Proceedings Paper:
Ruiz-Garcia, A, Elshaw, M, Altahhan, A orcid.org/0000-0003-1133-7744 et al. (1 more author) (2017) Stacked deep convolutional auto-encoders for emotion recognition from facial expressions. In: 2017 International Joint Conference on Neural Networks (IJCNN). International Joint Conference on Neural Networks (IJCNN), 14-19 May 2017, Anchorage, Alaska, USA. IEEE , pp. 1586-1593. ISBN 978-1-5090-6183-9

https://doi.org/10.1109/IJCNN.2017.7966040

Stacked Deep Convolutional Auto-Encoders for Emotion Recognition from Facial Expressions

Ariel Ruiz-Garcia, Mark Elshaw, Abdulrahman Altahhan, Vasile Palade
School of Computing, Electronics, and Mathematics
Faculty of Engineering, Environnment, and Computing
Coventry University
Coventry, United Kingdom
Email: ariel.ruizgarcia@coventry.ac.uk

***Abstract*—Emotion recognition is critical for everyday living and is essential for meaningful interaction. If we are to progress towards human and machine interaction that is engaging the human user, the machine should be able to recognise the emotional state of the user. Deep Convolutional Neural Networks (CNN) have proven to be efficient in emotion recognition problems. The good degree of performance achieved by these classifiers can be attributed to their ability to self-learn a down-sampled feature vector that retains spatial information through filter kernels in Convolutional layers. Given the view that randomized initialization of weights can lead to convergence in non-optimal local minima, in this paper we explore the impact of training the initial weights in an unsupervised manner. We study the effect of pre-training a Deep CNN as a Stacked Convolutional Auto-Encoder (SCAE) in a greedy layer-wise unsupervised fashion for emotion recognition using facial expression images. When trained with randomly initialized weights, our CNN emotion recognition model achieves a performance rate of 91.16% on the Karolinska Directed Emotional Faces (KDEF) dataset. In contrast, when each layer of the model, including the hidden layer, is pre-trained as an Auto-Encoder, the performance increases to 92.52%. Pre-training our CNN as a SCAE also reduces training time marginally. The emotion recognition model developed in this work will form the basis of real-time empathic robot system.**

I. Introduction

Emotion recognition usually involves analizing a person's facial expressions, body language, or speech signals and classifying them as a specific emotion. It has been stated that emotion recognition is critical for everyday living and is essential for interaction with others [1], [2]. If we are to progress towards human and machine interaction, such as the interaction with a social robot, in a manner that is engaging for the human, the machine should be able to recognise and respond to the emotional state of the user [3].

In the work discussed in this paper, we compare the performance of a Deep Convolution Network when trained with a randomly initialized set of weights and when pre-trained as a Stacked Convolutional Auto-Encoder to classify facial expression images from the KDEF [4] dataset. Emotion recognition is performed through facial expressions images due the evident advantage offered over other forms of emotion recognition: in unconstrained environments, it can be difficult to isolate the speech signals from a particular subject, especially in a crowed environment. Similarly, the difficulty of capturing body language can be greater compared to obtaining an image of someone's face.

Compared to traditional feed forward networks, CNN have the ability to autonomously create a feature vector of salient features while retaining spatial information, such as shapes, through filter kernels. In the case of emotion recognition this is particularly important considering that classification of a given emotion depends predominately upon the shape of facial features such as the eyes, mouth, and eyebrows. However, due to the high complexity of facial expression images, CNN models often require a high number of Convolutional layers in order to extract a good set of features that best represent the data. The disadvantage of increased network depth is the complexity of the network and training time which can grow significantly with each additional layer. Moreover, increased network complexity often leads to a failure in finding the optimum network configuration, and such limitation might not allow the best possible emotion recognition capability.

Finding the right initialization parameters for deep networks in supervised learning is always challenging and requires a large number of attempts to move towards the best possible recognition performance. In the case of deep networks, including Deep CNN, this is very inefficient due to the lengthy training time required for each trial. Bengio [5] suggests that random initialization of a network can lead to convergence on local minima, and thus result in poor classification. To avoid this difficulty associated with random initialization, one can employ Auto-Encoders to pre-train each layer of a CNN in a greedy layer-wise unsupervised manner. This allows for an initialization of filter kernels in a CNN close to a good local minimum [5], which leads to improved feature extraction and classification performance.

