Recognizing Facial Expressions Using Deep Learning

Alexandru Savoiu
Stanford University
savoiu@stanford.edu

James Wong
Stanford University
jwhwong@stanford.edu

Abstract

In this project we applied various deep learning methods (convolutional neural networks) to identify the key seven human emotions: anger, disgust, fear, happiness, sadness, surprise and neutrality. We used the Kaggle (Facial Expression Recognition Challenge) and Karolinska Directed Emotional Faces datasets. The architectures we employed for our convolutional neural networks were VGG-16 and ResNet50. We used the support vector machine multiclass classifier as our baseline, which had an accuracy performance of 31.8%. To further improve our results, we leveraged ensemble and transfer learning techniques to achieve our best results. Thus, the accuracy using ensemble learning was 67.2% and with transfer learning was 78.3%, solid results given that the winner of the Kaggle Facial Expression Recognition Challenge had an accuracy of 71.2%, and those who ranked in the top 10 of the same competition only achieved accuracies starting at around 60%.

1. Introduction

More than 90% of the human communication is nonverbal [1].
Professor Albert Mehrabian, UCLA

Understanding human emotions is a key area of research, as the ability to recognize one's emotions can give one access to a plethora of opportunities and applications, ranging from more friendly human-computer interactions, to better targeted advertising campaigns, and culminating with an improved communication among humans, by improving the emotional intelligence ("EQ") of each of us. While there are multiple ways one can investigate the recognition of human emotions, ranging from facial expressions, posture of the body, speed and tone of the voice, in this paper we shall focus on only one area of this field - visual recognition of facial expressions.

One of the reasons we chose to focus on the area of facial expressions is because certain facial expressions have universal meaning, and these emotions have been documented for tens and even hundreds of years. Thus, nowadays, most databases containing facial emotions use the same key classification of the human emotions as it was originally presented in a paper by Ekman et al in 1971 - "Constants across cultures in the face and emotion" [2]. That paper identified the following six key emotions: anger, disgust, fear, happiness, sadness and surprise. These are the same emotions that are being used by current researchers to identify facial expression in computer vision, or in competitions such as Kaggle's Facial Expression Recognition Challenge, along with the addition of a seventh, neutral emotion, for classification.

Thus, our research is about using deep learning (a VGG-16 convolutional network and a ResNet50 convolutional network) to identify these seven main human emotions [3]. To us this problem is extremely relevant because of its broad spectrum of applicability in a variety of fields, such as systematic recruiting, while being also able to be integrated with a variety of technologies (i.e. smart glasses, VR, wearables, etc.). Emotions and facial responses can also serve as a new dimension of user information (i.e. imagine Facebook or Google analyzing your emotions and reactions to learn more about the user and serve better recommendations and ads).

To achieve our goals we will use a support vector machine (SVM) classifier baseline model and develop a convolutional neural network (CNN) to classify these emotions. In particular, we will use some of the current state of the art architectures - VGG-16 and ResNet50, while making some adjustments which include: applications of various deep learning techniques, and ensemble and transfer learning [5]. We chose to go with VGG-16 and ResNet50 because they won in the past the ImageNet challenge, achieved near state of the art results in terms of prediction accuracy, and follow a relatively standard CNN architecture. The two datasets we will leverage in our research are the Kaggle's Facial Expression Recognition Challenge and Karolinska Directed Emotional Faces (KDEF) datasets. We found these datasets to be representative because of their size, unstructured nature of faces (in terms of facial orientation, ethnicity, age, and gender of the subjects) and relatively uniform distribution of the data across the seven main human emotions (disgust being the only underrepresented one within the Kaggle dataset, at ~1.5%).

To evaluate the performance of our models, we will primarily be looking at the accuracy on the training, validation, and test sets. To facilitate the training and tuning

processes, we will be leveraging other standard statistics such as precision and recall to provide further insights on the efficacy of the models. We expect our best model to achieve at least 60% test set valuation because the winner of the Kaggle challenge achieved 71.2% accuracy and the top ten contestants achieved at least 60% accuracy.

