SOM: A Computer Vision Technique for Face Sketch Featured Database

Vineet Srivastava
Dehradun Institute of Technology
Dehradun, Uttarakhand
vineetsrivasta@gmail.com

***Abstract*— Self-Organizing Maps (SOM's) found to be an improved data management computer vision technique used for the closed matching of face vs sketch identification system based on neural network of untrained input images with trained database of images. Parameters for the SOM neural network are selected to be a minimum and maximum point for each row on a vector of training database. In this paper 64 minimum and 64 maximum pixel intensity values selected altogether using 8x8 image masking technique. Further for the design of SOM a set of 25 uniform image data used to create 5 different classes of a face image like left eye, right eye, nose, frontal face and lips for the training database. All the preprocessing for the image enhancement is done in the MATLAB software.**

***Keywords*-SOM; Masking; 2D-DCT; Computer Vision; Neural Network**

*****

I. INTRODUCTION

Human communication has two main aspects auditory (verbal) and visual (non-verbal). Non-verbal communication like facial expression, body movements, and physiological reactions provides significant information regards the state of the person. Computer vision aims to duplicate the human vision by electronically perceiving and understanding an image. Computer vision techniques use the results of artificial intelligence, pattern recognition, mathematics, computer science, psycho-physiology and other scientific areas.

II. KOHONEN SELF-ORGANIZING MAPS

Self-Organizing Maps (SOM) was introduced by a Finnish Professor, Teuvo Kohonen in 1982, thus SOM's are also sometimes known as Kohonen Maps. SOM are subtype of artificial neural network. They are trained using un-supervised learning to produce low dimensional representation of the training samples while preserving the topological properties of the input space. Thus SOM are reasonable for visualizing low-dimensional views of high-dimensional data.

Self-Organizing Maps are a single layer feed forward network where the output syntaxes are arranged in low dimensional grid i.e. 2D or 3D. Each input is connected to all output neurons attached to every neuron there is a weight vector with the same dimensionality as the input vectors. The number of input dimensions is usually a lot higher than the output grid dimension. A Self-organizing map is shown in below Figure 1.

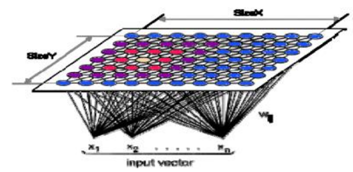

Figure 1: SOM graphical view

A. Network Architecture

Self-organizing maps are single layer feed-forward networks where the output syntaxes are arranged in low dimensional (usually 2D or 3D) grid. Each input is connected to all output neurons. Attached to every neuron there is a weight vector with the same dimensionality as the input vectors. The number of input dimensions is usually a lot higher than the output grid dimension. SOMs are mainly used for dimensionality reduction rather than expansion. The architecture for a simple self organizing map is shown in Figure 2.

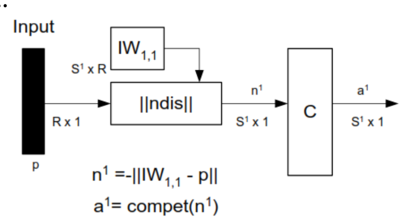

Figure 2: Network Architecture of SOM

The input vector *p* is the row of pixels of the image. The ||ndis|| box in the Figure 4.3.2.1.2 accepts the input vector *p* and the input weight matrix $IW_{1,1}$ produces a vector having S1 elements. The elements are the negative of the distances between the input vector and vectors $IW_{1,1}$ formed from the rows of the input weight matrix. The competitive transfer function accepts a net input vector for a layer and returns neuron outputs of 0 for all neurons except for the *winner*, the neuron associated with the most positive element of net input n^1. The winner's output is 1. The neuron whose weight vector is closest to the input vector has the *least* negative net input and, therefore, wins the competition to output a 1. Thus the competitive transfer function produces a 1 for output element a^1_i corresponding to i^* the winning neuron. All other output elements in a^1 are 0.

A self-organizing feature map network identifies a winning neuron using the same procedure as employed by a competitive layer. However, instead of updating only the winning neuron, all neurons within a certain neighborhood of the winning neuron are updated using the Kohonen rule.

III. DISCRETE COSINE TRANSFORM (DCT)

A. Overview

The discovery of Discrete Cosine Transform (DCT) in 1974 was an important achievement for research community working on image compression. It is a lossy data compression technique virtually used image processing. Compression standard like JPEG for compression of still images employ the basics of the DCT.

B. Background

The DCT algorithm similar to Fast Fourier Transform converts the data pixels into the sets of frequencies. The first frequencies in the set are the most meaningful than the latter i.e. least. To compress the data the least meaningful frequencies are stripped away based on the allowable resolution loss. DCT operates on a function at a finite number of discrete data points.

