A Comparison between Shallow and Deep Architecture Classifiers on Small Dataset

Kitsuchart Pasupa
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand
Email: kitsuchart@it.kmitl.ac.th

Wisuwat Sunhem
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand
Email: wisuwat.sun@gmail.com

***Abstract*—Many machine learning algorithms have been introduced to solve different types of problem. Recently, many of these algorithms have been applied to deep architecture model and showed very impressive performance. In general, deep architecture model suffers from over-fitting problem when there is a small number of training data. In this paper, we attempted to remedy this problem in deep architecture with regularization techniques including overlap pooling, flipped-image augmentation and dropout, and we also compared a deep structure model (convolutional neural network (CNN)) with shallow structure models (support vector machine and artificial neural network with one hidden layer) on a small dataset. It was statistically confirmed that the shallow models achieved better performance than the deep model that did not use a regularization technique. However, a deep model augmented with a regularization technique–CNN with dropout technique–was competitive to the shallow models.**

***Keywords*—machine learning; shallow learning; deep learning; small dataset.**

I. Introduction

Today, machine learning techniques are becoming widely used in real world applications [1]. They are now integrated in many modern applications, e.g., hand writing recognition, image understanding, and video suggestion. They create and improve models by learning from accumulated data. In the past, many machine learning techniques, e.g., Support Vector Machine (SVM), have shown their substantial performance and effectiveness in many real world applications [2]–[4]. However, these algorithms require good extracted hand-crafted features. Feature extraction is one of challenging tasks that needs expert knowledge to perform adequately. Features extracted from each data sample are fed into learning algorithms. We later refer to these algorithms as "shallow model" algorithms because they consist of very few layers of composition. These include Artificial Neural Network (ANN) with one hidden layer as well. In order to avoid complex feature extraction process, "deep model" was introduced. It aims to automatically learn feature hierarchy from low level to high level. It moves machine learning forward closer to the ultimate goal of artificial intelligence, that is, a machine that can think like a human does [5].

Although there are pieces of evidence that deep learning is able to achieve better performance than shallow learning [6], [7], it is well known that deep learning has limitations that should be considered. Examples of these limitations are the following:

i) High computational cost in training phase: it has higher computational complexity comparing to shallow learning and requires high performance computing unit to train– using GPU-based implementation can significantly reduce model training time comparing to using CPU-based implementation [8].
ii) Over-fitting problem when the data set is small: performance can drop dramatically due to over-fitting. Shallow learning techniques can overcome the problems of deep learning techniques when data is scarce. Many techniques have been proposed for solving over-fitting problem, such as overlap pooling, dropout and flipped-image augmentation. Another technique, transfer learning can be used when there is a small number of training sample. This can be done by training a model with related domain dataset that contains large number of samples. As a model converges, it fine-tunes all of the layers of the network by using an in-domain small sample data set. An example of usages of transfer learning technique can be found in a recent work by [9]. They attempted to use Convolutional Neural Network (CNN) with transfer learning technique for emotion recognition of human face on small datasets. However, it is not always possible to find a related domain dataset to train on first. CNN was also applied on ImageNet dataset [7]. This dataset consists of 1.2 million labeled images under 1000 different categories. Although the dataset is large, there are only roughly 1000 images in each of the 1000 categories. This is rather a small sample size compared to the number of categories. In this case, the authors applied a dropout technique to solve the over-fitting problem.

There has been a work that compared the performances of machine learning with shallow and deep architecture [10]. This paper concluded that combining them together gave the best result for text analysis task.

Our study investigated the same small face shape dataset that we used in one of our previous studies [11]. In that study, we showed that SVMs in conjunction with Radial Basis Function (RBF) outperformed other shallow machine learning

978-1-5090-4139-8/16/$31.00 ©2016 IEEE

algorithms on this dataset. In this study, we used a deep learning technique–CNN–and compared it with two well-known shallow model learning algorithms–SVM and ANN with one hidden layer. Because the dataset was very small, CNN model could be over-fitting, so we employed overlap pooling, flipped-image augmentation, and dropout regularization techniques to handle it.

