1 MAP Estimation

1. $\log\left(p(w_j)\right) = \log\left(\frac{\lambda}{2}\right) - \lambda\left|w_j\right|$

$$-\log(p(w)) = -\sum_{j=1}^{d} \log\left(p(w_j)\right) = constant + \sum_{j=1}^{d} \lambda\left|w_j\right|$$

Replacing the regularization term with $\lambda\|w\|_1$

2. $\log(p(y_i|x_i, w)) = \log\left(\frac{1}{2}\right) - |w^T x_i - y_i|$

$$-\log(p(y|X, w)) = constant + \sum_{i=1}^{n} |w^T x_i - y_i|$$

Replacing the training term with $\|Xw - y\|_1$

3. $\log(p(y_i|x_i, w)) = \log\left(\frac{1}{\sqrt{2\sigma^2\pi}}\right) - \frac{(w^T x_i - y_i)^2}{2\sigma^2}$

$$-\log(p(y|X, w)) = constant + \sum_{i=1}^{n} \frac{(w^T x_i - y_i)^2}{2\sigma^2}$$

Replacing the training term with $\frac{1}{2\sigma^2}\|Xw - y\|_2^2$

4. $\log(p(y_i|x_i, w)) = \log\left(\frac{1}{\sqrt{2\sigma_i^2\pi}}\right) - \frac{(w^T x_i - y_i)^2}{2\sigma_i^2}$

$$-\log(p(y|X, w)) = constant + \sum_{i=1}^{n} \frac{(w^T x_i - y_i)^2}{2\sigma_i^2}$$

Replacing the training term with $\frac{1}{2}(Xw - y)^T Z(Xw - y)$ where Z is a matrix with $\frac{1}{\sigma_i^2}$ terms on the diagonal

$$5. \log\big(p(y_i|x_i,w)\big) = \log\left(\frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{v\pi}\,\Gamma\left(\frac{v}{2}\right)}\right) - \left(\frac{v+1}{2}\right)\log\left(1+\frac{(w^T x_i - y_i)^2}{v}\right)$$

$$-\log\big(p(y|X,w)\big) = \ constant + \left(\frac{v+1}{2}\right)\sum_{i=1}^{n}\log\left(1+\frac{(w^T x_i - y_i)^2}{v}\right)$$

Replacing the training term with $\left(\frac{v+1}{2}\right)\sum_{i=1}^{n}\log\left(1+\frac{(w^T x_i - y_i)^2}{v}\right)$

The loss coming from the student t distribution is very robust because it depends on the log of the squared loss instead of just the squared or absolute loss.

2 Naïve Bayes

2.1 Naïve Bayes by Hand

(a)

$$p(y=1) = \frac{6}{10} = \frac{3}{5}$$

$$p(y=0) = \frac{4}{10} = \frac{2}{5}$$

(b)

$$p(x_1=1|y=1) = \frac{3}{6} = \frac{1}{2}$$

$$p(x_2=1|y=1) = \frac{4}{6} = \frac{2}{3}$$

$$p(x_1=1|y=0) = \frac{4}{4} = 1$$

$$p(x_2=1|y=0) = \frac{1}{4}$$

(c)

$$\begin{aligned} p(y=0|x_1=1,x_2=1) \ &\propto \ p(x_1=1,x_2=1|y=0)\,p(y=0) \\ &= p(x_1=1|y=0)\,p(x_2=1|y=0)\,p(y=0) \\ &= 1\left(\frac{1}{4}\right)\left(\frac{2}{5}\right) = \frac{1}{10} \end{aligned}$$

$$\begin{aligned} p(y=1|x_1=1,x_2=1) \ &\propto \ p(x_1=1,x_2=1|y=1)\,p(y=1) \\ &= p(x_1=1|y=1)\,p(x_2=1|y=1)\,p(y=1) \\ &= \left(\frac{1}{2}\right)\left(\frac{2}{3}\right)\left(\frac{3}{5}\right) = \frac{1}{5} \end{aligned}$$

Since $p(y=1|x_1=1,x_2=1) > p(y=0|x_1=1,x_2=1)$, the label for $\hat{x}$ is 1.

2.2 Naïve Bayes Implementation

Code: https://github.ubc.ca/cpsc340-2017S/sopida_zhenxil_a4/tree/master/code/naive_bayes.py

Random Forest Validation error: 0.202

Naive Bayes Validation error: 0.188

2.3 Runtime of Naïve Bayes for Discrete Data

For each test example, we are multiplying $p(x_{ij}|y_i)$ for each d feature in each k class which costs $O(kd)$. Hence, the cost for classifying t test examples is $O(tkd)$.

3 Principal Component Analysis

3.1 PCA by Hand

1. Since PCA assume features have a mean of 0,
   Mean of x1 is 0, which is ok.

   Mean of x2 is 1, so we need to subtract each x2 with 1.

   x1: -2 -1 0 1 2

   x2: -2 -1 0 1 2

   In this case, $X \approx ZW$ with d=2 and k =1.

   Z is (n*1) and W is (1*2)

   In order to find the principal component W, we need a PCA that maximizes the variance after projection. So we pick the line x2 = x1 where the gradient w is (1, 1).

   Since $||w|| = 1$ and the first principal component w is $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$.

