Documentation

Problem Statement:
In the context of computer vision and image recognition, the task is to develop a classification model that can accurately classify geometric shapes into one of three classes: "Circle," "Triangle," or "Rectangle."

Approach:
Data Collection: I started this project by searching for a relevant dataset, that can classify the “geometric shapes” and I can find one dataset from Kaggle. The dataset consists of three sub-folders for each class ‘Circle’, ‘Triangle’, and ‘Square’.

Creating Image Path: Then I make 3 different lists for each directory to create an image path, in this way, I get a path that holds a complete Path for each image in the directory. To achieve this I use list comprehension with a for loop.

Data splitting: I split the dataset into three sets ‘train’, ‘vali’, and ‘test’ in a ratio of 70-15-15 respectively. with each set containing separate folders for ‘Circle’, ‘Square’, and ‘Triangle’ init.

Data Loading: In the data loading step I create three variables ‘train_data’, ‘val_data’ and ‘test_data’, using Keras which returns me a Tensorflow object that yields batches of images and labels. Here I use a batch size of 2 and image are resized (64, 64) and the labels are converted to the ‘int’ data type. Data Normalization was performed on the image data by dividing it by 64.

Model Training:
1. I built a CNN model with a set of 3 convolutional layers, 3 max-pooling layers, and fully connected layers. convolutional layer with 32 filters, each of size 3x3. The activation function of the layer is 'relu'.
2. max pooling layer with a pool size of 2x2. The max pooling layer reduces the spatial dimensions of the feature maps by taking the maximum value from each 2x2 window.
3. Flattend the output from the previous layer into a one-dimensional vector.
4. fully connected layer with 128 hidden units. The activation function of the layer is 'relu'.
5. dropout layer with a dropout rate of 0.5. Dropout is a regularization technique that helps to prevent overfitting by randomly dropping out neurons during training.
6. The output layer has 3 units, which corresponds to the 3 classes that the model is trying to classify. The activation function of the layer is 'softmax', which produces a probability distribution over the 3 classes.
7. The model was compiled with the Adam optimizer and the sparse categorical crossentropy loss function.
8. The model was trained on the training set and evaluates its performance on the validation set, for 10 epochs.