USE OF NEURAL NETWORKS IN AUTOMATIC CARICATURE GENERATION: AN APPROACH BASED ON DRAWING STYLE CAPTURE

R.N.Shet, K.H.Lai, E.A.Edirisinghe, P.W.H.Chung

Department of Computer Science, Loughborough University, UK
rupesh_shet@hotmail.com Fax: +44(0)1509 211586

Keywords: Caricature, Neural Networks, EDFM, Cascade Correlation, Pattern Recognition.

Abstract

Caricature is considered a rendering that emphasizes the distinctive features of a particular face. A formalization of this idea, Exaggerating the Difference from the Mean (EDFM) is widely accepted among caricaturists to be the driving factor behind caricature generation. However the caricatures created by different artists have distinctive features, which depend on their drawing style. No attempt has been taken in the past to identify these distinct drawing styles. Yet the proper identification of the drawing style of an artist will allow the accurate modelling of a personalised exaggeration process, leading to fully automatic caricature generation with increased accuracy. In this paper we provide experimental results and detailed analysis to prove that a Cascade Correlation Neural Network (CCNN) can be successfully used for capturing the drawing style of an artist and thereby used in realistic automatic caricature generation. This work is the first attempt to use neural networks in this application area and have the potential to revolutionize existing automatic caricature generation technologies.

1 Introduction

Caricature is an art that conveys humour or sarcasm to people via drawing human faces. The basic concept is capturing the essence of a persons face by graphically exaggerating their distinctive facial features. Many approaches have been proposed in literature to automatically generate facial caricatures by computers [1-6]. Most of these approaches use fixed geometrical exaggerations based on simple image analysis techniques. Others use linguistic approaches, where exaggerations are based on variations linguistically requested by a user. However within the process of creating a caricature even a professional caricaturist would not be able to geometrically or linguistically quantify all the exaggerations he/she is likely to introduce. It is observed that these exaggerations often depend on the individual drawing style adopted by an artist. The fact that we are able to identify caricatures drawn by famous caricaturists, regardless of the original image, supports this observation. Unfortunately none of the existing state-of-the-art automatic caricature generation techniques attempt to capture the drawing style of an individual artist. Yet the accurate capture of this detail would allow more realistic caricatures to be generated. From the artists' point of view, it is difficult for them to explain how they draw caricatures. This is because the rules governing their drawing style are embedded in their subconscious mind and often unexplainable. Automatic identification of an artist's drawing style using state-of-the-art image analysis and artificial intelligence techniques could provide a solution for this.

The human brain has an innate ability of remembering and recognising thousands of faces it encounters during a lifetime, where most of the faces are metrically similar. Psychologists [5,6] suggested that human beings have a "mean face" recorded in their brain, which is an *average* of faces they encounter in life. A caricaturist compares one's face with this so-called mean face, and uses their inborn talents to draw caricatures by exaggerating the distinctive facial features. This caricature drawing approach is widely accepted among psychologists and caricaturists [1,7]. Within the wider aspect of our research we are currently investigating the full automation of the above mentioned drawing style capture and related caricature generation process. The work presented in this paper limits the investigation to capturing the drawing style adopted by a caricaturist in exaggerating a single, selected facial component. It should be noted that capturing the drawing style adopted over a complete face is a challenging task due to the large number of possible variations and non-linearity of exaggerations that a caricaturist may adopt for different facial components. However non-linearity in exaggerations could be found even in the deformations made to a single facial component. This observation undermines previous research, which assumes semi-linear deformations over a single facial component such as an eye, mouth, chin, nose etc. Fortunately neural networks have the ability to capture the non-linear relationship between the input and output values in a training set. Within the research context of this paper we provide experimental results and analysis to prove that a Cascade Correlation Neural Network (CCNN) [11,12] can be trained to accurately capture the drawing style of a caricaturist in relation to an individual facial object. Further we use the results to justify that the trained CCNN could then be used to automatically generate a caricature (drawn by the same artist) of the same facial component belonging to either the same original facial figure or of a different one.

