

## Research Article

# Research on UI Design and Optimization of Digital Media Based on Artificial Intelligence

Meng Xia<sup>1</sup> and Yucui Zhou <sup>2</sup>

<sup>1</sup>Guangdong Polytechnic of Science and Technology, Guangdong Zhuhai 519090, China

<sup>2</sup>Tianjin University of Technology, Tianjin 300384, China

Correspondence should be addressed to Yucui Zhou; [zyc2003@email.tjut.edu.cn](mailto:zyc2003@email.tjut.edu.cn)

Received 11 March 2022; Accepted 27 April 2022; Published 24 May 2022

Academic Editor: Yuan Li

Copyright © 2022 Meng Xia and Yucui Zhou. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

At present, artificial intelligence technology has developed rapidly and has been gradually applied to various fields of society. With the popularization and development of computers, under the influence of artificial intelligence technology, digital media art and artificial intelligence technology merge with each other, which makes modern culture bloom with greater charm. Based on the era of artificial intelligence, this paper discusses the development core of the integration of artificial intelligence technology and digital media art, analyzes the current development status of digital media art and technology, and finally puts forward the innovative development direction and future trend. At present, China's artificial intelligence technology has gradually entered a prosperous stage in the continuous development and has become one of the high and new technologies in the new era, bringing many changes to people's work and life. With the development of society, modern people are pursuing the two-way needs of material and spirit, and art, as a special field, has attracted much attention. By involving artificial intelligence in the art field, it has become a new challenge to be faced at present, which will promote art and technology to present different forms of expression. Of course, the artistic content of digital media art will be richer, and the attention will be higher. Under the background of artificial intelligence era, the integration of artificial intelligence and digital media is the core industry at present, which promotes the development of China's economy in a better direction.

## 1. Introduction

This work explains artificial intelligence through the definition of intelligent agent and its role in production system, reactive agent, real-time conditional planner, neural network, and rich decision system. This book emphasizes the importance of the task environment as a decisive factor in the proper design of agents [1]. Integer programming benefits from many innovations in models and methods, and some promising directions for expounding these innovations in the future can be seen from a framework linking artificial intelligence and operational research perspectives. We looked at four key areas. Each shows useful features with the development on the horizon [2]. Logic is now widely regarded as one of the basic disciplines of computing and has applications in almost every aspect of the discipline, from software engineering and hardware to programming

languages and artificial intelligence. After extensive discussion between the author of the manual and the second reader, the catalogue of these two volumes was finally determined [3]. This paper introduces constrained programming, which is a powerful example of combinatorial search problem, and it absorbs a wide range of technologies from artificial intelligence, computer science, database, programming language, and operational research [4]. This work explains artificial intelligence through the definition of intelligent agent and its role in production system, reactive agent, real-time conditional planner, neural network, and rich decision system. This book emphasizes the importance of the task environment as a decisive factor in the proper design of agents [5]. This article introduces the evolution of this book from a lecture for students with little knowledge of calculus, and readers do not need any prerequisites related to knowledge of programming languages. In university settings,

this book can be used as an introductory course in computer science, information systems, or engineering departments [6]. Pattern recognition techniques can usually improve efficiency by limiting the application of machine methods to appropriate problems. Pattern recognition, together with learning, can be used to generalize based on accumulated experience and further reduce searches [7]. Since 2010, Ant Conference has officially devoted itself to the whole field of swarm intelligence, without any specific research direction. The full-text document was presented orally at the plenary meeting, and an expanded version of the best papers presented at the meeting will be published in a special issue of the journal [8]. In the uncertain subjective and objective world, randomness is the most fundamental. This paper discusses the relationship between randomness and fuzziness, and the result is the automation of representation, processing, and thinking of uncertain information and knowledge [9]. International e-commerce conference aims to bring together researchers interested in current applications. The technology involves not only lower-level technical problems, but the revised best papers are published in CCIS series [10]. This paper discusses the implementation of artificial intelligence (AI) technology on the World Wide Web. These implementations include a pilot agent with natural language processing and rule-based artificial intelligence, which is a new complex [11]. In artificial intelligence, the process of searching for problem solutions can be realized without domain knowledge or with domain knowledge [12]. This paper examines the organizational dynamics when communication becomes a prominent part of the organizational structure. That is, to understand this change in large-scale action networks, it is necessary to distinguish at least two possible logics: the familiar collective action logic related to high-level organizational resources and the formation of collective identity [13]. A system for controlling access to digitized data is provided. A transmitter program capable of communicating with a security rights management server is provided to a non-secure client. Prior to use, an authentication process is performed between the transmitting pad and the server to authenticate the browser, which expires if authentication does not occur within a predetermined time period [14]. A system for allowing the use of media content in an interactive digital media program is described in which object mapping data and programs are downloaded to a user terminal and used in conjunction with the presentation of media content [15]. The new digital media has greatly changed the communication environment, especially for young people. These communication platforms provide new tools for young people to participate in promoting sexual health and reducing risks. Many comparative studies of follow-up and measurement of behavioral results show the effectiveness of new digital media in changing adolescent sexual behavior [16]. The purpose of this study is to understand the cognitive process of designers when sketching in digital and traditional media. The results show that traditional media has advantages over digital media and these results also provide enlightenment for computer aided architectural design [17]. The Internet has changed the way people get news, do business, communicate with each other, socialize,

and interact with government officials. A number of researchers, almost all working in the countries they analyzed, contributed to the project by studying Internet-related laws and practices, testing the accessibility of selected websites and interviewing a wide range of sources [18]. Digital media strategy is an important part of contemporary political activities, which makes citizenship thinner in that people can easily become political speakers without substantive participation [19]. Interactive tables can enhance teamwork and collaborative work in many fields. One application of this new technology is collaborative search of digital content. The results show that these two strategies are equally effective and the possible benefits of team-centered disadvantages exceed expectations [20]. Digital media watermarking discusses the new aspects of digital watermarking in the world. Not only in terms of technology, but also in terms of business and law, this book discusses digital watermarking as it relates to many areas of digital media [21]. An apparatus, program product, and method utilize a set of rules associated with particular individuals to restrict or otherwise control the use of character substitution or similar techniques for similarity data of those individuals in media presentation. Thus, during generation of the media presentation, a personalized version of the media presentation may be generated [22]. This paper presents a method for managing digital objects. You can search and display objects by dragging and dropping tags onto them, and then according to the degree to which the tag search criteria match. In addition, visual cues can indicate whether the displayed objects meet their own search criteria [23]. The invention relates to an armrest-type personal digital media system that allows individuals to use portable, mobile digital devices combined with an armrest system, allows users to enjoy their own digital entertainment options, and also provides users with multiple functions such as seat armrests including memory interface modules [24].

## 2. The Performance Characteristics of Digital Media

*2.1. Interactivity.* Interactivity is an important feature of digital media technology. People can choose and receive information according to their own thinking habits and subjective consciousness and can form mutual communication and feedback between people and machines. When people can choose and control information according to their own subjective wishes and when people interact with machines, they will get more vivid information displayed by rich multimedia and then change the original boring visit into a two-way interactive way of information acquisition, so people will feel more entertaining.

*2.2. Virtuality.* Digital media technology has created a fictional world for people, and “0” and “1” have become the “materials” of this world and constructed the “realistic scene” of this world. Upgrading technology can change ideas and make the world better understood. Virtuality enriches the true meaning of “reality.” Reality is no longer just daily

life, but more into the virtual reality of the world to experience the virtual unreality and reality of the feeling.

**2.3. Spectacularity.** Digital media can restore the impossible or imaginary scenes in the real world and then make people feel the shocking audiovisual experience that they have never experienced before. The success of this phenomenon shows the superb ability of digital media.

**2.4. Fusion.** Artistic creations based on digital technology and information technology break down the barriers between art categories caused by production and dissemination tools, put different types of artistic creation on one platform, and use unified digital technology tools and language so that different artistic categories can be freely integrated, eliminating the estrangement and occlusion between artistic categories, and finally presenting a more colorful artistic experience.

### 3. Artificial Intelligence Industry Upgrading Mechanism

**3.1. Technical Factor Model of Economic Growth.** The theoretical research in this section is based on Romer's R&D model. In the AK model, the production function is expressed as

$$F_i = F\left(k_i, \sum_{i=1}^n K_i, \bar{x}_i\right). \quad (1)$$

$F_i$  Represents the output level of manufacturers,  $k_i$  represents the professional knowledge during production,  $\bar{x}_i$  represents the vector of production factors, and  $\sum_{i=1}^n K_i$  represents the cumulative level of social knowledge.

