AN EFFECTIVE ENHANCED FACE RECOGNITION SYSTEM BASED ON CNN

[1]Dr. T. CHARAN SINGH, [2] SNVASRK PRASAD, [3] D. KEERTHI REDDY

[1]Associate Professor, Department of Computer Science and Engineering, Sri Indu College of Engineering and Technology, Hyderabad, Telangana-501510
[2, 3] Assistant Professor, Department of Computer Science and Engineering, Sri Indu College of Engineering and Technology, Hyderabad, Telangana-501510

ABSTRACT:

This paper targets learning robust image representation for single training sample per person face recognition. Motivated by the success of deep learning in image representation, we propose a supervised auto-encoder, which is a new type of building block for deep architectures. There are two features distinct our supervised auto-encoder from standard auto-encoder. First, we enforce the faces with variants to be mapped with the canonical face of the person, for example, frontal face with neutral expression and normal illumination; Second, we enforce features corresponding to the same person to be similar. As a result, our supervised auto-encoder extracts the features which are robust to variances in illumination, expression, occlusion, and pose, and facilitates the face recognition. We stack such supervised auto-encoders to get the deep architecture and use it for extracting features in image representation. Experimental results on the AR, Extended Yale B, CMU-PIE, and Multi-PIE datasets demonstrate that by coupling with the commonly used sparse representation based classification, our stacked supervised auto-encoders based face representation significantly outperforms the commonly used image representations in single sample per person face recognition, and it achieves higher recognition accuracy compared with other deep learning models, including the deep Lambertian network, in spite of much less training data and without any domain information. Moreover, supervised auto-encoder can also be used for face verification, which further demonstrates its effectiveness for face representation.

INTRODUCTION

PROBLEM DEFINITION:

One sample per person (OSPP) face recognition is an opening problem in face recognition community. Samples in OSPP are limited, only one for each subject. Lack of samples makes it difficult to describe intra-class variations of the subject, so the performance of most face recognition algorithms, such as linear discriminant analyses , support vector machine ,subspace recognition [4], locally linear embedding and sparse representation-based classifier (SRC) , will deteriorate in OSPP.

In practical applications, face images of some subjects under different environments can be obtained, If the intra-class image variations of the multi-sample subjects can be generalized to one sample subjects, the recognition performance of OSPP will be improved [1].

OBJECTIVE:

In this paper, we propose a supervised auto-encoder, and use it to build deep neural network architecture for extracting robust features for SSPP face representation. By introducing a similarity preservation term, our supervised auto-encoder enforces faces corresponding to the same person to be represented similarly [2]. Experimental results on the AR, Extended Yale B, and CMU-PIE datasets demonstrate clear superiority of

this module over other conventional modules such as DAE or DLN. In view of the size of images and training sets, we restrict the image size to be 32 × 32, and only images of a handful of subjects are used to train the network. For example, only 20 subjects are used on the CMU-PIE and AR datasets, and only 28 subjects on the Extended Yale B dataset. Obviously more training samples will improve the stability of the learnt network and larger images will improve the face recognition accuracy .

SYSTEM ARCHITECHTURE

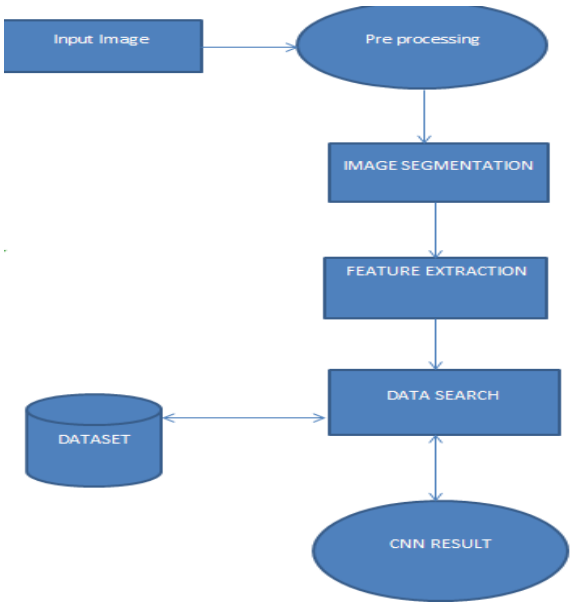


Fig-1 Architecture

MODULES AND SUBMODULES

PREPROCESSING:

In Pre-processing stage preparing the image for segmentation since image quality varies according to the conditions of acquisition. For instance, the image could be acquired under some undesired conditions, such as unevenly illuminated, noisy or low-contrasted images, which obviously influence the performance of segmentation algorithm. Hence the acquired RGB image has to undergo a sequence of preprocessing steps, optic disc removal, and background normalization [3].

IMAGE HISTOGRAM:

An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image` It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance [4].

Image segmentation:

Image segmentation is the process of dividing an image into multiple parts. This is typically used to identify objects or other relevant information in digital images.

Feature Extraction:

Feature extraction a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector. This approach is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as image matching and retrieval [5].

Noise Removal

Digital images are prone to various types of noise. Noise is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene. There are several ways that noise can be introduced into an image, depending on how the image is created. For example:

If the image is scanned from a photograph made on film, the film grain is a source of noise. Noise can also be the result of damage to the film, or be introduced by the scanner itself.

- If the image is acquired directly in a digital format, the mechanism for gathering the data (such as a CCD detector) can introduce noise.
- Electronic transmission of image data can introduce noise.

IMAGE FILERING:

Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement [6].

Project design

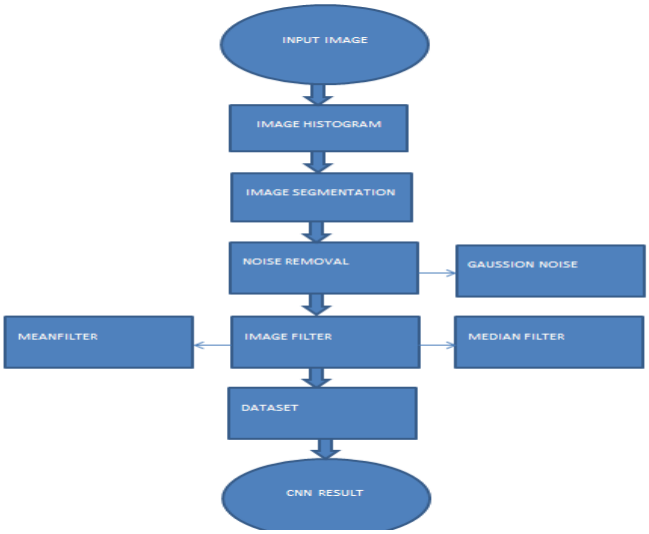


Image Database:

Image Database A face image database was created for the purpose of benchmarking the face recognition system. The image database is divided into two subsets, for separate training and testing purposes.

CNN RESULT:

The results presented in this work show that CNNs perform very well on various facial image processing tasks, such as face alignment, facial feature detection and face recognition and clearly demonstrate that the CNN technique is a versatile, efficient and robust approach for facial image analysis [7].

METHODOLOGY

Convolution neural network algorithm is a multilayer perceptron that is the special design for identification of two-dimensional image information . Always has more layers: input layer, convolution layer, sample layer and output layer. In addition, in a deep network architecture the convolution layer and sample layer can have multiple [20]. CNN is not as restricted boltzmann machine, need to be before and after the layer of neurons in the adjacent layer for all connections, convolution neural network algorithms, each neuron don't need to do feel global image, just feel the local area of the image. In addition, each neuron parameter is set to the same, namely, the sharing of weights , namely each neuron with the same convolution kernels to deconvolution image [8] [19].