2. Point (3,3)
   The mean for (x_1, x_2) is (0, 1)

   Principal component w is $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$

   Since $||w|| = 1, Z = \hat{X}W^T$

   So $z = xw^T = \begin{bmatrix}(3-0)\\(3-1)\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix} = \frac{5}{\sqrt{2}}$

$$f(W,Z) = ||ZW - X||_F^2 = (w_1^T z_1 - x_1)\text{^}2 + (w_2^T z_2 - x_2)\text{^}2$$

$$= \left(\frac{1}{\sqrt{2}} * \frac{5}{\sqrt{2}} - (3-0)\right)\text{^}2 + \left(\frac{1}{\sqrt{2}} * \frac{5}{\sqrt{2}} - (3-1)\right)\text{^}2 = \frac{1}{2}$$

   Therefore, the reconstruction error is $\frac{1}{\sqrt{2}}$.

3. Point (3,4)

$$z = xw^T = \begin{bmatrix}(3-0)\\(4-1)\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix} = \frac{6}{\sqrt{2}}$$

$$f(W,Z) = \|ZW - X\|_F^2 = (w_1^T z_1 - x_1)\text{\textasciicircum}2 + (w_2^T z_2 - x_2)\text{\textasciicircum}2$$

$$= (\frac{1}{\sqrt{2}} * \frac{6}{\sqrt{2}} - (3-0))\text{\textasciicircum}2 + (\frac{1}{\sqrt{2}} * \frac{6}{\sqrt{2}} - (4-1))\text{\textasciicircum}2 = 0$$

Therefore, the reconstruction error is 0.

3.2 Data Visualization

Code: https://github.ubc.ca/cpsc340-2017S/sopida_zhenxil_a4/tree/master/code/main.py