Although AK model reveals that technological progress is the result of the efforts of economic activity subjects, but it cannot explain the motivation of the main body of economic activities to carry out technological innovation. Combined with the framework of imperfect competition, Romer revises the original model and constructs R&D model, which internalizes technological progress and thinks that technological progress determines the increase of intermediate product types, while the input of intermediate product types directly affects economic growth, so innovation leads to productivity growth.

The production function Y of the product is expressed as

$$Y = H_y^\theta L^\mu \sum_{i=1}^{\infty} x_i^{1-\theta-\mu}. \quad (2)$$

When there is technological progress, the infinite diminishing effect disappears. Given the values of human capital  $H_y$ , and labor force L, the total demand function of durable goods based on this can be obtained as shown as

$$\max \int_0^{\infty} \left[ H_y^\theta L^\mu \sum_{i=1}^{\infty} x_i^{1-\theta-\mu} - p(i)x(i) \right] di. \quad (3)$$

The first-order condition for maximization is

$$p(i) = (1 - \theta - \mu) H_y^\theta L^\mu x_i^{-\theta-\mu}. \quad (4)$$

In the formula, the profit maximization formula is

$$\pi = \max p(x)x - r\varphi x, \quad (5)$$

where  $\bar{x}$  is the maximum value of  $\bar{p}$  on the demand curve.

The producer enters the intermediate product sector and buys a new design with a value  $P_A$  equal to the discounted value of the net income that can be obtained as shown as

$$\int_t^{\infty} e^{-\int_t^\tau r(s)ds} \pi(\tau) d\tau = P_A(t). \quad (6)$$

If  $P_A$  is a constant, Formula (7) can be obtained:

$$\pi(t) - r(t) \int_t^{\infty} e^{-\int_t^\tau r(s)ds} \pi(\tau) d\tau = 0. \quad (7)$$

Formula (8) can be obtained by substituting  $P_A$ :

$$\pi(t) = r(t)P_A. \quad (8)$$

According to Ramsey model, Formula (9) can be deduced:

$$\int_0^{\infty} u(C(t))e^{-\rho t} dt, \quad (9)$$

where  $u(C(t))$  is shown:

$$u(C(t)) = \frac{c^{1-\sigma} - 1}{1 - \sigma}, \sigma \in [1, \infty). \quad (10)$$

The profit that the seller can obtain from continuous investment is  $\pi = (\lambda + \mu)\bar{p}\bar{x}$ , and if the profit that can be obtained is equal to the newly designed price  $P_A$ , Formula (11) can be deduced:

$$P_A = \frac{\pi}{r} = \frac{1}{r}(1 - \theta - \mu)H_y^\theta L^\mu x_i^{-\theta-\mu}. \quad (11)$$

The total income of human capital from the research department is  $P_A\delta A$ , as shown as

$$H_y = \frac{1}{\delta} \frac{\lambda}{(\lambda + \mu)(1 - \theta - \mu)} \gamma. \quad (12)$$

When the values of  $H_y$  and  $\bar{x}$  are fixed, L and A have the same growth rate; If the value of  $\bar{x}$  is fixed, then K and A have

the same growth rate. If the growth rates of A, Y and K are Z, then the relationship between Z and interest rate R is shown:

$$z = \delta H_A = \delta H - \frac{\lambda}{(\lambda + \mu)(1 - \theta - \mu)} \gamma. \quad (13)$$

Therefore, economic growth rate is directly proportional to human capital, inversely proportional to technology, and has nothing to do with population size, while technological progress is the driving force of economic growth. In addition, Romer also stressed that government behavior can support technological innovation, and the positive externalities of technology and knowledge elements will lead to increasing returns to scale, thus promoting economic growth. Under the maximum utility, the government needs to increase investment in scientific research and education to ensure sufficient knowledge output.

**3.2. Model of Intelligent Technology Expanding Product Category.** Taking artificial intelligence technology as an expansion of the categories of raw products and consumer goods, the production function Y of Formula (2) can be rewritten as shown as

$$Y = H_y L^\mu \sum_{j=1}^N x_{ij}^{1-\mu}. \quad (14)$$

$x_{ij}$  is the consumption of type j transition products, and the decomposability of transition products shows that their number is not equal to the current number of items, that is to say, new products are not the products directly copied and used by the original technology, nor do they supplement the original technology, so the discovery of new products will not cause the obsolescence of existing products.

N denotes the number of transition products used directly, and the increase in N denotes the utility brought about by technological development. In order to study this utility, the transition products are expressed as established units, and assuming that the quantities used are consistent, then Formula (14) can be further rewritten as

$$Y = H_y L^\mu N x_i^{1-\mu} = H_y L^\mu (N x_i)^{1-\mu} N^\mu. \quad (15)$$

If  $H_y$ , L, and  $N x_i$  are given, Formula (15) shows that the value of Y increases with the increase of N. This reflects the connotation of technological progress. In this paper, it is indicated that the income increased by artificial intelligence technology by allocating a given number of intermediate products  $N x_i$  to a more number of transitional products N. For a given L, if the increase of  $N x_i$  is realized by increasing  $x_i$ , there will be a diminishing return, and if the increase of  $N x_i$  is realized by using existing  $x_i$  and increasing N, there will be no diminishing return. Therefore, using artificial intelligence technology to increase the number of transition products will not cause a decline in profits.

**3.3. Model of Intelligent Technology to Improve Product Quality.** The improvement of product quality is accompa-

nied by the replacement of old products. Referring to the research of Aghion and Howitt (1992, 1998), it is assumed that the final product is unique, the transitional product category is N, and the production function of the enterprise is expressed as

$$Y = H_y L^\mu \sum_{j=1}^N \bar{x}_{ij}^{1-\mu}, \quad (16)$$

where  $\bar{x}_{ij}$  represents the quality change of the intermediate product of Class J. If the initial quality of the intermediate product is 1 and the quality after m changes is  $Q^m$ , then the input of the J department after quality adjustment is

$$\bar{x}_{ij} = Q^{mj} x_{ij}. \quad (17)$$

Assuming that quality levels within departments can be completely substituted, the total input of departments can be regarded as the weighted sum of the quality of each level, and the net present value of the total profit of J department undergoing quality adjustment at time point  $t_{m_j}$  can be expressed as

$$V(m_j) = \int_{t_{m_j}}^{t_{m_{j+1}}} \pi(m_j) e^{-\int_{t_{m_j}}^w r(v) dv} dw. \quad (18)$$

$t_{m_j}$  indicates the time of the  $m_j$  quality adjustment,  $t_{m_{j+1}}$  indicates the time of the competitor's next adjustment, and the time for the product to maintain its advantage after the  $m_j$  quality adjustment is  $t_{m_{j+1}} - t_{m_j}$ . With the competitor developing new adjustment means, the old product loses its leading position.

The product category expansion model reflects the basic innovation research, and will not replace the old products, while the product quality improvement model reflects the progress of production technology, which may replace and eliminate the old products. Both of them show the contribution of the improvement of technical level in economic growth.

## 4. Experiment and Result Analysis

**4.1. Experiment 1.** In this paper, Python 3.6 is used as the main programming language, and neural network is constructed based on Pytorch 1.6. NVidia GeForce GTX-1080Ti is used as the main graphics card. The specific environment configuration is shown in Table 1.

### 4.1.1. Details of the Experiment

(1) *Data Preparation.* The interface image is preprocessed. In order to adapt to the horizontal screen interface, the interface image is rotated at 90 degrees and 270 degrees in this paper, and the offline rotation is carried out instead of random online rotation, which is equivalent to expanding the number of interface images in the training set to three times the original, that is, nearly 171 K pairs of training data. In this paper, different sizes of images are scaled for different

TABLE 1: Experimental software and hardware environment.

Type of experimental environment	Experimental environment
CPU	Intel(R) Core(TM) i5-9400F@2.20GHz
Memory	16 GB
Graphics card	NVIDIA GeForce GTX-1080Ti
Operating system	Ubuntu 18.04
Neural network framework	Pytorch 1.6
Parallel computing framework	cuda 10.1, cudnn 7.6

backbone networks (the downsampling magnification of images in backbone networks is different, so adaptive adjustment is needed). Based on the consideration of video memory and time, when simple convolution neural network is used as backbone network, the interface images are scaled to  $201 * 301$  (consistent with UI2code, which is convenient for comparative experiments). When VGG16 and ResNet 34 are used as backbone networks, the image is scaled to  $480 * 850$ . Because of the image rotation operation, the width and height of the above scaling size are selected according to the original interface image, and the larger one of the widths of the interface image is scaled to 301 or 850, and the smaller one is scaled to 201 or 480.