The paper is organized as follows: Section II presents the methodology used in this paper; we explain our experimental framework and settings in Section III; The results are discussed and concluded in Section IV and V, respectively.

II. Methodology

Many machine learning algorithms have been introduced and re-introduced. They can be categorized into two types according to their model structure: shallow and deep.

A. Shallow model learning

Shallow model learning is a type of machine learning algorithms can generate good generalized predictive model with only a few layers of composition. It requires samples with well-studied discriminative features extracted by experts. It can perform well even though only a limited number of samples is available. This work focused on two well-known shallow machine learning algorithms: ANN with one hidden layer and SVM.

1) Artificial Neural Network (ANN): ANN was inspired by human brain operation. Its complexity can be increased by adding more hidden layers into the model and/or more neurons into each layer. ANN with one hidden layer is considered a shallow structure model while ANN with more than one hidden layers is a deep model. The model can be trained with different loss functions such as cross entropy error and classification error functions. A gradient decent algorithm is usually applied to minimize a selected loss function in order to obtain optimal variables.

2) Support Vector Machine (SVM): SVM is an instance-based learning that classifies data into two classes. The model selects and utilizes proper representative instances from a training set, so-called "support vector". SVM tries to generate a maximum-margin hyper-plane between support vectors of each class. Basically, it performs linear classification. In order to enable it to be a nonlinear classifier, a kernel function is applied. Kernel function maps data from its input space to a new space that SVM can perform nonlinear classification of data. SVM can also be extended to solve regression tasks.

Both models can perform nonlinear prediction. SVM usually performs better than ANN as shown in [12]–[14]. SVM performance can drop when applied to noisy data whereas ANN tolerates noise better and is more robust in some tasks [15], [16]. These algorithms are powerful machine learning techniques that can be applied to solve a complex problem; however, they require discriminative features extracted by human.

B. Deep model learning

The first deep learning technique so-called "Convolutional Neural Network" (CNN) was first introduced by Fukushima in 1980 [15]. It was inspired by real biological learning process. A complex combination of cells in an animal's visual cortex is called a receptive field. The field processes small sub-pictures of an image and combines them to recognize the context [17]. In deep learning, a region in an input image covered by a mask is a receptive field for a neuron in hidden layer. The mask is shifted by one or more pixel throughout the input image. A sub-image has similar features to the others if their output values are close to each other. The number of masks represents the number of neurons in the hidden layers. Hidden layers are structured hierarchically as layers in ANN. The weights of each mask can be adjusted to achieve an optimal performance. This process is called "feature learning" [18]. It performs automatic feature extraction prior to the classification phase. Feature learning is an important ability of deep learning algorithms that can learn to extract features from raw data without human aid. In this work, we also investigated CNN. CNN consists of three main layers which are (i) convolutional layer, (ii) pooling layer, and (iii) fully connected layer. Convolutional layer operates as a filter–receptive field–that can be tuned to gain an optimal model with good feature extraction. Pooling layer performs a data summarizing operation such as Mean, Max, and Min in order to reduce the spatial size of the representation in the network and control over-fitting of the model. The architecture of the last type of layer, fully connected layer, is similar to that of a traditional neural network that uses extracted features from the layers before it. With all of these components, the model can be trained with any kinds of raw data, especially, image data. It is known that deep learning algorithms require a high performance computing unit to generate a model and a large dataset to obtain a good model [19]. If there is a small number of training data, the model can be easily over-fitting. The following special techniques are needed to solve this problem–overlap pooling, dropout, data flipped-image augmentation [7].

i) Overlap pooling: In a standard pooling layer, the mask is shifted (stride) so that the next position does not overlap with the current one. The technique is applied to collect more input information with the size of the mask previously mentioned. Hence, it can solve the over-fitting problem. Moreover, in order to enhance the performance of the network, the mask can also be overlapped [20].

