

EEC 289A Assignment 1

K-Mean Clustering of 5x5 Image Patches

Randall Fowler

ECE Department, UC Davis

rlfowler@ucdavis.edu

Introduction:

This paper is a submission to an unsupervised learning course homework assignment. It addresses the issue of representing shaping information within an image as well as reconstructing an image with a trained model. By using K-Means algorithm to cluster patches of images, shapes can be found by grouping the data points found from the patch's pixels. These found clusters represent the learned dictionary to display shaping information and support an encoded format of an image for reconstruction.

Methodology:

K-Means:

The K-Means algorithm is a clustering algorithm designed to group data points within the same region. It is a method that is considered to be unsupervised learning as these regions, or groups, are not known. Each group will have a center defined as a centroid, and centroids will be the only information required to be saved to store the model. Only the data and number of centroids are required to begin training a model using K-Means. The number of dimensions for the sample space and number of centroids may add to the time complexity of the algorithm. This is due to the increase in search space by iterating through more dimensions and clusters.

With a specified number of centroids, the clusters will be formed with an initial position of centroids. Data points will be mapped to the nearest centroids to determine which cluster it will belong to. Once all data points have been assigned to a cluster, the centroid will move to the mean of all the data points in its cluster. The goal of this algorithm is to minimize the distance

from each data point to a centroid. After the centroids have moved to the mean of a cluster, the closest centroid to each data point may change. For each time the centroid moves, the data points will be mapped to the nearest centroid, and the centroid will move toward the mean of the newly formed cluster. This will repeat until little to no movement of the centroids occurs.

Many clustering algorithms will implement a K-Means method with different strategies of finding the starting locations of the centroids. Random placement is a common approach as it will fill the sample space uniformly to group data. However, this could be a deceptive approach depending on the distribution of data. A method called k-means++ will carefully place centroids with respect to the distribution of data [1]. It will randomly select a data point to place the first centroid. Subsequent centroids will be placed upon data points based on probability density. The probability of every point in the sample space will be calculated as the square distance to the nearest centroid. There are many other centroid placement techniques, but this is the method utilized in this paper.

Image Dictionary:

While biological creatures are capable of perceiving geometry from images, software does not intuitively notice shapes. Relativity can be exploited to help define the shapes of objects [2]. By zooming into an image, shapes can become small changes amongst a patch of pixels. For this section, shapes will be discussed as a grouping of image patches to make up an image dictionary of known types of patches.

Patches of an image may vary in size depending on the resolution of an image. Many of the patches may hold no shape if in a region where all pixels hold the same value. It was advised to remove the blank patches, but these will be needed to properly reconstruct the images. Extraction of patches will be done iteratively to get a smaller image, and patches may overlap depending on the stride.

Once patches are extracted from all images, these patches will represent the data points in the sample space. For a 5 by 5-pixel patch, this can represent a data point with a dimensionality of 25. K-Means can then be applied to the sample space for all the patches of all the images. The resulting centroids will represent an image patch similar to the clustered data points. With a

reasonable number of centroids, all the arrangements of pixels to create the necessary shapes for an image should be known. These learned centroid locations make up the image dictionary.

Experiment:

As this paper is a class assignment, the focus will be on demonstrating the results of a learned image dictionary on the MNIST dataset. K-Means algorithm will be used to cluster image patches of images of decimal numbers. The patches of the learned dictionary will be analyzed as well as the reconstructed images from the dictionary. Source code for this project can be found within the reference Github repository [3].

To prepare for the K-Means algorithm, all patches were extracted using a stride of 1 with no padding and a size of 5 by 5 pixels. The number of centroids varied between 100 and 10,000, but not many dictionaries were trained due to the time requirement; for this reason, the number of centroids was 100, 500, 1000, 5000, and 10,000.

After training, the resulting dictionaries are shown in figures 1 through 5. Dictionaries larger than 1000 may be difficult to visualize, but the larger dictionaries may not be needed. There are many permutations for filling a 25-pixel area, but centroids may be nearly duplicates in the larger dictionaries. Some of the dictionaries contain no pixels in plots, and these could be due to dead centroids that share the same position with another.

