

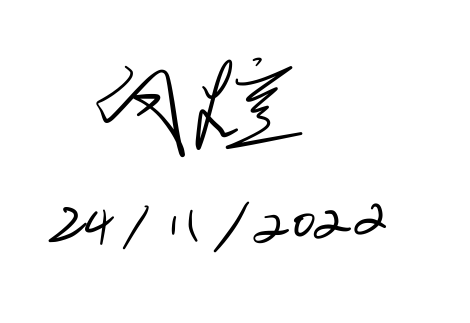
Faculty of Applied Sciences  
Bachelor of Science in Computing

**COMP490 Final Year Project  
Progress Report**Academic Year 2022/23

|  |  |
| --- | --- |
| Text Detection and Recognition in Image-Based on Deep Learning | |
|  |  |
| Project number: | 30 |
| Student ID: | P1908370 |
| Student Name: | Eden Xiang |
|  |  |
| Supervisor: | Jacky Tang |
| Assessor: | Charles Lam |
|  |  |
| Submission Date: | 24/11/22 |

Declaration of Originality

I, Xiang Xuan, declare that this report and the work reported herein was composed by and originated entirely from me. This report has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given in the bibliography.



Abstract

Automated marking systems of today suffer from a number of limitations, the most notable of which are their high cost and difficult operation. This is a proposal for an autonomous marking system that combines three distinct but related tasks: detection, identification, and marking. We made use of certain traditional approaches, but based on them, we also developed some new ways that better fit the job. This project enables users to freely upload images and answers of exam papers, which are then automatically assessed by the system. The goal of this project is to lessen the workload of instructors by reducing the amount of work that has to be graded.

Table of Contents

[1 Introduction 7](#_Toc120498988)

[1.1 Background 7](#_Toc120498989)

[1.2 Project Description 9](#_Toc120498990)

[1.3 Objectives 10](#_Toc120498991)

[1.4 Risk Assessment 10](#_Toc120498992)

[1.4.1 Ethics 10](#_Toc120498993)

[1.4.2 Incorrect Inputs 10](#_Toc120498994)

[1.4.3 Malicious attacks 10](#_Toc120498995)

[1.4.4 Software Supports 11](#_Toc120498996)

[1.5 Summary 11](#_Toc120498997)

[2 Background and Related Work 12](#_Toc120498998)

[2.1 Machine Learning Background 12](#_Toc120498999)

[2.1.1 Neural 12](#_Toc120499000)

[2.1.2 Activation function 12](#_Toc120499001)

[2.1.3 Computer Vision (CV) 13](#_Toc120499002)

[2.1.4 Natural Language Processing (NLP) 13](#_Toc120499003)

[2.2 Related Work 14](#_Toc120499004)

[2.2.1 Dataset 14](#_Toc120499005)

[2.2.2 Detection 14](#_Toc120499006)

[2.2.3 Recognition 15](#_Toc120499007)

[3 Completed Work 17](#_Toc120499008)

[3.1 Dataset 17](#_Toc120499009)

[3.1.1 Resize 18](#_Toc120499010)

[3.1.2 Normalization 18](#_Toc120499011)

[3.1.3 Label Generation 18](#_Toc120499012)

[3.2 Detection 21](#_Toc120499013)

[3.2.1 Result 22](#_Toc120499014)

[3.3 Recognition 23](#_Toc120499015)

[3.3.1 Result 24](#_Toc120499016)

[4 On-going and Future Work 25](#_Toc120499017)

[4.1 Marking 25](#_Toc120499018)

[4.2 User Interface 26](#_Toc120499019)

[5 Conclusion 27](#_Toc120499020)

[6 References 28](#_Toc120499021)

Table of Figures

[Figure 1: Today's electronic marking systems 7](#_Toc120499022)

[Figure 2: Project workflow 9](#_Toc120499023)

[Figure 3: Probability impact matrix before proposed solution 11](#_Toc120499024)

[Figure 4: Activate functions 13](#_Toc120499025)

[Figure 5: CTPN model structure 15](#_Toc120499026)

[Figure 6: CRNN model structure 15](#_Toc120499027)

[Figure 7: Sample dataset 17](#_Toc120499028)

[Figure 8: The interpretation of intersection of union 20](#_Toc120499029)

[Figure 9: CTPN feature extraction 21](#_Toc120499030)

[Figure 10: Sample output of CTPN model 22](#_Toc120499031)

