

Article

Design on Intelligent Feature Graphics Based on Convolution Operation

Ying Li ¹ and Ye Tang ^{2,3,*} ¹ School of Arts, Anhui Polytechnic University, Wuhu 241000, China; ly408493042@163.com² School of Mechanical Engineering, Anhui Polytechnic University, Wuhu 241000, China³ Department of Mechanics, Tianjin University, Tianjin 300350, China

* Correspondence: tangye2010_hit@163.com; Tel.: +86-188-9531-6533

Abstract: With the development and application of artificial intelligence, the technical methods of intelligent image processing and graphic design need to be explored to realize the intelligent graphic design based on traditional graphics such as pottery engraving graphics. An optimized method is aimed to be explored to extract the image features from traditional engraving graphics on historical relics and apply them into intelligent graphic design. For this purpose, an image feature extracted model based on convolution operation is proposed. Parametric test and effectiveness research are conducted to evaluate the performance of the proposed model. Theoretical and practical research shows that the image-extracted model has a significant effect on the extraction of image features from traditional engraving graphics because the image brightness processing greatly simplifies the process of image feature extraction, and the convolution operation improves the accuracy. Based on the brightness feature map output from the proposed model, the design algorithm of intelligent feature graphic is presented to create the feature graphics, which can be directly applied to design the intelligent graphical interface. Taking some pottery engraving graphics from the Neolithic Age as an example, we conduct the practice on image feature extraction and feature graphic design, the results of which further verify the effectiveness of the proposed method. This paper provides a theoretical basis for the application of traditional engraving graphics in intelligent graphical interface design for AI products such as smart tourism products, smart museums, and so on.

Keywords: convolution operation; intelligent graphic design; brightness feature; traditional engraving graphics



Citation: Li, Y.; Tang, Y. Design on Intelligent Feature Graphics Based on Convolution Operation. *Mathematics* **2022**, *10*, 384. <https://doi.org/10.3390/math10030384>

Academic Editors: Xiang Li, Shuo Zhang and Wei Zhang

Received: 1 December 2021

Accepted: 24 January 2022

Published: 26 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, with the development of artificial intelligence (AI), the application of intelligent image processing has been found in many fields such as handwriting recognition, image caption, automatic vehicle navigation, and so on [1–4]. As a deep learning network in the field of artificial intelligence, the convolutional neural network (CNN) has been maturely used in image processing field, especially for the image recognition [5,6] or image feature extraction. For example, Ding et al. studied the intelligent image identification method to roughly handle the express packages by using the intelligent recognition method of the gated recursive unit in the convolutional neural network, which can be used as an intelligent fusional model [7]. Li et al. investigated the extracted method of global features from images of typical infrared targets such as people and vehicles by designing semantic segmentation algorithms, and achieved good results [8]. As the core computing module is acted in convolutional neural network, the convolution operation plays an important role in intelligent image processing [9] and image feature extraction. For example, Zhan studied the method of extracting image features of tea materials by convolution operation [10]. Chen studied the method and effect that extracted brightness features in the interest regions from the images by the improved Itti–Koch model based on the convolution operation [11].

By the other side, there is a rapid, essential, and global change in the graphic design due to the effect of artificial intelligence [12]. AI has been widely used by designers, which is considered to be an important assistant for them [13]. Intelligent algorithms play an increasingly important role in graphic design field such as real-time graphics generation, virtual scene visualization [14], 3D Graphics Engines [15], and so on. In Ref. [16], Li and Xv studied the generation and conversion method of the woodcut print style by applying a deep learning algorithm, which can be used to generate a wood engraving texture effect. In Ref. [17], Tian et al. discussed the generation method of multi-style ancient book textures via the layout analysis and style transfer techniques based on the deep learning. In Ref. [18], Liu studied the method of generating image features such as specific color, shape, and texture by using the synthetic method of artificial intelligence and data mining. On the basis of the previous studies, we use the technical method of intelligent image processing and intelligent graphic design to realize the intelligent graphic design based on traditional graphics such as pottery engraving graphics. For this purpose, the image feature extracted model is constructed to extract the image features from and apply them to intelligent graphic design with the algorithms for intelligent graphic design. This research will promote the spreading of traditional culture and artistic features in the field of intelligent application.

2. Method of Image Feature Extraction Based on Convolution Operation

2.1. Principle of Image Feature Extraction

During the 1950s and 1960s, Professor Frank Rosenblatt of Cornell University, invented the perceptron by imitating the visual system architecture of automatic pattern recognition of human body [19]. The perceptron is a simple learning algorithm, which plays an important role in the AI field. As the early prototype of deep learning network mentioned in [20], the perceptron consists of an input layer, an output layer, and a set of structures connecting with them, which is called a hidden layer in a deep neural network [21]. It can recognize, extract, and classify the images input into the perceptron through the judgment of input information. The process of machine recognition is to divide a complete picture into many small parts, extract and summarize the features in each small part, which is realized based on the perceptron. Image features such as brightness, pixel strength, and contour are extracted from the original input image and weighted, which are used as the basis of image feature recognition and classification.

2.2. Image Feature Extraction Method Based on Convolution Operation

Based on the principle of perceptron, convolutional neural network is developed. Image features can be extracted based on the structure of convolutional neural network, in which the central role is existed in convolution operation. When it is applied in the image feature extraction, the basic structure consists of the feature extraction and mapping layers [22]. It is advantaged that images can be directly input with the form of three-dimensional data [23], resulting in reducing the preprocessing process of the input original signal, and weakening the complicated extent of the recognition model by sharing the weight and worth. According to the structure of convolutional neural network, the method of image feature extraction can be described as follows.

Convolutional layer is used to extract preliminary image features. The image feature extraction is obtained via the convolution operation, of which the process contains as: inputting the original images as pixel matrices, and then setting the convolutional kernel to move on and cover the pixel matrix of original image sequentially, in which the moving interval unit in each time is called step. It realizes the extraction of the image feature such as the brightness, pixel intensity, and outline etc., by the judgment of weighted sums of convolutional kernel and the covered image pixel matrix in each movement of convolutional kernel.

Pool layer is used to enhance and extract the main features of the image. The working principle of the pooling layer is to multiply the original data output from the convolutional

layer with the corresponding convolutional kernel to obtain a new matrix, which is used to strengthen and extract the main features of the image.

