AUTOMATIC CARTOON GENERATION BY LEARNING THE STYLE OF AN ARTIST

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BETÜL KURUOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2012

Approval of the thesis:

AUTOMATIC CARTOON GENERATION BY LEARNING THE STYLE OF AN ARTIST

submitted by **BETÜL KURUOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** ________________

Prof. Dr. Fatoş Tünay Yarman Vural
Supervisor, **Computer Engineering Department METU** ________________

Examining Committee Members:

Prof. Dr. Göktürk Üçoluk
Computer Engineering, METU ________________

Prof. Dr. Fatoş Tünay Yarman Vural
Computer Engineering, METU ________________

Prof. Dr. Faruk Polat
Computer Engineering, METU ________________

Prof. Dr. İ. Hakkı Toroslu
Computer Engineering, METU ________________

Dr. Onur Pekcan
Civil Engineering, METU ________________

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: BETÜL KURUOĞLU

Signature :

ABSTRACT

AUTOMATIC CARTOON GENERATION BY LEARNING THE STYLE OF AN ARTIST

Kuruoğlu, Betül
M.S., Department of Computer Engineering
Supervisor : Prof. Dr. Fatoş Tünay Yarman Vural

September 2012, 65 pages

In this study, we suggest an algorithm for generating cartoons from face images automatically. The suggested method learns drawing style of an artist and applies this style to the face images in a database to create cartoons.

The training data consists of a set of face images and corresponding cartoons, drawn by the same artist. Initially, a set of control points are labeled and indexed to characterize the face in the training data set for both images and corresponding caricatures. Then, their features are extracted to model the style of the artist. Finally, a similarity matrix of real face image set and the input image are constructed. With the help of the similarity matrix, Distance-Weighted Nearest Neighbor algorithm calculates the exaggeration coefficients which caricaturist would have designed for the input image in his mind. In caricature generation phase, Moving Least Squares algorithm is applied to distort the input image based on these coefficients. Caricatures generated by this approach successfully cover most of the caricaturist's key characteristics in his drawing.

Keywords: Art Application, Face Caricatures,Image Warping, Style Learning

ÖZ

BİR ARTİSTİN STİLİNİN ÖĞRENİLEREK OTOMATİK KARİKATUR ÜRETİMİ

Kuruoğlu, Betül
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Fatoş Tünay Yarman Vural

Eylül 2012, 65 sayfa

Bu çalışmada, yüz görüntülerinden otomatik karikatürler oluşturmak için bir algoritma öneririlir. Önerilen yöntem bir sanatçının çizim tekniğini öğrenir ve karikatür oluşturmak için bu tekniği veri kümesi içindeki yüz görüntülerine uygular.

Eğitim verileri, veri kümesi içindeki yüz görüntüleri ile sanatçı tarafından bu görüntüleri esas alarak çizilmiş karikatürlerden oluşur. Başlangıçta, her yüz görüntüsü ve karikatür ikilisi için belirli sayıda kontrol noktası işaretlenir ve bu noktalar sayesinde yüzün karakteristiği çıkarılmış olur. Ardından, yüz özellikleri sanatçının tarzını modellemek için çıkartılır. Son olarak, gerçek yüz görüntüsü ile girdi görüntünün benzerlik matrisi inşa edilir. Benzerlik matrisi yardımıyla, Distance-Weighted Nearest Neighbor algoritması karikatüristin zihninde yüz görüntüsü için tasarlamış olacağı abartma katsayılarını hesaplar. Karikatür üretim aşamasında, Moving Least Squares algoritması ve bu abartma katsayıları kullanılarak girdi görüntü deforme edilir. Bu yaklaşımla üretilen karikatürler, karikatüristin çizimlerinde kullandığı kilit özellikleri içerir.

Anahtar Kelimeler: Sanat Üzerine Uygulama, Yüz Karikatürleri, Resim Bükme , Stil Öğrenme

To my fiance Ozan

ACKNOWLEDGMENTS

First and foremost, I want to express my deepest gratitude to my supervisor Prof. Dr. Fatoş Tünay Yarman Vural for her extremely inciting guidance, useful advices, directing criticisms and motivating encouragements she unsparingly presented throughout the research. With her lively character, diligence, and inspiring view of life, she has been and will be an excellent role model for me.

I am also overwhelmed with gratitude for the very kind contributions of the members of Image Processing and Pattern Recognition Group of the Department of Computer Engineering. Without their knowledge and assistance this study would not have yield a success.

I have been lucky enough to have the support of many good friends. Life would not have been the same without Selçuk, Nazlı, Gülşah and Cansın, Derya and Erdem, Gözde and Burhan.

I would like to offer my deepest appreciation and thanks to my working company TUBITAK (Turkish Institution of Scientific and Technological Research) that gives importance to my personal development as much as myself; to my project manager Özgür Yürekten for his kind supports; to my team leader Halime Gök who, with her lovely jokes, cheered me up and uplifted my spirit especially when I am in a miserable mood; to İbrahim Taşyurt whom I believe I pestered with my endless questions about the thesis process; and to all my office-mates who passed through nearly all the stages of the thesis together with me.

I would like to thank Zülfe Eyles for spending her valuable time to proof-read my thesis. I highly appreciate her patience and support. I am also very thankful to Şermin Korkusuz for being extremely helpful, encouraging and kind during the wiriting stage of my thesis.

Most importantly, however, as none of all these would have been possible without the loving care, earnest support and endless patience of my dearest mother and father. My immediate family have always been a constant source of love, concern, support and strength all these years. I deeply thank to my sister and brother-in-law, who spared no effort whenever I needed help, and to my nephews Yusuf and Yunus who profoundly changed my life perspective.

I feel indebted to Şebnem who patiently listened to me and anytime I had a trouble, tried her best to find a solution and if she could not do anything, she just stood by me and shared all my feelings. Although I am physically quite away from her, this distance has never been able to prevent us to be on each other's side in difficult times.

Last, but surely not least, I would like to offer my heartfelt thanks to my dearly beloved Ozan. I am deeply grateful to him, since he, in my most distressful times during preparing this thesis, put up with my changing moods, has always been very kind and caring and supported and guided me out of the hardships I have fallen in. It is surely thanks to him that this thesis is now at your hands, ready to be read.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Is it possible for a fan to copy a drawing of his/her favorite caricaturist? Can he/she capture the essence of the artist's drawing style? We know that the answer to these questions is 'No' for many people. But still, in this thesis we started a quest to teach computers to imitate the magical harmony between hands and mind of a caricaturist.

Since the day a baby is born, its brain immediately starts executing pattern recognition and modeling processes. Even ancient people relied on patterns to identify and make sense of the world they live in. Classifying which animals to hunt and eat, which ones to ride and which ones to avoid requires a learning approach in its own sense.

Today, pattern recognition and modeling is not only limited the use of human mind but also used in solving complex problems that computers encounter. Several fields of study include and/or require pattern recognition techniques in wide range of applications. For example, voice recognition, weather forecast and character recognition all use pattern recognition in solving problems, although they all have their own particular difficulties. Not a single day passes that a new study with better results and different approaches emerge. As hinted above, this study also falls into this ever growing research field that has lots of ongoing studies on pattern recognition and modeling.

Face caricature is an imitation of a human face created by a caricaturist by exaggerating facial features. Previous works [2] show that, everyone draws a mean face in their mind by collecting common features of all the faces they encounter in their lifetime.

Within the scope of this study, a caricaturist is an artist who compares the mean image in his mind with the facial image of the real person, and then can draw exaggerated or distorted features of the face of a person. The output of his/her art is the caricature. In this thesis, it is aimed that the art of drawing caricature, the caricaturist's point of view, and the caricaturist's talent of exaggerating can be learned and reproduced by a computer.

The caricaturist creates caricatures parallel to his/her talent. All the works of art created during lifetime free caricaturist of his/her mortal being and grant immortality as long as these artworks remain. Well then, what about the talent he/she possesses? Nevertheless, talent is limited to the lifetime of artist. Modeling talent or style of an artist enables him/her to be immortal not only through artworks created in lifetime, but also through his/her talent that will be used to create new caricatures even after death. However, this modeling process mentioned is difficult to implement, because it denotes modeling brain functions. Also, due to the fact that face caricature drawing process is exaggerating and drawing facial features in a non-linear way, modeling this process is troublesome. During modeling, several parameters are taken into account. Furthermore, complexity is increased owing to the impossibility of applying exaggeration techniques on all points in the face. Previous studies were not able to present success of their models arithmetically. This is also another proof of the challenge this field of study offers. Therefore, comments on the results are limited to mere visual evaluation of output images. Further in this study, modeling process described above is implemented. Moreover, an arithmetic proof of success is shown in detail with the help of a new technique.

A solution to this challenge can be found in the study of Chiang et al [6]. This system handles all facial components (eye, nose, eyebrows etc.) separately. Average features for every component is calculated by using all the images in the learning set. In addition to the average features, a normative range of values (in terms of minimum and maximum) for each component is also defined. For example; if normative range for eye width is assumed 50 to 100 pixels, an eye with 49 pixels width is labeled as ”narrow” while another eye with 101 pixels width is labeled as ”wide”. After defining both average features and normative values, all facial components in an input image are evaluated with respect to these values in order to define whether the component is in the defined range or not. The components those falls into the range are kept unchanged, as the components those are not in the range are redrawn by using exaggeration techniques. This approach requires focusing on each component rather than whole image by labeling each component as normal or not and calculating the exaggeration rate in each component group afterwards. Another example of similar approach is suggested by Lin Liang et al. [7]. This is a sample-based approach to observe and learn the artist's drawing style and produce new sketches. In Liang's system first, exaggeration prototypes, which describe greatest variance in the training data, are selected. During the test phase, the most suitable prototype for the test image is found in order to estimate the exaggeration techniques and amounts. At this stage, however, describing the caricaturist's technique with one prototype may not be enough for the input image. If a face consists of dominant facial characteristics of different prototypes and is used as an input image, the caricature generated by the machine will not be similar to that of the caricaturist's drawing. Additionally, labeled data does not contain the position of ears or any labeled hair properties. Therefore, the caricature generated by the machine does not capture the style of the artist on ears and hair. In this study, a new approach that proposes a better solution for modeling the style of a caricaturist is proposed.

This study may sound more interesting if the benefits of learning a caricaturist's drawing style are detailed. First of all, the ability to learn an artistic approach and producing art pieces will mean that a computer can mimic and model mind-hand harmony of a human being. Also, because of the fact that facial features are exaggerated, face recognition algorithms applied on generated caricatures will probably give better results than they are applied on real images. Same thing can be said in algorithms for detecting facial expressions. These two fields will be our future work based on this study. There is another area of use for this study which may seem a bit off the beaten track. In order for the computer to learn the caricaturist's style, the caricaturist must actually have a very own style. This leads to the point that by improving this study, it would be possible to identify if a caricature belongs to a certain artist or not. Even it may be possible to comprehend whether a particular artist has a consistent style. Another exciting aspect of this study is that it is not only applicable on caricatures but also on sketches. Moreover, it is designed in an object independent approach that enables to be applied not only on human faces but on many objects. For example, an artist's car sketches can be mimicked by learning from the sketches he/she has drawn.

1.2 Contributed Work

We present the method, named ACGen as an abbreviation for ”Automatic Cartoon Generator”, that we suggest for modeling the style of an artist. Our approach is based on the following assumption: if a caricaturist draws the caricatures of twins at different times, he draws nearly the same caricatures for both twins. This assumption is based on the fact that the caricaturist takes the difference between the mean face in his mind and the model face. Therefore, we assume that as the distances between the mean face and two similar faces are close to each other, the cartoons of these similar faces are also mathematically similar.

Our method consists five steps: Firstly, we form the training set, which consists of a set of face images and their cartoon versions. Next, we compute a similarity matrix using Principal Component Analysis. Then, we propose an exaggeration technique using the Distance-Weighted Nearest Neighbor Algorithm. After that, we automatically deform the test face image using the Moving Least Squares Algorithm. Finally, we create the caricature of the test face image using an edge detection technique. Chapter 4 presents the experimental results of the proposed method in detail.

In our study, a new similarity metric, that is inspired from a commonly used metric [9] in face recognition algorithms, is used to calculate the similarity between two faces.

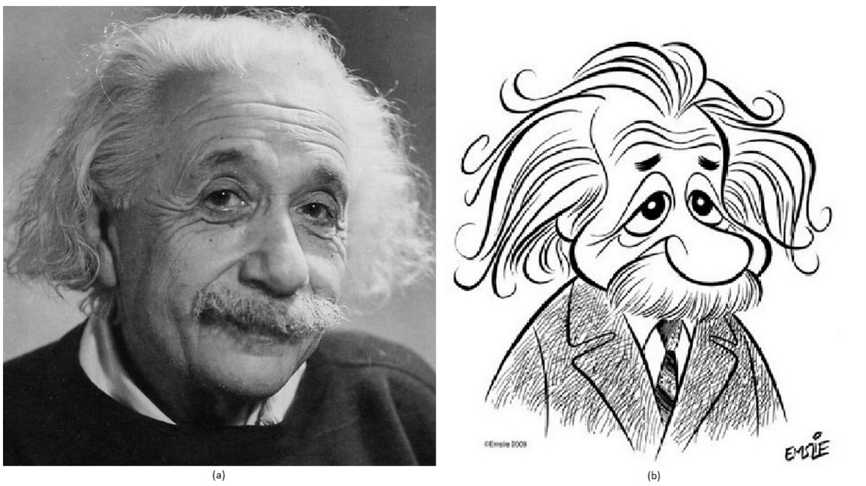


Figure 1.1: (a) Einstein’s Face Image, (b) Einstein’s Caricature

1.3 Organization

The next chapter, Chapter 2, includes studies previously made on automatic caricature generation and learning caricaturist drawing style. Before moving into the details of ACGen, existing studies are explained in order to give the reader basics of this field. Chapter 3 gives a detailed description of the ACGen. These details show the unique features of ACGen as well as describing important details of known methods. In this chapter, especially data generation, and then feature extraction and feature space creation processes are shown in detail. Afterwards, the role that PCA played in making generated feature space sparse is explained. Then, the equation, which is prepared for calculating the similarity between two images, and its technicalities are presented. Later, the role of Distance Weighted Nearest Neighbor Algorithm in ACGen is detailed. Technicalities of Moving Least Square Algorithm, a deformation technique, are thoroughly explained. At the end of this chapter, a successful edge detection method, which is used in ACGen, is mentioned. Chapter 4 focuses on practical utilization. Most of the methods mentioned in Chapter 3 use method-specific parameters. Several experiments were performed in order to determine the optimum values of these parameter. These experiments, the results of them along with the results ACGen that used the results of experiments are presented to the reader. It is difficult to prove the success of this study mathematically. Yet, in this chapter it is proven via using similarity calculation method proposed in this thesis. The last chapter lists pros and cons of using ACGen. Moreover, some further research topics, that base on this study, are listed.

CHAPTER 2

BACKGROUND ON AUTOMATIC CARICATURE GENERATION

In this chapter, previous studies made on caricature generation are provided. The existing studies are explained in order to give reader some background knowledge on the subject area.

2.1 Studies on Automatic Caricature Generation

”Caricature is a graphical coding of facial features that seeks, paradoxically, to be more like a face than the face itself. It is a transformation which amplifies perceptually significant information while reducing less relevant details. The resulting distortion satisfies the beholder's mental model of what is unique about a particular face.” [1]

Shet et al. [2] defined caricature drawing as the process of comparing one's face with the 'mean face', which is the average of the faces encountered during a lifetime, and drawing caricatures relying on the self-exaggerated distinctive facial features. This idea suggests that different caricaturists living in different environments and interact with different people have distinct mean faces in their minds. Hence, caricatures of the same faces, drawn by different artists are vastly different. If we want to model the drawing style of a caricaturist, we must take the mean face in the mind of the caricaturist into account.

As being a caricaturist herself and have been studying on cognitive science for nearly thirty years , Brennan pioneered in defining basics of caricature generation and pointing out obstacles encountered in improving the success. In her study [1], she prepared a system that generates caricatures by comparing two distinct face images and exaggerates the differences between the two faces and draws them as caricatures. However, although she was successful in automatically creating caricatures, her study did not include any type of learning system for the drawing style.

Automatic Caricature generation has been the subject of various fields in art and engineering. However, in this study we restrict ourselves to the methods available within the computer vision. As an example, Akleman [4] presented a tool to obtain caricatures by deforming the real photographs while Chen et al. [3] focused on automatic face sketch generation from face images rather than caricaturizing them. Moreover, Junfa Liu et al. [5] proposed an automatic facial caricature generation system, which use caricatures, collected from the Internet. Nevertheless, none of these studies deals with learning the style of an artist for caricature generation, which is the focal point of our study.

2.2 Studies on Caricature Generation with Caricaturist's Style Modelling

In addition to the mentioned methods that solely focus on drawing caricatures, there are some other studies which mimic the drawing style of a caricaturist. An example of this approach can be found in the study of Chiang et al [6]. This system offers an automatic caricature generation technique that highly focuses on the unique features of a face. They created the face mesh by using 119 points and edges that use these points as vertices. Also, in order to determine the positions of these vertices on the mesh, they are divided into 8 groups. Depending on this new hierarchy of the face and using the caricature drawn by the artist, the system warps the image and thus generates the caricature.

In terms of results, this study generates caricatures successfully even using only one source caricature. However, these results are critically dependent on the mesh structure constructed. And the mesh structure is fragile, because the number of vertices used is very high and grouping mechanism is not clearly expressed. In other words, a small change in these groups would cause a big difference in the generated caricature.

Another example of similar approach is suggested by Lin Liang et al. [7]. This study suggests a different approach that divides the process of caricature generation into two steps, named shape exaggeration and texture style transferring. Firstly, Partial Least Square Algorithm (PLS) is applied on the training set that includes 91 image-caricature pairs of different people.

As a result of PLS, exaggeration prototypes, which describe greatest variance in the training data, are selected. Then, the best prototype suited to the test image, which estimates the exaggeration techniques and amounts, is selected. After that, features of the selected prototype are applied on the test image. For example, if the prototype includes the changes like stretching nose for 50%, widening the mouth for 20% and minimizing the eyes for 20%, these will all be applied on the test image. Finally, edge detection techniques will be applied on the image in order to create the output caricature of ACGen.

This study is successful in terms of results; nevertheless it is highly dependent on facial features presented in the dataset. For example, assume that the dataset is created using Japanese faces and a Caucasian face image is used as the test image. Chosen prototype will not be suitable for the test image. Because, the overall effect of changing eyes would be different for a Japanese face and a Caucasian face. As another point, this study requires manual marking of 70 points on a face in face extraction phase and even choosing 70 specific points to mark is a difficult process. Also, none of these 70 points cover ears or hair, so there is not any exaggeration made in these facial parts. So this system can only be called a partial caricature generation system for human face.