For clarity of presentation the paper is organised as follows: section-2 introduces the CCNN and discusses its suitability for the application domain. Section-3 presents the proposed methodology for the use of CCNN in identifying the drawing style of an artist. Section-4 presents experimental results and a detailed analysis proving the validity of the proposed concepts use in capturing the drawing style of a caricaturist. Finally section-5 concludes with an insight into further research that is currently being considered as a result of it.

2 The Cascade Correlation Neural Network

Artificial neural networks are the combination of artificial neurons that are similar to biological neurons [8,12,13]. These artificial neurons (simply called neurons here after) are usually connected in three layers. The first layer is an input layer, consisting of neurons that receive information (inputs) from the external environment. The second layer, which performs essential intermediate computations, is hidden from view (not directly visible from the external world) and is referred to as the hidden layer. The third layer is an output layer (target/output) that communicates the result of the weighted, summed output to the external environment or to the user. At the input layer, a linear input function computes the weighted sum of the inputs. Subsequently a non-linear transfer function transforms the weighted sum into final output values. Thus in general, all neural network architectures/topologies are based on the concept of input/output neurons, number of layers, a training function and transfer functions. Past research in neural network technology has resulted in the design of several architectures that are capable of solving specific problems. After testing and analysing various neural networks we found that the CCNN is the best for the application domain under consideration.

The CCNN [10-12] is a new architecture and is a generative, feed forward, supervised learning algorithm for artificial neural networks. It is similar to a traditional network in which the neuron is the most basic unit. However an untrained CCNN will remain in a blank state with no hidden units. Its output weights are trained until either the solution is found, or the progress stagnates. A hidden neuron is 'recruited' when training yields no appreciable reduction of error. Thus a pool of hidden neurons is created with a mixture of non-linear activation functions. The resulting network is trained until the error reduction halts. The hidden neuron with the greatest correspondence to the overall error is then installed in the network and the others are discarded. The new hidden neuron 'rattles' the network and significant error reduction is accomplished after each inclusion. Note that the weights of hidden neurons are static, i.e., once they are initially trained, they are not subsequently altered. The features they identify are permanently cast into the memory of the network, which means that it has the ability to detect the features from training samples. Preserving the orientation of hidden neurons allows cascade correlation to accumulate experience after its initial training session. Few neural network architectures allow this. The above features justify its use within the application domain under consideration. In addition the CCNN has several other advantages [11] namely: 1) It learns very quickly and is at least ten times faster than traditional back-propagation algorithms. 2) The network determines its own size and topology and retains the structure. 3) It is useful for incremental learning in which new information is added to the already trained network.

Once the architecture has been selected and the input signals have been prepared (unique properties have been found) the next step is to train the neural network. We use the Levenberg-marquardt backpropagation training function [12] due to its significant speed of operation. It would be unwise to design a network, train it and then put it into practise immediately. Its accuracy and capabilities should first be tested, evaluated and scrutinized. The testing process is known as validation. It can be said that the validation process is more important, as small errors could result in a misleading output from a network, which will be unreliable and incorrect. In section 4 we validate the use of the above network within the application domain under consideration.

3 Capturing the Drawing Style of a Caricaturist: The Proposed Methodology

Figure 1 illustrates the block diagram of the proposed drawing style capture algorithm. A facial component extractor module subdivides a given original facial image, its corresponding caricature drawn by the artist and the mean face into distinguishable components such as eye, nose, chin, mouth etc. Subsequently geometrical data from a given component of an original image and data from the corresponding component of the mean image are entered as inputs to the neural network module. The relevant data from the caricature component is entered to the module as the output. The above data is used to train the neural network. Once sufficient data points have been used in the above training process, we show that the neural network is able to predict the caricature of a novel image depicting the same facial component that was used in the training process. Below is a more detailed description of the processes involved.

Step 1: Generating Mean Face: For the purpose of our present research which is focused only on a proof of concept, the mean face (and thus the facial components) was hand drawn for experimental use and analysis. However, in a real system one could use one of the many excellent mean face generator programs [15] made available in the World Wide Web.

