Facial Expression Recognition using Transfer Learning

Soodamani Ramalingam
Centre for Engineering Research, School of Engineering & Technology, University of Hertfordshire College Lane Campus, Hatfield, Hertfordshire AL10 9AB, UK

Fabio Garzia
Safety & Security Engineering Group – DICMA, SAPIENZA – University of Rome, Rome, Italy

***Abstract*— In this paper, we investigate Deep Learning architectures for the recognition of facial expressions. In particular, we consider the concept of Transfer Learning whereby features learnt from generic images of large scale datasets can be used to train models of smaller databases without losing the generalization ability.**

Keywords—facial expression recognition, transfer learning, deep learning, Convolutional Neural Networks

I. INTRODUCTION

Facial Expression Recognition (FER) has gained importance and popularity among the Vision Community since the series of emotion recognition competitions such as FER2013 [1] and EmotiW [2] made it possible to acquire sufficient training data from real-world. In particular, Deep Learning (DL) techniques have shown to cope well with emotion recognition in the wild. Training data is key to all DL techniques. However, the variations in the training sets pose a problem of insufficient samples leading to the common 'overfitting' or lack of generalisation issues as well as large intra-class variability.

Like any other Computer Vision task, DL techniques comprise of three main stages namely, pre-processing, feature learning and feature classification. Standard pre-processing tasks such as face alignment and image normalisation is often needed. In addition, *Data Augmentation* is a key pre-processing technique deployed to generate synthetic datasets from existing data through geometric transformations to make up for the lack of sufficient expression samples.

Deep Learning hierarchical architectures have the ability to provide abstraction through derivations from multiple transformations and representations. Popular DL architectures include Convolutional Neural Networks (CNNs), Deep Belief Network (DBN), Deep Autoencoder (DAE), Recurrent Neural Network (RNN) and others. In this paper we consider CNNs due to their good generalisation ability and invariance to geometrical transformations such as translation, rotation, and scaling.

II. CONVOLUTIONAL ARCHITECTURES FOR FEATURE LEARNING AND CLASSIFICATION

A. Convolutional Neural Networks

Convolutional Networks (ConvNet) typically consist of three layers namely CONV, POOL and FC (fully connected). These layers are stacked to form a full ConvNet architecture. The *Convolutional Layer* (CONV) has a set of learnable filters that are convolved with the original images providing as output specific activation feature maps. This layer is followed by *Pooling Layer* to reduce the spatial size of the feature maps and the computational cost. The last layer is the *Fully Connected Layer* and connects all neurons in the last to the previous layer and simultaneously converts 2D feature vectors to 1D maps (*Flatten)* that are useful for further representation and classification. Further, *Rectified Linear Unit (ReLU)* apply an elementwise activation function. See Figs. 1-2.

Symbolically, such a network can be described by [INPUT-CONV-RELU-POOL-FC]. It is common for ConvNet architectures to stack a few CONV-RELU layers, followed by POOL layers and then repeat this pattern until the image has been spatially reduced to a small size. Once the features are learnt through ConvNet architectures, classification of FER takes place. DL techniques can be used to classify by adding a loss layer at the end of the network to regulate back-propagation error. This enables to determine as output the prediction probability of each sample. Alternatively, classifiers such as Support Vector Machines (SVM) or Random Forest may be added to the learned features.

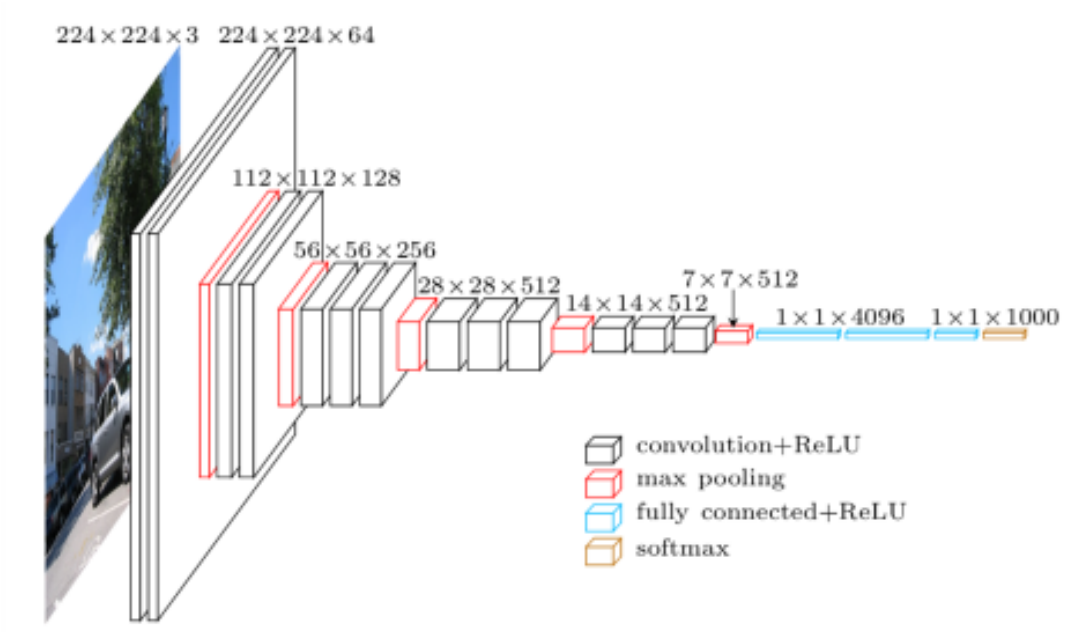

Fig. 1. VGG16 and VGG19 architecture (source)

