E9-246 ADVANCED IMAGE PROCESSING

ASSIGNMENT 1: REPORT

NAME: *HARSHIT DUBEY*

SR. NO.: *04-03-06-10-51-23-1-23043*

DEGREE: *M.TECH AI*

Question 1:

Details of your implementation:

Keypoint and Orientation Detection:

Scale Space Creation:

- Generates a stack of blurred images (octaves) using Gaussian filters with increasing sigma values.
- Computes Difference of Gaussians (DoG) between consecutive blurred images to capture potential keypoints.

Extrema Detection:

- Localizes keypoints by identifying pixels that are maxima or minima in their 3x3x3 neighborhood within the DoG layers.
- Applies a contrast threshold to filter out weak keypoints.

Orientation Assignment:

- Calculates gradient magnitude and orientation at each keypoint using Sobel filters.
- Constructs a histogram of gradient orientations in a neighborhood around the keypoint.
- Assigns the dominant orientation (bin with maximum count) as the keypoint's orientation.

Descriptor Generation:

- Extracts a 41x41 patch around each keypoint, oriented according to its assigned orientation.
- Calculates horizontal and vertical gradients within the patch using Sobel filters.
- Concatenates the gradients into a single vector.

PCA Reduction:

- Applies Principal Component Analysis (PCA) to reduce the dimensionality of the gradient vectors to 100.
- Retains the most significant information for robust matching.

Additional Functions:

- PCA_SIFT (image): Main function that calls the keypoint detection and descriptor generation steps.
- show_keypoints_with_orientation (image, keypoint_info): Visualizes the detected keypoints and their orientations on the image.

Output Format:

- The code produces a list of lists as its output, where each inner list represents a single keypoint and contains the following information:
  - Keypoint coordinates (x, y)
  - Scale
  - Octave
  - Orientation angle
  - PCA-SIFT descriptor (100-dimensional vector)

Building

Books

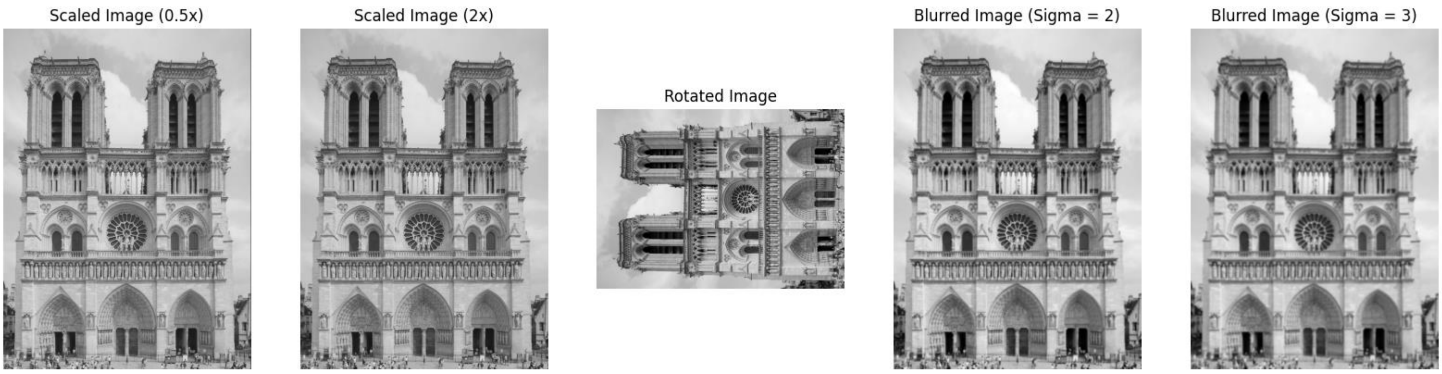
Output Format:

