ADDIE Instructional Design Model combined with AI Technology in the Design of an Auxiliary Correction System for IPOTA

Yung-Hui Chen[1*], Chun-Hsiung Tseng[2], Lin Hui[3], Tony C. T. Kuo[4], Jian-Yu Chen[1], Jeffrey Kurniawan Chandra[1]

[1] Department of Computer Information and Network Engineering, Lunghwa University of Science and Technology, Taoyuan City, Taiwan, ROC.
[2] Department of Communications Engineering, Yuan Ze University, Taoyuan City, Taiwan, ROC.
[3] Department of Computer Science and Information Engineering, Tamkang University, New Taipei City, Taiwan, ROC.
[4] Department of Management and Information, National Open University, New Taipei City, Taiwan, ROC.
[1]cyh@mail.lhu.edu.tw, [2]lendle_tseng@seed.net.tw, [3]amar0627@gmail.com, [4]tony@mail.nou.edu.tw, [1]G1102181009@gm.lhu.edu.tw, [1]jeffreykurniawan14@gmail.com,

Abstract. The experimental analysis of "IP-oriented template assignments (IPOTA)" uses each student's unique seat number as part of the experimental IP to distinguish each student's network experiment results. At the same time, it can prevent students from copying other students' assignment. However, the uniqueness of each assignment made the grading process excessively time-consuming. Therefore, in order to effectively reduce the correction time and at the same time improve the learning performances, this paper proposes a system that adopts the theory of the ADDIE instructional design model, combining the traditional course teaching mode and the "Auxiliary Correction System (ACS)" with the application of the neural network to assist the instructor in the teaching process and shorten the time as well as cost for improving the teaching quality. Exam results have always been the best proof of students' understanding of the course. However, the problem is that assessing students' learning progress usually occurs only after the end of the semester. In order to be able to know the students' learning status in advance before the end of the semester, this paper applies the supervised learning method, predicts the learning performance through the regression model, and predicts the students' test scores based on the students' assignment scores so as to predict the students' next test results. If the result is not good, it can be known immediately, and it can strengthen students' understanding of courses and exams in the next teaching. Therefore, the results of this experiment performed on a personal computer show that when the ACS performs the correction assignments, the average time required for correction of each assignment can be shortened to 20.35 seconds, the image recognition can achieve 100% accuracy in the training set, and the accuracy of the prediction model in the training set and test set can reach 74.44% and 64.29%, respectively.

Keywords: ADDIE Instructional Design Model, Artificial Intelligence (AI), Auxiliary Correction System (ACS), IP-oriented Template-based Assignment (IPOTA), Prediction Model.

1 Introduction

In this era of information explosion, traditional teaching methods are gradually unable to meet students' learning needs. Therefore, many educational scholars have tried to research teaching methods that can improve teaching quality and increase students' learning interest and efficiency. Among these research methods, how to design assignments that are easy for students to operate, complete, and actually learn from, and also prevent students from copying others' work, has become a concern. Therefore, the application of artificial intelligence-related technologies emerges. Based on the basic framework of neural networks, various methods have been developed to address different problem characteristics. For example, the Convolutional Neural Networks (CNNs) are particularly effective in solving image recognition-related problems (Simonyan & Zisserman, 2014)[1], while the Recurrent Neural Networks (RNNs) are useful for solving problems related to time series (Mikolov, et al., 2010)[2], and so on.

The CNN is one of the popular deep learning architectures used today, which is mainly used for image recognition (Simonyan & Zisserman, 2014)[1] and is a deep learning approach. Its main structure is a deep feed-forward neural network with properties such as local connections, weight sharing, and translation invariance. Typically, CNNs are formed by stacking convolutional layers, pooling layers, and fully connected layers. By using backpropagation algorithm for training, CNNs are able to extract features from images. The architecture introduced in this article is the LeNet-5 convolutional neural network model, which is mainly divided into eight parts, namely the Input Layer (INPUT), Convolutional Layer (Convolutions, C1), Subsampling Layer (Subsampling, S2), Convolutional Layer (C3), Subsampling Layer (Subsampling, S4), Convolutional Layer (C5), Fully Connected Layer (F6), and Output Layer (OUTPUT) (LeCun, et al., 1998)[3]. The CNNs have a wide range of practical applications. For example, they can be used for "Sentence Classification" (Kim, 2014)[4] and "Natural Language Matching" (Hu, Lu, Li & Chen, 2014)[5] to overcome previously difficult problems with sentence classification and matching. In addition to image recognition, these are the most well-known types of applications.

Projection involves using an imaginary transparent plane, called the projection plane, and placing it between the object and the observer. Using a defined projection algorithm, the contours of the object are projected onto the imaginary plane using point projection. The points on the projection plane are then connected with lines to create the projected image of the object. This is called the projection of the object onto the projection plane. There are two main types of projection based on the method of projection: perspective projection and parallel projection. Perspective projection involves projecting the object onto the projection plane from a single projection point. It is more in line with visual habits as the object appears larger when closer to the observer and smaller when farther away. When the object is too far, it disappears. Parallel projection, on the other hand, projects the object onto the projection plane with parallel light rays.

This method assumes that the observer is observing the object from an infinite distance, and the projected size of the object is not affected by the distance between the object and the projection plane, making it easier to measure actual distances. Therefore, in this study, image projection will be used to complete the required extraction tasks (Zoizou, Zarghili & Chaker, 2020)[6].

The ADDIE model is a commonly used generic process by instructional designers and training developers. The five phases of analysis, design, development, execution, and evaluation are relatively flexible extensions, making it easy to build effective courses and performance validation tools. Most of the current instructional design models are derivatives of the ADDIE model. The recognized improvement in these models is the use of rapid prototyping and the idea of receiving continuous or formative feedback when creating instructional materials (Mannaz, 1998)[7]. Therefore, the operational steps and content of the ADDIE model are the following:

(1) Analyze: The content includes analysis of learners, course content, training tools, and training environment, among others.
(2) Design: The content includes drafting course outlines, planning course systems, and writing training objectives, among others.
(3) Develop: The content includes designing course presentation formats, teaching activities, interface design, and feedback design, among others.
(4) Implement: The content includes programming, script writing, and graphic design, among others.
(5) Evaluate: The content includes evaluating course content, interface, and effectiveness, among others.

Thus, the study adopts the process of ADDIE Instructional Design Model to analyze the design, implementation, and evaluation of Auxiliary Grading and Scoring" and to improve template-based assignments and explore the evaluation of instructional effectiveness. The main purpose of the research is to reduce the cost of human operation by designing an ACS for the IPOTA to replace the manual grading of text in the Wireshark image screenshots required for course assignments. This is achieved through the use of deep learning CNN, RNN, and image projection methods to extract text from the Wireshark image screenshots required for course assignments.

Therefore, session 2 introduces the design and production of the ACS for IPOTA. And then introduces the design and production of the "Deep Learning" image recognition architecture for this thesis in session 3. Session 4 then describes the application of the ADDIE instructional design model to the ACS, and analyzes the preliminary experimental results of the ACS in session 5. Finally, the conclusion and recommendations of the research are presented in the last session.

2 Design and Development of an ACS for IPOTA

The design and analysis of the architecture and experimental process of the ACS for IPOTA are shown in the flowchart in Fig. 1. To reduce the manpower and time costs of educators in grading assignments, this study will optimize traditional text-based assignments (IPOTA) and develop an ACS for these optimized assignments. The systematic approach of the system aims to reduce the manpower and time costs of educators in grading assignments. The developed system is specifically designed for the IPOTA used in the "Network Engineering Lab.(1)" course, using data normalization to read the assignments and algorithms and databases to assist in grading. This system replaces the laborious and complex manual review, comparison, and verification process of grading assignments with manpower. This structure is mainly divided into two parts, including the first part of the design and production of the ACS and the second part of the "Database Content Design".

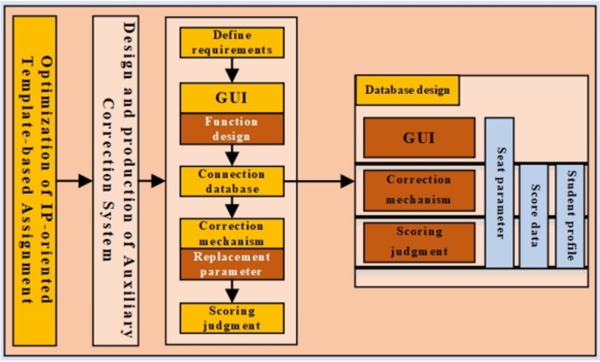
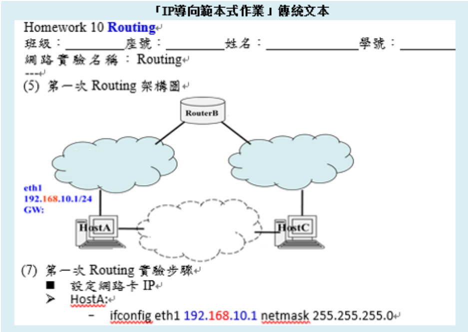


Fig. 1. Structural flowchart of systematic development of ACS for IPOTA

- **The Design and Production of the ACS**

The ACS includes defining requirements, designing a graphical user interface, connecting to a database, implementing correction mechanisms, and establishing scoring criteria. First, we need to optimize the text of the IPOTA used in the past, such as the example shown in Fig. 2(a), by standardizing the data content of the text to facilitate the subsequent development process. From Fig. 1, it can be seen that the traditional text of the IPOTA contains many information, including student information, assignment instructions, images, and required answers. In order for the ACS to read this information accurately, we differentiate the different types of information, as shown in the optimized text flow chart in Fig. 2(b). However, by defining the file content that students submit in advance, we can achieve input file consistency for the ACS. Therefore, we only need to define the algorithm in the ACS to grade and score the assignments of two or more students, solving the problem of handling multiple files. Due to the input file consistency, we can also simplify the graphical user interface (GUI) design of the ACS, as shown in Fig. 3.



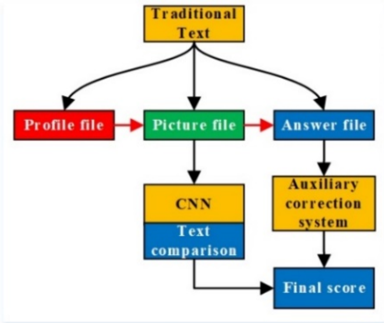


Fig. 2(a). Traditional Text of IPOTA. **Fig. 2(b).** Optimization flowchart of traditional text

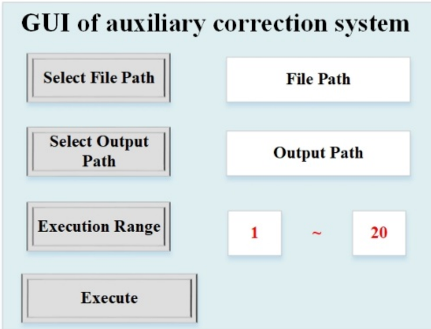


Fig. 3. GUI of the ACS.

From the graphical interface of the ACS in Fig. 3, we only need to define four functions and corresponding buttons, including "Select File Path", "Select Output Path", "Execution Range", and "Execute", to meet our purpose. The operating principle is illustrated in Fig. 4. First, the user needs to specify the path of the file to be graded and the path of the output result file, and define the range of system execution. Finally, clicking the execute button will start the system operation. The "Text Correction" mechanism is achieved through data normalization, defining rules for extracting content from the text, comparing the extracted data with the standardized database, and finally returning the score for text correction, as shown in Fig. 5. The system then waits for the score for image correction to be added, and calculates the final score for the assignment.

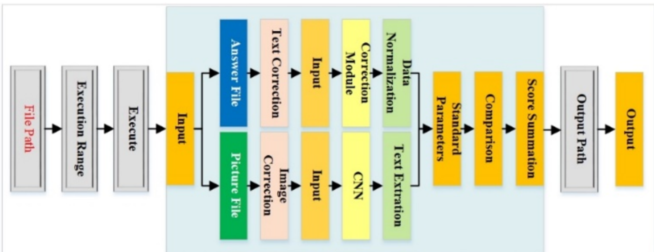


Fig. 4. Operating principle of ACS.

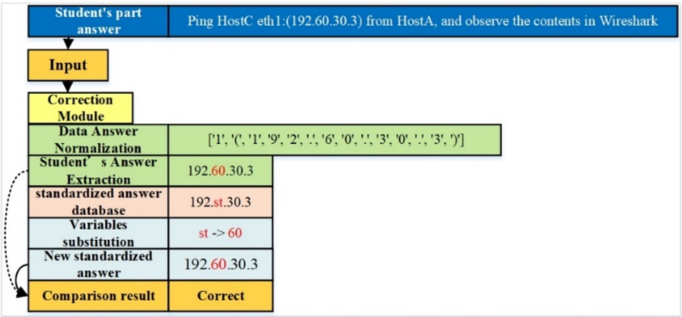


Fig. 5. Text correction mechanism

- **Database design**

The database content includes student information data, seat number parameter data, homework score data, and exam score data. According to the defined requirements at each stage, we also design and establish independent standardized database content for the correction mechanism and scoring judgment. In order to ensure the fairness of the ACS, we will standardize the answer to be compared, so that all students' assignments can be graded against the same standard answer, as shown in Fig. 6.

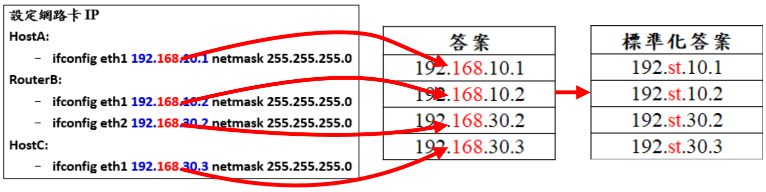


Fig. 6. Database design for the standardized answer of the assignment.

Therefore, Fig. 6 shows the content of the IPOTA implemented in the "Network Engineering Lab.(1)" course, where we transformed the answers of the assignment from traditional text format into a standardized answer database. The number 168 is an example of a teacher's seat number. In order to enable the correction mechanism of the ACS to grade each student's assignment smoothly, we define the seat number of the standardized answer database as a variable "st". This allows for standardized grading of different students' assignment by simply adjusting the value of "st" corresponding to the seat number read by the system. As shown in Fig. 7, if a student's seat number is 60, the seat number after substitution should be 60. Through this approach, we only need to design one standardized database to grade assignment for multiple students.

According to the content of the "Design and Production of Deep Learning Image Recognition" phase, we will establish a database for storing the training samples required for the prediction model. The content will be divided into two sections. The semester is divided into eighteen weeks, and the mid-term exam and final exam should be taken after five homework assignments. The primary key is based on the seat number, with a distinction between the homework assignments and exam scores for the first to ninth weeks of the semester, and the homework assignments and exam scores for the tenth to eighteenth weeks of the semester, with a distinction between the mid-term exam and final exam scores.

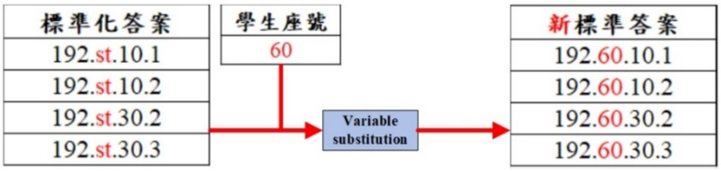


Fig. 7. The process of standardizing and substituting answer.

3 The Design and Production of Deep Learning of Image Recognition Architecture

This section will use the characteristics of CNN and RNN to achieve text extraction. We will also use the Wireshark screenshots submitted by students in the "Network Engineering Lab. (1)" course as training samples for image recognition and design its architecture and analysis. We will explain the neural network layout of this image recognition method and finally test the feasibility of this architecture in text extraction, as shown in the structural flowchart of Fig. 8. It is mainly divided into two stages, including the design of "Image Recognition" in the first stage and the production of "Image Recognition" in the second stage.

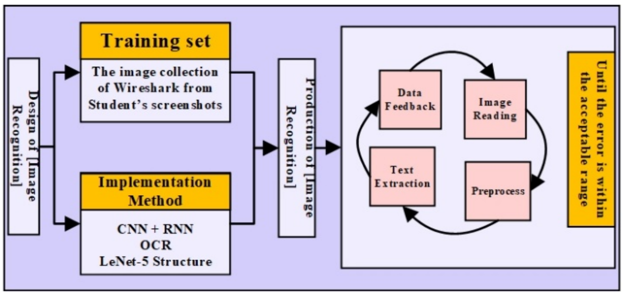


Fig. 8. Structural flowchart of the Image Recognition.

- **The Design of Image Recognition**

It includes the establishment of the training set data and the design of implementation methods. The text to be extracted in this study is the optical characters in Wireshark screenshots. Since the font in Wireshark screenshots is different from the fonts commonly used in computer input, and the area we need to recognize in the Wireshark screenshot is relatively small compared to the entire screenshot, students may compress pixels during the process of taking screenshots and uploading assignments, which makes it more difficult to recognize the fonts in the screenshot, as shown in the example in Fig. 9. One of the assignment requirements in the IPOTA is to capture an image with a packet content that is in Echo reply state and has a Time to live (TTL) of 63. We can also see from this image that the area where we want to extract the text is the area marked with a red line segment and highlighted. In order to enable the neural network to perform image recognition and text extraction smoothly in the area we want, we can use the difference in color of the highlighted area and the other parts of the image to segment the image, so as to achieve recognition and extraction only in the area we need and avoid processing too many areas.

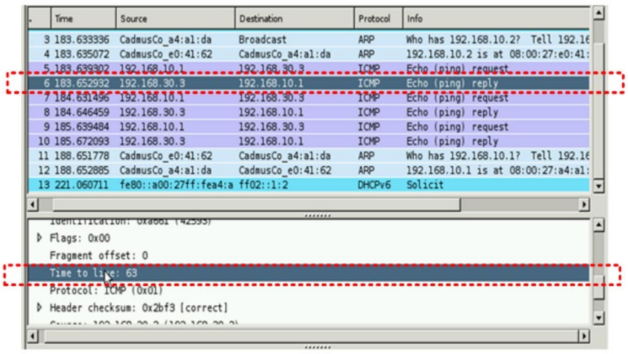


Fig. 9. Example of extracting text from the required area of the Wireshark screenshot.

This study will refer to the CNN structure of LeNet-5 and use the TensorFlow deep learning framework and image projection method in Python to apply the neural network architecture of optical character recognition (OCR), as shown in Fig. 11. Taking Fig. 10 as an example, the pre-training and recognition area is first extracted through color masking, and then horizontal and vertical projection is performed. Each character is cut and saved as a separate image file for classification. Then, "0-9, a-z, A-Z" feature training is performed using CNN, and the training results are input into the RNN network layer. We will use CNN to extract the features of the image, and input the extracted features sequentially into the RNN network layer. In this layer of the network, the neural network will learn from the feature vectors input from the previous stage to facilitate prediction. Finally, the output of the previous layer is subjected to the Softmax function to obtain various different characters. That is, the recognition result defined by us is obtained by repeating and integrating the labels output by the RNN through various operations.

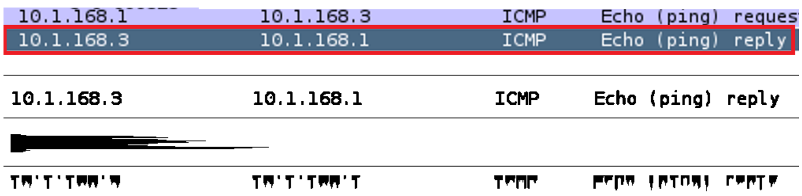


Fig. 10. Color masking and projection segmentation.

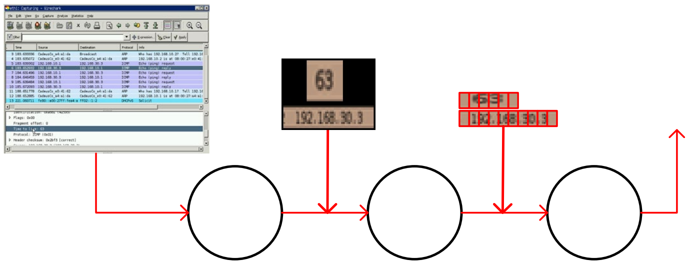


Fig. 11. The network structure of OCR.

- **The Production of Image Recognition**

According to the OCR network structure in Fig. 11, we first preprocess the Wireshark screenshot image with image segmentation to extract the area required for the task. Since the segmented image is too small, we need to enlarge the image before inputting it into the CNN network. This is to magnify the image features obtained through segmentation to facilitate the training of the CNN network and generate better feature maps. However, simply enlarging the segmented image may cause distortion and other issues. Therefore, we need to perform Super-Resolution imaging (SR) on the segmented image as a pre-processing step. From the example in Fig. 12 (Dong et al., 2015)[8], it can be seen that the structure of SRCNN (Super-Resolution Convolution Neural Network) is mainly divided into three steps: patch extraction and representation, non-linear mapping, and reconstruction. That is, the original image to be super-resolved is input, and then the convolutional neural network is used to fit the non-linear mapping of the features to enlarge the original image to the target size. Finally, the image is reconstructed to output the high-resolution image result.

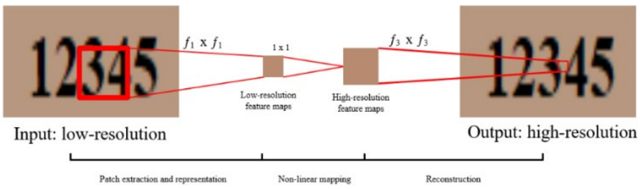


Fig. 12. The structural flowchart of SRCNN.

- **Patch extraction and representation**

Firstly, the low-resolution image blocks of the original resolution image are enlarged to the target size using Bicubic Interpolation, as shown in Equation 1. In this equation, F1(Y) represents the output of this layer, where Y represents the image that has already passed through the Bicubic Interpolation operation. W1 and B1 represent the filters and bias, respectively, where W1 represents n1 filters of size c × f1 × f1, where c is the number of input image channels, and f1 is the filter spatial size. The symbol * represents the convolution operation, and max represents the activation function RELU. At this stage, the image at the enlarged size is still a low-resolution image (as shown in the

Input of the above figure). The image features are extracted from the original input image and are reassembled into high-dimensional feature maps.

$$F_1(Y) = max(0, W_1 * Y + B_1) \quad \textbf{Equation 1}$$

- **Non-linear mapping**

In the non-linear mapping step, the n1-dimensional feature vectors extracted in the first step are transformed into n2-dimensional feature vectors, as shown in Equation 2. Each output n2-dimensional feature vector represents a patch that will be used to reconstruct the high-resolution image.

$$F_2(Y) = max(0, W_2 * F_1(Y) + B_2) \quad \textbf{Equation 2}$$

- **Reconstruction**

In the reconstruction step, multiple related high-dimensional feature vector blocks are averaged to reconstruct the final complete image, as shown in Equation 3. W3 is a set of linear filters, and in this model, the weights and biases of all filters will be optimized.

$$F(Y) = W_3 * F_2(Y) + B_3 \quad \textbf{Equation 3}$$

- **Training**

In SRCNN, the function F mapping between end-to-end needs to estimate the parameters of the neural network Θ={W1, W2, W3, B1, B2, B3} by minimizing the loss between the reconstructed image, which is achieved using the Mean Square Error (MSE) loss function. The MSE loss function calculates the difference between each pixel of the reconstructed image and the original image, as shown in Equation 4. Here, Xi represents the high-resolution image, Yi represents the low-resolution image, F(Yi;θ) represents the SRCNN reconstruction of Yi, and n represents the number of images used for training the neural network or the number of images in a dataset.

$$L(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \| \mathrm{F}(Y_i;\theta) - X_i \|^2 \quad \textbf{Equation 4}$$

4 The application of the ADDIE Instructional Design Model to an ACS

The study will also adopt the theoretical framework of the ADDIE Instructional Design Model to optimize the teaching mode of the "Network Engineering Lab.(1)" course and the text of the IPOTA. We will also develop an ACS to replace traditional teaching aids and achieve the effect of reducing labor costs for educators through automated processes. We will follow the five operational steps and contents of the analysis phase, design phase, development phase, implementation phase, and evaluation phase in the

ADDIE model to collect and analyze data, design courses and programs, develop teaching materials and algorithms, and implement the new system. Finally, we will evaluate the effectiveness of the new system in the classroom, verify the improvement of student performance through the ADDIE-designed course, and the reduction of educators' time costs through the ACS. We will also explore and resolve any problems that arise and thereby enhance the learning effectiveness of the course through the IPOTA and the ACS, as shown in Fig. 13.

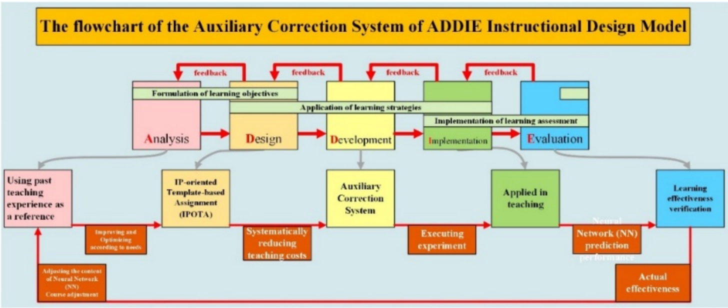


Fig. 13. The flowchart of the ACS of ADDIE Instructional Design Model

5 Analysis of Preliminary Experimental Results of the ACS

According to the time spent on manual grading, time spent on auxiliary grading, and the accuracy of the auxiliary grading, Table 1 shows the results of the ACS analysis. The ACS significantly improves the time spent on grading assignments. Table 1 shows the actual time taken to grade each assignment. The time for manual grading is measured using a stopwatch, starting from when the document is opened, while the time for the ACS is measured from when the first file is selected. The average time required to grade each assignment is calculated by averaging the time spent grading ten assignments to the second decimal place. Since manual grading requires checking each answer one by one, it takes relatively longer. After optimizing the assignments, the number of questions decreased, and the content and answers became more regular and systematic, so there was a certain degree of reduction in the time spent on manual grading. For the ACS, there was no significant increase or decrease in the number of steps involved in grading before and after optimizing the assignments, so the execution time did not differ significantly.

Table 1. Time-consuming to correct an assignment.

	Original Assignments	Optimized Assignments
Manual Correction	70.32s	55.73s
Auxiliary Correction System	20.35s	20.33s

6 Conclusions and suggestions

To solve the problem of time-consuming and laborious manual grading of assignments and to improve the learning effectiveness of network engineering courses, we applied Deep Learning methods such as the CNN and RNN, as well as Image Recognition through Image Projection, to develop an ACS. Additionally, our system includes a standardized database and algorithm utilizing regular expressions to handle different input types and avoid grading errors that could cause system crashes or serious errors in the correction mechanisms. To achieve this, we use data normalization technology, including artificial intelligence, in the development of the ACS and Deep Learning techniques such as CNNs and RNNs in Image Recognition. Finally, we use the ADDIE Instructional Design Model to ensure that the entire development process from analysis, design, development, implementation to evaluation is feasible and efficient for the ACS.

Acknowledgment

This work was supported by the Ministry of Science and Technology, R.O.C., under Grant MOST 111-2410-H-262-001 -.

References

1. Simonyan, K. & Zisserman, A. (2014) "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556.
2. Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In 11th annual conference of the international speech communication association.
3. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). "Gradient-Based Learning Applied to Document Recognition," Proceedings of the IEEE, 86(11), 2278-2324.
4. Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification, In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) pp. 1746-1751.
5. Hu, B., Lu, Z., Li, H. & Chen, Q. (2014). Convolutional neural network architectures formatching natural language sentences, In Advances in Neural Information Processing Systems.
6. Zoizou, A., Zarghili, A., & Chaker, I. (2020). "A New Hybrid Method for Arabic Multi-Font Text Segmentation, and A Reference Corpus Construction," Journal of King Saud University-Computer and Information Sciences, 32(5), 576-582.
7. Mannaz, M. (1998). "An expert teacher's thinking and teaching and instructional design models and principles: An Ethnographic study", Educational Technology Research and Development, 46(2), 37-64.
8. Dong, C., Loy, C. C, He, K. & Tang, X. (2015). "Image Super-Resolution Using Deep Convolutional Networks", IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(2), 295-307.