When reconstructing an image from the dictionaries, patches are first extracted, nearest centroids are found, and the closest centroids will then replace each image patch. Combining patches back into images consists of placing the patches in their original position and averaging between the overlapping pixels. This strategy may not be ideal, but it will show how well the dictionary is able to preserve shaping information. Figures 6 through 10 display the results of this reconstruction, and as you can see, most of these figures show nearly identical images. However, figure 6 appears to be a blurred version of the original image; this can be explained as an averaging between patches of similar but different shape.

Conclusion:

The time required to train the image dictionaries was a surprising problem. With the smallest size considered, the training time varied between 15 and 20 minutes. As the number of centroids increased, the training time increased to hours. For a dictionary of size 5000, this required an overnight training, and the 10,000 size required an entire day. This constraint reduced the available time for exploration, but the assignment goal could still be fulfilled.

For the number of clusters needed to represent the image shapes, it seems that the dictionary size 500 could reconstruct the images well. This can be seen in figure 7, and figure 6 appears to have blurring images with the small image dictionary. Increasing the number of centroids didn't seem to have vast improvements after 500 centroids, so this could be considered close to covering the entire image patch space.

The interpretable meaning behind the centroids would be the estimated shape between many patches. Some patches can be very similar, these patches would most likely be within the same cluster. A cluster can be considered as a given shape with slight variations, but as the number of centroids increases, there could be multiple clusters that are nearly the same. Digits can be formed from these clusters as each cluster should be close to another. Looking at the dictionaries in figures 1 through 5, most patches appear to be lines, curves, or a small dash. When considering how digits are written, every line in a patch should be connected to another line, whether curved or shortened. Distance between clusters shouldn't be large in this case since one cluster will need to lead to the next.

Appendix:

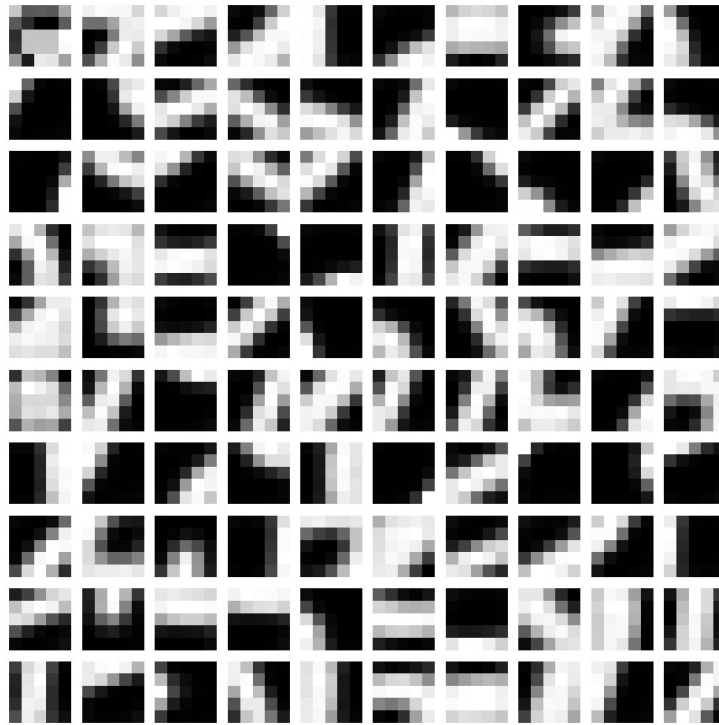


Fig. 1: Image Dictionary of size 100

Fig. 2: Image Dictionary of size 500

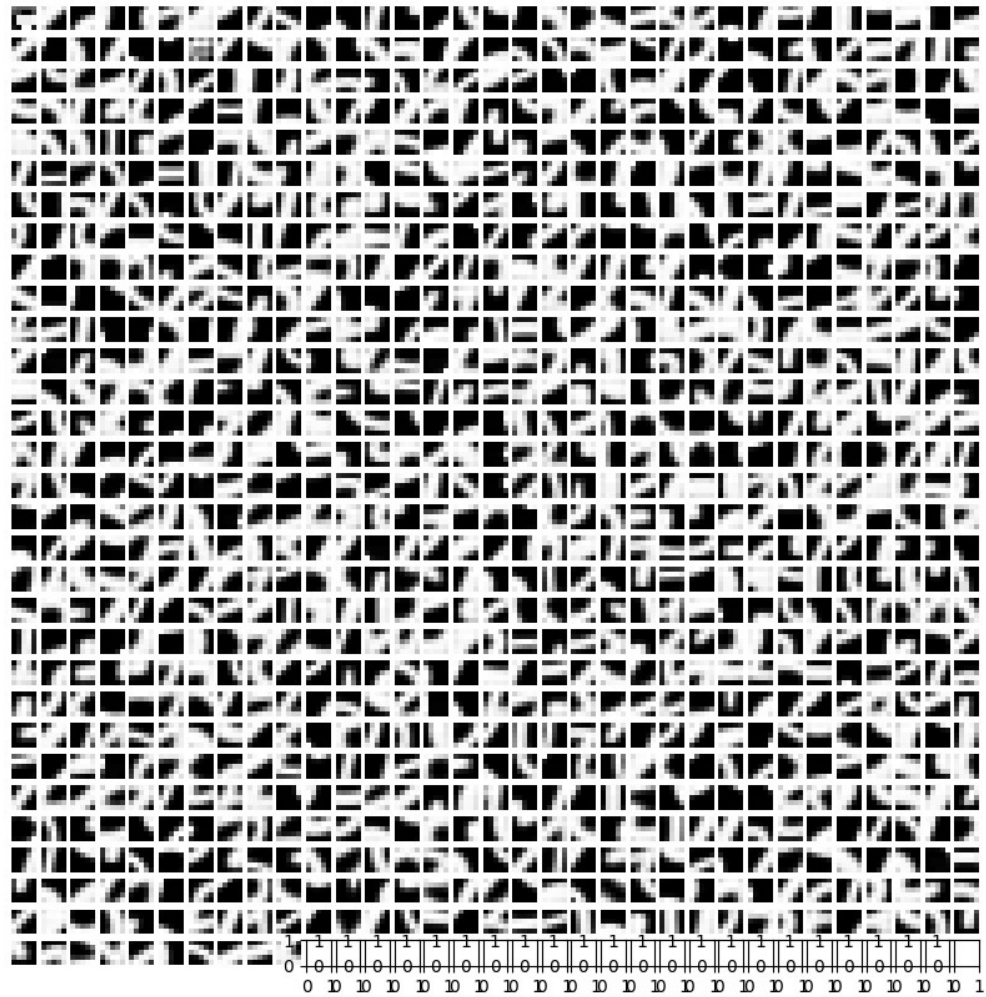


Fig. 3: Image Dictionary of size 1000

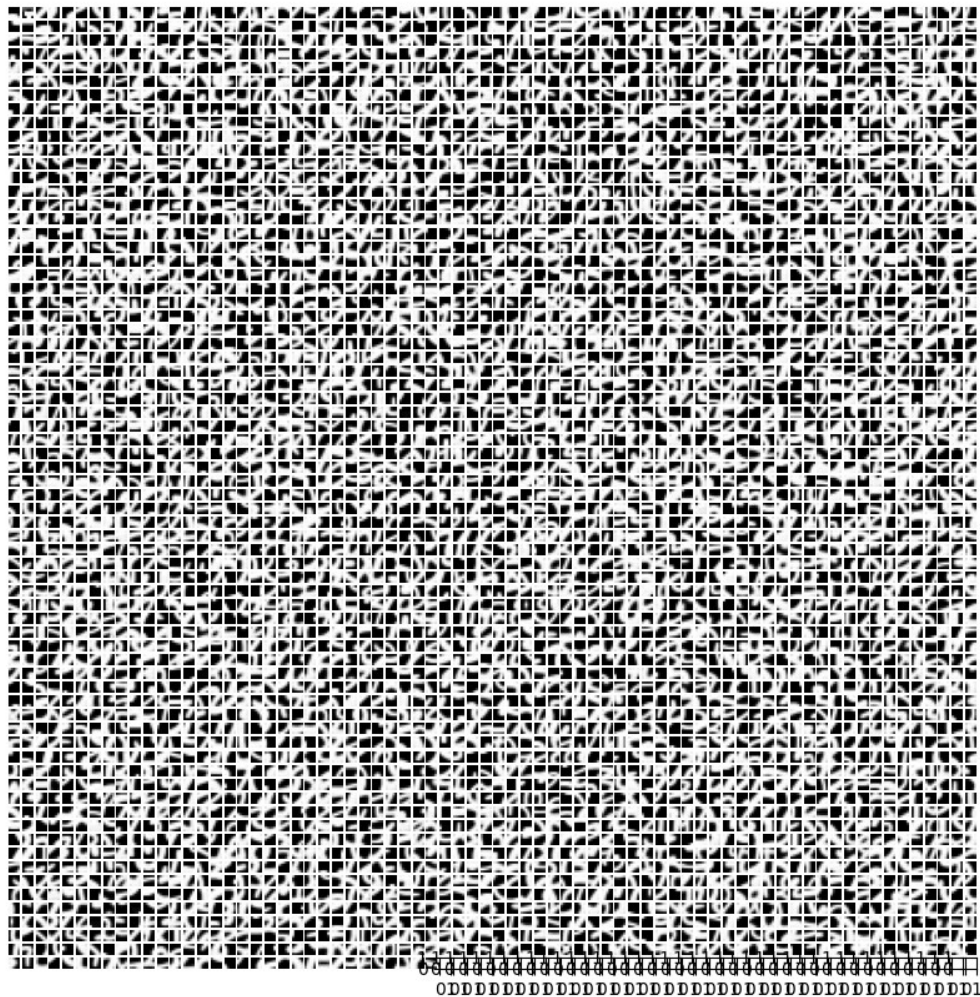


Fig. 4: Image Dictionary of size 5000

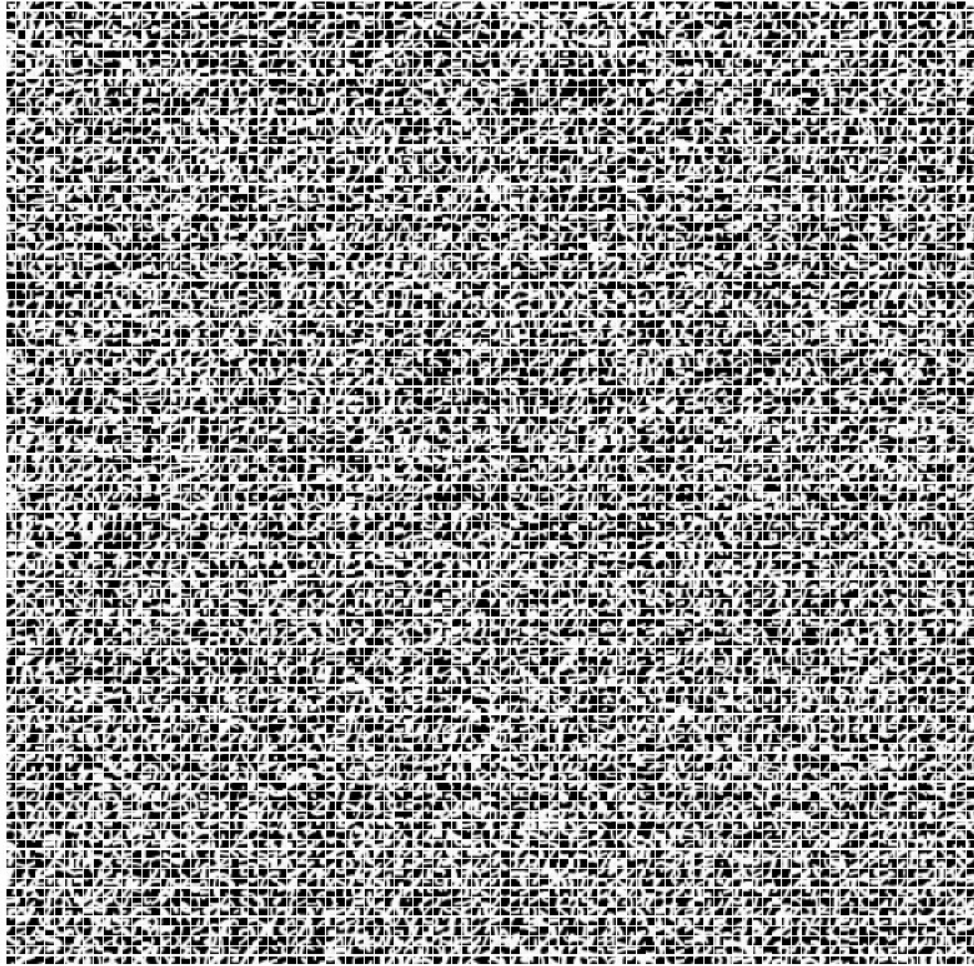