[Figure 11: CRNN feature extraction 23](#_Toc120499032)

[Figure 12: Sample Data 24](#_Toc120499033)

[Figure 13: Gantt chart 25](#_Toc120499034)

List of Tables

[Table 1: Table of risks 11](#_Toc120498987)

# Introduction

## Background

It has now become popular in mainland China to have students answer questions on a special type of answer card, which generally uses positioning points to locate answers and scan them into the computer by dividing them according to the pre-set position. This allows the teacher to mark the paper from anywhere, anytime, via computer or cell phone. Using such methods can effectively reduce the teacher's marking burden and speed up the marking efficiency. In addition, machine marking can be more objective in judging the scores of subjective questions, and can also avoid the original test papers being tampered with, and the confidentiality of the test papers is stronger, which can better protect the interests of the candidates.



Figure 1: Today's electronic marking systems

In terms of practical applications, today's electronic marking systems are divided into two main categories, one using specially designed answer sheets for students to answer on, and the other designing a computer program for students to use electronic devices to answer on the program.

However, these methods are still not good enough. For a method such as using answer cards, it requires an answer sheet specifically designed for the test and may require a special optical recognition machine. And For a method like online quizzing, it is not easy to use in a secondary school classroom where electronic devices are regulated. Another problem is that existing scoring systems can only judge multiple-choice questions [1], neither fill-in-the-blank questions with standard answers nor questions that do not strictly correspond to standard answers, such as reading comprehension and writing. In addition, these methods require additional learning by the user (e.g., how to fill in answer sheets and how to use the online system).

From a technical point of view. The major challenge of this project is the special characteristics of handwritten Chinese characters that pose a challenge to the recognition task [2] [3].

Currently there is a large number of English is other jobs. This type of task includes only 26 classes of letters, while Chinese characters have more than 3000 classes. Moreover, if we directly apply the recognition technology of English characters to Chinese characters, we will face the problem of category imbalance.

The structure of Chinese characters itself is very complex, with upper and lower structures, left and right structures, semi-encircled and fully encircled structures. And many Chinese characters are very similar to each other.

For handwritten Chinese characters, everyone's writing style is different. There are some handwriting styles that are difficult to recognize even for the writers themselves. Characters often stick together in handwritten Chinese characters.

This project employs deep learning technology to construct an intelligent scoring system in order to cut costs and increase usability. Students are permitted to answer examination questions on any white paper. The system analyzes the student's responses and assigns a grade based on their responses. There are three components to the main program: handwriting detection, recognition, and response comparison.

## Project Description

This project plans to use deep learning technology to train neural networks to perform Chinese character detection and recognition tasks. Figure 2 shows the workflow of this model.

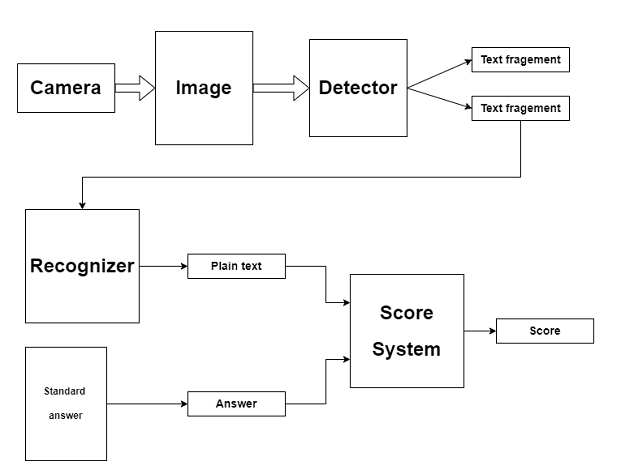


Figure 2: Project workflow

First the camera takes a picture of the test taker's answer and transfers it to the detector. detector detects and locates the text information in the picture and splits the picture into text fragment which subsequently will be recognized by the recognizer. After the recognizer recognizes the candidate's handwriting font, the scoring system finds the answer corresponding to this question and compares it. Finally, a reference score will be given based on the result of the comparison.

## Objectives

The objectives of this project is to develop a test paper marking system which contains the image detection, recognition and natural language processing capabilities using deep learning technology.

* This project should be well integrated with all functions to implement an automatic marking system.
* The model should read and process paper images from camera.
* The model should identify text in the image and segment it based on the question.
* The model should recognize the text in question answer and output the plain text.

