CS6011 Kernel Methods for Pattern Analysis
Assignment-2
C-SVM, ν-SVM, Convolutional Neural Network, RBM

Arulkumar S (CS15S023), Divya Saglani (CS15M041), Nitish (CS15S028)

3rd March 2016

1 Introduction

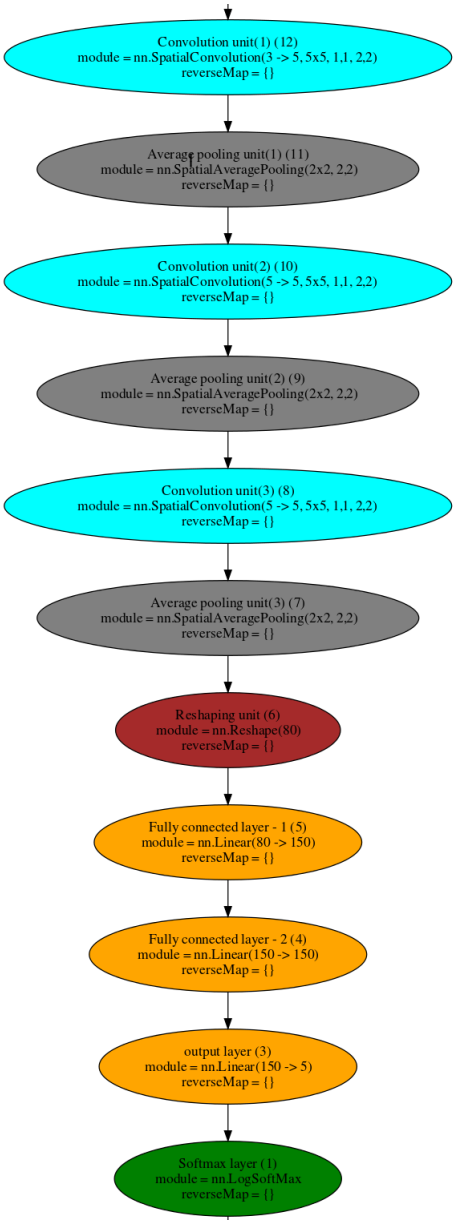
2 Analysis of SVM with 2D data

3 PCA, Auto-encoder, Stacked Auto-encoder

4 Deep convolutional neural network (DCNN)

4.1 Model

The model skeleton is shown below.



The model consists of three sets of "convolution + average pooling" layers followed by two Fully connected layers and output layer. The output layer gives Softmax output (5 nodes) for each class.

The model is trained using Stochastic Gradient Descent optimization technique with the following parameters.

- Learning Rate = 0.02
- Batch Size = 128
- Weight decay = $5e^{-4}$
- Momentum = 0.9

4.2 Data

The given image data (of size 200 x 200 pixels) are resized into 32 x 32 pixels sized images and splitted into Training (75%), Validation(12.5%) and Test(12.5%) set.

Total number of images given are:

- cow: 302 (train = 226 images, validation = 38 images, test = 38 images)
- boat: 500 (train = 375 images, validation = 62 images, test = 63 images)
- sheep: 316 (train = 237 images, validation = 39 images, test = 40 images)
- bottle: 757 (train = 567 images, validation = 95 images, test = 95 images)
- bird: 753 (train = 564 images, validation = 94 images, test = 95 images)

4.3 Model selection

- The model selection is carried out by varying the "number of output feature maps" in 3 convolution layers and "number of nodes" 2 Fully connected layers.
- For convolution layers, the "number of output feature maps" applied are 5, 15. & the filter sizes applied are 3, 5.
- For Fully connected layers, the "number of nodes" applied are 150, 300.
- Totally, 64 CNNs are trained based on the combinations from Convolution layers and Fully connected layers.
- The best model is selected based on the validation accuracy of the trained model.

According to validation criteria, the best performing models are found to be,

Model Number	conv1	conv2	conv3	filter-size	FC1	FC2	Training(%)	Validation(%)
1	5	15	15	5	150	150	51.142	53.048
2	5	15	5	5	300	150	50.634	53.048

The above models will be referred as **model-1** & **model-2** respectively in upcoming pages.

4.4 Confusion matrices

4.4.1 Model-1 (after 167 epochs)

4.4.1.1 Training set

	cow	boat	sheep	bottle	bird	Total correct (%)
cow	70	26	11	52	67	30.973
boat	5	222	1	46	101	59.200
sheep	33	31	21	55	97	8.861
bottle	8	44	5	399	111	70.370
bird	19	83	19	148	295	52.305
					Accuracy	51.142

4.4.1.2 Validation set

	cow	boat	sheep	bottle	bird	Total correct (%)
cow	9	5	4	9	11	23.684
boat	3	39	1	12	7	62.903
sheep	2	6	12	9	10	30.769
bottle	1	11	3	64	16	67.368
bird	3	11	5	25	50	53.191
					Accuracy	53.048

4.4.1.3 Test set

	cow	boat	sheep	bottle	bird	Total correct (%)
cow	12	5	2	7	12	31.579
boat	2	29	0	6	26	46.032
sheep	11	6	5	7	11	12.500
bottle	2	9	0	71	13	74.737
bird	4	12	2	30	47	49.474
					Accuracy	49.546

