Handwritten Text Recognition

Team #6
21700082 김도영
21800097 김성민
21800201 김현욱
22000372 송다빈

1. Problem description

- With the development of electronic devices, we are less likely to write letters. You can type on a keyboard, voice recognition, and recently take pictures to search for items. However, when applying for complaints from public institutions or creating a new bank account at a bank, it is often necessary to write it in writing. Employees visually check written documents and enter them directly into electronic documents on the computer. This is a very inefficient method, and depending on the employee's proficiency, inexperienced employees may enter incorrect information.
- In recent cases, in 2020, when COVID-19 was rapidly infected, restaurant A was closed due to accidental damage caused by an employee of a public health center in Busan. The health center staff saw the related documents and posted the movements of the confirmed patients on the city hall's COVID-19 website. The route posted on the website included Restaurant A, which was actually the day when Restaurant A was not open. This was incorrectly stated due to the mistake of the health center staff. As it was rumored to be a store visited by confirmed patients due to the wrong human traffic record, the restaurant ended up closing down as the number of customers decreased.
- The above case is a problem that occurred in the process of writing documents into electronic documents on a computer. Converting written documents into electronic documents immediately will significantly reduce the likelihood of problems and increase the efficiency of work. Therefore,

our team thought that we should be able to accurately recognize the written text, which is the step before transmitting data from written documents to electronic documents.

- Therefore, our team decided to create an AI model that accurately recognizes handwriting data.

2. Background and conventional approach

- Of course, there is a technology called Optical Character Recognition (OCR), which is similar to our project topic, which recognizes human writing on paper and converts it into text data. However, through a class called AI_project, it was difficult to implement OCR technology and decided to utilize the most similar mnist data. To prepare for this project, we used the mnist dataset that we learned in class. The Mnist dataset is a simple computer vision dataset and consists of human handwritten images, as shown in the picture below.

- Since it utilizes that data and includes alphabets, it is necessary to properly preprocess the dataset for training to increase recognition accuracy.

3. Proposed method

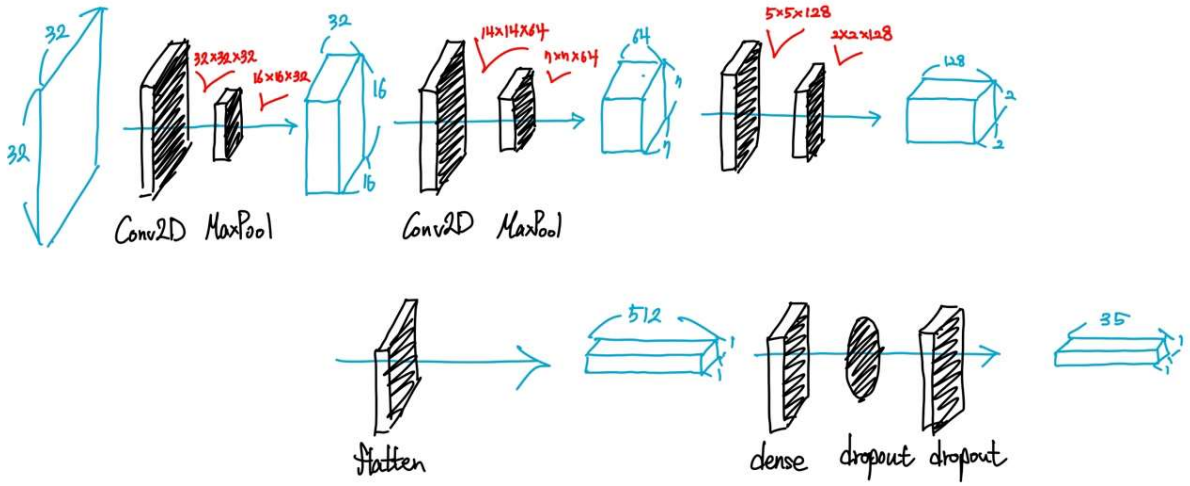
- Our team trains with mnist datasets to create models that can accurately recognize handwritten numbers and alphabets. An essential mnist dataset for training was collected from an open source platform called kaggle.

A. Data collection and preparation

- The mnist dataset collected by kaggle had only numbers and alphabets. I couldn't find Korean because I couldn't find the data.
- The data to be tested were tested with word photos found on the Internet or screenshots of letters written directly on a tablet PC.