The full connection layer is used to summarize and output the features of each part of the image. The working principle of the full connection layer is to convert the vector matrix output by the pooling layer into some vectors, multiply it by the weight matrix, add the offset value, and then use the ReLU (rectified linear units) activation function to optimize its parameters.

3. Image Feature Extracted Model

Based on the principle of perceptron, convolutional neural network is developed, which plays an important role in the image feature extraction. Then, we use the structure of convolutional neural network to propose the new method of image feature extraction, and construct the image feature extracted model based on convolution operation. Then, the image brightness processing is adopted to further optimize this extracted model.

3.1. Image Brightness Feature Processing

Because the shapes of traditional engraving graphics on historical relics mostly take the form of lines and gullies, formed by pressing and engraving, there is a significant brightness difference between the carved lines and the surrounding of the images. Therefore, the brightness features can reflect the features of the lines in the image well. According to the image features of the traditional engraving graphics on historical relics, an algorithm for extracting brightness feature from images is designed. The image feature extracted model is constructed based on convolution operation for the batch extraction of sample image features. To highlight the brightness features of the image, simplify the process of feature extraction, and improve the accuracy, the image brightness feature processing is performed before using the convolution operation to extract the image feature of traditional engraving graphics. Taking the pottery engraving graphics from the Neolithic Age as an example, the image brightness processing and image feature extraction are carried out. The image brightness processing comprises a series of brightness feature operations, such as conversion and enhancement of image brightness value as well as the image threshold processing.

1. Conversion of image brightness value is taken to convert the storage mode of image information. It is known that the images are stored by the form of color value matrix in the computer. In addition, it is more convenient for the image feature extraction by converting the image storage mode from the color value matrix to the brightness value matrix, as showed in Figure 1.

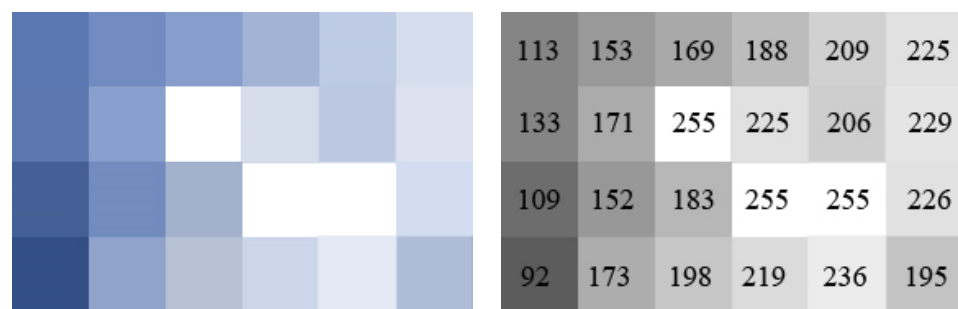


Figure 1. The conversion from color value mode to brightness value mode for image.

2. Enhancement and inverse operation of the image brightness value are exerted on the image to further emphasize the brightness feature of engraving lines, which is more convenient for extracting the image feature of line part according to the high value.

We use Equation (1) to realize the conversion of and enhancement of image brightness value, which yields

$$L = 255 - k \frac{r + g + b}{3} \tag{1}$$

where r , g , and b represent the color value matrices of red, green, and blue, respectively, and L is the brightness value matrix. Taking the 112×112 -px image of pottery engraving graphic in Figure 2a as an example, the RGB color value of the pixel at coordinates (69, 69) is (4, 14, 151) and the brightness value is considered to be 56. By using this method, the image storage mode of the input image is converted to a brightness value matrix. k is the brightness enhancement coefficient, which represents the multiple of brightness enhancement compared with the original image. It is demonstrated that with the increase of the whole image brightness, the engraving lines can be displayed better. For example, most of the pottery engraving lines can be revealed better as the whole image brightness is increased to 2.5 times.

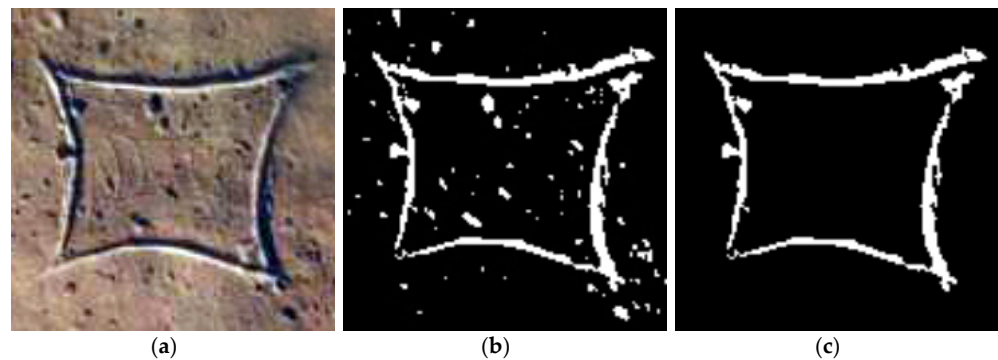


Figure 2. The image brightness processing of the pottery engraving graphic. (a) Input image. (b) Image threshold processing. (c) Removing discrete points.

3. Threshold operation is exerted on the image to further magnify the brightness features. We use Equation (2) to realize the image threshold processing, which yields

$$L(x, y) = \begin{cases} \text{maxval} & \text{if } L(x, y) > \theta_T \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

when the brightness value of the pixel $L(x, y)$ is greater than the threshold θ_T , it is set to the maximum value maxval , but is set to 0 in other cases. If maxval and θ_T are set as 255 and 128, respectively, the resulting image is as shown in Figure 2b, it is concluded that the binary conversion of image information is realized in the method and the brightness feature is highlighted to the utmost extent. To further enhance the image feature of lines, we use the method of `remove_small_objects` (T_A) to remove scattered small areas of an image, and the size of removed area can be controlled by setting the connected area threshold T_A . After these operations, the brightness value map is obtained when the connected area threshold value $T_A = 25$, as shown in Figure 2c.

After a series of image brightness processing including the conversion and enhancement of the image brightness value as well as the image threshold processing, the image brightness feature is more obvious, because the binary conversion of image brightness value is achieved and the engraving line parts is highlighted with the higher brightness values. It greatly simplifies the extraction process of image features and improves the accuracy.