2. Related Work

2.1. Psychological Framework

Last years represented a flourishing era for research in the field of human emotions recognition [8, 9, 10], and a dominant psychological framework for describing the facial movements emerged - the Facial Action Coding system (FACS) [11]. FACS is a system that classifies the human facial movements by their appearance on the face using Action Units (AU). An AU is one of 46 atomic elements of visible facial movement or its associated deformation; an expression typically results from the accumulation of several AUs [8, 9]. Among the research in the area to detect the basic AUs of the FACs, some that stand out are:

- Tian *et al.* who developed an automatic face analysis (AFA) , with a 95.6% recognition rate on the Cohn-Kanade Database, Version 1 (CK) [11].
- Donato *et al*. who were able to achieve 96.9% recognition using Gabor wavelet decomposition [12].
- Bazzo and Lamar who invented a pre-processing step based on the neutral face average difference and used a neural-network-based classifier combined with Gabor wavelet to obtain recognition rates of 86.55% and 81.63%, respectively, for the upper and the lower faces [13].

Recent developments in terms of the techniques used for facial expression recognition include: Bayesian Networks, Neural Networks and the multi-level Hidden Markov Model (HMM) [14, 15].

2.2. Area of Focus

Overall, papers in this area have been focused on recognizing human emotion in the context of video footage or based on audiovisual data (mixing speech recognition and video techniques). Many papers seek to recognize and match faces (e.g. [16]), but most papers do *not* use convolutional neural networks to extract emotions from still images. An exception to this is a paper by Kahou et al. which ([17]) actually trains a deep convolutional neural network on a set of static images, but then applies this to video data.

2.3. Dedicated Competitions

Dedicated to this topic, there are two major competitions: the Kaggle one, from which we used the dataset, and the *Emotion Recognition in the Wild Challenge.* The winner of the Kaggle competition used a deep neural net (based on CIFAR-10 weights) to extract features and then SVM for classification while the winners of the Emotion Recognition Competition from 2016 used convolutional neural networks (CNN-RNN and C3D Hybrid Networks).

3. Methods

3.1. Support Vector Machine (SVM)

As our baseline, we used a linear classifier trained with multi-class support vector machine loss, that has the following score function (Equation 1):

$$f(x_i, W, b) = Wx_i + b \qquad (1)$$

where x_i is an image's pixel data flattened to a K × 1 vector, W is a C×K weight matrix, and b is a C×1 bias vector. The output of the function is a C × 1 vector of class scores, where C is the number of classes. As the score for a class is the weighted sum of an image's pixel values, we can interpret the linear classifier as how much an image matches the "template" for a class.

After we computed the class scores, we use a loss function to quantify how well the classifier performs, where the the i^{th} loss has the formula (Equation 2):

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + \Delta) \qquad (2)$$

where y_i is the correct class for x_i. The SVM loss will be non-zero for a class j y_i when the score for class j is not at least Δ lower than the score for the correct class y_i. A commonly used value for Δ, and one adopted here, is Δ = 1.

To discourage the weights from taking on arbitrarily large values, we add an L2 regularization term to the loss function (Equation 3).

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + \lambda \sum_{j=1}^{C} \sum_{k=1}^{D} W_{j,k}^2 \qquad (3)$$

where $W_{j,k}$ is the (j, k) entry of the weight matrix and λ is a hyper-parameter determined through cross-validation.

The goal of training is to minimize the loss across training data. Each element of the weight and bias is initialized as a Gaussian with mean zero and some small standard deviation. At each iteration, the derivative of the loss is calculated with respect to W and b, and the parameters are updated using stochastic gradient descent. We leveraged the scikit-learn implementation of SVM [18].

3.2. VGG-16

VGG-16 represents one of the state of the art architectures for convolutional neural networks, with 16 CNV/FC layers and with an extremely homogenous architecture that only performs 3x3 convolutions and 2x2 pooling from the beginning to the end (Figure 1). The downside of the VGG-16 is that is more expensive to evaluate and uses significantly more memory and parameters (140 millions), where most of these parameters are located in the first fully connected layer. Like a linear classifier, convolutional neural networks have learnable weights and biases; however, in a CNN not all of the image is "seen" by the model at once, there are many convolutional layers of weights and biases, and between convolutional layers are nonlinear functions that in combination allow the model to approximate much more complicated functions than a linear classifier.

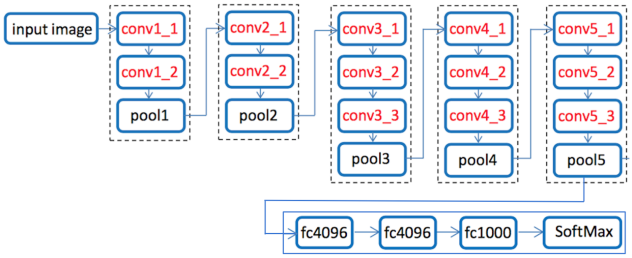

Figure 1: VGG-16 architecture diagram.

