Facial Emotion Representation and Reconstruction with Deep Neural Networks

Alberto Barradas-Chacon[1], Salvador Rocha, Eduardo[2], and Thorben Werner[3]

[1] Universitt Hildesheim, Germany
barradas@uni-hildesheim.de
[2] Universitt Hildesheim, Germany
salvador@uni-hildesheim.de
[3] Universitt Hildesheim, Germany
wernerth@uni-hildesheim.de

Abstract. In this paper we propose deep neural network architectures that can be used to learn, classify and reconstruct images with facial expression of emotions. We also explore the latent representations of emotions, and the consequence of different representations on the reconstructed samples.

Keywords: Deep Learning · Emotions · Facial Expressions · CNN.

1 Problem Setting

Emotion detection from faces is a common problem in the area of affective computing that relies on visual cues from faces to detect specific emotional states of individuals. This problem, first described by Ekman [3] in 1992, is commonly solved by experts trained in the observation of specific facial features produced when a subject involuntarily contracts face muscles. The set of muscles that are activated during the expression of an emotion are also known as Action Units (AUs).
Since the problem of emotion detection on faces heavily relies on visual cues, Convolutional Neural Networks (CNNs) can be used to automatize the clasification problem. Burkert first showed a version of this in 2015[2], followed by Khorrami [7], who also showed that feature maps can be trained to selectively detect AUs.

1.1 Dataset

The Cohn-Kanade Plus (CK+) dataset [10] has been used for this problem. It contains the faces of 210 adults of ages between 18 and 50 years. It is relatively balanced, as described by Cohn "69% female, 81%,Euro-American, 13% Afro-American, and 6% other groups." [10]. Each subject was asked to preform the expression of an emotion. A sequence of images was recorded as they did. From the original set, 593 sequences from 123 subjects were included in the CK+. Each emotion can be classified by one or multiple sets of AUs. In CK+ six emotions where labeled in the data:

1. Anger
2. Contempt
3. Disgust
4. Surprise
5. Happiness
6. Sadness

The labels were one-hot encoded, they follow no particular structure for each actor, therefore the total number of samples per emotion varies as shown in figure 1.

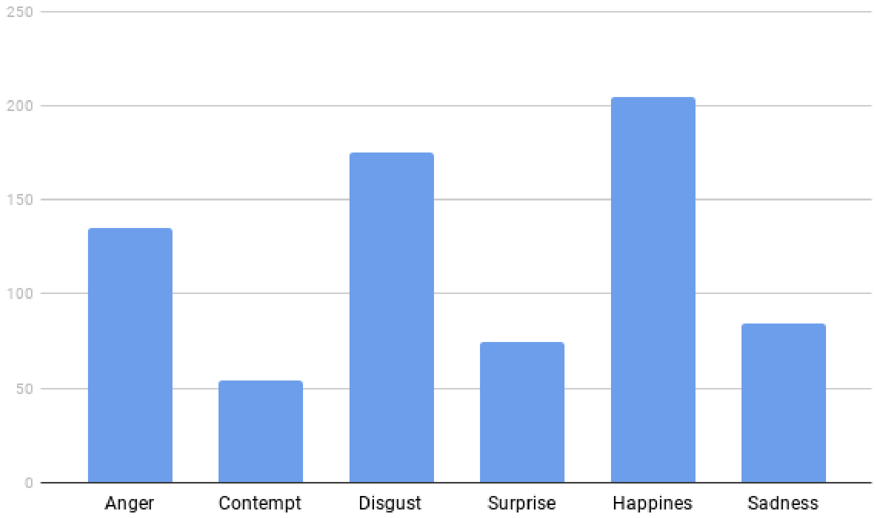

Fig. 1. Distribution of emotions in the CK+ dataset

The CK+ has been used on numerous experiments for emotion classification. [2, 7, 9, 11, 12, 4, 6, 14] Zhou presented a model of facial expression transfer in 2017, that uses adversarial Variational Auto Encoders (VAE) to transfer the expression from the face of one subject, to its neutral version. [16] Qiao presented, in 2018, an architecture for a generative model of emotions using Generative Adversarial Networks (GANs) and a geometric representation of faces, by including a points-of-interest (POI) extraction process in the data preprocessing, that can transfer the emotion not only to the same subject, but to new faces. [13]

In this paper, we explore three different generative architectures for facial expressions, their latent representations of emotions, and the role that this representation plays on the generated samples.

