Summer Semister1-Deep Learning Assignment1-IITJ

M22AI608-Kadapala Rakesh Reddy

Link to Data Source: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/#testdata

Step1: Model Selection: Choosing a CNN model trained on the ImageNet classification dataset.

- **resnet50:** Residual Networks is a convolutional neural network, used to load a pretrained version of the neural network trained on more than a million images.

torchvision.models: This Package provides pre-trained model for Image Classification and Object Detection.

Step2: Extracting features using the pre-trained CNN model:

- Loading the Dataset PASCAL VOC 2007 contains JPEG Images, Annotations folders.
- Extracting features and class labels from images using the ResNet50 model.
- The pre-trained ResNet50 model is loaded, which has been trained on a large dataset called ImageNet.
- In the Function **extract_features_and_labels** , takes image paths, annotation paths, and the pre-trained model as input, and loops over the images and annotations to obtain feature vectors and class labels.
- Recognizing objects from a number of visual object classes in realistic scenes

The object classes that have been selected are:

- Person: person
- Animal: bird, cat, cow, dog, horse, sheep
- Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
- Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

```
C:/Users/Admin/M22AI608-DeepLearning-Assignment1/VOCdevkit/VOC2007\JPEGImages\009962.jpg
1/1 [==============================] - 1s 588ms/step
chair  classlabel
person  classlabel
person  classlabel
person  classlabel
diningtable  classlabel
```

- Printing the shape of the extracted features and class labels and getting all

```
Features shape: (4952, 2048)
Class labels shape: (14976,)
```

Step3: Generate normalized Features and Labels Binary:

❖ Converting class labels into binary vectors and normalizing the features

- Label binarization is done using the LabelBinarizer from scikit-learn. It converts the class labels into binary format, where each class is represented as a binary vector.
- Feature normalization is done using the Standard Scaler from scikit-learn. This scales the features such that they have zero mean and unit variance.

Step4: Converting features and labels to NumPy arrays:

❖ converting features and class labels to NumPy arrays and aligns them to have the same size.

```
Features shape: (4952, 2048)
Class labels shape: (4952,)
```

Step5: perform multi-label binarization on labels:

Multi-label binarization converts categorical labels into a binary matrix. Each label becomes a separate binary column, indicating its presence (1) or absence (0).

- converting the class labels into a binary matrix representation,
  Here The shape of the binarized labels:

```
Binarized labels shape: (4952, 20)
```

Step6: Split data into train and validation sets:[features_aligned]

❖ The data is split into training and validation sets using scikit-learn's train_test_split function. The features (**features_aligned**) and corresponding labels (labels_binary) are divided into X_train (training features), X_val (validation features), y_train (training labels), and y_val (validation labels).

- The test_size parameter specifies the proportion of the data to be used for validation (0.2=20%).
- The random_state parameter ensures reproducibility of the split.

```
Training set shape: (3961, 2048) (3961, 20)
Validation set shape: (991, 2048) (991, 20)
```

Step7: Train binary SVM classifiers:

❖ In this step binary Support Vector Machine (SVM) classifiers for each class in a loop. For every class, it prepares the training data by extracting the respective labels from y_train. Then, an SVM classifier is created with a linear kernel and trained using the training data. The trained classifiers are appended to a list and saved individually using the joblib.dump() function with unique filenames.

- **joblib** is a library in Python used to save Python objects to disk.
- **joblib.dump()** is a function allows Store and reuse the model without retrain it.

Step8: Step 5: Evaluate classification accuracy and confusion matrix:

- Loading the SVM Classifiers trained for each class. Then I predict the class labels for the validation set using these classifiers and The predicted labels are converted into a numpy array for further analysis.
- I calculated the overall accuracy to check the how much accurately classifiers predicting the labels across all classes and also calculated the accuracy for each individual class.
- Finally Calculated confusion matrix, which presents a detailed view of the predicted and true labels. It helps in understanding the classification performance, identifying any patterns of misclassification, and evaluating the effectiveness of the SVM classifiers.

