Objectives

1. Practice the use of dimensionality reduction as a preprocessing step and study its impact on classification performance.
2. Gain an in-depth understanding of unsupervised learning algorithms by formulating an image compression problem in this context.

Guide

Dimensionality Reduction

Implement Fisher's Linear Discriminant (FLD)
- Objective: Perform supervised dimensionality reduction using class label information.

Implement Principal Component Analysis (PCA)
- Objective: Apply unsupervised dimensionality reduction to MNIST dataset and find the lowest dimension achieving less than 10% error rate..

Impact of Dimensionality Reduction on Classification
- Objective: Analyze the effect of dimensionality reduction (FLD and PCA) on classification performance.

Unsupervised Learning

k-Means and Winner-Take-All (WTA) Clustering
- Objective: Perform image compression by clustering RGB values of a flower image using k-Means and WTA clustering measuring by RMSE while testing different k-values.

Hierarchical Clustering
- Objective: Solve the image compression problem using hierarchical clustering and compare performance to other models.

Summary of Results

- Dimensionality Reduction: FLD+PCA demonstrated better class separability than PCA alone.
- Classification Performance: Reduced datasets (fX and pX) achieved significant computation time reductions with minimal impact on accuracy.

- Image Compression: k-Means offered the fastest and most consistent results, while hierarchical clustering provided better quality for larger clusters.

File Structure

- `src/`: Source code for all implementations.
- `data/`: MNIST and flower image datasets.
- `notebooks/`: Jupyter Notebooks for analysis and visualization.
- `results/`: Generated graphs, tables, and reports.

How to Run

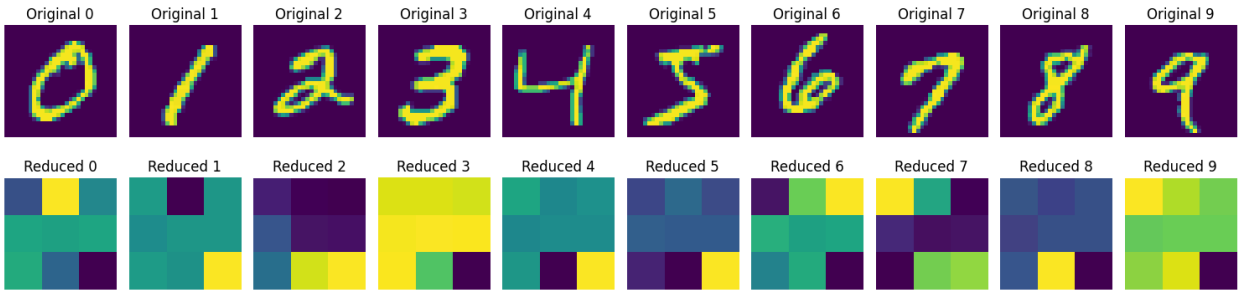
1. Clone this repository.
2. Install required dependencies (listed in `requirements.txt`).
3. Run the provided Jupyter Notebooks or Python scripts in the `src/` directory for individual tasks.

