotional state pretty well (at least when our interlocutors are not intentionally hiding it).

But for a machine, it is not an easy task. We tried to visualize images in two-dimensional space using t-SNE algorithm [1]. As you can observe, facial expressions are not the things that the algorithms use to differentiate between images. It is skin colour, age gender what algorithm notices.

If we asked any human to take a look on faces (for example, those [2]) and say whether there is a substantial visual difference between them, we would certainly hear "yes". But, in fact, they are not that different. It is our ability to recognize these differences that makes humans consider them substantial. In general, there are few reasons why machine learning model usually perform much worse compared to humans in recognizing facial expressions (while outperforming in other classification tasks):

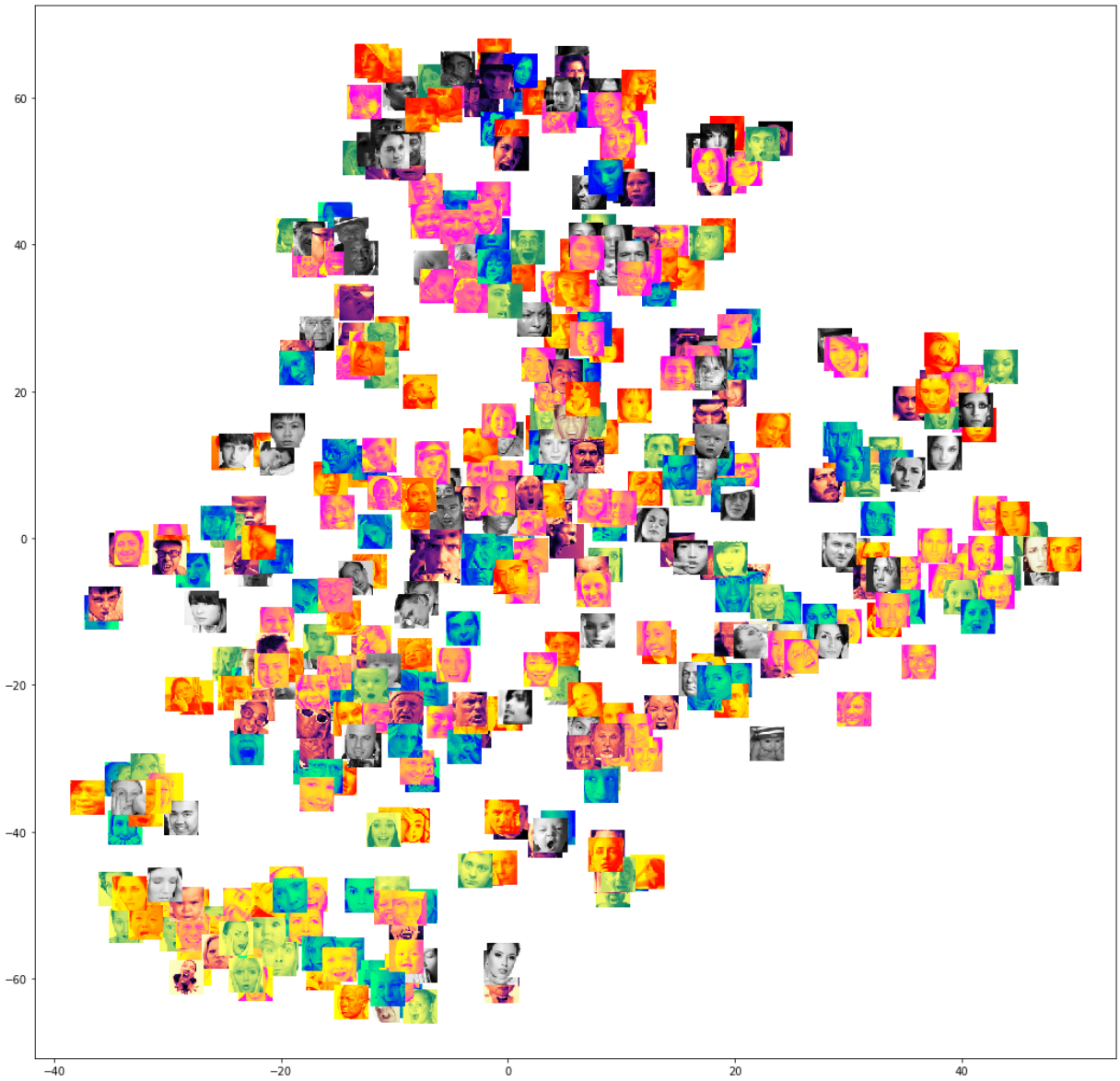

Figure 1: Visualization of 400 random images from `fer2013` data set using t-SNE algorithm

- Humans do not perceive facial expressions separately from other parts of body language. For facial expressions, context is very important, because it provides a great deal of additional information.

