IMAGE SIMILARITY CHECKING USING SSIM

Airish Tom
Department of Computer Applications
Amal Jyothi College of Engineering Kanjirappally, India
airishtom2023a@mca.ajce.in

Sr. Elsin C. SH
Department of Computer Applications
Amal Jyothi College of Engineering Kanjirappally, India
elsinchakkalackal@amaljyothi.ac.in

***Abstract*— The Study focuses on developing and implementing the Structural Similarity Index Measure (SSIM) for automated image comparison tasks. The SSIM method is widely used to assess how similar the two images are to one another. Here, we use SSIM to find the similarity of paintings uploaded by the artists. The similarity is predicated on the notion that the structural information, brightness, and contrast of two similar images should be identical. By comparing the mean, standard deviation, and covariance of the pixels in the two images, SSIM calculates these variables. The code in this study compares an input image to a set of reference images by using the SSIM approach. Prior to converting both images to grayscale, the algorithm resizes the input image to match the size of the reference images. The highest score is then returned for each pair of images once the SSIM score is determined for each pair of images.**

Keywords- SSIM, OpenCv, skimage, Numpy, Grayscale, Image Similarity

I. INTRODUCTION

Since the beginning of the Stone Age, artwork and paintings have been an essential component of human civilization. For ages, art galleries have played a significant role in the development of culture and the arts. With the development of digital technology, our perception of art has changed, and online art galleries are becoming more and more common. Paintings offer more knowledge about any subject than papers and the Old Testament. The artwork of many painters is preserved through the archiving of digital versions of paintings. The projected work aims to be similar to the artwork of other painters. However, ensuring the authenticity of the art being sold presents a challenge for running an online art gallery.

Fine art painting artist identification is a challenge that is mainly handled by art historians with substantial training and competence. This issue has been investigated in many earlier papers by specifically defining similarity features. Image similarity has long been a major issue in the fields of computer vision and image processing. When you compare two images, Image Similarity gives you a result that indicates how physically similar the images are. This test provides a numerical measurement of the degree of correspondence between the images in concern. This has been addressed using an image processing system that looks for image similarity using SSIM.

SSIM is a frequently used method for comparing the structure, brightness, and contrast information of two images to determine how similar they are. SSIM considers the perceptual qualities of images as opposed to other similarity measures that only consider pixel values, making it more resistant to noise and image compression artifacts. In order to do automated image comparison tasks, this paper describes an implementation of the SSIM technique utilizing Python, OpenCV, NumPy, and Flask.

II. LITERATURE REVIEW

Varshini Appana, Divya Shree, Himasri Kurra, Tulasi M anasa Guttikonda and Shahana Bano[1] in their paper titled “Similarity Score of Two Images using Different Measures” finds similarity score between the two images using several similarity metrics. SSIM, Pixel Similarity Measure, Earth Mover's Distance, and ORB were the similarity measures that produced effective scores. When given the input of two images, it can initially resize and normalize the image with a fixed size provided using height and width parameters. We can choose any image size so that it can be normalized and later altered for use in processing. The images' histograms are then acquired and utilized to compare the images. Histogram is a term used to describe a visual representation of the frequency of various color values in a picture. To prepare the images for additional processing, it normalizes the exposure. It calculates the normalization values for each location of the CDF (Cumulative Distribution Function) while normalizing by adding the values accumulated by each position in the histogram. Using each similarity metric, we compute the similarity score between each image by passing the path of the images as arguments, which computes the similarity score and outputs a float-valued similarity score. The similarity match between the submitted images is greater if the similarity score is close to 1. Thus, there is a higher likelihood that the images are similar. The offered images were not very similar if the score was close to 0

DOI: 10.5281/zenodo.7950300
ISBN: 978-93-5906-046-0@2023, Dept. of Computer Applications, Amal Jyothi College of Engineering Kanjirappally, Kottayam

Yiyu Hong and Jongweon Kim[2], in their paper titled "Art Painting Identification using Convolutional Neural Network" detects the copyrighted art paintings from the contents using the Convolutional Neural Network. The process is frequently divided into two steps: 1) segmenting the images that may contain art paintings or extracting the region of interest and (2) using image identification techniques to determine whether the region contains copyrighted art paintings. In the datasets of paintings, which represent simulated scenarios in which paintings might appear in contents, we trained a CNN. CNN will produce a determination of the identity of an image if one is input. The study suggested a way for using CNN to identify paintings of art. In order to depict diverse scenarios regarding the art painting, first created distorted representations of the paintings. The images are then sent to CNN, which has trained an algorithm to recognize paintings. Three CNN designs and associated parameters were examined, and the results demonstrated that even a little modification to the CNN topologies or parameters would have a considerable impact on performance.

