Facial Emotion Recognition Using Convolutional Neural Network

Aswin Matthews Ashok, Dept. of Electrical and Computer Engineering, University of Florida, aashok@ufl.edu

***Abstract*—Facial emotion recognition is an integral part of psychology, forensics and social media. In all these fields, a high degree of reliability of classification is imperative. Currently human judgement is used in these fields but humans are not always accurate. Therefore, there is a need for a reliable and quick way of identifying human emotions.**

The recent advances in machine learning and pattern recognition has offered several algorithms to recognize human emotions. One such algorithms is the Convolutional Neural Network (CNN). The CNN is capable of high-speed image processing with excellent reliability. In this project one such CNN is used to recognize facial emotions. It is seen that with proper training this method can yield very high accuracy of classification.

I. Introduction

THERE are a total of seven human emotions which can be identified from facial expressions. They are anger, fear, disgust, happiness, sadness, surprise and contempt. Recognition of such emotions is valuable in several fields and is currently done manually. But human judgement can be flawed in many cases. Errors can occur due to various factors like insufficient knowledge of facial expressions and their meaning, biased judgement, stress and the monotonous nature of the work. Such misinterpretations can have very severe results depending on the field. Therefore, it is necessary to come up with an automated solution to this problem with minimal errors, high speed and reproducible results.

Facial emotion recognition has long been a topic of study in machine learning and deep learning. Several machine learning algorithms such as Support Vector Machines (SVM) [1], Naïve Bayes and Maximum entropy have been applied to detect human facial emotions. Deep learning algorithms such as Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) [2] have also been used in the field of facial emotion recognition. In this project the Convolutional Neural Network has been used in order to perform facial emotion recognition.

Convolutional Neural Networks offer various advantages over an Artificial neural network for image processing applications. The CNN is a partially-connected network where each neuron is connected to only a subset of neurons from the previous layer. This subset of neurons is from a small region in the image. The key advantage in this approach is that since every neuron is connected to a small region from the previous layer, the spatial properties of the image are often maintained. This enables the network to easily identify patterns in an image. The second important advantage of a CNN over an ANN is that the number of training parameters for the CNN is much lesser due to the fact that each neuron has fewer connections with the previous layer this greatly improves training and classification speed and also combats the problem of overfitting.

In this project the CNN is used to analyze the complex facial expressions, analyze key patterns which correspond to the seven facial emotions, perform statistical analysis and determine the facial emotion which has the highest probability. An eighth class called Neutral is used in this project in order to handle scenarios where the facial emotion cannot be identified with a good degree of certainty.

II. Project Description

A. Frontal Face Image Database

This project requires a frontal face image dataset with a wide variety of facial expressions which correspond to the seven emotions. For this purpose, the Cohn-Kanade AU-Coded Expression Database [3, 4] was used. This database is one of the leading databases used for facial emotion recognition. It contains subjects with several facial features and varying degree of facial expressions classified into the seven emotion which this project aims to recognize.

B. Data Pre-processing

Data which was downloaded from the Cohn-Kanade AU-Coded Expression Database was further processed to better suit the requirements of the project in the following ways.

Face Detection

The CNN used in this project aimed to classify facial emotions and required only face portion of the images in the dataset. This cannot be done manually since the dataset contained thousands of images and it was also necessary to be able to automatically detect faces in the case of real time applications. In order to quickly detect and extract the face portion of the images, an LBP Cascade Classifier available in OpenCV was used. This classifier offers high speed

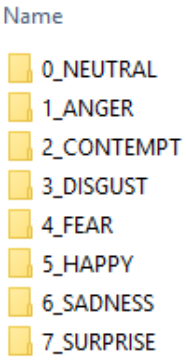
classification and sufficient accuracy for real-time applications.

Manual Verification of Data

In order to ensure best results, the data must be manually inspected to find and remove any flaws before training the CNN. To facilitate this process, the detected faces were resized to 100X100 images and stored in a separate output location under directories named after the different emotions. In addition to this, a directory for neutral images were also created to house those faces whose expressions were deemed to be unclear.

