Abstract

We present a new end-to-end network architecture for facial expression recognition with an attention model. It focuses attention in the human face and uses a Gaussian space representation for expression recognition. We devise this architecture based on two fundamental complementary components: (1) facial image correction and attention and (2) facial expression representation and classification. The first component uses an encoder-decoder style network and a convolutional feature extractor that are pixel-wise multiplied to obtain a feature attention map. The second component is responsible for obtaining an embedded representation and classification of the facial expression. We propose a loss function that creates a Gaussian structure on the representation space. To demonstrate the proposed method, we create two larger and more comprehensive synthetic datasets using the traditional BU3DFE and CK+ facial datasets. We compared results with the PreActResNet18 baseline. Our experiments on these datasets have shown the superiority of our approach in recognizing facial expressions.

1. Introduction

Human beings are able to express and recognize emotions as a way to communicate an inner state. Facial expression is the main form to convey this information and its understanding has transformed the treatment of emotions by the scientific community. Traditionally, scientists assumed that people have internal mechanisms comprising a small set of emotional reactions (e.g. happiness, anger, sadness, fear, disgust) that are measurable and objective. Understanding these mental states from facial and body cues is a fundamental human trait, and such aptitude is vital in our daily communications and social interactions. In fields such as Human-Computer Interaction (HCI), Neuroscience, and Computer Vision, scientists have conducted extensive research to understand human emotions. Some of these studies aspire to create computers that can understand and respond to human emotions and to our general behavior, potentially leading to seamless beneficial interactions between humans and computers [29, 12]. Our work aims to contribute to this effort, more specifically in the area of Facial Expression Recognition, or FER for short.

Deep Convolutional Neural Networks (CNN) have recently shown excellent performance in a wide variety of image classification tasks [17, 32, 35, 34]. The careful design of local to global feature learning with a convolution, pooling, and layered architecture produces a rich visual representation, making CNN a powerful tool for facial expression recognition [18]. Research challenges such as the Emotion Recognition in the Wild (EmotiW) series\textsuperscript{1} and
Kaggle’s Facial Expression Recognition Challenge\(^2\) suggest the growing interest of the community in the use of deep learning for the solution of this problem.

Recent developments for the facial expression recognition problem consider processing the entire image regardless of the face crop location within the image [42]. Such developments bring in extraneous artifacts, including noise, which might be harmful for classification as well as incur in unnecessary additional computational cost. This is problematic as the *minutiae* that characterizes facial expressions can be affected by elements such as hair, jewelry, and other environmental objects not defining the actual face and as part of the image background. Some methods use heuristics to decrease the searching size of the facial regions to avoid considering objects beyond the face itself. Such approaches contrast to our understanding of the human visual perception, which quickly parses the field of view, discards irrelevant information, and then focus the main processing on a specific target region of interest – the so-called visual attention mechanism [14, 39]. Our approach tries to mimic this behavior as it aims to suppress the contribution of surrounding deterrent elements by segmenting the face in the image and thus concentrating recognition solely on facial regions. Figure 1 illustrates how the attention mechanism works in a typical scene.

Attention mechanisms have recently been explored in a wide variety of contexts [38, 15], often providing new capabilities for known neural networks models [7, 8, 4]. While they improve efficiency [26] and performance on state-of-the-art machine learning benchmarks [38], their computational architecture is much simpler than those comprising the mechanisms in the human visual cortex [2]. Attention has also been long studied by neuroscientists [36], who believe it is crucial for visual perception and cognition [1] as it is inherently tied to the architecture of the visual cortex and can affect its information.

Our contributions are summarized as follows: (1) We propose a CNN-based method using attention to jointly solve for representation and classification in FER problems; (2) We introduce a new dual-branch network to extract an attention map which in turn improves the learning of kernels specific to facial expression; (3) A new loss function is formulated for obtaining a facial manifold represented as a Gaussian Mixture Model; and (4) We offer a new synthetic generator to render face expressions which significantly augments training data and consequently improves the overall classification.

