DEEP CONVOLUTIONAL NEURAL NETWORKS FOR FACIAL EMOTION RECOGNITION

M Geetha Yadav¹, Rajasekhar Nennuri², Roshna Sanjana K³, Sai Ramana Deekonda⁴, Shivangi Solanki⁵

¹,²,³,⁴,⁵Department of Computer Science Engineering, Institute of Aeronautical Engineering, Hyderabad. geethayadav22@gmail.com¹, rajasekhharnenuri@gmail.com², roshnasanjana21@gmail.com³, sairamanadeekonda9999@gmail.com⁴, shivangi.solanki27@gmail.com⁵

ABSTRACT

Artificial intelligence's rapid development has made a significant contribution to the technological environment. Machine Learning was created as a result of conventional algorithms failing to fulfill human needs in real time. Algorithms for learning (ML) and deep learning have had a lot of success in a variety of implementations, including classification systems, Pattern identification, recommendation systems, and so on. Emotion is essential in evaluating a person's emotions, actions, and feelings. Using the benefits of deep learning, an emotion recognition system can be developed, and various applications such as feedback analysis, face unlocking, and so on can be implemented with high accuracy. Here the main goal is to develop a Deep Convolutional Neural Network model that can classify five different human facial emotions. The manually collected image dataset is used to practice, evaluate, and validate the model.

KEYWORDS

Artificial intelligence’s, Emotion recognition system, Deep learning Deep Convolutional Neural Network.

I. INTRODUCTION

Computer systems, applications, and networks are rapidly evolving and becoming more widely used. These systems play an important role in our daily lives and make life much simpler for us. In this era, facial expression recognition systems are very important because they can capture human behaviour, emotions, and intentions. Traditional approaches have limited speed and precision, while a facial emotion recognition technology based on deep learning has proven to be superior. This framework aims to create a deep convolutional neural network model that recognises five distinct human facial emotions, which can be used for customer feedback research, face unlocking, and other applications.

As we all know that machine learning is an emerging technology in the field of computer science that is expected to have a more than 90%-95% effect in the upcoming years. Artificial neural networks, which are algorithms inspired by the human brain, are used in deep learning, a branch of machine learning. Convolutional Neural Networks are a form of deep neural network that employs convolution as a mathematical process. Since the dataset is made up of images, the computer uses Two-dimensional Convolutional Neural Networks for recognition. The proposed deep convolutional neural network is not only capable of recognising five distinct human facial emotions, but it also does so with a high degree of accuracy. The model is trained using the dataset, which was collected manually using a cell phone camera.

II. PROPOSED MODEL

A. Convolutional Neural Network

A neural network is a series of algorithms that simulate the human brain, and it uses these algorithms to find a relationship between data in order to find solutions. CNN is a type of Neural Network in which Convolution [7]-[9] is the mathematical operation used to find the relationship between the data. When it comes to complex problems like image classification, video classification, pattern recognition, and so on, traditional neural networks struggle miserably, but CNN has had great success in these areas, yielding high accuracy.
Traditional neural networks fail miserably when it comes to complex problems like image classification, video classification, pattern recognition, and so on, but CNN has excelled in these fields, yielding high accuracy. The convolutional layer is made up of small patches that use the filter values to transform the entire image. The formula for creating feature maps, i.e. the contribution from the convolutional layer, is Equation (1).

The output of the convolution layer is transferred to a pooling layer, which reduces the size of the output without losing any information. The flatten layer converts these 2-dimensional arrays to a single-dimensional vector, which can then be fed to the neural network for classification. The neural network employs the back-propagation algorithm, which adjusts the weights based on the errors, lowering the error (loss) function. The weight is modified with the aid of (2).

\[
G[m, n] = (f * h)[m, n] = \sum_j \sum_k h[j, k]f[m - j, n - k] \tag{1}
\]

where \(f\) is the input image, \((m,n)\) is the size of the generated matrix, and \((m,n)\) is the filter.

\[
W_i = W_i + \Delta W_i \tag{2}
\]

where is the weight(\(W_i\)) and is calculated using the delta law(\(\Delta W_i\)), as shown in (3).

\[
\Delta W_i = n \frac{dE}{dw_i} x_i \tag{3}
\]

where \(n\) is the rate of learning, \(E\) is the error function, and \(X_i\) is the data input.

**B. Proposed CNN Model**

Figure 1 shows the architecture of the proposed facial emotion recognition model. The model employs two convolution layers, with dropouts between them. The first convolution layer receives the input image, which is resized to 32 x 32 pixels.

An activation function is applied to the output of the convolution layer, which is referred to as the feature map. The activation function used here is ReLU (Rectified Linear Unit), which reduces negative values to zero while maintaining positive values. This feature map is applied to a pooling layer with a pool size of 2 x 2 in order to reduce the size without sacrificing any detail. To avoid overfitting, a dropout layer is used. This procedure is repeated for the next convolution sheet. Finally, some function values are stored in a two-dimensional array.