Once the programs which populated these directories with faces were executed, the faces were manually inspected. The false positives from the LBT Cascade Classifier were removed. Faces with unclear emotions were moved to the neutral folder.

Figure 1: Directories Containing Labeled Faces



Data and Label Files

After a thorough inspection it was found that some of the emotion directories contained fewer specimens than others. This could adversely affect stability of the CNN and will cause misclassifications. The classifier will be more prone to identify faces as belonging to the category which had the greatest number of samples during training. In order to overcome this problem, virtual sampling was done. Virtual sampling is a technique which improves the representation of classes with insufficient samples through re-sampling and adding some noise to the additional samples.

While training the network it will be highly inefficient to access the image files every time since this would involve system calls to access every image. To improve training speed, the images in all the eight emotion directories were consolidated into a file named as data.npy and their corresponding emotion labels were saved into a file called labels.npy. It is to be noted that in a gray scale image, the three RGB values of a pixel are always equal. Therefore, to remove this redundancy, only one of the three basic color values are stored in the data.npy file for each pixel.

Data Pre-processing Modules

To achieve the above-mentioned functionality three python modules were created.

The faceDetectorInterface.py module was made to leverage the LBP Cascade Classifier for frontal face detection available in OpenCV in order to detect the human faces in the given images. The returned images are grayscale images.

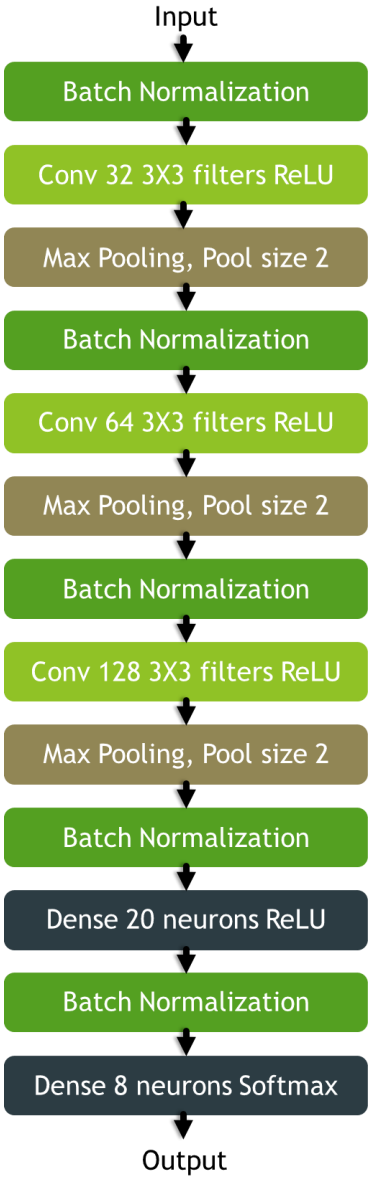
The ckPlusInterface.py module was built to navigate the CK+ dataset, use the faceDetectorInterface.py module to detect faces in the dataset and store the detected faces in separate output location under the appropriate emotion directories.

The dataInterface.py module was created to allow the user to control the data preprocessing steps and also to create the data.npy and labels.npy files.

C. Neural Network Design

The Neural network consists of three convolutional layers and two fully connected dense layers. Each of the convolutional layers used filters of dimensions 3X3. The entire network was designed using Keras a python package which uses Tensor Flow as the backend package. The structure of the network is as seen in Figure 2.

Figure 2: Layers in the Neural Network



Convolutional Layers

The first convolutional layer was connected to the input and it used a total of 32 filters, the second layer consisted of 64 filters and the third filter consisted of 128 filters. They all used ReLU as the activation function. Each of the convolutional layers were preceded by a batch normalization layer and succeeded by a max pooling layer.

Max Pooling

Max pooling is the process of reducing image size by substituting a group of matrix elements with a single element whose value is equal to the maximum value in the group. Max pooling was used in this project to reduce the size of the convolutional layer output matrices thus boosting training and prediction speed.

Dense Layers

The dense layers are fully connected layers which were connected after the third convolutional layer. The data was flattened before being sent to the dense layers. There was a total of two dense layers. The first dense layer contained 20 neurons each of which used ReLU activation function. The second dense layer contained 8 neurons with SoftMax activation. Each of the dense layers were preceded by a batch normalization layer. The second dense layer was the output layer and each neuron represent one of the eight classes.