## 2. Related Works

Liu et al. [22] introduced a facial expression recognition framework using 3DCNN together with deformable action parts constraints to jointly localize facial action parts and learn part-based representations for expression recognition. Liu et al. [21] followed by including the pre-trained Caffe CNN models to extract image-level features.

In 2015, Yu and Zhang [43] achieved state-of-the-art results in the EmotiW challenge using CNNs. They used an ensemble of CNNs each with five convolutional layers and showed that randomly perturbing the input images yielded a 2-3% boost in accuracy. Specifically, they applied transformations to the input images at training time. At testing time, their model generated predictions for multiple perturbations of each test example and voted on the class label to produce a final answer. They used stochastic pooling [6] rather than max pooling due to its good performance on limited training data. Mollahosseini et al. [27] have also obtained state of the art results with a network consisting of two convolutional layers, max-pooling, and four inception layers, the latter introduced by GoogLeNet.

Another recent method the De-expression Residue Learning (DeRL) [40], trains a generative model to create a corresponding neutral face image for any input face. Then, another model is trained to learn the deposition (or residue) that remains in the intermediate layers of the generative model for the classification of facial expression.

Zhang et al. [46] proposed an end-to-end learning model based on Generative Adversarial Network (GAN). The architecture incorporates a generator, two discriminators, and a classifier. The GAN is used for generating multiples variation of one image, which is used to train a convolutional neural network.

## 3. Methodology

In this section, we describe our contributions in designing a new network architecture, in the formulation of the loss function used for training, and in the method to generate synthetic data.

### 3.1. Network architecture

Given a facial expression image \(I\), our objective is to obtain a good representation and classification of \(I\). The proposed model, Facial Expression Recognition with Attention Net (FERAtt), is based on the dual-branch architecture [9, 19, 28, 48] and consists of four major modules: (i) an attention module \(G_{\text{att}}\) to extract the attention feature map, (ii) a feature extraction module \(G_{f1}\) to obtain essential features from the input image \(I\), (iii) a reconstruction module \(G_{\text{rec}}\) to estimate a proper attention image \(I_{\text{att}}\), and (iv) a representation module \(G_{\text{rep}}\) that is responsible for the representation and classification of the facial expression image. An illustration of the proposed model is shown in Figure 2.

**Attention module.** We use an encoder-decoder style network, which has been shown to produce good results for many generative [33, 48] and segmentation tasks [31]. In
Figure 2: **Architecture of FERAtt.** Our model consists of four major modules: attention module $G_{att}$, feature extraction module $G_{ft}$, reconstruction module $G_{rec}$, and classification and representation module $G_{rep}$. The features extracted by $G_{att}$, $G_{ft}$ and $G_{rec}$ are used to create the attention map $I_{att}$ which in turn is fed into $G_{rep}$ to create a representation of the image. Input images $I$ have $128 \times 128$ pixels and are reduced to $32 \times 32$ by an Averaging Pooling layer on the reconstruction module. Classification is thus done on these smaller but richer representations of the original image.

In particular, we choose a variation of the fully convolutional model proposed in [31] for semantic segmentation. Also, we applied four layers in the coder with skip connections and dilation of 2x. The decoder layer is initialized with pre-trained ResNet34 [10] layers. This significantly accelerates the convergence. The output features of the decoder are denoted by $G_{att}$, which is used to determine the attention feature map. This is a probability map that is not the same as a simple segmentation procedure.

**Feature extraction module.** Four ResBlocks [20] were used to extract high-dimensional features for image attention and to maintain spatial information; no pooling or strided convolutional layers were used. We denote the extracted features as $G_{ft}$ – see Figure 3b.

**Reconstruction module.** The reconstruction layer adjusts the attention map to create an enhanced input to the representation module. This module has two convolutional layers, a Relu layer, and an Average Pooling layer which, by design choice, resizes the input image of $128 \times 128$ to $32 \times 32$. This reduced size was chosen for the input of the representation and classification module (PreActivation-ResNet [11]), the image size number we borrowed from the literature to facilitate comparisons. We plan to experiment with other sizes in the future. We denote the feature attention map as $I_{att}$ – see Figure 3d.

