Detailed Design

Introduction:

A web application for handwriting exercises for children, with AI feedback based on an OCR model. Adults guiding the children on this journey can view their progress.

System Overview:

The foundation of the system will consist of the following features:

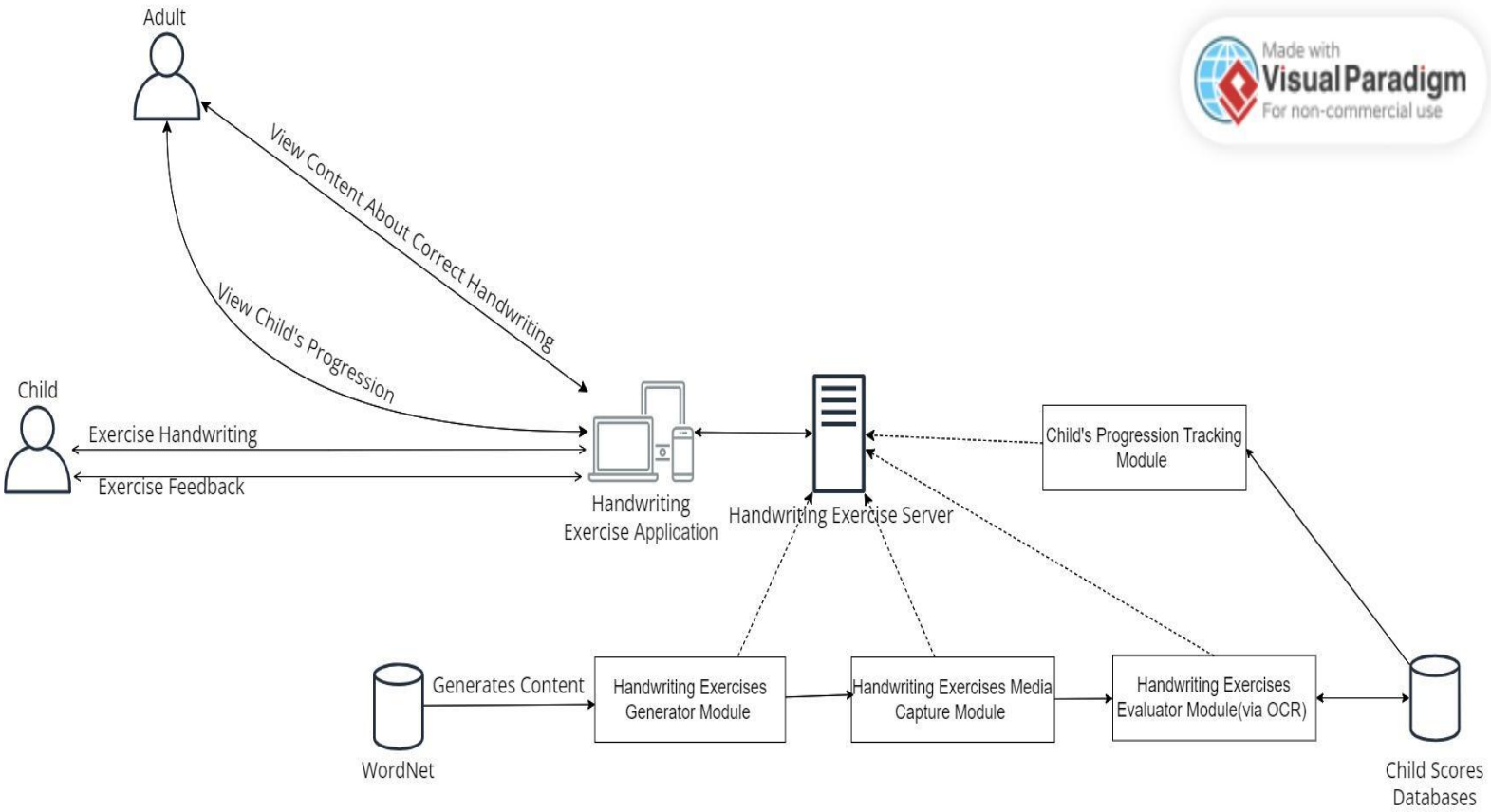
- Flexible AI-based feedback on the writings, partly based on the OCR's analysis.
- A set of tools to capture the child's writings, such as the device camera or gallery.
- A database synchronized with changes according to the child's scores, helping to keep track of the child's current handwriting level.
- Exercises will be provided based on the child's current level, thus providing varied and gradual experience.
- Analytics provided to the adult will be displayed by interactive graphs that will show each child's progression and performance over time, allowing for more focused face-to-face exercising sessions between the adults and children if needed.