Step 2: Facial Component Extraction/Separation: A simple image analysis tool based on edge detection, thresholding and thinning was developed to extract/separate various significant facial components such as, ears, eyes, nose and mouth from the original, mean and caricature facial images (see figure 2). Many such algorithms and commercial software packages [14] exists that could identify facial components from images/sketches.

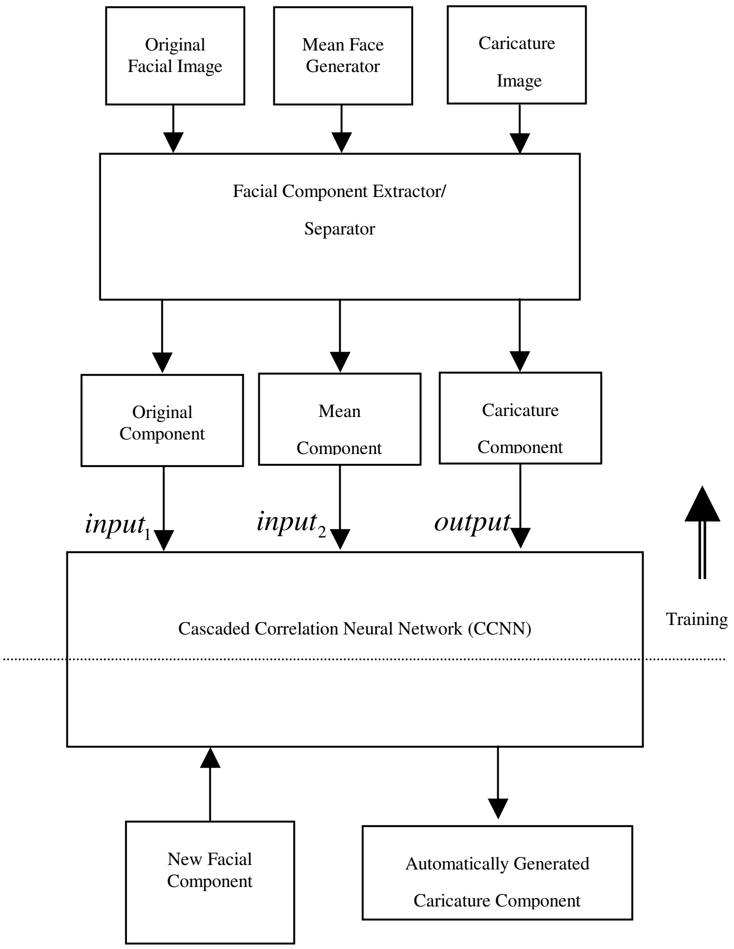


Figure 1: Proposed Drawing Style Capture Algorithm

Step 3: Creating Data Sets for Training the Neural network: Once the facial components have been extracted, the original, mean and caricature images of the component under consideration are overlapped, assuming an appropriate common centre point (see figure 3). E.g., for an eye, the centre of the *iris* could be considered the most appropriate centre point. Subsequently using cross sectional lines centred at the above point and placed at equal angular separations, the co-ordinate points at which the lines intersect the components are noted. This is done following a clockwise direction as noted by points 1,2,…8 of the caricature image data set of figure. 3. Note that figure 3 is for illustration purposes only (not to scale) and thus may not represent an accurately scaled/proportioned diagram.