**Representation and classification module.** For the representation and classification of facial expressions, we have chosen a Fully Convolutional Network (FCN) of PreActiveResNet [11]. This architecture has shown excellent results when applied on classification tasks. The output of the FCN, vector $z$, is evaluated in a linear layer to obtain a vector $\hat{z} \in \mathbb{R}^d$ with the desired dimensions. $f_{\theta} : \mathbb{R}^D \rightarrow \mathbb{R}^d$, the network function, builds a representation for a sample image $x \in \mathbb{R}^D$, (e.g. $D = 128 \times 128$ pixels) in an embedded space of reduced dimension $\mathbb{R}^d$ (we use $d = 64$ in our experiments). Vector $\hat{z}$ is then evaluated in a regression layer to estimate the probability $p(w | \hat{z})$ for each class $w_j$, $w = [w_1, w_2, \ldots, w_c]$.
3.2. Loss functions

The FERAtt network generates three outputs: a feature attention map \( I_{att} \), a representation vector \( \hat{z} \), and a classification vector \( \hat{w} \). In our training data, each image \( I \) has an associated binary ground truth mask \( I_{mask} \) corresponding to a face in the image and its expression class vector \( w \). We train the network by jointly optimizing the sum of attention, representation, and classification losses:

\[
\min_{\Theta} \{ L_{att}(I_{att}, I \otimes I_{mask}) + L_{rep}(\hat{z}, w) + L_{cls}(\hat{w}, w) \}
\]

where \( \Theta \) represents the collective parameters that need be optimized. We use the pixel-wise MSE (Mean Square Error) loss function for \( L_{att} \), and for \( L_{cls} \) we use the BCE (Binary Cross Entropy) loss function. We propose a new loss function \( L_{rep} \) for the representation, defined below.

3.3. Structured Gaussian Manifold Loss

Let \( S = \{ x_i | x_i \in \mathbb{R}^D \} \) be a collection of i.i.d. samples \( x_i \) we want to classify into \( c \) classes, and let \( w_j \) represent the \( j \)-th class, for \( j = 1, \ldots, c \). The class function \( l(x) = \arg \max p(w|f_\theta(x)) \) returns the class \( w_j \) of sample \( x \) — maximum a posteriori probability estimate — for the neural net function \( f_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^d \) drawn independently according to probability \( p(x|w_j) \) for input \( x \). Suppose we separate \( S \) in an embedded space such that each set \( C_j = \{ x \in S | l(x) = w_j \} \) contains the samples belonging to class \( w_j \). Our goal is to find a Gaussian representation for each \( C_j \) which would allow a clear separation of \( S \) in a reduced space, \( d \ll D \).

We assume that \( p(f_\theta(x)|w_j) \) has a known parametric form, and it is therefore determined uniquely by the value of a parameter vector \( \theta_j \). For example, we might have \( p(f_\theta(x)|w_j) \sim N(\mu_j, \Sigma_j) \), where \( \theta_j = (\mu_j, \Sigma_j) \), for \( N(\ldots) \) the normal distribution with mean \( \mu_j \) and variance \( \Sigma_j \). To show the dependence of \( p(f_\theta(x)|w_j) \) on \( \theta_j \) explicitly, we write \( p(f_\theta(x)|w_j) \) as \( p(f_\theta(x)|w_j, \theta_j) \). Our problem is to use the information provided by the training samples to obtain a good transformation function \( f_\theta(x) \) that generates embedded spaces with a known distribution associated with each category. Then the a posteriori probability \( P(w_j|f_\theta(x)) \) can be computed from \( p(f_\theta(x)|w_j) \) by the Bayes’ formula:

\[
P(w_j|f_\theta(x)) = \frac{p(f_\theta(x)|w_j, \theta_j)}{\sum_i p(f_\theta(x)|w_i, \theta_i)}
\]

