COMPUTER VISION BASED MODEL FOR ART SKILLS ASSESSMENT

Thesis Submitted to

The College of Arts and Sciences of the UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for The Degree of

Master of Science in Computer Science

By

Asaad Saeed M Alghamdi Dayton, Ohio December 2022

COMPUTER VISION BASED MODEL FOR ART SKILLS ASSESSMENT

Name: Alghamdi, Asaad Saeed M APPROVED BY:

|  |  |  |
| --- | --- | --- |
| Dr. Tam Nguyen  Advisory Committee Chairman Assistant Professor  Department of Computer Science |  | Dr. James Buckley Assistant Professor  Director of Graduate Program in Computer Science  Associate Professor  Department of Computer Science |
| Dr. Tom Ongwere Assistant Professor  Department of Computer Science | | |

© Copyright by Asaad Saeed M Alghamdi

All rights reserved 2022

ABSTRACT

COMPUTER VISION BASED MODEL FOR ART SKILLS ASSESSMENT

Name: Alghamdi, Asaad Saeed M University of Dayton

Advisor: Dr. Tam Nguyen

Drawing is assumed to be a talent in people, which makes drawing a natural process. Based on the scenario in consideration, there are times when it is necessary to assess drawing skills. In this thesis, I intend to develop an algorithm that will measure the skill of drawing by matching the hand-drawn image with the original template. The techniques that are already available make use of a complicated process. Notably, computers can be trained to identify the match at a human level which will resolve the tedious and overwhelming traditional process. Image similarity involves identifying the level of similarities in an image using a reference image; computer vision applications are used. The SIFT method and Siamese Network are analyzed and implemented to measure image similarity. The results show that it is possible to measure the skill level of an art. Via the analysis of the features, SIFT-based implementation was able to detect better than VGG16.

*KEYWORDS: Computer Vision, OpenCV, Matching Object, SIFT, Siamese Network*

This thesis is dedicated to my late father: Saeed Mohammed Alghamdi 08/04/1940 – 05/02/2021

Though you never got to see this research, you are in every page.

ACKNOWLEDGMENTS

Though beyond what words can express, I would like to express my special thanks to my advisor Dr. Tam Nguyen for providing me with the information necessary for this project's success and offering direction on this thesis. I value your patience and feedback and appreciate your commitment throughout the project. Additionally, the process was made much smoother with the generous treatment from my mother, my wife, and my siblings, who have offered moral support, which is a significant element of success. Finally, I also would like to thank my friends who offered me editing help, feedback, and academic criticism.

TABLE OF CONTENTS

[ABSTRACT 3](#_bookmark0)

[DEDICATION 4](#_bookmark1)

[ACKNOWLEDGMENTS 5](#_bookmark2)

[LIST OF FIGURES 8](#_bookmark3)

[LIST OF TABLES 9](#_bookmark4)

[LIST OF ABBREVIATIONS AND NOTATIONS 10](#_bookmark5)

[CHAPTER I INTRODUCTION 11](#_bookmark6)

* 1. [*Thesis Structure* 11](#_bookmark7)
  2. [*Background* 11](#_bookmark8)
  3. [*Motivation* 12](#_bookmark9)
  4. [*Research objective* 13](#_bookmark10)
  5. [*Problem statement* 13](#_bookmark11)

[CHAPTER 2 RELEVANT RESEARCH 15](#_bookmark12)

[CHAPTER 3 METHODOLOGY 19](#_bookmark13)

* 1. [*Overview* 19](#_bookmark14)
  2. [*SIFT Algorithm* 19](#_bookmark15)
     1. [*The multi-stage framework of SIFT model* 23](#_bookmark18)

1. [Detection of classes 23](#_bookmark20)
2. [Art skill level detection 25](#_bookmark22)
3. [Classification of art skill 28](#_bookmark28)
   1. [*Siamese Network-based approach* 29](#_bookmark30)
      1. [*Convolution Neural Networks* 29](#_bookmark31)
         1. [*Siamese Networks* 31](#_bookmark33)
            1. [*The VGG16 approach using Cosine & Euclidean similarity* 33](#_bookmark35)

[3.3 Data collection 37](#_bookmark40)

[CHAPTER 4 RESULTS AND DISCUSSIONS 38](#_bookmark42)

* 1. [*General Overview* 38](#_bookmark43)
  2. [*System Specifications* 38](#_bookmark44)
  3. [*Operational Parameter* 39](#_bookmark45)
  4. [*Failure modes and actions on failure* 39](#_bookmark46)
  5. [*Limitations and Restrictions* 39](#_bookmark48)
  6. [*Feature Detection Results* 40](#_bookmark49)
  7. [*State of the Art* 46](#_bookmark52)

[CHAPTER 5 CONCLUSION AND FUTURE WORK 48](#_bookmark53)

[REFERENCES 51](#_bookmark55)

[APPENDIX - A Data analysis sheet 55](#_bookmark56)

LIST OF FIGURES

[**Figure 1:** Lowe's pyramid scheme. 20](#_bookmark16)

[**Figure 2:** SIFT keypoints are circular image regions with an orientation. 22](#_bookmark17)

[**Figure 3:** Framework of SIFT model 23](#_bookmark19)

[**Figure 4:** Model Training 24](#_bookmark21)

**Figure 5:** Accuracy and data loss trends 25

[**Figure 6:** Pre-processing Steps 26](#_bookmark23)

[**Figure 7:** Keypoints calculating formula 26](#_bookmark24)

[**Figure 8:** Flann-Based Matcher 27](#_bookmark25)

[**Figure 9:** SIFT Matches 27](#_bookmark26)

[**Figure 10:** With KNN, good matches are calculated 28](#_bookmark27)

[**Figure 11:** Skill levels roles 29](#_bookmark29)

[**Figure 12:** Two hidden layer Siamese network for binary classification with logistic prediction.](#_bookmark32)

[................................................................................................................................................. 30](#_bookmark32)

[**Figure 13:** Siamese network architecture. 32](#_bookmark34)

**Figure 14:** The VGG16 using Cosine and Euclidean methods 33

[**Figure 15:** Cosine Similarity Equation 34](#_bookmark36)

[**Figure 16:** Euclidean Distance Equation 34](#_bookmark37)

[**Figure 17:** Three functions of the VGG16 model 35](#_bookmark38)

[**Figure 18:** Other functions to find image similarity by VGG16. 36](#_bookmark39)

[**Figure 19:** The collage of some images that we used in this research 37](#_bookmark41)

**Figure 20:** Matching Results 40

**Figure 21:** Comparison between results from human judgment and results from SIFT algorithm

................................................................................................................................................. 41

[**Figure 22:** A value is not considered to be a valuable match if there is a difference in contrast](#_bookmark50) [between two points (light on dark and dark on light background) 42](#_bookmark50)

[**Figure 23:** Performance of SIFT descriptor-based image matching 44](#_bookmark51)

[**Figure 24:** VGG16 Architecture. 49](#_bookmark54)

LIST OF TABLES

[**Table 1:** Failure modes and actions on failure 39](#_bookmark47)

LIST OF ABBREVIATIONS AND NOTATIONS

|  |  |
| --- | --- |
| OpenCV | Open-Source Computer Vision Library |
| SIFT | Scale Invariant Feature Transform |
| SURF | Speeded Up Robust Features |
| ORB | Oriented Fast and rotated Brief |
| VGG | Visual Geometry Group (16-19) |
| HOG | Histogram of Oriented Gradients |
| DoG | Differences of Gaussian |
| CNN | Convolutional Neural Network |
| RNN | Recurrent neural network |
| KNN | K-Nearest Neighbors’ algorithm |

CHAPTER I INTRODUCTION

* 1. *Thesis Structure*

The thesis is organized into several chapters; Chapter 1 introduces the background, motivation, research objective, and problem statement. Chapter 2 describes the relevant research. Meanwhile, Chapter 3 introduces the methodology conducted in this research. Chapter 4 presents the experimental results and discussions. Finally, Chapter 5 concludes this thesis and paves the way for future research.

* 1. *Background*

We are in an era where computers can do most things humans can. Interestingly, some technology research argues that computer enjoys other advantages over humans; better memories, the ability to hold a large amount of information, and do not require sleep like humans; hence their ability to calculate and analyze tasks tirelessly around the clock. Smartphone apps like virtual trainers and virtual exercise assistance are developed using OpenCV. OpenCV is an excellent tool for image processing and computer vision-related tasks. The platform has been designed as an open library that allows users to perform tasks such as face detection in images, objection, and landmark detection (1). OpenCV offers flexibility to developers since it can be designed using multiple languages such as Python, Java, or C++. Through hundreds of functions and algorithms available in the library, users find the element of convenience in the apps. In virtual trainers, the pose is estimated for the correct guidance, and computer vision techniques are used in many medical fields, like anomaly detection and tumor detection. Moreover, it is utilized in agriculture, such as plant disease and leaf disease. For the development of self-driving cars, the first scenario to keep in mind is a vision done using OpenCV.

# 11

* 1. *Motivation*

For many years, visual art was defined by painting, drawing, and art sculpture. In 1826, a new visual art photography format was invented; however, there are some cases when a drawing is a preferred art format. For instance, drawings communicate the technical detail of a project in a standard format (2). Law enforcement agencies have automatic face recognition that helps the investigation process. The system of photographic evidence enables law enforcement agencies to identify participants involved in a crime by retrieving their faces in their photographic archives. In extreme cases, there might be no photographic archives available and in such a case, a suspect is identified using the memory of the eyewitnesses through the production of forensic sketches. While cases of crime involving murder have increased over the past decades, a more reliable approach of matching a face sketch to an existing database of a photograph is required (2).

Research shows that face photo recognition technology has been advancing. However, the complexity of matching a sketch to a photograph is also in existence. The sketch and photo are two different moderates, and different correspondence between the intensity value of the Pixel never exists. Unlike a photograph, a sketch is a drawing using repeated line-stroke and shades with different textures (2). Using the same approach, we are confident that it can develop a program that will assist in defining the skill level of an artist who draws a sketch based on a photograph.

OpenCV is used to achieve a rapid and robust matching method necessary in the application of the below fields:

* + - Robotics.
    - Image processing.
    - Computer vision.

Several methods are available to handle this task in the present age, such as ORB, SURF, SIFT, and KAZE. SIFT is commonly used since it allows users to create internal presentations

using the original image as a reference which is essential in maintaining the scale of invariance. Modern tools like artificial neural networks and deep neural networks have taken place that can outperform depending on data used for machine learning. The whole methodology of image similarity depends on the feature vectors of the images. These feature vectors can differ depending upon the methodology used, which means that the feature vectors for both images are compared to get similar results. For example, the face photo recognition technology used in forensics to identify a suspect can be applied in designing a program that will define the level of similarity between a drawn cartoon sketch and the actual image.

* 1. *Research objective*

This research aims to develop a program that offers users a technique that can measure art skills. The program will give value to the level of accuracy between a photo and a sketch developed using the same. To achieve the objectives, feature-matching techniques will be needed to be evaluated to produce the most accurate results possible. Creating a system that uses feature matching to analyze artistic abilities will also be conducted. Using OpenCV, we will create a system that displays the proportion of a drawing that matches a real image. Verification of outcomes will be done by comparing Siamese Networks and human assessments, which involves asking some experts about their opinions regarding the sketches.

* 1. *Problem statement*

Naturally, the human brain can recognize and analyze different sketches using distinct features hence judging the level of skill possessed by the artist. To compare photos or, more precisely, portions of images, we must have a superior computing tool. To manually spot the differences and similarities between a drawing and an actual photograph is a laborious effort. Notably, different people perceive things. People select different aspects of an image and pay attention to the aspect based on what interests them in an image. Most artists argue that drawing from an image is much more difficult than drawing from a real-life image that offers a 3D model.

One of the renowned artists argued that drawing from a photo involves copying contours and colors presented in the image, and an ability to copy defines the level of skill an artist has (3).