In this work we look at the development of an emotion recognition model with the right trade-off balance between depth and classification performance. Since very deep networks are often not suitable for applications in which response delay has to be kept to a minimum, for example when a robot is interacting with a user the robot needs to avoid delayed responses to maintain interaction, we attempt to build an emotion recognition model with a reduced number of Deep Learning (DL) layers that produces similar classification performance to very deep networks. Taking into account that

this model is intended to be incorporated within a social robot for real-time emotion recognition, we study the effect of reducing the number of Convolutional layers and the effect of pre-training a CNN as a Stacked Convolutional Auto-Encoder (SCAE). Furthermore, we analyse the effect of batch normalization [6] on Convolutional and fully connected layers during pre-training and fine-tuning. These criteria are applied to a Deep CNN with four Convolutional layers.

The structure of this paper is as follows: Section II introduces existing state-of-the-art emotion recognition approaches based on DL and literature on Auto-Encoders used as a pre-training method. Section III describes the dataset used in our experiments along with a detailed description of our experiments. Section IV presents results and a discussion of these. Section V describes future direction of our work followed by a list of references

II. Previous Work

Due to the inherent non-linearity of deep networks, empirical training methods such as Stochastic Gradient Decent (SGD) may fail if the parameters are not initialized appropriately or if the network topology is not ideal. Imprecise network configurations can lead to large or small gradients and problems obtaining a set of weights that provide optimal generalization of the training data. Where the topology or parameters of the network are not ideal it often requires a lengthy training process, particularly for very deep models. Random weight initialization is often the preferred choice amongst researchers and is intended to provide the network with a weight distribution that does not favor any particular class. However, recent studies [5], [7] show that random initialization of weights can lead to convergence in local minima that are far away from an optimal global solution. As a result, a number of initialization methods targeting this issue have been devised in recent years. This exploration of previous research discusses prominent initialization methods including pre-training of networks and methods designed to eliminate the need of fixed initialization, such as batch normalization. Furthermore, this section explores existing emotion recognition models which employ deep learning for feature extraction and classification.

A. Weight Initialization Methods

Initializing a network with the right set of weights can lead to good generalization of the training data. However, it is often difficult to find this optimal initial set of weights and so this is an area of interest with regards to DL networks. Krähenbühl et al. [8] have introduced a data-dependent initialization method for CNN which forces all the weights within a layer to train at a similar rate. According to the authors, when combined with pre-training methods, their data-dependent initialization method outperforms existing methods and avoids vanishing or exploding gradients. Remero et al. [9] proposed using a trained teacher network to train a student network that has greater depth but is thinner and has less parameters. This approach uses the intermediate representations learned by the teacher network to improve training of the student network, which outperforms the teacher network and generalizes faster. Romero et al. refer to this approach as FitNets and essentially consists of compressing a deep and wide network into a deeper but thinner one.

Srivastava et al. [10] have introduced the concept of Highway Networks which allows the training of very deep networks with hundreds of layers using SGD. Highway networks are inspired by Long Short Term Memory of recurrent networks and regulate information flow through gating units, allowing information flow across layers without debilitation. Although the authors propose a novel way to efficiently train very deep architectures, the application of very deep networks to real-life problems seems unpromising considering the lengthy training process and computational power required for each network.

Mishkin and Matas [11] propose an initialization method, which they refer to as layer-sequential unit-variance (LSUV), consisting of initializing Convolution layers with orthonormal matrices and then normalizing the variance of the output of each layer in the network, including non-Convolutional layers, to be equal to one. The authors argue that LSUV outperforms more complex methods such as Highway networks and FitNets and has an advantage of working with a number of activation functions

B. Deep CNN Normalization

Since Glorot et al. [12] showed that rectifying neurons can be used to train networks and obtain similar or better results than deep models that employ unsupervised pre-training methods. Rectified Linear Unit (ReLU) layers, along with Max Pooling, have become essential components of Convolutional networks. Most, if not all, recent Deep CNN architectures use rectifier neurons to normalize the output of Convolutional layers. Variations of ReLU layers have been proposed by: He et al. [13], in the form of the Parametric Rectified Linear Unit (PReLU); Maas et al. [14] who introduced leaky ReLU; and Xu et al. [15] who proposed the Randomized leaky ReLU.

Further improvements to Convolutional networks were proposed by Krizhevsky et al. [16] who introduced the Local Response Normalization (LRN) layers for CNN using ReLU layers in order to allow the detection of high-frequency features with big neuron responses. Other optimization methods include the use of Dropout, Learning Decay, along with Weight Decay. In the work presented in this paper, we use Learning Decay as discussed in Section III.

