# 2AMM10 2021-2022 Assignment 1: Group 22

Ruichen Hu(1674544) Simin Sun(1692933) Siyue Chen(1657402)

May 20, 2022

## 1 Problem Formulation

Image retrieval is a problem in the field of deep learning, which aims at finding one or multiple images belonging to the same class of the query image from a dataset. The approaches resolving this problem usually involve convolutional neural networks (CNN), as they are invariant and equivariant to spatial image transformations due to the convolutional architecture.

This report focuses on building a model to recognize the images from the Omniglot dataset that represent the same character as the query image within a tuple. To formulate the problem in a formal way, given a training set containing 10000 tuples of images $T_i (1 \leq i \leq 10000)$, the tuple size $|T_i| = 6$ and indices of the images $j \in [0, 5]$, we need to determine for each tuple whether the images $x_j (0 \leq j \leq 4)$ in the support set belong to the class of the query image $x_5$. The trained model is then evaluated on a test set, in which there are 1000 tuples with images not shown in training set.

The main obstacle is there is no class label given for query images. That means for each character represented by the query image, we have to train the model merely relying on the images within the same tuple, which leads to fairly small amount of training data. This problem corresponds to image recognition task with little training data, which is similar to one-shot learning task described in [1], thus we decide to apply the metric learning method. Compared to traditional supervised learning, metric learning requires much less amount of data, because it learns from all input data in a metric space instead of focusing on data of a certain class. Specifically, it uses the similarity or distance between the objects to judge whether they are in the same category or not. The model based on this method usually updates the parameters by calculating distance of the objects. This indicates that the features of all the objects are taken into consideration. Besides, as images are typically embedded in a vector space in metric learning rather than transferred into the predicted probabilities of fixed classes, the models generalizes well when recognizing the unseen images in the test set. In other words, the similarity of the two test images is directly measured by the similarity of the two image embeddings. The images with close similarity can be regarded as belonging to the same class. By this means, we can convert all similarity values into predicted labels, which are then evaluated by the true labels.

## 2 Model Formulation

As mentioned in the preceding part, metric learning is a suitable solution for the given problem. Specifically, we decide to reproduce the siamese neural network proposed by [1], but with some modifications on model architecture and loss function. Instead of building twin networks, we construct a single network with shared parameters for the both input images. Every time when a pair of input images is forwarded, the two images are processed separately, and we get an embedding at the output layer for each forward pass. The two embeddings are further utilized in loss function. As for the remaining parts of the model, they are similar to common components of a CNN that contain multiple convolutional blocks including convolutional layer, pooling, padding and regularization layer as well as multiple

fully connected blocks(Figure 1). To accommodate to the siamese network, image tuples are decomposed by combining each image of the support set with the query image in the same tuple to form a pair $(x_j, x_5), j \in [0, 4]$, and assigning a corresponding true label.

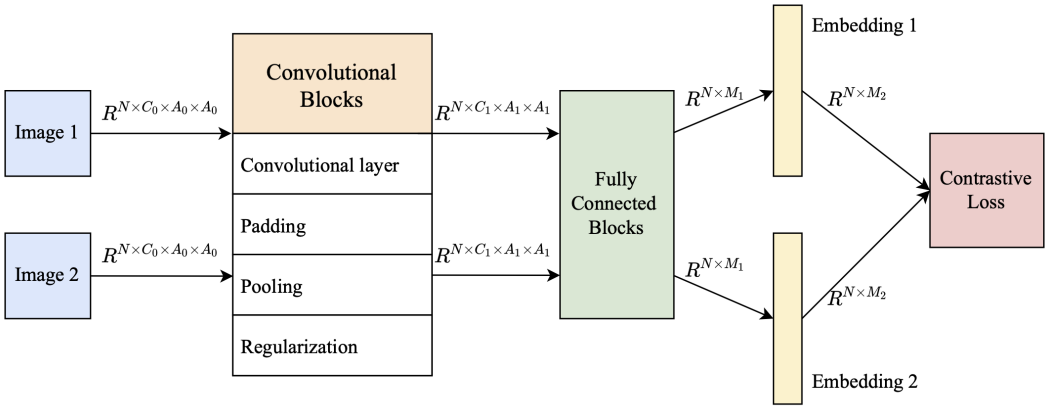


Figure 1: Architecture of the siamese neural model($N$ denotes the batch size. $C_i$ denotes the number of channels. $A_i$ denotes the dimension of images or layer outputs. $M_i$ denotes the dimension after dense layers. )

The reason for the above modifications is that this architecture is simpler and works under the same mechanism proposed in [1]. Besides, as we do not have the category label of the characters, it makes no sense to use one-hot encoding coupled with cross entropy loss at output layer. Instead, we use the contrastive loss, which requires the two output objects to be vectors such that the distance between the two objects can be derived. The formula of contrastive loss is as follows [2], assuming the batch size to be $n$:

$$L(x_1, x_2) = (1 - Y)\frac{1}{2}\|f(x_1) - f(x_2)\|^2 + Y\frac{1}{2}(max(0, \tau - \|f(x_1) - f(x_2)\|))^2$$

where $Y \in \{0, 1\}^N$ is a vector of true labels indicating whether each one of $N$ image pairs is similar($Y = 0$) or not ($Y = 1$), $f(x_1) \in R^{N\times M}$ and $f(x_2) \in R^{N\times M}$ denote the embedded representations in the dimension $M$ of $N$ query images and their counterparts, and $\tau$ means a margin, which acts as a hyperparameter in our model.