C. Definition

The DCT is regarded as a discrete-time version of the Fourier –cosine series. Hence it is considered as a Fourier-related transform similar to the Discrete Fourier Transform (DFT) using only real numbers.

D. 1D-DCT

The Discrete Cosine Transform is a linear invertible function or an N x N square matrix like:

F: RN -> RN
where, R denotes set of real numbers

Mathematically,1D-DCT X[k] of a sequence x[n]of length N can be defined as:

$$X[k] = \alpha[k] \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi(2n+1)k}{2N}\right), \quad k = 0,1,...,N-1$$

Each element of the transformed list X[k] in above equation is the inner dot product of the input list x[n] and a basis vector. Constant factors are chosen so the basis vectors are orthogonal and normalized. The DCT can be written as the product of a vector (input list) and the N x N are orthogonal matrix whose rows are the basis vectors.

E. 2D-DCT

The two-dimensional discrete cosine transform (2D-DCT) used for image processing. The 2D-DCT resembles the 1D-DCT transform because it is a separable linear transformation. For example, in an n x m matrix, S, the 2D-DCT is computed by applying it to each row of Sand then to each column of the result. Figure 3 shows the generic 2D-DCT architecture of an N x M input image.

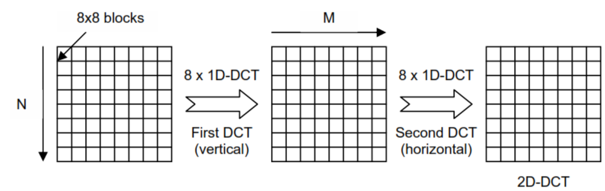

Figure 3: 2D-DCT

IV. DESIGN AND IMPLEMENTATION

In Computer Vision, detecting a Face Sketch in a digital image involves segmentation, extraction and verification of faces and possibly facial features like Left-Eye, Right-Eye, Nose, Lips and frontal face from an uncontrolled background.

A. Face Sketch Recognition

Face Sketch recognition algorithm can be classified into two main categories i.e. generative and discriminative approaches.

1) Genrative Approach: To model a digital image in terms of sketches and then match it with the query sketch or vice-versa.

Wang and Tang proposed Eigen transformation based approach to transform a digital photo into sketch before matching. In another approach, they presented an algorithm to separate shape and texture information and applied Bayesian classifier for recognition. Liu et al. proposed non-linear discriminative classifier based approach for synthesizing sketches by preserving face geometry. Li et al. matched sketches and photos using a method similar to the Eigen-transform after converting sketches to photos. Recently, Wang and Tang proposed Markov Random Fields based algorithm to automatically synthesize sketches from digital face images and vice-versa.

2) Discriminative Approach: This performs feature extraction and matching using the given digital image and sketch pair and do not generate the corresponding digital image from sketches or the sketch from digital images.

Uhl and Lobo proposed photometric standardization of sketches to compare it with digital photos. The sketches and photos were geometrically normalized and matched using Eigen analysis. Yuen and Man used local and global feature measurements to match sketches and mug-shot images. Zhang et al. compared the performance of humans and PCA-based algorithm for matching sketch-photo pairs with variations in gender, age, ethnicity and inter-artist variations. They also discussed about the quality of sketches in terms of artist's skills, experience, exposure time, and distinctiveness of features. Similarly, Nizami et al. analyzed the effect of matching sketches drawn by different artists. Klare and Jain proposed a Scale Invariant Feature Transform (SIFT) based local feature approach where sketches and digital face images were matched using the gradient magnitude and orientation within the local region. Bhatt et al. extended Uniform Local Binary Patterns to incorporate exact difference of gray level intensities to encode texture features in sketches and digital face images. Klare et al. extended their approach using Local Feature Discriminant Analysis (LFDA) to match forensic sketches. In their recent approach, Klare and Jain proposed a framework for heterogeneous face recognition where both probe and gallery images are represented in terms of nonlinear kernel similarities. Recently, Zhang et al. proposed an information theoretic encoding band descriptor to capture discriminative information and random forest based matching to maximize the mutual information between the sketch and the photo. [7]

B. Face Image Pre-processing

The programming language used to design and implement the forensic face sketch identification system code is MATLAB. The reason for using MATLAB in this project is due to its Neural Network and Image Processing toolbox that helped to obtain an efficient code.

Face image processing consists of the following steps:

- Data Gathering
- Import Faces Images to MATLAB
- Image Resize in MATLAB
- Featured Cropping

1. Data Gathering

Face images of different were taken from CUHK database which are stored under uniform light conditions and frontal position with similar dimensions. Figure 4 shows some of the original face vs sketch pictures.