3.2. Image Brightness Feature Extraction Based on Convolution Operation

The image brightness feature is extracted based on the brightness value map output from the image feature extracted model, which can greatly simplify the extracted process of image brightness feature and improve the accuracy. Convolution operation is applied to

extract image features from the image matrix input with the defined convolution kernel. The method can be expressed as

$$conv = \sigma(imgMat \otimes K_i + b) \tag{3}$$

in which σ represents the active function, $imgMat$ is the image brightness value matrix, K_i is the defined convolution kernel with the size i , \otimes means the convolution operation, b is the offset value. The convolution kernel is applied to extract the image feature and obtain an initialized set of feature vectors. Here, we use the Sobel convolution kernel. Because the image features have been highlighted greatly after the processing of image brightness feature, more accurate feature vectors can be obtained by a simple convolution operation. Only two convolution kernels are applied to extract the brightness feature of the image, and they can also obtain an obvious result.

Sobel—G(x) and Sobel—G(y) convolution kernels are used to carry out convolution operation with the input images, which respectively represent the kernels of horizontal and vertical directions. Substituting Sobel—G(x) and Sobel—G(y) into Equation (3) respectively, and then adding each element of this matrix with the offset value b , and inputting the results into the activation function yields

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

Combining the results in the convolution operation derived by G(x) and G(y) convolution kernels, we get the best feature vector matrix, which can be manifested as a feature map in python environment. After the brightness value processing, the image feature extraction has been greatly simplified because the input image is displayed with the binary mode of black and white. Therefore, we can get good extraction results for image features with only twice convolution operations derived by G(x) and G(y) convolution kernels.

3.3. Image Feature Extracted Model

The image feature extracted model based on the above research is shown in Figure 3. Sample images are put into this model, and the brightness feature matrix is obtained by convolution operation after a series of image feature processing including the conversion and enhancement of the image brightness value as well as the image threshold, which can be displayed as the brightness feature map used for the intelligent graphic design in python environment.

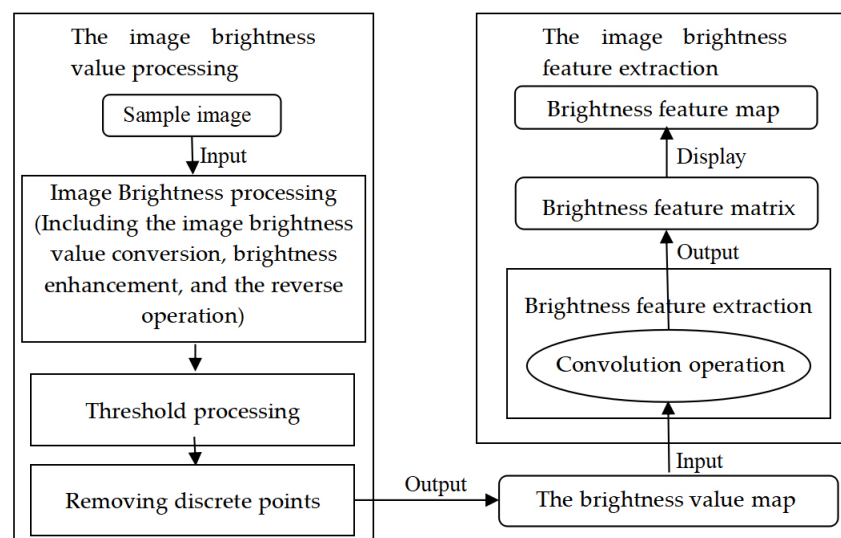


Figure 3. The image feature extracted model.

3.4. Model Parametric Test and Effect Evaluation

3.4.1. Model Parametric Test

Model performance based on the parametric test is carried out by using simple variable method, adjusting one parameter while other parameters are fixed and comparing with the results, we get the optimal parameters. The specific process involves collecting the original images and inputting them into the extracted model to test its extracted effect and find the optimal parameters. Taking the pottery engraving graphics from the Neolithic Age for example, selecting 50 sample images of representative pottery engraving graphics from Shuangdun Site in Bengbu City of Anhui province as the test samples, we input them into the model after preprocessing the images, and conducted brightness feature processing and extraction. We compared the effects of brightness feature maps, as obtained by different parameters to find that when the brightness enhancement coefficient k increases the other parameters remain fixed; the image feature extraction effects are shown in Figure 4a. Figure 4b exhibits the influence of the threshold θ_T on the image feature extraction. It is concluded from Figures 4 and 5 that the extracted effect of image brightness feature is closely related to the model parameters, which directly affects the results of intelligent graphic design.

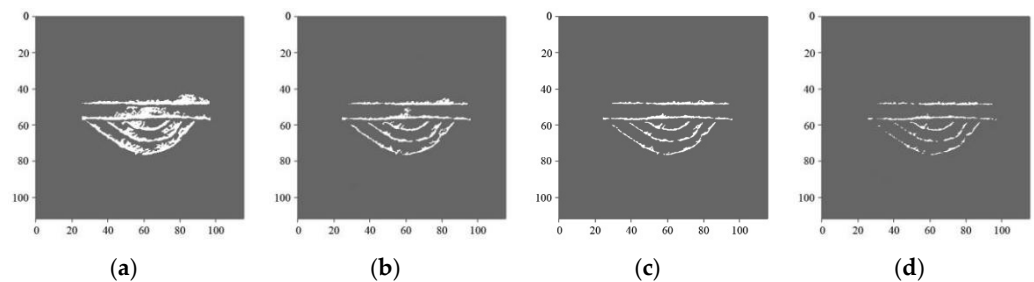


Figure 4. Influence of brightness enhancement coefficient k on image feature extraction. (a) $k = 1.5$, $\theta_T = 128$. (b) $k = 2.0$, $\theta_T = 128$. (c) $k = 2.5$, $\theta_T = 128$. (d) $k = 3.0$, $\theta_T = 128$.

