International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 5 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT195523

3D Character Generation from Images using Convolutional Neural Networks and 3D-Character Factory

Anuj Potdar*[1], Mitali Ghotgalkar[2]
*[1]Cuddle.ai-Fractal Analytics, Mumbai, Maharashtra, India
[2]Technology Services, Fractal Analytics, Mumbai, Maharashtra, India

ABSTRACT
Face recognition using convolutional neural networks can be utilised for constructing 3D characters from a photograph or from a live person. By using convolutional neural networks human attributes like the color of the eyes, skin tone, hair color, presence or absence of facial hair, body type and so on can be identified and provided as a response to a 3D-Character in a game engine like Unity3D using factory pattern approach to make the 3D Character look like the subject.

Keywords : Convolution Neural Networks, 3D Characters, Factory Pattern for Developing 3D Characters

I. INTRODUCTION

Facial recognition although easily performed by humans, has been a challenging task for a computer system. Various deep learning algorithms have been recently devised for aiding computers in performing face recognition at an advanced level. These are widely used for restricting users from accessing a system, to grant certain privileges to certain users, etc. This paper highlights the use of Convolutional Neural Networks as a means for face recognition and extends this concept for generating 3D models using the face recognition data provided by CNN. Convolution Neural Networks is a powerful methodology for extracting feature maps and generating recognition patterns. This approach has a wide variety of applications including identifying patterns and matching signatures or handwriting, analyzing documents, advertising, etc.

CNNs consist of a layer of neurons or perceptrons. These layers of neurons are connected to each other. One layer is connected to the next and so on. There are three parts of a CNN - input, output and multiple middle or hidden layers. These middle layers are convolved with each other by means of a dot product of their weights. The input to the CNN is a visual image or a photograph of an object.

The output layer of the Convolution Neural Network provides all the extracted features of the input object Once the CNN provides the desired output of features, these features can serve as an input to the factory pattern approach.

Factory pattern approach is very useful when complex object creation is involved. A photograph of a human contains a huge amount of features. For example, the colour of the skin, height and width of the face, distance of the nose from the eyes, placement of the eyebrows and eyeballs, color of the eyeballs, colour of the hair, etc. These features can be classified into different classes and defined as enumeration types. For instance, color can be an enumeration type and its values can be blue, red, green, black and so on. Once these enumerations are defined, the inputs from the

CNN can be fed to a java backend program which classifies these features and provides a JSON response using which the client side 3D-Character factory builds a human 3D Character.

II. METHODS AND MATERIAL

A. Methodology

Figure 1 illustrates the methodology of extracting features from CNN and generating 3D models from these features.

The entire process can be broken down into two sub-processes:

1) Extracting feature information from the photograph

A photograph can be captured on the spot with the smartphone's camera or an existing photograph can be used for this experiment. This photograph is then stored on a remote storage which will be providing the necessary connection between CNN and the uploaded data. Once the CNN receives this input, it generates feature maps on the basis of a trained model and extracts all the necessary features needed. The CNN gives the output only as a medium of classification and the response needs to be compounded and provided as a "Human" object with all the attributes in it. This task is carried out by the backend server and it gives the response as an object with the help of JSON Object which is easily consumable by any client side software.

2) Feeding feature information to a program which generates the 3D model

The CNN provides feature characteristics, which are then fed into a 3D character factory program which generates dynamic objects on the basis of the input provided. The parameters are introduced into the client side software by means of a REST API call. Once this 3D character factory is populated with an object, a client side software made with a game engine like Unity 3D then accepts these parameters and generates the necessary 3D models using the Character Factory it has setup on its own end. These characters resemble the ones provided as an input to the application. Furthermore, these 3D characters are rendered on the smartphone screen.

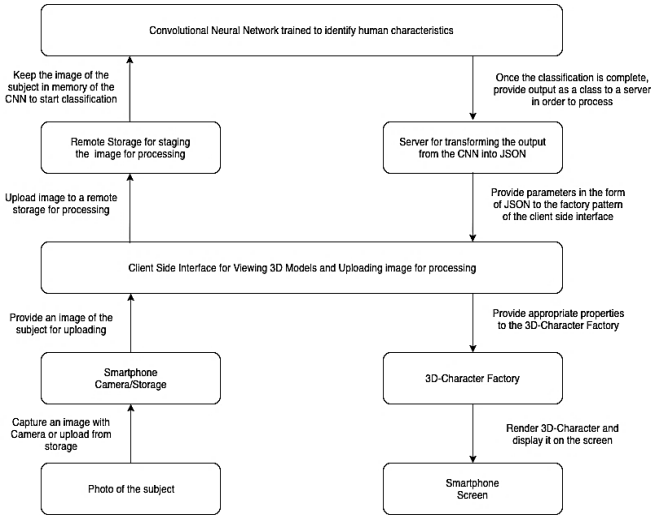

Figure 1. Methodology

B. 3D Character Factory Pattern

The responses from the Convolution Neural Network are fed to a Java program. A part of the code snippet is illustrated in Figure 2 below:

```
public enum HairColor
{
    RED, BLACK, BROWN
}

public static Hair buildHairColor(HairColor color)
{
    Hair humanHair = new Hair();

    switch(color){

        case RED:
        humanHair = new RedHair();
        break;

        case BLACK:
        humanHair = new BlackHair();
        break;

        case BROWN:
        humanHair = new BlackHair();
        break;

        default: break;

    }

    return humanHair;
}
```

Figure 2. 3D Character Factory Pattern

For this particular example, the Hair object is passed to the Human object along with other "feature" objects. These objects can then be aggregated to produce all the features which the 3D model needs as an input. In the example mentioned above, a "Human" object will be returned which will possess all the attributes and features needed to construct a 3D model.

