FANCY: Review Note Auto Generation Application*

Capstone Design Project Proposal

Minji Cha[1 [2018311658]],
Daehee Kim[1 [2015313077]],
Eunji Lee[1 [2017314519]], and
Daeyeol Jo[1 [2015311711]]

Sungkyunkwan University
2066, Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do, Republic of Korea
{cookkyh99,dehee1030,hello06210,dyjo.seoul}@gmail.com

Abstract. Using review note is helpful for learning. However, it takes a lot of effort to make a review note, so most of the students think it is such a hassle. Several services have launched to help users make review notes, but those services still make users to grade, cut and paste problems by themselves. Therefore, we decided to develop a service that helps users to make review notes easily and comfortably. Our service automatically grade the answers and provide additional functions that could help students while studying math: regenerating problem, and shuffling answer options.

Keywords: Review note · Auto scoring · Object detection

1 Introduction

The review note refers to a note in which students organize the problems that they answered incorrectly. With review notes, students can intensively check and train their weaknesses in knowledge. However, the process of making review notes is never easy. Usually students transcribe problems to another notes, or cut the problem area and paste it to another note. To solve the inconvenience in making review notes, several review note generator services have launched. However, those services still have limitations that users have to grade the answers by themselves and select area to be cut. Therefore, we planned an AI-based review note service that can free users from inconveniences. In this project, our team proposes the service that automatically creates review notes using photos of problems and answers.

2 Related work

There was no prior approach to use AI for creating review notes, but since there are many review note applications in app store, we first analyzed about

* Supported by SKKU.

functions and limitations of those applications. Also, we compared how our service would be different from them.

First, there is the application called '1등 오답노트'. User can take a picture of problem and save it. Later, the user can print these problems in PDF format. However, this application does not provide additional functions, so it is not different from offline review notes other than that the user takes pictures of problems instead of writing down those problems. After all, this application just provides a storage for wrong problems. In fact, when we searched applications with the keyword 'review note', most of them were in this form.

Second, there is one application called 'AI 오답노트' that uses AI technique. User still has to grade the problem by themselves, but with this application detects the wrong problems automatically from user's picture. It recognizes and distinguishes O/X marks scored by the user and create a review note with the wrong answers. It is more convenient than the previous application, but it still doesn't provide additional features to help studying.

Lastly, there is '콴다(QANDA)', one of the popular studying applications with AI. But QANDA just shows answers of questions, so many students tend to depend on the service, not truly learning something.

Since many applications doesn't provide functions to easily create review notes and additional features for studying, we decided to implement those functions in our service.

3 Proposed service

3.1 Main features of the project

Auto scoring

User takes a picture of the workbook and uploads it. Then our YOLO model detects the area of problem, problem number, page number, and answer options. Based on the result of YOLO, we extract the user's answer and recognize the page number and problem number with tesseract. We compare user's answer with the answer saved in our database, and finally save the graded result in database.

Regenerating Problems

Whenever user clicks "regenerate question" button, we change the variables of the question to let user solve the question again, By this function, user can solve the same type of problem with different variable, so it can help user to solve specific type of problem by their own.

Shuffle Answers

We knew that regenerating problem is a very hard technique, and there might be many limitations of that function. So we support additional feature, shuffling answer, to help users study math. We randomly shuffle the multiple choice question's options to make user not memorize the answer number and solve the question again.

3.2 Ideal circumstance

It would be wonderful to support all the extreme cases. But because of the lack of the resources, we limited the input like as follow to narrow the scope:

1. Our model can also detect subjective question area, but we only implemented auto grading functions for multiple choice questions.
2. User should mark the answer like as marking on the OMR card.
3. Since it was hard to create a real life dataset, our model was trained with PDF files with little bit of augmentation like rotating and adding a noise. Due to the lack of the data, our model seemed like it was working well with a scanned image. So the user should take a picture as if it was scanned.

