Medical Image Classification using Deep Learning

Department of Computer Science and Engineering,

Khulna University of Engineering & Technology

Report written by

Tanmoy Tapos Datta, Roll:1307006

Arunima Mandal, Roll:1307018

Supervised by

Prof. Dr. Pintu Chandra Shill

Signature ____________________

17 Feb, 2018

Acknowledgement

All the praise to the almighty God, whose blessing helped me to successfully complete this thesis work. We show significant and indescribable gratefulness to our supervisor Dr. Pintu Chandra Shill, Professor, Department of Computer Science and Engineering, Khulna University of Engineering & Technology for his outstanding helpful contribution in giving suggestion and encouragement. We acknowledge his constant co-operation and proper guidance throughout the development process. He has been a great source of effective and feasible ideas, profound knowledge and all time feedback for us.
We are also too grateful to Md. Sakib Galib Sourav, Assistant Professor, Department of Electrical Electronic Engineering and Mr. Sudipta Singha Roy , Lecturer, Institute of Information and Communication Technology (IICT), Khulna University of Engineering & Technology for their incredible cordial help.
We thank all the teachers of the Department of Computer Science and Engineering who helped us providing guidelines to perform the work. We would also like to thank our friends and family for their cordial support.

Abstract

In our work, we used a very simple deep learning network for image classification that is based on very basic data processing components: 1) cascaded principal component analysis (PCA); 2) binary hashing; and 3) blockwise histograms. In the proposed architecture, the PCA is employed to learn multistage filter banks. This is followed by simple binary hashing and block histograms for indexing and pooling. This architecture is thus called the PCA network (PCANet) and can be extremely easily and efficiently designed and learned.
Several have extensively tested this on many benchmark visual data sets for different tasks, including Labeled Faces in the Wild (LFW) for face verification; the MultiPIE, Extended Yale B, AR, Facial Recognition Technology (FERET) data sets for face recognition; and MNIST for hand-written digit recognition.Surprisingly, for all tasks, such a seemingly naive PCANet model is on par with the state-of-the-art features either prefixed, highly hand-crafted, or carefully learned [by deep neural networks (DNNs)]. Even more surprisingly, the model sets new records for many classification tasks on the Extended Yale B, AR, and FERET data sets and on MNIST variations.We have used this method in a medical image dataset, particularly on Breast histology dataset detecting whether it is invasive ductal carcinoma (IDC) or non-invasive ductal carcinoma (NIDC).

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Deep learning (also known as deep structured learning or hierarchical learning) is the part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

Deep learning models are related to information processing and communication patterns in a biological nervous system, such as neural coding that attempts to define a relationship between various stimuli and associated neuronal responses in the brain.

Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics and drug design,where they have produced results comparable to and in some cases superior to human experts.

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more), digital image processing may be modeled in the form of multidimensional systems.

1.2 Problem statement

Image classification based on visual content is a very challenging task, largely because there is usually a large amount of intra-class variability, arising from different lighting conditions, misalignment, non-rigid deformations, occlusion and corruptions. Numerous efforts have been made to counter the intra-class variability by manually designing low-level features for classification tasks. Representative examples are Gabor features and local binary patterns (LBP) for texture and face classification and Scale Invariant Feature Transform (SIFT) and the Histogram of Oriented Gradients (HOG) features for object recognition. Although the low-level features can be hand crafted with great success for certain data and tasks, designing effective features for new data and tasks usually requires new domain knowledge because most hand-crafted features cannot simply be adapted to new conditions [1], [2].

1.2.1 Image classification using Deep Learning

Learning features from data of interest is considered as a plausible method of remedying the limitations of hand-crafted features. An example of such methods is learning through deep neural networks (DNNs), which has recently garnered significant attention [1]. The idea of deep learning is to discover multiple levels of representation with the hope that higher level features can represent more abstract semantics of the data. Such abstract representations learned from a deep network are expected to provide greater robustness to intra-class variability. One key ingredient to the success of deep learning in image classification is the use of convolutional architectures [3]–[10]. A convolutional deep neural network (ConvNet) architecture [3]–[5], [8], [9] consists of multiple trainable stages stacked on top of each other followed by a supervised classifier. Each stage generally consists of "three layers" – a convolutional filter bank layer, a nonlinear processing layer, and a feature pooling layer. To learn a filter bank in each stage of ConvNet, a variety of techniques, such as restricted Boltzmann machines (RBM) [7] and regularized auto-encoders and their variations, has been proposed; see [2] for a review and references therein. In general, such a network is typically learned using a stochastic gradient descent (SGD) method. However, learning a network that is useful for classification critically depends on expertise in parameter tuning and various ad hoc tricks.

