MANTECH Publications

Journal of Image and Signal Processing
Volume 2 Issue 2

# *Automatic Caricature Generation*

***Nikhil Biju[1], Akhil A[2], Sanath Hemanth T[3], Eldho P Elias[4], Surekha Mariam Varghese[5]***

*Department of ECE*

*M A College of Engineering, Kothamangalam, India*

***Corresponding Authors:*** *nikhilbiju67@gmail.com[1]*

***Abstract***

*Caricature is a popular artistic media widely used for effective communications. The fascination of caricature lies in its expressive depiction of a person's prominent features, which is usually realized through the so called exaggeration technique. The system comprises the facial feature analysis and the generation of caricatures. The different facial landmarks are identified from the face and the caricature is drawn with exaggeration For the input face, the system automatically compute the feature vectors of facial parts and hairstyle, and do the exaggeration on the most prominent features by using the average face model. Using an average face model , the system is capable of producing caricatures of different persons effectively and efficiently. Python's turtle framework is used to draw the exaggerated caricature of the person in the input image.*

***Keywords:*** *Dlib Library, Turtle Graphics, Open CV*

## I. INTRODUCTION

Caricatures serve as effective media for communication in many settings. These "deformed" images, which emphasize a person's most prominent features to make it easy for observers to identify the subject at a glance, range from artistic portraits to satirical or parodic illustrations and comics. People now use them as everything from gifts to resources for criminal investigations and satires of politicians. Caricatures are a highly individualized form of art, but not

everyone is capable of drawing a caricature on his or her own. Given the challenges involved, people who want caricatures for personal use would benefit greatly from a generation system that anyone could use to produce caricatures without much difficulty. This paper proposes a new automatic caricature generation technique: an example-based caricature generation system that uses exaggerations of visual appearance features. We implemented a system that take an image as input and gives out the required caricature by performing the exaggeration technique on the input image.

This paper will unfold into three sections whereby first will focus on the technology used second will focus on the implementation details while the last part showcase the result.

## II. FRAMEWORK

### *1. dlib Library*

Dlib is a general purpose cross-platform software library written in the programming language C++. Its design is heavily influenced by ideas from design by contract and component-based software engineering. Thus it is, first and foremost, a set of independent software components. It contains software components for dealing with networking, threads, graphical user interfaces, data structures, linear algebra, machine learning, image processing, data mining, XML and text parsing, numerical optimization, Bayesian networks, and many other tasks. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows it to be used in any application, free of charge. The major features are:

***Documentation***

Unlike a lot of open source projects, this one provides complete and precise documentation for every class and function. There are also debugging modes that check the documented preconditions for functions. When this is enabled it will catch the vast majority of bugs caused by calling functions incorrectly or using objects in an incorrect manner.

***High Quality Portable Code***

Good unit test coverage. The ratio of unit test lines of code to library lines of code is about 1 to 4.No other packages are required to use the library. Only APIs that are

provided by an out of the box OS are needed.

***2. Turtle Graphics***

Turtle graphics is a term in computer graphics vector graphics using a relative cursor (the "turtle") upon a Cartesian plane. Turtle graphics is a key feature of the Logo programming language. The turtle has three attributes: a location, an orientation (or direction), and a pen. The pen, too, has attributes: color, width, and on/off state. The turtle moves with commands that are relative to its own position, such as "move forward 10 spaces" and "turn left 90 degrees". The pen carried by the turtle can also be controlled, by enabling it, setting its color, or setting its width.

A full turtle graphics system requires control flow, procedures, and recursion: many turtle drawing programs fall short. From these building blocks one can build more complex shapes like squares, triangles, circles and other composite figures. The idea of turtle graphics, for example is useful in a Lindenmayer system for generating fractals.

***3. OpenCV***

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation.

OpenCV runs on a variety of platforms. Desktop: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD; Mobile: Android, iOS, Maemo.

**III. IMPLEMENTATION**

The proposed system mainly consist of three parts: the facial feature extraction, exaggeration module and the drawing module. Each of these modules works one after another to create the exaggerated caricature of persons.

The facial landmarks detection module identifies the different facial features such as eyebrows, eyes, nose, lips and cheek. We use 68 facial landmark points to identify these features. The exaggeration module use the help of an average face to do the

exaggeration. It compares the given face with the average face model and decides the amount of exaggeration that is needed. The drawing module uses python's turtle framework to draw the caricature.

### *A. Facial Feature Extraction*

This section defines the face mesh employed in our caricature generation system. The characteristics of the different types of constituent nodes are elucidated. We then proceed to compute the parameters of the average face and discuss the method to apply exaggeration to the face components.

To control the shape and appearance of each facial feature, dlib define a set of 68 points based on the MPEG-4 face definition parameters (FDP) and face animation parameters (FAP). These nodes are categorized into 5 groups, namely, face contour, eyebrows, eyes, nose and lip. The extracted facial landmark points are stored in an array. In the figure 1, the 5groups are identified and the corresponding x, y coordinate values are stored in a 68*2 array.

***Figure 1: Facial landmarks***

Hair regions are identified by applying a mask on input image. The mask is created in such a way that its output will be the dark regions in the input image. Before applying the mask the input image is converted to grayscale. And when the mask is applied the output will be a binary image where the black represents the hair regions in the image.