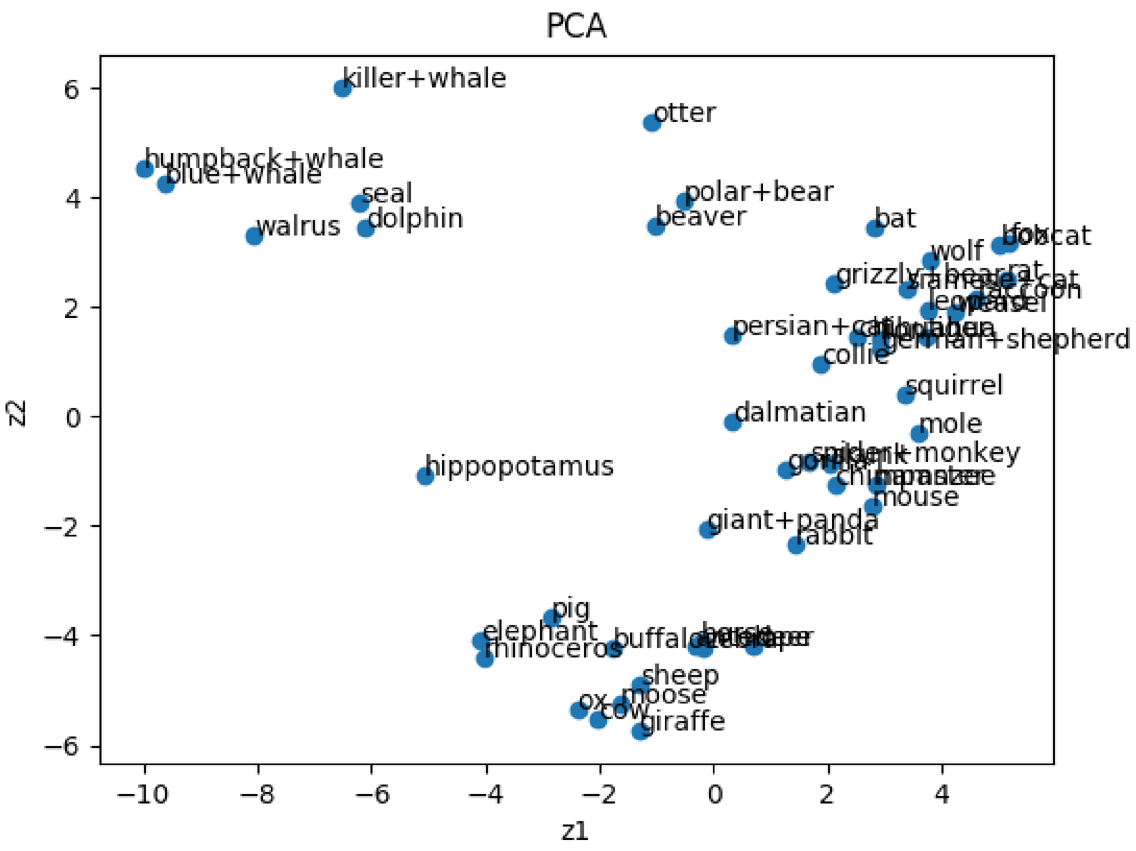


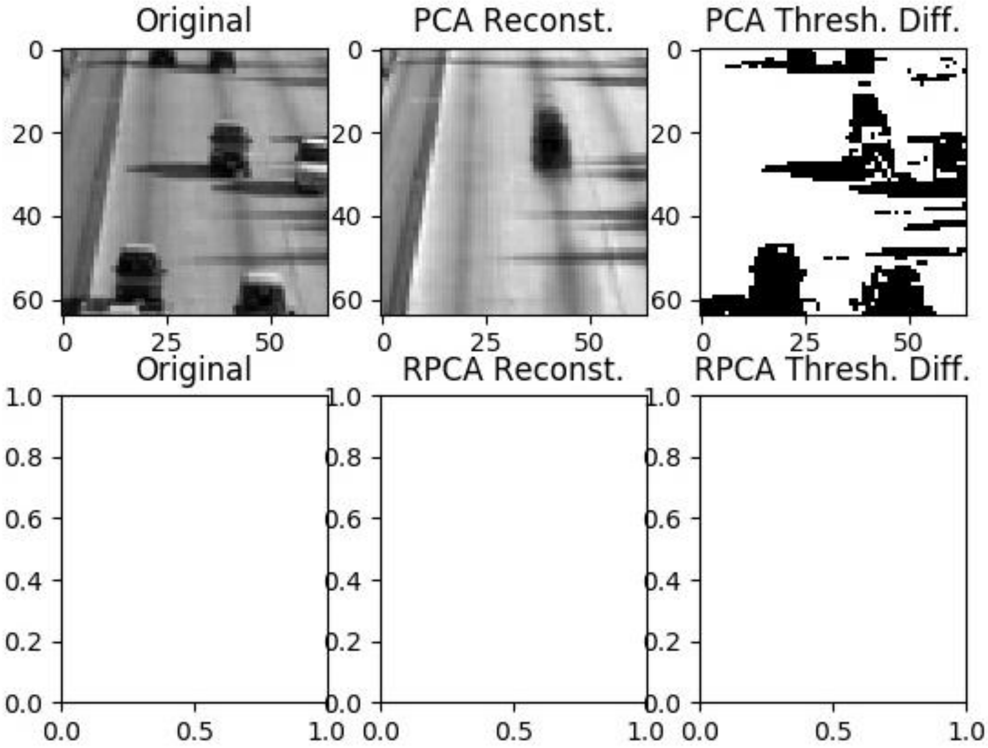
Fig: https://github.ubc.ca/cpsc340-2017S/sopida_zhenxil_a4/tree/master/figs/Q3.2_PCA.png

