PROJECT PROPOSAL: DIGITAL IMAGE RECOGNITION OF HANDWRITTEN MATHEMATICAL EXPRESSIONS

Maeesha Biswas
Student# 1003931392
maeesha.biswas@mail.utoronto.ca

Meherin Shamim
Student# 1007158754
meherin.shamim@mail.utoronto.ca

Muhammad Ahmed
Student# 1007364825
mm.ahmed@mail.utoronto.ca

Tracy Sun
Student# 1006981438
tracy.sun@mail.utoronto.ca

1 INTRODUCTION

Mathematics is an integral part of an engineering student's life and with everything becoming digital, it is imperative to be able to identify handwritten digits, letters, mathematical symbols, and complex structure expressions and convert them into a digital format.

Our project aims to design such a model using Convolutional Neural Networks (CNN) which can recognize handwritten equations and output their equivalent in MathML or LaTeX. The LaTeX or MathML code can then be exported to digital notes or documents. Our project will also implement a simple web application with a user interface that accepts a digital input of a mathematical expression through a web based drawing canvas to demonstrate the explained functionality supported through existing open-source code (LiterallyCanvas, 2018).

Because of mathematics' vital role in all sciences, our model will help improve the application of math. It would assist the visually impaired to read handwritten math equations and would allow digitization of handwritten math documents (Shinde et al., 2017). Students would be able to share their handwritten notes in a neater format and could export them into digital notebooks. Old handwritten mathematical texts could be digitized for easier access.

Deep learning is a reasonable approach because compared to traditional computer vision techniques, neural networks in deep learning are not hard coded but are trained so they can take into account different writing styles when recognizing handwritten equations (O'Mahony et al., 2019). Also advances in computing power and memory capacity have improved the cost effectiveness and performance of deep learning techniques (O'Mahony et al., 2019). Compared to traditional computer vision, deep learning is more accurate in image classification and semantic segmentation (O'Mahony et al., 2019). And since our model involves classifying individual symbols in a mathematical equation, deep learning would be the reasonable approach.

1.1 PROJECT ILLUSTRATION

Project illustration can be seen in Figure 2 at the end of the report.

2 BACKGROUND & RELATED WORK

The recognition of handwritten text is generally achieved through using neural network-based systems, and most existing approaches are based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) (Li et al., 2021). RNNs are typically used to get high-performance results, and have won several handwriting recognition competitions (Voigtlaender et al., 2016). Conversely, the training time when using RNNs is normally long due to its recurrence feature (Doetsch et al., 2014). On the other hand, CNNs are efficient and reliable in image understanding, but their performance is slightly inferior to RNN based system (Chaudhary & Bali, 2022). However, since CNNs have high efficiency and relatively good performance on mathematical symbols and digits, they are commonly studied for tasks about handwritten math classification (Ramadhan et al., 2016).

Several studies were conducted to improve the CNN models for better performance. In 2019, Yousef et al. proposed a fully convolutional CNN architecture for text recognition (Yousef et al., 2020). This architecture makes extensive use of batch normalization and layer normalization, which allows it to achieve a high degree of regularization and effectiveness. This model has accomplished human-level performance on CAPTCHA, and won the International Conference on Frontiers of Handwriting Recognition (ICFHR) 2018 Competition on Automated Text Recognition.

In 2022, Chaudhary et al. designed an advanced version of the Efficient And Scalable TExt Recognizer (EASTER) which they published in 2020 (Chaudhary & Bali, 2020). The EASTER2.0 is a novel CNN-based architecture, and uses only a one-dimensional convolution (Chaudhary & Bali, 2022). They also proposed a new data augmentation technique called "Tiling and Corruption (T ACo)". T ACo augmentations refer to generating a new image by randomly covering up several areas on the input image, thus producing a larger training set. This augmentation makes EASTER2.0 capable of achieving high accuracy with limited training data.