In an attempt to address the challenges, previous approaches offer a transformation of a photo into a sketch or vice versa in an attempt to reduce the modality gap. One research by Khan et al. (2012) found that "Such a transformation may be indeterministic and, if learned from training data, is likely to over-fit the sketch artist's drawing technique”. In this project, I will design a program that uses an image similarity model to compare sketches and their actual images. Comparing images involves comparing various elements and returning a value demonstrating how visually similar the sketch is to the actual image (2). The computer program will be able to frame the issue by comparing self-similarities between the cartoon's actual image and its sketch.

The program will be a self-similarity descriptor, and it will be obtained through the correlation of small patches within its neighborhood. Through the program, there will be no need for modality transformation, and the intramodality gap will be significantly reduced. The facial self-similarity we shall design will use a database containing elements of sketch photo pair and the real image. We expect our program to have above 97% accuracy. The program should also outperform the available techniques, such as one used android and windows apps developed using OpenCV. Users of the program can select the artist's degree of expertise. The software, which will take the shape of an app, will be able to compare drawn drawings scanned in and produce numbers that indicate how close they are aesthetic. Other image similarity approaches are utilized to validate results and obtain the best outcomes.

CHAPTER 2 RELEVANT RESEARCH

This section explores previous research and studies on the topic that would be essential in contributing to the research. The use of computer vision techniques to measure art skills is a field that still leaves a gap. We, however, discovered a significant amount of research publications that give comparative findings on a variety of feature detectors and descriptors, which are critical to my study. In this area, we have explored various approaches used in feature detection: SIFT. SURF, ORB, among others.

The technology of computer vision assessment has a high chance of improving the art skills of humans if adopted (3). We aim to have a program that can assess a hand-drawn object and evaluate the artistic talent of an artist. Various research demonstrates various ways to assess the quality of a hand-drawn image using a computerized program (4). Using a pre-determined scale of the results, sketches can be matched to a respective image and evaluate the skill level.

In this research, more focus will be on SIFT algorithm as the major algorithm that locates keypoints and gives descriptors on two fronts, one on the pre-determined image and the other on the sketch. There is a high similarity between SIFT method and ORB; however, SIFT is said to be slower (5). Based on its performance, SIFT is also considered rotation invariant and more resistant to noise. Therefore, researchers recommend using a fast-rotated approach as an alternative to SWIFT since it can block noise and use a fast function to conduct the computations (4). In research by Karami, Ebrahim, Siva Prasad, and Shehata, an extensive study of image matching was conducted using SIFT, BRIEF, ORB, and SURF. Applications on computer vision and robotics have developed a need for image matching that is fast and robust (6). During the study, Karami et al. compared the different approaches SIFT, BRIEF, ORB, and SURF, which built an essential reference for this research.

In a study by Karami, Shehata, and Smith, which was meant to evaluate the performance of SIFT matching algorithm, it was found that SIFT is preferred over scaling, rotation, and other image detection techniques. The research also found that the ability of a technique used to find similarities in a given image depends on various use cases, such as visual search (7). In another study, Luo et al. presented various enhanced ORB methods and image-matching techniques based on feature points. According to the study, though the ORB technique offers excellent real-time performance and significant gain in the computation process, there are several flaws with the technique. Therefore, Luo et al. suggest that combining ORB with another algorithm would improve its performance (8).

In research by H. Bay, the SURF detected was found to be based on the Gaussian scale- space analysis of an image (9). The SURF detector is determined by Hessian Matrix and uses integral pictures to raise the speed of feature detection. SURF's 64-bin descriptor characterizes each feature detected using a Harr wavelet dispersion. Though SURF is characterized by elements such as rotation and scale-invariant, low invariance is always a problem. The good news is that the descriptor can be expanded to a 128-bin value, making it capable of dealing with bigger perspective shifts. SURF outshines SIFT in computational costs. At scale σ, Hessian Matrix in

point x=x,y is represented by the equation: 𝐻(𝑥, 𝑦) = 𝐿𝑥𝑥(𝑥, 𝜎) 𝐿𝑥𝑦(𝑥, 𝜎)

𝐿𝑥𝑦(𝑥, 𝜎) 𝐿𝑦𝑦(𝑥, 𝜎)

𝐿𝑥𝑥(𝑥, 𝜎) - the convolution of Gaussian second-order derivative with the image “𝐼” in point “𝑥”. Similarly, 𝐿𝑥𝑦(𝑥, 𝜎), and 𝐿𝑦𝑦(𝑥, 𝜎),.

Alcantarilla et al. (2012) proposed the KAZE feature to leverage non-linear scale-space via non-linear diffusion filtering (10). It is expected that pictures blur and become locally adaptable to feature points. The conversion can lower noise and keep borders in the subject image. The scale-normalized determinant of the Hessian Matrix, computed at various scale levels, serves as the foundation for the KAZE detector. The detector response maxima are recognized as feature

points using a moving window. The property of rotation invariance is integrated into the feature description by identifying the dominant orientation in a circular area surrounding each detected feature. Rotation, scale, and restricted affine invariant properties of KAZE provide more uniqueness at different scales at a little longer processing time.

Using both hidden and specific layers, CNN is used for image recognition, segmentation, and detection in the research (11). VGG16 is considered to have a large database and is more connected. In addition, VGG16 has trainable layers that make it possible to fit accurately in more complex functions. For example, VGG16 is used to train the model for evaluating the artistic skills of a sketch based on the original image as the point of reference (12). Many studies suggest that VGG16 architecture is convenient for accurately performing complex functions. This is because of VGG19 architecture has more trained layers compared to VGG16. VGG19 architecture would work better in complex functions because it has more connected layers, which is suitable for performing complex functions (13).

Deep learning and neural networks (CNN and RNN) techniques are also used in image matching. However, the named techniques are more accurate because they are based on a much- improved design. In 2015, Tsung-Yi, Yin, Serge, and James proposed and implemented a very interesting idea. The authors planned to use deep networks to geolocative images by matching images without using ground-level as a reference image (14). Though their research is related to my project, different and specific input data is evaluated. Such research would pave the way for more discoveries on machines and computers to improve accuracy and save computation time. This can be achieved by using more accurate and fast detection methods, recognition methods, and segmentation methods by involving more connected layers and keypoints to improve the accuracy of image detection and recognition (15). Moreover, databases of the pre-determined image require to be enlarged as it can be very important in having different images that could be

compared to the sketches through the computer or machine learning which are specific and already trained to the machine.

In conclusion, the thesis is important in growing and advancing artistic skills by evaluating through a computer-generated application and paves the way for more advances in machine learning and machine training fields.

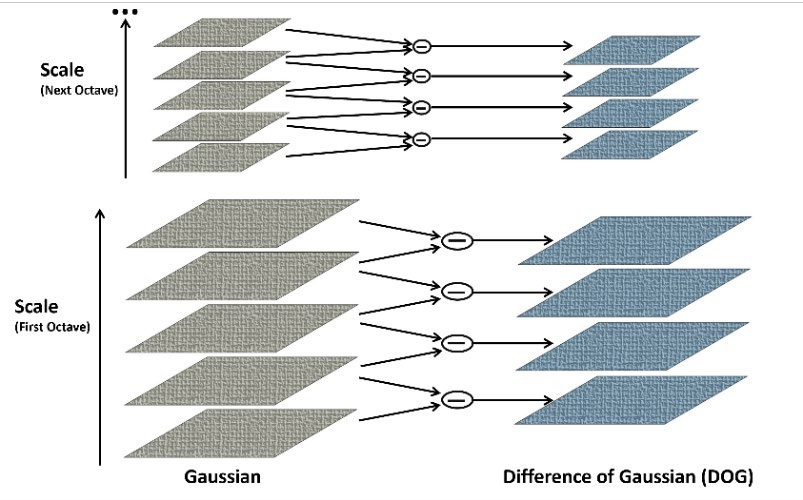
CHAPTER 3 METHODOLOGY

* 1. *Overview*

Methods used in achieving research objectives are outlined in this section. We focus our attention on image matching, as well as the main objective of the research is to develop a hand- drawing measuring model that uses computer vision. The proposed project implements the calculation of image similarity between two images using SIFT as the feature extraction method and then using the Siamese Network based on the Keras deep learning model (VGG16) pre-trained model that uses Cosine and Euclidean similarity for further classification of which class the images belong to. Since VGG16 works based on CNN principles, an extensive description of neural networks' essential aspects will be explained. The proposed study will be based on the underlying belief that some algorithms work best in particular types of images while others work best in certain images. This enhances the hypothesis's hypothesis claiming that an algorithm's performance to detect art skills depends on the type of image. However, SIFT algorithm works better in drawn arts than VGG16, hence the best algorithm for measuring art skills.

* 1. *SIFT Algorithm*

In 1999, Lowe proposed SIFT to assist image rotation, affine transformations, changes in viewpoints, and intensity in matching features (16). The actual process involves four steps; estimating a scale-space extrema using DoG, localization of keypoints by eliminating low contrast points, orientation, and assignment of a keypoint based on local image gradient. Finally, a descriptor generator is used to compute the local image descriptor based on the magnitude and gradient of the image for each keypoint (17). As a result, this research has led to a huge revolution in the computer vision field!



**Figure 1:** Lowe's pyramid scheme.

To illustrate that, there are some fundamental phases in the SIFT method. Below are the five major phases of computation involved in generating the set of image features using SIFT method:

* + 1. Scale-space extrema detection: The first step in selecting crucial points in a picture is to identify the region of interest. This phase involves using DoG to identify invariants to scale and orient possible interest points.
    2. Keypoint localization: keypoint candidates are refined and localized by removing low contrast spots where the intensity of the pixel is calculated, and if the intensity is below a specified threshold, the pixel is rejected to further process. Edges are also detected and removed using their eigenvalues and ratios. Therefore, the low contrast and edge keypoints are removed, and only the strong interest points are taken to process further. By default, for SIFT algorithm, the contrast threshold value is set to 0.03, and the edge threshold value is set to 10.
    3. Orientation assignment: based on the gradient direction of the local image, each keypoint is assigned one or more orientations. The orientation is taken at the angles of

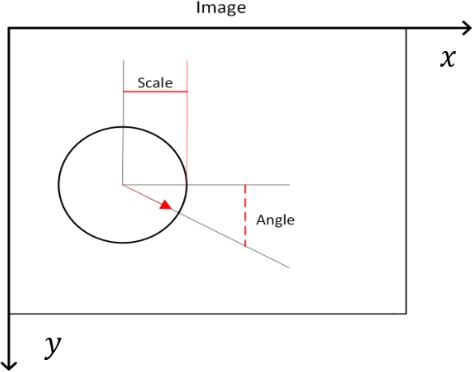
multiples of 45 degrees, so a total of 8 orientations are possible. The histogram of these orientations is calculated, which tells the frequency of these orientations in the image. The orientation with the highest peak in the histogram is considered along with the orientations of the peaks greater than 80.

* + 1. Keypoint descriptor: the local image gradients are measured at the selected scale around each keypoint. Keypoints are then transformed to allow a significant level of distortion of the local image. In other words, descriptors are vectors of size 128 achieved from the histogram. The pixel considers the neighbors in a 16 × 16 grid which is further down-sampled in a 4×4 neighbors’ grid, and each neighbor has one of 8 orientations, accounting for a total of 4 × 4 × 8 = 128.
    2. At the end, keypoints matching is done by identifying the nearest neighbors.

Tests are performed on different types of images and states. Although seeing all the results SIFT method takes more time. Nevertheless, it is more accurate. To be specific, from all the research and requirements of the project, the SIFT technique for feature matching and Neural Networks (TensorFlow Keras) is used for class detection. However, as SIFT has been used previously to improve, this KNN is used to find good matches. KNN is used to find good matches and eliminate mismatches. Moreover, both images can have different backgrounds and some noise which can decrease accuracy since an extra little line can have keypoints that can defect the results. A combination of OpenCV functions is used to pre-process both pictures to remove noise and obtain only the needed sketch.