One of the most recent improvements to deep networks is Batch Normalization (BN) which normalises the distribution of each input feature at every layer [6]. BN is rapidly becoming the main approach in deep networks to accelerate training and improve classification performance given that it significantly improves training time and in some instances boosts classification performance. Furthermore, according to Ioffe and Szegedy [6], BN eliminates the need for Local Response Normalization and Dropout. The main advantage of BN seems to be faster training times which also lead to larger learning rates and faster learning rate decays. Given the

benefits offered, our SCAE emotion recognizer incorporates BN for both Convolutional and fully connected layers as explained in Section III.

C. Unsupervised Pre-Training

According to Erhan et al. [17] pre-training deep networks in an unsupervised fashion guides the learning towards better minima and results in better generalization of training data. Restricted Boltzmann Machines (RBM) have often been used to pre-train Deep Belief [18] and CNN models [19]. Norouzi et al. [20] introduced a novel extension of RBMs which they refer to as Convolutional Restricted Boltzmann Machines (CRBM). Compared to traditional RBMs, this variation preserves spatial structure of images. Abdel-Hamid et al. [19] take a similar approach and use stacks of CRBMs to pre-train a CNN designed for speech recognition. The authors found improvements in performance when pre-training with the CRBMs proposed by [20]. Other popular methods prior to the use of Auto-Encoders include the use of PCA [21][22] and ICA [23].

The improvements in performance provided by initialization of weights through RBMs comes with added complexity, which leads to increased difficulty in finding the optimum network topology, particularly for very deep networks. Moreover, training with randomized weights often proved to not give the optimal weight distribution and thus a need for better initialization methods. Auto-Encoders are seen as an alternative for data-dependent feature extraction methods. Auto-encoders are a special kind of feed forward artificial neural networks that learn to reconstruct the input data at the output layer [23].

Auto-Encoders are used for data dimensionality reduction, are trained in an unsupervised greedy layer-wise manner and learn to encode the input vector into a down-sampled representation of the input. Masci et al. [24] showed that initializing CNN with filters of a SCAE significantly improves the performance of CNN. In this paper we follow a similar approach and use SCAE to pre-train a CNN for emotion recognition. Moreover, we incorporate the findings by Ioffe et. al [6] by incorporating BN during training and pre-training.

D. Emotion Recognition Using CNN

Convolutional networks have an ability to self-learn important features necessary for classification while preserving spatial information. This unique ability portrayed by CNN make them an appealing choice for computer vision problems and has led them to become a common choice for classification problems in which spatial information plays an important role in the classification. In the field of emotion recognition, where classification depends upon the shape of facial features, CNN have already set state-of-the-art classification benchmarks. Burkert et al. [25] have devised an emotion recognition model, which they refer to as DeXpression, consisting of a pair of parallel feature extraction blocks consisting of Convolutional, Pooling, and ReLU layers. The authors achieve an average 99.6% accuracy rate on the CKP dataset after a 10-fold cross-validation.

Other approaches include using pre-trained networks, or networks trained for different classification problems, to initialize the weights of new networks [26]. Ouellet [27] presented a model which relies on a deep CNN, originally trained with 1.2 million images from ImageNet, for feature extraction and a Support Vector Machine for classification. The author reports a recognition rate of 94.4% on the CK+ dataset after training with a 10-fold cross-validation method.

A similar approach was followed by Levi and Hassner [28] who use Local Binary Pattern features as input to a number of CNN ensembles to obtain a performance of 54.56% on the Static Facial Expression in the Wild dataset. This network was then use by Duncan et al. [29] who transfered the weights onto a new CNN model consisting of five Convolutional layers and also designed for emotion recognition. The authors train their model on a variety of datasets: a dataset created by the authors and the CK+ and Jaffe datasets, and obtain an accuracy rate of 57.1% and a peak performance of 90.7% during training. Raghuvanshi and Choksi [30] proposed two CNN models trained on the Kaggle Facial Expression Recognition Challenge dataset which consists of images taken in unconstrained environments. The networks achieved an accuracy rate of 48%.

III. Methodology and Experimental Design

Emotion recognition continues to be an area of interest in the research community. Despite a vast number of emotion recognition models being developed, a model that offers a good degree of performance with fast training while being quick enough for real-time recognition is yet to be developed. In this work we try to find the right balance between classification performance and prediction time. We develop a SCAE for Emotion Recognition and compare this with a conventional CNN with BN for emotional recognition. Moreover, we incorporate the findings by [6] and make use of batch normalization to speed up training and improve classification performance. This section of the report outlines our methodology employed.

