Automatic assessment of creativity in images

Jingkai Yuan

4th Year Project Report
Honours Project Report
School of Informatics
University of Edinburgh

2023

Abstract

Traditional assessments of image creativity have primarily relied on human experts scoring, but this requires significant time and effort. At the same time, due to the subjectivity of creativity, human experts may disagree when scoring. To resolve these issues, we propose an automated image creativity scoring system based on the Torrance Test of Creative Thinking (TTCT) in this project. The measurement relies on the core concepts of the figural test in TTCT, namely Fluency, Flexibility, and Originality. In addition, we use a Convolutional Neural Network (CNN)-based deep learning model to quantify each measurement. The experimental results show that in a set of images, this automatic scoring system can distinguish the images with high creativity and those with low creativity. This automated creativity scoring system is the first computer vision approach to assess creativity that combines TTCT as the measurement and CNN-based deep learning models as quantification methods.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Jingkai Yuan*)

Acknowledgements

Great thanks to Dr. Kobi Gal, the supervisor of this project, for the valuable advice and guidance, as well as planning throughout the project. In addition, I want to thank Dr. Benjamin Paassen for his valuable comments and ideas in computer vision fields. I would also like to thank Professor Frank Ma and the 50 users who joined the creativity experiments. In the end, I would like to thank my family and friends for their support and love, as always.

All the participants, including Professor Frank Ma, and the 50 users are my friends and undergraduate students at the university. In addition, this is a pilot study, so there is no need to take approval, and participants' information will not be included in the appendix.

Table of Contents

Chapter 1

Introduction

People often think about what creativity is and why creativity is so important. Creativity is a unique and comprehensive human skill and refers to the ability to generate new ideas, to discover and create new things[1]. Creativity is significant because it is an essential power for producing and creating scientific, literary, and artistic inventions. The study of creativity has attracted widespread attention in the fields of psychology, education, sociology, culture, art, and science. Creativity is expressed in the creation of new concepts, theories, technologies, equipment, methods, works, etc.

An important part of human creativity is found in pictures. Picture refers to a visual representation of a person, object, or scene, such as a painting, drawing, photograph, etc. It can convey a wide range of ideas and emotions with versatility and expressiveness at the same time. This article will discuss the automatic assessment of creativity in pictures(images).

1.1 Research Problem

Creativity assessment plays an important role in previous creativity research. The creativity assessment has two difficulties; the first one is the subjective nature of creativity. Everyone may have different opinions on the level of creativity of the same artwork. For example, an image of a cat composed of colorful color blocks (See Figure 1.1). Rater A thinks this image is very creative because it uses many different color blocks to piece together the appearance of a cat, which makes the whole work more vivid; rater B thinks this image is not very creative. Because it depicts just a cat, whereas a creative work would combine the cat with other elements. The second difficulty is that creativity assessment needs human experts to rate it. Scoring involving human raters requires a lot of time and effort, and human experts may not be up to the job when there is a need to score the creativity of many artworks.

Automated creativity scoring can solve the disadvantages of relying on human experts in traditional creativity scoring, and it is more efficient. At the same time, automated scoring can alleviate the problem of disagreement caused by different experts due to the subjectivity of creativity. To achieve automated scoring, we must consider two issues

Figure 1.1: An image of a cat composed of colorful color blocks

for evaluating creativity. The first is what measure of creativity should be used for evaluation. The second is what method should be used to quantify each measurement standard.

What measure of creativity should be used for evaluation? – Since the last century, people have gradually become interested in evaluating creativity. Many creativity measurements have been proposed, such as Torrance's test of creative thinking (TTCT)[2] in the early stage, and Williams' tests on creative thinking[3]. Later, with the improvement of awareness of divergent thinking diversity, the figure test gradually appeared in TTCT, which is mainly used to measure individuals' divergent thinking ability and creative level in visual and image thinking. Inspired by TTCT figural tests and the approaches to evaluate creativity in the Sctach project by Kobi et al.[4], we decided to evaluate creativity in images on three scales: Fluency, Flexibility, and Originality.

- **Fluency** refers to the total number of objects presented in an image.
- **Flexibility** refers to the number of different regions in which an image can be divided into.
- **Originality** refers to how similar an image is compared to other images.

What method should be used to quantify each measurement standard? – Once we have defined the measures of creativity, we need to find a way to quantify them. The three defined standards, Fluency, Flexibility, and Originality, all need to do feature extraction on images.

Considering traditional feature extraction methods such as Scale-Invariant Feature Transform (SIFT)[5] and Histogram of Oriented Gradients (HOG)[3], the essence of the SIFT algorithm is to find critical points (feature points) in different scale-spaces and calculate the direction of key points. The key points found by SIFT are some very prominent points that will not change due to factors such as illumination, affine transformation, and noise, such as corner points, edge points, bright spots in dark areas, and

dark points in bright areas. However, SIFT sometimes extracts fewer feature points (such as blurred images) and cannot accurately extract features for objects with smooth edges (such as images with smooth edges, too few feature points are detected, and it is powerless for circles). The basic idea of the HOG algorithm is to vote on the local gradient magnitude and direction of the image to form a histogram based on the gradient characteristics and then stitch the local features together as the full feature. Although HOG can quantify the position and orientation space to a certain extent, it can suppress the impact of translation and rotation. However, HOG still has the problems of a long descriptor generation process, slow speed, and sensitivity to noise.

With the development of deep learning, various neural networks began to appear. The notion of a Convolutional Neural Network (CNN) was first proposed by Yann LeCun et al.[6]. It is a feed-forward neural network with artificial neurons that respond to surrounding cells within a specific coverage area. Therefore, it has excellent performance for large-scale image processing. A CNN consists of convolutional layers with fully connected layers and associated weight and pooling layers. This structure enables CNNs to exploit the two-dimensional system of the input data. At the same time, CNN can automatically perform feature extraction and train the model faster when it can handle high-dimensional data. Therefore, CNN is an ideal model to implement and choose for this study.

1.2 Result

As a result, we tested the automated scoring method on two datasets, the cat and human datasets, and compared it with the scoring of human experts and 50 people without art backgrounds. The experimental results show that the automated scoring system can find a few images with high creativity and a few with low creativity in a set of pictures. The results of Kendall Tau show a moderate correlation between the automated ranking result and a large amount of the ranking results of 50 users, which once again proves the feasibility of the automated scoring system to a certain extent. However, since the existing scoring mechanism cannot explain Originality very well, the automated scoring system will ignore some innovative features at the abstract level, so it will judge some images that humans think are highly creative as low creative images.

Chapter 2

Background

In this chapter, we will introduce all the techniques applied in this project.

2.1 TTCT

The Torrance Tests of Creative Thinking (TTCT)[2, 7] is a standardized test designed to measure a person's creativity in various areas such as art, writing, and problem-solving. The test was developed by E. Paul Torrance, a psychologist and creativity researcher, in the 1960s and has been widely used in educational and psychological settings since then[8].

The TTCT consists of various tests, including a figural test, which assesses the ability to generate original ideas and designs, and the creative test, which can assess originally written ideas and idea generation. The figural analysis considers various aspects of creativity, including the articulation, flexibility, originality, expressiveness, and quality of the visual ideas generated by the individual being invisible. The test also assesses an individual's ability to use symbols and visuals to communicate ideas. The TTCT has been used in various settings, including schools, research studies, and performance evaluations. It is a reliable and valid measure of vocabulary and has been used to identify gifted and gifted students and assess the effectiveness of vocabulary instruction programs.

However, some researchers[9, 10] have criticized the TTCT for its cultural bias and focus on individual creativity rather than the collaborative and socially-oriented aspects of creativity. Others have pointed out that the test may not accurately capture all aspects of creativity and may be influenced by factors such as motivation and personality. Despite these limitations, TTCT has been improved for half a century since it was proposed in the 1960s. Its comprehensive assessment content, wide application range, and detailed feedback make it the most popular and widely-used measure of creativity in educational and psychological settings.[9]

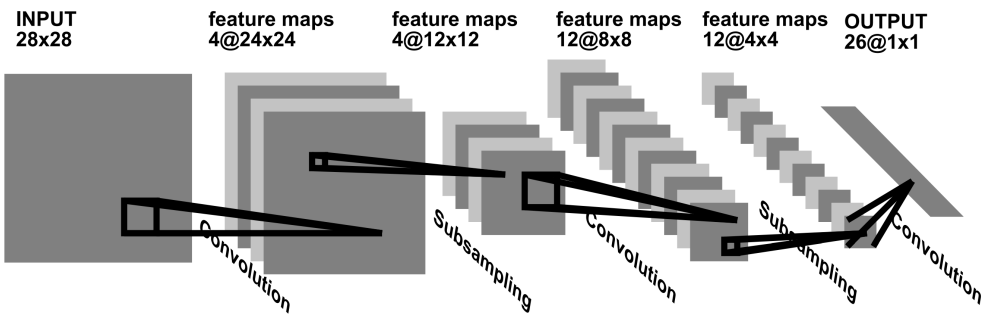


Figure 2.1: Convolutional Neural Network Structure[6]

2.2 CNN and ResNet-50

2.2.1 CNN

CNN, known as Convolutional Neural Network, is a type of deep learning algorithm commonly used in image and video analysis tasks, such as object detection and recognition. It was first introduced by Yann LeCun[6] in the 1990s and has since become one of the most widely used neural network architectures in computer vision applications.(See Figure 2.1)

The advantage of CNNs lies in their ability to automatically learn and extract features from images without manual feature engineering. This is achieved through a series of convolutional layers, where filters are applied to the input image to detect specific patterns and features, such as edges, corners, and textures. The output of each convolutional layer is then passed through a non-linear activation function, such as ReLU, which introduces non-linearity into the model.

CNNs also use pooling layers to downsample the feature maps, reducing the dimensionality of the input and making the model more efficient. Finally, the output of the convolutional and pooling layers is flattened and fed into a fully connected layer, which performs the classification or regression task. The advantage of CNNs over traditional machine learning algorithms is their ability to handle complex and high-dimensional data, such as images and videos. They can also learn from large datasets and generalize well to new, unseen data. As a result, CNNs have been successfully used in various applications[11], including image classification, object detection, facial recognition, and medical imaging.

2.2.2 ResNet-50

ResNet-50 is a deep convolutional neural network (CNN) architecture developed by Kaiming He et al.[12] (See Figure 2.2). It is a subset of the Residual Network family and was developed to address the issue of vanishing gradients, which frequently occur in very deep neural networks.

