

---

# Writeup

## Project Overview

In our project, we write a function to apply a kernel to an image, and then In this project, i generated a dataset by pasting images on to various background with different textures. I then trained a CNN to classify the images based on the shapes that were pasts onto the image.

## Implementation Detail

### 0.1 Pre-Processing

The images for the dataset come from the Cal Tech 101 Silhouettes data set. From this dataset, I selected four shapes to use for our model. We used the anchor, the ceiling fan, the crab, and the scorpion. I chose these shapes because they all had a similar structure and would still be a tough task. There were other possible subsets of shapes that looked nothing like each other that would have been too easy for the model. Once I got all the shapes I wanted to use, I created a data set of the images by pasting the shapes in a random location on one of the background textures I generated. The background textures are randomly chosen to be one of the eight I made. These include gradients, stripe patters with horizontal and vertical lines, chess board pattern, a triangle tiling pattern, along with random noise and a white background. The same thing is done with the validation and test data sets.

Figure 1: A scorpion on top of the checkered triangle background



Figure 2: An anchor shape on top of a gradient background



Figure 3: A ceiling fan shape on top of a striped background



---

Figure 4: A crab shape on top of the random noise background



## **0.2 Neural Network**

I made a CNN for the classifier network. The CNN has 3 layers of convolution and relu, max pooling, and batch normalization, followed by a fully connected classifier layer.

## **Result**

After testing the model, it usually ends up with around an 85 – 80% accuracy.

## **1 Instructions**

To run the project, just run the main.py file from the command line or from your text editor. The accuracy results will be shown on the command line.