Design considerations:

Due to the project time and resource limitations, we will make the following considerations:

- The OCR's model accuracy: Due to computing and financial limitations, we will use an open-source OCR model called paddleOCR, instead of the more accurate analysis offered by paid API calls to popular models. In addition, our ability to train the model is limited by the free GPU offerings in platforms like Kaggle and Google Collab and the lack of free qualitative handwritten text datasets.
- Language supportability: Due to limited time and resources, we won't be able to train PaddleOCR's model on handwritten text in every language it supports. This may result in less accurate recognition of languages other than English, even if we add support for them.
- Performance: Due to limited resources, we won't be able to deploy such an app that can maintain many users simultaneously, it will likely maintain up to 20 users simultaneously.

System Architecture:



Component design:

1. User management component
    - Adults accounts login/registration and authentication
    - Linking childrens to an adult account.

2. Handwriting exercise component
    - Generates handwriting exercises based on the child’s progression.
    - Retrieves words by category from WordNet.

3. Handwriting media handling component
    - Handles upload and temporary storage of the handwriting media.
    - Integrates the device’s camera for capturing the child’s handwriting.

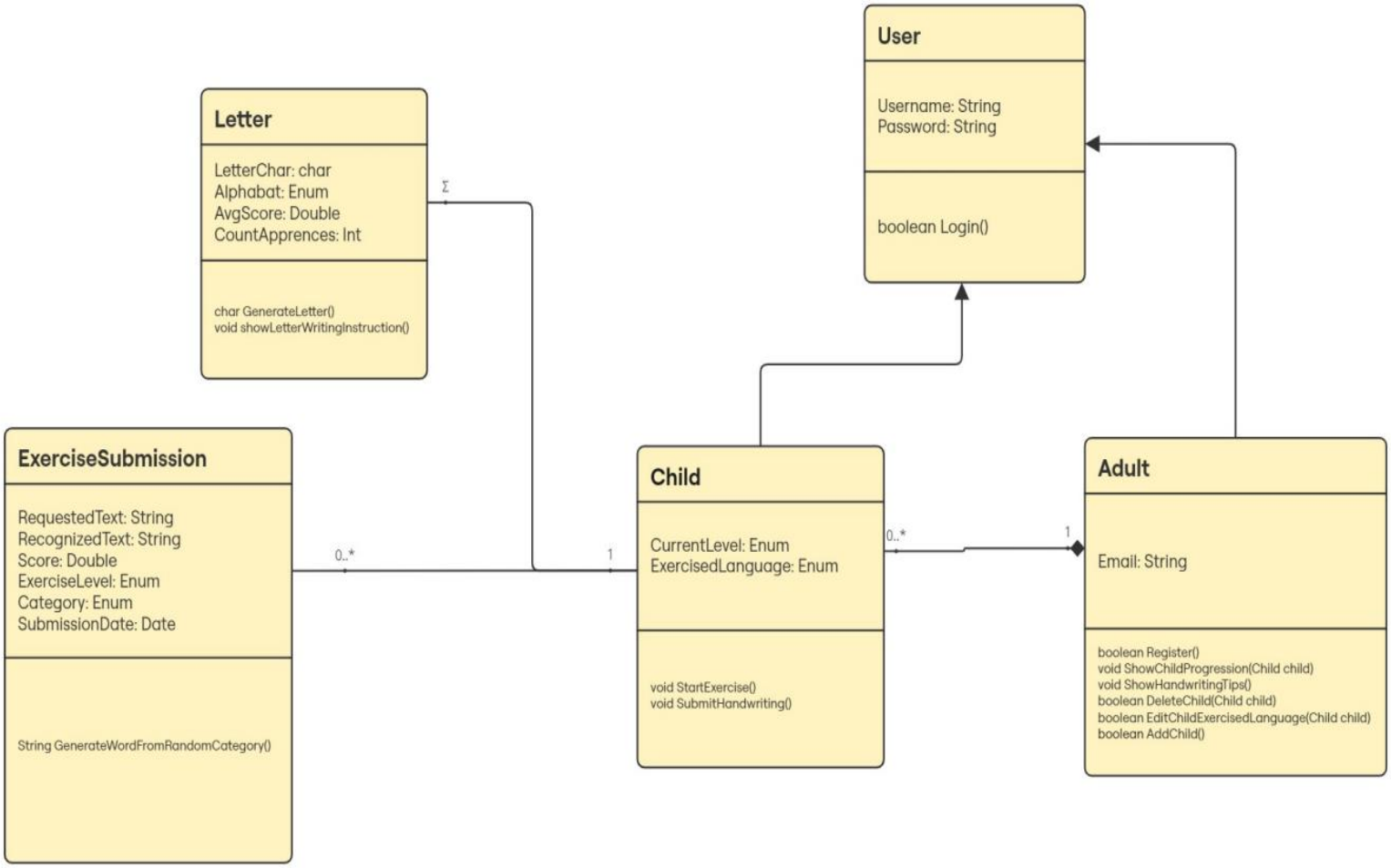
4. OCR evaluation component
    - Process handwriting images by an OCR model to evaluate accuracy and provide feedback.
    - Sends the results to the child progression tracker and the database components.