A. Facial Expression Corpus

This work uses the Karolinska Directed Emotional Faces database (KDEF) [4] due to the high number of participants it contains and taking into consideration that it was created to be particularly suitable for perception, attention, emotion, memory and backward masking experiments [4]. The KDEF database contains a set with 70 individuals: 35 males and 35 females between 20 and 30 years old and each displaying seven different emotional expressions: sad, surprised, neutral, happy, fear, disgust, and angry. Faces are centred within the image and moth and eyes are fixed in specific coordinates. We use a subset of 980 front angle images split into our training and testing sets. In our experiments 70% of this subset is used for training, and pre-training of the SCAE, and the remaining 30% for testing. Each class has the same number of samples in both testing and training sets. Faces are extracted from the original image and grey-scaled in order to reduce dimensionality. Figure 1 illustrates sample face images obtained from the KDEF database.

Fig. 1. Sample extracted face images from the KDEF database. Subject F05 displaying seven emotions: angry, disgust, fear, happy, neutral, sad, surprise.

B. Convolutional Neural Networks with Batch Normalization

The first network described here for real-time emotion recognition departs from empirical CNN models with very large depth and built a model with it only having four Convolutional layers. This approach also adapts the empirical CNN by making use of BN for both, Convolutional and fully connected layers. Essentially, our CNN is composed of Convolutional, BN, ReLU, and Max Pooling layers, except for the last block which does not have a Max Pooling layer. The Convolutional layers contain 20, 40, 60, and 80 planes. The first two Convolutional layers use kernels of 5×5 with zero padding of 1 and 2 over width and height dimensions. The last two Convolutional layers use kernels of size 3×3 with zero padding of 2. First two Max Pooling layers use zero padding of size 1 whereas the last one uses zero padding of size 2. The last block is connected to a fully connected layer which in effect is a Multilayer Perceptron (MLP) with 100 neurons, also with BN and ReLU layers. The output of Convolutional layers is defined by:

$$C(X_{u;v}) = (x+a)^n = \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} f_k(i,j)x_u - i, u - j \quad (1)$$

where f_k is the filter with a kernel size $m \times n$, applied to the input x.

This CNN emotion recognition model is initially trained using mini-batch SGD for 500 epochs as follows: weights are randomly initialized, each mini-batch contains 49 randomly selected training samples, momentum was set to 0.6, the learning rate was set to 0.1 and dynamically adjusted down with a decay of 0.01. Let λ represent the initial learning rate, θ represent the learning rate decay, and ω the current epoch, the learning rate LR is adjusted according to:

$$LR = \frac{\lambda}{1 + \omega \times \theta} \quad (2)$$

During training, the output of the network is shaped by a SoftMax operator and the cross-entropy loss y is defined by:

$$y = -\sum_{ij} \left(x_{ijc} - \log \sum_{d=1}^{D} \exp^{x_{ijd}} \right) \quad (3)$$

C. Stacked Auto-Encoders

In an attempt to improve training time and classification performance of our CNN emotion recognition model, we decided to pre-train as a SCAE Essentially, each Convolutional layer and its subsequent layers: BN, ReLU, and Max Pooling, are treated as a single block and an Auto-Encoder is created for each one of these blocks. However, since compared to traditional Auto-Encoders composed of one dimensional layers, MLPs, Convolutional Auto-Encoders are more difficult to train due to the Max Pooling applied to the output of Convolutional layers and therefore each block of layers is used as the encoder component of the Auto-Encoder and a new block of layers which replaces Max Pooling with Upsampling layers is used as the decoder component. Refer to Figure 2 for a pictorial representation of the SCAE model.

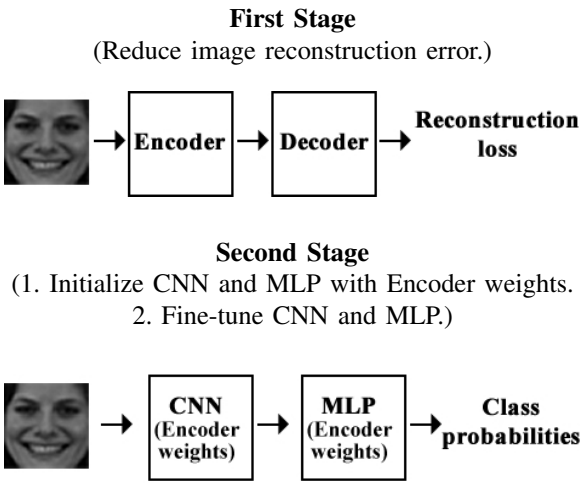

Fig. 2. Illustration of the SCAE architecture. First stage shows training of SCAE which learns to reconstruct the input image and associate the MLP with a corresponding label. Second stage shows CNN with hidden and classification layers. MLP has ReLU and BN layers. Face image corresponding to subject F07 from the KDEF dataset.