Generating a “scale space” makes it possible. A “scale-space” is the product of the convolution process of a Gaussian blur at a different scale within the image acting as an input and is given by the function L (x,y,σ) whenever there is a blurring of an image while using the Gaussian function, the concept of gaussian blur is said to be occurring. Notably, Gaussian blur is a common

concept in graphic software (5). More details on SIFT algorithm image matching technique will be discussed.



**Figure 2:** SIFT keypoints are circular image regions with an orientation.

The process of implementing the algorithm to detect image similarity will also be demonstrated. The process involved filtering raw images using Euclidean distance and extracting features regarded as keypoints. SIFT keypoints is a circular section in an image that shows orientation. SIFT keypoints describe the geometric frame of the four parameters of an image; the radius and orientation of an image are demonstrated by the center of x and y coordinates (17).

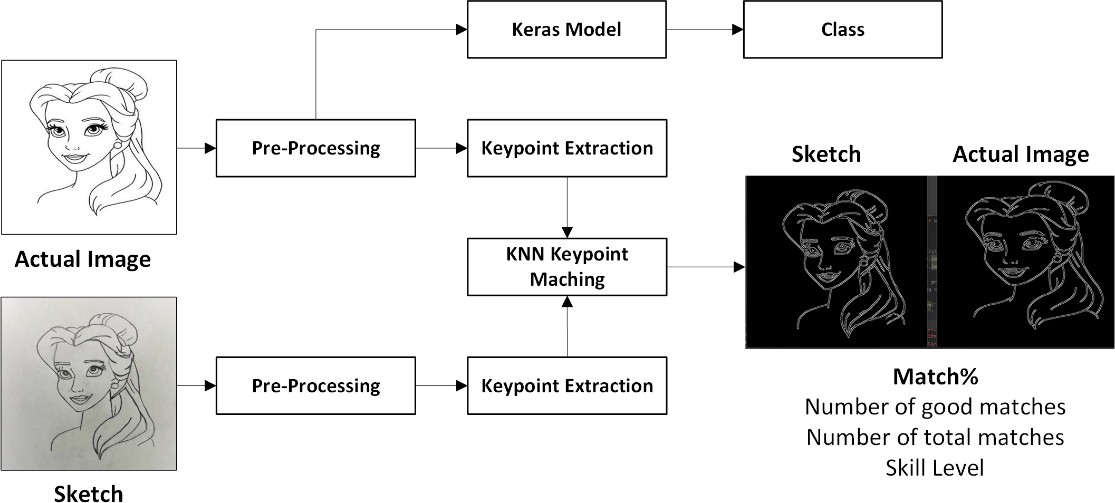
Keypoints in SIFT algorithm give accurate results compared to other techniques because of the keypoints represent the scale and orientation invariant. For implementation, a set of already available images is available in local storage, which these images are used to get keypoints. This set of images is available locally to fulfill the model, as there is always an image on which the sketch is based. To get similarity, the model is fed with the perfect original sketch, and the sketch we want to calculate similarity. Then, the keypoints of both images are calculated and further processed to get similarities (18). Compared to HOG features, SIFT features have a significant benefit because they are not affected by the size and orientation of an image.

SIFT characteristics may be applied to virtually any job that needs the detection of matching spots between pictures. For example, the authors of this publication (6) offered a novel strategy for using SIFT descriptors for multiple item identification applications. The suggested multiple-item detection techniques are evaluated on aerial and satellite photos.

In conclusion, SIFT is a feature extraction method that offers a number of benefits and is regarded as a useful and approachable tool. Reducing the dimensions of the feature space by avoiding redundant features has a substantial impact on how machine learning is performed in many applications.

*3.2.1 The multi-stage framework of SIFT model*

Practical guidance on how to carry out a design study is provided using a multi-stage framework. The framework involves many stages categorized into three major sections: detection of classes, art skill level detection, and classification of art skills. **Figure 3** below illustrates the framework of SIFT tool.



**Figure 3:** Framework of SIFT model.

1. Detection of classes

In this model, 10 classes of different cartoon characters are being utilized in our model. Those characters were chosen since they will be familiar with the participants and do not require

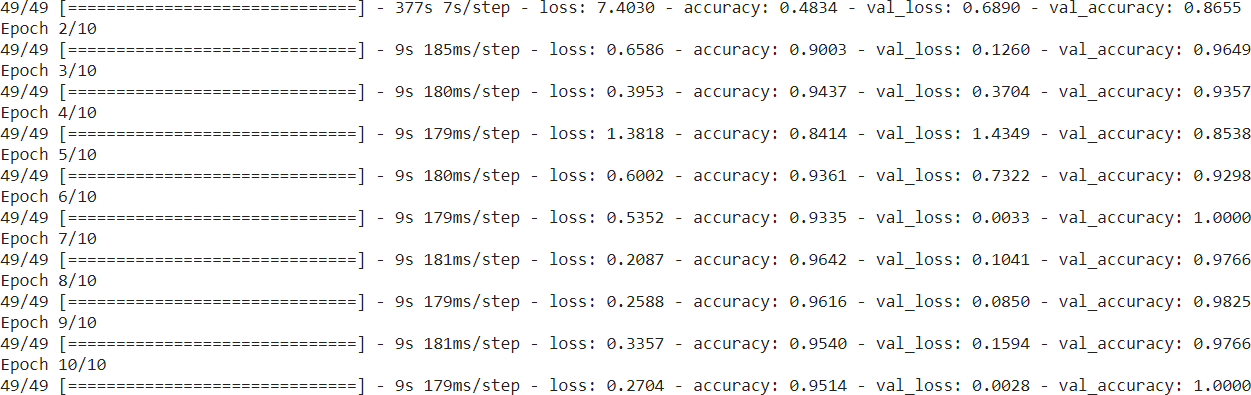
much time to draw any of them. The following steps are used for class detection:

* 1. *Data collection and classification*

Data collection is the first step to training a model, and images and sketches from all classes are gathered and put in their respective folders. Then, in the end, there is only one folder with ten folders, and all folders are named according to their classes.

* 1. *Data training*

Data consists of different graphics; therefore, using Google Collab greatly results in training the GPU. After 10 Epochs Model, accuracy has reached 100%, which can be seen below:



**Figure 4:** Model Training.

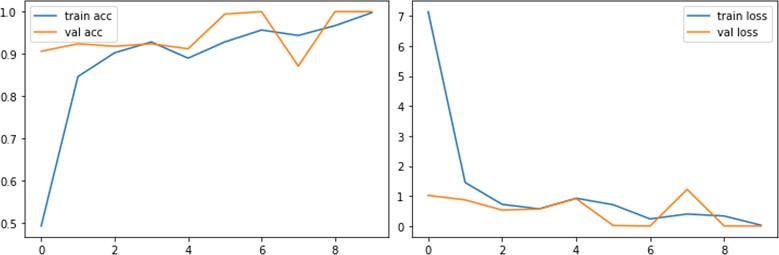
* 1. *Export model*

The training model was exported after training, which has an extension of .h5, and saved

locally.

* 1. *Training graph*

When training is started initially, we have more data loss as weights are not that accurate. However, as the number of training cycles (epochs) increases, data loss decreases, and accuracy increases. Trends for our model are given in the next paige.



**Figure 5:** Accuracy and data loss trends.

* 1. *Class detection*

For class detection model is loaded into the program, and a prediction is made. The input image to the model is our original image which also is used to compare with the artwork.

1. Art skill level detection

For skill level detection, SIFT 2D features and Flann Based matcher with KNN is utilized for skill level detection. The following steps are used for skills level:

* 1. *Pre-Processing*

Prior to matching both images, there are pre-processed steps for finding keypoints. As both images are different and require a different level of pre-processing, a GUI window enables users to change thresholding levels in real time. The following steps are included in pre-processing of the image:

* + 1. Resizing images: the image is resized to be of size 500 × 500.
    2. Grey scaling: the image is converted from RGB to greyscale to reduce the color channels from 3 to 1 and minimize the processing.
    3. Gaussian Blur: The Gaussian Blur filter is applied to the image to blur the image in order to smoothen the edge detection process, dilation, and SIFT calculation.
    4. Auto canny function calculated using the Sigma value of an image: the canny edge detection is applied to the image for converting the image into the binary image with edges

in the white color.

* + 1. OpenCV trackbars to change image detail level in real-time in case of auto canny does not provide accurate results.
    2. In black and white images, broken lines or gaps are filled using Erode Function.
    3. Morphological Operations such as erode and dilate are used where value could be changed in real-time to suppress noise.
    4. Finally, an image is returned and passed to the SIFT algorithm.

The following snippet contains the pre-processing steps performed on the images:



**Figure 6:** Pre-processing Steps.

* 1. *Extracting Keypoints*

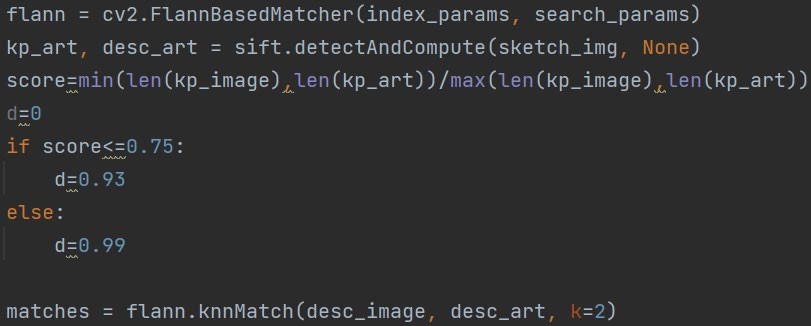
SIFT 2D features are used for detecting and computing the keypoints of images. In this part, we will use the nearby pixels' alignments and magnitudes to create a unique fingerprint for this keypoint called a 'descriptor.' Following is the function used to calculate keypoints:



**Figure 7:** Keypoints calculating formula.

* 1. *Matching Keypoints*

By locating their nearest neighbors, keypoints between two photos are matched. In other circumstances, the second-closest match may be extremely close to the first. For example, it might happen due to noise or other causes. Nonetheless, the closest distance to the second-closest distance percentage is used, which is 98%; if it is more than 98% match, it is discarded. Keypoints of both images are matched using Flann Based Matcher as seen below:



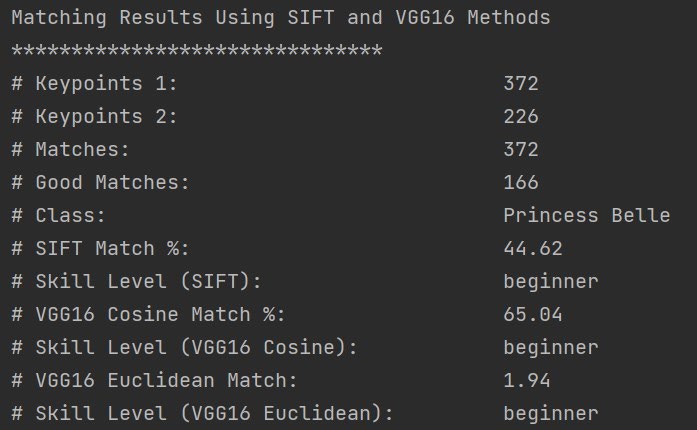
**Figure 8:** Flann-Based Matcher.

All matches are also shown in the plot to infer the results. Moreover, these matches help to judge the result since some noises in the image could be seen as a match. However, it is not so that it could assist in making pre-processing step better.



