Facial Emotion Recognition on Static Facial Expressions in the Wild dataset by Convolutional neural network and fine-tuning ResNet

Chen Huang

Research School of Computer Science,
Australian National University
u6735118@anu.edu.au

Abstract. The deep learning is prevalently applied in the image recognition tasks in recent decade and it obtains considerably good performance, especially on the database with lab-controlled properties. In many deep neural networks, deep convolutional neural network stands a crucial role, and it is widely used on image related tasks. However, many deep neural networks fail to deal with the small unconstrained image database. In this paper, we basically construct several CNN classifiers, CNN Autoencoder and fine-tuning ResNets, to deal with the facial recognition challenge on SFEW dataset which consists of close to real-world properties, and the size of the database is small. To avoid overfitting, we also compare several regularization strategies, including weight decay, distinctiveness pruning and dropout to generalize the facial emotion recognition model. In this paper, we obtain a CNN classifier with maximum test accuracy around 48.68% and the fine-tuning ResNet18 gains the highest test accuracy about 57.43%.

Keywords: SFEW, CNN, Autoencoder, neural network, FER, fine-tuning, weight decay, distinctiveness pruning, dropout

1 Introduction

Since Krizhevsky, Sutskever and Hinton achieved low error rate on ImageNet recognition challenge via deep convolutional neural network (CNN) in 2012, computer vision has gained considerable development in recent decade. The technology explosion of object recognition based on image or video facilitates the rapid growth of artificial intelligent (AI) related applications and products. Facial Expression Recognition (FER) is a popular task which gains enormous interests from the experts throughout the world. Many research experiments obtain pretty good accuracy by constructing the machine learning model based on the images generated in lab-controlled environment (Dhall et al., 2011). However, the classification on images or videos taken from real-world condition are more complicated as the real-world environment introduces considerable noises, such as illumination, face angle, which increases the difficulties to get high recognition accuracy.

Static Facial Expressions in the Wild (SFEW) extracting the temporal image data from Acted Facial Expressions in the Wild (AFEW) is a database comprising 675 screenshots from movies (Dhall et al., 2011). The images in SFEW possess close to real world properties, such as varied head posture, large age range and realistic illumination, as well as the resolution of the images are unconstrained (Dhall et al., 2011). Dhall et. al (2011) has applied the local phase quantization (LPQ) and the pyramid of histogram of oriented gradients (PHOG) descriptors to process the images in SFEW and used a non-linear support vector machine (SVM) as the classifier to obtain a baseline accuracy about 19.0%. We would like to construct some new FER classifiers on the SFEW dataset to improve the recognition accuracy, as well as compare the effects of the classifiers.

Krizhevsky et al. (2012) trains a deep neural network (DNN) firstly stacking convolutional layers, obtaining 17% top-5 error rate and 26.2% second-best error rate on ImageNet LSVRC-2010 consisting of 1000 classes. Therefore, we devise a deep convolutional neural network on the images in SFEW to train a FER classifier and evaluate the performance. In addition, Autoencoder can be feature selection model as it forces to learn discriminative features from input (Wang et al., 2017). Thus, we design an Autoencoder to extract high-level features of images and use the extracted features to train a FER classifier. We would like to compare the performances between extracting features from Autoencoder and extracting features by the LPQ and PHOG descriptors. Furthermore, He et al. (2016) devise a very deep CNN for object detection task which achieves state of art result on ImageNet ILSVRC 2015 dataset containing 1000 classes, about 3.57% top-5 error rate. The authors utilize residual connections to skip some layers to decrease the neural network complexity, which enables the possibility to construct up to 152 layers network and address the degradation issue in DNN. Thus, we would like to use the different layer number models of ResNet (He et al., 2016) as the pretrained models to apply different fine-tuning strategies to obtain some new models on SFEW and analyze the effects.

In DNN tasks, complex network with too many layers easily encounters effectiveness degradation due to gradient vanishing and exploding (Glorot & Bengio, 2010). In addition, the performance of the training model decreases dramatically as the overfitting is occurred easily when training data is small (Pasupa & Sunhem, 2016). SFEW is a small dataset contains 675 screenshot images from movies, and we would like to train a deep CNN on the dataset,

which potentially encounters the degradation and overfitting issues. Thus, we would like to apply some strategies, including distinctiveness pruning (Gedeon & Harris, 1991), dropout (Srivastava et al., 2014) and L2 weight decay (Loshchilov & Hutter, 2017), in our DNN models to prevent the degradation and overfitting issues in our DNN models, as well as compare the effects of the three regularization strategies.