Upsampling is done using the nearest neighbour approach with a scale of 2. Let u and v represent image coordinates of the input image, α the scale, then upsampling f is defined as:

$$f(u,v) = \lfloor \frac{u-1}{\alpha} \rfloor + 1, \lfloor \frac{v-1}{\alpha} \rfloor + 1 \quad (4)$$

In the SCAE emotion recognition model, the first Auto-Encoder learns to reconstruct raw pixel data. The second Auto-Encoder learns to reconstruct the output of the first encoder: raw pixel data passed through the first encoder component of the first auto-encoder, and so on. Finally, because the network uses a fully connected layer with 100 hidden units, this layer is trained to encode the output of the last Convolutional encoder and instead of reconstructing it, it learns to associate it with its corresponding label.

All individual Auto-Encoders are trained for only ten epochs using mini-batch SGD. Mini-batches are of size 49 and in the case of the Convolutional Auto-Encoders the loss is measured using a mean absolute value criterion. In the case of the fully connected layer the loss between input x and out y is measured by the cross-entropy criterion referred by equation 3.

Once all the Auto-Encoders are trained, the weights corresponding to each one of the encoders are used as a Stacked Convolutional Auto-Encoder. This SCAE is then fine-tuned as a single unit for only 20 epochs also using SGD and a SoftMax cross-entropy criterion. When trained for higher

Fig. 3. Sample output of first Convolutional layer of the emotion recognition model pre-trained as a SCAE and fine tuned as a CNN. Left to right, subject F05 of the KDEF [4] dataset illustrating: fear, sad, and happy emotions.

number of epochs the performance of the network dropped or remained the same. Learning rate for fine-tuning was set to 0.1 and decayed by a factor of 0.001, whereas momentum was initialized to 0.6.

The encoder is a function f that maps the input $x \epsilon \mathbb{R}^{d_x}$ to a hidden representation $h(x) \epsilon \mathbb{R}^{d_x}$. It has the form:

$$h = f(x) = s_f\big(Wx + b_h\big) \tag{5}$$

where s_f is a ReLU activation function. The decoder function g maps the hidden representation h back to a reconstruction y:

$$y = g(h) = s_g\big(W'h + b_y\big) \tag{6}$$

where s_g is the decoder's activation function. The decoder's parameters are a bias vector $b_y \epsilon \mathbb{R}^{d_x}$, and matrix W'. Training consists in finding parameters $\theta = W, b_h, b_y$ that minimize the reconstruction error on a training set of exampled D_n, which corresponds to minimizing the following objective function:

$$JAE(\theta) = \sum_{x \epsilon D_n} L\Big(x, g\big(f(x)\big)\Big) \tag{7}$$

where L is the reconstruction error [31].

IV. Results and Discussion

The CNN with BN and the SCAE emotion recognisers are trained and tested using the KDEF [4] dataset. When trained from scratch for 500 epochs and with a random weight initialization, the deep CNN model with BN produces an accuracy rate of 100% on the training set and a peak performance of 91% on the testing set. Further training seems to cause overfitting whereas smaller number of epochs decreases accuracy rate. The training set consisted of 98 randomly selected images per class whereas the testing set consisted of 42 images per class, also randomly selected.

In an attempt to improve recognition performance while reducing training time, we investigated the effect of pre-training our model as an Auto-Encoder to learn to reconstruct the input image. To accomplish this, we treated each Convolutional layer, and its subsequent layers, as the encoder component of an Auto-Encoder. We used a similar configuration as the corresponding decoder component except we replaced max pooling layers with spatial nearest neighbour upsampling layers. The encoders were trained individually and then stacked and fine-tuned as a SCAE for 20 epochs. Applying this pre-training technique increased our model's performance to 92.52% and dramatically reduced the training time. Table 1 illustrates the confusion matrix of this model when pre-trained as a SCAE.

As it can be observed in Table 1 our CNN emotion recognition model performs well on the emotions Happy, Neutral, Sad, and Surprised and only misclassifies them once or twice. The worst performance is on the emotion Fear which often tends to be confused with other emotions such as Sad. We observed Fear to always be the most misclassified class when training with different network configurations and parameters. Moreover, we previously observed fear to be one of the most misclassified classes in [32]. A similar correlation was observed with the classes happy and neutral always being the most correctly classified.