Data Augmentation For all experiments an augmented version of CK+ was used: To increase the number of samples and to normalize the label distribution, the dataset was augmented using image processing techniques.
This work does not employ sequence models, therefore only single images of the video sequence are used. For each recorded video the last three frames, which display the emotion in full effect, are used as three independent samples. In addition to the six emotions neutral expressions are for each actor are gathered. As each video starts with a neutral expression and transfers over the the emotion, a plethora of neutral images are included in the dataset.
Finally, each extracted image was cropped to the face of the person and rescaled to 128x128 pixels. To find the face of the person and crop the image, OpenCV [1] was used. The selected technique were Haar Cascade Classifiers, with the pre-trained frontal face model, provided by OpenCV
The following procedure was used to augment the dataset:

```
D = originalImages
for e : {'Anger', ..., 'Sadness'}:
    for i : [1 .. N - |e|]:
        img = sampleImageByEmotion(e)
        with prob. 0.5:
            img = flipHorizontally(img)
        D.append(applyRandomTransformation(img))

for i : [1 .. N]:
    D.append(sampleImageByEmotion('Neutral')
```

Where N is the final number of samples per emotion in the augmented dataset and $|e|$ is the number of samples of emotion e in the original data.

For each emotion, random images with that label are flipped horizontally with a probability of 0.5 and augmented with one of the following techniques:

- Random Brightness/Contrast Adjust
- Random Perspective Transform
- Random Rotation

Examples for each operation are displayed in figure 2.

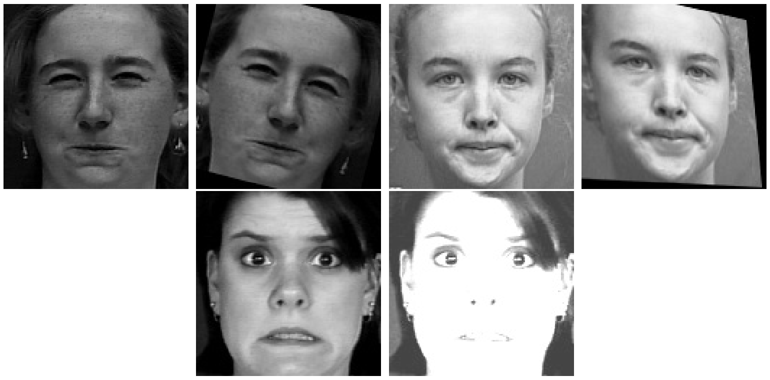

Fig. 2. Examples for the augmentation techniques: First pair: rotation; Second pair: Perspective Transform; Third pair: Brightness/Contrast

Lastly, N images were sampled from the original neutral images and added to the dataset with a zero-label, effectively introducing a seventh category to the labels. This was done to ensure the presence of a neutral expression in the generative models.
The final dataset label distribution is shown in figure 3.

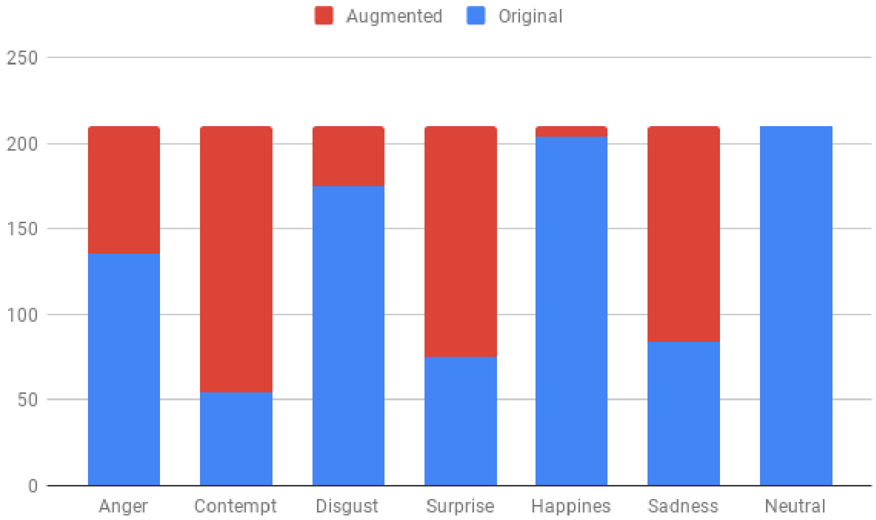

Fig. 3. Distribution of emotions in the augmented dataset

2 Methodology

Three different generative models have been tested. Each of them represents emotions in a latent space, that we use to explore how neural networks can learn emotions as a concept.