4.4.2 Model-2 (after 183 epochs)

4.4.2.1 Training set

	cow	boat	sheep	bottle	bird	Total correct (%)
cow	68	30	17	55	56	30.088
boat	9	213	3	41	109	56.800
sheep	31	27	34	62	83	14.346
bottle	8	46	9	397	107	70.018
bird	20	82	25	152	285	50.532
					Accuracy	50.634

4.4.2.2 Validation set

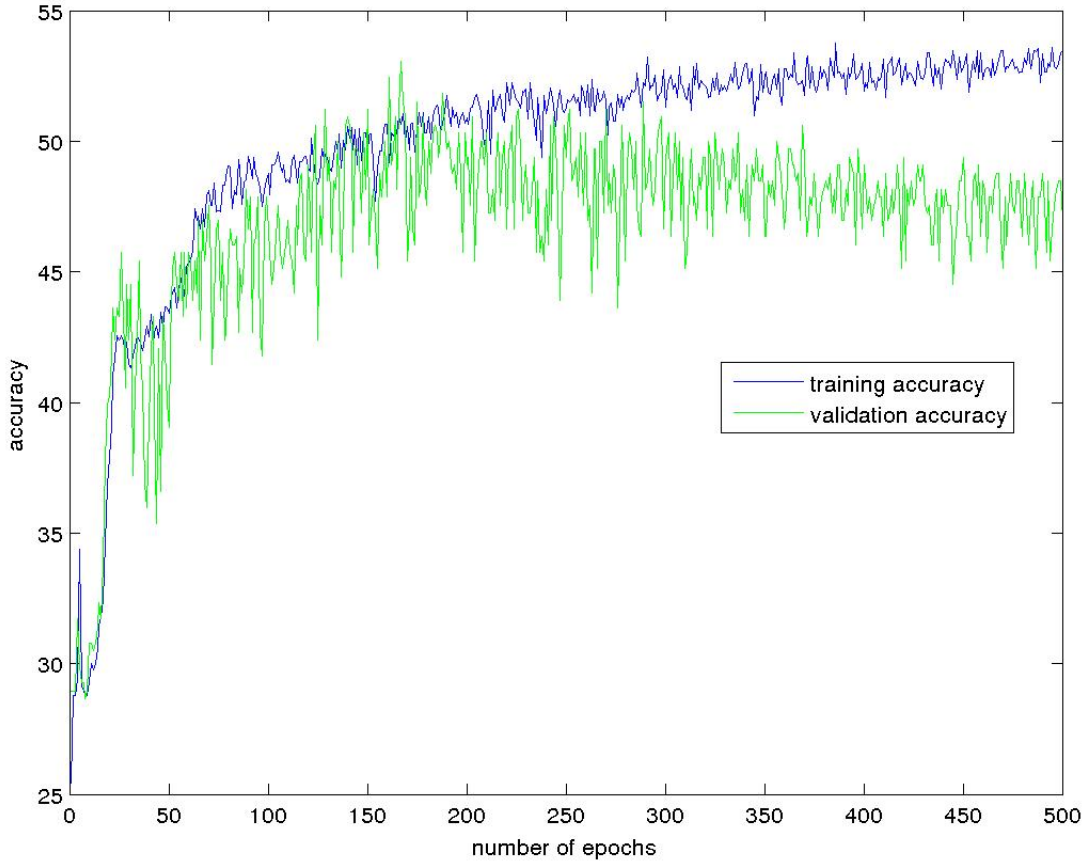
	cow	boat	sheep	bottle	bird	Total correct (%)
cow	10	3	4	10	11	26.316
boat	3	32	2	15	10	51.613
sheep	2	6	11	8	12	28.205
bottle	1	8	2	66	18	69.474
bird	2	10	3	24	55	58.511
					Accuracy	53.048

4.4.2.3 Test set

	cow	boat	sheep	bottle	bird	Total correct (%)
cow	11	2	3	7	15	28.947
boat	2	29	0	6	26	46.032
sheep	8	4	10	6	12	25.000
bottle	3	6	0	71	15	74.737
bird	3	12	3	29	48	50.526
					Accuracy	51.057

4.5 Training and validation accuracy

During training phase, after each epoch of training, the accuracy of validation set is being noted down to detect overfitting on the training data. As shown in the picture, the model starts to overfit on the training data after certain epoch (180 epochs). So, it is important to keep track of validation accuracy and select the appropriate model for testing.



4.5.1 Observations

- During initial epochs of training, the model is biased with the classes having more data (bottle, bird) and predicts only those classes (bottle, bird) for all the data. Once, the gradient is backpropagated for multiple epochs (20 epochs), the model starts to learn all other classes.
- Due to the imbalance in the data, the classes which has more data (bottle, bird) are learned effectively than the classes having less data.
- For 3 convolution layers, the model is overfitting after ˜180 epochs. It can be accounted to the fact that only less amount of data is available for training.