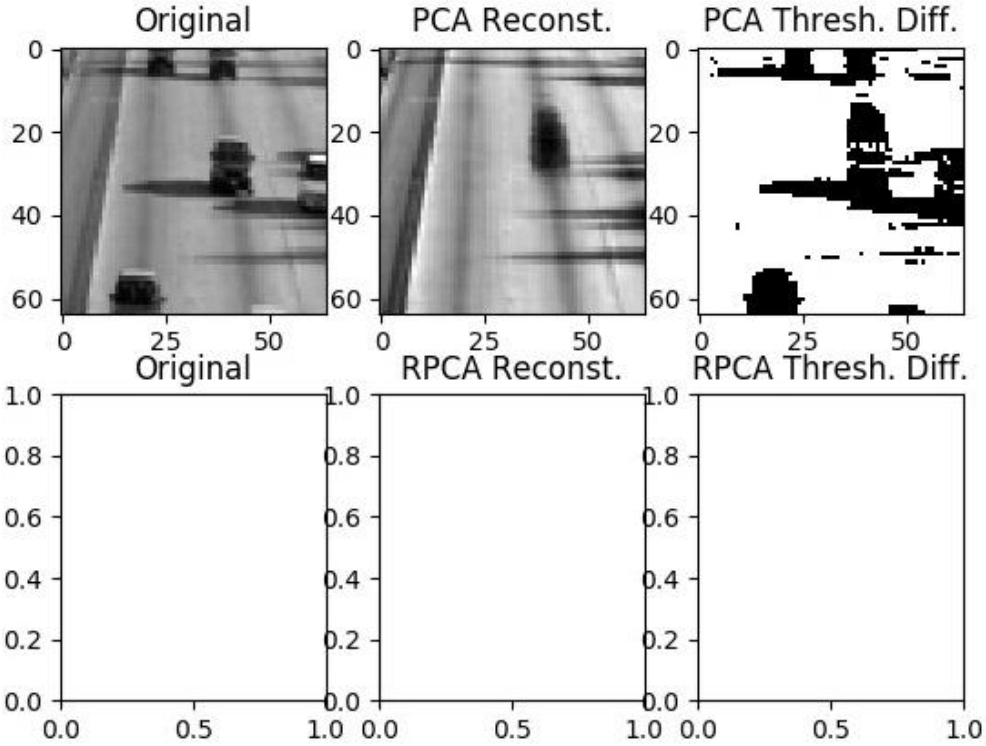
3.3 Data Compression

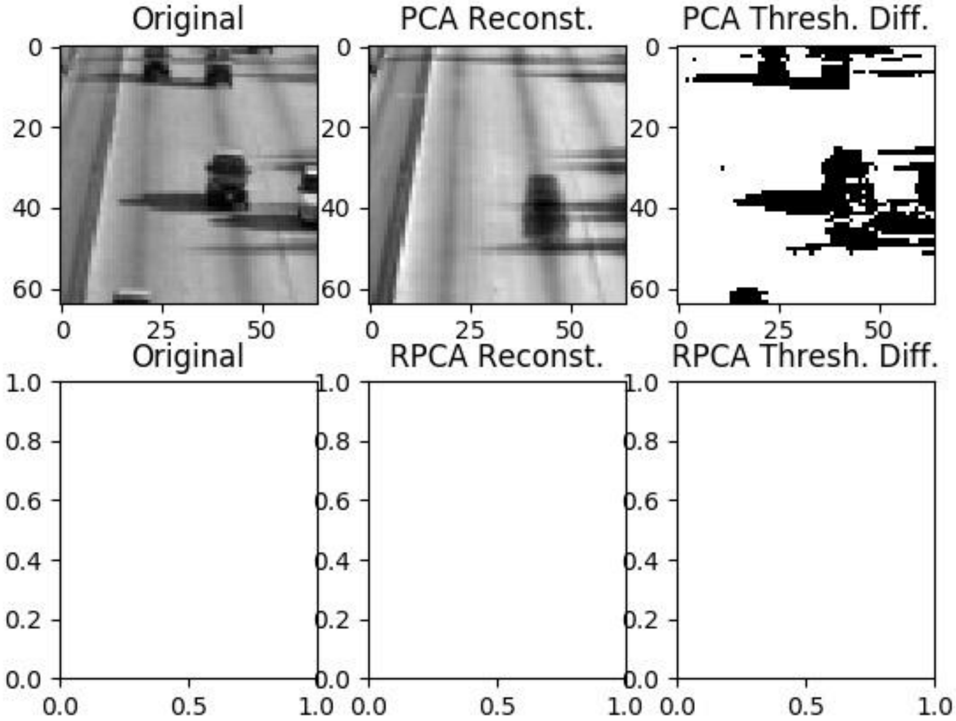
1. When k=2, variance is explained by: 30.19%.
2. When k=5, variance is explained by: 50.59%. Therefore, to explain 50% of the variance, k must be greater than 4.