CHAPTER 3

A MODEL FOR THE DRAWING STYLE TO GENERATE CARICATURE AUTOMATICALLY

In this chapter, Automatic Caricature Generation (ACGen) is introduced. Known methods, taken from referenced studies, used in ACGen and roles they have taken in this study are described in detail. Moreover, unique aspects that this study offers are also recounted.

3.1 Architecture of the Automatic Caricature Generation

3.2 Preparation of the Training Dataset

For the training dataset, FEI Face Database is used under permission of the dataset owners [10]. From this data set, 50 real face images are used. Three different caricaturists drew the caricature of 50 people. 30 of the real images and three set of corresponding caricatures are used for training (The reason why we used 30 images but not 40, for example, is explained in detail in Chapter 4). The rest of the images are preserved for the test dataset. Figure 3.2 shows the sample face images and the corresponding caricatures.

In order to generate the training dataset, a fixed number of control points (In Chapter 4, it is explained how we determined the number of control points) are defined on the face, heuristically. These control points are assumed to represent the characteristic properties of the face. Every control point stands for a point in the image which is defined with (x, y), **x and y values on the coordinate system**. Figure 3.3 indicates the control points on a sample face image. In this study, for each image 43 control points are labeled by hand. The total 30 of the real images and their corresponding caricatures in the training dataset are indexed by a

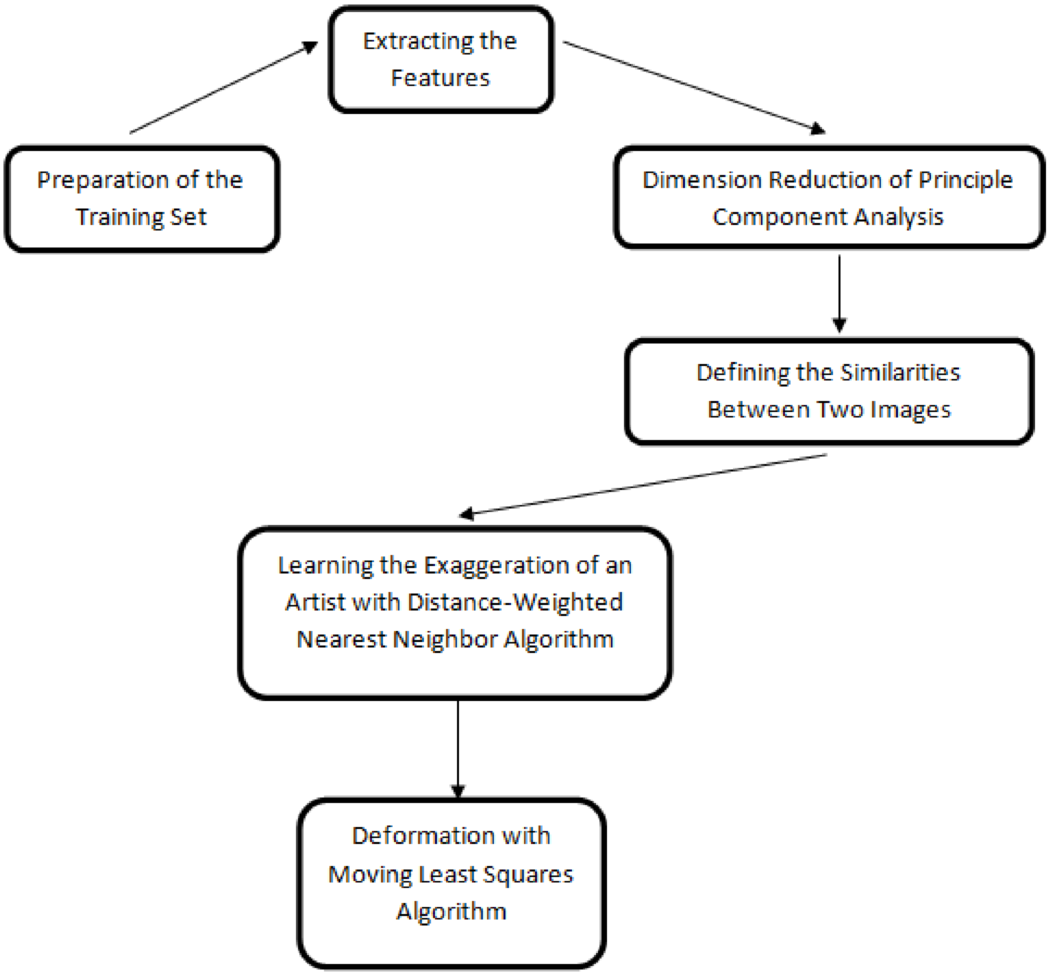


Figure 3.1: Architecture of the Automatic Caricature Generation

simple user interface.

3.3 Extracting the Features

Using the control points, we extracted a set of features for every real image. We did the same thing for each corresponding caricature. Below, the details of the process of extracting the features from the control points labeled for each image are explained where **m** is the number of labelled control points (43 for this work) and **n** is the number of training images (30 for this work):

1. The images are aligned according to a specific reference point. In this study, the tip of

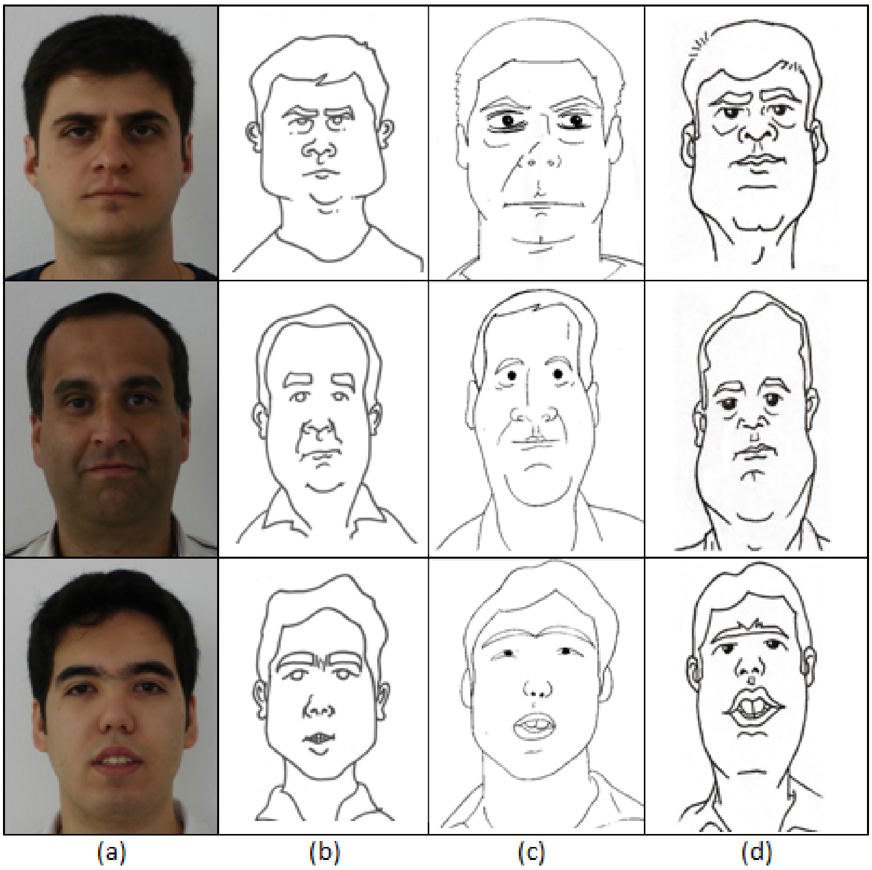


Figure 3.2: (a) Real Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

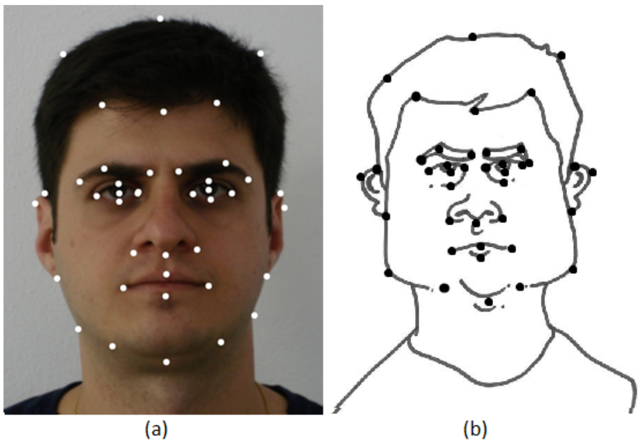


Figure 3.3: A Sample image, 43 points labelled, (a) Control Points in Original Image, (b) Control Points in Caricature

the nose is taken as the reference point. Each image and each caricature are represented as follows:

j^{th} Image:

$$I_j = \begin{pmatrix} i_{j1} & i_{j2} & . & . & i_{jm} \end{pmatrix} \tag{3.1}$$

$$where\ i_{jk} = i_{jk}(x, y)$$

indicates the coordinate of k^{th} control point on j^{th} image.

j^{th} Caricature:

$$C_j = \begin{pmatrix} c_{j1} & c_{j2} & . & . & c_{jm} \end{pmatrix} \tag{3.2}$$

$$where\ c_{jk} = c_{jk}(x, y)$$

indicates the coordinate of k^{th} control point on j^{th} caricature.

2. The mean value of the control points for each index label for real images (I_{mean}) is calculated with the following formula:

$$I_{mean} = \begin{pmatrix} i_{mean,1} & i_{mean,2} & . & . & i_{mean,m} \end{pmatrix}, \tag{3.3}$$

where for each k^{th} control point from 1 to m, $I_{mean,k}$ is calculated by

$$I_{mean,k} = \frac{\sum_{t=1}^{n} i_{tk}}{n}. \tag{3.4}$$

3. The mean value of the control points for caricatures (C_{mean}) is calculated with the following formula:

$$C_{mean} = \begin{pmatrix} c_{mean,1} & c_{mean,2} & . & . & c_{mean,m} \end{pmatrix}, \tag{3.5}$$

where for each k^{th} control point from 1 to m, $C_{mean,k}$ is calculated by

$$C_{mean,k} = \frac{\sum_{t=1}^{n} c_{tk}}{n}. \tag{3.6}$$

4. ΔI_j is the subtraction of j^{th} image from the mean image. For every image a set of features (ΔI_j) is prepared. Each image (I_j) can be represented as follows:

$$I_i = I_{mean} + \Delta I_i. \tag{3.7}$$

5. ΔC_j is the subtraction of j^{th} image from the mean image. For every image a set of features (ΔC_j) is prepared. Each image (C_j) can be represented as follows:

$$C_j = C_{mean} + \Delta C_j. \tag{3.8}$$

.

Finally, the following matrices are constructed.

The real image feature matrix M_r is defined as follows,

$$M_r = \begin{pmatrix} \Delta i_{11} & \Delta i_{12} & \cdots & \Delta i_{1m} \\ \Delta i_{21} & \Delta i_{22} & \cdots & \Delta i_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta i_{n1} & \Delta i_{n2} & \cdots & \Delta i_{nm} \end{pmatrix} \tag{3.9}$$

$$where\ \Delta i_{nm} = \Delta i_{nm}(x, y).$$

Corresponding caricature feature matrix M_c is defined as follows,

$$M_c = \begin{pmatrix} \Delta c_{11} & \Delta c_{12} & \cdots & \Delta c_{1m} \\ \Delta c_{21} & \Delta c_{22} & \cdots & \Delta c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta c_{n1} & \Delta c_{n2} & \cdots & \Delta c_{nm} \end{pmatrix} \tag{3.10}$$

$$where\ \Delta c_{nm} = \Delta c_{nm}(x, y).$$

3.4 Similarity Matrix with Principal Component Analysis

A feature space is created by combining features of all images in training set and features of test image. Some of the features in feature space is close to zero, although some others are directly dependent on other features. In order to eliminate non-essential features and ensure sparsity in feature space, ACGen uses PCA. With the help of created feature space, all the images in training set and test image can be represented with far less features.

The feature matrix M_r for the test image is computed using Equation (3.9). During the feature matrix extraction, mean control points for real images (I_{mean}) are calculated from the training set via the equation (3.4).

Test image feature vector is defined as follows,

$$I_{test} = I_{mean} + \Delta I_{test}. \tag{3.11}$$

$$\Delta I_{test} = \begin{pmatrix} \Delta i_{test,1} & \Delta i_{test,2} & . & . & \Delta i_{test,m} \end{pmatrix} \tag{3.12}$$

Algorithm 1 Extract Features

Require: n Training ”Image-Caricature” Sets and a Test Face Image

  for $j = 1 \to n$ **do**

    $I_j \leftarrow$ Representation of j^{th} Image with m points

    $C_j \leftarrow$ Representation of j^{th} Caricature with m points

    for $k = 1 \to m$ **do**

      $I_{jk} \leftarrow I_{jk}$ - the Specific Reference Point (the Tip of the Nose)

      $C_{jk} \leftarrow C_{jk}$ - the Specific Reference Point (the Tip of the Nose)

    end for

    $I_{mean} \leftarrow$ Mean (n Images)

    $C_{mean} \leftarrow$ Mean (n Caricature)

    for $k = 1 \to m$ **do**

      $\Delta I_{jk} \leftarrow I_{jk}$ - $I_{mean,k}$

      $\Delta C_{jk} \leftarrow C_{jk}$ -$C_{mean,k}$

    end for

  end for

  $I_{test} \leftarrow$ Representation of Test Image with m points

  for $k = 1 \to m$ **do**

    $I_{test,j} \leftarrow I_{test,k}$ - the Specific Reference Point (the Tip of the Nose)

    $\Delta I_{test,j} \leftarrow I_{test,k}$ - $I_{mean,k}$

  end for

  return Training ”Image-Caricature” Sets Features & Test Image Features

$$where\ \Delta i_{test,k} = \Delta i_{test,k}(x, y)$$

indicates the subtraction of test image's k^{th} control point from mean image's k^{th} control point.

Test image feature vector is concatenated to the real image feature matrix at the last row to obtain the augmented matrix,

$$M_a = \begin{pmatrix} \Delta i_{11} & \Delta i_{12} & \cdots & \Delta i_{1m} \\ \Delta i_{21} & \Delta i_{22} & \cdots & \Delta i_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta i_{n1} & \Delta i_{n2} & \cdots & \Delta i_{nm} \\ \Delta i_{test,1} & \Delta i_{test,2} & \cdots & \Delta i_{test,m} \end{pmatrix}. \tag{3.13}$$

The columns of the (n+1)×(m) augmented matrix are correlated. At this point Principal Component Analysis is used to remove the correlated columns and reduce the dimension of the feature space. For this purpose, the eigenvalues and eigenvector of the covariance matrix of M_a are computed. All non-zero eigenvalues are chosen and new feature matrix is obtained. The reduced matrix RM_a has r columns ($r \ll m$) and n+1 rows, so each row will represent an image with new vector.

$$RM_a = \begin{pmatrix} \Delta ri_{11} & \Delta ri_{12} & \cdots & \Delta ri_{1r} \\ \Delta ri_{21} & \Delta ri_{22} & \cdots & \Delta ri_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta ri_{n1} & \Delta ri_{n2} & \cdots & \Delta ri_{nr} \\ \Delta ri_{test,1} & \Delta ri_{test,2} & \cdots & \Delta ri_{test,r} \end{pmatrix}. \tag{3.14}$$

3.5 Defining the Similarities Betvween Two Images

Finally, similarities between the test image and each real image from training set (s_{jt}) are calculated by the following equation that is inspired from commonly used metric [9], which calculates the cosine value of the angle between the two images' coefficients. This formula has a significant role in the originality of the work presented in this study.

Let,

$$u = \Delta ri_j, y = \Delta ri_{test}. \tag{3.15}$$

The cosine value of the degree between u and y:

$$\cos\theta = \frac{u \cdot y}{||u|| \times ||y||} \tag{3.16}$$

and

$$\frac{u \cdot y}{||u||} = \cos\theta \times ||y||. \tag{3.17}$$

Equation of similarity between two images' coefficients :

$$s_{jt} = \frac{\arccos(\frac{u \cdot y}{||u||} \times ||y||)}{\arccos(\frac{u \cdot u}{||u||} \times ||u||)}. \tag{3.18}$$

Since the degree between u and u is 2π and cosine value of the degree is 1, following equation is derived:

$$s_{jt} = \frac{\arccos(\cos\theta \times ||y||^2)}{\arccos(\cos 2\pi \times ||u||^2)}. \tag{3.19}$$

The similarity metric that is used in this work is

$$s_{jt} = \frac{\arccos(\cos\theta \times ||y||^2)}{\arccos(||u||^2)}. \tag{3.20}$$

The similarity vector S has entries which measures the amount of similarity of the test image and the images in the training set;

$$S = \begin{pmatrix} s_{1,test} & s_{2,test} & . & . & s_{n,test} \end{pmatrix}. \tag{3.21}$$

The vector S entries are normalized between 0 to 1 .

3.6 Learning the Exaggeration of an Artist with Distance-Weighted Nearest Neighbor Algorithm

k-nearest neighbors algorithm (kNN) is a method for classifying objects based on closest training examples in the feature space. Compared to the kNN, Distance-Weighted Nearest Neighbor Algorithm is refined as it weights each of k neighbours' contribution based on their distance to the query point by weighting closer points more [11].

We suggest a model in which the concept of exaggeration is used to represent the style of a caricaturist. We assume that the style of a caricaturist can basically be modelled by defining

Algorithm 2 Principle Component Analysis

Require: Training Real Face Images Features, Test Real Face Image Features

for $j = 1 \rightarrow n$ **do**

Real Face Images Feature Matrix (j) = Δi_j

end for

Real Face Images Feature Matrix (n+1) = Δi_{test}

Eigenvalues = Calculate Eigen Values (Real Face Images Feature Matrix)

Eigenvectors = Calculate Eigen Vectors (Real Face Images Feature Matrix)

r=0

for $j = 1 \rightarrow m$ **do**

if Eigen Values (j) $\neq 0$ **then**

Reduced Feature Space (r^{th} column) $\leftarrow$ Eigen Vectors (j)

r $\leftarrow$ r + 1

end if

end for

for $j = 1 \rightarrow n + 1$ **do**

for $k = 1 \rightarrow n + 1$ **do**

I_j = Reduced Feature Space (j)

I_k = Reduced Feature Space (k)

Similarity Matrix (j, k) = Similarity Calculation (I_j, I_k)

end for

end for

return $SimilarityMatrix$

the type and amount of certain prominent characteristics of a face. For example, if a caricaturist draws a face of a person with large nose, he draws the nose even larger to make him more distinguishable than the other faces. We call this overstating process as exaggeration style. The model for exaggeration is based on the feature matrices M_a and M_r, defined in the previous section. As we mentioned before, we assume that "a caricaturist draws similar caricatures for similar faces". Therefore, the exaggeration style of a caricaturist can be captured from the similar images in the training data set. In other words, images taken from the training set, that are similar to the input image, affect the exaggeration of the caricature more than the others. This exaggeration is calculated with the help of Distance-Weighted Nearest Neighbor Algorithm by using all the training images via equation (3.23) . Each image in the training set impacts the yielded result, proportional to its similarity to the input image.