ii) Flipped-image augmentation: CNN is definitely over-fitting if the training set is small. To cope with this problem, a new set of images is created by flipping images in the dataset horizontally and augmenting them to the training set. Therefore, the number of samples is now twice that of the samples in the old training set. It also increases the diversity of training set.

iii) Dropout: this is a popular technique that handles over-fitting by randomly removes some neurons in the hidden layer of the fully-connected architecture in the training phase, but in the test phase, all neurons in the hidden layer are still used. A probability of retain unit, p, is required for adjustment to get a good model. [21] suggested that if n is an optimal number of nodes used in a standard ANN, a good dropout network should have $\frac{n}{p}$ neurons.

III. Experimental Framework

A. Face Shape Dataset

Images of women faces were collected from the Internet in order to build a hairstyle recommendation system [11]. They were labeled by volunteers who had passed a qualifying test. This dataset contained 500 samples of five face shape categories: heart-shape, oval-shape, oblong-shape, round-shape, and square-shape. Each class consisted of 100 samples. This dataset was later used in the developed hairstyle recommendation system for woman to analyze the face shape of a user and suggest proper hairstyles for her [22]. The system could also simulate the hairstyle she was wearing. Nineteen features were carefully extracted to obtain discriminative features for five different face shapes. It was found that SVM in conjunction with RBF was the best contender for this dataset.

B. Data pre-processing

In this work, we investigated three types of data representation.

1) Hand-crafted features: The 19 features proposed in [11] were applied to SVM and ANN.

2) Raw image: We simply utilized raw images shown in Fig. 1a. All images were resized to 48×48 pixels and converted to gray level scale because color domain is not necessary for face shape classification. This type of data representation were used in our deep model.

3) Reconstructed image: Raw images are very complex and composed of components that may not be necessary for performing classification and can lead to poor performance. Therefore, from raw images, we reconstructed new images that had only the necessary components for classification. We generated 61 feature points from the Active Appearance Model (AAM) and face color-based segmentation as described in [11]. All of these points were connected to represent the shape of a face as shown in Fig. 1b. The reconstructed images were used in the CNN.

C. Experimental Setting

We aimed to find a set of optimal parameters for each model. In order for us to be able to do that, we divided the data into three sets: 400 samples for training set, 50 samples for validation set, and 50 samples for test set. The set of optimal parameters were selected based on the prediction accuracy on the validation set. Once it was obtained, we again trained the model on the combined training and validation set. Here, we compared the prediction accuracy and area under the receiver operating characteristic (AUROC) on the test set between all of the contenders. The adjustable parameters of each algorithm were tuned as follows:

1) SVM: As SVM is a binary classifier, a one-vs-one scheme was used to enable SVM to perform multi-class classification task. RBF kernel was employed. The regularization and kernel parameter range was $\{10^{-5}, 10^{-4}, ..., 10^{4}, 10^{5}\}$.

2) ANN with one hidden layer: We varied the number of neurons in each layer from 1 to 30.

3) CNN: There was a large number of parameters to be considered for CNN that needs high computational resource. Hence, we investigated only CNN with one to five convolutional layers to get an optimal model. We considered the following two structures: (i) CNN with one convolutional layer: the 32-dimension deep convolutional layer used a mask of 5×5 size that shifted one pixel at a time. We utilized Max pooling with a 2×2 filter in the pooling layer. In this layer, the filter shifted 2 pixels at a time in order to reduce the number of variables which was the output volume. This is shown in Fig. 2. (ii) CNN with two to five convolutional layers: The structure of this one was similar to the previous structure but we inserted 1–4 more convolutional layers and one more pooling layer between the current pooling layer and the fully-connected layer. In the additional convolutional layer, its depth dimension was 64 and the size of the mask was 3×3 with 1-pixel step shift, set to preserve the context of the image. The size of the image in the additional convolutional layer was $m \times m$ where $m = 22 - 2 \times (n - 1)$ and n was the number of convolutional layers. Additional pooling layer settings were set to be the same as those of the previous one in order to reduce the number of parameters in the classification phase. This architecture is shown in Fig. 3. In both architectures, a hidden layer in a fully connected architecture contained 256 neurons with five outputs.