License

Report Begins On Next Page

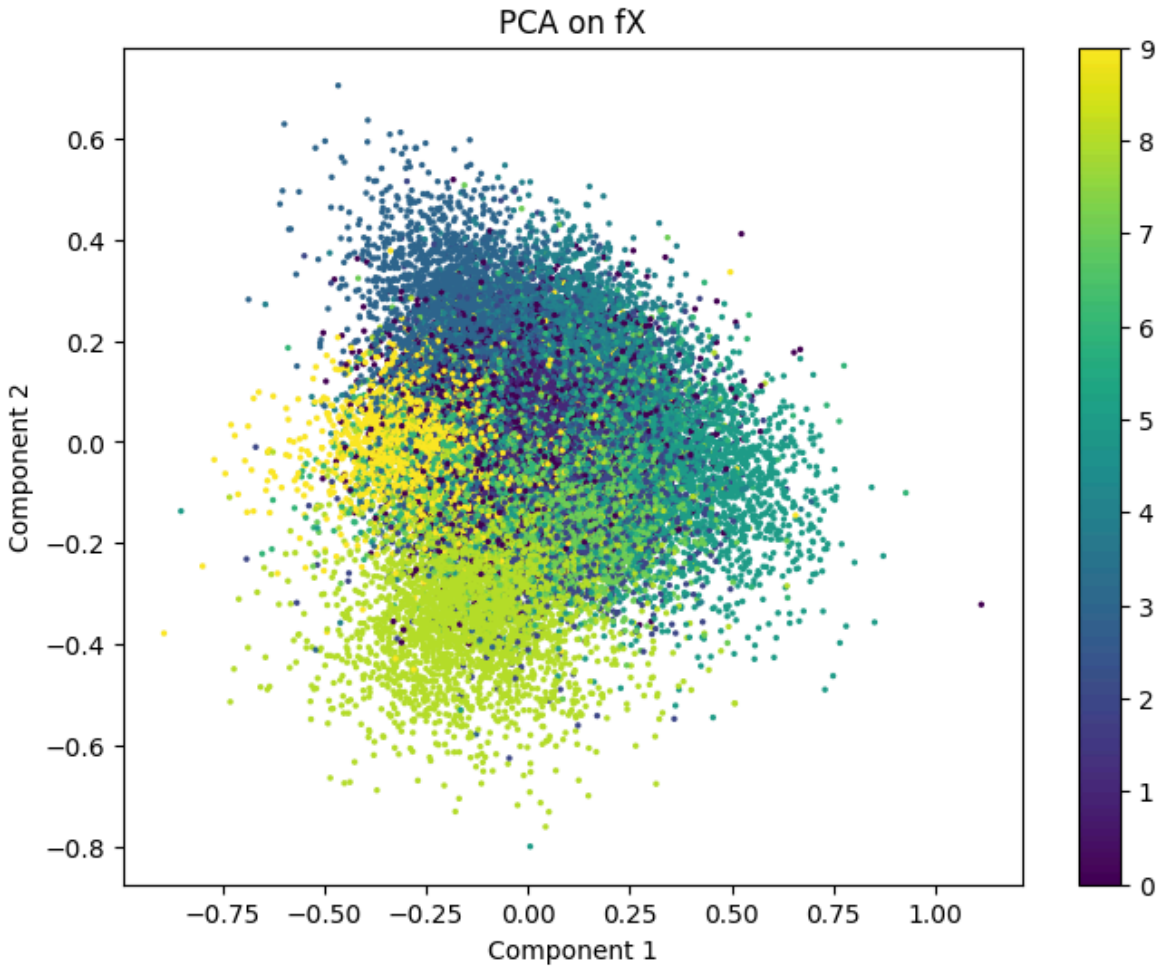
Dimensionality Reduction

FLD on images to 9 dimensions

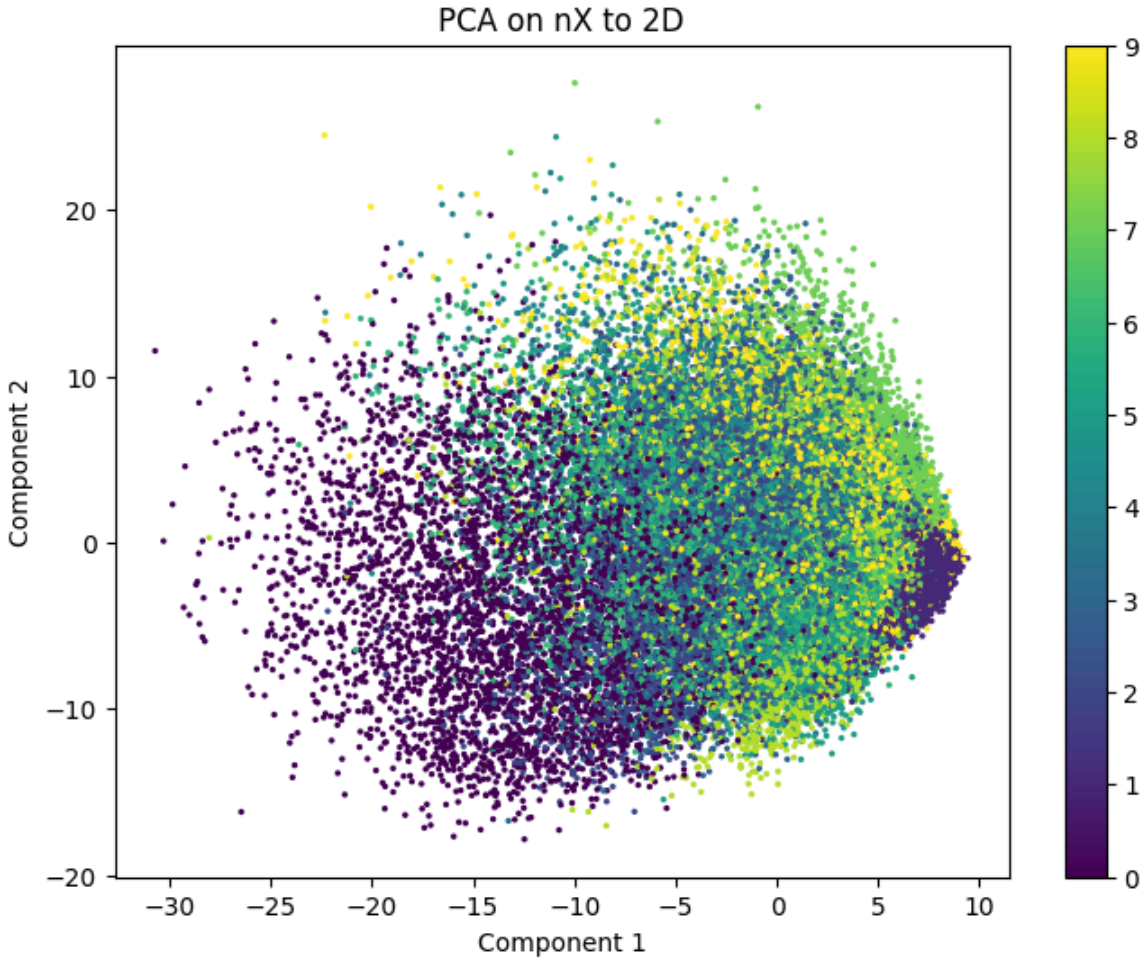