In this work, we are using the normal density function for \( p(x|w_j, \theta_j) \). The objective is to generate embedded subspaces with a defined structure. We use Gaussian structures:

\[
p(f_\theta(x)|w_j, \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{D/2}\Sigma_j^{1/2}} \exp(-\frac{1}{2}X^T\Sigma_j^{-1}X)
\]

where \( X = (f_\theta(x) - \mu_j) \). For the case \( \Sigma_j = \sigma^2 I \), where \( I \) is the identity matrix:

\[
p(x|w_j, \mu_j, \sigma_j) = \frac{1}{\sqrt{(2\pi)^\sigma\sigma_j}} \exp(-\frac{||f_\theta(x) - \mu_j||^2}{2\sigma_j^2})
\]

In a supervised problem, we know the a posteriori probability \( P(w_j|x) \) for the input set. From this, we can define our structured loss function as the mean square error between the a posteriori probability of the input set and the a posteriori probability estimated for the embedded space:

\[
L_{rep} = \mathbb{E} \{ ||P(w_j|f_\theta(x_i)) - P(w_j|x_i)||^2 \}
\]

3.4. Synthetic image generator

A limiting problem of currently available face expression datasets for supervised learning is the reduced number of correctly labeled data. We propose a data augmentation strategy to mitigate this problem in the lines of what has been introduced in [5]. Our image renderer \( R \) creates a synthetic larger dataset using real face datasets by making background changes and geometric transformations of face images. The example in Figure 4 shows a synthetic image generated pipeline by combining an example face of the CK+ dataset and a background image.

The generator method is limited to make low-level features that represent small variations in the facial expression space for the classification module. However, it allows creating a good number of examples to train our end-to-end system, having a larger contribution to the attention component. In the future we plan to include high-level features using GAN from the generated masks [13].

The renderer \( R \) adjusts the illumination of the face image so that it is inserted in the scene more realistically. An alpha matte step is applied in the construction of the final composite image of face and background. The luminance channel of the image face model \( I_{face} \) is adjusted by multiplying it by the factor \( \frac{I_r}{I_{face}} \), where \( I_r \) is the luminance of the region that contains the face in the original image.

4. Experiments

We describe here the creation of the dataset used for training our network and its implementation details. We discuss two groups of experimental results: (1) Expression recognition result, to measure the performance of the method regarding the relevance of the attention module and the proposed loss function, and (2) Correction result, to analyze the robustness to noise.
Figure 4: The pipeline of the synthetic image generation. The horizontal alignment of the image (b) is based on the inner points of the eyes (red points in (a)). The face is obtained as the convex hull of the landmarks set (c) and a random transform matrix is generated (d). The face image is projected on the background image (e). A face image and a general cropped background image are combined to generate a composite image (f). By using distinct background images for every face image, we are able to generate a much larger training data set. We create a large quantity of synthetic new images for every face of a database: approximately 9,231 synthetic images are generated for each face in the CK+ database, and 5,000 for the BU-3DFE database. This covers a great variety of possible tones and different backgrounds.

4.1. Datasets

We employ two public facial expression datasets, namely Extended Cohn-Kanade (CK+) [24] and BU-3DFE [41] to evaluate our method. We apply in all experiments person-independent FER scenarios [45]. Subjects in the training set are completely different from the subjects in the test set, i.e., the subjects used for training are not used for testing. The CK+ dataset includes 593 image sequences from 123 subjects. We selected 325 sequences of 118 subjects from this set, which meet the criteria for one of the seven emotions [24]. The selected 325 sequences consist of 45 Angry, 18 Contempt, 25 Fear, 69 Happy, 28 Sadness and 82 Surprise [24] facial expressions. In the neutral face expression case, we selected the first frame of the sequence of 33 random selected subjects. The BU-3DFE dataset is known to be challenging mainly due to a variety of ethnic/racial ancestries and expression intensity [41]. A total of 600 expressive face images (1 intensity x 6 expressions x 100 subjects) and 100 neutral face expression images, one for each subject, were used [41].

