<https://link.springer.com/search?query=Handwritten%20mathematical%20expression%20recognition&facet-discipline=%22Computer%20Science%22>

Questions to Answer:

Background

* What is the problem/question that you will be investigating?

We would like to improve existing methods of optical character recognition (OCR)

Can we use computer vision to automate the grading process for math problems?

* Why is the problem interesting/important?

Students benefit greatly from receiving feedback but teachers and professors are limited in the amount of feedback they can give because grading is so time-consuming.

<https://www.edweek.org/teaching-learning/how-teachers-spend-their-time-a-breakdown/2022/04>

According to the Merrimack College Teacher Survey conducted in 2022 by the nonprofit, nonpartisan EdWeek Research Center, teachers spend a median of 5 hours a week grading.

* What are the most relevant readings (2-4 papers)?
  + <https://www.semanticscholar.org/paper/Deep-Columnar-Convolutional-Neural-Network-Majumdar-Jain/74e5317af9eea8dcf7e197e5b6c58d39c3654bdd>
  + A Tutorial on Optical Character Recognition in the Mathematical Domain

<https://www.researchgate.net/publication/333539203_A_Tutorial_on_Optical_Character_Recognition_in_Mathematical_Domain#full-text>

* + <https://arxiv.org/pdf/2308.05820.pdf>

Method

* What data will you use?
  + MNIST number dataset
  + AIDA Calculus Math Handwriting Dataset
* What are the existing methods? Are their implementations available?
  + <https://towardsdatascience.com/computer-vision-auto-grading-handwritten-mathematical-answersheets-8974744f72dd>
    - Uses a deep columnar convolutional neural network (DCCNN), an architecture that has high performance on image classification challenges such as the MNIST dataset.
  + <https://www.researchgate.net/publication/333539203_A_Tutorial_on_Optical_Character_Recognition_in_Mathematical_Domain#full-text>
    - Uses a Recursive Neural Network with Connectionist Temporal Classification. Also has a solution architecture consisting of 1) image preprocessing, 2) character segmentation, and 3) character recognition.
    - We can use openCV’s [text\_segment](https://github.com/divyaprabha123/Autograding-handwritten-mathematical-worksheets/blob/d074738c2e1db6b4619e9930f576439b309d1f37/utils_functions.py#L636-L725) function
  + <https://mathpix.com/ocr> = company providing an API with OCR for STEM.
  + <https://arxiv.org/pdf/2308.05820.pdf> = just for vertical addition and subtraction
  + Object detection: <https://arxiv.org/abs/2207.02696>
* What method or algorithm will you use, and why? What motivates your choice of this approach?

1) object detection to segment the characters. We will use an existing model such as YOLOv7, which was used in [this paper](https://arxiv.org/pdf/2308.05820.pdf) and has both high speed and accuracy on the MS COCO dataset.

2) character recognition using a DCCNN architecture, which has had high performance on image classification challenges such as those on the MNIST and CIFAR-10 datasets.

* What computing resources will you use to train and run your model(s)?

We will be running the model as a Jupyter Notebook on Colab using the T4 GPU to train and evaluate the model. We will connect to the datasets we use via their provided API.

* How will you evaluate your results?

The loss function used by the YOLO.v7 object detection model is the bounding box regression which will be used to update the weights during training.

During training, the results of the transcriptor will be evaluated by computing the mean squared error between the outputted LateX expression and its ground truth label where the difference is equal to the number of different characters.

Expected Results and Impact

* Qualitatively, what kind of results do you expect (e.g. plots or figures)?

The results of the object detection step will be the original image overlaid by colorful boxes indicating the bounds of a character.

The final output will be the LateX expression.

* Quantitatively, what kind of analysis (performance metrics, statistical tests etc.) will you use to evaluate and/or compare your results?

We will compute the average precision to evaluate the object detection algorithm. We will also output a précision-recall curve and calculate the area under the curve.

We will compute the expression recognition rate, which is the percentage of correctly recognized latex characters, to evaluate the transcriptor.

We will do a paired t-test to determine if the trained model performs significantly better on the test set than a model with randomly initialized weights.

* If your approach is successful, what difference will this contribution make?

Our mathematical expression recognition model, when combined with a grading software, can help automate the grading process for math exams and relieve the stress of teachers. Additionally, it uses a combination of state-of-the-art methods, including YOLO.v7 and a DCCNN architecture, that has the potential to achieve higher speed and accuracy than previous models. Both of these improvements would be helpful for the grading process and other optical character recognition projects.

**Relevant Papers’ Methods**

* [Recognizing Handwritten Mathematical Expressions of Vertical addition and subtraction](https://arxiv.org/pdf/2308.05820.pdf)
  + Creates their own dataset pretty much
  + Object detection stage: evaluates multiple models in first step. In second step, eliminates bounding boxes with low confidence level.
  + Transcription stage: classification algorithm
  + Implementation:
    - Used k-fold cross validation
    - Used pre-trained weights at COCO dataset
    - Default parameters
    - Resized images to 320x320 pixels and trained for 300 epochs
    - Used Optuna to optimize parameters for second step of object detection in the post-processing stage using validation set
* [This article has some code/implementation specifics](https://vipul-gupta73921.medium.com/handwritten-equation-solver-using-convolutional-neural-network-a44acc0bd9f8)
* [Recognition and Solution for Handwritten Equation Using Convolutional Neural Network](https://www.researchgate.net/publication/331617158_Recognition_and_Solution_for_Handwritten_Equation_Using_Convolutional_Neural_Network?_tp=eyJjb250ZXh0Ijp7InBhZ2UiOiJwdWJsaWNhdGlvbiIsInByZXZpb3VzUGFnZSI6bnVsbH19)
  + Image pre-processing:
    - Transformed to gray-scale
    - Binarization
    - Low pass filtering to reduce noise
* [Another good article](https://cs231n.stanford.edu/reports/2015/pdfs/mohan_lu_cs231n-project-final.pdf) this paper is rlly cool, uses HMMs, Gibbs to see if an expression is likely
  + Uses distortion to increase size of dataset
  + Has good explanation for classifying symbols such as =
  + Character-level and expression-level classification
* [Syntax-Aware Network for Handwritten Mathematical Expression Recognition](https://arxiv.org/pdf/2203.01601.pdf)

**Other datasets used:**

* Just symbols: <https://www.kaggle.com/datasets/xainano/handwrittenmathsymbols>
* HME 100k: <https://github.com/Phymond/HME100K>
* CHROME dataset: <http://www.iapr-tc11.org/mediawiki/index.php/CROHME:_Competition_on_Recognition_of_Online_Handwritten_Mathematical_Expressions>

Implementations:

<https://github.com/OlehOnyshchak/OCR/blob/master/RecognitionModelTraining.ipynb>

* Data pre-processing