Hand-written Mathematical Formula Recognition Using Machine Learning

By: Chithiraikkayalvizhi Chinnusami

Advisor: Sion Yoon

WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Contents

- Motivation
- Purpose
- Problem Statement
- Background
- Related Work (Literature Review)
- Approach
- Data Collection
- Data Analysis
- Finding
- Conclusion
- Future Work

WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Motivation

- Still an unsolved research topic. Even though number and character recognition systems are very mature and have human like accuracies.
- Great improvement for existing note taking apps like OneNote and Evernote. Existing handwriting recognition works great for characters but struggles with math formulae.
- Personal Motivation
  - Try to solve a real-world research problem.
  - Learn Machine Learning and apply it on a real-world problem.
  - Learn Tensorflow

WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Purpose

- Solve the research problem of Mathematical formula recognition.
- Nice add-on feature to various note taking applications like OneNote and Evernote
- Extract formulae from various math related research and other documents where these formulae are in images.

WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Problem Statement

- Given a formula with its corresponding stroke data, the goal if this project is to develop a system that will identify the formula and output the result in a standard text format like Latex, MathML etc.
- Alternative input for the system could be an image representation of the formula.

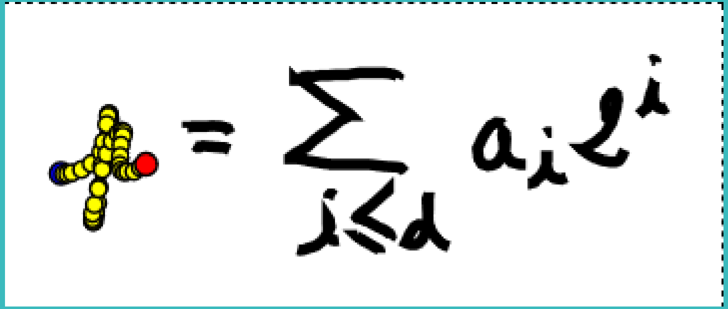
WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Background

- **INKML:** The input data format. The InkML format enables to make references between the digital ink of the expression, its segmentation into symbols and its MathML representation.
- **MATHML:** A standard (XML) format to represent a mathematical formula.
- **TensorFlow:** TensorFlow is a machine learning system that operates at large scale and in heterogeneous environments. Makes it easier to implementing Machine Learning algorithms based in existing libraries in the framework.
- **CROHME:** Competition on Recognition of Online Handwritten Mathematical Expressions

INKML vs MATHML



```
<annotationXML type="truth" encoding="Content-MathML">
    <math xmlns='http://www.w3.org/1998/Math/MathML'>
        <mrow>
            <mi xml:id="p_1">p</mi>
            <mrow>
                <mo xml:id="=_1">=</mo>
                <mrow>
                    <munder>
                        <mo xml:id="\sum_1">\sum</mo>
                        <mrow>
                            <mi xml:id="i_1">i</mi>
                            <mrow>
                                <mo xml:id="\leq_1">\leq</mo>
                                <mi xml:id="d_1">d</mi>
                            </mrow>
                        </mrow>
                    </munder>
                    <mrow>
                        <msub>
                            <mi xml:id="a_1">a</mi>
                            <mi xml:id="i_2">i</mi>
                        </msub>
                        <msup>
                            <mn xml:id="2_1">2</mn>
                            <mi xml:id="i_3">i</mi>
                        </msup>
                    </mrow>
                </mrow>
            </mrow>
        </mrow>
    </math>
</annotationXML>
```

Related Work

- The systems that participated in the original CROHME competitions will serve as our baseline systems to compare our results against.
- (Lu, C., et al. 2015) and (Zhang, J., et al. 2017)
- Our solution is an amalgamation of various ideas and develop a system based on modern Machine Learning frameworks and tools.
- Based on the existing literature we'll break down the problem into two sub-tasks
  - Symbol Recognition
  - Structure Recognition
- Multiple Sub-systems solve specific smaller tasks working in tandem to tackle the overall problem statement.

WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Approach

- Provide a brief sentence or two regarding how you approached the project. For example, was this a quantitative or qualitative project? Was a computer program written? Was this a research project of an existing topic?
- We’ll use an iterative approach to build the systems and subsystems.
- We’ll start with a quick and trivial implementation.
- Evaluate the performance, and make improvements as required
- The focus of the project will not be in trying to implement niche machine learning algorithms but instead to leverage the framework to its fullest capacity and put together a system that would compete and/or exceed the baseline performances set previously

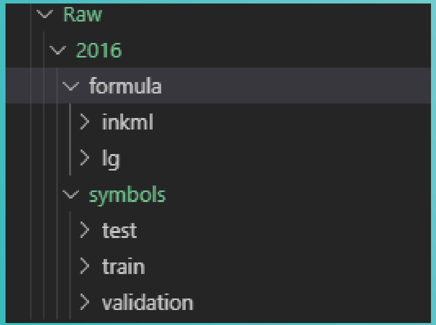
WE’RE ALL ABOUT THE FINISH

CityU of Seattle

Data Collection

- CROHME has been collecting the data by various methods and has this data available in a W3C standard format (InkML).
- We'll use this data as-is as our raw input data.
- We'll process the data to the right format as per the needs of our Machine Learning framework.

Data Collection



- Manual Data Organization
  - Organizing the raw data in the file structure in an optimal format.
  - Removed redundant data and settled on a simple file structure.
- Manual Data Validation:
  - Validated file counts from original dataset.
  - Randomly validated ~100 data points to verify the ground truth is visually similar to the stroke data.
- Manual Data Visualization:
  - Used InkML viewer to visualize provided ink data.
- Automated Data Validation:
  - Python scripts to validate data.
  - Run on each pull-request in the repository.
- Automated Data Visualization
  - Python tool to display multiple InkML files in grid format for quick and easier visual analysis.

Data Analysis

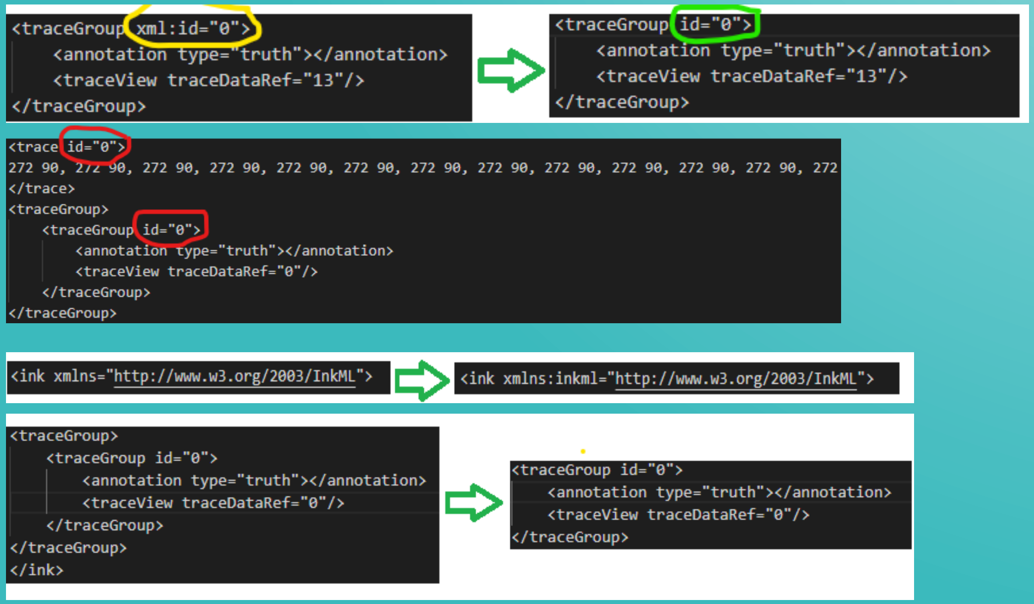
- Initial visual ad-hoc analysis performed with a tool provided by CHROME. InkmlViewer
- Custom python scripts to perform bulk views and analysis.
- This was useful in to validate the correctness of the pre-processing steps performed on the data.
- Data and Models stored in MongoDB for offline analysis.

