ECEN 5060 (Deep Learning) Final Project Proposal

Haya Monawwar, Sungjoo Chung

March 2025

1 Handwritten Mathematical Expression Recognition

Handwritten mathematical expressions (HMEs) are indispensable in various domains, such as engineering, education, and science. The development of pen-based or touch-based devices has provided a user-friendly interface to input handwritten mathematical expressions, which is more natural and convenient than editors such as Microsoft Equation Editor or LaTeX. Thus, such devices have been widely adopted in various environments, such as offices and educational institutions, especially during the outbreak of COVID-19. This phenomenon has necessitated the development of an accurate model for recognizing HMEs.

The handwritten mathematical expression recognition (HMER) problem differs from the traditional optical character recognition (OCR) problem due to three main reasons:

- The two-dimensional (2D) nature of HMEs adds additional complexity compared with the one-dimensional OCR problem. This can be demonstrated by the following mean squared error (MSE) function: $\frac{1}{N}\sum_{i=1}^{N}(p_i - t_i)^2$.
- The existence of more than 1,500 unique symbols [BFS17], which are often difficult to distinguish from each other (for example, "O", "o", "0", "$\mathbf{O}$", and "$\mathcal{O}$"), especially when considering the variation in handwriting styles. Combined with its structural nature, this leads to an infinite number of combinations of symbols and spatial relationships in MEs.
- The existence of long-term dependencies and correlations among symbols in MEs. For example, "(" and ")" are used to contain subexpressions, and if those subexpressions contain other long subexpressions, these dependencies are challenging to learn.

On a more personal note, as engineering students, we read, write, and share HMEs on a daily basis. Initially suggested by Haya, we thought that applying

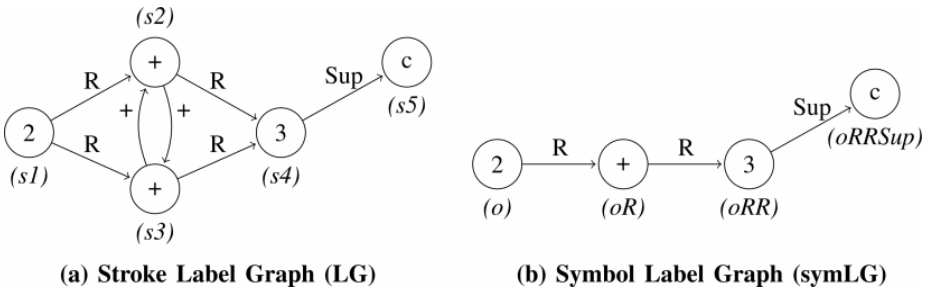

Figure 1: Different graph representation of "$2 + 3^c$" [Mah+19]

the concepts we have learned in the Deep Learning course to tackle the HMER problem would not only be interesting but also very relevant.

We will utilize the database provided for the 6th Competition on Recognition of Handwritten Mathematical Expression (CROHME) [Mah+19]. It consists of roughly 12,000 HMEs collected from previous CROHME competitions (2011, 2012, 2013, 2014, and 2016), with annotations in symbol layout graph (symLG) format. SymLGs is a structured representation of HMEs that captures both individual symbols and their spatial relationships and has been adopted as the default representation mode since 2019 CROHME due to the success of encoder-decoder-based systems. Fig. 1 illustrates the two different types of graph representations used in previous CROHME competitions. More details on these representations can be found in [Mah+19].

While the 2019 CROHME contains data in InkML format, which stores pen stroke sequences as a series of (x, y, timestamp) points, and greyscale image format, in this project, we will only focus on recognizing HMEs in the image format. Greyscale images in the 2019 CROHME dataset are of 1000×1000 pixels, with 5 pixels of padding. As the 2019 CHROME dataset consists of 8,836 and 1,199 images for training and testing, respectively, we believe that the CROHME dataset is sufficient to train a deep network. As the project progresses, we hope to extend our work to include the generation of the latex string file from the input image.

1.1 Deep Learning Network

For this task, we intend to use 2 models to perform a comparative study:

- **TrOCR (Transformer-based OCR)**: TrOCR employs a Vision Transformer (ViT) encoder and an autoregressive decoder, making it highly effective for structured handwriting recognition [Li+21].
- **CRNN (Convolutional Recurrent Neural Network)**: CRNN combines CNNs for feature extraction and bidirectional LSTMs for sequential text decoding, using a Connectionist Temporal Classification (CTC) loss function for alignment-free recognition [SBY16].

For starters, the models will be used in their standard implementation but will be fine-tuned along the way. The 2019 CROHME dataset is conveniently separated into training, validation, and testing datasets. Thus, models will be trained, fine-tuned, and evaluated on corresponding datasets.

1.2 Implementation Framework

Both models will be implemented using **PyTorch**, with TrOCR fine-tuned using the Hugging Face Transformers library and CRNN trained from scratch or initialized with pre-trained text OCR weights. CRNN will use a CNN backbone (ResNet or VGG) followed by a bi-directional LSTM (BiLSTM) decoder trained with CTC loss. A BiLSTM decoder trained with CTC loss is a common approach used in sequence-to-sequence tasks like OCR and speech recognition where input and output sequences do not have explicit alignments. Additional help will be taken from existing Kaggle competitions on the subject as well as existing publications to implement the frameworks in this project.

1.2.1 Evaluation Metrics

The performance of the network will be evaluated with the same metrics implemented in the 2019 CROHME which uses the Label Graph Evaluation Tools (LgEval) [ZMV13]. Evaluation metrics of LgEval include the expression recognition rate (ExpRate), the expression structure prediction rate (ESPR), recall, precision, and F1 scores. ExpRate measures the percentage of predicted mathematical expressions matching the ground truth by the following equation:

$$\text{ExpRate}_{\leq n} = \frac{N_{\text{correct},\leq n}}{N_{\text{total}}} \times 100 \tag{1}$$

where $\text{ExpRate}_{\leq n}$ indicates that the ExpRate is tolerable with at most n symbol-level errors, and $N_{\text{correct},\leq n}$ denote the number of correctly predicted expressions with at most n symbol-level errors, respectively. ESPR is calculated by the percent of MEs whose structure is recognized correctly irrespective of the symbol labels.

1.3 Project Schedule

The project is intended to follow the timeline below:

- **Week 1 / 18-25 March**: Literature review of existing works.
- **Week 2 / 26 - 1 April**: Pre-processing dataset (image resizing, augmentation).
- **Week 3 / 2 - 8 April**: Fine-tuning the CNN and TrOCR models on mathematical expressions to perform a comparative study.
- **Week 4 / 9 - 15 April**: Model training and hyperparameter optimization.

- **Week 5 / 16 - 22 April** : Further testing on potentially unseen handwritten formulas.
- **Week 6 / 23 - 29 April** : Model improvement and ablation studies.
- **Week 7 / 30 - 6 May**: Preparing final results and presentation, submitting the final report, and updating the project GitHub.

References

[ZMV13] Richard Zanibbi, Harold Mouchère, and Christian Viard-Gaudin. "DRR - Evaluating structural pattern recognition for handwritten math via primitive label graphs". In: *SPIE Proceedings*. Vol. 8658. SPIE, Feb. 4, 2013, pp. 865817–. DOI: 10.1117/12.2008409. URL: https://lens.org/034-750-771-175-279.

[SBY16] Baoguang Shi, Xiang Bai, and Cong Yao. "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.11 (2016), pp. 2298–2304.

[BFS17] Barbara Beeton, Asmus Freytag, and Murray Sargent. *Unicode support for mathematics*. The Unicode Consortium, Mountain View, CA, USA, Tech. Rep. 25. May 2017.

[Mah+19] Mahshad Mahdavi et al. "ICDAR 2019 CROHME + TFD: Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection". In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. 2019, pp. 1533–1538. DOI: 10.1109/ICDAR.2019.00247.

[Li+21] Minghao Li et al. "TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models". In: *arXiv preprint arXiv:2109.10282* (2021).