(2) *Model Implementation.* The overall training flow of neural network translator based on Transformer proposed in this chapter is as follows: Firstly, screenshots of application screen interface are sent to convolution neural network in batches for image feature extraction. Then, it is stretched and embedded with position coding and then sent to the encoder (the input of each layer needs to be encoded with spatial position). The encoder encodes the image features into a context vector  $C$ . Then, the interface tree text corresponding to the interface screenshot is sent to the decoder for mask operation so that the decoder can observe the specific text in time steps, and then the context vector  $C$  generated by the encoder and the text after mask calculation are calculated together to calculate the probability distribution of output prediction, and finally, the control text search is carried out to finally generate the interface tree.

In this paper, the task of generating interface tree is regarded as the task of image description (generating depth-first traversal sequence of interface tree from high-fidelity interface image). Because text sequence generation is essentially a multiclassification task, cross entropy loss function is adopted in model training, and Adam optimizer is used to update weights. Parameters are set  $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$ , learning rate is set to 1, hot start is added, warmup = 10000 is set, and learning rate is dynamically adjusted. In this paper, the model dimension of Transformer is set to 512, and the number of multihead self-attention of the model is set to heads = 8. Based on the consideration of video memory and time, the batch size of training data is set to 10, and the layers of encoder and decoder are set to

3. In order to prevent over-fitting, dropout = 0.1 is added in the training process, and the training rounds are set to 30.

For the consideration of model performance and translation time, the greedy search strategy is used to generate the interface tree. In order to further improve the effect of generating the interface tree, the model with the best performance under the greedy search strategy is used to generate the interface tree by cluster search.

4.1.2. *Evaluation Indicators.* In this paper, the generation effect of interface tree is mainly evaluated by the following three evaluation indicators.

(1) *Perfect Matching Rate.* Perfect matching rate is a relatively strict evaluation index, that is, when the generated GUI interface tree is exactly the same as the real label in the data set, it is regarded as a successful matching of the interface tree; otherwise, it is a failure. In this paper, the perfect matching rate is used to strictly distinguish the performance of models, That is, the model with higher perfect matching rate of the generated interface tree performs better. However, there is only one case where the generated interface tree matches the real interface tree successfully. However, there may be many cases of matching failure, that is, the effect of one wrong node and 100 wrong nodes is the same, that is to say, the perfect matching rate can only judge how many interface trees have been inferred successfully, but cannot judge how close the generated interface tree is to the real interface tree in case of error.

(2) *BLEU Score.* BLEU is commonly used in machine translation to evaluate the similarity between the generated text and the target text as shown as follows:

$$BLEU = BP * \exp \left( \sum_{n=1}^N w_n \log p_n \right). \quad (19)$$

BP is the length penalty factor, which is set to 1 when the generated text length is greater than the target text length; otherwise, it is  $e^{1-r/l}$ ;  $R$  represents the target text length;  $C$  represents the length of the generated text;  $p_n$  represents the accuracy of n-gram;  $w_n$  denotes the weighting of n-gram precision, and the sum is 1.

The range of BLEU score is [0, 1], which is often expressed as percentage score. The higher the score, the higher the similarity between the generated text and the target text. If it matches completely, the BLEU is 1, which is 100%. In this paper, the highest n-gram is 4.

(3) *Edit Distance.* Besides using BLEU, this paper also uses edit distance to measure the similarity between the generated interface tree and the real interface tree. This article uses two editing distances: One is the Levenshtein editorial distance, mainly used to judge the editing distance of string text, because the interface tree can be regarded as a text string of depth-first traversal sequence, Levenshtein edit distance is used to measure its similarity, and the other is tree edit

distance, which is mainly used to judge the edit distance between two trees. This paper uses it to judge the structural similarity between the generated interface tree and the real interface tree.

The Levenshtein edit distance algorithm measures the similarity of two strings by calculating the minimum number of edits a string needs to make to become another string. The fewer edits required, the smaller the edit distance between two strings, that is, the more similar they are.

The difference of tree editing distance is that preprocessing operation is needed, that is, calculating the key node set of two trees. The so-called key node refers to the highest-level node with the same leftmost leaf node. In the calculation of tree edit distance, the following order traversal sequence of the tree is mainly used to calculate the edit distance similar to Levenshtein, and the dynamic programming algorithm is used to gradually move up from the lower level of the tree, and the forest edit distance may evolve in the calculation process.

*4.2. Experimental Results and Analysis.* In this paper, 5 K screenshots of application interfaces are randomly selected in the data set for testing, and greedy search and cluster search are used to generate interface trees, respectively. First, greedy search is used to judge the advantages and disadvantages of the model, and on this basis, the best results are selected for cluster search to further explore the performance of the model. Except for cluster search, greedy search is used as the generation method of experimental results. Because only UI2code is similar to this paper in the existing research, which can directly generate the interface tree represented by Android real controls, but pix2code and ReDraw cannot do it, so the experimental results of this paper are mainly compared with UI2code.

In order to further explore the performance of the model, this chapter chooses several different convolutional neural networks as the backbone network to extract image features and adjusts the number of layers of the model encoder and decoder. In this paper, three kinds of convolutional neural networks are mainly used: simple network, VGG16, and ResNet34. The loss curves of model training are shown in Figure 1, Figure 2 and Figure 3, and the experimental results of greedy search are shown in Table 2.

As can be seen from Figures 1, 2, and 3, the model of ResNet34 as the backbone network converges faster, and the loss curve is smoother. The ours model in Table 2 represents the Transformer-based neural network translator model proposed in this chapter, and the suffix simple represents different backbone networks adopted by the model. According to the experimental results displayed, the following conclusions can be drawn:

- (1) The Transformer-based neural network translator proposed in this section is obviously affected by the ability of image feature extraction, when a relatively simple convolutional neural network is adopted. Because the extracted image features are not rich enough, the overall performance of the model cannot be brought into full play. When using convolution

neural network with stronger image feature extraction ability, such as ResNet34, the ability of the model can be brought into full play, and the perfect matching rate between the generated interface tree and the real interface tree can be obviously improved. Specifically, the simple convolutional neural network with only six convolution layers and four pooling layers has limited extraction ability. The perfect matching rate of the generated interface tree is relatively low. The perfect matching rate of the generated interface tree is only 67.56%, and the BLEU is 92.59%. The average Levenshtein edit distance (LD) of the randomly selected 5K test images is 3.49, and the average tree edit distance (TD) is 2.61. However, with the deepening of the convolutional neural network and the introduction of the residual network, the effect of the model is significantly improved. When VGG16 is used as the backbone network, the perfect matching rate between the generated interface tree and the real interface tree can reach 68.96%, BLEU can also reach 93.18%, LD is further reduced to 3.16, and TD is as low as 2.34; when ResNet34 is used as the backbone network, the effect of the model can be further improved, the perfect matching rate of the interface tree can reach 70.44%, the BLEU can reach 93.21%, and the LD and TD can be further reduced to 3.09 and 2.31. Affected by the backbone network, the perfect matching rate between the interface tree generated by the model in this chapter and the real interface tree increased from 67.56% of simple to 70.44% of ResNet34, an increase of nearly 3 percentage points. At the same time, BLEU also increased from 92.59% to 93.21%, an increase of nearly 0.6 percentage points, and LD and TD also decreased obviously

- (2) When the simple convolutional neural network is used in the backbone network, the number of layers of the encoder and decoder has a great influence on the model effect. If the model uses the same six-layer structure as the original transformer, then the perfect matching rate of the generated interface tree is low, only 64.88%; when the number of model layers is reduced to, the perfect match rate is 67.56% at 3 layers. The semantic complexity of interface tree generation is lower than that of machine translation task. Other indicators have been significantly improved. According to this discovery, all subsequent experiments will use three-layer encoder and decoder structure
- (3) The Transformer-based neural network translator proposed in this section can generate an interface tree directly from high-fidelity UI images. In 5 K test images, 3522 images (70.44%) can directly generate a completely correct interface tree, and generally speaking, the generated interface tree is similar to the real interface structure (BLEU reaches 93.21%, LD and TD are only 3.09 and 2.31, respectively)

In order to prove the superiority of the neural network translator proposed in this paper more obviously, it is

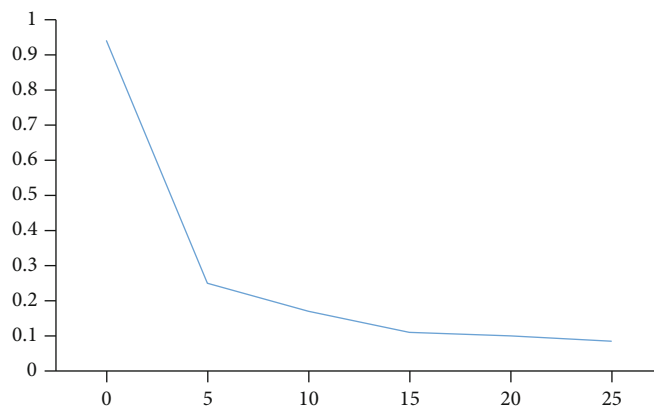


FIGURE 1: Loss change curve of simple model.