B. AI approach

model. Sequential()



- It is a classification model that obtains a probability value for each class as a result of performing feature extract on data entering the input through the Convolution layer and the Max Pooling layer.
- To obtain the model's loss, we used the category_crossentropy function and used Adam as the optimizer.

- Learning was conducted with epoch = 50 and batch_size = 32, and the final acuity was 0.9408.

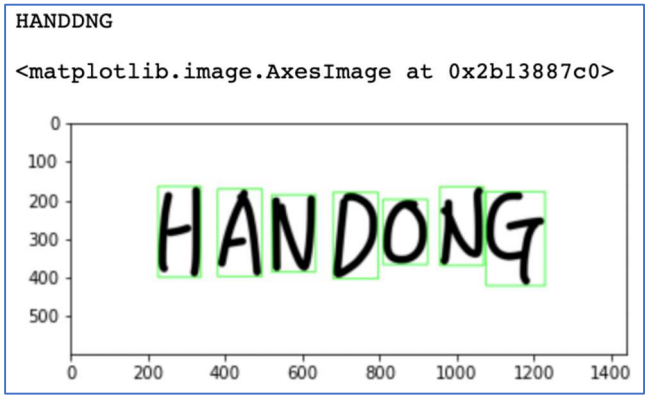
C. Evaluation criteria

- Relevance: It relates to the learned image transformation, and we were able to learn how to accurately recognize handwriting except in a few cases using a handwriting recognition model.
- Efficiency and Achievement: One of the most important criteria in this project is how accurately numbers and alphabets can be recognized in writing when running a model with a data set. Most of our models were recognizable accurately.

4. Results

- Handong has the alphabet 'O'. Therefore, the alphabet 'O' is recognized as the number '0' or the alphabet 'D', and other test data accurately predicts the character

- The results are as follows:

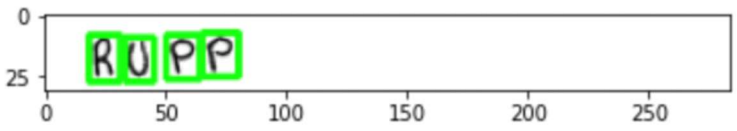
1. 

<data : [HANDONG]>

2.

```
RUPP

<matplotlib.image.AxesImage at 0x7f60301ea6d0>
```



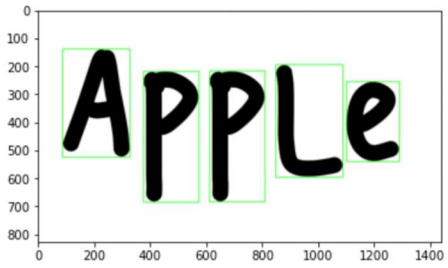
<data : [RUPP]>

3.

```
In [152]:
1 letter,image = get_letters("./archive/our_train/train_6.jpeg")
2 plt.imshow(image)
3 word = get_word(letter)
4 print(word)
5 plt.imshow(image)

APPLE

Out[152]: <matplotlib.image.AxesImage at 0x2ff427760>
```

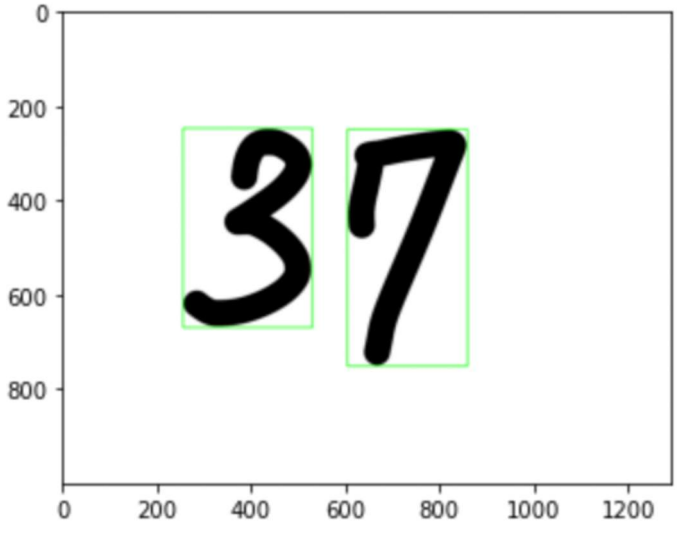


<data : [APPLE]>

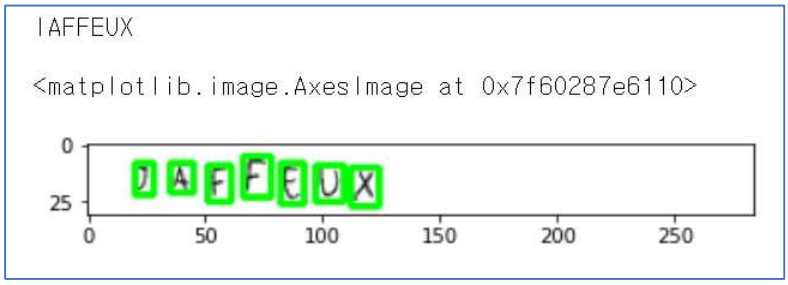
4.

```
37

<matplotlib.image.AxesImage at 0x2ff727b50>
```