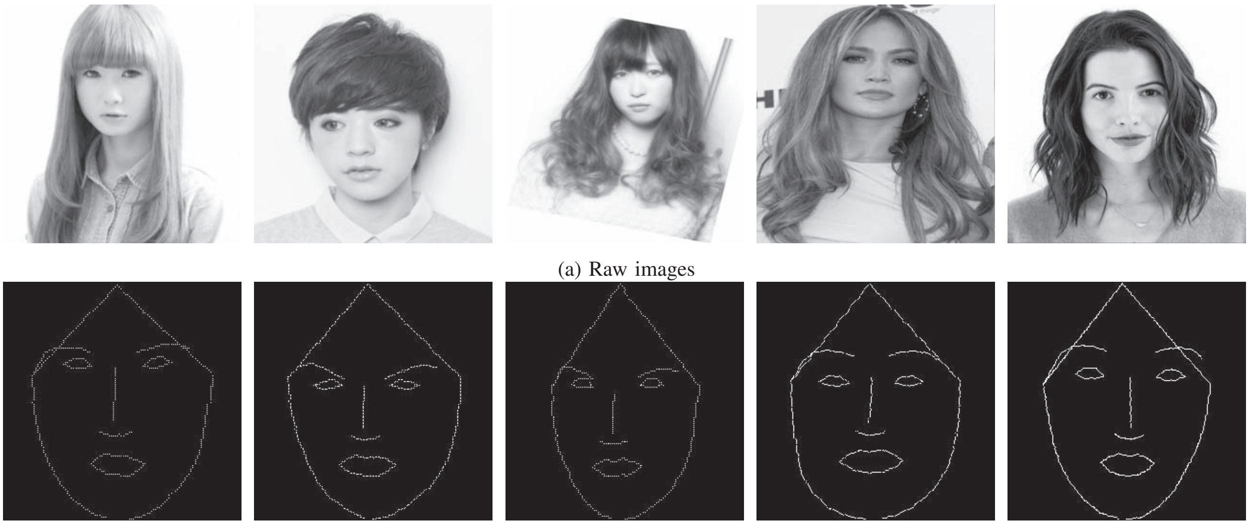
We applied all of the regularization techniques mentioned in the previous section one-by-one in this experiment as follows:

1) Overlap pooling method: the size of the mask in the first pooling layer was changed from 2×2 to 3×3 but the shift was still 2 pixels at a time, so the masks were overlapped.
2) Image augmentation technique: each image were horizontally flipped then augmented to the training set.
3) Dropout technique: We followed a good practice by [21] explained in the previous section. p range was $\{0.1, 0.2, \ldots, 1.0\}$ while n was set to 256. Therefore, the number of neurons in the hidden layer became $\frac{256}{p}$.

The experiment was run 20 times with different random seeds.

IV. Results and Discussion

Table I and II show the accuracy and AUROC of each method for 20 runs, respectively. These results were presented as violin plots that clearly show their distribution in Fig. 4a and 4b. We applied one-way analysis of variance (ANOVA) to analyze the differences between group means, and it was found that all of the means were different, confirming that there was a statistically significant difference between at least one pair of means at $p < 0.001$ for both accuracy and AUROC. Then, multiple comparisons were performed. The p-values are shown in Table III. It should be noted that we only discussed the accuracy because, overall, AUROC was very much the same



(a) Raw images

(b) Reconstructed images

Fig. 1: Examples of images trained by CNN. There were five-classes of face shapes: oval, round, oblong, square and heart (from left to right).