The input to our VGG-16 is a 48x48 RGB image. The only preprocessing we do is subtracting the mean RGB from each pixel. The image is passed through a stack of convolutional layers, where we use 3x3 filters. In one of the configurations we also utilize 1 × 1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of convolutional layer input is such that the spatial resolution is preserved after convolution (i.e. the padding is 1 pixel for 3 × 3 conv. layers). Spatial pooling is carried out by five max-pooling layers, which follow some of the convolutional layers (not all the convolutional layers are followed by max-pooling). Max-pooling is performed over a 2 × 2 pixel window, with stride 2.

A stack of convolutional layers is followed by three Fully-Connected (FC) layers: the first two have 4096 channels each, the third performs 7-way ILSVRC classification and thus contains seven channels (one for each class). The final layer is the softmax layer. The configuration of the fully connected layers is the same in all networks. All hidden layers are equipped with the rectification (ReLU) nonlinearity.

To conclude, VGG-16 consists of 16 weight layers that include 13 convolutional layers with filter size of 3x3 and 3 fully-connected layers. The stride and padding of all convolutional layers are fixed to 1 pixel. All convolutional layers are divided into 5 groups and each group is followed by a max-pooling layer (Figure 1). Max-pooling is carried out over a 2x2 window with stride 2. The number of filters of convolutional layer group starts from 64 in the first group and then increases by a factor of 2 after each max-pooling layer, until it reaches 512. We leveraged the keras implementation of VGG-16 [19].

3.3. ResNet50

ResNet50 is another current state of the art convolutional neural network architecture. It is similar in architecture to networks such as VGG-16 but with the additional identity mapping capability (Figure 2).

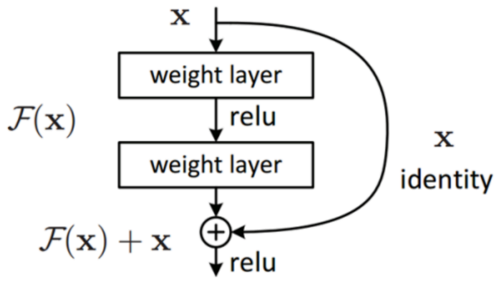

Figure 2: ResNet residual block diagram with identity mapping.

Rather than fitting the latent weights to predict the final emotion at each layer, ResNet models fit a residual mapping to predict the delta needed to reach the final prediction from one layer to the next. The identity mapping enables the model to bypass a typical CNN weight layer if the current layer is not necessary. This further helps the model to avoid overfitting to the training set. From an overall architecture and performance perspective, ResNet allows for much deeper networks while training much faster than other CNNs. In the case of ResNet50, there are 50 layers. We leveraged the keras implementation of ResNet50 [19].

3.4. Ensemble Learning

While VGG-16 and ResNet50 are currently two of the state of the art deep learning architectures, we attempt to combine these two models by leveraging an ensemble approach. From the second to last layers, we obtain a vector of weights which can be treated as feature vectors. These feature vectors represent the latent representation of the input image which each model learned. We combine these latent representations by concatenating the feature vectors to form an overall feature vector which is inputted into logistic regression models for the final emotion prediction (Figure 3). We train one logistic regression for each emotion, for a total of seven models, and taking the model with the highest score as the prediction. So for each image we compute nine predictions: one from VGG-16, one from ResNet50, and seven from the logistic regression models. We leveraged the scikit-learn implementation of logistic regression [18].

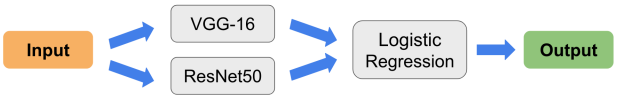

Figure 3: Ensemble learning architecture with VGG-16 and ResNet50 as input models into the logistic regression ensemble model for final predictions.

3.5. Transfer Learning

Transfer learning is a commonly applied technique which takes the learned weights of a model from a larger dataset (e.g. ImageNet) and applies those by fixing various layers and retraining the remaining layers or fine tuning the network. In this project, we apply transfer learning by taking the learned weights from the Kaggle dataset, a significantly larger and broader dataset, and retraining a few, later layers on the KDEF dataset, a smaller dataset. We chose this approach because both KDEF and Kaggle contain similar data, images of subjects displaying one of the seven emotions.