Confusion Matrix:

[[TN FP]

[FN TP]]

```
Overall Accuracy: 0.08375378405650857
Accuracy for class aeroplane: 0.9616548940464178
Accuracy for class bicycle: 0.9525731584258325
Accuracy for class bird: 0.9152371342078708
Accuracy for class boat: 0.9424823410696267
Accuracy for class bottle: 0.900100908173562
Accuracy for class bus: 0.9727547931382442
Accuracy for class car: 0.8012108980827447
Accuracy for class cat: 0.9535822401614531
Accuracy for class chair: 0.7870837537840565
Accuracy for class cow: 0.9566094853683148
Accuracy for class diningtable: 0.9656912209889001
Accuracy for class dog: 0.9132189707366297
Accuracy for class horse: 0.9495459132189707
Accuracy for class motorbike: 0.9495459132189707
Accuracy for class person: 0.5317860746720484
Accuracy for class pottedplant: 0.9212916246215943
Accuracy for class sheep: 0.9656912209889001
Accuracy for class sofa: 0.9465186680121089
Accuracy for class train: 0.970736629667003
Accuracy for class tvmonitor: 0.9656912209889001
Confusion Matrix:
[[17884   945]
 [  816   175]]
```

Difference B/W features_aligned & normalized_features used

features_aligned	normalized_features
features_aligned is obtained by subsetting the original features array to ensure that it has the same number of samples as the class_labels_aligned array.	normalized_features are obtained by applying feature normalization or scaling techniques, specifically using the StandardScaler from sklearn.preprocessing.
features_aligned represents the raw features extracted from the dataset without any additional transformations or normalization applied.	normalized_features represents the features after scaling, where each feature column has been transformed to have a mean of 0 and a standard deviation of 1.

the main difference between features_aligned and normalized_features lies in the processing steps applied to them. features_aligned represents the original raw features, while normalized_features represents the same features but after applying normalization for improved performance in subsequent machine learning tasks.

Another Method

Split data into train and validation sets:[normalized_features]

- **Data Splitting:** The data is split into training and validation sets using the train_test_split function from sklearn.model_selection.**(normalized_features)**
- **Training SVM Classifiers:** Binary SVM classifiers are trained for each class using a linear kernel. The training data for each class is prepared, and the classifiers are trained and saved.
- **Evaluation on Validation Set:** The trained classifiers are evaluated on the validation set. For each class, the validation data is prepared, and the SVM classifier predicts the labels. Classification metrics, such as precision, recall, and F1-score, are calculated using classification_report from sklearn.metrics and printed for each class.
- **Confusion Matrix:** The confusion matrix provides an overview of the predicted and true labels for all classes in a tabular format.

¶ **-:Report:-**

```
Classification Report for class sheep:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       970
           1       0.00      0.00      0.00        21

    accuracy                           0.96       991
   macro avg       0.49      0.49      0.49       991
weighted avg       0.96      0.96      0.96       991

Classification Report for class sofa:
              precision    recall  f1-score   support

           0       0.98      0.96      0.97       972
           1       0.00      0.00      0.00        19

    accuracy                           0.95       991
   macro avg       0.49      0.48      0.49       991
weighted avg       0.96      0.95      0.95       991

Classification Report for class train:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99       974
           1       0.00      0.00      0.00        17

    accuracy                           0.97       991
   macro avg       0.49      0.50      0.49       991
weighted avg       0.97      0.97      0.97       991
```

```
Classification Report for class tvmonitor:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       973
           1       0.05      0.06      0.05        18

    accuracy                           0.96       991
   macro avg       0.52      0.52      0.52       991
weighted avg       0.97      0.96      0.97       991

Confusion Matrix:
[[17803  1026]
 [  948    43]]
```

Classification Report for class aeroplane:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	975
1	0.00	0.00	0.00	16
accuracy			0.97	991
macro avg	0.49	0.49	0.49	991
weighted avg	0.97	0.97	0.97	991