Kaustav Mondal and H.B Anita[3], in their paper titled "Categorization of artwork images based on painters using CNN" uses the Convolution technique in the field of image processing to condense the meaning of an image and distil its information to a single, easily digestible point. Python and TensorFlow are used in the presented work's implementation. The suggested work uses the Inception V3 classifier. Srikar and Appalaraju Vineet Chaoji, in their paper titled "Image similarity using Deep CNN and Curriculum Learning" used the method, SimNet, which uses a multi-scale Siamese network to look for comparable images to a new image.

Srikar Appalaraju and Vineet Chaoji[4], in their paper titled "Image similarity using Deep CNN and Curriculum Learning" uses the method, SimNet, which is a multi-scale Siamese network to look for comparable images to a new image. It uses the CNN that creates state-of-the-art feature descriptors by self-learning features. When using metric-based learning, the similarity of images is accurately determined by learning a distance metric from labelled training samples in an embedding space. SimNet learns a 4096-dimensional embedding of an image using a multi-scale CNN in a Siamese network.

Zhizhe Liu , Luo Sun, and Qian Zhang[5], in their paper titled "High Similarity Image Recognition and Classification Algorithm Based on Convolutional Neural Network", suggests a convolutional neural network-fused high similarity image recognition and classification algorithm. Through image training, the method extracts texture features and optimizes parameter values. Based on textural variations, the image is divided into smaller pieces, from which energy features are then retrieved for categorization. Different kernel sizes are used for convolution feature fusion, and convolutional neural networks are used to transform the input image feature vector into a 1D vector. The network parameters are refined using training and data, and the most accurate model is then selected. Convolutional neural networks, subimage decomposition, and texture feature extraction are used in the technique to accurately classify images.

Mehmet Oguz Kelek, Nurullah Calik and Tulay Yildirim[6], in their paper titled "Painter Classification Over the Novel Art Painting Data Set via The Latest Deep Neural Network" uses CNN to classify and recognise images. CNN employs attributes that distinguish between living things or things in the image to categorize or characterize the given images.

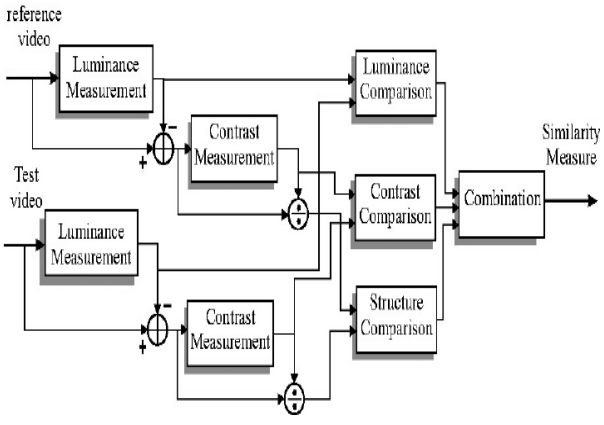
Nitin Viswanathan[7], in the paper titled "Artist Identification with Convolutional Neural Networks", made an effort to distinguish artists by outlining the characteristics that set them apart. A new strategy, however, has been put out that involves training CNNs to identify an artist's distinctive style and produce the finest feature representation for their paintings. This theory assumes that each artist has a distinctive artistic style and that employing a CNN can enhance currently used techniques for identifying artists.

III. PROPOSED METHODOLOGY

The structural similarity between the new image and each artist image in the dataset is determined using the SSIM technique. The artist image with the highest structural similarity score will be chosen by the algorithm as the new image's closest match. The three main aspects that the Structural Similarity Index (SSIM) metric extracts from an image are:

- Luminance
- Contrast
- Structure

These three features are used as the basis for comparing the two images. The layout and operation of the Structural Similarity Measurement system are depicted in Fig. 1 below. Signals X and Y point to the Reference and Sample Images, respectively.



