Emotion Classification based on SFEW using Modern Image Classification Methods

Xingnan Pan

Research School of Computer Science, Australian National University, Australia
u6744662@anu.edu.au

Abstract. This paper uses the SFEW emotion dataset to implement emotion classification. We will implement emotion image classification in two ways: the traditional and the modern way to classify images. For the traditional way in this paper, we use PHOG and LPQ to extract the features vectors, use the first five principal components as the input of a vanilla neural network classifier; while in the modern way, we use deep learning algorithm, the pretrained ResNet and ResNext, and modified it to fit our task. In this paper, we aim to reproduce the modern methods to implement Image/Emotion classification.

Keywords: Deep learning Emotion Classification, ResNext, ResNet, Face detection, Transfer learning

1 Introduction

In the past 10 years, machine learning and deep learning have become the hot spots of the times. Countries and companies are constantly exploring and seeking the transformation of data-driven business models. Among them, neural networks are the key to deep learning. The emotion classification is the relatively nascent research area, and it plays a vital role to psychology and sociology. Emotion classification, similar to image classification, generally has two methods to achieve. The traditional method is to first extract the feature vector of the picture, and then use the feature vector as input to train a classifier; while the modern approach is to use a deep learning model to directly use the picture as an input and then train the model to get the predicted value. It is an end-to-end method. Compared with traditional methods, deep learning models are easier to implement and have higher accuracy. However, since deep learning models have millions of parameters, deep learning generally requires a lot of computing power, and the interpretability of the model is very poor. In this article, we will use the SFEW emoticon dataset to implement algorithms. And see how the algorithms perform on relatively small dataset. In particular in this paper, we will implement vanilla neural network, ResNet, ResNext, Transfer learning, Image pre-processing and other modern image classification related methods.

2 Dataset & Pre-processing

The Face Emotion dataset, Static Facial Expressions in the Wild (SFEW) [1] is used in this paper. The SFEW is extracted from a temporal facial expression dataset Acted Facial Expression in the Wild (AFEW) [3], in which the facial emotion images are extracted from movies. The images in the dataset are more natural and realistic because those images are captured in a close to real-world environment instead of lab-control environment. **Figure 1** shows the sample images from the SFEW dataset.

The dataset has **675** items in total, and contains **seven** facial expressions: Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise. The images' size is 720×576 with RBG colour space.

PCA Extraction and Normalization

In this paper, we will use PCA vector as the input of the vanilla neural network model. The images in dataset are processed by two descriptors: LPQ [4] & PHOG [5] to generate the pattern vectors. In order to reduce the input data complexity, Principal Component Analysis (PCA) is applied for both pattern vectors. The first five components are kept which contain 98% of the variance. Therefore, the input dimension is 10 in total by combining two of first-five PCA vector.

The normalization is applied to the input by using (1) equation. Normalization can map input values in different dimension to similar range of values, leading each principal component to have similar contribution during the training neural network process. Also, the original PCA element value is small. For instance, the values in the first dimension have the range of [-0.0109, 0.0096]. Normalization can ensure the input roughly in the range of [-1, 1],

which can allow the network to learn more quickly. Where $\tilde{x}$ means the normalized value, and μ means the mean value, and σ stands for standard deviation.

$$\tilde{x} = \frac{x - \mu}{\sigma} \quad (1)$$

Figure 1. Sample images from the SFEW dataset [1]

Image Preprocessing

Original dataset. To fit the image size to the original Resnet model, all the images resize to 224×224×3, representing height, width, and channel respectively. And all images are normalized by the mean [0.485, 0.456, 0.406] and the standard deviation [0.229, 0.224, 0.225] on three channels respectively, using the formula (1). This mean and standard deviation value are calculated on ImageNet [2] millions of images. It is officially suggested by the PyTorch.

The images in training set and test set are processed in different ways. For the images in the training set, we use some random image processing methods to increase the generalization of the model. In particular, the images in training set are processed by random horizontal flip and random affine.