Face images were then preprocessed as:

- Image conversion from RGB color to 8-bit grayscale
- Image resizing to 512 x 512 pixels

Figure 4: Face vs Face Sketches Images from the CUHK Database

2. Import of Face Images to MATLAB

All face images were preprocessed in Adobe Photoshop, then they were imported into MATLAB. The MATLAB command *imread* was used to load pictures into the workspace.

3. Image Re-size in MATLAB

After all face images were imported into MATLAB, they were resized further from 512 x 512 pixels to *8 x 8 pixels*. For this purpose the MATLAB command *imresize* was used to resize the imported pictures.

4. *Block Preparation*

For the block preparation an image is divided into individual blocks. A block consists of 8x8 pixels. Images are divided into blocks because each block is treated individually (compression steps are applied onto individual blocks, not onto the image as a whole). Figure 5 illustrates block preparation by dividing an image into a block of 8x8 pixels.

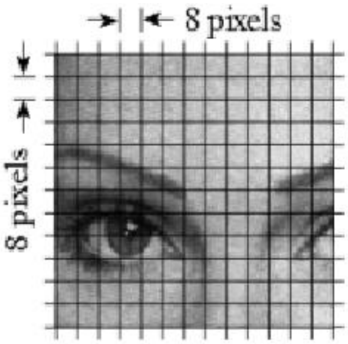

Figure 5: Image Block of 8x8 Pixels

5. Quantization

The block of 8x8 DCT coefficients are divided by an 8x8 quantization table. In quantization the low DCT coefficients of the high frequencies are discarded. Thus, quantization is applied to allow further compression of entropy encoding by neglecting insignificant low coefficients.

6. Featured Cropping

After all face images were resized, they were cropped as Left-Eye, Right Eye, Nose, Lips , frontal face with hair removes and saved under different filenames, using the MATLAB *imwrite* command.

Left Eye Right Eye Nose Frontal Face Lips

Figure 6: Featured Class Databases

V. FACE IMAGE CLASSES DCT

After all featured cropped face images were resized to 8 x 8 pixels and saved, the next step was to compress them by applying the 2D blocked DCT. When the 2D DCT is applied with a mask, high-coefficients are in the image discarded. Then the 2D IDCT is applied to regenerate the compressed image, which is blurred due to loss of quality and also smaller in size. To find a technique to apply the 2D DCT to a face image, the MATLAB help was searched in the Image Processing Toolbox. A program for 2D DCT image compression in MATLAB help was found with its source code.

The source code found from the MATLAB image processing toolbox help was used to DCT all face images, after few modifications. Before DCT compression, the image data of the resized images needed to be converted into the double format. This was achieved by using the MATLAB *double* command. The mask used for the 2D-DCT was of 8 coefficients out of 64. Figure 7 shows the DCT masking matrix.

```
mask = [1 1 1 1 0 0 0 0
        1 1 1 0 0 0 0 0
        1 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 ];
```

Figure 7: DCT Masking Matrix

Tthe masking host was set to 8 coefficients and all featured face/face sketch images were compressed using the DCT. The newly compressed face images were saved under a different filename.

FUTURE SCOPE AND CONCLUSION

In this paper I demonstrated the preprocessing techniques for face vs face sketch recognition based system using SOM neural network under a approach of computer vision. Generally, image based data have large numbers of pixel values therefore to perform any computation on those pixel values a mind frame is needed for a particular region of interest because image's intensity and color information are mostly based on very less in deviation with the nearby pixel values. In the SOM algorithm, we calculates the deviation against the higher side and lower side of image's data and meanwhile the closed pixels values discarded to save our self from duplicate values.

SOM is a fast technique based on masking of 8x8 matrix data to prepare a vector of 64 values which will be further used to match against lower and higher side values of claimed face sketch image. In future scope of this paper one can apply a pattern recognition technique based on all masking pixel values to make a set of different featured base database that could be used to faster matching of the claimed image.

REFERENCES

[1] http://www.willam ette.edu/~gorr/classes/cs449/Unsupervised/SOM.html Kohonen's Self-Organizing Map.

[2] http://www.ai-junkie.com/ann/som/som1.html, Kohonen's Self Organizing Feature Maps, Tutorial.

[3] Anil K. Jain, Brendan Klare, and Unsang Park, "Face Matching and Retrieval in Forensics Applications", IEEE Multimedia Forensic, Security and Intelligence, January-March, 2012.

[4] Himanshu S. Bhatt, Samarth Bharadwaj, Richa, Mayank Vatsa, "Memetic Approach for Matching Sketches with Digital Face Images", Submitted to IEEE Transactions on PAMI.

[5] http://dmr.ath.cx/gfx/dct/, The 2D Discrete Cosine Transform and ImageCompression.

[6] **http://mmlab.ie.cuhk.edu.hk/facesketch.html**