4. Dataset & Features

4.1. Facial Expression Recognition Challenge

As mentioned, we wanted to choose those databases that not only provide a representative number of images, but also that contain data which is rather uniformly distributed across the race, sex, ethnicity and gender of the subjects, and with a relatively even distribution across the emotions of these subjects. The Kaggle dataset (from the Facial Expression Recognition Challenge) meets all the following attributes:

- 35,887 images
- Image Format: 48 x 48 pixels (8-bit grayscale)
- Various individuals across the entire spectrum of: ethnicity, race, gender and race, with all these images being taken at various angles
- Contains the seven key emotions (Figure 4)

Figure 4: Example images of the seven emotions in the Kaggle dataset.

- These seven key emotions are relatively equally distributed with the one exception being disgust, at ~1.5% (Figure 5)

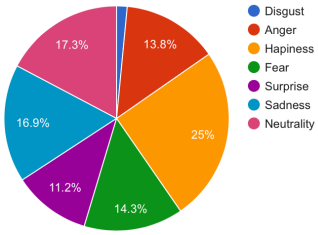

Figure 5: Data distribution of the Kaggle dataset across the seven emotions.

4.2. Karolinska Directed Emotional Faces

As mentioned, one of the techniques we wanted to investigate in this paper was transfer learning. To accomplish this, we needed another database, with similar features to those exhibited by the main database (in our case, Kaggle), but with even an greater detail richness in terms of the angle selection of the photos of the subjects. KDEF almost perfectly fits these requirements, with the one remark that we will need to do some data processing for KDEF images to have the same color and format as those from the Kaggle database. Below are the key attributes of KDEF:

- 4900 images
- Image Format: 562 x 762 (32-bit RGB)
- 70 individuals, each displaying the seven different emotional expressions, and each expression is photographed twice from five different angles
- Representative across ethnicity, race, sex and gender
- The seven key emotions are uniformly distributed

4.3. Data processing

For both datasets, we mean centered the raw pixel data. For KDEF, we applied colorimetry (luminance-preserving) conversion to grayscale (from 32-bit RGB to 8-bit grayscale) and resized the original 562 x 762 images to 127 x 94 images (Figure 6). Reducing the size of the images improved training time. For ResNet50, the minimum image size input is 200 pixels on both dimensions, so we scaled the images from both datasets accordingly.

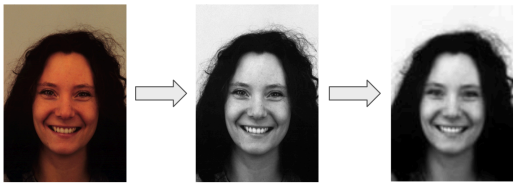

Figure 6: Original image from KDEF, converted to grayscale, and resized.

5. Results

To assess the performance of our models, we used a combination of accuracy, precision, and recall. Accuracy measures the proportion of true results amongst the

evaluated set, precision shows us the positive predictive value, and recall captures the sensitivity or true positive rate of the models. To compute the overall precision and recall, we use micro-averages to combine the results across all seven emotions. For both the Kaggle and KDEF datasets, we used a 80-10-10 split for the train, validation, and test sets. To further understand and assess our models, we examined the metrics for each emotion as well as the confusion matrix.

In Table 1 below, we see the results of the SVM (baseline), VGG-16, ResNet50 and ensemble learning models on the Kaggle dataset. Our baseline SVM accuracy was 31.8% while VGG-16 and ResNet50 had accuracies of 59.2% and 65.1%. Because ResNet50 contains identity bypass layers, it is possible that this is helping the model achieve better performance in terms of accuracy, precision, and recall compared to VGG-16. The ensemble learning model, which effectively combines VGG-16 and ResNet50, achieved an accuracy of 67.2%, 2.1% greater than either VGG-16 or ResNet50 individually.

	Accuracy	Precision	Recall
SVM (baseline)	31.8%	43.7%	54.2%
VGG-16	59.2%	70.1%	69.5%
ResNet50	65.1%	76.5%	74.8%
Ensemble	67.2%	79.4%	78.2%

Table 1: Kaggle dataset performance (accuracy, precision, and recall) for SVM, VGG-16, ResNet50, and ensemble learning models.

