International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019

Automatic Facial Expressions and Identification of different face reactions using Convolutional Neural Network

Anand R, Kalkeseetharaman P. K., Naveen Kumar S

Abstract*: Automatic Face expression is the significant device in computer apparition and a predictable knowledge discovery application in automation, personal security and moveable devices. However, the state-of-the-art machine and deep learning (DL) methods has complete this technology game altering and even better human matching part in terms of accurateness. This paper focuses on put on one of the progressive deep learning tools in face expression to achieve higher accuracy. In this paper, we focusses on Automatic Facial Expressions and Identification of different face reactions using Convolution Neural Network. Here, we framed our own data and trained by convolution neural networks. Human behavior can be easily predicted using their facial expression, which helps marketing team, psychological team and other required team to understand the human facial expression more clearly.

***Keywords: Convolutional Neural Network, Pooling layer, Deep learning, Artificial Intelligence, Face expression*.**

I. INTRODUCTION

Facial expression is an important indicator of a person's emotion. Computers and other electronic devices in our daily lives will become more user-friendly if they can adequately interpret a person's facial expressions, thereby improving human-machine interfaces. Facial expression recognition can be implemented in all computer interfaces, automated psychological research and treatment, robots or even polygraphs Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using rules to reach approximate or definite conclusions) and self-correction. Particular applications [1] [2] of AI include expert systems, speech recognition and machine vision.

AI can be categorized as either weak or strong. Weak AI, also known as narrow AI, is an AI system that is designed and trained for a particular task. Virtual personal assistants, such as Apple's Siri, are a form of weak AI.

Revised Manuscript Received on October 30, 2019.
***** Correspondence Author
Anand R*, Assistant Professor, Department of ECE, Salem.
Kalkeseetharaman P.K, Department of ECE, Sona College of Technology, Salem.
Naveen Kumar S, Department of ECE, Sona College of Technology, Salem.

Strong AI, also known as artificial general intelligence, is an AI system with generalized human cognitive abilities. When presented with an unfamiliar task, a strong AI system is able to find a solution without human intervention. Because hardware, software and staffing costs for AI can be expensive, many vendors are including AI components in their standard offerings, as well as access to Artificial Intelligence Service (AIS) platforms. AI as a Service allows individuals and companies to experiment with AI for various business purposes and sample multiple platforms before making a commitment. Popular AI cloud offerings include Amazon AI services, IBM Watson Assistant, Microsoft Cognitive Services and Google AI services. Examples of AI technology, a) Automation, b) Machine learning c) Machine vision d) Natural language Processing e) Robotics f) Self-driving cars. Getting into Deep Learning (DL) and AI is not an easy task but is a critical part of data science programs. Many aspiring professionals and enthusiasts find it hard to establish a proper path into the field, given the enormous amount of resources available today. The field is evolving constantly, and it is crucial that we keep up with the pace of this rapid development. In order to cope with this overwhelming speed of evolution and innovation, a good way to stay updated and knowledgeable on the advances of DL, is to engage with the community using Python is the best way because of the following reasons. The programmers of big companies use Python as it has created a mark for itself in the software development with characteristic features like-
Interactive. Interpreted, Modular, Dynamic, Object-oriented Portable, High level &Extensible in C++ & C.

II. OBJECTIVES OF RESEARCH

This project is a Deep Learning model approach of Facial Expression prediction depending on different facial expression images. We use classification based Deep learning algorithm to train the model. We have here predicted 7 expressions namely: Angry, Disgust, Fear, Happy, Neutral, Sad and Surprise depending on the facial expression for different expression [11,3]. However, this method can be used for other expressions prediction provided with the images to train the model for the other possible expression. Facial expression is used to reconcile emotional state conveyed by a face as a reflection of a person's state of mind as an outer make-up. As it's aimed to provide us a clear picture about how one looks/appears when s/he has an emotional state to outer world[10],[11].