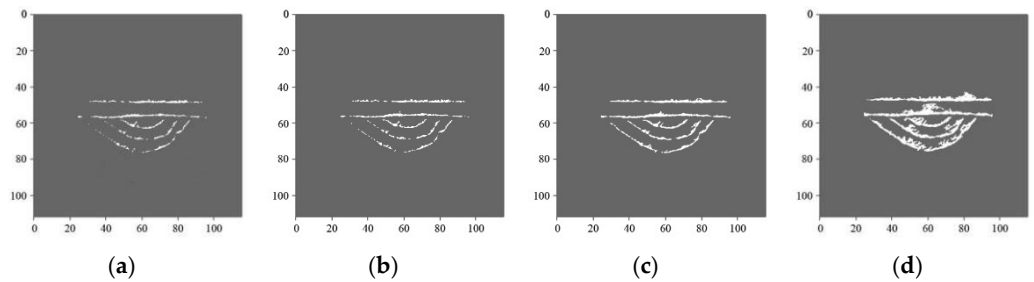


Figure 5. Influence of threshold θ_T on image feature extraction. (a) $k = 2.5$, $\theta_T = 96$. (b) $k = 2.5$, $\theta_T = 112$. (c) $k = 2.5$, $\theta_T = 128$. (d) $k = 2.5$, $\theta_T = 140$.

In the image brightness value processing, we adjust the connected area threshold T_A of the `remove_small_objects()` method to remove the most discrete points, and adjust the brightness enhancement coefficient k and the brightness threshold θ_T to enhance the image brightness features. From the sample testing, we conclude that, when the connected area threshold $T_A = 25$, independent discrete points can be removed well, preserving the effective parts of the engraving lines. When the brightness enhancement coefficient $k = 2.5$ and the image brightness threshold $\theta_T = 128$, the good displayed effects are obtained and the feature maps are more true, accurate, and distinctive, compared with the original image. In the image brightness feature extraction, we define the horizontal convolution kernel $G(x) = [-1, 0, 1; -2, 0, 2; -1, 0, 1]$, and the vertical convolution kernel $G(y) = [1, 2, 1; 0, 0, 0; -1, -2, -1]$. We found that the better results are achieved under these convolution kernels. When the offset value $b = 2$, the locations of the feature maps are more accurate for most of the 112×112 images of the pottery engraving graphics. The parameters are adjusted to

tailor an accurate effect of the feature map for a few images when the extracted effect is not ideal.

3.4.2. Model Effectiveness Research

In order to evaluate its stability and availability, model test is conducted. Total of 50 sample images of traditional engraving graphics are inputted into the model to extract their image features. Test results are shown in Table 1, and it illustrates that image feature extraction of 45 traditional engraving graphics achieves a good result under standard parameters condition. But the defect results are obtained for the few sample images, and the extracted feature can be tuned/tailored well by adjusting the modal parameters slightly. Therefore, the application of the proposed model to the image feature extraction of traditional engraving graphics results in very good efficiency. We can conclude that the results of the image feature extraction based on the proposed model are accurate and clear for most traditional engraving graphics.

Table 1. Model test results statistics.

Number of Pottery Engraving Graphics	45	2	1	1	1
Sets of parameters in image brightness processing	$k = 2.5;$ $\theta_T = 128;$ $T_A = 25$	$k = 2.5;$ $\theta_T = 150;$ $T_A = 25$	$k = 2.0;$ $\theta_T = 128;$ $T_A = 25$	$k = 2.0;$ $\theta_T = 150;$ $T_A = 25$	$k = 3.0;$ $\theta_T = 128;$ $T_A = 25$
Sets of parameters in convolution operation	$G(x) = [-1, 0, 1; -2, 0, 2; -1, 0, 1]; G(y) = [1, 2, 1; 0, 0, 0; -1, -2, -1]; b = 2$				

We also compare the extracted image features, as obtained by the proposed method and complex convolutional neural networks and algorithms mentioned in literatures [8,10,11], as shown in Figure 6. It is shown that this proposed method is more simple and effective, the reason being that the binary conversion of image brightness value is achieved via conducting the image brightness processing, which consequently simplifies the process of extracting image features by virtue of the convolution operation. Furthermore, we find that image features extracted by using complex convolutional neural networks and algorithms have more detailed information, but the image features are not prominent. Image features extracted by the proposed model reflects the image features more clearly and accurately, which is more suitable for designing the feature graphic in the intelligent user interface and conveying the line features well. It is of superiority that this proposed method is used in the field of intelligent graphic design for traditional engraving graphics.

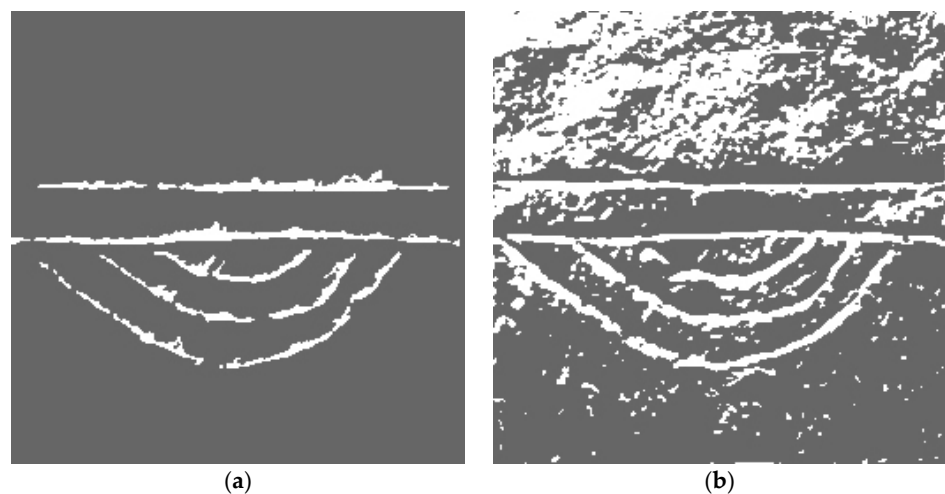


Figure 6. Comparison of image feature extraction with and without image brightness processing. (a) The proposed results. (b) The results obtained by the complex algorithms.

Based on the parametric test and effectiveness evaluation of the proposed model, we establish its execution metrics as: this extracted feature model is suitable for traditional engraving graphics on historical relics in intelligent graphic design. Through the parametric test, it can achieve an ideal effects with the optimal parameters, which are derived as the brightness enhancement coefficient $k = 2.5$, the image brightness threshold $\theta_T = 128$, the connected area threshold $T_A = 25$, the horizontal convolution kernel $G(x) = [-1, 0, 1; -2, 0, 2; -1, 0, 1]$, and the vertical convolution kernel $G(y) = [1, 2, 1; 0, 0, 0; -1, -2, -1]$, $b = 2$. However, we should adjust the parameters to obtain the appropriate effect for the non-ideal sample images.