There are also numerous studies on how to implement CNN specifically for better recognition of mathematical expression. In 2018, D'souza et al. proposed an offline handwritten mathematical expression recognizer (D'souza & Mascarenhas, 2018). Their model first obtains the grayscale version of input images, then the model is trained with SpNet architecture. After 100 epochs with a learning rate of 1, this model can accurately recognize 83 types of mathematical symbols (Ramadhan et al., 2016).

In 2019, Marques et al. introduced a model that can recognize handwritten polynomials (Marques et al., 2019). This model used Fractional Order Darwinian Particle Swarm Optimization (FODPSO) to separate components in the input image, then three CNN layers are used to identify each component individually in order. The first CNN layer classifies if that element is a number or symbol. Then, there is a CNN layer dedicated to number recognition, and a CNN layer for symbol recognition. The data is passed to the corresponding CNN layer after classification. After recognizing all the components on the input image, the results are merged together to produce the final output. The model achieves an average accuracy of 99% in recognizing polynomials with various handwriting styles (Marques et al., 2019).

In 2021, Sultan et al. developed a handwritten mathematical symbol classifier that can greatly improve the recognition of handwritten mathematical formulas (Sultan et al., 2021). Their model has a novel CNN architecture. It has 6 convolutional layers, after every two layers, there is a max pooling layer. This model uses different adaptive learning rates for each model parameter, thus achieving a high overall accuracy (Jepkoech et al., 2021). The average accuracy of the model is 99.19%, which is higher than most handwritten mathematical symbol classifiers (Sultan et al., 2021).

We are using a combination of handwritten and printed mathematical expression datasets so the final outcome of our project can handle and recognize both handwritten and printed mathematical expressions. However, there will be some data pre-processing completed to ensure data congruence.

3 Data Processing

3.1 Data Sources

The handwritten and printed math expression dataset will be split into training, validating and testing sets, to avoid overfitting. We will use cross-validation to assess the fit of our model (Brownlee, 2020).

3.1.1 Handwritten Dataset

The handwritten mathematical expression datasets have been sourced from the international Competition on Recognition of Online Handwritten Mathematical Expressions (CROHME) found here which take place at universities across the world (Mouchère et al., 2016). The competition's 2013 dataset combines datasets used at 3 international competitions (Mouchère et al., 2016).

The original data is in inkML format (an XML markup language), has over 10,000 expressions, and up to 101 unique symbols (Mouchère et al., 2016).

Figure 1: Sample handwritten mathematical expression sample from the CROHME Dataset

3.2 Characteristics of Data

Below are details on how we will ensure consistency in the dataset and what those characteristics are as they affect training and validation.

- Dark writing (like a pencil or pen) on a white background or dark printed typed text on white paper. All images will be in greyscale.
- All PNG image files that can be handled by the Matplotlib library and our CNN model.
- All image files will be rescaled to be the same dimension, and their margins will be removed (no peripheral whitespace).
- Ensure the encoding of the PNG files is base64 (we can do this at the time of converting the original format of the InkML in the CROHME dataset to image files)

3.3 Data Pre-processing

Group 10 will take the following steps to reach data congruenc prior to training our model.

- Convert inkML data to PNG using the Pandas library, an open source tool found here and encode the images using base64 (RobinXL, 2022).
- Crop margins from each image, and rescale all images to the same size.

4 Architecture

Deep learning can be applied for recognizing handwritten mathematical expressions. When identifying specific expressions and notations, a digital image of the handwritten characters can be used as an input for an Artificial Neural Network(ANN). When dealing with image inputs, it is common to use Convolutional Neural Networks(CNN), as they are often used for classification and tasks related to computer vision (Education, 2020).

A CNN is a supervised learning regime (Alzubaidi et al., 2021) and has three main components - a convolutional layer, pooling layer, and fully-connected layer (Education, 2020).

The convolutional layer is where most of the computation occurs; It assigns kernel weights through a collection of filters (Alzubaidi et al., 2021). Essentially this component creates feature maps, where an image is translated into numerical values, and then finds patterns with weighted values (Education, 2020).

