Feature extraction performances of convolutional networks under noisy environments

Daniel Hao

Research School of Computer Science, Australia National University u6055952@anu.edu.au

Abstract. In recent years, convolutional networks have achieved state-of-the-art performances in numerous computer vision tasks such as image classification and object tracking. They have demonstrated promising results in extracting high- and low-level details from images. Despite its wide usage and impressive performances, convolutional networks suffer from the same problem as any other artificial neural network; they require large amounts of data and are extremely sensitive to the environment. In this paper, we explore the feature extraction performances of a convolutional network under noisy environments for a image classification task using a small data set. We compare the final accuracies of a classifier using features extracted by a convolutional network to a classifier using features extracted by pyramid histogram of oriented gradients (PHOG) and local phase quantisation (LPQ). Results demonstrate that convolutional neural networks can extract high quality features, performing better than PHOGs and LPQs even under noisy environments.

Keywords: Artificial neural networks · Convolutional neural network · Image clasffication · emotions.

1 Introduction

1.1 Background

Convolutional neural networks (CNN) are an extension of traditional artificial neural networks by adding extra depth and constraints to early layers. They take pixel values of images as input and outputs an array of features extracted by the convolutional layers. The usage of CNNs has long been applied in the field computer vision dating by to the early 1980s [1], however it's only with the recent boom of computing power and availability of big data that it has finally had its chance to shine. Despite its glory, CNNs shares the same Achilles' hill as tradition artificial networks, requiring large amount of training data and extremely sensitive to environment. Under certain circumstances, CNNs may not be the ideal choice for image classification regardless of its ability to model abstract representations. As the focus of the research shifts to understanding the feature extraction behaviour of convolutional networks, network minimization techniques such as Casper[12] was not implemented. While they do offer slight advantages such as better final performances and minimal network sizes[11], we've decided to abandon them to maintain consistencies and fairness in the feed forward network during comparison and evaluation.

1.2 Data set

The data set used for the image classification task is from the Static Facial Expressions in the Wild database[2]. The data set contains 675 images that have been labelled for six basic expressions: *angry, disgust, fear, happy, sad, surprise* and also including the *neutral* expression[2]. The 675 images consist of a balanced mix of expressions of 100 each except for the disgust class, which only has 75 samples in contrast to 100. The task of the classifier is to classify each image to one of the expression classes. The expressions are labelled 1 - 7 as demonstrated in table 1

Original images in PNG formats as well as local phase quantization descriptors (LPQ) and pyramid of histogram of oriented gradient (PHOG) descriptors were used for the classification task. For the convolutional neural network, images are directly fed into the network with no further preprocessing of facial expressions. In contrast, PHOG and LPQ descriptors received images of faces localised by the Viola-Jones face detector [2] and then performed feature extraction on the localized images. Figure 1 demonstrates the values extracted by LPQ and PHOG descriptors.

Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
1	2	3	4	5	6	7

Table 1. Expression labels

label	First 5 Principal Components of Local Phase Quantization (LPQ)					First 5 Principal Components of Pyramid of Histogram of Gradients (PHOG) features				
1	-0.0008172	0.0034698	-0.0075171	-0.010912	-0.0054296	0.0095511	0.0067755	0.0035193	-0.0009996	0.0043082

Fig. 1. Values extracted by LPQ and PHOG

The SFEW database[2] was selected as it contained relatively realistic images of various expressions with large amount of noise. Unlike the MMI database[7], which contained facial expression captured in lab-controlled environments, SFEW extracted its data from movies scenes, allowing it to have more realistic expressions in a dynamic environment.

2 Method

2.1 preprocessing

Origin data extracted by LPQ and PHOG contained 12 columns of data, including the emotion labels, the file name of the original image file and 5 features each for the LPQ descriptor and PHOG descriptor. For the purpose the experiment the file names were removed as it has no correlation with the emotions the images contain. The labels were also relabelled from 1 - 7 to 0 - 6 respectively for serving as the target of classification as shown in table 2. All 10 features from the LPQ and PHOG descriptors were used as inputs to the networks. 25 extra *disgust* data samples were artificially added to the data set to create a more balanced data set, resulting in a data set of size 700.

Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
0	1	2	3	4	5	6

Table 2. processed expression labels

Image fed in the CNN was first rescaled from the original resolution of 720×576 to 360×288 for faster training, the aspect ratio of the image was maintained as it may impact the quality of features extracted [10]. No cropping was made, since the location of faces in the images were extremely inconsistent, random cropping may result in important features being removed. Random horizontal flips and normalizations of mean 0 and standard deviation of 1 was applied to the images to combat overfitting and improve convergence through reducing the sparseness of input data.

For both data sets the data is shuffled randomly, independent of its source and the subject of the expression, such that we have a Partially Person Independent data set (PPI)[2]. The shuffled PPI data is then split with a 90/10 ratio respectively for training and testing. Strictly Person Independent (SPI) data sets were not constructed as we do not possess information about the individual in the image, but only the movie source of the image. It is thus impossible to split the data set into training and testing sets in a way such that no subjects in the training set are present in the testing set. Splitting the data set according to the movie source of the image was considered, however it was not proceeded with as there is not guarantee that different movies contained completely different actors, especially since it contained movies with sequels, such as Harry Potter.

2.2 PHOG and LPQ with feed forward network

The artificial neural network using features extracted by LPQ and PHOG with 2 hidden layers, 9 and 8 neurons in each layer respectively will serve as the baseline for the experiment. The network uses RPROP as the optimiser with a learning rate of 0.2. Hyperbolic tangent function was chosen as the activation function as it yielded better results than sigmoid[5] and does not suffer from the dying Relu problem. The weights were initialized with Pytorch default weight initializations and cross entropy loss was used for loss calculation. Random dropout was also applied to the fina hiddenl layer to combat overfitting. The resulting network was trained over a total of 1200 epochs, iterating through each entry in the training set around twice. The training length was decided through experimental trials, resulting network was able to generalize reasonably with the provided data without overfitting on the training data.