2 Dataset

2.1 Dataset overviews

SFEW dataset. The report research bases on the unconstrained image data from SFEW (see Fig. 1) consisting of 675 screenshots extracted from the temporal data in AFEW (Dhall et al., 2011). Basically, the static images possess close to real-world conditions that the head posture, person age and illumination are similar with the realistic situations as they are originated from the movies (Dhall et al., 2011). The SFEW labels data with 7 classes comprising "Angry", "Disgust", "Fear", "Happy", "Neutral", "Sad" and "Surprise" and they are denoted by integers from 0 to 6 respectively in this research task. The entire data distribution along with the facial emotion class is balanced that each class contains 100 data points except the class "Disgust" contains 75 data points (see Fig. 2). In addition, the facial emotion images are captured from 102 subjects aged between 1 and 70, and the data distribution along with the subject is imbalanced (see Fig. 2).

Fig. 1. Sample images from the SFEW dataset

In addition, Dhall et al. (2011) train the FER model by using the features extracted by LPQ and PHOG, and the features are only kept the first five principal components (CPA) respectively, remaining about 98% variance, which can retain most information stored in the original images. The authors combine the top-5 CPA features of LPQ and PHOG and feed the combine vectors to a non-linear SVM, gaining a baseline accuracy about 19%. In our experiments, we use the 10-dimension features of LPQ and PHOG as well, as we would like to compare with the FER performance of extracting features from images via CNN. Basically, the 10-dimension CPA features are stored with the corresponding image names and facial expression labels, and they have the same order with the raw SFEW images.

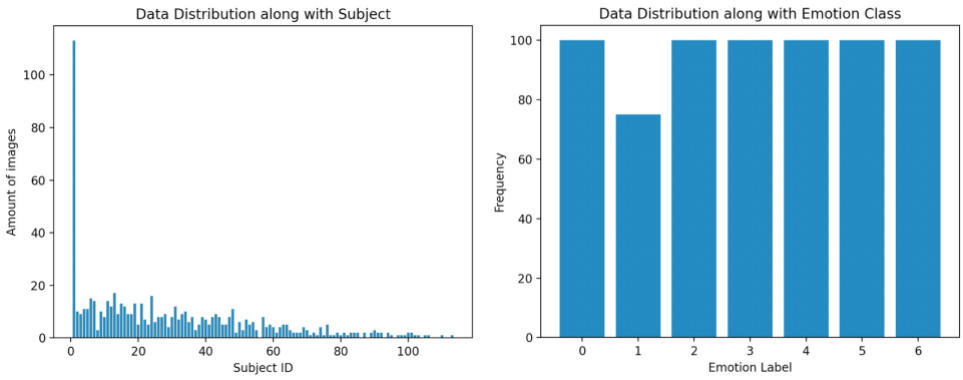


Fig. 2. Data distribution along with the subject and data distribution along with the facial emotion class in the SFEW database

Data Preprocessing. The experiment conducts three protocols to partition the data into training dataset and test dataset, consisting of strictly person specific (SPS), partial person independent (PPI) and strictly person independent (SPI) (Dhall et al., 2011). In practice, the facial emotion recognizer potentially classifies the facial expressions on the images containing people have not been seen. According to Dhall et al. (2011), SPS is the ideal situation that we suppose the model see the facial emotions from all people during the training phase; in contrast, SPI is the worst situation that the model does not see the facial emotions of any people in the testing dataset; the PPI partitions facial emotion data

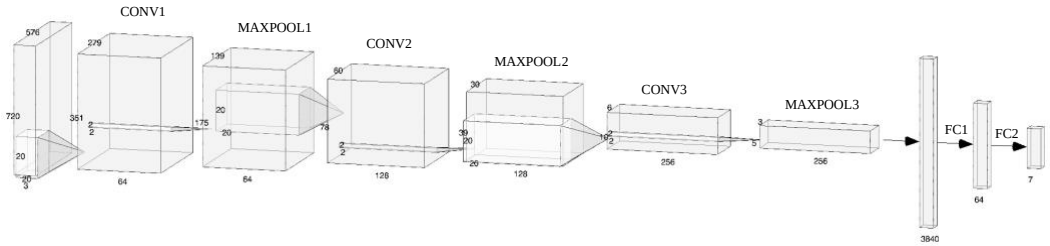
without considering the person information, which resembles the real-world situation. Therefore, the evaluation results of facial emotion recognizer obtained by PPI protocol is closer to the performance in practice. In this paper, we mainly use the PPI protocol to evaluate the performance of our FER classifiers and We briefly compare the performances under the three protocols.

The image file name contains the movie name and character IDs, for example, "Hangover_010805134_00000019" is a name of an image that "Hangover" is the movie name and "00000019" is the character ID in the movie (Dhall et al., 2011). Thus, the person information can be extracted from the image file name which can be used in the SPI and SPS protocols. In addition, Facial emotions labels are transformed to integers from 0 to 6. Images are fed into the CNN models directly and we apply normalization on each batch of dataset. The data of CPA features extracted by LPQ and PHOG are normalized on the entire dataset by subtracting the mean and dividing by the deviation.

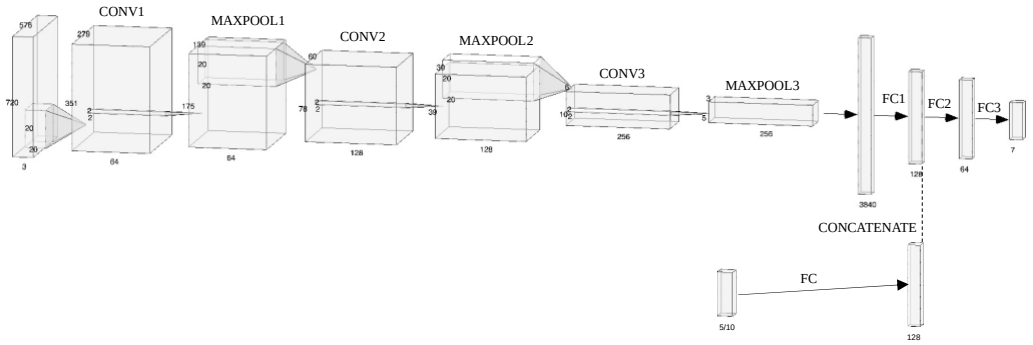
3 Method

3.1 CNN model

The FER classifier is basically implemented by a deep CNN model with 3 convolutional layers and 2 full-connected layers (see Fig. 3(a)). Basically, each convolutional layer doubles the filters, thus, the channel numbers are 64, 128 and 256 after each convolutional layer respectively. In addition, we stack a max pool layer after each convolutional layer to decrease the complexity. In the deep CNN model, we use LeakyReLU as the activation function in each convolution layer to avoid too sparse feature maps since some images are too dark that the people and the background are similar. Furthermore, we normalize the hidden features learned by each convolutional layer on the batch of data. After convolutional layers, we flatten the learned features of each image to a long vector to be fed into the last 2 full-connected layers for classification. In each full connected layer, we use ReLU as the activation function. In addition, we put a dropout regularizer at the first full-connected layer to avoid degradation and overfitting issues. As the FER model aims at classifying images into 7 classes, the activation function in the output layer is the logarithmic softmax. Therefore, we can use the negative log likelihood loss function to calculate the loss during the training. The optimizer in the experiments is stochastic gradient descent (SGD) with momentum.



(a)



(b)

Fig. 3. (a) The structure of the basic facial emotion classifier[1]. This is a DNN with 3 convolutional layers and 2 full-connected layers. (b) The structure of the facial emotion classifier processing the images and data of LPQ and PHOG features simultaneously[2]. The model concatenates a CNN and a full-connected model to train the images and LPQ-PHOG features respectively. And the classifier combines features learned from CNN and full-connected model to do the classification.

Based on the basic deep CNN model, we also implement a model concatenate a deep CNN model with a full-connected model (see Fig. 3(b)). The CNN model aims to train the images and the full-connected model aims to train the LPQ and

[1] The model is named as CNN_base in the paper

[2] The model is named as CNN_and_PHOG_LPQ in the paper

PHOG feature data. The FER model concatenates the feature vector learned by CNN and the feature vector learned by full-connected model to get a new feature vector which combines the features learned by CNN, LPQ and PHOG. In our experiments, we would like to compare the classification performances by different combination of features. In other words, we would like to compare four feature combinations, including pure CNN features, CNN and LPQ features, CNN and PHOG features, and all features. The activation functions in convolutional layers and full-connected layers, loss function and optimizer are same with the basic CNN model.

3.2 Autoencoder based on CNN

The Autoencoder is a feature selector which can learn high-level features from input data. It combines an encoder and a decoder which learns high-level features and reconstructs images respectively (see Fig. 4). The encoder contains 3 convolutional layers and there is a max pool layer follows by each convolutional layer. Symmetrically, the decoder contains 3 deconvolutional layers and each deconvolutional layer is stacked after a max unpool layer. Similarly, the activation function after each convolutional layer is LeakyReLU expect the last convolutional layer. The activation function in last layer is ReLU as the value range of the RGB images is 0-255. The learned features in each convolutional layer are normalized before passing to the next convolutional layer. In addition, the optimizer is SGD which is same with the approach in basic CNN model. As the decoder aims to reconstruct the images by the features learned from decoder, we use mean square error (MSE) as the loss function to calculate the similarity between raw images and reconstructed images.

