Human emotions considered to be universal are happiness, sadness, angriness, surprised, fearful, and neutral. Excluding disorders that disrupt emotional functioning such as alexithymia, it is safe to say everyone has experienced the above emotions and they play an important role in verbal and non-verbal communication. Most people can typically distinguish between happiness and sadness or angriness and surprised by looking at others facial expressions. Deep learning has been used to classify emotions with high accuracy (Mollahosseini et al., 2016; Kanjo, Younis & Ang, 2019). The aims of this experiment were to explore how accurate deep learning can be in classifying emotions. It is hypothesised a convolutional neural network (CNN) can provide high accuracy in classifying images depicting happiness, sadness, angriness, surprised, fearful, and neutral emotions.

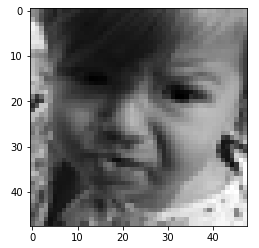
**Method**

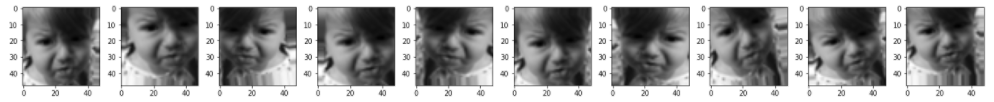
**Dataset background**

The facial emotion recognition dataset was originally released in 2013 and consisted of images depicting different emotional expressions including happiness, sadness, angriness, surprised, disgusted, fearful, and neutral. The dataset consisted of 35,685 samples divided into the training and test set and all images were 48 x 48 pixel grey scaled. The dataset was downloaded from Kaggle.

**Data augmentation: training set**

The original dataset was imbalanced among the 7 classes: the disgusted class was removed from the analysis considering it had minimal samples and this helped reduce the complexity of the analysis to 6 classes to classify. The remaining 6 classes were imbalanced and ranged from 3,171 to 7,215 samples. Therefore, to remove data imbalance and increase training set size, data augmentation using keras API was done for every class so each class would consist of ~ 15,000 samples. The total amount of samples in the training set was ~ 90,000. Figure 1 shows an image from the angry emotion class and data augmentation performed to the image 10 times.





*Figure 1.* Top image represents original image and bottom images are various augmentations done to original image

**Validation set**

Considering the original dataset did not have any validation sets, a new validation set was created. 10% of data from each class in the training set was moved to a corresponding validation set. Therefore, there was now a new validation set directory with each of the 6 classes. Validation set was ~ 9000 samples.

**Convolutional neural network architecture: model 1**

Keras sequential API was used to stack the layers in the CNN and rectified linear unit activation function was used for all layers: the first layer was a 2D convolutional layer with 32 filters and filter size of 3 x 3. Batch normalization was then applied to standardize the inputs, and this was then followed with a max pooling layer which had a pool size of 2 x 2 and 2 strides. The second 2D convolutional layer consisted of 64 filters and a filter size of 3 x 3. Batch normalization followed by max pooling with a pool size of 2 x 2 and 2 strides was then applied. The CNN was then flattened before the fully connected layers. The first dense layer consisted of 64 nodes followed by batch normalization and a dropout of 0.5 was then used. The second dense layer consisted of 32 nodes followed by batch normalization and a dropout of 0.5 was used again. The final output layer consisted of 6 nodes and the soft max function was used.

The model was compiled with Adam optimizer with a learning rate of 0.0001 and the loss function was categorical cross entropy. The model was fitted with the training and validation data: batch size was 32 and there were 50 epochs. Learning rate reduction call back was used in the fit function: if validation loss didn’t decrease in 3 epochs, the learning rate was reduced by a factor of 0.90. The early stopping call back was also used in the fit function: if validation accuracy did not decrease in 8 epochs, the neural network would stop training and the program would finish.