WE'RE ALL ABOUT THE FINISH

CityU of Seattle

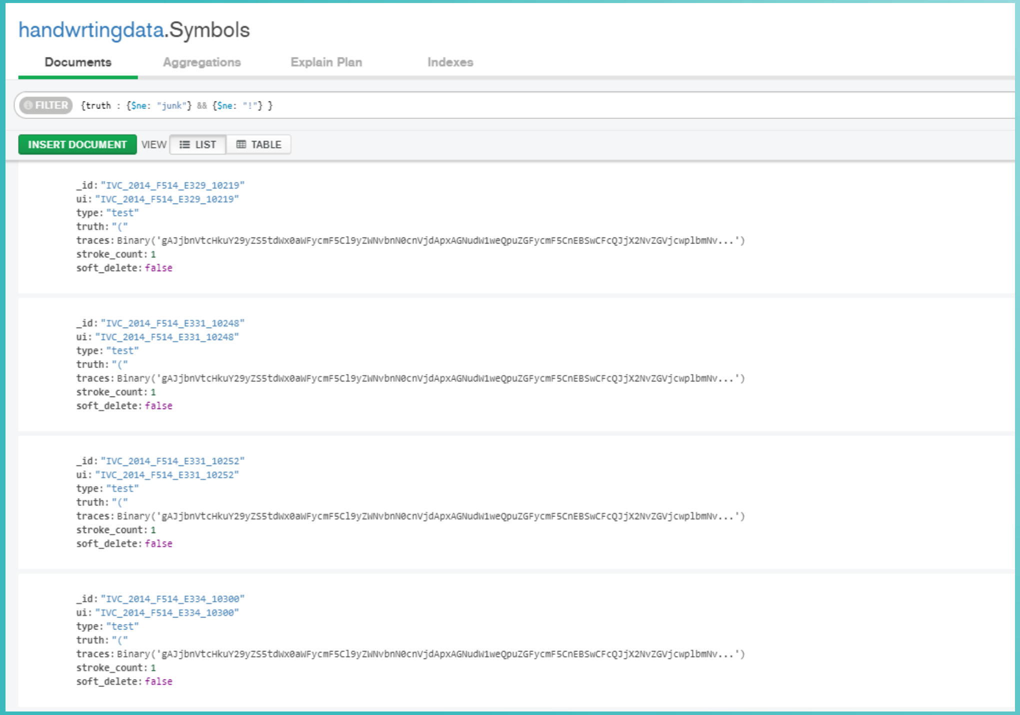
Data Cleansing

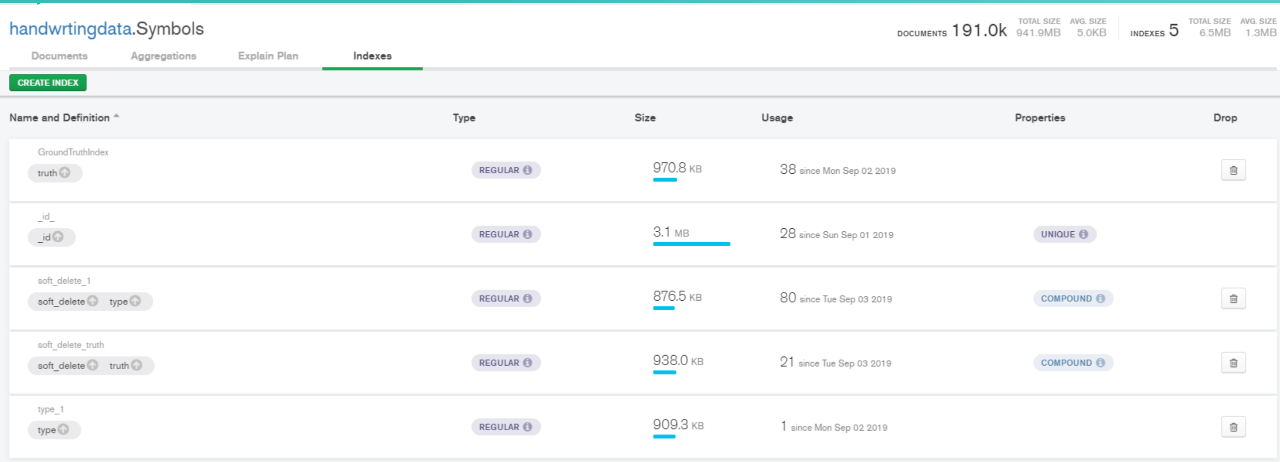
- Several issues addressed in the initial dataset.
    - “xml:id” issue
    - non-unique id attributes
    - invalid namespace
    - Redundant nesting
- Python scripts to cleanse the datasets.
- Python scripts to validate the datasets.



Data Storage

- Processed dataset stored in MongoDB for quick retrieval and advanced query capability.
- Indexes on columns for faster querying.
- Added the ability to prune datasets by soft deleting certain datasets.
- Python scripts use the pyMongo client library to retrieve da from the DB to create Machine Learning models.



