CIMON: Towards high quality Hash Codes

Group Members:

1. **Hitarth Kothari** **2019A7PS0178G**
2. **Sushant Sreeram Swamy** **2019A7PS1031G**

Table of Contents:

About the Paper

The paper that we have chosen is called CIMON which stands for **C**omprehensive s**I**milarity **M**ining and c**O**nsistency lear**N**ing in which the authors propose their model for Deep Hashing related tasks. The results in the paper show that the proposed model is able to learn high quality hash codes.

More About Deep Hashing and Retrieval based tasks

With growing demands for browsing, searching and retrieving images from a large image database, content-based image retrieval has become a hot topic. During deep hashing, images are mapped to compact binary hash codes according to extracted image features. Since nearby hash codes indicate similar contents, hashing methods retrieve images with similar hash codes to query images. The binarized representation of images leads to great efficiency improvement on storage and computation compared to standard techniques.

Proposed Method in short

The aim of the paper was to learn good hash codes so as to get good results in retrieval performance. The model that has been proposed was to learn robust hash codes that are tolerant to small augmentations of images, i.e, images that are similar to each other should have a similar hash code. The other problem that the authors have tried to solve was that from the space of a pre-trained model, many false similar pairs can occur between the boundary points of two manifolds. This can lead to the introduction of false positives and negatives.

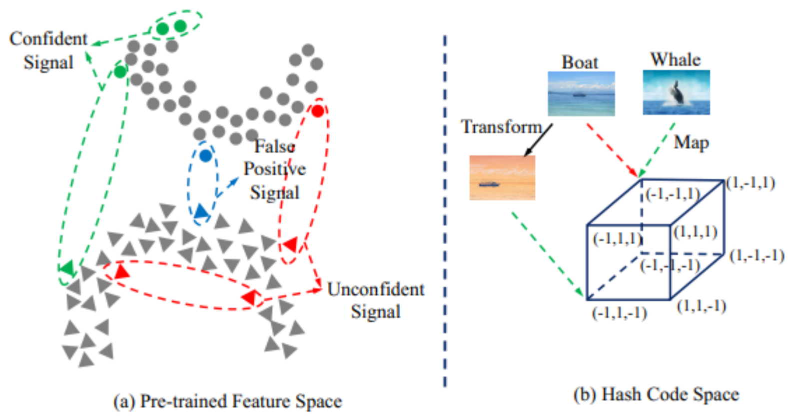


Figure 1: Motivation of our model. (a) The "triangle" points and the "circle" points are belong to different categories. False signals (blue) and unconfidence signals (red) will misguide the hash code learning. (b) Two different images (the first line) are mapped to the same hash code (collision) and the hash code is sensitive to the transformation, which implies the hash code is of low quality.

Figure from the paper which illustrates the problems tackled by the authors.

The proposed model is as follows:

- Pretrained VGG-F without the last fully connected layer is adopted to extract deep features, which is used to generate the similarity graph and the confidence-based weight matrix.
- Consistency learning: The hashing network is modified from VGG-F by replacing the last fully-connected layer with a fully-connected layer with L hidden units to incorporate the hash code learning process.
- A novel consistency learning framework is adopted to learn high-quality hash codes.

Aim of the project

Deep hashing networks are able to extract features from the images into a compact representation. We were curious to know if these hash codes can be used for **classification** tasks, instead of traditionally using these networks for retrieval based tasks. A thorough literature review was conducted, and we were not able to get any work that has used hashing networks for classification purposes. Through this project, we aim to study if we can use CIMON's rich hash codes for classification purposes.

Salient Features and Contributions

Salient Features:

- The dataset that we have chosen is the STL10 dataset. We chose this dataset because of the rich availability of unlabeled data that can be used for learning good hash codes.
- Since the unlabeled dataset is very huge (100000 images), in order to reduce the training time, we have used only a random 20000 images from the dataset.
- To be as close to the model proposed by the authors, we have used a smaller VGG network, the VGG-11 model with Batch Normalization layers for the feature extraction network and the hash code network
- The training dataset that the authors have chosen is a random sample of the retrieval set. They constructed the similarity graph, confidence matrix using the entire dataset. These quantities are constructed only once before training, and are used repeatedly during their main training loop. Our approach to this part has been a bit different. We are constructing these quantities for each mini-batch separately during the training procedure. These are the following justifications for the above approach:
    - One of the main problems was resource constraints on Google Colab. Since we have a limited amount of memory, we could not afford to use the whole dataset.
    - For a retrieval task, the whole dataset would matter since they would need to capture the correlation between images in the dataset. But we are using the hash codes for classification purposes. So, even if we have the images as batches, we would be able to learn hash codes for the images in general.
- Most of the hyperparameters of the model that we used are taken directly from the paper. We changed a few of them to suit our purposes and added a few:
    - BATCH_SIZE : 256
    - N_CLUSTERS = 5. This is the number of clusters that is given as input to the spectral clustering algorithm. The authors have used a