**Convolutional neural network architecture: model 2**

Keras sequential API was used to stack the layers in the CNN and rectified linear unit activation function was used for all layers: the first layer was a 2D convolutional layer with 16 filters and a filter size of 3 x 3. Batch normalization was then applied followed by a max pooling layer with a pool size of 2 x 2 and 2 strides. The second 2D convolutional layer consisted of 32 filters with a filter size of 3 x 3. Batch normalization was then applied followed by max pooling with a pool size of 2 x 2 and 2 strides. The third 2D convolutional layer consisted of 64 filters with a filter size of 3 x 3. Batch normalization was then applied followed by max pooling with a pool size of 2 x 2 and 2 strides. The CNN was then flattened before the fully connected layers. The first dense layer consisted of 64 nodes followed by batch normalization and a dropout of 0.5 was then used. The second dense layer consisted of 32 nodes also followed by batch normalization and a dropout of 0.3. the third dense layer consisted of 16 nodes followed by batch normalization and a dropout of 0.1. The final output layer consisted of 6 nodes and the soft max function was used.

The model was compiled with Adam optimizer with a learning rate of 0.0001 and the loss function was categorical cross entropy. The model was fitted with the training and validation data: batch size was 32 and there were 50 epochs. Learning rate reduction call back was used in the fit function: if validation loss didn’t decrease in 3 epochs, the learning rate was reduced by a factor of 0.90. The early stopping call back was also used in the fit function: if validation accuracy did not decrease in 10 epochs, the neural network would stop training and the program would finish.

**Results**

Model 1 early stopping occurred on epoch 26 due to the validation accuracy not increasing over several epochs. Training accuracy kept increasing but validation accuracy was hovering around 52% for 13 epochs indicating the model was overfitting and could not generalize. Therefore model 1 was not used on the test set and will not be mentioned for the remainder of this research.

Model 2 early stopping occurred on epoch 41 due to validation accuracy not increasing over several epochs. As like model 1, training accuracy kept increasing but validation accuracy was hovering around 56% for 13 epochs indicating model 2 was also overfitting and could not generalize well. Although validation accuracy was slightly better than model 1 scoring 56% over several epochs. Therefore model 2 was used on the test set.

|  |  |
| --- | --- |
| Class | Accuracy % |
| Total | 56 |
| Angry | 49 |
| Fearful | 29 |
| Happy | 79 |
| Neutral | 57 |
| Sad | 46 |
| Surprised | 61 |

*Table 1*. Test set accuracy for each class and total accuracy

As can be seen from table 1, the total accuracy for the test set was 56% which was the same as the validation set accuracy. There was wide variation in accuracy for each class ranging from 29-79% with happiness emotion scoring the highest at 79% and fearful emotion scoring the lowest at 29%.

**Discussion**

The hypothesis that a CNN can predict various emotions with high classification accuracy was not supported. The total classification accuracy across the various emotions was only 56%. Although when exploring each emotion separately, there is wide variation between the level of accuracy the neural network could classify. For instance, the neural network could classify happiness with high accuracy scoring 79% while it struggled to classify images of fearful which scored 29%. The neural network classified surprised emotion at 60%, neutral at 57%, angry at 49% and sad was 45%. For reasons unknown, the neural network was much better at distinguishing some emotions from others. Perhaps happiness may be the easiest emotion to recognise compared to the others, further research should run studies using neural network models to see if similar patterns emerge in terms of some emotions being easier to distinguish compared to others. Research should also investigate how accurate humans are at distinguishing the various emotions used in this research. The takeaway implication from this study is some emotions may be harder to distinguish than others. Sad and fearful may be two emotions that are hard to tell apart for neural networks and humans would also probably have a hard time differentiating the two emotions.

In conclusion, the CNN overall classification accuracy was 56% but more importantly, further exploration found that the CNN was much better or worse at classifying some emotions than others.

References

Kanjo, E., Younis, E. M., & Ang, C. S. (2019). Deep learning analysis of mobile

physiological, environmental and location sensor data for emotion detection. *Information Fusion*, *49*, 46-56.

Mollahosseini, A., Hasani, B., Salvador, M. J., Abdollahi, H., Chan, D., & Mahoor, M. H.

(2016). Facial expression recognition from world wild web. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 58-65).