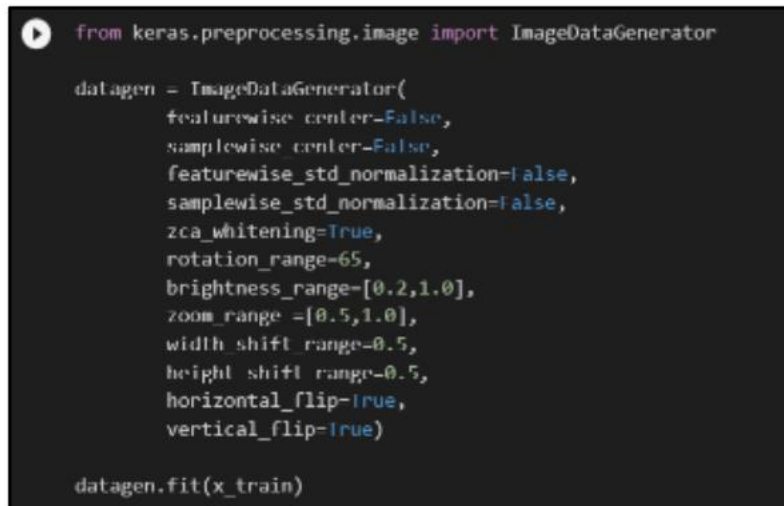# METHODOLOGY

Firstly, this project was implemented on Google Colab platform as it enables the user to connect to the GPU. Libraries such as NumPy, Matplotlib, tensorflow, keras and dataset are imported.

```python
from keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(
        featurewise_center=False,
        samplewise_center=False,
        featurewise_std_normalization=False,
        samplewise_std_normalization=False,
        zca_whitening=True,
        rotation_range=65,
        brightness_range=[0.2,1.0],
        zoom_range =[0.5,1.0],
        width_shift_range=0.5,
        height_shift_range=0.5,
        horizontal_flip=True,
        vertical_flip=True)

datagen.fit(x_train)
```

The main aim of this project was to import libraries and dataset, apply data augmentation, generate a new dataset, execute a convolution neural network, and obtain the desired results. The problem to be solved is decreased accuracy, problem of overfitting and limited amount of ground truth data. Data augmentation is a technique of generation of new training samples by applying various a list of transformations. The transformations might include random shifts, cropping, erasing, addition of noise, random flips, zooming and many more. This project implements the data augmentation parameters as shown in figure.

ZCA whitening is a process where the data displays decorrelated features. This parameter was set to a boolean value of "True". The horizontal and vertical flip were set to "True" which indicated that the pixels of rows and columns were reversed. Further, the rotation angle was set to a value of 45. This shows that every image of handwritten digit was rotated at an angle of 45 degrees. Rotation is a parameter which takes the input ranging from 0 to 360 degrees. Floating point numbers can be seen in the shift parameters which implies that the images undergo a horizontal and vertical shift by 50%. The zoom and brightness parameters are specified in floating point ranges. Here, the zoom parameters were specified as [0.5,1.0]. the

zoom operation enables the pixel zooming in a uniform manner throughout the image. The brightness range for this project was set to [0.2,1.0]. The aim of specifying the brightness parameter is to enable the model to be accustomed to a dataset which contains images of different brightness levels.

The above data augmentation parameters were further applied to a CNN network. The CNN network consisted of three convolutional layers of 64, 32 and 32 neurons respectively with ReLU activation function. The kernel size was set to 5. The output layer of the convolutional neural network consisted of 10 units with softmax activation function.

```
model.summary()
Model: "sequential_1"

 Layer (type)                  Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)            (None, 28, 28, 64)         1664

 max_pooling2d_3 (MaxPooling  (None, 14, 14, 64)         0
 2D)

 conv2d_4 (Conv2D)            (None, 14, 14, 32)         51232

 max_pooling2d_4 (MaxPooling  (None, 7, 7, 32)           0
 2D)

 conv2d_5 (Conv2D)            (None, 7, 7, 64)           51264

 max_pooling2d_5 (MaxPooling  (None, 3, 3, 64)           0
 2D)

 flatten_1 (Flatten)          (None, 576)                0

 dense_3 (Dense)              (None, 128)                73856

 dropout_2 (Dropout)          (None, 128)                0

 activation_2 (Activation)    (None, 128)                0

 dense_4 (Dense)              (None, 512)                66048

 dropout_3 (Dropout)          (None, 512)                0

 activation_3 (Activation)    (None, 512)                0

 dense_5 (Dense)              (None, 10)                 5130

=================================================================
Total params: 249,194
Trainable params: 249,194
Non-trainable params: 0
_____
```

The above model was compiled with the Adam optimizer and categorial cross entropy loss function. The number of epochs and batch size were set to 100 and 128 respectively. After the successful execution of the above specified model, the accuracies are determined.