B. *Literature Review*

It is well known that direct application of DL architectures on relatively small databases suffer from overfitting. Two approaches exist that attempt to overcome this issue by either using additional task-oriented data to pre-train the networks or use well known pre-trained models such AlexNet,[3] VGG [4], VGG-face [5], GoogleNet [6] and the like. Recent studies show significant improvement of poorly performing FER systems. An ensemble of networks approach has also shown to improve performance of an individual network. Such a network ensemble should have characteristics of being complementary, that is diversified and a mechanism to aggregate such ensembles [7]. Because of the intra-variations in expressions influenced by factors such as age, gender, culture, etc., for FER systems to perform well, it is required to have abundant samples during training. Further, such samples need annotations based on the factors mentioned above. In that context, the concept of transfer learning in DL is emerging.

C. *FER Databases*

Several databases exist in the literature[7]. We consider the following databases for our paper:

The Extended Cohn-Kanade Dataset (CK+) [8], the Japanese Female Facial Expression JAFFE [9] database, and the FER2013 database[1](Fig.3). All of these databases consist of 7 expressions. CK+ consists of 123 subjects and 593 samples, JAFFE database has only 10 subjects with 213 samples and FER2013 has 35,887 sample images.

The aim of our paper is to compare the performance of transfer learning on different datasets namely FER2013, CK+ and JAFFE databases.

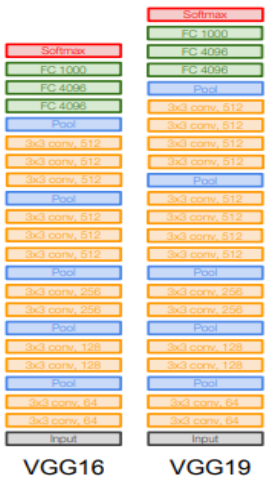

Fig 2. VGG Nets with Small Filters and Deeper Nets (source)

Fig. 3. FER Datasets (source)

III. TRANSFER LEARNING

The notion of representations learnt from pre-trained networks for a particular task such as object detection being transferred to a different task of facial expression recognition is explored. Thus, we propose to use Transfer Learning from Deep Convolutional Networks to recognise facial expressions [10]. Two DL architectures that are deployed here are VGG16 and VGG19 whose architectural diagrmas are shown in Fig.2.

A. Experiments with Transfer Learning

We test three exisitng algorithms that vary slightly in the approach to suit each database. The first of the algorithm in [11] is as follows:

Algorithm A. *With Data Augmentation on Kaggle FER2013 Database [11]*

In its simplest form, data augmentation makes up for the lack of data by applying transformations on existing data. This algorithm performs data augmentation on the exisitng database.

Stage 1. Pre-Processing

1) *Load data:* Database-FER2013, image size: 48x48 = 2304 vector. #classes=7 =[0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, and 6=Neutral]
2) *Split data:* (training: test) = (28273,7067)
3) *Augment data:* rotation, scaling, shift along X and Y axes

Stage 2. Creating the Network
Add laeyrs sequentially: [CONV-CONV-NORM-RELU-POOL]x3 →[FC]

Stage 3. Training the Network

1) Num_epochs=100
2) Fit model on batches with real-time augmentation

Stage 4. Learning decision
Determine loss on training and test sets over the training epochs

Stage 5. Making Predictions

1) Test on individual images
2) Evaluate trained model on test set.

In original paper [11], samples for the expression 'Disgust' have been removed as they were fewer in number. In our work, we've put back this category and included data augmentation. This algorithm has been simply adopted as it is to verify that the accuracy of the model is about 60%.

Algorithm B. *Algoritm B-Transfer Learning: Pre-trained Bottleneck Features of VGG16 with Data Augmentation on Kaggle FER2013 Database [12]*

The Keras blog in[12]presents a technique for building a powerful image classifier with few training samples in the order of few 100s -1000s samples/class. It is demonstrated to perform a binary classification of dogs vs cats from Kaggle dataset [13]. Three possible approaches to transfer learning are outlined in the Blog. We choose to adapt the bottleneck features of a pre-trained network to build a model for the FER2013 dataset. This is a more refined mechanism to leverage a network that has been pre-trained on a large dataset and in this case the VGG16 architecture that contains 1000 classes. It is expected that this will improve the generalisation ability on the FER2013 dataset.

The VGG16 architecture is represnted as [[CONVx2-POOL]x3→[CONVx3-POOL]x2 →FCx3]. The features learnt from VGG16 only up to the convolutional model up to the fuly connected layers is instantaited. This model is run on the traing and validation data of FER2013 once, thus recording the bottleneck features from VGG16 model. The model is then trained with a small fully connected model on top of the stored features.