2.1 2D Latent Space Visualization

In order to obtain a visual representation of the images in a latent space, a classifier was trained and the output of the last dense layer was extracted and processed with a TSNE [15]. Let C_k be a convolution-MaxPooling-Relu layer with k filters. Convolutions use a kernel size 3x3, stride 1 and SAME padding, while Max pooling use kernel size 2x2 with a stride 1 and SAME padding. LetF_l be a Fully Connected-Relu layer of size l. Let D_p be a Drop out layer with drop rate p.

The classifier consists of the following layers:

$$C_8 - C_{32} - FlatteningLayer - F_{256} - F_{100} - D_{0.5} - D_7$$

2.2 Siamese Architecture

Idea Guide the autoencoder to focus on emotion relevant features instead of hair color and other strong features commonly extracted by autoencoders.

Employ a siamese comparator network, which tries to classifiy, if two given pictures of faces have the same emotion. By comparing an arbitrary face with the original emotion of the AE input with the reconstructed image the output of the siamese network can be incorporated into the AE loss.

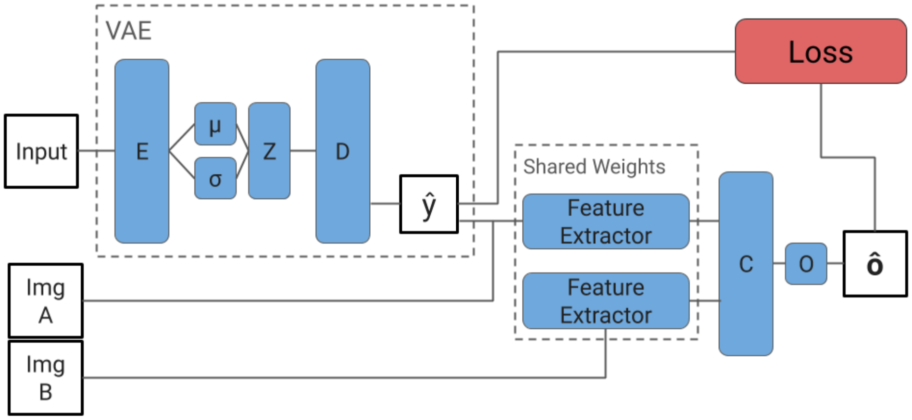

Fig. 4. Architecture of the VAE with siamese support network

The employed architecture is shown in figure 4. The loss for this network is

$$L = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) - \lambda \hat{o}(\hat{y}, ImgB)$$

Where the first part is a standard recontruction loss and the second part tries to maximize the response from the siamese network.
This architecture encourages to autoencoder to generate images that have a recognizable emotions in them, which maximize the response from the siamese network.

2.3 Fader Network Architecture

The main idea of a Fader Network is to constrain the latent space to be invariant to an attribute of interest. This approach is based on an encoder-decoder architecture where an image x with attribute y is mapped to a latent representation z s.t. z is identical for all possible attribute values of interest (i.e. a single latent representation can encode different images that share a common structure but with different attribute values), and the decoder is trained to reconstruct x given (z, y). The attribute of interest for this network was $Emotion - NoEmotion$.

As mentioned in [8], one of the principal features of this kind of networks is it's ability to learn a disentangled latent space with explicit control on the attributes of interest. In the implemented network presented in this section, in addition to a reconstruction loss two additional classification losses were used for its training: a multiclass cross entropy loss to enforce the disentanglement in the encoder's latent space (based on the type of emotion, i.e. "sadness", "happiness", etc...) and a Binary cross entropy loss ($Emotion - NoEmotion$) to construct an adversarial loss on the level of the latent representation.