Fig. 5: Image Dictionary of size 10000

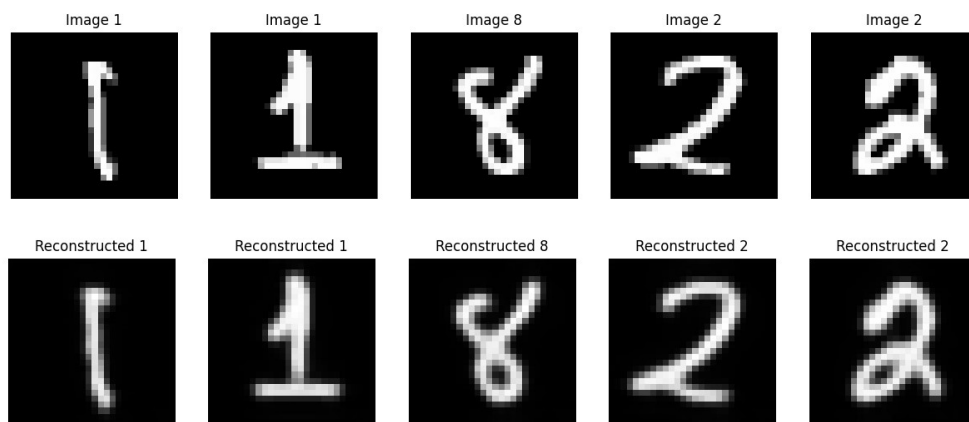


Fig. 6: Reconstructed Images from Dictionary size 100

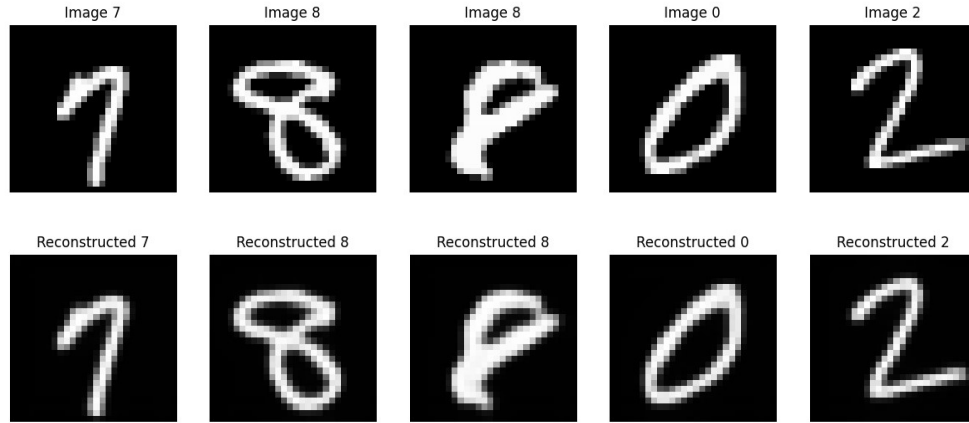


Fig. 7: Reconstructed Images from Dictionary size 500

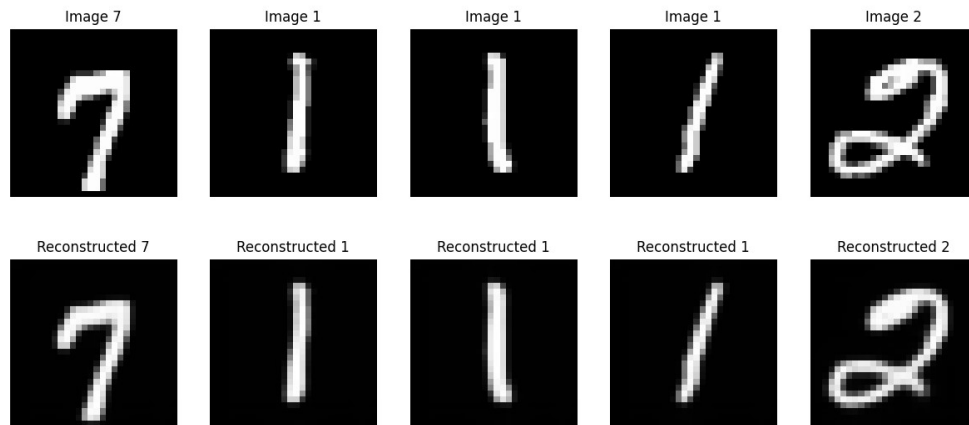