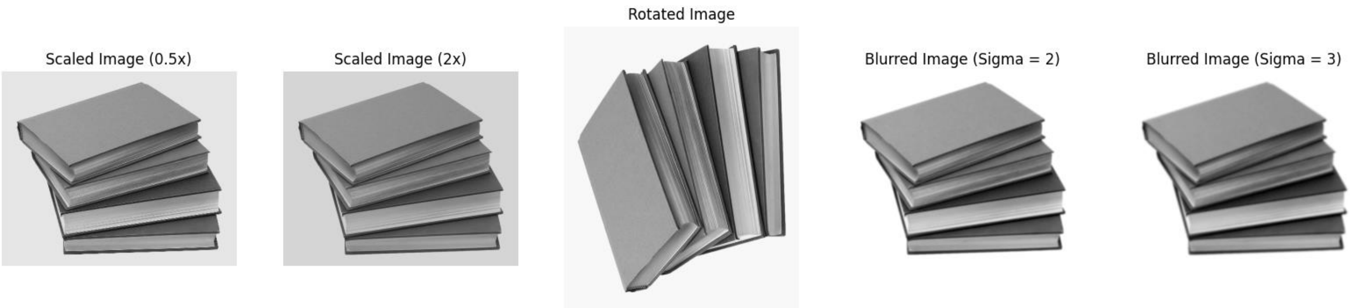
```
Coordinates of the keypoint is: (43, 498)
Scale of the keypoint is: 1
Corresponding octave of the keypoint is: 1
Orientation_Angle of the keypoint is :195
Descriptors of the keypoint is:
[-1.07117474e-01  3.93118522e-01  1.83789514e-02 -1.88730363e-01
 -3.45481796e-02 -1.35323090e-01 -1.51301204e-01 -8.67923293e-02
 -3.80593578e-02  3.70335290e-02 -1.93744649e-02 -4.92655508e-02
 -1.35839434e-02 -5.65353711e-02 -2.11662558e-02  2.05961236e-02
  3.96688447e-02 -3.32548632e-03 -6.76182196e-02 -2.88613629e-02
  2.33147590e-02  2.17930261e-02  2.55067855e-02  7.46143163e-03
  1.96480327e-02 -2.95152981e-03 -3.36647590e-02 -8.46706207e-03
 -2.46839190e-03  1.14813864e-02 -4.07427329e-02 -4.28514734e-02
 -3.87674312e-02 -4.60710718e-03 -3.37527166e-03 -2.68573552e-02
  3.96465985e-03  6.70838426e-04  2.20852339e-02  1.92310587e-02
  2.48364491e-02 -1.48587569e-02  6.45575756e-03 -7.59928731e-03
 -1.59966187e-02  2.75586880e-03  2.16966927e-03  3.86254449e-03
 -1.23301962e-03  4.55343958e-02 -5.11257424e-03 -3.17286279e-03
  5.24230120e-03 -6.95583548e-04  2.67492209e-02  1.54148660e-03
  3.24315014e-03 -6.07184539e-03 -7.72897561e-03  3.22873764e-02
 -1.80290479e-02 -1.19864928e-02 -1.60041738e-02 -1.15443149e-02
  1.97629126e-02  1.89917174e-02  1.79821426e-03  1.01741819e-02
 -1.21657031e-02  3.01131484e-04 -7.58768710e-03  4.49353245e-03
 -6.10162674e-03 -8.30001527e-03 -4.05815056e-03 -1.17192438e-02
  1.94664588e-03  1.70539855e-02 -7.90235542e-04  1.28211055e-02
 -1.06206811e-02  3.28260609e-03 -1.33863212e-02 -1.06871891e-02
  2.84253655e-03 -1.18611122e-02 -1.56474006e-03 -4.34062849e-03
 -1.02038481e-03 -9.56641835e-03  1.26643416e-02  3.25253718e-02
 -6.15910939e-03  1.05462699e-03  1.09884355e-02  3.51588591e-03
 -2.18494593e-02 -2.20576741e-02  2.38675770e-02  3.55763575e-03]
```

Analysis:

Number of Keypoints:

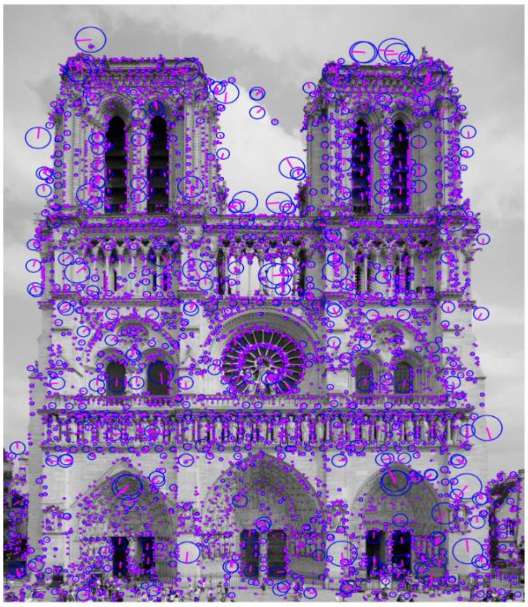
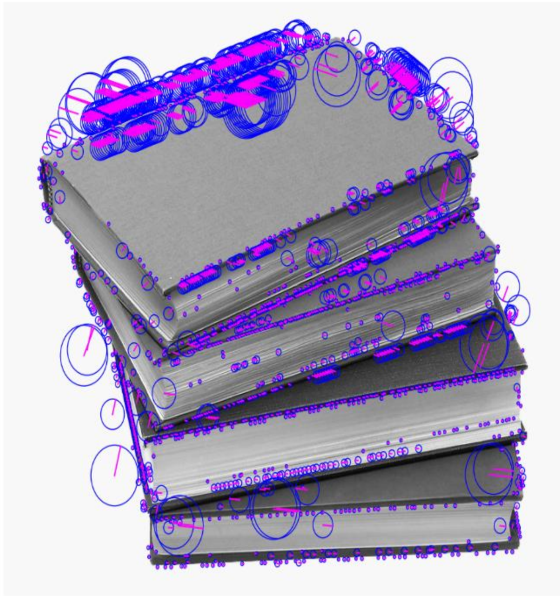
Type	Books	Building
Original	1628	6621
Rotation	1628	6621
Scaled(x0.5)	482	1779
Scaled(x2)	3726	28108
Gaussian Blur(sigma = 2)	837	3838
Gaussian Blur(sigma = 3)	661	1983





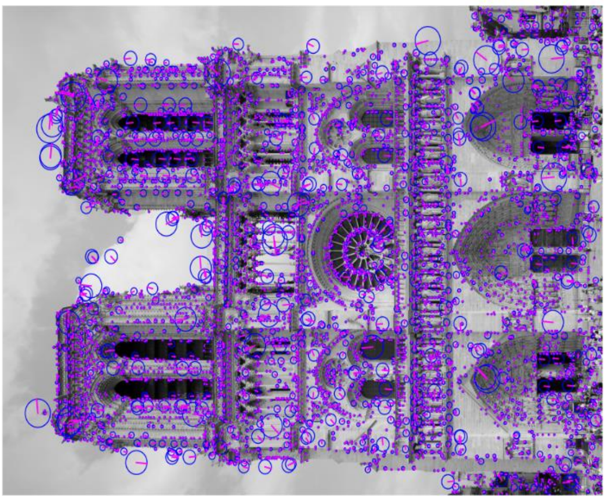
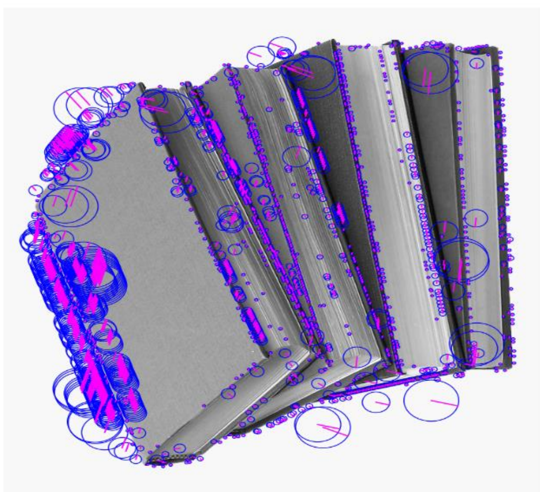
Original:

- Books: 1628
- Building: 6621

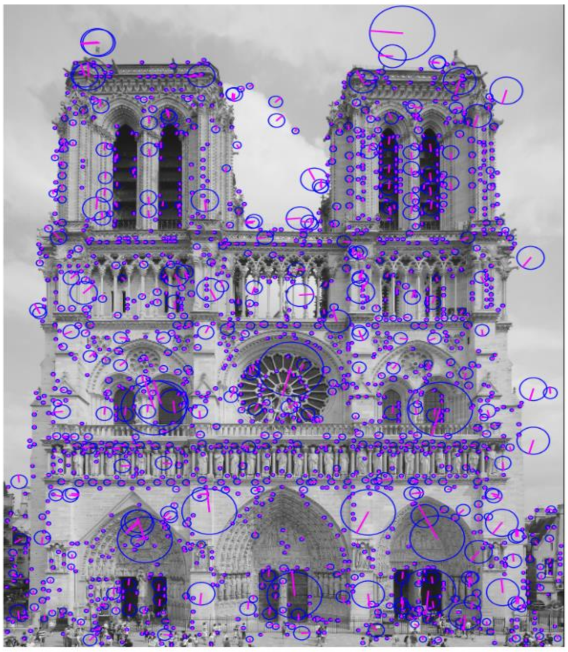
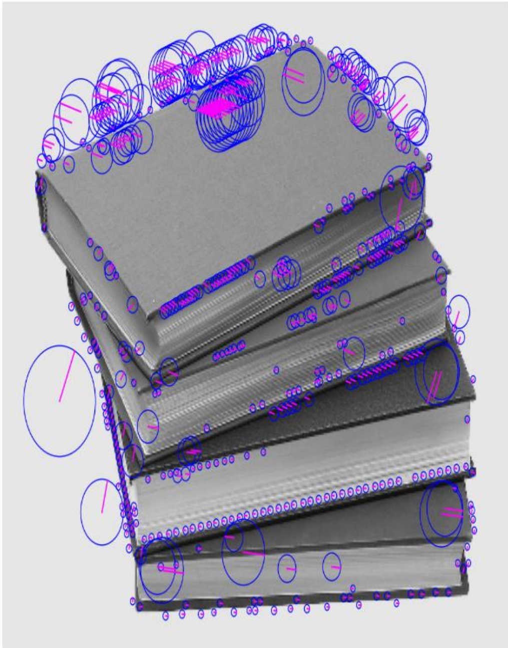
Rotation:

- For 90 degrees no. of features are almost same as expected as the image is intact and is just rotated by 90 degrees. Observed numbers (1628 and 6621) confirm this. For other angles some amount of change in number of keypoints are expected.