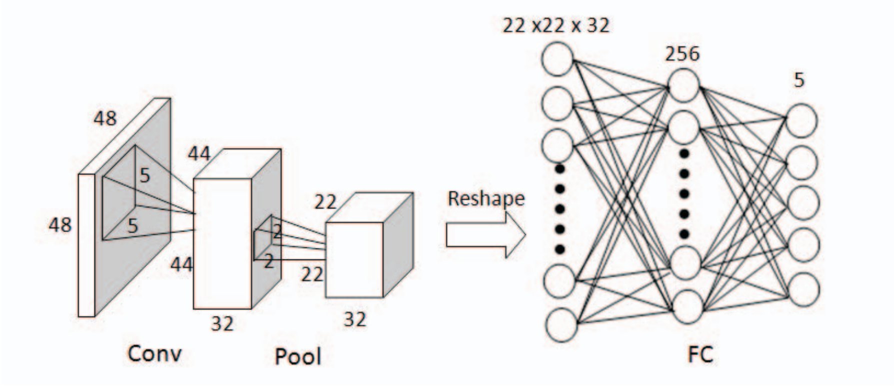


Fig. 2: CNN architecture with one convolutional layer.

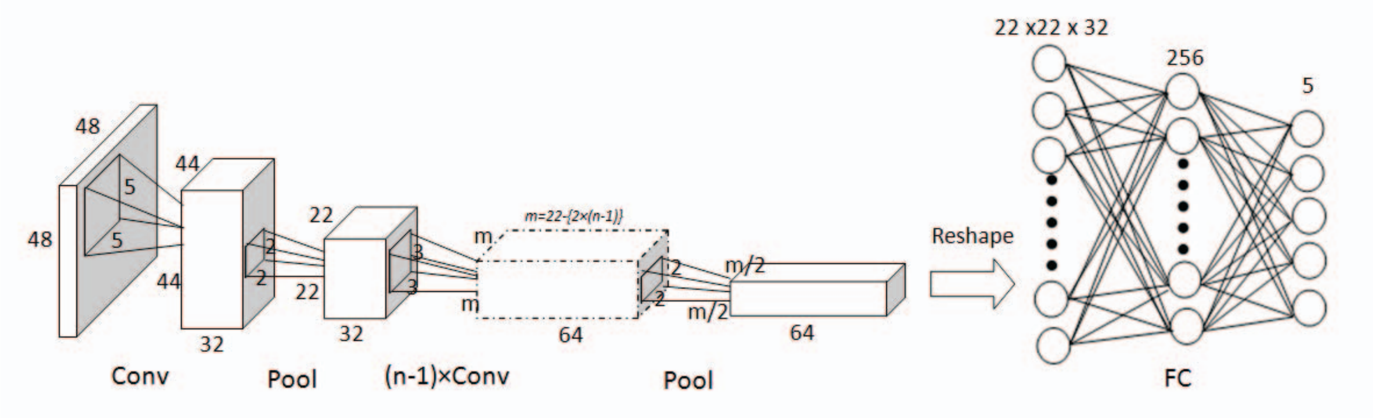


Fig. 3: CNN architecture with n convolutional layers (2–5).

as accuracy. According to the results, ANN achieved a better accuracy than SVM did at 60.2% and 58.7%, respectively, as shown in Table I, but the conclusion was statistically inconclusive ($p = 0.9863$), as shown in Table III.

We compared two types of data representation–raw and reconstructed images described in Section III-B. It is clear that using reconstructed images in CNN gave significantly better performance in accuracy than using raw images, at 53.00%, and 30.20% respectively ($p < 0.001$). Therefore, we only used reconstructed images in CNN with different types of

TABLE I: The classification accuracy of each algorithm on the test set.