Let z be defined as the output of the encoder:

$$z = \theta(x)_{enc}$$

Binary Cross entropy : The probability that an image represents an emotion other than neutral.

$$P_{BC} = -(y \log(z) + (1 - y) \log(1 - z))$$

Multiclass Cross entropy : Classify the images' type of emotion to enhance latent space disentanglement

$$P_{MC} = -y \log(z)$$

Adversarial Loss : L2 as reconstruction Loss, binomial classification as adversarial loss. During experiments λ value was set to 0.1

$$AL = (\theta(z)_{dec} - x)^2 - \lambda \log(1 - P_{BC})$$

Network Structure : Let TC be a Transpose Convolution (alias: Deconvolution) - Relu layer with m kernels of size n and strides s. The Encoder (enc) architecture is the same as the one used for the TSNE 2D-visualization (section 2.1). The Decoder architecture is:

$$F_{8192} - TC_{18,3,2} - TC_{12,3,2} - TC_{6,3,2} - TC_{1,1,1} - F_{16384}$$

The network was trained using Keras's Adam Optimizer.

2.4 Semi-supervised Beta-VAE Architecture

For this architecture, the idea of β-VAEs [5] has been used, to focus on the latent representation of emotions as a concept. To do this, an unsupervised β-VAE is not enough, since many image features are more salient in the dataset, as AUs are. This is the case of intensity, brightness, or contrast in the image. These concepts are easier for a β-VAE to learn. Thus, a semi-supervised β-VAE was created, where the latent space is used directly as a classification layer to predict

the label of each image.
The latent space of the VAE is still being represented as a normal distribution, but by uisng a softmax layer for the classification, the distribution of the corresponding label is pushed away from the standard mean of 0, and towards one.

The final loss of the model is optimized by minimizing three different losses:

Reconstruction Loss This is the binary cross entropy loss, comparing the input image, and the image generated by the VAE:

$$RL = -(y \log(p) + (1-y)\log(1-p))$$

Latent Loss The KL divergence of the latent variables. This is the standard loss used for VAEs' latent space:

$$LL = -\frac{1}{2}(1 + 2\log(\sigma) - \sigma^2 - \mu^2)$$

Classification Loss The softmax crossentropy loss of the latent space:

$$CL = -(y \log(\text{softmax}(z)) + (1-y)\log(1-\text{softmax}(z)))$$

The three losses are each multiplied by three coefficients: $\alpha, \beta, \text{and} \gamma$, respectively, and summed up to generate the total loss:

$$\mathcal{L} = \alpha \cdot RL + \beta \cdot LL + \gamma \cdot CL$$

This way the contribution of each loss can be balanced to optimize the results for what we want from the model. If β and γ are zero, we have a normal Auto Encoder. If γ is zero, the result is a traditional β-VAE. If β is zero the latent space is no longer represented as a probability distribution, but a sampling process from a normal distribution is still being done during the encoding. This is optimized using tensorflow's ADAM optimizer implementation. The architecture of this network is show in figure 5.

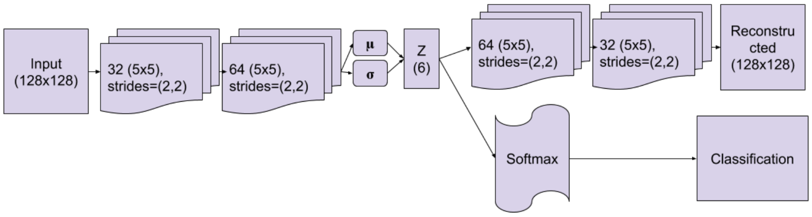

Fig. 5. Architecture of classifying VAE

Considering this architecture, it is expected that images with a neutral facial expression fall on a point close to the origin of the 7-dimensional latent space. An image with the expression of an emotion, should then be pushed along the axis used to represent the specific emotion being shown, but remain close to zero in all other dimensions.

3 Results

3.1 Siamese Architecture

To use the architecture depicted in figure 4 the comparator network has to be pretrained in order to do useful comparisons between the reconstructed image and an arbitrary face. To train the siamese network a second dataset has been constructed, consisting of image pairs, that have been labeled 1 if the emotions match and 0 otherwise. This dataset not only includes raw images, but also averaged images of a set of faces that have one emotion. This was done to simulate the output of the autoencoder.