5. Child progression tracker component
    - Calculate child progression scores.
    - Visualizes scores data.

6. Educational content component
    - Store tips, tutorials, and educational content on children’s handwriting for adult accounts.

7. Database component
    - Stores all the system's data, including user info, child scores, and tips education content.

Data design:


User
Username: String
Password: String
boolean Login()
Letter
LetterChar: char
Alphabat: Enum
AvgScore: Double
CountApprences: Int
char GenerateLetter()
void showLetterWritingInstruction()
Σ
ExerciseSubmission
RequestedText: String
RecognizedText: String
Score: Double
ExerciseLevel: Enum
Category: Enum
SubmissionDate: Date
String GenerateWordFromRandomCategory()
0..*
1
Child
CurrentLevel: Enum
ExercisedLanguage: Enum
void StartExercise()
void SubmitHandwriting()
0..*
1
Adult
Email: String
boolean Register()
void ShowChildProgression(Child child)
void ShowHandwritingTips()
boolean DeleteChild(Child child)
boolean EditChildExercisedLanguage(Child child)
boolean AddChild()

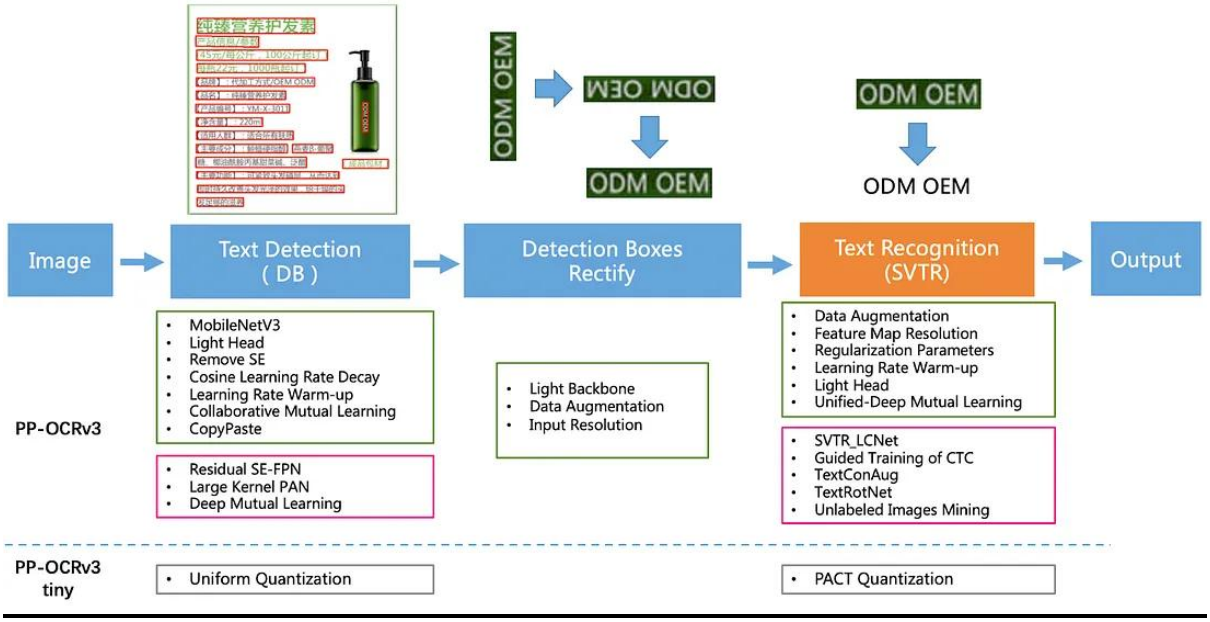
Algorithms Description:

1. PaddleOCR's detection and recognition:
    - PaddleOCR receives the handwritten image and processes it through 3 separate steps, where the goal is to reach the recognized text from the photo.

    A. The text detection step involves preprocessing the image to fit the model requirements and segmenting it into separate "boxes" of text.

    B. The detection boxes rectify step involves rotating each box to fit the correct reading direction in the selected language.

    C. The text recognition step: The trained recognition model interprets the written content of each detected box, makes predictions, and determines the confidence score of each box prediction. We have adjusted the open-source library to get the model's confidence scores per character, allowing us to give more detailed feedback to both the child and the adult on the handwriting.



OCR Algorithm Time Complexity:

1. **Text detection:**

   DBNet algorithm (Differentiable Binarization), splitted into 3 parts:

   1.1 Backbone network for extraction of image features- ResNet50

   O(W * H * C * K^2 * 50) where:

   W, H: width and height of the input image

   C: Number of input channels (3 for RGB)

   K^2: kernel size (for example, 3x3)

   50: number of layers in ResNet50

   1.2 FPN (Feature Pyramid Network) for feature enhancement

   O(W * H * D) where:

   W, H: width and height of the feature map at each pyramid level.

   D: Number of the feature channels

   1.3 Head network for calculation of the probability map of the text region

   O(W' * H' * D) where:

   W', H': width and height of the downsampled feature map.

   D: Number of the feature channels

2. **Text recognition:**

   SVTR_LCnet algorithm, splitted into 3 parts:

1. **2.1** Lightweight CNN Backbone (**LCNet**): used to extract features from the input text images

   O(W * H * C * K^2 * L) where:

   W, H: width and height of the input image

   C: Number of input channels (3 for RGB)

   K^2: kernel size

   L: number of layers convolutional layers

2. **2.2** Scalable Vision Transformer (**SVTR**): A transformer-based architecture that processes the sequential features produced by the LCNet.

   2.2.1 Self-attention model

   O(T^2 * D) where:

   T: sequence length

   D: hidden dimension

   2.2.2 Feed-Forward Network (FFN)

   O(T * D^2)

   T, D - (same as 2.1)

   Overall: O((T^2 * D) + (T * D^2))

3. **CTC** Decoder: Decodes the output of the SVTR into character predictions.

   O(T * C) where:

   T: sequence length

   C: number of possible output characters

2. **Adaptive handwriting learning**

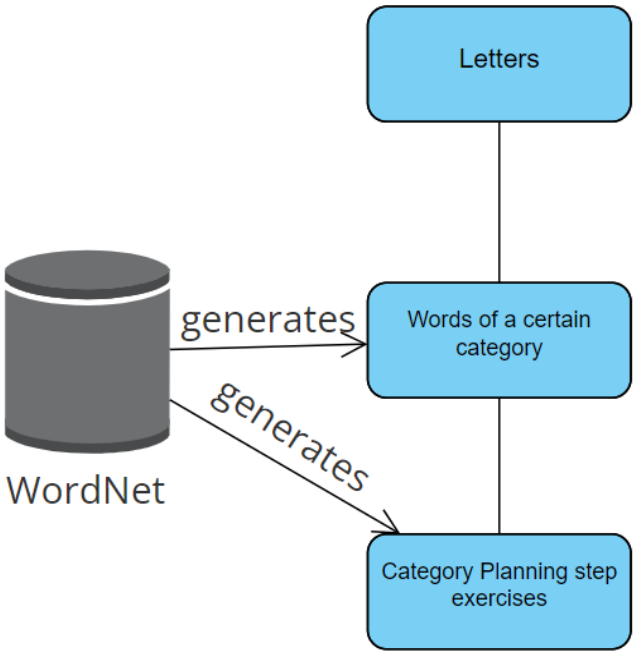
When tracking the learner's data, we can offer him personalized exercises that fit his current level based on his past performance and help him improve weaker points that were identified by suggesting more exercises on such subjects.
To offer such adaptive behavior we will consider these aspects:

A. The current level of the child on each letter of his alphabet: for each time the child exercises and correctly writes a letter that he has been requested to write (based on the model) - his confidence score will be calculated to the letter's score in his saved data. For example, if the child fails to write the letter 'a' - we will lower the score of that misrecognized letter. A low level on a certain letter will amount to a higher chance of getting an exercise on that letter, and a certain score on every letter will have to be reached until the child is offered a random word of chosen categories.

B. Once the child's score has passed a certain score on each letter, he will be offered exercises to write certain words, each letter of the word will contribute to the letter score, and the confidence score of the word itself will contribute to the score the child has on words.

C. Once the child's score on words has passed a certain score, he could be offered to write a word of that category (for example- "Name an animal"). According to our literature review, these exercises should help the child with the planning step of the handwriting process, which is the hardest step for those who struggle with handwriting.
We will keep track of the weighted confidence score in these practices to indicate if the child struggles with this step.

The current level of the child will be recalculated after each of his exercises following these principles, while the possible levels are: letters, words, and planning. When being in a higher level the child could be offered with exercises of lower levels in lower probability.



The algorithm's time complexity:

According to these steps, each time we will want to generate a new exercise for the child according to his level we will need to:

1. If his current level is letters, generating one of them is O(1)
2. If his current level is words, generating one of them is requiring:
    - generating a random category from a chosen set: O(1)
    - retrieving all synsets of that category from wordnet(the group of words that represent that category): O(1)
    - retrieving all the subcategories(called hyponyms) of these synsets - O(S) while S is the number of synsets retrieved in the last step.
    - choosing a random subcategory O(1)
    - retrieving all the words(which are called lemmas) from wordnet: O(L) while L is the number of lemmas in that hyponym.
    - choosing a random lemma from that group - O(1)
    - Total: O(S + L)
3. If his current level is categories, generating one will only require generating one of the chosen categories - O(1)

UI design:

1. Landing page:
   1.1 App logo and name.
   1.2 Login/register button.
2. Register Page for adults(parents/teachers etc..) : username, password and email fields.
3. Login Page for both children and adults: username and password fields.

Pages designed only for the children:

4. Handwriting practice page
   4.1 Instruction panel: GIF/video guiding proper letter writing that fits the given exercise.
   4.2 Exercise panel: displaying the letter/word/category for the child to write.
   4.3 Submission form: Accessible button to open the device camera/gallery.
   4.4 Feedback panel: Positive and simple feedback based on the OCR's evaluation.

Pages designed only for the guiding adults:

5. Children account manager page
   A grid layout of all the children that are connected to that adult, in each row:
   5.1 The child's username
   5.2 A button to delete the child's account.
   5.3 A form to change the child's exercising language
   5.4 A button to access that child's progress tracking page.
   5.5 An option to add a new child that will be connected to this adult account, by choosing his username and password.

6. Progress tracking page
   6.1 Child's performance chart in each section - Letters, words, planning.
   6.2 Success rates per each letter in the language, displayed as bar charts.
   6.3 Line chart showing success rate for each of the child's submissions indicating his improvement over time.

7. Content about correct handwriting
   A list of articles/references which contains:
   7.1 Handwriting Best Practices
   7.2 Handwriting Improvement Strategies