4.6 Convolution layer-1 filters (Model-1)

The visualization of convolution layer-1 filters is shown below. From the filters, it can be inferred that the first convolution layer filters learn the color information (mainly, green, reddish, blue colors) in the training images.

4.7 Hidden layer output visualization

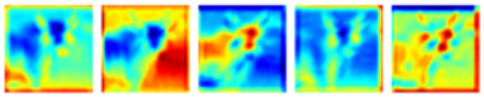
The output of convolution layers for each class image(s) is shown below. As we can see in the layer outputs, the initial convolution nodes are able to learn low level features such as Edges, the second convolution layer is able to give high activation value when a particular object is found. The linear fully connected layers are acting as classifiers for the classes based on the learned features from convolution layers.

4.7.1 Model-1

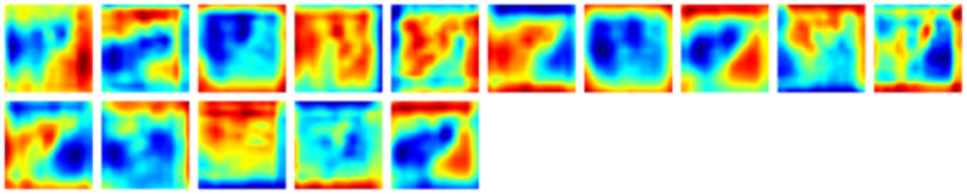
4.7.1.1 class 'cow'

cow

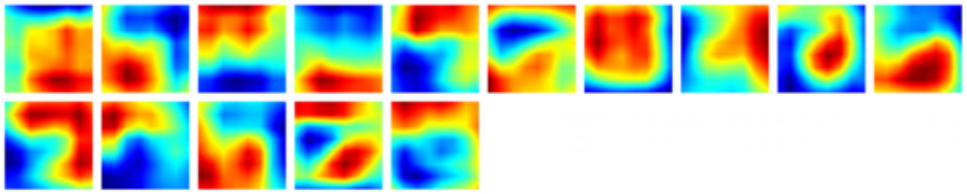
convolution layer 1



convolution layer 2



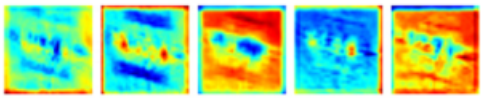
convolution layer 3



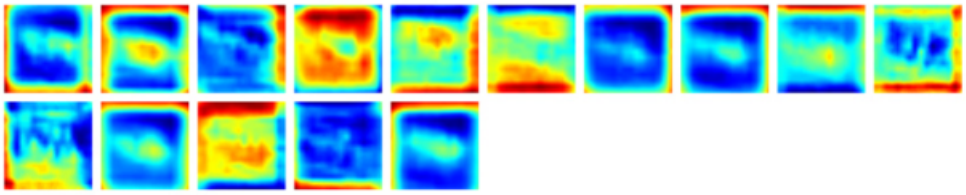
4.7.1.2 class 'boat'

boat

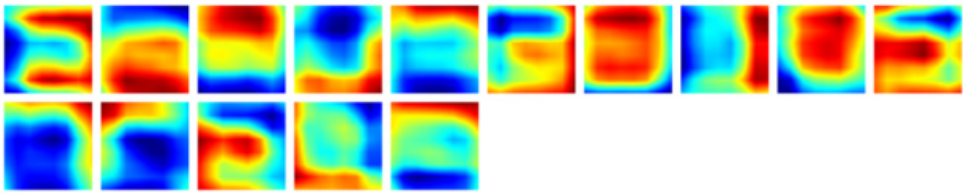
convolution layer 1



convolution layer 2



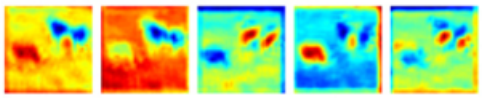
convolution layer 3



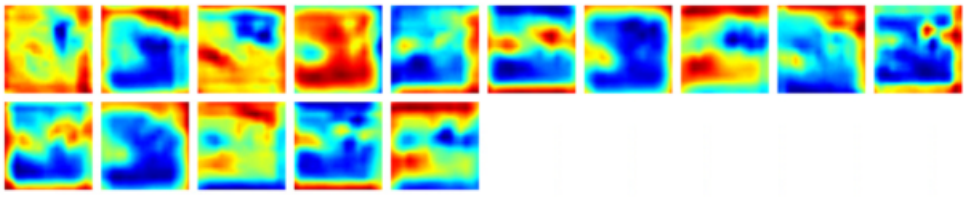
4.7.1.3 class 'sheep'

