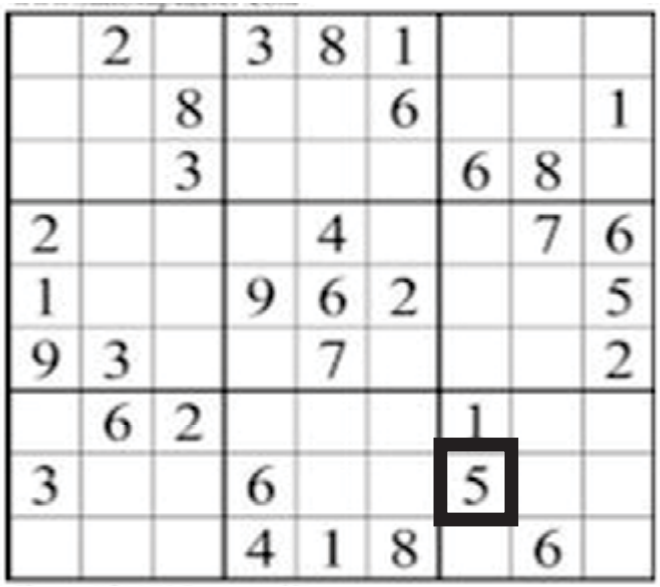
Project Goal
CNN to solve Sudoku puzzles

Summary
As a part of a team project, I created a training data set for our convolutional neural network, CNN, model. The program input is an unsolved Sudoku puzzle image. Then, the boxes or cells with prefilled numbers are extracted. The CNN model then predicts the numbers present, and based on these predictions the program solves the Sudoku puzzle.

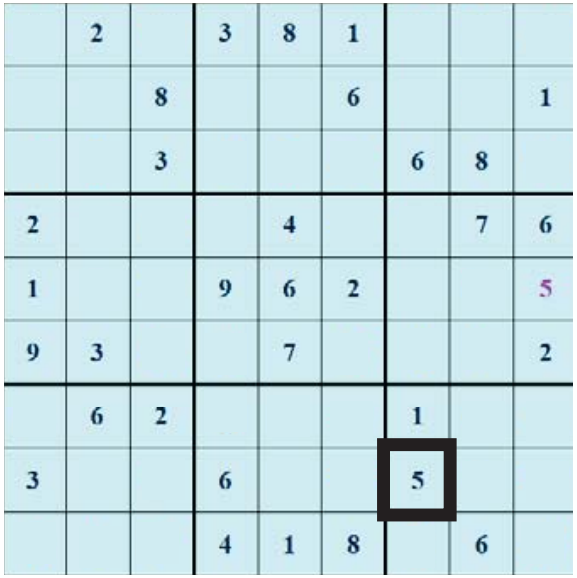
I created 37,800 labeled images with printed digits, 4,200 images of each digit 1-9. The training data set has 60 different fonts and was created using InDesign and Python. The original 540 images were exported in order, maintaining the order of the images in the file directory. With this predictability, the corresponding labels were created with numpy in Python.

Based on poor early CNN performance, noise and borders were added to more accurately reflect the numbers extracted from the Sudoku puzzles. In Python, the original 'clean' images were transformed 70 times using numpy transformations and overlaying images using openCV. The additional corresponding labels were created alongside the transformations.

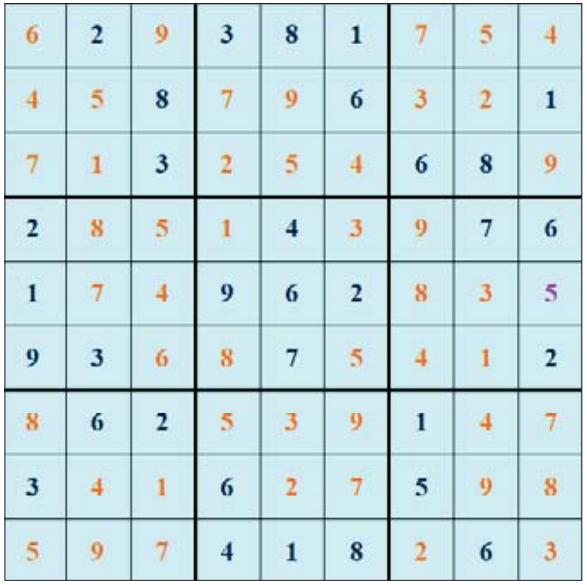
Included
Images of training data creation process, overall project flow, and model results



Scan and interpret Sudoku puzzle by cell

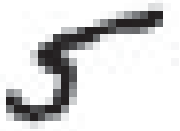


Predict digits present in Sudoku puzzle using CNN

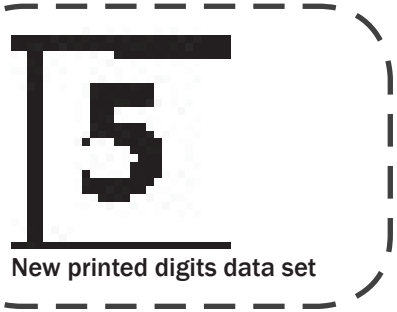


Sudoku puzzle solved and results displayed

Extract numbers from puzzle and use training data sets in CNN to predict numbers present



Handwritten MNIST data set



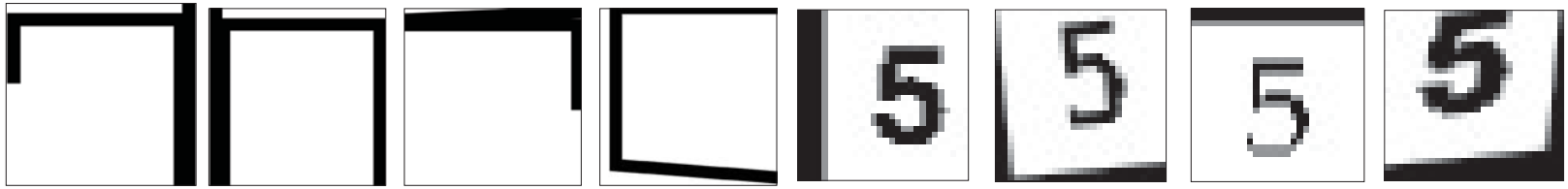
New printed digits data set

Future applications

With the printed digits and handwritten MNIST data sets we could identify prefilled printed and handwritten digits using a CNN. Then the model and program could correct or aid puzzle solution attempts.

Considerations:
Image and digit proportions, font weights, natural shifts of digits between fonts

60 different fonts per 1-9 digit
540 images total



Considerations:
Maintain digit visibility within image frame

Overlay 40 different borders, skew and shift 30 times
70 transformations per original image

Considerations:
Maintain space between ‘noise’ and digits

Examples of new printed digits data set
37,800 total images

Results

With the printed digits training data set, we achieved 96.2% accuracy rather than 23.1% accuracy with the well known handwritten MNIST data set. Note the use of a convolutional neural network was a project requirement. OCR, optical character recognition, is a potentially a more reliable approach within our Sudoku project than the CNN model.