Real-time Convolutional Neural Networks for Emotion and Gender Classification

Octavio Arriaga
Hochschule Bonn-Rhein-Sieg
Sankt Augustin Germany
Email: octavio.arriaga@smail.inf.h-brs.de

Paul G. Plöger
Hochschule Bonn-Rhein-Sieg
Sankt Augustin Germany
Email: paul.ploeger@h-brs.de

Matias Valdenegro
Heriot-Watt University
Edinburgh, UK
Email: m.valdenegro@hw.ac.uk

Abstract—In this paper we propose an implement a general convolutional neural network (CNN) building framework for designing real-time CNNs. We validate our models by creating a real-time vision system which accomplishes the tasks of face detection, gender classification and emotion classification simultaneously in one blended step using our proposed CNN architecture. After presenting the details of the training procedure setup we proceed to evaluate on standard benchmark sets. We report accuracies of 96% in the IMDB gender dataset and 66% in the FER-2013 emotion dataset. Along with this we also introduced the very recent real-time enabled guided back-propagation visualization technique. Guided back-propagation uncovers the dynamics of the weight changes and evaluates the learned features. We argue that the careful implementation of modern CNN architectures, the use of the current regularization methods and the visualization of previously hidden features are necessary in order to reduce the gap between slow performances and real-time architectures. Our system has been validated by its deployment on a Care-O-bot 3 robot used during RoboCup@Home competitions. All our code, demos and pre-trained architectures have been released under an open-source license in our public repository.

I. INTRODUCTION

The success of service robotics decisively depends on a smooth robot to user interaction. Thus, a robot should be able to extract information just from the face of its user, e.g. identify the emotional state or deduce gender. Interpreting correctly any of these elements using machine learning (ML) techniques has proven to be complicated due the high variability of the samples within each task [4]. This leads to models with millions of parameters trained under thousands of samples [3]. Furthermore, the human accuracy for classifying an image of a face in one of 7 different emotions is 65% ± 5% [4]. One can observe the difficulty of this task by trying to manually classify the FER-2013 dataset images in Figure 1 within the following classes {"angry", "disgust", "fear", "happy", "sad", "surprise", "neutral"}.

In spite of these difficulties, robot platforms oriented to attend and solve household tasks require facial expressions systems that are robust and computationally efficient. Moreover, the state-of-the-art methods in image-related tasks such as image classification [1] and object detection are all based on Convolutional Neural Networks (CNNs). These tasks require CNN architectures with millions of parameters; therefore, their deployment in robot platforms and real-time systems

Fig. 1: Samples of the FER-2013 emotion dataset [4].

Fig. 2: Samples of the IMDB dataset [9].

becomes unfeasible. In this paper we propose an implement a general CNN building framework for designing real-time CNNs. The implementations have been validated in a real-time facial expression system that provides face-detection, gender classification and that achieves human-level performance when classifying emotions. This system has been deployed in a care-O-bot 3 robot, and has been extended for general robot

platforms and the RoboCup@Home competition challenges.

Furthermore, CNNs are used as black-boxes and often their learned features remain hidden, making it complicated to establish a balance between their classification accuracy and unnecessary parameters. Therefore, we implemented a real-time visualization of the guided-gradient back-propagation proposed by Springenberg [11] in order to validate the features learned by the CNN.

II. RELATED WORK

Commonly used CNNs for feature extraction include a set of fully connected layers at the end. Fully connected layers tend to contain most of the parameters in a CNN. Specifically, VGG16 [10] contains approximately 90% of all its parameters in their last fully connected layers. Recent architectures such as Inception V3 [12], reduced the amount of parameters in their last layers by including a Global Average Pooling operation. Global Average Pooling reduces each feature map into a scalar value by taking the average over all elements in the feature map. The average operation forces the network to extract global features from the input image. Modern CNN architectures such as Xception [1] leverage from the combination of two of the most successful experimental assumptions in CNNs: the use of residual modules [6] and depth-wise separable convolutions [2]. Depth-wise separable convolutions reduce further the amount of parameters by separating the processes of feature extraction and combination within a convolutional layer.