4. Intelligent Graphic Design Based on the Brightness Feature Map

According to the brightness feature maps output from the extracted model based on convolution operation, intelligent feature graphics are generated by using the python graphic tools, and then it is applied to design the intelligent graphical interface.

4.1. Created Method of Intelligent Graphics

Many traditional engraving graphics are composed of basic lines and graphics. Their basic graphic units are defined as dots, straight lines, oblique lines, curves, arcs, circles, and so on, which can be created by python graphic tools. Python environment provides powerful graphic tools for drawing regular geometry graphics. Taking the python graphic tool of turtle for example, the general graphic drawing method is listed in Table 2.

Table 2. Drawing method of common geometric lines with python-turtle tools.

Graphics	Drawing Method	Describing
Round dot	dot (r)	Draw a round dot of specified radius (r).
Straight line	forward ($distance$) backward ($distance$)	Move a distance ($distance$) forward or backward to draw a straight line.
Oblique line	Straight line + right ($degree$) Straight line + left ($degree$)	Draw oblique lines with a clockwise or anti-clockwise angle ($degree$) based on the straight line method.
Circle	circle (r)	Draw circles with a radius (r).
Arcs	circle() + up() + down() + left() + right()	Draw arcs with different curvature by changing in different direction of up, down, left and right based on the circle method.
Curve	circle() + up() + up() + left() + right() + seth()	Draw curves with different curvature and shape by changing of angle and directions based on the circle method.

Based on the brightness feature map, more accurate graphics can be generated by Python graphic drawing tools and displayed on the user interface of intelligent products. We can also use a lots of mathematical functions provided for array operation such as sine and cosine functions, tangent and cotangent functions, linear function, quadratic function, and so on to generate geometric lines under some geometric laws. Figure 7 shows some geometric lines created by python graphic tools.

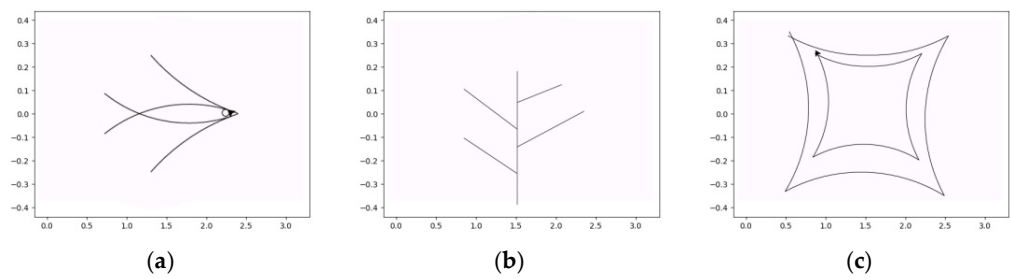


Figure 7. Geometric lines generated by python graphic tools. (a) Fish-shape graphic. (b) Tree-shape graphic. (c) Double-box-shape graphic.

Based on the created graphics, dynamic graphics can also be created, which are commonly applied in intelligent graphical interfaces. Recent research show that dynamic graphic system shows superiority in data exchange and information understanding [24,25], especially on intelligent user interface. In their study, the visual performance of dynamic graphics system in information dissemination is analyzed and evaluated. For intelligent user interface, Chen and Jiu studied a rapidly convert system of 3D dynamic graphics, which is suitable for the Android platform, and evaluated the stability and effectiveness of the algorithm for generating 3D dynamic graphics [26]. Technical method and model for generating dynamic graphics have also been paid attention to. For example, Ding studied the technical method of dynamic graphics with Visual C++ 2005 in the frame of Net framework to realize the dynamic change of points, lines, rectangles, and other shapes [27]. Castillo etc., proposed a dynamic graphic model associated with the graphic structure and studied the transition-probabilities of the proposed model by the method of unobservable variables via estimating the model parameters [28]. Based on the existing studies, an effective method for generating dynamic graphics is necessary to be applied for designing intelligent graphical interface. In this study, we explore an accurate and effective generated method of dynamic graphics in python environment. The specific method used to generate dynamic graphics includes: the basic animation environment is built, the graphic change method is defined and the graphic animation is generated, as shown in Figure 8.

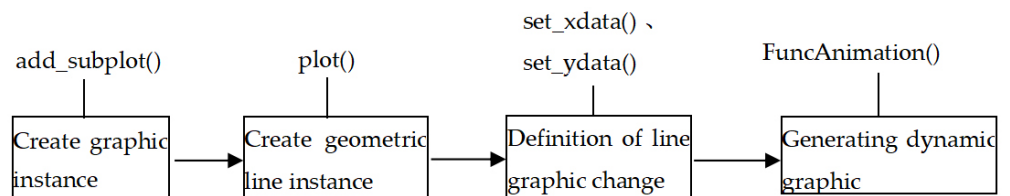


Figure 8. Flow chart of generated method of dynamic graphic.

The animation environment is created by using the matplotlib tools, in which graphic instances are then created by using the plot() method. The change method of line graphic from these graphic instances is defended with the data change in x or y directions, which is set with the methods of set_xdata() and set_ydata(). Then the graphic animation is generated by calling the animation class of FuncAnimation with the graphic change method defined in advance. Figure 9 shows the dynamic graphic of double box line generated by the above method.

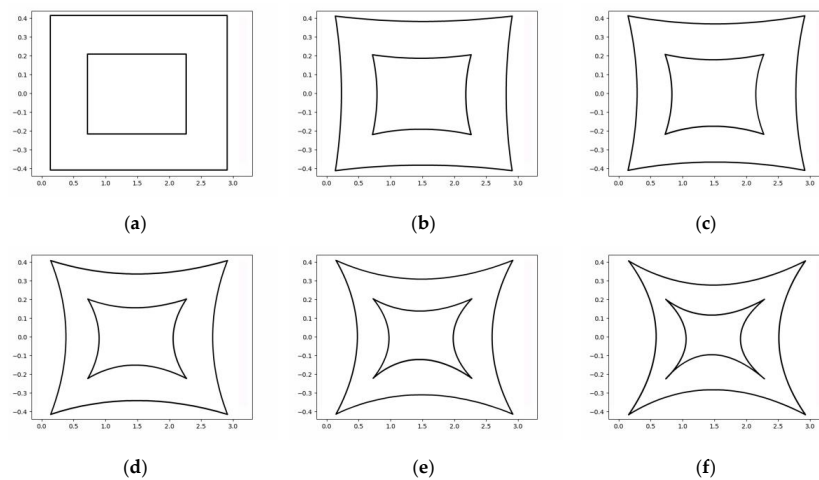


Figure 9. Dynamic graphic of double box line created by generated method of dynamic graphics. (a) Dynamic graphic at frame 1. (b) Dynamic graphic at frame 3. (c) Dynamic graphic at frame 4. (d) Dynamic graphic at frame 6. (e) Dynamic graphic at frame 8. (f) Dynamic graphic at frame 10.