- Regardless of their universality, facial expressions are very diverse and vary substantially due to the race, age, gender, nationality and culture.

- People are brilliant at determining how they feel. But in some cultures, it is normal to hide your real emotions behind the neutral or happy facial expression. It is especially distinguishable in western cultures, where excessive emotionality is not considered as an element of effective communication. Machine learning model doesn't have this prior knowledge and hence use a generalized approach to cases where cultural nuances are crucial.

- Some kinds of facial expressions are very similar (for example, without context, it's hard to distinguish surprise from fear, or sadness from neutrality). And image similarity is a fundamental property when the model is trying to determine class depicted on it. The model fails to generalize well and makes mistakes by focusing on the wrong features.

In this project, we investigated deep learning approaches to the task of facial expression recognition and developed an end-to-end machine learning system of facial expression recognition in natural images. It consists of two parts: a face detector and expression recognizer. Unfortunately, we haven't found a data set to evaluate our end-to-end system (we need a data set that contains images with faced in different contexts labelled with facial expression classes). So, we evaluated our results in conjunction with state-of-the-art performance on `fer2013` data set, which is equal to 75.2%

3 Related Work

Byoung Chul Ko in a paper "A Brief Review of Facial Emotion Recognition Based on Visual Information"[3] gives an overview of researches in the field of FER conducted over the past decades. To tackle the problem of FER researchers tried such techniques :

1. Conventional FER approaches along with the representative categories of FER systems and their main algorithms.

2. Deep-learning-based FER approaches

3. Hybrid deep-learning approach combining a convolutional neural network (CNN)for the spatial features of an individual frame and long short-term memory (LSTM) for temporal features of consecutive frames.

Papers in this area are usually focused on recognizing human emotion in the context of video data (consecutive frames)[4]. In terms of this homework project, we investigated deep-learning-based approaches, that solve the problem for from still images (individual frames). An overview of various deep learning methods applicable to this particular

Figure 2: Examples of images from `fer2013` data set

problem was also made by Alexandru Savoiu and James Wong in [6]. Our contribution was to learn and implement different approaches, create own models, train them on the conjoint data-set, evaluate and then compare.

4 Project idea. Data

The project idea was inspired by Kaggle competition "Challenges in Representation Learning: Facial Expression Recognition Challenge". It was organized in 2013, but we think that the topic is still relevant today. The `fer2013` data set [1] for this competition was prepared by Pierre-Luc Carrier and Aaron Courville.

The `fer2013` data set consists of 48×48 pixel gray-scale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

The data set is divided into train, validation and test splits. The training set consists of $28,709$ examples. The validation set consists of $3,589$ examples. The final test set consists of another $3,589$ examples.

Expression	Train	Validation	Test	% of samples
Angry	3995	467	491	13.8
Disgust	436	56	55	**1.52**
Fear	4097	496	528	14.27
Happy	7215	895	879	**25.05**
Neutral	4965	607	626	17.27
Sad	4830	653	594	16.93
Surprise	3171	415	416	11.15

Table 1: Distribution of classes in the data set.

The classes in the data set are not well-balanced. First of all, we have highly under-represented `Disgust` class. Also, 25.05% of samples correspond to the `Happy` class, which is more than the fraction of `Angry` and `Surprise` altogether (24.95%).

This imbalance issue imposes additional activities in our development process:

- We performed data augmentation in order to make it more robust.
- We used minibatch normalization, to reduce the covariance shift and to increase training rates.
- As an evaluation metric, we used the confusion matrix to determine the weaknesses of the model and possible improvement strategies.

Besides the imbalance, `fer2013` data set contains trash samples (which do not contain a face) and several misclassified examples. These imperfections make the classification harder because the model has to generalise well and be robust to incorrect data.

5 Approach to the solution

5.1 Face detection

Since we haven't found a data set on which we can assess both face detection and facial recognition models performance we decided to use a pretrained Haar Cascade model for front-face detection from `cv2` python module. It works well in general but sometimes fails to detect faces in unusual conditions or under unusual aspect. But it is absolutely fine if we take into account the scope and setting of our project.