Furthermore, the state-of-the-art model for the FER2-2013 dataset is based on CNN trained with square hinged loss [13]. This model achieved an accuracy of 71% [4] using approximately 5 million parameters. In this architecture 98% of all parameters are located in the last fully connected layers.

The second-best methods presented in [4] achieved an accuracy of 66% using an ensemble of CNNs.

III. MODEL

We propose two models which we evaluated in accordance to their test accuracy and number of parameters. Both models were designed with the idea of creating the best accuracy over number of parameters ratio. Reducing the number of parameters help us overcoming two important problems. First, the use of small CNNs alleviate us from slow performances in hardware-constrained systems such robot platforms. And second, the reduction of parameters provides a better generalization under an Occam's razor framework. Our first model relies on the idea of eliminating completely the fully connected layers. The second architecture combines the deletion of the fully connected layer and the inclusion of the combined depth-wise separable convolutions and residual modules. Both architectures were trained with the ADAM optimizer [8].

Following the previous architecture schemas, our initial architecture used Global Average Pooling to completely remove any fully connected layers. This was achieved by having in the last convolutional layer the same number of feature maps as number of classes, and applying a softmax activation function
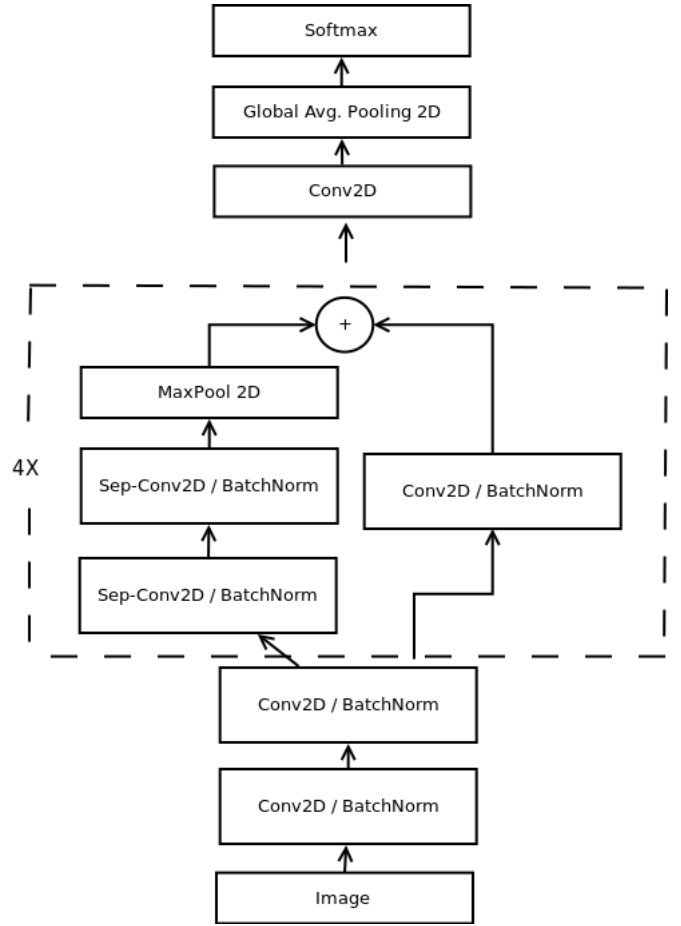

Fig. 3: Our proposed model for real-time classification.

to each reduced feature map. Our initial proposed architecture is a standard fully-convolutional neural network composed of 9 convolution layers, ReLUs [5], Batch Normalization [7] and Global Average Pooling. This model contains approximately 600,000 parameters. It was trained on the IMDB gender dataset, which contains 460,723 RGB images where each image belongs to the class "woman" or "man", and it achieved an accuracy of 96% in this dataset. We also validated this model in the FER-2013 dataset. This dataset contains 35,887 grayscale images where each image belongs to one of the following classes {"angry", "disgust", "fear", "happy", "sad", "surprise", "neutral"}. Our initial model achieved an accuracy of 66% in this dataset. We will refer to this model as "sequential fully-CNN".