Fig. 8: Reconstructed Images from Dictionary size 1000

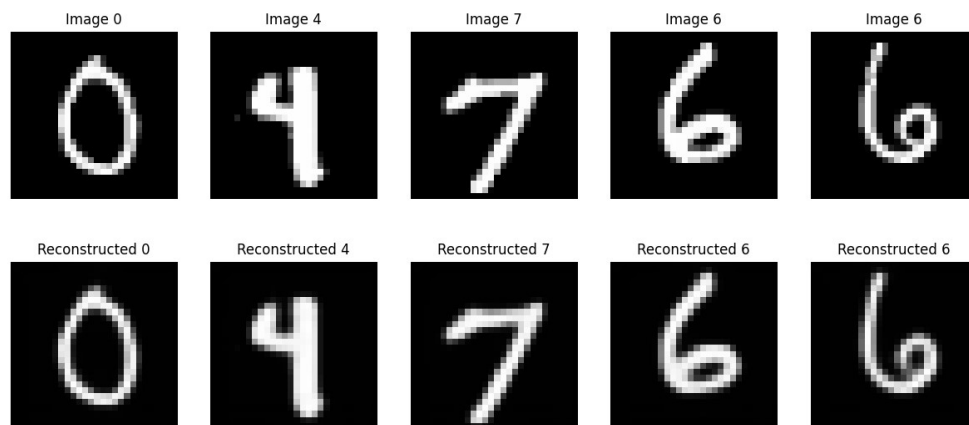


Fig. 9: Reconstructed Images from Dictionary size 5000

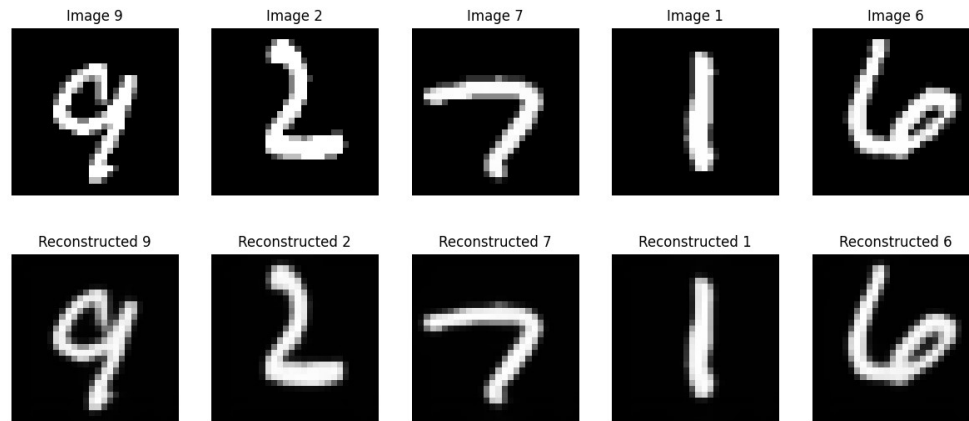


Fig. 10: Reconstructed Images from Dictionary size 10000

References:

- [1] David Arthur and Sergei Vassilvitskii. 2007. K-means++: the advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '07). Society for Industrial and Applied Mathematics, USA, 1027–1035.
- [2] Chen, Yubei, (2024). “The Principles – A Discrete Perspective,” Lecture Notes.
- [3] Github Repository: <https://github.com/Th3RandyMan/K-Mean-Clustering-MNIST>