Classification Report for class bicycle:

	precision	recall	f1-score	support
0	0.97	0.98	0.98	966
1	0.00	0.00	0.00	25
accuracy			0.95	991
macro avg	0.49	0.49	0.49	991
weighted avg	0.95	0.95	0.95	991

Classification Report for class bird:

	precision	recall	f1-score	support
0	0.97	0.94	0.96	962
1	0.00	0.00	0.00	29
accuracy			0.91	991
macro avg	0.48	0.47	0.48	991
weighted avg	0.94	0.91	0.93	991

Classification Report for class boat:

	precision	recall	f1-score	support
0	0.96	0.98	0.97	953
1	0.00	0.00	0.00	38
accuracy			0.94	991
macro avg	0.48	0.49	0.48	991
weighted avg	0.92	0.94	0.93	991

Classification Report for class horse:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	965
1	0.08	0.08	0.08	26
accuracy			0.95	991
macro avg	0.53	0.53	0.53	991
weighted avg	0.95	0.95	0.95	991

Classification Report for class motorbike:

	precision	recall	f1-score	support
0	0.97	0.98	0.98	960
1	0.00	0.00	0.00	31
accuracy			0.95	991
macro avg	0.48	0.49	0.49	991
weighted avg	0.94	0.95	0.95	991

Classification Report for class person:

	precision	recall	f1-score	support
0	0.65	0.62	0.64	640
1	0.37	0.40	0.38	351
accuracy			0.54	991
macro avg	0.51	0.51	0.51	991
weighted avg	0.55	0.54	0.55	991

Classification Report for class pottedplant:

	precision	recall	f1-score	support
0	0.96	0.96	0.96	949
1	0.06	0.05	0.05	42
accuracy			0.93	991
macro avg	0.51	0.51	0.51	991
weighted avg	0.92	0.93	0.92	991

Classification Report for class chair:

	precision	recall	f1-score	support
0	0.91	0.86	0.89	903
1	0.11	0.17	0.13	88
accuracy			0.80	991
macro avg	0.51	0.52	0.51	991
weighted avg	0.84	0.80	0.82	991

Classification Report for class cow:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	966
1	0.04	0.04	0.04	25
accuracy			0.95	991
macro avg	0.51	0.51	0.51	991
weighted avg	0.95	0.95	0.95	991

Classification Report for class diningtable:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	971
1	0.00	0.00	0.00	20
accuracy			0.97	991
macro avg	0.49	0.50	0.49	991
weighted avg	0.96	0.97	0.97	991

Classification Report for class dog:

	precision	recall	f1-score	support
0	0.96	0.95	0.96	948
1	0.04	0.05	0.04	43
accuracy			0.91	991
macro avg	0.50	0.50	0.50	991
weighted avg	0.92	0.91	0.92	991

Classification Report for class bottle:

	precision	recall	f1-score	support
0	0.96	0.95	0.95	954
1	0.04	0.05	0.04	37
accuracy			0.91	991
macro avg	0.50	0.50	0.50	991
weighted avg	0.93	0.91	0.92	991

Classification Report for class bus:

	precision	recall	f1-score	support
0	0.98	0.99	0.98	968
1	0.00	0.00	0.00	23
accuracy			0.97	991
macro avg	0.49	0.50	0.49	991
weighted avg	0.95	0.97	0.96	991

Classification Report for class car:

	precision	recall	f1-score	support
0	0.91	0.88	0.89	899
1	0.08	0.11	0.09	92
accuracy			0.81	991
macro avg	0.49	0.49	0.49	991
weighted avg	0.83	0.81	0.82	991

Classification Report for class cat:

	precision	recall	f1-score	support
0	0.97	0.98	0.98	961
1	0.00	0.00	0.00	30
accuracy			0.95	991
macro avg	0.48	0.49	0.49	991
weighted avg	0.94	0.95	0.95	991