1.3 Objectives

Our objectives is to use PCANet and SVM for the purpose of classification of medical image dataset so that we can obtain a better accuracy result than any other conventional methods.

1.4 Motivation

An initial motivation of our study is the desire to resolve certain apparent discrepancies between ConvNet and ScatNet. We want to achieve two simple goals: First, we want to design a simple deep learning network that should be very easy, even trivial, to train and to adapt to different data and tasks. Second, such a basic network could serve as a good baseline for people to empirically justify the use of more advanced processing components or more sophisticated architectures for their deep learning networks.

The solution comes as no surprise: We use the most basic and easy operations to emulate the processing layers in a typical (convolutional) neural network mentioned above: The data-adapting convolution filter bank in each stage is chosen to be the most basic PCA filters; the nonlinear layer is set to be the simplest binary quantization (hashing); and for the feature pooling layer, we simply use the block-wise histograms of the binary codes, which are considered as the final output features of the network. For ease of reference, we call this data-processing network a PCA Network (PCANet).

At least one characteristic of the PCANet model seems to challenge common wisdom regarding building a deep learning network such as ConvNet [4], [5], [8] and ScatNet [6], [10]: no nonlinear operations in the early stages of the PCANet until the very last output layer, where binary hashing and histograms are utilized, to compute the output features. Nevertheless, as we will see through extensive experiments, such a drastic simplification does not appear to undermine the performance of the network on various typical datasets.

1.5 Contributions

Although our initial intention of studying the simple PCANet architecture is to obtain a simple baseline for comparing and justifying other, more advanced deep learning components or architectures, our findings lead to various pleasant but thought-provoking surprises: The very basic PCANet, in a fair experimental comparison, is already quite as good as, and often better than, state-of-the-art features (prefixed, hand crafted, or learned from DNNs) for almost all image classification tasks, including face images, hand-written digits, texture images, and object images. More specifically, for face recognition with one gallery image per person, the model achieves a 99.58% accuracy on the Extended Yale B dataset and a greater than 95% accuracy across disguise/illumination subsets in the AR dataset. On the FERET dataset, the model obtains a state-of-the-art average accuracy of 97.25% and achieves its best accuracy of 95.84 and 94.02% on the Dup-1 and Dup-2 subsets, respectively.

Chapter 2

Literature Review

2.1 Background

Although many variations of deep convolutional networks have been proposed for different vision tasks and their success is usually empirically justified, arguably the first instance that has led to a clear mathematical justification is the wavelet scattering networks (ScatNet) [6], [10]. The only difference in that case is that the convolutional filters in ScatNet are prefixed – they are simply wavelet operators; hence, no learning is required. Somewhat surprisingly, such a prefixed filter bank, once utilized in the similar multistage architectures of ConvNet or DNNs, has demonstrated superior performance over ConvNet and DNNs in several challenging vision tasks such as hand-written digits and texture recognition [6], [10].

However, as we will observe in our following work, such a prefixed architecture does not generalize very well to tasks such as face recognition or image classification, where the intra-class variability includes significant illumination changes and corruption.

2.2 Principal Component Analysis(PCA)

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of distinct principal components is equal to the smaller of the number of original variables or the number of observations minus one. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

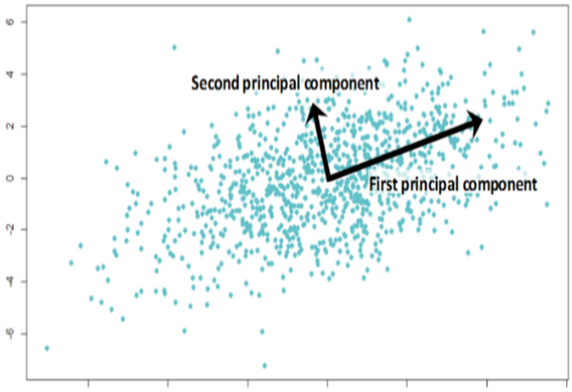

Figure 2.1: PCA

2.3 PCANet

Principal component analysis network (PCANet) is a novel deep learning algorithm for feature learning with the simple network architecture and parameters. In this architecture, the PCA is employed to learn multistage filter banks. This is followed by simple binary hashing and block histograms for indexing and pooling. This architecture is thus called the PCA network or PCANet.