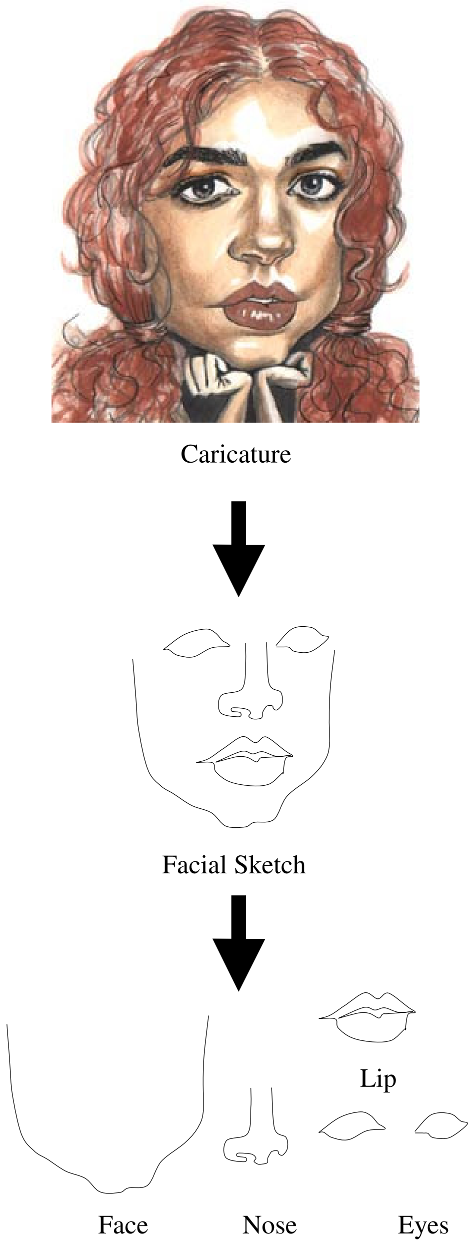


Figure 2: Facial component extraction from a caricature image

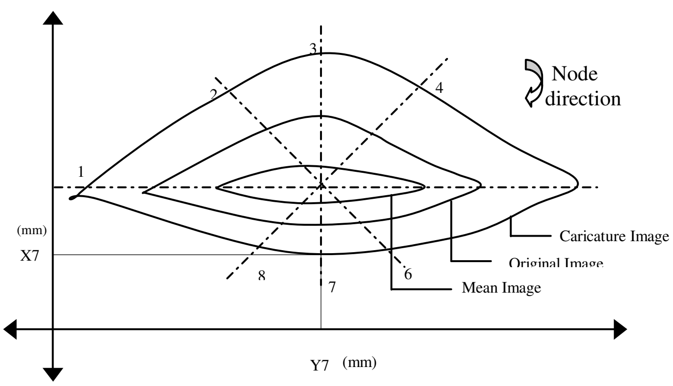


Figure 3: Creating Data Sets for Training

Step 4: Tabulating Data Sets: After acquiring the X-Y coordinate points as in step-3, they are tabulated as depicted in Table-1. The higher the number of cross sectional lines that are used, the more accurate the captured shape would be. However for clarity of presentation and ease of experimentation, we have only used four cross sectional lines in figure 3, which results in eight data sets.

	Original	Mean	Caricature
X1	25	37	13
Y1	99	99	99
X2	47	53	37
Y2	108	102	118
X3	56.8	56.8	56.8
Y3	109	102	125
X4	66	59	76
Y4	109	102	119
X5	86	75	100
Y5	99	99	99
X6	62	60	68
Y6	93	95	87
X7	56.8	56.8	56.8
Y7	92	95	86
X8	50	52	45
Y8	93	95	87

Table 1: Training Data Set

Step 5: Data Entry: Considering the fact that the neural network should be trained to automatically produce a caricature of a given facial component drawn by a particular artist, we consider the data points obtained from the caricature image above to be the output training dataset of the neural network. Furthermore the neural network is provided with the data sets obtained from the original and mean images to formulate input data. This follows the widely accepted strategy used by the human brain to analyse a given facial image in comparison to a known mean facial image.

Step 6: Setting up the Neural Network: We propose the use of the following training parameters for a simple, fast and efficient training process.