However, the pretraining of the siamese network didn't succeed. Over a wide variety of tested architectures the network didn't generalize at all. Even as the training loss was steadily decreasing the performance on the test set stayed on 50% accuracy.

Without a functioning comparator network the full architecture has never been tested.

3.2 2D Latent Space Visualization

In figure 6, the clusters obtained by the TSNE technique are well defined with just a few mismatches. It is interesting to note that in this representation each of the emotions seems to be spatially located on the contrary side of the said emotion e.g. the cluster of images corresponding to sadness is located in the opposite side of happiness, etc. while all the emotions labeled as neutral are located at the center of all the emotions suggesting that the structure of a neutral face (in terms of features of convolutional layers) is closer to the structure of all other motions.

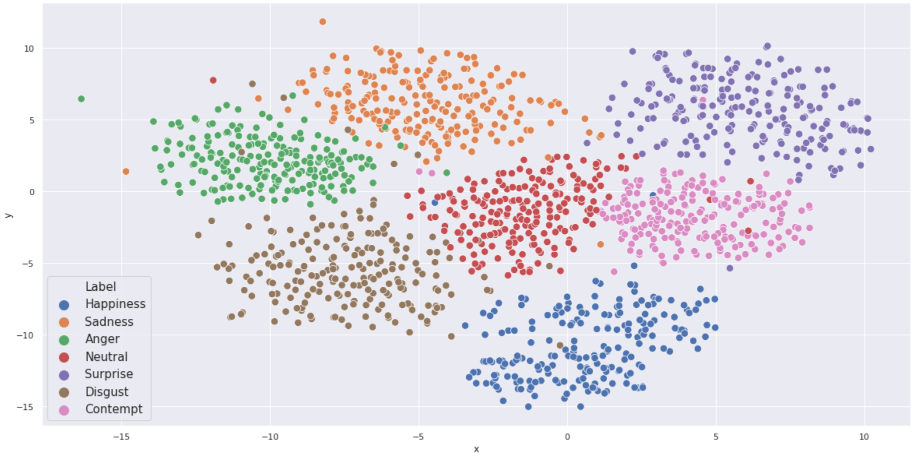

Fig. 6. 2D- Representation of the Image's Latent Space

3.3 Fader Network

As mentioned before, one of the greatest features behind this network is to have a disentangled latent space where each latent variable corresponds to one emotion (i.e. sampling only from that latent variable, the network should reconstruct only that emotion).

This disentanglement can be observed in figure 7. To create the plot, a set of images corresponding to a single emotion were fed into the network and the activation values generated by the encoder (latent space) were stored. It can be seen that for each different emotion the distribution of the activations was higher for a specific latent variable (exemplified by a different color) i.e. displaced to a higher positive values than the other latent variables. It can also be noted that for some cases, one emotion also disentangled another one, for example in figure 7*a*), images with emotion tag of "Disgust" were fed and two disentanglements occurred: the one in the positive quadrant belonging to the emotion fed and one in the negative quadrant belonging to the contrary emotion found in the TSNE visualization (figure 6).