**Figure 9:** SIFT Matches.

In the previous figure, it can be seen how noise can cause inaccuracy. Furthermore, acceptable matches are calculated based on KNN distance, giving unique results. For example, the following result is shown with great matches:

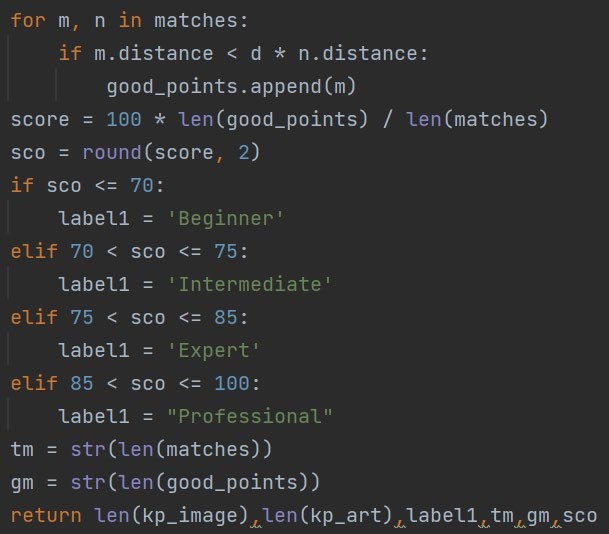


**Figure 10:** With KNN, good matches are calculated.

Without a great match, results are always between 90-100%, which is not accurate in all cases.

1. Classification of art skill

This step takes two images and calculates the keypoints and descriptors for both images. Then, the descriptors are passed to the calculateMatches() method, which computes the points similar to two images. Based on the number of matches, a score is calculated, which tells how similar or different the images are from each other. Next, the “getPlotFor()” method takes the images, keypoints, and matches to concatenate the images and draws the lines between the matched points on the images. Based on the calculated score, the art skill level is determined and printed on the screen along with the concatenated image with similar points. Based on match percentages, art skills levels are calculated following these roles as shown in the following figure:

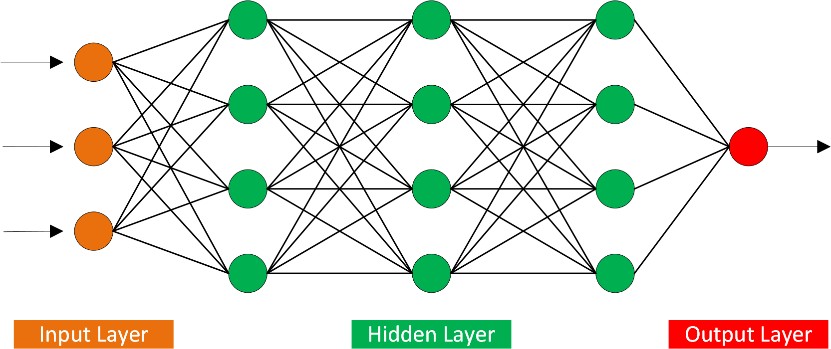


**Figure 11:** Skill levels roles.

* 1. *Siamese Network-based approach*
     1. *Convolution Neural Networks*

As the name implies, Neural Networks were designed to learn and act like the human brain. Neural Networks is a deep learning algorithm that combines a series of algorithms that mimics brain operation (4). Kunihiko Fukushima first put up the concept of convolution in 1980 under the name Neocognitron. In order to train the settings of a deep neocognitron so that it might learn internal representations of incoming data, Fukushima suggested various guided and unstructured learning algorithms (19). Yann LeCun's 1998 paper, which first introduced convolutional neural networks as we know them today, is credited with this. To recognize handwriting, LeCun suggested a CNN named LeNet (20). Alex Krizhevsky utilized a CNN model known as AlexNet to win the ImageNet Large-Scale Visual Recognition Contest in 2012. Since Krizhevsky trained AlexNet on GPUs, CNN models could potentially be developed more quickly, sparking increased interest and new work involving CNNs (21).

This structure contains specific parameters that may be adjusted to perform certain tasks. Each layer in a neural network serves a specific function, and the simplest form of a neural network has three layers, as depicted in the figure below.



**Figure 12:** Two hidden layer Siamese network for binary classification with logistic prediction.

Twin networks are created by duplicating the network's structure in the top and bottom regions. The twin networks share weight matrices at every layer. These layers are made up of nodes, each with a unique purpose. Depending on the needs, a neural network might include multiple hidden layers (19). For example, the input layer takes input signals, transmits them to the subsequent layer, and gathers data from the outside environment. All of the calculation's back-end processes are managed by the hidden layer.

There can be no hidden layers in a network. On the other hand, a neural network contains at least one hidden layer and the ultimate result of the hidden layer's computation results in the output layer (20). The basic operation of all neural networks is the same. With the training data, we must set some input parameters. All these input parameters have weights, and they decide our results. One more important parameter of the neural network is the activation function, which decides whether to activate neurons or not. This means whether the neuron’s input is important or not during the prediction process (21).

The following learning process of neural networks is used to make a perfect prediction:

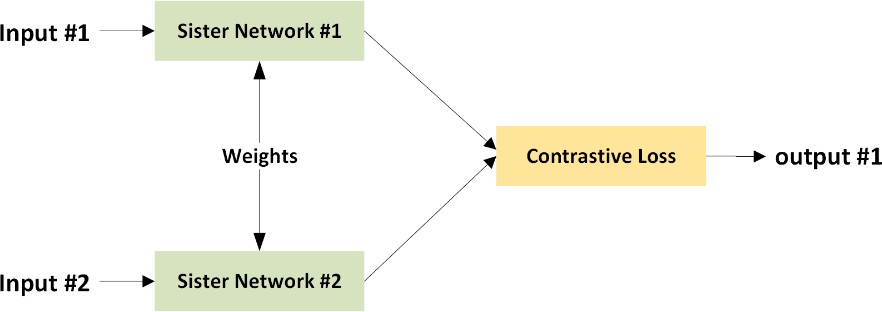
* + - 1. Weights are initialized.
      2. For each input and weight, the output is predicted.
      3. All these predictions are used to calculate the accuracy of the model.
      4. The gradient for each weight is calculated. Which predicts how changing weights will change the loss.
      5. All the weights are changed according to step 4.
      6. Then step two is repeated with new weights.
      7. These steps are repeated until the final decision. Which is also called several epochs.
      8. *Siamese Networks*

Bromley et al. introduced the technology of Siamese networks in the early 1990s. During the making, the technology was meant to assist in signature verification through image matching (22). This technique has two main characteristics:

* + - * It guarantees that its forecasts are consistent. Since each network computes the same function, weight tying ensures that two highly similar pictures cannot be translated to very different places in feature space by their networks.
      * The network is symmetric, if we show two different pictures to the twin networks, the top conjoining layer will calculate the same metric as if we showed the same two photos to the opposite twins.

The Siamese neural network, also known as the convolutional neural network or twin neural network, comprises two conjoined twin networks that receive different inputs but are connected by an energy function at the top (12). However, the twin networks take in many inputs and are connected at the top by an energy function (**Figure 13**). Some metrics between the highest-

level features represented on either side are generated by the function. The twin networks' shared parameters are connected.



**Figure 13:** Siamese network architecture.

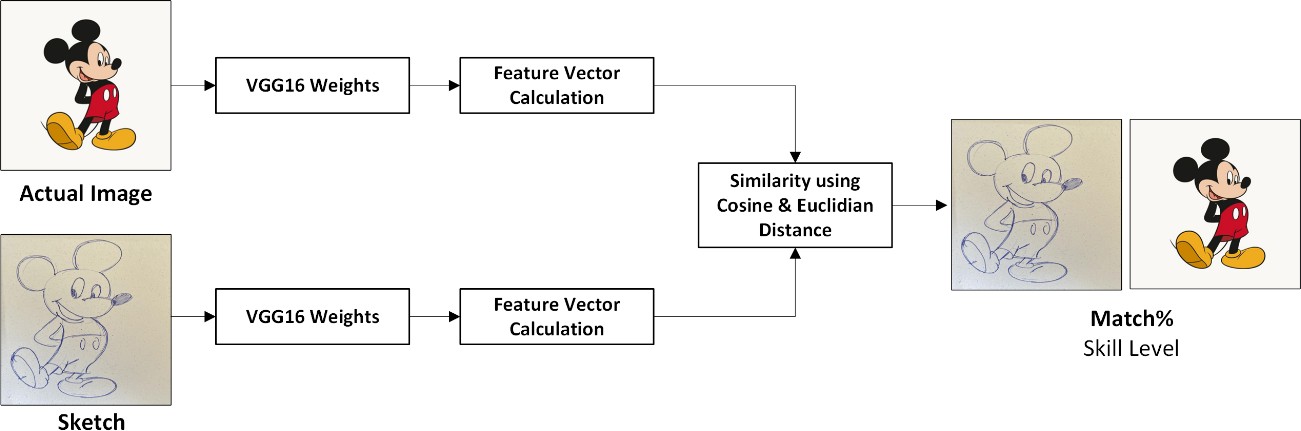
One of these networks receives each picture in the image pair. Similarity measures can be employed where a twin network would be useful, such as identifying handwritten checks, automated recognition of faces in camera pictures, and matching searches with indexed texts (23). Face recognition is likely the most well-known application of Siamese networks, in which known photos of individuals are efficient and matched to an image from a turnstile or similar.

Siamese networks are used to identify the similarity of input. The process involves comparing the feature vectors of the inputs. In this project, we will be using the VGG16 convolutional neural network model for ImageNet. Based at Oxford University's Department of Science and Engineering, the Visual Geometry Group or VGG, as it is more often known is a research group. Starting with VGG and progressing through VGG16 to VGG19, it has developed a variety of convolutional network models that can be used for image classification and biometrics. The image identification and classification algorithm VGG16 can detect the images with an accuracy rate of 92.7% using 1000 images from 1000 different categories (11). It is a well-liked and user-friendly system for sorting images that uses transfer learning.VGG16 is commonly used in learning applications due to its advantages over other available options. In 2014, VGG16, a CNN

architecture, was used to win ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (11). However, until now, VGG16 is termed one of the best vision architectures.

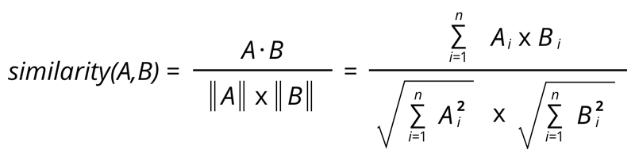
The pre-trained VGG16 model uses the Cosine similarity to calculate vectors for both input and output. When a Siamese network is used to compare images, images to be compared are passed through subnetworks that share the weight. As a result, images that are categorized as the same class have identical 4096-dimensional representations. The output feature vector generated from each subnetwork is combined through subtraction, and the outcome is passed through an operation fully connected with a single output. A sigmoid operation converts these values to a probability ranging from 0 to 1, indicating whether images are similar or dissimilar. During training, the network is updated using binary cross-entropy loss that stands between network prediction and the true label (13). VGG16 consists of 41 layers, as well as it includes over 135 million trainable parameters, and its accuracy reaches 92.7% using ImageNet weights.

* + - * 1. *The VGG16 approach using Cosine & Euclidean similarity*

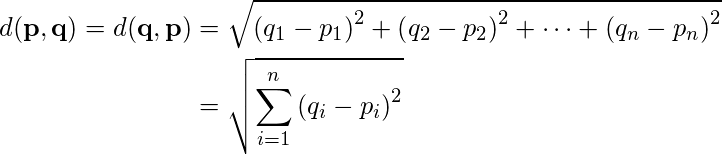
The proposed work finds the similarity between two images by employing transfer learning.

# **Figure 14:** The VGG16 using Cosine and Euclidean methods.

It calculates the percentage of similarity between two images using Cosine similarity (**Figure 15**) and Euclidean distance (**Figure 16)**, as shown in the next page.