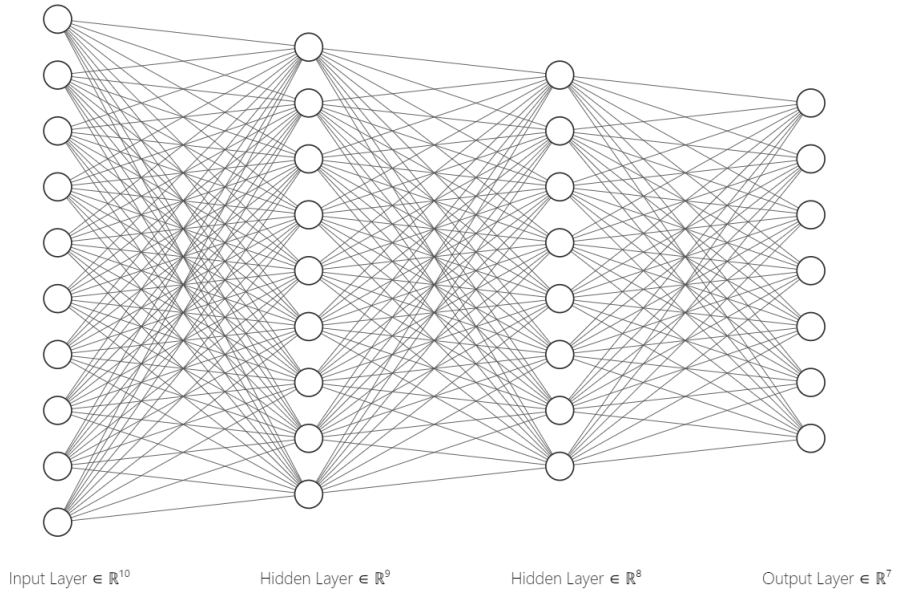

Fig. 2. Fully connected neural network

2.3 CNN structure

The convolutional network built for this 7 class classification tasks takes inspirations from the classical network structure VGGNet[9]. Mimicking the structures of VGGNet, we construct a convolutional network starting with 32 filters and doubles after every max pooling layer. Max pooling follows the pattern of pooling every 2 convolutional layer, performed over a 2×2 pixel windows with a stride of 2. Dropout is applied within both the fully connected layers and as well as convolutional layers[13]. Random dropouts was applied between convolutional layers to reduce overfitting by preventing co-adaption of feature detectors [3]. Dropout rates are initialized with a low probability of 0.2 [8] , applied after each max pooling layer. Batch normalization was also applied after every convolutional layer to aid regularization and reduce overfitting[4]. Due to significantly smaller data sets, we do not fully follow the design of the VGGNet[9] of going up to 16 convolutional layers and simply stop after 4 convolutional layers. We use a relatively small batch size of 16 for the same reason, by choosing a smaller batch size, the network was able to train effectively over the small amount of data.

As shown in figure 3, the convolutional layers is followed by a single hidden layer with 200 hidden neurons. Both the fully connected layers as well convolutional layer uses the leaky relu as the activation function [14], relu was not

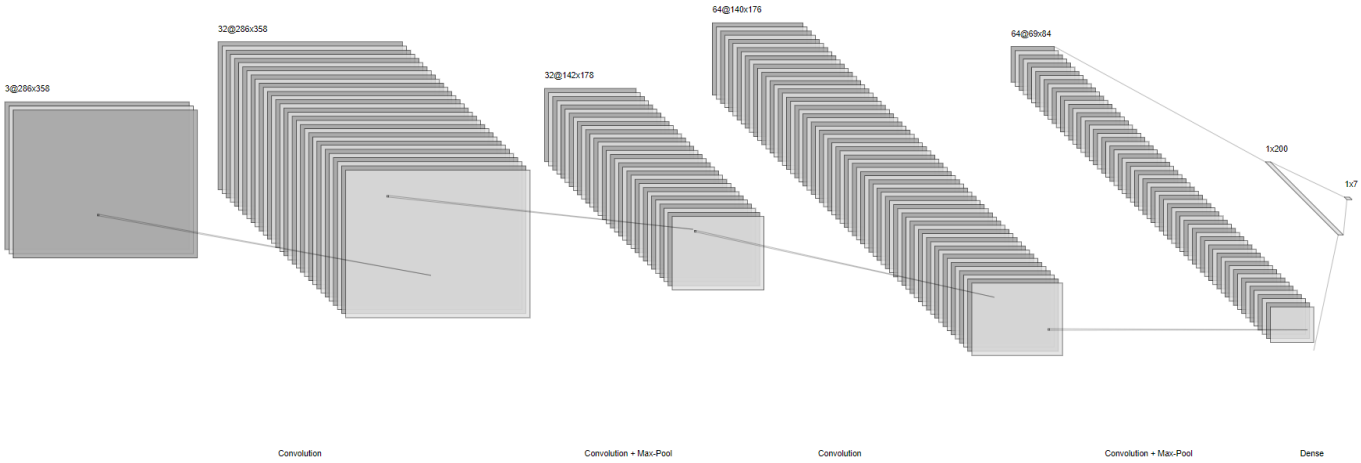

Fig. 3. Convolutional neural network architecture

used as the network suffered heavily from dying relu, leaky relu was chosen to combat this problem. The network was optimized with stochastic gradient descent with a learning rate of 0.01 and a momentum of 0.5. Training was regularized by a weight decay value of 0.0005 as well random dropouts. For loss calculation, we use cross entropy loss; a combination of logarithmic SoftMax and negative log likelihood loss. Training was completed over 12 epochs; empirical evidence show that further training did not improve generalization of the final network.