2.4 Support Vector Machine(SVM)

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support vector clustering algorithm created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.

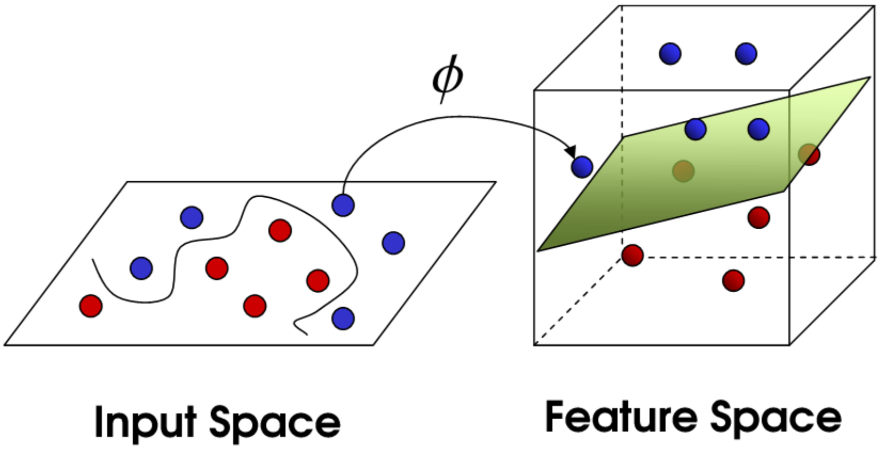

Figure 2.2: SVM

2.5 Difference between PCANet vs regular neural network

The term "regular" neural network arise since a large variety of techniques have been studied under the connectionist umbrella. However, three obvious major differences from the most fashionable deep Neural Network techniques these days are:

- PCANet does not use back propagation for training.
- PCANet is trained by stacking. The layers are not trained jointly.
- The layers are trained unsupervised. The filters are just the Principle Components obtained by PCA on the training examples or (N-1)th layer activation maps of training examples while stacking.

2.6 Related works on Image classification

Lots of work has been done using deep learning and image processing for image classification. We have studied that people tested these basic networks on many benchmark visual data sets for different tasks, including Labeled Faces in the Wild (LFW) for face verification, the MultiPIE, Extended Yale B, AR, Facial Recognition Technology (FERET) data sets for face recognition, and MNIST for hand-written digit recognition.

The very basic PCANet, in a fair experimental comparison, is already quite on par with, and often better than, state-of-the-art features (prefixed, hand crafted, or learned from DNNs) for almost all image classification tasks, including face images, hand-written digits, texture images, and object images. More specifically, for face recognition with one gallery image per person, the model achieves a 99.58% accuracy on the Extended Yale B dataset and a greater than 95% accuracy across disguise/illumination subsets in the AR dataset. On the FERET dataset, the model obtains a state-of-the-art average accuracy of 97.25% and achieves its best accuracy of 95.84 and 94.02% on the Dup-1 and Dup-2 subsets, respectively.

2.7 Implemented method

In medical science PCANet has not been used widely so far. In this paper, we propose a medthod to use PCANet for image classification that is based on very basic data processing components:

- Principal component analysis (PCA)
- Binary hashing
- Blockwise histograms

Chapter 3

Theoretical Consideration

In recent years, representation learning, which automatically learns useful information and discovers appropriate representation directly from data instead of handcrafted features, has gained its good reputation for histopathological images. As a typical representation learning method sparse representation (SR) and its variants have shown their effectiveness for histopathological images. More recently, deep learning (DL) has also been applied to histopathological images. DL is almost the most successful representation learning method now. DL is suitable to learn feature representation for histopathological images, as it can directly learn effective image representation from pixel (or low) level features to discover high level shape and edge interactions. In the last couple of years, various DL algorithms, such as convolutional neural networks (CNN), autoencoder and its variants, have been applied to histopathological images for classification and detection, and achieved the state-of-the-art performance.