The result shown an improved accuracy of around 76% which is significant.

Algorithm C. *Transfer Learning: Pre-trained Features of VGG19 on JAFFE, CK+ and FER2013 databases [14]*

We then adapt the work in [14] for use of pre-trained CNNs for learning and classifying samples from smaller databases of JAFFE and a subset of CK+. In this technique, features extracted from VGG19 from each pooling layer and the first fully connected layer are extracted. Principal Component Analysis (PCA) is further used for dimensioanlity reduction. No data augmentation is carried out.

In comparison to the FER2013 database, the JAFFE and a subset of CK+ databases have far too few samples. The VGG19 architecture is described by [[CONVx2-POOL]x2→ [CONVx4-POOL]x3→[FCx2]. These are then reduced in dimensionalty using PCA. The number of PCA components N_{PCA} ={ 50, 100, 150, 200} is investiaged for idntifying an optimal set of parameters whose combination produces best results of training and test accuracies:

(i) the layer of VGG19
(ii) N_{PCA} elements

Feature selection is based on the combination of the best perfroming N_{PCA} and VGG19 layer of feature extraction. A Support Vector Machine (SVM) with a linear kernel is used for classification. Data splitting of Training: Test :: 80:20. A 10-fold cross validation is implemented. Also, a leave-one-out startegy is tested. Results show that N_{PCA}=100 for CK+ dataset N_{PCA}= 200 for JAFFE dataset alongwith Block 4-Pool layer features. For the FER2013 dataset, two varaiations in samples sizes namely 30 samples/class to keep in line with the other databses and 100 samples/class were selected and results obtained. Results are verified as reported as follows:

Table 1 Performance Analysis of Transfer Learning on Datasets

	Training Accuracy	**Testing Accuracy**	**Mean Score on Leave-One-Out**
CK+	0.8924 (+/- 0.09)	0.9048	0.892 (+/-0.014)
JAFFE	0.76 (+/-0.033)	0.7381	0.784 (+/-0.032)
FER2013: 30 samples/class	0.202 (+/-0.031)	0.2143	0.202 (+/-0.025)
FER2013: 100 samples / class	0.232 (+/-0.033)	0.1429	0.205 (+/-0.037)

B. Discussions

From Table I, it is seen that features learnt from VGG19 works well on JAFFE and CK+ databases which are much smaller in size compared to FER2013. The results on FER2013 database, immaterial of the two different samples/class sizes indicate utterly poor performance. Some of the reasons that this may be due to th fact:

1) Intra-class variation is high in FER2013 even for human beings to classify them apprpriately.
2) No consistent N_{PCA} was a winner for the FER2013 database. N_{PCA}=150 was the consistent under-performer.
3) No best performing layer emerged with VGG19.

IV. FUTURE WORK

All of the three algorithms showed good performance improvement with CK+ and JAFFE databases. Data augmentation ceratinly helped with the increased performance as it learnt the geometrical transformations along with the original datasets. FER2013 is a more trying dataset and data augmentation will be added to it with transfer learning of VGG19 and similar netwrok architectures.

V. REFERENCES

1. Goodfellow, I.J., et al., *Challenges in representation learning: A report on three machine learning contests.* Neural Networks, 2015. **64**: p. 59-63.
2. Dhall, A., et al., *EmotiW 2016: video and group-level emotion recognition challenges*, in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*. 2016, ACM: Tokyo, Japan. p. 427-432.
3. A. Krizhevsky, I. Sutskever, and G.E. Hinton, *Imagenet Classification with Deep Convolutional Neural Networks.* Advances in Neural Information Processing Systems, 2012: p. 1097-1105.
4. Simonyan, K. and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*. CoRR, 2014. **abs/1409.1556**.
5. M. Parkhi, O., A. Vedaldi, and A. Zisserman, *Deep Face Recognition*. Vol. 1. 2015. 41.1-41.12.
6. Szegedy, C., et al. Going deeper with convolutions. in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015.
7. S. Li and W. Deng, Deep Facial Expression Recognition: A Survey. CoRR, 2018. **abs/1804.08348**: p. 22.
8. P. Lucey, et al. 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 (CVPR Workshops). 2010. San Francisco, CA, USA, USA.
9. O. Langner, et al., Presentation and validation of the radboud faces database. Cognition and Emotion, 2010. **24**(8): p. 1377-1388.
10. M. Xu, et al. Facial Expression Recognition Based On Transfer Learning From Deep Convolutional Networks. in 2015 11th International Conference on Natural Computation (ICNC). 2015. Zhangjiajie.
11. Sosnovshchenko, A., Machine Learning with Swift",. Getting Started with Machine Learning. Feb. 2018: Packt Pub.
12. Chollet, F., Building powerful image classification models using very little data, in Tutorial. 2015.
13. Kaggle: Dataset of Dogs Vs Cats, Kaggle, Editor. 2018.

14. Ravi, A., Pre-Trained Convolutional Neural Network Features for Facial Expression Recognition. Mar. 2018, Dept. of Systems Design Engineering, University of Waterloo: Canada.