The exaggeration amounts of the caricature are given by:

$$\Delta C_{test} = \begin{pmatrix} \Delta c_{test,1} & \Delta c_{test,2} & . & . & \Delta c_{test,m} \end{pmatrix}. \tag{3.22}$$

For each k^{th} component from 1 to m, $\Delta\ c_{test,k}$ is calculated. For this formula, M_c and S matrices are used.

Exaggeration Formula:

$$\Delta c_{test,k} = \frac{\sum_{t=1}^{n} \Delta c_{tk} \times s_{t,test}}{\sum_{t=1}^{n} s_{t,test}}. \tag{3.23}$$

The exaggeration rates are ready to be applied to the real image.

Algorithm 3 Distance-Weighted Nearest Neighbor Algorithm

Require: Similarity Matrix & Training Caricatures Features

for $k = 1 \rightarrow m$ **do**

Exaggeration $Amounts_{test,k}$ = Exaggeration Calculation (Similarity Matrix, Training Caricatures Features)

end for

return Exaggeration Amounts for the Test Image's m Number of Representation Points

3.7 Deformation with Moving Least Squares Algorithm

”The Moving Least Squares algorithm is a deformation technique that allows to compute a map f:R2 → R2 from the transformation of a set of N pivot points p in the new positions q.” [11]

For simlicity, during the application of Moving Least Square algorithm on this study, **the ”p” value is I_{test} and ”q” value is C_{test}**.

In this phase, I_{test} is used as a set of control points of the Test Image. Besides, deformed positions (C_{test}) is calculated by the following formula which is mentioned in Section 3.2. We already have C_{mean} and Δ C_t values, as

$$C_{test} = C_{mean} + \Delta C_{test}. \tag{3.24}$$

The best affine transformation l_v that minimizes the equation below for a point v given in test image[11]:

$$\sum_i w_i |l_v(p_i) - q_i|^2, \tag{3.25}$$

where the weights w_i are calculated from the following equation:

$$w_i = \frac{1}{|p_i - v|^{2\alpha}}. \tag{3.26}$$

Below is the function f of deformation where f(v) = $l_v(v)$. A **linear transformation matrix M** and a **translation T** constitutes transformation function:

$$l_v(x) = xM + T. \tag{3.27}$$

A linear system of equations is obtained via deriving the formula partially based on the free variables in T (translation). And when solved for T,

$$T = q_* - p_* M, \tag{3.28}$$

as p_* and q_* represents weighted centroids,

$$p_* = \frac{\sum_i w_i p_i}{\sum_i w_i}, q_* = \frac{\sum_i w_i q_i}{\sum_i w_i}. \tag{3.29}$$

Results above will lead to the change in l_v transformation function as:

$$l_v = (x - p_*)M + q_*. \tag{3.30}$$

Minimization problem of least square can now be presented as:

$$\sum_i w_i |\hat{p}_i M - \hat{q}_i|^2, \tag{3.31}$$

where,

$$\hat{p}_i = p_i - p_*, \hat{q}_i = q_i - q_*. \tag{3.32}$$

3.7.1 Rigid Deformations

A rigid deformation is a deformation technique that preserves the distance. Rigid deformations should be preferred to create realistic shapes and even uniform scaling should not be included as a deformation technique. Rigid Deformation has an obvious superiority against Affine and Similarity Deformation Methods in giving more realistic results. This success is directly related to the ability of maintaining rigidity and local scaling. This study requires extensive use of partial and discrete deformations of the face. Rigid Deformation is not only provides this, but also offers smooth passes between deformed parts[11].

Reference study [11] states that M value in Rigid Deformation is represented as $M^T M = I$. So the challenge of minimization will look like:

$$\min_{M^T M = I} \sum_i w_i |\hat{p}_i M - \hat{q}_i|^2 \tag{3.33}$$

After derivations of the minimization problem equation (3.33), the following formulas are obtained [11].

$$M = \frac{1}{\mu_r} \sum_i w_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^{\perp} \end{pmatrix} \begin{pmatrix} \hat{q}_i^T & -\hat{q}_i^{\perp T} \end{pmatrix} \tag{3.34}$$

where,

$$\mu_r = \sqrt{(\sum_i w_i \hat{q}_i \hat{p}_i^T)^2 + (\sum_i w_i \hat{q}_i \hat{p}_i^{\perp T})^2}, \tag{3.35}$$

$$A_i = w_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i{}^{\perp} \end{pmatrix} \begin{pmatrix} v - p_* \\ -(v - p_*)^{\perp} \end{pmatrix}^T, \tag{3.36}$$

$$\vec{f_r} = \sum_i \hat{q}_i A_i, \tag{3.37}$$

and

$$f_r(v) = |v - p_*| \frac{\vec{f_r}(v)}{|\vec{f_r}(v)|} + q_*. \tag{3.38}$$

In this study, equation (3.38) is used as the transformation function to deform the images. For each point on the images, new position information is calculated with it by the help of control points as p (I_t) and deformed positions as q (C_t). An example of this application can be seen Figure 3.4.

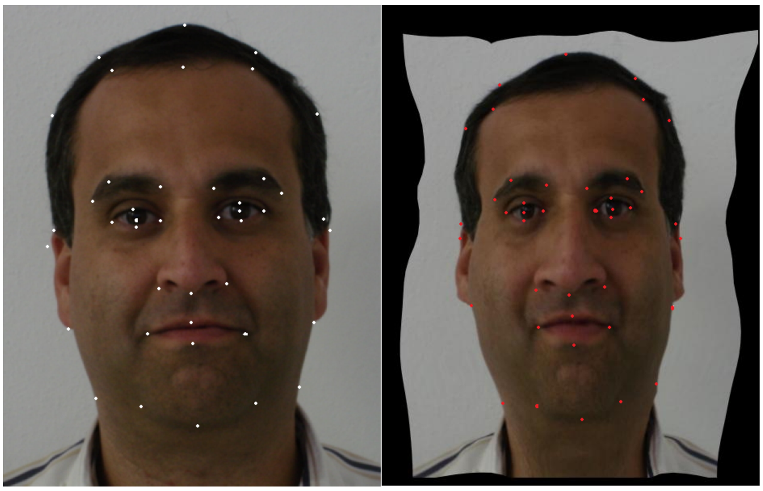


Figure 3.4: Image on the left is real image while the image on the right is MLS applied version of the same image. is the set of white dots on real face image and is the set of red dots on deformed image.

3.8 Edge Detection

When MLS Algorithm is applied on the test image, the output image will become like the examples shown in Figure 3.4. Yet, this image cannot be defined as a caricature. In order to generate a real caricature, the image is introduced into the edge detection process which will extract a caricature out of the input image. As a solution to the edge detection challenge,

Algorithm 4 Moving Least Squares Algorithm

Require: Test Image, a set of control points (p) of the Test Image and the deformed positions (q)

for v → each point in the Test Image **do**

l_v= Calculate Best Affine Transformation Funtion (p, q)

$v_{newposition} = l_v(v)$;

Plot $v_{newposition}$ on the output caricature image

end for

a method called ”Informative Binarization Based on Unsharp Masking” stands out as most plausible solution. Before deciding on this method, several edge detection techniques were evaluated. Unfortunately, most of them could not offer a satisfactory of success. Firstly, the edge detection techniques those came to the mind first (canny, prewitt, sobel etc.) are used. Yet, the resulting images obtained do not conserve the facial integrity that deformed images have. Moreover, they include inconsistent lines and so these methods were not used.

Three edge detection methods those give the best results during the tests of creating cartoons from deformed images are given below. Experiments show that among many edge detectors, steerable filters, canny, sobel etc., Informative Binarization Based on Unsharp Masking gives the best result.

3.8.1 Steerable Filter

”Steerable filters basically provide directional edge detection since they behave as band-pass filters in a particular orientation. The edge located at different orientations in an image can be detected by splitting the image into orientation sub-bands obtained by basis filters having these orientations” [15]. Steerable filter edge detection method is applied on image on the left in Figure 3.5. In this example, the image is split into orientation sub-bands with angles of 15 degrees.

Figure 3.5: Image after steerable filter application

3.8.2 Multi-level edge detectors based on the convolution matrices of base lengths 2 and 3

This method [13] uses the following parameters and it is similar to the third method which is explained below in detail. The filtering operation applied with the following mask for the output image shown in Figure 3.6.

$$Mask = \begin{pmatrix} 1 & -9 & -20 & -9 & 1 \\ -9 & 0 & 18 & 0 & -9 \\ -20 & 18 & 76 & 18 & -20 \\ -9 & 0 & 18 & 0 & -9 \\ 1 & -9 & -20 & -9 & 1 \end{pmatrix};$$

The threshold value is chosen 113.

3.8.3 Informative Binarization Based on Unsharp Masking

This is an edge detection technique that uses **effective unsharp masks on an image to enhance the edges** and **t**hresholds the image in order to obtain binarized version of the image.

1. Mask applying process

   (a) 'Unsharp Mask' is one of the six special masks used in the study [14] in detail (labeled as 'P6' in the study). (k = 25) The most important feature of the mask

Figure 3.6: Image after ”Multi-level edge detectors based on the convolution matrices of base lengths 2 and 3” application

used, compared to the other masks illustrated in the study, is that it does not cause any distortion around the edges.

$$\text{Mask} = 1/4 \times \begin{pmatrix} -2\times(k-1) & (k-1) & -2\times(k-1) \\ (k-1) & 4\times k & (k-1) \\ -2\times(k-1) & (k-1) & -2\times(k-1) \end{pmatrix};$$

(b) As mentioned in reference study[14], to generate better informative binary images, the original 3×3 mask and following smoothing filter are used to obtain a 5×5 unsharp mask.

$$\text{Smoothing Filter} = 1/9 \times \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix};$$

(c) Convoluted Mask is calculated by the process below.

Convoluted Mask = Mask * Smoothing Filter

(d) Also, Edge Enhanced Image is generated by applying the above calculated convoluted mask on the image.

Edge Enhanced Image = Image * Convoluted Mask

2. Binarization

(a) Histogram of an edge enhanced image T_p is obtained after filtering with an unsharped mask.

(b) T_p value is accepted as the threshold and after that, Edge Enhanced Image is binarized by using this threshold. The output is the generated caricature.

Figure 3.7: Image after ”Informative Binarization Based on Unsharp Masking” application

Algorithm 5 Edge Detection

Require: TestImage = The Output Image of MLS Algorithm

Image = Read Image (TestImage);

Gray Scale Image = Convert Image into Grayscale (Image) ;

Unsharp Mask = $1/4 \times \begin{pmatrix} -2\times(k-1) & (k-1) & -2\times(k-1) \\ (k-1) & 4\times k & (k-1) \\ -2\times(k-1) & (k-1) & -2\times(k-1) \end{pmatrix}$;

Smoothing Filter = $1/9 \times \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$;

Larger Unsharp Mask = Unsharp Mask * Smoothing Filter ;

Edge Enhanced Image = Image * Larger Unsharp Mask;

Scaled Edge Enhanced Image =Scale **Edge Enhanced Image** Between 0 and 255;

T_p = Threshold of **Scaled Edge Enhanced Image**;

Caricature Image = Binarize **Scaled Edge Enhanced Image** with Threshold T_p;

CHAPTER 4

EXPERIMENTS ON AUTOMATIC CARICATURE GENERATION

This chapter is designed to cover information about practical usage of ACGen and it contains details of the experiments performed. The way that data generation, feature extraction and parameter validation processes, those form the preliminary work section of ACGen, are carried out is introduced along with the results of these processes. Furthermore, experiments those prove the success of ACGen and the outcomes of these experiments are presented in ”Experimental Results” section.

4.1 Training and Test Data Generation

Right after deciding the thesis topic, thesis progress proceeds with preparing dataset. A dataset of face images is required. After searching a suitable face dataset, ”FEI Face Dataset” is selected and required academic permission is obtained. This dataset contains mug shots of 200 people aged between 19 and 40 with unique appearances taken from 14 different angles. However, only frontal images of first 100 individuals are used in this study. After real face image dataset is chosen, caricature generation phase is started. Three amateur caricaturists, those all have unique styles of their own, draw three sets of caricatures. They were requested to draw all caricatures with the same pen and emphasizing their styles. They are given the real image set, and three sets of 100 caricatures were created. Finally, the dataset containing 100 real images and 300 caricatures is created. All the images and caricatures in this set are sized 468*596 pixels. All dataset are given in Appendix C.

4.2 Control Point Extraction

After the dataset is prepared, control points are marked in all the images in this dataset with using a program developed in MATLAB environment. This program lets x-y coordinates of specific control points (tip of the node, left end of the left eye, etc.) to be marked on the face image and stored in a matrix.

For this study, 43 points are selected to represent a face. These points are shown in Figure 3.3. In choosing these control points, all facial components (eyes, eyebrows, nose, ears, chin, etc.) and general frame of the face are taken into account. However the real focus in determining control points is directed to finding optimum number and locations of these, so that facial components will be represented efficiently. After an intense study on each component, 43 points are selected to represent a face.

Control point extraction program explained above is implemented in order to let 43 selected points to be marked and saved on the face image. For every image (both real image and caricature), 43 points are marked one by one by clicking by an easy to use interface. At this point, it is important that this whole process is carried out by only one person because there is also a subjective style for clicking these points. Because, every individual perceives and defines these points with his/her own perspective and creates a selection model accordingly. This dilemma is encountered during the clicking process and only one hand is used in order to prevent biasness.

4.3 Validation of the Parameters

Number of Training Images

The sample space of the learning data directly affects the performance of the method mentioned in the thesis. A preliminary study is held in order to understand this effect and to estimate the most appropriate value for learning data size. As a result of this study, the answer to the question ”how many pictures in the training set one should need in order to obtain a visually satisfactory result” is obtained. During the preliminary experiments, the number of learning images is switched between 5 and 35, with the number of test images are kept constant at 10. This preliminary study was repeated separately for three caricaturists.

During the experiments, if a resemblance between caricaturist-drawn and system-drawn caricatures has been achieved, the process is accepted as successful. This resemblance is calculated via Equation (3.20), that calculates the similarity of the two face images .

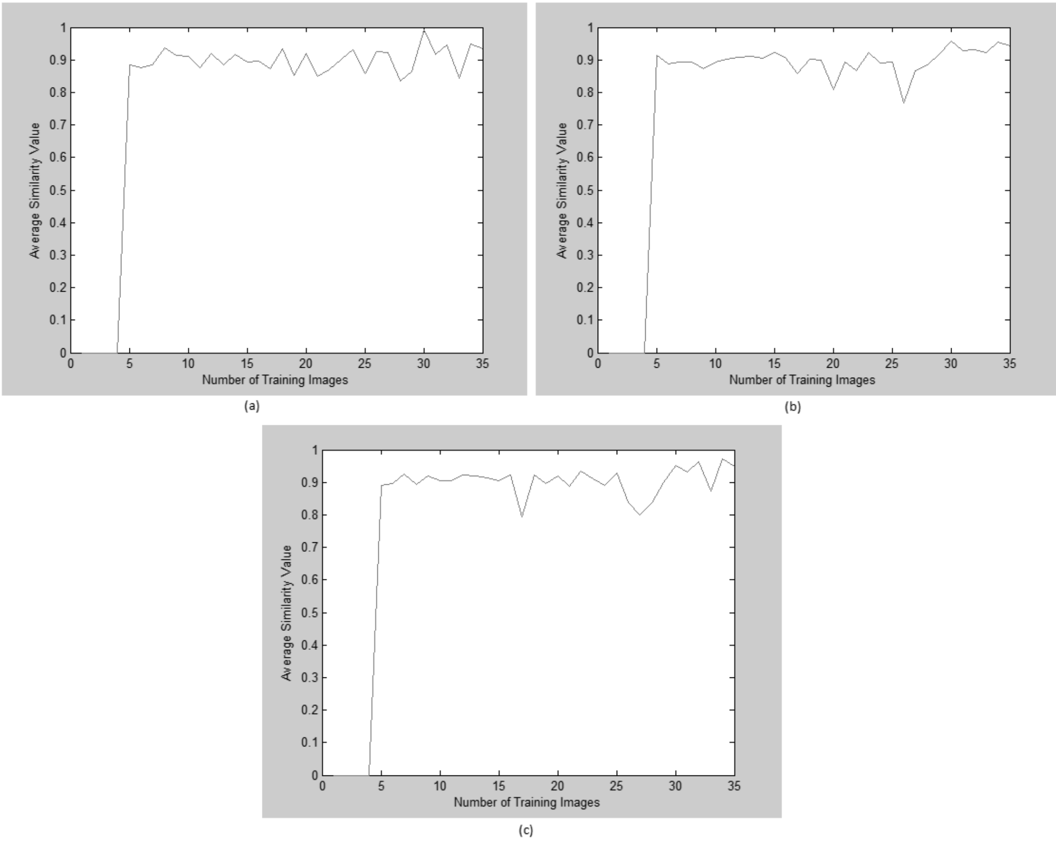


Figure 4.1: The number of images in the learning set and average similarity rate of ten test data. (a) 1^{st} Caricaturist's Dataset, (b) 2^{nd} Caricaturist's Dataset, (c) 3^{rd} Caricaturist's Dataset

Figure 4.1 shows average success values obtained separately for the three caricaturists, by calculating the average similarity value according to the number of training images. Values contained in this chart, enables to defining the number of image-caricature duo required for an optimum learning set to have a successful outcome (producing caricatures those have similar characteristics with a real caricature). When examining the graph, optimum number of data sets is shown as 30 for first two datasets and 34 for the last dataset. Also, the average success for the last dataset, 30 is the third highest value and very close to the first two values. Nevertheless, ACGen can give close to the highest similarity value with 7-8 images, even though the best similarity value is acquired by using 30 training images.