Parameter	Choice
Neural Network Name	Cascade Correlation
Training Function Name	Levenberg-marquardt
Performance Validation Function	Mean squared error
Number of Layers	2
Hidden Layer Transfer Function	Tan-sigmoid with one neuron at the start
Output Layer Transfer Function	Pure-linear with eight neurons

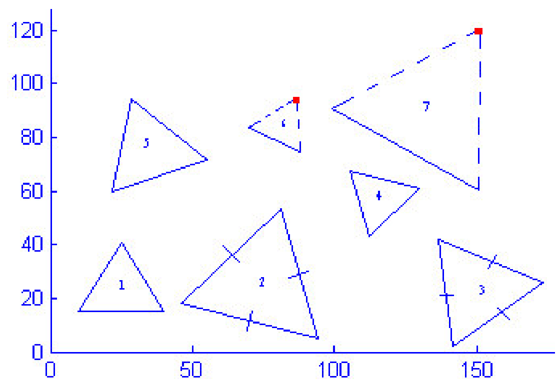
Table 2: Neural Network Specifications

Step 7: Testing: Once training has been successfully concluded as described above, the relevant facial component of a new original image is sampled and fed as input to the trained neural network along with the matching data from the corresponding mean component. In section-4 we provide experimental evidence in support of our proof of concept that a CCNN is able to capture the drawing style of a caricaturist.

4 Experiments and Analysis

Several experiments were designed and carried out to prove the suitability of using a CCNN to capture the drawing style of a caricaturist. The MATLAB neural network toolbox and associated functions [9] were used for the simulations. Three of these core experiments are presented and analysed in detail in this section. Note that experiments 1 and 2 use simple geometrical shapes for testing.

Experiment 1: In this experiment we train the CCNN using vertices of five equilateral triangles (numbered 1-5 in figure 4(a)). The first two vertices of each triangle, taken in a counter-clockwise direction are used as inputs to the neural network and the third vertex is used as the output. Subsequently given two ordered points in the Cartesian co-ordinate system, the neural network is used to predict the third point. It is found that the neural network is able to predict the third point to be a vertex of an equilateral triangle it forms with the two given points. In addition to this, the three points are arranged in the counter-clockwise direction, similar to the order used by the training datasets. The triangles numbered 6 and 7 in figure 4(a) illustrate the results of two such experiments. This proves that the CCNN is able to predict the orientation and direction with accuracy.



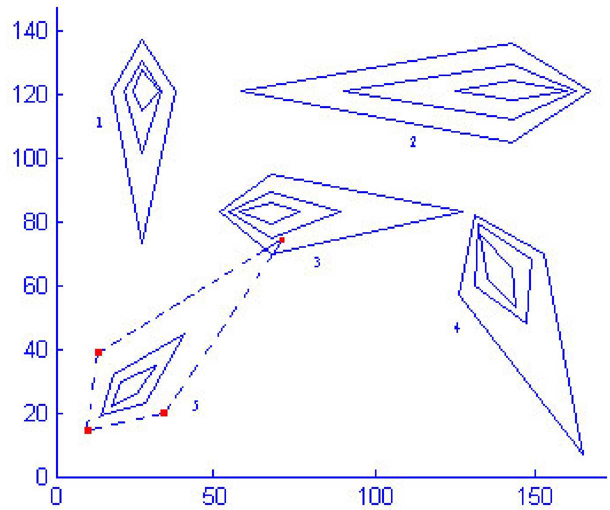
4. (a)

	X1	Y1	X2	Y2	X3	Y3
Training Data Set						
1	10.00	15.00	40.00	15.00	25.00	40.98
2	46.00	18.00	94.50	5.00	81.20	53.00
3	136.90	41.85	141.55	2.05	173.60	26.05
4	105.80	67.35	112.30	43.20	130.00	61.10
5	22.00	60.00	55.00	71.65	28.38	94.41
Test Data Set & Result						
6	70.05	83.7	88	74.53	87	94.5
7	99.2	90.8	151.05	60.58	151.3	120.55

4. (b)

Figure 4: Experiment-1 Data (a) graphical (b) tabular

Experiment 2: The results of Experiment 1 proved that the CCNN is capable of accurately predicting orientation and direction. This experiment is designed to prove that it is able to accurately predict exaggeration in addition to the qualities tested under experiment 1. The four training objects denoted by 1-4 in figure 5(a) represent the training cases. In each training object, the innermost shape denotes the mean component, the middle shape denotes the original component and the outermost denotes the caricature component. Note that the exaggeration in one direction is much greater than in the other three directions for all training objects. Object 5 in figure 5(a) denotes the test case. The input shapes (mean and original) are illustrated by continuous lines and the output (i.e. generated caricature) shape is denoted by the dotted shape. Note that the CCNN has been able to accurately predict exaggeration along the proper direction, i.e. along the direction where exaggeration is the most when the original is compared with the mean in the test object.