2.4 Cross Validation

To evaluate the final performances of the convolutional network, we use 10-fold cross validation to reduce the variance in the results[6]. 10-fold cross validation randomly splits the data set into 10 even subsets, in every iteration a different subset is chosen to be the test set while the rest becomes the training set, this process is repeated 10 times and a final average accuracy is produced from this validation method. By using cross validation we've wasted a minimal amount of a data and minimized the variance in the tests results.

3 Results and Discussion

The final Convolutional neural network was trained and evaluated with 10-fold cross validation, table 3 shows the results of one single run of 10-fold cross validation.

Run No.	Final Train Accuracy (%)	Test Accuracy (%)
1	52	31
2	52	35
3	49	35
4	52	38
5	51	32
6	48	37
7	50	28
8	46	44
9	47	34
10	53	31
Average	49	35

Table 3. 10 fold cross validation results

The network was able to learn to generalize consistently, however judging from the difference in accuracies on the training set and testing set, it seems that the network may have overfitted slightly on the training set. To better understand the performances of the network, we plot an accuracy graph of the network during training and testing, shown in figure 4.

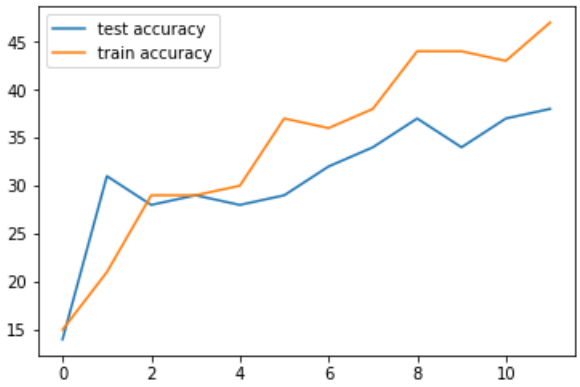

Fig. 4. Training vs Testing accuracy

In figure 4, the domain gap between training and testing was within reasonable range and was still able to improve its performances on the testing data despite much higher accuracies on the training data, thus we conclude that the network did not overfit during training.

In multi class classification problems, classifiers may become bias towards predicting one or few classes, thus limiting its performances on more generalized data sets. This is not well reflected by the final accuracies of the classifier as it may still achieve high overall accuracies from only recognizing limited number of classes. This is not a desirable behavior as we want the network to learn a generalized representation for all classes. Figure 5 shows the distribution of predicted classes, demonstrating that the network was able to make fair judgments for all classes and was not heavily bias towards any particular class.

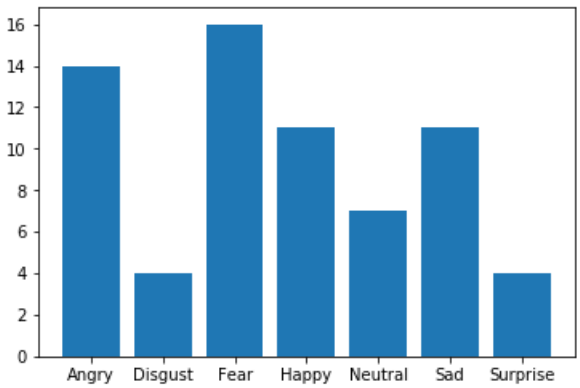

Fig. 5. Distribution of class predictions

To evaluate the feature extraction performances of the convolutional network, we compare it to another baseline neural network using features extracted by LPQs and PHOGs as described in chapter 2.2. The baseline classifier was able to achieve an average accuracy of 25% with evenly distributed predictions. Despite that the features used by the baseline classifier had an unfair advantage; the faces in the images were first localized by the Viola-Jones face detector [2], it still performed much worse than the CNN. In comparison to the baseline classifier, the CNN needed to locate the faces in the images as well as perform feature extraction on them, making the task significantly harder.