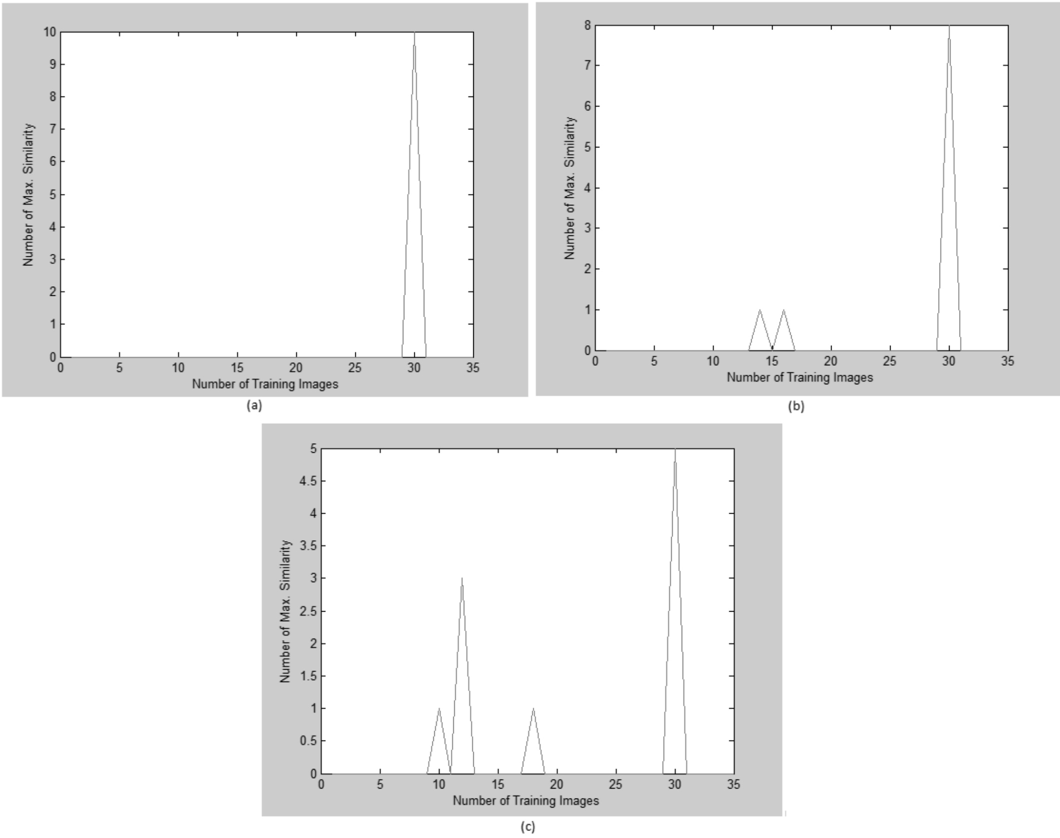


Figure 4.2: The number of images in the learning set and maximum similarity value for test data. (a) 1^{st} Caricaturist's Dataset, (b) 2^{nd} Caricaturist's Dataset, (c) 3^{rd} Caricaturist's Dataset

The graphs in Figure 4.2 shows the number of test data achieved maximum success, taken from the ten test data applied on each learning data set, calculated for three caricaturists. The data from these charts, shows the number of image-caricature duos required to create a learning set which results in maximum similarity values. If the graphs are examined, it is observed that the best results for first two datasets are obtained with 30 image-caricature pairs and the same number of pairs that will be effective for at least half of the last dataset. However, one can easily employ sample set of size 10-15 to obtain similar results.

The two experiments described above show that using 30 image-caricature pairs for the learning data set gives the satisfactory results for both the number of maximum similarities and the average similarities for almost all the caricaturists. Therefore, the experiments that have been conducted from this stage onwards are carried out by using 30 image-caricature pairs as the learning set.

Table 4.1: Success values in the Sample Test Image 1

	1st Caricaturist	**2nd Caricaturist**	**3rd Caricaturist**
Sample Test Image 1	0,984758349	0,903169998	0,917775398

Machine is learned the styles of the caricaturist by using proposed methodology. For this learning process a test has been performed. When the caricatures of test images drawn by the artist are compared with the five test outputs below, similarities between them, which will be proved arithmetically later, can be discerned.

When the caricatures created by the system are analyzed, caricatures of the same person are noted to be drawn by the three different styles matching three separate learning processes. This outcome certifies that this method not only generate caricatures automatically, but also create different caricatures depending on different caricaturist styles.

Although the similarity between the caricatures drawn by the artists and those generated by the system is satisfactory, during the visual inspection, arithmetical measure of this similarity is nevertheless described in the tables below.

4.4 Experiments and Results

The real image of the first test subject the caricatures based on the three caricaturists' styles generated by the system and the caricatures drawn by the artists are given in 4.3. A visual examination of the images and the corresponding caricatures reveals the following observations:

(a) It can be observed that the system is capable of learning the facial features of the caricatures which form the caricaturists' style to a visually acceptable extent. The chin, the cheek, the line between the hair and the forehead, the outer contour line of the hair in these system generated caricatures look very alike with the original caricatures.

(b) The features of the eyebrows (thickness, length, angle etc.), which were drawn rather linear in the original caricatures, were correctly discerned and reflected into the generated caricatures.

(c) The system made an almost one-to-one prediction of the structure of the mouth (as in

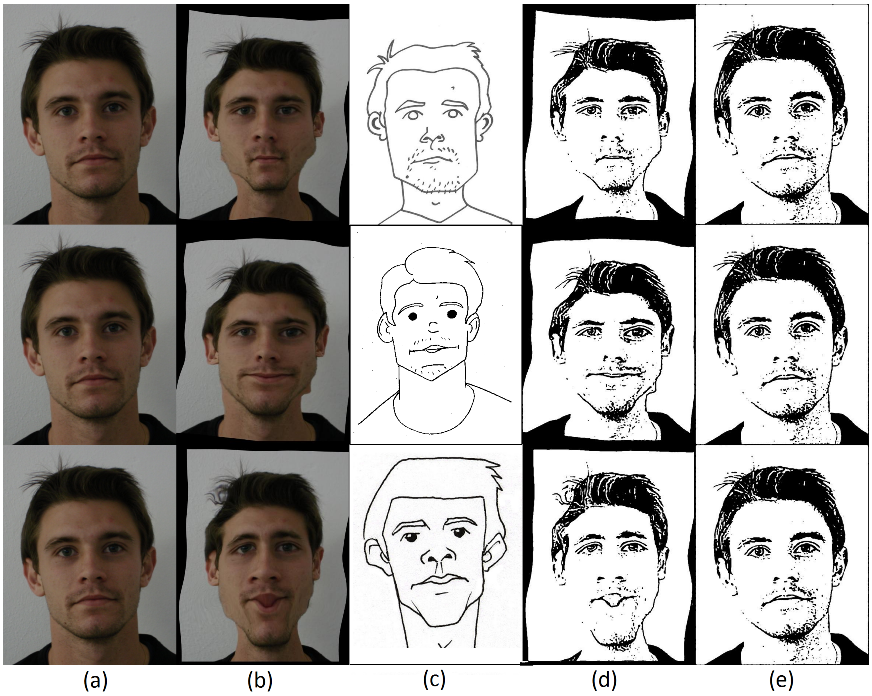


Figure 4.3: (a) Original Image, (b) Deformed Image, Learned By The System, Generated With Respect To The Caricaturist's Style, (c) Caricature Drawn By The Artist, (d) Caricature Created by the System, (e) Read Image After Edge Detection Applied First Row-1st Caricaturist Related Results, Second Row-2nd Caricaturist Related Results, Third Row-3rd Caricaturist Related Results

Table 4.2: Success values in the Sample Test Image 2

	1st Caricaturist	2nd Caricaturist	3rd Caricaturist
Sample Test Image 2	0,983334131	0,905398578	0,91457893

terms of width, length, position on the face etc.) for the caricatures. The second caricature has a widened thin mouth, while the third caricature has again widened but also a bit shrunk mouth. For the first caricature unlike the others, a narrow and very thin mouth was drawn. In addition to these, easily discerned details, positions and sizes of the nose and eye are very much akin to these features in the original caricatures.

The original image of the second test subject, the caricatures based on the three caricaturists' styles generated by the system and the caricatures drawn by the artists are given in 4.4. A visual examination of the images and the corresponding caricatures reveals the following observations:

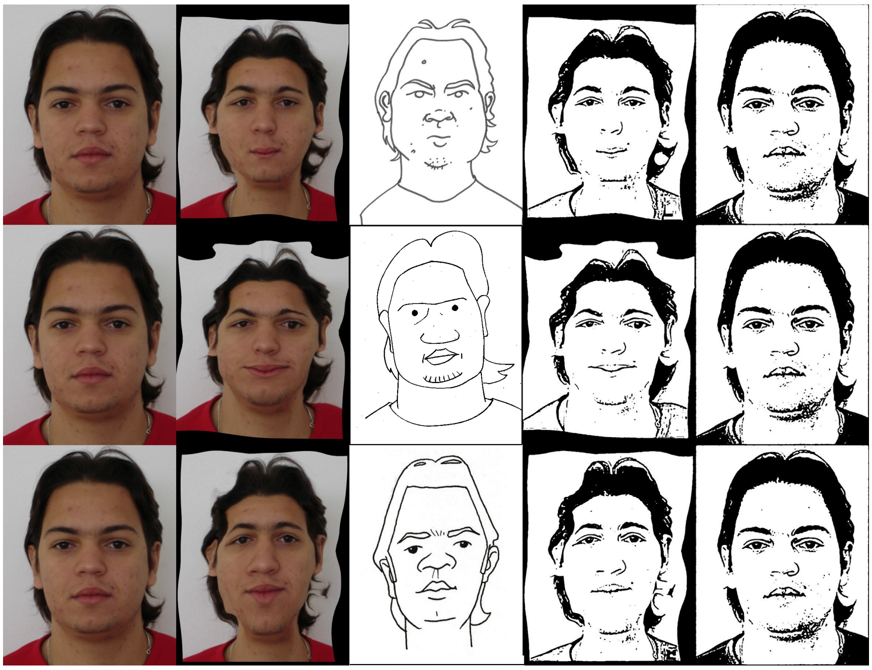


Figure 4.4: (a) Original Image, (b) Deformed Image, Learned By The System, Generated With Respect To The Caricaturist's Style, (c) Caricature Drawn By The Artist, (d) Caricature Created by the System, (e) Read Image After Edge Detection Applied First Row-1st Caricaturist Related Results, Second Row-2nd Caricaturist Related Results, Third Row-3rd Caricaturist Related Results

Table 4.3: Success values in the Sample Test Image 3

	1st Caricaturist	**2nd Caricaturist**	**3rd Caricaturist**
Sample Test Image 3	0,978961562	0,900143387	0,919655799

(a) It can be seen from the results that the system can successfully draw the mouth structure. In the second caricature, the mouth is a bit widened, the lips are thicker while in the third and the first one the mouth is drawn both narrow and thin.

(b) From the results, it can be perceived that the positions of the eyes in the generated caricatures are very similar to the positions of the eyes in the real caricatures. The left eye in the second caricature lean leftwards, while the eyes in the third caricature are more straight. However, in the first caricature, they are symmetrically aligned in the center of the face. These visual analyses indicate the success for predicting the eye positions.

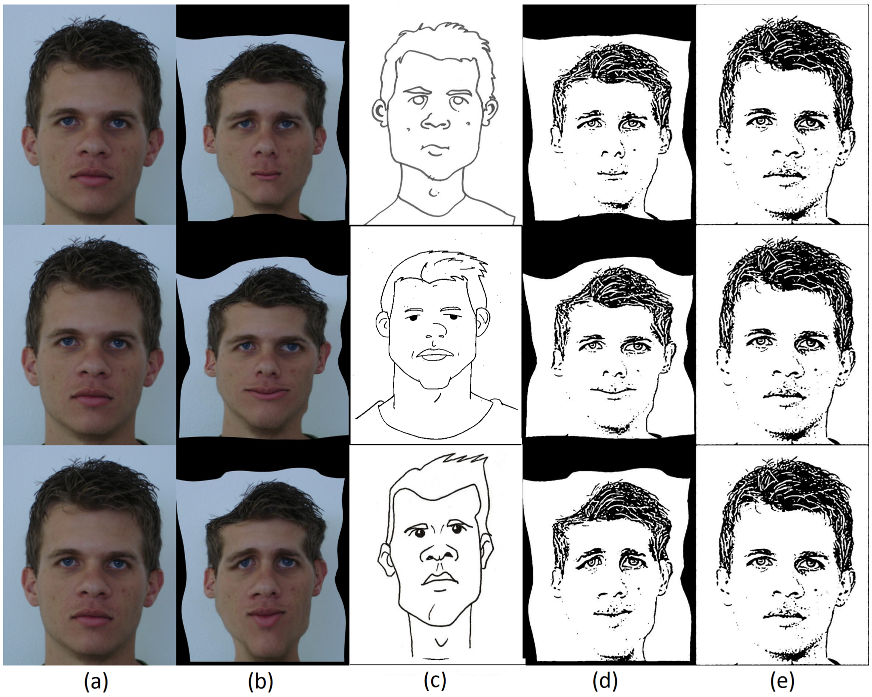


Figure 4.5: (a) Original Image, (b) Deformed Image, Learned By The System, Generated With Respect To The Caricaturist's Style, (c) Caricature Drawn By The Artist, (d) Caricature Created by the System, (e) Read Image After Edge Detection Applied First Row-1^{st} Caricaturist Related Results, Second Row-2^{nd} Caricaturist Related Results, Third Row-3^{rd} Caricaturist Related Results

The real image of the third test subject, the caricatures based on the three caricaturists' styles generated by the system and the caricatures drawn by the artists are given in 4.5. A visual examination of the images and the corresponding caricatures reveals the following observations:

(a) With this system, the hair structure, which is not normally taken into account in most face similarity algorithms, can also be learned with this system. The hair is shoved towards left in the second and third caricatures. However in the first, it is more tidily drawn. The same features can also be seen in the real caricatures.

(b) Here, the facial features have also successfully been mimicked. For the chin and jaw of the third caricature, sharp lines are used, in contrast to the rather soft transitions in the first caricature. For the second caricature, on the other hand, cheeks were emphasized and keen lines were used in chins.

(c) We should also give attention to the details of the eyes drawn by the system. All the three

Table 4.4: Success values in the Sample Test Image 4

	1st Caricaturist	2nd Caricaturist	3rd Caricaturist
Sample Test Image 4	0,984636837	0,905613737	0,918569934

caricatures offer distinct representation for a rather standard eye structure that test image has while the second caricature has aligned eyes, the third one has a left eye that is warped downward. Finally, the first one has a right eye slightly pulled upward. All these details match the original changes made by the three caricaturists in the original caricatures.

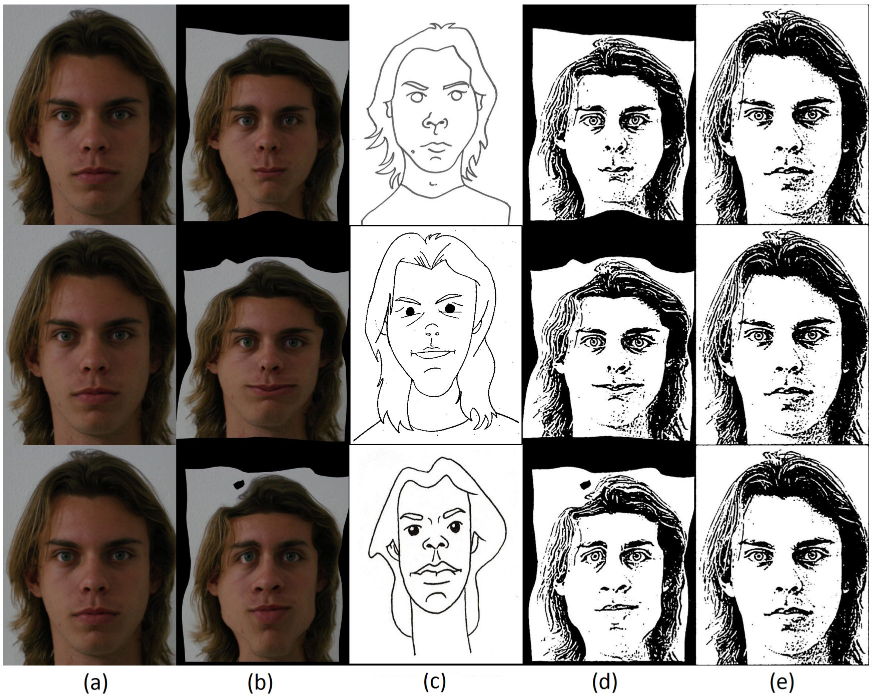


Figure 4.6: (a) Original Image, (b) Deformed Image, Learned By The System, Generated With Respect To The Caricaturist's Style, (c) Caricature Drawn By The Artist, (d) Caricature Created by the System, (e) Read Image After Edge Detection Applied First Row-1st Caricaturist Related Results, Second Row-2nd Caricaturist Related Results, Third Row-3rd Caricaturist Related Results

The real image of the fourth test subject, the caricatures based on the three caricaturists' styles generated by the system and the caricatures drawn by the artists are given in 4.6. A visual examination of the images and the corresponding caricatures reveals the following observations:

(a) The system suggested in this study also makes successful predictions about the nose shape. In third caricature, a distinct and flat nose drawn, instead of the smaller ones drawn in other two caricatures. At the same time, noses in first and second caricatures do not resemble each other at all. First caricature has a rather symmetrical nose structure; on the contrary, second caricature has a nose slightly pulled right. Parallel inclinations can be noted in the positions of the nostrils in original drawings.

(b) The width of the forehead is similarly drawn in both generated and real caricatures. Forehead in the third caricature is narrow, though the foreheads in other two caricatures are wider. Fore head is wider both horizontally and vertically in the first caricature but it is horizontally narrower in the second. Again, these results are parallel to the forehead features of the real caricatures.

