**PREDICTING**
We wanted to create a pipeline that automates the repetitive, tedious process of writing up math equations into LaTeX. Our project implements the first stage of this task: character segmentation and classification using a CNN we implemented in Keras. We feed in an image of a handwritten equation and output the characters present along with their corresponding bounding box information, which we plan to later use in constructing the equivalent LaTeX expression.

**DATA**
We trained our model using Xai Nano’s Handwritten Math Symbols Dataset found on Kaggle, which has examples for 32 classes (numbers, symbols, Greek letters, etc.), which we perform image binarization on. We used 147,234 train and 37,161 test examples.

**FEATURES**
Our input data is simply the pixels of a binarized image, so we hypothesize that the model learns features including curvature, boundary noise, stroke count, distance, and length in deeper layers.

**MODEL**
Initially, we tried to implement YOLO for this task. However, that was a bad idea. We then switched to implementing a character segmentation class and building our own CNN.

**RESULTS**

<table>
<thead>
<tr>
<th>Model</th>
<th>Loss after 40k steps</th>
<th>Accuracy &amp; Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv2-Tiny</td>
<td>Loss around 0.31</td>
<td>Inaccurate &amp; low conf.</td>
</tr>
<tr>
<td>Custom CNN</td>
<td>Batch size: 64, Epochs: 100, Learning Rate: 0.001</td>
<td>Test Accuracy: 0.99</td>
</tr>
</tbody>
</table>

**DISCUSSION**
We found that separating the segmentation and classification tasks is a much better approach for handwritten character recognition than YOLO. We achieved reasonably good results, as our pipeline can successfully segment characters from an image and predict the correct class with good accuracy for non-ambiguous symbols. The model confuses ambiguous pairs like ‘1’ and ‘ascii_124’ = ‘|’; however that can also be confusing for humans. It also has difficulty with superscripts. Overall, we feel that this was a successful start towards a complete handwriting to LaTeX conversion pipeline.

**FUTURE**
We plan on continuing this project over the coming months to complete the pipeline by building an RNN that can take the structured data we obtain from our segmentation/classification step and output the corresponding LaTeX expression.

**RESOURCES & RELATED WORK**
4. H. Chen, “Combining Neural Networks for Offline Handwriting Recognition,” 2014,