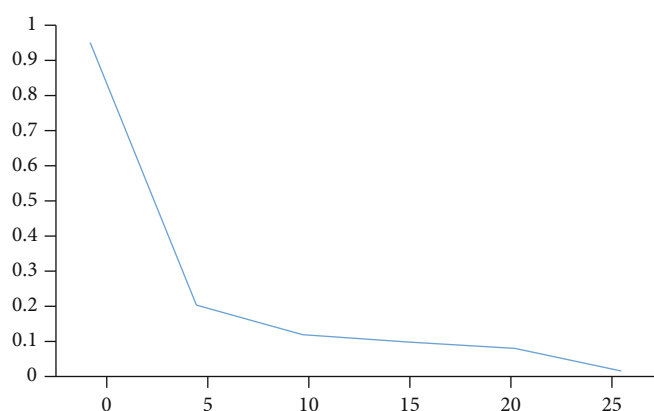


FIGURE 2: Loss variation curve of VGG16 model.

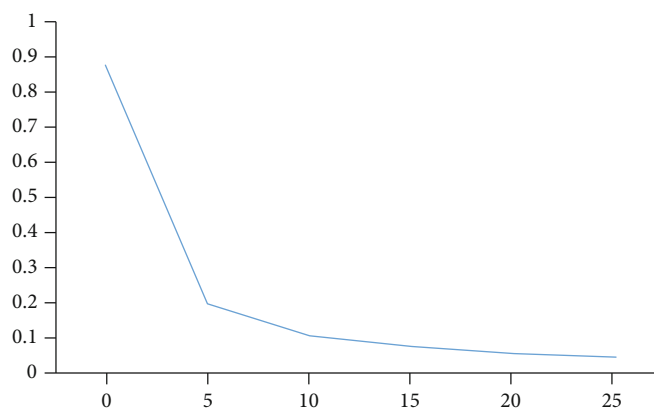


FIGURE 3: Loss change curve of ResNet34 model.

compared with the UI2code model with a better effect at present, and the experimental results of greedy search are shown in Table 3.

As shown in Table 3 (the postJustOnce model suffix indicates that spatial position coding is added only to the first layer of the encoder, and all layers of the encoder are added unless otherwise specified), when the model proposed

in this section uses the same simple convolution neural network as UI2code, compared with it, the perfect matching rate can be improved by 7.1%, 8.67% on BLEU, and 3.31 and 2.96 on LD and TD, respectively. This is mainly due to Transformer's solution to the problem of feature information length dependence and strong feature coding and decoding ability, which proves Transformer's good

TABLE 2: Comparison of greedy search results in backbone networks.

Type of experimental environment	Experimental environment
CPU	Intel(R) Core(TM) i5-9400F@2.20GHz
Memory	16 GB
Graphics card	NVIDIA GeForce GTX-1080Ti
Operating system	Ubuntu 18.04
Neural network framework	Pytorch 1.6
Parallel computing framework	cuda 10.1, cudnn 7.6

TABLE 3: Comparison of greedy search results of each index of the model.

Scheme model	Perfect matching rate	BLEU	LD	TD
UI2code	60.46%	83.92%	6.8	5.57
ours-simple	67.56%	92.59%	3.49	2.61
ours-ResNet34	70.44%	93.21%	3.09	2.31
ours-ResNet34-posJustOnce	69.50%	93.19%	3.12	2.33

performance in interface tree generation. Furthermore, when ResNet34 is used in convolution neural network, 3522 (70.44%) GUI interface trees generated in this model can completely match the real application interface structure, while only 3023 (60.46%) of UI2code completely match the real interface tree, which is 9.98% higher than that in this paper. At the same time, on BLEU, the BLEU score of the neural network model proposed in this paper is 93.21%, and the BLEU score of UI2code is only 83.92%; compared with this article, it is improved by 9.29%. In terms of editing distance, the model proposed in this paper only needs 3.09 string editing operations or 2.34 tree editing operations to get the real interface tree on average, while UI2code needs 6.8 string editing operations or 5.57 tree editing operations on average. Compared with UI2code, the interface tree generated by this research method is closer to the real interface tree. In particular, this chapter explores different spatial position coding methods. When spatial position coding is only used in the first layer of the encoder, the perfect matching rate is 69.50%, while when spatial position coding is added in each layer of the encoder, the perfect matching rate is improved to 70.44%. Therefore, the latter position coding method is adopted in the subsequent experiments of this paper.

In order to further explore the interface tree generation ability of the model, this chapter will use the cluster search strategy to generate the interface tree for the model with the best performance under the greedy search strategy. The specific interface tree generation effect is shown in Table 4 and Figures 4, 5, 6, and 7.

In Figure 4, Figure 5, Figures 6, and 7, broken lines with different colors represent different models. The models proposed in this chapter are represented by blue, UI2code is

represented by orange, the  $x$ -axis represents Beamwidth, that is, the number of results maintained at the same time during cluster search, and the  $y$ -axis displays numerical values, which are divided into perfect match rate, BLEU, Levenshtein editing distance, and tree editing distance. As shown in Figures 4 and 5, with the increase of Beamwidth, the perfect matching rate and BLEU of the model will be improved, especially when Beamwidth = 2. At the same time, the model proposed in this paper is better than UI2code in all values of Beamwidth. As shown in Figures 6 and 7, as Beamwidth becomes larger, the edit distance between the generated interface tree and the real interface tree (whether it is Levenshtein edit distance or tree edit distance) is reduced, which indicates that the proximity between the generated interface tree and the real interface tree is gradually increasing. The edit distance between the generated interface tree and the real interface tree is much lower than that of UI2code. From the specific values in Table 4, it can be seen that the method in this section can achieve the best generation effect when Beamwidth = 5, with a perfect match rate of 71.16%, BLEU of 93.36%, LD of 2.90, and TD of 2.14. Compared with UI2code under the same search conditions, the perfect match rate is improved by 8.3%, BLEU is improved by 5.05%, LD is reduced by 1.82, and TD is reduced by 1.32.

It should be pointed out in particular that in the process of tree editing distance calculation, this paper finds that in the generation of interface tree, there are some generated text sequences that cannot form a tree (“{” and “}” in the text sequence cannot be closed perfectly). For this reason, this paper counts the occurrence times of “{” and “}” in the generated text sequence. When the occurrence times of “{” are less than the occurrence times of “}”, “}” is removed from the end of the text until all “{” and “}” match each other; when “{” occurs more often than “}”, add “}” at the end of the text until all “{” and “}” match each other. According to the above operation, the interface tree can be filled simply and quickly. In 5K test images, the number of incomplete interface trees generated by each model is shown in Figure 8. As can be seen, although the model in this chapter has completely surpassed UI2code in evaluation indicators such as perfect matching rate, it can be seen from the perfect matching rate and TD (tree editing distance) that the interface tree generated by the model in this chapter is more similar to the real interface structure, but in the same way, there are still nearly 100 interface images that cannot generate a complete interface tree; even if different backbone networks are replaced, this problem has not been significantly improved, which proves that the problem mainly lies in the codec stage of Transformer. After in-depth analysis of the model, it is considered that it is mainly caused by the self-attention mechanism of Transformer and the next section will analyze this problem.

This chapter describes the software and hardware environment of the experiment and the training parameters; after that, the experimental results are analyzed and compared with the existing research. It is found that the perfect matching rate and BLEU score of GUI interface generated by the Transformer-based neural network translator are much higher than those of the existing research, and the



TABLE 4: Comparison of cluster search results of each index of the model.

