Real-time Face Expression Recognition using Convolutional Neural Networks

Anmol Singh Suag
University of Massachusetts Amherst
asuag@umass.edu

Abstract

I have developed a real-time facial expression recognition system using Convolutional Neural Networks. An existing VGG-16 network trained on CASIA-Webface dataset was fine-tuned on a collection of images from various datasets including JAFFE, CK+ as well as a personal dataset. The fine-tuned network has an overall training accuracy of 75.05% and an overall test accuracy of 75.21%. Individually, the network has a test accuracy of 78.13% on CK+ and 84.8% on the personal dataset. A live video stream from the webcam is used as input and the classified frames are plotted with the face bounding boxes and the classified expression. The real-time expression recognition system has a 0.347 sec/image classification time on a commodity machine and a small pre-processing overhead.

1. Introduction

Humans communicate with each other using not only speech, but also emotions. One of the major ways humans communicate emotions is through facial expressions. Therefore, recognizing facial expressions is a very important part of human-machine interaction and its applications are endless [3]. The plethora of applications include robot-human interactions, targeted advertisement, surveillance, etc. Real-time expression recognition has myriad exciting applications : An autonomous vehicle changing music, speed and lighting based on its owner's emotions, a real-time analysis of customers satisfaction at restaurants, etc.

However complex, some human expressions have been ubiquitous and have not changed since thousands of years. Most datasets use the same classification of emotions as used by Ekman et al [6]. These 6 emotions include Anger, Disgust, Fear, Happiness, Sadness and Surprise. Many datasets includes a 7th emotion, Neutral, for unclassified results or as the starting point of a transition to other emotions. There have been significant advancements in facial recognition and expression recognition using feature extractions and various classifications. Creating an automated and fast system is a challenge [12] owing to the lack of extensive datasets, variance in different datasets, illumination differences as well as latency due to preprocessing and classification times.

Convolutional Neural Networks facilitate automatic feature extractions and promise robust results. Motivation behind this project is the endless applications of such a system as well as to experiment with various depth of CNNs and face bounding box recognition techniques, reaching a balance between accuracy and real-time processing feasibility. I visualise filters and outputs of various CNN layers as well as confusion matrices to understand what the network learns and what confuses it.

2. Background and Related Work

In face and expression recognition, the focus recently shifted from traditional machine learning techniques like Bayesian Classifiers and SVMs to Convolutional Neural Networks. In 2016, A. Mollahosseini et al used CNNs on 7 data-sets DISFA, CK+, FERA, SFEW, MMI, MultiPIE and FER2013 to achieve state-of-the-art accuracies [11]. P. Barros et al tackled challenges with illumination and face positioning using spatial-temporal hierarchical features [4]. Gil Levi and Tal Hassner used Mapped Binary Patterns with CNNs to counter illuminations variances and trained an ensemble of VGG-16 networks on CASIA-Webface dataset [7].

A motivating implementation of CNN for real-time expression detection of is by S. Oullet [13]. He improvised a game where a subject's facial expressions were captured and the video input stream was used with CNNs. The paper demonstrated the feasibility of using CNNs for real-time face and expression recognition. Dan Duncan et al used a pretrained VGG-16 and fine-tuned on a home-brewed dataset to achieve 57.1% accuracy and created a real-time detection application [5]. Recently, Andre T. Lopes et al focused on various image pre-processing techniques including spatial normalization, intensity normalization and downsampling to reach start-of-the-art accuracy with lesser training time [9].

3. Approach

I used a pretrained VGG-16 network trained on CASIA-Webface dataset provided in Caffe Model Zoo by Levi and Hassner [8]. The model specified 55.3% accuracy on Emotion Recognition in the Wild Challenge 2015. On testing the dataset on JAFFE dataset and CK+ dataset, I could not receive more than 19.13% accuracy even after trimming the faces to face bounding boxes.