5. (a)

Object	X1	Y1	X2	Y2	X3	Y3	X4	Y4
1M	32	121	26.5	127.5	24	121	26.5	114.5
1O	21	121	26.5	130.5	33	121	26.5	101.5
1C	17	121	26.5	137	37	121	26.5	73
2M	143	124	160	121	143	118	125	121
2O	143	129	163	121	143	112	90	121
2C	143	136	167	121	143	105	58	121
3M	67	79	57	83	67	86	76	83
3O	67	75	54	83	67	89	89	83
3C	67	70	51	83	67	95	127	83
4M	135	62	133	76	143	65.5	144	53
4O	131	60	132	79	149	68	147	48
4C	126	57	131	82	153	70	165	7
5M	25	26	17	22	20	30	31	35
5O	28	23	14	19	18	32	40	45
Test Result								
5C	33.25	19.23	10.33	15	12.68	38.44	71.27	74.63
(M-mean) (O-Original) (C-Caricature)								

5. (b)

Figure 5: Experiment 2 Data (a) graphical (b) tabular

Experiment 3: Experiments 1 and 2 were performed on basic shapes and proved that the CCNN is capable of accurately predicting orientation, direction and exaggeration. In this experiment we test CCNN on a more complicated shape depicting a mouth (encloses lower and upper lips). Figure 6 illustrates six training cases out of 20 cases used in the experiment. In each training case, the innermost shape corresponds to a *mean mouth* sampled at eight points. For all training and test cases the shape of the mean mouth has been maintained as constant [Note: To reduce experimental complexity, the sampling points were limited to 8. This does not undermine the experimental accuracy. However more sampling points would have allowed us to train the neural network on a more regular and realistic *mouth* shape.] The middle shape corresponds to the *original mouth* and the outermost shape represents the *caricature mouth*. Both these shapes have been sampled at 8 points as illustrated in training case 1 of figure 6. Note the non-linearity in exaggeration that is shown in the training cases across the shape of the mouth. Our set of 20 training cases was carefully selected so as to cover all possible exaggerations in all eight directions. This is a must in order for the CCNN to be able to predict exaggerations accurately in all of the eight directions.