The overall accuracies along with precision and recall on the KDEF dataset are greater than those on the Kaggle dataset. SVM achieved an accuracy of 37.9% while VGG-16 and ResNet50 achieved accuracies of 71.4% and 73.8%, respectively (Table 2). The ensemble approach achieved an accuracy of 75.8% and continued to perform better than the individual deep learning models. The ranking of the four models is the same for KDEF as it is for Kaggle. We found it surprising that all four models performed better on the KDEF, a significantly smaller dataset than Kaggle. We conjecture that this may be a result of the structure and uniformity of the KDEF dataset in terms of the subjects' postures and number of examples for each subject and each emotion. The images in the KDEF dataset are also of higher quality. Aside from better image resolution, there were examples in the Kaggle dataset where there was, for example, text overlay in the background of the image.

	Accuracy	Precision	Recall
SVM (baseline)	37.9%	50.1%	54.9%
VGG-16	71.4%	81.9%	79.4%
ResNet50	73.8%	83.3%	80.7%
Ensemble	75.8%	85.0%	82.3%

Table 2: KDEF dataset performance (accuracy, precision, and recall) for SVM, VGG-16, ResNet50, and ensemble learning models.

Applying transfer learning further improved the results. After training the VGG-16 and ResNet50 models on the Kaggle dataset, we fixed the layer weights aside from the last few layers of these models and retrained on the KDEF dataset. This led to a 2.5% accuracy improvement in our ensemble model which was our best performing model (Table 3). Precision and recall were similarly improved. This shows that the model was able to leverage the learnings from the faces of the Kaggle dataset which contained a wider and more abundant distribution of data and transfer those learnings to the smaller KDEF dataset.

	Accuracy	Precision	Recall
VGG-16	73.6%	84.2%	81.1%
ResNet50	76.0%	86.1%	82.5%
Ensemble	78.3%	87.3%	84.3%

Table 3: KDEF dataset performance (accuracy, precision, and recall) with transfer learning from Kaggle models.

To help assess the model performance on each individual emotion, we summarize the findings in Figure 7. The minimum accuracy, precision, and recall are 56.1% (neutral), 48.2% (sad) and 56.1% (neutral). Sadness and neutrality, as we further discuss later on, possess similar facial features as each other and a couple other emotions. We also note that we performed the best on happiness, which may be due to having the most data coverage for this emotion. While it is surprising, due to the lack of data coverage, we achieved an 81.8% accuracy on disgust, the low precision indicates that the model may not have learned to distinguish disgust amongst other emotions and is predicting disgust more often than it should.

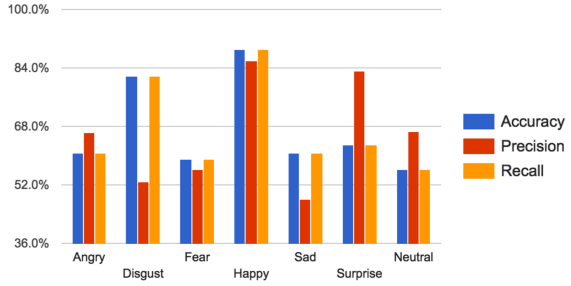

Figure 7: Ensemble learning performance on Kaggle dataset (accuracy, precision, recall).

Figure 8 shows the confusion matrix for our best performing model on the Kaggle dataset. The correlations between actual and predicted emotion hold for the other three models we experimented with. The matrix reveals that anger, disgust, fear, and neutrality tend to get miscategorized with sadness. Conversely, sadness tends to be miscategorized with the same set of emotions. Looking at the raw images, we can qualitatively see that the facial expressions for sadness have commonalities with that for those emotions, especially the aspects of the mouth area (aside from anger). Since we did not add additional features aside from the processed image pixels, it isn't surprising that these emotions are confused with one another. Lastly, surprise is confused with both fear and happiness.

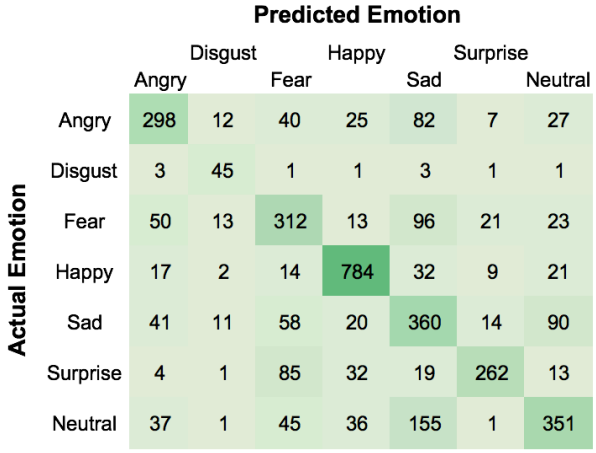

Figure 8: Confusion matrix with actual (true) emotion rows and predicted emotion columns (Kaggle, ensemble learning).