## Risk Assessment

### Ethics

Collecting sensitive information about users when gathering information may lead to an ethical crisis for the project. Although most of the training data can be found on the internet, this project may still need human participants and collect their handwriting. This project should avoid collecting sensitive data and protect the rights of participants. For data collected, all data stored should be anonymized and should not contain sensitive data about living individuals.

### Incorrect Inputs

Accepting a data that the program cannot process may cause the whole program to crash. This project should avoid this situation and can take measures such as restricting the file upload type and handling incorrect data in the backend.

### Malicious attacks

Since this item can be deployed on a web page, it may be a potential target for attackers. A third-party platform can be used to protect the website.

### Software Supports

Since this project is software development, there may be situations where the tools or frameworks used are no longer supported or deprecated in future versions. This could result in the system becoming completely unusable in the near future. Popular software or software developed by a large corporation should be used, because they are more likely to be supported over time and less vulnerable.

Table 1: Table of risks

|  |  |
| --- | --- |
| Priority | Risk Identifier and Description |
| 1 | Risk 1: Ethic risks |
| 2 | Risk 2: Incorrect Inputs |
| 3 | Risk 3: Malicious attacks |
| 4 | Risk 4: Software Supports |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Probability** | High | Risk 4 |  |  |
| Medium |  | Risk 3 | Risk 2 |
| Low |  |  | Risk 1 |
|  |  | Low | Medium | High |
|  |  | **Impact** | | |

Figure 3: Probability impact matrix before proposed solution

## Summary

The structure of this report is as follows: Chapter 2 covers the knowledge base and neural network model used for this project. The third chapter describes my design methodology and innovations. Chapter 4 summarizes the tasks to be completed throughout the following semester.

# Background and Related Work

## Machine Learning Background

Most of the functionality of this project is done by deep learning models. This section will introduce some of the basics of machine learning that were used in this project.

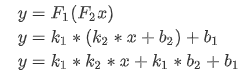
### Neural

Inspired by biological neurons and for computational convenience, an AI neuron can accept values in the interval -1 to 1 and output a value of -1 to 1. Neurons are combined in layers to form a neural network.

A neural network is a machine learning technology that teaches computers to process data like the human nervous system. Deep learning is a type of machine learning that involves interconnected nodes or neurons in a layered structure similar to the human nervous system. It offers a framework that enables computers to continually grow and learn from their failures. Thus, artificial neural networks strive to perform difficult tasks, such as document summarization and facial recognition, with greater precision.

### Activation function

Activation functions are extremely important for constructing a neural network. Since the models we want to fit using neural networks are usually complex, if we do not use an activation function, i.e., if we only use the linear relation *y = kx + b* to fit, the result will only be an activation function (see Proof 1). So, we need the activation function to provide a nonlinear variation to the neural network. A simple demonstration of why linear relations cannot fit complex curves follows.



*k1* and *k2* can be combined and considered as a new weight *k3*, and *k1 \* b2 + b1* can be considered as a new constant value

The activation functions used in this project are as follows:

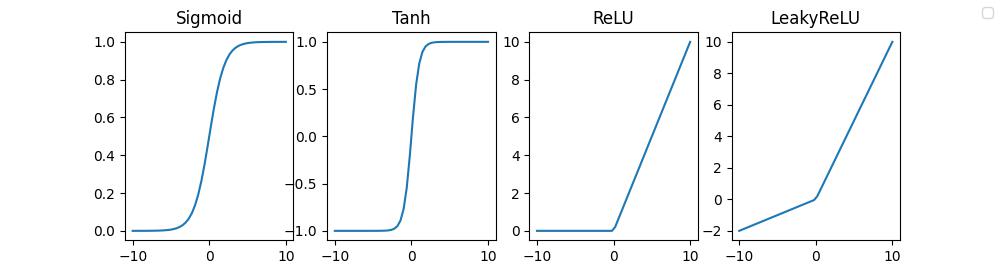


Figure 4: Activate functions

Sigmoid and Tanh are very good activate function, they can convert the output in to specific value. But using Sigmoid and Tanh may cause the vanishing gradient problem, which can be a hurdle while updating parameters value in back propagation stage [4] Therefore, we only use them in the last layer.