I proceeded to create a custom dataset with JAFFE [10], CK+ [14] and a collection of personal photos. In the project, I aim to classify all 7 emotions : Angry, Disgust, Fear, Happy, Neutral, Sad and Surprise. CK+ images dont have a well-defined Neutral emotion and many research approaches use the beginning frame of emotion transitions as Neutral. Carefully analysing the CK+ images, I decided to not include Neutral images from CK+ as the first frame wasnt always Neutral as seen in Fig.3.1c. Moreover, CK+ defines another emotion, Contempt, which I replaced with Disgust. JAFFE and personal dataset adhered to the 7 specified labels. Fig. 3.1a,b show sample images from JAFFE and CK+.

Fig. 3.1a Sample JAFFE Image (Happy)

Fig. 3.1b Sample CK+ Image (Surprise)

Fig. 3.1c Sample CK+ Image (First Frame)

I fine-tuned the EmotiW_VGG_S model on the custom dataset and received an overall 90.4% training accuracy, but 21.3% test accuracy. The model when tested on live camera feed and I found that it was extremely sensitive to camera angle, placement of the face, face inclination and lighting. The model was clearly overfitted to the conditions of the training images. To solve the problem of illumination variations, on the personal dataset, I applied illumination variations to every image to create 6 more illumination variant images with the same emotion label. Fig 3.2 shows the illumination variations.

Fig. 3.2 : Images with illumination variations

The final custom dataset contained 214 images from JAFFE, 328 images from CK+ and 631 personal images including illumination variants. 20% of the dataset was randomly selected and kept aside for testing. The custom dataset composition is shown in Fig. 3.3.

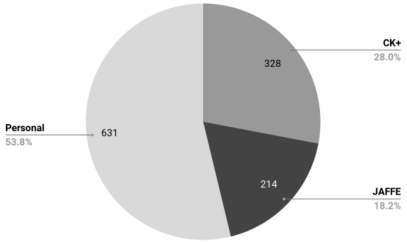

Fig. 3.3 : Custom dataset composition

To improve the quality of training, the images were cropped to face bounding boxes and the face tilts were handled. To find face bounding boxes, I used a pre-trained network provided by K. Zhang et al based on their Joint Face Detection and Alignment using Multi-task Cascaded CNNs [15]. State-of-the-art MTCNN model uses 3 CNN models, P-net, R-net and O-net to find faces using a coarse-to-fine architecture. The standard bounding boxes were further shrunk by 10% to not include any part outside the face. Using the 5 facial points provided by MTCNN, the bounding boxes were rotated to be vertical. Next, the cropped image is converted to grayscale and an adaptive histogram equalization technique, CLAHE is applied. Provided in OpenCV, Contrast Limited Adaptive Histogram Equalization (CLAHE) [1] divides the images into tiles of specified pixels and applies histogram equalization to the tiles. This technique preserves local features. I used a tile side of 8 pixels in the project.The resulting images were resized to 224*224 pixels and converted to 3 channels by copying the single channel values to the other 2. The pre-processing steps for training images are shown in Fig. 3.4.

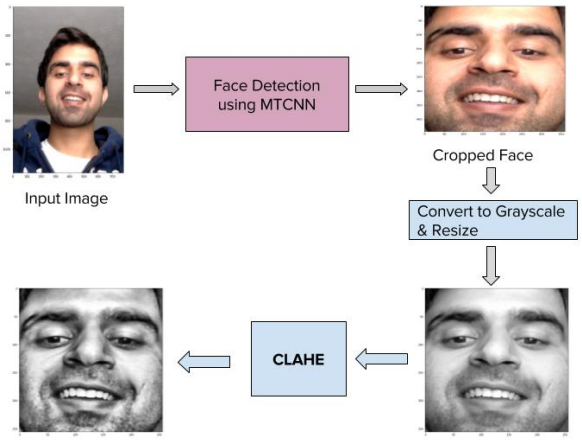

Fig. 3.4 : Image pre-processing pipeline

The processed training images were used to finetune the EmotiW network. The pre-trained EmotiW network has the architecture as shown in Fig. 3.5.

