BHARGAV KILAPARTHI

1. This report's goal was to describe the procedures followed and the results obtained for a binary image classification job using a deep learning classifier and an SVM. The assignment involved classifying pictures of cats and dogs from the CIFAR-10 dataset. Convolutional Neural Networks (CNN) were used for deep learning, and Support Vector Machines (SVM) were used for more conventional machine learning. This paper examined the performance indicators for both models and provided a thorough analysis of the code implementation.

2.Dataset Preparation

The CIFAR-10 dataset included 60,000 color pictures with a 32x32 pixel resolution, distributed across 10 different classes. Only the cat (class 3) and dog (class 5) classes were considered for the binary classification task. Cat and dog samples were separated for both the training and testing sets once the dataset had been loaded.

Following data filtering, the labels were converted into binary values, where 0 denoted cats and 1 denoted dogs. The pixel values of the photos were normalized by dividing them by 255, ensuring that the pixel values varied between 0 and 1, in order to provide the input data for the models.

3.The following architectural design was used to create a CNN model:

- a 32-filter convolutional layer with ReLU activation, each measuring 3x3.
- a maximum pooling layer with a pool of 2 by 2.
- a convolutional layer using ReLU activation and 64 filters, each measuring 3 by 3.
- a maximum pooling layer with a pool of 2 by 2.
- a convolutional layer with 128 3x3-sized filters activated by ReLU.
- a maximum pooling layer with a pool of 2 by 2.
- a flatten layer that produces a 1D vector from the 3D output of the preceding layer.
- a 256-unit, fully linked, thick layer with ReLU activation that is named "feature_extraction."
- a single unit in a dense, fully linked output layer with sigmoid activation for binary classification.

4.The accuracy metric, binary cross-entropy loss, and Adam optimizer were all used in the model's construction. Using the pre-prepared training set, the model was trained for 20 iterations before being verified on the test data.

5.Feature Extraction using CNN

The samples in the training and testing sets were used to extract features from the fully connected layer 'feature_extraction'. With the same architecture as the CNN model up to the 'feature_extraction' layer, a new model called feature_extractor was created. The training and testing sets' features, X_tr_feat and X_te_feat, were predicted using this model.

6.Support Vector Machine (SVM) Model

An SVM with a linear kernel was trained using the characteristics that were retrieved from the CNN model. The corresponding labels y_tr and X_tr_feat were used to fit the SVM model. By predicting the labels for X_te_feat and calculating the performance measures, such as accuracy, F1 score, precision, and recall, the SVM model was tested after training.

```
SVM Accuracy: 0.76
SVM F1 Score: 0.76
SVM Precision: 0.76
SVM Recall: 0.76
```

7.Results and Discussion

The SVM model's performance was demonstrated by the results in terms of accuracy, F1 score, precision, and recall. These metrics provided information on the effectiveness of applying the feature extraction from the deep learning model to train a conventional machine learning model like the SVM. By contrasting the SVM model's performance