4 Design and Implementation

4.1 System Architecture

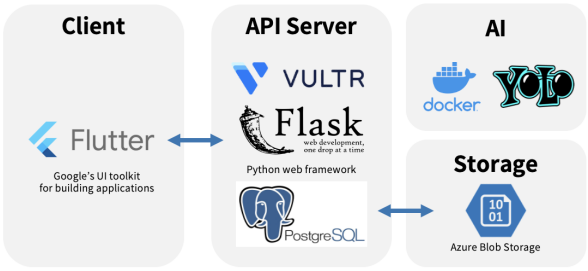


Fig. 1. System architecture

To make users easily upload images to our service, we developed an application. Especially, we chose Flutter to support both Android and iOS platform.

We developed our server with python Flask because it has very simple architecture and all of the members were familiar with using Python. Since vultr provided $100 credit for new users, we deployed our server on vultr. We set postgresql database on our web server, and we used azure blob storage to save images, to access image quickly from the application.

Finally, to run AI model in our server, we used darknet YOLO model. The interaction between the app and the user should be instantaneous, so we considered AI model's speed to be one of the most important feature. That's the reason we chose the fast object detection model, YOLO. We created a docker container including the YOLO model and some essential libraries, and ran commands in our web server to retrieve a result from YOLO.

With this architecture, the application cannot directly access the database, so it increased security. Also, the client(application) could access to image directly by url, thanks to azure blob storage. So it didn't have to get raw image from the database, so this increased the application's overall speed.

4.2 Frontend

4.2.1 UI/UX

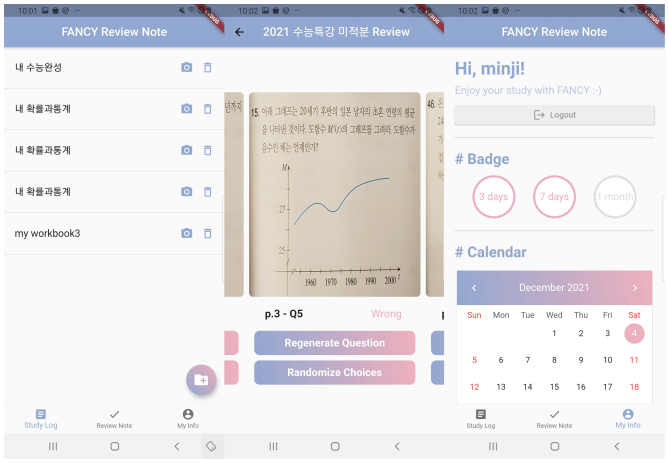


Fig. 2. UI/UX of Application

Our main page is workbook list page. Users can check the workbook list they created, and user can create new workbook by clicking button in the lower right corner. User can take picture of problem they solved by clicking the camera button in each entry. The problem is automatically scored after taking a picture, and the information is stored in the database.

When users click a entry in a workbook list, they can check problems included in the workbook. On this page, user can create a new review note by selecting the problems they want to include in a review note.

In 'Review Note' tab, users can check the review note list they created. When users click a entry in the list, they can check problems included in the review note. Users can see the bigger image of the problem by clicking each problem, and users can regenerate problem or randomize choices by clicking each buttons.

Moreover, when users click the 'My info' tab, user can check their attendance on the calendar. To motivate users, our service gives badges in case users attends 3, 7, 30 days consecutively. Users can check collected badges on this page.

4.2.2 How to Crop Images

There was a need for a technique that enables users to crop the image area after taking a picture of questions. So we utilized 'edge_detection', one of the flutter plugins. With this, users can detect edges of objects, scan paper, and freely adjust the box area to crop the image.

4.3 Backend

The main challenge from backend part was finding a good way to save image, and also finding a good way to run YOLO model in server.

4.3.1 How to Save Images

To save image in database, the image should be converted into bytes. In this case, the image should be decoded every time whenever we show an image to the user. This process is unnecessary and time-consuming, so we decided to save images in azure blob storage. Based on azure-blob-storage library in python, we implemented some functions to download and upload image to our blob container. With this implementation, it was able to show image to the user quickly, presenting image directly from web without getting image data from the server.

4.3.2 How to Run YOLO in Server

After checking our YOLO model is working in our web server with CPU, we created a docker container containing the model - based on darknet image pre-build for cpu use. The container contains model, learned weight, and other essential libraries like tesseract and opencv.