Run	Shallow Model		Deep Model				
	19 Features		Raw	Reconstructed Image			
	SVM	ANN	CNN	CNN			
	RBF	1-hidden	Standard	Standard	Overlap	Aug	Dropout
1	56.00	54.00	18.00	54.00	54.00	50.00	56.00
2	60.00	60.00	26.00	46.00	52.00	54.00	52.00
3	56.00	60.00	34.00	44.00	56.00	46.00	50.00
4	62.00	56.00	32.00	57.00	57.00	56.00	66.00
5	58.00	58.00	28.00	54.00	52.00	56.00	57.00
6	62.00	60.00	36.00	44.00	44.00	44.00	50.00
7	68.00	62.00	28.00	60.00	57.00	60.00	70.00
8	56.00	60.00	22.00	56.00	60.00	46.00	46.00
9	64.00	68.00	30.00	52.00	60.00	54.00	70.00
10	58.00	58.00	38.00	52.00	50.00	57.00	57.00
11	66.00	60.00	30.00	57.00	60.00	62.00	60.00
12	38.00	62.00	24.00	48.00	46.00	52.00	50.00
13	54.00	60.00	34.00	62.00	57.00	60.00	54.00
14	56.00	58.00	34.00	50.00	54.00	50.00	56.00
15	48.00	62.00	24.00	54.00	64.00	52.00	64.00
16	66.00	68.00	42.00	56.00	66.00	66.00	70.00
17	54.00	60.00	30.00	56.00	62.00	57.00	64.00
18	62.00	64.00	34.00	52.00	56.00	64.00	66.00
19	70.00	58.00	30.00	56.00	62.00	57.00	57.00
20	60.00	56.00	30.00	50.00	46.00	52.00	48.00
Mean	58.70	60.20	30.20	53.00	55.75	54.75	58.15
STD	7.26	3.55	5.69	4.91	6.09	5.96	7.73

TABLE II: The area under the ROC curve of each algorithm on the test set.

Run	Shallow Model		Deep Model				
	19 Features		Raw	Reconstructed Image			
	SVM	ANN	CNN	CNN			
	RBF	1-hidden	Standard	Standard	Overlap	Aug	Dropout
1	75.00	72.50	56.25	68.75	66.25	70.00	67.50
2	72.50	71.25	48.75	71.25	71.25	68.75	72.50
3	75.00	75.00	53.75	66.25	70.00	71.25	70.00
4	72.50	75.00	58.75	65.00	72.50	66.25	68.75
5	76.25	72.50	57.50	73.75	73.75	72.50	78.75
6	73.75	73.75	55.00	71.25	70.00	72.50	73.75
7	76.25	75.00	60.00	65.00	65.00	65.00	68.75
8	80.00	76.25	55.00	75.00	73.75	75.00	81.25
9	72.50	75.00	51.25	72.50	75.00	66.25	66.25
10	77.50	80.00	56.25	70.00	75.00	71.25	81.25
11	73.75	73.75	61.25	70.00	68.75	73.75	73.75
12	78.75	75.00	56.25	73.75	75.00	76.25	75.00
13	61.25	76.25	52.50	67.50	66.25	70.00	68.75
14	71.25	75.00	58.75	76.25	73.75	75.00	71.25
15	72.50	73.75	58.75	68.75	71.25	68.75	72.50
16	67.50	76.25	52.50	71.25	77.50	70.00	77.50
17	78.75	80.00	63.75	72.50	78.75	78.75	81.25
18	71.25	75.00	56.25	72.50	76.25	73.75	77.50
19	76.25	77.50	58.75	70.00	72.50	77.50	78.75
20	81.25	73.75	56.25	72.50	76.25	73.75	73.75
Mean	74.19	75.13	56.38	70.69	72.44	71.81	73.94
STD	4.54	2.22	3.56	3.13	3.84	3.77	4.82

regularization and evaluated them.

The accuracy of CNN using overlap pooling, flipped-image augmentation, and dropout were 55.75%, 54.75%, and 58.15%, respectively. These results were clearly better than those obtained by conventional CNN on the average. Unfortunately, it was ambiguous to conclude that overlap pooling and flipped-image augmentation techniques were able to enhance the performance of CNN in this case, as the comparison resulted in high 0.7784 and 0.9698 p-value, respectively. Fortunately, dropout was able to improve the performance of CNN at $p = 0.0976$.

It was clear that the shallow models were able to achieve better performance than the deep models on the average. However, they were significantly better than only the deep model that did not use any regularization techniques ($p < 0.05$).

TABLE III: p-values from multiple comparisons of accuracy & AUROC.