3.1 Why PCANet?

Despite the wide applications of various DL algorithms, they usually suffer from the problem of parameter tuning, which is not only time-costing, but also in need of much expertise knowledge. The principal component analysis network (PCANet) is a novel framework for unsupervised DL, which consists of only three simple basic components: cascaded PCA as a deep network, binary hashing as a nonlinear layer, and block-wise histograms for feature pooling layer. The network architecture of PCANet is very simple with few parameters. Moreover, PCANet indeed achieves competitive and even better performance for image classification compared with other state-of-the-art DL algorithms. Several improved PCANet algorithms are then proposed, such as DLANet, SRDANet, and SPCANet. The successful application of PCANet for image representation makes it feasible to histopathological images.

The variants of PCANet mainly focus on applying more effective filters instead of PCA in the PCANet framework to improve representation performance. In fact, the way of binary hashing in PCANet also affects the representation performance, which has not been deeply investigated yet. In current PCANet and its variants, the binary hashing approach just simply encodes the quantized binary values according to the sequence of principal components (PC), that is to say, the first PC is assigned to the most significant bit, while the last PC is corresponding to the least significant bit. It is worth noting that the current binary hashing operator is one of the n 2 binary codes for an n-bit vector, therefore, only limited information is obtained, which affects classification performance. Random theory based methods, such as the random subspace and random projection algorithms, have been widely used in machine learning.

Chapter 4

Methodology

4.1 PCANet Algorithm

The PCANet algorithm is cascades two filterbank convolutions with an intermediate mean normalization step, followed by a binary hashing step and a final histogramming step. Training involves estimating the filterbanks used for the convolutions, and estimating the classifier to be used on top of the ultimate histogram-derived features.

4.1.1 Filterbank Convolutions

The filterbanks are estimated by performing principal components analysis (PCA) over patches. We extract all of the 7×7 from all of the images and vectorize them so that each patch is a flat 49-entry vector:

$$v \;\in\; R^{7\times 7} \rightarrow vec\; v \;\in\; R^{49}\; where\; visanimage.$$

For each patch vector we take the mean of the entries (the DC-component) and then subtract that mean from each entry of the vector so that all of our patches are now zero mean. We perform PCA over these zero-mean patch vectors and retain the top eight components $W \in\; R^{49}$. Each principle component (a column of W) is a filter and may be converted into a 7×7 kernel which is convolved with the input images. The input images are zero-padded for the convolution so that the output has the same dimension as the image itself. So, using the eight columns of W we take each input image and convert it into eight output images, where

$$1 \leq l \leq 8$$

4.1.2 Second Layer

The second layer is constructed by iterating the algorithm from the first layer over each of the eight output images. For each output image we take the dense set of flattened patch vectors, remove the DC-component. The patches produced by the different filters are then concatenated together and we estimate another PCA filterbank (again with eight filters). Each filter $w_{2,k}$ from the layer-2 filterbank is convolved to produce a new image. Repeating this process for each filter in the filterbanks produces 64=8 $\times$ 8

4.1.3 Hashing and Histogramming

The 64 images have the same size as the original image thus we may view the filter outputs as producing a three-dimensional array $J \in \; R^{H \times W \times 64}$ where $H \times W$are the dimensions of the input image. Each of the 64 images is produced from a layer one filter l1 and a layer two filter l2 so we denote the associated image as. Each pixel (x,y) from the image has an associated 8-dimensional feature vector. These feature vectors are converted into integers by using a Heaviside step function H sum: Kl1(x,y)=

$$\sum_{z=1}^{8} 2^{z1}.h$$

in the layer one filterbank so this means that we have eight images after the hashing operation and the images are all integers. Histogramming

We then take 77 blocks of the hashed images K and compute a histogram with bins over the values observed. These blocks can be disjoint (used for face recognition) or they can be overlapping (useful for digit recognition). The histograms formed from these blocks and from the several images are all concatenated into a feature vector. Classification is then performed using this feature vector.