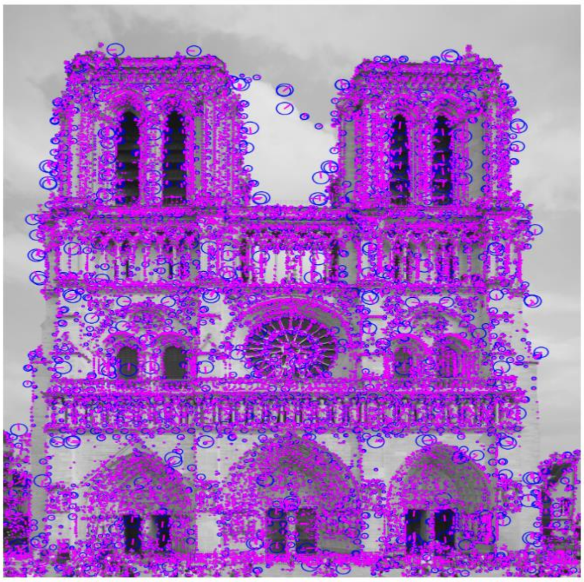
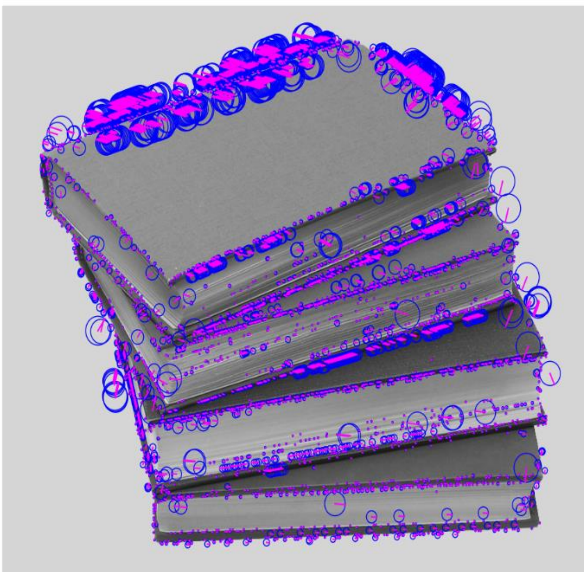
Scaled (x0.5):

- Both images show a significant decrease in keypoints (482 for Books, 1779 for Building). This is expected as scaling down reduces the image size, leading to fewer potential keypoint locations.

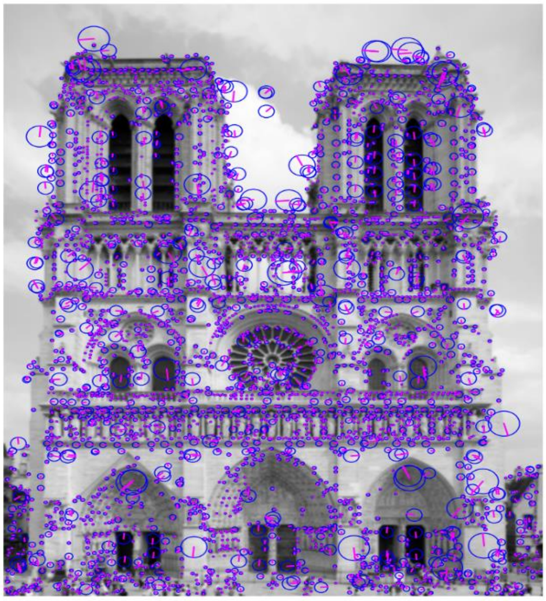
Scaled (x2):

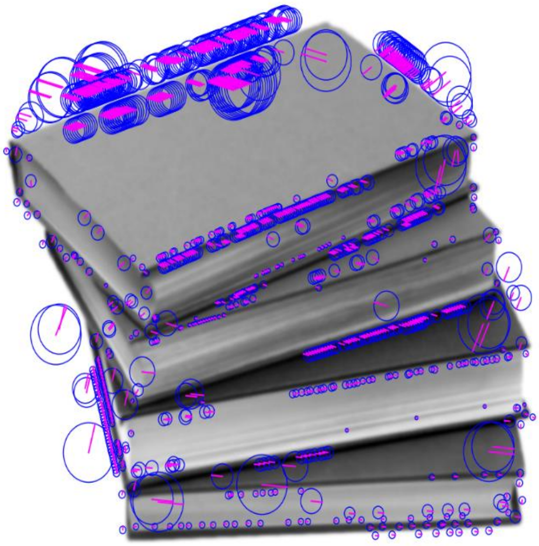
- Both images show a significant increase in keypoints (3726 for Books, 28108 for Building). Scaling up introduces more details and potential keypoint locations.