Scheme model	Beam width	Perfect matching rate	BLEU	LD	TD
UI2code	1	60.46%	83.92%	6.8	5.57
UI2code	2	61.76%	88.19%	5.07	3.81
UI2code	3	62.12%	88.25%	4.88	3.61
UI2code	4	62.52%	88.27%	4.74	3.5
UI2code	5	62.86%	88.31%	4.72	3.46
ours-ResNet34	1	70.44%	93.21%	3.09	2.31
ours-ResNet34	2	71.12%	93.34%	2.91	2.16
ours-ResNet34	3	71.14%	93.36%	2.9	2.16
ours-ResNet34	4	71.16%	93.36%	2.9	2.16
ours-ResNet34	5	71.16%	93.36%	2.9	2.14

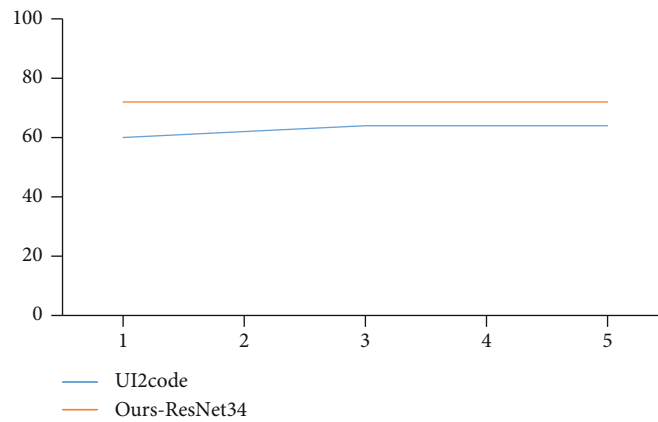


FIGURE 4: Comparison of cluster search results of each index of perfect matching rate.

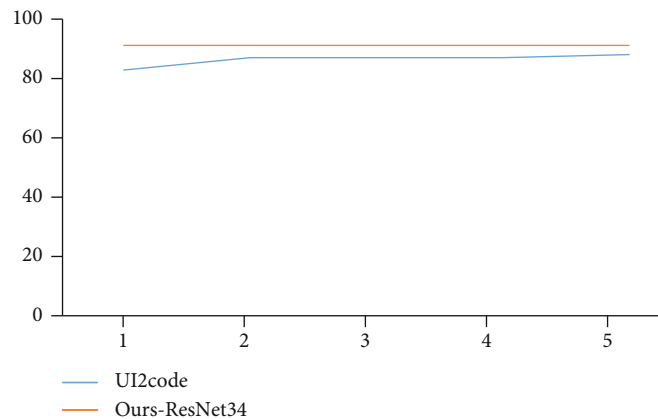


FIGURE 5: Comparison of cluster search effect of each index of BLEU score.

Levinstein editing distance and tree editing distance are obviously lower than those of the existing research, which proves the effectiveness of the method in this chapter.

4.3. *Experiment 2.* The experimental environment in this chapter is consistent with that in the previous section. Python 3.6 is used as the main programming language, and neural network is constructed based on Pytorch1.6. The graphics

card mainly adopts NVIDIA GeForce GTX-1080Ti, and the specific environment configuration is shown in Table 1.

#### 4.3.1. Details of the Experiment

(1) *Data Preparation.* The data used in this chapter is consistent with the previous section. There are 57 K pairs of data (interface image and interface tree label) in the training set,

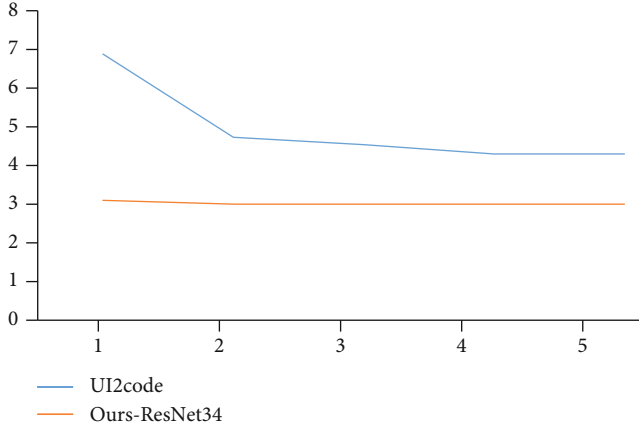


FIGURE 6: Comparison of cluster search results of various indicators of Levenshtein editing distance.

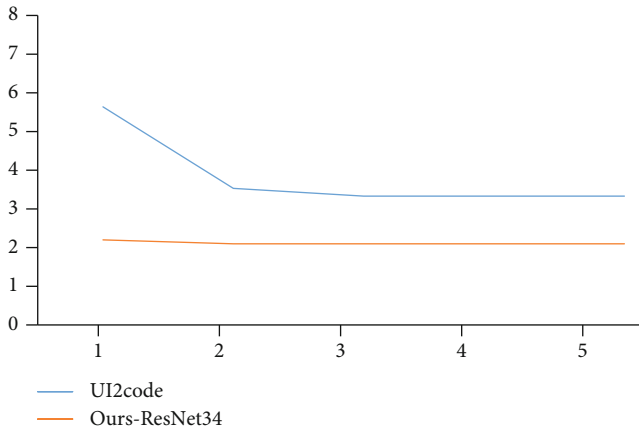


FIGURE 7: Comparison of cluster search effect of each index of tree editing distance.

3 K pairs in the verification set, and 5 K pairs in the test set. The training set is expanded to 171 k pairs of data after two different rotation enhancements. There are also two scaling scales for image data in this chapter:  $201 * 301$  for simple network and  $480 * 850$  for VGG16 and ResNet is 34.

(2) *Model Implementation.* The overall training process is as follows: Firstly, screenshots of application screen interface are sent to convolution neural network in batches for image feature extraction, Then, it is stretched and encoded with spatial position information and sent to the encoder (each layer of the encoder needs to be encoded with spatial position information). In each layer of the encoder (except the highest layer), in addition to the original self-attention calculation module, there is also a priori memory module, using it to extract prior information. At the same time, the prior information is propagated to all subsequent encoder layers to participate in the attention calculation of this layer. The encoder encodes the image features into a context vector C. Then, the interface tree text corresponding to the interface screenshot is sent to the decoder for masking operation

so that the decoder can observe the specific text in time steps, and the context vector C and the text after masking calculation are calculated together to output the probability distribution predicted by the decoder, and finally, the control text search is carried out to finally generate the interface tree.

4.3.2. *Model Training.* Because the model is mainly changed in the encoder stage, however, the backbone network and decoder of image feature extraction have not been changed, so when the model is trained concretely, it is still as shown in 4.1. 1b, using cross entropy loss function, using Adam optimizer to update the weight, keeping the parameters set  $\beta_1 = 0.9$  and  $\beta_2 = 0.98$ , setting the learning rate to 1, and still adding hot start, setting warmup = 10000, and dynamically adjusting the learning rate. In this section, the model dimension of Transformer is set to 512, and the number of multi-head self-attention of the model is set to heads = 8. Based on the consideration of video memory and time, the batch size of training data is set to 10, and the respective layers of encoder and decoder are layers = 3. In order to prevent over-fitting, dropout [71] = 0.1 is added in the training process, and the training rounds are set to 30.

4.3.3. *Experimental Results and Analysis.* In this section, the perfect matching rate, BLEU and editing distance are still used to evaluate the generation effect of interface tree. The 5 K screenshots of application interfaces used in the test are also consistent with the previous section. At the same time, greedy search and cluster search are used to generate interface trees, respectively. First, greedy search is used to judge the advantages and disadvantages of the model. On this basis, the best results are selected for cluster search to further explore the performance of the model. Except for cluster search, greedy search is used as the generation method of experimental results. The experimental results in this section are mainly compared with the improved model and UI2code proposed in this paper.

(1) *Overall Performance Analysis.* In order to fully explore the performance of prior memory self-attention proposed in this chapter, several different convolution neural networks are selected as backbone networks for image feature extraction. In this paper, three convolution neural networks are mainly used: simple convolution neural network, VGG16, and ResNet34. The loss curves of improved model training are shown in Figure 9, Figure 10, and Figure 11, and the experimental results of greedy search are shown in Table 5.