The *ReLU* function is simple and improve the computation efficiency a lot. It Derivative for its valid area is always be1 [5]. However, using a large number of *ReLU* as the activation function will result in a large number of "dead" neurons in the neural network, whose parameters will never be updated afterwards. Therefore, *LeakyReLU* provides a small gradient for those neurons that are not useful to be updated, and this activation function will be used extensively in this project's network.

### Computer Vision (CV)

Computer vision refers the ability of computers to extract information from images or videos. It is one of the hottest subfields of machine learning area now. Computers can distinguish and recognize images by using this information extract from neural networks. There are some typical tasks in CV: image categorization, object recognition, instance segmentation, and object tracking [6].

### Natural Language Processing (NLP)

Natural language processing means to give a computer the ability to comprehend meaning of words in a human-like manner. Since the introduction of the Bert model, which will be mentioned in the marking part, the NLP branch has attracted the attention of researchers again. Now NLP plays a growing role in enterprise business operations [7].

## Related Work

### Dataset

Due to the possibility that the dataset of handwritten Chinese characters is insufficient, the dataset of the ICPR2018 competition is utilized to train a basic Chinese character detection model. For the recognition model, a Kaggle dataset of handwritten Chinese characters is used.

ICPR2018 is an AI competition that is being hosted by the Chinese Association of Automation (CAA) and the Institute of Automation of the Chinese Academy of Sciences [8]. It provides many datasets for training, including but not limited to images, presentations, videos, biometrics data. Here, one of the datasets from the Text Detection of Web Images-Competition is utilized.

Kaggle is a community that provides a platform for users to communicate and offers several publicly accessible datasets for training purposes. The dataset of handwritten Chinese characters utilized for this study comprises 7330 characters with around two hundred data points per character [9].

### Detection

Object detection is a component of Computer Vision that refers to a system that can figure out the presence and position of a desired object or body inside an image. Identification tasks include a variety of real-world applications, such as autonomous vehicles, face detection, target tracking, pose detection, etc. In this project, CTPN is employed to complete the detection task.

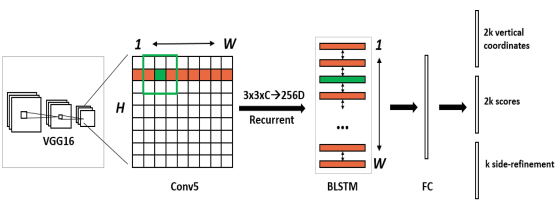


Figure 5: CTPN model structure

Text detection in real-life scenarios is a complex task and there are several approaches to achieve it. One approach is based on sliding windows [10] However, this method is less efficient and the detection of different scaling and rotation is not ideal.

Another well-known approach is region-based approach [11] Despite being faster than sliding-window methods, the region-based approach needs additional steps to do the region detection, therefore the false detections require additional classification steps.

This project made some modification in Connectionist Text Proposal Network (CTPN) [12] which is based on VGG model [13] and BLSTM model [14].

### Recognition

For the recognition, the Convolutional deep neural network (CDNN) shows very good results on classification tasks, therefore many models developed since then are basically based on CDNN. like image classification [15].

There are also attempts to deconstruct text for recognition [3]. This approach is a good solution to the problem of exhaust *softmax* layers in representational capability due to the large character set that needs to be recognized, such as Chinese characters (more than 7,000 commonly used characters).

This project will use the Convolutional recurrent neural network (CRNN) [16] to do the recognition work. Considering that Chinese characters are based on pictographs, recurrent neural networks can better extract the relationship between parts and parts.

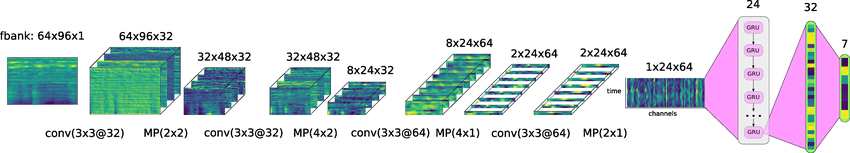
**

Figure 6: CRNN model structure

CRNN, also known as Convolutional Recurrent Neural Network, is a very classic model used primarily for end-to-end recognition of sequence data of undefined size, without cutting content into single characters, but converting text recognition into a time-series-dependent sequence learning problem.

A Connectionist Temporal Classification Loss (CTC Loss) is developed for issues involving complex connections between sequences, such as matching each letter to its position in a voice recording. It computes the separation among a continuous time - series data and the target. The *CTC loss* function has a wide range of applications in Speech Recognition and lipreading. *CTC loss* is already integrated in *PyTorch* and can be called directly.

