Personality recognition of images using EmotionGCN

Revanth A [1] Prathibamol C P [2]

[1]Amrita Vishwa Vidyapeetham [2]Amrita Vishwa Vidyapeetham

Problem

The area of emotion recognition offers tremendous scope to the human computer interaction, healthcare, robotics, biometric system, behavioural modelling. Emotion recognition can be done by extracting and analysing the features of facial expression data, text data, body movements, voice, brain, or heart signal. Here, image dataset with facial expression is used for emotion recognition. Emotion recognition is the study of recognizing six universal expressions such as anger, fear, joy, happiness, sadness, and surprise using various computer science techniques. Data extracted from the voice, facial expressions or brain signals are called as soft biometrics. Here, with the image dataset, the-state-of-art CNN model is used for these applications. But the problem is, the same facial image can express different emotion for different model. Two or more emotions have high correlation with one another, which is difficult for classification. Another problem is a number of layers. The performance of the CNN increases with increase in number of layers. This will be problematic if the model must be fine-tuned for any specific application, and hence it is computationally expensive. So here we present GCN specifically trained for emotion recognition called as EmotionGCN.

EmotionGCN

From [2], [3], for the emotion recognition, there is need for the colour. Thus, the images in the dataset will be converted to greyscale. Unlike CNN, the equivalent graph for the image is created by assigning each pixel to a node and their values to the corresponding edges by thresholding the average of pixels. From this graph, a sparse diagonal matrix g will be created with intensities values on the diagonal. This sparse matrix g will be trained with single or multilayer GCN to get the distribution graph for each emotion in the emotion wheel separately.

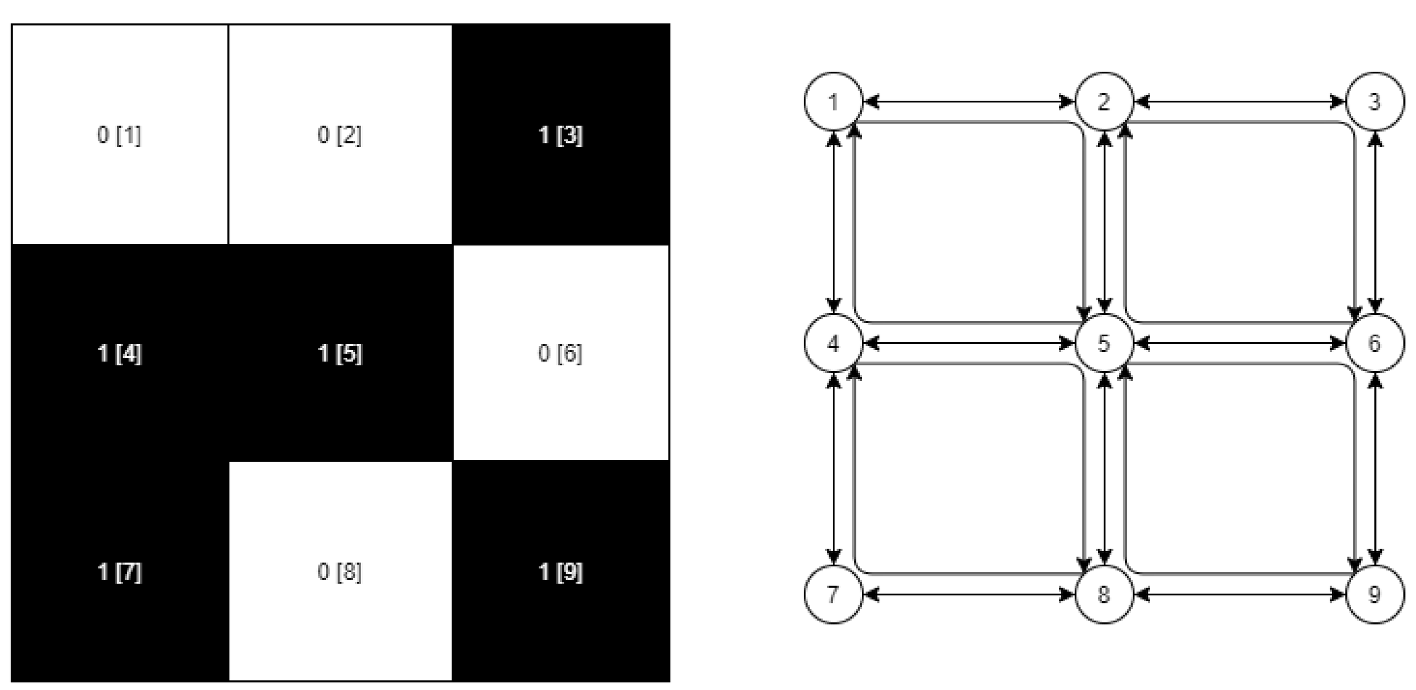

Figure 1. Conversion of image pixel data to graph.