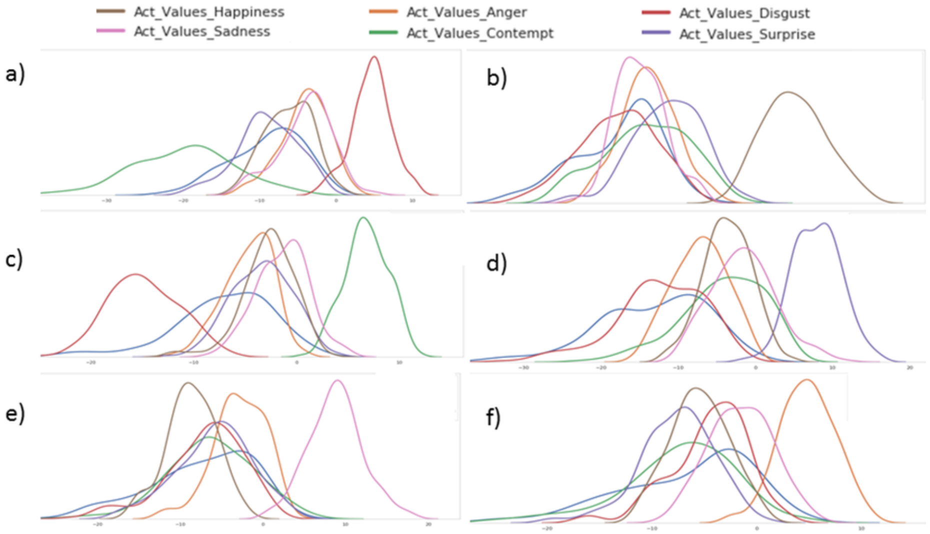

Fig. 7. Disentangled Latent Variables using a Fader Network.

To reflect the power of the Fader Network's Loss function, three small experiments were performed varying the grade of complexity of it: 1) Single L2 reconstruction loss, 2) L2 + multiclass loss, 3) L2 + multiclass loss + binomial adversarial loss. These results can be seen in figure 8. In order to find the optimal values to sample for the latent variables, after training a network, the same procedure to generate figure 7 was performed. Then six equally spaced values from within the maximum and minimum values found for the latent variable of interest were fed to the decoder (maintaining the remaining latent variables constant) in order to create images ”sampled” from the latent variable (figure 8). When the network was set to train on condition 1), no disentanglement occurred and the sampling from a latent variable produced images that weren't necessarily belonging to an emotion, instead random areas of the faces were changed at a different values.When the network was set to train on condition 2), a disentanglement occurred, but the network only remembered the main attributes of such emotion (figure 8, middle row). When trained on condition 3), all the emotions started with a neutral face, and when a sampling of the latent variable was performed, it was observed the change of the expresion from neutrality to such emotion. It should be noted though, that the reason of this is due the binomial loss trying to predict whether an image belonged to a neutral expression or not; if instead of a neutral expression, the binomial loss was trained to identify an emotion like happiness to the others, what we would observe should be the change from a a happiness face to the other emotions.

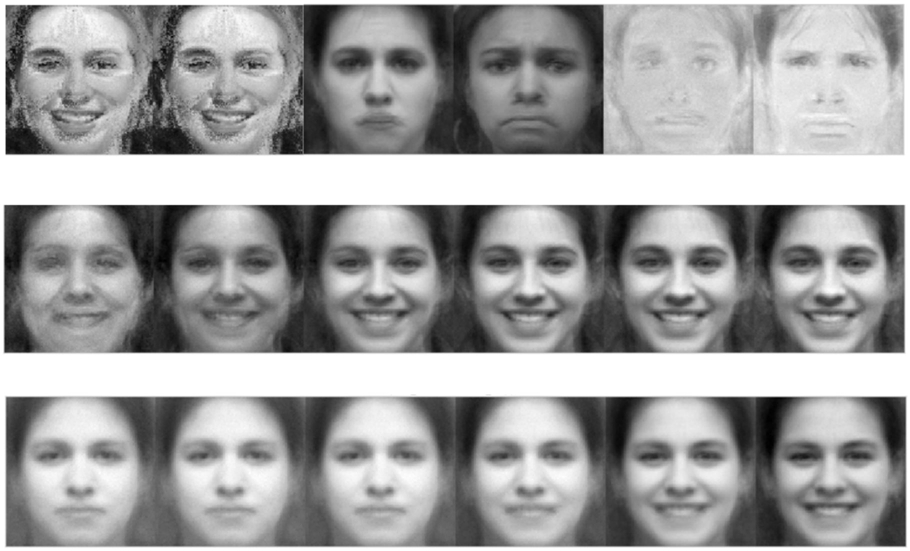

Fig. 8. Reconstructed Images Obtained from sampling a single latent variable. Each row represent a network trained using a different objective function. First Row: Simple L2 reconstruction loss in the encoder (no disentanglement), Second Row: L2 reconstruction loss and Multiclass loss on decoder (disentanglement, no control of features on latent space), Third Row: L2 loss with adversarial binomial loss and Multiclass loss on decoder (disentanglement and control over latent space.