# Completed Work

This semester's work focuses primarily on the preceding portion of the workflow described in Chapter 1, including the training of the detection and recognition models. Additionally, we must often process the training data prior to training the models. The improvement of this project and the work performed in these areas will be explained separately below.

## Dataset

In order to strengthen the model's ability to recognize handwritten Chinese characters, a program that automatically generates handwritten images was developed specifically for this project.

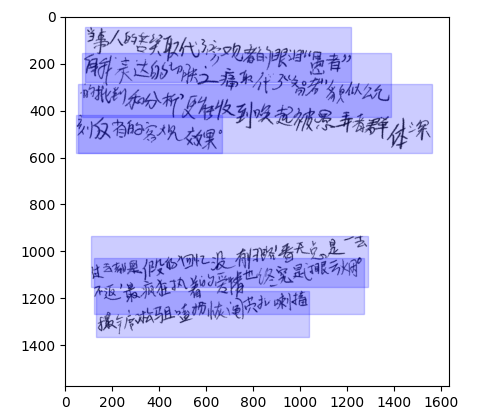


Figure 7: Sample dataset

Since we need to ensure the generalizability and stability of the model, the generated images for training are rotated and scaled. To make transfer learning easier, the generated dataset labels will be generated according to the ICPR dataset format (x1, y1, x2, y2, x3, y3, x4, y4, label). In addition to this we need to do some pre-processing of the data, including resize and normalization.

### Resize

Since the input size of images is uncertain, scaling large files to the right size is necessary. However, we cannot simply rescale image to the desired, this approach also brings a new problem that we need to process the target coordinates of the data annotation as well to ensure that the new coordinates can match the scaled image.

### Normalization

By default, the value of image range is [0, 255]. The normalization method ensures that the input to the neuron is always between 0 and 1, which increases the stability of the training. In contrast to the typically utilized batch normalization approach, the instance normalization method is employed here for normalizing since it has proven to be more effective in image processing [17].

### Label Generation

This model has two tasks, a classification task of text and background and a regression task of text position. To facilitate the model calculation, we segment the image into equal-width sliding windows of different heights for recognition, called anchor boxes, with the total number of windows being (N x H x W). Then the intersection of union (IoU) of each small window is calculated to get the confidence level of each window.

Anchor boxes of different sizes are generated at all possible positions. *Numpy's* matrix computation is used here to improve operational efficiency. The basic step is to generate a set of anchor boxes of different sizes at the first position and then move to the next point until all positions have anchor boxes generated.