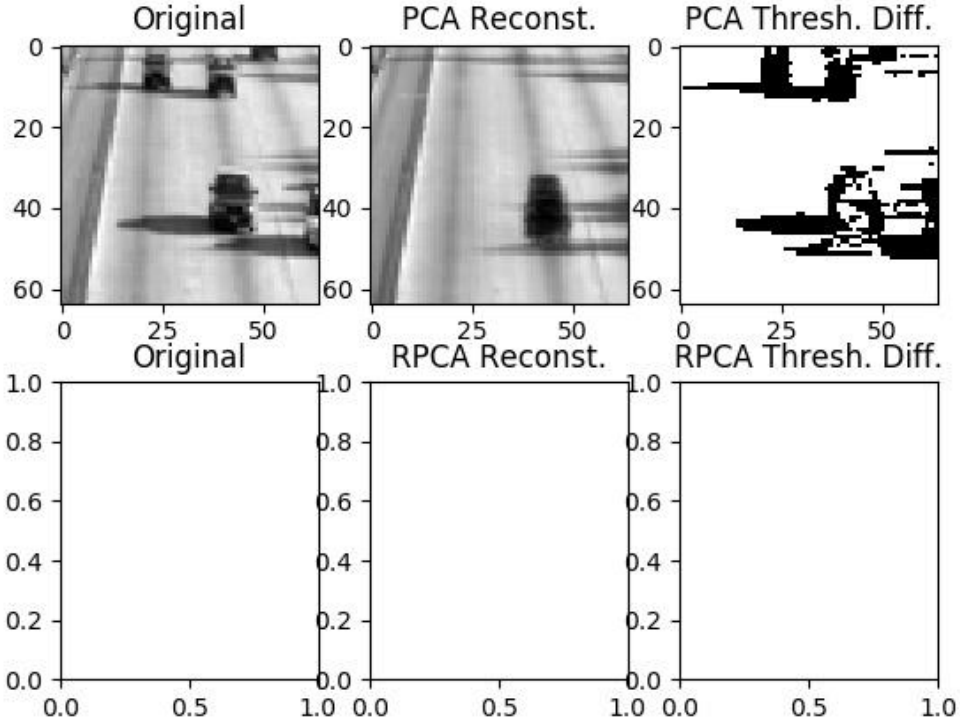
4 Robust PCA

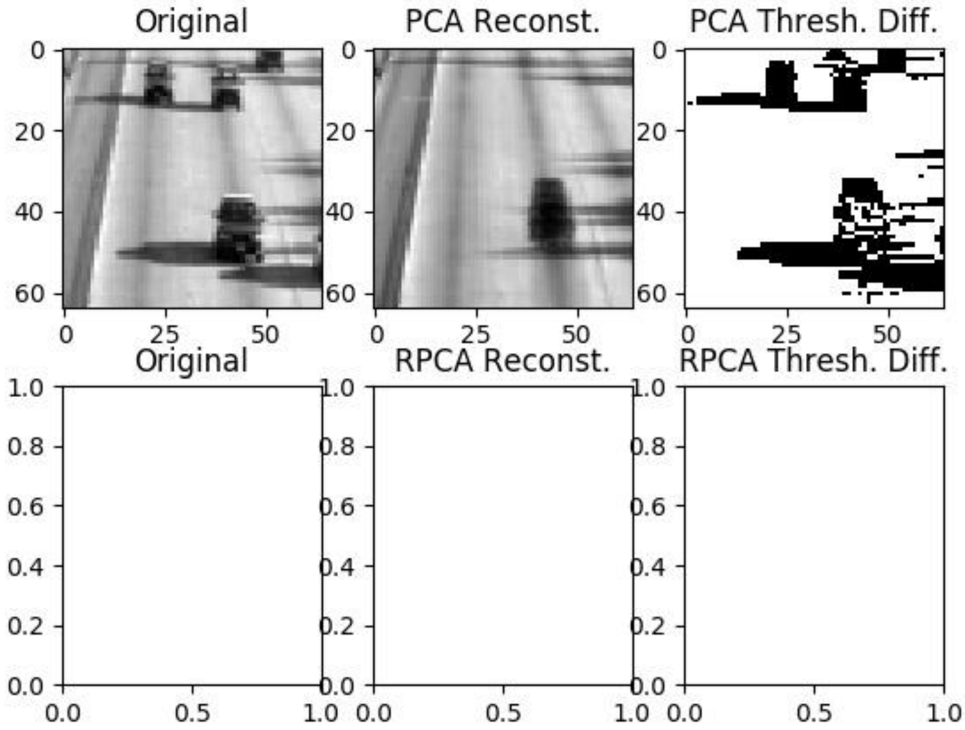
Code: https://github.ubc.ca/cpsc340-2017S/sopida_zhenxil_a4/tree/master/code/pca.py

