Homework 2 - Image Classification

Naomi Derel, 325324994 Shachar Frenkel, 326548153

18.12.2024

1 Classic Classifier

1.1 Loading Dataset

The first 5 images in the CIFAR-10 dataset are shown in Figure 1.

Figure 1: First 5 images in CIFAR-10 dataset

1.2 K-Nearest Neighbors

We load 10,000 samples from the training dataset, and concatenate the RGB channels of a flat image into a single vector. We then train a K-Nearest Neighbors classifier from `sklearn.neighbors` with $k = 10$.

1.3 Evaluation on Test Set

We load 1,000 samples from the test dataset, and evaluate the classifier on the test set. The accuracy of the classifier is 0.288. We also compute a confusion matrix between the true labels and the predicted labels, shown in Figure 2. We note that the results are not very good, with low accuracy and many misclassifications. The best performance was achieved on the 'ship' class, which might have to do with the distinguished background color or object itself.

1.4 Analysis of K Value

The comparison of the model accuracy as a function of the number of neighbors k, between 1 and 30, is shown in Figure 3. We note that the accuracy is pretty low for all values of k and remains below 0.3.

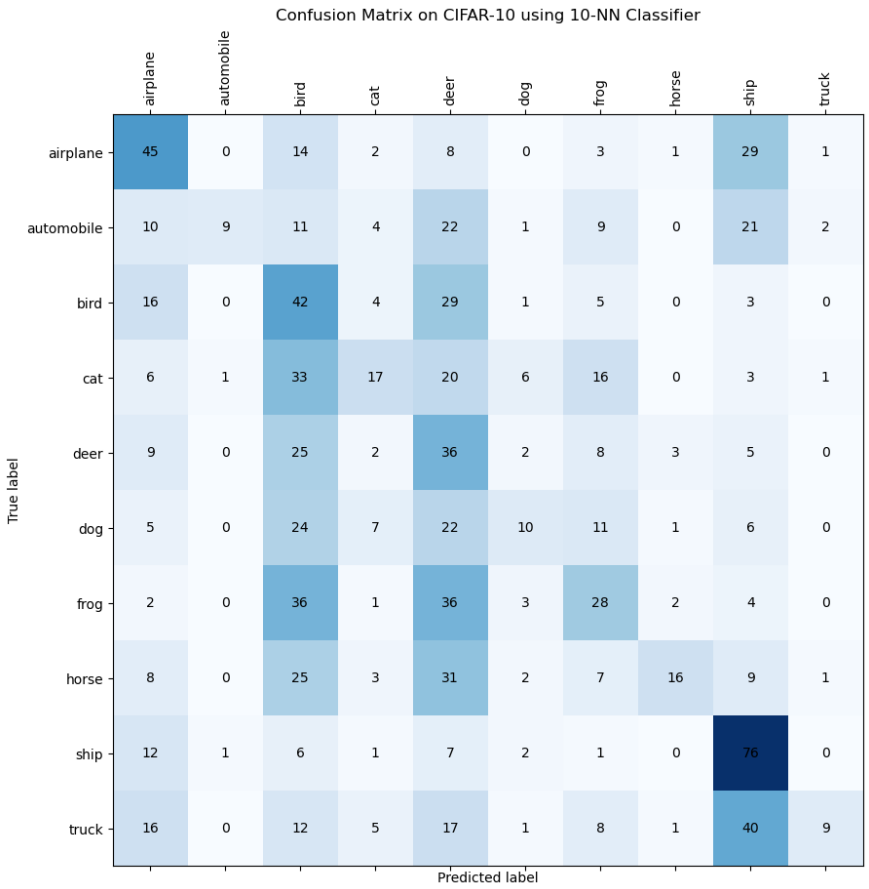

Figure 2: Confusion matrix of K-Nearest Neighbors classifier

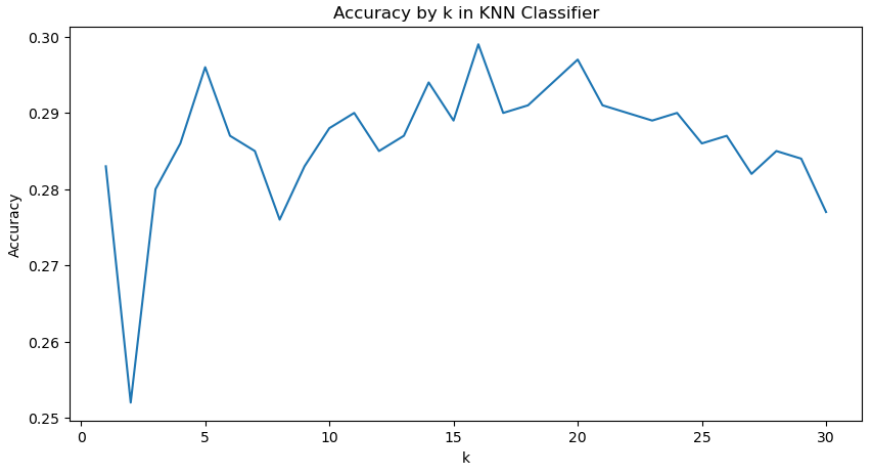

Figure 3: Model accuracy as a function of the number of neighbors k

2 Convolution Neural Network

2.1 Baseline CNN

We train the given CNN model with the hyper-parameters from the tutorial: learning rate of 10^{-4}, batch size of 128, 20 epochs and the Cross-Entropy loss function.

The baseline accuracy achieved on the test is 78.05%. The confusion matrix in Figure 4 shows that the model performs well on all classes. The most confusion occurs between the classes 'cat' and 'dog', with the model often predicting 'cat' when the true label is 'dog'.