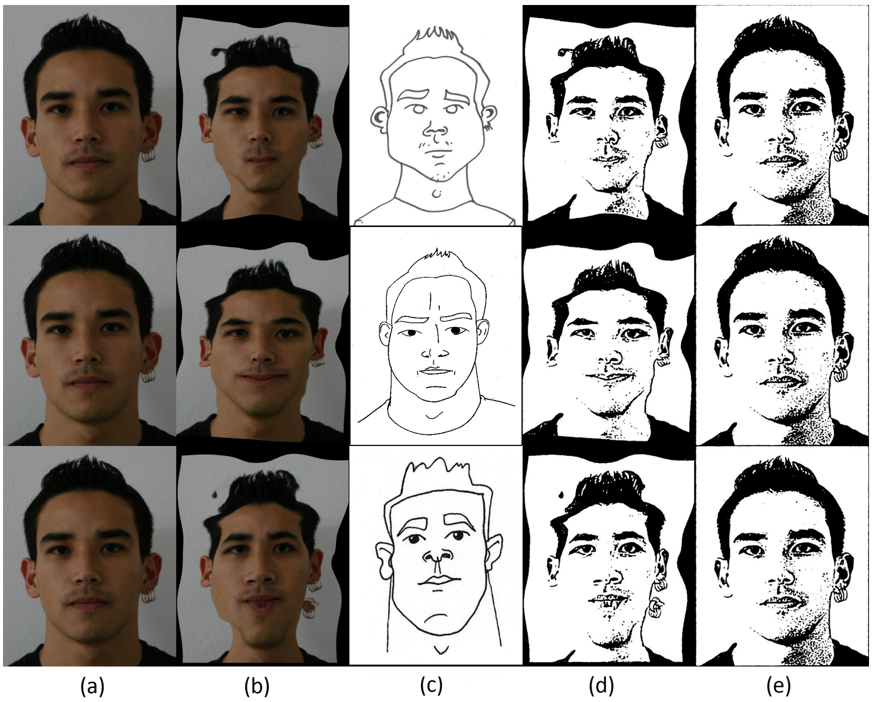


Figure 4.7: (a) Original Image, (b) Deformed Image, Learned By The System, Generated With Respect To The Caricaturist's Style, (c) Caricature Drawn By The Artist, (d) Caricature Created by the System, (e) Read Image After Edge Detection Applied First Row-1st Caricaturist Related Results, Second Row-2nd Caricaturist Related Results, Third Row-3rd Caricaturist Related Results

The real image of the fifth test subject, the caricatures based on the three caricaturists' styles generated by the system and the caricatures drawn by the artists are given in 4.7. A visual examination of the images and the corresponding caricatures reveals the following observa-

Table 4.5: Success values in the Sample Test Image 5

	1st Caricaturist	2nd Caricaturist	3rd Caricaturist
Sample Test Image 4	0,973584706	0,903713597	0,906799927

tions:

(a) Similarities or differences of details on the hair features are eye catching in these examples. Especially, the details in drawings could easily be distinguished as further proofs of system's success. Hair is combed up to a point in the second caricature, contrary to the hair in third caricature that is spread to the sides. Hair in the first caricature is similar to the second one but a bit messy, also shows real similarity to the hair in real caricature.

(b) Features of the mouth and noses present obvious similarities to the features in real caricatures. Also, in addition to the similarities, the gap between the nose and the lips can be estimated successfully. This gap is wide in first caricature, tight in second caricature and although it is tight as in second caricature, in third caricature thicker lips caused this rather than upper mouth position. Real caricatures have exact same features explained above.

Results indicate that the suggested ACGen method is very successful at grasping the details of the style of a caricaturist. The details about eyes, brows, mouth, hair may very well be based on the distance between them or positions in face as well as their shapes. Although each of the listed details proves to be exclusively important in creating caricatures; they only generate plausible results when used together. At this point, it is important to look at the big picture as a whole without getting lost on details. With this approach, similarity between the generated caricature and the real caricature sinks in. Especially, it is vital to feel the same about the model's look and mood while looking at the real caricature and generated one. This is directly affected by the look, facial features and jawbones of the face caricature. This resemblance can be felt if the caricatures compared in terms of this approach. Nevertheless, because it is impossible to show this feeling of resemblance in numbers, a similarity metric can be used to give an arithmetical proof. Then again, which comparisons do not only depends on inner-face-features, but also takes facial frame into account in a more out of the box style. And these results are given above in terms of similarity percentages which are calculated by the similarity function mentioned in Chapter 3. Just speaking on these data, results can be labeled to be very successful.

Along with the detailed examples, first caricaturist is chosen in order to show that all test results are successful. 15 real images, their system generated counterparts created after caricaturist's style is learned and the caricatures drawn by the artist are in Appendix A. The similarity percentages are also in Appendix B.

Along with the similarity percentages residing in Appendix B, similarities of the caricatures generated by using styles of different caricaturists for the same test image are shown in Appendix D. Although drawn by the same system for the same images, these caricatures proved to be not similar in terms of these values.

The reason why the Similarity Metric value is chosen as L2 norm in ACGen is explained with this study also. As an alternative to the L2 norm value used in similarity calculation, Lp norm and cosine values are picked. Lp norm and cosine values are used separately for similarity calculation and a sample deformed image is generated. As the image in Appendix E also suggests, using Lp norm or cosine values did not give any better results than using similarity metric of ACGen.

Notice that the suggested method successfully captures the style of the artists. The suggested method not only captures the main style but also the mimics of the real images such as bored, tired etc. In another words, the system also mimics caricaturists' talent of animating emotions.

CHAPTER 5

CONCLUSIONS AND FUTURE DIRECTIONS

This study introduces a new model for learning the drawing style of a caricaturist and generating caricatures that carries the caricaturist's key characteristics used in his drawing. In the suggested model, input caricature is compared to a set of 30 real face images. Next, similarity rates of test face and each training set images are calculated. The style of an artist is modelled by a feature vector, called exaggeration amounts The exaggeration amounts are computed for the new input image, for caricature generation. Rather than using a threshold for the rates and picking samples similar to the input image, every image and their corresponding caricatures in training set affect the result caricature. The caricaturizing techniques of the caricaturist applied on a test image computed with similarity rates of the image and exaggeration amounts of all the training caricatures. Only by warping the input image based on these caricaturizing techniques lets the system to generate a caricature that is very similar to the caricaturist's original sketch.

The approaches suggested in many studies [2], [3], [5], [6], [7] studies made on this thesis's topic leave the measurement of the success to the hands of user comments and do not present any mathematical evaluation. As a result, it is not possible to compare these previous studies with this study in terms of numbers. Yet, the success of our study is measured by a similarity calculation method, even that it is still premature and under development. Moreover, even if a survey is not made for measuring the success, the results are demonstrated to members of METU Computer Engineering Department ImageLab group and they are approved to be successful. There have been comments about the ACGen results that they contain the characteristics of the real caricatures.

As can be observed by the detailed analysis in Experiments chapter, ACGen is a robust system.

For illustration, even the number of images in learning set is reduced down to 7 or 8 images, the system works fairly well. Furthermore, this system can generate the caricature for an image, even if it is not much similar to the images in the training dataset, and make the caricature look like as if drawn by the caricaturist.

AcGen contains consecutive 5 or 6 methods those proceed serially depending on the end of preceding process. In other words, ACgen cannot operate in a parallel fashion. Yet, even with this setting, ACGen generates caricatures in terms of milliseconds. Besides, the highest time consumed in this method application is in the step that includes saving the generated image on the storage disk. Because of that, ACGen gives results in far less time if employed on a pc with solid state disk.

In the proposed model, initially facial features are labelled by hand. This process is the most time consuming part while the other parts execute in milliseconds. Therefore, first and supplementary future work will be directed towards automatic labelling of the images in order to have a fast and self sufficient system working just with training set and test images. As a result, our system will need minimum user interaction in caricature generation process.

As mentioned in Chapter 1, face recognition algorithms applied on generated caricatures will probably give better results than they are applied on real images. Accordingly, another future work will be the application of the face recognition algorithms or detecting facial expressions on generated caricatures.

Moreover, if the caricaturist has a style, the computer is capable of learning it. A new system, that enables identifying whether a caricature belongs to a certain artist or not, will be developed. Even more than that, ACGen can act as a truly unbiased critic that can grade caricaturists depending on their consistency and professionalism in their art.

Besides, as mentioned in Introduction, since the system is developed from an object independent approach, it is possible for the system to learn and imitate the style of a certain artist, independent of art branch(drawing on sketches, nature, still life etc.) he/she specializes on. For this future work, only the dataset corresponding to this object should be prepared. ACGen is available to work on this object as it is.

REFERENCES

[1] S. Brennan, ”Caricature generator”, Master’s thesis, Cambridge, MIT, 1982.

[2] Rupesh N. Shet, Ka H. Lai, Eran A. Edirisinghe, Paul W. H. Chung, ”Use of Neural Networks in Automatic Caricature Generation: An Approach Based on Drawing Style Capture”, In Proceedings of IbPRIA (2) ’2005. pp.343 351

[3] H. Chen, Y. Xu, H. Shum, S. Zhu, and N. Zheng, ”Example based facial sketch generation with non-parametric sampling”, In ICCV01, pages II: 433-438, 2001.

[4] E. Akleman, ”Making caricature with morphing”, In Visual Proceedings of ACM SIGGRAPH’97, page 145, 1997.

[5] Liu J., Chen Y., Gao W., ”Mapping Learning in Eigenspace for Harmonious Caricature Generation”, ACM Multimedia, pp 683-686, 2006.

[6] Pei-Ying Chiang, Wen-Hung Liao, Tsai-Yen Li, ”Automatic Caricature Generation by Analyzing Facial Features”, 2004 Asian Conference on Computer Vision, Jeju Island, Korea, Jan 27-30,2004.

[7] L. Liang, H. Chen, Y-Q Xu, H-Y Shum, ”Example- Based Caricature Generation with Exaggeration”, Proceedings of 10th Pacific Conference on Computer Graphics and Applications, 2002.

[8] M. Tominaga, S. Fukuoka, K. Murakami, and H. Koshimizu., ”Facial caricaturing with motion caricaturing in picasso system”, In IEEE/ASME International Conference on Advanced Intelligent Mechatronics, page 30, 1997.

[9] Lu, X. ; Wang, Y. & Jain, A.K. (2003), ”Combining Classifiers for Face Recognition”, In IEEE Conference on Multimedia & Expo, Vol. 3, pp. 13-16

[10] Artificial Intelligence Laboratory of FEI in São Bernardo do Campo, São Paulo, Brazil, ”FEI Face Database”, June 2005 and March 2006

[11] T.M. Mitchell, Machine Learning. McGraw-Hill 1997

[12] Yeon Geol Ryu, Hyun Chul Roh, and Myung Jin Chung., ”Long-time video stabilization using point-feature trajectory smoothing”, In Consumer Electronics (ICCE), 2011 IEEE International Conference on, pages 189-190,2011.

[13] Y.K. Singh, ”Multi-level edge detectors based on the convolution matrices of base lengths 2 and 3”, ARPN Journal of Engineering and Applied Sciences. 6(1): 29-37, 2011.

[14] Y.K. Singh, ”Informative Binarization Based on Unsharp Masking”, ARPN Journal of Engineering and Applied Sciences. 6(1): 29-37, 2011.

[15] A. Ozmen,E. Akman, ”Edge Detection Using Steerable Filters and CNN”, XI European Signal Processing Conference (EUSIPCO), 2002.

Appendix A

Extra Sample Results

Figure A.1: First Caricaturist Related Results. First Row- Real Image, Second Row - Deformed Image, Learned By The System, Generated With Respect To The Caricaturist's Style, Third Row - Caricature Drawn By The Artist, Fourth Row - Caricature Created by the System, Fifth Row - Read Image After Edge Detection Applied

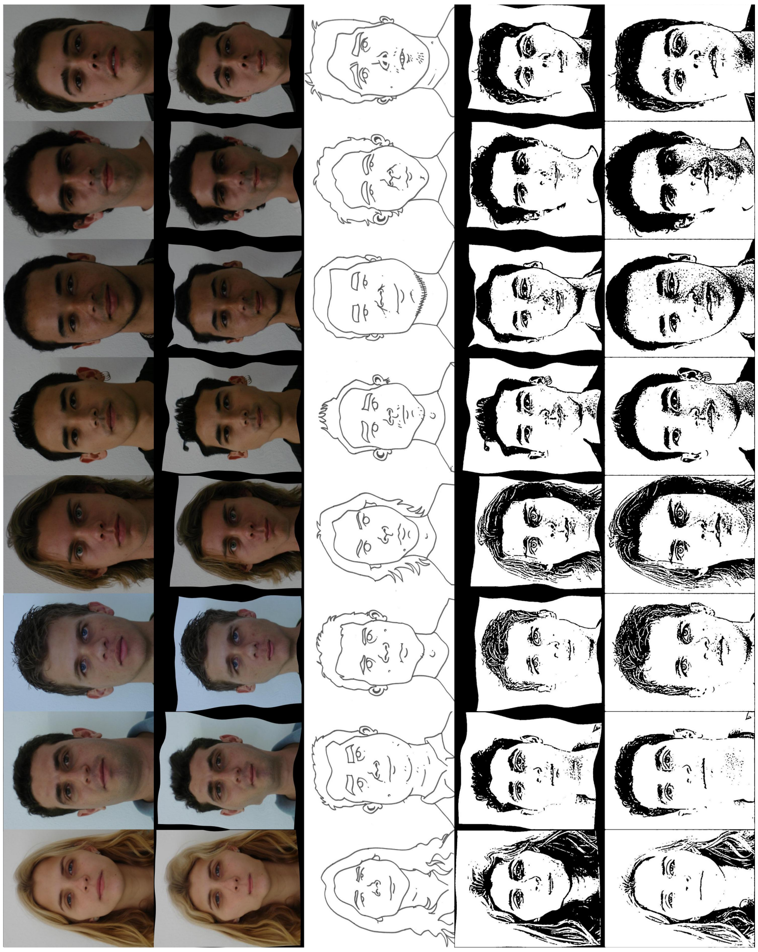


Figure A.2: First Caricaturist Related Results. First Row- Real Image, Second Row - Deformed Image, Learned By The System, Generated With Respect To The Caricaturist's Style, Third Row - Caricature Drawn By The Artist, Fourth Row - Caricature Created by the System, Fifth Row - Read Image After Edge Detection Applied

Appendix B

Success Achieved in the Experiments

Table B.1: Success achieved in Each Caricature Datase with ACGen Similarity Metric

	1st Caricaturist	**2nd Caricaturist**	**3rd Caricaturist**
Test Image 1	0,984758349	0,903169998	0,917775398
Test Image 2	0,989386395	0,899248522	0,917700107
Test Image 3	0,992569013	0,905427771	0,904821691
Test Image 4	0,985856294	0,905069946	0,906267762
Test Image 5	0,984021677	0,903891696	0,914938897
Test Image 6	0,983334131	0,905398578	0,914578933
Test Image 7	0,982222400	0,907610949	0,922716559
Test Image 8	0,987254825	0,905052262	0,915092213
Test Image 9	0,958288653	0,903963379	0,919384873
Test Image 10	0,978961562	0,900143387	0,919655799
Test Image 11	0,984636837	0,905613737	0,918569934
Test Image 12	0,973584706	0,903713597	0,906799927
Test Image 13	0,968927387	0,909458885	0,918590027
Test Image 14	0,990878506	0,903394803	0,914775633
Test Image 15	0,986631416	0,905174861	0,916146266

Table B.2: Success achieved in Each Caricature Dataset with Cosine Value Metric

	1st Caricaturist	**2nd Caricaturist**	**3rd Caricaturist**
Test Image 1	0,587639279	0,456426433	0,433125229
Test Image 2	0,592093375	0,442490449	0,424529497
Test Image 3	0,592360098	0,437972053	0,429883491
Test Image 4	0,584024716	0,493563702	0,434559680
Test Image 5	0,570288880	0,439750403	0,434171026
Test Image 6	0,588137093	0,500583886	0,436455481
Test Image 7	0,577748435	0,454099345	0,429508227
Test Image 8	0,578361197	0,479268300	0,432517607
Test Image 9	0,550232709	0,453923935	0,435394579
Test Image 10	0,578026417	0,435926559	0,426728281
Test Image 11	0,584995264	0,442542603	0,430205657
Test Image 12	0,572296162	0,439685701	0,430593684
Test Image 13	0,567034977	0,455750031	0,431068917
Test Image 14	0,592057167	0,439540668	0,431464005
Test Image 15	0,590595707	0,443176477	0,435702140

Appendix C

Original Image Dataset and 3 Different Caricature Datasets

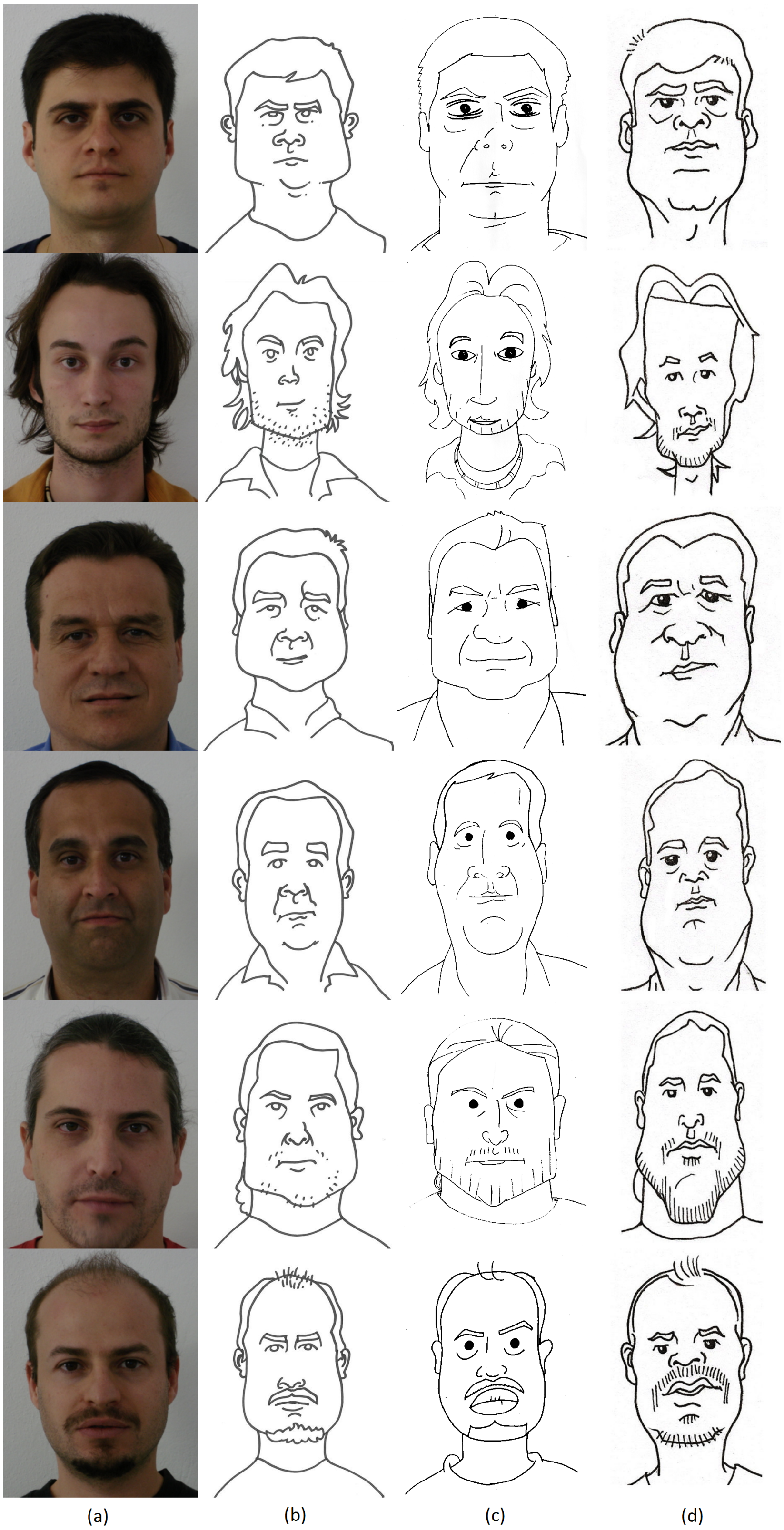


Figure C.1: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

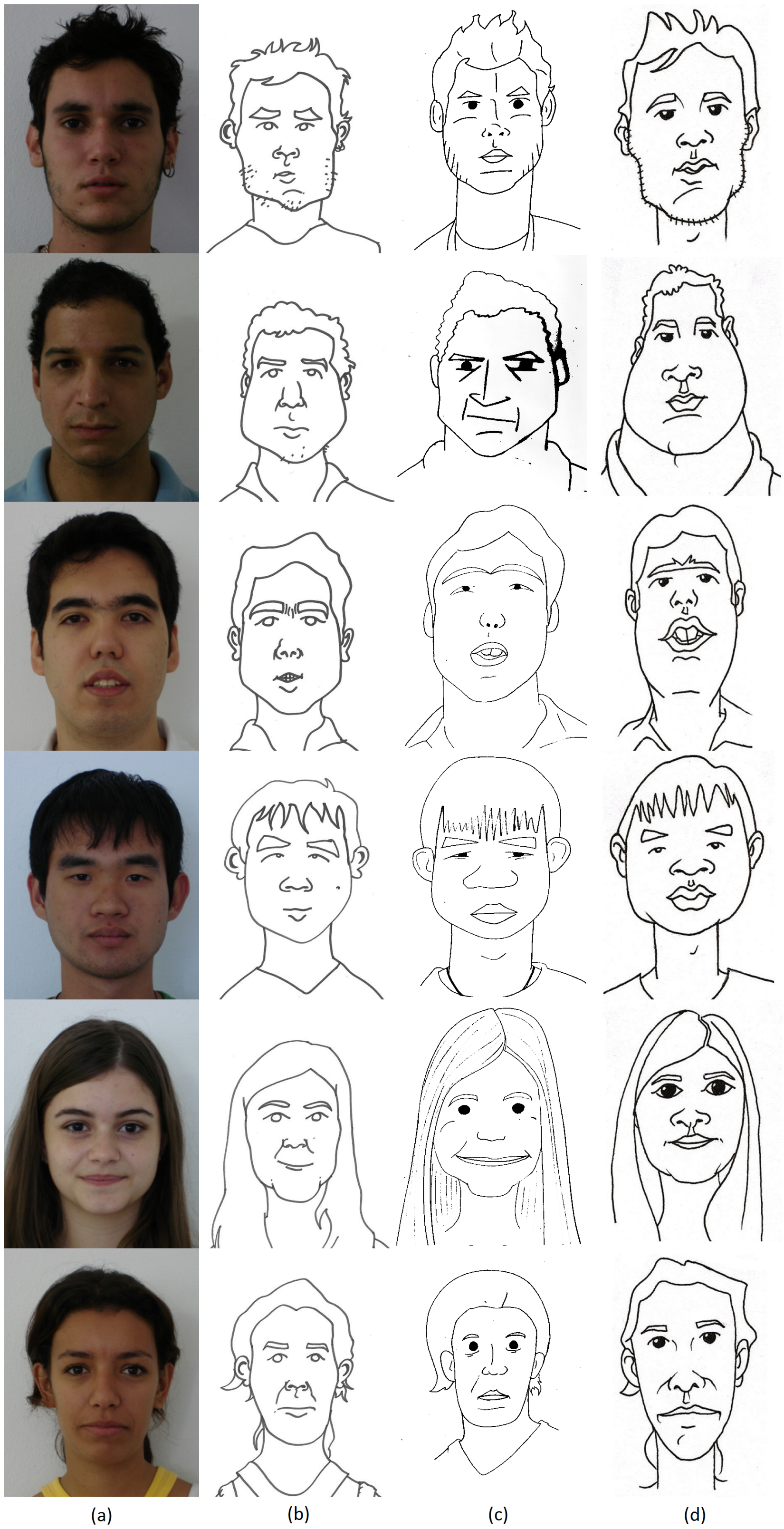


Figure C.2: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

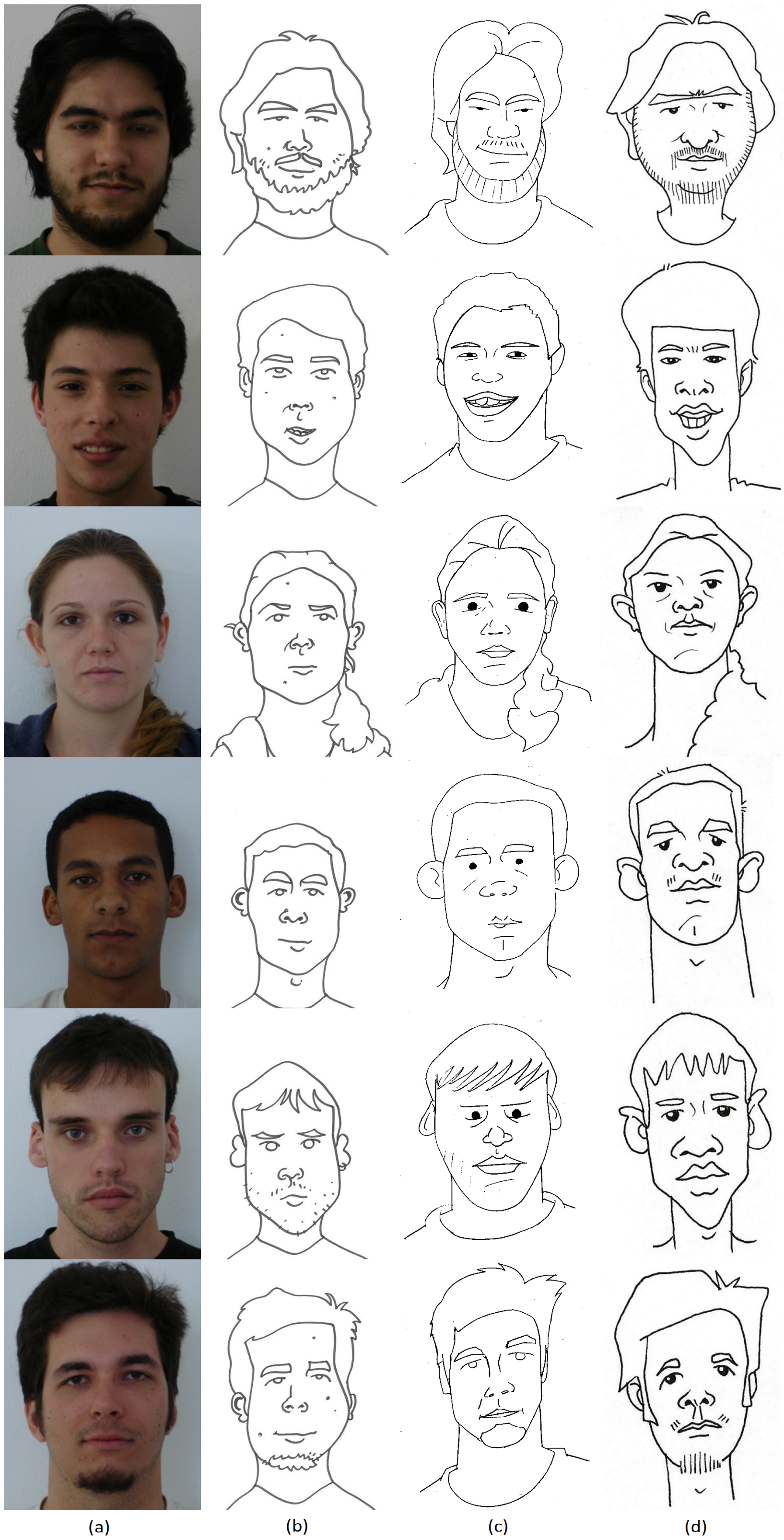


Figure C.3: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

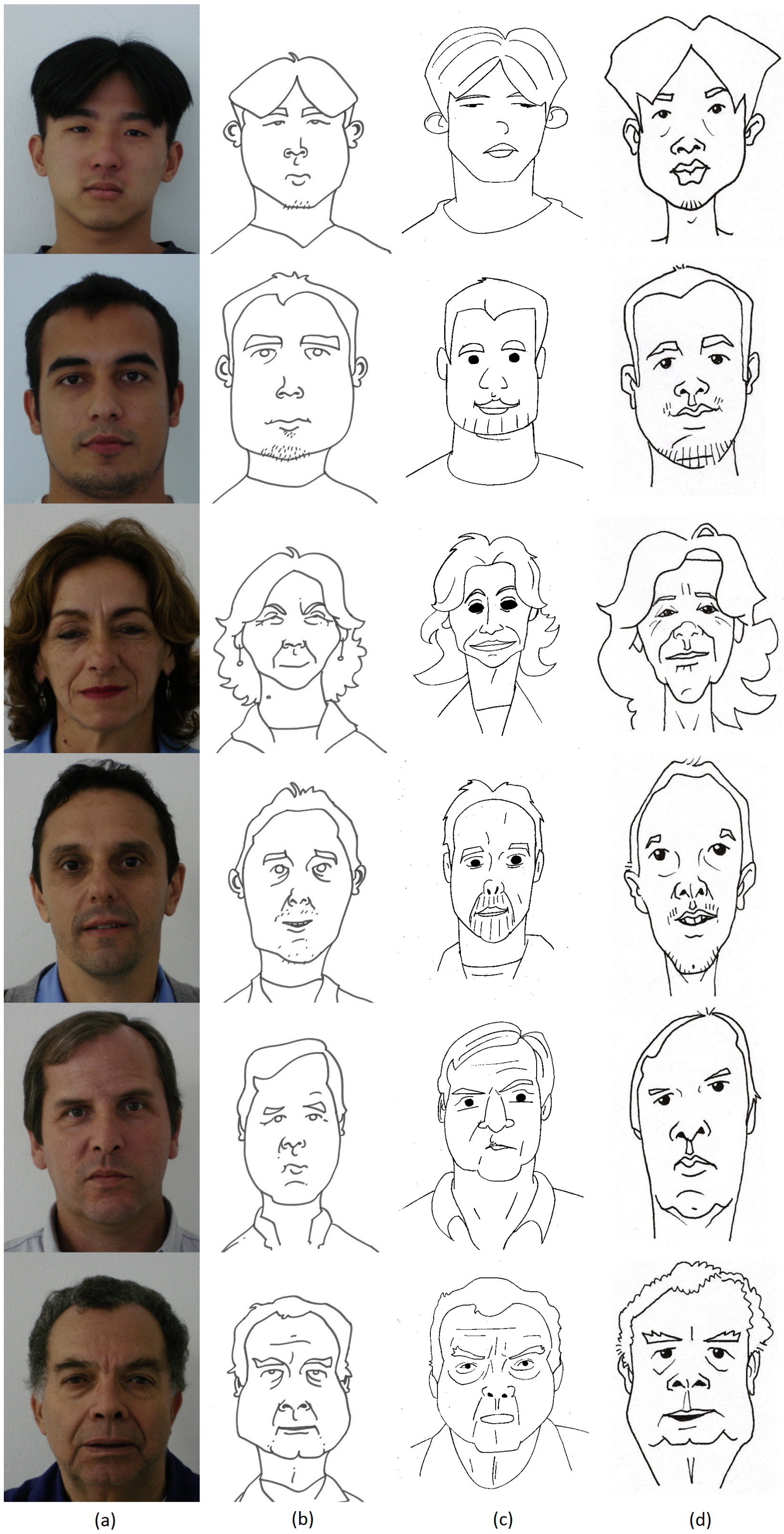


Figure C.4: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

Figure C.5: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