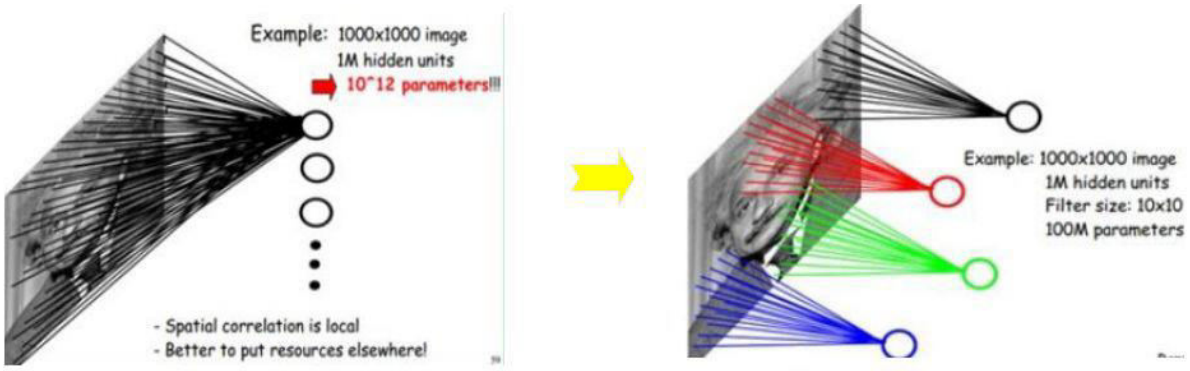
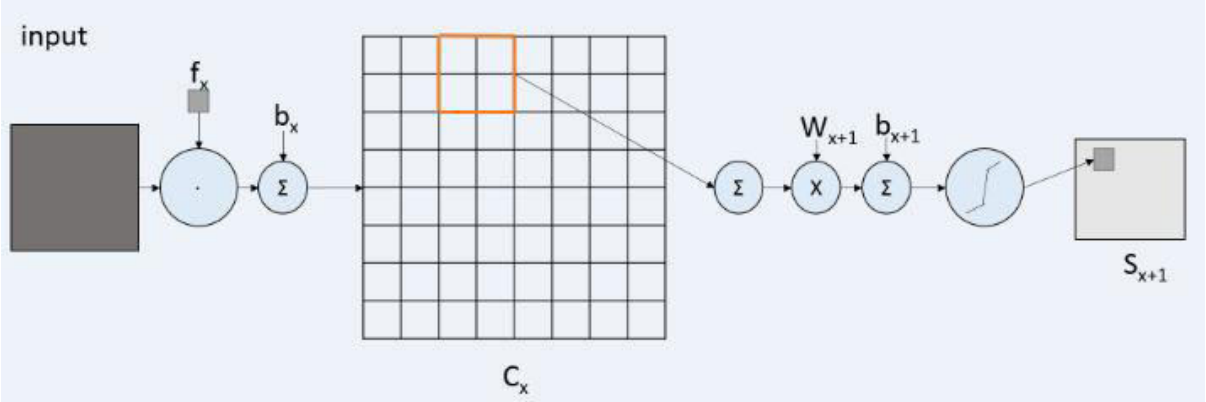


Fig-2: Fully connection vs partial connection

CNN algorithm has two main processes: convolution and sampling .

Convolution process: use a trainable filter Fx,deconvolution of the input image(the first stage is the input image, the input of the after convolution is the feature image of each layer, namely Feature Map), then add a bias bx, we can get convolution layer Cx [18].