We employed our renderer $R$ to augment training data for the neural network. $R$ uses a facial expression dataset (we use BU-3DFE and CK+, which were segmented to obtain face masks) and a dataset of background images chosen from the COCO dataset. Figure 5 shows some examples of images generated by the renderer on the BU-3DFE dataset.

4.2. Implementation and training details

In all experiments, we considered the neural network architecture PreActResNet18 for the classification and representation processes. We adopted two approaches: (1) a model with attention and classification, FERAtt+Cls, and (2) a model with attention, classification, and representation, FERAtt+Rep+Cls. These models were compared with the classification results. For representation, the last convolutional layer of PreActResNet is evaluated by a linear layer to generate a vector of selected size. We have opted for 64 dimensions for the representation vector $\hat{z}$.

All models were trained on Nvidia GPUs (P100, K80, Titan XP) using PyTorch$^3$ for 60 epochs for the training set with 200 examples per mini batch and employing Adam optimizer. Face images were rescaled to $32 \times 32$ pixels. The code for the FERAtt is available in a public repository$^4$.

$^3$http://pytorch.org/
$^4$https://github.com/pedrodiamel/ferattention
Table 1: Classification results for the Synthetic/Real BU-3DFE database (6 expression + neutral) and CK+ database (7 expression classes + neutral). Baseline: PreActResNet18[11], Acc.: Accuracy, Prec.: Precision, Rec.: Recall, F1: F1 measurement. Leave-10-subjects-out cross-validation is used for all experiments.

<table>
<thead>
<tr>
<th>Database</th>
<th>Method</th>
<th>Synthetic</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc.</td>
<td>Prec.</td>
<td>Rec.</td>
</tr>
<tr>
<td>BU-3DFE</td>
<td>Baseline</td>
<td>69.37</td>
<td>71.48</td>
</tr>
<tr>
<td></td>
<td>±2.84</td>
<td>±1.46</td>
<td>±2.76</td>
</tr>
<tr>
<td></td>
<td>75.15</td>
<td>77.34</td>
<td>75.45</td>
</tr>
<tr>
<td></td>
<td>±3.13</td>
<td>±1.40</td>
<td>±2.57</td>
</tr>
<tr>
<td></td>
<td>77.90</td>
<td>79.58</td>
<td>78.05</td>
</tr>
<tr>
<td></td>
<td>±2.59</td>
<td>±1.77</td>
<td>±2.34</td>
</tr>
<tr>
<td>CK+</td>
<td>Baseline</td>
<td>77.63</td>
<td>68.42</td>
</tr>
<tr>
<td></td>
<td>±2.11</td>
<td>±2.97</td>
<td>±1.91</td>
</tr>
<tr>
<td></td>
<td>84.60</td>
<td>74.94</td>
<td>76.30</td>
</tr>
<tr>
<td></td>
<td>±0.93</td>
<td>±0.38</td>
<td>±1.19</td>
</tr>
<tr>
<td></td>
<td>85.15</td>
<td>74.68</td>
<td>77.45</td>
</tr>
<tr>
<td></td>
<td>±1.07</td>
<td>±1.37</td>
<td>±0.55</td>
</tr>
</tbody>
</table>