Our second model is inspired by the Xception [1] architecture. This architecture combines the use of residual modules [6] and depth-wise separable convolutions [2]. Residual modules modify the desired mapping between two subsequent layers, so that the learned features become the difference of the original feature map and the desired features. Consequently, the desired features $H(x)$ are modified in order to solve an easier learning problem $F(X)$ such that:
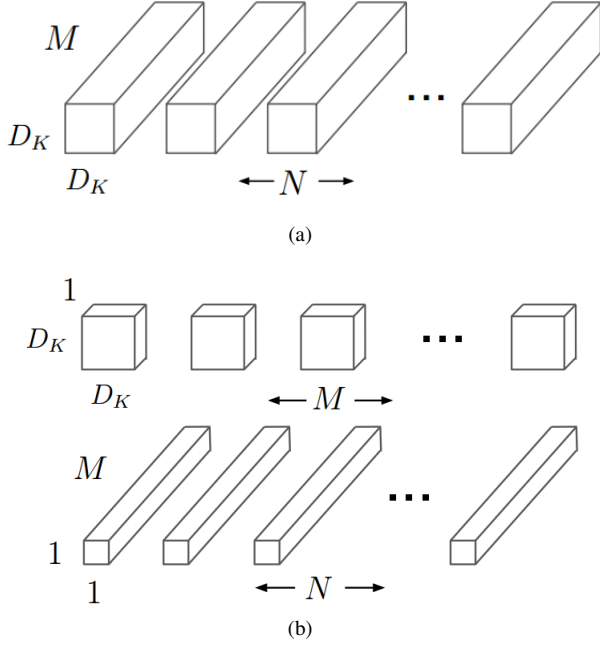
$$H(x) = F(x) + x \tag{1}$$

Fig. 4: [2] Difference between (a) standard convolutions and (b) depth-wise separable convolutions.

Since our initial proposed architecture deleted the last fully connected layer, we reduced further the amount of parameters by eliminating them now from the convolutional layers. This was done trough the use of depth-wise separable convolutions. Depth-wise separable convolutions are composed of two different layers: depth-wise convolutions and point-wise convolutions. The main purpose of these layers is to separate the spatial cross-correlations from the channel cross-correlations [1]. They do this by first applying a $D \times D$ filter on every M input channels and then applying N $1 \times 1 \times M$ convolution filters to combine the M input channels into N output channels. Applying $1 \times 1 \times M$ convolutions combines each value in the feature map without considering their spatial relation within the channel.

Depth-wise separable convolutions reduces the computation with respect to the standard convolutions by a factor of $\frac{1}{N} + \frac{1}{D^2}$ [2]. A visualization of the difference between a normal Convolution layer and a depth-wise separable convolution can be observed in Figure 4.

Our final architecture is a fully-convolutional neural network that contains 4 residual depth-wise separable convolutions where each convolution is followed by a batch normalization operation and a ReLU activation function. The last layer applies a global average pooling and a soft-max activation function to produce a prediction. This architecture has approximately $60,000$ parameters; which corresponds to a reduction of $10\times$ when compared to our initial naive implementation, and $80\times$ when compared to the original CNN. Figure 3 displays our complete final architecture which we refer to as mini-Xception. This architectures obtains an accuracy of 95% in gender classification task. Which corresponds

to a reduction of one percent with respect to our initial implementation. Furthermore, we tested this architecture in the FER-2013 dataset and we obtained the same accuracy of 66% for the emotion classification task. Our final architecture weights can be stored in an 855 kilobytes file. By reducing our architectures computational cost we are now able to join both models and use them consecutively in the same image without any serious time reduction. Our complete pipeline including the openCV face detection module, the gender classification and the emotion classification takes 0.22 ± 0.0003 ms on a i5-4210M CPU. This corresponds to a speedup of $1.5\times$ when compared to the original architecture of Tang.

We also added to our implementation a real-time guided back-propagation visualization to observe which pixels in the image activate an element of a higher-level feature map. Given a CNN with only ReLUs as activation functions for the intermediate layers, guided-back propagation takes the derivative of every element (x, y) of the input image I with respect to an element (i, j) of the feature map f^L in layer L. The reconstructed image R filters all the negative gradients; consequently, the remaining gradients are chosen such that they only increase the value of the chosen element of the feature map. Following [11], a fully ReLU CNN reconstructed image in layer l is given by:

$$R_{i,j}^l = (R_{i,j}^{l+1} > 0) * R_{i,j}^{l+1} \qquad (2)$$

IV. RESULTS

Results of the real-time emotion classification task in unseen faces can be observed in Figure 5. Our complete real-time pipeline including: face detection, emotion and gender classification have been fully integrated in our Care-O-bot 3 robot.