The model is updated by reducing the contrastive loss. If two images are similar but with different embeddings, then the model is punished such that it is forced to focusing more on learning the similar patterns of the two images, which contributes to output the similar embeddings and less losses. If the gap between the two different images is insufficiently large, namely lower than the margin, then the parameters are updated in a way that the different characteristics of the images are more significantly presented. After training, the model is invariant to translation such that different images of the same character can be recognized as the same.

# 3 Implementation

## 3.1 Dataset Construction

The original dataset doesn't include classes of the characters (and if so, there would be too many classes). So we don't use a traditional classification model. Since the task only focuses on distinguishing two images, we use a Siamese Network which takes three inputs: a query image, a supporting image, and a label stating whether the two images are the same. Thus we take the query image from each row combined with each supporting image from the same row as a pair. With a batch size 50, we transform the original data to the following form in Table 1:

| | original | reconstructed |
|---|---|---|
| training data | 10000∗5 | 1000 batches∗size(50) |
| test data | 1000∗5 | 100 batches∗size(50) |

Table 1: Dataset reconstruction

## 3.2 Network

The network structure is shown as follows:

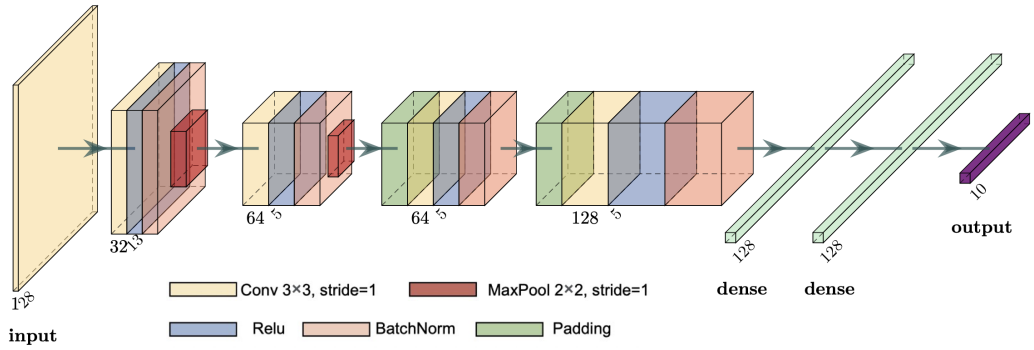


Figure 2: Structure of the siamese neural network

Our neural network contains four convolutional layers. The dimension of the image after each convolutional layer is shown in Figure2 . We use ReLU as the activation function to resist gradient vanishing and to make the network converge faster. We also apply batch normalization layers and max pool layers to reduce the sensitivity to initial values and mitigates overfitting. Compared to using fixed values for padding, the reflection padding approach we use may yield better convolution results. Followed by fully-connected layers, we map the learned distributed feature representation to the labeling space. Here we use three transformation with ReLU and output result.

We adopt two network with the same structure above as a Siamese twin. They join immediately after the fully-connected layer where the L1 component-wise distance between vectors is computed. Thus our model takes two images as inputs at a time, pass each of them through the twin neural network.

## 3.3 Contrastive Loss

The purpose of the twin architecture is not to classify the input images, but to distinguish them. Therefore, a classification loss function (e.g., cross entropy) is not the most suitable choice. Instead, we use the contrastive loss that calculate the Euclidean distance of two sample features. When the samples are similar, if the Euclidean distance in the feature space is relatively large, we need to increase the loss, and vice versa. We set a threshold margin, and when the distance exceeds the margin, the loss is treated as 0 (i.e., the loss should be very low for dissimilar features that are far away; and for similar features that are far away, we need to increase the loss, so as to continuously update the matching degree of the sample).

## 3.4 Training and Testing

We use a batch training in each epoch. In each batch, we take 50 image pairs with their labels from the dataloader. After 15 epoches, the model converges and the loss won't decrease.

We implement two ways of evaluation. For the first approach, we take L1 normalization, i.e. the mean value of distances between the five supporting images and the query image from the same row as the threshold. It is simple but comes with a limit when all supporting images are all the the same/different class as the query image.

Thus in the second approach, we adopt the Receiver Operating Characteristic (ROC) curve as the tool of evaluation. By finding the balance between the False Positive Rate (FPR) and True Positive Rate (TPR), we can reach the highest accuracy and the result is better than the previous approach. However, this also comes with a drawback that the true labels have to be taken into account. That means we can't apply it to a new query that we don't have the result.