We run code in docker by using system command in python. We retrieve the result and post-process the inference result from our web server, and save the processed result is saved in the database.

4.3.3 DB Schema

We applied third normal form(3NF) to our database schema. 3NF is a database schema design approach for relational databases. By applying 3NF in designing database schema phase, our database could reduce duplication of data, avoid data anomalies, ensure referential integrity.

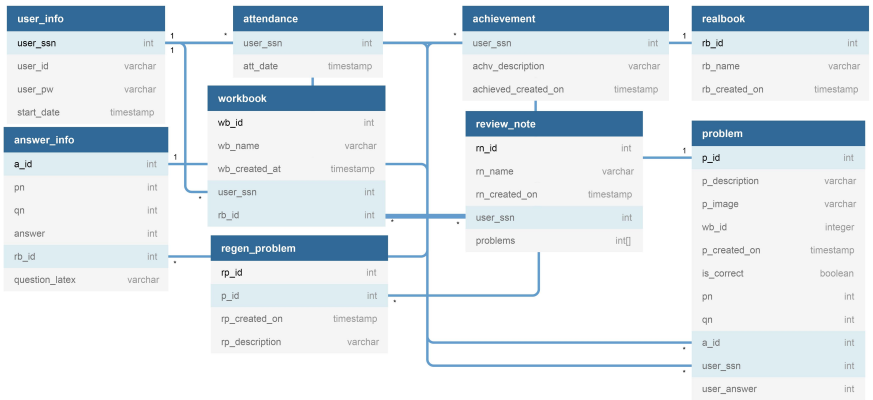


Fig. 3. Database schema

4.4 AI

YOLO You Only Look Once(YOLO), a Real-Time Object Detection model, was proposed by Joseph Redmon in 2016[1]. Prior to the publication of this paper, most of the deep learning-based object detection systems were just a modification of a classification model. The most popular model at that time was faster R-CNN[2], which proposes a place with a high probability of a bounding box and performs classification by putting the proposed box area as a classifier. R-CNN had three stages, 1) region proposal 2) classification, 3) box regression, and has a complex pipeline because the three stages must be individually trained. It was difficult to optimize a model due to such a complex pipeline and it had a very long inference time.

Contrary to this, YOLO handles finding bounding boxes and classifying them in one step. Through this special structure, YOLO is more powerful and faster than previous methods, and has a speed enough to even perform real-time object detection.

Because of these reasons, with a very fast speed and also great accuracy, YOLO is very famouse in this time.

5 YOLO related Implementation

5.1 Training Model

YOLOv4 is a complex model and therefore it has various hyperparameters. We customized hyperparameters based on the initial model developers:

batch	64	channels	3	subdivisions	16	max_batches	16000
width	416	height	416	angle	0	scales	.1, .1
momentum	0.949	decay	0.0005	learning_rate	0.001		
saturation	1.5	burn_in	1000	exposure	1.5	hue	.1
policy	steps	steps	12800, 14400				

Training was performed fine-tuning through our dataset based on pre-trained weights (yolov4.conv.137). All were conducted through the gpu server provided by google colab, and it was terminated after proceeding until learning was no longer progressing with a learning amount of about 300 epochs. We achieved 99% accuracy for validation mAP@50.

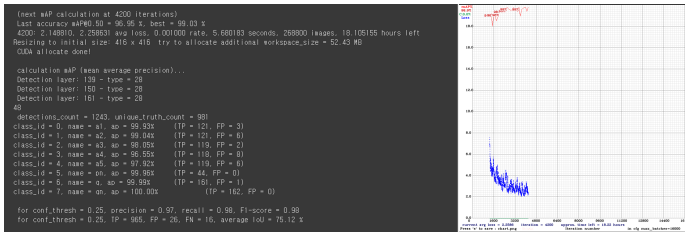


Fig. 4. Loss and accuracy in training

5.2 Dataset

Since there is no open dataset that perfectly suited our purpose, we created our own dataset. We used 8 books of "수능특강/수능완성", which EBS provides free pdf[4]. LabelImg was used as the laebling tool[5]. There was total 8 labels q, qn, a1, a2, a3, a4, a5, pn, and the meaning of each label was as following.

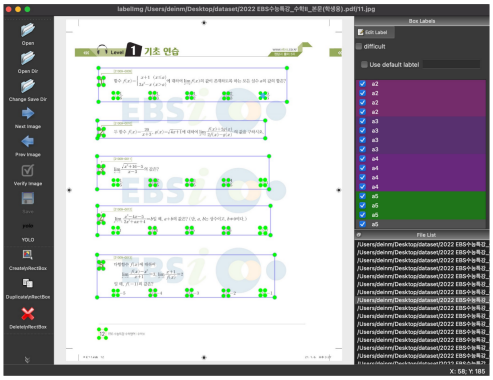


Fig. 5. labeling sample

After labeling 374 pages, we have done data augmentation to increase train set and to create a situation similar to the actual photographed image. Finally we created a dataset with a total of 906/51/26 train/validation/test images.

5.3 How to Detect Problem / Answer Area

Our model first starts by recognizing the problem area. When user takes a picture after solving a problem, first find the area occupied by each problem in the picture. In addition, by recognizing the question number and each circular number in the corresponding area, it is possible to determine whether the corresponding question is multiple choice/subjective, and reveal which answer was selected.
We tried to solve everything from recognizing problem areas to obtaining information about each problem through a single YOLO model.

5.4 How to Auto grading

The YOLOv4-darknet version we used outputs the object detection result in json format. The example of the json is as follows.


```
"filename":"test/test.jpg",
"objects": [
 {"class_id":7, "name":"qn", "relative_coordinates":{"center_x":0.101882, "center_y":0.467192, "width":0.019463, "height":0.018886}, "confidence":0.985022},
 {"class_id":7, "name":"qn", "relative_coordinates":{"center_x":0.110292, "center_y":0.756908, "width":0.027203, "height":0.022187}, "confidence":0.984115},
 {"class_id":7, "name":"qn", "relative_coordinates":{"center_x":0.101641, "center_y":0.608429, "width":0.019572, "height":0.019421}, "confidence":0.978645},
 {"class_id":7, "name":"qn", "relative_coordinates":{"center_x":0.101731, "center_y":0.187799, "width":0.019836, "height":0.019675}, "confidence":0.969553},
 {"class_id":7, "name":"qn", "relative_coordinates":{"center_x":0.102578, "center_y":0.324932, "width":0.018519, "height":0.021181}, "confidence":0.968845},
 {"class_id":6, "name":"q", "relative_coordinates":{"center_x":0.427016, "center_y":0.757734, "width":0.795912, "height":0.102917}, "confidence":0.997987},
```


Fig. 6. The result of yolo predict

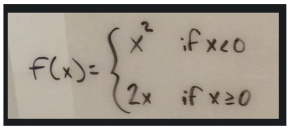
'classid' and 'name' show what kind of object the region is, and 'relative coordinates' indicates the relative position of the region, (center of x,y, width, height) Based on this information, we got the information value consisting of page number - question number - user's answer. The procedure is as follows.

1. OCR (tesseract) the page number and question number area to get the value
2. Determine whether objects other than pn belong to each question area.
3. After saving the recognized answers, the unrecognized number is determined as the user's answer.
4. If all numbers are recognized, the number with the darkest brightness among them is determined as the user's answer
5. Scoring proceeds by comparing the answers stored in the DB

5.5 How to Regenerate Math Problems

Regenerating math problem is one of our core functions. Since we thought training an equation OCR model was not the main object for our project, we decided to use API to implement this function.

Detect text and equation from image




```
{
 "auto_rotate_confidence": 0.0046418966250847404,
 "auto_rotate_degrees": 0,
 "confidence": 0.9849104881286621,
 "confidence_rate": 0.9849104881286621,
 "is_handwritten": true,
 "is_printed": false,
 "latex_styled": "f(x)=\\left\\{\\begin{array}{ll}x^{2} & \\text { if } x<0 \\\\\\n2 x & \\text { if } x \\geq 0\n\\end{array}\\right.",
 "request_id": "bd02af63ef187492c085331c60151d98",
 "text": "$f(x)=\\left\\{\\begin{array}{ll}x^{2} & \\text { if } x<0 \\\\ 2 x & \\text { if } x \\geq 0\\end{array}\\right.$"
}
```


Fig. 7. Input image and API response of Mathpix API

Before regenerating the math problem, we should detect the text and equation in the problem image. We used OCR API, Mathpix API, which is specialized in detecting equations. The API gives response containing the text in latex format from the image. We thought many students will use our application with same workbooks, so we decided to save the detected latex string into database to import efficiency. We saved API result in database, and get that value again whenever user wanted to regenerate the problem.

Detect variable from latex string

When user trys to regenerate the problem, we detect $ sign and get the numbers inside of it. Then we randomly regenerate the variable, comparing two variables at each time, and considering which number is bigger. Since the output is still in latex string format, the api server returns the result in html format. We used katex to render the latex string. Since some problems may not make sense if we just randomly change the numbers, we assumed that the input is a simple calculation question, and we decided to also provide random shuffle function.

5.6 How to Shuffle answers

1. Crop answer options
   Based on YOLO result, crop answer options. Had implemented some algorithms to crop options Even some of the options are not detected by YOLO. Save image in blob.
2. Shuffle options and grading
   From frontend side, shuffle the options and Show the options to the user. Grade the answer by getting the correct answer from API.

5.7 Open sources adopted

Mathpix API

We converted problem image into latex string to regenerate the question, useing Mathpix API. Mathpix API is an OCR API which is specialized in recognizing equations. It even show a good performance for printed Korean sentences.

azure-storage-blob library

To easily upload and download image from azure blob storage, we used python library which supports I/O of azure blob storage. Based on this library, we developed some functions to convert image into binary data and upload the binary image data into blob.

Embed webview within the application

To embed the Android's webview within the Flutter app, we utilized a flutter plugin called 'webview_flutter 3.0.0'. This plugin provides a WebView widget. Using this, we showed the webpage of regenerated questions. Also, we added floating action button on the right bottom of the screen which reloads the webpage when clicked, so users can regenerate the questions as many times as they want.

6 Limitations and Discussions

Answer information in the database

Through discussion, we concluded that extracting answers from the answer sheet is difficult. So we decided to save answers in database in advance. But by this implementation, we could make users only take pictures of the problem page. Also, since there are few main famous workbooks that student use for studying, maybe this might be even practical for some point of view. We don't have to run model again and again to grade the same page because we save the answer data in database.

Accuracy of model

With good image with good conditions, our model worked well. But if there was a 3d-rolling on the image, the performance of the model and the result of

OCR was so bad. Also, if user writes something on the workbook, the question area was detected according to the user's handwriting rather than the printed text. So the question-question number matching was not working well. We think that we can solve those problems by training model with additional datasets - 1) images with 3d rolling augmentation and 2) images which has handwritten letters on it.

Post-processing Algorithms

We implemented some post-processing algorithms to make up some errors caused by our YOLO model. We had defined some rule based algorithms to detect all 5 answer options even when model fails to detect the all options. We also implemented some algorithms to check the front part of each options to detect the answer that user have chosen - for the case when model fails to detect user's answer.

Of course, the best solution might be training a best model, but we think that it is possible to get better result by implementing an advanced post-processing algorithm. Because of time constraint, we couldn't consider all the various and extreme cases which can be handeled by post-processing algorithms.

7 Evaluation

7.1 Evaluation of model

In object detection problem, there are lot's of evaluation metrics.
However, since it is dependent on the dataset, it is usually used to compare the performance between models with one dataset (coco, etc.).

We used the most commonly used mAP@0.5 to measure the learning progress of the model. The average of the AP calculated for all classes is the mAP for object detection. The mAP calculated at IOU threshold 0.5 is designated as mAP@0.5. Finding the area under the precision-recall curve is the general definition of the Average Precision (AP). The precision recall curve represents the process of plotting the model's precision and recall as a function of the confidence threshold. Precision is a metric that determines how precise your predictions are. i.e. the percentage of correct predictions. Recall is a metric that measures how well you remember all of the good things. Precision and recall are calculated using the HM (Harmonic Mean) score.

7.2 Qualitatively evaluation

In the real object detection problem, it is necessary to qualitatively evaluate whether unseen data is well detected during actual testing.

Detecting question/answer

It can be seen that both types of problem books that we selected as targets recognize problem areas well.

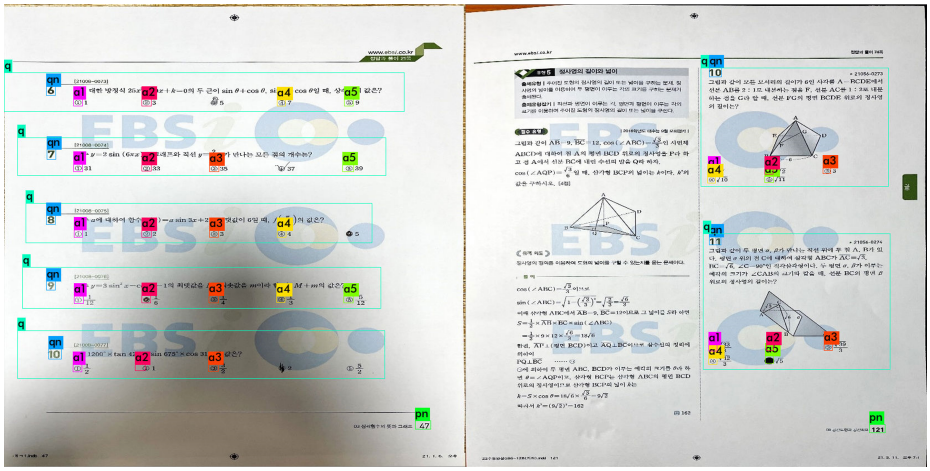


Fig. 8. detecting sample

```
Page :47 / Question : 6 / Useranswer :3
Page :47 / Question : 7 / Useranswer :4
Page :47 / Question : 8 / Useranswer :5
Page :47 / Question : 9 / Useranswer :2
Page :47 / Question : 10 / Useranswer :4
Page :121 / Question : 10 / Useranswer :2
Page :121 / Question : 11 / Useranswer :5
```

Fig. 9. Grading sample

Auto grading

It finds the answer checked by the user, and even if all five are recognized, the checked answer is found through post-processing.

Regenerating problems

For the basic calculation problems, our algorithm generated solvable regenerated questions.

Fig. 10. Regenerating and Shuffling sample

Shuffling problems The answer area is accurately divided, and the order of the answers is changed and displayed at random each time it is refreshed.

7.3 User experience testing

Regarding performance, we measured the time taken for getting the auto-graded results after the user takes a picture or selects image from gallery. On average, it took only about 6-7 seconds for delivering one image data to the model through the backend, processing in the model, and returning the result back to the user.

8 Conclusion

8.1 Lessons learned

Creating a service was accompanied by many difficulties. Especially, a lot of consideration was needed to make it easy for users who use this service for the first time to understand the flow of our service. Nevertheless, it was very rewarding to create service from which users can be benefit. We could be convinced that our project is meaningful because we were able to offer solution to user's discomfort.

In this course, students from different background gathered with a common interest in software technology. It was surprising how the common goals were gradually achieved, as team members used their own comparative advantage to meet the needs of the project. In a problematic situation, we were able to find a better option through exchange of opinions. Also, a lots of unexpected vulnerability was discovered by team members, making our service more robust. This project was a valuable opportunity for us to feel the value of collaboration in software development and to experience the methodology of collaboration.

References

1. YOLO Homepage, https://pjreddie.com/darknet/yolo/. Last accessed 4 Oct 2017
2. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards realtime object detection with region proposal networks," in International Conference on Neural Information Processing Systems, 2015, pp. 91–99.
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. https://www.ebsi.co.kr/ebs/pot/potg/beConnected2017BookDowonload.ebs
5. https://github.com/tzutalin/labelImg
6. LNCS Homepage, http://www.springer.com/lncs. Last accessed 4 Oct 2017