An example of our complete pipeline can be seen in Figure 6 in which we provide emotion and gender classification. In Figure 7 we provide the confusion matrix results of our emotion classification mini-Xception model. We can observe several common misclassifications such as predicting "sad" instead of "fear" and predicting "angry" instead "disgust".

A comparison of the learned features between several emotions and both of our proposed models can be observed in Figure 8. The white areas in figure 8b correspond to the pixel values that activate a selected neuron in our last convolution layer. The selected neuron was always selected in accordance to the highest activation. We can observe that the CNN learned to get activated by considering features such as the frown, the teeth, the eyebrows and the widening of one's eyes, and that each feature remains constant within the same class. These results reassure that the CNN learned to interpret understandable human-like features, that provide generalizable elements. These interpretable results have helped us understand several common misclassification such as persons with glasses being classified as "angry". This happens since the label "angry" is highly activated when it believes a person is frowning and frowning features get confused with darker glass frames. Moreover, we can also observe that the features learned in

Fig. 5: Results of the provided real-time emotion classification provided in our public repository
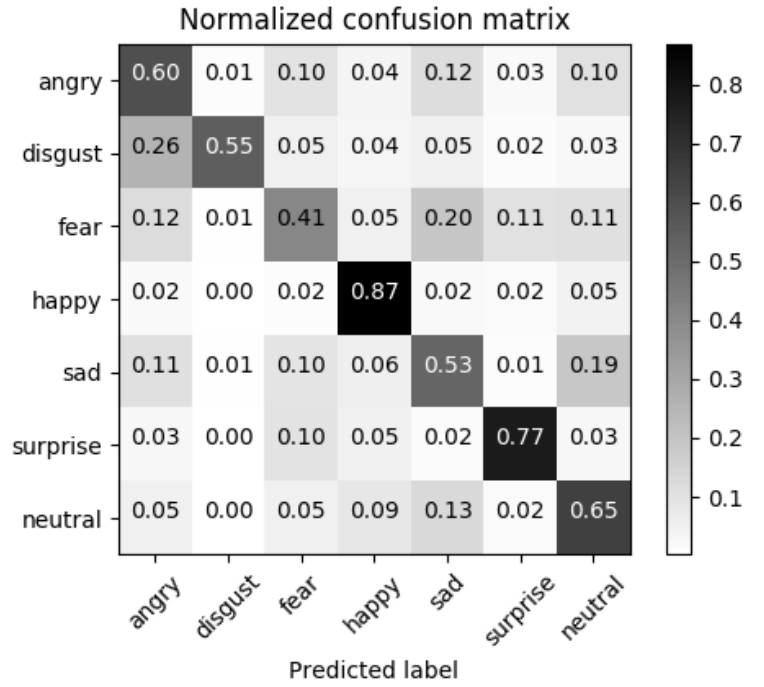

Fig. 7: Normalized confusion matrix of our mini-Xception network.

our mini-Xception model are more interpretable than the ones learned from our sequential fully-CNN. Consequently the use of more parameters in our naive implementations leads to less robust features.

V. FUTURE WORK

Machine learning models are biased in accordance to their training data. In our specific application we have empirically found that our trained CNNs for gender classification are biased towards western facial features and facial accessories. We hypothesize that this misclassfications occurs since our training dataset consist of mostly western: actors, writers and cinematographers as observed in Figure 2.