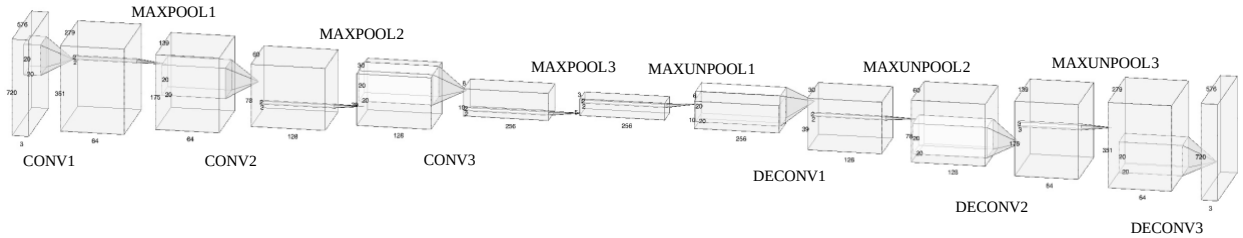


Fig. 4. The structure of the Autoencoder based on CNN. It combines by an encoder and a decoder. The encoder contains 3 convolutional layers, and the decoder contains 3 deconvolutional layers.

Basically, the CNN Autoencoder learns features for the images in the entire SFEW dataset. The shape of the feature map in the bottleneck layer is (256, 5, 3) which denotes 256 channels, width 5 and height 3. The features in the bottleneck layer are learned features from the encoder and we would like to use the bottleneck layer features to train a facial emotion classifier. The facial emotion classifier is implemented by a one-hidden-layer neural network and all layers are fully connected. We transform the bottleneck features of an image learned from the CNN Autoencoder to a long vector before feeding into the classifier. Therefore, the input layer contains 3840 neurons to fits the dimension of input data. There are 64 hidden neurons and the activation function at hidden layer is a TanH function to band the output value of hidden neuron between -1 and 1. The output layer contains 7 neurons as the SFEW database consists of 7 classes and the activation function of the output layer is a softmax function.

The shallow classification neural network is trained by K-fold cross validation as the SFEW database is quite small. The K-fold cross validation approach randomly divides the training dataset into K subsets. In each training epoch, we iteratively use K-1 subsets as the training data and the rest subset as the validation data. Hence, each subset can be used for training K-1 times, and we can pretend to have much more data which can be fed to the neural network. The cross-entropy loss function is used for training the classifier. It evaluates the probability distribution difference between the ground truth and the predicted result and attempts to minimize it, however, it might lead to overfitting (He et al., 2019).

Basically, we also use the shallow neural network trains on the combined LPQ and PHOG feature data. Therefore, the input layers should be adjusted to 10 neurons to fit the dimension of LPQ and PHOG feature data. Then, we can compare the classification performance based on CNN Autoencoder learned features and LPQ-PHOG feature.

3.3 Fine-tuning ResNet

The fine-tuning ResNet models implement some transfer learning by using the ResNet as the pretrained neural network and fine-tuning the model on the SFEW dataset. As ResNet is a state of art object detection model trained on the ImageNet database which contains image with real-world conditions, the SFEW dataset possess the similar real-world properties as well. It enables us to attempt a transfer learning model by using the pretrained ResNet on SFEW. In our experiments, we apply three fine-tuning approaches, including last layer fine-tuning (LLF) and all layer fine-tuning (ALF). LLF aims to discard the final full-connected layer of the ResNet and add a new full-connected layer with 7 neurons as the output layer. Then fine tuning the added output layer merely and the weights of the convolutional layers in ResNet are not updated. ALF aims to discard the final full-connected layer of the ResNet and add a new full-connected layer with 7 neurons as the output layer, which is similar with the LLF. However, it updates the weights of

all layers in the ResNet and the new added layer. In addition, we would like to evaluate the performance of the fine-tuning models based on different ResNet. In other word, we use the ResNets with 18 layers, 34 layers, 50 layers, 101 layers and 152 layers as the pretrained model. The loss function is cross-entropy as the activation function at the output layer is softmax. The optimizer is SGD without momentum as the transfer learning model is required to fine tune the parameters merely.

