**Motivation**

- Typing up mathematical equations has become quite a necessity when submitting academic papers, writing and solving problem sets, presenting mathematical concepts, and much more.
- We’re looking to automate the process of converting handwritten expressions to LaTeX using modern Machine Learning techniques.

**Datasets**

- Our dataset includes the traces of 10,000 handwritten expressions mapped to the corresponding ground truth LaTeX expression (obtained from Kaggle).
- We split the data into 80% train, 10% dev, and 10% test. We use traces for CSeg, normalized pixel arrays for OCR, and traces with segmented characters for SA.

**Features**

- **Segmentation:** features come from data. SVM uses features including overlap, inverse horizontal, vertical, and centroids distance, and horizontal containment, which are the main ingredients for grouping traces.
- **Character Recognition:** SVM and NN use data based features include normalized flattened greyscale pixel size. We derived Histogram of Oriented Gradients (HOG) from the pixel array – improves image recognition by using overlapping local contrast normalization.

**Methods**

- **Character Segmentation (CSeg)**
  - **Input:** list of all traces for equation. **Output:** traces grouped by characters.
  - **Overlap:** merges overlapping strokes. All other strokes are separate.
  - **SVM:** uses a binary SVM with a linear Kernel, L1 regularization, and C value 50. Binary SVM loss function: \( L = \frac{1}{2} \sum_{x \in D} (y \cdot h(x) - t)^2 \)
  - **NN (binary classification):** with single hidden layer, softmax and sigmoid activation, and cross entropy loss. Input is a 32X32 flattened pixel array of a single or multiple strokes. Output is whether or not the image is a valid LaTeX symbol.
  - **Beam Search:** we use beam search to explore possible segmentations. We maximize the segmentation score, which is based on NN prediction (confidence of it being a valid character). \( \text{score} = \arg\max_{\theta \in \Theta} \sum_{g \in \Theta} \log P(g) \)

- **Character Recognition (OCR)**
  - **Input:** normalized pixel array. **Output:** character’s LaTeX representation.
  - **SVM:** uses multiclass SVM with a linear Kernel, L2 regularization.
  - **Multiclass SVM loss:** \( L = \frac{1}{2} \sum_{x \in D} \max_{y \in Y} f(y; x) - f(y; x) + \delta_{y; \hat{y}} + \frac{1}{C} \sum_{e \in \Theta} \text{regularization} \)
  - **NN (multiclass):** uses same internal structure as NN in CSeg. Output is the classification of the LaTeX symbol. \( CE(y, \hat{y}) = - \sum_{y \in Y} y \log \hat{y}_y \)
  - **Cross entropy loss:**
  - **CNN:** multiclass CNN with 5 hidden layers, ReLU activation function, and cross entropy loss.

**Experimental Results**

- **OCR Accuracy:**
  - **CNN:** 44% NA
  - **SVM:** 53% 53%
  - **Binary NN:** 64% 96%

- **SA Analysis Accuracy:**
  - **Baseline:** Test 14% 65%
  - **Heuristics:** Test 14% 65%

**Discussion**

- Most character recognition errors come from symbols that do not appear as frequently in dataset (e.g. \( \mu \)).
- Segmentation: NN beam search and the binary SVM achieve similar accuracy (64%). Beam search tends to group traces that should remain separate. Binary SVM tends to keep strokes separated.
- We are working on improving our Structural Analysis. We are struggling with how to accurately test the output and train models and are currently reporting accuracy by hand to preserve correctness.

**Future Work**

- **CSeg:** implement a combined model using NN beam search and SVM predictions to increase accuracy. Use Adaboost and multi-scale shape context features.
- **OCR:** Experiment with advanced OCR techniques such as Random Forests to help improve the accuracy of our CNN and SVM. Use data augmentation techniques to add examples of uncommon symbols.
- **SA:** We plan on finding or implementing a LaTeX parser that will enable us to train an SVM to aid our current model in correctly building square root, exponents, and other more complex equations.
- **E2E:** implement a CNN for an end to end approach (similar to Google’s Tesseract).

**References**