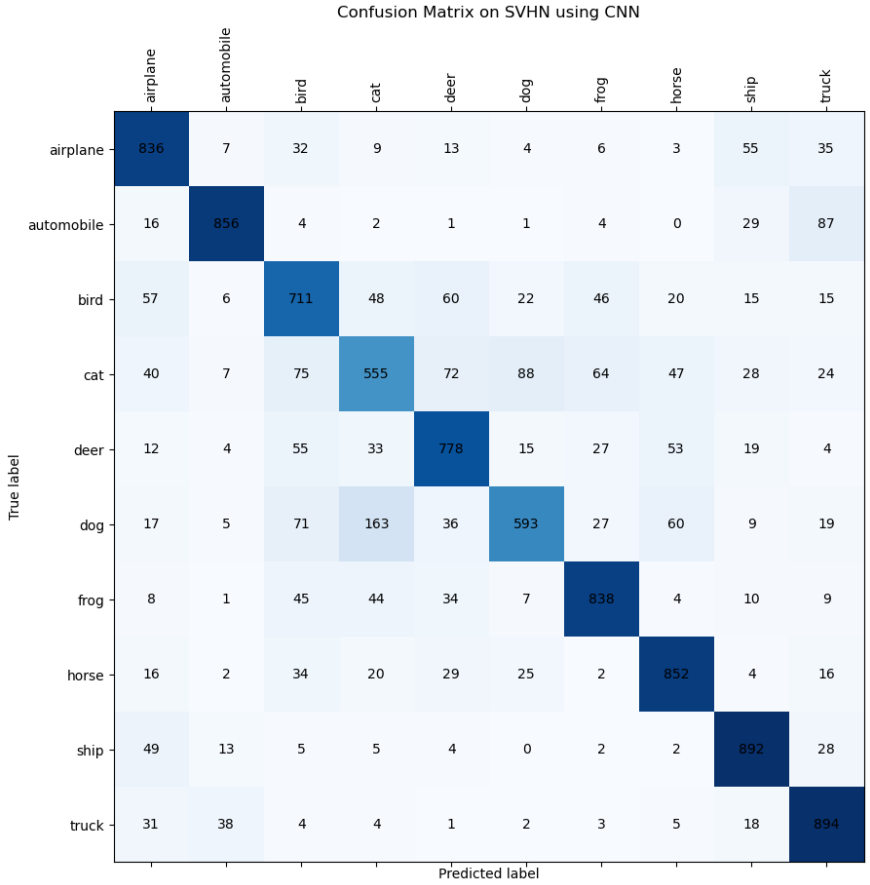

Figure 4: Confusion matrix of the baseline CNN

2.2 Designing Our Own CNN

We notice that the baseline model has a training accuracy of 96.576%, much higher than the test accuracy, which indicates the model is overfitting. To address this, we expand the given architecture with more dropout and batch normalization layers.

We also conduct training using the cross-validation method, allowing us early stopping instead of overfitting.

2.3 Training and Evaluation

This training results in a test accuracy of 79.29%, which is slightly better than the baseline model.

3 Foundation Models

3.1 Embedding Space of T-SNE

We load all the images from the 'data/clip_images' directory, which includes 'cats' and 'dogs' images. We then embed the images using clip, and project them using T-SNE (with random seed=18). The resulting plot is shown in Figure 5. We note that the 'cats' and 'dogs' images are well separated in the embedding space, with the 'cats' images on the right and the 'dogs' images on the left.