5.2 Baseline model

Our baseline facial expression recognition model is a 4-layer CNN with 3 dense layers. The model architecture is visualized in [3].

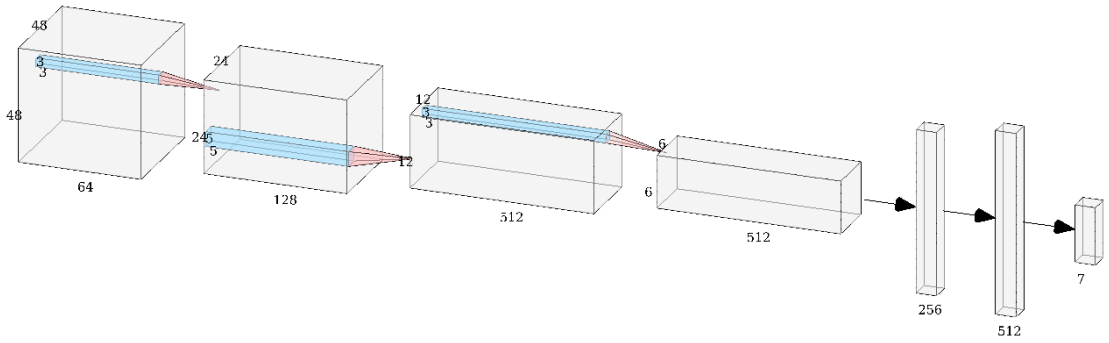

Figure 3: Baseline model architecture

On each layer except output one, we used ReLU activation function. On the output layer, we used Softmax function. We applied batch normalization after each convolutional layer in order to make our model converge faster and omit getting stuck in local minima. Besides it, we applied the Dropout regularization technique at each layer to prevent over-fitting and make our model generalize better.

5.3 Baseline model with data augmentation

As we mentioned in the previous section, the data set we were training our model on was very unbalanced. We decided to tackle this problem using data augmentation for the train set. The general approach was as following: we randomly augmented 100% of `Disgust` class images, 50% of `Fear` and `Angry` classes and 10% of `Surprise`, `Happy`, `Neutral` and `Sad` ones. We used 5 different augmentation techniques for each image: horizontal flipping, left/right rotation by 15 degrees, noise and blur introduction. As a result, we increased our train set size by 32490 images. Our baseline model was simple enough to handle this amount of data without a substantial increase in training time.

We didn't apply data augmentation for modification of VGG-19 and Original VGG-19, because it would take much longer time to train those models with 2× more data.

Data augmentation increased the performance of the baseline model by nearly 3%, which means that it was a reasonable procedure to perform. One important thing to note is it was crucial for the model that dropout rate at each layer was equal to 0.5. Otherwise, model over-fitted and performed poorly on test data.

5.4 Baseline model with HOG

During the data analysis stage of our project, we had an idea that the introduction of additional features to the model could increase its performance. There are a lot of ways to extract additional information from the image. In our setting the most applicable were Face Landmarks and HOG (Histogram of Oriented Gradients). Each of them employs a different approach to feature extraction and, hence, results in different amount of features. For example, using Face Landmarks, we usually get less than 100 features and using HOG we can get up to $n \times n$ features (where n is image height).

Observing results obtained in [12], we concluded that HOG features/Face Landmarks do not considerably improve the model performance in our problem setting. But we still decided to check this assumption.

When choosing among Face Landmarks and HOG we chose the second one, because it is much easier to implement. We added a branch with HOG input to the baseline model, and trained it on the regular data without augmentation. The model obtained performed really poorly (accuracy decreased by 4%) and consequently we decided not to use additional features in the future modelling process.

5.5 AlexNet model

One more model we tried was an adaptation of the classic AlexNet model. This model consists of 6 convolutional layers and 4 dense layers. ReLU activation function was used on each layer except the output one, where Softmax is employed. In order to simplify this model we decreased the number of neurons in the dense layers (first - by half, second - by 3/4 and third - by half). Besides it, we had to decrease kernels and pooling size in the first layers of the network, since the original model was developed for 224×224 images, and our images are much smaller.

The AlexNet model architecture is depicted in [4].

AlexNet model proved to perform worse than the baseline in both regular and augmented train set settings.