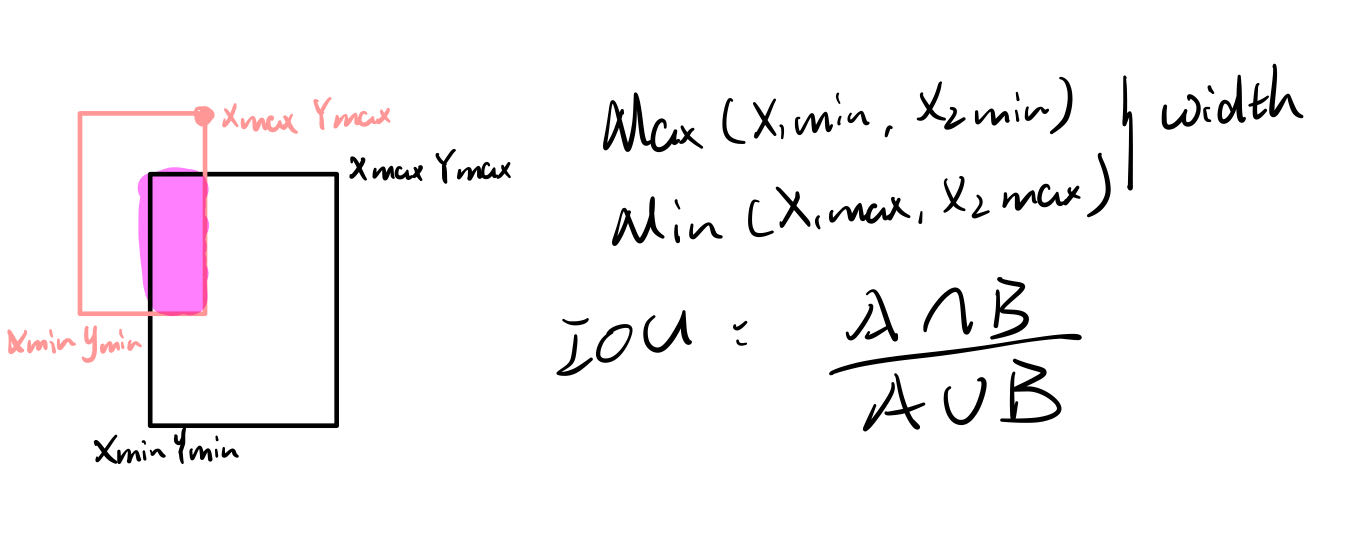


Figure 8: The interpretation of intersection of union

IoU is a measure of the overlap between two bounding boxes. If the IoU is greater than 0.5, we can assume that this box contains the target, and the higher the IoU, the more precise this box is. The formula and code to calculate the IoU is shown below.





## Detection

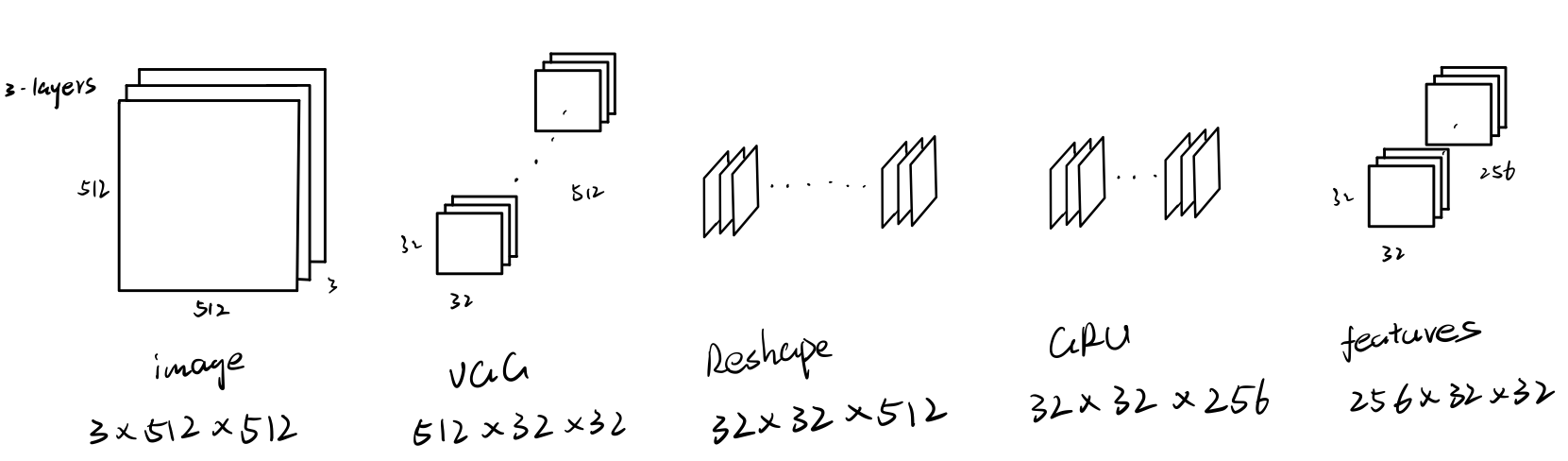


Figure 9: CTPN feature extraction

In CTPN, in order to be able to pass features extracted by VGG network to BLSTM model when performing, it needs to flatten features in to 1D vector, which actually diminishes the relationship between the features in the vertical direction. I make some modifications of this network. It can be seen that the original BLSTM layer is changed to a GRU layer in this project, so that the extracted features do not have a loss of longitudinal relationships. We believe this little modification can make the performance of the neural network even better.

After passing the images into the neural network, the neural network outputs the score of each anchor and the coordinates of the anchor, which is used to determine whether it contains content as text or background. Unlike the original CTPN, because our target detection task is relatively simple, the detection of text does not contain a complex background, so not using refinement can reduce the workload a lot

### Result



Figure 10: Sample output of CTPN model

Each anchor is assigned a proper height and a score by the network. We can presume that anchors with a score greater than 0.5 contain the text that we wish to identify. The algorithm next analyses the anchor points with positive predictions and merges them into a number of text-containing picture segments that will be given to the following recognition model.