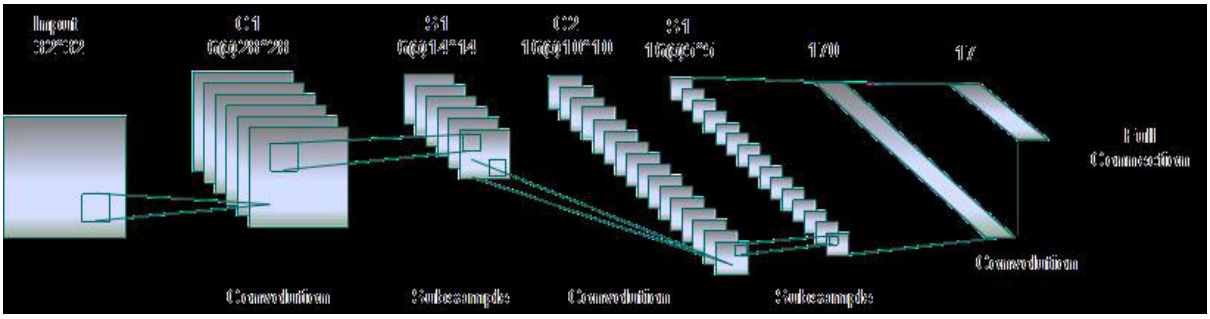
A sampling process: n pixels of each neighborhood through pooling steps,become a pixel, and then by scalar weighting Wx + 1 weighted, add bias bx + 1, and then by an activation function, produce a narrow n times feature map Sx + 1.



The key technology of CNN is the local receptive field, sharing of weights ,sub sampling by time or space, so as to extract feature and reduce the size of thetrainingparameters.The advantage of CNN algorithm is that to avoid the explicitfeature extraction, and implicitly to learn from the training data;The same neuronweights on the surface of the feature mapping, thus network can learn parallelly ,reduce the complexity of the network;Adopting sub sampling structure by time orspace, can achieve some degree of robustness, scale and deformation displacement;Input information and network topology can be a very good match, It has unique advantages in speech recognition and image processing [9][10].

CNN Architecture Design

CNN algorithm need experience in architecture design, and need to debug unceasingly in the practical application, in order to obtain the most suitable for a particular application architecture of CNN [17]. Based on gray image as the input of 96 ´ 96, in the preprocess stage, turning it into 32 32 of the size of the image. Design depth of the layer 7 convolution model: input layer,convolution layer C1, sub sampling layer S1, convolution layer C2, sampling layer S2, hidden layer H and output layer F [11].



To summarize, the Conv Layer:

- Accepts a volume of size W1×H1×D1W1×H1×D1
- Requires four hyperparameters:
  - Number of filters KK,
  - their spatial extent FF,
  - the stride SS,
  - the amount of zero padding PP.
- Produces a volume of size W2×H2×D2W2×H2×D2 where:

  - W2=(W1−F+2P)/S+1W2=(W1−F+2P)/S+1
  - H2=(H1−F+2P)/S+1H2=(H1−F+2P)/S+1 (i.e. width and height are computed equally by symmetry)
  - D2=KD2=K
- With parameter sharing, it introduces F·F·D1F·F·D1 weights per filter, for a total of (F·F·D1)·K(F·F·D1)·K weights and KK biases.
- In the output volume, the dd-th depth slice (of size W2×H2W2×H2) is the result of performing a valid convolution of the dd-th filter over the input volume with a stride of SS, and then offset by dd-th bias.
- A common setting of the hyperparameters is F=3,S=1,P=1F=3,S=1,P=1. However, there are common conventions and rules of thumb that motivate these hyperparameters. See the ConvNet architectures section below [12].
- **Convolution Demo**. Below is a running demo of a CONV layer. Since 3D volumes are hard to visualize, all the volumes (the input volume (in blue), the weight volumes (in red), the output volume (in green)) are visualized with each depth slice stacked in rows. The input volume is of size W1=5,H1=5,D1=3W1=5,H1=5,D1=3, and the CONV layer parameters are K=2,F=3,S=2,P=1K=2,F=3,S=2,P=1. That is, we have two filters of size 3×33×3, and they are applied with a stride of 2. Therefore, the output volume size has spatial size (5 - 3 + 2)/2 + 1 = 3. Moreover, notice that a padding of P=1P=1 is applied to the input volume, making the outer border of the input volume zero. The visualization below iterates over the output activations (green), and shows that each element is computed by elementwise multiplying the highlighted input (blue) with the filter (red), summing it up, and then offsetting the result by the bias [13].
- **Implementation as Matrix Multiplication**. Note that the convolution operation essentially performs dot products between the filters and local regions of the input. A common implementation pattern of the CONV layer is to take advantage of this fact and formulate the forward pass of a convolutional layer as one big matrix multiply as follows:
- The local regions in the input image are stretched out into columns in an operation commonly called **im2col**. For example, if the input is [227x227x3] and it is to be convolved with 11x11x3 filters at stride 4, then we would take [11x11x3] blocks of pixels in the input and stretch each block into a column vector of size 11*11*3 = 363. Iterating this process in the input at stride of 4 gives (227-11)/4+1 = 55 locations along both width and height, leading to an output matrix X_col of *im2col* of size [363 x 3025], where every column is a stretched out receptive field and there are 55*55 = 3025 of them in total. Note that since the receptive fields overlap, every number in the input volume may be duplicated in multiple distinct columns [14].
- The weights of the CONV layer are similarly stretched out into rows. For example, if there are 96 filters of size [11x11x3] this would give a matrix W_row of size [96 x 363].
- The result of a convolution is now equivalent to performing one large matrix multiply np.dot(W_row, X_col), which evaluates the dot product between every filter and every receptive field location. In our example, the output of this operation would be [96 x 3025], giving the output of the dot product of each filter at each location.
- The result must finally be reshaped back to its proper output dimension [55x55x96].

