Face emotion recognition in real time

Jimenez G. Osman D., Niquefa V. Diego S. and Vergara D. Manuel A.

***Abstract*— We trained a Convolutional Neural Network to classify pictures of faces in 7 different categories: Angry, Neutral, Disgust, Fear, Happy, Sad and Surprised. We used transfer learning on the VGG-16 network, removing the output layer and adding 3 new fully connected layers. Using this network and a library to find the bounding boxes of the faces in an image, we implemented a program to detect the emotions of the people on the web-camera video in real time. We used a very small dataset with 517 images and got good results in good lightning conditions, when close to the camera and with exaggerated expressions.**

I. INTRODUCTION

Humans express and recognize emotions as part of our communication and interactions, this is also the case for animals[1]. We humans express emotions in several ways, including: with our face expression, with our tone when talking, with the words we use and with our behavior. We will focus on the first one, face expression, since it is easy to apply machine learning techniques when we have many pictures of human faces labeled with the emotions they show.

II. RELATED WORK

Over the last few years we have a lot of improvements in image recognition. Face emotion recognition is no exception. Advances on deep learning are a big part of the improvements we have had. Deep Convolutional Neural Networks (CNNs) are being used commonly and successfully, for example: top submission to the 2015 Emotions in the Wild (EmotiW 2015) used CNNs[3], [4].

Duncan, Shine and English show that it is possible to apply transfer learning to train a face emotion recognition CNN from CNN that was trained for face recognition, and then use this CNN to do face emotion recognition in real time[2].

III. DATASETS

We collected 517 face pictures with labels corresponding to one of the emotions: Angry, Neutral, Disgust, Fear, Happy, Sad and Surprised. We merged two datasets we found with an original one made by us. These are the datasets:

- Dataset 1: The Extended Cohn-Kanade Dataset (CK+)[5]. A dataset with 326 images labeled with the 7 emotions we used. This dataset was made with the purpose of promote the research in this area.
- Dataset 2: The Japanese Female Facial Expression (JAFFE) Database[6]. The database contains 213 images of the 7 facial expressions posed by 10 Japanese models. We discarded some images we considered did not show the emotions strongly enough or showed different emotions. After this process we ended up with 112 images.

TABLE I
SUMMARY OF DATASET

Emotion	Images on dataset
Angry	57
Neutral	63
Disgust	88
Fear	36
Happy	112
Sad	48
Surprised	113
Total	517

- Dataset 3: Our original dataset, containing 79 images of us and some relatives, labeled by us.

We want to highlight that in the process of labeling Datasets 2 and 3 we often disagreed on what emotions are shown on the pictures, we only kept the pictures on which we agreed on the label. As a result the images used for trained all show very clearly (at least in our opinion) the emotion they are labeled with. This also shows that emotion recognition from just a picture of a face is a hard task even for humans.

IV. MODEL

Since it has been showed that transfer learning works well for the task at and[2], we decided to apply transfer learning. Ideally, we wanted to use a pre-trained CNN for face recognition. But we did not find a small CNN that we could use, it needs to be small since we want to be able to do feedforward very fast to be able to achieve real time emotion recognition. So the CNN we used is the VGG16 CNN[7] which achieves a 7.4% top-5 test error on ImageNet 2012, a competition where you must build a classifier to distinguish between 1000 different types classes of objects [8].

We removed the last layer of the VGG16 model and appended 2 fully connected layers with 256 nodes and the output layer with 7 nodes (one for each emotion we classify). We let all parameters of the model (pre-trained and newly added) to be trainable since this gave much better results. Since we have a very small dataset. We used 90% of the data (469 images) for training and the rest (48 images) for testing. After training we got a 52% accuracy on the test set, although the error on the accuracy might be big since the test set is so small.

V. STATIC IMAGE EMOTION RECOGNITION

Before recognizing emotion in real time (live video), first we need to recognize the emotions for a static image. For this process we use a library for face recognition[9] which finds the bounding boxes of all faces in an image ($faceBoundingBoxes$ method on the pseudocode). Then,