3.4 Regularization approaches

As the SFEW database contains small number of data and the full-connected layers of the shallow neural network in the CNN Autoencoder model section are wide, the training is potentially overfitting when training small dataset on a large parameter matrix. Generally, regularization strategies can help the model avoid overfitting. In our experiments, we compare the performances of three regularization methods, including weight decay, distinctiveness pruning and dropout.

Weight Decay. In the deep neural network training, overfitting might be occurred when training data is small, or the complexity of the neural work is high. And overfitting not only significantly impacts on the performance of neural network, but also decreases the computation efficiency of the model. Typically, the overfitting implies the parameters of the neural network are complicated that they can perfectly fit the training data, however, it fails to perform well on unseen data as the model is not generalized. To overcome the issue, weight decay is a prevalent strategy to generalize the model. Basically, it adds a regularizer on the loss function, commonly applying L1 norm and L2 norm. L1 norm can cause a sparse weight matrix that it prefers to zero some weights and L2 norm is more stable than the L1 norm (Luo et al., 2016). In our case, we use the L2 norm in the Adam optimizer to implement the weight decay purpose. According to the formulas (1), the first term denotes the cross-entropy loss, and the second term is the L2 norm regularizer. The weight decay focus on minimizing the L2 norm of parameter matrix, thus, it decreases the complexity of the neural network. Furthermore, the coefficient of the L2 norm is vital importance as a large coefficient may lead to underfitting.

$$\mathbf{L} = \sum_{c} \mathbf{y}^{(c)} \log \mathbf{p}^{(c)} + \lambda \|\boldsymbol{\omega}\|_2 \tag{1}$$

Pruning. Pruning is another popular approach to decrease the network complexity by removing some hidden neurons based on some benchmarks. In our case, we implement a pruning strategy by considering the distinctiveness between the functionality of hidden neurons, which is firstly applied in the research by Gedeon and Harris (1991). According to Gedeon and Harris (1991), the distinctiveness between a pair of hidden neurons reflects the angle separation of the hidden neuron functionality vectors. The authors indicate that the functionality of a hidden neuron is defined as the activated output of the neuron having the same dimensionality with the number of the input in a training batch. Since the activation function of hidden layer in our full-connected classifier is a simple TanH function banding the hidden output values in the range from -1 to 1, hence, we can calculate the cosine similarity between each pair of neurons' functionality vectors and the angle separation between each pair of neurons are bounded in 0º to 180º (Gedeon & Harris, 1991). According to Gedeon and Harris (1991), the angle separation which is lower than 15 º implies the two hidden neurons are too similar and one of them should be removed. On the other hand, a pair of neurons with 165 º or higher angle separation are functional complementary, and both should be removed.

Dropout. We also experiment the dropout strategy to regularize our full-connected classifier in CNN Autoencoder section. Dropout is a typical regularization method prevalently applied in the image recognition tasks based on the convolutional neural network (CNN). According to Krizhevsky et al. (2012), dropout approach aims at randomly mask 50% of hidden neurons in each epoch of training. When training process is completed, the entire neural network will be used for testing. Basically, it can reduce the complexity of the neural network structure to speed up the training phase and prevent overfitting.

3.5 Evaluation metrics

We basically use the testing accuracy as the evaluation metrics to compare the performances of different models with varied hyperparameters, training methods and regularization approaches. In addition, we compare our model performances with the baseline about 19% which is obtained by the non-linear SVM classifier in the research of Dhall et al. (2011). The expressions of accuracy can be seen in formulas (2), where *tp*, *tn*, *fp*, *fn* denotes true positive, true negative, false positive and false negative.