As can be seen from Figure 9, Figure 10, and Figure 11, when ResNet34 is used as the backbone network of the improved model, the convergence speed of the model is faster than that of the other two backbone networks, and the loss curve is smoother. The ours-PM model in Table 5 represents the improved Transformer neural network translator based on prior memory and self-attention proposed in this chapter, and the suffix simple represents different selected backbone networks. From the experimental results shown, the following conclusions can be drawn:

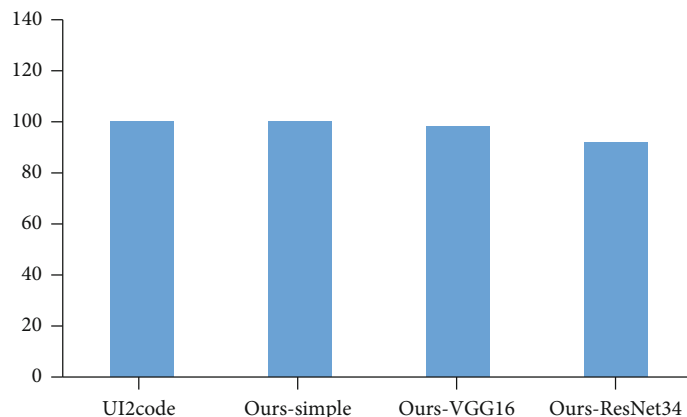


FIGURE 8: Number of incomplete interface trees generated by each model.

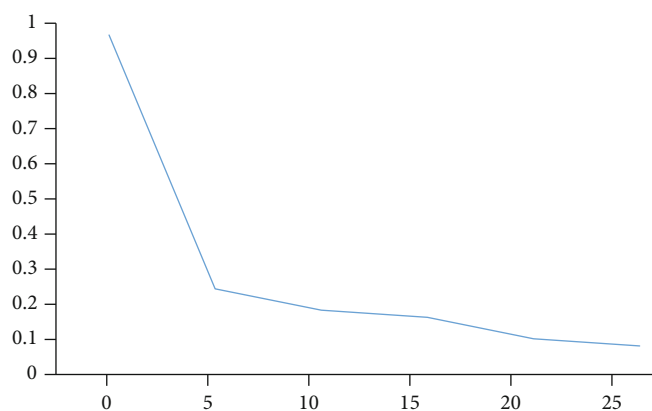


FIGURE 9: Loss change graph of improved simple.

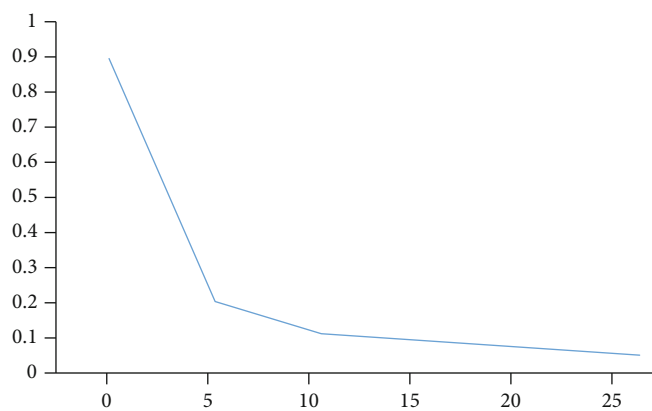


FIGURE 10: Loss change curve of improved VGG16.

(1) The improved Transformer neural network translator based on prior memory self-attention proposed in this chapter can show better effect when the image feature extraction ability is strong. When using simple convolution neural network for image feature extraction, it will seriously affect the model to extract prior information, resulting in the generation effect

of interface tree is not significantly improved. When using convolution neural network with stronger image feature extraction ability, the ability of the improved model can be better reflected. Specifically, when using the simple convolution neural network constructed in this paper, the perfect matching rate of the generated interface tree can only reach

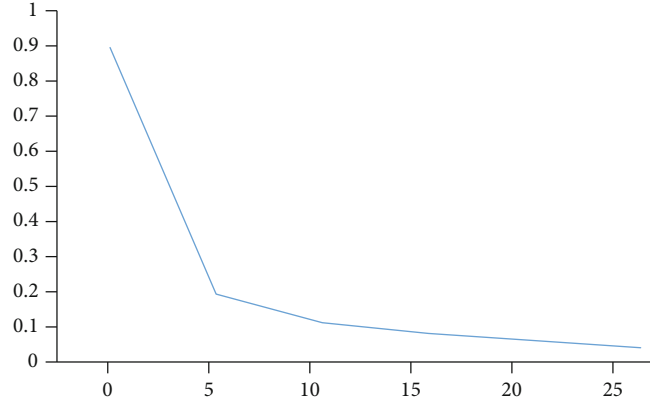


FIGURE 11: Loss change graph of improved ResNet 34.

TABLE 5: Comparison of greedy search results of backbone network of improved model.

Scheme model	m	Perfect matching rate	BLEU	LD	TD
ours-PM-simple	10	66.33%	92.13%	3.62	2.72
ours-PM-simple	20	67.76%	92.67%	3.47	2.56
ours-PM-VGG16	20	70.28%	93.42%	3.01	2.32
ours-PM-ResNet34	20	71.32%	93.66%	2.91	2.11

67.76%, which can only increase by 0.2% compared with that before improvement; BLEU can only reach 92.67%, which is only 0.08% higher than that before improvement. LD (Levinstein editing distance) and TD (tree editing distance) are 3.47 and 2.56, respectively, which are 0.02 and 0.05 lower than that before improvement. When the backbone network is replaced by VGG16 with stronger feature extraction ability, the perfect matching rate between the interface tree generated by the model and the real interface tree can reach 70.28%, which is 2.52% higher than simple; BLEU reaches 93.42%, which is 0.75% higher; and LD and TD are 3.01 and 2.32, respectively, which are 0.46 and 0.24 lower than simple, respectively. When ResNet34 is used as the backbone network of image feature extraction, the performance of the model can be further improved; the perfect matching rate can reach 71.32%, which is 1.04% higher than VGG16; the BLEU reaches 93.66%, which is 0.24% higher; and the LD and TD are 2.91 and 2.11, respectively, which are 0.1 and 0.21 lower than VGG16

- (2) The setting of  $m$  can greatly affect the interface tree generation effect of the improved model, when simple convolution neural network is also used as the backbone network. At the same time, set  $m$  to 10. The perfect matching rate can only reach 66.33%. When  $M$  is expanded to 20, the perfect matching rate can reach 67.76%, and other indicators are improved to varying degrees. This may be due to different sizes of  $M$ , which can obtain different prior

knowledge. When  $M = 10$ , the prior knowledge is too little to model well, and when  $M = 20$ , the modeling ability of the model can be improved. Therefore, the follow-up experiments in this paper adopt  $M = 20$

- (3) The improved Transformer neural network translator based on prior memory self-attention proposed in this chapter can more accurately generate interface trees directly from high-fidelity UI images. In 5K test images, 3566 (71.32%) interface images can directly generate a completely matched interface tree. Generally speaking, compared with the improved network, the generated interface tree can further improve the ability of generating interface tree, and the generated interface tree is closer to the real interface structure (BLEU reaches 93.66%, and LD and TD further decrease to 2.91 and 2.11, respectively)

In order to directly prove the advancement of the improved Transformer neural network translator based on prior memory and self-attention proposed in this chapter, we compare it with the improved Transformer neural network translator and compare it with the best UI2code model and the existing improved Transformer model in the existing research. The experimental results of greedy search are shown in Table 6.

As shown in Table 6, in all 5K test UI images, when the improved Transformer neural network translator based on prior memory self-attention proposed in this chapter adopts the same simple convolution neural network as UI2code, compared with it, the perfect matching rate can be improved

TABLE 6: Comparison of greedy search results of each index of the model.