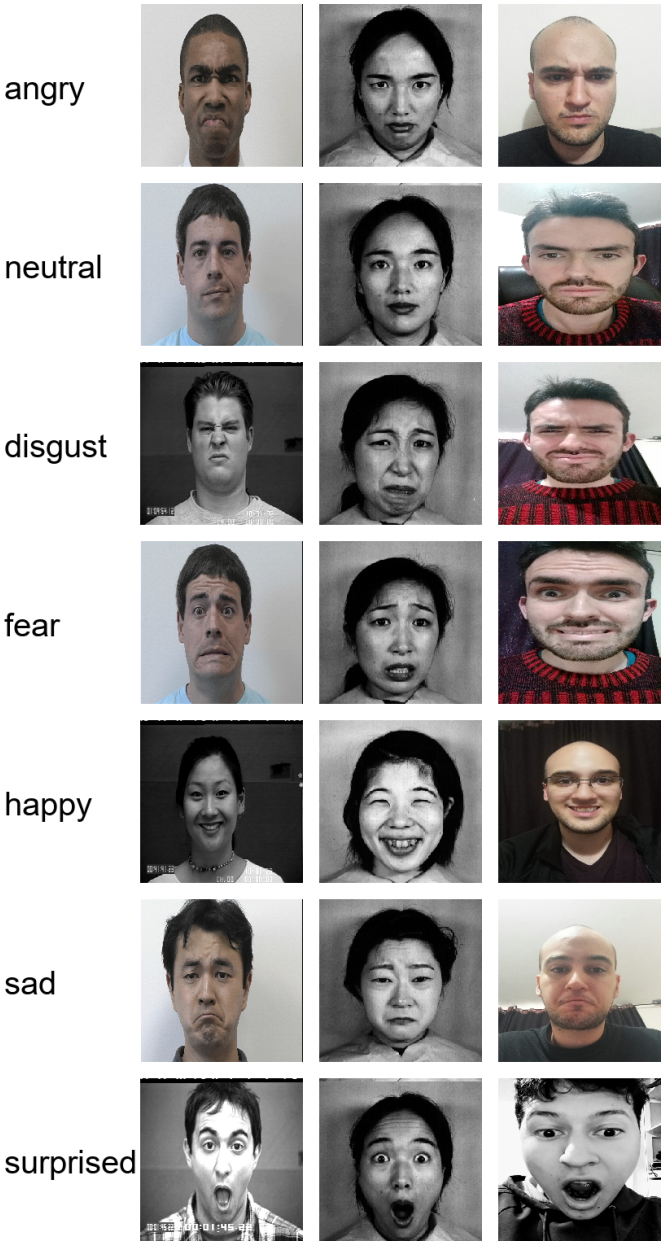

Fig. 1. One example from each dataset for each emotion

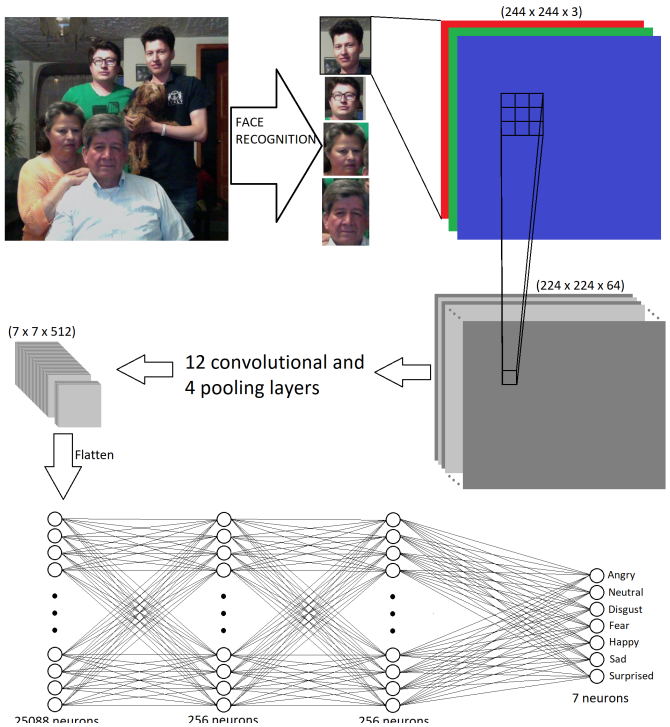

Fig. 2. CNN arquitecture

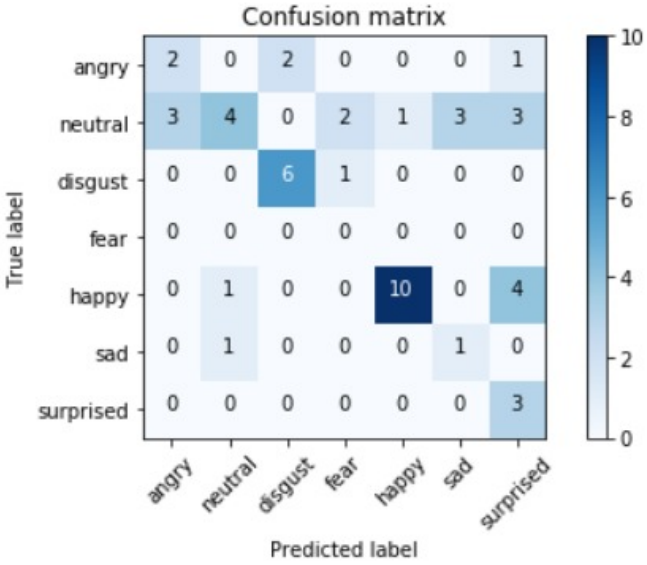

Fig. 3. Confusion matrix for test set

we crop all the faces from the image and use them as inputs to feed-forward our CNN and get the predictions of the emotion of each face in the image.