**Figure 15:** Cosine Similarity Equation.



**Figure 16:** Euclidean Distance Equation.

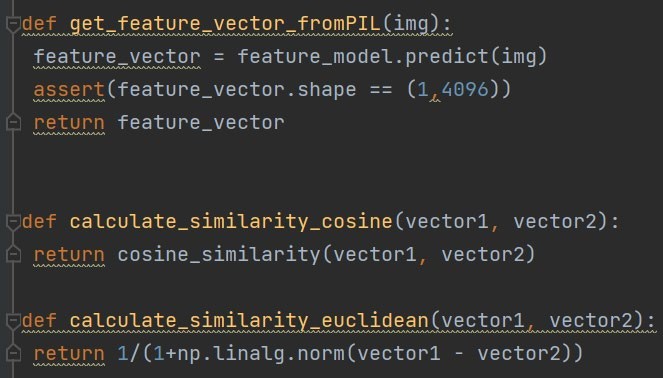
Euclidean Distance Equation uses the concept of transfer of learning to extract features from an image and calculate the distance instead of utilizing a histogram to extract features. Euclidean Distance Equation involves the application of the pre-trained model to an existing problem by making minor adjustments to fit the required problems. Given that the model involves a large number of parameters, the process is time-consuming (24). Therefore, models such as VGG16, InceptionNet, and ResNet50 are trained over a huge dataset to make them work better.

The implementation of the problem statement has been performed using the Keras API of the TensorFlow library by Google. The API contains these pre-trained models, which anyone can access through the API and use according to the required need. After downloading the weights and the architecture, the model can be fine-tuned as needed. By default, the layers are not frozen; if the model is trained, the whole weights will be trained. Therefore, the layers are usually frozen first before starting the retraining over the dataset.

The Convolutional Neural Networks extract the features from the images by convolving the different filters of a specific size over the whole image. These filters extract the features like vertical and horizontal edges of objects from the images. After the feature extraction, the similarity between two images can be calculated by finding mathematical measures like Cosine similarity or

Euclidean distance (25). These measures calculate the distance in percentage, and different labels are returned based on the percentage that is calculated, such as beginner if the image similarity percentage is under 70%, intermediate if the similarity is 70% ≥ 75%, expert if the percentage is 75% ≥ 85%, and professional if the images are 85% ≥ 100%.

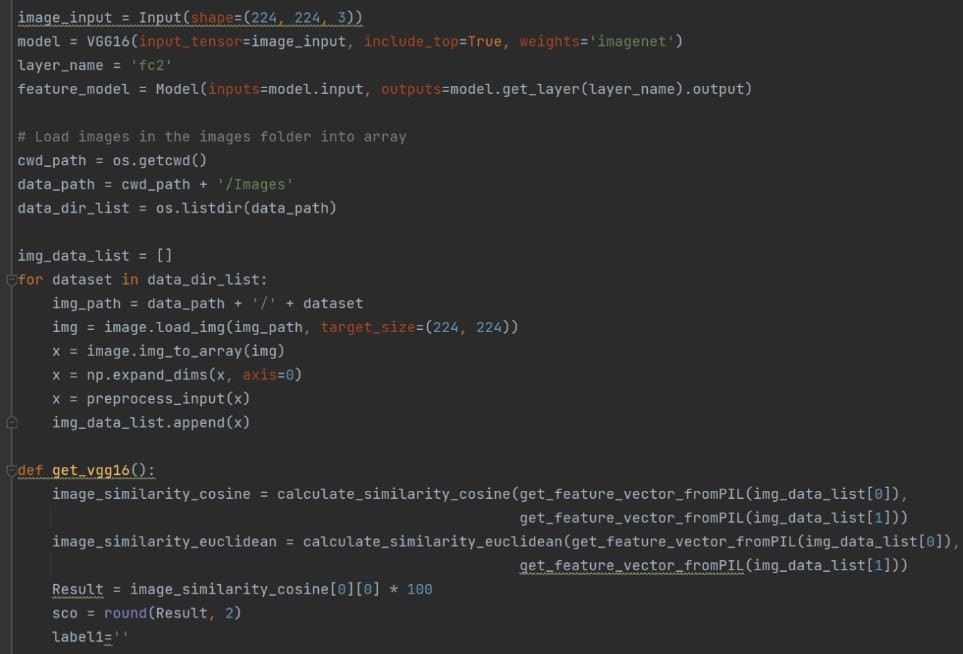
The implantation of VGG16 that we use to find image matching is explained below:



**Figure 17:** Three functions of the VGG16 model.

The above code snippet contains three functions used in the proposed methodology. The method “get\_feature\_vector\_fromPIL” receives as an argument an image, passes it to the deep learning model VGG16 to predict the class (label/target value) of the image, and creates a feature vector.

The other two functions, named “calculate\_similarity\_cosine” and “calculate\_similarity\_euclidean”, take two feature vectors as the parameters of the function and return the similarity and distance between those images accordingly.

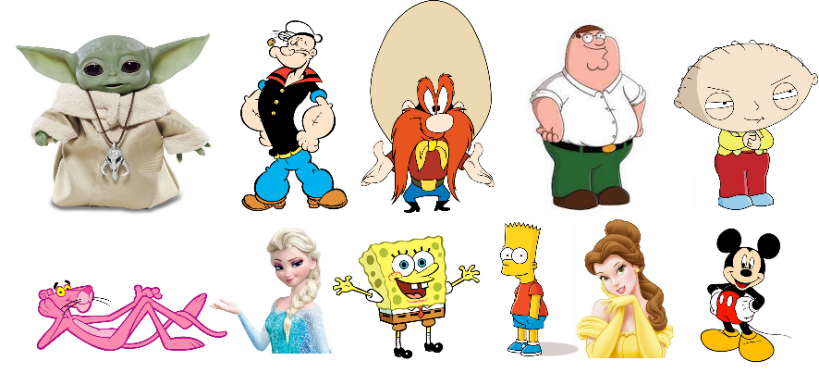


**Figure 18:** Other functions to find image similarity by VGG16.

The above code snippet first defines the model by creating an object of the pre-trained VGG16 model imported from the Keras API by defining the input shape and importing the weights of the models that are trained on the ImageNet dataset. Then, the code uses the “os” module of the python packages to look for all of the images in the specified folder. The image is loaded, converted to the array, and then reshaped and appended to the list of all of the images in a python list. Then the similarity calculation methods created in the upper code snippet are bypassing the two images to the function, however, the images are first passed to the “get\_feature\_vector\_fromPIL” function returns the feature vectors, and those feature vectors are then passed to functions to calculate their similarity. Finally, the code checks for the percentage of the similarity that is returned by the functions and uses the conditional statements to print the right labels according to the thresholds shown previously.

*3.3 Data collection*

This section highlights the process used to gather drawn pictures from source participants, which would help us test the accuracy of our model. During our testing, we avoided sketches that contained negative values of factors such as noise, distortion, and sharpness to enhance accuracy during the process hence quality data. One hundred two people participated in drawing 189 sketches, and the dataset was collected through in-person and online interviews. As mentioned, there were ten classes of cartoon characters that included many pictures for each class, such as SpongeBob, Pink Panther, and Baby Yoda. In addition, individual differences between participants were considered by choosing uncomplicated images to draw them.



**Figure 19:** The collage of some images that we used in this research.

We randomly selected 50 sketches for research to test through our two models. In addition, 25 art experts from several countries have cooperated to assess those sketches to compare human and AI assessments.

CHAPTER 4 RESULTS AND DISCUSSIONS

* 1. *General Overview*

Chapter 4 presents the experimental results and discussion. The proposed project implements the calculation of image similarity using SIFT algorithm as the feature extraction method. Siamese Network based on Keras Deep Learning Model (VGG16) pre-trained model that uses Cosine and Euclidean distance is used to find similarity and compare it with SIFT. Further, to classify the class of the images, Keras Transfer Learning is used. However, the system is unable to identify both input pictures that contain the same character when a sketch results around the beginner level of art skill, then the system finds some difficulties to detect that class. Nevertheless, the results of the experiment were synchronous with the expected as well as it is important to pay attention to an image dataset, which consists of images representing a typical real-world scene. In some cases, the image size is required to be modified to promote the efficiency and rate of algorithm performance. Neuroscience considers object recognition among the fascinating abilities possessed by humans. Humans can easily generalize by observing a set of objects and separating various features in each object (15). Human measuring was used to categorize the results of the algorithms used as beginner, intermediate, expert, or professional.

* 1. *System Specifications*

The system takes two images as input, then compares both images with multiple methods and gives a judgment based on similarity percentage, providing us with the skill level. Another function of the program is to detect the class of each art out of the total ten classes only selected for the system. The system works on already captured pictures and a set of original pictures.

* 1. *Operational Parameter*

The following are the operational parameters of our complete system:

* + - Properly configured OpenCV.
    - Accurate model for class detection.
    - DNN models with high accuracy.
    - White background for images.
    - Visible art so that it could be detected by the system easily.
  1. *Failure modes and actions on failure*

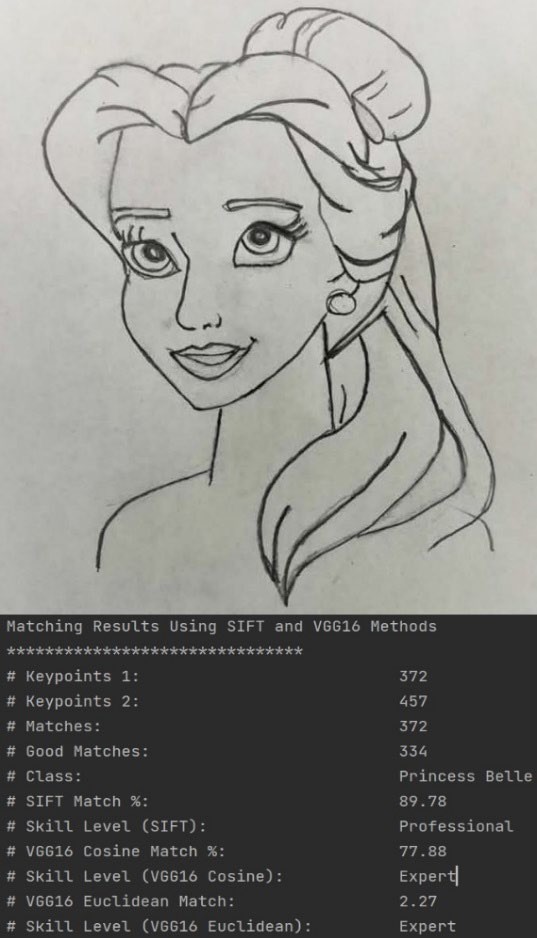
The Table is shown below:

**Table 1:** Failure modes and actions on failure.

|  |  |
| --- | --- |
| **Failure** | **Solution** |
| Different Background or Color | Used canny edge detection |
| Canny edge detection parameters | Built Auto Canny Parameter function |
| Can’t verify Auto Canny Parameters | Change Auto Canny Parameters in real-time |
| Color is a feature of VGG16 | Used black and white processed image |
| Noise becomes keypoint for SIFT | Pre-Processed image |
| Unwanted keypoint disturbing results | KNN use for only good keypoints |
| Double-lined sketches produce noise | OpenCV Morphological Operations to reduce  noise in real-time |

* 1. *Limitations and Restrictions:*

1. Poor lighting conditions were not considered during experiments.
2. White background is preferred for art.
3. Sketches should be according to the provided art.
   1. *Feature Detection Results*

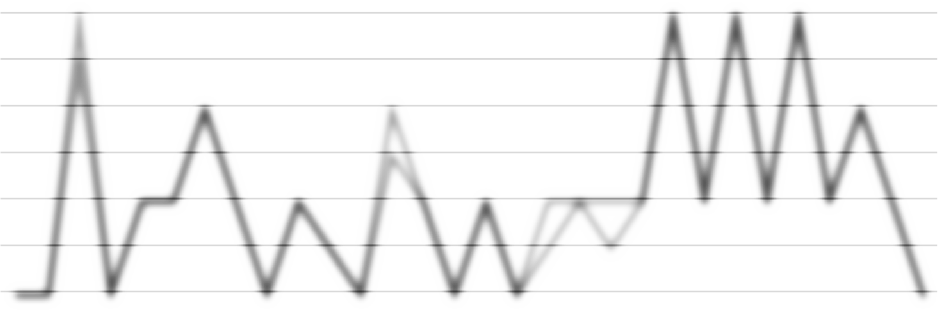
During the implementation of the features, it was found that SIFT can detect more features (Keypoints Detection) compared to VGG16 (Feature Vectors). The results of these features are shown in the first part of the excel sheet attached. It is not a surprise that SIFT detected many features in the image as it is proven to be a robust detector, as seen in **Figure 20**. Using arrows to represent the number of features detected by the SIFT approach, we can assert that the approach gives plenty of features. However, it is worth noting that SIFT model detects features in areas that do not seem to have sufficient

information in most images. In this scenario, parts of the image considered good features are edges, highly

**Figure 20:** Matching Results.

contrasting areas, and corners. As evident in **Figure 20**, many features were detected in the edges and corners of the image. On the other hand, very few features could be detected around the cheek, forehead, and nose. An exception to note, in some cases, SIFT does not detect excellent features.