This emotional look of a person has the ability to communicate a lot about the person's internal condition as emotions are the outer make-up of all the state of mind, and observers can easily evaluate and predict a Person's emotional state in order to assist him or give feedback about his outer makeup or how he/she appears to the outer world. Even Nurses in hospital observe such thing in order to improve mental health.

In the health care centers this expression prediction helps Doctors in making a well diagnosed treatment plan for patients accordingly. In short it's very important to understand meaning of each emotion conveyed by different people. It is also very important to understand the reason behind a emotion and different emotions in selection, hospitals, dating, in schools and counseling centers.

The goal of this project is to predict, which emotion the facial expression conveys, from the grayscale picture of a person's face. Our evaluation metric will be the accuracy for each emotion (fraction of correctly classified images), supplemented by a confusion matrix (in later headings) which highlights which emotions are better recognized than others. In short,

- Input: 64 by 64 grayscale image of a face
- Output: Emotion conveyed by facial expression

III. DATA PREPROCESSING

For, all the 7-expression used Chinese women's expression images from "Jaffee", where for each expression (Angry, Disgust, Fear, Happy, Neutral, Sad and Surprise), thirty different images were recorded and cropped by removing unwanted data from the image.

Later for the purpose of the deep learning model the images got split into Train and Test images with six image per expression for testing and twenty-four images for each expression.

Therefore, the total number of images used for training is 24*7=168 training images. The total number of images used for testing the model is 6*7=42 testing images[3],[4]. This image data will be used for further proceedings.

IV. METHODOLOGY

After separating image dataset into train and test, it is time to decide, which methodology to use. Since we have dataset as direct images, it is not directly possible to give directly an image as an input to the neural network. So we need extract the important features from the image as a matrix consist of important information pixels of the image, then that pixel values ought to be given as the input to the Neural Network.

A Neural Network is simply a network of Neurons, where a Neuron is simply a mimic of bio neuron with input as pixel of the images, output as predictions. There are different layers namely input, middle and output layer[6,7]. We may increase the number of intermediate layers as per our requirement. Layers are the combination of more neurons.

Here the feature extraction can be done by applying filters to the image. Initially the image is allowed to the convolution process, the output of the convolution is passed as the input to the Pooling filter, and then Flattening is done to make all the pixels into one single dimension matrix[9].

The output of the flatten layer is further proceeded with Neural network concepts, simply flatten pixels are given input to the Artificial Neural Network (ANN), All these can be achieved using Convolution Neural Network (CNN) Algorithm (Detailed in Data Modeling).

V. RESULTS AND DISCUSSION

a.) Data modelling

CNN is an algorithm helps to extract important features from the image and train them. CNN has four steps to extract the important features from the image namely, 1 .Convolution, 2.Pooling, 3.Flattening, 4.Fully Connections

In Convolution operation, the input image will be convolved with feature detector/filters to get a feature map. The important role of the feature detector is to extract the features from the image. By applying convolution operation, the size of the image reduced, due to that we may lose some information, but feature detector filter helps to extract main feature from image and remove unwanted pixels. The main features are that we find or see to recognize an image. To one single input image, we apply number of filters or feature detector. These important features are parked in a matrix called feature map. The group of feature maps is called as a convolution layer. Pooling is a technique in CNN which helps us to avoid overfitting of data, spatial invariance and distortion. By applying Max pooling operation, we are neglecting 75% of the unwanted information or unwanted data, as we are taking the maximum of the pixels from the feature map depending on the size of the Kernel. Here developer can decide the size of the Kernel [5]. As we are applying poling to the feature detector in convolution layer, we will get pool layer. Pooling applied feature map is also called as pooled feature map. The group of pooled feature map is called as pooling layer. Along with the max pooling technique, we also have other two commonly used pooling techniques they are sum pooling and average pooling. Now the pooled feature map will be converted into a one single dimension matrix or map, where each pixel in one single column, nothing but flattening. Flattening layer converts multi dimension matrix to one single dimension layer. The output of the flattening is a flattening layer, which is the input of the future ANN [8]. In establishing full connections, dense layers like middle layer and output layers are defined with the number of neurons for training in each layer. The following describes the details of the layers used:

{'name': 'sequential_1',
 'layers': [{'class_name': 'Conv2D',
   'config': {'name': 'conv2d_1',
    'trainable': True,
    'batch_input_shape': (None, 64, 64, 3),
    'dtype': 'float32',
    'filters': 32,
    'kernel_size': (3, 3),
    'strides': (1, 1),
    'padding': 'valid',
    'data_format': 'channels_last',
    'dilation_rate': (1, 1),
    'activation': 'relu',
    'use_bias': True,

'kernel_initializer': {'class_name': 'VarianceScaling',
'config': {'scale': 1.0,
'mode': 'fan_avg',
'distribution': 'uniform',
'seed': None}},
'bias_initializer': {'class_name': 'Zeros', 'config': {}},
'kernel_regularizer': None,
'bias_regularizer': None,
'activity_regularizer': None,
'kernel_constraint': None,
'bias_constraint': None}},
{'class_name': 'MaxPooling2D',
'config': {'name': 'max_pooling2d_1',
'trainable': True,
'pool_size': (2, 2),
'padding': 'valid',
'strides': (2, 2),
'data_format': 'channels_last'}},
{'class_name': 'Flatten',
'config': {'name': 'flatten_1',
'trainable': True,
'data_format': 'channels_last'}},
{'class_name': 'Dense',
'config': {'name': 'dense_1',
'trainable': True,
'units': 120,
'activation': 'relu',
'use_bias': True,
'kernel_initializer': {'class_name': 'RandomUniform',
'config': {'minval': -0.05, 'maxval': 0.05, 'seed': None}},
'bias_initializer': {'class_name': 'Zeros', 'config': {}},
'kernel_regularizer': None,
'bias_regularizer': None,
'activity_regularizer': None,
'kernel_constraint': None,
'bias_constraint': None}},
{'class_name': 'Dense',
'config': {'name': 'dense_2',
'trainable': True,
'units': 7,
'activation': 'sigmoid',
'use_bias': True,
'kernel_initializer': {'class_name': 'RandomUniform',
'config': {'minval': -0.05, 'maxval': 0.05, 'seed': None}},
'bias_initializer': {'class_name': 'Zeros', 'config': {}},
'kernel_regularizer': None,
'bias_regularizer': None,
'activity_regularizer': None,
'kernel_constraint': None,
'bias_constraint': None}}]}

The model is built based on the above layers detailed in the above layer description with 3690123 parameters; simply will see the summary for the simple understanding about the model.