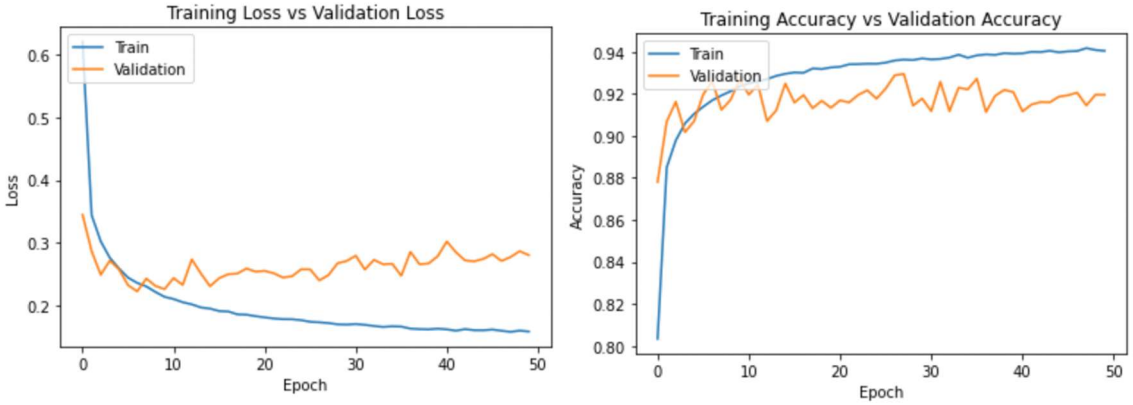
<data : [37]>

5. 

<data : [IAFFEUX]>

- We wrote the word in the test data and it recognized each character correctly. For recognition, we draw several bounding boxes for each character in the output image. All data sets are uppercase and must be capitalized on the test data set. This is the limitation of our project and data set.
- Most of our models recognize it correctly, but we have found that we output unstable results when recognizing the alphabet 'D' in several cases. It found that there was no data on the alphabet 'O' in the mnist dataset received from kaggle. Therefore, it seems that 'O' was recognized as 'D' or the number '0' in 'HANDONG'.

5. Conclusion



- The left image is a comparison of the loss values and the right image is a result of accuracy. We estimated each loss value and accuracy for history as 'loss' and 'val_loss' and 'val_loss'. The total Epoch is 50 and it seems to have been predicted correctly.

- Finally, we learned to create deep learning models and process datasets to achieve our goals. The most interesting thing was the data set.
- We preprocessed the data according to the purpose, in order to achieve the desired result. However, we were not well aware of the importance of identifying training datasets. So we didn't recognize the letter 'O' properly.

6. Division of work

- All the team members collaborated to the end. Data, data collection (mnist), writing and presenting slides, etc. for selection of topics.
- In addition, our team used an open-source model, and AI hub's data was installed by all team members, but it was not implemented because it did not understand the model accurately or because it was different from the environment written by the author. So I chose the model I found on the open source platform called kaggle, but this was also done accurately to only one team member. We had time to think about what parts to modify for the target function of the project in open source.
- I researched Kaggle training data and problem and doing some analysis of model sequences. I didn’t have much memory on my notebook, so another person trained the MNIST and Kaggle Data set. Also, I recorded the presentation of PPT last in 14th week

7. Discussion (per each member)

- We wanted to make OCR Model to detect some handwritten data. But It’s too difficult to implement the detail things detecting the location of data, classifying the number, letters of given. Therefore, we found some open-source model and analysis about how this model can classify the data. We didn’t learn exact concept of optimizer and activation function in the back propagation. I think it’s our task to study the detail operation on deep learning model.

8. Reference

- Kaggle model source
  - https://www.kaggle.com/code/aryantiwari123/handwriting-recognition-deep-learning-tensorflow/notebook
- PDF of processor Choi in LMS