Figure 2. Original image and processed image

Crop faces dataset. Adam have published an easy way to detect faces on images [6]. It is an algorithm by using HOG to get feature vector and training SVM linear classifier to get the result. We use this algorithm to detect and crop faces from the original images. However, this method is not perfect, having **238** images haven't been detected faces. The poor performance may be caused by the light of the movie images is relatively dark, and the characters are easy to blend with the background and images have complex backgrounds. Consequently,we use the following policy: for those images are falsely detected faces, we keep the original image, while for those positivelt detected, we crop the face and save as the new input. **Figure 2** shows samples of the original images and the coresponding crop faces. **We only use this dataset in resnet50.**

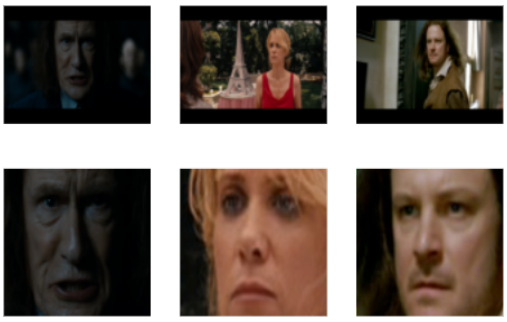

Figure 3. Original faces image (upper) & Crop faces (bottom)

3 Methods

Vanilla neural network

Vanilla neural network is a classical machine learning algorithm. We construct a three-layer vanilla neural network with one hidden layer. The input of the model is two first five principal components of two feature vector, generated by LPQ and PHOG respectively. Hence, the input size is 10. The activation function of the hidden layer is Sigmoid. The experience will try different hidden layer size: [16, 32, 64, 128], which the latter size just doubles the previous size. The **Figure 4** shows the network structure, where *n* in brackets represent the number of layer size. The activation function is Tanh function with the output range: [-1, 1].

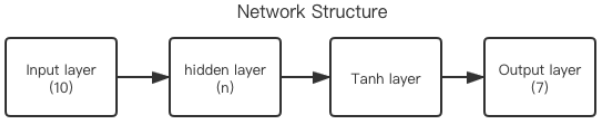

Figure 4. Network Structure

ResNet

ResNet [7] is a classical deep learning image classification algorithm. Before ResNet was invented, the deep learning algorithms suffered from the problem that the more layers the model have, the higher probability of the model would have gradient vanishing or exploding so that the model cannot be trained. ResNet address this issue by "residual connection", which is type of skip-connection that learn residual functions with reference to the layer input. **Figure 5** shows the structure of the residual connection.

Denoting that, $H(x)$ is the desired underlying mapping, but it is difficult to learn. We use another non-linear mapping $F(x)$ and the identity x to represent the $H(x)$, having $F(x) = H(x) - x$, the residual mapping. Hence the model change to learn the residual mapping rather than the original mapping. Intuitively, it is easier to optimize the residual mapping. For example, if the identity is optimal, then the model would find it easier to push the residual to zero than to fit an identity mapping by a stack of non-linear mapping. This paper will experience on ResNet50, with the network structure show in **Figure 7.**

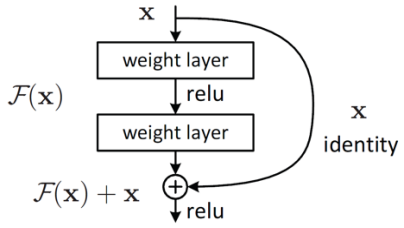

Figure 5. Residual Connection [7]

ResNext

ResNext is the enhanced variant of the ResNet, introduced by Xie and his team [8]. The traditional method to improve the accuracy of the model is to deepen or widen the network, but as the number of hyperparameters increases (such as the number of channels, filter size, etc.), the difficulty of network design and the computational overhead will also increase. Therefore, the ResNext structure can improve the accuracy without increasing the complexity of the parameters, while also reducing the number of hyperparameters.

The main difference between ResNet and ResNext is the building block. In **Figure 6,** the LHS figure shows the building block structure of ResNet while RHS shows the block structure of ResNext. As we can see, the building blocks' structure are very similar. Compare the ResNet, ResNext aggregated a set of transformation with the same topology, where the size of the set of transformation is measure by *Cardinality*. In the Figure 6 RHS, the architecture includes 32 same topology building block, hence the Cardinality is 32. Meanwhile, because the block uses the same topology, fewer parameters are needed. The original paper [8] states that results demonstrate that increasing cardinality is a more effective way to increace the accuracy than made the model deep or wider.

Figure 7 shows detailed network structure of ResNext compare to ResNet.

