KMPA: Assignment 2 Report

Group 1: CS13B053, CS13B060, CS13B061

November 25, 2017

1 Classification using SVMs

For this task, C-SVM and ν-SVM were trained on different data sets and for different kernels. For a given data set and the kernel, best hyper parameters were selected based on performance of the model on the validation set. General performance of the best selected SVM was quantified by the performance of the SVM on the test data. The two kernels used for the experiment are as follows:

Polynomial Kernel:

$$K(\overline{x}, \overline{y}) = (\gamma\ \overline{x}^T \overline{y} + r)^d$$

RBF Kernel:

$$K(\overline{x}, \overline{y}) = e^{-\gamma\ \|\overline{x} - \overline{y}\|^2}$$

1.1 Models on Dataset 1(a): Linearly Separable Classes

1.1.1 C-SVM with Polynomial Kernel

- **Parameters:**
  The model with the following parameters gave best performance (100%) on the validation set:

$$\begin{gathered} C = 10 \\ \gamma = 1 \\ r = 0 \\ d = 1 \end{gathered}$$

- **Decision Region Plot** obtained is shown in Figure 1. The black points in the figure are the Support Vectors.

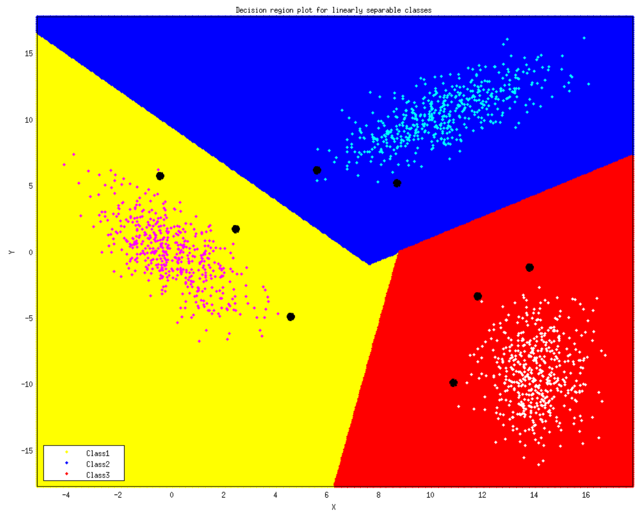

Figure 1: Decision region plot obtained for C-SVM on Dataset 1(a)

- **Confusion Matrix** obtained is shown in Figure 2.

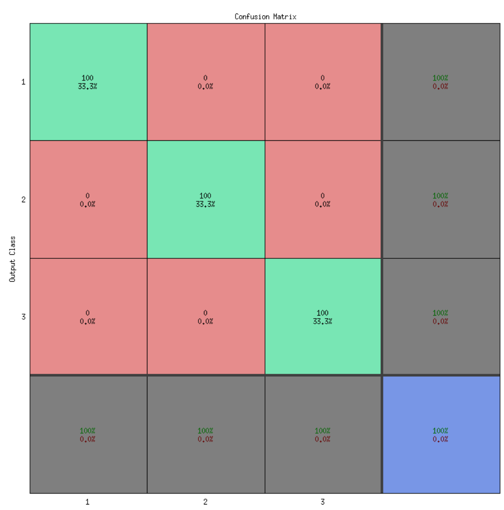

Figure 2: Confusion Matrix obtained for C-SVM on Dataset 1(a)

- **Image Representation of the Kernel Gram Matrix** for polynomial

kernel on training dataset is shown in Figure 3.

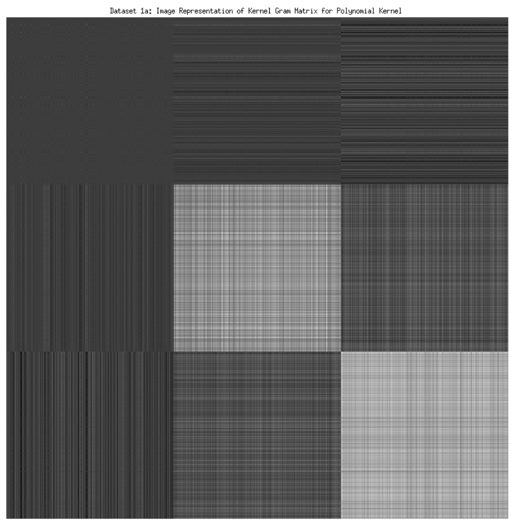

Figure 3: Kernel Gram Matrix for Polynomial Kernel on Dataset 1(a)

- **Observations and Inferences:**
  C-SVM with polynomial kernel gives 100% test accuracy and perfectly classifies the linearly separable data.
  For polynomial kernel, kernel gram matrix is not a good measure to evaluate parameters, because the value of kernel depends on spatial location of the points. Here for class1 (Yellow class in Figure 1), the points lie close to origin and hence have small intra-class kernel values as compared to points of class2 (Blue class in Figure 1) whose points lie in 1st quadrant having high intra-class kernel values.

1.1.2 C-SVM with Gaussian Kernel

- **Parameters:**
  The model with the following parameters gave best performance (100%) on the validation set:

$$C = 1$$
$$\gamma = 0.03125$$

- **Decision Region Plot** obtained is shown in Figure 4. The black points in the figure are the Support Vectors.

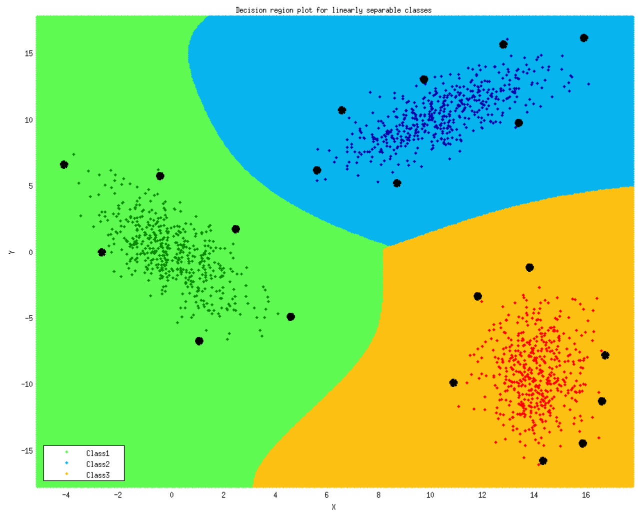

Figure 4: Decision region plot obtained for C-SVM on Dataset 1(a)

- **Confusion Matrix** obtained is shown in Figure 5.

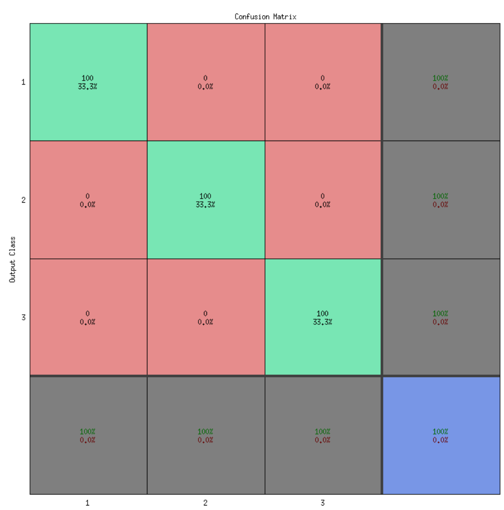

Figure 5: Confusion Matrix obtained for C-SVM on Dataset 1(a)

- **Image Representation of the Kernel Gram Matrix** for polynomial

kernel on training dataset is shown in Figure 6.