ResNet-50 has 50 layers, including convolutional layers, batch normalization layers,

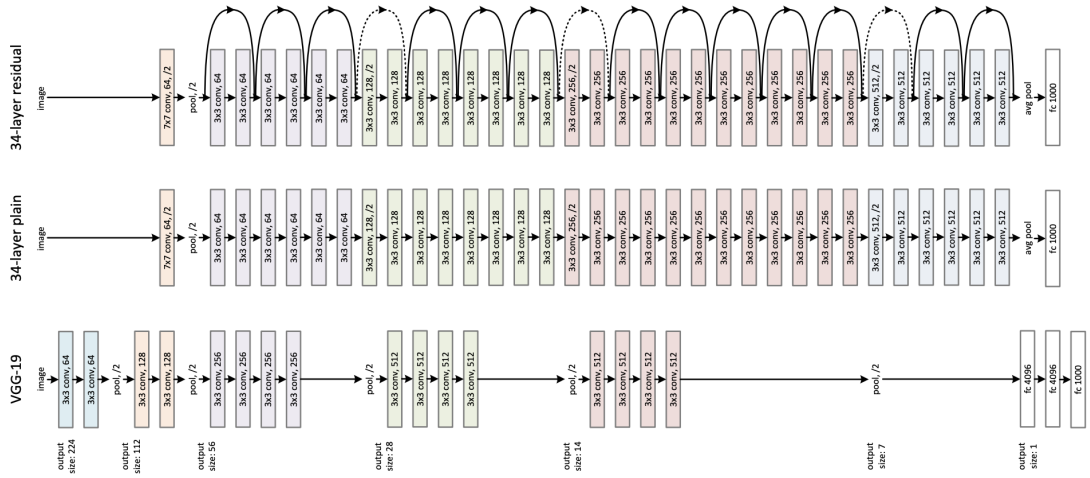


Figure 2.2: ResNet-50 backbone structure[12]

max-pooling layers, and fully connected layers. The use of residual connections, which allow information to bypass certain layers and flow directly to deeper layers in the network, is the key innovation in the ResNet-50 architecture. This reduces the vanishing gradient problem and allows the network to be trained much deeper than previous CNN architectures.

ResNet-50 has produced cutting-edge results in a variety of computer vision tasks, including image classification, object detection, and semantic segmentation. It was the winning model in the 2015 ImageNet Large Scale Visual Recognition Challenge (ILSVRC)[13], a benchmark task in image classification.

The ResNet-50 architecture has been used in various applications and is now a popular choice for studying evolution in computer vision. Several deep frameworks, such as PyTorch and TensorFlow, have pre-trained ResNet-50 models that can be successfully developed on new datasets with small data sizes.

In this article, we used the pre-trained ResNet-50 as the convolutional neural network model for extracting image embeddings.

2.3 Image Embedding

Image embedding[14] is converting an image into a compact and dense fixed-size representation, typically a vector of numbers, that captures the essence of the image content. Image embedding can be considered a high-level summary of the visual information present in the image that can be used for image retrieval, classification, and segmentation.

The earliest proposal to use a convolutional neural network (CNN) to extract image embeddings is in the AlexNet proposed by Alex Krizhevsky et al. [15]. Following pre-training of the CNN model, the network's last layer, which is typically a softmax layer for classification, is removed, and the output of the preceding layer is used as the

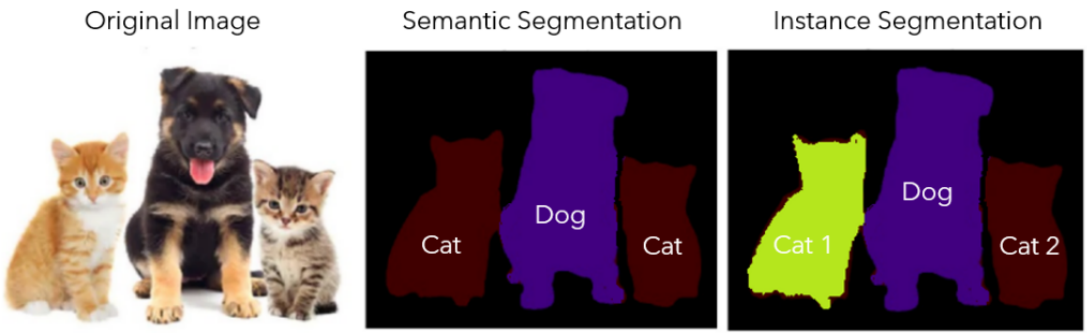


Figure 2.3: Comparison between Instance Segmentation and Semantic Segmentation[19]

image embedding. This output is a high-dimensional vector representing the network's features learned for that specific image. To reduce its dimensionality and make it more suitable for downstream tasks, this embedding can be further processed using techniques such as principal component analysis (PCA)[16] or t-distributed stochastic neighbor embedding (t-SNE)[17].

2.4 Instance Segmentation

Instance segmentation[18] is a computer vision task that entails identifying and segmenting each individual object instance contained within an image. In contrast to semantic segmentation, which divides pixels into regions corresponding to different object categories, instance segmentation differentiates between different object instances within each category and assigns each one a unique label.

We consider such a picture 2.3, Semantic Segmentation can mark the pixels of cats and dogs, but it cannot indicate how many cats and dogs are in the image. But with instance segmentation, it is possible to find the bounding boxes of each instance (in this case belonging to a dog and two cats) and the object segmentation map for each instance, thus knowing the number of instances (cats and dogs) in the picture.

Instance segmentation has many practical applications, including object detection, image and video analysis, autonomous driving, robotics, and more. It is often used in applications that require precise object detection and tracking, such as traffic monitoring, security surveillance, and medical imaging.

There are several methods for performing instance segmentation, including supervised learning techniques such as convolutional neural networks[6], unsupervised learning techniques such as clustering algorithms, and hybrid approaches that combine both supervised and unsupervised techniques. Some of the most popular instance segmentation models include Mask R-CNN[20], YOLACT[21], and CenterMask[22].

In this article, Mask-Rcnn[20] will be used for Instance Segmentation, and the purpose is to identify the types and quantities of different objects in an image

$$cosineSimilarity(A, B) = \frac{dotProduct(A, B)}{(magnitude(A) * magnitude(B))}$$

Figure 2.4: Cosine similarity formula[23]

2.5 Cosine Similarity

Cosine similarity[23] measures the similarity between two vectors in a multi-dimensional space.

When we compare two vectors using cosine similarity, the vectors should be normalized to a unit length, and this means the vector's length is scaled to 1. This could ensure that the magnitude of the vectors does not affect the cosine similarity measurement. Once we have normalized the vectors, the cosine similarity between them is calculated by taking the dot product of the two vectors and dividing it by the product of their magnitudes. The result is a value between -1 and 1, where 1 indicates that the two vectors are identical, 0 indicates that they are orthogonal, and -1 indicates that they are opposed. The formula to calculate the cosine similarity between vectors A and B is shown in Figure 2.4.

In this article, we will use cosine similarity to measure the Originality of a set of pictures

2.6 K-means Clustering

K-means[24] is an unsupervised machine learning algorithm used for clustering data. It is a simple and efficient algorithm that groups a set of data points into clusters based on their similarity. The K-means algorithm begins by selecting a fixed number of clusters (K) that the data points will be grouped into 2.5. The algorithm then randomly initializes K centroids, which are the center points of each cluster. The data points are assigned to the closest centroid based on their similarity, usually measured by the Euclidean distance.

Once the initial assignment is made, the algorithm updates the position of the centroids by calculating the mean of all data points assigned to each cluster. The data points are then reassigned to the closest centroid, and the process is repeated until convergence. K-means is an iterative algorithm, and convergence is guaranteed. But it may get stuck in the local optima depending on the initial positions of the centroids. Therefore, multiple initializations of the algorithm are typically performed, and the best clustering result will be selected based on a pre-defined evaluation metric.

The K-means cluster algorithm is suitable for large data sets due to its simplicity and efficiency. However, the algorithm also has some drawbacks. For example, we need to pre-determine the number of K clusters. In addition, the K-algorithm is sensitive to the initial position of the centroids, and the result of the cluster will vary depending on the onset. Despite these limitations, K-means is a popular algorithm for clustering data.

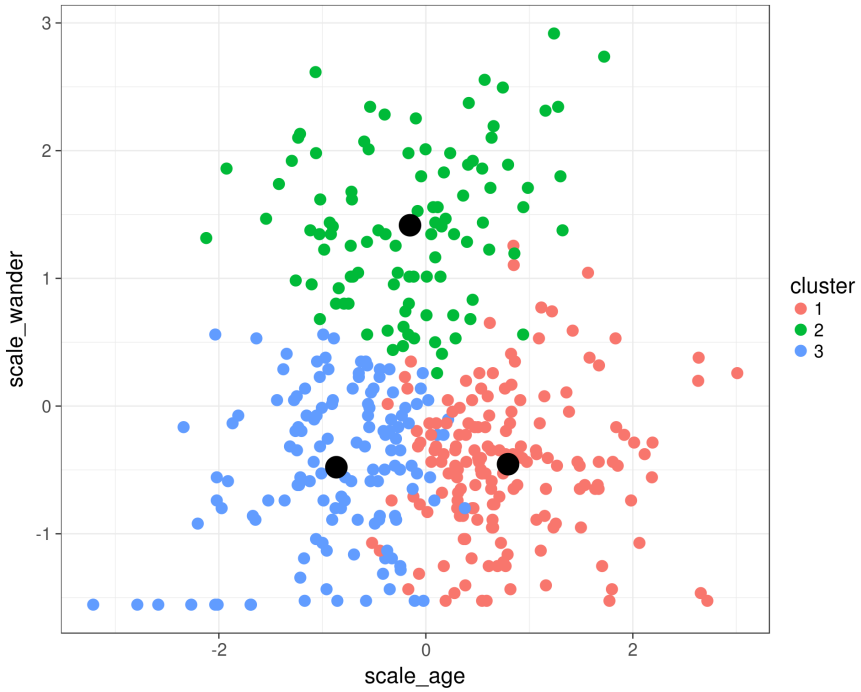


Figure 2.5: An example of K-means clustering[25]

In this article, we will use the K-means clustering algorithm to calculate the Flexibility in the creativity evaluation standard, that is, how many regions an image can be divided into.

2.7 Kendall rank correlation coefficient

Kendall's rank correlation coefficient[26] was proposed by Maurice Kendall in 1938, and it is a statistical measure used to quantify the degree of agreement between two rankings or ordinal data. It is a non-parametric correlation measure, which means it does not require the data to follow any particular distribution.

The Kendall rank correlation coefficient, ranges from -1 to 1. A value of 1 indicates a perfect positive correlation, which means that the two rankings agree completely. A value of -1 indicates a perfect negative correlation, indicating that the two rankings are completely contradictory. A value of 0 indicates no relationship between the two rankings.

Kendall's rank correlation coefficient is calculated by pairwise comparison of the scores of the two items. The agreement or disagreement of each pair of observations is determined by whether the order of the two objects is the same or different. Then, both congruent and incongruent pairs are counted, and the coefficient is calculated as the difference between the number of congruent pairs and the number of incongruent pairs to each other, divided by the sum of both.

Kendall's rank correlation coefficient is commonly used to analyze the relationship between two ordinal variables in fields such as social sciences, economics, and biology.

It is a useful measure when the data does not follow a normal distribution or contains outliers or extreme values. However, when the data contains ties, or the sample size is small, Kendall's rank correlation coefficient may not be appropriate.

In this article, Kendall's tau is used to calculate how much agreement between student ranking and the ranking of automated ranking.

Chapter 3

Literature Review

In this chapter, we discuss previous work on measuring the creativity of images.

3.1 Modeling Creativity in Visual Programming

”Modeling Creativity in Visual Programming: From Theory to Practice”[4] is a research paper by Kobi Gai et al. in 2021. This paper introduces a new method to measure creativity in visual programming and explores the importance of creativity in visual programming.

First, the paper introduces the concepts and applications of visual programming and discusses the importance of creativity in visual programming. Visual programming refers to creating computer programs using a graphical interface rather than writing code. Since visual programming can make programming more intuitive and easy to learn and use, it is gaining popularity in education and industrial applications. However, creativity needs to be effectively measured and evaluated to enable creative design and development in visual programming.

This paper proposes a machine learning-based creativity measurement method that analyzes programmers' behavioral data to predict their creativity in Sctach projects. Inspired by the Torrance Test of Creative Thinking (TTCT)[7], the method combines three elements of creativity: flexibility, fluency, and originality to measure creativity in visual programming. Fluency refers to the number of concepts used in the program, flexibility to the distance between concepts, and originality to the distance to specific programs.

To verify the effectiveness of the method, the authors conducted two experiments. In the first experiment, the authors recruited 14 participants, asked them to complete a simple visual programming task, and assessed their creativity using the proposed method. The results show that the method provides a good measure of creativity in visual programming and an effective tool to evaluate and compare the creativity of different programmers. In the second experiment, the authors used a more complex visual programming task and compared the performance of the proposed method with traditional human evaluation methods. Automated creativity scores using machine

learning models were more closely aligned with expert rating rankings than even the agreement between expert ratings. The results show that creativity can be measured more accurately using the proposed method, and significant time and cost savings can be achieved.

3.2 Automated scoring of figural creativity using a convolutional neural network

In the paper "Automated scoring of figural creativity using a convolutional neural network" [27], the author proposes an automatic way to assess figural creativity using CNN based on the Test of Creative Thinking–Drawing Production (TCT-DP) standard[7], which was proposed by E.Paul Torrance in 1966.

First, the authors used an autoencoder-based convolutional neural network to extract the features of paintings. The self-encoder consists of two parts: an encoder and a decoder, where the encoder part can convert the painting into a low-dimensional feature vector. This feature vector contains essential information about the painting and can be used to evaluate its creativity level.

Next, the authors used a pre-trained MobileNet model[28] to extract features of the paintings (See Figure 3.1). The model can convert paintings into a high-dimensional feature vector and reduce computation while maintaining feature quality. The authors then perform dimensionality reduction on the eigenvectors using Principal Component Analysis (PCA)[16]. PCA is a commonly used data dimensionality reduction technique, which can reduce the dimensionality of feature vectors, thereby reducing computational complexity. Through dimensionality reduction, the creative characteristics of paintings can be better described. Finally, the authors use feature vectors to calculate a creative score for each drawing. To calculate the creativity score, the authors employed a variance-based assessment method that compares the difference between the creativity score of a drawing and a benchmark score to assess its level of creativity.

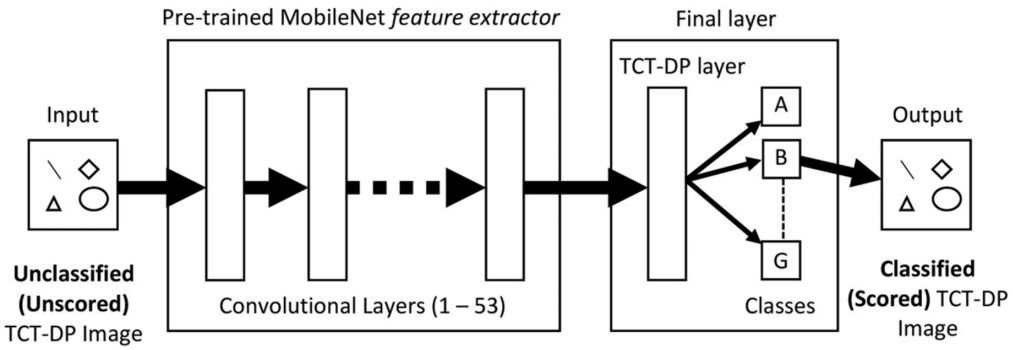
To evaluate the performance of the automated scoring method, the authors conducted experiments on three datasets and compared their results with human scoring. Experimental results show that this method can effectively evaluate the creativity level of paintings, and its scores are highly correlated with human scores. In addition, the evaluation speed of the method is also fast, which can be quickly and automatically evaluated. In conclusion, this paper proposes an automated scoring method based on a convolutional neural network and principal component analysis for evaluating the creative level of paintings. The method has high accuracy, repeatability, and practicality and can be applied in art education and research.

3.3 Quantifying Creativity in Art Networks

In the paper "Quantifying Creativity in Art Networks"[29], the author proposed an approach to quantify creativity based on Historical-Creativity[30] in figure 3.2. For instance, H-creativity is reflected in the directed graph (Art Networks) that every painting

Figure 2

Convolutional Neural Network for Figural Creativity Assessment



Note. TCT-DP = The test of Creative Thinking–Drawing Production.

Figure 3.1: Convolutional Neural Network for Figural Creativity Assessment based on TCP-DP standard[27]

comes with a time label indicating the date it was created, denoted by t_{p_i}, and the parameters will be assigned considering the originality of the paintings t_{p_i}. Suppose P is a set of paintings P = p_i, i = 1...N; correspondingly, assign a creative score C = C_{p_i}, i = 1...N to each painting. The composed time of each figure labels to create a directed graph, and each directed edge is given a positive weight. This weight is used as a key factor affecting the measurement of visual aspects (e.g., Color, Theme, Stroke, etc.). Creativity scores are induced through these adjacency matrices.

The creativity score can be controlled by changing the size of the edge weight to achieve creativity propagation. To calculate the creativity score of each picture, the paper proposes a definition formula that defines an assigned creativity score, shows the construction of the Creativity Implication Network reduces the problem of computing the creativity scores to a traditional network centrality problem. The article also proposes how to obtain different feature representations of images through traditional computer vision techniques, that is, based on Historical Creativity, using a Bayesian probability model to draw a Gaussian probability density model for each image. Each utterance is scored based on all previous paintings connected to that painting in the directed graph while limiting the bias of modern and earlier paintings.

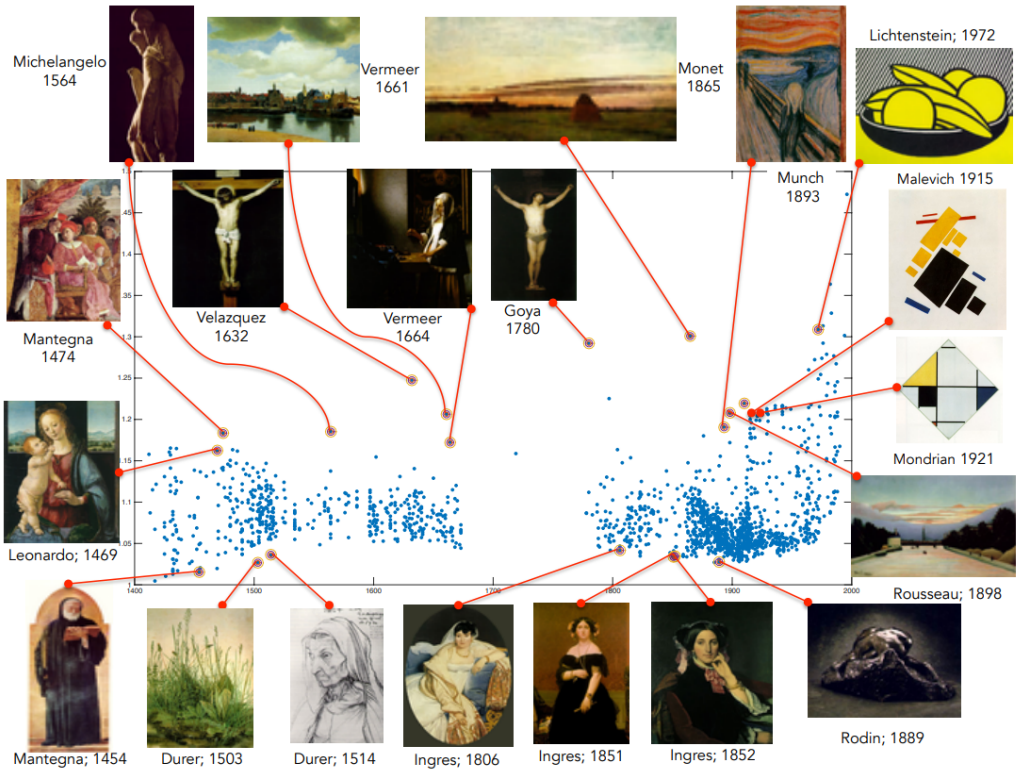


Figure 3.2: The graph of creativity score verse the year of the painting[29]

Chapter 4

Methdology

In this chapter, I will introduce the specific creativity measurement process using Torrance Tests of Creative Thinking(TTCT)[7], including Fluency, Flexibility, and Originality and their corresponding implementation models and calculation methods.

4.1 Input data

The input to the model is an RGB image: usually, a three-channel (red, green, blue) color image is used as the input to the model, and the image can be of any size. The image's content can be any form of figural representation, such as photos, artificial painting, pictures generated by AI, etc. However, the objects or elements in the picture need to express the nature of the characteristics of the object, and the too abstract artistic expression may cause the network model to be unable to distinguish the object.

4.1.1 DALL-E*2 Image

DALL-E 2[31] is an advanced image generation AI model developed by OpenAI that can generate high-quality, diverse and creative images. It is an improved version of the DALL-E model with higher resolution and faster generation.

The DALL-E 2 can accept many forms of text input, including descriptive language and question-answer formats. It also supports generating multiple objects and can control the number, position, size, and orientation of objects to generate highly customized images. Compared with DALL-E, DALL-E 2 can generate higher resolution images (4096 x 4096 pixels) and can generate more images in less time. In addition, DALL-E 2 also supports a wider range of object and scene types, such as animals, people, buildings, etc.

Figure 4.1 is an image generated from the DALL-E 2 model using the prompt "A cat with tie is eating apples and bananas using forks". The resolution of this image is 1024 x 1024 pixels. This image will be used to demonstrate how the creativity score is assessed in Fluency, Flexibility and Originality.

Figure 4.1: An image generated from the prompt ”A cat with tie is eating apples and bananas using forks” in the DALL-E 2 model

4.2 Assessing Fluency

4.2.1 Approach selection

As we mentioned before, Fluency refers to the total amount of Ideas generated in TTCT. Similarly, we can apply this idea to images, that is, use the total number of different objects presented in an image as a reference to the Fluency quantification standard. There are many ways to detect the sum of the number of different objects in an image:

- Object Detection: This is an advanced technique for detecting objects in an image and determining their location and category. Object detection can use deep learning techniques such as Convolutional Neural Networks (CNN)[6] or Region Proposal Networks (RPN)[32], etc.
- Instance Segmentation: This is a technique of dividing an image into different parts, where each part represents an object or part of an object. By counting the number of objects in the segmentation result, the sum of the numbers of different objects in the image can be determined.
- Feature Extraction: This is a technique of extracting key features from an image that can be used to identify objects and count them. This can be achieved using feature descriptors such as Scale Invariant Feature Transform (SIFT)[5].
- Edge Detection: This is a technique for detecting the edges of objects in an image. Edge detection can be used to count objects by counting the number of edges in an image.
- Thresholding Methods: This technique converts an image into black and white, where pixels of similar color are assigned the same value. By counting the number of objects in the segmentation result, the sum of the numbers of different

objects in the image can be determined.

In this project, I used the instance segmentation technique in image segmentation to implement it. Instance segmentation is an advanced image segmentation technique that can accurately detect and segment each object in an image. Compared with other methods, instance segmentation can provide more accurate object segmentation results and can simultaneously detect and count the number of different objects in the image. Therefore, instance segmentation is a very effective way to detect different objects in an image and their number.

4.2.2 model selection

First, we need to build a deep learning model for training to achieve instance segmentation, which is used to detect different types of objects in the picture and their number. Since there are a variety of different objects in the picture, identifying each object requires training on different types of object data during model training. However, it is unrealistic to implement the recognition of each object. Nevertheless, we can consider other ways to solve this problem.

- **Use the same common class for different objects:** Treat all objects as the same class, such as ”object” or ”target” and use instance segmentation to detect and count the number of different objects.
- **Recognize some objects:** Choose to train the model to recognize some objects. For example, only animals, but not all objects, or only cars and people, but not other objects. This can reduce the workload of data collection and model training.
- **Using pre-trained models:** Use pre-trained models that have been trained for instance segmentation and object counting. These pre-trained models have been trained on massive datasets to detect and count many different kinds of objects.

After consideration, using a pre-trained model seems to be a reasonable and feasible way to implement it. I chose **Maskrcnn_resnet50_fpn**[33] as the pre-training model. Figure 4.2 displays an example of instance segmentation using Maskrcnn_resnet50_fpn in PyTorch.

Maskrcnn_resnet50_fpn[33] is a deep learning model implemented based on PyTorch, which is a pre-trained instance segmentation model. The model uses Mask R-CNN, a modified version of the Faster R-CNN algorithm[34], to detect and segment objects in images. Different from the Faster R-CNN algorithm, the Mask R-CNN model[34] adds a mask branch based on the Faster R-CNN model[34], which can output pixel-level object segmentation masks to detect object boundaries and shapes more accurately.

The model uses a ResNet-50-FPN backbone network as a feature extractor. ResNet-50-FPN is a deep convolutional neural network that can extract features from feature maps of different scales to improve the accuracy of object detection and segmentation. The model’s pre-trained weights were trained using the COCO dataset[35], which contains 91 common categories of objects, such as people, cars, animals, and furniture.

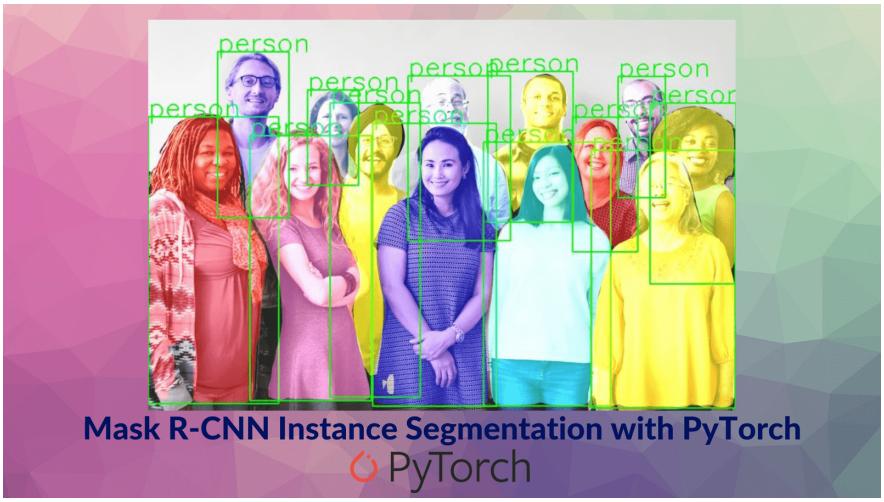


Figure 4.2: An example of the results generated using the instance segmentation model Maskrcnn_resnet50_fpn[33] in Pytorch

4.2.3 Processing steps

- **Load the model:** Firstly, we use the torch-vision library in PyTorch to load the pre-trained Maskrcnn_resnet50_fpn model, set the model parameters to (Pretrain=True, Progress=True), and specify to use pre-trained that can recognize 91 different object types Version.
- **Image preprocessing:** Import and convert the image into RGB format, then convert the image into tensor for use as the input format of the network, and save the original image before inputting it to the network, so as to apply the mask to the image later.
- **Image forward propagation:** We input the preprocessed image into the model and perform forward propagation. As a result, the candidate frames and segment candidate frames will be returned.
- **Post-processing:** Then we need to adjust the threshold hyperparameter to filter the probability of all detected objects. According to the bounding box and mask information output by the model, the overlapping bounding boxes are removed, and the candidate boxes are segmented using the mask information.
- **Visualization output:** Superimpose the processed bounding box and mask information with the original image to form a logo with the target box, and then save the superimposed image.

4.2.4 Output assessment

The instance segmentation algorithm will return the bounding box and probability corresponding to all possible objects in the image. Here, we set a threshold to filter the detected objects below a particular probability value to ensure that all marked objects presented in the final output are above this value.

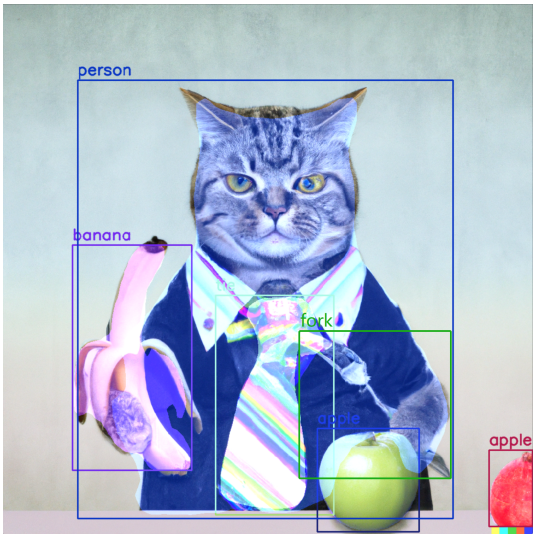


Figure 4.3: The output of instance segmentation with threshold equals to 0.9

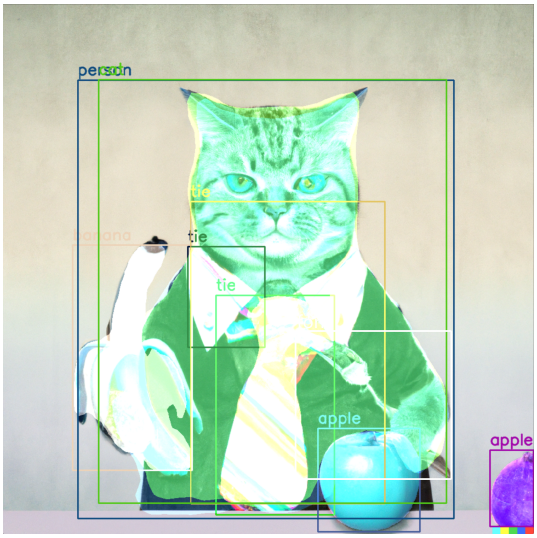


Figure 4.4: The output of instance segmentation with threshold equals to 0.5

Figure 4.3 shows the output of the instance segmentation with a threshold equals to 0.9 **(keep the objects whose predicted probability is higher than 90%)**. The output image contains six boundary boxes with five different categories of objects. They are a person, a banana, a tie, a fork, and two apples.

- From the perspective of human vision, we see an image with a cat's head and a person holding a fork to eat fruit. Usually, most people would recognize this image as an anthropomorphic cat.
- From the perspective of computer vision, the instance segmentation model was trained based on authentic object images. Usually, cats eating fruit with a fork in a human pose would not appear in the training dataset. Therefore, considering an object with a cat looking in human pose, the instance segmentation model will extract more human features than cat features and eventually misidentify the object as a person object.

Figure 4.4 shows the output of the instance segmentation with a threshold equals to 0.5 **(keep the objects whose predicted probability is higher than 50%)**. In this image, the instance segmentation model recognizes that the object may also be a cat while recognizing the person, but the probability is lower than that of a person.

In addition, after reducing the threshold value, we found that there are two tie-type objects are detected in the final output. They all contain the characteristics of a tie as a whole, but the probability is lower. Inspired by this, sometimes, when the object is quite abstract or cannot be recognized due to the angle, we can properly reduce the threshold to achieve the goal of recognizing the object.

4.2.5 Fluency score

When calculating the final image fluency score, two aspects will be considered: **the total number of recognized objects** and **the number of different types of objects recognized**.

If, in an image, a certain object appears more than 5 times, then we will only record 5 as the number of occurrences of the object category. This is to prevent a certain image from appearing too much, which may lead to Fluency's score being unusually high.

In this example, we take Figure 4.3 as the final output. The final model predicts that the picture contains a person, a banana, a fork, a tie, and two apples. Hence, **the total number of recognized objects** = 1 + 1 + 1 + 1 + 2 = 6 and **the number of different types of objects is recognized** = 4 (plus 1 for each additional type). The final score of Fluency will be the sum of these two terms, which is **10**.

4.3 Assessing Flexibility

4.3.1 Approach Selection

Flexibility refers to the number of different regions in which an image can be divided into. So we need to find an image segmentation method to determine how to divide the region. We consider the following approaches:

- **Superpixel-based segmentation methods:** These methods segment an image into regions of superpixels, which are small blocks of adjacent pixels. The number of regions can be determined by analyzing the similarity and distance between superpixels. For example, algorithms such as k-means, mean shift, etc., can be used to cluster superpixels.
- **Edge-based segmentation methods:** These methods use edge detection algorithms to detect edges in an image and determine the number of regions based on the connectivity of the edges.
- **Texture and color-based segmentation methods:** The technique counts the number of regions in an image based on variations in texture and color to identify various sections in the image. For instance, one can segment images using clustering techniques and texture descriptors like HOG[3] and SIFT[5], etc.

However, superpixel-based segmentation methods are challenging to deal with irregularly shaped images and need to set the number and size of superpixels manually. Edge-based segmentation methods are susceptible to edge detection results, so they may not be able to detect image edges with complex textures and shapes accurately. They will also be affected by the image quality. Color variations influence texture and color-based segmentation methods and are also challenging to handle images with complex textures.

In coordination with the thesis supervisors, I chose a method of using **region segmentation based on extracting sub-image image embeddings**. Specifically, we divide an image into multiple sub-images of the same size. For each sub-image, use the

Convolutional Neural Network(CNN) to extract the embedding vector and perform normalization on these extracted embedding vectors. Finally, we apply a clustering algorithm on the embedding vectors to determine how many clusters the image can be divided into.

The advantage of this method is that it can be applied to different types of images, regardless of image size, color, or content, and can be segmented in the same way. This method provides excellent scalability, and the number and size of image divisions can be adjusted to meet different task needs. At the same time, it can also be optimized according to other clustering algorithms and feature extraction methods. Furthermore, this method is robust to noise and outliers in the image. If some sub-images appear abnormal, these subgraphs can be excluded from the clustering algorithm for better segmentation results.

4.3.2 Model Selection

In the CNN model selection, we chose to use pre-trained ResNet-50 to extract image embeddings.

ResNet-50[12] is a deep convolutional neural network with 50 layers, and the feature extraction ability of this deep convolutional network is very strong. Furthermore, the residual block is an essential part of ResNet-50, which allows information to be directly transmitted across multiple layers, avoiding the gradient disappearance and model degradation caused by excessive model stacking, making ResNet-50 easier to train, and also improving the performance of the model.

The benefits of Residual blocks are also shown when ResNet-50 is used to extract image embeddings. Since image embedding needs to extract the advanced features of the image, the Residual block has a robust feature extraction ability. Furthermore, it can effectively extract the abstract features of the image. In addition, ResNet-50 uses a global average pooling layer, which can convert input images of different sizes into fixed-length feature vectors, making image embeddings more comparable. At the same time, the pre-trained ResNet-50 model has good migration ability in the image embedding task, which can effectively solve the problem of small data sets and improve the accuracy and generalization performance of image embedding.

4.3.3 Processing steps

- **Load the model:** First, we directly load the pre-trained ResNet-50 model in PyTorch Hub and then adjust the model to inference mode.
- **Data preprocessing:** First, we divide the image into 64 sub-images with the size of 128*128 pixels and save them seperately. (Figure 4.5 shows the example)

  For each sub-image, convert the image into RGB format, then convert the image into tensor for use as the input format of the network. Since Pytorch's default image backend is Pillow, PyTorch will automatically convert all images between 0 and 1, so we don't need to perform normalization on the images here.

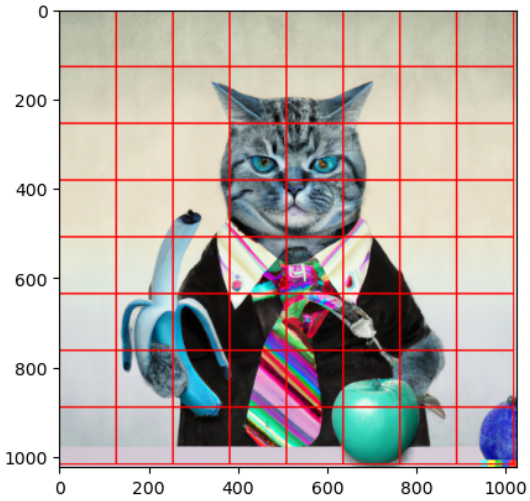


Figure 4.5: An example of dividing an image into many sub-images of the same size

- **Select feature extraction layer:** CNN's convolutional layers extract image features, and then fully connected layers handle classification and return class probabilities. To generate image embeddings, we need to get the features before the last fully connected layer for classification. So we need to get the previous layer of the fully connected layer, also called the penultimate layer. Use a specific module in PyTorch to get the embedding.
- **Image forward propagation:** We then perform batch forward propagation on the dataset using the model's inference mode.
- **Post-processing:** Flatten the output of the batch data, and finally store the embedding vector in the data frame of pandas

4.3.4 K-means Clustering

When we get the embedding image vectors of the 64 sub-images, we use these vectors as input and then use K-means Clustering to cluster the 64 data points.

The K-Means algorithm is a commonly used clustering algorithm. Its K value represents the number of clusters, so how to choose the K value is an important factor for the quality of this algorithm. Here, we choose **Elbow Method**[36] to determine the K value. That is, under different K values, calculate the Sum of Squared Errors of each K value clustering result and draw the K-SSE curve, and then determine the appropriate K value for the number of clusters according to the position of the "elbow" on the curve.

Figure 4.6 represents the K-SSE curve of K-means clustering for 64 image embedding vectors. We set the variation range of the K value between 2 and 20, and it can be seen from the figure that an Elbow shape is formed when the K value is 8, 9, and 10. So we can determine the value of K as 9.

But there are many problems with using the elbow method alone:

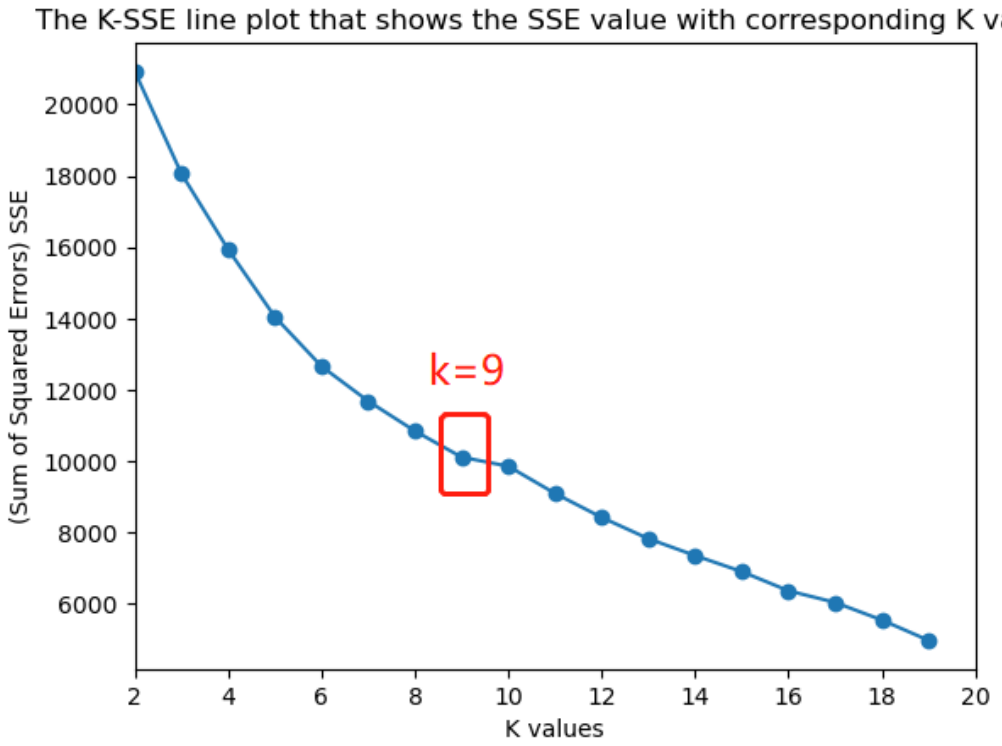


Figure 4.6: The K-SSE curve of K-means clustering for 64 sub-image embedding vectors that shows the elbow points

- **Elbow not obvious:** The K-SSE curve may lack evident elbows when the data distribution is asymmetrical or the number of clusters exhibits no clear distinction, making the choice of K value challenging.
- **Subjective influence:** The elbow method's outcomes could be impacted by subjective human factors. This may lead to the chosen K value being incorrect because the elbow position judgment must be made subjectively.
- **Single metric:** The elbow technique overlooks other aspects of clustering findings, such as the closeness inside the cluster and the gap between clusters, and only considers SSE when evaluating the clustering impact. Hence, some K values with better clustering effects may be disregarded by the elbow technique.

To prevent the impact of the aforementioned issues, we must combine the elbow technique with the actual circumstance while determining the K value. To aid in determining the ideal K value, you can also experiment with different K value selection techniques, such as the Silhouette Coefficient[37]. But in this work, we solely evaluate the K value using elbow methods.

To better understand the region division of K-means clustering, the cluster label of each sub-image is recorded. We added it to the position of each sub-image. (See Figure 4.7)

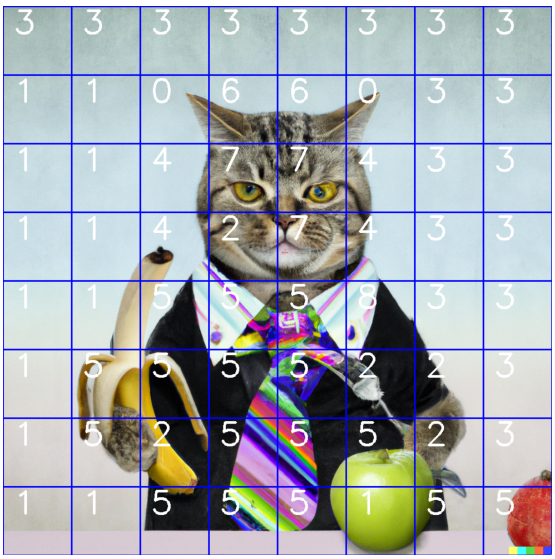


Figure 4.7: The Region-divided images with corresponding K-means clustering labels

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2$$

Figure 4.8: The Mean Square Error formula for calculating the similarity between two images[38]

4.3.5 Flexibility Score

The final flexibility score is equal to the number of regions that can be divided into, which is **9** for this image.

4.4 Assessing Originality

4.4.1 Approach Selection

Originality refers to how similar an image is compared to other images. In a group of images, the lower the similarity between an image and all other images, the higher the originality of this picture. Therefore, we need to find a method that can measure the similarity between images. Among the more common methods are:

- **Mean Square Error(MSE)[38]:** In image similarity comparison, we sum the square of the difference in each pixel value of the two images and divide by the number of pixels to get the average. The formula is given in Figure 4.8 where N is the number of pixels in the picture, and X and Y are the vectors after flattening the image pixels, respectively.
- **Structural Similarity Index(SSIM)[39]:** The structural similarity index is a more complex index that compares the structure, contrast, and brightness of two images to determine whether they are similar. The SSIM value ranges from -1

to 1, with the higher the value, the more similar the two images are.

However, using SSIM or MSE to calculate image similarity will also expose some serious shortcomings:

- The MSE and SSIM metrics are not directly comparable for images of different resolutions or sizes. Because they are based on pixel-level comparisons rather than structural or semantic information, for example, two images with different resolutions may have large MSE and SSIM values despite having similar contents.
- In some cases, the MSE and SSIM metrics may not accurately reflect the human perception of image similarity. MSE and SSIM metrics, for example, may produce inaccurate results in image noise reduction or compression because they need to account for the characteristics of the human visual system.

Inspired by assessing flexibility, we can use image embeddings for similarity comparison. Because the convolutional neural network can automatically extract essential features in the image and ignore some unimportant features, it can capture the semantic and structural information of the image more accurately than SSIM and MSE. Then use cosine similarity to calculate the similarity between embedding vectors.

4.4.2 Calculate the Originality Score

The calculation of originality is discussed in a group of images. Figure 5.1 shows the example dataset. This is a cat theme dataset generated by DALL-E 2. Each image was generated with different prompts in purpose so that it could result in images with clear distinctions in creativity.

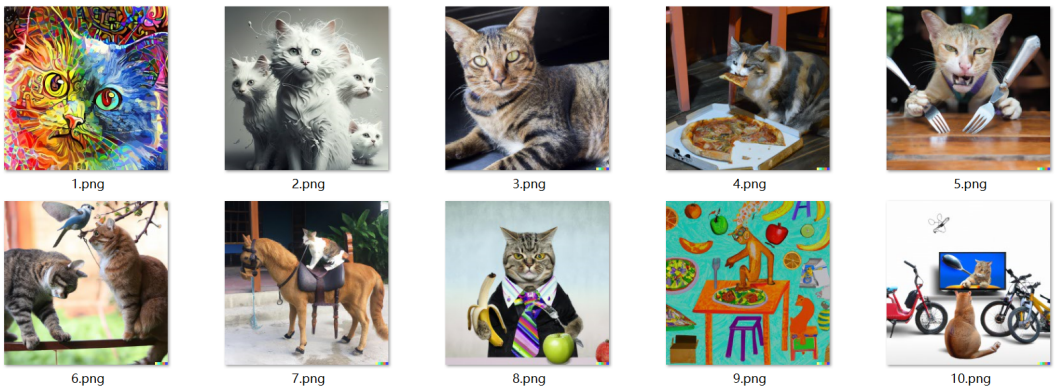


Figure 4.9: a cat theme dataset generated by DALL-E 2 with different prompts

Firstly, We will extract the embedding vectors of these ten images. For our Input image (the cat eating fruit with a fork), we will calculate the Cosine Similarity of this image and the remaining nine images, respectively. Then calculate the sum of these nine cosine similarity values, and finally subtract this sum from N (N is the total number of images in this group).

For this image, the sum of the nine cosine similarity values is 6.452. Then the final Originality score is **10 - 6.452 = 3.548**

4.5 Combined Creativity Score

After we calculate the Fluency, Flexibility, and Originality of each picture, in order to compare the results in a standardized manner, we need to standardize each item of data first. Inspired by the calculation of CCS in the visual programming environment by Kobi et al., I adopted the **Min-Max normalization**[40] method to set the scale of Fluency, Flexibility, and Originality between 0 and 1. In a list of values, for each one, we standardize it using the Min-Max normalization using the formula in Figure 5.1. So the maximum score of the Combined Creativity Score is 3 (since we have three measurements, each measurement has a maximum score of 1), and the lowest score is 0.

$$\frac{ObservationValue - MinimumValue}{MaximumValue - MinimumValue}$$

Figure 4.10: The formula of Min-Max standardization[40]

We can also weigh the standardized values of Fluency, Flexibility, and Originality according to the specific conditions of different data sets. For example, in human portraits, Originality will have a more significant impact on the creativity of the image, so we can balance this problem by increasing the weight of Originality.

Chapter 5

Experiments

In this chapter, I will introduce two examples of datasets, a cat dataset, and a human dataset, to verify the feasibility of the automated scoring system. Also, I will discuss the user study result using the Kendall rank correlation coefficient.

5.1 Experiment with Cat dataset

This cat dataset 5.1 includes ten cat-themed images, and all of them are generated using DALL-E 2 with different prompts (see specific prompts in appendix A). The reason for using different prompts is to make the generated images intentionally have different levels of creativity so that human raters can easily identify the difference between high-creativity images and low-creativity images. We will use the original file name as the initial serial number of each picture.

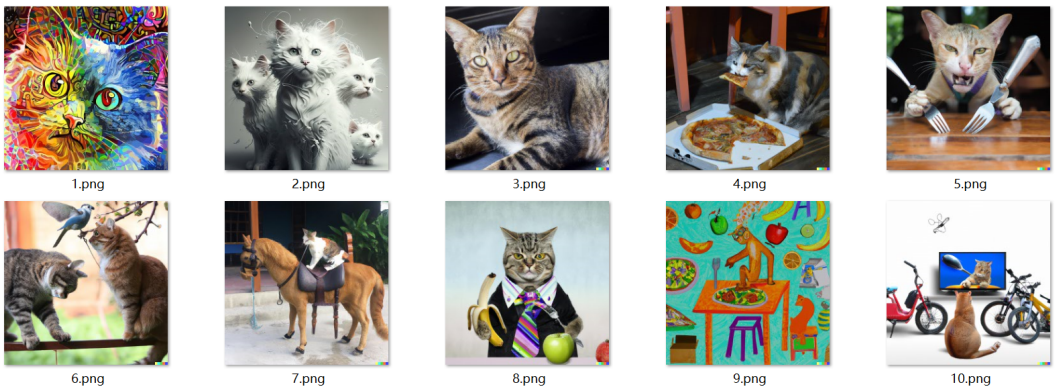


Figure 5.1: The cat dataset generated by DALL-E 2 with different prompts

Then, We calculate the Fluency, Flexibility, and Originality of each image separately. Figure 5.2 shows the Flexibility score, Fluency score, Originality score, Combined Creativity Score, and corresponding rankings for all images. The numbers inside the parentheses are the score after the Min-Max standardization. Figure 5.3 shows the ranking of the automated scoring result using the original images.

Images	Flexibility Score	Fluency Score	Originality Score	Combined Score	Ranking
Image 1	13 (1.0)	1 (0.0)	4.555 (0.887)	1.89	2
Image 2	9 (0.5)	4 (0.166)	3.519 (0.015)	0.68	7
Image 3	9 (0.5)	1 (0.0)	3.646 (0.122)	0.62	8
Image 4	5 (0.0)	3 (0.111)	3.636 (0.113)	0.22	9
Image 5	10 (0.625)	6 (0.277)	3.535 (0.028)	0.93	6
Image 6	5 (0.0)	4 (0.166)	3.501 (0.0)	0.17	10
Image 7	8 (0.375)	3 (0.111)	4.135 (0.534)	1.02	5
Image 8	9 (0.5)	10(0.5)	3.548 (0.039)	1.03	4
Image 9	11 (0.75)	19 (1.0)	4.688 (1.0)	2.75	1
Image 10	6 (0.125)	11 (0.666)	4.021 (0.438)	1.23	3

Figure 5.2: The chart that contains the Flexibility score, Fluency score, Originality score, Combined Creativity Score, and corresponding rankings for all images in the cat dataset (The number in the brackets is the score after normalization)

Figure 5.3: The automated scoring result on the cat dataset ranked according to the combined creativity score)

It can be seen from Figure 5.3 that image No.9 is ranked in first place by the automated scoring system. Combined with Figure 5.2, we can find that the Fluency and Originality scores of image 9 are the highest among the ten images. Its high Fluency is because image No.9 contains various objects, such as apples, bananas, pizza, forks, tables, chairs, etc. At the same time, its high Originality is attributed to the abstract style of the image, in which the cat sitting on the table looks more like a person under the abstract style. This makes image No.9 less similar to the other images and thus gets the highest Originality score. At the same time, due to the wide variety of colors and lines of different shapes in the composition, its Flexibility score is relatively high, and finally obtained a combined creativity score of 2.75.

Image No.1 is ranked second due to its high combined creativity score comes from its Flexibility and Originality score. For its high Flexibility, it is because image. No.1 is composed of many small polygons. In addition, the shape and many irregular lines make the local features of this image diverse, so the features extracted in the image embedding also become various. At the same time, the image uses many different colors, which will also lead to the diversity of picture feature extraction. This ends up making it the highest Flexibility of all the images, which is 1. Again, this composition of Image No.9 is very different from the other nine images, which explains why it has a relatively high originality score (0.887). However, for its Fluency score, although its expression is very complex and unique, the final result is just the appearance of a cat. This results in image No.1 having the lowest Fluency score compared to the rest, which is 0 after the standardization.

The Flexibility and Originality score of image No.6 is the lowest among all pictures, and its Fluency score is also relatively low. In the end, it was ranked last in this group of pictures with a Combined Creativity Score of 0.17.

Figure 5.4: The human expert scoring result on the cat dataset ranked according to the combined creativity score

To verify the scoring level of the automated scoring system for this group of images, Frank Ma, a professor from the China Academy of Art, is invited as the human expert to rank this group of images. Figure 5.4 shows the human expert scoring result on this cat dataset.

Compared the human expert result with the automated scoring result, they both put image No.9 in the first place. The images ranked second, third, and fourth in the automated scoring ranking were respectively ranked third, fourth, and fifth in the Human expert ranking. Although images No.3, No.4, and No.6 rank orders in automated scoring ranking and human expert ranking are different, they are all placed in the last three positions in both rankings.

However, the most significant disagreement between the two rankings is image No.2 (Four white cats). In the automated scoring ranking, image No.2 is placed seventh. Conversely, in the human expert ranking, image No.2 is placed second. To figure out the disagreement on this image, I asked the human expert for his opinion of ranking image No.2 in second place, and he gave the following answers:

- ”The cats’ fur in this image goes from solid to liquid and spreads out in different directions. At the same time, the direction of divergence is also the direction in which the four cats look. The four cats’ facial expressions are also different, giving people a feeling between realistic and unrealistic. Furthermore, the color is relatively clean, and this sense of emanation has more visual impact, which aligns with my definition of creativity.”

The reason that image No.2 was placed in seventh (relatively low creativity) is that it has a very low fluency and originality score (0.166 and 0.015, respectively). For the automated scoring system, image No.7 can only be read as four cats, and each cat looks similar to the cats in the rest of the images. Considering the comments by the human expert, the automated scoring system could not capture the features like: ”The cats’ fur goes from solid to liquid and spreads out in different directions.”. It couldn’t correlate the direction the cat was looking with the direction in which the cat’s fur spread. It also couldn’t explain whether the cat’s facial expression would increase creativity. Although this is just a human expert’s subjective view of the image, it can reflect that the automated scoring system cannot incorporate a human understanding of some abstract art into the scoring mechanism.

Except for some images of a particular artistic or abstract nature, the automated scoring system could roughly find a few images with relatively high creativity and a few with relatively low creativity in a set of images.

5.2 Experiment on the human dataset

The human dataset contains ten human-themed images (See Figure 5.5). It contains images of different artistic creation styles of human beings, such as hand-drawn comics, abstract paintings, realistic portraits, etc. This group of images is selected on Google Images. Similar to the previous cat dataset, there is a certain tendency in the selection of images where some images have high creativity, and some have low creativity.

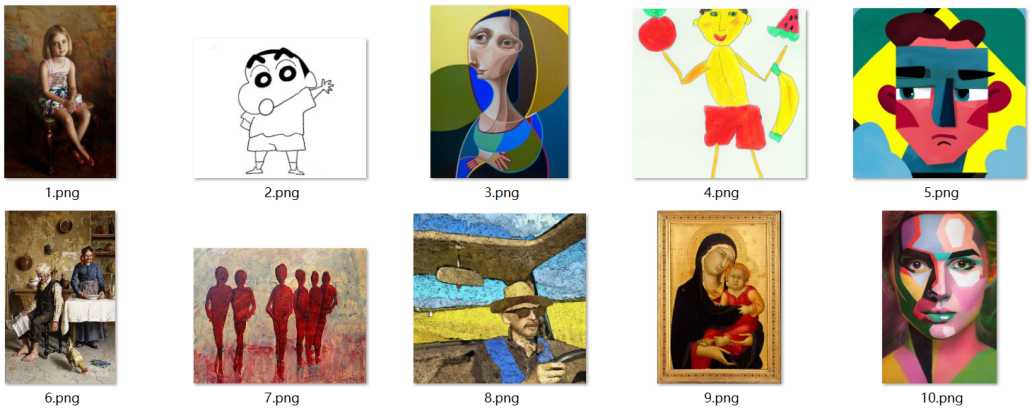


Figure 5.5: The human dataset collected from the Google images

Similarly, we calculate the Fluency, Flexibility, and Originality of each picture in this group of pictures separately. However, unlike the previous cat dataset, we use the weighted Combine creativity score to calculate the final result. Considering that this set of data sets is a human-themed data set, the focus of human-centered image creativity is not on the number of elements in a picture (Fluency) but on the compositional diversity of people in the entire image (Flexibility) and the uniqueness of characters (Originality). Therefore, a weighted Combined Creativity score is used with 0.6 on Originality, 0.3 on Flexibility, and 0.1 on Fluency.

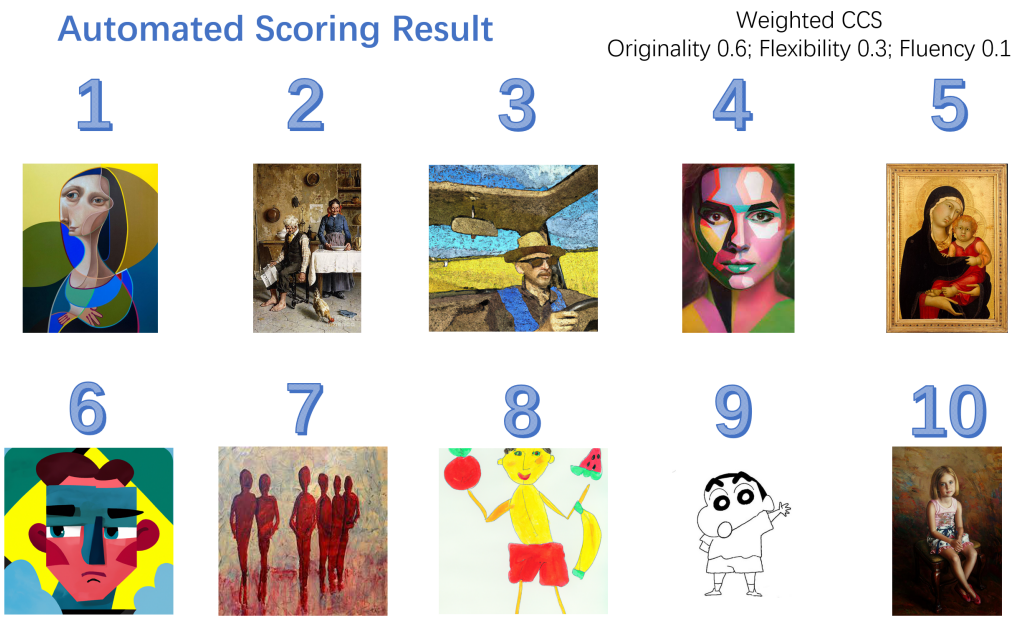


Figure 5.6: The automated scoring result on the human dataset ranked according to the combined creativity score with weighed CCS

Figure 5.6 and Figure 5.8 show the automated scoring result and human expert scoring result on the human dataset, respectively. Figure 5.7 shows the automated scoring details of the human dataset.

Images	Flexibility Score	Fluency Score	Originality Score	Weighted Combined Score	Ranking
Image 1	8 (0.43)	3 (0.33)	5.398 (0.0)	0.49	10
Image 2	5 (0.0)	1 (0.0)	5.771 (0.52)	0.94	9
Image 3	11 (0.86)	1 (0.0)	6.112 (1.0)	2.57	1
Image 4	6 (0.14)	4 (0.5)	5.967 (0.80)	1.71	8
Image 5	9 (0.57)	1 (0.0)	5.917 (0.73)	1.82	6
Image 6	12 (1.0)	7 (1.0)	5.804 (0.57)	2.22	2
Image 7	9 (0.57)	3 (0.33)	5.833 (0.61)	1.72	7
Image 8	10 (0.71)	5 (0.67)	5.927 (0.74)	2.18	3
Image 9	8 (0.43)	2 (0.17)	6.005 (0.85)	1.97	5
Image 10	10 (0.71)	1 (0.0)	5.954 (0.78)	2.04	4

Figure 5.7: The chart that contains the Flexibility score, Fluency score, Originality score, Weighted Combined Creativity Score, and corresponding rankings for all images in the human dataset

Compared Figure 5.6 with 5.8, we can find that both two scoring results put the image No.1 and image No.3 in the first and third places, respectively; and both two scoring results puts image No.1 , image No.2, and image No.4 in the last three among all images(although the relative order is different). Among them, the most significant difference is the ranking of the No.5 image and No.6 image. The Automated scoring system ranks image No.6 in the second place and image No.5 in the sixth place; on the contrary, the human expert ranks image No.6 in the sixth place and image No.5 in the second place with the reason:

- Image No.5 combines simple lines and geometric shapes to create a dynamic and vivid character. The color matching of red and green on the characters' faces can even reflect the melancholy emotions of the characters, making the whole work more profound and creative.
- Consider Image No.6; it is an oil painting created in a realistic style, the characters in it are realistically depicted, and the objects on the background wall of the characters are very realistic. The chicken on the ground, the sitting old man, and the old woman preparing food has produced a perfect positional relationship. However, the whole picture gives the feeling of a realistic and nostalgic style, which does not express the characteristics of creativity.

From figure 5.7, the high score of the No. 6 image is the highest in Flexibility and Fluency among all pictures. Although we use weighted CCS, the total score of the

Figure 5.8: The human expert scoring result on the human dataset ranked according to the combined creativity score with weighed CCS

first two items plus the Originality score(0.57) ranks it second. Compared with the No. 5 image, its Flexibility score is relatively average, and Fluency is the lowest among all images. Even though its Originality score is 0.73 (relatively high), the total score is only 1.82, and it ranked sixth. The originality scores of the two images are not much different. However, according to the feedback given by the human expert, the originality of picture No. 5 should be much higher than that of picture No. 6. This shows that it is not comprehensive enough to only use the similarity between a group of images as a measure of Originality. In a group of images, if there are several highly original and similar images in the form of composition at the same time, then using relative similarity alone as a measure will cause these images to have low originality (because they are very similar to each other).

A question that also needs to be considered is, Fluency, when only one element appears in the image, such as a person or an animal. It should be considered whether multiple other elements inside the object make up the object (such as different lines, polygons, etc.), and thus adopt rules for scoring these sub-elements. For example, we can count the number of polygons that make up the face instead of the sum of the different recognizable objects.

5.3 User Study

To further study the accuracy of the automated scoring system, I recruited 50 participants without any art background (mostly university students from different countries) to rank these two datasets.

These two images represent the degree of coincidence between automated ranking and

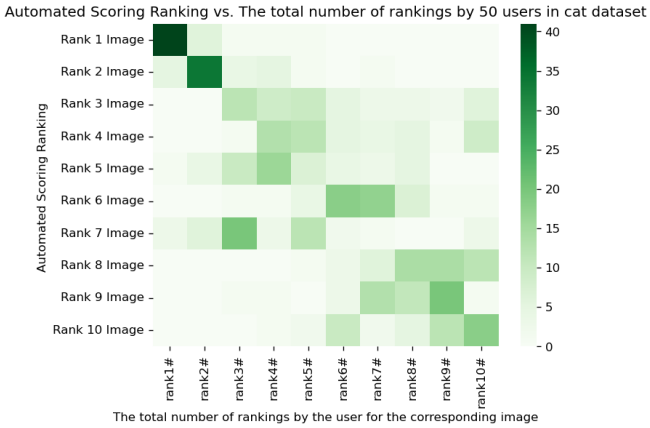


Figure 5.9: The heat map that shows the agreement between automated ranking and 50 users' ranking on the cat dataset

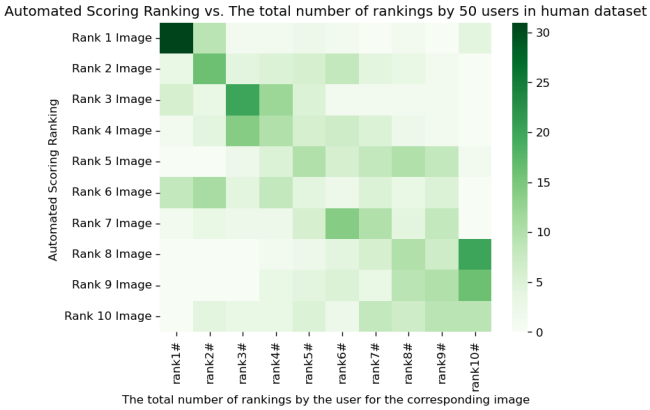


Figure 5.10: The heat map that shows the agreement between automated ranking and 50 users' ranking on the human dataset

the ranking of 50 users. The y-axis represents each image corresponding to automated ranking from 1 to 10, and the x-axis represents the total number of rankings of 50 users for different images from 1 to 10. For instance, the upper left square indicates the total number of users who ranked the image that is the first place in the automated scoring result as their first place. The colors represent the total number of identical selections by the user on different rankings and images (The higher the number, the darker the color). If all the colors are concentrated on this diagonal line from the upper left to the lower right, it means that the ranking result and the automated ranking result of users are consistent

From Figure 5.9, it is found that, in the cat dataset, the images ranked first and second in the automated scoring ranking were also ranked first and second (41 users and 34 users, respectively). It is also found that in the three-by-three area in the lower left corner, the dark color blocks are gathered in this area, which means that most of the 50 users chose the three pictures with the same lowest creativity as the automated scoring result. However, The big difference is that the image of "four white cats" ranked seventh in the automated scoring result and was ranked in the top three by most of

the 50 users. For the images ranked in the middle of the Automated scoring result, 50 users gave a relatively different ranking order. Still, the overall trend is centered on the diagonal line from upper left to lower right.

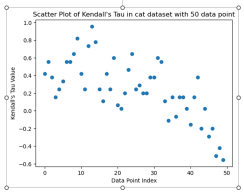
From Figure 5.10, it is found that, in the human dataset, most of the 50 users reached a consensus on the top three images that has the highest creativity with the automated scoring result. For the images ranked 8th and 9th by automated scoring, most of the 50 users ranked these two images in the rear position. Compared with the results of the cat dataset, according to the degree of dispersion of the color blocks, it could be found that 50 users have more different opinions on the ranking of images with medium creativity on the human dataset (although there is still a certain relationship with the diagonal line, they are more different than the result of cat dataset).

5.4 Kendall rank correlation coefficient

To figure out the degree of agreement between automated ranking and the users' ranking in a numerical way, the Kendall rank correlation coefficient[26] is presented. Figure 5.11 and Figure 5.12 show the scatter plot and histogram plot with corresponding statistics in both datasets.

Kendall's Tau on the cat dataset

Dataset	Mean	Median	Maximum	Minimum	Standard deviation
Cat dataset	0.246	0.244	0.956	-0.556	0.336



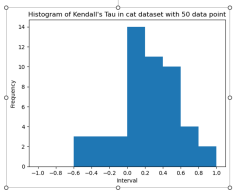
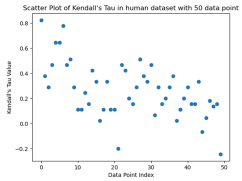


Figure 5.11: The combination of Statistics, scatter plots, and histograms of Kendall Tau scores from 50 users on the cat dataset

Kendall's Tau on the human dataset

Dataset	Mean	Median	Maximum	Minimum	Standard deviation
Cat dataset	0.265	0.266	0.822	-0.244	0.336



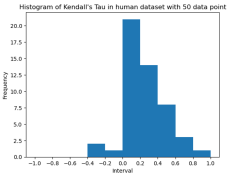


Figure 5.12: The combination of Statistics, scatter plots, and histograms of Kendall Tau scores from 50 users on the human dataset

By calculating Kendall's Tau score of 50 users on the two data sets, it can be found that the two data sets show similar results. The mean and median of Kendall's Tau score of the two data sets are around 0.25. According to the comparison of the scatter plot and the histogram, we can find that the 50 data points of each of the two sets of data are concentrated in the range of 0.0 to 0.6. At the same time, there are also some Kendall's Tau scores in the two data sets that exceed 0.6, and there are also a small number of Kendall's Tau scores that are lower than 0.0.

The experimental results show a moderate correlation between the automated ranking result and a large amount of the ranking results of 50 users, and a small part of the data shows a very strong correlation. However, due to people's subjective nature of creativity, some ranking results are opposite to the automated ranking result.

Chapter 6

Conclusions

6.1 Summary

In this project, we first introduce the idea of measuring creativity through the importance of creativity in human society. Then, we discuss two difficulties in measuring image creativity in traditional methods: 1) the subjectivity of creativity: each person may have a different definition of creativity; 2) traditional creativity assessments mostly rely on human experts, but it takes a lot of time and effort. Therefore, I propose an automated scoring system to assess the creativity of images that uses the Torrance Test of Creative Thinking (TTCT) as a measurement standard and uses a CNN-based deep learning model as a quantification method.

Inspired by TTCT, we divide the measurement of image creativity into three aspects: Fluency, Flexibility and Originality, where Fluency refers to the total number of different objects presented in an image; Flexibility refers to the number of different regions in which an image can be divided into; Originality refers to how similar an image is compared to other images.

In terms of implementation, we use different CNN-based deep learning models to quantify these three measurements. For Fluency, use the Maskrcnn pre-trained on the COCO dataset as a model, and use the instance segmentation method to identify the number of elements in the image. For Flexibility, use pre-trained Resnet-50 to extract image embedding, then use the K-means algorithm for clustering, and finally use elbow methods to determine the number of clusters (the number of clusters is the score of Flexibility). For Originality, similarly, use Resnet-50 to extract image embeddings, and then in a set of images, calculate the cosine similarity to determine the similarity of each image to the remaining images. Images with low total similarity have higher Originality scores.

In terms of experiments, we experimented with the automated creativity scoring method on two datasets, a cat-themed dataset generated by DALL-E 2 and a human-themed dataset collected on Google Images. We conducted comparative experiments of the automated scoring system and human expert scoring on these two datasets. The experimental results show that the automated scoring system can figure out the images with

higher creativity and those with lower creativity in a group of images. However, the Automated scoring system will judge some images that human experts think are more creative as images with lower creativity. It is because the existing scoring mechanism (especially the originality) cannot explain the human's definition of image creativity on an abstract level. Such as, it cannot connect the direction of the cat's eyes and the direction of the cat's fur; it cannot understand that sometimes the images with simple lines and clean color schemes were associated with higher creativity; it is unable to explain the higher creativity when two different objects were combined as a whole.

To further verify the feasibility of the automated scoring system, we invited 50 users to sort the two sets of data sets according to their level of creativity. We got similar results to human experts. Hence the conclusion is:

- The automated scoring system could distinguish the most and least creative images in a group of images.
- The automated scoring system could not quite correctly rank images with moderate levels of creativity due to the subjective nature of creativity.
- The measurement mechanism of Automate the scoring system makes it unable to explain people's definition and understanding of creativity at an abstract level which causes it defines some images that humans consider high-creativity as low-creativity.

6.2 Future Work

The defects exposed by the automated scoring system in the experiment show that our measurement method cannot fully explain people's definition of creativity at the abstract level. In future work, we need to complement our measurement method to strengthen it Interpretation of creativity at different levels.

In most cases, the understanding of creativity on this abstract level comes from the measurement of Originality. Only calculating the relative similarity of images in a group of pictures cannot fully explain the Originality of creativity of individual images. Therefore, we need to use more methods to measure Originality. Considering we have an image in Figure A.10, the combination of the clock head and the human body under the style of surrealism is a unique and imaginative art form that breaks through the traditional way of visual expression and arouses people's sensual resonance. The fusion of different elements forms a unique visual effect. Creativity evaluations for such works are based on their groundbreaking originality. Overall, this combination of a walking clock head and a human body in a surreal painting style has won high creative evaluations for its unique imagination, visual effects, and technical skills. However, it is easy for computer vision to recognize that this image contains a human body and a clock, respectively. But the difficulty lies in what kind of method we should use to make computer vision understand that these two objects are actually a creative fusion. A feasible method may be that we use some edge detection methods to judge whether different objects can form a new object. Then analyze the semantics of the identified objects to determine whether the combination of these objects is highly creative at the

Figure 6.1: Clocks walking in surreal style

semantic level. But this is a complex and uncertain work (due to the subjectivity of creativity), and we need to invest much time and energy in future work to understand similar abstract creativity and implement it using computer vision methods.

On the other hand, due to the subjectivity of creativity, we can conduct comparative experiments on different types of people in future work. For instance, we can select 50 people in different age groups to sort a group of images and then calculate the average Kendall Tau score of all people in each age group so that we can know which age group has a result that is closer to the automated scoring result. Then we adjust some parameters in the measurement method according to these results to have better robustness for assessing images created by different age groups. Similarly, we can conduct experiments on aspects such as gender, occupation, and artistic style hobbies to explore the influencing factors of creativity's subjectivity and how to better deploy the automated scoring system's creativity measurement method in different situations.

Bibliography

[1] Mark A Runco and Garrett J Jaeger. The standard definition of creativity. *Creativity research journal*, 24(1):92–96, 2012.

[2] E Paul Torrance. Predictive validity of the torrance tests of creative thinking. *The Journal of creative behavior*, 1972.

[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[4] Anastasia Kovalkov, Benjamin Paassen, Avi Segal, Kobi Gal, and Niels Pinkwart. Modeling creativity in visual programming: From theory to practice. *International Educational Data Mining Society*, 2021.

[5] Tony Lindeberg. Scale invariant feature transform. 2012.

[6] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[7] Ellis Paul Torrance. *Torrance tests of creative thinking: Norms-technical manual: Verbal tests, forms a and b: Figural tests, forms a and b*. Personal Press, Incorporated, 1966.

[8] Maria M Clapham. The convergent validity of the torrance tests of creative thinking and creativity interest inventories. *Educational and Psychological Measurement*, 64(5):828–841, 2004.

[9] Kyung Hee Kim. Can we trust creativity tests? a review of the torrance tests of creative thinking (ttct). *Creativity research journal*, 18(1):3–14, 2006.

[10] Bonnie Cramond. We can trust creativity tests. *Educational Leadership*, 52(2):70–71, 1994.

[11] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[14] Image embeddings. [Online], 2020. https://rom1504.medium.com/image-embeddings-ed1b194d113e.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[16] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[17] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[18] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, 9(3):171–189, 2020.

[19] The definitive guide to instance segmentation. [Online], 2023. https://www.v7labs.com/blog/instance-segmentation-guide.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[21] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9157–9166, 2019.

[22] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020.

[23] Cosine similarity. [Online], 2020. https://www.learndatasci.com/glossary/cosine-similarity/.

[24] J MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297. University of California Los Angeles LA USA, 1967.

[25] K-means clustering tutorial. [Online], 2018. https://rpubs.com/cyobero/k-means.

[26] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[27] David H Cropley and Rebecca L Marrone. Automated scoring of figural creativity using a convolutional neural network. *Psychology of Aesthetics, Creativity, and the Arts*, 2022.

[28] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[29] Ahmed Elgammal and Babak Saleh. Quantifying creativity in art networks. *arXiv preprint arXiv:1506.00711*, 2015.

[30] Margaret A Boden. Creativity and artificial intelligence. *Artificial intelligence*, 103(1-2):347–356, 1998.

[31] Gary Marcus, Ernest Davis, and Scott Aaronson. A very preliminary analysis of dall-e 2. *arXiv preprint arXiv:2204.13807*, 2022.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[33] Cosine maskrcnn_resnet50_fpn. [Online], 2020. https://pytorch.org/vision/main/models/generated/torchvision.models.detection.maskrcnn_resnet50_fpn.html.

[34] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[36] MA Syakur, BK Khotimah, EMS Rochman, and Budi Dwi Satoto. Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In *IOP conference series: materials science and engineering*, volume 336, page 012017. IOP Publishing, 2018.

[37] S Aranganayagi and Kuttiyannan Thangavel. Clustering categorical data using silhouette coefficient as a relocating measure. In *International conference on computational intelligence and multimedia applications (ICCIMA 2007)*, volume 2, pages 13–17. IEEE, 2007.

[38] David M Allen. Mean square error of prediction as a criterion for selecting variables. *Technometrics*, 13(3):469–475, 1971.

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[40] SGOPAL Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

Appendix A

DALL-E 2 Image Generating Prompts

A.1 The generating prompts for all images in the cat dataset

Figure A.1: Image No.1 Prompt: A colorful cat composed of different blocks

Figure A.2: Image No.2 Prompt: Four white cats with milky fur

Figure A.3: Image No.3 Prompt: One cat

Figure A.4: Image No.4 Prompt: A cat is eating a pizza

Figure A.5: Image No.5 Prompt: A cat is eating dinner with fork and knife

Figure A.6: Image No.6 Prompt: Two cats are playing with a bird

Figure A.7: Image No.7 Prompt: A cat is riding on a horse with chair beside it

Figure A.8: Image No.8 Prompt: A cat with tie is eating apples and bananas using forks

Figure A.9: Image No.9 Prompt: On a table with apples, bananas, oranges, pizza, hot dogs, a cat is making a chair and holding a fork

Figure A.10: Image No.10 Prompt: A cat is watching tv with different transportation tools