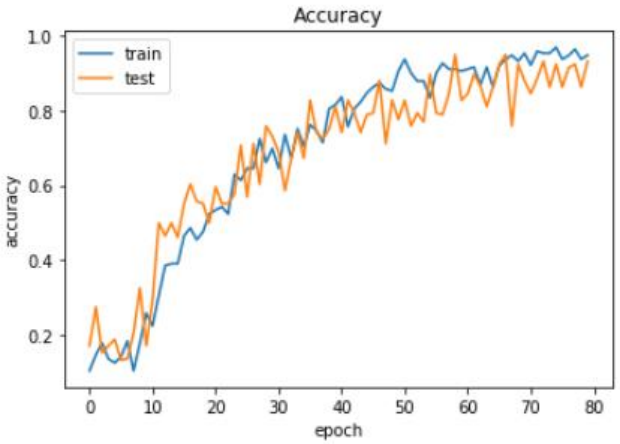

Figure 1: Training Accuracy for Training and Testing

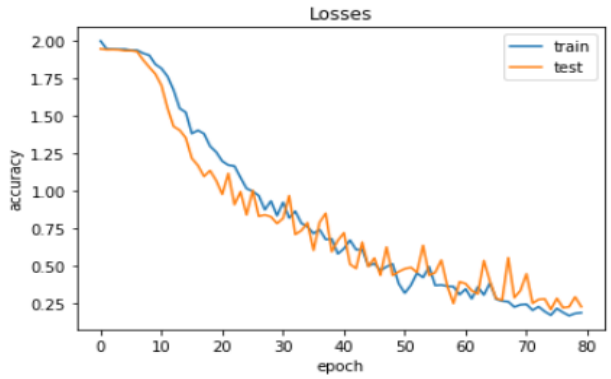

Figure 2: Losses During Training and Testing dataset

The figure 1 & 2 shows the training and testing for accuracy and losses. This is how the model is built and during the forward propagation and backward propagation weights get updated and training was done with the number of epochs, the following graph describes you in detail about the training accuracy and testing accuracy during each and every epoch and also the losses occurred during training and testing: Epoch 1/80: loss: 1.9988 - acc: 0.1043 - val_loss: 1.9470 - val_acc: 0.1724 Epoch 80/80: loss: 0.1894 - acc: 0.9467 - val_loss: 0.2276 - val_acc: 0.9310. At the end of 80th epoch we got 94% and 93% training and testing accuracy with the training and testing losses of 18% and 22%, from the first epoch as 10% and 17% traing and testing accuracy with more than 1 as losses. In Evaluation, one more use full concept is confusion matrix, with some parameters used to evaluate the model and understand the evaluation in better way.The entities TP, TN, FP and FN for each class can be evaluated using equations and the explanation of each parameter as follows: True positive (TP) – Number of defective images correctly classified as defective image. True negative (TN) – Number of non-defective images correctly classified as non-defective image. False positive (FP) – Number of non-defective images wrongly classified as defective image.

False negative (FN) – Number of defective images wrongly classified as non-defective image. In a multi classification model all these four parameters need to be found for individual classes, the formula for a class say A is defined as follows:

True positive (TP) of class $A = X_{AA}$, True negative (TN) of $\sum_{i=1}^{n} X_{iA} - -X_{AA}$, False negative (FN) of class A= $\sum_{i=1}^{n} X_{Ai} - -X_{AA}$; n for number of classes used which is shown in Table 1 & Table 2. Using the above table, we can easily define some performance metrics listed below in table class A = $\sum_{i=1}^{n} X_{ii} - X_{AA}$, False positive (FP) of class A= 3:Hence, the above were the evaluation metrics of the model, which helps us to understand the model's output, efficiency and other performance metrics more clearly which is shown in table.

Table 1: Confusion Matric for different facial expression

CONFUSION MATRIX	ANGRY	DISGUST	FEAR	HAPPY	NEUTRAL	SAD	SURPRISE
ANGRY	5	1	0	0	0	0	0
DISGUST	1	5	0	0	0	0	0
FEAR	1	0	4	0	1	0	0
HAPPY	0	0	0	4	1	1	0
NEUTRAL	1	0	0	0	5	0	0
SAD	0	0	0	0	0	5	1
SURPRISE	0	0	0	0	0	1	5

Table 2: Confusion matrix attributes

Parameters	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
TP	5	5	4	4	5	5	5
TN	28	28	29	29	28	28	28
FP	3	1	0	0	2	2	1
FN	1	1	2	2	1	1	1

Table 3: Performance Metrices

Performance Metrics	Formula	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Sensitivity	$\frac{TP}{TP+FN} * 100$	83.33	83.33	66.66	66.66	83.33	83.33	83.33
Specificity	$\frac{TN}{TN+FP} * 100$	90.32	96.55	100	100	93.33	93.33	96.55
False positive rate (FPR)	$\frac{FP}{TN+FP}$	0.096	0.034	0	0	0.066	0.066	0.034
False negative ratio (FNR)	$\frac{FN}{TP+FN}$	0.166	0.166	0.333	0.333	0.166	0.166	0.166
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN} \times 100$	89.18	94.28	94.28	94.28	91.66	91.66	91.66
Precision	$\frac{TP}{TP+FP}$	0.625	0.833	1	1	0.714	0.714	0.833
Negative Predictive value	$\frac{TN}{TN+FN}$	0.965	0.965	0.935	0.935	0.965	0.965	0.965
FI - Score	$\frac{2 * TP}{2TP+FP+FN}$	0.714	0.833	0.8	0.8	0.769	0.769	0.833