3.4 Semi-supervised VAE

For the training of the semi-supervised VAE, the three losses were balanced as follows: $\alpha = 1, \beta = 2, \gamma = 10$. Thus giving a much higher weight for the latent classification of the emotions. As a consequence, the network is not very good at reconstructing photo-realistic images of faces. Instead faces look as an approximate average of the individuals on the dataset. An example can be seen on figure 9.

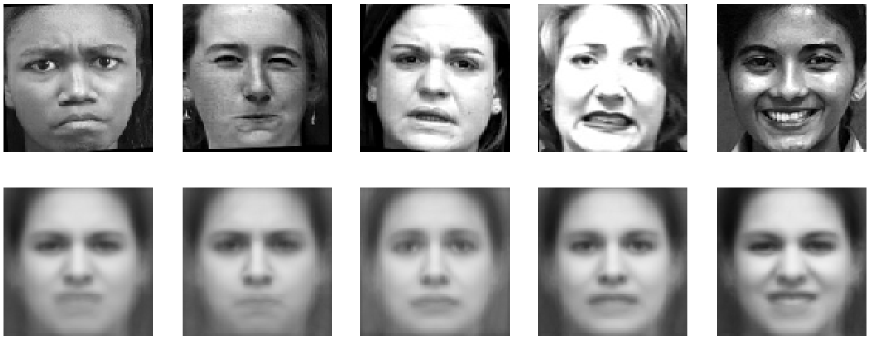

Fig. 9. Reconstruction of original Images with Semi-supervised VAE

The distribution of the latent variables is being learned as expected. A plot of the distributions can be observed in figure 10.

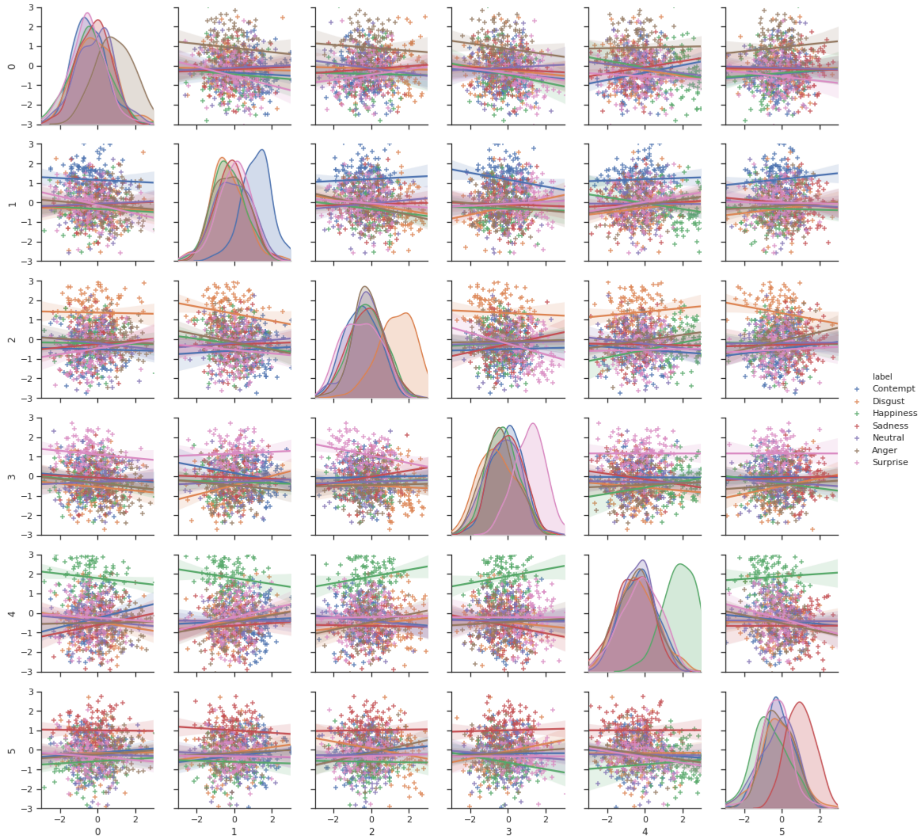

Fig. 10. Distribution of latent variables in Semi-supervised VAE

As a consequence of the network's focus on the expression of emotions, marked by the labels of the classification loss, and not on the individual characteristics of images. The reconstructed images contain a characteristic-neutral face, in matter of gender, age, race, or skin-color, but very clearly represent the AUs to every emotion.