- This approach has the downside that it can use a lot of memory, since some values in the input volume are replicated multiple times in X_col. However, the benefit is that there are many very efficient implementations of Matrix Multiplication that we can take advantage of (for example, in the commonly used BLASAPI). Moreover, the same *im2col* idea can be reused to perform the pooling operation, which we discuss next [15].
- **1x1 convolution**. As an aside, several papers use 1x1 convolutions, as first investigated by Network in Network. Some people are at first confused to see 1x1 convolutions especially when they come from signal processing background. Normally signals are 2-dimensional so 1x1 convolutions do not make sense (it's just pointwise scaling). However, in ConvNets this is not the case because one must remember that we operate over 3-dimensional volumes, and that the filters always extend through the full depth of the input volume. For example, if the input is [32x32x3] then doing 1x1 convolutions would effectively be doing 3-dimensional dot products (since the input depth is 3 channels) [16].

CONCLUSION

In this work, we presented an experimental evaluation of the performance of proposed CNN. The overall performances were obtained using the different number of training images and test images. The convolutional neural networks achieve the best results so far. Using complex architectures, it is possible to reach accuracy rates of about 98 %. Despite this impressing outcome, CNNs cannot work without negative impacts. Very huge training datasets lead to a high computation load and memory usage, which then needs high processing power to be able to be applied usefully. In our case, the largest tested face dataset consists of 1521 grey scale images with a resolution of 384 286 pixel (Bio ID Face Database) . This database contained 23 different test images (persons). The obtained experimental results based on this database will be published in our next work. Thanks to the development of better and faster hardware, it is no problem anymore to cope with the vast amount of parameters. It can be seen, that every object algorithm has different advantages and disadvantages. Hence, it is almost not possible to create a complete, meaningful ranking, as too many different aspects have to be considered. It depends on the target application which algorithm shall be used.