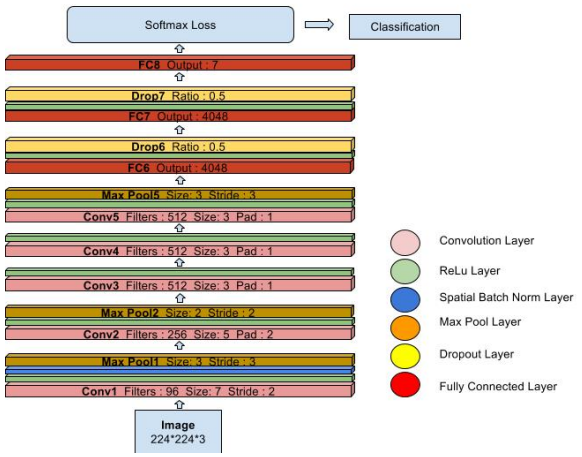

Fig. 3.5 : Architecture of EmotiW CNN

About 10% of training images were set aside as validation images. The network was re-trained for about 30 hours on CPU mode in Caffe on a commodity machine without GPU support. As the training dataset is moderately sized and a batch-size of 250 was used, a jagged profile for loss and training accuracy was observed. Fig 3.6 visualises the loss history as well as training and validation set accuracies which were calculated after every 10 epochs. Training was stopped at 400 epochs and the final test accuracy was nearly 100%, validation accuracy was nearly 90% and loss was below 10e-5.

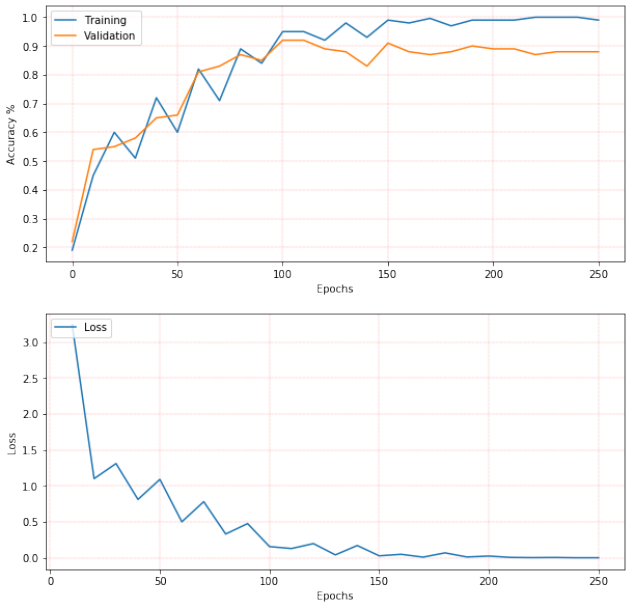

Fig. 3.6 : Training History Visualization

4. Experiments

As mentioned before, the first attempt to re-train the network fetched 90.4% training accuracy and only 21.3% test accuracy. In that attempt the images werent processed in the form as described in Fig. 3.4 and neither were illumination variants added in the training dataset. By cropping the images to a little over the bounding boxes, handling face tilts, converting to grayscale, resizing to a 224*224 size and applying CLAHE, there was a major improvement in the test results. The model received a 75.21% on the test dataset. The training set when tested as a whole was found to have 75.05% accurate. The reason why the training accuracy during training was displayed as nearly 100% is because its the accuracy for that particular batch, not on the entire training data-set. The confusion matrix of models performance on the overall test dataset is shown in Fig. 4.1.