Batch Normalization

The batch normalization layers are extremely vital for to the stability of the network. Without batch normalization, the network will train on high input values especially with the ReLU activation function. This will in turn result in large weights. If the network has large weights then even small changes in the input caused by noise will be amplified across the layers thus making the predictions extremely unstable. The batch normalization ensures that the data is normalized within a range of 0 and 1. This will in turn result in smaller network weights. Thus, noise will not be amplified and the network predictions will be stable.

Training and Saving the Network

The data and the corresponding labels were imported from the data.npy and labels.npy files respectively. The data and the labels were shuffled and split into Training, Testing and Validation sets in the ration 80:10:10 respectively. While training, the training data and the validation data were used. The training data was used to train the values of the network parameters while the validation was used to measure accuracy after each epoch in order to prevent overfitting. After the training was completed the model was tested on the unknown test data and the performance was measured. Finally, the model was saved so that it can be used for predictions later.

Network Modules

The emotionRecognitionNetwork.py module is built to create and train the network according to given specification. This module also stores the network in a file and exposes interfaces for predictions to the cnnInterface.py module.

The cnnInterface.py module is used to specify network configurations, test the network and assess its performance and to expose prediction interface to the applications. This module also does the test, train and validation data split.

D. Applications

To demonstrate the functionality of the CNN, three applications were developed. Each of these applications obtain images from different sources, detects the faces in them using the faceDetectorInterface.py module, predicts the emotion of the detected faces using the cnnInterface.py module, reformats the image to display the predictions and displays the reformatted image on the screen. One application handles image from the webcam, another application handles images from a video file and the third application handles the images from the computer screen. All three application codes are in the applicationInterface.py module.

The masterControl.py module is used to provide a centralized user interface through which the entire project can be handled.

III. Evaluation

After the CNN was built, the network was put through several tests to comprehensively evaluate the performance of the network. These tests and the corresponding outcome are explained in detail below.

Training Statistics

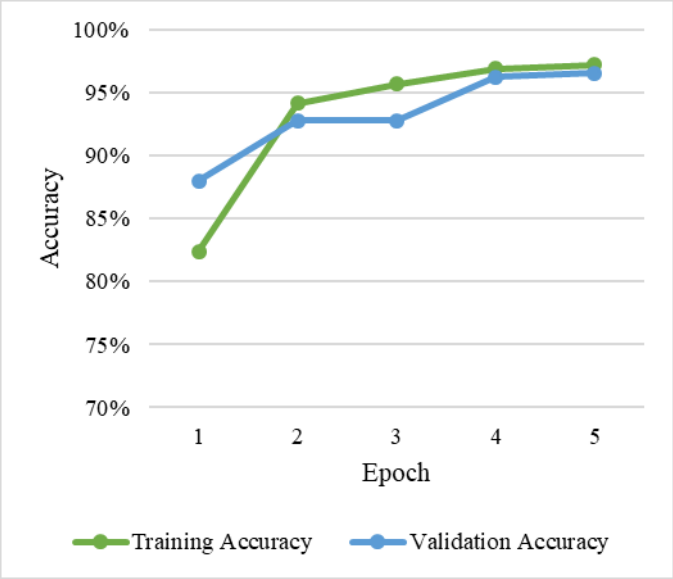
Proper training of the CNN is of paramount importance if the network is to have high prediction accuracy on unknown dataset. While training the network may begin to memorize the samples while omitting the key features which are common to the samples in order to improve accuracy on the training data. This causes the network to have poor prediction accuracy on unknown data. This may occur due to overtraining, insufficient training data or too may trainable variables in the network. This phenomenon is called overfitting.

In order to avoid overfitting, it is essential to keep track of the network's performance on unknown data and continue training only if there is an improvement in performance on both the training and the unknown data. For this purpose, 10% of the data was reserved as validation data and the network's performance on training and validation data after each epoch.