PCA

5. Error Rate: 0.62378



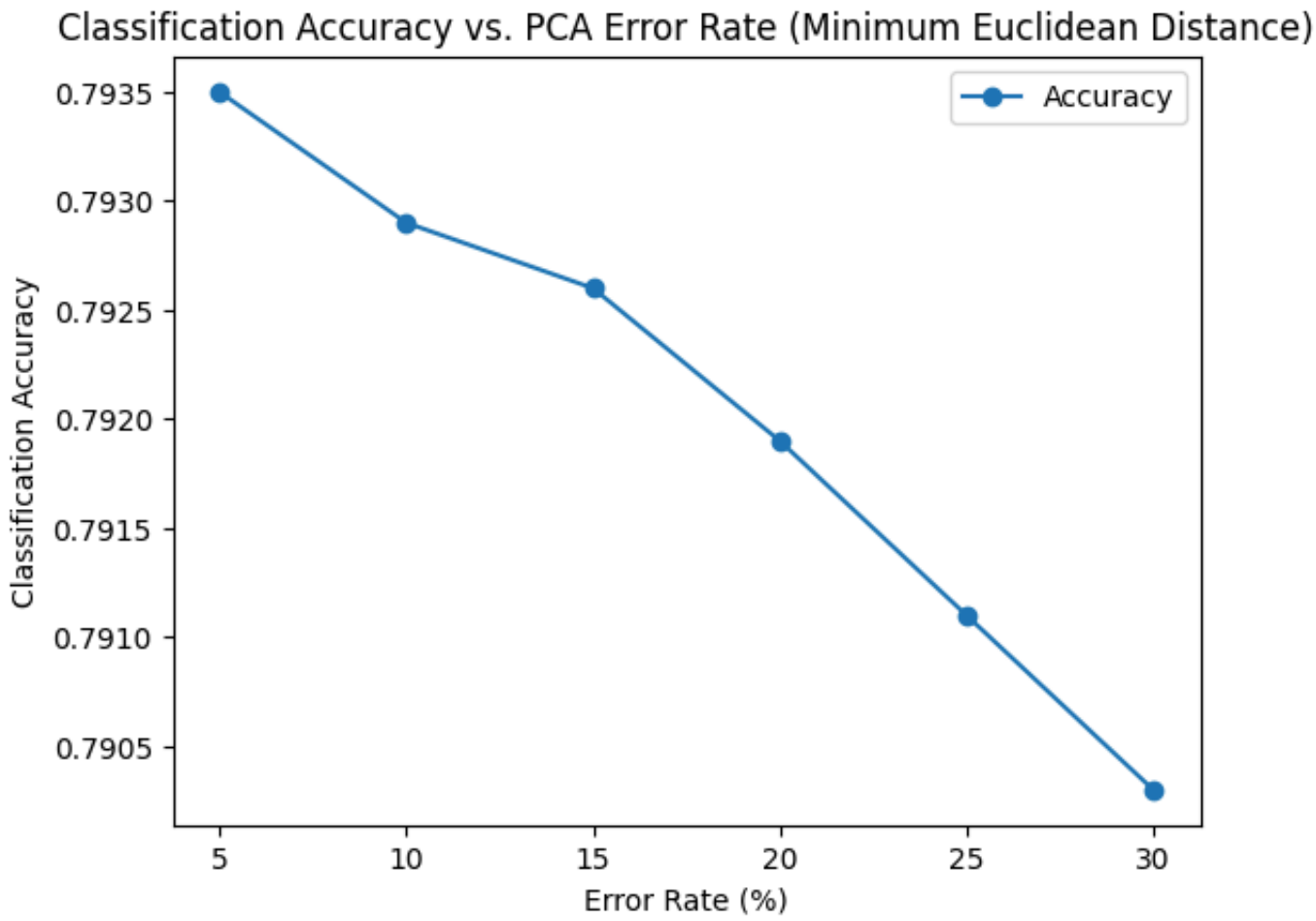
6. Error Rate: 0.90316055

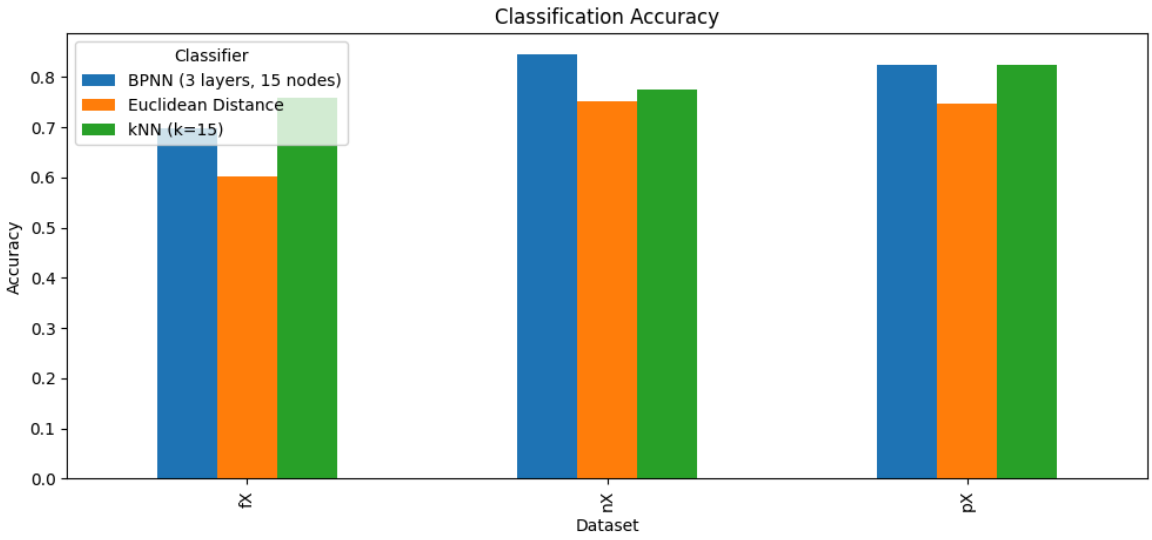


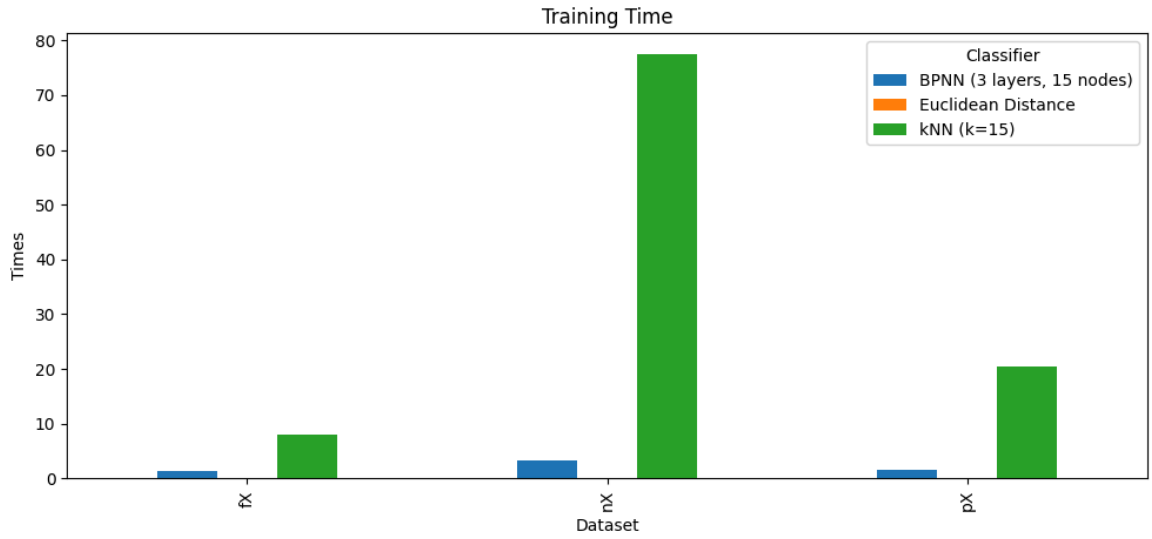
7. In FLD+PCA, we can see better class separability as classes appear more distinct, enabling clustering. With PCA alone, there is more overlap between classes, as PCA maximizes variance globally, not for each class. This highlights how reduction like FLD+PCA can improve visualization or separability over methods like PCA by itself.