5.6 VGG-19 model

The original VGG network described in [7] uses an architecture with very small (5x5; 3x3) convolution filters, consecutively with depth 64, 128, 256, 512, pushing size up to 19, depending on the problem. This architecture showed sufficient accuracy for classification tasks of large scale RGB images .

We adjusted the VGG network for images of size $48 \times 48 \times 1$. During training, the best result of 66.5% accuracy was obtained with a 7-layer VGG network. The architecture of this adjusted VGG network is shown in [5]. The only preprocessing we did was

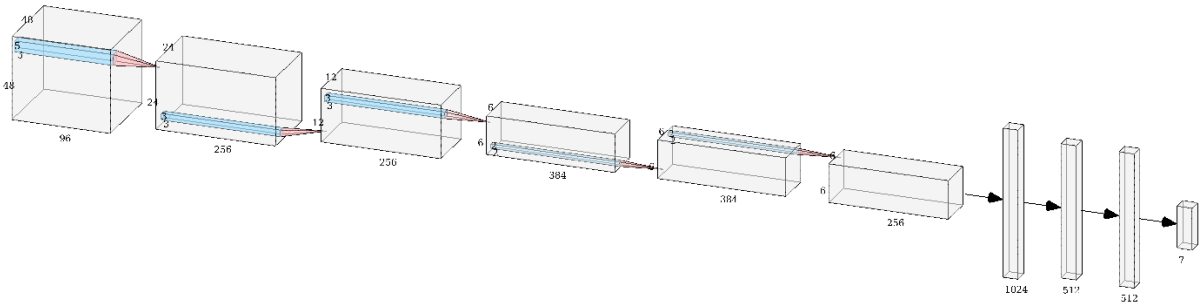

Figure 4: AlexNet model architecture

image scaling to the range [0:1] by the division of each pixel by 255. Then, the image is passed through the convolutional layer, with receptive field of size 5×5, and then through stack of convolutional layers, where we use filters with a very small receptive field 3x3 (which is the smallest size to capture the notion of left/right, up/down, centre [7]). The convolution stride is fixed to 1 pixel and the padding is adjusted in a way that after each convolution the size of the image remains the same. Also, for downsampling only the important features of the images, as in [9], we decided to add MaxPooling after the first convolution and the Average Pooling after each second convolution. Each pooling reduces the height and length of the layer by two. To prevent overfitting even more, as suggested in [10] and [11], we decided to use dropout in all layers with a fraction 0.5.

On each layer except the output one, we used ReLU activation function. On the output layer, we used Softmax function. To address the problem of internal covariance shift, we performed the normalization for each training mini-batch. Batch Normalization also acts as a regularizer and achieves the same accuracy with fewer training steps [8].

5.7 Ensamble approach

The models we trained (even the best ones) showed pretty mediocre results on the test set, comparing to the state-of-the-art models proposed, for example in [14]. In this specific paper, authors achieved astonishing accuracy of 75.2% using an ensemble of 8 complex models (VGG, ResNet and Inception). We decided that this approach would be reasonable in our setting as well.

We tried two different ensembles: combinations of 3 strongest models (in our case - Baseline with Augmentation, VGG and Original VGG) and combination of 5 strongest models (those used in previous ensemble plus Baseline and AlexNet with Augmentation). As a result, we got a substantial increase in accuracy for both validation and test sets. As expected, an ensemble of 5 models performed better than the ensemble of 3.

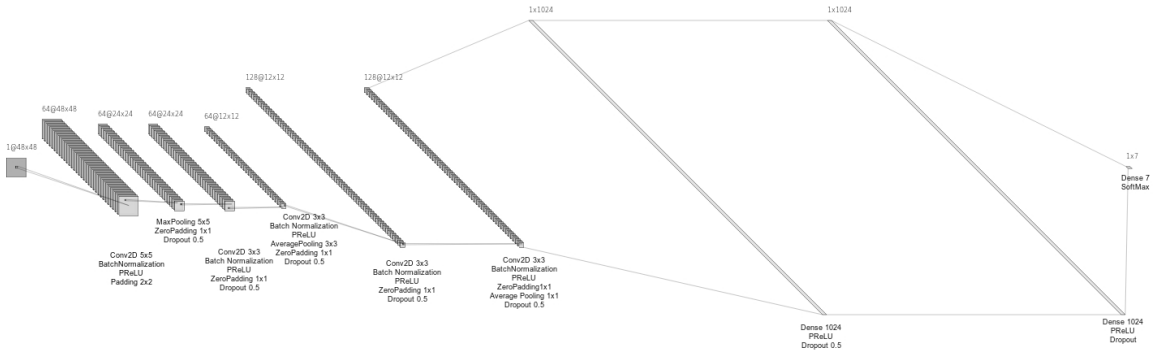

Figure 5: Customized VGG model architecture

For the ensemble of 5 models, we have used "smart" voting system by weighting models' votes using their validation accuracy. It was very beneficial in those cases when two models predicted one class and two - the other one. This approach allowed us to fight ambiguity and let better models to make the decision.