The art assessment system outperforms the few systems available in the market. Auto canny systems work fine, and real-time trackbars are an additional feature if users think it is insufficient to meet the criteria. Random 30 results are plotted, showing that SIFT is closest to human results, as shown in **Figure 21.**



**SIFT- Human Match**

4.5

4

3.5

3

2.5

2

1.5

1

0.5

0

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

SIFT

Human

# **Figure 21:** Comparison between results from human judgment and results from SIFT algorithm

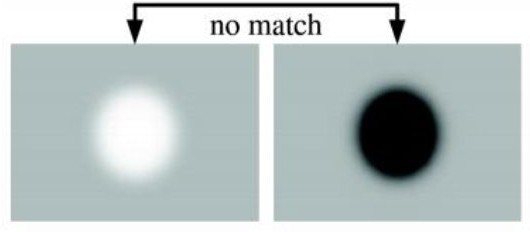
Although SIFT has some drawbacks, the gradients of each pixel in the patch need to be computed, and these computations cost time. However, it has more advantages as follows:

1. It is more accurate than any other descriptors.
2. It is rotation and scale invariant.
3. Provides correct assessment that is close to human rates.
4. SIFT works with any image irrespective of color, however, for VGG16, color is the feature that affects the results and is less required.
5. It is not affected by rotation, scaling, or image brightness.

The above part of this chapter discussed keypoints detection in SIFT approach. However, it should be noted that the number of keypoints should not be the sole measure of the algorithm's effectiveness. First, it is necessary to analyze the number of robust keypoints to survive further filtering. Then, I will offer a discussion on the performance of some images. The results of this are compiled in the form of an excel worksheet attached.

1. Most keypoints were detected in image No. 23 by SIFT. The high number of keypoints was expected due to the image's large amount of texture. On the other hand, the lowest detection was found in image No. 15. This result complies with the hypothesis of SIFT performance which claims a higher performance on textured images.
2. Though VGG16 is also a textured-based matching algorithm, it demonstrated confusion in textured images with changes in illumination. As a result, VGG16 does not offer satisfactory performance with the pair. For instance, the image that showed the most features in the Cosine match was image No. 10, and the image that showed the lowest features in the Cosine match was image No. 27. Based on the results, it can be asserted that VGG16 is a texture-based algorithm and does not perform well when tested on a planar image.

The experiment aimed for accurate feature detection to facilitate matching. During feature matching, the features detected in an image are tested each at a time to ensure features detected to contain the same information as the database holding all the values. To maintain accuracy as well as the effectiveness of the matching process, the steps outlined in section 3.2.1of the methodology section were followed. Images (**Figure 1** and **Figure 2**) showing the image detection used for SIFT are in section 3.2 of the methodology. Below is a discussion of algorithms that matches performance for images:



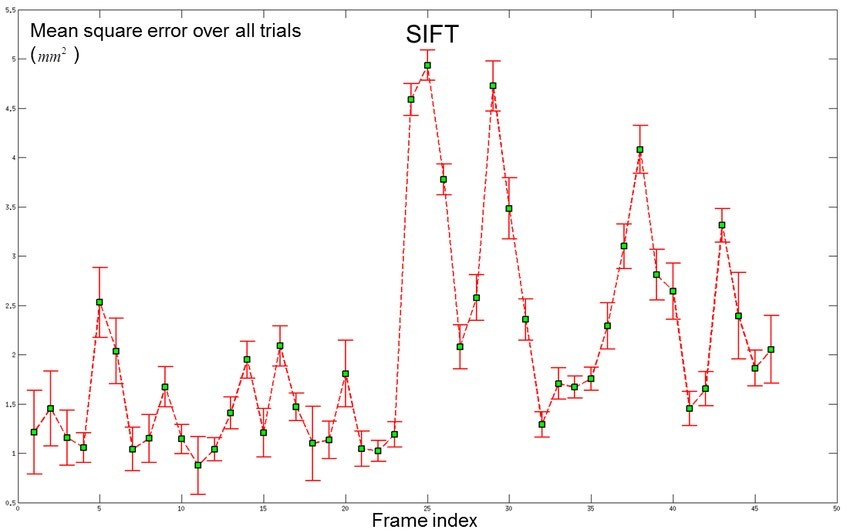
**Figure 22:** A value is not considered to be a valuable match if there is a difference in contrast between two points (light on dark and dark on light background).

SIFT, as earlier mentioned, detected keypoints, while VGG16 did not since it works based on feature vectors, which was in line with expectations. SIFT detected above 95% of keypoints that related to each sketch and 70% of good matches with their actual images. To illustrate that, in the attached sheet it can observe that in the picture No.3, the keypoints of the actual image were 372 and 452 in the sketch, as well as the good matches, reached 327; therefore, that indicates the high performance of SIFT detector. VGG16, this model recorded lower matches found among the images. VGG16 does not have an automated feature-matching extraction. Therefore, very few matches could be detected in most images. However, this does not ascertain a conclusion since measuring accuracy and efficiency rather than the quantity of the same is necessary. Generally, compared to the number of features detected, matching was low. This insinuates an argument that the number of matchings cannot be considered good by itself. Therefore, there is a need to compare the number of features detected to take note of the percentage of the features used in the algorithm. The excel attached contains the matching from SIFT and VGG16.

Of more importance is the quality of the matches and not the amount. We measured the accuracy of the matches to offer an analysis. This was done by matching the component of the algorithm while comparing it with the automatically detected matches. After automatically detecting the matches, the accuracy of the same could now be analyzed visually. Human judgment will be used as a point of reference when measuring the accuracy and effectiveness of the algorithms.

The results obtained from the experiment were assessed based on the number of errors incurred during the process of accurate matching by the algorithm. As a measure of success, errors will be categorized into type I and type II. Type I error occurs when the algorithm fails to detect real matches. In this case, the algorithm recognizes both the original image and the drawn art as the same. Type II error is involved when the algorithm mismatches a feature. Typically, such errors appear as crossing lines between matches. A discussion on errors will be explained below.

To assess type I errors, the number of matches the algorithm failed to identify as matches compared to those that had manually been identified was determined. We aim to ensure a lower number of type I errors. A high number of type I errors would indicate a failure in the algorithm used. In essence, a lower number of type I errors is preferred to preserve the accuracy of the algorithm. After collecting and analyzing the results, a comparison between the values arrived from each approach (SIFT and VGG16) are evaluated. The comparison will be necessary for deriving and shaping a conclusion for this research. **Figure 23** is an illustration of the performance of SIFT descriptor-based image matching. The x-axis represents the frame index, and the y-axis represents the number of matched SIFT features.



**Figure 23:** Performance of SIFT descriptor-based image matching.

Comparing algorithms in SIFT and VGG16 in terms of the amount of type 1 and type II errors, SIFT demonstrated more errors compared to VGG16. This is because the many features detected in SIFT were not as accurate as the few features that arrived from VGG16. It is important to note that this analysis does not compare the image types considered most suitable for one

algorithm. SIFT and VGG16 are texture-based algorithms that work similarly with all texture algorithms. Therefore, it can be concluded that the general features detected by the algorithms cannot accurately provide good enough features. This is due to the number of features detected in insignificant areas of the images.

Of more importance when it comes to matching is the quality of the matches. Measuring the accuracy of matches was one way of determining the quality of the matches. After automatically detecting the matches, a visual analysis of the images was conducted, and a comparison was made with the manual matches conducted earlier. The process involved in matching highly depends on the features detected. For good matching, it is expected that all the matches be visually or manually detected. In cases where the algorithm fails to detect manual matches, the condition is considered that the matching component of the algorithm used does not interfere with accuracy.

On the contrary, a mismatch is counted as a type II error. In this experiment, the amount of type II errors detected matters more than the amount of type I error. Type II error represents a condition of matching two features with no match in the image pair. On the other hand, type I error reflects the weaknesses in the algorithm used when it comes to detecting all the matches. Type I error reflects a failure in the algorithm regarding matching features without necessarily being conjugate points.

Furthermore, since these keypoints are scale- and orientation-invariant, they could provide results with more precision than alternative approaches. For implementation, a collection of pre- existing photos is kept in local storage and utilized to get keypoints. Since the sketch is always based on a picture, this set of photos may be found locally to complete the model. To get similarity, the model is fed with the original sketch, which is perfect, and the sketch we want to calculate similarity. Keypoints of both images are calculated and further processed to get similarities. Because of they are unaffected by picture size or orientation, SIFT features offer a substantial advantage over edge technique or HOG features. An overall accuracy of 80%-98% was attained

using SIFT. This equated to a presumption that the samples' descending portions could be accurately categorized using this method. SIFT strategy was that SIFT is arguably the most preferred algorithm used to match under different scales, lighting, and rotation. However, compared to VGG16, SIFT is relatively slow. A wide range of implementations is available on the web as open-source codes. This study compiled different implementations, such as implementations published by Bruno, Greco, and La Cascia. This implementation was the most applicable and aligned with our study's results (5).

Various image and state categories are the subjects of tests. Even though it takes longer to examine all the findings, the SIFT approach is more accurate. To be specific, from all the research and requirements of the project, the SIFT technique for feature matching and Neural Networks (TensorFlow Keras) is used for class detection. However, since SIFT has previously been used to improve this KNN is used to find good matches. KNN is utilized to identify good matches and eliminate mismatches. Moreover, both images can have different backgrounds and some noise which can decrease accuracy since an extra little line can have keypoints that can defect the results. A combination of OpenCV functions is used to pre-process both pictures to remove noise and obtain only the needed sketch.

* 1. *State of the Art*

There are two common categories of most essential object recognition approaches.

* + - Feature-based algorithms.
    - Geometry-based approaches.

The research made use of feature-based as the primary object recognition approach. Object recognition algorithms are based on appearances. Nixon et al. developed an algorithm that would recognize an overlapped image. The dimensions of the extracted features account for hundreds in the recognition system based on view. 3D recognition can be compiled from the perspective of

pattern recognition (26). The appearance of an object is based on features such as the pose of the scene, shape, illumination condition, and reflectance properties of the image. Karami et al. (2015) also developed an object recognition system that used SIFT descriptors features that are constant, regardless of the scaling, rotation, and translation of an image. SIFT descriptors are also partially variant to the illumination changes and 3D projections (7). Algorithms in SIFT are detected using a filtering approach that can extract stable points in scale space. Keypoints in SIFT are used as inputs. Many authors have also proposed a method that can perform object pose estimation.

Our method is unique in that the algorithm is based on visual models. After completing the recognition phase, 25 experts are involved in rating the art as either beginner, intermediate, expert, or professional. The art with a higher percentage of professional ratings and lower rating in beginner ratings would be considered the best. As shown in the excel attached, art 22 is considered professional art.