The pooling layer simplifies these large feature maps into smaller ones (Alzubaidi et al., 2021). This is done to reduce complexity, increase efficiency, and limit risk of over-fitting (Education, 2020). There are three types of pooling layers - average pooling, maximum pooling, and global average pooling. Average pooling is when a filter moves through the pixels in the smaller feature map and calculates the average to carry out to the output array. Maximum pooling is when the filter returns the maximum value of the smaller feature map. Global average pooling is when the filter calculates the average of the large feature map altogether (Education, 2020).

In the fully-connected layer, classification of images occur through the connections made from all the previous convolutional and pooling layers. This part of the process is where the image is related to a specific label.

In this project, the neural network that will be created will have a similar architecture. It will have the main components of a CNN.

5 BASELINE MODEL

The open source handwritten multi-digit calculator on GitHub will be used as the baseline model of our project. Similar to our model, this model uses CNN to recognize handwritten math equations and convert them into digital expressions.

This model has 3 hidden layers. Two of which each have two convolutional layers and a dropout layer. The last hidden layer contains a fully connected layer and a dropout layer. The handwritten multi-digit calculator is trained with the MNIST dataset. After training with a batch size of 86, this model achieves a validation accuracy of 91.43% (PyMyCode, 2020).

To test the performance of our model, the same set of testing data will be input into both our model and this handwritten multi-digit calculator. The accuracy of our model should be no less than this baseline model.

6 ETHICAL CONSIDERATIONS

Since our model can digitize documents with mathematical notation, users can use it to digitize someone's handwritten notes or lectures and sell them or use them in personal work without their permission. The unauthorized and unintended uses of this technology such as this is one of the ethical considerations of our model. Also handwriting is unique (Srihari et al., 2002), so if in a future state of this project, if data of a user's input handwriting is stored, it could be linked back to and used as an identifying feature which gives rise to another ethical issue based on privacy. Our model may be limited by a specific class of expressions it can not identify or it may not be able to segment some expressions into individual symbols because of the complexity of handwriting (e.g. all the symbols joined together with no separation).

7 PROJECT PLAN

The platforms that will be used for collaboration are the following: Discord, Notion, Overleaf, and GitHub.

- Discord will be used to schedule meeting times through chat and meeting over voice call to discuss project deliverables.
- Notion will be used to record details discussed in the meeting as well as organizing deadlines and reminders.
- Overleaf will be used to write out reports in LaTex.
- GitHub will be used to collaboratively write code for the project. This way every change that is made is pushed to the repository and there is a way of retrieving previous code. This will ensure members do not overwrite each others code.

The following is a Link to the team's GitHub repository: `https://github.com/tsyh06/APS360project.git`

See Table 1 for information about how tasks for the Project Proposal were delegated along with their deadlines.

Table 1: Plan for divided tasks assigned to member for each deliverable. Names of members that are in bold are the Task Leads.

