Facial Expression Recognition through Machine Learning

Sivasankari Rajamanickam
Queen Mary University of London
Mile End Rd, Bethnal Green, London E1 4NS
s.rajamanickam@se21.qmul.ac.uk

1. Introduction

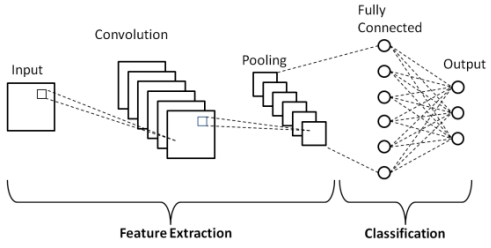
Over the past few years, real-time facial emotion recognition have been a major topic of investigation. In this paper, we are interested in the extraction techniques and classification in the facial expression recognition classification (FER). On the Aff-Wild2 database, we disclose the performance of training models using several classifier techniques and select the top model. The chosen model is subjected to an ablation study, and the performance is assessed using a new database.

2. Task 1: Main Machine Learning Model

2.1. CNN:

Convolutional neural network (CNN) is a frequently employed neural network in the field of image recognition.[1] Convolutional neural networks were created by neurocognitive machines using the visual field concept seen in the human brain. It has the traits of a local connection and parameter sharing, as well as outstanding time-shift invariance processing qualities. The usage of CNN has been successful in various pattern recognition systems today, including handwritten character identification, face recognition, and speech recognition. An input layer, two convolutional layers, two pooling layers, two fully connected layers, and output layers make up the eight layers of the CNN network structure.[2] Then, the nth feature of the convolution layer node in the current layer T, given the mth input graph of the convolutional layer to be Xm, Wn, and m represent the convolution kernel from the mth input graph to the nth feature graph. There are two types of pooling operations: maximum pooling and average pooling.[3] The pooling layer's primary function is to reduce overfitting, compress the image, and facilitate optimization. All of the retrieved features can be used to train a classifier, such as the softmax classifier, after the features have been acquired by convolution, however this presents a computational complexity barrier. The tail of the convolutional neural network contains the fully connected layer. It connects all of the features in the convolution output's two-dimensional feature map to create a one-dimensional vector before sending the output value to a classifier, like Softmax classifier.

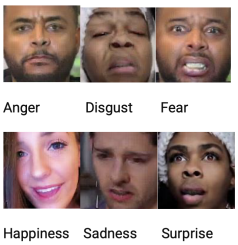
Figure 1. CNN



2.1.1 Experimentation:

Training and testing are two distinct phases of the facial expression recognition process. There are two sets of data in the Aff-Wild2 database: a training set and a test set. The test set is used to evaluate the performance of this model and determine whether it can more accurately complete the Facial Expression Recognition task. The training set is used to train the model. The first is to train the dataset to generate a model and save it, then use the trained model to test the picture, and finally get the classification results.

Figure 2. Aff-Wild2 database



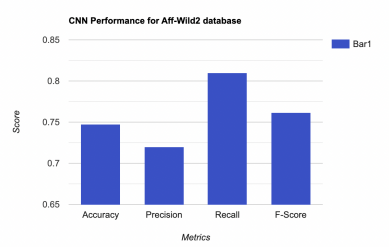
Six classes—Anger, Disgust, Fear, Happiness, Sadness, and Surprise—that individually represent an emotion are included in the Aff-Wild2 database. The tensorflow library is

used to load the dataset. There are 43515 photos altogether. Training and testing datasets are created from this dataset. The training dataset would be used to create the model, and the testing dataset would be used to evaluate the completed model. Based on their class, each image in the collection is given a label. Currently, data is taken for training at a rate of 80%, validation at 10%, and testing at 10%. The emotions predicted by CNN approach is compared against labelled data. The efficiency of CNN approach is shown in terms of Accuracy, Precision, Recall and F-Score.

Figure 3. CNN Performance

Accuracy	Precision	Recall	F-Score
0.7475	0.72	0.81	0.762

Figure 4. CNN Performance Bar Graph



3. Task 2: Baseline Machine Learning Model

3.1. SVM:

Vapnik [4] created the support vector machine (SVM) for binary classification. Its goal is to identify the best hyperplane with characteristics x belongs to R^m to divide classes in a given dataset, f (w, x) = w x + b. By resolving an optimization issue, SVM learns the parameters w.

$$min\frac{1}{p}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{p} max(0, 1 - y_i'(\mathbf{w}^T\mathbf{x}_i + b))$$

where y is the actual label, w^T x + b is the prediction function, C is the penalty parameter (which may have any value or be chosen using hyper-parameter tuning), and $w^T w$ is the Manhattan norm (also known as the L1 norm). Equation 1 is the L1-SVM with the common hinge loss. L2-SVM (Eq. 2), its differentiable counterpart, offers more consistent outcomes[5]. where the squared hinge loss is present

$$min\frac{1}{p}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{p} max(0, 1 - y_i'(\mathbf{w}^T\mathbf{x}_i + b))^2$$

and w_2 is the Euclidean norm, also known as the L2 norm.