Gaussian Blur (sigma = 2):

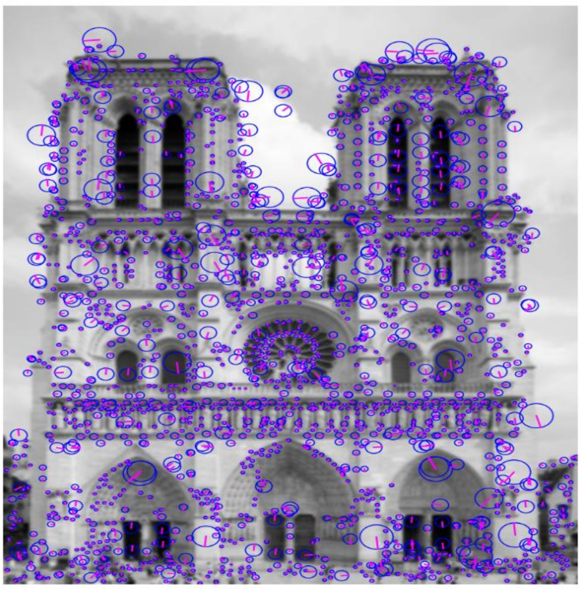
Both images show a decrease in keypoints (837 for Books, 3838 for Building). Blurring smoothens out details, reducing potential keypoint locations.

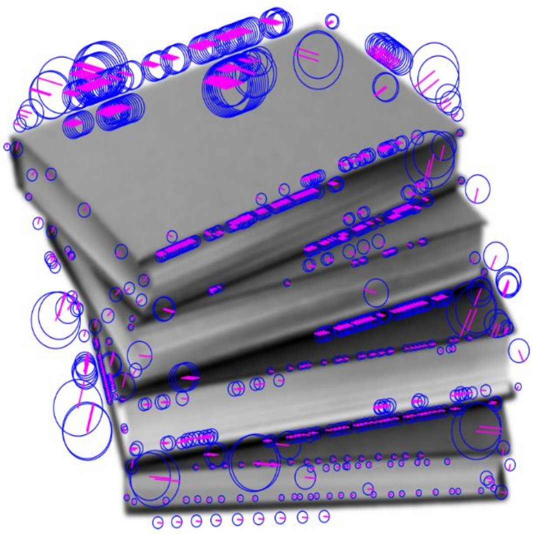




Gaussian Blur (sigma = 3):

Both images show a further decrease in keypoints compared to sigma = 2 (661 for Books, 1983 for Building). Higher blur results in more smoothing and fewer keypoints.





Qualitative Analysis:

- ***Rotation:*** Rotation by 90° preserves keypoints as the image content stays the same. Other rotations may change them as features align differently with the detector.
- ***Scaling:*** The expected behaviour is observed, with scaling down reducing and scaling up increasing the number of keypoints due to changes in image size and detail.
- ***Blurring:*** As the blur sigma increases, the number of keypoints decreases as expected due to the loss of detailed features.

Additional considerations:

- The choice of keypoint detection parameters (e.g., contrast threshold) can affect the number of detected keypoints.
- The nature of the images (e.g., texture, presence of edges) can also influence the number and distribution of keypoints.

Question 2:

Details of your implementation:

Data Preprocessing:

- extract_classnames: This function utilizes the '.decode('utf-8')' method to convert byte strings from the label_names key in the meta_data dictionary to human-readable class names.
- reshape: This function reshapes the raw data from a (samples, 3072) format to a (samples, 32, 32, 3) format. This transformation creates a 4D tensor where each 2D slice represents a colour image (3 channels) of 32x32 pixels. Transposing (0,2,3,1) ensures the correct ordering for convolutional layers: samples-first, followed by height, width, and channels.
- extract_labels: This function simply extracts the numerical labels (int values) from the provided data batch.

CNN Architectures:

- Both architectures: They share the same basic structure of convolutional layers followed by dense layers. They differ in the activation function used in the convolutional layers: sigmoid in the first and relu in the second.
- Convolutional layers: Each convolutional layer applies 64 filters of size 3x3 to the input. These filters extract spatial features from the images. Max pooling layers (2x2) then reduce the dimensionality and introduce some translational invariance.
- Dense layers: After flattening the output of the last convolutional layer, two dense layers are used for classification. The first dense layer has 64 neurons and the final layer has 10 neurons with SoftMax activation, one for each class.