***B .The exaggeration Module***

The exaggeration of the facial features is done by this module. This module use an average face to determine the amount of exaggeration that is needed. The facial features of the average face is analyzed and is compared with the input face and the features are compared to determine whether the exaggeration is needed or not.

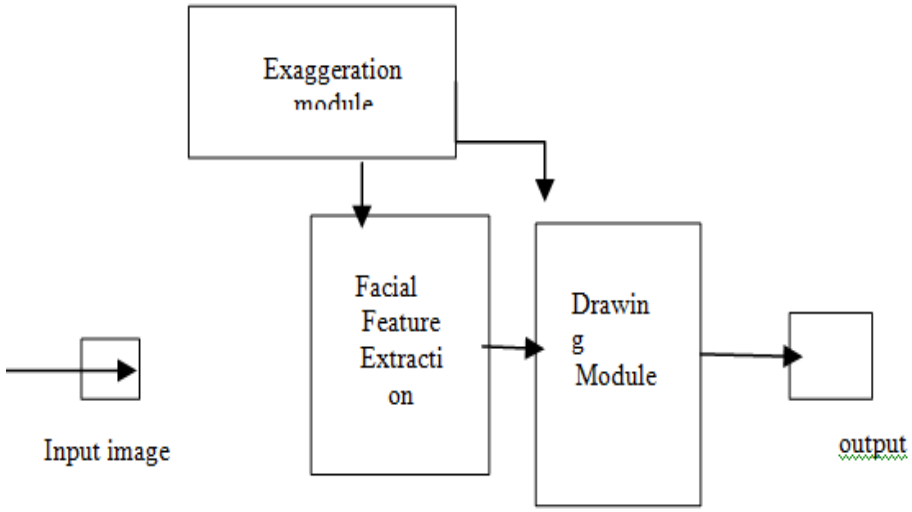
The best human average face is selected for the comparison the input image. The facial points of the average face is compared with the 68 facial features of the input image to determine the amount of facial exaggeration. The information about the exaggeration needed is given to the drawing module.

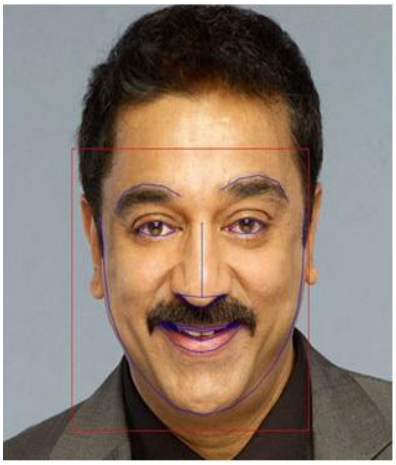
***C. The drawing module***

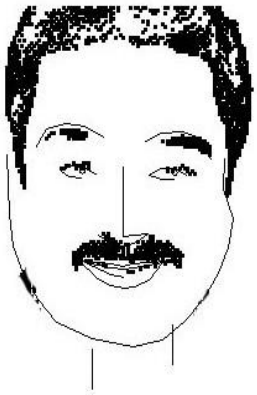
The drawing module draws the picture based on the facial features extracted by the facial feature extraction module and using the exaggeration information obtained from the exaggeration module the caricature with exaggeration is drawn. At first the 68 facial points are drawn. The facial points are suitably jointed to draw the face. Some of the facial feature coordinates altered by the exaggeration module to give the exaggeration.

The caricature is drawn according to the exaggeration information given by the exaggeration module and also from the facial landmark identification module. The caricature of the person is drawn with artistic effect.

The drawing is done using the python's turtle framework. The drawing coordinates are given to the turtle to draw the caricature The turtle module provides turtle graphics primitives, in both object-oriented and procedure-oriented ways. Because it uses Tkinter for the underlying graphics, it needs a version of Python installed with Tk support.





***(a).Input image*** ***(b) Output generated***

***Fig 3 .The given input image and the caricature generated.***

**CONCLUSION**

The proposed system aims to generate caricature by analyzing facial features of the person in the input In this paper, proposed a approach for generating caricature drawings using only one existing model as the reference. An organized way for defining and estimating the face components and the

associated parameters has been developed. By combining effective feature analysis with the proposed face model, the caricature generated. The current prototype considers only the spatial relationship among the groups. Future system will also pay attention to the relative node position within each group to achieve more accurate shape definition. In addition, hair style classification will be included to enable the production of more appealing caricatures.

**REFERENCES**

1) Hughes, Caricatures (Learn to Draw), HarperCollins Publishers, London, 1999.

2) L. Redman, How to Draw Caricatures, Contemporary Books, Chicago, 1984.

3) S. Brennan, Caricature generator, Master's thesis, Cambridge, MIT, 1982.

4) L. Liang, H. Chen, Y-Q Xu, H-Y Shum. "ExampleBase Caricature Generation with Exaggeration," Proceedings of 10th Pacific Conference on Computer Graphics and Applications, 2002.

5) L. Liang, H. Chen, Y-Q Xu, H-Y Shum. "ExampleBased Caricature Generation with Exaggeration," Proceedings of 10th Pacific Conference on Computer Graphics and Applications, 2002.

6) H. Koshimizu, M. Tominaga, T. Fujiwara, and K.Murakami. "On Kansei Facial Processing for Computerized Facial Caricaturing System Picasso," Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, pp. 294-299, 1999.