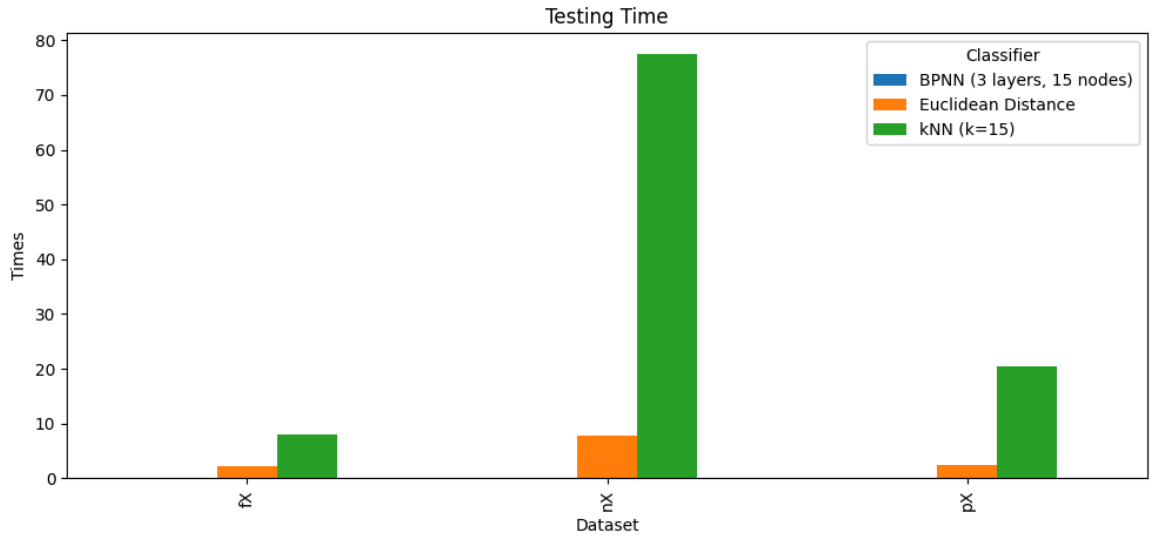
8. Dimension for 10% Error Rate: 235

Impact of Dimensionality Reduction on Classification




Classification Accuracy
Classifier
BPNN (3 layers, 15 nodes)
Euclidean Distance
kNN (k=15)
Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
fX
nX
pX
Dataset


Training Time
Classifier
BPNN (3 layers, 15 nodes)
Euclidean Distance
kNN (k=15)
Times
0
10
20
30
40
50
60
70
80
fX
nX
pX
Dataset