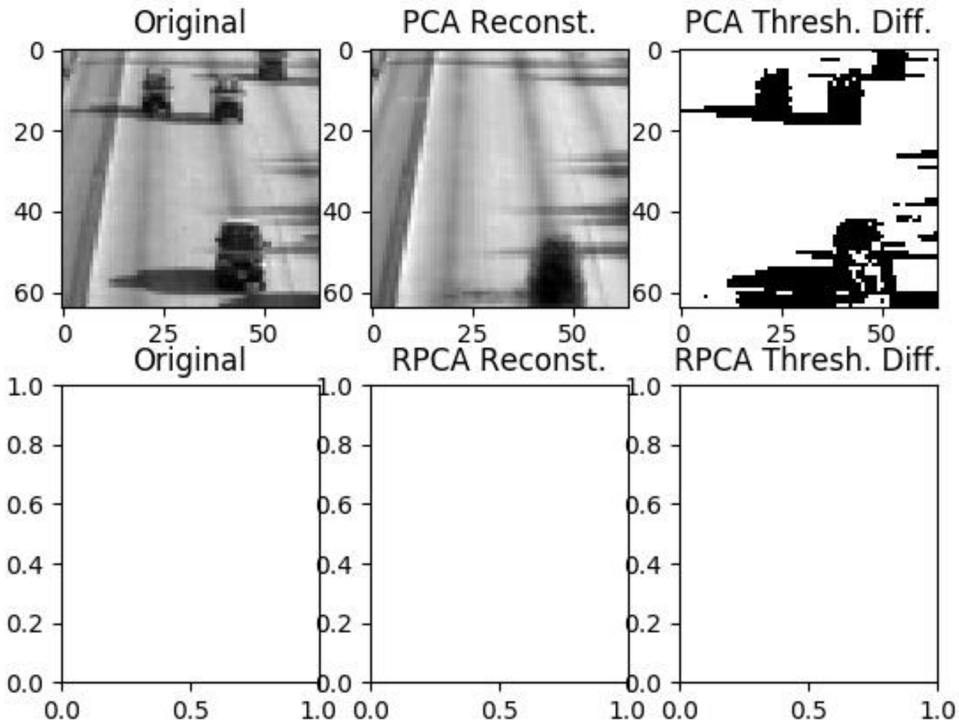


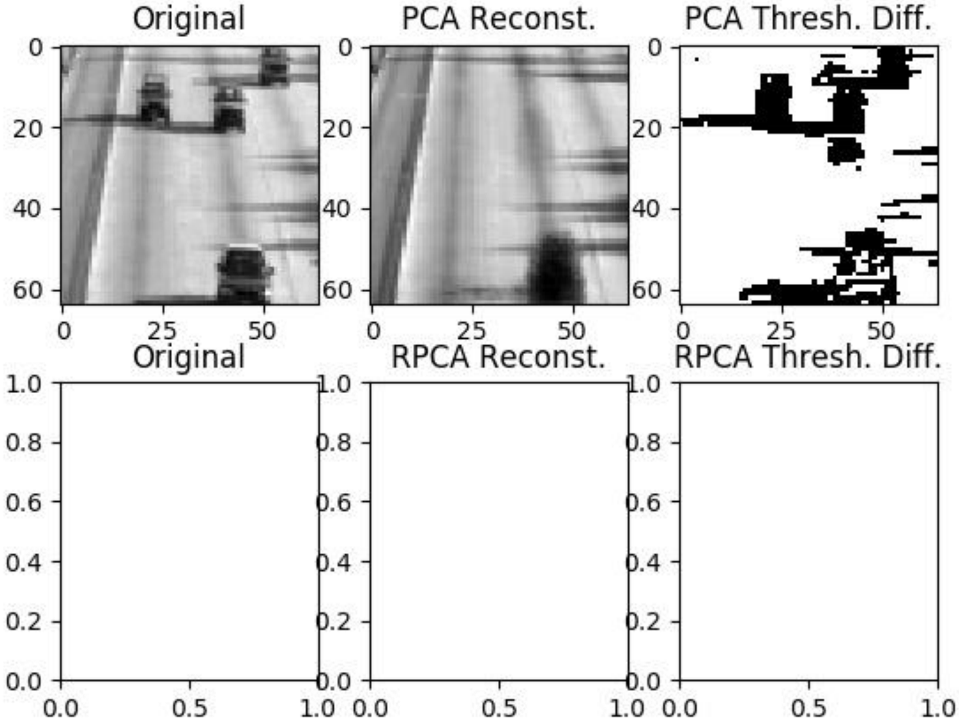


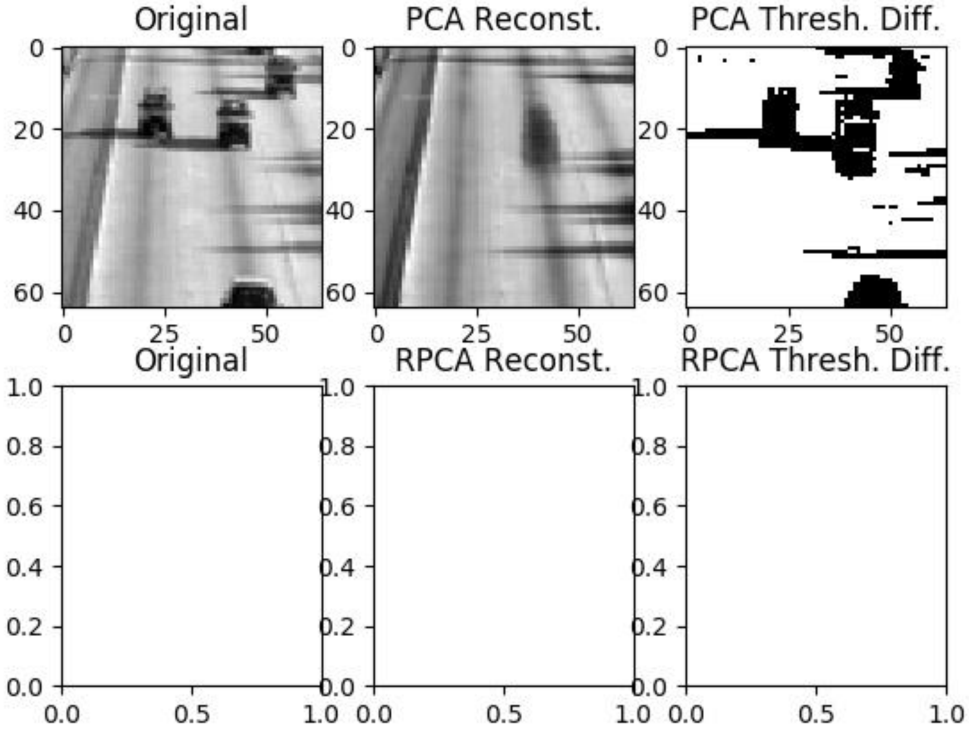

Original
PCA Reconst.
PCA Thresh. Diff.
0
20
40
60
0
25
50
Original
RPCA Reconst.
RPCA Thresh. Diff.
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0

