FACIAL EMOTION RECOGNITION USING DEEP LEARNING

Nadendla Venkata Guru Swapna [1], Ponakala Priyanka[2,] Marella Swathi[3], Murari DivyaPrasanna[4], P Prashant[5]
[1,2,3,4] B.Tech, computer Science, Vasireddy Venkatadri Institute of Technology, Andhra Pradesh, India
[5] Associate Professor, Computer Science, VasireddyVenkatadri Institute of Technology, Andhra Pradesh, India

Abstract:

The use of machines to perform different tasks is constantly increasing in society. Providing machines with perception can lead them to perform a great variety of tasks; even very complex ones such as elderly care. Machine perception requires that machines understand about their environment and interlocutor's intention. Recognizing facial emotions might help in this regard. During the development of this work, deep learning techniques have been used over images displaying the following facial emotions: happiness, sadness, anger, surprise, disgust, and fear. In this research, a pure convolutional neural network approach outperformed other statistical methods' results achieved by other authors that include feature engineering. Utilizing convolutional networks involves feature learning; which sounds very promising for this task where defining features is not trivial.

Keywords: Machine learning, Python Programing, OpenCV, Facial emotions

1. INTRODUCTION

The use of machines in society has increased widely in the last decades. Nowadays, machines are used in many different industries. As their exposure with humans increase, the interaction also has to become smoother and more natural. In order to achieve this, machines have to be provided with a capability that let them understand the surrounding environment. Specially, the intentions of a human being.

When machines are referred, this term comprises to computers and robots. A distinction between both is that robots involve interaction abilities into a more advanced extent since their design involves some degree of autonomy. When machines are able to appreciate their surroundings, some sort of machine perception has been developed. Humans use their senses to gain insights about their environment. Therefore, machine perception aims to mimic human senses in order to interact with their environment. Nowadays, machines have several ways to capture their environment state trough cameras and sensors. Hence, using this information with suitable algorithms allow to generate machine perception. In the last years, the use of Deep Learning algorithms has been proven to be very successful in this regard. For instance, Jeremy Howard showed on his Brussels 2014 TEDx's talk how computers trained using deep learning techniques were able to achieve some amazing tasks. These tasks include the ability to learn Chinese language, to recognize objects in images and to help on medical diagnosis.

Affective computing claims that emotion detection is necessary for machines to better serve their purpose. For example, the use of robots in areas such as elderly care or as porters in hospitals demands a deep understanding of the environment. Facial emotions deliver information about the subject's inner state [74]. If a machine is able to obtain a sequence of facial images, then the use of deep learning techniques would help machines to be aware of their interlocutor's mood. In this context, deep learning has the potential to become a key factor to build better interaction between humans and machines, while providing machines with some kind of self-awareness about its human peers, and how to improve its communication with natural intelligence.

Figure 1: Example of primary universal emotions.

2. RELATED WORK

Affectiva is the worlds leading commercial research group on emotion recognition. Its current patent portfolio is the largest, compared to startups in this field. Their research has adopted deep learning methodologies since its private corpus consists of 3.2 million facial videos. Also, their data gathering has been done in 75 countries, which prevents the research to fall on cultural or regional behaviors. In order to measure its detector accuracy, the area under a Receiver Operating Characteristic (ROC) curve is used. The value of ROC score ranges between 0 and 1. The classifier is more accurate when the value is closer to 1. Some

emotions such as joy, disgust, contempt, and surprise have a score greater than 0.8. While expressions such as anger, sadness, and fear achieve a lower accuracy since they are more nuanced and subtle. Moreover, Affectiva has been able to successfully identify facial action units on spontaneous facial expressions without using deep learning techniques.

Kotsia et al. focused on the effect of occlusion when classifying 6 facial emotion expressions. In order to achieve this, several feature engineering techniques and classification models were combined. Gabor features, which is a linear filter used for edge detection, and Discriminant Non-negative Matrix Factorization (DNMF), which focuses on the non-negativity of the data to be handled, are the feature extractors techniques. To classifiy these features multiclass support vector machine (SVM) and multi-layer perceptron (MLP) were used. The results over Cohn-Kanade are the following: Using a MLP with Gabor 91.6% and with DNMF: 86.7%. While using SVM achieved 91.4%. Another corpus used was JAFFE: Gabor combined with MLP achieved 88.1% and when using it with DNMF, it resulted on 85.2% classification accuracy.

3. PROPOSED METHOD

In this work, automatic facial expression recognition using convolution neural network features is investigated. Here, we are using one publicly available dataset FER2013 to carry out the experiment. Pre-processing step involves face detection for the above dataset. The frontal faces are detected and cropped using OpenCV. Then facial features are extracted using the CNN framework.

4. PREPROCESSING

In general scenario, human vision system, first detects the faces, and then subsequently it recognizes the emotion associated with that face. In the same way, in this work, face detection is the pre-processing or prior work of the emotion recognition task. Face detection task has been done using Viola Jones algorithm.

Haar Feature-Based Cascade Classifier is applied on all the images. This forms a bounding box around the face in the images. The area inside the bounding box is cropped and reshaped into 48 × 48 pixels. After pre-processing, the dataset consists of 11,246 images of the 7 emotions of which 1456 are angry, 240 are disgust, 1414 fear, 3235 happy, 1304 sad, 1362 surprise and 2235 are neutral. All the images are of frontal face. Non frontal faces and non-relevant images were removed.