Project Deliverable	Project Task	Member(s)	Deadlines
Project Proposal	Introduction	Muhammed	2022-06-04
	Illustration/Figure	Maeesha	2022-06-04
	Background and Related Work	Tracy	2022-06-04
	Data Processing	Maeesha	2022-06-04
	Architecture	Meherin	2022-06-04
	Baseline Model	Tracy	2022-06-04
	Ethical Considerations	Muhammed	2022-06-04
	Project Plan	Meherin	2022-06-04
	Risk Register	**Maeesha***, Muhammed, Tracy, Meherin	2022-06-04
	Structure, Edits, Grammar, References	**Tracy***, Maeesha, Muhammed, Meherin	2022-06-05
Progress Report	Process of Cleaning Data	Maeesha, Muhammed, Tracy, Meherin	2022-07-08
	Brief Project Description	Muhammed	2022-07-09
	Individual Contributions and Responsibilities	Meherin	2022-07-09
	Data Processing (Written Portion)	Maeesha	2022-07-09
	Baseline Model	Tracy	2022-07-09
	Primary Model	Meherin	2022-07-09
	Structure, Edits, Grammar, References	**Maeesha***, Tracy, Muhammed, Meherin	2022-07-10
Project Presentation	Problem	Muhammed	2022-08-03
	Data	Maeesha	2022-08-03
	Data Processing	Tracy	2022-08-03
	Model	Meherin	2022-08-03
	Demonstration	Muhammed	2022-08-03
	Quantitative Results	Maeesha	2022-08-03
	Qualitative Results	Tracy	2022-08-03
	Takeaways	Meherin	2022-08-03
	Video Editing	**Muhammed***, Maeesha, Tracy, Meherin	2022-08-06
Final Report	Introduction	Muhammed	2022-08-10
	Illustration/Figure	Maeesha	2022-08-10
	Background and Related Work	Tracy	2022-08-10
	Data Processing	Maeesha	2022-08-10
	Architecture	Meherin	2022-08-10
	Baseline Model	Tracy	2022-08-10
	Quantitative Results	Meherin	2022-08-10
	Qualitative Results	Maeesha	2022-08-10
	Evaluate Model on New Data	Muhammed	2022-08-10
	Discussion	Meherin	2022-08-10
	Ethical Considerations	Muhammed	2022-08-10
	Project Difficulty/Quality	Tracy	2022-08-10
	Structure, Edits, Grammar, References	**Meherin***, Tracy, Muhammed, Maeesha	2022-08-13

Table 2: Modelling Project Risk Areas as functions of Probability and Impact

High Impact	1. Model Taking a long time to train (>3 days) 2. A team member dropping the course	
Low Impact	5. To contribute net new test data in a user interface, if a submitted image is not congruent with the testing methodoloy, or training image characteristics, the system may fail.	3. The model is limited in the type or level of math it recognizes and symbols recognized. 4. Large size of dataset might require time consuming workarounds to manage and train the model with.
	Low Probability	**High Probability**

8 Risk Register

We have considered likely risks and anticipated solutions beforehand to navigate them below (Table 2).

- 1. Model Taking a long time to train ($>$ 3 days)
  - Experiment with hyperparameters early in the learning process to get feedback on learning speed early.
- 2. A team member dropping the course
  - Approach: Knowledge Transfer with team member with team who dropped the course to ease the transition
  - Action: Redistribute the workload among fewer people and possibly change project expectations
- 3. The model is limited in the type or level of math it recognizes and symbols recognized. The dataset is capped at 10,000 expressions, and likely the number and types of expressions.
  - Ensure we are using a reasonable testing dataset size based on an accepted heuristic supported by machine learning literature to assess performance better.
  - Once we know the specific symbolic notation limitations of the model's recognition capabilities, mention them in the project's documentation. This will also pave the way for future areas of focus in the project.
  - Use other CROHME competition papers to support our observations.
- 4. Large size of dataset might require time consuming workarounds to manage pre-processing and post-processing of the the model.
  - Approach: Explore techniques to manage data with a Teaching Assistant for support.
  - Technical Solution: Consider a data augmentation technique to achieve good model training with smaller amount of data (Chaudhary & Bali, 2022)
- 5. To contribute net new test data in a user interface, if a submitted image is not congruent with the testing methodology, or training image characteristics, the system may fail.
  - Allow users to enter a handwritten math expression/equation on a web based canvas that adheres to current image graphics formatting, including size, format, and encoding. To avoid this issue altogether, We will not allow users to submit a file for expression recognition.

References

Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al Dujaili, Ye Duan, Omran Al Shamma, J Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning : Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8(1):1–74, 2021.

Jason Brownlee. A gentle introduction to k - fold cross - validation, Aug 2020. URL https://machinelearningmastery.com/k-fold-cross-validation/.

Kartik Chaudhary and Raghav Bali. Easter: Efficient and scalable text recognizer. *arXiv preprint arXiv : 2008.07839*, 2020.

Kartik Chaudhary and Raghav Bali. Easter2. 0: Improving convolutional models for handwritten text recognition. *arXiv preprint arXiv : 2205.14879*, 2022.