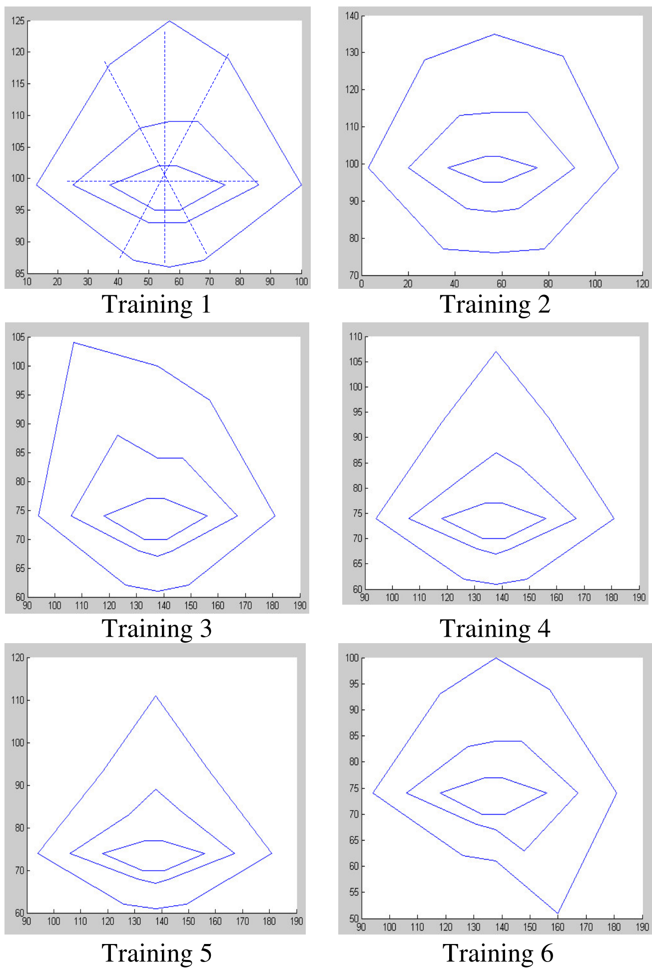


Figure 6: Testing CCNN on a real facial object under limited sampling – the training cases

Figures named “result 1-4” in figure 7 below, illustrate the test cases. They demonstrate that the successful training of the CCNN has resulted in it’s ability to accurately predict exaggeration of non-linear nature in all directions. Note that an increase in the amount of the training data set would result in an increase of the prediction accuracy for a new set of test data.

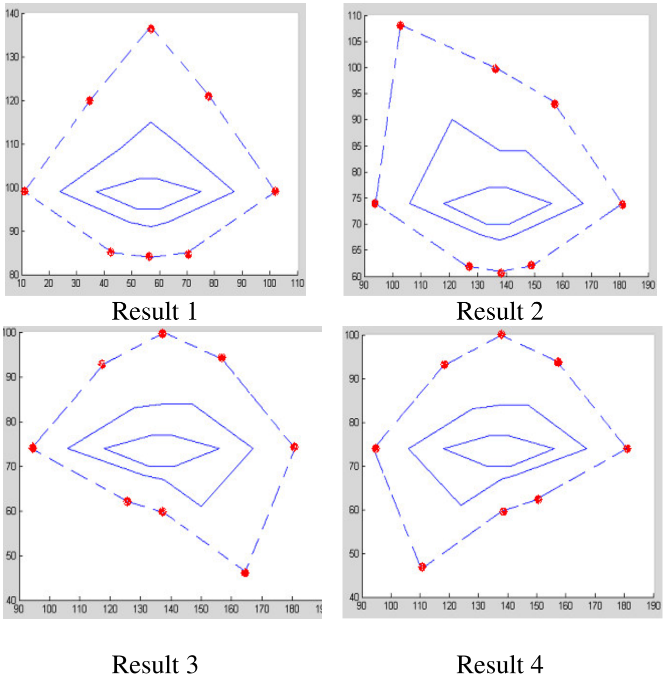


Figure 7: Testing CCNN on a real facial object under limited sampling – the test cases

4.2 Analysis: Use of CCNN in Automatic Caricature Generation

Our experiments above were designed to support the proof of concept that the CCNN can be used in capturing the drawing style of an artist and subsequent automatic caricature generation. Here we provide justifications as to why the experiments performed on limited shapes, with limited sampling would still prove enough evidence in support of the proposed idea.

Figure-8 illustrates the mean, original and caricature (drawn by two artists) images of a human eye. The original eye shows a noticeable difference in shape from the mean eye at the two ends. In the left end, the eye is curved up whereas at the right end it is curved down.

The drawing style of artist-1 shows no difference being made to the left side but a noticeable exaggeration to the difference (curved nature) in the right side. This could be a trait of this artist’s drawing style. I.e. the artist makes no exaggerations in any cartoon he draws, in the left corner of the eye, but exaggerates considerably in the right corner. The proposed CCNN based approach is able to learn this rule as proved by the results of experiments 2. Performing experiments on a larger set of original eyes (belonging to different people but caricatured by the same artist-1) will help improve prediction further. Using more sampling points around the surface of the eye (rather than 8 in our experiments) will increase the accuracy of approximating the actual shape of the eye.

In figure 8, the drawing style of artist-2 shows exaggerations being done at both ends of the eye. As justified above and supported by evidence from experiments 2 and 3, CCNN would be capable of accurately capture the drawing style of artist-2 as well. Given a new original eye, it would then be able to automatically generate the caricature, incorporating the artist’s style.