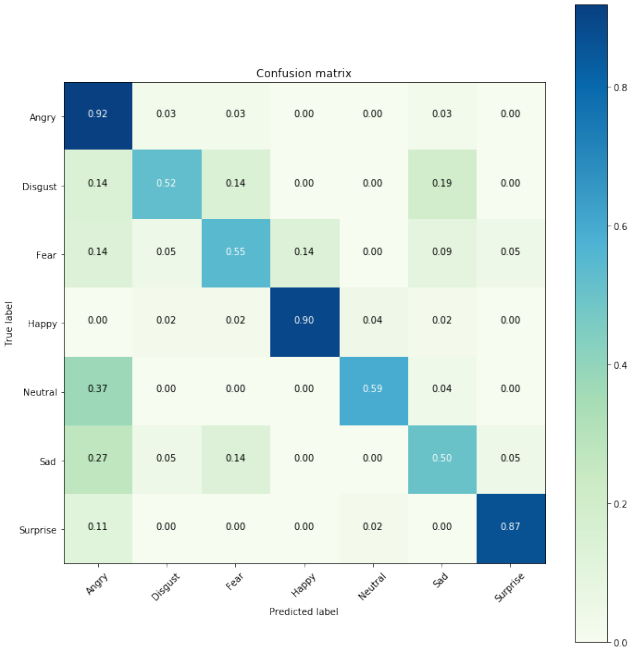

Fig. 4.1 : Confusion Matrix for overall Test set

It can be seen from Fig. 4.1 that the network predicts Happy, Angry and Surprise with nearly 90% accuracy. It is 52% accurate in predicting Disgust, but often confuses it with Sad and Angry. It is 59% accurate in predicting Neutral, but it majorly confuses it with Angry. Sad emotion is the hardest to tell apart for this model and it is only 50% accurate for Sad expression. Next, the model was tested on the test slice of personal dataset alone and received 83.2% accuracy. The confusion matrix for the same is shown in Fig. 4.2.

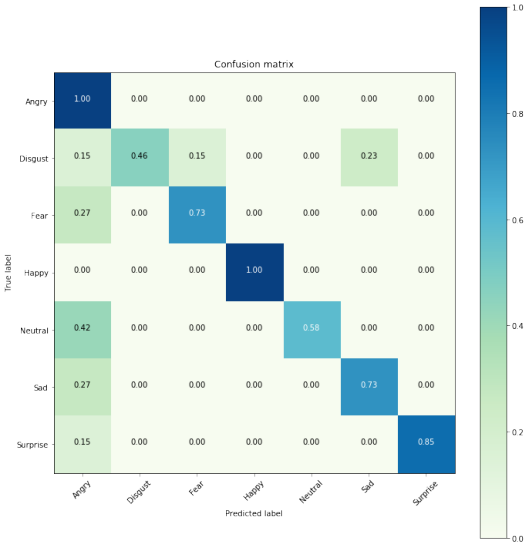

Fig. 4.2 : Confusion Matrix for Test set of personal dataset

For personal test set, the model is 100% accurate for Happy and Angry. It is fairly accurate for Surprise, Sad and Fear. The model confuses Disgust the most with Angry. Moreover, it confuses my Neutral emotion with Angry which is explainable as my Neutral expression is close to Angry. Hence, it is important to have a dataset containing a large number of images of different subjects for a generalised learning. Lastly, the model was tested on CK+ test set and its confusion matrix was analysed. The confusion matrix as seen in Fig 4.3 shows that the model performs poorly for Fear and Sad. As Neutral images were not taken from CK+ as they were not well-defined, the Neutral row in the confusion matrix is not plotted. The test accuracy of CK+ was 78.125%.

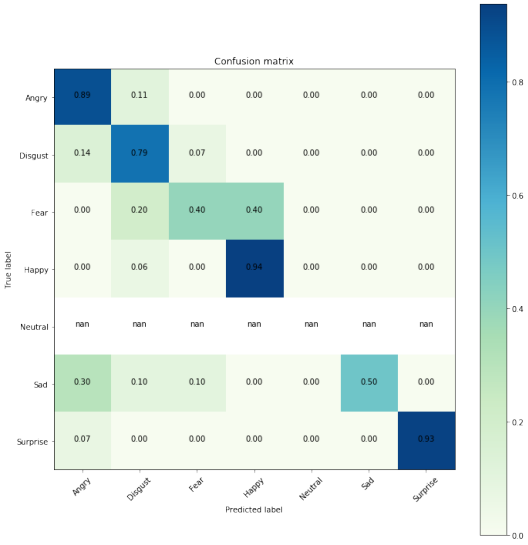

Fig. 4.3 : Confusion Matrix for CK+ Test set

I have visualised the 96 filters in the first Convolutional layer in Fig. 4.4. Fig 4.5 shows the outputs of these 96 filters for an input image that was the processed cropped image in Fig. 3.4.This greatly helps to understand the features that the model has learned to extract for classification.