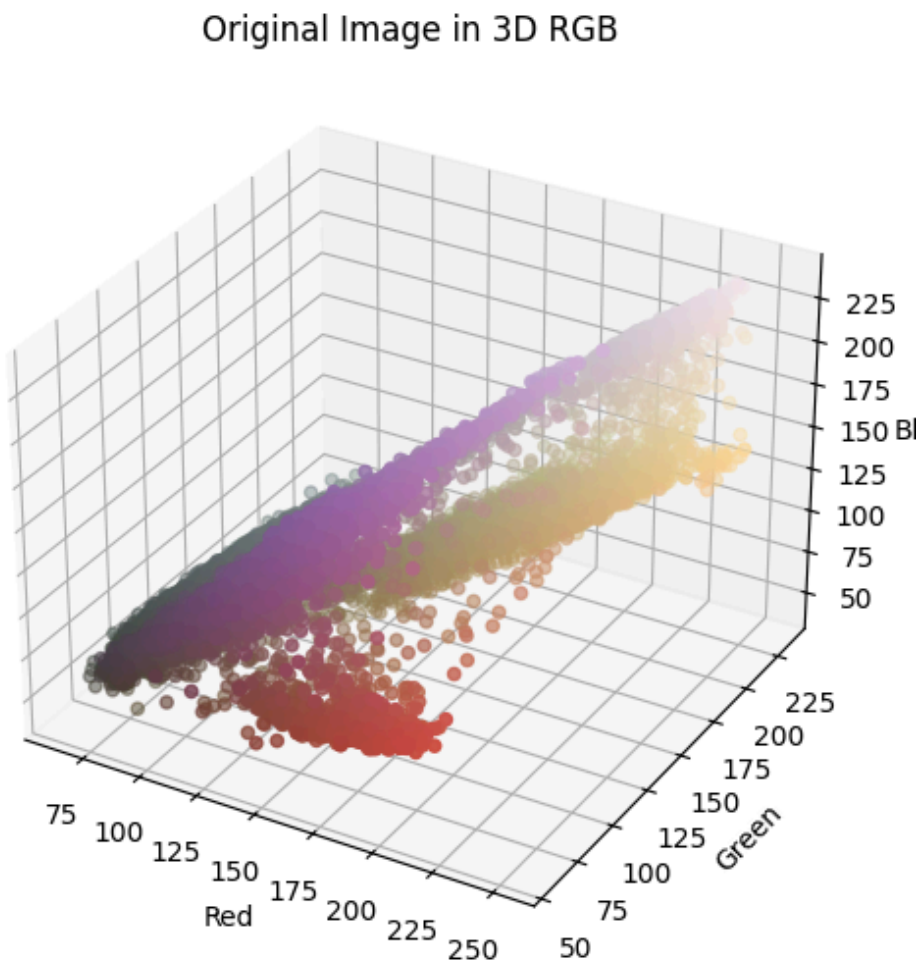

Testing Time
Classifier
BPNN (3 layers, 15 nodes)
Euclidean Distance
kNN (k=15)
Times
0
10
20
30
40
50
60
70
80
fX
nX
pX
Dataset

Classification Performance:

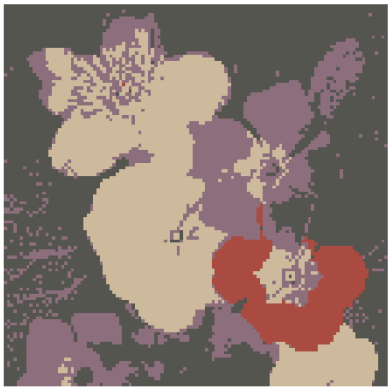
	Classifier	Dataset	Accuracy	Train Time	Test Time
0	Euclidean Distance	nX	0.755420	0.014987	7.699192
1	Euclidean Distance	fX	0.870971	0.003245	4.171664
2	Euclidean Distance	pX	0.750232	0.002201	3.473627
3	kNN (k=15)	nX	0.779551	362.254604	362.254604
4	kNN (k=15)	fX	0.875449	39.782191	39.782191
5	kNN (k=15)	pX	0.829058	57.896135	57.896135
6	BPNN (3 layers, 15 nodes)	nX	0.838667	4.547128	0.630827
7	BPNN (3 layers, 15 nodes)	fX	0.870551	1.218038	0.082589
8	BPNN (3 layers, 15 nodes)	pX	0.820652	1.401997	0.058200