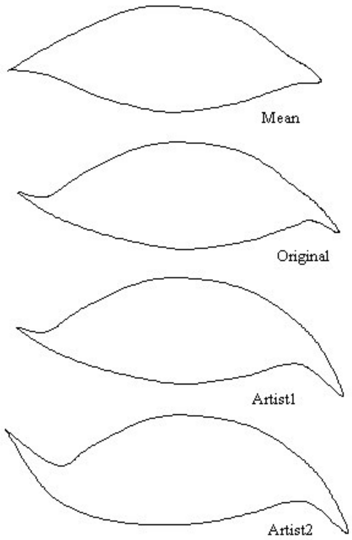


Figure 8: Comparison of the mean and an original human eye with a caricature eye drawn by two different artists

5 Conclusion

In this paper we have identified an important shortcoming of existing automatic caricature generation systems in that their inability to identify and act upon the unique drawing style of a given artist. We have proposed a Cascade-Correlation Neural Network based approach to identify the said drawing style of an artist by training the neural network on unique non-linear deformations made by an artist when producing caricature of individual facial objects. The trained neural network has been subsequently used successfully to generate the caricature of the facial component automatically. We have shown that the automatically generated caricature consists of various unique straits adopted by the artist in drawing free-hand caricatures.

The above research is a part of a more advanced research project that is looking at fully automatic, realistic, caricature generation of complete facial figures. One major challenge faced by this project includes, non-linearities and unpredictabilities of deformations introduced in exaggerations done between different objects within the same facial figure, by the same artist. We are currently extending the work of this paper in combination with artificial intelligence technology to find an effective solution to the above problem.

References

[1] S. E. Brennan. "Caricature generator: the dynamic exaggeration of faces by computer", *LEONARDO*, Vol. 18, No. 3, pp170-178, 1985

[2] Z. Mo, J.P.Lewis, and U. Neumann. Improved Automatic Caricature by Feature Normalization and Exaggeration, SIGGRAPH 2003 Sketches and Applications, San Diego, CA: *ACM SIGGRAPH*, July (2003).

[3] P.J. Benson, D.I. Perrett. "Synthesising Continuous-tone Caricatures", *Image & Vision Computing*, Vol 9, 1991, pp123-129.

[4] H. Koshimizu, M. Tominaga, T. Fujiwara, K. Murakami. "On Kansei Facial Caricaturing System PICASSO", *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, 1999, pp294-299.

[5] G. Rhodes, T. Tremewan. "Averageness, Exaggeration and Facial Attractiveness", *Psychological Science*, Vol 7, pp105-110, 1996.

[6] J.H. Langlois, L.A. Roggman, L. Mussleman. "What Is Average and What Is Not Average About Attractive Faces", *Psychological Science*, Vol 5, pp214-220, 1994.

[7] L. Redman. "How to Draw Caricatures", *McGraw-Hill Publishers*, 1984.

[8] "Neural Network" http://library.thinkquest.org/C007395/tqweb/index.html (Access date 13th Oct 2004)

[9] The Maths Works Inc, User's Guide version 4, Neural Network Toolbox, MATLAB

[10] http://www.cs.cornell.edu/boom/2004sp/ProjectArch/AppoNeuralNet/BNL/NeuralNetworkCascadeCorrelation.htm (Access date 13th Oct 2004)

[11] S. E. Fahlman, C. Lebiere. "The Cascade-Correlation Learning Architecture", Technical Report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, 1990.

[12] Carling, A. (1992). "Introducing Neural Networks". Wilmslow, UK: Sigma Press.

[13] Fausett, L. (1994). "Fundamentals of Neural Networks". New York: Prentice Hall.

[14] http://www.jasc.com/products/ (Access date 11th Nov. 2004)

[15] http://www.asahi-net.or.jp/~FX6M-FJMY/mop00e.html, M.Fujimiya. "Morpher" (Access date Nov 11, 2004)