Table 1: Accuracy on Training and Validation Data After Each Epoch

Epoch	Training Accuracy	Validation Accuracy
1	82.40%	88.02%
2	94.14%	92.77%
3	95.71%	92.77%
4	96.89%	96.24%
5	97.22%	96.53%

Figure 3: Plot showing the Accuracy on Training and Validation Data After Each Epoch



Tests on Native Dataset

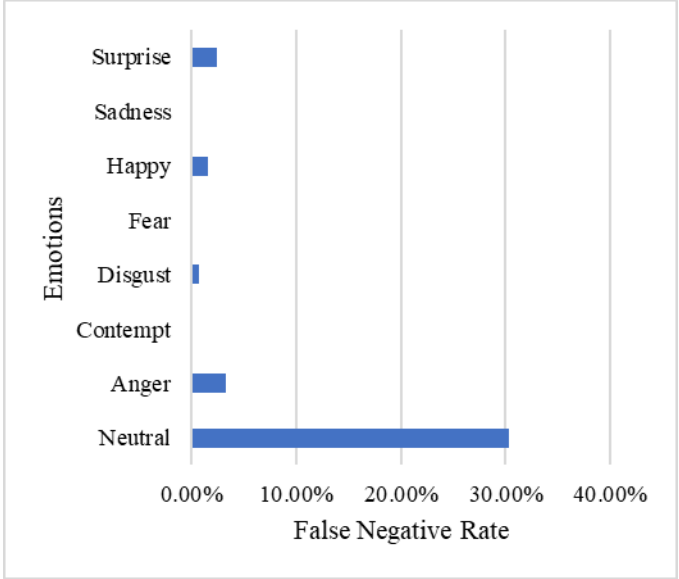
Before training the CNN the data 10% of the data from the CK+ database was reserved for testing purposes. This data was never used during any part of the network training process. Therefore, the performance of the network on this data proved as an excellent way of assessing the ability of the network to learn the critical facial features for emotion recognition without overfitting. The trained CNN network was used on the test data to predict the emotions. The output from the CNN contained the confidence of the prediction for each of the images. The emotion which was predicted with the highest confidence value was identified for each image and the results were compared with the ground truth. Based on this prediction, a confusion matrix was built and the overall accuracy of classification was determined.

Figure 4: Confusion Matrix

		Predictions							
		Neutral	Anger	Contempt	Disgust	Fear	Happy	Sadness	Surprise
Actual	Neutral	94	3	1	14	2	8	5	8
	Anger	1	118	0	0	0	0	3	0
	Contempt	0	0	144	0	0	0	0	0
	Disgust	0	0	0	133	0	0	0	1
	Fear	0	0	0	0	100	0	0	0
	Happy	0	1	0	0	0	126	1	0
	Sadness	0	0	0	0	0	0	127	0
	Surprise	1	0	0	2	0	0	0	118

Based on the confusions matrix it is easily seen that the classifier performs extremely well on the seven original. The Neutral emotion category is the main source of errors. This particular category contains several false negatives. Therefore, a plot of the false negative rates for each class was created as seen in Figure 5.

Figure 5: Plot Showing the False Negative Rate for Each Class



The reason for the abnormally high false negatives on the Neutral class is that the Neutral class was a manually created class which contains faces with unclear or subtle emotions. There may be several miss classifications due to human error in this particular class since it is solely based on the opinions of a single person. The trained CNN is able to correctly identify several faces depicting valid emotions even among the Neutral images which are ambiguous to us. Further analysis and proper classification of the Neutral faces can yield much better results. The current overall accuracy of the CNN on unknown data from the native dataset is 94.96%. With proper classification of the faces misclassified due to human error could yield an accuracy of more than 97%.

