Assignment 2 - Report

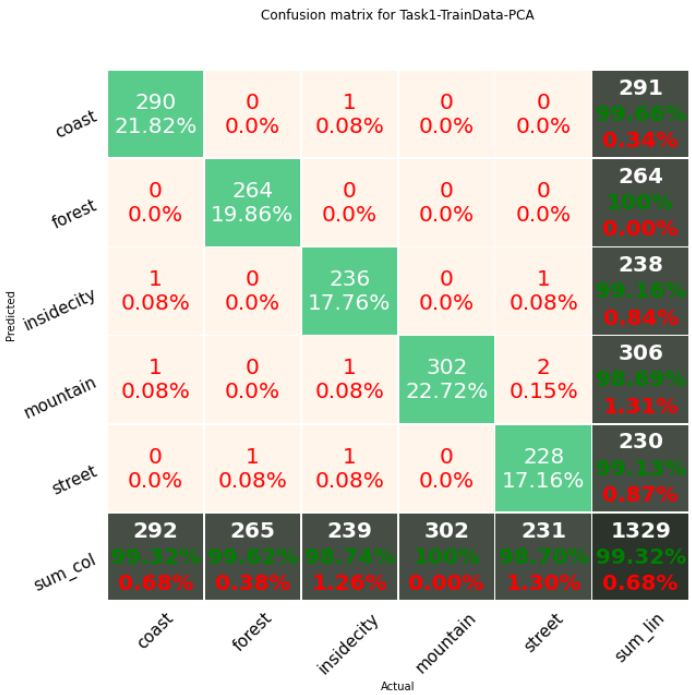
By
Allumalla Ravi Kiran - CS17B038
Allada Praharsh - CS17B036
Arabhi Subhash - CS17B005

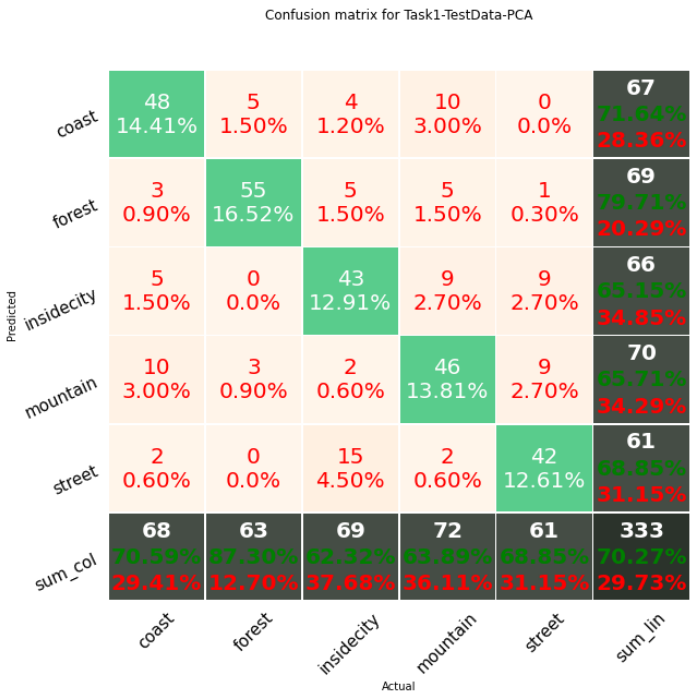
Task 1: Dimension reduction using PCA and AANN and Classification with MLFFNN on extracted data

Note Points

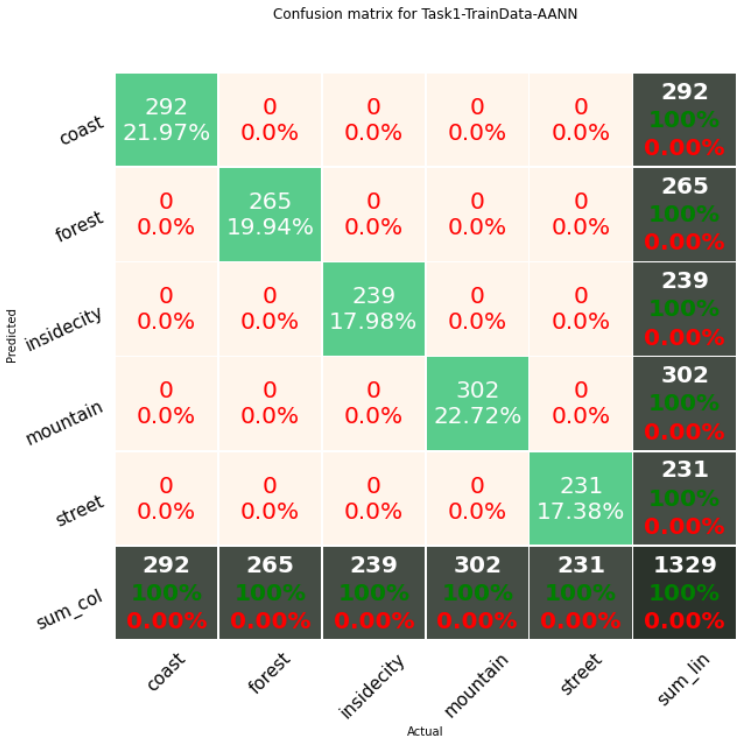
1. The first one, dimensional reduction using PCA is a linear method whereas the second is using AANN a nonlinear one
2. Here we reduced dimensions from 828 to 500
3. Tuned Parameters - PCA (Epochs for convergence based on running loss)
4. h1_dim = 50
5. learning_rate = 0.001
6. num_epochs = 600
7. batch_size = 32
8. Parameters - AANN
9. h1_dim = 50
10. learning_rate = 0.001
11. num_epochs = 500
12. batch_size = 32
13. We can see from the results that, with AANN the convergence is faster
14. The accuracy should also be higher for AANN but we got 70 and 66, which are similar

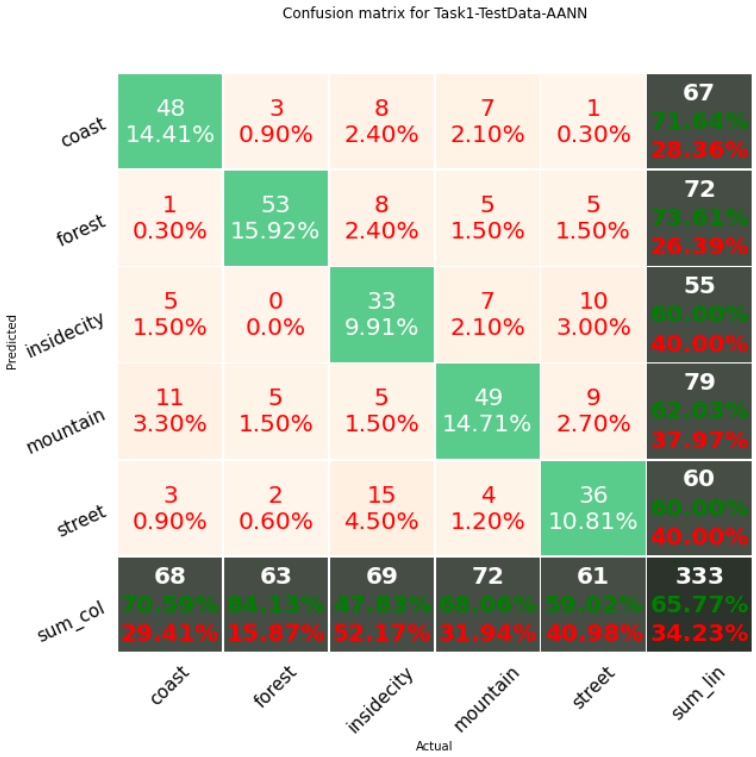
PCA



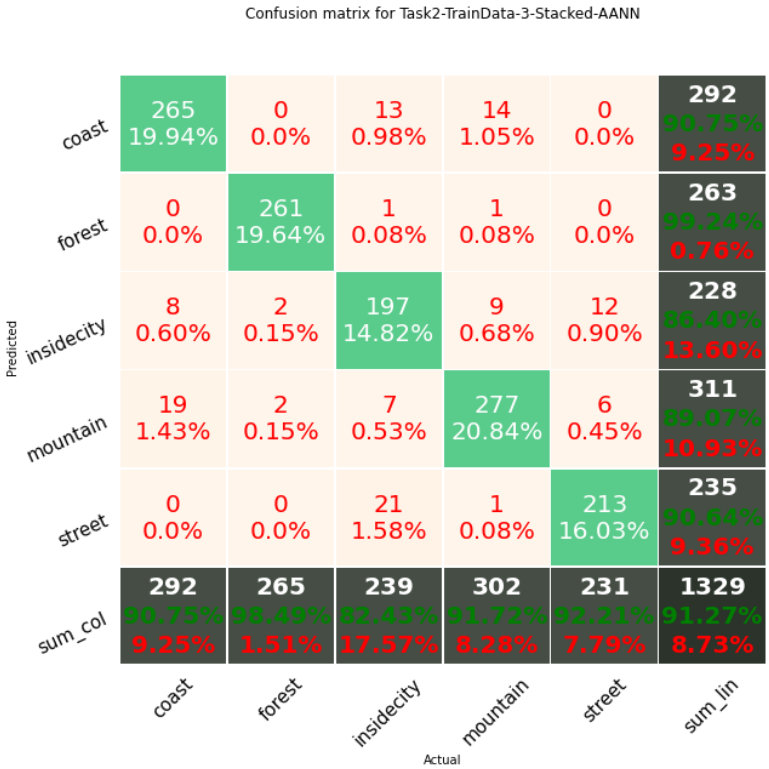


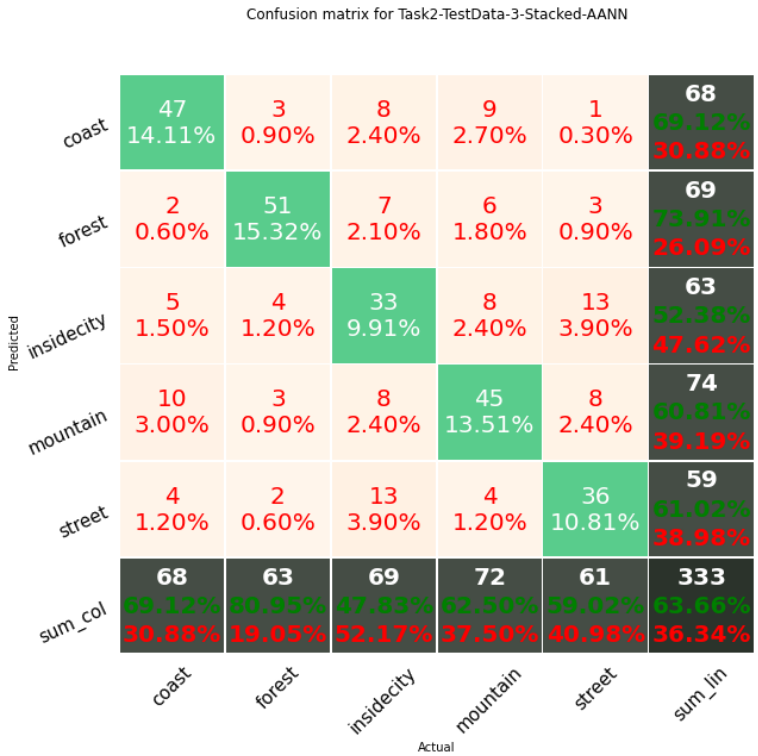
AANN





Task 2: Stacked Autoencoder based pretraining of a DNN based classifier for extracted data



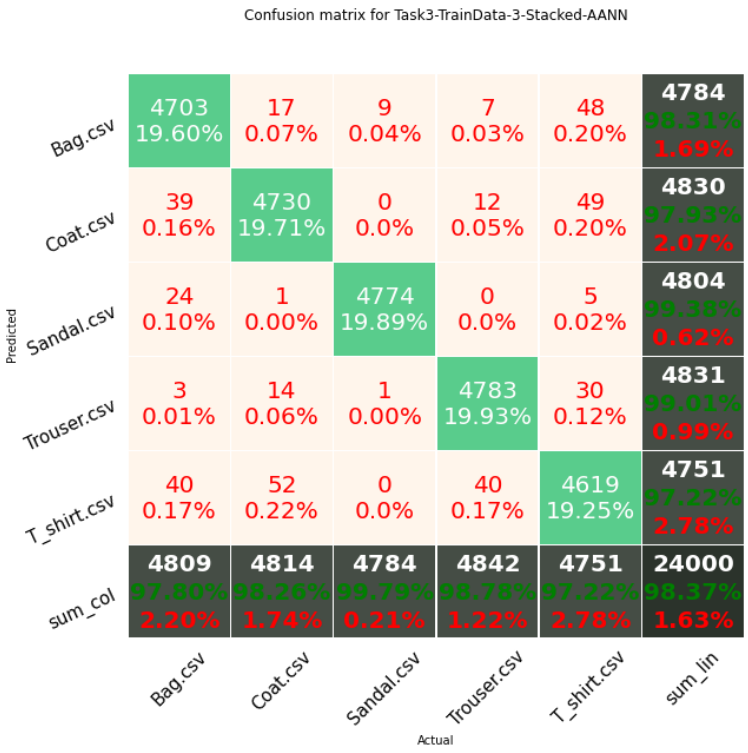


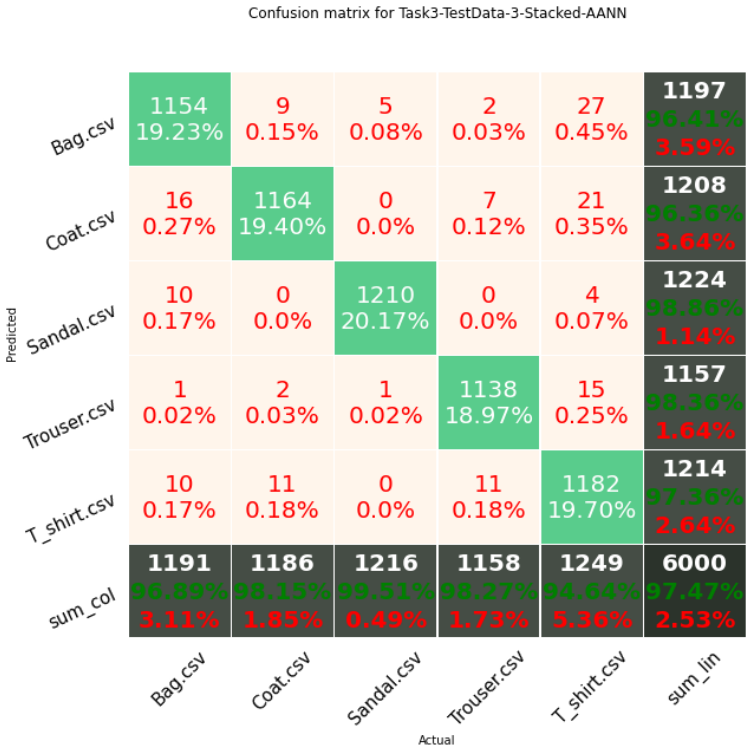
Note Points

1. The 3 AANNs are similar, together with reduced the dimensions to 300
2. Parameters - Adagrad, Mean Square error loss

3. h1_dim = 1000, 800, 600
4. h2_dim = 800, 600, 400
5. h3_dim = 700, 500, 300
6. h4_dim = 900, 600, 400
7. batch_size = 64
8. learning_rate = 0.001
9. num_epochs = 3000, 1500, 1000
10. num_workers = 8
11. The convergence is faster for 2nd and third encoders because of reduced dimensions. This makes it easier to train compared to single AANN with many hidden layers
12. The accuracy is almost equal to that of task 1
13. A similar DNN is used for all the problems below for the sake of comparison with parameters -
14. h1_dim = 50
15. h2_dim = 28
16. learning_rate = 0.001
17. momentum = 0.9
18. num_epochs = 1000 (around)
19. batch_size = 32
20. The optimizer is Schaostic Gradient Decent without Nesterov momentum and criterion is cross-entropy loss

Task 3: Stacked Autoencoder based pretraining of a DNN based classifier for Pixel Data



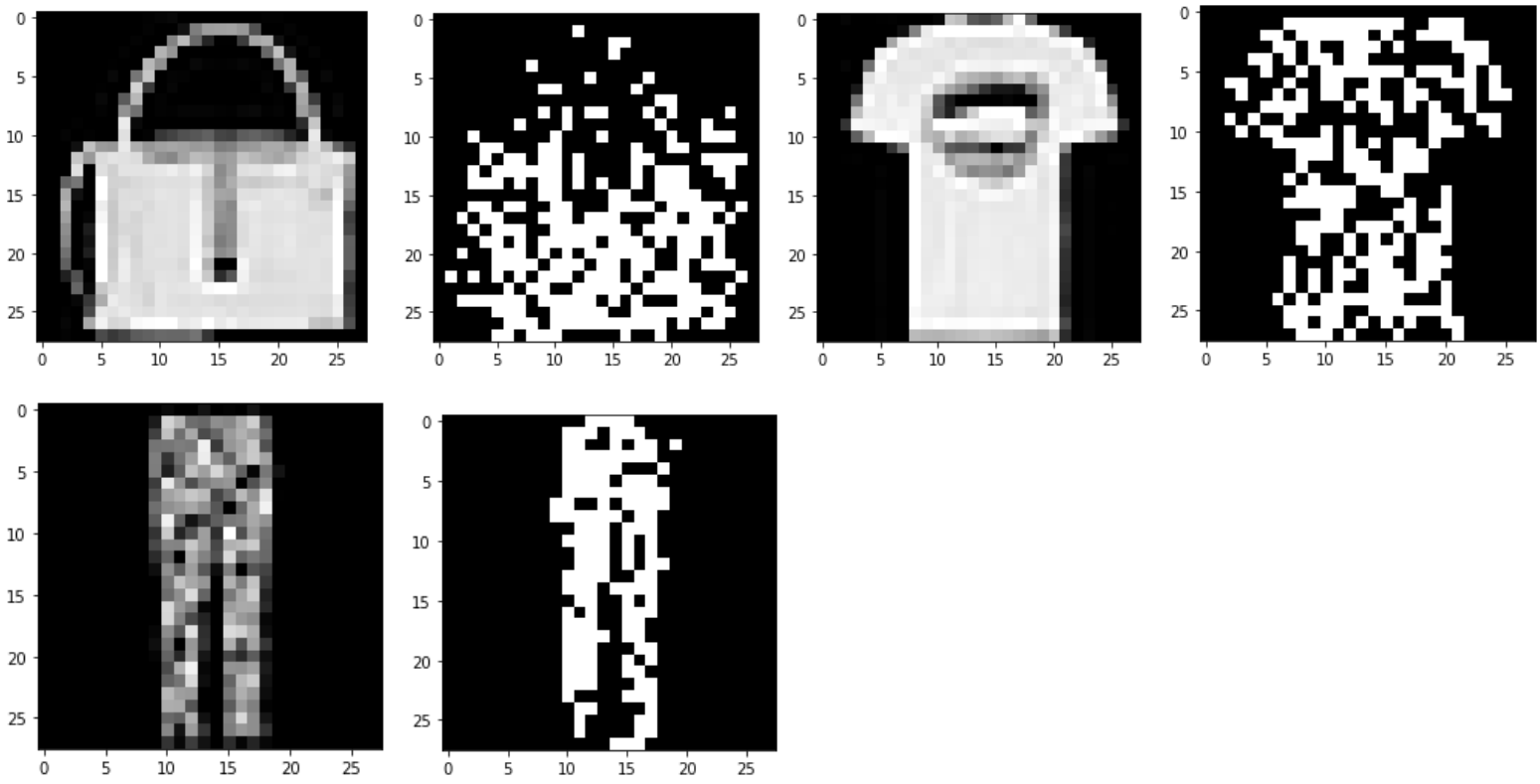


Note Points

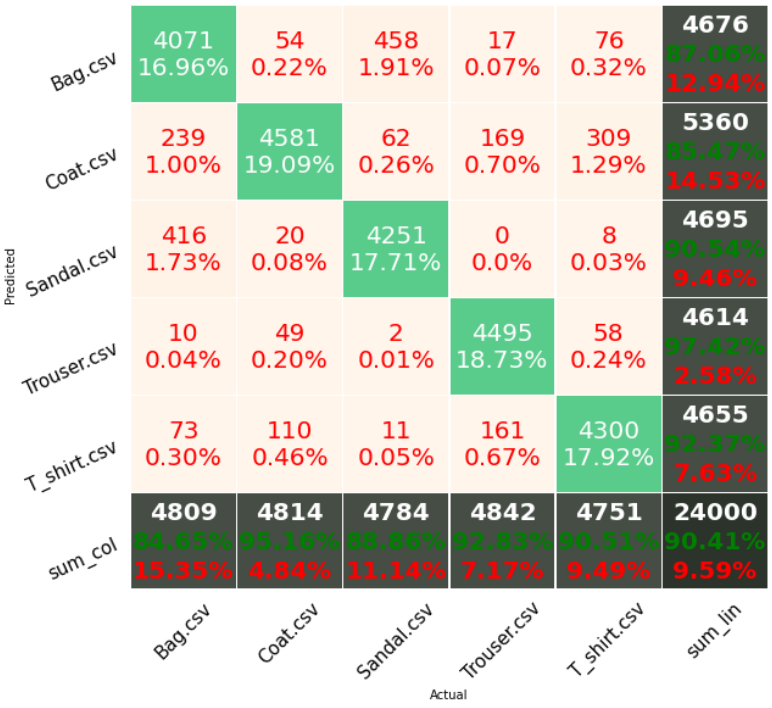
1. Convergence is taking longer for raw pixel data even with proper initialization (owing to the fact that data consists of integers, 0-255) so before passing on to first autoencoder data is gaussian normalized
2. The parameters are almost the same except for the fact that we are starting with 23x23 = 784 dimensions
3. Convergence is much faster (within 1000 epochs) because of 2 factors, first is fewer input dimensions and normalization

4. The accuracy (97) is much better compared to that of extracted data (around 70) maybe because the functions used for extraction couldn't preserve the object information

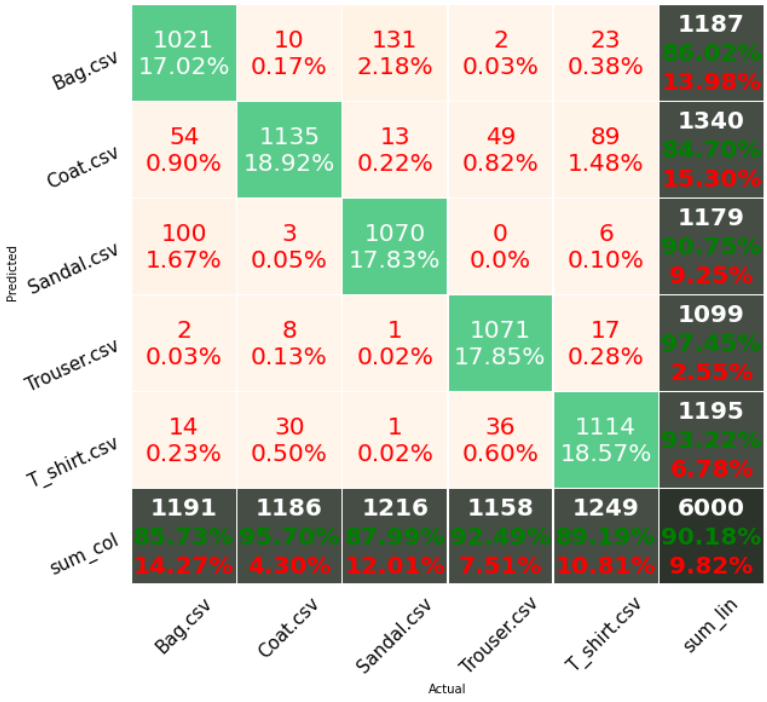
Task 4: Stacked RBM based pre-training of a DNN based classifier for Pixel data, using Binary-Binary RBMs



Confusion matrix for Task4-TrainData-3-Stacked-BBRBM



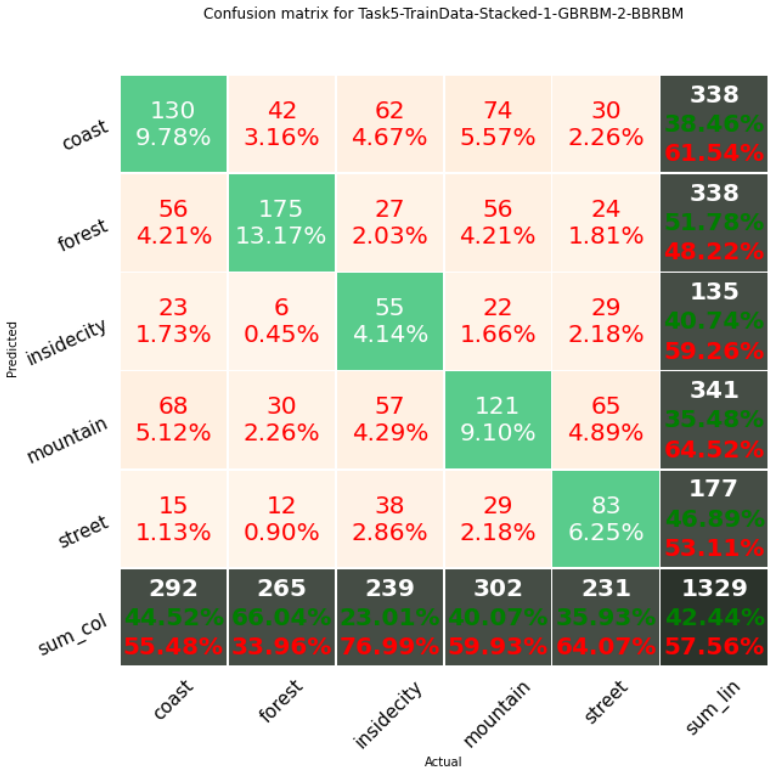
Confusion matrix for Task4-TestData-3-Stacked-BBRBM

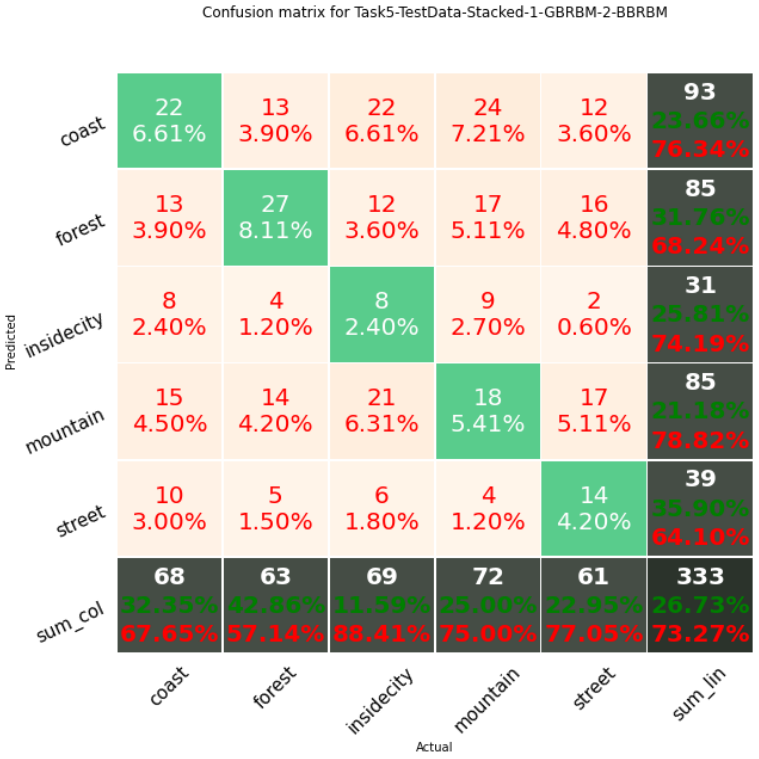


Note Points

1. The first few images are pixel data and their reconstruction after passing through the first BBRBM. We can see that though data is converted to binary format (much less space) the object information is preserved this is the reason for accuracy of 90%
2. Min-max normalization is applied before passing on to the first RBM
3. Parameters
4. n_vis = 784, 600, 400
5. n_hid = 600, 400, 300
6. batch_size = 64
7. learning_rate = 0.001
8. num_epochs = around 100 for all three RBMs
9. k = 3 (3 step contrastive divergence)
10. BBRBMs converge very faster, we can see that in the number of epochs, compared to that of AANN. This is mainly because of the binary aspect

Task 5: Stacked RBM based pre-training of a DNN based classifier for extracted data, using Gaussian-Binary RBMs





Note Points

1. The stack has 1 GBRBM and 2 BBRBM and so far it is pretty bad classifier with 40% accuracy in training
2. The reason may be that the relationship between different extracted points might not represent the image
3. The parameters are the same as that of the previous one except for the tuned learning rate of GBRBM, which is 0.002
4. Convergence is faster similar to the previous task
5. The difference is in sampling hidden and calculating free energy rest all is same between GBRBM and BBRBM