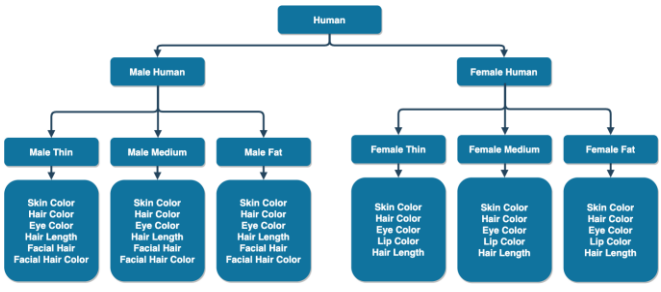

Figure 3. Segregation of Features

As illustrated in Figure 3, the output from the java backend program is segregated into the following broad categories:

1) Female Human
2) Male Human

This classification is further drilled down to feature level segregation where each aspect of the photograph is taken into consideration and a holistic model of either a male or a female is created which resembles the photograph.

III. RESULTS AND DISCUSSION

A. End-To-End Approach

Figure 4 illustrates how this approach is carried out on a female photograph end-to-end. The photograph is processed by CNN wherein feature maps are produced for matching and performing permutations and combinations. These features, as seen to the right of the diagram, are provided as an output from CNN. In this particular case, the photograph has black hair, brown eyes, white skin colour and black eyes.

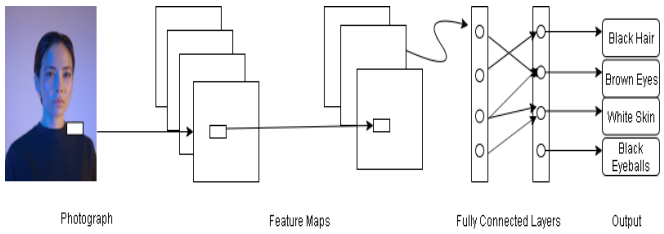

Figure 4. End-To-End Approach

B. Applications in Real World

1) 3D Model Generation using Convolution Neural Networks can be a powerful tool in game engines where the character in the game resembles the person playing that game.
2) This concept can be used for simulation purposes in the field of education for students to get a good knowledge of the subject.
3) In the field of cinematography and entertainment, this approach can help generate 3D characters that resemble freedom fighters and political leaders without having to develop a model from scratch by making a sophisticated factory and a CNN that

recognizes a much larger scope of attributes of human appearances.

4) In making augmented reality greeting cards or portraits, this CNN and 3D-Character factory based approach can be used to develop real-time greeting cards where the 3D-Characters will resemble the people in the cards.

IV. CONCLUSION

3D Model Generation using Convolution Neural Networks and Factory Pattern aids in extending the application of Convolution Neural Networks for generating 3D models that resemble the photograph provided as an input to CNN. This approach does not require much investment from an end-user perspective. This method can help save a lot of time and money spent on 3D Designers for developing 3D-Characters resembling a particular person, with a highly trained and sophisticated model the resemblance This methodology provides a novel and optimal way of using the face recognition feature already prevalent in the society.

V. REFERENCES

[1] A. B. Author, “Title of chapter in the book,” in Title of His Published Book, xth ed. City of Publisher, Country if not

[2] L. Wiskott, J.-M. Fellous, N. Kr¨uger, and C. von der Malsburg, “Face recognition and gender determination,” in Proc. Int. Wkshp. Automat. Face Gesture Recognition, Z¨urich, Switzerland, 1995.

[3] R. Brunelli and T. Poggio, “Face recognition: Features versus templates,” IEEE Trans. Pattern Anal. Machine Intell., vol. 15, pp. 1042–1052, Oct. 1993.

[4] JSON. json.org. http://www.json.org

[5] S. Klarr, “Javascript: What is json?,” 2007. http://www.scottklarr.com/topic/18/javascriptwhat-isjson

[6] B. Miller, “Vital signs of identity,” IEEE Spectrum, pp. 22–30, Feb. 1994.

[7] R. Brunelli and T. Poggio, “Face recognition: Features versus templates,” IEEE Trans. Pattern Anal. Machine Intell., vol. 15, pp. 1042–1052, Oct. 1993.

[8] R. Chellappa, C. L. Wilson, and S. Sirohey, “Human and machine recognition of faces: A survey,” Proc. IEEE, vol. 83, pp. 5, pp. 705–740, 1995.

[9] AUBEL, A., AND THALMANN, D. 2000. Realistic deformation of human body shapes. In Proceedings of Computer Animation and Simulation 2000, 125-135.

[10] R. Brunelli and T. Poggio. Face recognition: Features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(10):1042{1052, 1993.

[11] E. Bienenstock and C. von der Malsburg. A neural network for invariant pattern recognition. Europhysics Letters, 4:121{126, 1987.

[12] M. Lades et al. Distortion invariant object recognition in the dynamic link architecture. IEEE Trans. Comput., 42(3):300{311, 1993.

[13] T. Maurer and C. von der Malsburg. Single view based recognition of faces rotated in depth. In Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition, Zurich, 1995.

[14] E. Bienenstock and C. von der Malsburg. A neural network for invariant pattern recognition. Europhysics Letters, 4:121{126, 1987.

[15] P. Kalocsai, I. Biederman, and E.E. Cooper. To what extent can the recognition of unfamiliar faces be accounted for by a representation of the direct output of simple cells. In Proceedings of ARVO, May 1994.

Cite this article as :

Anuj Potdar, Mitali Ghotgalkar, "3D Character Generation from Images using Convolutional Neural Networks and 3D-Character Factory ", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 5, pp. 176-179, September-October 2019. Available at doi : https://doi.org/10.32628/CSEIT195523

Journal URL : http://ijsrcseit.com/CSEIT195523