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{2}$$

4 Results and Discussion

4.1 CNN model performance

We implement a basic CNN model as the baseline of our experiments, and we compare the classification performances with different settings under PPI protocol. Basically, we use the test accuracy as the benchmark to evaluate the settings in the basic CNN model (see Fig. 5). We change the kernel size of all convolutional layers, and the CNN classifier obtains the highest test accuracy 41.73% when kernel size is 20. However, all test accuracies with different kernel sizes are located between 40% and 42%. The kernel size does not significantly impact on the CNN classifier performance as the dimension of the output long vector after the final convolutional layer is a 768 when kernel size is 25, which preserves sufficiently fine-grained information from the input images. In addition, we compare the classification performance by using the average pool and max pool after each convolutional layer respectively, the max pool approach introduces better performance than the average pool approach. The test accuracy of the model applied max pool and average pool accuracies are around 35% and 40% respectively. Furthermore, we compare the performance of the CNN classifier by using different combinations of kernel size and stride, including (10, 2), (10, 4) and (25, 2)[3]. And the CNN model with the combination of kernel size and stride (10, 2) obtains the highest performance. Furthermore, the LeakyReLU with different negative slopes do not impact the performance of the CNN classifier dramatically. According to Fig. 5, the test accuracy of 0, 0.1, 0.01 and 0.001 negative slopes are all around 45% to 49%, amongst, the model using LeakyReLU with 0.01 negative slope obtains the highest test accuracy about 48.68%. Lastly, we compare the CNN with different number of full-connected layers and the effect of dropout. Basically, we have a 2 full-connected layer CNN classifier and a 4 full-connected layer CNN classifier, and the dropout regularizer is put at the first full-connected layer in these two models. According to Fig. 5, the two models present similar performance as the convolutional layers have learned the features well. Therefore, the deepness of the final full-connected layers cannot help to learn more features. In addition, when we add dropout regularizer on all hidden full-connected layers in the 4 full-connected layer CNN classifier, the convergence speed of the model significantly decreases, however, it can obtain the similar test accuracy with the 4 full-connected layer CNN classifier which only contains dropout regularizer in the first full-connected layer. Therefore, the dropout regularizer can help the CNN model get a more stable training result.

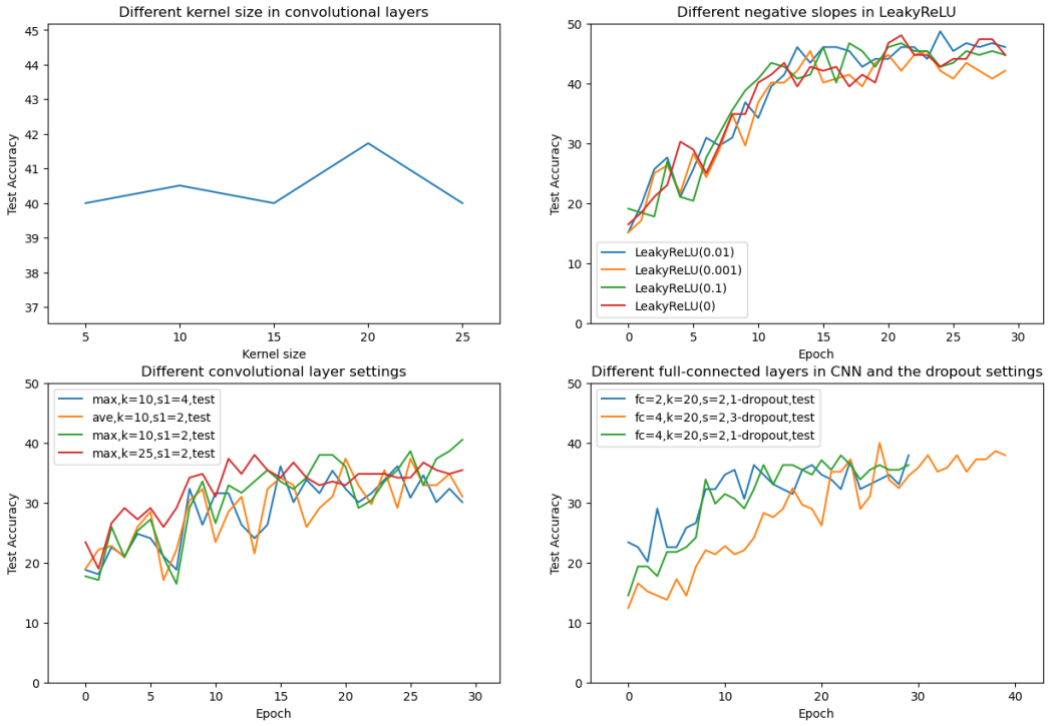


Fig. 5. The test accuracy of basic CNN model with different settings.

In addition, we attempt different data partition protocols when training the basic CNN classifier and the result can be seen in Table 1. Obviously, the CNN model improves the FER performance on SFEW significantly. Under SPI protocol, the test accuracy of the CNN classifier is 39.39%, which is much higher than the baseline accuracy, 19% (Dhall et al., 2011). And the "BaseModel_K_fold" which is a full-connected neural network with 1 hidden layer also obtains better performance when using the LPQ-PHOG feature data, about 27.27% test accuracy under SPI protocol. Generally, the

[3] (10, 2), (10, 4) and (25, 2) are combination of kernel size and stride: (kernel size, stride)

convolutional layers can learn better features than the LPQ and PHOG descriptors and the output layer using full-connected layer performs better than the SVM.