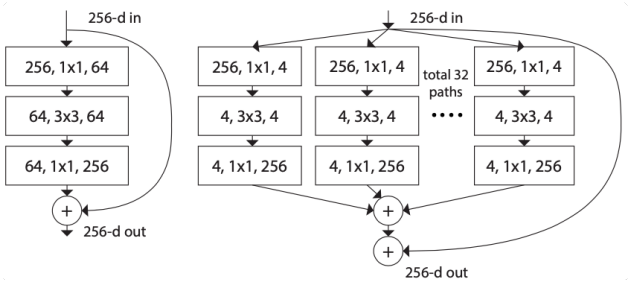

Figure 6. Building block structure of ResNet (left) and ResNext (right)

stage	output	ResNet-50	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 128 3×3, 128, C=32 1×1, 256] ×3
conv3	28×28	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 256 3×3, 256, C=32 1×1, 512] ×4
conv4	14×14	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 512 3×3, 512, C=32 1×1, 1024] ×6
conv5	7×7	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 1024 3×3, 1024, C=32 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		**25.5**×10^6	**25.0**×10^6
FLOPs		**4.1**×10^9	**4.2**×10^9

Figure 7. ResNet & ResNext architecture

Transfer learning.

The transfer learning is taking the knowledge of the neural network has learn from one task and apply that knowledge a separate task, which means the knowledge is transferred from one model to another. Using transfer learning, we can solve a particular task using full or part of a pretrained model on different task. Transfer learning is very useful in our case, since our dataset only contain a small amount of data, 675 images for 7 class in total. As a result, for the Resnet and ResNext, we use the official pretrained model from PyTorch, the pretrained **resnet50**, **resnext50_32x4d**.

Since the pretrained models are not designed to solve our problem, we modified the last fully connection layer to two linear layers and adding activation functions. The modified layer structure shows in Figure 7.

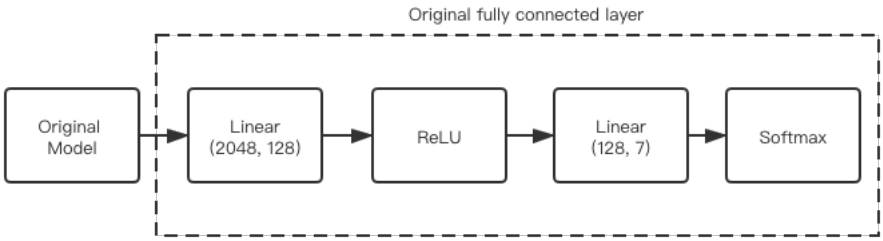

Figure 8. Modified layers structure

4 Result and Discussion

PCA vanilla neural network

Different hidden layer sizes have been tested: [16, 32, 64, 128]. The latter size just doubles the previous size. The network of this structure is constructed because it is hoped that by constructing a double size network, it could be more intuitively see the impact of different hidden layer sizes on the accuracy of the network. Other relevant hyperparameter are listed here. Batch size is 16; learning rate is 1e-4.

Measurement of model. In this model, the **Accuracy** is used as the main performance descriptor. The experiences run though the five-fold cross validation script and take the average value. The Accuracy is evaluated by the test set. The numerical results show in the **Table 1**. The results are the average value of five fold's results.

The benchmark of the dataset in original paper is 19%, which is used non-linear SVM to classify. In comparison, it is surprising that the accuracy of all models has exceed 19% which can demonstrate that neural network has better Accuracy performance than SVM for this task. In the table 1, it is obvious that bigger network tent to have better performance than the smaller network or at least has similar performance. Among all models, the model with 128 hidden units has the highest accuracy.

The class-wise accuracy of model with 128 hidden units shows in **Table 2.** Among all emotions, Disgust has the worst accuracy, while the Neutral emotion is second worst. The result is similar to the original dataset paper. These two emotion have common character that it does not involve fewer facial muscles than other emotions , which may be reason for the poor performance.

Table 1. Vanilla neural network accuracies over different hidden size

PCA Vanilla neural network				
Hidden size	**16**	**32**	**64**	**128**
Accuracy	24.47	26.61	27.26	28.25

Table 2. Hidden 128 Vanilla neural network class-wise accuracy

class-wise accuracy							
class	**Angry**	**Disgust**	**Fear**	**Happy**	**Neutral**	**Sad**	**Surprise**
Accuracy	27.25	15.25	41.47	32.25	19.71	31.69	29.84

Deep learning algorithms results

We experience the SFEW dataset on three different models. All three models would run on 5-fold cross validation script, and take the average value over five folds. All models trained by 300 epoches, with the learing rate is 1e-5 and batch size is 16. All experiences in this paper run on Google Colab with GPU acceleration.