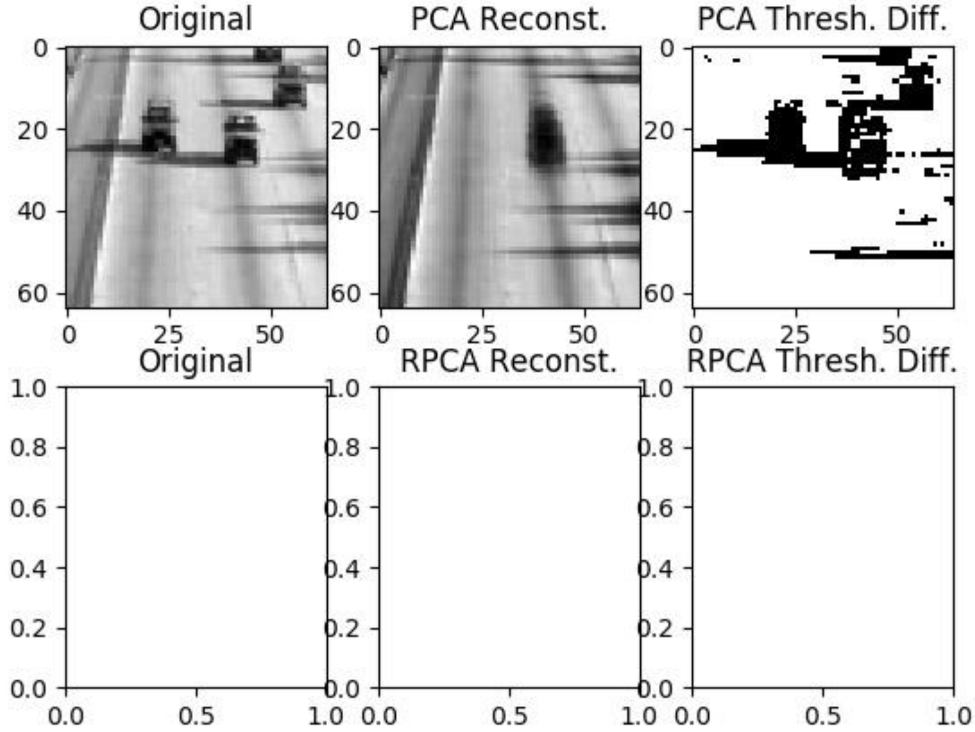

Original
PCA Reconst.
PCA Thresh. Diff.
0
20
40
60
0
25
50
Original
RPCA Reconst.
RPCA Thresh. Diff.
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0

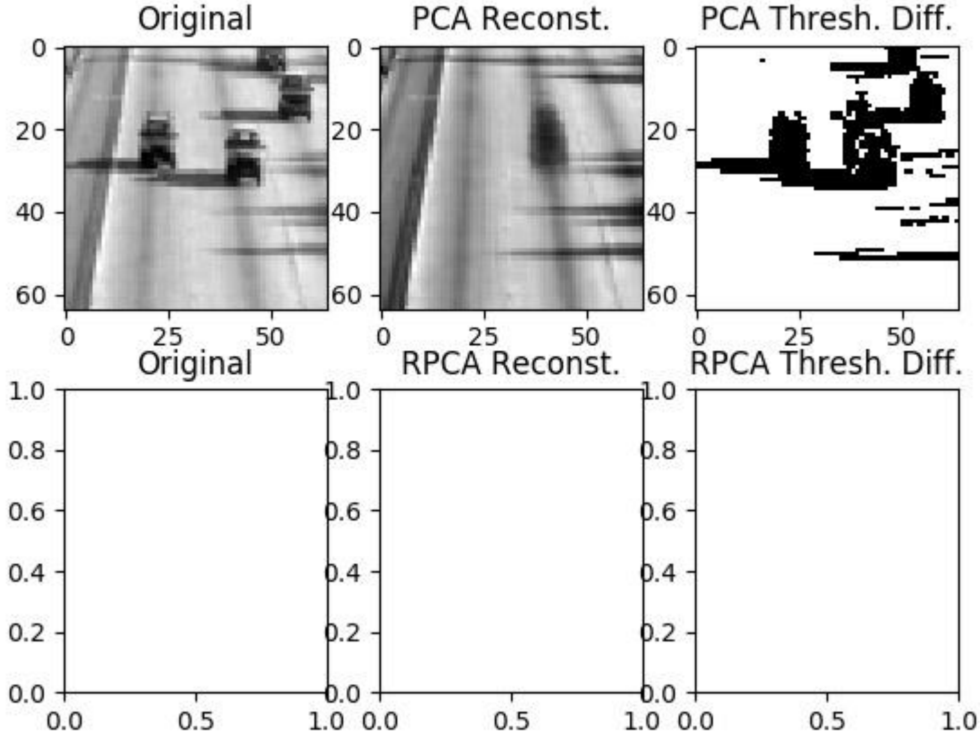

Original
PCA Reconst.
PCA Thresh. Diff.
0
20
40
60
0
25
50
Original
RPCA Reconst.
RPCA Thresh. Diff.
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0


Original
PCA Reconst.
PCA Thresh. Diff.
0
20
40
60
0
25
50
Original
RPCA Reconst.
RPCA Thresh. Diff.
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0


Original
PCA Reconst.
PCA Thresh. Diff.
0
20
40
60
0
25
50
Original
RPCA Reconst.
RPCA Thresh. Diff.
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0


Original
PCA Reconst.
PCA Thresh. Diff.
0
20
40
60
0
25
50
Original
RPCA Reconst.
RPCA Thresh. Diff.
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0