Table 1. Test accuracies of models based on PPI, SPI and SPS protocols

Model Name	PPI	SPI	SPS
CNN_base	41.73%	39.39%	42.42%
BaseModel_K_fold[4]	31.58%	27.27%	25.62%
SVM (baseline) [5]	-	19.00%	-

Furthermore, we train a CNN_and_PHOG_LPQ model concatenating a CNN and a full-connected model demonstrated in Fig. 3(b) and the performance can be seen in Fig. 6. Obviously, the performance of the model combining the CNN features and LPQ/HPOG features is higher than the model with pure CNN model. It implies that the combination of different feature selection approaches can extract more useful features as different approaches focus on different aspects of data. The combination strategy can preserve the advantages from different feature selection methods, which can obtain better performance.

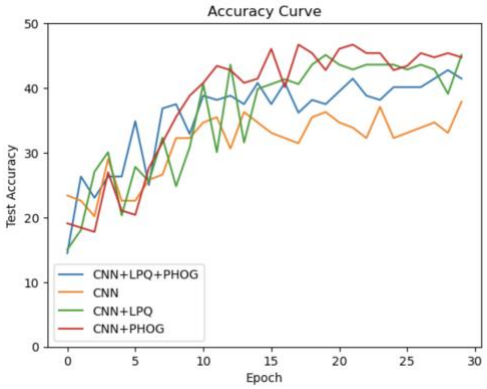


Fig. 6. The test accuracy of CNN_and_PHOG_LPQ model.

4.2 CNN Autoencoder model performance

In our experiments, we implement a CNN Autoencoder to learn the high-level features of images in SFEW dataset. We train the CNN Autoencoder with 100 epochs to get convergence, and the sample of reconstructed image can be seen in Fig. 7. We use the feature maps in the middle bottleneck layer in the CNN Autoencoder as the input to the full-connected 1 hidden layer neural network. Before feeding the learned features into the shallow neural network, we flatten the feature maps of each image to a long vector with length 3840. In addition, we apply L2 norm weight decay, distinctiveness pruning and dropout on the shallow neural network to compare the classification performances (see Table 2). Both L2 norm weight decay and distinctiveness pruning can obtain higher test accuracies, about 43.51% and 43.18% respectively. It implies the two regularization approaches can effectively generalize the model and avoid overfitting without sacrificing the classification performance. However, the test accuracy of model applying dropout regularizer is much smaller than the accuracy of the model without any regularization methods. It seems that the dropout regularization can perform well on CNN model, but it cannot preserve a good performance on a shallow neural network even it contains large number of hidden neurons, as the loss is oscillated dramatically that the shallow neural network would easily jump out from a local minimum.

Reconstructed image Raw image

Fig. 7. The sample reconstructed image obtained by CNN Autoencoder

[4] The model is a 1-hidden layer neural network useing the LPQ-PHOG feature data as input
[5] The model in the research from Dhall et al. (2011)

Table 2. Test accuracies of the full-connected classifier trained on features learned by CNN Autoencoder over different regularization strategies based on PPI protocol

Model Name	Regularization Method	Test accuracy
BaseModel_K_fold	-	41.73%
BaseModel_K_fold	Weight decay[6]	43.51%
DistinctivenessPruning_K_fold	Pruning [15º, 165º]	43.18%
ModelDropout_K_fold	Dropout	32.84%

4.3 Fine-tuning RestNet performance

In our experiments, we compare the performances of different fine-tuning ResNet model and different fine-tuning method (see Fig. 8). We fine-tune the ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152 with 0.0005 learning rate. The suffix FC of label names in legend in Fig. 8 denotes the LLF fine-tuning method and the suffix ALL denotes the ALF fine-tuning method. Obviously, the transfer models applying ALF fine-tuning method obtain higher test accuracies than the models applying LLF, as the SFEW dataset is not very similar with the ImageNet dataset. All models trained by ALF gain the test accuracies around 45% to 58%, where ResNet18 obtains the largest test accuracy about 57.43%. And the ResNet with larger layer number performs worse, because the deeper ResNets learn more specific features from ImageNet dataset that they cannot apply the pretrained weights to learn the features well on SFEW dataset.