4.2 PCANet Diagram

PCANet is based on three basic data processing components like cascaded principal com-ponent analysis (PCA), binary hashing and blockwise histograms. Fllowing diagram explain this steps:

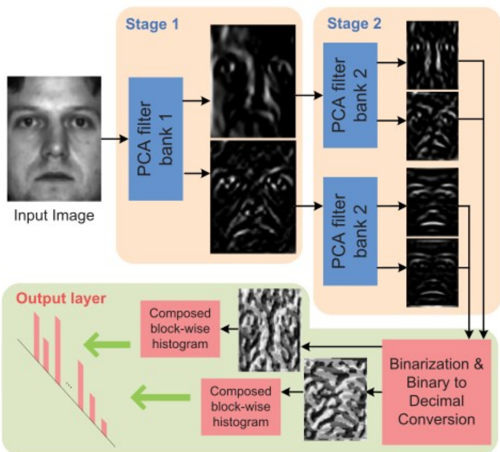

Figure 4.1: Illustration of how PCANet extracts features from an image through the three simplest processing components: PCA filters, binary hashing, and histograms.

In our work we used a two stage PCANet model. The detailed block diagram of this model is given below:

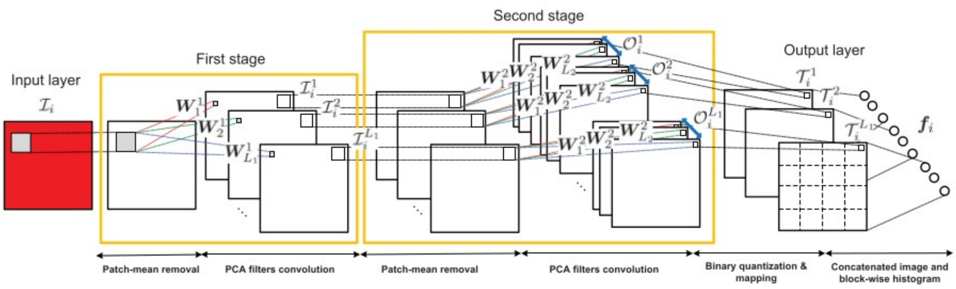

Figure 4.2: A detailed block diagram of the two-stage PCANet

4.3 Workflow

The workflow of our work is described below:

1. Generate image list using python script.
2. Prepr train and test dataset using opencv.
3. Train PCANet model with train dataset.
4. Train Support Vector Machine(SVM) with PCANet output image obtained from train dataset.
5. Get output of PCANet model for each test data in test dataset.
6. Acquire prediction result from Support Vector Machine(SVM) model using every image of test dataset obtained from previous step.
7. Calculate accuray and analysis result.

Chapter 5

Implementation

5.1 Datasets

In our work we used a dataset consists of breast cancer histology images of size 50 x 50, curated from Andrew Janowczyk website and used for a data science tutorial at Epidemium. The goal is to classify cancerous images (IDC : invasive ductal carcinoma) vs non-IDC images. An pictorial view of the dataset is given below:

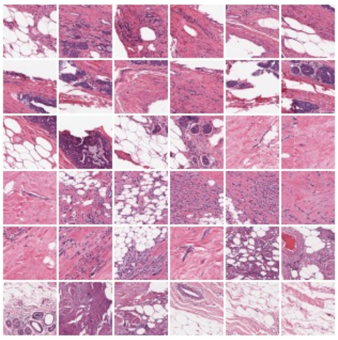

Figure 5.1: Class 0 (non-invasive ductal carcinoma) image dataset.

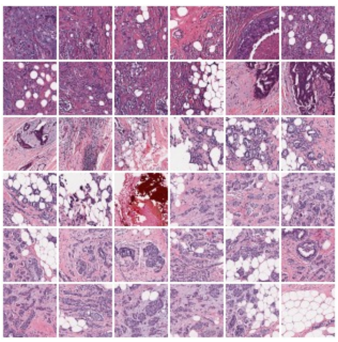

Figure 5.2: Class 1 (invasive ductal carcinoma) image dataset.

5.2 Environment

We used the following environment for our work:

- Pycharm: For listing dataset names.

- QT and openCV: For generating dataset from the images.
- Matlab: For Implementing PCANet and SVM.

5.3 Working Procedure

5.3.1 Training PCANet model with train dataset

At first we trained PCANet model with train dataset.

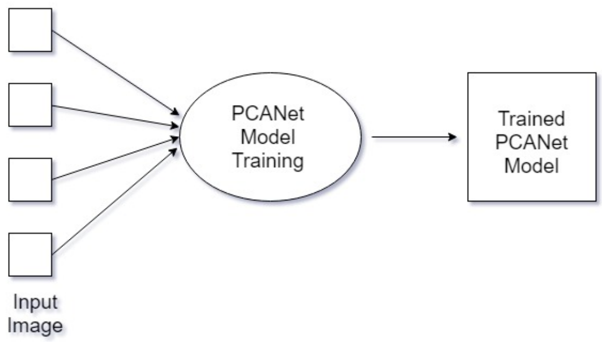

Figure 5.3: Training PCANet model

5.3.2 Taking output for train dataset from PCANet trained model

In this step we took the PCANet output corresponding to train dataset.

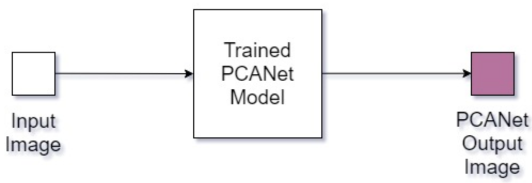

Figure 5.4: Taking output for train dataset from PCANet trained model

5.3.3 Training SVM for Classification purpose

We trained SVM with the output image of PCANet trained model for classification purpose.

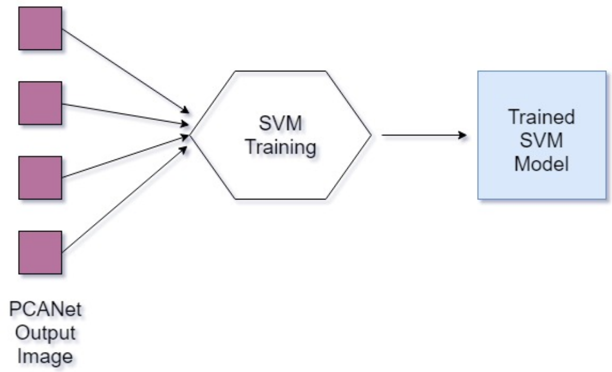

Figure 5.5: Training SVM for Classification purpose

5.3.4 Predicting results from SVM trained model

In this step at first we get the output from PCANet trained model for every test dataset. Then we predict the outcome of the image got from PCANet model.

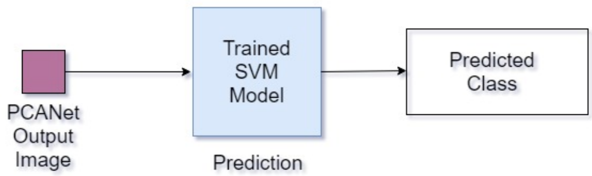

Figure 5.6: Predicting results from SVM trained model

So the overall prediction step will be like this:

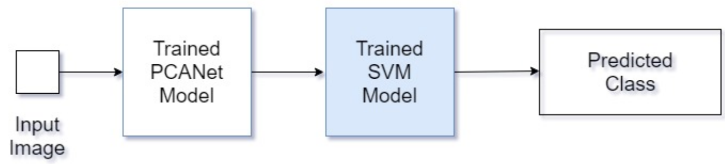

Figure 5.7: Prediction step

Chapter 6

Simulated Result

We have tried enough to get better accuarte result. We have obained the following result with respect to the described dataset int he Implementation section.

We have used 12,000 random images to train PCANet and SVM. 4,474 images was checked as test data corresponding to the train dataset. We get about 82% accuracy.

The accuracy curve over all the test dataset is given below:

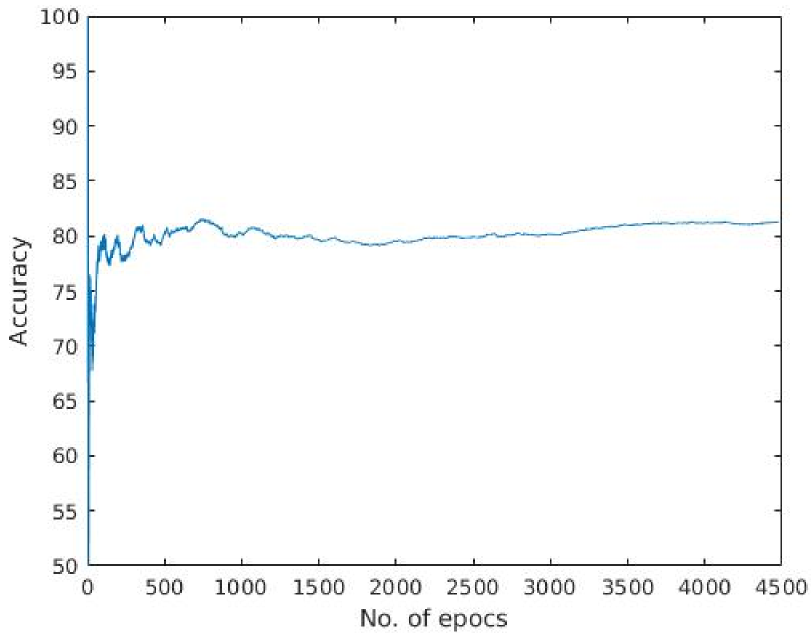

Figure 6.1: Accuracy Curve

And the confusion matrix for the obtained result is:

Prediction outcome

		p	**n**	**total**
actual value	**p′**	True P 1667	False N 736	P′
	n′	False P 540	True N 1531	N′
	total	P	N	

We have simulated the result for small dataset because of computation limitation. If it can be simulated for large dataset, then the accuracy result will be better.

Chapter 7

Conclusions and Recommendation

7.1 Summary

In our work, we used arguably the simplest unsupervised convolutional deep learning network— PCANet. The network processes input images using cascaded PCA, binary hashing, and block histograms. Like most ConvNet models, the network parameters, such as the number of layers, the filter size, and the number of filters, must be given to the PCANet. Once the parameters are fixed, training the PCANet is extremely simple and efficient because the filter learning in the PCANet does not involve regularized parameters or require numerical optimization solvers. Moreover, building the PCANet consists of only a cascaded linear map followed by a nonlinear output stage. Such simplicity offers an alternative yet refreshing perspective on convolutional deep learning networks and could further facilitate mathematical analysis and justification of their effectiveness.

Regardless, the extensive experiments given in this paper sufficiently demonstrate two facts: 1) the PCANet is a very simple deep learning network that effectively extracts useful information for the classification of faces, digits, and texture images, and 2) the PCANet can be a valuable baseline for studying advanced deep learning architectures for large-scale image classification tasks.

7.2 Limitation

It is also worth noting that, because of limited computational resources, we had to heavily rely on our experience in the choice of learning hyperparameters. We did not perform a systematic search for optimal hyperparameters, which often has a great impact on the performance of a neural network in limited data scenarios. The methods we used in this work are powerful and our results can be improved simply by the means of applying more computational resources without significantly changing the methodology.

7.3 Future Works

The method hardly contains any deep or new techniques, and our study so far is entirely empirical.Nevertheless, a thorough report on such a baseline system has tremendous value to the deep learning and visual recognition community, therein sending both sobering and encouraging messages: On the one hand, for future study, the PCANet can serve as a simple but surprisingly competitive baseline for empirically justifying advanced designs of multistage features or networks. On the other hand, the empirical success of the PCANet (and even that of RandNet) reconfirms certain remarkable benefits of cascaded feature learning and extraction architectures. More importantly, because the PCANet consists of only a (cascaded) linear map followed by binary hashing and block histograms, it is amenable to mathematical analysis and justification of its effectiveness. This could lead to fundamental theoretical insights about general deep networks, which currently seem to be an urgent need in deep learning.

7.4 Conclusion

So far deep learning based application provided positive feedback, however, but due to the sensitivity of healthcare data and challenges, we should look more sophisticated deep learning methods that can deal complex healthcare data efficiently. Lastly we conclude that there are unlimited opportunities to improve healthcare system.

Bibliography

[1] T. H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng and Y. Ma, “PCANet: A Simple Deep Learning Baseline for Image Classification”, IEEE Transaction on Image Processing, vol. 24, issue. 12, pp. 5017-5033, 2015.

[2] G. E. Hinton, S. Osindero, and Y.-W. Teh, ”A fast learning algorithm for deep belief nets,” Neural Comput., vol. 18, no. 7, pp. 1527–1554, 2006.

[3] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[4] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout networks,” in Proc. 30th ICML, 2013, pp. 1–9.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[6] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?” in Proc. IEEE 12th ICCV, Sep./Oct. 2009, pp. 2146–2153.

[7] J. Bruna and S. Mallat, “Invariant scattering convolution networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1872–1886, Aug. 2013.