Table 2: Comparison of the average recognition accuracy with state-of-the-art FER methods for the BU-3DFE database. NE: number of expressions, †: six basic expressions + neutral class. Leave-10-subjects-out cross-validation is used for all methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
<th>NE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lopes[23]</td>
<td>72.89</td>
<td>7†</td>
</tr>
<tr>
<td>Jampour[16]</td>
<td>78.64</td>
<td>7†</td>
</tr>
<tr>
<td>Zhang[47]</td>
<td>80.10</td>
<td>7†</td>
</tr>
<tr>
<td>Zhang[46]</td>
<td>80.95</td>
<td>7†</td>
</tr>
<tr>
<td>Our</td>
<td>82.11</td>
<td>7†</td>
</tr>
</tbody>
</table>
Table 3: Comparison of the average recognition accuracy with state-of-the-art FER methods for the CK+ database. NE: number of expressions, †: six basic expressions + neutral class and contempt class, ‡: six basic expressions + contempt class (neutral is excluded). ∗: the value in parentheses is the mean accuracy, which is calculated with the confusion matrix given by the authors. Leave-10-subjects-out cross-validation is used for all methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy*</th>
<th>NE</th>
</tr>
</thead>
<tbody>
<tr>
<td>IACNN[25]</td>
<td>95.37</td>
<td>7‡</td>
</tr>
<tr>
<td>DSAE[44]</td>
<td>95.79 (93.78)</td>
<td>7‡</td>
</tr>
<tr>
<td>DeRL[40]</td>
<td>97.30 (96.57)</td>
<td>7‡</td>
</tr>
<tr>
<td>Our</td>
<td>97.50</td>
<td>7‡</td>
</tr>
<tr>
<td>DSAE[44]</td>
<td>89.84 (86.82)</td>
<td>8†</td>
</tr>
<tr>
<td>Our</td>
<td>90.30</td>
<td>8†</td>
</tr>
</tbody>
</table>

Figure 6: Barnes-Hut t-SNE visualization [37] of the Gaussian Structured loss for the Real CK+ database. Each color represents one of the eight emotions including neutral. Observe the clear separation of classes with a really small amount of misclassified neutral images. The distance between classes presented in the figure shows the expected. For example, happiness and anger are far apart while neutral appears approximately halfway between them.

Figure 7: Top-5 images retrieved using FERAtt+Rep+Cls for the Real CK+ database embedded vectors.

Figure 8: Barnes-Hut t-SNE visualization [37] of the Gaussian Structured loss for the Real CK+ database. Each color represents one of the eight emotions including neutral. Observe the clear separation of classes with a really small amount of misclassified neutral images. The distance between classes presented in the figure shows the expected. For example, happiness and anger are far apart while neutral appears approximately halfway between them.

4.4. Robustness to noise

The objective of this set of experiments is to demonstrate the robustness of our method to the presence of image noise when compared to the baseline architecture PreActResNet18.

Protocol. To carry out this experiment, the Baseline, FERAtt+Class, and FERAtt+Rep+Class models were trained on the Synthetic CK+ dataset. Each of these models was readjusted with increasing noise in the training set ($\sigma \in [0.05, 0.30]$). We maintained the parameters in the training for fine-tuning and used the real database CK+, so that 2000 images were generated for the synthetic dataset for test.

Results. One of the advantages of the proposed approach is that we can evaluate the robustness of the method under different noise levels by visually assessing the changes in the attention map $I_{att}$. Figure 8 shows the attention maps
for an image for white zero mean Gaussian noise levels $\sigma = [0.01, 0.05, 0.07, 0.09, 0.1, 0.2, 0.3]$. We observe that our network is quite robust to noise for the range of 0.01 to 0.1 and maintains a distribution of homogeneous intensity values. This aspect is beneficial to the subsequent performance of the classification module. Figures 9 and 10 present classification accuracy results of the evaluated models in the Real CK+ dataset and for 2000 synthetic images. The proposed method FERAtt+CLs+Rep provides the best classification in both cases.

5. Conclusions

In this work, we present a new end-to-end neural network architecture with an attention model for facial expression recognition. We create a generator of synthetic images which is used for training our models. The results show that, for these experimental conditions, the attention module improves the system classification performance in comparison to other methods from the state-of-the-art. The loss function presented works as a regularization method on the embedded space. For future work, we plan to incorporate a transformer component in the architecture for automatic alignment of the face. We want to train the network for extreme condition such as dark light and occlusion.

6. Acknowledgment

The authors thanks the financial support from the Brazilian funding agency FACEPE and CETENE for usage of the computational facility.
References


[27] Ali Mollahosseini, David Chan, and Mohammad H Ma`hoor. Going deeper in facial expression recognition using