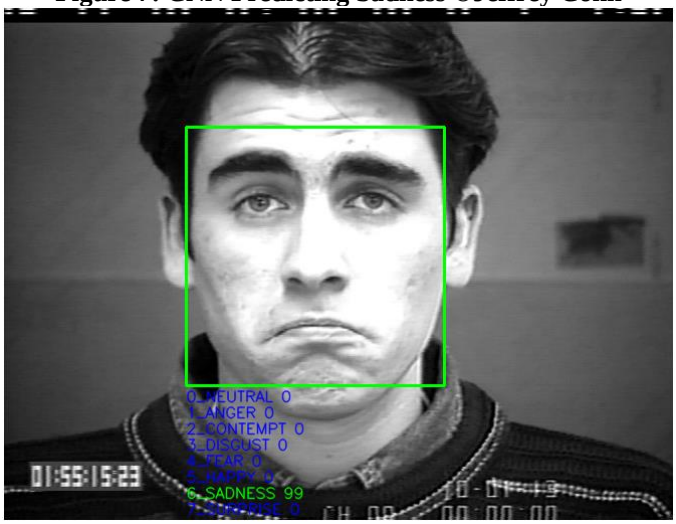
Performance on Real-world Applications

The truest indication of a well-built system is its ability to solve real world problems. This holds true in the case of our CNN which was built with real world applications in mind. The LBP Cascade Classifier used for face detection ensures that the faces in the images are detected with speed and reliability. The well trained and simple CNN guarantees that the emotions of the detected faces will be classified with ease. Therefor this setup is well suited for real world applications were speed and accuracy of classification is of paramount importance.

In this section we will discuss the ability of the network to classify images in real time. The CNN network was used to classify several faces. The network performed very well on images seen on the computer screen. The confidence of the network was very high for each classification and it was able to perform fairly well even if the scale of the image is altered within reasonable limits. It also worked well for images with varying illumination. Similar results were seen in images from webcam feeds and videos.

Figure 6: CNN Predicting of Anger ©Jeffrey Cohn

Figure 7: CNN Predicting Sadness ©Jeffrey Cohn



The CNN despite its good performance on unknown data has a lot of room for improvement. It is to be noted that the training dataset of the network primarily consisted of posed images. Therefore, the CNN has some difficulty identifying candid facial emotions in everyday images. The CNN categorizes people wearing glasses as having the expression of disgust. This is due to the fact that the nose piece between the eyes is identical to the wrinkled skin found in the same area. Similarly, the training dataset does not contain any subjects with beard or mustache. Therefore, the CNN is less confident while predicting the emotions of such subjects. These problems in the CNN network are easily remedied by altering the training data to contain more diverse samples. Therefor, it cannot be said that these are flaws in the approach of using for facial emotion recognition.

Another notable flaw is that the CNN was not very confident about facial emotions while classifying the emotions of faces in a video. This is because the images were non-posed and there was lip movement in most of the images since the subjects were trying to speak. This issue can be mitigated by combining the predictions from facial emotion recognition with those of vocal emotion recognition systems. But vocal emotion recognition is beyond the scope of this project.

IV. Background And Related Work

There are two main steps involved in automatic facial emotion recognition. They are detection of faces in an image and the recognition of facial emotions from the detected image. Detection of faces have been done through several ways in the past. Some of these approaches involve identification of certain landmark features and their geometric orientation in order to detect the faces in an image. The use of Local Binary Patterns in order to detect these landmark features is discussed in the paper by Hadid *et al.* [5] In this paper a new approach to detect facial features through overlapping sections of the image is discussed. This approach allows very fast detection of faces in low resolution images which use relatively short feature vectors as opposed to the longer vectors used in approaches using non-overlapping sections of images.

There are several techniques used to solve the problem of automated facial emotion recognition. Most algorithms follow one of two basic approach. The first approach uses the Facial Action Coding System (FACS) suggested in the Ekman *et al.* [6, 7]. FACS labels each human facial movement which when combined forms an expression. These labels are called Action Units (AU). Certain such sequences of AUs are unique to a particular emotion. Many algorithms have been designed to identify these AU's in a given facial image. The combination of AUs and their intensities are then used to determine the facial emotion. The second technique is to directly label the faces as a whole and allowing the classifier to learn the distinguishing features. This approach requires minimum human intervention and is well suited for implementations which use neural networks to classify the images. In this project, we have used the second approach to classify the images by facial emotions.

Deep neural networks have been used to come up with several solutions for automated facial emotion recognition. The Artificial Neural Network is a network of artificial neurons which are connected. Each neuron is activated under a certain condition and their outputs are dictated by activation functions. The network is trained to learn parameters which determine which inputs from the previous layer affects the neuron in a particular layer. Then these learned parameters are applied upon any input and the activation of certain neurons are used to classify the input.