The misclassification of images belonging to the class fear can be attributed to their similarity to sad images, notice that sad images were only confused with fear ones: the shape of facial features, particularly of the eyes and eyebrows tend to be very alike. Figure 3 above illustrates the representations learnt by the first Convolutional layer of the CNN which are passed down to lower layers in the network. The left most image is labelled as fear, the middle image as sad, and the right image as happy. It can be observed that the representations learnt for the sad and fear images are relatively identical, whereas the representation learnt for a happy image is very different, particularly the area around the eyes. In effect, this explains the misclassification of such images and exposes the challenge faced by models intended for real-time emotion recognition: since people express emotions in a number of ways, particularly if ethnic backgrounds are different, it can be difficult to create a model that can efficiently differentiate emotions that are expressed in similar ways.

Figure 3 above also allows us to observe that the filters learnt by the first Convolutional layer resemble those produced

TABLE I
SCAE CONFUSION MATRIX: LEFT TO RIGHT; ANGRY, DISGUST, FEAR, HAPPY, NEUTRAL, SAD, SURPRISE. RIGHT MOST COLUM DENOTES AVERAGE ACURACY RATE PER CLASS AND TOTAL AVERAGE.

Label	*An*	*Di*	*Fe*	*Ha*	*Ne*	*Sa*	*Su*	*total*
An	38	1	0	0	1	2	0	90.48
Di	1	38	0	0	0	3	0	90.48
Fe	1	0	35	0	0	4	2	83.33
Ha	0	0	0	41	1	0	0	97.62
Ne	0	0	1	1	40	0	0	95.24
Sa	0	0	2	0	0	40	0	95.24
Su	0	0	1	0	0	1	40	95.24
								92.52

by a bank of Gabor filters which are often used for edge detection. Nevertheless, Convolutional layers have the added advantage of being able to learn to extract these salient features necessary for emotion recognition instead of extracting fixed features. One of the main issues we observed when using Stacked Auto-Encoders as means of pre-training was that if the loss for the first Auto-Encoder becomes too small, then the network fails to learn a model that generalizes the training data. We speculate this this is due to overfitting in the first layer, which is passed down to lower layers in the model and the deeper the layer the higher the error. However, during training this is difficult to detect since the error continues to decrease but the deeper layers are only learning to replicate the bad representation learnt by the first layer. Furthermore, the error for the lower layers tends to decrease much slower than for the first layers. We speculate that training lower layers with higher learning rates and or for longer periods of time could solve this issue. This will be explored in future work.

We have attributed performance and training time improvements to the use of SCAE, though much of this improvement was only possible due to the use of Batch Normalization within our network. By employing BN, we were able to set much higher learning rates and train our initial model, before the use of SCAEs, for only 500 epochs. Before incorporating BN our model produced an average peak performance rate of 86% when trained for the same number of epochs. Moreover, when training without BN the initial learning rate had to be set to 0.0001. In addition to this, when using BN for the Convolutional layers only, classification performance decreases about 2% on average, though this might be due to the higher learning rate used.

The state-of-the-art performance achieved by our SCAE emotion recognition model is comparable to emotion recognition models using deep learning [25], [27], [29]. Moreover, our model achieves similar performance to the model proposed by [33] which uses Gabor filters for feature extraction. The SCAE proposed in this work self-learns Gabor-like filters with the first Convolutional layer and improves the feature vector through lower Convolutional layers. Furthermore, although we cannot compare our model directly to those proposed by [25], [27], [29] due to the different datasets used, we believe that our model has a slight advantage given that it only has four Convolutional layers and was trained for only 70 epochs in total, pre-training and fine-tuning, compared to the model used by [27] which was originally trained with 1.2 million images.

To the best of our knowledge we are the first ones to propose the use of SCAE in conjunction with BN for emotion recognition through facial expression images. The only emotion recognition models that use SCAE, and that we are aware of, perform recognition through speech instead of facial expressions [34]. Another model which makes use of Auto-Encoders, although not stacked, is presented in [35]. However, the model proposed by the authors of [35] only pre-trains one Convolutional layer and keeps its weights fixed during fine-tuning. This approach is often employed due to the added complexity of training SCAE: Since the number of output planes of Convolutional layers is typically high, 20 or more, reconstructing these many planes in the second layer tends to be difficult. Moreover, it is easy for the gradients to vanish if the parameters are not initialized appropriately or the network topology is not ideal.