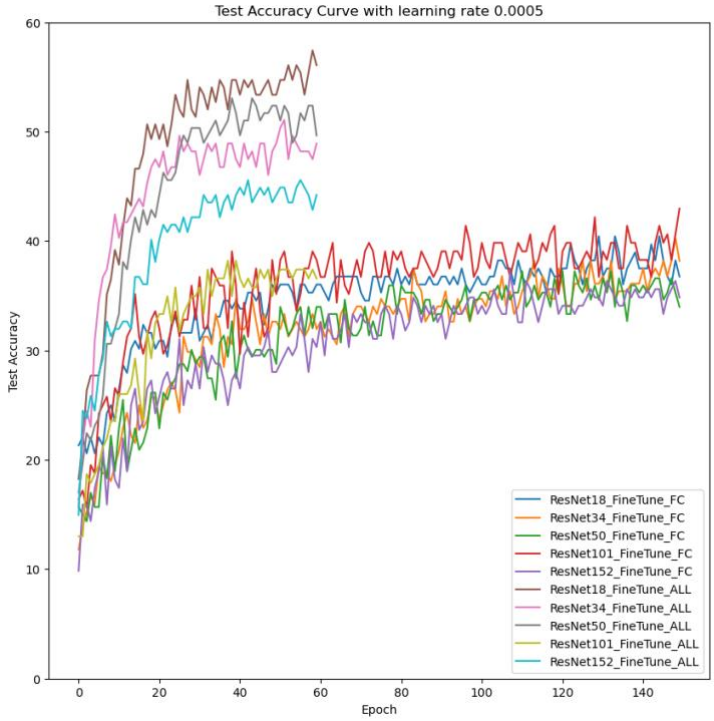


Fig. 8. The test accuracy of fine-tuning ResNets.

4 Conclusion and Future Work

In this report, we construct several neural network models to perform as a facial emotion classifier based on SFEW which possesses the near real-world properties. In comparison with the baseline accuracy implemented by a non-linear SVM classier, both basic CNN classifier and the shallow neural network with one-hidden layer perform better FER capacities, with 39.39% and 27.27% testing accuracies under SPI protocol respectively. In addition, the basic CNN classifier obtains the highest classification performance when the 20-kernel size, 2-stride value, 0.01-LeakyReLU negative slope and max pool method in each convolutional layer, which obtains maximum 48.68% test accuracy. Furthermore, the CNN_and_LPQ_PHOG model concatenating the features learned by CNN with the features of LPQ-PHOG can obtain better performance than the pure CNN model. Moreover, the full connected neural network with 1-hidden layer can get an acceptable classification accuracy around 41.73% when using the learned features from CNN Autoencoder as input. And the L2 norm weight decay and distinctiveness pruning can decrease the model complexity

[6] L2 norm ($\lambda = 0.001$)

without sacrificing the performance which can get 43.51% and 43.18% test accuracy respectively. However, the dropout method sacrifices the model performance as it may easily jump out from the local minimum in the shallow neural network. Lastly, the ResNet18 with ALF fine-tuning method can obtain highest classification performance on SFEW with 57.43% test accuracy. And the deeper ResNets are not fitting well with the SFEW dataset as they learn too specific features on ImageNet dataset.

Currently, we implement some deep neural network models on SFEW with good performance. However, we mainly use the raw images as the input for the CNN models, which triggers considerable computation complexity. In addition, the background in some images occupies large area which may cause worse learning result. There are many facial cropping methods on real-world images which can eliminate many noises from images (Koestinger et al., 2011; Li et al., 2015). Thus, we can apply some facial detection methods to crop the faces of people form the SFEW images before applying the FER classification to try to get higher performance.

References

Dhall, A., Goecke, R., Lucey, S., & Gedeon, T. (2011, November). Static facial expressions in tough conditions: Data, evaluation protocol and benchmark. In *1st IEEE International Workshop on Benchmarking Facial Image Analysis Technologies BeFIT, ICCV2011.*

Gedeoal, T. D., & Harris, D. PROGRESSIVE IMAGE COMPRESSION.

Gedeon, T. D., & Harris, D. (1991). Network reduction techniques. In *Proceedings International Conference on Neural Networks Methodologies and Applications* (Vol. 1, pp. 119-126).

Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256). JMLR Workshop and Conference Proceedings.

He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., & Li, M. (2019). Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 558-567).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.

Koestinger, M., Wohlhart, P., Roth, P. M., & Bischof, H. (2011, November). Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)* (pp. 2144-2151). IEEE.

Luo, X., Chang, X., & Ban, X. (2016). Regression and classification using extreme learning machine based on L1-norm and L2-norm. *Neurocomputing*, 174, 179-186.

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101.*

Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2015). A convolutional neural network cascade for face detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5325-5334).

Pasupa, K., & Sunhem, W. (2016, October). A comparison between shallow and deep architecture classifiers on small dataset. In *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)* (pp. 1-6). IEEE.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research,* 15(1), 1929-1958.

Wang, S., Ding, Z., & Fu, Y. (2017, February). Feature selection guided auto-encoder. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 31, No. 1).