    value of 70, but their graph is of size 5000x5000 while ours is 256x256. So, we reduced the number of clusters
  - We also increased the threshold parameter to 0.3 compared to 0.1 which was used by the authors.

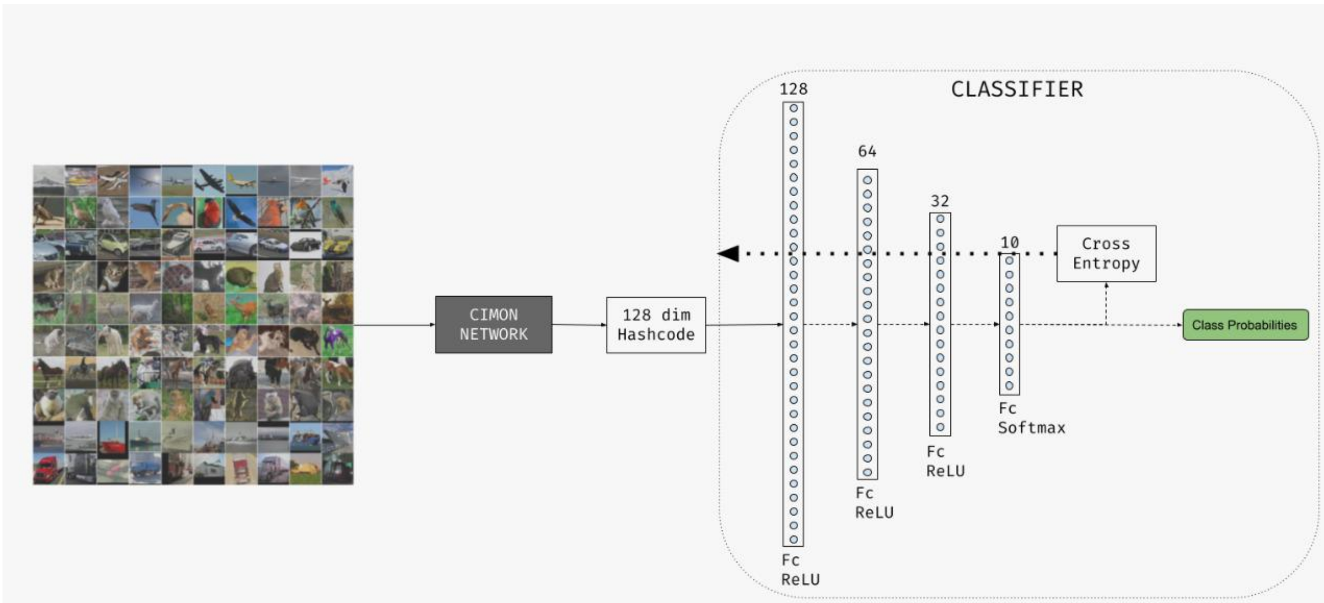
Contributions:

- We present the first network that uses learned deep hash codes for classification purposes.
- We first train the CIMON network in a (modified manner as mentioned above) to learn a network that can learn good hash codes for images.
- We train a small MLP (as shown in Figure 1) using the training set of the STL10 dataset. This is done as follows:
  - The image is passed through the trained network to obtain the hash code for the image.
  - Using the hash code as the input to the MLP, we train the MLP to classify the image given the hash code.
  - Just like any other network, using the cross entropy loss, we train the MLP

- We have also added visualizations of the hash codes that the network has learnt. The visualizations support our hypothesis that classification can be performed from the hash code space because for the classes of images taken, the network was able to produce hash codes that form visibly separate clusters in low dimensional space. (Figure 2)
- There are only 5000 training examples in the STL10 dataset, and we were able to achieve a good accuracy by first training an unsupervised learning model to learn hash codes, and then training a supervised learning model on just 5000 images. These are results that we obtained on the STL10 dataset:
  - Best test accuracy : **86.56 %**
  - Best train accuracy : **84.9 %**