Algorithm 1	Algorithm 2	p-value	
		Accuracy	AUROC
SVM-RBF	ANN	0.9863	0.9866
SVM-RBF	CNN-Raw	< 0.001	< 0.001
SVM-RBF	CNN-Rec	< 0.050	0.0532
SVM-RBF	CNN-Rec-Overlap	0.7156	0.7667
SVM-RBF	CNN-Rec-Aug	0.3691	0.4237
SVM-RBF	CNN-Rec-Dropout	1.0000	1.0000
ANN	CNN-Raw	< 0.001	< 0.001
ANN	CNN-Rec	< 0.050	< 0.050
ANN	CNN-Rec-Overlap	0.2272	0.2703
ANN	CNN-Rec-Aug	0.0642	0.0819
ANN	CNN-Rec-Dropout	0.9353	0.9557
CNN-Raw	CNN-Rec	< 0.001	< 0.001
CNN-Raw	CNN-Rec-Overlap	< 0.001	< 0.001
CNN-Raw	CNN-Rec-Aug	< 0.001	< 0.001
CNN-Raw	CNN-Rec-Dropout	< 0.001	< 0.001
CNN-Rec	CNN-Rec-Overlap	0.7784	0.7667
CNN-Rec	CNN-Rec-Aug	0.9698	0.9660
CNN-Rec	CNN-Rec-Dropout	0.0976	0.0939
CNN-Rec-Overlap	CNN-Rec-Aug	0.9985	0.9985
CNN-Rec-Overlap	CNN-Rec-Dropout	0.8704	0.8726
CNN-Rec-Aug	CNN-Rec-Dropout	0.5589	0.5639

V. Conclusion

We statistically compared deep learning with shallow learning technique on a small dataset–a face shape dataset. We showed that SVM and ANN in conjunction with hand-crafted discriminative features were able to achieve better performances than deep learning techniques that used either raw or reconstructed images but no regularization technique. Moreover, it is ambiguous to say that a shallow model gave a higher accuracy than a deep model that used regularization techniques did, especially the case of CNN using dropout. Thus, we can conclude that using the dropout technique can handle overfitting in small dataset. We were able to build a comparable model of deep learning on a small dataset to a shallow model by using image transformation and regularization techniques.

Acknowledgement

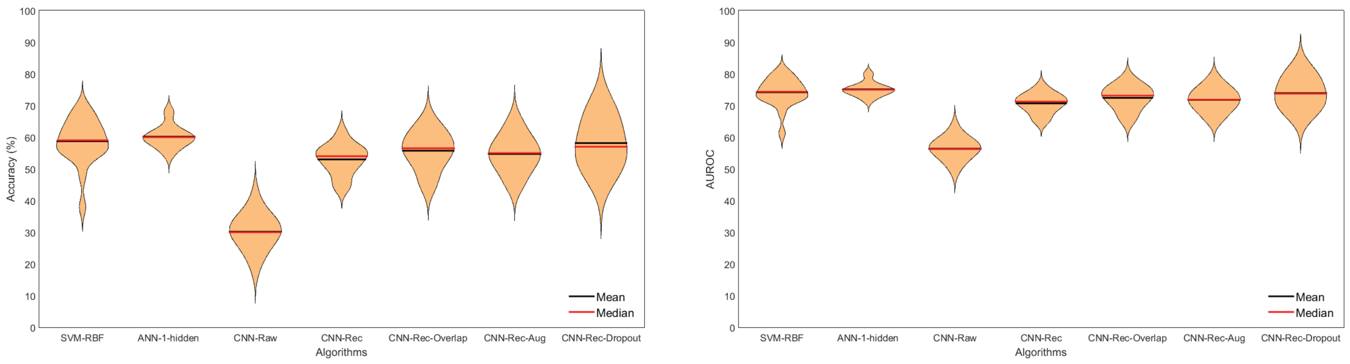
We would like to thank the Information Security Advisory Group (ISAG) and Digital Image Processing Laboratory (DIP) at the Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang for giving us access to their computing resources. This work was supported by the Thailand Research Fund under grant agreement number TRG5680090.