The ResNet50 is a pre-trained model with modified last two layers. The ResNext50_32x4d is also a pre-trained model with modified last two layers. On the other hand, ResNet50 with crop faces uses the same network as ResNet50, but uses the crop face dataset which contain mostly are faces and few are orginal images data.

Table 3 shows the accuracy result of these models. The benchmark of the original dataset is about 46%, where LPQ is 43.71% and PHOG is 46.28%. As we can see that, all model performance is significantly higher than the benchmark. It could demonstrates that, deep learning algorithms works better than machine learning in this field.

The ResNet50 has almost the same performance with the ResNext50, mainly because these two models have a very similar network structure. Though, the overall accuracy of ResNext50 is slightly higher than the ResNet 50.

Among three models, ResNet50 with crop faces has the worst performance. The crop face does not help model gain accuracy improvement, instead it decreases the model performance. It may be caused by the different data distribution of the dataset, as 243 images are kept using the original data. The original data have an obviously different distribution with the crop faces. As a result, the model might be confused to learning which kind of knowledge.

Table 3. Class-wise and overall Accuracy of ResNet, ResNext50_32x4d, ResNet50 crop face

Class-wise Accuracy								
Class	**Angry**	**Disgust**	**Fear**	**Happy**	**Neutral**	**Sad**	**Surprise**	**overall**
ResNet50	0.51	0.58	0.62	0.58	0.33	0.59	0.43	0.52
ResNext50_32x4d	0.50	0.58	0.54	0.58	0.36	0.59	0.50	0.53
ResNet50 with crop faces	0.37	0.53	0.46	0.58	0.30	0.58	0.43	0.48

Compare the Deep learning model with the PCA vanilla model, it is no doubt that deep learning models have better performance. However, an interesting finding is that the Disgust emotion is no longer the class having the worst accuracy instead it has a relatively high accuracy compare to other emotions.

5 Conclusion and Future work

In summary, this paper introduced the more realistic emotion dataset SFEW. Also, paper has introduced one machine learning algorithm and popular image deep learning algorithm: ResNet and its extension ResNext, and briefly explain its theory and their connection.

The result shows that although the vanilla network is simple and have limited inputs, the performance of vanilla network is surprising. The accuracy significantly exceeds benchmark 19% at about 28%, which shows that it can effectively make prediction to some extension. But the Disgust emotion is barely classified by the vanilla network, with about 15% accuracy.

On the other hand, three different deep learning model have similar performance about 50% accuracy on seven emotion classification. Among three models, ResNet50 using crop face dataset has the worst performance. It may cause by inconsistent data distribution of the dataset as 25% of this dataset keep using the original image. Meanwhile the Disgust emotion could be easily classified by these models, having about 55% accuracy.

Future work

The crop face dataset is not perfect in this paper and the result of this model is not convincing. As a result, it might use other face detection or manual extract feature from images to construct the dataset. For object detection or face detection, the YOLO algorithm is suggested.

In terms of other model image classification method, Autoencoder and Variational Autoencoder are suggested. They both are the popular algorithm for feature extraction and classification.

Reference

[1] Dhall, A., Goecke, R., Lucey, S., & Gedeon, T. (2011). Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 1–7. https://doi.org/10.1109/iccvw.2011.6130508

[2] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255).

[3] Dhall, A., Goecke, R., Lucey, S., & Gedeon, T. (2011). Acted facial expressions in the wild database. Australian National University, Canberra, Australia, Technical Report TR-CS-11, 2, 1.

[4] V. Ojansivu and J. Heikkil. Blur Insensitive Texture Classi- fication Using Local Phase Quantization. In Proceedings of the 3rd International Conference on Image and Signal Pro- cessing, ICISP'08, pages 236–243, 2008.

[5] Bosch, A. Zisserman, and X. Munoz. Representing Shape with a Spatial Pyramid Kernel. In Proceedings of the ACM International Conference on Image and Video Re-trieval, CIVR '07, pages 401–408, 2007.

[6] Geitgey, A. (2020, September 24). *Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning*. Medium. https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

[7] He, K., Zhang, X., Ren, S. & Sun, J. (2015). Deep Residual Learning for Image Recognition (cite arxiv:1512.03385Comment: Tech report)

[8] Xie, S., Girshick, R.B., Dollár, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5987-5995.