Algorithm for EmotionGCN

1. For an image I_i, its feature f_i will be determined by $f_i = F(I_i)$.
2. The emotion correlation matrix g will be obtained from the sparse diagonal matrix from the graph and fed to GCN.
3. From this the final weight matrix $\mathrm{W} = \mathcal{G}\left(\mathbf{W}_p,\ \mathrm{g}\right)$ can be determined.
4. Compute the loss and proceed to back propagation.
5. These optimized parameters will be updated to the GCN model.
6. This process will be repeated for a contact number of epochs.
7. After the training is completed, the emotion distribution graph will be generated for each emotion in the emotion wheel.
8. The final form of multilayer GCN be $\mathbf{Z} = \mathrm{Softmax}\left(\widetilde{A}\sigma\left(\widetilde{A}\mathbf{X}\mathbf{W}^{(0)}\right)\mathbf{W}^{(1)}\right)$. Where, $\widetilde{A}$ is the normalized adjacent matrix $\hat{\mathrm{D}}^{-\frac{1}{2}}\hat{\mathrm{A}}\hat{\mathrm{D}}^{-\frac{1}{2}}$.

Analysis of Algorithm

From [1], let the graph generated from the pixels of the image dataset be $G = (V, E)$ where, V is the vertex or node in the graph and E be the edge. This graph structure will be in the form of sparse diagonal matrix. This will produce a node level matrix in which each row represents the output feature f_i. Here, the operation perform by every GCN layer is:

$$f\left(\mathrm{H}^{(l)}, \mathrm{A}\right) = \sigma\left(\mathrm{A}\mathrm{H}^{(l)}\mathbf{W}^{(l)}\right) \tag{1}$$

where,

- $\mathrm{H}^{(l)}$ be the input node feature matrix.
- $\mathbf{W}^{(l)}$ be the transformation matrix with learnable weights.
- σ be the nonlinear activation function.

In practice, the performance of the single layer GCN will be not up to the mark. So multiple GCN layers can be stacked to form the multilayer GCN where the output feature from the i^{th} GCN layer will normalized and fed to the $(i+1)^{th}$ GCN layer. The output feature vectors from the final GCN layer will be applied to a nonlinear activation function such as softmax or reLu to perform the task specified. The output from the two-layer GCN will be in the form:

$$\mathbf{Z} = \mathrm{Softmax}\left(\widetilde{A}\sigma\left(\widetilde{A}\mathbf{X}\mathbf{W}^{(0)}\right)\mathbf{W}^{(1)}\right) \tag{2}$$

where, $\widetilde{A}$ is the normalized adjacent matrix $\hat{\mathrm{D}}^{-\frac{1}{2}}\hat{\mathrm{A}}\hat{\mathrm{D}}^{-\frac{1}{2}}$.

After the graph convolution operation is done, the next thing is to find the emotion correlation graph. This graph can be constructed by following one of the psychological models. Each psychological model represents a different set of emotions. Here angry, happy, sad, surprise emotions will be considered based upon the Mikel's wheel. The emotion correlation matrix can be determined by finding the distance matrix for each emotion separately using the cosine distance metric. Then the self-connections are removed and form a non-directed graph. This graph is called the emotion correlation graph for emotion, and the process is repeated for all other emotions in the set. This method similar to finding the probability of i^{th} emotion given j^{th} emotion p(i|j) which is defined as,

$$p(i \mid j) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d(i,j)^2}{2\sigma^2}\right) \tag{3}$$

where, $d(i,j)$ is the pairwise cosine distance between the emotions i and j. When the input matrix E and correlation graph g is fed to the two layer GCN network, then the two layer GCN can be expressed as,