4.2. Created Method of Intelligent Feature Graphics Based on Brightness Feature Map

Traditional culture and artistic features can be conveyed in the field of intelligent applications by applying the feature graphics of traditional engraving graphics into the intelligent graphical interface design. So, the created method of feature graphics including dynamic feature graphics based on the brightness feature map output from the extracted model is researched.

4.2.1. Created Method of Feature Graphics Based on Brightness Feature Map

Intelligent feature graphics are created based on the image brightness feature map output from the proposed model. Specifically, the process includes feature point sampling, optimization, and connection, as shown in Figure 10.

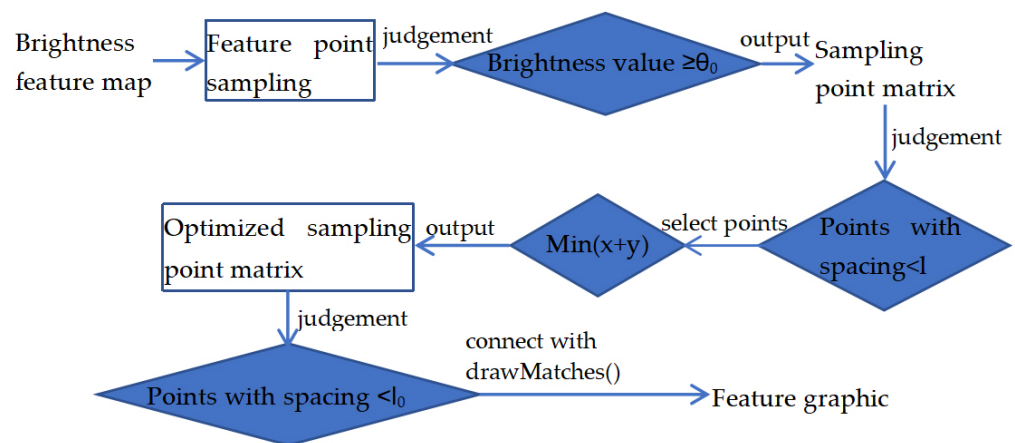


Figure 10. Flow chart of intelligent graphic created algorithm.

Step 1. Feature point sampling. The output brightness feature map is sampled, and the appropriate step size is set as the sampling interval in X and Y directions. In the array of $img_array[n \times s, n \times s]$ (where s is the sampling interval, n is the number of samples, and $0 \leq n \times s \leq 112$), the pixels whose brightness value is greater than the preset brightness threshold θ_0 are sampled and output, forming the sampling matrix of feature points. This is the initial pix point sampling process of the image brightness feature map.

Step 2. Optimizing the sampling point matrix. To facilitate the connection of feature points, the sampling point matrix is further optimized, and the optimized spacing l is set. During the sampling point the spacing is less than l , the point with $\min x + y$ is

output, forming the optimized sampling point matrix, where x and y respectively are the horizontal longitudinal coordinates of the sampling point. We obtain key effective points of the sampling matrix of the image brightness feature.

Step 3. Connecting the optimized sampling points to form the feature graphics. The optimized sampling points are connected to form the feature graphics, and we set the connection space as l_0 (usually $l_0 > l$). If the space is less than the set value l_0 , the optimized sampling points are connected by the drawing method of `drawMatches()` for feature point matrix to form the feature graphics, and some smoothing calculations are performed.

4.2.2. Generated Method of Dynamic Feature Graphics

Dynamic feature graphics is generated to realize the dynamic displayed effect of feature graphics on the user interface of AI products. We use the generated method of dynamic graphic mentioned above to generate dynamic feature graphics. The specific implementation method is described as follows.

We select some points such as the middle point of a line, an intersection point, or the highest (or lowest) point of an arc of feature matrix points as the key frame feature points on the line. By setting the data change in the X or Y direction, the key frame feature points drive other feature points to form dynamic feature graphics. Specifically, the implementation method is divided into the following key steps. 1. The basic animation environment is built, and a graphic instance is created to load the feature graphics. 2. The change of the graphics is defined, with the key frame feature points as the reference, we set the date change of graphic data in the X or Y direction through the methods `set_xdata()` and `set_ydata()`, to define the animation method. 3. The animation method defined in the previous step is used and the animation class `FuncAnimation` is called to realize the dynamic effect of feature graphics. Taking the dynamic graphic of a line graphic for example, the pseudo code for algorithm implementation can be described as follows briefly:

```
# import graphic and animation tools:
import matplotlib.pyplot as plt from matplotlib;
import FuncAnimation as animation class;
# create graphic instance:
fig = plt.figure();
ax1=fig.add_subplot(num, num, num);
line1= ax1.plot(num, curve, color, linewidth);
# set keyframe feature points:
point_ani=plt.plot(x1, y1, 'point1', x2, y2, 'point2', ... );
# def updata(num):
point_ani.set_data(x[num], y[num]);
return [point_ani];
# generate dynamic graphic:
ani = animation.FuncAnimation(fig=fig,func=updata,frames=np.arange(num,num), interval=num).
```

5. Practice on the Brightness Feature Extraction and Feature Graphic Design

Taking a square engraving graphic from the Shuangdun Site as an example, the practice is conducted on the image feature extraction and intelligent feature graphic design. Specifically, the process is given as follows.

1. Image brightness feature extraction;

The 112×112 image after preprocessing is input into the image feature extracted model. With the optimal parameters are set as the brightness enhancement coefficient $k = 2.5$, brightness threshold $\theta_T = 128$ and connected region threshold $T_A = 25$, the brightness feature map output from the extracted model is shown in Figure 11a.