5.8 Training

We trained our models for a different number of epochs depending on such parameters as cross-entropy loss and categorical accuracy. We tried different approaches and tuned hyper-parameters many times to make our models more robust (we changed learning rates, dropout rates, added/removed layers, decreased their size, changed optimizers and enhanced existing models with augmented data). Besides it, we always reused pretrained weights in the future iterations to accelerate computations and iterate faster.

It should be noted that validation loss didn't fall below 0.97 in any of our experiments. During the training process, it usually oscillated around 1 and then, after a large number of epochs began to slowly but gradually increase till the end of the training. We tried a lot of things to change this behaviour but didn't succeed.

6 Evaluation

The accuracies of the model used can be seen in 2. Two types of models performed best in the FER task: simple models trained on aggressively augmented training data (BaseAug) and complex models trained on regular training data. Those models that were somewhat in the middle (AlexNet) didn't really perform well. We explain it by the specificity of the task. A simple model (shallow, if we are talking about CNN) can't learn complex

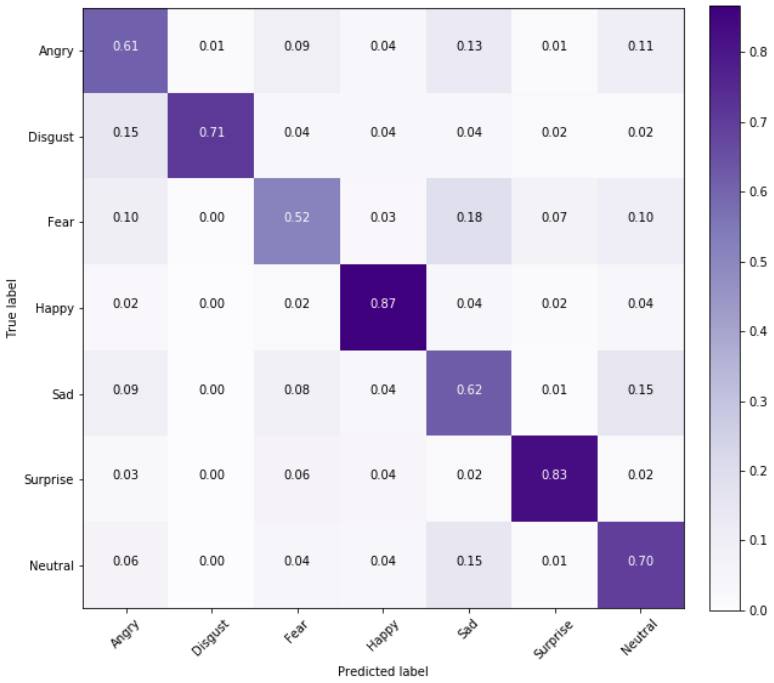

Figure 6: Normalized confusion matrix

hypothesis. That's why it benefits a lot from augmented data. Complex, deep CNN models, on the other hand, have this ability so that they can perform well even without data augmentation.

As we can observe from the confusion matrix [6], constructed using predictions of ensemble of 5 models, it does a great job recognizing `Happy` and `Surprise` expressions, while failing to recognize accurately such expressions as `Fear`, `Sad` and `Angry`. In general, it's not an unexpected result. Those are complex and ambiguous facial expressions that are often misinterpreted if seen without corresponding context.

	Base	BaseAug	Alex	AlexAug	VGG	VGG-O	E3	E5
Validation	0.636	0.655	0.616	0.635	0.646	0.662	0.668	**0.681**
Test	0.642	0.672	0.635	0.643	0.670	0.675	0.693	**0.703**

Table 2: Performance of the models and ensembles. Base: Baseline, BaseAug: Baseline with Augmentation, Alex: AlexNet, AlexAug: AlexNet with Augmentation, VGG: Modified VGG, VGG-O: Original VGG, E3: Ensemble of 3 strongest models, E5: Ensemble of 5 strongest models

7 Conclusions

As a result of this project, we developed an efficient end-to-end system for facial expression recognition in natural images. We didn't succeed to achieve state-of-the-art accuracy on the data set, but still obtained performance that would provide us a second absolute place in the original Kaggle leaderboard (70.3%).