References

[1] R. F. Harrison and K. Pasupa, "Sparse multinomial kernel discriminant analysis (sMKDA)," *Pattern Recognition*, vol. 42, no. 9, pp. 1795–1802, 2009.

[2] D. Inman, "Machine learning applied to recognising hand-written Japanese," in *Proceeding of 4th IEEE International Workshop on Robot and Human Communication (RO-MAN 1995)*, 1995, pp. 117–122.

[3] C. Agarwal and A. Sharma, "Image understanding using decision tree based machine learning," in *Proceeding of International Conference on Information Technology and Multimedia (ICIM 2011)*, 2011, pp. 1–8.



(a) Accuracy

(b) AUROC

Fig. 4: Violin plot of the performances shown by all of the methods.

[4] L. M. López-López, J. J. Castro-Schez, D. Vallejo-Fernandez, and J. Albusac, "A recommender system based on a machine learning algorithm for B2C portals," in *Proceeding of IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2009)*, 2009, pp. 524–531.

[5] S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards, *Artificial intelligence: a modern approach*. Prentice Hall Upper Saddle River, 2003, vol. 2.

[6] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceeding of Advances in Neural Information Processing Systems (NIPS 2012)*, 2012, pp. 1097–1105.

[8] NVIDIA. (2015) GPU-based deep learning inference: A performance and power analysis. [Online]. Available: https://www.nvidia.com/content/tegra/embedded-systems/pdf/jetson_tx1_whitepaper.pdf

[9] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proceeding of ACM International Conference on Multimodal Interaction (ICMI 2015)*, 2015, pp. 443–449.

[10] J. Wagner, J. Foster, and J. van Genabith, "A comparative evaluation of deep and shallow approaches to the automatic detection of common grammatical errors," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 112–121.

[11] W. Sunhem and K. Pasupa, "An approach to face shape classification for hairstyle recommendation," in *Proceeding of 8th International Conference on Advanced Computational Intelligence (ICACI 2016)*, 2016, pp. 390–394.

[12] N. Naseer, K.-S. Hong, M. J. Khan, and M. R. Bhutta, "Comparison of artificial neural network and support vector machine classifications for fNIRS-based BCI," in *Proceeding of 15th International Conference on Control, Automation and Systems (ICCAS 2015)*, 2015, pp. 1817–1821.

[13] K. Kianmehr, S. Gao, J. Attari, M. M. Rahman, K. Akomeah, R. Alhajj, J. Rokne, and K. Barker, "Text summarization techniques: SVM versus neural networks," in *Proceedings of 11th International Conference on Information Integration and Web-based Applications & Services (iiWAS 2009)*, 2009, pp. 487–491.

[14] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of 23rd International Conference on Machine Learning (ICML 2006)*, 2006, pp. 161–168.

[15] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[16] S. Anwar and R. Ismal, "Robustness analysis of artificial neural networks and support vector machine in making prediction," in *Proceeding of IEEE 9th International Symposium on Parallel and Distributed Processing with Applications (ISPA 2011)*, 2011, pp. 256–261.

[17] T. D. Team. Convolutional neural networks (lenet). [Online]. Available: http://deeplearning.net/tutorial/lenet.html

[18] T. Dettmers. (2015) Deep learning in a nutshell: Core concepts. [Online]. Available: https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/

[19] S. Raschka. (2016) When does deep learning work better than SVMs or random forests. [Online]. Available: http://www.kdnuggets.com/2016/04/deep-learning-vs-svm-random-forest.html

[20] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Proceeding of International Conference on Artificial Neural Networks (ICANN 2010)*, 2010, pp. 92–101.

[21] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] W. Sunhem, K. Pasupa, and P. Jansiripitikul, "Hairstyle recommendation for women," in *Proceeding of 5th ICT International Student Project Conference (ICT-ISPC 2016)*, 2016.