Task 1:

In task 1, I’ve been provided with the two images and had to merge the images. Here, I’ve used geometrical/ perspective transformation. Simply, I’ve got the corner pixels of the source image and got the image (without resizing it), we can even resize them. Then got the corner pixel of the image where i’ve combined the first image and weights are provided.

Task 2:

For sharpening the image from scratch using convolution, I’ve used the sharpeing matrix. Then the core operation is to convolution of input image with the matrix which is done separately for each color channel. Loops of 3 has been applied for over color channel. Here, cv2.filter2D function is used to convolution of image.

Then in other sharpen image is provided by using OpenCV.

Task 3:

Following are the difference between the k-means clustering and singular value decomposition for image compression:

|  |  |
| --- | --- |
| K-means clustering | Singular value decomposition |
| K-Means groups similar pixel values in images, reducing them to cluster centroids for compression. | SVD decomposes the image matrix into U, Σ, and V^T matrices, enabling compression by truncating smaller singular values in Σ. |
| K-Means compression offers better quality at the expense of lower compression ratios as K (number of clusters) increases. | SVD compression achieves higher compression ratios by truncating more singular values, but this results in a trade-off with image quality. |
| Especially for large images, k-means clustering is intense processes for compression as high values of numbers of clusters are there, more iteration should be applied. So, it is slow process. | It is faster than the k-means clustering compression process, due to its mathematical foundation. |
| K-Means does not have a standardized compression format and can be implemented with variations in approach. | SVD-based compression is commonly used in formats like JPEG and MPEG, which means it's compatible with many different devices and software. This ensures that images compressed using SVD can be easily opened and viewed on various devices and software programs. |
| Adjusting the number of clusters varies the compression ratio. Higher the number of clusters, better image quality but less compression ratio. | You can decide how much quality you want in your compressed image by picking which parts to keep and which to leave out. This gives you more control over how much you're willing to trade quality for a smaller file size. |

Task 4:

References

<https://medium.com/@jorgesleonel/linear-regression-307937441a8b>

Some sort of AI has been used to generate the plotting graph code..

Gradient descent and pseudo-inverse, both works on aiming to minimize the sum of squared residua. These residuals give us knowledge about how well the solution fits in the equation. For pseudo-inverse, it minimizes the sum of squared residual in least-squared sense and gives solution best fit to equation. In other hand, Gradient Descent adjust the value of x1 and x2 continuous loop to reduce the sum of squared residuals. Pseudo inverse provides the solution directly while gradient descent gives solution iteratively.

Task 5:

Firstly, the MNIST dataset has been downloaded. Then, the provided FeatureExtractor class has been defined. Within it, a model has been provided to us to train the dataset. Then, I’ve loaded the pre-trained feature extractor. Then, I’ve created the MultiTaskModel class which takes an input, extracts features using a shared feature extractor, and then uses separate neural networks to perform two-digit classification and binary classification based on the shared features. The output of the forward pass consists of logits for both tasks.

Then, I’ve trained the images with the digit and binary labels. After that evaluation has been done in which accuracy for digit prediction come 99.10% and binary prediction comes 99.42%. After that, I’ve randomly selected the three images and predict them.