## Recognition

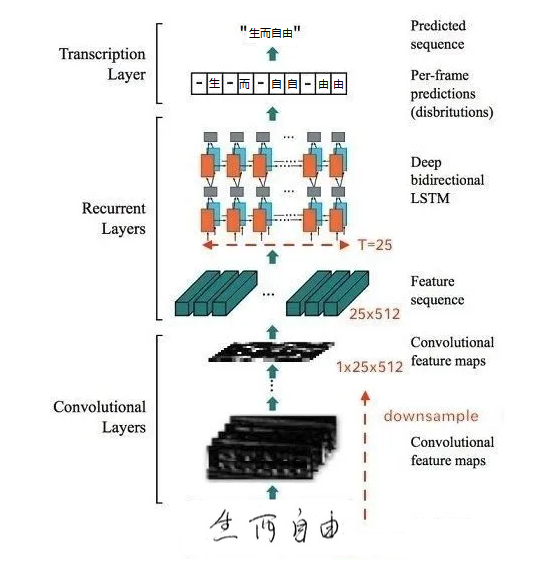


Figure 11: CRNN feature extraction

This project's recognition tasks will be handled by a Convolutional recurrent neural network (CRNN). In light of the fact that Chinese characters are based on pictographs, recurrent neural networks can extract the relationship between parts and parts more effectively.

### Result

The results for each time frame are acquired following the application of the RNN model to the text graphics in order to recognize them. In order to create the anticipated output sequence, the sequences of these characters are concatenated, and the blank symbol (which is represented by a minus sign) is inserted between the repeated letters in the text labels.



Figure 12: Sample Data

For instance, the RNN model applied to the image above this page may produce the output string "运动-员--回到-起起-跑--线线上." After that, we transcribe this sequence, which means that we remove any letters in the sequence that are repeated consecutively, and then we clear the paths of any and all blank symbol so that we may receive an accurate prediction of "运动员回到起跑线上".

# On-going and Future Work

So far, we have completed a review and analysis of existing detection and recognition techniques and completed training of the associated neural networks. In this progress report we point out the possible shortcomings of the existing neural networks and suggest possible improvement directions for these problems. However, throughout the project, there are still many unfinished parts. One of the main tasks is the design and training of the model for judging text similarity, but we also need to consider the design of the user interface and the testing and optimization of the model.

In the next semester, we will focus on completing these two tasks and optimizing the two models completed this semester so that they can work together better.

In the next two subsections the ideas for the implementation of the two tasks of marking model and user interface will be discussed, as well as the possible difficulties to be overcome and the related considerations.

Here is a Gantt chart of the tasks for the next semester

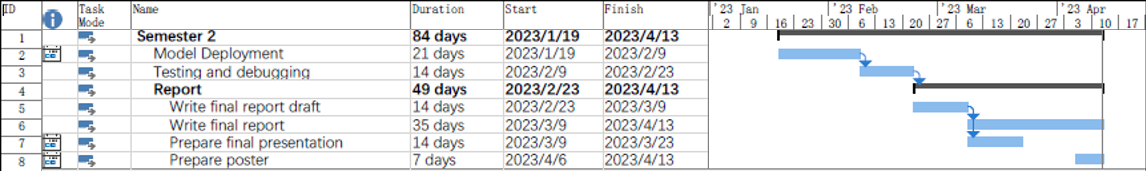


Figure 13: Gantt chart

## Marking

The intent is to create the full scoring system using the Bert model. Bert (Bidirectional Encoder Representations from Transformers), a Google-developed NLP algorithm introduced in 2018, is a language modelling tool. It has been demonstrated to be an efficient language model has been used in practice in various languages.

The project intends to use this model to analyse the outcomes of the identification of standard answers for sentence meaning correlation and to grade the answers based on the degree of correlation.

## User Interface

User interface design, often known as UI design, describes the connection that exists between a website or design and its visitors. It includes every component of the design that has to be there in order for users to be able to navigate the site and perform the desired actions. The interaction between the user and the design when it is being used is what is meant by this term. The user interface (UI) and the user experience (UX) has a close association. A good UI can not only improve the UX, but it can also lower the amount of time and effort that users need to invest in order to learn how to use the system. The end consequence is that the user will have an easier time locating the information that they require, which will lead to an increase in their level of satisfaction while simultaneously reducing the amount of time spent searching.

# Conclusion

In conclusion, the current method of marking still has a great number of deficiencies, such as a high cost and an absence of convenience. These issues are tackled head-on by this project, which involves the development of an intelligent marking system based on an integrated neural network model. During the course of this semester, we will discuss some of the probable drawbacks of the currently available neural networks and offer some suggestions for how these issues may be improved. In addition, data augmentation and transfer learning are utilized in order to address the issue of an inadequate number of samples encountered throughout the training process.

Despite this, there are still a significant number of incomplete aspects across the project. The creation and instruction of a model for determining the degree to which two pieces of text are same is one of the most important jobs, but we also need to think about the design of the user interface as well as the testing and optimization of the model.

# References

|  |  |
| --- | --- |
| [1] | Catalan and J. Antonio, “A Framework for Automated Multiple-Choice Exam Scoring with Digital Image and Assorted Processing using Readily Available Software,” *LSU Research Congress,* 2017. |
| [2] | N. Sahu, R. K. Rathy and I. Kashyap, “Survey and Analysis of Devnagari Character,” *International Journal of Computer Applications (0975 – 888),* June 2012. |
| [3] | G. Kim, J. Son, K. Lee and J. Min, “Character decomposition to resolve class imbalance problem in Hangul OCR”. |
| [4] | A. Syed, “Why You Need to Stop Using Sigmoids,” 28 1 2021. [Online]. |
| [5] | AskPython, “Python ReLu function – All you need to know!,” [Online]. Available: https://www.askpython.com/python/examples/relu-function. [Accessed 1 11 2022]. |
| [6] | J. Couto, “An Introductory Guide to Computer Vision,” 30 04 2019. [Online]. Available: https://tryolabs.com/guides/introductory-guide-computer-vision. [Accessed 01 11 2022]. |
| [7] | I. C. Education, “Natural Language Processing (NLP),” IBM, 2 7 2020. [Online]. Available: https://www.ibm.com/cloud/learn/natural-language-processing. [Accessed 1 11 2022]. |
| [8] | C. Wolf; J.M. Jolion, “International Journal of Document Analysis,” *Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms,* 2006. |
| [9] | P. Bliem, “Handwritten Chinese Character (Hanzi) Datasets,” [Online]. Available: https://www.kaggle.com/datasets/pascalbliem/handwritten-chinese-character-hanzi-datasets. [Accessed 1 11 2022]. |
| [10] | K. Wang, B. Babenko and S. Belongie, “End-to-end scene text recognition,” *International Conference on Computer Vision (ICCV),* November 2011. |
| [11] | C. Yao, X. Bai, W. Liu, Y. Ma and Z. Tu, “Detecting texts of arbitrary orientations in natural images,” *IEEE Conference on Computer Vision and Pattern Recognition,* June 2012. |
| [12] | Z. Tian, W. Huang, T. He, P. He and Y. Qiao, “Detecting Text in Natural Image with Connectionist Text Proposal Network,” *Computer Vision – ECCV,* 2016. |
| [13] | K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv:1409.1556,* 2014. |
| [14] | Z. Tian, W. Huang, T. He, P. He and Y. Qiao, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *ECCV Computer Vision ,* 2016. |
| [15] | D. Ciregan, U. Meier and J. Schmidhuber, “Multi-column deep neural networks for image classification,” *IEEE Conference on Computer Vision and Pattern Recognition,* 2012. |
| [16] | X. Fu, E. Ch’ng, U. Aickelin and S. See, “CRNN: A Joint Neural Network for Redundancy Detection,” *IEEE International Conference on Smart Computing,* 2017. |
| [17] | Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky, “Instance Normalization: The Missing Ingredient for Fast Stylization,” Nov 2017. |
| [18] | R. Smith, “An Overview of the Tesseract OCR Engine,” *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007),* September 2007. |
| [19] | A. Gupta, A. Vedaldi and A. Zisserman, “Synthetic Data for Text Localisation in Natural Images,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2016. |
| [20] | M. Busta, L. Neumann and J. Matas, “FASText: Efficient Unconstrained Scene Text Detector,” *IEEE International Conference on Computer Vision (ICCV),* 2015. |
| [21] | “Optical character recognition - Wikipedia,” 30 08 2022. [Online]. Available: https://en.wikipedia.org/wiki/Optical\_character\_recognition. |