Our SCAE emotion recognition provides state-of-the-art classification performance and has an added advantage of learning relatively fast compared to traditional CNN models. With these observations and results we conclude that Batch Normalization and Stacked Auto-Encoders can efficiently improve emotion recognition models that use deep learning for feature extraction and classification.

V. CONCLUSIONS AND FUTURE WORK

In this work we have proposed two CNN models: A CNN model that combines BN and fewer layers than an empirical CNN and a SCAE that pre-trains the weights to the CNN element using Auto-Encoders. Both methods provide state-of-the-art classification performance with the SCAE being relatively faster to train. With the evident advantage portrayed by SCAE, future work will look at ways to improve this model. Moreover, we also plan to explore the effect of pre-training Auto-Encoders as a single unit rather than layer by layer. This method has proven to be efficient by Zhou et al. [36] who found that training deep Auto-Ancoders can also be done jointly instead of layer-wise.

Despite the state-of-the-art results achieved on the KDEF [4] data set we are aware that these images used were from front on and all of the same quality. Given it is our goal to make use of this SCAE on a robotic system, it is likely that the images will vary in terms of the angle of the user's face and the light conditions. Hence in the future we will explore the model's resistance to these situations.

In this work we show that employing SCAE as a pre-training method for deep CNN improves not only performance but training time and have a positive impact on the performance of the recognition rate. Due to the very fast convergence observed, in part due to the use of BN, we speculate that

our architecture would perform better if trained with bigger datasets. In addition to this, since the network reaches a good local minimum relatively fast, a deeper network with similar properties should take advantage of fast learning and could produce higher accuracy rates.

REFERENCES

[1] M. Lewis, J. M. Haviland-Jones, and L. F. Barrett, "Handbook of Emotions." *Contemporary Sociology*, vol. 24, no. 3, p. 298, may 1995.

[2] A. Chavhan, S. Chavan, S. Dahe, and S. Chibhade, "A Neural Network Approach for Real Time Emotion Recognition," *Ijarcce*, vol. 4, no. 3, pp. 259–263, 2015. [Online]. Available: http://ijarcce.com/upload/2015/march-15/IJARCCE 62.pdf

[3] C. D. Cameron, K. A. Lindquist, and K. Gray, "A Constructionist Review of Morality and Emotions: No Evidence for Specific Links Between Moral Content and Discrete Emotions," *Personality and Social Psychology Review*, vol. 19, no. 4, pp. 371–394, nov 2015. [Online]. Available: http://psr.sagepub.com/cgi/doi/10.1177/1088868314566683

[4] D. Lundqvist, A. Flykt, and A. Öhman, "The Karolinska Directed Emotional Faces - KDEF CD ROM from Department of Clinical Neuroscience, Psycology section," *Karolinska Institutet*, pp. 3–5, 1998.

[5] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009. [Online]. Available: http://www.nowpublishers.com/article/Details/MAL-006

[6] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," feb 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[7] S. Tan and B. Li, "Stacked convolutional auto-encoders for steganalysis of digital images," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, no. 2012lym 0117. IEEE, dec 2014, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/document/7041565/

[8] P. Krähenbühl, C. Doersch, J. Donahue, and T. Darrell, "Data-dependent Initializations of Convolutional Neural Networks," *International Conference on Computer Vision*, pp. 1–12, nov 2015. [Online]. Available: http://arxiv.org/abs/1511.06856

[9] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for Thin Deep Nets," pp. 1–13, dec 2014. [Online]. Available: http://arxiv.org/abs/1412.6550

[10] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway Networks," *arXiv:1505.00387 [cs]*, may 2015. [Online]. Available: http://arxiv.org/abs/1505.00387

[11] D. Mishkin and J. Matas, "All you need is a good init," *Computers & Mathematics with Applications*, vol. 31, no. 11, p. 135, nov 2015. [Online]. Available: http://arxiv.org/abs/1511.06422

[12] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 9, pp. 249–256, 2010.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," feb 2015. [Online]. Available: http://arxiv.org/abs/1502.01852

[14] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," *Proceedings of the 30 th International Conference on Machine Learning*, vol. 28, p. 6, 2013. [Online]. Available: https://web.stanford.edu/ awni/papers/relu_hybrid_icml2013_final.pdf

[15] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," *ICML Deep Learning Workshop*, pp. 1–5, may 2015. [Online]. Available: http://arxiv.org/abs/1505.00853

[16] A. Krizhevsky, L. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *NIPS*, pp. 1106–1114, 2012.