$$\mathbf{W}^{(1)} = \mathrm{ReLU}\left(\tilde{\mathrm{g}}\mathrm{E}\mathrm{U}^{(0)}\right) \tag{4}$$

$$\mathbf{W}^{(2)} = \mathrm{Softmax}\left(\tilde{\mathbf{g}}\mathbf{W}^{(1)}\mathbf{U}^{(1)}\right) \tag{5}$$

where,

- $\mathbf{g}$ be the normalized version of g.
- $\mathrm{U}^{(0)}$, $\mathrm{U}^{(1)}$ be the learnable weights in GCN.
- $\mathbf{W}^{(2)}$ be the final output weights of distribution prediction.

After every GCN layer operation, the weight matrix should be normalized before applying it to another GCN layer. In this approach, l^2 normalization along the row of weight matrix W is performed. The l^2 normalization of weight matrix W be,

$$w_{ij} = \frac{w_{ij}}{\sqrt{\sum_{j=1}^{d} w_{ij}^2}} \tag{6}$$

Results and Discussions

Emotion wheel	Correlation graph edges	Emotion efficiency
Angry	250706	0.4496240601503759
Happy	267144	0.5168
Sad	242263	0.4047244094488189
Surprise	300470	0.7196850393700788

Table 1. Performance of GCN in emotion wheel.

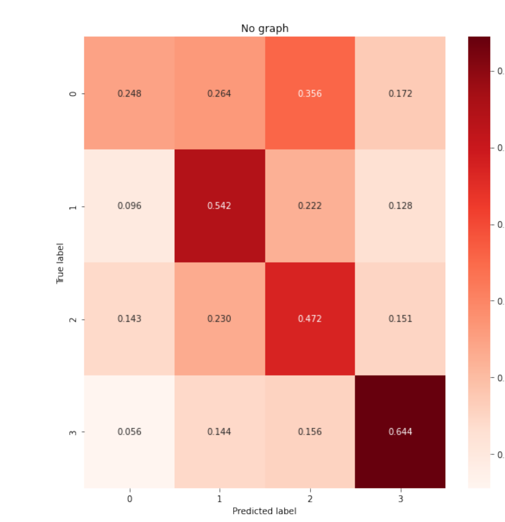

Figure 2. Confusion matrix of CNN model.

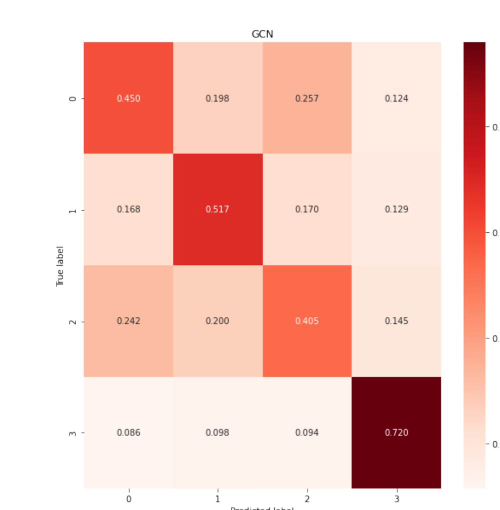

Figure 3. Confusion matrix of GCN model.

Conclusion

In this model for the emotion recognition, we consider the interrelationship between different emotions using the emotion correlation graph generated using EmotionGCN. By introducing GCN instead of CNN, the number of layers is reduced and hence the computational complexity also reduced as the model only uses two layers of GCN compared to the CNN which has a greater number of layers with same accuracy of this model. From the confusion matrix of the CNN and EmotionGCN model, the EmotionGCN better classifies the emotions even when there is huge variance in number of images for each class. So, using the EmotionGCN in emotion recognition is better when compared to the state-of-the-art CNN.

References

[1] Tao He and Xiaoming Jin. Image emotion distribution learning with graph convolutional networks. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 382–390, 2019.

[2] Yangtao Wang, Yanzhao Xie, Yu Liu, and Lisheng Fan. G-cam: Graph convolution network based class activation mapping for multi-label image recognition. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pages 322–330, 2021.

[3] Yangtao Wang, Yanzhao Xie, Yu Liu, Ke Zhou, and Xiaocui Li. Fast graph convolution network based multi-label image recognition via cross-modal fusion. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1575–1584, 2020.