The emotions predicted by SVM approach is compared against labelled data. The efficiency of SVM approach is shown in terms of Accuracy, Precision, Recall and F-Score.

Figure 5. SVM Performance Bar Graph

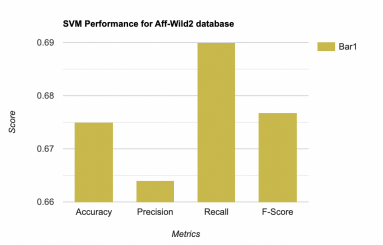


Figure 6. SVM Performance

Accuracy	Precision	Recall	F-Score
0.675	0.664	0.69	0.6768

4. Task 3: Ablation Study

To accurately anticipate, various datasets require various sets of hyperparameters. However, it can be challenging to pick which hyperparameter to use given the abundance of options. There is no definitive answer on the optimal number of layers, neurons, or optimizers for each dataset. Finding the potential optimal sets of hyperparameters to create the model from a particular dataset requires hyperparameter tuning. The number of neurons, activation function, optimizer, learning rate, batch size, and epochs are the hyperparameters that need to be tuned. The number of layers must be tuned in the second stage. Other traditional algorithms do not possess this. The accuracy may be affected by various layers. To perform ablation study, we modify the learning rate, batch size and epoch as mentioned:

Figure 7. Tuned Hyperparameters for CNN

```
[45] # Set paramaters
     params_nn ={
          'learning_rate':0.001,
          'batch_size':128,
          'epochs':(10, 140)
     }
```

After tuning the hyperparameters, the following accuracy and loss is noted:

Model accuracy remains the same across the epoch variations, but seems to drop drastically around an epoch of 110. The numerical results in this part helped to clarify the impact of each hyperparameter and the various tuning techniques to increase accuracy levels. Although neural networks have achieved extraordinary accuracy in automating the solution of complicated problems, little is understood about the precise principles behind the recognition process.

Figure 8. Graph of Variations in Accuracy with Epoch

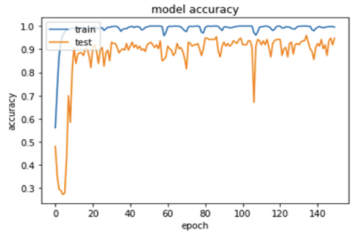
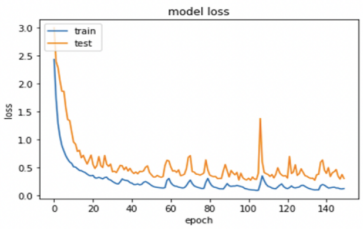


Figure 9. Graph of Variations in Loss with Epoch



5. Task 4: Production Data

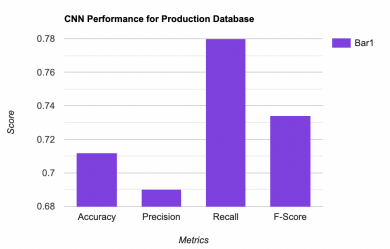
CNN was implemented using Aff-Wild2 Database and achieved an accuracy of 74.75%. SVM on the other hand, achieved an accuracy of 67.5%. The study suggests that CNN shows greater accuracy on the Database than SVM. Now that we know the best model, we implement CNN on a production dataset. This dataset consists of images that have been generated artificially from the Aff-Wild2 Database. The 6 basic expressions are used to annotate each image in the provided dataset (i.e., happiness, surprise, disgust, fear, sadness, anger). There are 26,130 imaged in total and we split the dataset into training and testing. 80% of the dataset is taken for training and remaining 20% of the dataset is used to evaluate the model. The CNN framework used on the Aff-Wild2 Database is replicated on the production database the approach shows the following efficiency: The accuracy decreases on the production dataset.

Figure 10. CNN Performance on production dataset

Accuracy	Precision	Recall	F-Score
0.712	0.69	0.78	0.734

This shows the impact of dataset size on the CNN Model. This could be because of model overfitting which increases with a reduced dataset size. Thus, as the database size increases, the estimates confidence increases, uncertainty decreases and the precision is higher.

Figure 11. CNN Performance on Production Dataset Bar Graph



References

[1] Changxing Ding, Dacheng Tao, Trunk-branch ensemble convolutional neural networks for video-based face recognition[J], IEEE Trans. Pattern Anal. Mach. Intell. (2016) 1 PP(99).

[2] Wei Shen, Mu Zhou, Feng Yang, Multi-crop Convolutional Neural Networks for lung nodule malignancy suspiciousness classification[J], Pattern Recognit. 61 (61) (2017) 663–673.

[3] Y. Ioannou, D. Robertson, J. Shotton, et al., Training convolutional neural networks with low-rank filters for efficient image classification[J], J. Bacteriol. 167 (3) (2016) 774–783.

[4] C. Cortes and V. Vapnik. 1995. Support-vector Networks. Machine Learning 20.3 (1995), 273–297. https://doi.org/10.1007/BF00994018

[5] Yichuan Tang. 2013. Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239 (2013).