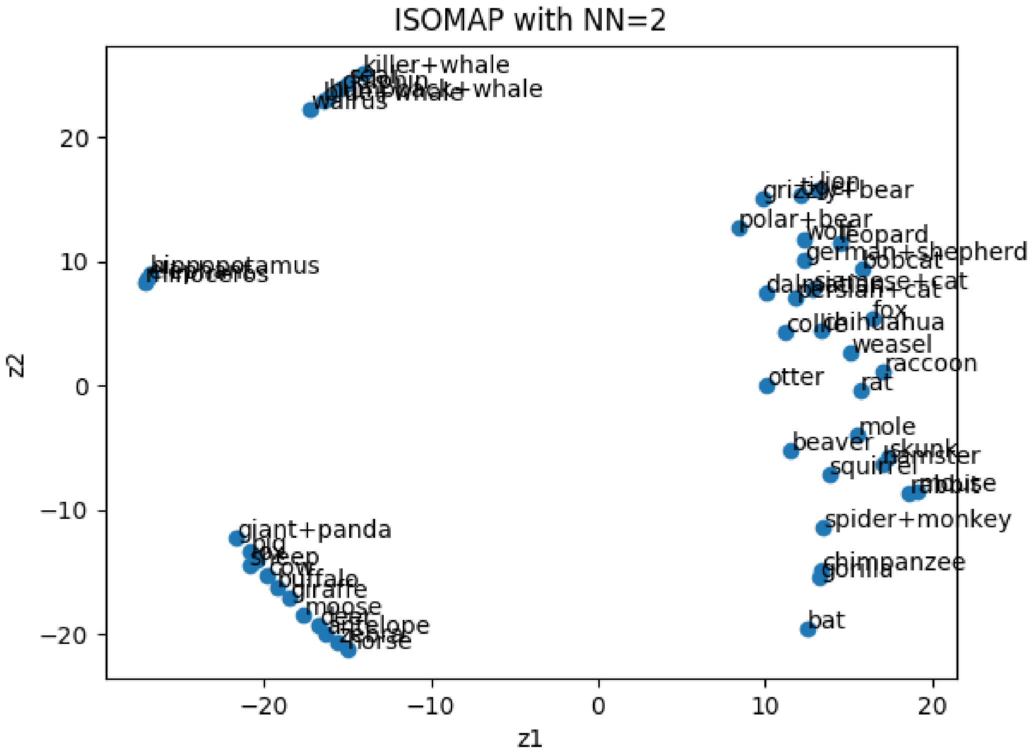

Original
PCA Reconst.
PCA Thresh. Diff.
0
20
40
60
0
25
50
Original
RPCA Reconst.
RPCA Thresh. Diff.
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0

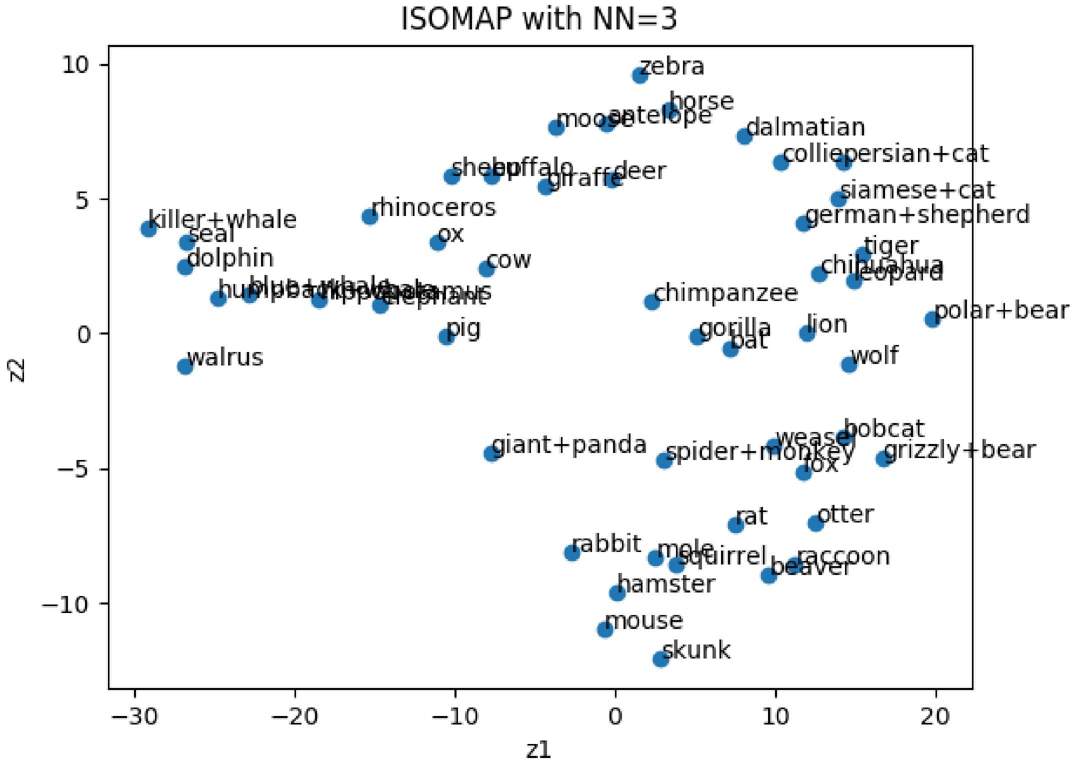

Original
PCA Reconst.
PCA Thresh. Diff.
0
20
40
60
0
25
50
Original
RPCA Reconst.
RPCA Thresh. Diff.
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0

5 Multi-Dimensional Scaling

5.1 ISOMAP

Code: https://github.ubc.ca/cpsc340-2017S/sopida_zhenxil_a4/tree/master/code/manifold.py


ISOMAP with NN=2


ISOMAP with NN=3

5.2 Reflection

PCA: Animals are separated into clusters with aquatic animals on the top left and land animals at the bottom. A big crowding cluster of dissimilar animals in the top right corner.

MDS: Disperse individual animals in evenly spaced distance. Difficult to tell groups of similar animals.

ISOMAP: With 2-nearest neighbours, ISOMAP separates animals into distinct clusters which are far apart from one another. With 3-nearest neighbours, there is no distinct clusters. The aquatic animals are on the top left then large land animals as we move to the right. As we move down, the size of animals is getting smaller.

ISOMAP did the best job because it doesn’t have crowding effect like PCA and it does a better job at separating different types of animals.