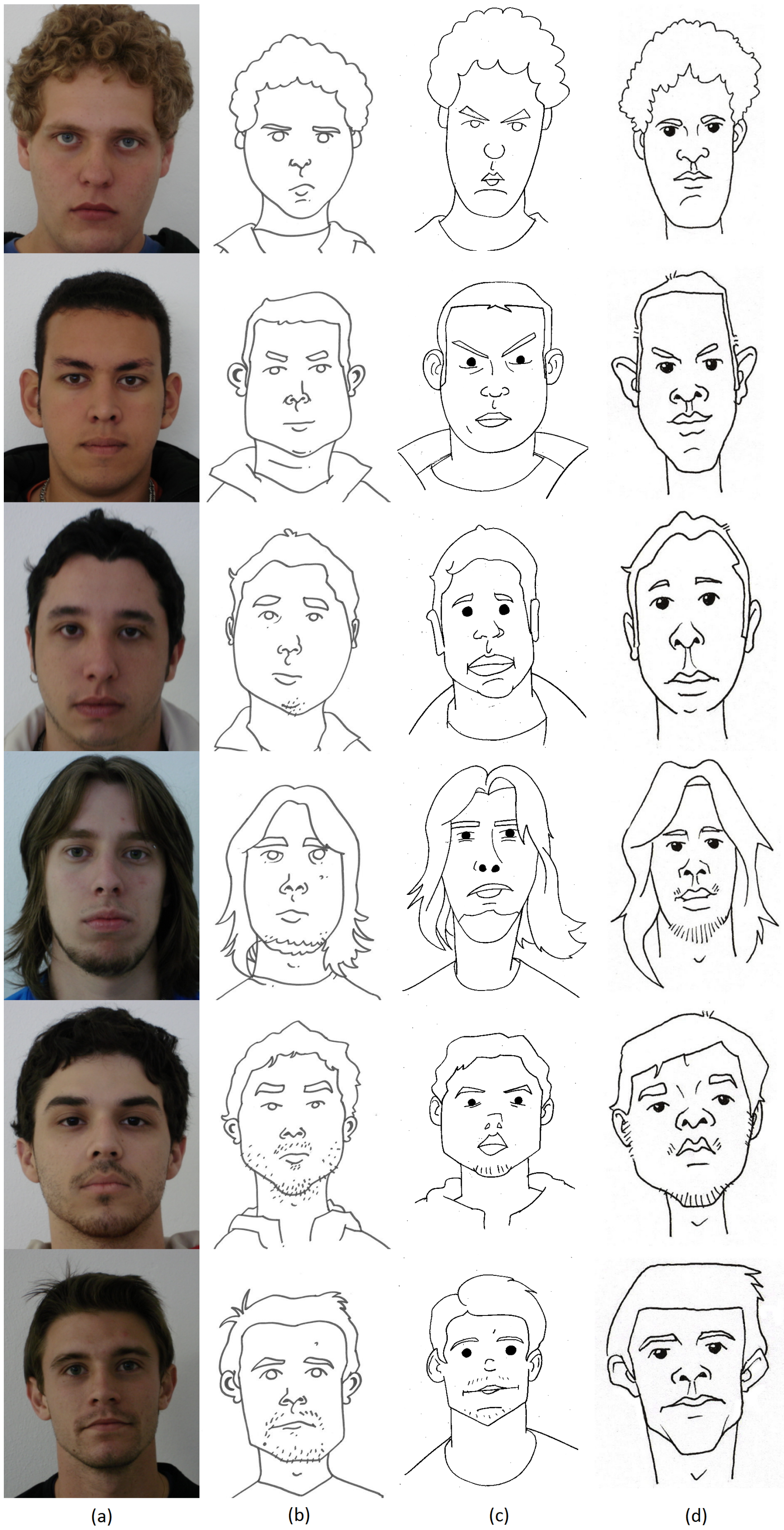


Figure C.6: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

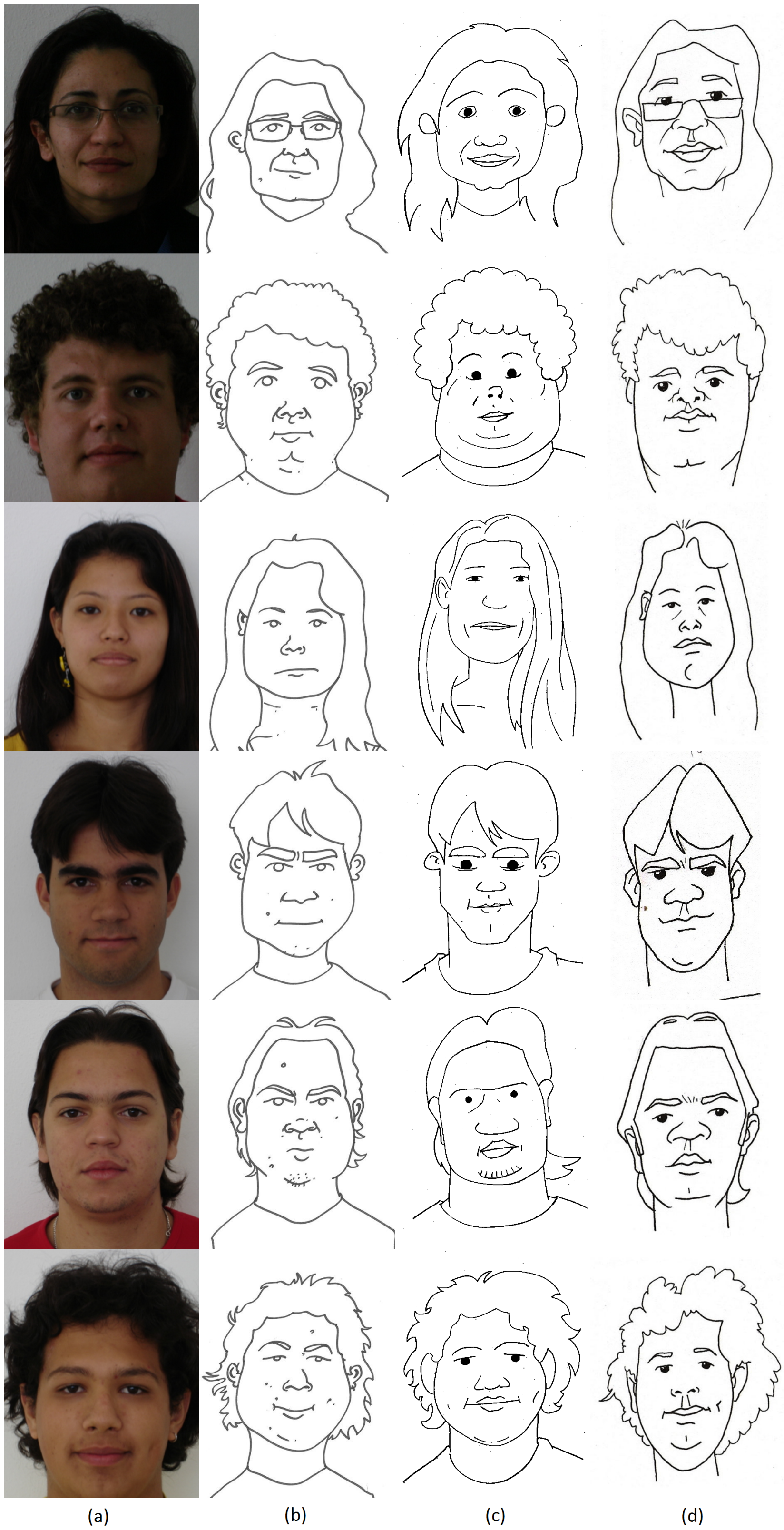
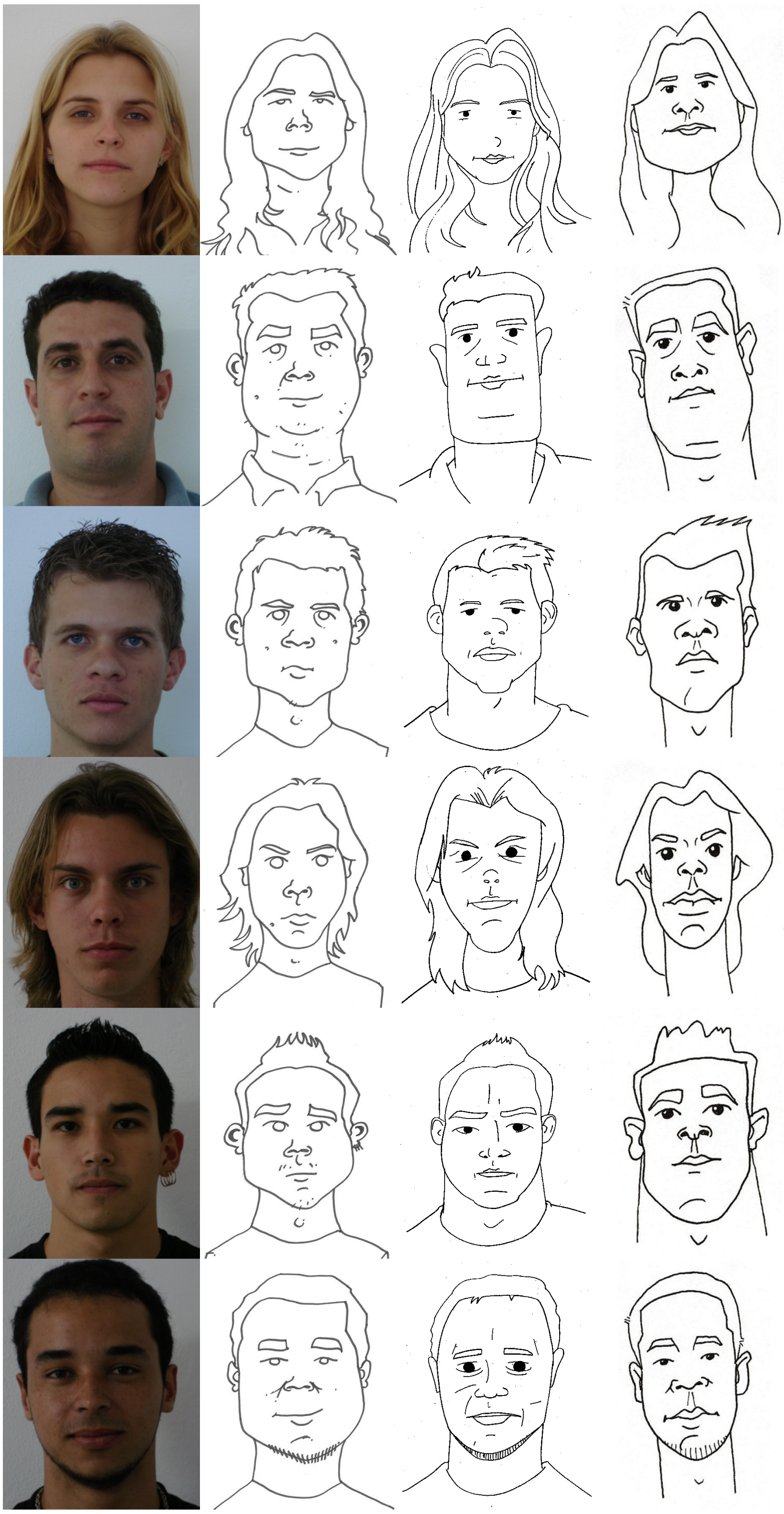


Figure C.7: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)



(a) (b) (c) (d)

Figure C.8: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

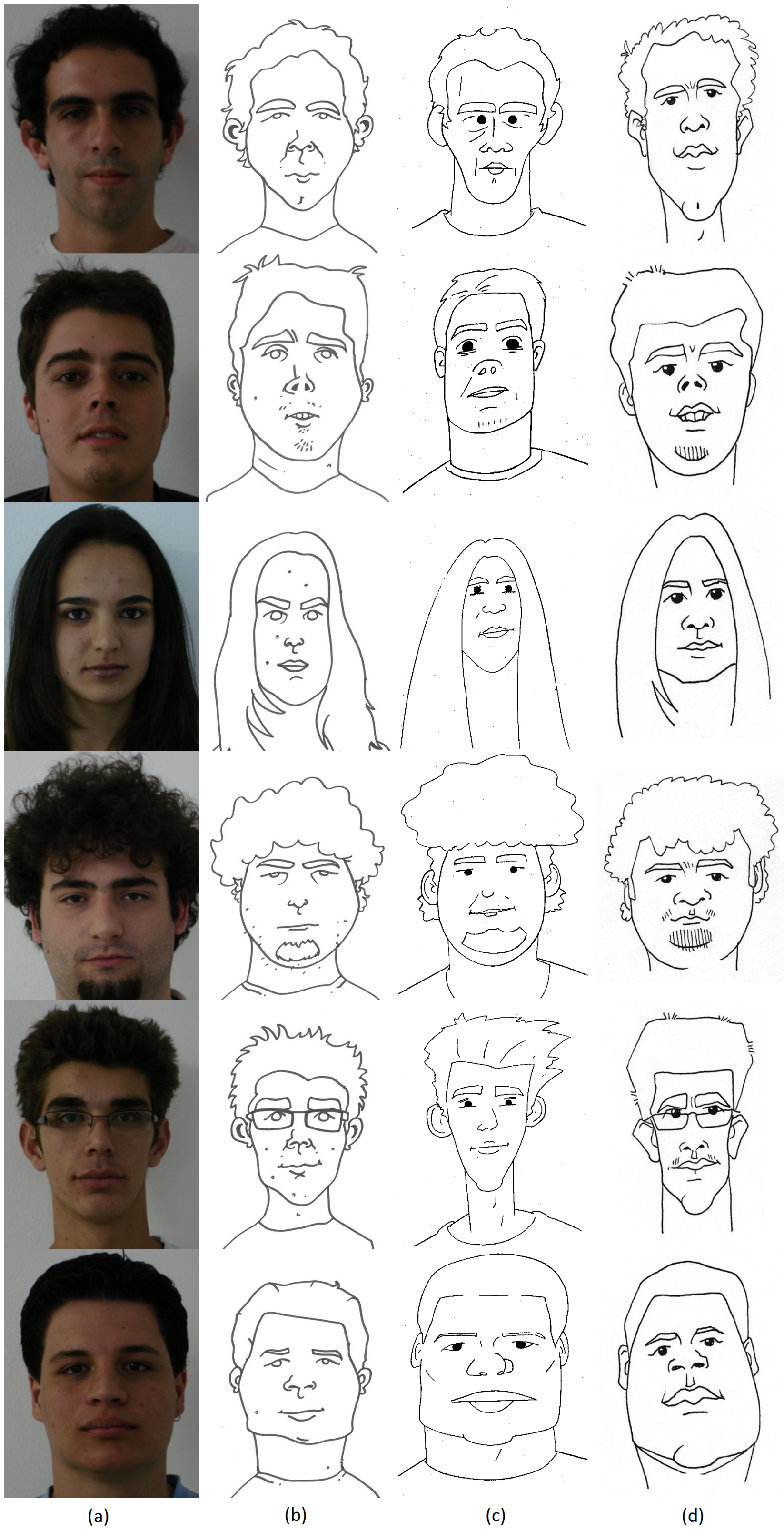
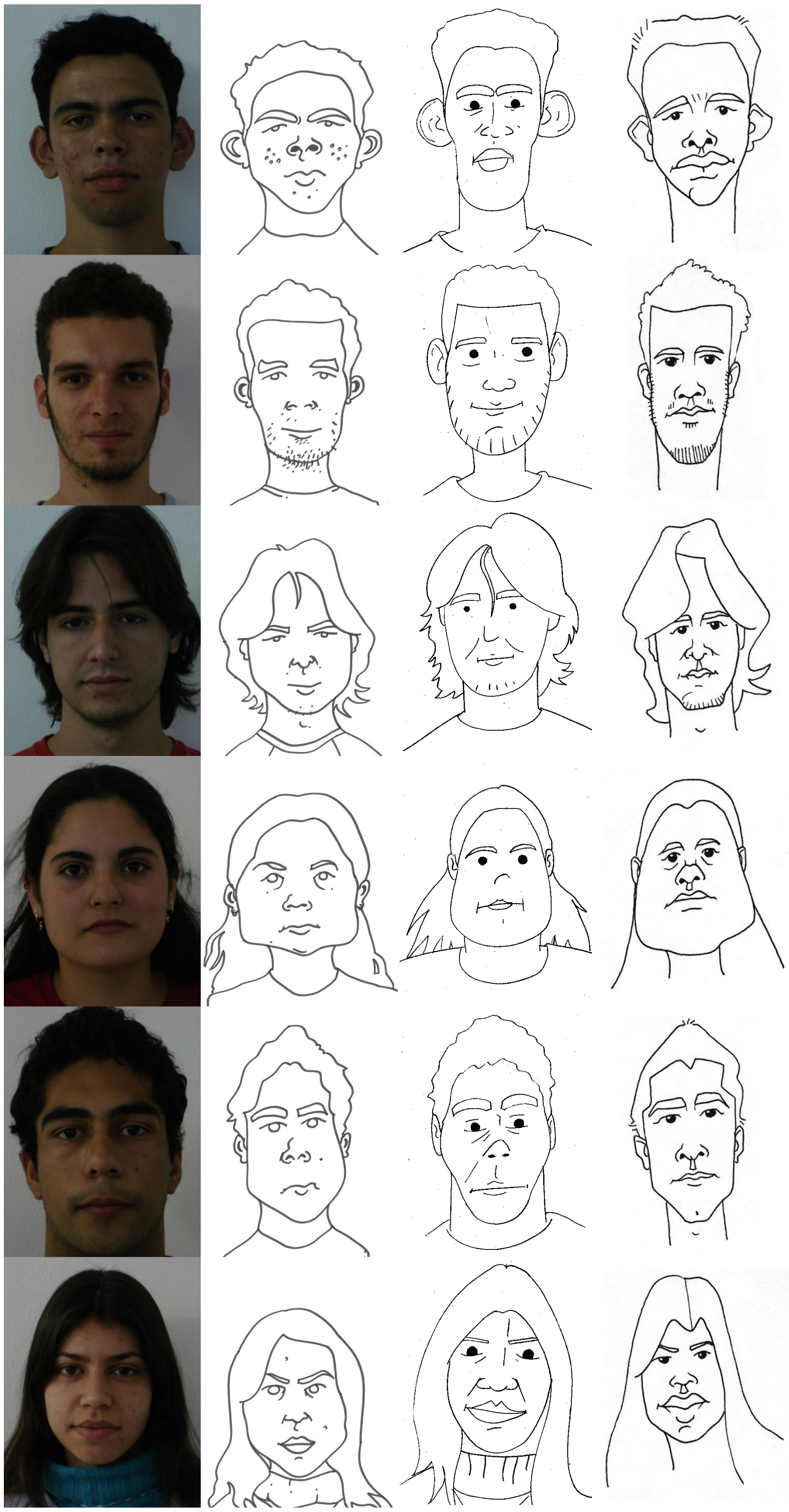


Figure C.9: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)



(a) (b) (c) (d)