REFERENCE

[1] A. A. S. Kumar, K. Ovsthus, and L. M. Kristensen, ``An industrial perspective on wireless sensor networks: A survey of requirements, protocols, and challenges," IEEE Commun. Surveys Tuts., vol. 16, no. 3, pp. 1391-1412, 3rd Quart., 2014.
[2] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, ``Energy-efficient routing protocols in wireless sensor networks: A survey," IEEE Commun. Surveys Tuts., vol. 15, no. 2, pp. 551-591, 2nd Quart., 2013.
[3] H. Yoo, M. Shim, and D. Kim, ``Dynamic duty-cycle scheduling schemes for energy-harvesting wireless sensor networks," IEEE Commun. Lett., vol. 16, no. 2, pp. 202-204, Feb. 2012.
[4] G. M. Shafiullah, S. A. Azad, and A. B. M. S. Ali, ``Energy-efficient wireless MAC protocols for railway monitoring applications," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 2, pp. 649-659, Jun. 2013.
[5] C. Wei, C. Zhi, P. Fan, and K. B. Letaief, ``AsOR: An energy-efficient multi-hop opportunistic routing protocol for wireless sensor networks over Rayleigh fading channels," IEEE Trans. Wireless Commun., vol. 8, no. 5, pp. 2452-2463, May 2009.
[6] S. Parikh, V. M. Vokkarane, L. Xing, and D. Kasilingam, ``Node-replacement policies to maintain threshold-coverage in wireless sensor networks," in Proc. IEEE Conf. Comput. Commun. Netw., Aug. 2007, pp. 760-765.
[7] S. Sudevalayam and P. Kulkarni, ``Energy harvesting sensor nodes: Survey and implications," IEEE Commun. Surveys Tuts., vol. 13, no. 3, pp. 443-461, 3rd Quart., 2011.

[8] B. Tong, G. Wang, W. Zhang, and C. Wang, ``Node reclamation and replacement for long-lived sensor networks," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 9, pp. 1550-1563, Sep. 2011.
[9] K. Bhargavi. An Effective Study on Data Science Approach to Cybercrime Underground Economy Data. Journal of Engineering, Computing and Architecture.2020;p.148.
[10] M. Kiran Kumar , S. Jessica Saritha. AN EFFICIENT APPROACH TO QUERY REFORMULATION IN WEB SEARCH, International Journal of Research in Engineering and Technology. 2015;p.172
[11] K BALAKRISHNA,M NAGA SESHUDU,A SANDEEP. Providing Privacy for Numeric Range SQL Queries Using Two-Cloud Architecture. International Journal of Scientific Research and Review. 2018;p.39
[12] K BALA KRISHNA, M NAGASESHUDU. An Effective Way of Processing Big Data by Using Hierarchically Distributed Data Matrix. International Journal of Research.2019;p.1628
[13] P.Padma, Vadapalli Gopi,. Detection of Cyber anomaly Using Fuzzy Neural networks. Journal of Engineering Sciences.2020;p.48.
[14] Kiran Kumar, M., Kranthi Kumar, S., Kalpana, E., Srikanth, D., & Saikumar, K. (2022). A Novel Implementation of Linux Based Android Platform for Client and Server. In A Fusion of Artificial Intelligence and Internet of Things for Emerging Cyber Systems (pp. 151-170). Springer, Cham.
[15] Kumar, M. Kiran, and Pankaj Kawad Kar. "A Study on Privacy Preserving in Big Data Mining Using Fuzzy Logic Approach." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 11.3 (2020): 2108-2116.
[16] M. Kiran Kumar and Dr. Pankaj Kawad Kar. “Implementation of Novel Association Rule Hiding Algorithm Using FLA with Privacy Preserving in Big Data Mining”. Design Engineering (2023): 15852-15862
[17] K. APARNA, G. MURALI. ANNOTATING SEARCH RESULTS FROM WEB DATABASE USING IN-TEXT PREFIX/SUFFIX ANNOTATOR, International Journal of Research in Engineering and Technology. 2015;p.16.
[18] Z. Han, J. Wu, J. Zhang, L. Liu, and K. Tian, ``A general self-organized tree-based energy-balance routing protocol for wireless sensor network," IEEE Trans. Nucl. Sci., vol. 61, no. 2, pp. 732-740, Apr. 2014.
[19] D. Estrin, ``Wireless sensor networks tutorial part IV: Sensor network protocols," in Proc. MobiCom, 2002, pp. 23-28.
[20] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, ``An application-specific protocol architecture for wireless microsensor networks," IEEE Trans. Wireless Commun., vol. 1, no. 4, pp. 660-670, Oct. 2002.