6. Conclusion

We explored the VGG-16 and ResNet50 architectures for recognizing facial emotions using deep learning. The results demonstrated that we were able to achieve acceptable results in comparison to other Kaggle contestants and researchers leveraging the KDEF dataset. We further improved these models by developing an ensemble model to combine the outputs from the two neural networks. Coupled with transfer learning, we achieved 67.2% accuracy on the Kaggle dataset and 78.3% accuracy on the KDEF dataset. For context, the winner of the Kaggle Facial Expression Recognition Challenge achieved an accuracy of 71.2% and the top 10 finalists achieved accuracies of at least 60%.

7. Future Work

In our work for this project, we trained the models using a pre-processed version of the raw image pixels. To further improve model performance, we wish to explore adding various facial and image features. We would also like to explore recognizing emotions in color images and to perform these predictions across the duration of a video. Lastly, we wish to explore leveraging deep learning beyond the seven basic emotions and extend our work to assess attributes such as confidence, composure, and credibility derived from the subject's micro-expressions.

References

[1] Albert Mehrabian. Silent Messages, University of California Los Angeles, 1971.

[2] P. Ekman and W. V. Friesen. Emotional facial action coding system. Unpublished manuscript, University of California at San Francisco, 1983.

[3] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1701-1708.

[4] Very deep convolutional networks for large-scale image recognition. Visual Geometry Group, Department of Engineering Science, University of Oxford, 2015.

[5] Ruiz-Garcia A., Elshaw M., Altahhan A., Palade V. (2016) Deep Learning for Emotion Recognition in Faces. In: Villa A., Masulli P., Pons Rivero A. (eds) Artificial Neural Networks and Machine Learning – ICANN 2016. ICANN 2016. Lecture Notes in Computer Science, vol 9887. Springer, Cham.

[6] Hiranmayi Ranganathan, Shayok Chakraborty, Sethuraman Panchanathan, "Transfer of multimodal emotion features in deep belief networks", Signals Systems and Computers 2016 50th Asilomar Conference on, pp. 449-453, 2016.

[7] Liu W., Zheng WL., Lu BL. (2016) Emotion Recognition Using Multimodal Deep Learning. In: Hirose A., Ozawa S., Doya K., Ikeda K., Lee M., Liu D. (eds) Neural Information Processing. ICONIP 2016. Lecture Notes in Computer Science, vol 9948. Springer, Cham.

[8] Y. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2), 2001.

[9] M.S. Bartlett, G. Littlewort, M. Frank, C. Lain- scsek, I. Fasel, and J. Movellan. Fully automatic facial action recognition in spontaneous behavior. In Proceedings of the IEEE Conference on Automatic Facial and Gesture Recognition, 2006.

[10] M. Pantic and J.M. Rothkrantz. Facial action recognition for facial expression analysis from static face images. IEEE Transactions on Systems, Man and Cybernetics, 34(3), 2004.

[11] P.Ekman,W.Friesen,Facial Action Coding System: A Technique for the Measurement of Facial Movement, Consulting Psychologists Press, 1978.

[12] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, October 1999.

[13] J. J. Bazzo and M. V. Lamar. Recognizing facial actions using gabor recognizing facial actions using gabor wavelets with neutral face average difference. In *Sixth IEEE International Conference on Automatic Face and Gesture Recogni- tion*, May 2004.

[14] D. Matsumoto, D. Keltner, M. N. Shiota, M. G. Frank, and M. O'Sullivan. *Handbook of emotions*, chapter What's in a

face? Facial expressions as signals of discrete emotions, pages 211–234. Guilford Press, New York, 2008.

[15] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler. Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 443–449, November 2015.

[16] Thai Hoang Le. Applying artificial neural networks for face recognition. *Advances in Artificial Neural Systems*, 2011:15, 2011.

[17] Samira Ebrahimi Kahou, Christopher Pal, Xavier Bouthillier, Pierre Froumenty, C ̧aglar Gu ̈lc ̧ehre, Roland Memisevic, Pascal Vincent, Aaron Courville, Yoshua Bengio, Raul Chandias Ferrari, et al. Combining modality specific deep neural networks for emotion recognition in video. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 543–550. ACM, 2013.

[18] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[19] Chollet François, Keras, (2015), GitHub repository, https://github.com/fchollet/keras