CHAPTER 5 CONCLUSION AND FUTURE WORK

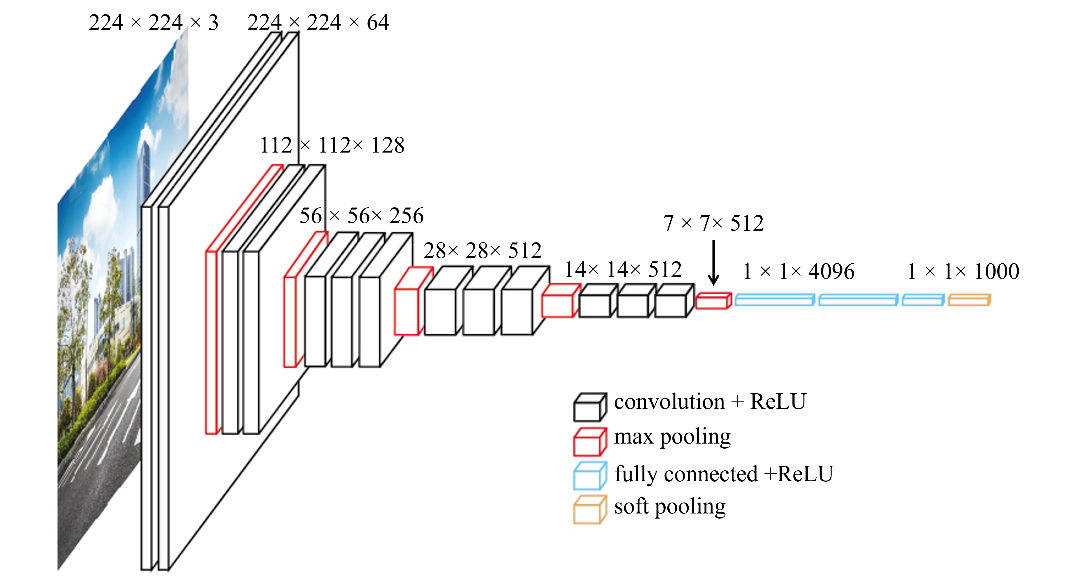
Chapter 5 concludes this thesis and paves the way for future research. In summary, from the results obtained from the study, it can be concluded that:

We can teach AI to assess human drawing skills. The number of features detected in either of the models, SIFT or VGG16, depends on the type of image used. SIFT is a texture-based detector that detects more features in highly textured images. VGG16 is more accurate in performance where SIFT and SURF fail. This translates to the assumption that VGG16, a feature-based algorithm, performs better when tested on images containing planner objects. The algorithm's success is not determined by the number of features detected but by the quality. The number of features detected is proportional to the number of matches. The algorithm used in SIFT was able to detect more features in all images. The ability of an algorithm to detect more matches is directly proportional to the number of matches identified.

In all images, SIFT detected more features as well as matched all images. While matches in SIFT were more compared to VGG16, they were less effective simultaneously. The number of matches was more promising than the effectiveness ratios. Even in the case of a double line, which is supposed to be single, our system can overcome that issue too and can enhance the results. In addition, the amount of Type 1 and Type II errors in SIFT and VGG16 demonstrated that both algorithms need improvement to make them more robust.

The overall aim of this research is to develop a program that allows users to measure their art skills of art. The proposed project implements the calculation of image similarity between two images using SIFT as the feature extraction method and then using the Siamese Network based on the Keras deep learning pre-trained model (VGG16) that uses Cosine and Euclidean similarity for further classification of which class the images belong to. VGG16 is considered an excellent vision

model architecture; however, SIFT is also. VGG16 is a convolution neural network (CNN) architecture that consistently uses convolution layers throughout the architecture. For example, the 16 in VGG16 indicates that the model has 16 layers that carry weights. This creates a pretty large network with more than 138 million parameters.



**Figure 24:** VGG16 Architecture.

The SIFT model keypoints offer descriptions of the object recognition problem. Significant attention was put on a problem that comes along with the recognition of specific areas and instances in art. Notably, like VGG16, SIFT separates processing into feature extraction and matching. In feature extraction, discrete primitives are detected. In feature matching, the stored models are matched against the features detected.

The study aimed to prove the hypothesis that the algorithm's performance depends on the image, and SIFT algorithm works better in measuring the skill of art over VGG16. The program developed was expected to give a value to the level of accuracy between a photo and a sketch developed using the same. Results showed that the algorithm's performance highly depends on the type of image. However, the main problem is that the system cannot identify outliers and useful

data. Therefore, pre-processing is the major part focused on in this research. The SIFT method has the most pre-processing steps, making it very close to human results, as seen in the graphs. Experiments show that the results are mostly affected by extra lines, which are outliers. Although there are some computer vision methods to fill the gaps of double lines, the system may fail to identify useful data or noise. That’s why a GUI-based trackbar is added to the system where the user can adjust the level in real-time to keep maximum information and remove outliers. After extensive pre-processing steps, systems can reject mostly outliers and have an accuracy of above 90% as compared with human observation.

Creating a visual system that equals human cognitive ability may be difficult. A vision recognition system's development process is anticipated to handle scene illumination variations, an object's position about the camera, and generalization from a collection of exemplary photographs. No matter the lighting conditions at the time of capture or the stance, a good identification system must be able to extract and detect patterns in photographs. Object recognition models should be constructed in a more condensed manner to reduce computing complexity during the recognition phase. Additionally, the system can be upgraded to include live images captured by a camera, and multiple classes can be added. It should be noted that the representation might be 2D or 3D. The hand-drawn artwork, in this instance, was 2D.

REFERENCES

1. **Plataniotis, Konstantinos N. and Venetsanopoulos, Anastasios N.** *Color Image Processing and Applications.* s.l. : Springer, 2000. 9783662041864.
2. *Facial Self Similarity for Sketch to Photo Matching.* **Khan, Zohaib, Hu, Yiqun and Mian, Ajmal.** Fremantle, WA, Australia : IEEE, 2012. Vol. 12. 9781467321792.
3. **Chie, Yi T.** Difference between Drawing from Photo Reference vs Real Life. *YouTube.* [Online] 02 17, 2016. https://[www.youtube.com/watch?v=XIhEiP0TvGE&ab\_channel=TeohYiChie.](http://www.youtube.com/watch?v=XIhEiP0TvGE&ab_channel=TeohYiChie)
4. *Application of the residue number system to reduce hardware costs of the convolutional neural network implementation.* **Valueva, M., et al.** Stavropol, Russia : s.n., 2020.
5. **Bruno, Alessandro, Greco, Luca and La Cascia, Marco.** Object Recognition and Modeling Using SIFT Features. *Advanced Concepts for Intelligent Vision Systems.* 2013, Vol. 10.1007, 978-3-319-02895-8\_23, pp. 250-261.
6. *Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images.* **Karami, Ebrahim, Prasad, Siva and Shehata, Mohamed.** Oxford : A Treatise on Electricity and Magnetism, 3rd ed., 2017, Vol. 2. arXiv: 1710.02726.
7. *Image Identification Using SIFT Algorithm: Performance Analysis against Different Image Deformations.* **Karami, Ebrahim, Shehata, Mohamed and Smith, Andrew.** St. john’s, Canada : Proceedings of the 2015 Newfoundland Electrical and Computer Engineering Conference, 2015.
8. *Overview of Image Matching Based on ORB Algorithm.* **Luo, Chuan, et al.** s.l. : Journal of Physics: Conference Series, 2019. 199012935.
9. *Speeded-up robust features (SURF).* **Bay, Herbert , et al.** 2008, Vol. 110, pp. 346-359. No. 3.
10. *KAZE Features.* **Alcantarilla, Pablo Fernández and Bartoli, Adrien.** s.l. : European Conference on Computer Vision (ECCV), 2012, Vol. 7577, pp. 214-227.
11. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* **Simonyan, Karen and Zisserman, Andrew.** San Diego, USA : International Conference on Learning Representations, 2015.
12. *Siamese Network Features for Image Matching.* **Melekhov, Iaroslav, Kannala, Juho and Rahtu, Esa.** Cancun, Mexico : 23rd International Conference on Pattern Recognition, 2017. 978- 1-5090-4847-2.
13. *Siamese network features for image matching.* **Melekhov, I., Kannala, J., & Rahtu, E.** s.l. : IEEE, 2016, international conference on pattern recognition (ICPR), pp. pp. 378-383.
14. *Learning Deep Representations for Ground-to-Aerial Geolocalization.* **Lin, Tsung-Yi, et al.**

Boston, USA : Computer Vision and Pattern Recognition, 2015.

1. **Loy, James.** *Neural Network Projects with Python.* Birmingham, UK : Packt Publishing Ltd., 2019. 9781789138900.
2. *Object Recognition from Local Scale-Invariant Features.* **Lowe, David G.** Kerkyra, Greece : IEEE, 1999. 0-7695-0164-8.
3. *Distinctive Image Features from Scale-Invariant Keypoints.* **Lowe, David G.** s.l. : Accepted for publication in the International Journal of Computer Vision, 2004, Vol. 50. 2.
4. *Urban-Area and Building Detection Using SIFT Keypoints and Graph Theory.* **Sırmaçek, Beril and Ünsalan, Cem.** 4, s.l. : IEEE Transactions on Geoscience and Remote Sensing, 2009, Vol. 47, pp. 1156-1167.
5. **Elgendy, Mohamed.** *Deep Learning for Vision Systems.* New York, USA : Manning, 2020.

9781617296192.

1. **Davide, Chicco.** *Siamese neural networks: An overview.* s.l. : Artificial Neural Networks, 2021. pp. 73-94.
2. *Learning a similarity metric discriminatively, with application to face verification.* **Chopra, Sumit, Hadsell, Raia and LeCun, Yann.** San Diego, CA, USA : IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, Vol. 1. 0-7695-2372-2.
3. *Signature verification using a "Siamese" time delay neural network.* **Bromley, Jane, et al.** 4,

s.l. : World Scientific Publishing, 1993, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 7, pp. 669–688.

1. *DeepFace: Closing the Gap to Human-Level Performance in Face Verification.* **Taigman, Yaniv, et al.** Columbus, OH, USA : IEEE Conference on Computer Vision and Pattern Recognition, 2014. 1063-6919.
2. **Sonka, Milan, Hlavac, Vaclav and Boyle, Roger .** *Image Processing, Analysis, and Machine Vision, 4th edition.* Boston, MA : Cengage Learning, 2015. 101133593607.
3. **Michelucci , Umberto.** *Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection.* USA : Apress; 1st ed. edition (September 28, 2019), 2019. 978-1-4842- 4976-5.
4. **Nixon, Mark and Aguado, Alberto.** *Feature Extraction and Image Processing for Computer Vision, 4th edition.* Cambridge, MA : Academic Press, 2019. 0128149760.
5. *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.* **Fukushima, Kunihiko.** 4, Tokyo, Japan : Biological Cybernetics, 1980, NHK Broadcasting Science Research Laboratories, Vol. 36, pp. 193–202.
6. *Gradient-based learning applied to document recognition.* **LeCun, Yann, et al.** 11, s.l. : IEEE, 1998, Proceedings of the IEEE, Vol. 86, pp. 2278 - 2324. 1558-2256.
7. *ImageNet Classification with Deep Convolutional.* **Krizhevsky, Alex, Sutskever, Ilya and E. Hinton, Geoffrey.** s.l. : Publication History, 2012, Advances in Neural Information Processing Systems, Vol. 1, pp. 1097–1105.
8. *ImageNet Large Scale Visual Recognition Challenge.* **Russakovsky, Olga, et al.** 115, 2015, International Journal of Computer Vision, pp. 211–252.