Patrick Doetsch, Michal Kozielski, and Hermann Ney. Fast and robust training of recurrent neural networks for offline handwriting recognition. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pp. 279–284, 2014. doi: 10.1109/ICFHR.2014.54.

Lyzandra D'souza and Maruska Mascarenhas. Offline handwritten mathematical expression recognition using convolutional neural network. In *2018 International Conference on Information , Communication, Engineering and Technology (ICICET)*, pp. 1–3, 2018. doi: 10.1109/ICICET.2018.8533789.

IBM Cloud Education. What are convolutional neural networks ?, Oct 2020. URL https://www.ibm.com/cloud/learn/convolutional-neural-networks.

Jennifer Jepkoech, David Muchangi Mugo, Benson K Kenduiywo, and Edna Chebet Too. The effect of adaptive learning rate on the accuracy of neural networks. *International Journal of Advanced Computer Science and Applications*, 12(8), 2021.

Minghao Li, Tengchao Lv, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. Trocr: Transformer - based optical character recognition with pre - trained models. *arXiv preprint arXiv : 2109.10282*, 2021.

LiterallyCanvas. Literally canvas, Oct 2018. URL http://literallycanvas.com/.

F.C.F. Marques, T.P. De Araujo, C.C. Nator, A.A. Saraiva, J.V.M. Sousa, A.M. Pinto, and R.T. Melo. Recognition of simple handwritten polynomials using segmentation with fractional calculus and convolutional neural networks. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pp. 245–250, 2019. doi: 10.1109/BRACIS.2019.00051.

Harold Mouchère, Christian Viard Gaudin, Richard Zanibbi, and U. Garain. Icfhr2016 crohme : Competition on recognition of online handwritten mathematical expressions. In *2016 15th International Conference on Frontiers in Handwriting Recognition(ICFHR)*, pp. 607 – 612, 2016. doi: 10.1109/ICFHR.2016.0116.

Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep learning vs.traditional computer vision. In *Science and information conference*, pp. 128–144. Springer, 2019.

PyMyCode. Building a handwritten multi - digit calculator, Dec 2020. URL https://towardsdatascience.com/building-a-handwritten-multi-digit-calculator-f03cf5028052.

Irwansyah Ramadhan, Bedy Purnama, and Said Al Faraby. Convolutional neural networks applied to handwritten mathematical symbols classification. In *2016 4th International Conference on Information and Communication Technology (ICoICT)*, pp. 1–4, 2016. doi: 10.1109/ICoICT.2016.7571941.

RobinXL. Convert inkml to image, April 2022. URL https://github.com/RobinXL/inkml2img.

Sagar Shinde, RB Waghulade, and DS Bormane. A new neural network based algorithm for identifying handwritten mathematical equations. In *2017 International Conference on Trends in Electronics and Informatics(ICEI)*, pp. 204–209. IEEE, 2017.

Sargur N Srihari, Sung Hyuk Cha, Hina Arora, and Sangjik Lee. Individuality of handwriting. *Journal of forensic sciences*, 47(4):856–872, 2002.

Rafi Ibn Sultan, Md. Nahid Hasan, and Mohammad Kasedullah. Recognition of basic handwritten math symbols using convolutional neural network with data augmentation. In *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pp. 1–6, 2021. doi: 10.1109/ICEEICT53905.2021.9667794.

Paul Voigtlaender, Patrick Doetsch, and Hermann Ney. Handwriting recognition with large multidimensional long short - term memory recurrent neural networks. In *2016 15th International Conference on Frontiers in Handwriting Recognition(ICFHR)*, pp. 228 – 233, 2016. doi: 10.1109/ICFHR.2016.0052.

Mohamed Yousef, Khaled F Hussain, and Usama S Mohammed. Accurate, data - efficient, unconstrained text recognition with convolutional neural networks. *Pattern Recognition*, 108:107482, 2020.