Model Compilation:

- Optimizer: Adam is an adaptive learning rate optimization algorithm that dynamically adjusts individual learning rates for each parameter based on their past gradients.
- Loss function: Sparse categorical cross-entropy is used for multi-class classification problems where each sample belongs to a single class. It measures the difference between the predicted probability distribution and the one-hot encoded true label.
- Metrics: Accuracy is a simple metric that measures the percentage of correctly classified samples.

Model Training:

- fit function: This function trains the model on the provided training data for 15 epochs. An epoch represents one complete pass through the entire training dataset.

Evaluation:

Classification report-Sigmoid (Test Set 1)

	precision	recall	f1-score	support
0	0.69	0.58	0.63	1000
1	0.63	0.77	0.70	1000
2	0.51	0.37	0.43	1000
3	0.44	0.39	0.41	1000
4	0.49	0.56	0.52	1000
5	0.53	0.47	0.50	1000
6	0.61	0.72	0.66	1000
7	0.63	0.67	0.65	1000
8	0.62	0.73	0.67	1000
9	0.67	0.60	0.63	1000
accuracy			0.59	10000
macro avg	0.58	0.59	0.58	10000
weighted avg	0.58	0.59	0.58	10000

Classification report-Relu (Test Set 1)

	precision	recall	f1-score	support
0	0.75	0.66	0.70	1000
1	0.72	0.82	0.77	1000
2	0.59	0.52	0.55	1000
3	0.44	0.53	0.48	1000
4	0.64	0.59	0.61	1000
5	0.54	0.53	0.53	1000
6	0.68	0.80	0.74	1000
7	0.77	0.68	0.72	1000
8	0.77	0.80	0.79	1000
9	0.75	0.70	0.72	1000
accuracy			0.66	10000
macro avg	0.67	0.66	0.66	10000
weighted avg	0.67	0.66	0.66	10000

Classification report-Sigmoid (Test Set 2)

	precision	recall	f1-score	support
0	0.70	0.58	0.63	1000
1	0.63	0.77	0.69	1000
2	0.51	0.37	0.43	1000
3	0.44	0.37	0.40	1000
4	0.47	0.53	0.50	1000
5	0.54	0.46	0.50	1000
6	0.58	0.72	0.64	1000
7	0.63	0.66	0.64	1000
8	0.60	0.73	0.66	1000
9	0.66	0.59	0.62	1000
accuracy			0.58	10000
macro avg	0.58	0.58	0.57	10000
weighted avg	0.58	0.58	0.57	10000

Classification report-Relu (Test Set 2)

	precision	recall	f1-score	support
0	0.73	0.58	0.64	1000
1	0.64	0.80	0.71	1000
2	0.54	0.37	0.44	1000
3	0.42	0.43	0.43	1000
4	0.54	0.48	0.51	1000
5	0.58	0.43	0.49	1000
6	0.45	0.89	0.60	1000
7	0.76	0.61	0.68	1000
8	0.79	0.66	0.71	1000
9	0.67	0.69	0.68	1000
accuracy			0.59	10000
macro avg	0.61	0.59	0.59	10000
weighted avg	0.61	0.59	0.59	10000

Overall Performance:

- ReLU consistently outperformed sigmoid on both test sets, achieving higher accuracy scores:
  - Test Set 1: ReLU (66%) vs. Sigmoid (58%)
  - Test Set 2: ReLU (59%) vs. Sigmoid (57%)

Class-Level Performance:

- ReLU generally outperformed sigmoid across most classes in terms of precision, recall, and F1-score.
- Some notable differences:
  - Classes 0, 1, 6, 7, 8, and 9 saw consistently higher scores with ReLU.
  - Class 4 had similar scores with both functions.
  - Class 2 and 3 performances varied slightly between test sets and functions.

Additional Considerations:

- Hyperparameter tuning: Experimenting with different hyperparameters (e.g., learning rate, number of neurons, epochs) could potentially improve performance for both models.
- Dataset characteristics: The nature of the dataset and task can influence which activation function performs better.

Confusion Matrices: