Kaggle Competition Report

Stephen Fay, Tomer Moran, and Lin Xiao Zheng

**Implementation of the model**

**We have attempted multiple implementations of standard and convolutional neural networks using Keras. Various architectures were tested, each described below.**

**Results (Model performance + best model)**

**Best Model: STEEVEEEE**

**Other Models:**

1. **CNN: 9 hidden layers, ReLU, max pooling, dropout, 48 to 128 filters.**  
   **96.9% validation acc.; 81.3% testing acc.**
2. **NN: 2 hidden layers, sigmoid, regularization.  
   95.7% validation acc.; 77.6% testing acc.**

**Challenges**

**Allowing the model to generalize beyond the data it has seen.**

**Conclusion (what we learned)**

**How to use a new library (Keras)**

**Individual Contribution**

**Stephen (Pre-Trained):**

**Tomer (NN & CNN):** I developed and trained models 1 and 2. I also developed a noise removal filter which processes the images and removes the black dots you guys maliciously added. It does so by running a convolution with a kernel looks at relative differences; black dots are then replaced with the average of the neighbors.

**Lin (Preprocessing):**

**I wrote a denoising autoencoder using keras inspired from models I found online. I trained my model using a set training images with the obvious noise removed and another set with more noise added on. Using it along with Tomer’s CNN, it got a 92% accuracy on the training set, but only 48% on the test set. Then, I tried using convolutional filters hoping each filter could make some features of the image stand out. I trained three models of that CNN on training images that were sharpened, blurred, and both and then carried out a vote to gather the final results. The training accuracy of each models was around 94%, but the test accuracy dropped down to 60%.**