# 4 Experiments and Results

**Training:** In the experiments, we reconstruct the data to format the inputs images. In the training process, we constantly adjust the number of epochs, the learning rate, and number of layers, etc. Finally, as shown in Table 2, we have an optimal set of parameters that makes our model works fine. In each batch, we take 50 image pairs with their labels from the dataloader. The learning rate is 0.001, the regularization rate is 0.05, and the total epochs is 15 since the loss converges and changes slightly even if we increase the number of epochs.

We collect the loss over the epoch during the training process to analyze the Siamese Model's performance on the training task. The loss decreased during the training process, as seen in Figure 3. On the last epoch, the initial loss of 0.011 was reduced to 0.006.

**Testing:** We implement two ways of evaluations which described in Section 3.4. The evaluation method using L1 normalization output test accuracy of 0.8844 while using ROC curve gives a better accuracy of 0.9044.

| epoch | loss | epoch | loss | epoch | loss |
|---|---|---|---|---|---|
| 0 | 0.011348750566840171 | 5 | 0.006682599625587464 | 10 | 0.006152719916105271 |
| 1 | 0.007859797753691673 | 6 | 0.006584761443734169 | 11 | 0.006083149826228619 |
| 2 | 0.007182168480455876 | 7 | 0.0064162091067433355 | 12 | 0.005925096845030784 |
| 3 | 0.006922982262670994 | 8 | 0.006329963045716286 | 13 | 0.005950722154974938 |
| 4 | 0.006781164486706257 | 9 | 0.006259964982271195 | 14 | 0.005920068034678698 |

Table 2: Loss Summary

# 5 Results and Analysis

According to the Figure 3, it's worth noting that the loss decreased significantly during the first two epochs, then gradually dropped after that. The total number of epochs is 15, as expanding the amount of epochs may cause the model overfitting.

We initially use L1 normalization, which uses the average distance between the five supporting images and the query image from the same row as the threshold for evaluation. It's straightforward, but there's a restriction if all supporting photos are of the same or different class as the query image.

Then we use ROC as an assessment approach, which results in higher accuracy. To find the threshold, we try to balance between the False Positive Rate (FPR) and True Positive Rate (TPR) as shown in Figure 4. However it requires the usage of real labels when computing accuracy. That means we won't be able to utilize it on a new query without a result. We also attempted using L2 normalization and softmax to improve the vector of same-class photos while lowering the weights of different-class shots, but the results were not good as expected.

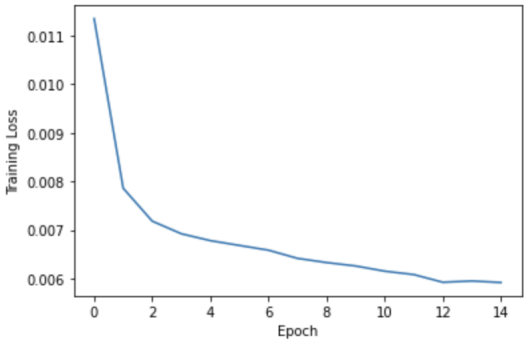


Figure 3: Loss over Epoch curve

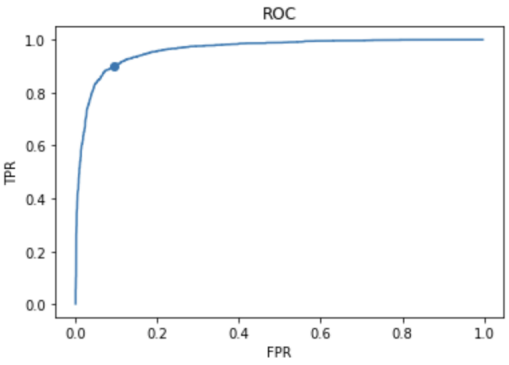


Figure 4: ROC curve

# 6 Conclusion

According to experiments and results we get from this assignment, it is possible to conclude that the Siamese model can be used to detect the same classification images provided query images without categorizing information or a large number of examples. In our example, the L1 normalization and ROC curve are useful tools for assessing the Siamese model.

However, our model's evaluation phase, particularly how we determine the threshold, could be enhanced. Both L1 normalization and the ROC curve have limitations, so we're seeking for a better function to apply to the dist we obtained in order to maximize the vector of same-class images and decrease the weights of different-class images. Softmax and L2 normalization were attempted, but the accuracy was inferior.

In the future, we could adopt a Triplet Network [3] with Triplet Loss. It allows features with the same label to be as close as possible in spatial location, while features with different labels are as far away as possible. But based on our dataset, we may end up with too many triplets. Thus for dataset construction, we could use informative and representative triplet selection [4] to select a small set of the most representative and informative triplets. It allows faster convergence and reduce the computational complexity of training without significant loss on the performance.

# References

[1] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

[2] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006.

[3] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

[4] Gencer Sumbul, Mahdyar Ravanbakhsh, and Begüm Demir. Informative and representative triplet selection for multi-label remote sensing image retrieval. *arXiv preprint arXiv:2105.03647*, 2021.