VI. CONCLUSION

Human behavior can be easily predicted using their facial expression, which helps marketing team, psychological team and other required team to understand the human facial expression more clearly. In Video feed and other feeds provide good recommendations to the users. We can provide good environment to the user of the corresponding product helps the company to improve its customer base.

REFERENCES

1. Anand, R., Veni, S., & Aravinth, J. (2016, April). An application of image processing techniques for detection of diseases on brinjal leaves using k-means clustering method. In 2016 International Conference on Recent Trends in Information Technology (ICRTIT) (pp. 1-6). IEEE.
2. Anand, R., Veni, S., & Aravinth, J. (2017, September). Big data challenges in airborne hyperspectral image for urban landuse classification. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1808-1814). IEEE..
3. Sabeenian, R. S., Paramasivam, M. E., Anand, R., & Dinesh, P. M. (2019). Palm-Leaf Manuscript Character Recognition and Classification Using Convolutional Neural Networks. In Computing and Network Sustainability (pp. 397-404). Springer, Singapore.
4. Paul, E., Anand, R., Karthick, P. V., & Paramasivam, M. E. (2019). Image Contrast Enhancement Using Particle Swarm Optimization. Journal of Advanced Research in Dynamic and Control Systems, 11(04-Special Issue), 1192-1196.
5. Ekman, P., & Keltner, D. (1997). Universal facial expressions of emotion. Segerstrale U, P. Molnar P, eds. Nonverbal communication: Where nature meets culture, 27-46..
6. Dimberg, U., Thunberg, M., & Elmehed, K. (2000). Unconscious facial reactions to emotional facial expressions. Psychological science, 11(1), 86-89..
7. Ekman, P. (1999). Facial expressions. Handbook of cognition and emotion, 16(301), e320.
8. Tottenham, N., Tanaka, J. W., Leon, A. C., McCarry, T., Nurse, M., Hare, T. A., ... & Nelson, C. (2009). The NimStim set of facial expressions: judgments from untrained research participants. Psychiatry research, 168(3), 242-249.
9. Mollahosseini, A., Chan, D., & Mahoor, M. H. (2016, March). 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.
10. Ng, H. W., Nguyen, V. D., Vonikakis, V., & Winkler, S. (2015, November). Deep learning for emotion recognition on small datasets using transfer learning. In Proceedings of the 2015 ACM on international conference on multimodal interaction (pp. 443-449). ACM.
11. Mavani, V., Raman, S., & Miyapuram, K. P. (2017). Facial expression recognition using visual saliency and deep learning. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2783-2788).
12. Zeng, N., Zhang, H., Song, B., Liu, W., Li, Y., & Dobaie, A. M. (2018). Facial expression recognition via learning deep sparse autoencoders. Neurocomputing, 273, 643-649.

AUTHORS PROFILE

Anand R is currently working as an Assistant Professor in Sona College of Technology, Salem, Tamil Nadu, India. He has more than 2 years of teaching experience. He received his bachelor's in engineering from Kongu Engineering college (Anna University) and his Master of Technology in Communication Engineering and Signal processing from Amrita University, Coimbatore. He is a team member in the research group named Sona SIPRO (**SONA Signal and Image PROcessing Research Centre**) center located at the Advanced Research Centre in Sona College of Technology, Salem. He has published papers in International, National Journals and Conferences. His areas of interest include signal and image classification, machine and deep learning