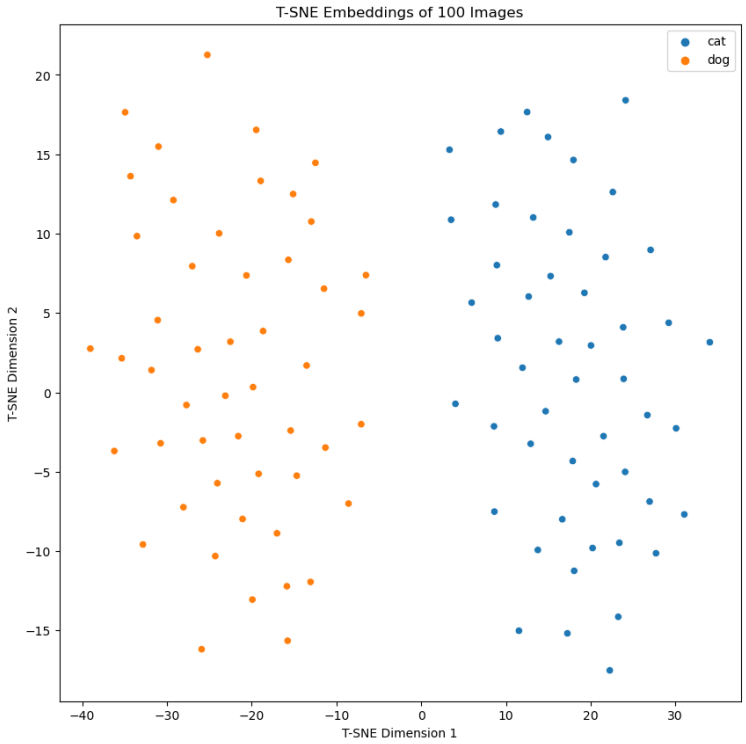

Figure 5: T-SNE plot of the 'cats' and 'dogs' images

3.2 CLIP Nearest Neighbors

We compare the image of Alfie (Figure 6) with the 5-Nearest Neighbors in the CLIP embedding space (Figure 7).

Figure 6: Alfie

Figure 7: 5-Nearest Neighbors of Alfie in the CLIP embedding space

First, the results are all cat images and no dog images, which matches our expected result. Second, we notice that most of the cats have a white coloring around the same location as Alfie, and some have a general gray-

black coloring as well. This is a good indication that the model is able to understand the context of the image and not just the image itself.

3.3 Classification with CLIP Textual Embeddings

We load the images and labels from the 'data/clip_images/test' directory.

3.4 Counting Objects in Images