Scheme model	Perfect matching rate	BLEU	LD	TD
UI2code	60.46%	83.92%	6.8	5.57
ours-simple	67.56%	92.59%	3.49	2.61
ours-PM-simple	67.76%	92.67%	3.47	2.56
ours-ResNet34	70.44%	93.21%	3.09	2.31
ours-PM-ResNet34	71.32%	93.66%	2.91	2.11
M <sup>2</sup> -ResNet34	70.46%	93.63%	2.93	2.16

by 7.3%. It can be improved by 8.75% on BLEU and reduced by 3.33 and 3.01 on LD and TD, respectively. On the one hand, it benefits from Transformer’s powerful feature information long dependence modeling ability; on the other hand, the prior knowledge of prior memory self-attention modeling proposed in this chapter also contributes to some extent (compared with the traditional Transformer before improvement, the above four indicators have been improved to some extent). In addition, when ResNet34 is used in convolution neural network, the ability of prior memory and self-attention can be brought into full play, and 3566 (71.32%) generated interface trees can completely match the real interface trees, which is 10.86% higher than UI2code and 0.88% higher than before improvement. Compared with UI2code, BLEU has increased by 9.74% and 0.45% compared with before improvement. Compared with UI2code, LD is reduced by 3.89 and 0.18 compared with that before improvement. Compared with UI2code, TD is 3.46 lower and 0.2 lower than before improvement. Compared with UI2code and the improved model, the improved Transformer neural network translator based on prior memory and self-attention proposed in this chapter can learn the transformation knowledge from high-fidelity UI image to interface tree more fully, and the generated interface tree is closer to the real interface tree. In particular, this paper compares the improved model with the Transformer improved model M2Transformer, which performs best on MSCOCO data set at present. M2Transformer model uses self-learning  $m_k$  and  $m_v$  matrices to model prior knowledge. Different from this model, the learning of prior knowledge is calculated according to self-attention mechanism, and the prior knowledge of the lower layer of encoder is transmitted to all subsequent higher layers. Each layer of M2Transformer decoder does not take the output of the last layer of encoder as a context vector matrix, but takes the output of all layers of encoder as a context vector input, establishes mesh connection with each layer of decoder, and calculates their respective weight parameters. When ResNet34 is used as the backbone network, the perfect matching rate of the interface tree generated by M2Transformer can reach 70.46%. Compared with the improved network proposed in this paper, it has certain advantages, but the perfect matching rate of the improved model based on prior memory self-attention can reach 71.32%, and other indicators have been improved to varying degrees, which further proves the superiority of the improved model proposed in this paper.

At the same time, in order to further explore the interface tree generation ability of the model, the model with the best performance under the greedy search strategy will use the cluster search strategy to generate the interface tree. The specific generation effect is shown in Table 7 and Figures 12, 13, 14, and 15.

In Figures 12, 13, 14, and 15, orange represents UI2code model, blue represents Transformer-based neural network translator proposed in this paper, and red represents improved Transformer neural network translator based on prior memory self-attention proposed in this paper. The  $x$ -axis represents Beamwidth, and the  $y$ -axis displays values, which are divided into perfect match rate, BLEU, Levinstein edit distance, and tree edit distance. As shown in Figures 12 and 13, with the increase of Beamwidth, the perfect matching rate and BLEU of the improved model have a certain improvement, which is more obvious when Beamwidth = 2. As shown in Figures 14 and 15, with the increasing Beamwidth, the editing distance of the improved model has decreased to a certain extent, which indicates that the generated interface tree is more and more close to the real interface tree. Generally speaking, when Beamwidth takes different values, the improved model proposed in this chapter is better than the model before improvement and is far better than UI2code.

At the same time, it can be seen from Table 7 that the improved model proposed in this chapter can achieve the best interface tree generation effect when Beamwidth = 5, with perfect matching rate reaching 72.22%, BLEU 93.80%, LD 2.75, and TD 2.0. Compared with the improved model, under the same search conditions, the perfect matching rate increased by 1.06%, BLEU increased by 0.44%, LD decreased by 0.15, and TD decreased by 0.14. Compared with UI2code, under the same search conditions, the perfect match rate increases by 9.36%, BLEU increases by 5.49%, LD decreases by 1.97, and TD decreases by 1.46.

At the same time, it should be noted that in view of the problem of generating incomplete interface trees in the model before improvement, the prior memory self-attention proposed in this chapter effectively alleviates this situation. In 5k test images, the number of incomplete interface trees generated by the model is shown in Figure 16, which is reduced by nearly 50% compared with UI2code and the model before improvement.

According to the above experimental results, the interface tree generation effect of improved Transformer neural network translator based on prior memory self-attention

TABLE 7: Comparison of cluster search results of each index of the model.

Scheme model	Beam width	Perfect matching rate	BLEU	LD	TD
UI2code	1	60.46%	83.92%	6.8	5.57
UI2code	2	61.76%	88.19%	5.07	3.81
UI2code	3	62.12%	88.25%	4.88	3.61
UI2code	4	62.52%	88.27%	4.74	3.5
UI2code	5	62.86%	88.31%	4.72	3.46
ours-ResNet34	1	70.44%	93.21%	3.09	2.31
ours-ResNet34	2	71.12%	93.34%	2.91	2.16
ours-ResNet34	3	71.14%	93.36%	2.9	2.16
ours-ResNet34	4	71.16%	93.36%	2.9	2.16
ours-ResNet34	5	71.16%	93.36%	2.9	2.14
ours-PM-ResNet34	1	71.32%	93.66%	2.91	2.11
ours-PM-ResNet34	2	71.90%	93.70%	2.79	2.01
ours-PM-ResNet34	3	72.10%	93.80%	2.76	2
ours-PM-ResNet34	4	72.22%	93.80%	2.76	2
ours-PM-ResNet34	5	72.22%	93.80%	2.75	2

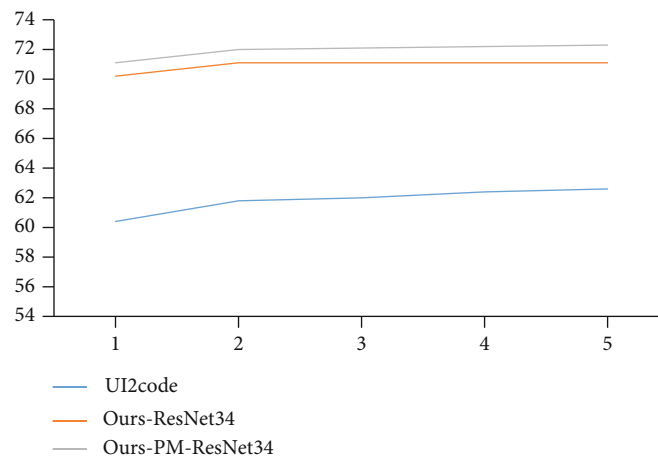


FIGURE 12: Comparison of cluster search results of each index in perfect matching rate model.

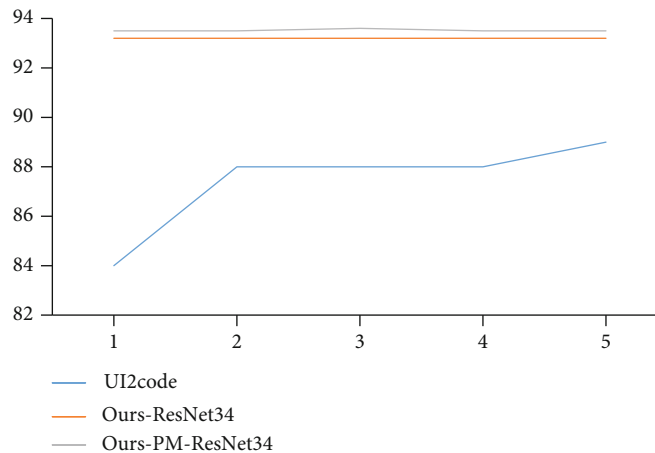


FIGURE 13: Comparison of cluster search results of each index in BLEU score model.

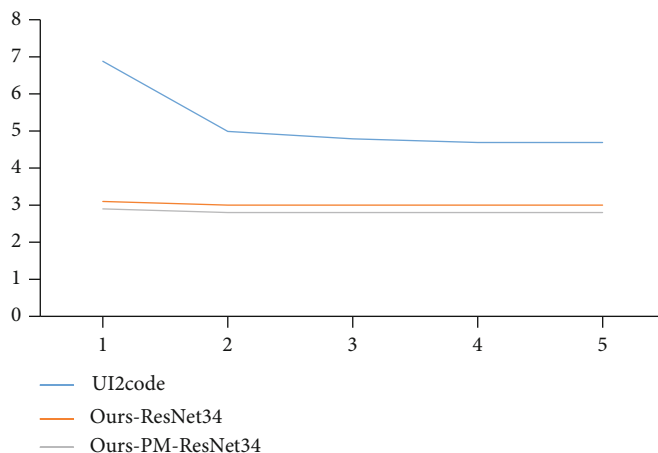


FIGURE 14: Comparison of cluster search results of each index in Levenshtein editing distance model.