sheep

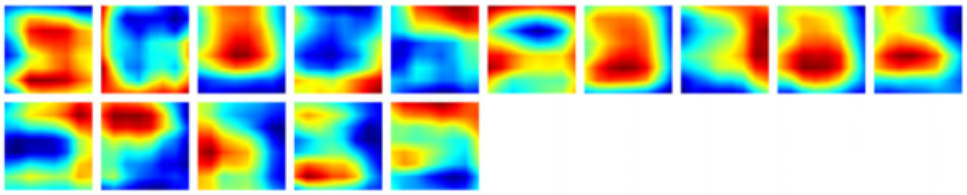
convolution layer 1



convolution layer 2



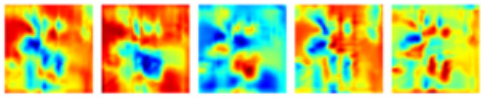
convolution layer 3



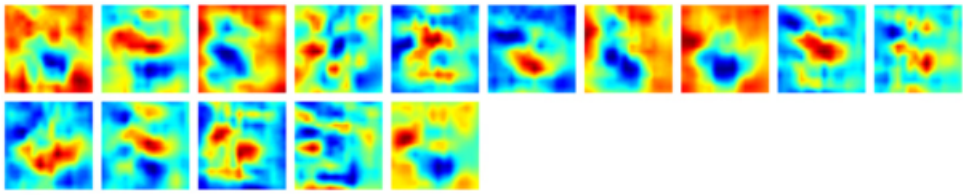
4.7.1.4 class 'bottle'

bottle

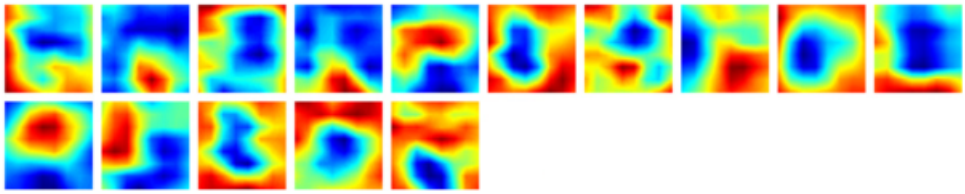
convolution layer 1



convolution layer 2



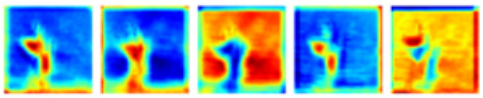
convolution layer 3



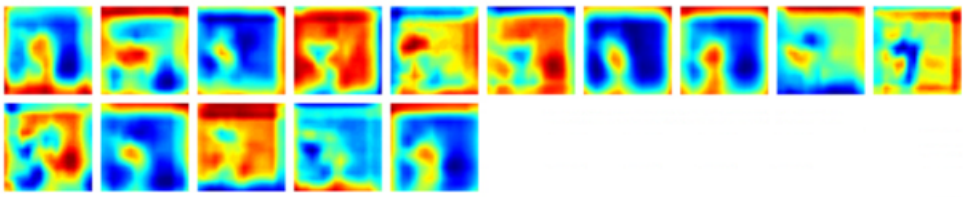
4.7.1.5 class 'bird'

bird

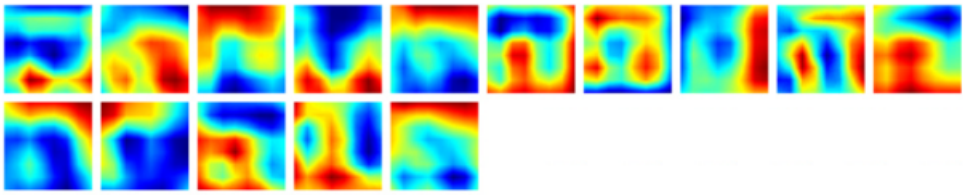
convolution layer 1



convolution layer 2



convolution layer 3

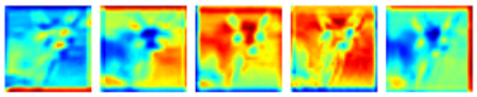


4.7.2 Model-2

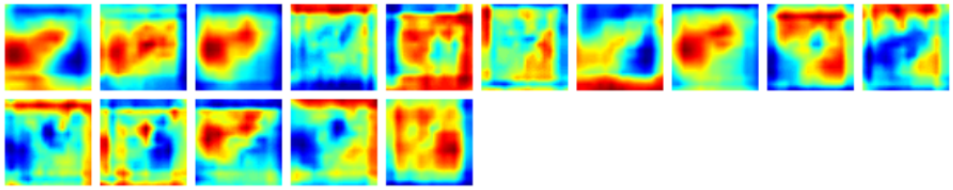
4.7.2.1 class 'cow'

cow

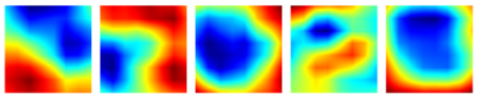
convolution layer 1



convolution layer 2



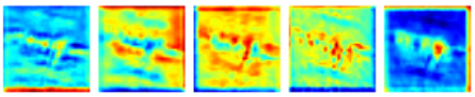
convolution layer 3



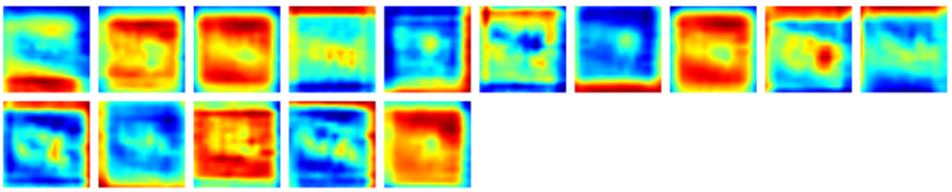
4.7.2.2 class 'boat'

boat

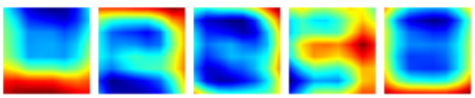
convolution layer 1



convolution layer 2



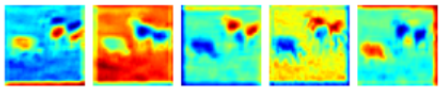
convolution layer 3



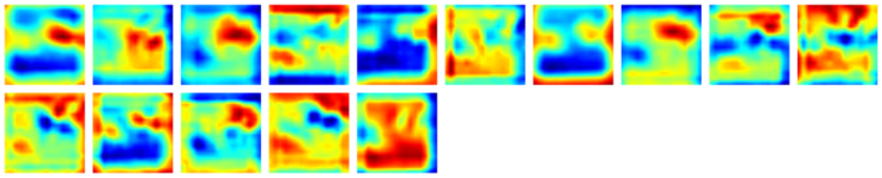
4.7.2.3 class 'sheep'

sheep

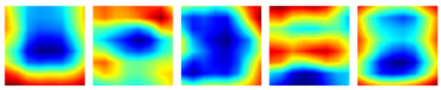
convolution layer 1



convolution layer 2



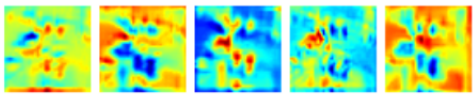
convolution layer 3



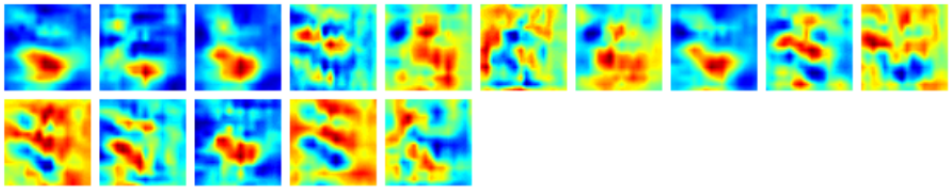
4.7.2.4 class 'bottle'

bottle

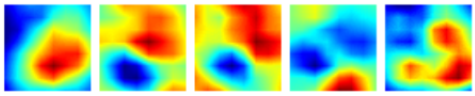
convolution layer 1



convolution layer 2



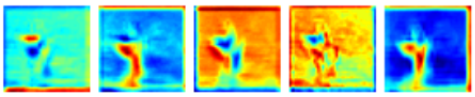
convolution layer 3



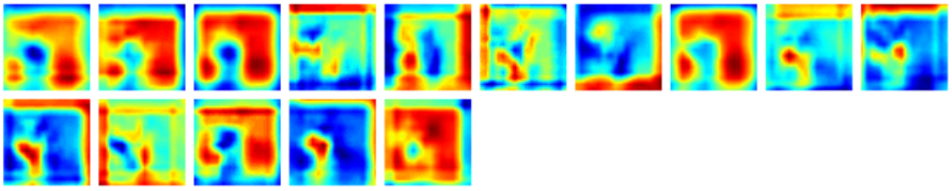
4.7.2.5 class 'bird'

bird

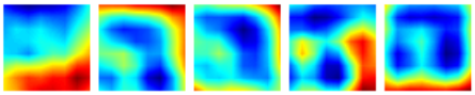
convolution layer 1



convolution layer 2



convolution layer 3



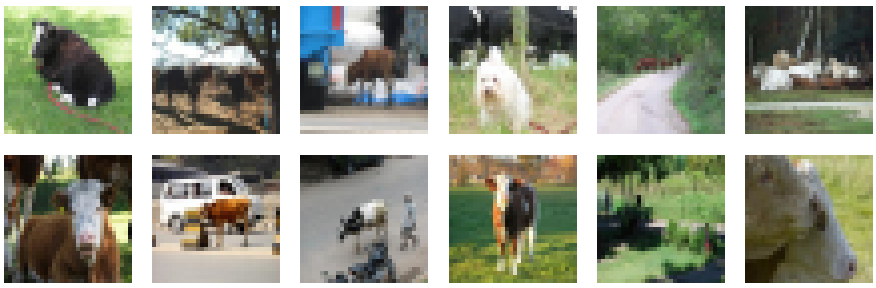
It can be clearly noticed from the convolution layer outputs (especially from conv2), the model is able to locate the object and provide higher activation value on object's location.

4.8 A closer look at misclassified images

4.8.1 actual: 'cow' & predicted : 'boat'

4.8.2 actual: 'boat' & predicted : 'cow'

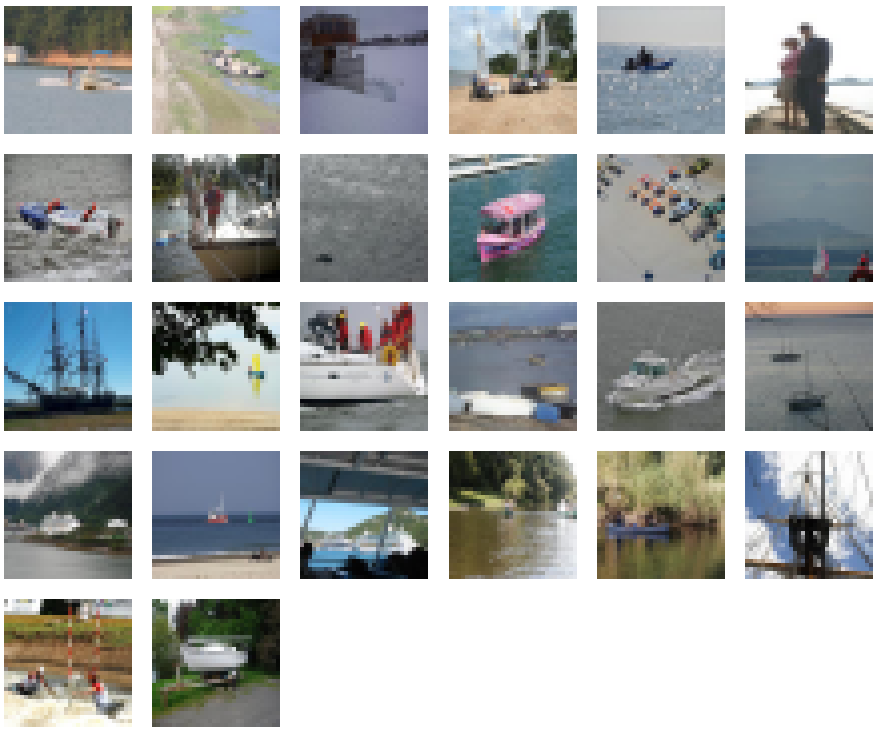
4.8.3 actual: 'cow' & predicted : 'bird'



4.8.4 actual: 'bird' & predicted: 'cow'

4.8.5 actual: 'bird' & predicted : 'boat'

4.8.6 actual: 'boat' & predicted : 'bird'



It can be inferred from the images that the 'boat' and 'bird' classes contain 'blue' color in most of their pictures which makes the model to misclassify these images.

4.8.7 actual: 'cow' & predicted : 'sheep'

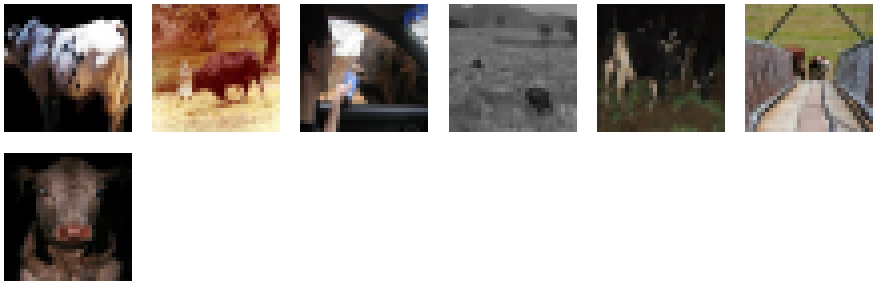
4.8.8 actual: 'sheep' & predicted : 'cow'

4.8.9 actual: 'bird' & predicted : 'sheep'

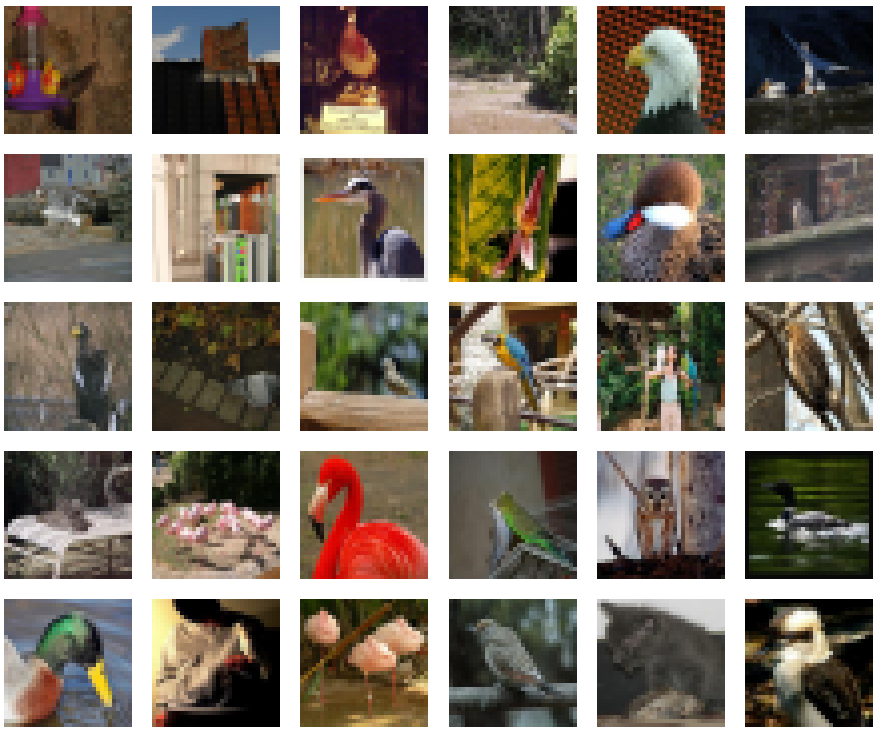
4.8.10 actual: 'sheep' & predicted : 'bird'

4.8.11 actual: 'bottle' & predicted : 'cow'

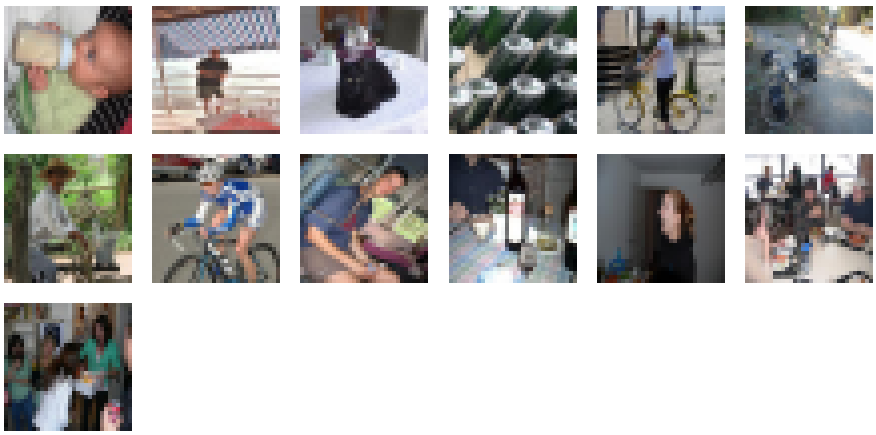
4.8.12 actual: 'cow' & predicted : 'bottle'



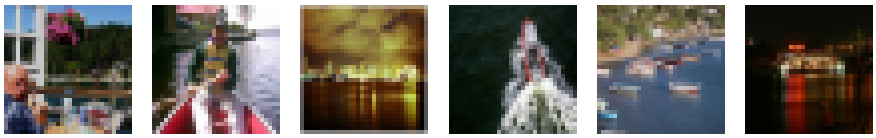
4.8.13 actual: 'bird' & predicted : 'bottle'



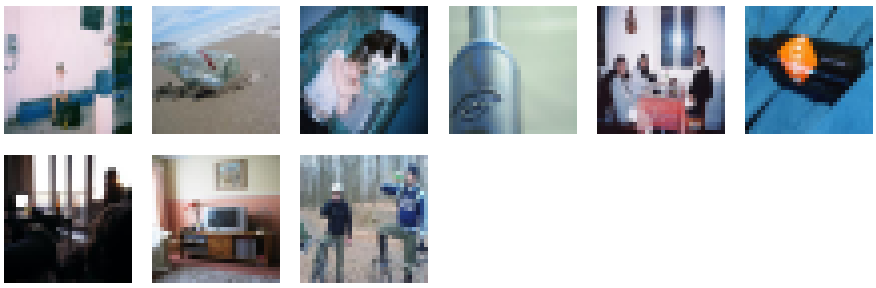
4.8.14 actual: 'bottle' & predicted : 'bird'



4.8.15 actual: 'boat' & predicted : 'bottle'



4.8.16 actual: 'bottle' & predicted : 'boat'



4.8.17 actual: 'sheep' & predicted : 'bottle'

4.8.18 actual: 'sheep' & predicted : 'boat'

4.9 Effect of Data augmentation strategies

- Since the data set is imbalanced (i.e., class 'bird' has double the data of class 'cow'), we tried to balance it by using Label preserving transformations. During data augmentation, the original images are translated randomly by [-0.1H to +0.1H] (i.e., 3 pixels) and each image is also horizontally flipped.

- After the data augmentation, the number of images is increased to ˜20000 with ˜3000 images per class. the model (conv1=5, conv2=15, conv3=15, FC1=150, FC2=150) is trained using the augmented dataset and the validation performance is measured. Eventhough, we increase the data by augmenting, the training accuracy (48%) and validation accuracy (46%) are not increased.