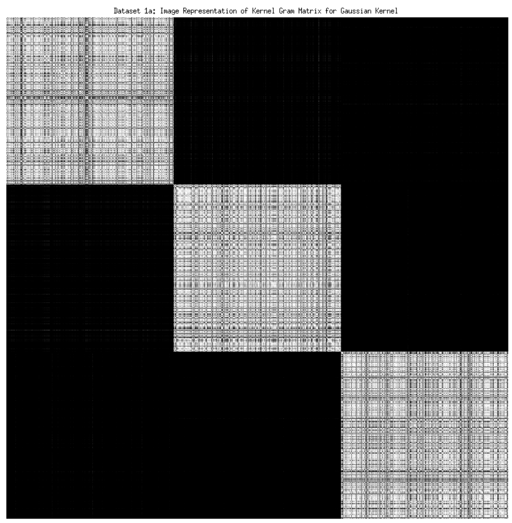

Figure 6: Kernel Gram Matrix for Gaussian Kernel on Dataset 1(a)

- **Observations and Inferences:**
  C-SVM with gaussian kernel gives 100% test accuracy and perfectly classifies the linearly separable data.
  As can be seen from Figure 6, the kernel gram matrix gives good estimate of the parameters chosen for gaussian kernel, as the kernel value of two data points depends on distance between them. For points of same class the distance is small, hence large kernel value and vice versa.

1.1.3 ν-SVM with Polynomial Kernel

- **Parameters:**
  The model with the following parameters gave best performance (100%) on the validation set:

$$\nu = 0.01$$
$$\gamma = 1$$
$$r = 0$$
$$d = 1$$

- **Decision Region Plot** obtained is shown in Figure 7. The black points in the figure are the Support Vectors.

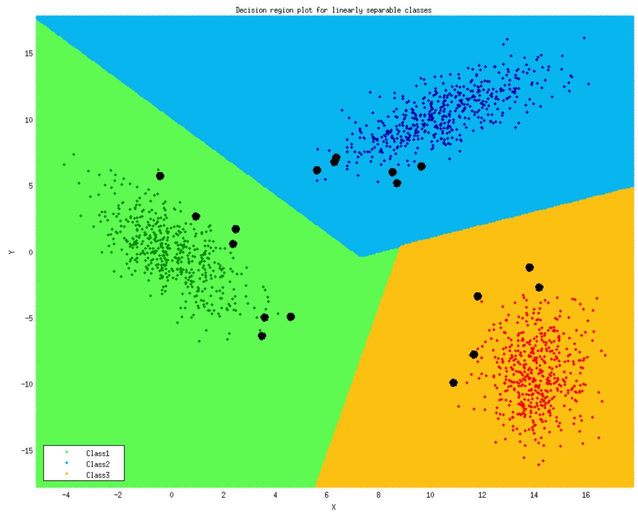

Figure 7: Decision region plot obtained for ν-SVM on Dataset 1(a)

- **Confusion Matrix** obtained is shown in Figure 8.

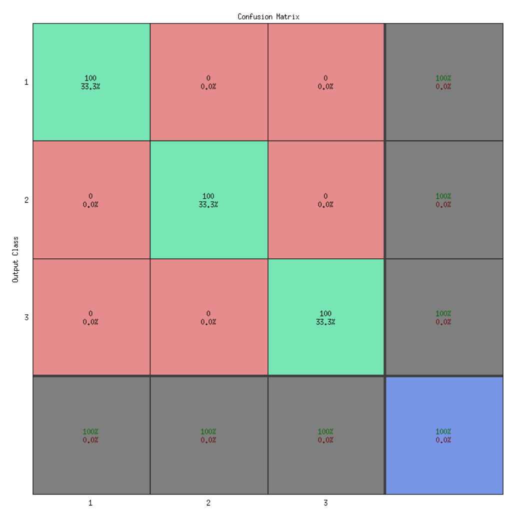

Figure 8: Confusion Matrix obtained for ν-SVM on Dataset 1(a)

- **Image Representation of the Kernel Gram Matrix** for polynomial

kernel on training dataset is shown in Figure 9.

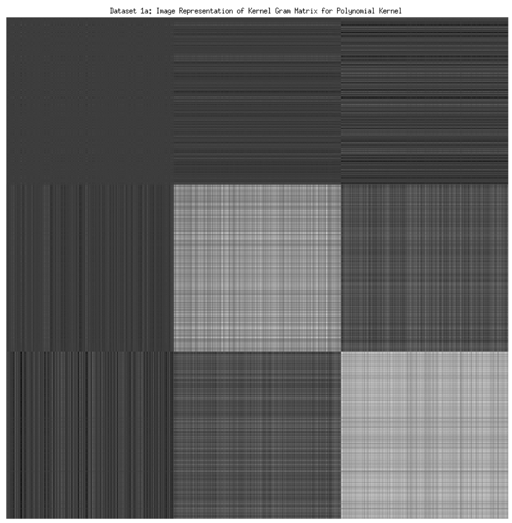

Figure 9: Kernel Gram Matrix for Polynomial Kernel on Dataset 1(a)

- **Observations and Inferences:** ν-SVM with polynomial kernel gives 100% test accuracy and perfectly classifies the linearly separable data. Here again, we can see that the Kernel Gram Matrix is not a good measure to evaluate polynomial kernel.

1.1.4 ν-SVM with Gaussian Kernel

- **Parameters:**
  The model with the following parameters gave best performance (100%) on the validation set:

$$\nu = 0.005$$
$$\gamma = 0.03125$$

- **Decision Region Plot** obtained is shown in Figure 10. The black points in the figure are the Support Vectors.

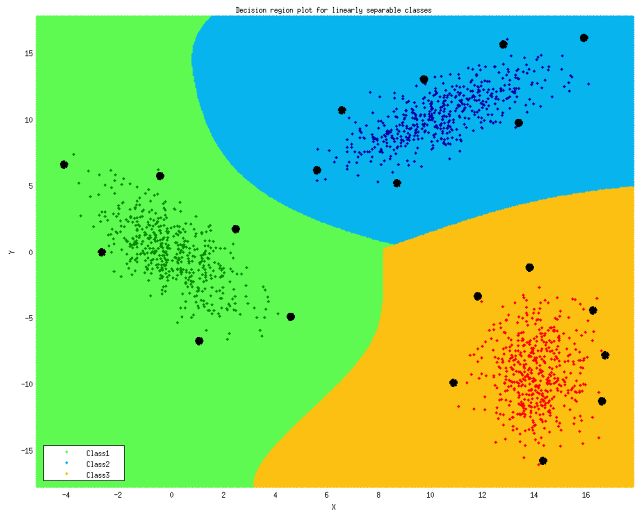

Figure 10: Decision region plot obtained for ν-SVM on Dataset 1(a)

- **Confusion Matrix** obtained is shown in Figure 11.

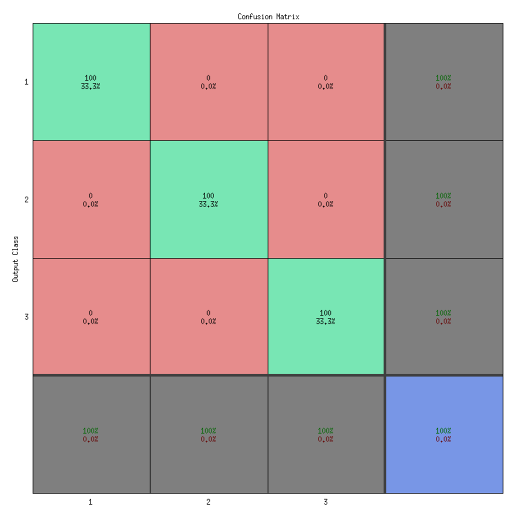

Figure 11: Confusion Matrix obtained for ν-SVM on Dataset 1(a)

- **Image Representation of the Kernel Gram Matrix** for gaussian kernel on training dataset is shown in Figure 12.