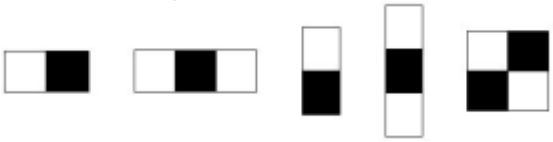

Figure 2(a): The 5 types of Haar-like templates; the value of each rectangle feature is computed by subtracting the sum of the black area, from the white area

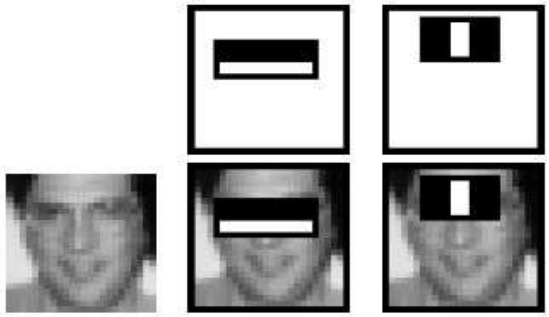

Figure 2(b): Method of applying the rectangle features on the 48x48pixels image of the face.

5. CONVOLUTIONAL NEURAL NETWORK

Convolution Neural Networks have the most influential innovations in the field of computer vision. It is biologically inspired from visual cortex and imitates the working of human brain for visual analysis.

All the networks described here are programmed using Keras, a deep learning python library on Tensorflow in the backend. This facilitated faster and easier experimentation. Convolutional Neural Network architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. The image is fed into the network and then the network analyses the features of the image. A brief description of the layers used in convolution neural network is:

Input Layer:

Input layer has the raw pixel values of the image as (w × h × c) where h and w are the height and width of the image and c represents the number of colour channels. Since the dimensions are fixed, pre-processing needs to be done before feeding the pixels in the input layer.

Convolutional Layer:

Convolution layer computes the dot product between the weights and a small region to which the neurons are connected to in the input layer. The number of filters is passed as one of the hyper parameters which are unique with randomly generated weights. The filter also called a kernel is convolved with image.

This generates a feature map that acts as feature map that acts as feature identifiers sensitive to the edges and the orientations that represent how the pixel values are enhanced. This result in (w × h × f), where f refers to the number of filters used.

Convolutional layers are followed by a pooling layer that down samples the dimensions along the width and the height to reduce the computational time due to a large number of convolutional layers. MaxPooling is used that reduces the dimensions of the map by a factor of window size and only the maximum pixel value in the original feature map window is retained.

Dense Layer (Fully Connected Layer):

This layer is fully connected with the output of the previous layer. These are typically used in the last stages of the convolution neural network to connect to the output layer and construct the desired

number of outputs. It transforms the features through layers connected with trainable weights. Sophisticated features in the image are identified by this layer that brings out the entire image.

Output Layer:

Output layer is connected to the previous dense layer and outputs the required classes or their probabilities.

Training process of the Convolution Neural Network:

Step1: All filters and weights with random values are initialized here.

Step2: A training image is taken by this network as input, goes through the forward propagation step (convolution, ReLU and pooling operations along with forward propagation in the Fully Connected layer) and finds the output probabilities for each class.

Step3: Calculate the total error at the output layer (summation over all 4 classes).

- *Total Error = ∑ ½ (target probability – output probability) ²*

Step4: To calculate the gradients of the error with respect to all weights in the network use Back propagation. To update all filter values and parameter values to minimize the output error use gradient descent.

Step5: Repeat steps 2-4 with all images in the training set.

The above steps train the Convolution neural network this essentially means that all the weights and parameters of the convolution neural network have now been optimized to correctly classify images from the training set.

When a new (unseen) image is input into the convolution neural network, the network would go through the forward propagation step and output a probability for each class (for a new image, the output probabilities are calculated using the weights which have been optimized to correctly classify all the previous training examples).

Here first we give a live video as input then the video is divided into frames. Each frame is converted into gray scale images. These gray scale images are giving as inputs to the CNN. In CNN the process for emotion detection is done. Then output image is displayed in another window with its emotion on the top of the face.

6. CONCLUSION

In this project, a research to classify facial emotions over static facial images using deep learning techniques was developed. This is a complex problem that has already been approached several times with different techniques. While good results have been achieved using feature engineering, this project focused on feature learning, which is one of DL promises. While the results achieved were not state-of-the-art, they were slightly better than other techniques including feature engineering. It means that eventually DL techniques will be able to solve this problem given an enough amount of labeled examples. While feature engineering is not necessary, image pre-processing boosts classification accuracy. Hence, it reduces noise on the input data.

REFERENCES

[1] *Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems. pages 1097–1105, and 2012.*

[2] Geoffrey Hinton et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine, IEEE, 29(6):82–97, 2012.

[3] https://arxiv.org/ftp/arxiv/papers/1701/1701.08257.pdf

[4] Andrej Karpathy. Stanford university: Cs231n convolutional neural networks for visual recognition. http://cs231n.github.io/neural-networks-1, 2016. Online; Accessed 07 June 2016.

[5] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar and Li Fei-Fei. Large-scale video classication with convolutional neural networks. In Proceedings of International Computer Vision and Pattern Recognition (CVPR 2014), 2014.