.

DOI: 10.5281/zenodo.7950300
ISBN: 978-93-5906-046-0@2023, Dept. of Computer Applications, Amal Jyothi College of Engineering Kanjirappally, Kottayam

Fig 1

Luminance: Averaging across all of the pixel data yields the luminance value. Its symbol is (Mu), and the formula is shown below,

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i. \qquad (2)$$

The luminance comparison function $l(\mathbf{x},\mathbf{y})$ is then a function of μ_x and μ_y.

Contrast: The standard deviation (square root of variance) of all the pixel values is used to calculate it. Its symbol is sigma, and the following formula serves as its representation:

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)^2\right)^{\frac{1}{2}}. \qquad (4)$$

The contrast comparison $c(\mathbf{x},\mathbf{y})$ is then the comparison of σ_x and σ_y.

Structure: To compare the structural properties of two signals, a consolidated formula is used (more on that later). In essence, the input signal's standard deviation is divided by the result's standard deviation so that the result has a unit standard deviation, enabling a more reliable comparison.

$$(\mathbf{x}-\mu_x)/\sigma_x$$

Where, x is the Input Image

We define the comparison functions, followed by the combination function that results in the value of the similarity index.

Luminance comparison function: A function called l(x, y) defines the luminance comparison function. It is illustrated below. The mean of a particular image is represented by (mu). The two images being compared are x and y.

$$l(\mathbf{x},\mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2+\mu_y^2+C_1}$$

Where, C1 serves as a constant to provide stability when the numerator goes to zero. C1 is supplied by,

$$C_1 = (K_1L)^2$$

L is the pixel values' dynamic range. K1, K2 are only normal constants.

Contrast comparison function: The function c(x, y) described below defines the contrast comparison function. The symbol denotes an image's standard deviation. The two photos being compared are x and y.

$$c(\mathbf{x},\mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2+\sigma_y^2+C_2}$$

Where, C2 is given by,

$$C_2 = (K_2L)^2$$

Structure comparison function: The function s(x, y), which is depicted below, defines the structure comparison function. The symbol denotes an image's standard deviation. The two photos being compared are x and y.

$$s(\mathbf{x},\mathbf{y}) = \frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3}.$$

Where, σ(xy) is defined as,

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y).$$

And finally, the SSIM score is given by,

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}.$$

Steps involved are:

1) Gathering the image dataset: we gather and store a variety of images of paintings by various painters.

2) Image comparison algorithm implementation: The image comparison algorithm implemented in this project compares the user's input image to the images saved in the folder using the SSIM approach. A popular metric for determining how structurally similar two images is called SSIM. The algorithm reads each image in the folder one at a time, computes the

DOI: 10.5281/zenodo.7950300
ISBN: 978-93-5906-046-0@2023, Dept. of Computer Applications, Amal Jyothi College of Engineering Kanjirappally, Kottayam

SSIM score between the user's input image and every image in the directory, and then stores the highest SSIM score.

3) Output: The system determined the SSIM score for each image in the directory.

```
Python > server.py > ...
1  import os
2  import cv2
3  import numpy
4  from flask import Flask, request, jsonify
5  from skimage.metrics import structural_similarity as compare_ssim
6
7  app = Flask(__name__)
```

The programme imports the required packages and libraries. An interface for dealing with the operating system is provided by the OS module. The OpenCV library for image processing is called cv2. The Python library for numerical computing is called numpy. This API was created using the Python web framework Flask. The Flask request and jsonify modules are used to manage requests and response information in the API. Finally, the SSIM score is computed using the skimage.metrics module.

```
6
7  app = Flask(__name__)
8
```

creates a Flask class instance, the primary application object.

```
9  @app.route('/compare', methods=['POST'])
10 def compare():
```

This Flask decorator defines the application's route as /compare and the HTTP method as POST. This indicates that an image file will be accepted as input data by the API through a POST request to this endpoint. The handler for this endpoint is the compare() method.

```
11  if 'file' not in request.files:
12      return jsonify({'error': 'Missing file'})
```

This line determines whether the request contains the input image file. If not, a JSON-formatted error message is returned.

```
file = request.files['file']
img1 = cv2.imdecode(numpy.fromstring(file.read(), numpy.uint8), cv2.IMREAD_COLOR)
```

