# Calculating Equations Using K-NN Number and Symbol Recognition

Ryan Williams and Carson Crick

## Our Algorithm

For our project, we implemented K-NN to classify numbers (0-9) and symbols (+,-,*,/) We use two different formulas to determine how similar two pictures are (performances of each formula down below). The first is classic Euclidean Distance. It takes coordinates that are above a certain whitespace value from the test picture and training pictures and finds the distances between each training picture and the test picture. The second formula is from a 3rd party and is Euclidean distance but instead of using coordinates, it uses grayscale values from each pixel in a 28x28 picture. The k lowest distances are then taken and the most prevalent label is the test picture's classification. Performances of different k-values with varying training data are seen below.

## Our Problem Space

For the numbers 0-9 we decided to use the MNIST dataset. It provides a training set of 60,000 images and a testing set of 10,000 images. All digits are handwritten and were written by numerous people to account for different handwriting styles. The handwritten digits were then centered in 28x28 images and each pixel in each picture was given a grayscale number of (0-255).
For our symbol data (+,-,*,/), we had to hand create it in Python. We used a similar system to MNIST, using 28x28 images and grayscale values. In total we have 41 training images and 31 testing images.

## Performances