Predicting
Our model attempts to determine the string encoded in a captcha. We liked this idea because it highlights the strength of deep learning. Captchas are intentionally designed to "prove you’re not a robot" but with deep learning, a robot could solve a captcha.

Features
We used all three of the pixel channels as features. We considered averaging the channels but decided that color differentiation was important in reading the captchas in some cases.

Results
<table>
<thead>
<tr>
<th>Model</th>
<th>Data Size</th>
<th>String Length</th>
<th>Training Accuracy (%)</th>
<th>Validation Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>128x128</td>
<td>10000</td>
<td>90%</td>
<td>85%</td>
</tr>
<tr>
<td>B</td>
<td>256x256</td>
<td>10000</td>
<td>88%</td>
<td>80%</td>
</tr>
<tr>
<td>C</td>
<td>512x512</td>
<td>10000</td>
<td>92%</td>
<td>90%</td>
</tr>
</tbody>
</table>

Data
Our data comes from a python api used to create captchas. Therefore we can generate more data as we need it. The generator takes a string and image size as input so we can experiment with different lengths, alphabets, and image size.

Model
In our early stages we experimented with fully connected models using relu activation and a sigmoid output layer. In order to examine the effects of data generation in training. We found it was best to keep the image size large enough to allow for training during every iteration but then to create more data if desired during the training process.

Discussion
The pixel bug
Generating data
More images vs fewer models
Flexibility of TF vs Keras
All in one web extension
Inherent difficulty
Human error solving captchas
Image overlap
Complex models vs Simple solutions
Lack of Computing Power as motivation
Collaboratory

References

CIS220 Hands-on session 6: "TensorFlow Bites with PyTorch Bites" Internet: https://edlab.research.google.com/drive/12xIPfD43Jg7TwUkCvUtPFc1stX4nAMb7uV5c2ZtSFDzOyaKPK53D, July 25, 2018 (Mon Dec 3).