Algorithm 1 shows a pseudocode of how we get the emotion of all faces given an image. The execution time of Algorithm 1 is roughly $t_1 + nt_2$ where t_1 is the time the library takes to find the bounding boxes of the faces, n is the number of faces found and t_2 is the time our CNN takes to do a feed-forward. The execution time can be considerably less n is big since the library tries to parallelize the feed-forwards if possible, but this is not the case in our test laptop. For our laptop the mean times are: $t_1 = 0.75s$ and $t_2 = 0.55s$.

Algorithm 1 Get emotions from static image

procedure GETEMOTIONS($image$)
  $bboxes \Leftarrow faceBoundingBoxes(image)$
  $emotions \Leftarrow arrayOfCeros(length(boxes))$
  for i `in` $\{0, 1, \ldots, length(boxes) - 1\}$ **do**
    $emotions_i \Leftarrow feedforward(crop(image, bboxes_i))$
  end for
  return emotions
end procedure

VI. REAL TIME EMOTION RECOGNITION

Doing the emotion recognition on real time is a big challenge since it is too hard to match the FPS (of 200 in the case of the laptop we have for testing). As we saw before, given how big t_1 and t_2 are, it is impossible to match the fps of the camera. To give a real time experience we use 2 extra threads in the program:

- Thread 1: This thread runs the library to find the bounding boxes of a frame, as soon as it finishes, it runs the library again on the new current frame. This is an endless cycle that keeps finding the face bounding boxes of the current frame, note that this thread does not find the bounding boxes in every single frame, the number of frames ignored depend on the speed of the computer. So, approximately, every t_1 seconds we get the bounding box of the frame shown t_1 seconds ago, and this bounding boxes will be shown until a new set of bounding boxes is calculated.
- Thread 2: This thread runs feed-forward on our CNN with the images cropped from the last found bounding boxes, as soon as it finishes, it runs again with the current last found bounding boxes. So when this thread finishes we have the labels of the faces in the frame shown $t_1 + nt_2$ second ago.

When Thread 2 finishes finding the emotions of some faces we have this problem: We have the emotions of the faces in the frame shown x seconds ago (frame A), but the bounding boxes being displayed were found y seconds ago ($y \leq x$) (frame B). The number of bounding boxes in frame A and frame B might even be different. To solve this problem, every time a new set of bounding boxes we find the best matching of the previous bounding boxes with the new ones and keep track of how the frames are moving. Finding the best matching can be formulated as a classical optimization problem, the assignment problem, and is solved by the Hungarian algorithm[10].

Let A_i be the center of the i-th bounding box in frame A, and B_i the center of the i-th bounding box in frame B. Then if let matrix C be such that $C_{ij} = \|A_i - B_i\|$. Solving the assignment problem for matrix C gives us the best matching between the bounding boxes on frame A and the bounding boxes on frame B. With this method we keep track of how the faces are moving every time a new set of bounding boxes is found. And when we finish finding the emotion of a set of faces in an old frame we can show these results in a newer positions of the faces. We used the Hungarian Algorithm implementation by Dedecko[11].

We uploaded our implementation to GitHub [12].

VII. CONCLUSIONS

We achieved our goal of implementing a real time face emotion recognition software, but the prediction accuracy can definitely be improved.

As we test the problem we noticed that the CNN is most accurate at recognizing a happy face and a surprised face. Happy and surprised are the two classes with more data. So in order to improve this classifier a bigger database will probably be very helpful. Another important observation is that sometimes the misclassification of a face, especially of happy and surprised, can be solved by improving the lightning. This problem has been addressed before and can be solved by adding more images to the dataset with different lightnings. Taking all these into account we believe the most important thing to improve the accuracy of our model is a to use much bigger dataset.

REFERENCES

[1] Hebb, D. O. (1946). Emotion in man and animal: an analysis of the intuitive processes of recognition. Psychological review, 53(2), 88.

[2] Duncan, D., Shine, G., & English, C. Facial emotion recognition in real time.

[3] Yu, Z., & Zhang, C. (2015, November). Image based static facial expression recognition with multiple deep network learning. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (pp. 435-442). ACM.

[4] Kim, B. K., Roh, J., Dong, S. Y., & Lee, S. Y. (2016). Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. Journal on Multimodal User Interfaces, 10(2), 173-189.

[5] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010, June). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on (pp. 94-101). IEEE.

[6] Psychology Department of Kyushu University. The Japanese Female Facial Expression (JAFFE) Database. Retrieved from http://www.kasrl.org/jaffe.html

[7] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

[8] ImageNet. Large Scale Visual Recognition Challenge 2012. http://www.image-net.org/challenges/LSVRC/2012/

[9] Geitgey, A. Face Recognition. GitHub repository. Retrieved from https://github.com/ageitgey/face_recognition

[10] Bruff, D. (2005). The assignment problem and the hungarian method. Notes for Math, 20, 29-47.

[11] Dedecko, T. Hungarian Algorithm. GitHub repository. Retrieved from https://github.com/tdedecko/hungarian-algorithm

[12] Jimenez O., Niquefa D. & Vergara M. facEmotion. GitHub repository. https://github.com/mavd09/facEmotion