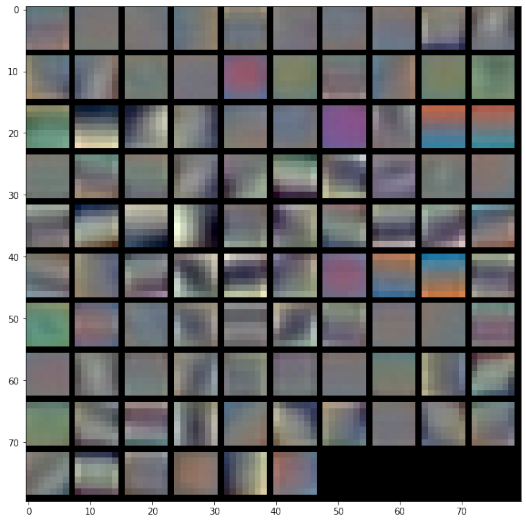

Fig. 4.4 : 96 Filters of the first Convolutional Layer

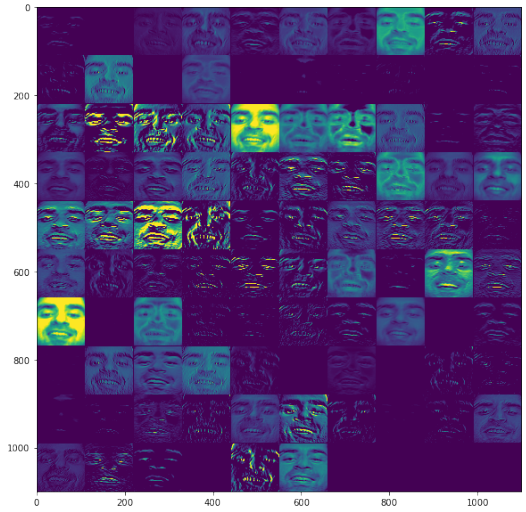

Fig. 4.5 : Outputs of filters of first Convolutional Layer

With the newly trained model ready, I moved on to create a real-time application for expression recognition. The network takes about 0.347 sec/image for classification. Preprocessing the image takes about 0.143 sec/image which includes the face bounding box detection using MTCNN. Another faster approach to find face bounding boxes is to use Haar Cascade Classifiers provided in OpenCV, but the reduced accuracy of Haar Cascades results in inaccurate multiple bounding boxes. These inaccurate boxes consume more time attempting to classify expressions than the time saved, therefore I have used MTCNN, which is a more accurate although slower method. The total classification time is about 0.49 sec/image on a commodity machine.

I use OpenCV for capturing the live frames from the webcam. The individual frames are preprocessed using the pipeline as described in Fig. 3.4. The resulting single or multiple bounding boxes are fed into the trained CNN model for expression recognition. A face bounding rectangle is drawn on the frame and the classified emotion as well as the confidence is plotted. OpenCV captures flipped frames in BGR color-space, the frames are flipped as well as converted to RGB before pre-processing. The process of real-time expression classification is shown in Fig. 4.6

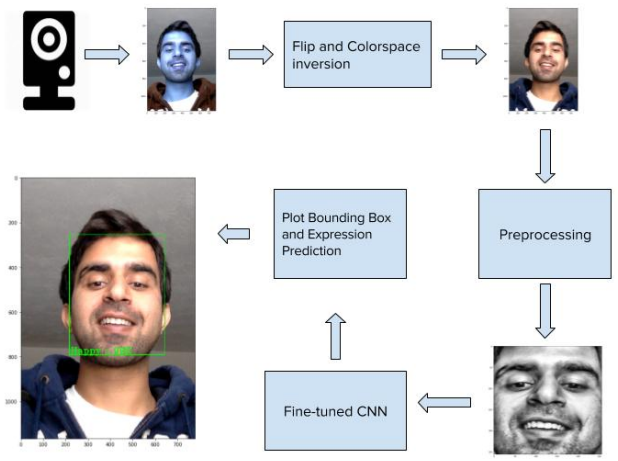

Fig. 4.6 : Real-time Expression detection using OpenCV

For multiple faces in a single frame, multiple bounding boxes are found and classification is done for all the faces. The processing time of 0.49 sec/image is for a single face. With more bounding boxes, the processing speed increases linearly. A demonstration video showing real-time classification of a single and multiple subjects can be seen here: `https://youtu.be/_fRp408fauI`

5. Conclusion

I was successful in implementing a real-time expression recognition system with a processing rate of about 0.4 sec/image including preprocessing. On a commodity machine it delivers 2-3 classified images in a second and the throughput could greatly increase on systems of higher configuration. After various trials of handling illumination variations, face tilts, different ratios of face bounding boxes and histogram equalizations, the final trained model received an overall test accuracy of over 75% and about 87% test accuracy on CK+. As the model was trained on grayscale images, it can work with colored images by converting them to grayscale. Fig. 5.1 shows some correctly classified images.