The flatten layer is used to transform these 2-dimensional arrays to a single dimensional vector, which is then fed into the dense layers of the neural network. A dropout layer is used to prevent overfitting. For the next convolution layer, repeat the process. Finally, a two-dimensional array is used to store certain feature values. The flatten layer converts these two-dimensional arrays into a single-dimensional vector, which is then fed into the neural network’s dense layers.
The proposed model uses a dataset of five different facial emotions: angry, positive, neutral, sad, and shocked. These are manually captured with a 48 MP sensor. Each image is 1920 x 2560 pixels in resolution. Table I shows how the dataset was divided. So that they are not biased, each class has the same number of training samples. The split between train, test, and validation is 8:1:1.

### III. METHODOLOGY

The model is implemented in Python as a programming language. The Jupyter Notebook is used to simulate the entire model. Keras, which runs on top of tensorflow, is used as the deep learning library for building the model, applying convolution layers, compiling, and fitting the model. Scikit-learn is the package that is used to find the confusion matrix, which contains information about the model's accuracy, precision, sensitivity, specificity, recall, and so on. Matplotlib and seaborn are used to plot the uncertainty matrix and other graphs such as accuracy and loss.

### RESULTS

The emotion image dataset is used to train CNN, which uses Adam as the optimizer and categorical cross-entropy as the loss function. Table II lists the model's parameters.
Adam is an optimization algorithm that can be used to update network weights with individual learning rates instead of the traditional stochastic gradient descent algorithm [11]. To adjust the learning rate, it uses first and second moment gradient estimations for each weight of the neural network. The random variable's nth moment is given in (4).

\[ m_n = E[X^n] \]  

(4)

where \( X \) is the random variable and \( m \) is the moment. The mean determines the first moment, while the uncentered variance determines the second. To measure the moments, Adam employs exponentially shifting averages. (5) and (6), respectively, are moving averages of gradient and squared gradient.

\[ m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t \]  

(5)

\[ v_t = \beta_2 v_{t-1} + (1 - \beta_2)g_t^2 \]  

(6)

where \( m \) and \( v \) are the moving averages are and, the hyperparameter is \( B \), and the gradient is \( g \). (7) and (8) are the final formulae for the bias corrected estimators for the first and second moments, respectively.

\[ m'_t = \frac{m_t}{1 - \beta_1^t} \]  

(7)

\[ v'_t = \frac{v_t}{1 - \beta_2^t} \]  

(8)

These moving averages scale the learning rate for each parameter individually, and the weight updation is performed using the formula in (9).

\[ W_t = W_{t-1} - n \frac{m'_t}{\sqrt{v'_t + \varepsilon}} \]  

(9)

where \( W_t \) is the updated weight, \( W_{t-1} \) is the previous weight and \( n \) is the step size.

Categorical cross-entropy, the loss (error or cost) function used for optimizing classification models, is given by (10).

\[ L(y, y') = - \sum_{j=0}^{M} \sum_{i=0}^{N} (y_{ij} \ast \log(y'_{ij})) \]  

(10)

<table>
<thead>
<tr>
<th>Model Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total images</td>
<td>2550</td>
</tr>
<tr>
<td>Activation</td>
<td>ReLU and Softmax</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.01</td>
</tr>
<tr>
<td>Epochs</td>
<td>11</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Loss function</td>
<td>Categorical Cross-entropy</td>
</tr>
</tbody>
</table>
where $y'$ is the expected result. This function will equate the expected values' distribution to the real values' distribution.

The normalised confusion matrix for the test samples using the proposed DCNN model is shown in Figure 3. Except for class 1, the specificity (recall), or the coverage of positive samples, indicates that the majority of them are expected to be positive (happy). The prediction results for Class 0 (angry) and Class 3 (neutral) are fine.

For the entire epochs, Figures 4 (a) and 4 (b) show the model accuracy and training loss, respectively. It is clear from the plots that the model is not overfitting. Table III shows the model's classification results based on precision, sensitivity, specificity, F1-score, and accuracy.

![Figure 3. Confusion Matrix](image)

![Figure 4. (a) Accuracy of training and testing data and (b) Model loss during the training process of CNN of the model](image)

<table>
<thead>
<tr>
<th>Classes</th>
<th>Precision</th>
<th>Sensitivity (Recall)</th>
<th>Specificity</th>
<th>F1 Score</th>
<th>Accuracy (in %)</th>
</tr>
</thead>
</table>
IV. CONCLUSION

For facial emotion recognition, this paper proposes a two-layer convolution network model. The model uses the image dataset to classify five different facial emotions. The model has comparable training and validation accuracy, indicating that it has the best fit and is generalizable to the data. The model reduces the loss function using an Adam optimizer, and it has been checked to have an accuracy of 78.04 percent. The work can be expanded to detect changes in emotion using a video series, which can then be used for a variety of real-time applications such as feedback analysis and so on. For efficient control of other electronic devices, this system can be combined with them.

REFERENCES

4. Lulingq Liu, "Human Face Expression Recognition Using Deep Learning and Deep Convolutional Neural Networks," 2019 International Conference on Smart Grid and Electrical Automation (ICSGÉA)
10. R Nennuri, AK Chaitanya, LP Malyula,” Implementation of data frame work system based on model driven architecture for MAS and Web based applications”, International Journal of Engineering & Technology 7 (2.20), 1-4
12. YB M Geetha yadav Golla Swetha , Vasista Kumar, “Intrusion Detection Scheme Using Machine Learning”, icrcsit-20