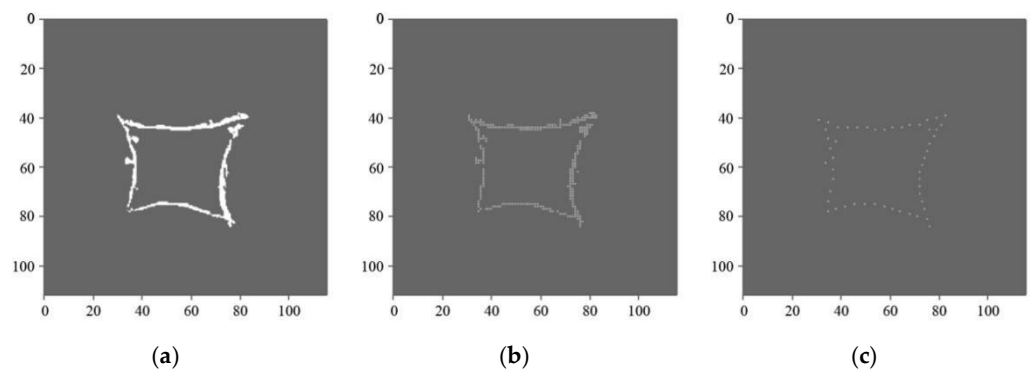


Figure 11. Feature pixel point sampling and optimization based on image brightness feature map. (a) Brightness feature map. (b) Initial sampling. (c) Sampling optimization.

2. Creating feature graphics;

The brightness feature map is sampled and optimized, where the sampling interval $s = 2$, the number of samples $n = 56$, and the optimal spacing $l = 4$ pixel units. The preliminary sampling result is shown in Figure 11b and the optimized sampling result is shown in Figure 11c. The specific process is that we take the progressive scanning and interval sampling to the image brightness feature map output from the proposed model with the sampling number $n = 56$ and the sampling interval $s = 2$. Then, we select the pixel points if their brightness value is higher than the preset brightness threshold $\theta_0 = 150$ and output them to form the brightness feature sampling matrix. Further optimization is conducted with the optimized spacing $l = 4$. If the space between the sampling points is less than $l = 4$ pixel units, we reserve the pixel point with $\min x + y$ to form the key feature point matrix. At last, the drawMatches() method is used to connect the optimized sampling points whose spacing is less than the set value $l_0 = 6$, and the feature graphics is formed, which consequently shows that the lines are partially smoothed.

3. Generation of dynamic feature graphic.

Dynamic feature graphic is generated by the generated method of dynamic graphic mentioned above, as shown in Figure 12. In the process of generating dynamic graphic, the key frame feature points are the middle points in each line. By using this method, dynamic feature graphic is generated, which can be applied to design the intelligent graphical interface. The displayed effect of this dynamic graphic shows that the better dynamic line graphics are obtained after some smoothing, which can accurately convey the image features of traditional engraving graphics and bring users to be a good visual experience.

The practical results show that the image feature extracted method proposed in this paper can effectively extract the image features of traditional engraving graphics. The feature or dynamic feature graphics created by the proposed method about the intelligent graphical design can be applied to the intelligent application interface, which can convey the image features of traditional engraving graphics well.

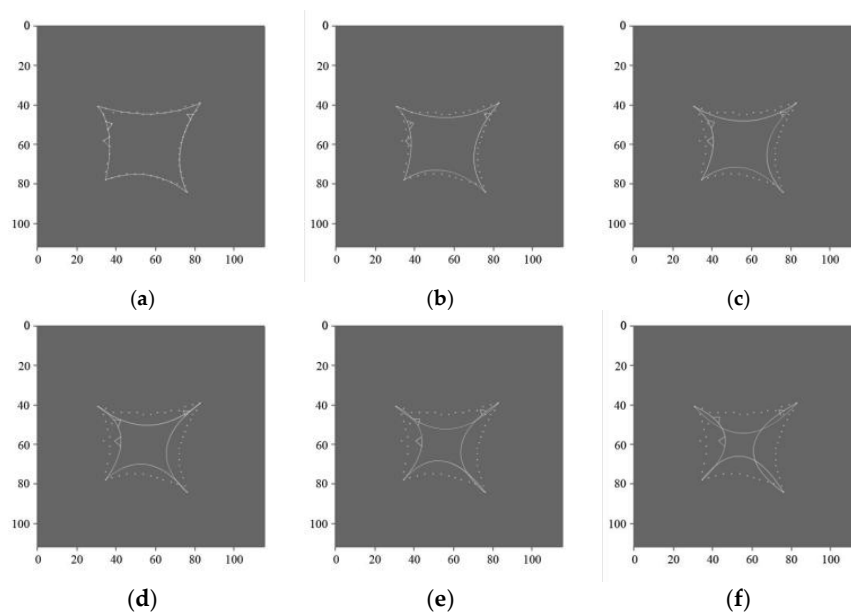


Figure 12. Dynamic graphic of square shape lines after smoothing. (a) Dynamic graphic at frame 1. (b) Dynamic graphic at frame 4. (c) Dynamic graphic at frame 7. (d) Dynamic graphic at frame 10. (e) Dynamic graphic at frame 13. (f) Dynamic graphic at frame 15.

6. Conclusions

In this study, we present the image feature extracted method, in which the image features are extracted by convolution operation after a series of image brightness feature processing. On the basis of the brightness feature map output from the proposed model, feature graphics are designed and applied on the intelligent graphical interface. Theoretical and practical research shows that the extracted model has a significant effect on the performance of the image features extracted from traditional engraving graphics. Moreover, the effect of image feature extraction is related to model parameters, which further affect the feature graphic design. The advantage of this method is that the brightness feature processing greatly simplifies the process of image feature extraction, and the extracted accuracy is improved by convolution operation. Take the pottery engraving graphics from the Neolithic Age as an example, the practice of the image feature extraction and intelligent dynamic graphic design is carried out, which further verify the effectiveness of the proposed method, especially in the field of intelligent feature graphic design and application. However, due to the limit of the sample amount and experiment condition, the image feature extracted model needs to further improve the extracted accuracy. The design efficiency of intelligent graphics needs to be further enhanced by undergoing more practice. More in-depth and extensive research needs to be taken in the next work.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L.; software, Y.L. and Y.T.; validation, Y.L. and Y.T.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets supporting the conclusion of this article are included within the article.

Acknowledgments: This work is supported by the Key Research Project of Humanities and Social Sciences in Colleges and Universities of Anhui Province (no.SK2019A0121); Support Program for Outstanding Young Talents in Colleges and Universities in Anhui Province (no. gxyq2020166); Scientific Research Project of Anhui Polytechnic University (no.Xjky2020115); Natural Science Research Project of Institutions of Higher Education in Anhui Province of China (no. KJ2017A114); National Natural Science Foundation of China (no. 11902001); Middle-aged Top-notch Talent Support Programs of Anhui Polytechnic University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Navaei, J.; Babakmehr, M.; Kalamdani, R.; Liu, Y.N.; Qiu, S.; Farhan, J.; Blackful, J. An intelligent image processing-based approach to optimize vehicle headlamp aiming. In Proceedings of the 18th International Multi-Conference on Systems, Signals & Devices (SSD), Monastir, Tunisia, 22–25 March 2021.
2. Chen, M.Y.; Wu, H.T. Real-time intelligent image processing for the internet of things. *J. Real-Time Image Process.* **2021**, *18*, 997–998. [[CrossRef](#)]
3. Singh, A.; Li, P.; Singh, K.K.; Saravana, V. Real-time intelligent image processing for security applications. *J. Real-Time Image Process.* **2021**, *18*, 1787–1788. [[CrossRef](#)]
4. LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. Handwritten digit recognition with a back-propagation network. *Adv. Neural Inf. Process. Syst.* **1990**, *2*, 396–404.
5. Duan, M.; Wang, G.P.; Niu, C.Y. Sample image recognition method on convolutional neural network. *Comput. Eng. Des.* **2018**, *39*, 224–229.
6. Liu, J.Z. Sample bark image recognition method on convolutional neural network. *J. Northwest For. Univ.* **2019**, *34*, 230–235.
7. Ding, A.; Zhang, Y.; Zhu, L.; Li, H.; Huang, L. Intelligent recognition of rough handling of express parcels based on CNN-GRU with the channel attention mechanism. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–18. [[CrossRef](#)]
8. Li, H.C.; Cai, Y.; Wang, L.X. Semantic segmentation of full convolutional network image based on global feature extraction. *Infrared Technol.* **2019**, *41*, 595–615.
9. Domínguez, A. A history of the convolution operation. *IEEE Pulse* **2015**, *6*, 38–49. [[CrossRef](#)] [[PubMed](#)]
10. Zhan, J. Technical analysis of intelligent image processing of tea. In Proceedings of the 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), Shenyang, China, 22–24 January 2021.
11. Chen, Z.L.; Zou, B.J.; Huang, M.Z.; Shen, H.L.; Xin, G.J. The influence of image brightness characteristics on ROI extraction. *J. Cent. South Univ. Sci. Technol.* **2012**, *43*, 208–214.
12. Maeda, J. *Design in Tech Report 2019*; MIT Press: Cambridge, MA, USA, 2019; pp. 10–21.
13. Xue, Z.R. *AI Changes the Design: Designer's Survival Manual in the Age of Artificial Intelligence*; Tsinghua University Press: Beijing, China, 2019; pp. 62–68.
14. Noguchi, K.; Okada, Y. IntelligentBox for web: A constructive visual development system for interactive web 3D graphics applications. In *Complex, Intelligent, and Software Intensive Systems*; Springer: Berlin, Germany, 2020; pp. 757–767.
15. Bao, W. The application of intelligent algorithms in the animation design of 3D graphics engines. *Int. J. Gaming Comput. Mediat. Simul.* **2021**, *13*, 26–37. [[CrossRef](#)]
16. Li, Y.T.; Xu, D. Deep learning algorithm of woodcut style conversion. *J. Comput. Aided Des. Graph.* **2020**, *32*, 1804–1812.
17. Tian, Y.L.; Chen, S.X.; Zhao, F.J.; Lin, X.Y.; Xiong, H.L. Handwritten layout analysis and multi style background fusion of ancient books. *J. Comput. Aided Des. Graph.* **2020**, *25*, 1115–1220.
18. Chen, L.Q.; Wang, P.; Dong, H.; Shi, F.; Han, J.; Guo, Y.; Childs, P.R.N.; Xiao, J.; Wu, C. An artificial intelligence based data-driven approach for design ideation. *J. Vis. Commun. Image* **2019**, *61*, 10–22. [[CrossRef](#)]
19. Sejnowski, T. *Deep Learning: The Core Driving Force in the Era of Intelligence*; Citic Press: Beijing, China, 2019; pp. 50–57.
20. Sejnowski, T. *The Deep Learning Revolution*; MIT Press: Cambridge, MA, USA, 2018; pp. 50–57.
21. Chen, D.X.; Zhan, Y.Y.; Yang, B. Application and analysis of deep learning technology in the field of educational big data mining. In Proceedings of the International Conference on Big Data and Social Sciences (ICBDSS), Xi'an, China, 14–16 August 2019; pp. 68–76.
22. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
23. Saito, K. *Introduction to Deep Learning: Theory and Implementation Based on Python*; Posts & Telecom Press: Beijing, China, 2018; pp. 201–203.
24. Morota, G.; Cheng, H.; Cook, D.; Tanaka, E. ASAS-NANP SYMPOSIUM: Prospects for interactive and dynamic graphics in the era of data-rich animal science 1. *J. Anim. Sci.* **2021**, *99*, skaa402. [[CrossRef](#)] [[PubMed](#)]

25. Zhou, N.G. Analysis on the design and information communication of motion graphic. *Packag. Eng.* **2017**, *38*, 147–151.
26. Chen, S.H.; Jiu, Z. A method of stereoscopic display for dynamic 3D graphics on android platform. *J. Web Eng.* **2020**, *19*, 849–864. [[CrossRef](#)]
27. Ding, Y.H. Methods for dynamic drawing based on .NET Framework. *J. Eng. Graph.* **2009**, *30*, 71–75.
28. Castillo, E.; Lacruz, B.; Lasala, P.; Lekuona, A. Estimating transition-probabilities in a dynamic graphic model with unobservable variables. *IEEE Trans. Reliab.* **2001**, *50*, 135–144. [[CrossRef](#)]