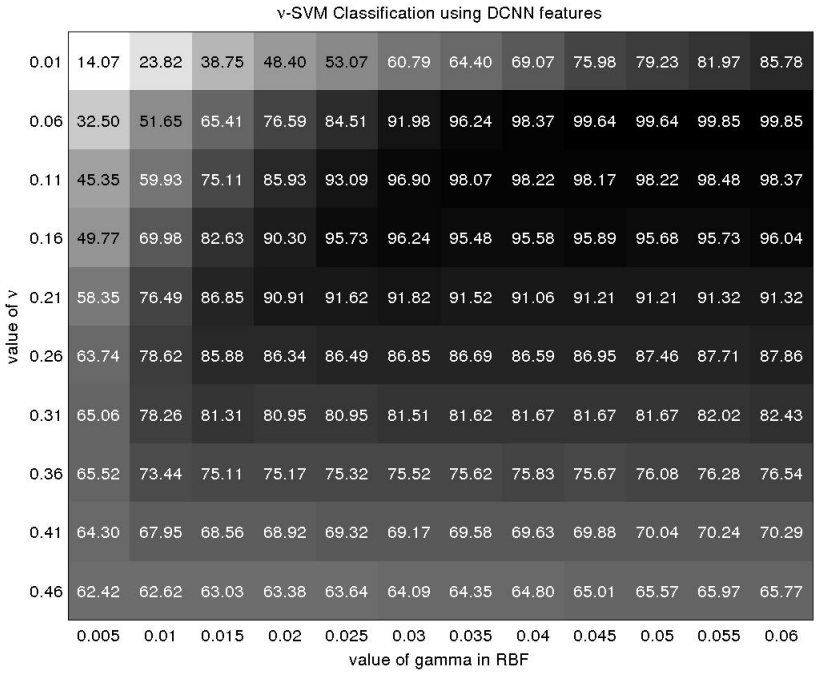
5 ν-SVM classification using DCNN features

The DCNN features output from Fully connected layer-2 (150 dimensional data) of Model-1 is considered for classification using ν-SVM.

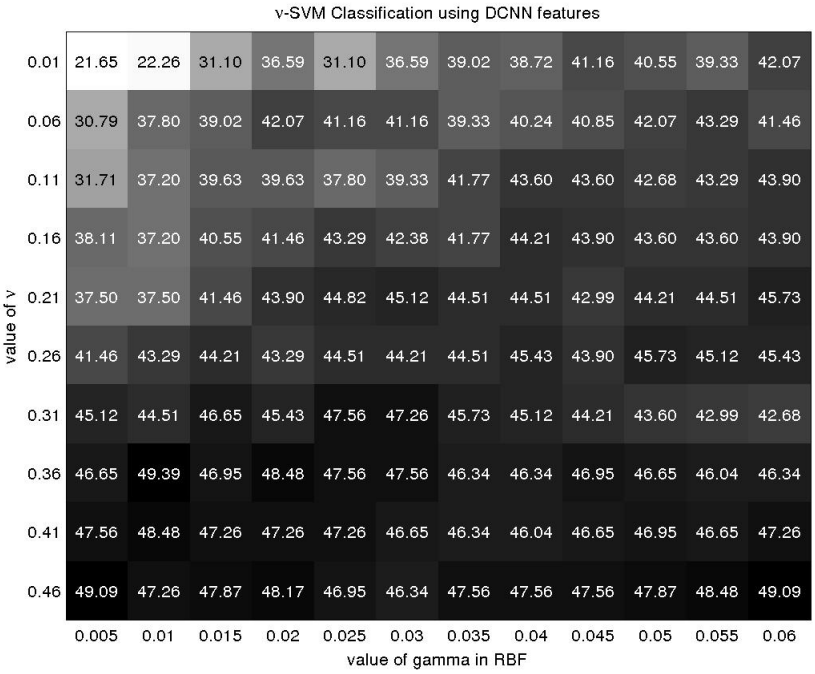
$\nu - SVM$ from the package "LibSVM" is used to classify the features obtained from DCNN.

5.1 Accuracy: ν vs RBF scale factor(gamma)

5.1.1 Training accuracy



5.1.2 Validation accuracy



5.2 Parameter selection

From the training accuracy, validation accuracy images, we can notice that

- when the ν parameter is increased (to ˜0.5), we are allowing more outliers, inturn the number of support vectors is increased.
- We need to find the balance between the number of support vectors and the accuracy. usually, $\nu = 0.05$ to 0.3 is preferred. So we shall choose a model which gives greater performance on validation set with $0.05 <= \nu <= 0.3$
- From the images, it can be seen that when ν is very small (˜0.06) & gamma is high, the model tries to fit training data perfectly (i.e., overfit). It is confirmed by looking at validation accuracy. Eventhough, training accuracy is ¿95%, validation accuracy is not increased.

The below parameters are choosen based on validation set performance of various models.

- $\nu = 0.26$
- RBF kernel scale factor (gamma) = 0.02

5.3 Confusion matrices

5.3.0.1 Training set

	cow	boat	sheep	bottle	bird
cow	171	9	1	24	21
boat	6	313	4	21	31
sheep	0	17	174	21	25
bottle	1	14	4	519	29
bird	4	15	6	16	523
				Accuracy	86.33

5.3.0.2 Validation set

	cow	boat	sheep	bottle	bird
cow	11	4	5	6	12
boat	7	29	2	12	12
sheep	5	6	13	5	10
bottle	6	15	6	47	21
bird	11	17	2	22	42
				Accuracy	43.297

5.3.0.3 Test set

	cow	boat	sheep	bottle	bird
cow	10	5	7	12	4
boat	3	35	3	5	17
sheep	7	8	9	3	13
bottle	6	12	5	61	11
bird	6	13	7	21	48
				Accuracy	49.245

5.4 Observations

- when ν is very less (˜0.05), the model tends to overfit on training data.
- When ν is increased, the number of support vectors, as well as the accuracy on validation data increases, due to the fact that we are allowing more outliers.