The imdecode() function of OpenCV is used to decode the input picture file and convert it into a NumPy array. The SSIM calculation is then performed using the array as input.

```
images_folder_path = os.path.join(os.path.dirname(__file__), '..', 'artist', 'images')
```

This line specifies the location of the folder holding the comparison's reference images.

```
18  max_ssim = 0.0
19  max_sim_img_path = None
20
```

It creates the greatest possible SSIM score as well as the path to the image with that score.

```
for filename in os.listdir(images_folder_path):
    if filename.endswith('.jpg') or filename.endswith('.jpeg') or filename.endswith('.png'):
        img2_path = os.path.join(images_folder_path, filename)
        img2 = cv2.imread(img2_path)
        img1_resized = cv2.resize(img1, (img2.shape[1], img2.shape[0]))
        gray_img1 = cv2.cvtColor(img1_resized, cv2.COLOR_BGR2GRAY)
        gray_img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
        ssim = compare_ssim(gray_img1, gray_img2)
        if ssim > max_ssim:
            max_ssim = ssim
            max_sim_img_path = img2_path
```

The SSIM score between each reference image and the input image is determined by reading each image from the reference folder one at a time and using this block of code. Since grayscale images are used to calculate the SSIM score, the cv2.imread() function reads the reference image and then the images to grayscale. Next, the compare_ssim() function from the skimage.metrics module is used to calculate the SSIM score. If a higher score is discovered, the maximum SSIM score and the path of the image with the highest score are updated.

```
if max_sim_img_path is None:
    return jsonify({'score': 0.0})
```

This line searches for comparable images. If not, a JSON response reflecting a score of 0.0 is returned by the function.

DOI: 10.5281/zenodo.7950300
ISBN: 978-93-5906-046-0@2023, Dept. of Computer Applications, Amal Jyothi College of Engineering Kanjirappally, Kottayam

```
print("================================================\n")
print("score : ", max_ssim)
print("score with rounded value (1): ", round(max_ssim,1))
print("\n")
print("================================================\n")
return jsonify({'score': round(max_ssim,1)})
```

These lines display the maximum SSIM score as well as the score in rounds.

IV. RESULT

Using the SSIM, the copyrighted art paintings are founded. The art painting images that are uploaded are compared with the images in the folder, and if there is a similar image, then it shows the copyright of the painting. It computes the similarity scores between every image in the folder and the inputted image.

V. CONCLUSION

In conclusion, this research described a technique for identifying artists' paintings based on the structural similarity index measure (SSIM). The method compares an input image to a set of reference images kept in a local directory. In order to expose a RESTful API endpoint that accepts an image file as input and returns the SSIM score between the input image and the reference image with the highest similarity in the directory, we used the Flask web framework. This technique successfully identifies the artist of a given painting with a high degree of accuracy, according to testing on a database of works by various artists. Due to its reliance on the structural features of the data, our method has the benefit of being computationally efficient and requiring little training data.

VI. REFERENCES

[1] Varshini Appana, Divya Shree, Himasri Kurra, Tulasi M anasa Guttikonda and Shahana Bano "Similarity Score of Two Images using Different Measures", April 20th, 2022

[2] Yiyu Hong and Jongweon Kim "Art Painting Identification using Convolutional Neural Network", 2017

[3] Kaustav Mondal and H.B Anita, "Categorization of artwork images based on painters using CNN", 2021

[4] Srikar Appalaraju and Vineet Chaoji, "Image similarity using Deep CNN and Curriculum Learning",September 26th , 2017

[5] Zhizhe Liu , Luo Sun, and Qian Zhang, "High Similarity Image Recognition and Classification Algorithm Based on Convolutional Neural Network", April 12th, 2022

[6] Mehmet Oguz Kelek, Nurullah Calik and Tulay Yildirim, "Painter Classification Over the Novel Art Painting Data Set via The Latest Deep Neural Network", 2019

[7] Nitin Viswanathan, "Artist Identification with Convolutional Neural Networks"

[8] Alexander Blessing and Kai Wen, "Using Machine Learning for Identification of Art Paintings"

DOI: 10.5281/zenodo.7950300
ISBN: 978-93-5906-046-0@2023, Dept. of Computer Applications, Amal Jyothi College of Engineering Kanjirappally, Kottayam