[8] H. Lee, R. Grosse, R. Rananth, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in Proc. 26th Annu. ICML, 2009, pp. 609–616.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural network,” in Proc. NIPS, 2012, pp. 1097–1105.

[10] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, "Learning convolutional feature hierarchies for visual recognition," in Proc. NIPS, 2010, pp. 1090–1098.

[11] L. Sifre and S. Mallat, "Rotation, scaling and deformation invariant scattering for texture discrimination," in Proc. IEEE Conf. CVPR, Jun. 2013, pp. 1233–1240.

[12] C. J. C. Burges, J. C. Platt, and S. Jana, "Distortion discriminant analysis for audio fingerprinting," IEEE Trans. Speech Audio Process., vol. 11, no. 3, pp. 165–174, May 2003.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. IEEE Comput. Soc. Conf. CVPR, Jun. 2005, pp. 886–893.

[15] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in Proc. IEEE Comput. Soc. Conf. CVPR, Jun. 2005, pp. 524–531.

[16] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition," IEEE Trans. Image Process., vol. 11, no. 4, pp. 467–476, Apr. 2002.

[17] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data—With application to face recognition," Pattern Recognit., vol. 34, no. 10, pp. 2067–2069, 2001.

[18] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," in Proc. 8th IEEE Conf. Autom. Face Gesture Recognit., Sep. 2008, pp. 1–8.

[19] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 6, pp. 643–660, Jun. 2001.

[20] A. Martínez and R. Benavente, "The AR face database," CVC, Barcelona, Spain, Tech. Rep. #24, 1998.

[21] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, “The FERET database and evaluation procedure for face-recognition algorithms,” Image Vis. Comput., vol. 16, no. 5, pp. 295–306, 1998.

[22] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” Univ. Massachusetts, Amherst, Amherst, MA, USA, Tech. Rep. 07-49, 2007.

[23] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face description with local binary patterns: Application to face recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[24] Y. Jia. (2013). Caffe: An Open Source Convolutional Architecture for Fast Feature Embedding. [Online]. Available: http:// caffe.berkeleyvision.org/

[25] X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” IEEE Trans. Image Process., vol. 19, no. 6, pp. 1635–1650, Jun. 2010.

[26] W. Deng, J. Hu, and J. Guo, “Extended SRC: Undersampled face recognition via intraclass variant dictionary,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 9, pp. 1864–1870, Sep. 2012.

[27] J. Lu, Y.-P. Tan, and G. Wang, “Discriminative multimanifold analysis for face recognition from a single training sample per person,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 39–51, Jan. 2013.

[28] N.-S. Vu and A. Caplier, “Enhanced patterns of oriented edge magnitudes for face recognition and image matching,” IEEE Trans. Image Process., vol. 21, no. 3, pp. 1352–1368, Mar. 2012.

[29] S. U. Hussain, T. Napoléon, and F. Jurie, “Face recognition using local quantized patterns,” in Proc. BMVC, 2012, p. 11.

[30] S. Xie, S. Shan, X. Chen, and J. Chen, “Fusing local patterns of Gabor magnitude and phase for face recognition,” IEEE Trans. Image Process., vol. 19, no. 5, pp. 1349–1361, May 2010.

[31] N.-S. Vu, “Exploring patterns of gradient orientations and magnitudes for face recognition,” IEEE Trans. Inf. Forensics Security, vol. 8, no. 2, pp. 295–304, Feb. 2013.

[32] Z. Chai, Z. Sun, H. Méndez-Vázquez, R. He, and T. Tan, "Gabor ordinal measures for face recognition," IEEE Trans. Inf. Forensics Security, vol. 9, no. 1, pp. 14–26, Jan. 2014.

[33] L. Wolf, T. Hassner, and Y. Taigman, "Effective unconstrained face recognition by combining multiple descriptors and learned background statistics," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 10, pp. 1978–1990, Oct. 2011.

[34] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz, "Fast high dimensional vector multiplication face recognition," in Proc. IEEE ICCV, Dec. 2013, pp. 1960–1967.

[35] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in Proc. IEEE Conf. CVPR, Jun. 2014, pp. 1701–1708.

[36] H. Fan, Z. Cao, Y. Jiang, and Q. Yin. (2014). "Learning deep face representation." [Online]. Available: http://arxiv.org/abs/1403.2802