Furthermore, as discussed previously, the use of glasses might affect the emotion classification by interfering with the features learned. However, the use of glasses can also interfere with the gender classification. This might be a result from the training data having most of the images of persons wearing glasses assigned with the label "man". We believe that uncovering such behaviours is of extreme importance when creating robust classifiers, and that the use of the visualization techniques such as guided back-propagation will become invaluable when uncovering model biases.
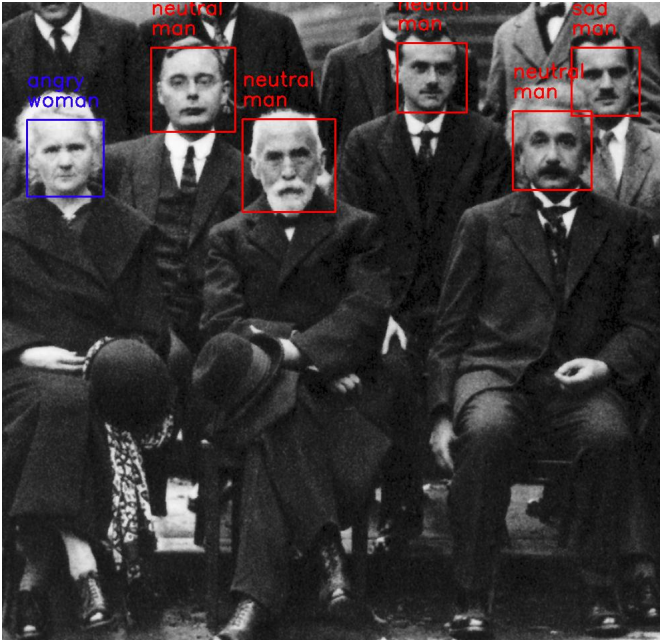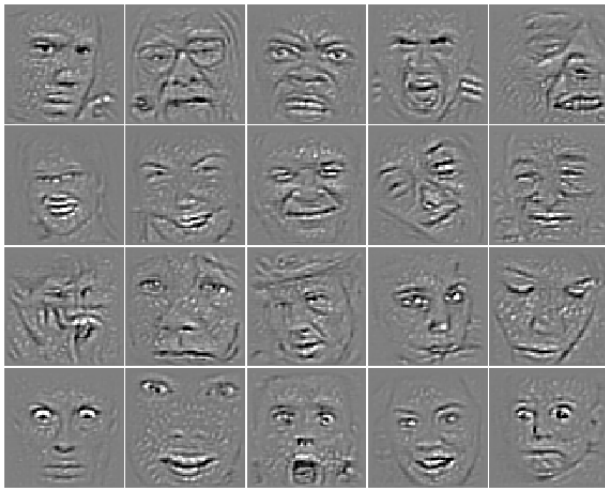

Fig. 6: Results of the provided combined gender and emotion inferences demo. The color blue represents the assigned class *woman* and red the class *man*

(a)

(b)

(c)

Fig. 8: All sub-figures contain the same images in the same order. Every row starting from the top corresponds respectively to the emotions {"angry", "happy", "sad", "surprise"} (a) Samples from the FER-2013 dataset (b) Guided back-propagation visualization of our mini-Xception model (c) Guided back-propagation visualization of our sequential fully-CNN.

VI. CONCLUSIONS

We have proposed and tested a general building designs for creating real-time CNNs. Our proposed architectures have been systematically built in order to reduce the amount of parameters. We began by eliminating completely the fully connected layers and by reducing the amount of parameters in the remaining convolutional layers via depth-wise separable convolutions. We have shown that our proposed models can be stacked for multi-class classifications while maintaining real-time inferences. Specifically, we have developed a vision system that performs face detection, gender classification and emotion classification in a single integrated module. We have achieved human-level performance in our classifications tasks using a single CNN that leverages modern architecture constructs. Our architecture reduces the amount of parameters $80\times$ while obtaining favorable results. Our complete pipeline has been successfully integrated in a Care-O-bot 3 robot. Finally we presented a visualization of the learned features in the CNN using the guided back-propagation visualization. This visualization technique is able to show us the high-level features learned by our models and discuss their interpretability.

REFERENCES

[1] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[2] Andrew G. Howard et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[3] Dario Amodei et al. Deep speech 2: End-to-end speech recognition in english and mandarin. *CoRR*, abs/1512.02595, 2015.

[4] Ian Goodfellow et al. Challenges in Representation Learning: A report on three machine learning contests, 2013.

[5] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[8] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, July 2016.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[11] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[13] Yichuan Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013.