Painfree LaTeX with Optical Character Recognition and Machine Learning

Joseph Chang, Shrey Gupta, Andrew Zhang {chang100, shreyg19, azhang97} @ stanford.edu

Overview and Objective

Current OCR technologies have proven quite effective on natural language, but recognition of mathematical expressions remains quite difficult due to their structural complexity. This project aims to explore modifications on typical techniques in order to convert images of typeset equation to LaTeX source code.

Dataset and Processing

Wikipedia Equation Dataset: 6,111 equations with ground truth LaTeX extracted by spidering Wikipedia's *List of Mathematical Equations* article.

- Used as training inputs for entire pipeline
- Preprocessing: Alpha channel removed, binarized, cropped, rendered to PNG.

$$\mathbf{A} = \frac{\mu_0}{4\pi}\frac{\mathbf{m}\times\mathbf{r}}{|\mathbf{r}|^3}$$

```
A=\frac{\mu_0}{4\pi}
\frac{m\times r}...
```

InftyCDB-3B Dataset: 70,637 characters with ground truth character codes

- Used to train Neural Net for OCR

Related Work

M. Okamoto and B. Miao, "Recognition of Mathematical Expressions by Using the Layout Structure of Symbols," *International Conference on Document Analysis and Recognition*.

P. S. Liang, "Neural networks, nearest neighbors," in *CS221 Fall 2016 Lecture*.

K.-F. Chan and D.-Y. Yeung, "Mathematical expression recognition: a survey," *International Journal on Document Analysis and Recognition*, vol. 3, no. 1, pp. 3–15, 2000.

Pipeline and Methods

Math expression recognition fundamentally involves four tasks:

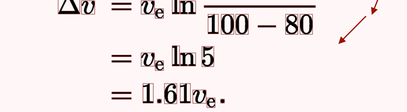
1. Character Segmentation
2. Character Recognition
3. Structural Analysis
4. Lexing/Parsing.

Inputs: Equation Images

$$\begin{aligned}\Delta v &= v_e \ln\frac{100}{100-80}\\ &= v_e \ln 5\\ &= 1.61 v_e.\end{aligned}$$

Character Segmentation (CSeg)

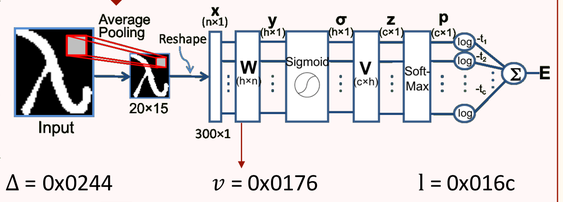
- Extract individual characters from equation
- Recursive Projection Profile Cuttings (RPPC) → recursively partitions by alternating vertical and horizontal separations.
- Post-processing handles disconnected chars (e.g. =, i) and slanted chars with subscripts



$$\begin{aligned}\Delta v &= v_e \ln\frac{100}{100-80}\\ &= v_e \ln 5\\ &= 1.61 v_e.\end{aligned}$$

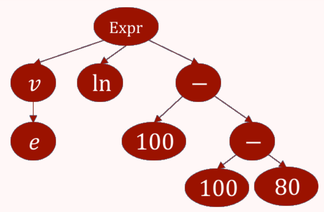
Character Recognition (OCR)

- Identity individual characters
- Neural Network: Processes pixels as image features and learns $\boldsymbol{W}$, $\boldsymbol{V}$ by solving backpropagation equations. (e.g. $\nabla_W E = \nabla_y E \cdot x^T + \lambda W$)



Structural Analysis (SA)

- Design expression structure tree
- Combination of rule-based system using layout data (e.g. "smaller characters above baseline are superscripts) and context-free grammars (e.g. "summation includes an index")



Lexing/Parsing

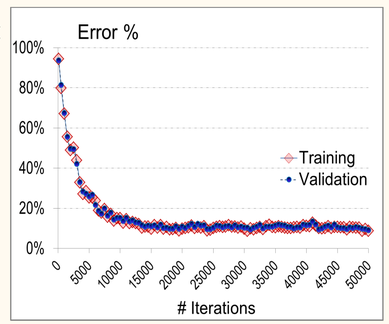
- Rule-based system for converting expression structure tree to LaTeX

Output: LaTeX Source

```
\Delta v = v_e \ln
\frac{100}{100-80}...
```

Results

- CSeg Performance: Achieves accuracy of **86%** (test on random sets of 50 training examples).
- OCR Performance: Attains 90% accuracy when using 60% of data for training, 20% for validation, 20% for testing.



Discussion

- CSeg works reasonably well using RPPC, but fails on cases like square roots that can't be vertically separated
- OCR is quite accurate, but gets confused on characters like l (letter) vs. 1 (number); perhaps contextual info may help
- SA tends to be a difficult problem due to ambiguities in expressions and variations on common grammars.

Future Directions

- Augment OCR with contextual feedback from CSeg (e.g. a character appearing next to two numbers is likely to be "1", not "l")
- For SA, apply a neural net approach for mapping small subexpressions to subtrees