Run No.	Baseline Accuracy (%)	CNN Accuracy (%)
1	25	31
2	30	35
3	30	35
4	25	38
5	26	32
6	21	37
7	16	28
8	28	44
9	24	34
10	23	31
Average	25	35

Table 4. Baseline vs CNN Accuracy

From the results, CNN was able to extract useful features from noisy environments, performing significantly better than LPQ and PHOG. While the CNN made a huge improvement over the baseline classifier, it wasn't able to achieve results similar to the original paper [2], which used a combination of LPQ and PHOG with a support vector machine. This is due a multitude of reasons, including but not limited to insufficient data and noisy environments. The sensitive nature of neural networks may also contribute to the inferior performances, not being able to generalize as well as support vector machines even when good features are extracted and provided. Table 4 clearly supports the claim that CNN were able to provide better features than LPQs and PHOGs, but the fully connected feed forward layers weren't able to utilize them to their full potential due to the noisy environments; neural networks may simply not be the best choice for this type of data sets.

4 Conclusion and Future Work

Convolutional neural networks provided respectable results even in noisy environments, it was able to learn to extract useful features despite of an extremely small data set. The relatively low performances of the CNN in contrast to support vector machines in such environments is an interesting topic to investigate. Future works will be focused on improving the performances of a CNN under noisy environment with a limited amount of data with techniques such as imputing extra data and detecting/removing outliers in images.

References

1. Becherer, N., Pecarina, J., Nykl, S., Hopkinson, K.: Improving optimization of convolutional neural networks through parameter fine-tuning. Neural Computing and Applications pp. 1–11 (2017). https://doi.org/10.1007/s00521-017-3285-0, https://doi.org/10.1007/s00521-017-3285-0
2. Dhall, A., Goecke, R., Lucey, S., Gedeon, T.: Static Facial Expression Analysis in Tough Conditions : Data , Evaluation Protocol and Benchmark Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia. Database pp. 2106–2112 (2011). https://doi.org/10.1109/ICCVW.2011.6130508, http://staff.estem-uc.edu.au/roland/wp-content/uploads/file/roland/publications/Conference/ICCV/BeFIT2011/dhall_goecke_lucey_gedeon_BeFIT2011_StaticFacialExpressio
3. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors pp. 1–18 (2012), http://arxiv.org/abs/1207.0580
4. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift (2015), http://arxiv.org/abs/1502.03167
5. Karlik, B.: Performance analysis of various activation functions in generalized MLP architectures of neural networks. International Journal of Artificial Intelligence and Expert Systems **1**(4), 111–122 (2015)
6. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2 **2**(0), 1137–1143 (1995)
7. Krumhuber, E.G., Skora, L., Küster, D., Fou, L.: A Review of Dynamic Datasets for Facial Expression Research. Emotion Review **9**(3), 280–292 (2017). https://doi.org/10.1177/1754073916670022
8. Park, S., B, N.K.: Computer Vision – ACCV 2016 **10111**, 189–204 (2017). https://doi.org/10.1007/978-3-319-54181-5, http://link.springer.com/10.1007/978-3-319-54181-5
9. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition pp. 1–14 (2014), http://arxiv.org/abs/1409.1556
10. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of CNN activations pp. 1–12 (2015), http://arxiv.org/abs/1511.05879
11. Treadgold, N.K., Gedeon, T.D.: A Cascade Network Algorithm Employing Progressive RPROP Second Hidden Unit First Hidden Unit Output Unit Input Bias Artificial Neuron L1 Weights L2 Weights L3 Weights (1993) (1996)
12. Treadgold, N.K., Gedeon, T.D.: Exploring constructive cascade networks. IEEE Transactions on Neural Networks **10**(6), 1335–1350 (1999). https://doi.org/10.1109/72.809079
13. Wu, H., Gu, X.: Towards dropout training for convolutional neural networks. Neural Networks **71**, 1–10 (2015). https://doi.org/10.1016/j.neunet.2015.07.007
14. Xu, B., Wang, N., Chen, T., Li, M.: Empirical Evaluation of Rectified Activations in Convolutional Network (2015), http://arxiv.org/abs/1505.00853