Figure C.10: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

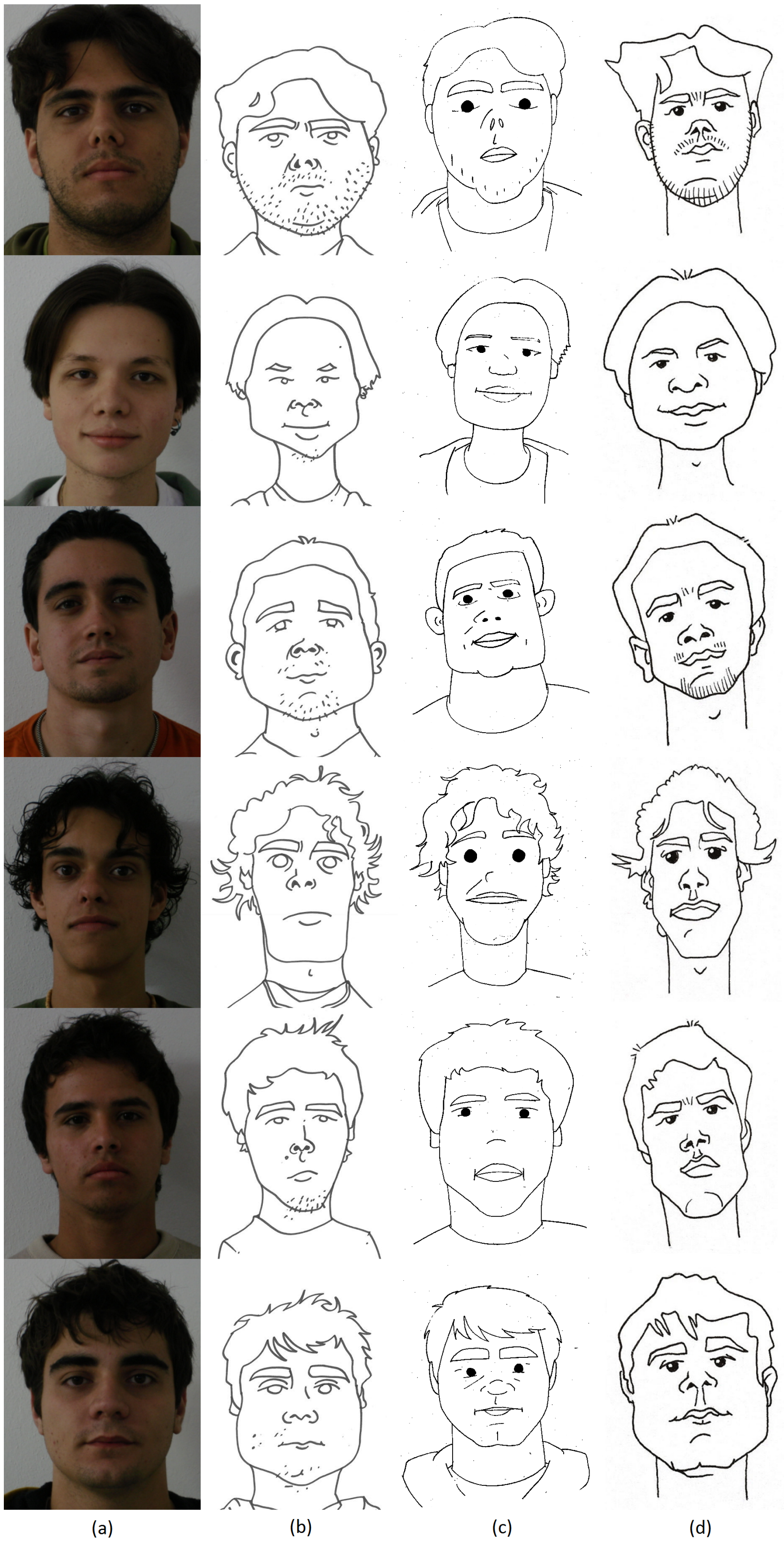


Figure C.11: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

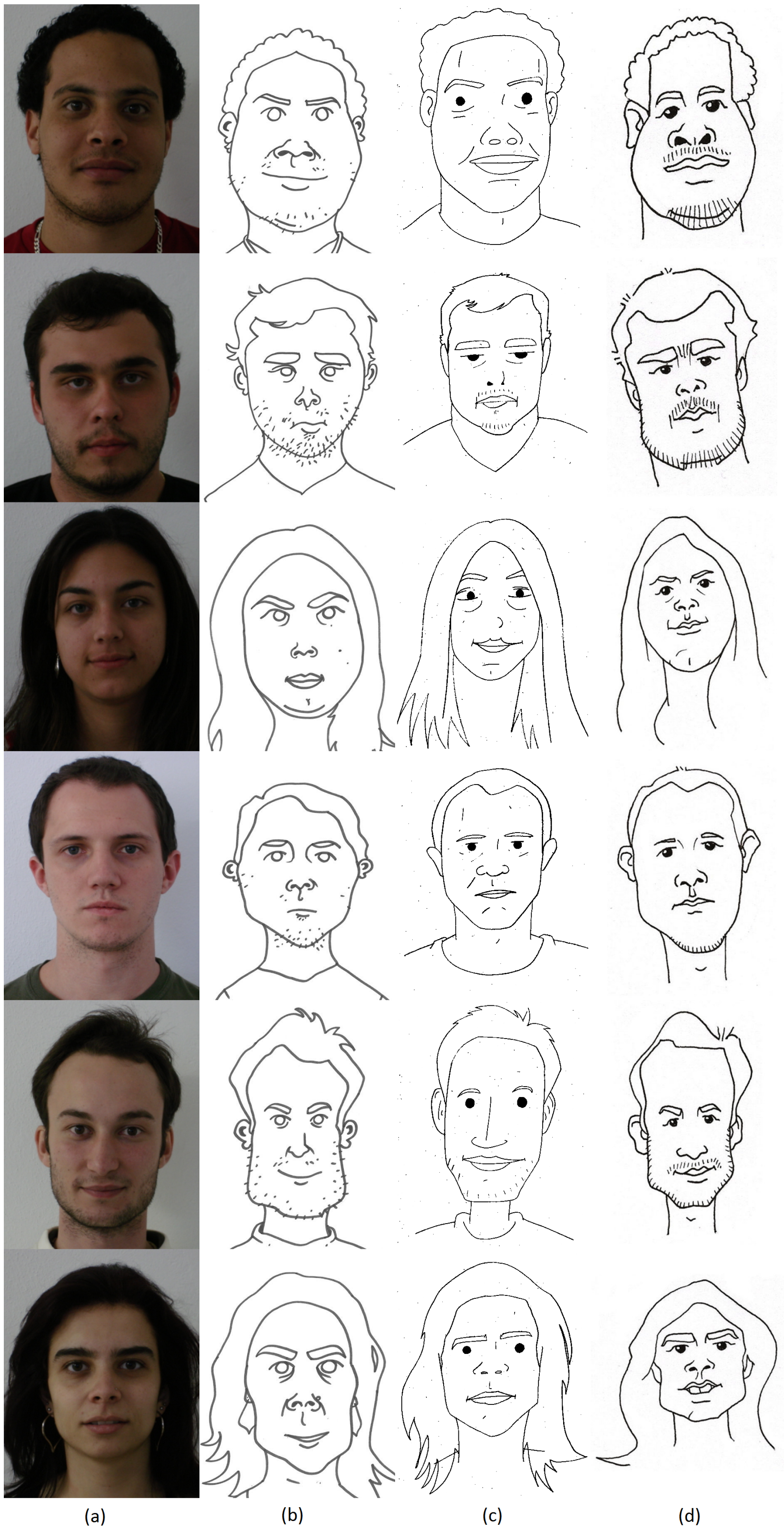


Figure C.12: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

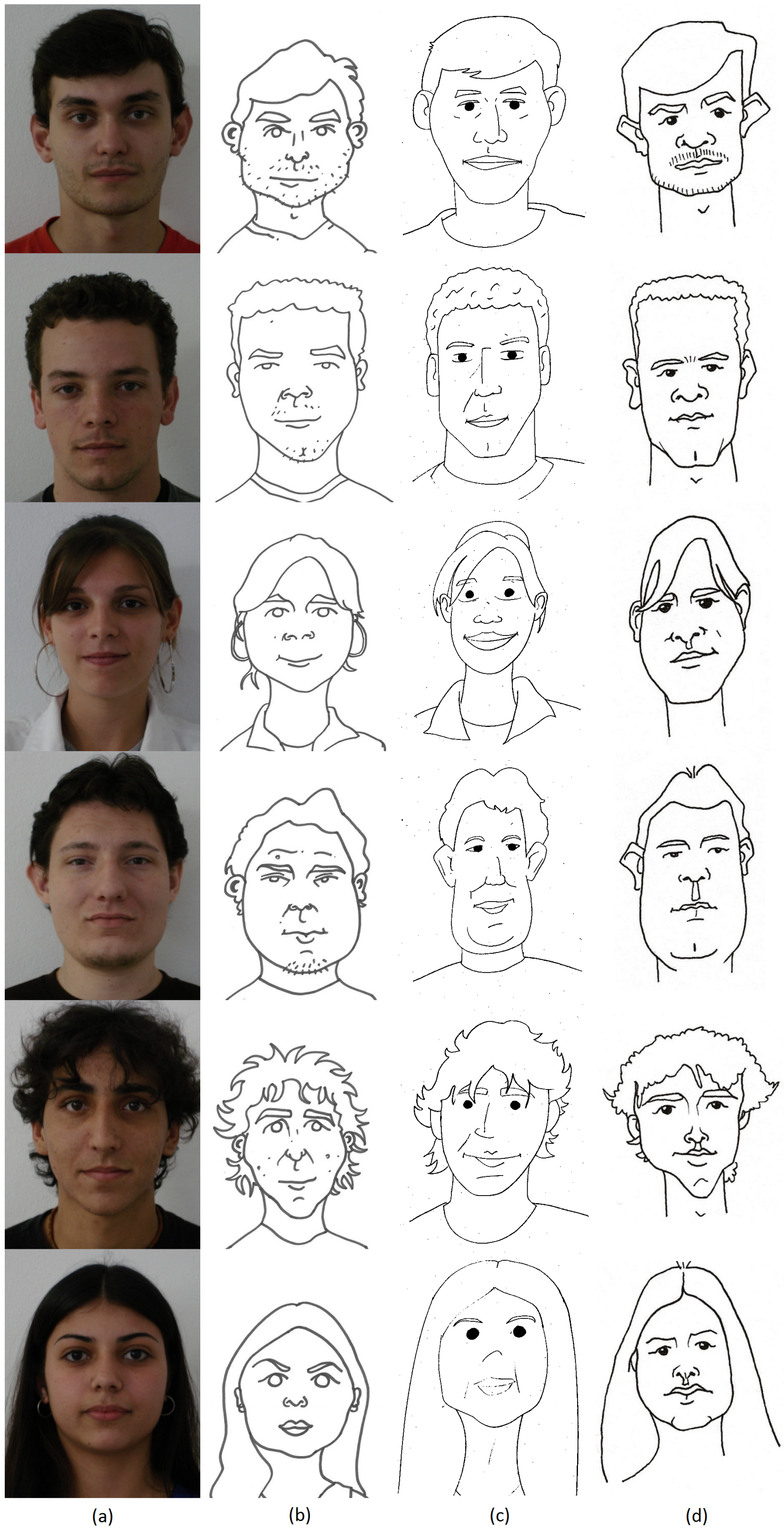


Figure C.13: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

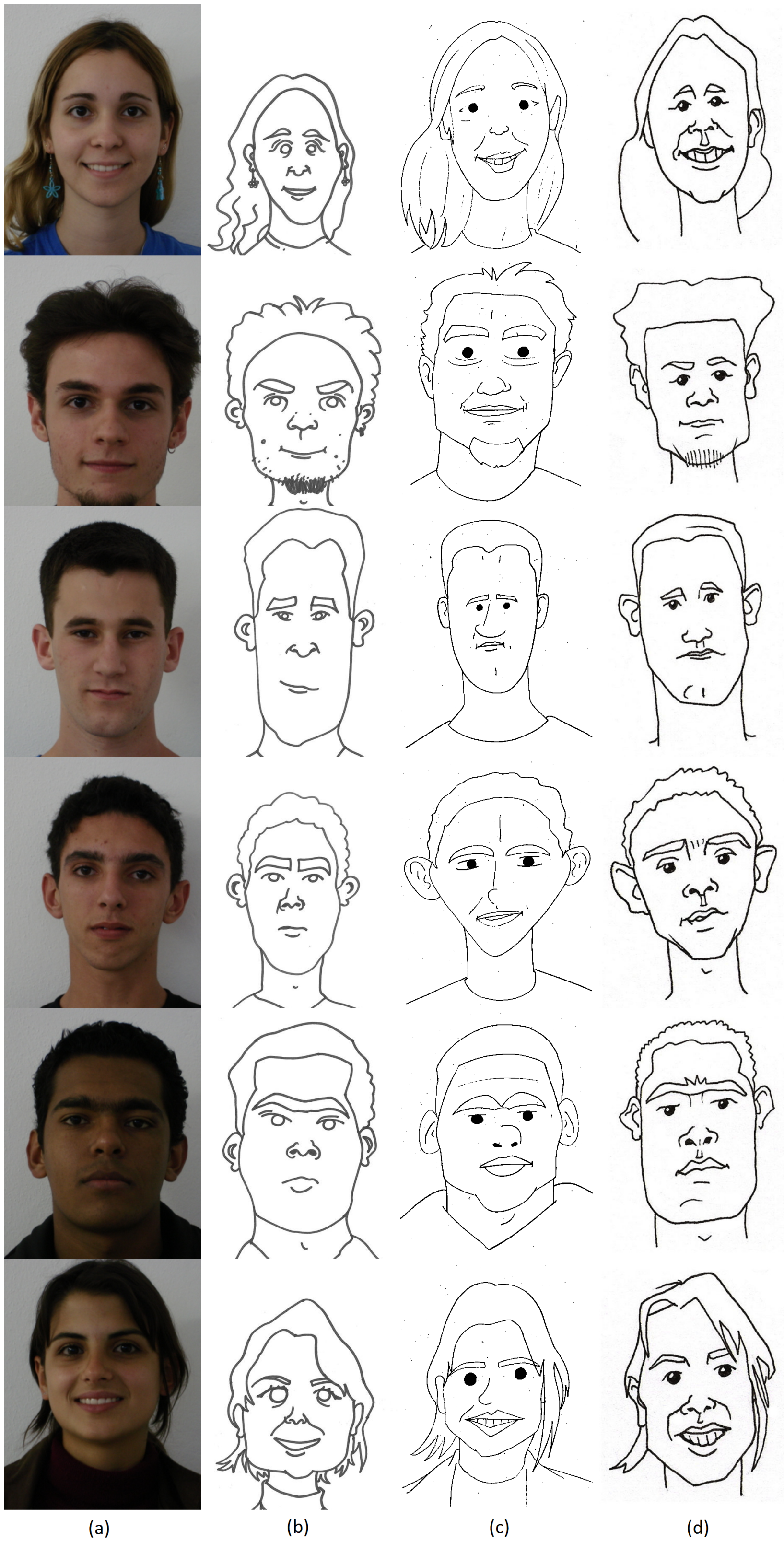


Figure C.14: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

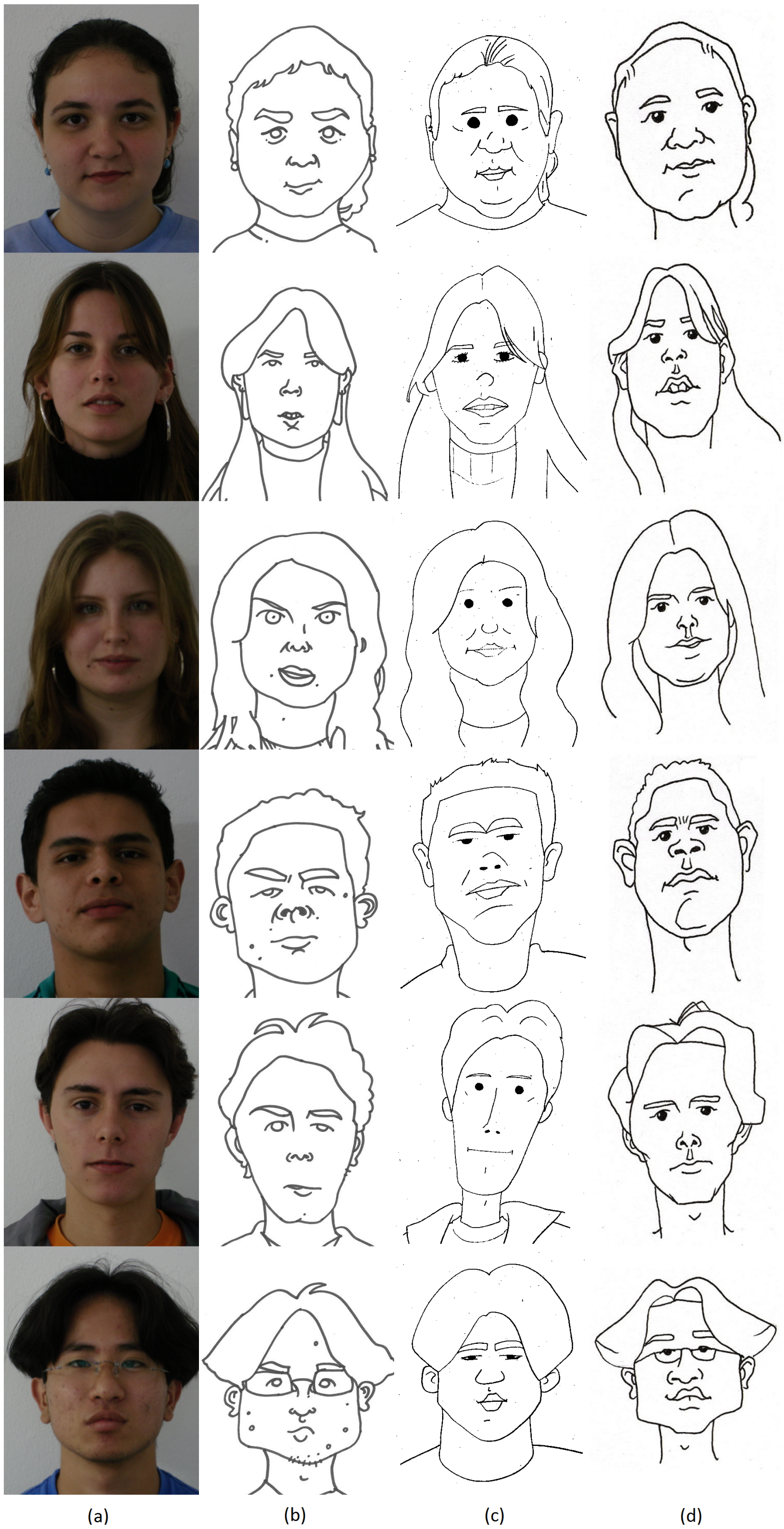


Figure C.15: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

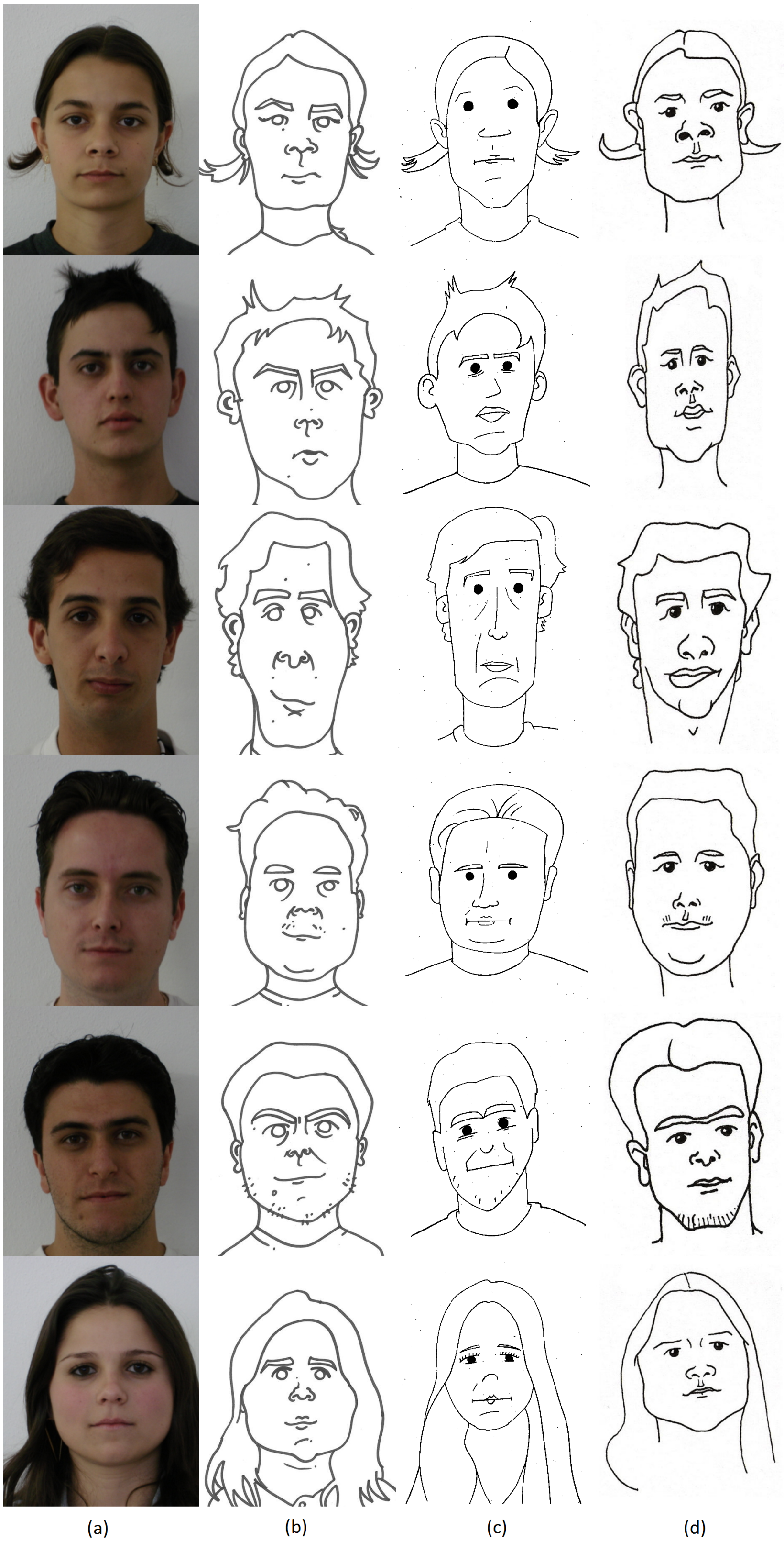


Figure C.16: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

Figure C.17: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Caricatures drawn by Uğur Erden (2nd Caricaturist), (d) Caricatures drawn by Armağan Yuksel (3rd Caricaturist)

Appendix D

Dissimilarities Between Different Caricatures of the Same Test Image

Table D.1: Similarity Values of the Caricatures Generated by Using Styles of Different Caricaturists for Same Test Images

	1st-2nd Caricaturist	**2nd-3rd Caricaturist**	**3rd-1st Caricaturist**
Test Image 1	0,449678834	0,47009013	0,446946912
Test Image 2	0,471670108	0,444400659	0,47587187
Test Image 3	0,443847309	0,434314035	0,422752825
Test Image 4	0,450596822	0,507896746	0,499671826
Test Image 5	0,444683105	0,438387901	0,441211977
Test Image 6	0,46079502	0,495733908	0,506952678
Test Image 7	0,473532697	0,466048015	0,451613984
Test Image 8	0,45566852	0,459287042	0,465925645
Test Image 9	0,496383481	0,446567901	0,507420393
Test Image 10	0,432775548	0,434504657	0,434204484
Test Image 11	0,432276264	0,444282226	0,446657263
Test Image 12	0,432457909	0,41316686	0,435172287
Test Image 13	0,458869632	0,457574529	0,450345607
Test Image 14	0,430464048	0,420466291	0,421944158
Test Image 15	0,481567583	0,442459114	0,489046244

Appendix E

Deformed Images with Different Similarity Metrics

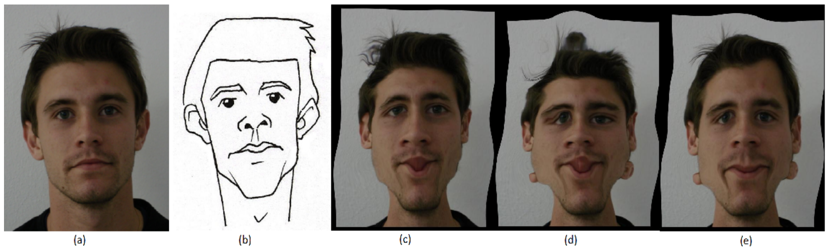


Figure E.1: (a) Original Images, (b) Caricatures drawn by Sinan Gürcan (1st Caricaturist), (c) Deformed Image Generated by Using L2 Form, (d) Deformed Image Generated by Using Lp Form, (e) Deformed Image Generated by Using Cosine Value