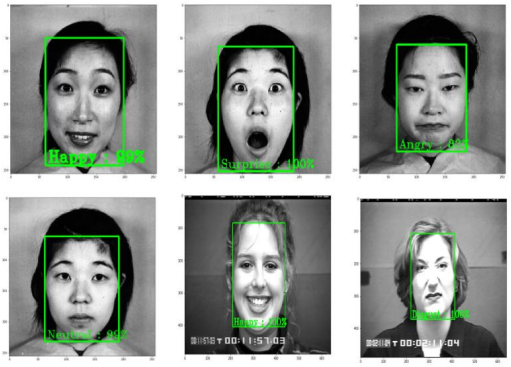

Fig. 5.1 : Classified results from JAFFE and CK+

Although the model has been improved to handle illumination variations and face tilts, it is still sensitive to shadows on the face that get exaggerated due to CLAHE. Moreover, the personal dataset contained images with exaggerated expressions that may not generalise to subtle expressions. A larger dataset and greater training time would possibly train a more robust model. Larger number of subjects would remove the classification skew due to intrinsic facial features of a few subjects. Furthermore, expressions are generally dynamic and there is a transition window of a few frames. Classifying the expressions while they form requires taking a running average of the frames. Using a faster machine, a running average of frames can be used for classifying dynamically forming expressions.

The project greatly improved my understanding of CNNs as well as computer vision. It was challenging to handle the sensitivity of CNN models to various factors as well as getting a balance between accuracy and throughput. I would like to thank the faculty of 682N for their constant support and motivation. The project was developed using Python 2.7, with Caffe. The source code can be downloaded from [2]

References

[1] `https://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.html`.

[2] `https://github.com/anmolsinghsuag/emotion_recognition`.

[3] M. S. W. S. A. Koakowska, A. Landowska and M. R. Wrobel. Human-computer systems interaction: Backgrounds and applications 3, ch. emotion recognition and its applications, 2014.

[4] W. C. Barros, P. and S. Wermter. Emotional expression recognition with a cross-channel convolutional neural network for human-robot interaction. in humanoid robots (humanoids), ieee-ras 15th international conference, 2015.

[5] G. S. Dan Duncan and C. English. Facial emotion recognition in real-time, 2016.

[6] P. Ekman and W. V. Friesen. Emotional facial action coding system. unpublished manuscript, university of california at san francisco, 1983.

[7] G. Levi and T. Hassner. Emotion recognition in the wild via convolutional neural networks and mapped binary patterns, November 2015.

[8] G. Levi and T. Hassner. Emotion recognition in the wild via convolutional neural networks and mapped binary patterns, 2016.

[9] A. T. Lopes, E. de Aguiar, and T. Oliveira-Santos. A facial expression recognition system using convolutional networks, Aug 2015.

[10] M. K. M. J. Lyons and J. Gyoba. "japanese female facial expressions (jaffe)," database of digital images, 1997.

[11] D. C. Mollahosseini, Ali and M. H. Mahoor. "going deeper in facial expression recognition using deep neural networks." applications of computer vision (wacv),, 2016.

[12] I. C. Y. S. T. G. T. S. H. Nicu Sebe, Michael S. Lew. Authentic facial expression analysis. image and vision computing, 2007.

[13] S. Ouellet. real-time emotion recognition for gaming using deep convolutional network features, corr, vol. abs/1408.3750, 2014.

[14] T. K. J. S. Z. A. P. Lucey, J.F. Cohn and I. Matthews. "the extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression", in the proceedings of ieee workshop on cvpr for human communicative behavior analysis, san francisco, usa,", 2010.

[15] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks, Oct 2016.