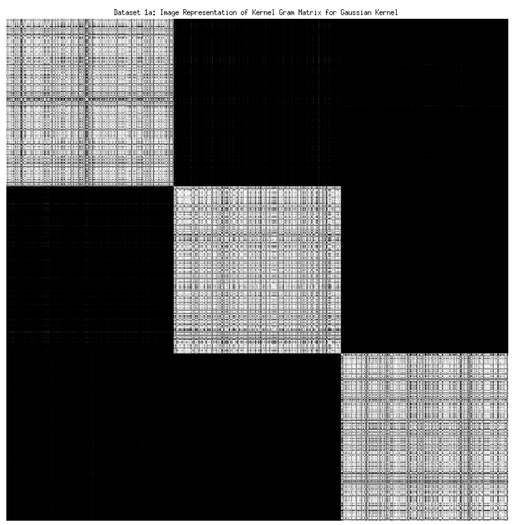

Figure 12: Kernel Gram Matrix for Gaussian Kernel on Dataset 1(a)

- **Observations and Inferences:**
  ν-SVM with gaussian kernel gives 100% test accuracy and perfectly classifies the linearly separable data.

1.2 Models on Dataset 1(b): Non-linearly Separable Classes

1.2.1 C-SVM with Polynomial Kernel

- **Parameters:**
  The model with the following parameters gave best performance (100%) on the validation set:

$$C = 10$$
$$\gamma = 1$$
$$r = 0$$
$$d = 2$$

- **Decision Region Plot** obtained is shown in Figure 13. The black points in the figure are the Support Vectors.

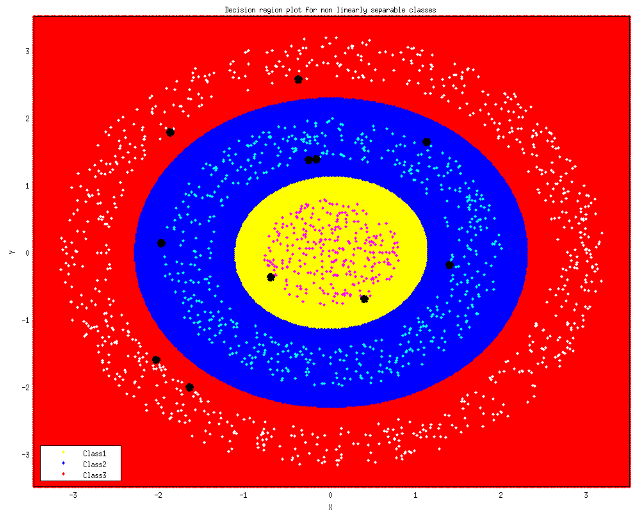

Figure 13: Decision region plot obtained for C-SVM on Dataset 1(b)

- **Confusion Matrix** obtained is shown in Figure 14.

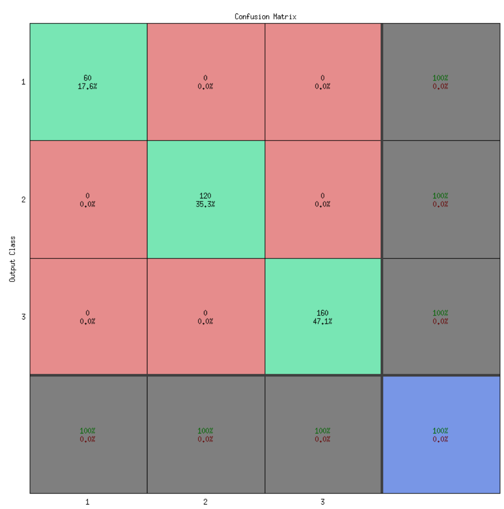

Figure 14: Confusion Matrix obtained for C-SVM on Dataset 1(b)

- **Image Representation of the Kernel Gram Matrix** for polynomial kernel on training dataset is shown in Figure 15.

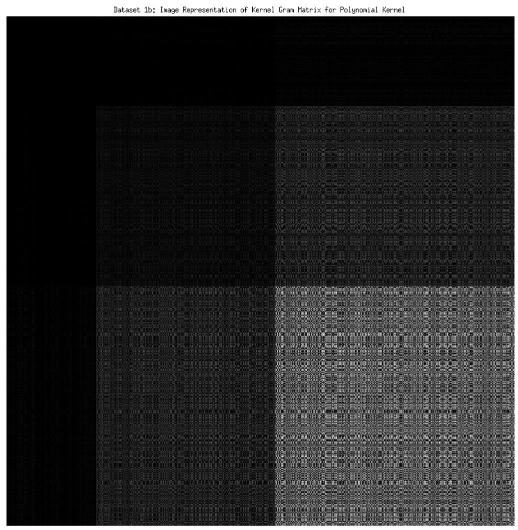

Figure 15: Kernel Gram Matrix for Polynomial Kernel on Dataset 1(b)

- **Observations and Inferences:**
  C-SVM with polynomial kernel of degree 2 gives 100% test accuracy and perfectly classifies the non-linearly separable data, as expected because degree 2 polynomial kernel can classify elliptical shaped classes.
  Again, the kernel gram matrix is not a good measure to evaluate polynomial kernels.

1.2.2 C-SVM with Gaussian Kernel

- **Parameters:**
  The model with the following parameters gave best performance (100%) on the validation set:

$$C = 1000$$
$$\gamma = 0.5$$

- **Decision Region Plot** obtained is shown in Figure 16. The black points in the figure are the Support Vectors.

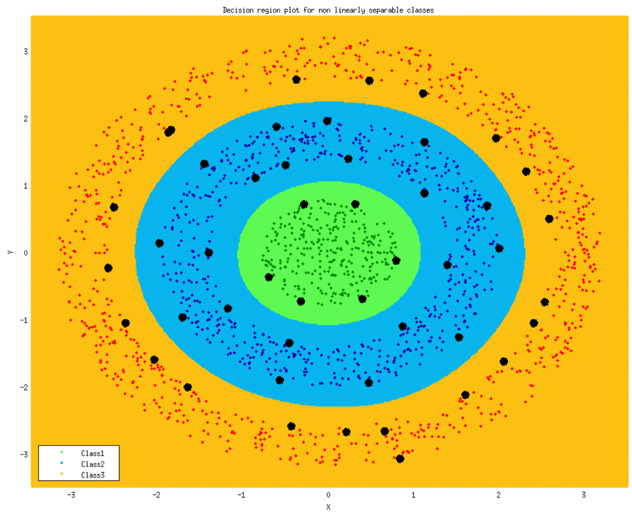

Figure 16: Decision region plot obtained for C-SVM on Dataset 1(b)

- **Confusion Matrix** obtained is shown in Figure 17.

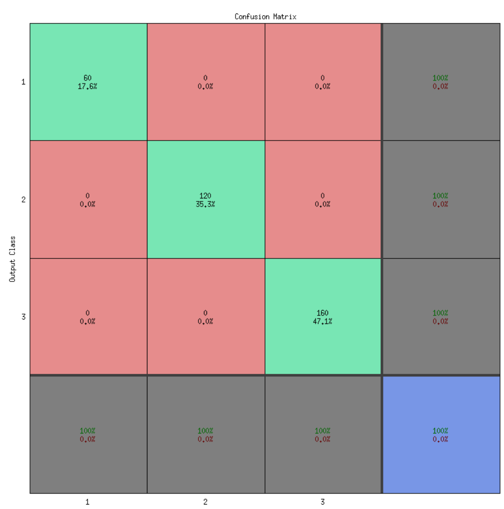

Figure 17: Confusion Matrix obtained for C-SVM on Dataset 1(b)

- **Image Representation of the Kernel Gram Matrix** for gaussian kernel on training dataset is shown in Figure 18.

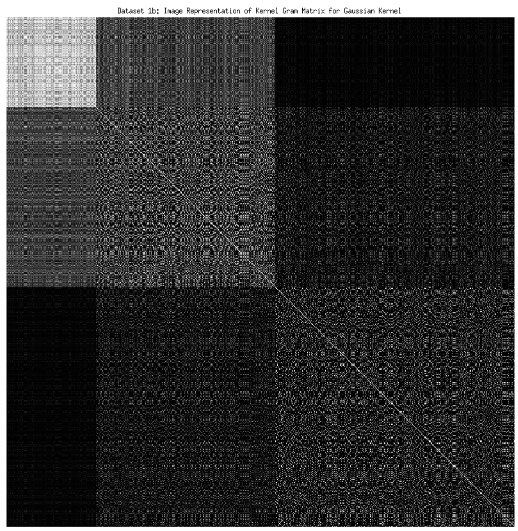

Figure 18: Kernel Gram Matrix for Gaussian Kernel on Dataset 1(b)

- **Observations and Inferences:**
  C-SVM with gaussian kernel gives 100% test accuracy and perfectly classifies the non-linearly separable data.

1.2.3 ν-SVM with Polynomial Kernel

- **Parameters:**
  The model with the following parameters gave best performance (100%) on the validation set:

$$\nu = 0.01$$
$$\gamma = 1$$
$$r = 0$$
$$d = 2$$

- **Decision Region Plot** obtained is shown in Figure 19. The black points in the figure are the Support Vectors.

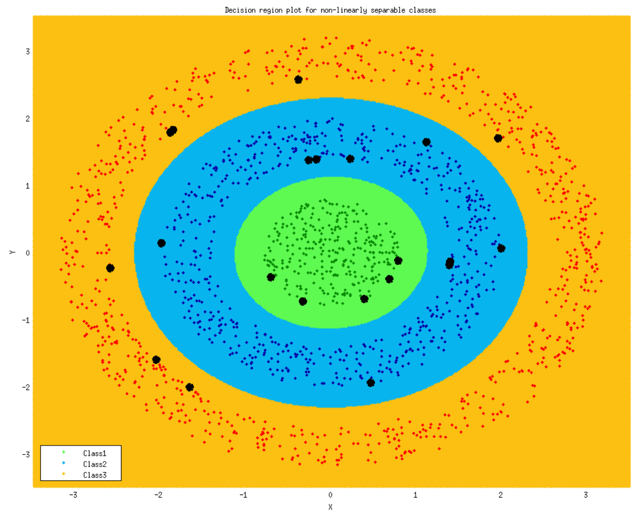

Figure 19: Decision region plot obtained for ν-SVM on Dataset 1(b)

- **Confusion Matrix** obtained is shown in Figure 20.

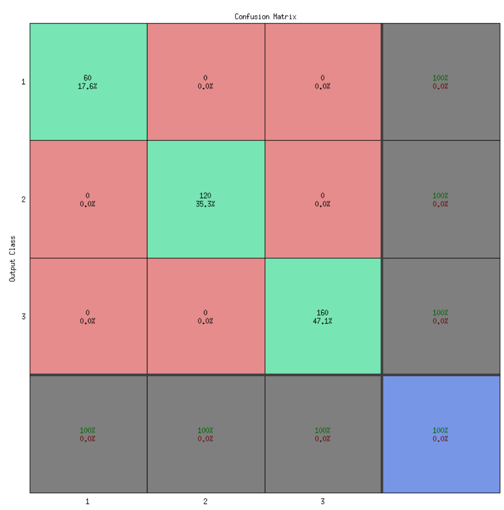

Figure 20: Confusion Matrix obtained for ν-SVM on Dataset 1(b)

- **Image Representation of the Kernel Gram Matrix** for polynomial kernel on training dataset is shown in Figure 21.

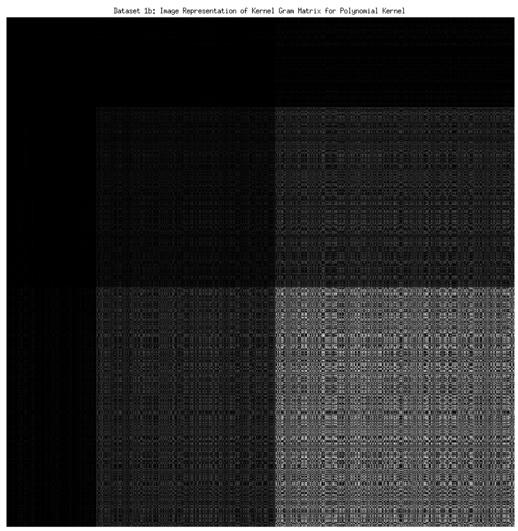

Figure 21: Kernel Gram Matrix for Polynomial Kernel on Dataset 1(b)

- **Observations and Inferences:**
  ν-SVM with polynomial kernel gives 100% test accuracy and perfectly classifies the non-linearly separable data.

1.2.4 ν-SVM with Gaussian Kernel

- **Parameters:**
  The model with the following parameters gave best performance (100%) on the validation set:

$$\nu = 0.005$$
$$\gamma = 0.5$$

- **Decision Region Plot** obtained is shown in Figure 22. The black points in the figure are the Support Vectors.

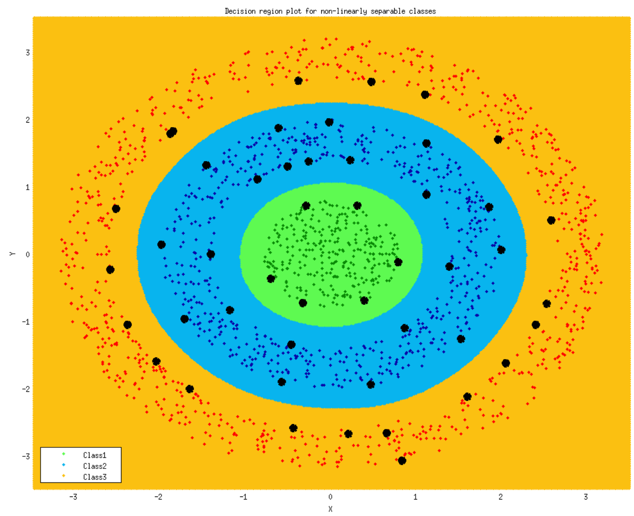

Figure 22: Decision region plot obtained for ν-SVM on Dataset 1(b)

- **Confusion Matrix** obtained is shown in Figure 23.

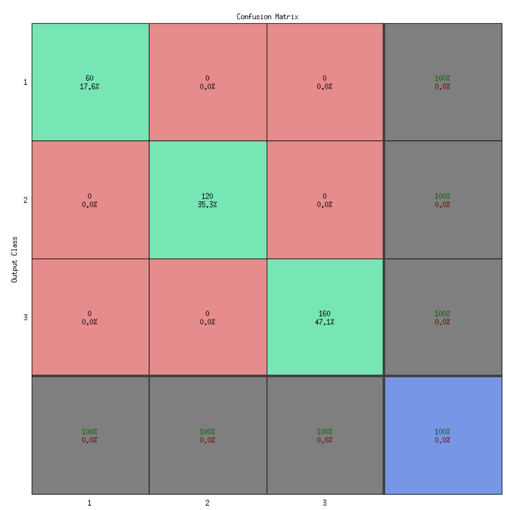

Figure 23: Confusion Matrix obtained for ν-SVM on Dataset 1(b)

- **Image Representation of the Kernel Gram Matrix** for gaussian

kernel on training dataset is shown in Figure 24.

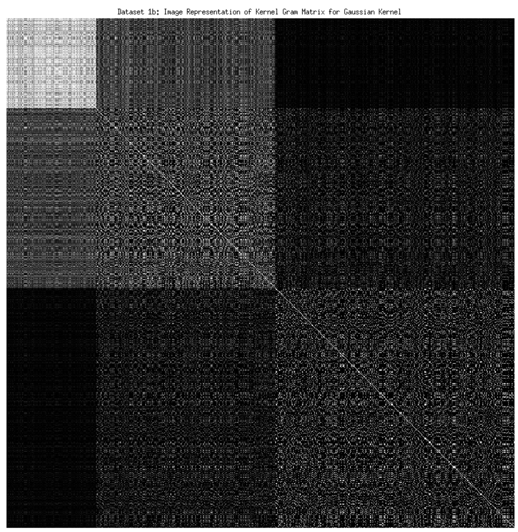

Figure 24: Kernel Gram Matrix for Gaussian Kernel on Dataset 1(b)

- **Observations and Inferences:**
  ν-SVM with gaussian kernel gives 100% test accuracy and perfectly classifies the non-linearly separable data.

1.3 Model on Dataset 1(c): Overlapping Classes

1.3.1 C-SVM with Polynomial Kernel

- **Parameters:** The parameters were chosen based on grid search on two level. Firstly, a grid search on γ and C values in powers of 2, then further exploiting the best solution in its neighbourhood.
  The model with the following parameters gave best performance on the validation set:

$$C = 32.25$$
$$\gamma = 2$$
$$r = 0.5$$
$$d = 3$$

- **Decision Region Plot** obtained is shown in Figure 25. The black points in the figure are the Support Vectors.

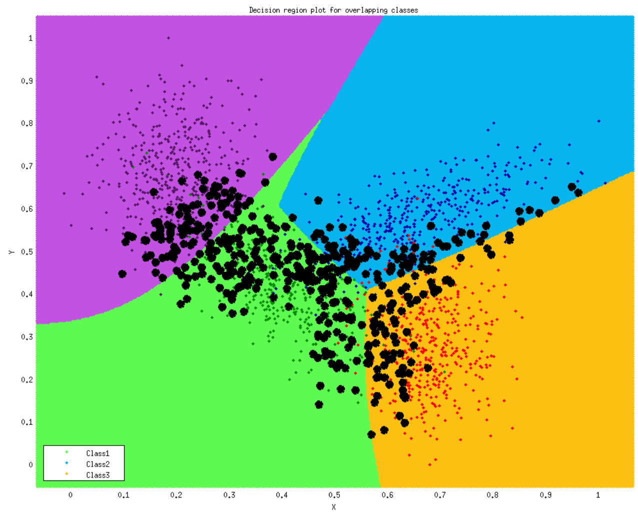

Figure 25: Decision region plot obtained for C-SVM on Dataset 1(c)

- **Confusion Matrix** obtained is shown in Figure 26.

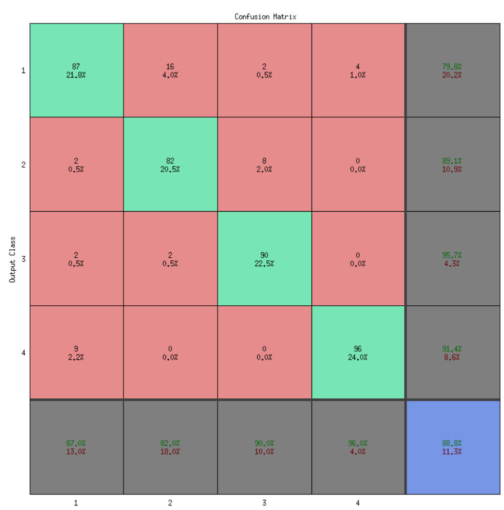

Figure 26: Confusion Matrix obtained for C-SVM on Dataset 1(c)

- **Image Representation of the Kernel Gram Matrix** for polynomial

kernel on training dataset is shown in Figure 27.

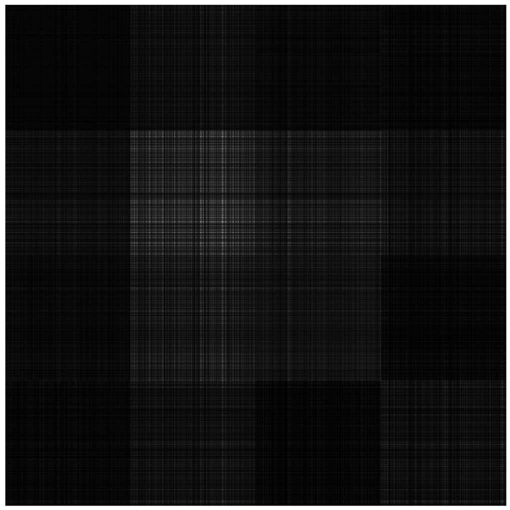

Figure 27: Kernel Gram Matrix for Polynomial Kernel on Dataset 1(c)

- **Observations and Inferences:**
  C-SVM with polynomial kernel with above parameter settings gives test accuracy of 88.8%.
  Degree 3 polynomial kernel gives better results than degree 2 polynomial kernel, as expected because the data is overlapping and a more complex model is required to solve the classification task.
  As can be seen from the decision region plot, the number of support vectors are more as compared to the other two data sets, because as the classification problem becomes harder more points contribute to the decision boundary.

1.3.2 C-SVM with Gaussian Kernel

- **Parameters:** Again, the parameters were selected based on two levels of grid search. The model with the following parameters gave best performance on the validation set:

$$C = 0.3125$$
$$\gamma = 16$$

- **Decision Region Plot** obtained is shown in Figure 28. The black points in the figure are the Support Vectors.

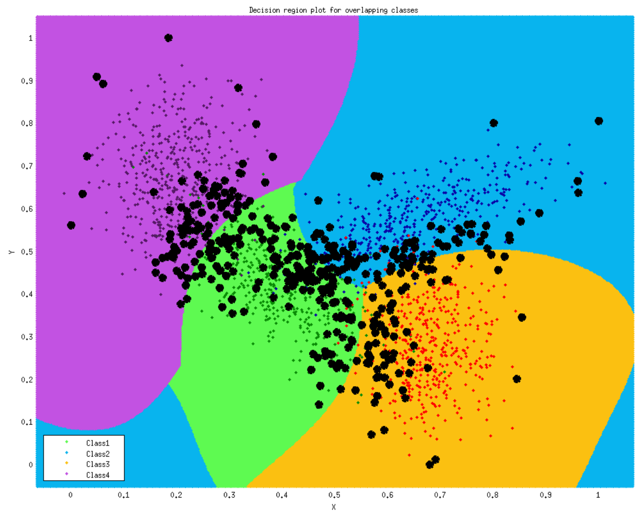

Figure 28: Decision region plot obtained for C-SVM on Dataset 1(c)

- **Confusion Matrix** obtained is shown in Figure 29.

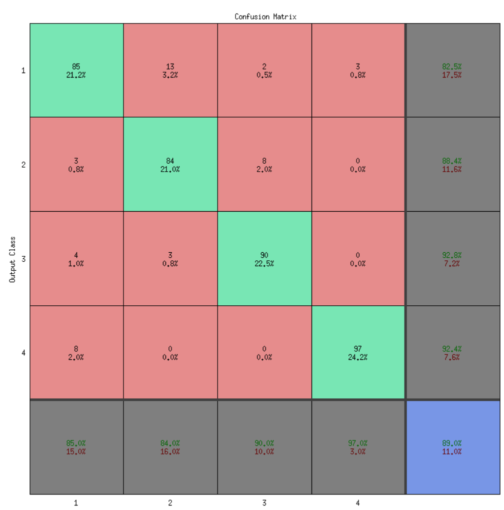

Figure 29: Confusion Matrix obtained for C-SVM on Dataset 1(c)

- **Image Representation of the Kernel Gram Matrix** for gaussian

kernel on training dataset is shown in Figure 30.

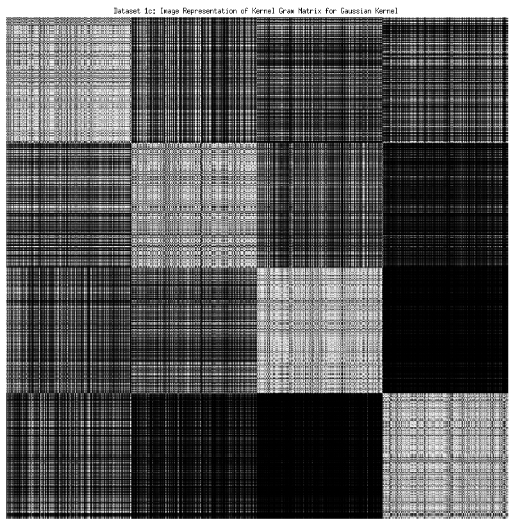

Figure 30: Kernel Gram Matrix for Gaussian Kernel on Dataset 1(c)

- **Observations and Inferences:**
  C-SVM with gaussian kernel with above parameters settings gives 89.0% test accuracy.
  Again, as the problem is harder to solve because of overlap between the classes, more support vectors are needed to decide the decision surface.
  The Kernel Gram Matrix is close to ideal for this dataset, having white regions for points of same class and grey regions for point of different class.

1.3.3 ν-SVM with Polynomial Kernel

- **Parameters:**
  The model with the following parameters gave best performance on the validation set:

$$\nu = 0.2188$$
$$\gamma = 2$$
$$r = 0.5$$
$$d = 3$$

- **Decision Region Plot** obtained is shown in Figure 31. The black points in the figure are the Support Vectors.

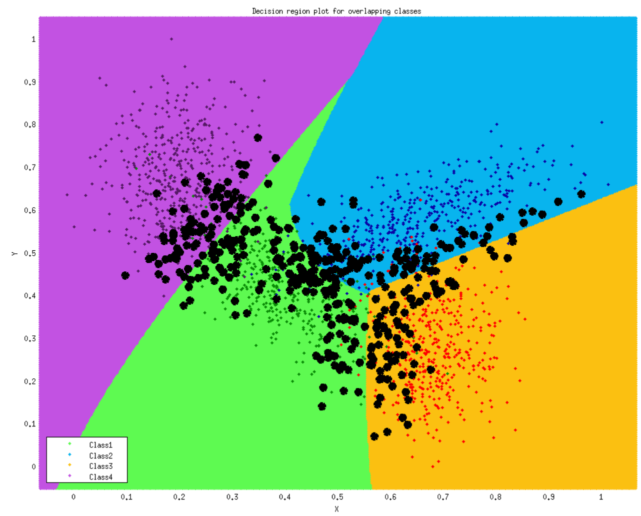

Figure 31: Decision region plot obtained for ν-SVM on Dataset 1(c)

- **Confusion Matrix** obtained is shown in Figure 32.

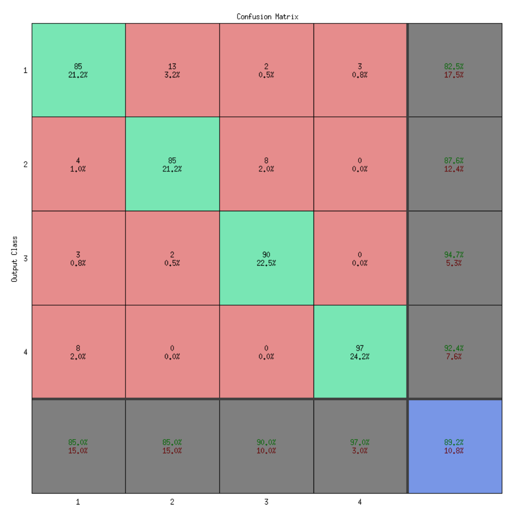

Figure 32: Confusion Matrix obtained for ν-SVM on Dataset 1(c)

- **Image Representation of the Kernel Gram Matrix** for polynomial

kernel on training dataset is shown in Figure 33.

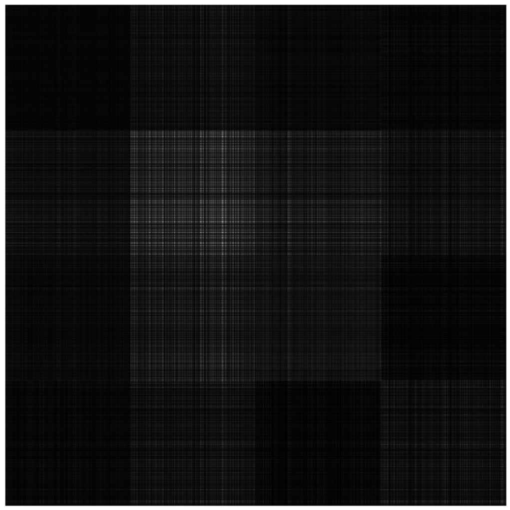

Figure 33: Kernel Gram Matrix for Polynomial Kernel on Dataset 1(c)

- **Observations and Inferences:**
  ν-SVM with polynomial kernel gives 89.25% test accuracy.
  Here also the Kernel Gram Matrix for polynomial kernel does not give any conclusive evaluation on the kernel parameters.

1.3.4 ν-SVM with Gaussian Kernel

- **Parameters:**
  The model with the following parameters gave best performance on the validation set:

$$\nu = 0.2422$$
$$\gamma = 16$$

- **Decision Region Plot** obtained is shown in Figure 34. The black points in the figure are the Support Vectors.

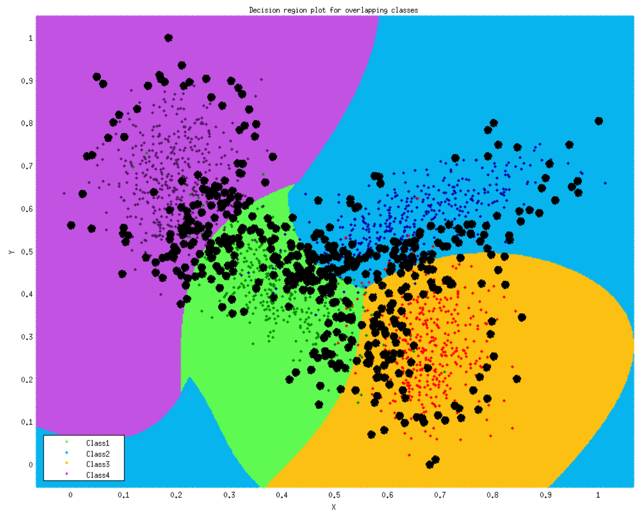

Figure 34: Decision region plot obtained for ν-SVM on Dataset 1(c)

- **Confusion Matrix** obtained is shown in Figure 35.

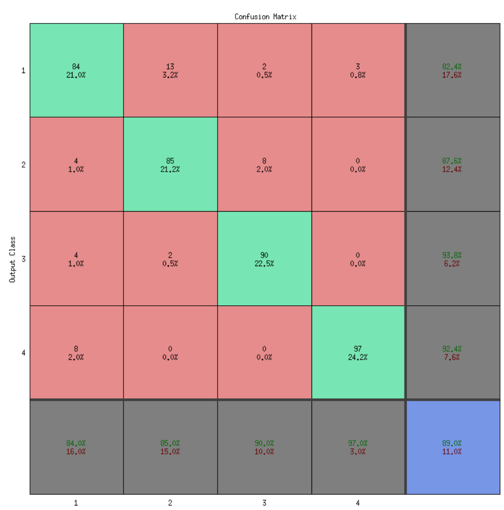

Figure 35: Confusion Matrix obtained for ν-SVM on Dataset 1(c)

- **Image Representation of the Kernel Gram Matrix** for gaussian

kernel on training dataset is shown in Figure 36.

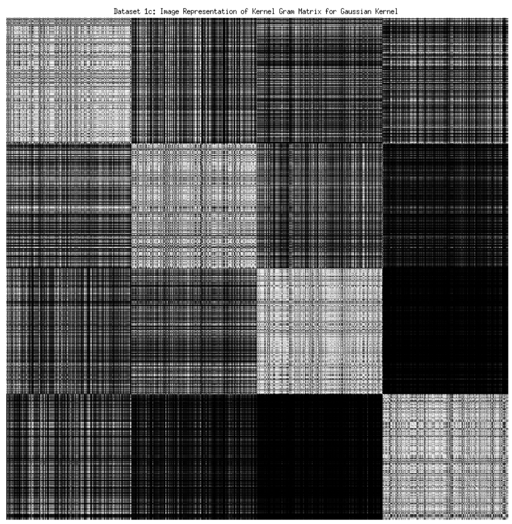

Figure 36: Kernel Gram Matrix for Gaussian Kernel on Dataset 1(c)

- **Observations and Inferences:**
  ν-SVM with gaussian kernel gives 89% test accuracy.
  The Kernel Gram Matrix for the training dataset is perfect for the above parameter settings, as for gaussian kernel the kernel value depends on the distance between the points not on their spatial location, as is the case of polynomial kernels.

2 Dimension Reduction using PCA, Autoencoder and Stacked Autoencoder

Four ν-SVM with Gaussian kernel were built, one without dimension reduction, and other three with dimension reduction using PCA, Autoencoder and Stacked Autoencoder.

2.1 Without Dimension Reduction

- **Parameters:**
  The model with the following parameters gave best performance on validation data set:

$$\gamma = 3.0$$
$$\nu = 0.15$$

- **Confusion Matrix** obtained is shown in Table 1.

Class	1	2	3	4	5
1	126	3	6	3	3
2	14	32	0	17	4
3	12	2	60	0	9
4	23	17	6	13	7
5	14	2	12	1	112

Table 1: Confusion Matrix obtained for ν-SVM on Dataset 2

- **Image Representation of the Kernel Gram Matrix** for Dataset 2 without dimension reduction with gaussian kernel is shown in Figure 37.

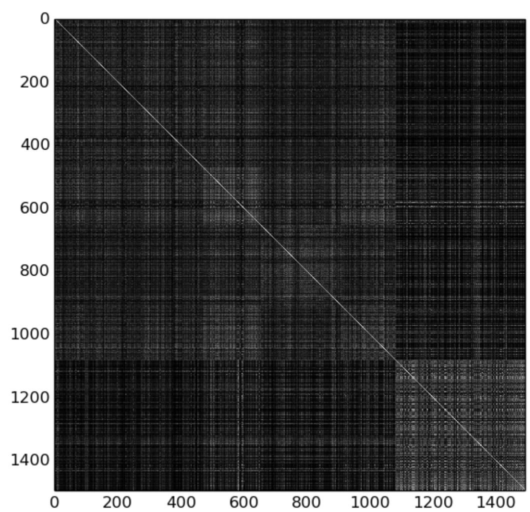

Figure 37: Kernel Gram Matrix for Gaussian Kernel on Dataset 2

- **Observations and Inferences:**
ν-SVM with Gaussian kernel gives 68.88% test accuracy for Dataset 2 without dimension reduction.
Even without any feature reduction the ν-SVM with Gaussian kernel gives comparable results to models with dimension reduction.
The kernel gram matrix for the training data set does not look conclusive, maybe because of the properties of the features selected or maybe because of the high dimensionality of the image dataset.

2.2 Dimension Reduction using PCA

- **Parameters:**
  The number of parameters were decided by listing the values of λs obtained, and choosing the significant λs.

  Features: 90
  $\gamma = 4.0$
  $\nu = 0.1$

- **Confusion Matrix** obtained is shown in Table 2.

Class	1	2	3	4	5
1	125	1	7	2	6
2	18	26	1	14	8
3	12	1	61	0	9
4	25	15	6	11	9
5	17	1	12	1	110

Table 2: Confusion Matrix obtained for ν-SVM on Dataset 2

- **Image Representation of the Kernel Gram Matrix** for gaussian kernel on Dataset 2 with dimension reduction using PCA is shown in Figure 38.

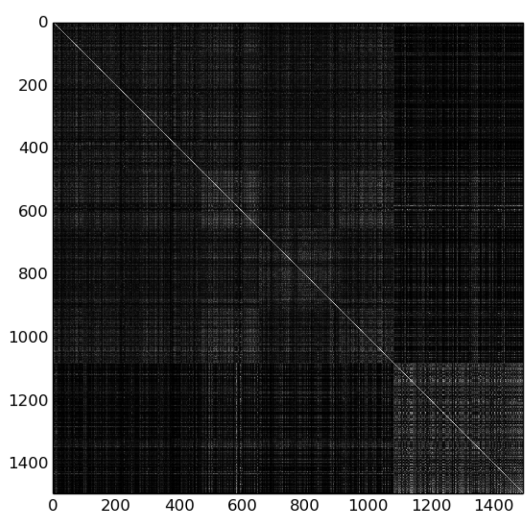

Figure 38: Kernel Gram Matrix for Gaussian Kernel on Dataset 2

- **Observations and Inferences:**
ν-SVM with Gaussian kernel gives 67.67% test accuracy for Dataset 2 with dimension reduction using PCA.
Even after reducing the number of features by more than a factor of 2, PCA is able to give a performance very close to the optimal performance obtained from keeping all the 512 features. PCA achieves this by choosing the linear directions along which the variance of the data is maximized. This ensures that not a lot of information is lost while performing feature reduction.

2.3 Dimension Reduction using Autoencoder

- The autoencoder had 3 hidden layers, with the outer hidden layers having sigmoidal activation function, and the bottleneck layer having linear activation.
- The outer hidden layers had $1.2x$ the number of hidden nodes as the input nodes.
- **Parameters:**
The model with the following parameters gave best performance on validation data set:

$$\text{Features: } 90$$
$$\gamma = 350.0$$
$$\nu = 0.2$$

- **Confusion Matrix** obtained is shown in Table 3.

Class	1	2	3	4	5
1	121	4	10	1	5
2	18	27	1	17	4
3	10	2	60	0	11
4	24	16	4	15	7
5	15	1	13	3	109

Table 3: Confusion Matrix obtained for ν-SVM on Dataset 2

- **Image Representation of the Kernel Gram Matrix** for Gaussian kernel on Dataset 2 with dimension reduction using Autoencoder is shown in Figure 39.

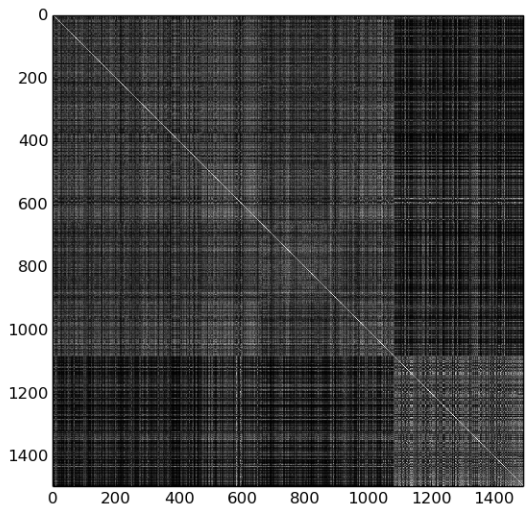

Figure 39: Kernel Gram Matrix for Gaussian Kernel on Dataset 2

- **Observations and Inferences:**
  ν-SVM with Gaussian kernel gives 67.87% test accuracy for Dataset 2 with dimension reduction using Autoencoder.
  We observe that Autoencoder is able to extract meaningful features from the data and gives accuracies very close to those obtained on using the entire feature set.
  Feature extraction using an Autoencoder can be thought of as performing a non-linear PCA on the given data.

2.4 Dimension Reduction using Stacked Autoencoder

- The stacked autoencoders was composed of 3 autoencoders, where each autoencoder had a structure similar to the autoencoders described above.

- Each autoencoder reduced the dimensionality by a fixed ratio, so as to get the desired output dimensionality.

- **Parameters:**
  The model with the following parameters gave best performance on validation data set:

  Features: 260
  $\gamma = 3.0 \times 10^6$
  $\nu = 0.2$

- **Confusion Matrix** obtained is shown in Table 4.

Class	1	2	3	4	5
1	117	4	8	7	5
2	16	29	1	13	8
3	9	1	61	1	11
4	25	17	4	13	7
5	12	1	7	3	118

Table 4: Confusion Matrix obtained for ν-SVM on Dataset 2

- **Image Representation of the Kernel Gram Matrix** for gaussian kernel on Dataset 2 with dimension reduction using Stacked Autoencoder is shown in Figure 40.

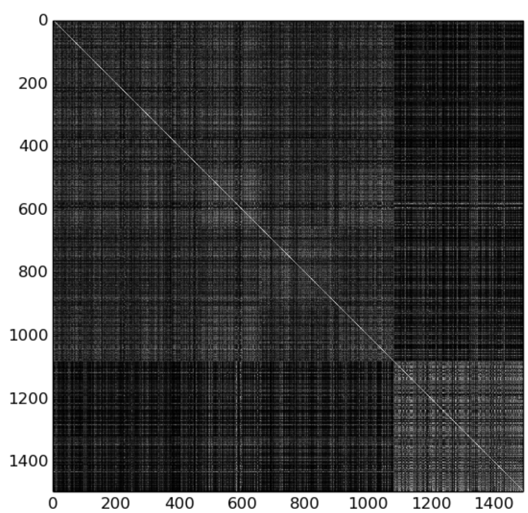

Figure 40: Kernel Gram Matrix for Gaussian Kernel on Dataset 2

- **Observations and Inferences:**
ν-SVM with Gaussian kernel gives 67.87% test accuracy for Dataset 2 with dimension reduction using Stacked Autoencoder.
We observe that Stacked Autoencoder is able to extract meaningful features from the data and gives accuracies very close to those obtained on using the entire feature set.

3 Classification using Deep Convolutional Neural Network (DCNN)

A Deep Convolutional Neural Network model is built to classify the images in Dataset 2.

- **Structure of DCNN:**
  The DCNN model built has 3 levels of Convolution layer followed by Pooling layer, 2 fully connected Hidden layers and an Output layer. An image in the dataset is of size $3 \times 32 \times 32$. The receptive field size for every convolution layer is 5×5. The 1st convolution layer has 6 feature maps (28×28), 2nd layer has 16 feature maps (10×10) and 3rd layer has 100 feature maps (1×1). The two hidden layers have 120 and 80 nodes respectively and output layer has 5 nodes (equal to number of classes). The number of feature maps are chosen based on cross validation.
  Each pooling layer is fully connected to next convolution layer. We also tried the LeNet like model, but the performance on validation set was not good.
  At each convolution layer, hyperbolic tangent activation function is used. At hidden layers also, hyperbolic tangent activation function is used. The output layer has logistic softmax activation function. Negative log likelihood was used as an error criteria. We also tried rectified linear and sigmoid activation functions, but performance on validation set was not better than hyperbolic tangent.
  The model was chosen based on best performance on validation set. The best model had 58.83% accuracy on validation set.

- **Confusion Matrix** obtained for test data is shown in Table 5.

Class	1	2	3	4	5
1	100	13	10	11	12
2	17	19	1	12	11
3	23	1	36	7	16
4	23	16	7	12	4
5	16	14	6	9	102

Table 5: Confusion Matrix obtained for DCNN on Dataset 2

- **Observations and Inferences:**
  The Deep Convolutional Neural Network model gives 54.02% test accuracy for classification of images in Dataset 2.
  It can be seen that the model performs fairly well even after significantly reducing the number of features per image from $3 \times 32 \times 32$ to 100.

4 Classification using DCNN features and SVM

The DCNN features obtained in Task 3 are used for training SVM for classification of images in Dataset 2.

- **Parameters:**
  γ was selected based on observing the image of the kernel gram matrix. Then ν was selected based on the cross validation method. The parameters for ν-SVM with Gaussian kernel for model with best performance on validation set are:

$$\gamma = 0.025$$
$$\nu = 0.1$$

- **Confusion Matrix** obtained for test data is shown in Table 6.

Class	1	2	3	4	5
1	94	16	21	22	12
2	13	22	1	11	4
3	16	0	48	6	8
4	13	11	3	15	6
5	10	11	10	8	117

Table 6: Confusion Matrix obtained for ν-SVM on DCNN features of Dataset 2

- **Image Representation of the Kernel Gram Matrix** for Gaussian kernel on training dataset is shown in Figure 41.

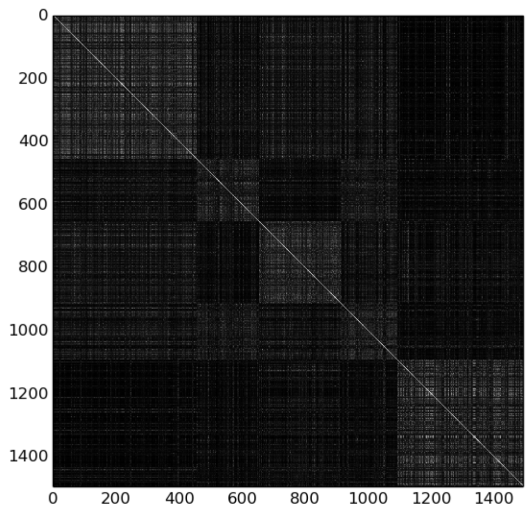

Figure 41: Kernel Gram Matrix for Gaussian Kernel on DCNN features of Dataset 2

- **Observations and Inferences:**
  ν-SVM with Gaussian kernel gives 61.24% test accuracy for DCNN features of Dataset2.
  We can also see that the image of the kernel gram matrix is close to ideal. This is because the features extracted by the DCNN are good in terms of distinguishing between the classes.
  The model performs reasonably well despite class imbalance due to a clear separation of the classes as seen in the kernel gram matrix.
  Also, we can see that ν SVC performs better than the DCNN model on the extracted feature from the DCNN.

5 Classification using Deep Boltzmann Machine (DBM)

A Deep Boltzmann Machine with three RBMs is used to learn the initial weights of a feedforward neural network. Then the feedforward neural network (with added output layer) is trained using the backpropagation algorithm. The number of nodes in the RBMs are chosen based on the performance of model on validation set. The best validation accuracy obtained was 90.48%.
In addition to this task, we also tried classification using a Neural Network on the features derived by the RBM (i.e. the activations / probabilities of the hidden nodes in the final layer of the RBM).

- **Confusion Matrix** obtained for test data is shown in Table 7.

Class	1	2	3	4	5
1	31	4	1	1	4
2	5	22	2	0	3
3	0	1	13	0	0
4	0	0	1	21	0
5	3	0	1	0	13

Table 7: Confusion Matrix obtained for NN with weights initialized using DBM for images in Dataset 3

- **Observations and Inferences:**
  The feedforward Neural Network with weights initialized using the Deep Boltzmann Machine gives 79.37% test accuracy.
  This works much better than classification using a Neural Network with randomly initialized weights, which gives ¡ 50% accuracy.
  It is hypothesised that initializing a Neural Network with these weights helps it avoid getting stuck in local minima present near the random starting weights.
  Classification using the derived features as inputs to a randomly initialized Neural Network also gave much better results than using the original features as inputs to a randomly initialized Neural Network.
  Hence we can also conclude the the DBN manages to extract a good higher level representation from the given binary data.