Unsupervised Clustering

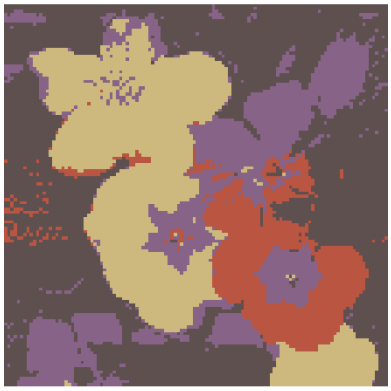
K-Means + WTA Clustering



K-means (k=4)
RMSE: 19.70, Time: 0.28s



WTA (k=4)
RMSE: 26.75, Time: 24.27s



K-means (k=16)
RMSE: 7.24, Time: 0.79s

WTA (k=16)
RMSE: 9.83, Time: 25.63s

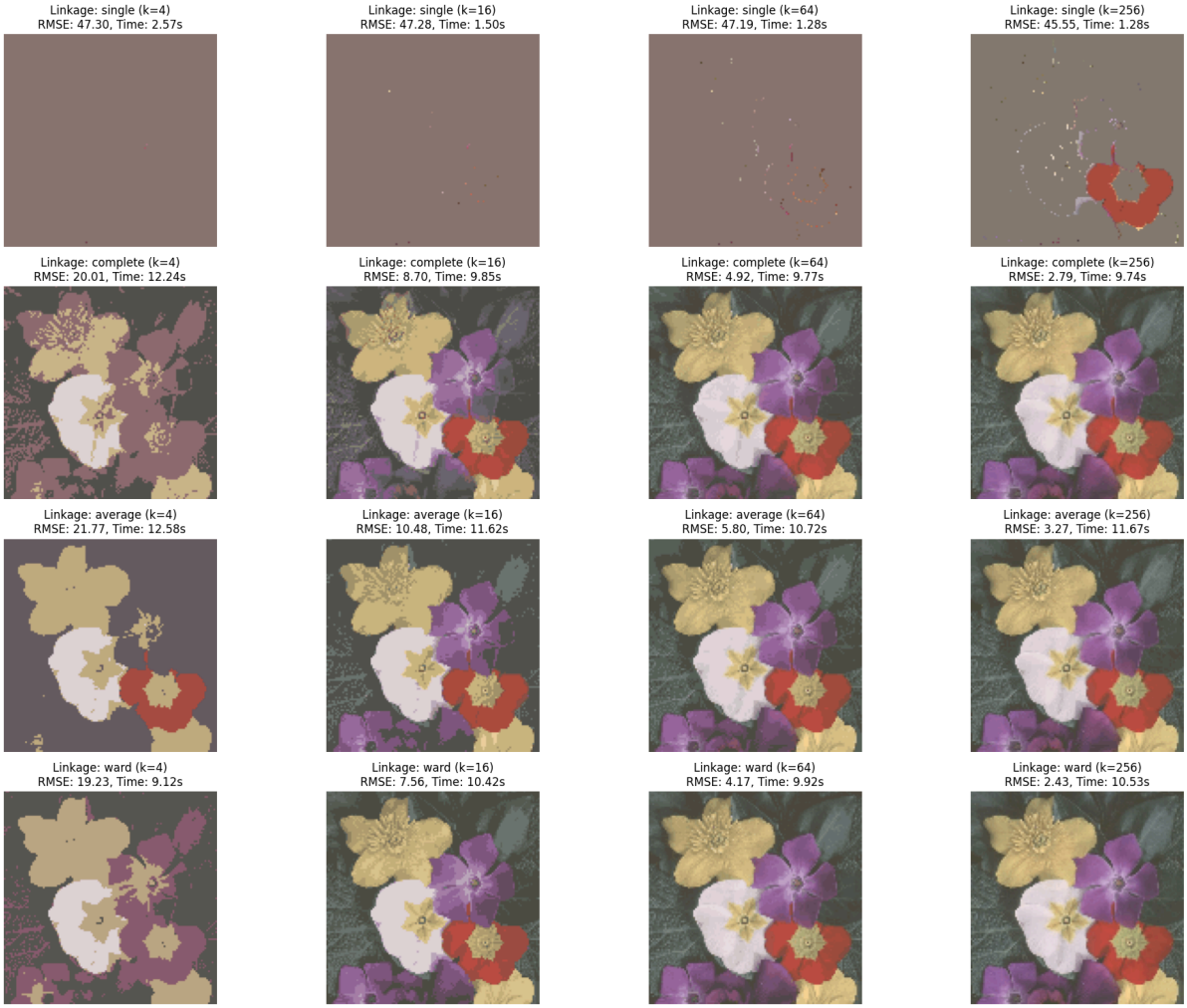
K-means (k=64)
RMSE: 4.15, Time: 2.97s

WTA (k=64)
RMSE: 4.57, Time: 28.56s

K-means (k=256)
RMSE: 2.56, Time: 10.58s

WTA (k=256)
RMSE: 2.48, Time: 38.92s

Hierarchical clustering + comparison



I added all linkage methods to the grid to display their effects as the “best” may be subjective. Ward starts the sharpest however, others are faster. (I also wanted to demonstrate how bad single was)

16. Image Quality: k-means provides clear decent compression at low k-values and sharpens steadily, while WTA adapts quickly but may be less stable and less clear at lower k values. Hierarchical clustering captures complex structures but may be less clear at larger k values.
Run-Time: k-means is fastest by far because it uses an iterative approach that quickly converges, and WTA is slowest by far because it updates weights incrementally for each point, requiring more iterations to converge with Hierarchical being in-between due to repeated distance calculations.
RMSE: RMSE’s across the models are pretty similar by the time they reach a k of 256 however, starting off WTA is the clear loser with k-means and Hierarchical coming close with Ward Hierarchical taking a slight edge victory.