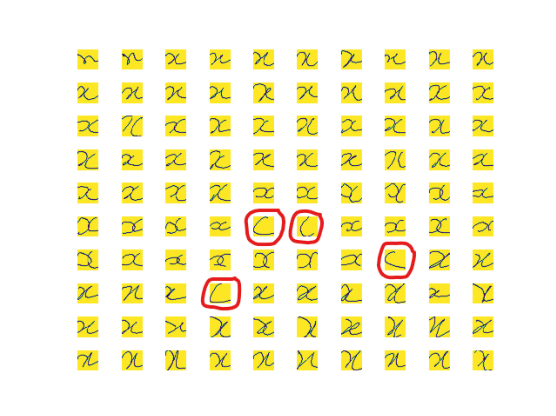


WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Findings - Visualizing the data

- Python's matplotlib and custom python scripts to view the data.
- Training data may not be clean. For example: datapoints circled in red.
- The system could be robust to handle these outliers due to the large dataset size. Example (5042 training samples for x)
- Worst performing symbols could be ones with low number of training samples. These will need to looked up closer and prune the training data if necessary.



WE'RE ALL ABOUT THE FINISH

CityU of Seattle

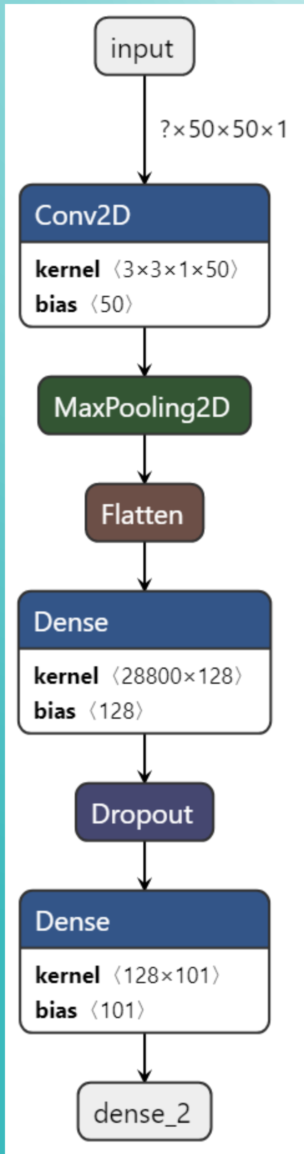
Findings – Generating Models

- Keras to model the Neural Networks.
  - High-level neural network API.
  - Simplifies Neural Network creation.
- Tensorflow as backend for Keras.
- GPU vs CPU
  - Training performed on Intel Core i7 4710MQ (CPU) vs NVIDIA GeForce GTX 880M(GPU)
  - GPU Performance: 975us/step or ~1024steps/sec
  - CPU Performance: 8ms/step or ~125steps/sec

```
model = Sequential()

model.add(Conv2D(50, kernel_size=(3,3), input_shape=symbol_input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten()) # Flattening the 2D arrays for fully connected layers
model.add(Dense(128, activation=tf.nn.relu))
model.add(Dropout(0.2))
model.add(Dense(101,activation=tf.nn.softmax))

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

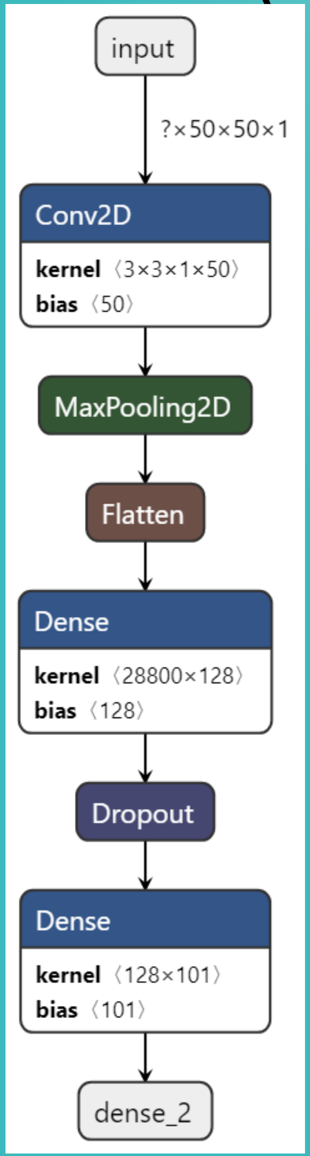