Our final solution for classification problem is an ensemble of 5 models: baseline CNN model, baseline model trained on augmented data, AlexNet model trained on augmented data, modified VGG network and Original VGG network. This is not the fastest or most efficient solution, but it generalizes well.

During the project implementation we made some important observations:

- Facial expressions are really hard to recognize. They are subtle, ambiguous and culture-specific. Creating a perfect classifier without additional contextual information is impossible.
- Complex CNN models (like VGG) really worth it. They may be hard to train or store, but at the same time, they can learn highly complex non-linear hypothesis which gives them a considerable advantage over simpler models.
- HOG doesn't worth it (at least in this problem). Facial expressions are too complex and subtle to be noticed by HOG, and as a result, it just introduces unnecessary noise which doesn't make the model perform better.
- Data augmentation is a very reasonable way to improve your model and make it generalize better by smoothing data imbalance. Other important question is what and how to augment, especially in cases when two classes are very similar (`Sad` and `Neutral` in our case).
- If you hit the ceiling with separate models, try ensemble. It will for sure make your model perform better.
- Computational resources are very important. They allow to iterate faster and to implement your idea in hours instead of days.
- There is no need to implement existing things from scratch if you want to move fast. In our case, for example, face detection problem was solved a long time ago (it is even a part of an existing python package), so we decided not to concentrate on it.

The code, experiments and scripts can be found in the project GitHub repository:

https://github.com/andreyurkiv/facial-expression-recognition.

References

[1] I. Goodfellow et al. *Challenges in Representation Learning: A report on three machine learning contests.*
https://arxiv.org/pdf/1307.0414.pdf

[2] Wikipedia contributors *Facial expression Wikipedia, The Free Encyclopedia.*
https://en.wikipedia.org/w/index.php

[3] B. C. Ko *A Brief Review of Facial Emotion Recognition Based on Visual Information.*
https://pdfs.semanticscholar.org/1e7a/e86a78a9b4860aa720fb0fd0bdc199b092c3.pdf

[4] S.Kahou, C. Pal, X. Bouthillier, P. Froumenty, C. Gulcehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio. *Combining Modality Specific Deep Neural Networks for Emotion Recognition in Video.*
http://www.professeurs.polymtl.ca/christopher.pal/icmi2013/icmi2013_grand_challenge_winner.pdf

[5] T. Le. *Applying Artificial Neural Networks for Face Recognition.*
https://www.researchgate.net/publication/258380240_Applying_Artificial_Neural_Networks_for_Face_Recognition

[6] A. Savoiu, J. Wong. *Recognizing Facial Expressions Using Deep Learning.*
http://cs231n.stanford.edu/reports/2017/pdfs/224.pdf

[7] K. Simonyan, A. Zisserman. *Very deep convolutional networks for large-scale image recognition.*
https://arxiv.org/pdf/1409.1556.pdf

[8] S. Ioffe, C. Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.*
https://arxiv.org/pdf/1502.03167v3.pdf

[9] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C.Courville, and Y. Bengio. *Maxout Networks.*
https://arxiv.org/pdf/1302.4389.pdf

[10] Stefan Wager, Sida Wang and Percy Liang. *Dropout Training as Adaptive Regularization.*
https://arxiv.org/pdf/1307.1493.pdf

[11] N. Srivastava, G. Hintonhinton, A. Krizhevskykriz, I. Sutskeverilya, R. Salakhutdinov. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting.*
http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

[12] A. Horseman *Facial expression recognition using CNN in Tensorflow*
https://github.com/amineHorseman/facial-expression-recognition-using-cnn

[13] A.Krizhevsky, I. Sutskever, G. Hinton *ImageNet Classification with Deep Convolutional Neural Networks*
https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-n
pdf

[14] C. Pramerdorfer, M. Kampel *Facial Expression Recognition using Convolutional Neural Networks: State of the Art*
https://arxiv.org/pdf/1612.02903.pdf

A Appendix

(a) Original image

(b) Predicted facial expressions

Figure 7: Baseline model results