4 Discussion

In figure 10 we can observe that the variable that is easiest to separate is happiness. In its paper, Lucey et al. [10] explain how some oft he facial expressions in the CK+ were simulated, while some others were Candid, specifically smiles. We found no label for this in the dataset, but believe this might be a reason for such a difference in the extraction of this characteristic.

As a consequence of representing emotions in a latent space with a normal distribution, and classifying emotions based only on the positive tail of the distribution we obtain what the network has learned to be the ”opposite” to every basic emotion. Although this is the abstraction of characteristics learned from faces expressing emotions, we think this might be an insight into the specific characteristics that have been developed though out the process of evolution and their relationships.
The best example we found for this phenomenon is the interaction between the expressions of joy and sadness. On figure 11 we present the reconstructed faces generated by sampling different points along the axis used to represent sadness. This image shows how towards the positive side of the distribution, the AUs of the emotion are activated on the reconstructed image, as expected. To the negative side we observe that the AUs of joy are activated.

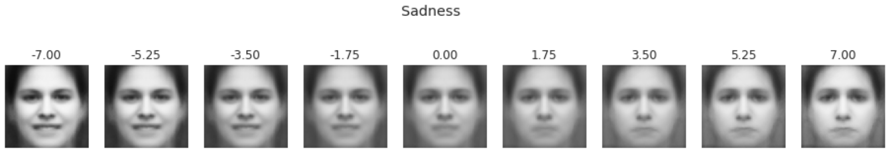

Fig. 11. Faces reconstructed by sampling the latent space along the axis of ”sadness”

We think this could be a consequence of the latent space representation, and the difference between the expression of emotions, represented in the latent space as a longer, or shorter distance. This can be explained in the context of Evolutionary Biology, since the expression and recognition of emotions is a key process in social animals.

Bibliography

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] P. Burkert, F. Trier, M. Z. Afzal, A. Dengel, and M. Liwicki. Dexpression: Deep convolutional neural network for expression recognition. *arXiv preprint arXiv:1509.05371*, 2015.

[3] P. Ekman. Are there basic emotions? 1992.

[4] S. Ghosh, E. Laksana, S. Scherer, and L.-P. Morency. A multi-label convolutional neural network approach to cross-domain action unit detection. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 609–615. IEEE, 2015.

[5] I. Higgins, L. Matthey, X. Glorot, A. Pal, B. Uria, C. Blundell, S. Mohamed, and A. Lerchner. Early visual concept learning with unsupervised deep learning. *arXiv preprint arXiv:1606.05579*, 2016.

[6] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim. Joint fine-tuning in deep neural networks for facial expression recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2983–2991, 2015.

[7] P. Khorrami, T. Paine, and T. Huang. Do deep neural networks learn facial action units when doing expression recognition? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 19–27, 2015.

[8] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato. Fader networks: Manipulating images by sliding attributes. *CoRR*, abs/1706.00409, 2017. URL http://arxiv.org/abs/1706.00409.

[9] A. T. Lopes, E. de Aguiar, A. F. De Souza, and T. Oliveira-Santos. Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. *Pattern Recognition*, 61:610–628, 2017.

[10] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 94–101. IEEE, 2010.

[11] Z. Meng, P. Liu, J. Cai, S. Han, and Y. Tong. Identity-aware convolutional neural network for facial expression recognition. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 558–565. IEEE, 2017.

[12] A. Mollahosseini, D. Chan, and M. H. Mahoor. Going deeper in facial expression recognition using deep neural networks. In *2016 IEEE Winter conference on applications of computer vision (WACV)*, pages 1–10. IEEE, 2016.

[13] F. Qiao, N. Yao, Z. Jiao, Z. Li, H. Chen, and H. Wang. Geometry-contrastive gan for facial expression transfer. *arXiv preprint arXiv:1802.01822*, 2018.

[14] K. Shan, J. Guo, W. You, D. Lu, and R. Bie. Automatic facial expression recognition based on a deep convolutional-neural-network structure. In *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 123–128. IEEE, 2017.

[15] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[16] Y. Zhou and B. E. Shi. Photorealistic facial expression synthesis by the conditional difference adversarial autoencoder. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 370–376. IEEE, 2017.