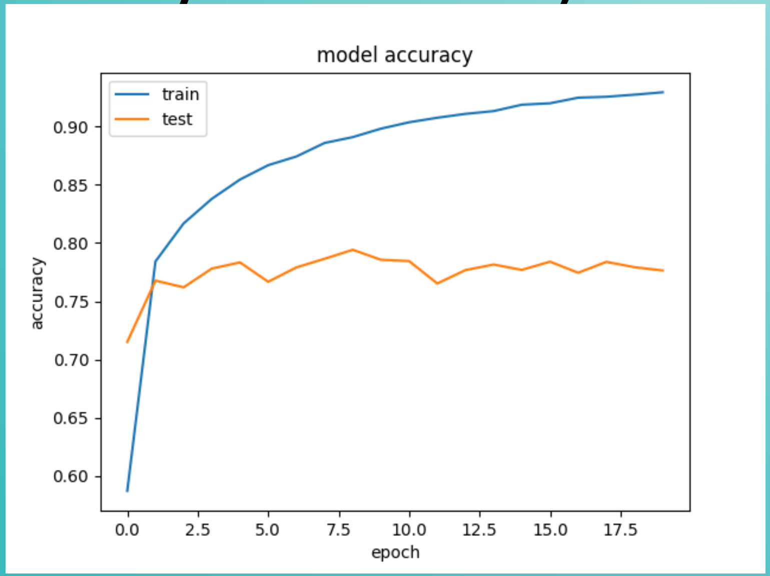
WE'RE ALL ABOUT THE FINISH

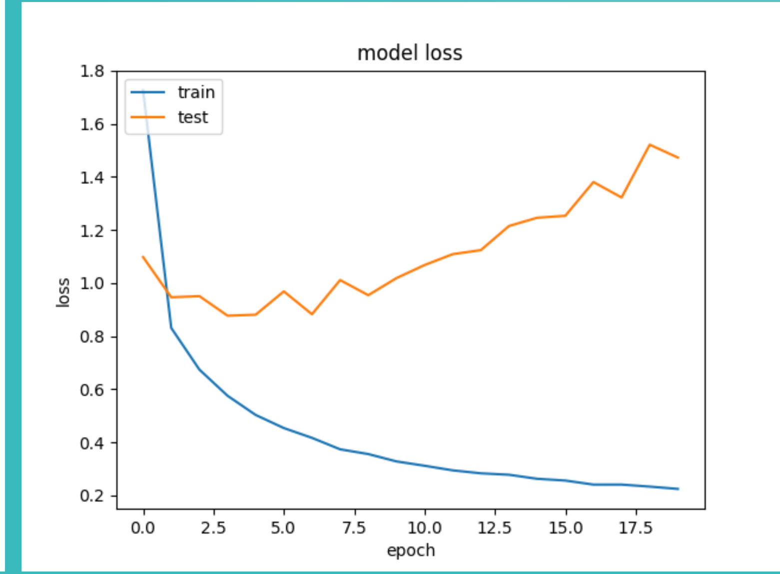
CityU of Seattle

Findings – Model Performance (1-Layer CNN)

- Performance on a Convolution Neural Network (CNN).
- Performance on Validation and Test datasets going flat at ~80%
- Model is overfitting the training data and unable to generalize.
- Clearly visible from increase in the loss function on the validation data while the training loss is going down.
- Next steps:
  - Tune other parameters like kernel sizes in Conv2D and first dense layer.
  - Look into the top error labels. Model could be having issues differentiating similar mathematical symbols like '/' vs '|' or '0' vs 'o'.
  - Tune other parameters like optimizer and loss functions or the dropout factor.
  - Compare the performance against the baseline systems' performances.





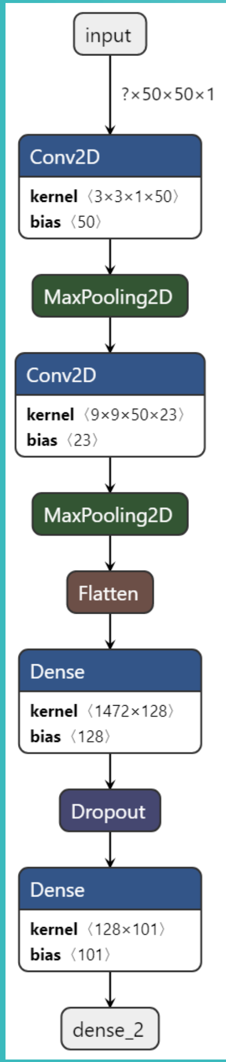


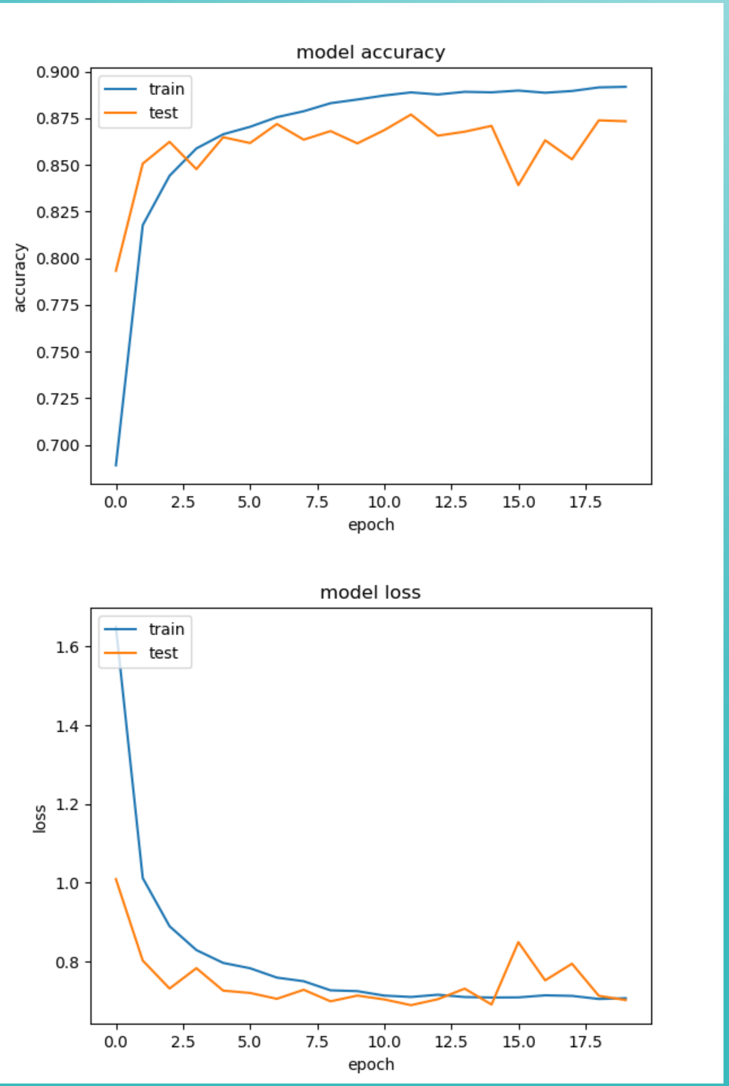
WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Findings – Model Performance (2-Layer CNN)

- ~82% validation and test accuracy on 1-layer even with tuning hyper parameters
- Validation loss always increasing after ~6 epochs.
- Extend the CNN with multiple convolutional layers.
- ~88% validation and test accuracy. Comparable to baseline models.
- Model loss on validation set clearly contained.
- Model is generalizing better





WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Findings – Formula segmentation

- Open CV contour API
- Can handle spatial relations with ease
- Can iteratively implement complex relations like subscripts, superscripts, fractions and square roots.
- Simple linear model is currently assumed.

WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Conclusion

- Designed a Machine Learning system end-to-end on commercial hardware
- Dedicated hardware is useful to improve performance
- Simple CNNs match performance of existing baseline systems
- Further scope for improvement with more complex CNN architecture
- Valuable data collected for further analysis on understanding the hyper parameters.

WE'RE ALL ABOUT THE FINISH

CityU of Seattle

Future Work

- Improve Symbol Recognition:
  - Further scope to improve the performance of symbol recognition
  - Complex CNNs could improve performance to ~95%
  - Multiple predictions per symbol for visually similar symbols like x and \times
- Formula Segmentation:
  - Current implementation is trivial and assumes linear structure
  - Wide variety of formula un-supported. Ex: polynomial equations
  - Add support for complex spatial relations in symbols. Example: Superscripts, subscripts, fractions and square roots.
- Demo tool:
  - Real test for the model.
  - Establish a feedback loop to drive future improvements.

WE'RE ALL ABOUT THE FINISH

CityU of Seattle