Deep network is an ANN which contains several hidden layers. This allows the network to learn complex features and characteristics. Deep networks have been used extensively for image classifications since the required processing speed to perform the complex computations of a deep network is currently available. The main issue when using an ANN is that it simply requires very high computation power and in the case of image processing applications, they are simply not fast

enough to solve real world problems. It should also be noted that each neuron in an ANN is connected to every neuron from the previous layer. This causes overfitting problems and any spatial features which might be very helpful for solving the problem may be lost. This results in misclassifications and unstable classification performance. There are a few ways through which this issue has been tackled. Regularization of weights, batch normalization and weight decay are few of the approaches which have been used to counter the problem of overfitting.

Another severe problem with deep networks it is harder to train them. Error back propagation is the process of transmitting classification errors across the ANN so that the network weights are adjusted accordingly. In deep networks, this back propagation of errors is not sufficient to reach the initial layers of the network. Therefore, the learning speed of the first few layers of neurons is very less. To mitigate this issue, networks are sometimes trained in smaller parts so that each of the weights are optimized. Then the network is put together and trained as a whole to solve classification problems.

The CNN is a sub category of the deep networks which is widely used in image processing applications. Unlike an ANN, each neuron of a CNN is connected only to a subset of inputs from the previous layers. This ensures that the spatial features of the input data are utilizable across the network. Another desirable aspect of the CNN is that the reduced number of connections boost speed of the network and remedies overfitting. To further improve the stability, speed and reliability of the CNN, concepts like Max Pooling and Batch Normalization are commonly used.

Facial emotion recognition techniques use the facial expressions alone in order to determine emotions. Another major source of features for predicting emotions is the tone of speech. Several deep networks make use of both facial emotion recognition and speech emotion recognition. Such a setup is highly suitable for video analysis applications. One such network is described in [8]. Here AlexNet CNN was used in conjunction with a deep network trained on audio information and a shallow network to process the movements of the mouth. The total accuracy of the network was 41.03%.

V. Summary And Conclusion

To summarize, in this project a highly accurate Convolutional Neural Network to recognize emotions from facial expressions was built. The LBP cascade classifier available in OpenCV was used to detect faces in images. A CNN network was trained using the detected faces to identify a total of eight emotions. The concept of batch normalization was used in the network to regularize the weights and improve stability. Virtual sampling was used to improve class representation of those classes with insufficient data. The trained CNN was tested on unknown data from the same dataset used for training. The performance on test data was analyzed.

Three applications were developed using the trained CNN. The first application detected the emotions of faces from a webcam feed. The second application detected emotions of faces in a video. The third application detected the emotions of faces on screen.

The performance of the CNN on real-time applications was analyzed. Further avenues for improving the performance of the network for real time applications was explored and documented.

References

[1] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998

[2] A. Mollahosseini, D. Chan, and M. H. Mahoor, “Going deeper in facial expression recognition using deep neural networks,” in 2016 *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–10, IEEE, 2016.

[3] T. Kanade, J. F. Cohn and Y. Tian, “Comprehensive database for facial expression analysis”, *Proc. 4th IEEE International Conf. on Automatic Face and Gesture Recognition (FG'00)*, Grenoble, France, 2000, pp. 46-53

[4] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, “The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression”, *Proc. 3rd International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010)*, San Francisco, USA, 2010, pp. 94-101

[5] T. Ahonen, A. Hadid, and M.Pietika¨inen, “Face Description with Local Binary Patterns: Application to Face Recognition,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037-2041, 2006.

[6] J. F. Cohn, Z. Ambadar, and P. Ekman. Observer-based measurement of facial expression with the facial action coding system. *The handbook of emotion elicitation and assessment*, pp 203–221, 2007.

[7] P. Ekman and W. V. Friesen. Facial action coding system. 1977

[8] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, C. Gülçehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R. C. Ferrari, et al. “Combining modality specific deep neural networks for emotion recognition in video.”, *Proc. 15th ACM on International conference on multimodal interaction*, pp 543–550. ACM, 2013.