[17] D. Erhan, Y. Bengio, A. Courville, P.-A. Mazagol, and P. Vincent, "Representation Learning: A Review and New Perspectives," *Journal of Machine Learning Research*, vol. 11, pp. 625–660, jun 2010. [Online]. Available: http://arxiv.org/abs/1206.5538

[18] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets." *Neural computation*, vol. 18, no. 7, pp. 1527–54, jul 2006.

[19] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition," in *INTERSPEECH*, 2013.

[20] M. Norouzi, M. Ranjbar, and G. Mori, "Stacks of Convolution Restricted Boltzmann Machines for Shift-Invariant Feature Learning," 2009.

[21] J. J. Lien, T. Kanade, A. J. Zlochower, J. F. Cohn, and C.-C. Li, "Automatically Recognizing Facial Expressions in the Spatio-Temporal Domain," pp. 94–97. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.30.2652

[22] B. A. Draper, K. Baek, M. S. Bartlett, and J. Beveridge, "Recognizing faces with PCA and ICA," *Computer Vision and Image Understanding*, vol. 91, no. 1-2, pp. 115–137, jul 2003. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1077314203000778

[23] C. C. Tan and C. Eswaran, "Reconstruction and recognition of face and digit images using autoencoders," *Neural Computing and Applications*, vol. 19, no. 7, pp. 1069–1079, oct 2010. [Online]. Available: http://link.springer.com/10.1007/s00521-010-0378-4

[24] J. Masci, U. Meier, D. Cirean, and J. Schmidhuber, "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6791 LNCS, no. PART 1, pp. 52–59. [Online]. Available: http://link.springer.com/10.1007/978-3-642-21735-7_7

[25] P. Burkert, F. Trier, M. Z. Afzal, A. Dengel, and M. Liwicki, "DeXpression: Deep convolutional neural network for expression recognition," *arXiv preprint*, pp. 1–8, 2015. [Online]. Available: http://arxiv.org/abs/1509.05371

[26] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," *Icml*, vol. 32, pp. 647–655, oct 2013. [Online]. Available: http://arxiv.org/abs/1310.1531

[27] S. Ouellet, "Real-time emotion recognition for gaming using deep convolutional network features," *CoRR*, vol. abs/1408.3, p. 6, aug 2014. [Online]. Available: http://arxiv.org/abs/1408.3750

[28] G. Levi and T. Hassner, "Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns," pp. 503–510, 2015.

[29] D. Duncan, G. Shine, and C. English, "Facial Emotion Recognition in Schizophrenia ."

[30] A. Raghuvanshi and V. Choksi, "Facial Expression Recognition with Convolutional Neural Networks."

[31] S. Rifai and X. Muller, "Contractive Auto-Encoders : Explicit Invariance During Feature Extraction," *Icml*, vol. 85, no. 1, pp. 833–840, 2011. [Online]. Available: http://www.icml-2011.org/papers/455_icmlpaper.pdf

[32] A. Ruiz-Garcia, M. Elshaw, A. Altahhan, and V. Palade, "Deep Learning for Emotion Recognition in Faces," in *Artificial Neural Networks and Machine Learning – ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II*, 2016, pp. 38–46. [Online]. Available: http://link.springer.com/10.1007/978-3-319-44781-0_5

[33] T. Ahsan, T. Jabid, and U.-P. Chong, "Facial Expression Recognition Using Local Transitional Pattern on Gabor Filtered Facial Images," *IETE Technical Review*, vol. 30, no. 12, p. 47, 2013.

[34] N. Cibau, E. Albornoz, and H. Rufiner, "Speech emotion recognition using a deep autoencoder," *XV Reunión de Trabajo en Procesamiento de la Información y Control*, no. i, pp. 934–939, 2013. [Online]. Available: http://fich.unl.edu.ar/sinc/sinc-publications/2013/CAR13/sinc_CAR13.pdf

[35] D. Hamester, P. Barros, and S. Wermter, "Face expression recognition with a 2-channel Convolutional Neural Network," in *2015 International Joint Conference on Neural Networks (IJCNN)*, vol. 2015-Septe, no. July. IEEE, jul 2015, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/7280539/

[36] Y. Zhou, D. Arpit, I. Nwogu, and V. Govindaraju, "Is Joint Training Better for Deep Auto-Encoders?" pp. 1–11, 2014. [Online]. Available: http://arxiv.org/abs/1405.1380