

Images to LaTeX

Group 10

Aditya Chetan,	Brihi Joshi,	Siddharth Yadav,	Taejas Gupta
2016217	2016142	2016268	2016204

Introduction

Problem Statement - To convert the picture of a handwritten mathematical expression into its corresponding compilable LaTeX representation

Procedure -

- Handwritten Equations (Easy/Moderate)
 - Segmentation of Handwritten Equation
 - Classification of Symbols
 - Structural Analysis to determine the LaTeX equation
- Typefaced Equations (Difficult)
 - Seq2Seq Attention Models

Dataset

- Handwritten Equations - The Kaggle [Handwritten Mathematical Equations Dataset](#) has handwritten equations in **.inkml**
 - The Dataset is a part of the CROHME contest organised at ICFHR 2014.
 - For classification, a dataset of segmented symbols taken from this CROHME contest dataset has been put up on Kaggle [here](#)
 - The dataset is in the .inkml format, which contains the following information
 - The trace data of the handwritten equations - Each stroke made by the user is recorded with a separate trace ID.
 - If two traces belong to the same word, the two components are added to a trace group (only present in the Training Set.)
 - The trace groups are ordered - The exact order in which the user has written the equation can also be read
 - Since our task was to deduce handwritten equations in images, we have converted each .inkml file to its corresponding image.
 - Fairly nontrivial as .inkml format is difficult to interpret initially.

Intermediate Results

- Segmentation

- The first step in our pipeline was segmentation
- In order to do this we employed 2 different algorithms-
 - **Contour finding**
 - **Watershed algorithm**
- Segmentation did not work too well
 - Classes of mathematical symbols like \rightarrow sin, cos, tan, log, etc. were not getting properly segmented as they are often not written in a single stroke. Thus, inaccurate segmentation would affect the classifier at a later stage
 - Overlapping symbols: This led to multiple symbols getting segmented as a single symbol

- Heuristic Development

- 9 Rules were developed for Structural Analysis
 - These are in no way exhaustive
 - Not mentioned here due to lack of space

● Classification

* trained on a subset of dataset

- Segmented images now proceed to classification. The dataset used for training the classifiers the dataset of mathematical symbols released by CROHME. It has 376139 samples and 83 classes.
- As proposed we implemented 7 classifiers for classifying the symbols. We have recorded their initial performances (without hyper-parameter tuning or cross-validation):

<u>Classification</u>	<u>Precision</u>	<u>Recall</u>	<u>F1-Score</u>	<u>Accuracy</u>
Naive Bayes*	0.56	0.47	0.48	0.472
Logistic Regression*	0.60	0.63	0.60	0.628
Random Forest*	0.63	0.60	0.58	0.600
SVM*	0.92	0.33	0.39	0.32
MLP	not calculated yet (nc)	nc	nc	VAL - 0.93383
CNN	nc	nc	nc	VAL - 0.97427
Shallow-CNN	nc	nc	nc	VAL - 0.98158

Next Steps

- Thorough Analysis of the Classification Algorithms
 - Hyperparameter Tuning using Grid Search
 - Resistance to overfitting using Cross-Validation techniques
- Complete Heuristic Development and Coding
 - The current heuristics are sufficient (and that too not always) for very simple equations
 - They are also not yet implemented. We plan to refine and implement them before the final deadline.
- Explore Seq2seq models
 - All previous works point to using Seq2seq. This is because heuristics are not enough for handwritten data
 - We have also faced a lot of problems with heuristics as they are non-exhaustive and cannot cover most of the cases. This is also supported by past literature on this problem.
 - Since Seq2seq models are end-to-end and require a larger dataset, we would like to explore using them on type-faced [data released by OpenAI](#).