APPENDIX - A

Data analysis sheet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 1** | **SIFT** | | **VGG16** | |
| A drawing of a person  Description automatically generated with medium confidence | **KP 1** | 372 | **Cosine %** | 68.62 |
| **KP 2** | 255 | **Skill Level** | Beginner |
| **G.M** | 150 | **Euclidean %** | 2.01 |
| **SIFT %** | 40.32 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 2** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 372 | **Cosine %** | 68.62 |
| **KP 2** | 227 | **Skill Level** | Beginner |
| **G.M** | 153 | **Euclidean %** | 2.01 |
| **SIFT %** | 41.13 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 3** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 372 | **Cosine %** | 77.88 |
| **KP 2** | 452 | **Skill Level** | Expert |
| **G.M** | 327 | **Euclidean %** | 2.27 |
| **SIFT %** | 87.9 | **Skill Level** | Expert |
| **Skill Level** | Professional |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 4** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 384 | **Cosine %** | 73.18 |
| **KP 2** | 536 | **Skill Level** | Intermediate |
| **G.M** | 148 | **Euclidean %** | 2.16 |
| **SIFT %** | 38.54 | **Skill Level** | Intermediate |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 5** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 246 | **Cosine %** | 79.24 |
| **KP 2** | 274 | **Skill Level** | Expert |
| **G.M** | 228 | **Euclidean %** | 2.41 |
| **SIFT %** | 74.68 | **Skill Level** | Expert |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 6** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 246 | **Cosine %** | 70.16 |
| **KP 2** | 214 | **Skill Level** | Intermediate |
| **G.M** | 232 | **Euclidean %** | 1.79 |
| **SIFT %** | 74.31 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 7** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 246 | **Cosine %** | 69.98 |
| **KP 2** | 271 | **Skill Level** | Beginner |
| **G.M** | 231 | **Euclidean %** | 1.8 |
| **SIFT %** | 84.9 | **Skill Level** | Beginner |
| **Skill Level** | Expert |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 8** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 246 | **Cosine %** |  |
| **KP 2** | 162 | **Skill Level** |  |
| **G.M** | 115 | **Euclidean %** |  |
| **SIFT %** | 70.75 | **Skill Level** |  |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 9** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 246 | **Cosine %** | 64.68 |
| **KP 2** | 163 | **Skill Level** | Beginner |
| **G.M** | 123 | **Euclidean %** | 1.68 |
| **SIFT %** | 50 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 10** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 250 | **Cosine %** | 82.71 |
| **KP 2** | 146 | **Skill Level** | Expert |
| **G.M** | 114 | **Euclidean %** | 2.22 |
| **SIFT %** | 75.6 | **Skill Level** | Intermediate |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 11** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 182 | **Cosine %** | 42.4 |
| **KP 2** | 364 | **Skill Level** | Beginner |
| **G.M** | 81 | **Euclidean %** | 1.13 |
| **SIFT %** | 44.51 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 12** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 256 | **Cosine %** | 64.87 |
| **KP 2** | 322 | **Skill Level** | Beginner |
| **G.M** | 222 | **Euclidean %** | 1.97 |
| **SIFT %** | 69.72 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 13** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 260 | **Cosine %** | 74.03 |
| **KP 2** | 259 | **Skill Level** | Intermediate |
| **G.M** | 227 | **Euclidean %** | 1.97 |
| **SIFT %** | 62.31 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 14** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 495 | **Cosine %** | 58.66 |
| **KP 2** | 423 | **Skill Level** | Beginner |
| **G.M** | 443 | **Euclidean %** | 1.5 |
| **SIFT %** | 84.89 | **Skill Level** | Beginner |
| **Skill Level** | Expert |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 15** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 495 | **Cosine %** | 62.9 |
| **KP 2** | 290 | **Skill Level** | Beginner |
| **G.M** | 178 | **Euclidean %** | 1.61 |
| **SIFT %** | 35.96 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 16** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 495 | **Cosine %** | 58.61 |
| **KP 2** | 266 | **Skill Level** | Beginner |
| **G.M** | 179 | **Euclidean %** | 1.55 |
| **SIFT %** | 36.16 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 17** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 328 | **Cosine %** | 74 |
| **KP 2** | 221 | **Skill Level** | Intermediate |
| **G.M** | 147 | **Euclidean %** | 2.21 |
| **SIFT %** | 74.82 | **Skill Level** | Intermediate |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 18** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 328 | **Cosine %** | 74 |
| **KP 2** | 167 | **Skill Level** | Intermediate |
| **G.M** | 134 | **Euclidean %** | 2.21 |
| **SIFT %** | 40.85 | **Skill Level** | Intermediate |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 19** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 328 | **Cosine %** | 67.42 |
| **KP 2** | 367 | **Skill Level** | Beginner |
| **G.M** | 287 | **Euclidean %** | 2.02 |
| **SIFT %** | 87.5 | **Skill Level** | Beginner |
| **Skill Level** | Professional |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 20** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 445 | **Cosine %** | 66.15 |
| **KP 2** | 173 | **Skill Level** | Beginner |
| **G.M** | 174 | **Euclidean %** | 1.86 |
| **SIFT %** | 39.1 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 21** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 445 | **Cosine %** | 66.15 |
| **KP 2** | 376 | **Skill Level** | Beginner |
| **G.M** | 400 | **Euclidean %** | 1.86 |
| **SIFT %** | 84.89 | **Skill Level** | Beginner |
| **Skill Level** | Expert |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 22** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 259 | **Cosine %** | 58.93 |
| **KP 2** | 241 | **Skill Level** | Beginner |
| **G.M** | 224 | **Euclidean %** | 1.67 |
| **SIFT %** | 86.49 | **Skill Level** | Beginner |
| **Skill Level** | Professional |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 23** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 314 | **Cosine %** | 57.48 |
| **KP 2** | 271 | **Skill Level** | Beginner |
| **G.M** | 279 | **Euclidean %** | 1.4 |
| **SIFT %** | 88.85 | **Skill Level** | Beginner |
| **Skill Level** | Professional |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 24** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 342 | **Cosine %** | 72.31 |
| **KP 2** | 157 | **Skill Level** | Intermediate |
| **G.M** | 145 | **Euclidean %** | 1.66 |
| **SIFT %** | 74.4 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 25** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 342 | **Cosine %** | 80.63 |
| **KP 2** | 214 | **Skill Level** | Expert |
| **G.M** | 159 | **Euclidean %** | 2.27 |
| **SIFT %** | 46.49 | **Skill Level** | Expert |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 26** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 214 | **Cosine %** | 78.78 |
| **KP 2** | 171 | **Skill Level** | Expert |
| **G.M** | 193 | **Euclidean %** | 2.18 |
| **SIFT %** | 74.19 | **Skill Level** | Intermediate |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 27** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 328 | **Cosine %** | 39.54 |
| **KP 2** | 268 | **Skill Level** | Beginner |
| **G.M** | 292 | **Euclidean %** | 1.44 |
| **SIFT %** | 73.82 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 28** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 227 | **Cosine %** | 56.54 |
| **KP 2** | 204 | **Skill Level** | Beginner |
| **G.M** | 197 | **Euclidean %** | 1.71 |
| **SIFT %** | 46.09 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
|  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 29** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 605 | **Cosine %** | 61.23 |
| **KP 2** | 481 | **Skill Level** | Beginner |
| **G.M** | 520 | **Euclidean %** | 1.44 |
| **SIFT %** | 85.95 | **Skill Level** | Beginner |
| **Skill Level** | Professional |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 30** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 607 | **Cosine %** | 58.55 |
| **KP 2** | 239 | **Skill Level** | Beginner |
| **G.M** | 239 | **Euclidean %** | 1.41 |
| **SIFT %** | 69.37 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 31** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 342 | **Cosine %** | 66.81 |
| **KP 2** | 234 | **Skill Level** | Beginner |
| **G.M** | 140 | **Euclidean %** | 1.79 |
| **SIFT %** | 40.94 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 32** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 313 | **Cosine %** | 66.81 |
| **KP 2** | 230 | **Skill Level** | Beginner |
| **G.M** | 116 | **Euclidean %** | 1.79 |
| **SIFT %** | 74.06 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 33** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 313 | **Cosine %** | 61.44 |
| **KP 2** | 222 | **Skill Level** | Beginner |
| **G.M** | 122 | **Euclidean %** | 1.92 |
| **SIFT %** | 38.98 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 34** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 313 | **Cosine %** | 74.85 |
| **KP 2** | 302 | **Skill Level** | Intermediate |
| **G.M** | 271 | **Euclidean %** | 2.4 |
| **SIFT %** | 74.58 | **Skill Level** | Expert |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 35** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 256 | **Cosine %** | 64.87 |
| **KP 2** | 322 | **Skill Level** | Beginner |
| **G.M** | 222 | **Euclidean %** | 1.97 |
| **SIFT %** | 74.72 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 36** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 272 | **Cosine %** | 67.09 |
| **KP 2** | 325 | **Skill Level** | Beginner |
| **G.M** | 233 | **Euclidean %** | 1.74 |
| **SIFT %** | 74.66 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 37** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 249 | **Cosine %** |  |
| **KP 2** | 228 | **Skill Level** |  |
| **G.M** | 230 | **Euclidean %** |  |
| **SIFT %** | 72.37 | **Skill Level** |  |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 38** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 683 | **Cosine %** | 66.66 |
| **KP 2** | 739 | **Skill Level** | Beginner |
| **G.M** | 596 | **Euclidean %** | 1.27 |
| **SIFT %** | 87.26 | **Skill Level** | Beginner |
| **Skill Level** | Professional |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 39** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 328 | **Cosine %** | 71.76 |
| **KP 2** | 191 | **Skill Level** | Intermediate |
| **G.M** | 152 | **Euclidean %** | 2.14 |
| **SIFT %** | 74.34 | **Skill Level** | Intermediate |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 40** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 328 | **Cosine %** | 54.86 |
| **KP 2** | 198 | **Skill Level** | Beginner |
| **G.M** | 138 | **Euclidean %** | 1.7 |
| **SIFT %** | 42.07 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 41** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 495 | **Cosine %** | 73.43 |
| **KP 2** | 239 | **Skill Level** | Intermediate |
| **G.M** | 184 | **Euclidean %** | 2.19 |
| **SIFT %** | 37.17 | **Skill Level** | Intermediate |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 42** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 495 | **Cosine %** | 55.31 |
| **KP 2** | 347 | **Skill Level** | Beginner |
| **G.M** | 188 | **Euclidean %** | 1.47 |
| **SIFT %** | 74.98 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 43** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 246 | **Cosine %** | 66.13 |
| **KP 2** | 258 | **Skill Level** | Beginner |
| **G.M** | 215 | **Euclidean %** | 1.7 |
| **SIFT %** | 74.4 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 44** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 601 | **Cosine %** | 66.13 |
| **KP 2** | 404 | **Skill Level** | Beginner |
| **G.M** | 311 | **Euclidean %** | 1.7 |
| **SIFT %** | 71.75 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 45** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 212 | **Cosine %** | 56.04 |
| **KP 2** | 170 | **Skill Level** | Beginner |
| **G.M** | 185 | **Euclidean %** | 1.71 |
| **SIFT %** | 74.26 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 46** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 256 | **Cosine %** | 41.66 |
| **KP 2** | 270 | **Skill Level** | Beginner |
| **G.M** | 223 | **Euclidean %** | 1.46 |
| **SIFT %** | 73.98 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 47** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 342 | **Cosine %** | 59.59 |
| **KP 2** | 213 | **Skill Level** | Beginner |
| **G.M** | 156 | **Euclidean %** | 1.55 |
| **SIFT %** | 45.61 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 48** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 313 | **Cosine %** | 72.32 |
| **KP 2** | 246 | **Skill Level** | Intermediate |
| **G.M** | 277 | **Euclidean %** | 1.95 |
| **SIFT %** | 69.5 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sketch 49** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 328 | **Cosine %** | 63.06 |
| **KP 2** | 206 | **Skill Level** | Beginner |
| **G.M** | 135 | **Euclidean %** | 1.94 |
| **SIFT %** | 41.16 | **Skill Level** | Beginner |
| **Skill Level** | Beginner |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |
| **Sketch 50** | **SIFT** | | **VGG16** | |
|  | **KP 1** | 336 | **Cosine %** | 64.08 |
| **KP 2** | 145 | **Skill Level** | Beginner |
| **G.M** | 152 | **Euclidean %** | 1.9 |
| **SIFT %** | 74.24 | **Skill Level** | Beginner |
| **Skill Level** | Intermediate |  | |
| **Human Measuring** | | | |
| Chart, pie chart  Description automatically generated | | | |