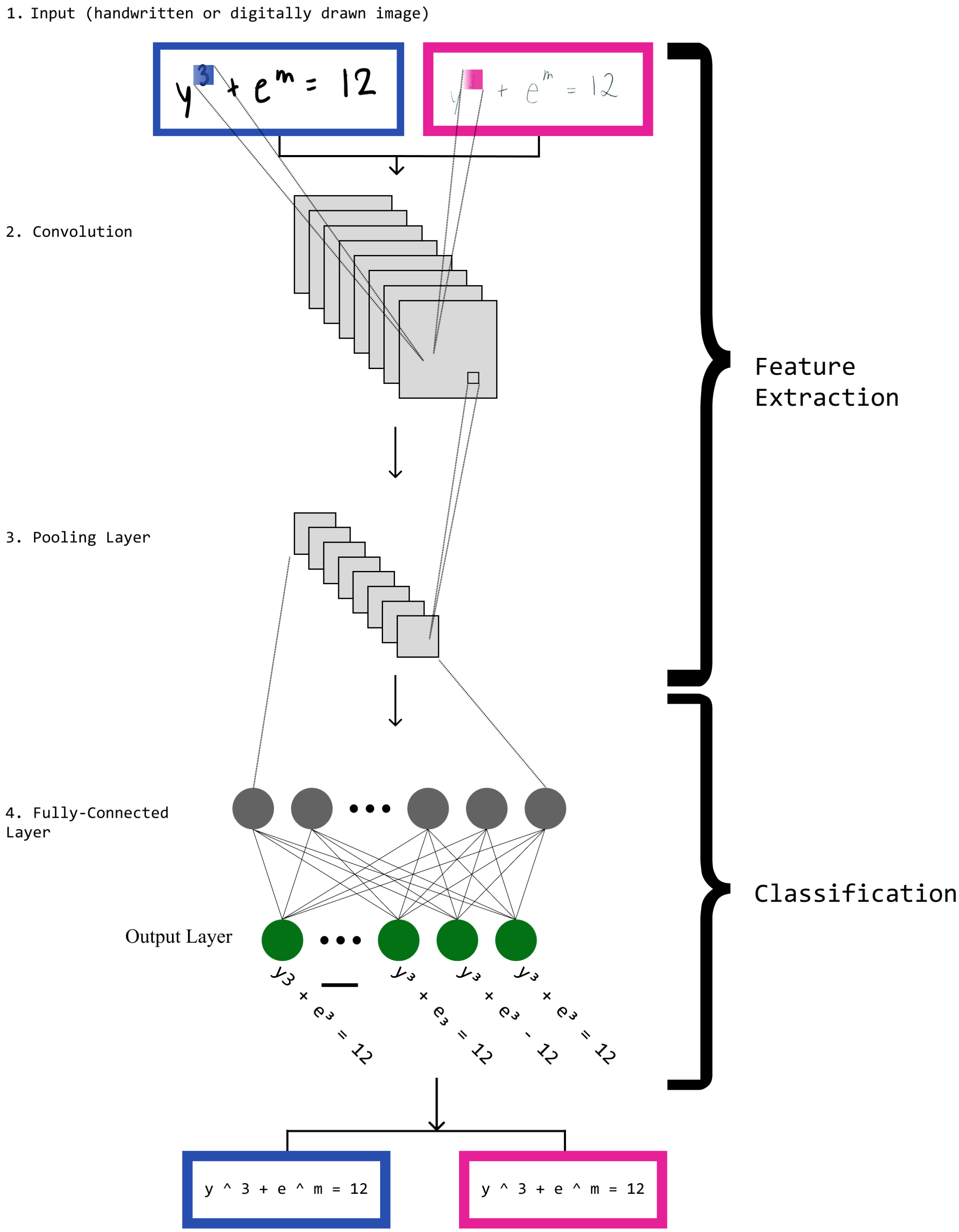

Figure 2: Illustration of Digital Image Recognition of Mathematical Expressions CNN Project