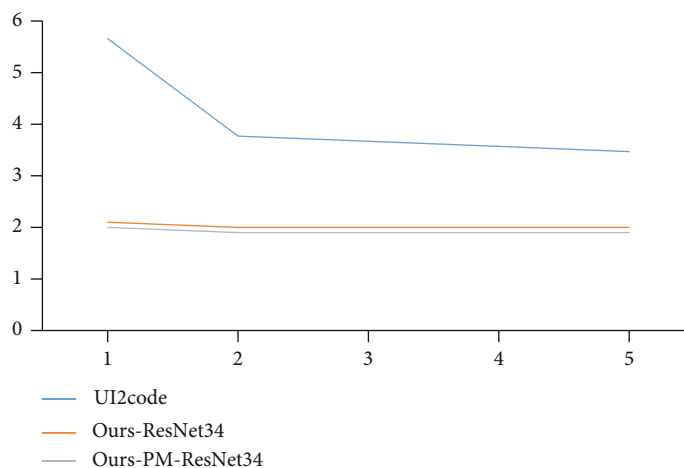


FIGURE 15: Comparison of cluster search results of each index in tree editing distance model.

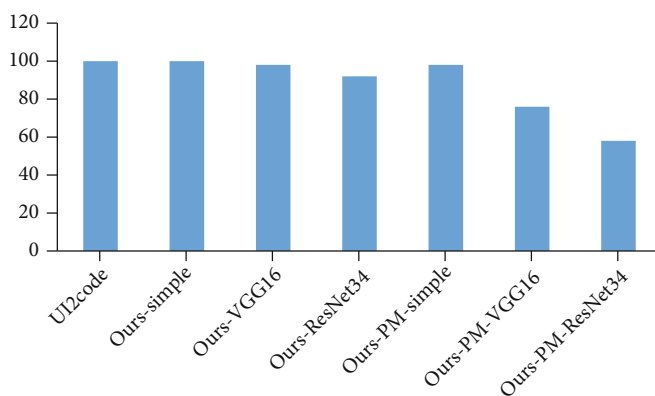


FIGURE 16: Number of incomplete interface trees generated by each model.

has better performance in perfect matching rate, BLEU, LD, TD and complete interface tree generation. The generated interface tree is more consistent with the real interface hierarchy, which proves that the prior memory

self-attention proposed in this chapter can enhance the understanding ability of the model to the interface image features and can effectively improve the generation effect of the interface tree.

## 5. Conclusion

In this paper, convolutional neural network and Transformer encoder and decoder are used to generate interface tree from high-fidelity UI design image. The results show that compared with the existing research, the model proposed in this paper has significantly improved the perfect matching rate and BLEU and the average Levenshtein editing distance and tree editing distance between the interface tree generated in this paper and the real interface tree are smaller. This shows that the interface tree generated by this method is more similar to the real application interface structure. But further analysis of the generated interface tree shows that at present, there are incomplete interface trees in the interface trees generated by the model (nearly 100 incomplete interface trees out of 5000), which shows that the learning ability of the model for the hierarchical structure relationship of interface trees needs to be improved. Further analysis of the proposed model shows that this is mainly caused by the characteristics of Transformer self-attention. To solve this problem, on the basis of Transformer's original self-attention mechanism, this paper puts forward the self-attention mechanism of transcendental memory. A priori knowledge is modeled by adding a memory module of continuous storage to each layer of Transformer's encoder. At the same time, the prior information of this layer is obtained by using it for attention calculation. It is transmitted to all the subsequent high level and participates in the attention calculation of the high level, which avoids that all the information in each layer of the encoder comes from a single input information source, which can enhance the understanding ability of the model to the image features and make the model better learn the hierarchical structure relationship of the interface.

## Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declared that they have no conflicts of interest regarding this work.

## Acknowledgments

This work was supported by the Project of Key Scientific Research Platform of Colleges and Universities of Guangdong Province of China (Grant No. 2020CJPT006); the Project of Higher Vocational Education Computer Specialty Teaching Steering Committee of Guangdong Province of China (Grant No. JSJJZW); the special project in key fields of "artificial intelligence" in Colleges and Universities of Guangdong Provincial Department of Education—research on speaker role intelligent analysis technology of multi-person conversation speech in complex noise scene (2019KZDZX1045 and 2021015); and the Teaching Reform Research Project (Grant No. JG201954).

## References

- [1] N. J. Nilsson, "Artificial intelligence: a modern approach," *Applied Mechanics & Materials*, vol. 263, no. 5, pp. 2829–2833, 2003.
- [2] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.
- [3] A. Robinson, J. K. Eaton, and L. Eaton, "Handbook of logic in artificial intelligence and logic programming," *Oxford University Press*, vol. 2, no. 4, pp. 391–429, 1993.
- [4] F. Rossi, P. V. Beek, and T. Walsh, "Handbook of constraint programming : foundations of artificial intelligence," *Elsevier Science Inc*, vol. 26, no. 19, pp. 643–691, 2006.
- [5] S. Russell and P. Norvig, "Artificial intelligence: a modern approach, 3rd edition," *Applied Mechanics & Materials*, vol. 263, no. 5, pp. 2829–2833, 1995.
- [6] M. Negnevitsky, "Artificial intelligence: a guide to intelligent systems," *Information & Computing Sciences*, vol. 48, no. 48, pp. 284–300, 2005.
- [7] M. Minsky, "Steps toward artificial intelligence," *Proceedings of the IRE*, vol. 49, no. 1, pp. 8–30, 1961.
- [8] F. Wang, "Proceedings of the national conference on artificial intelligence," *Springer Verlag*, vol. 36, no. 5, pp. 823–834, 2011.
- [9] D. Li, "Artificial intelligence with uncertainty," *International Conference on Computer & Information Technology*, vol. 15, 2008.
- [10] M. Dorigo, M. Birattari, and G. D. Caro, "Proceedings of the national conference on artificial intelligence," *World Scientific and Engineering Academy and Society (WSEAS)*, vol. 4, no. 3, 2007.
- [11] D. M. Gabbay, C. J. Hogger, and J. A. Robinson, "Handbook of logic in artificial intelligence and logic programming. Vol. 5: logic programming," *Oxford Univ Pr*, vol. 31, no. 27, pp. 863–904, 1994.
- [12] N. J. Nilsson, "Artificial intelligence: a new synthesis," *Morgan Kaufmann Publishers Inc*, vol. 9, no. 4, pp. 274–315, 1998.
- [13] M. R. Genesereth and N. J. Nilsson, "Logical foundation of artificial intelligence," *Brain broad research in artificial intelligence & neuroscience*, vol. 11, no. 5, pp. 372–435, 1987.
- [14] W. L. Bennett and A. Segerberg, "The logic of connective action: digital media and the personalization of contentious politics," *Perspectives on Politics*, vol. 31, no. 34, pp. 3728–3848, 2012.
- [15] M. W. Rabne, J. A. Barker, and T. Alrashid, "Rights management system for digital media," *US*, vol. 8, no. 13, pp. 384–396, 1999.
- [16] D. A. Wistendahl and L. K. Chong, "System for mapping hot spots in media content for interactive digital media program," *US*, vol. 8, no. 1, pp. 542–620, 1998.
- [17] K. Guse, D. Levine, S. Martins et al., "Interventions using new digital media to improve adolescent sexual health: a systematic review," *Journal of Adolescent Health Official Publication of the Society for Adolescent Medicine*, vol. 51, no. 6, pp. 535–543, 2012.
- [18] Z. Bilda and H. Demirkan, "An insight on designers' sketching activities in traditional versus digital media," *Design Studies*, vol. 24, no. 1, pp. 27–50, 2003.
- [19] P. Howard, M. Hussain, P. N. Howard, and M. M. Hussain, "Democracy fourth wave digital media and the Arab Spring," *Systematic Botany*, vol. 31, no. 4, pp. 805–821, 2013.



- [20] B. Herr-Stephenson, P. G. Lange, C. J. Pascoe, D. John, and T. Catherine, "MacArthur Foundation series on digital media and learning," *MIT Press*, vol. 7, no. 13, pp. 638–692, 2007.
- [21] F. House, "Freedom on the net: a global assessment of internet and digital media," *Freedom House*, vol. 12, no. 9, pp. 261–319, 2013.
- [22] C. Clar, M. Dyakova, K. Curtis et al., "Just telling and selling: current limitations in the use of digital media in public health: a scoping review," *Public Health*, vol. 128, no. 12, pp. 1066–1075, 2014.
- [23] P. N. Howard, "Deep democracy, thin citizenship: the impact of digital media in political campaign strategy," *The Annals of the American Academy of Political and Social Science*, vol. 597, no. 1, pp. 153–170, 2005.
- [24] M. R. Morris, A. Paepcke, and T. Winograd, "NNNNSearch: comparing techniques for co-present collaborative search of digital media," *IEEE international workshop on horizontal interactive human-computer systems*, vol. 5, no. 2, pp. 97–128, 2006.