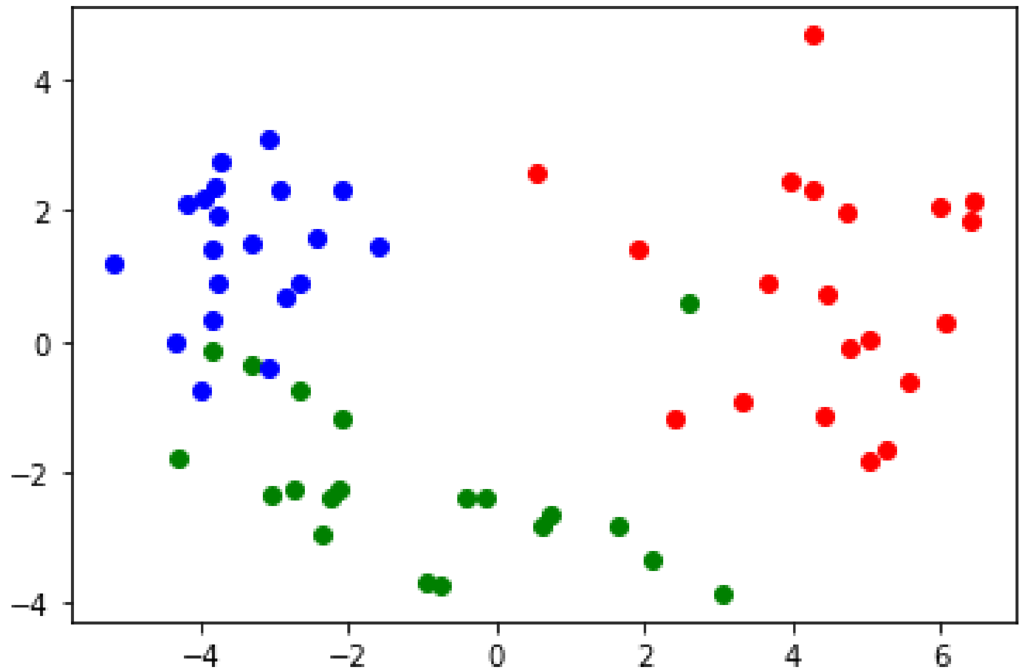
Plots and Visualizations

Figure 1:



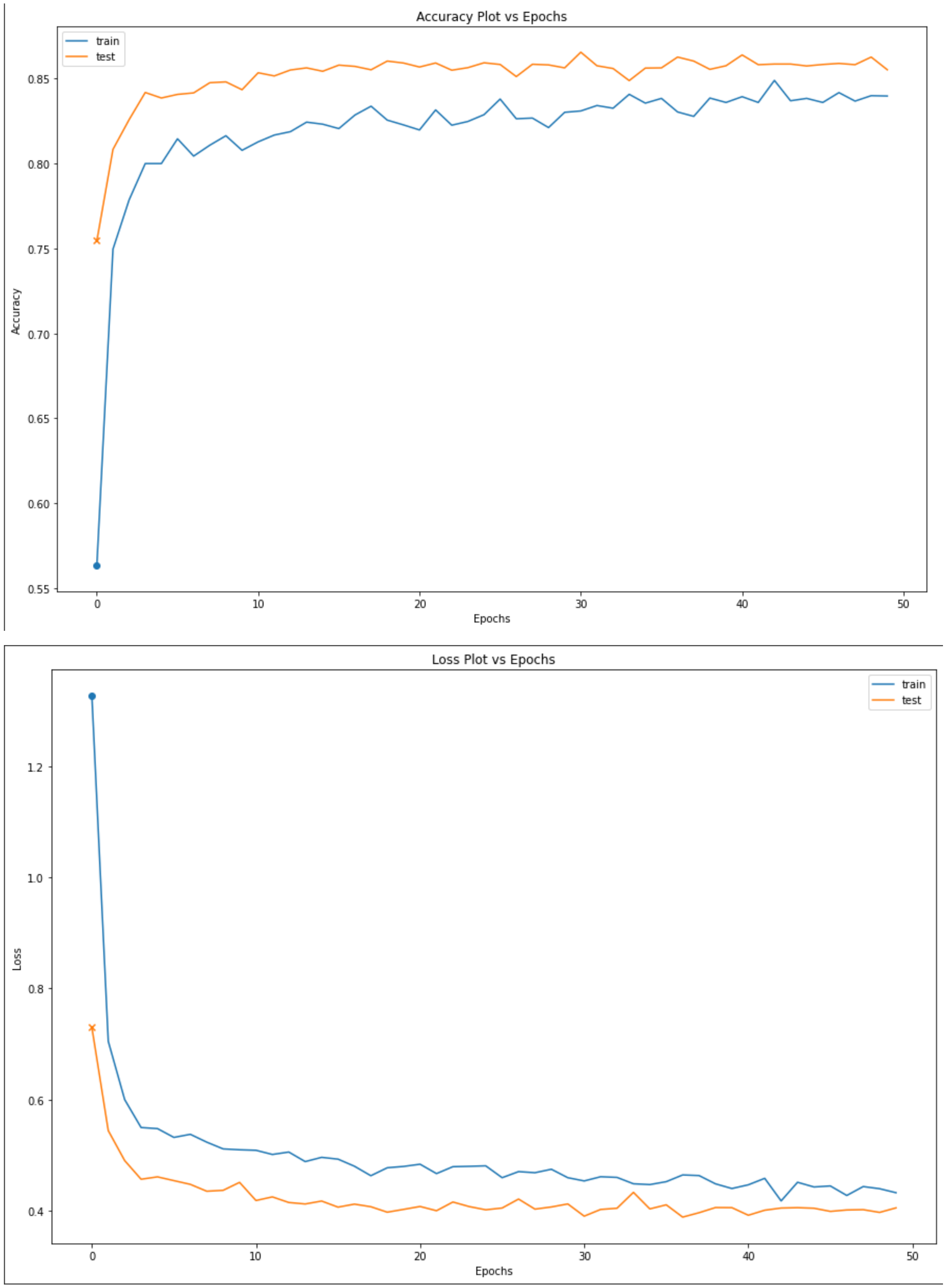
The model architecture is depicted in a pictorial format.

Figure 2:



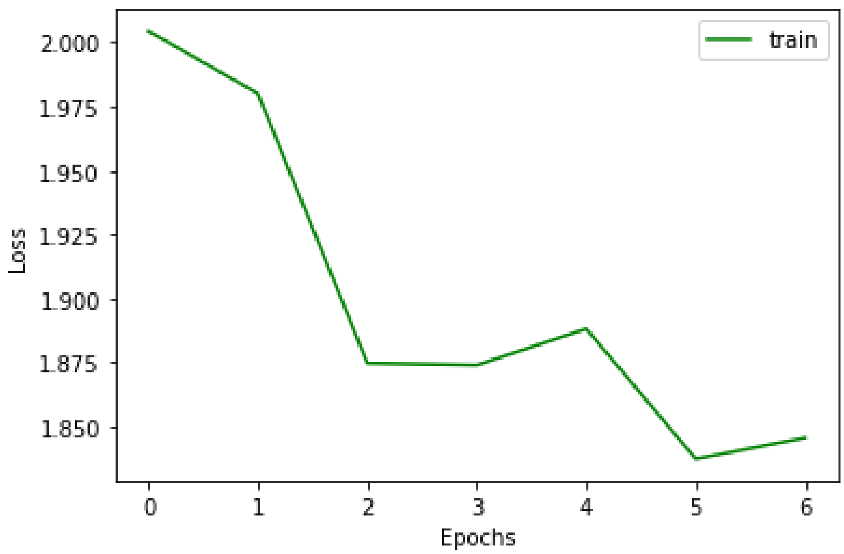
20 samples of 3 different classes were randomly selected, and 128 bit hash codes were generated. Using PCA, dimension was reduced to 2 (for visualization purposes). We can clearly observe three clusters in the above diagram. This gives us strong reasons as to why we were able to achieve good results on the test dataset.

Figure 3:



Training curves for the classification model.

Figure 4:



Training curve for the hash code model. We were able to train for only 7 epochs since the model takes a lot of time to train.

Conclusions

We presented a network that uses a deep hashing network (CIMON) combined with an MLP for classification based tasks. The network was able to learn decent hash codes which further helped the classification network to do well.

Due to time constraints, we were unable to train the hashing network for more epochs. We strongly believe that further training of the hashing network would improve the results.