HIROSHIMA UNIVERSITY

MASTER THESIS

Supervised Learning for Convolutional Neural Network with Barlow Twins

Author:
Murugan RAMYAA (M203877)

Supervisor:
Professor Takio KURITA
Sub Supervisor:
Associate Professor Junichi MIYAO
Professor Hiroaki MUKAIDANI
Associate Professor Sayaka KAMEI

A thesis submitted in fulfillment of the requirements for the degree of Master of Informatics and Data Science

in the

Division of Advanced Science and Engineering
Informatics and Data Science Program

August 9, 2022

Declaration of Authorship

I, Murugan RAMYAA (M203877), declare that this thesis titled, "Supervised Learning for Convolutional Neural Network with Barlow Twins" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Murugan Ramyaa

Date: 2022

HIROSHIMA UNIVERSITY

Abstract

Graduate School of Advanced Science and Engineering
Division of Advanced Science and Engineering
Informatics and Data Science Program

Master of Informatics and Data Science

Supervised Learning for Convolutional Neural Network with Barlow Twins

by Murugan RAMYAA (M203877)

Supervised learning has found many applications as one of the fundamental techniques in machine learning. Recently, Barlow Twins loss has been used in self-supervised learning to train Convolutional Neural Networks to extract invariant features by introducing perturbations to the input data and encouraging the network to learn features that are invariant to these perturbations. It has been shown that the extracted features can achieve good performance for new tasks. However, this kind of 'perturbation-invariance' is also necessary for the models trained with supervised learning. The main contribution of this research is to explicitly introduce the perturbation-invariance in supervised learning. To do this, we propose to train a pair of identical networks using the standard classification loss and an additional Barlow Twins loss. The different sets of transformations are applied to the two network branches as the perturbations. We show empirically that the proposed method can learn invariant features and results in a higher classification accuracy than the baseline method.

Acknowledgements

I would like to express my sincere gratitude to Professor Takio Kurita, Associate Professor Junichi Miyao, Professor Hiroaki Mukaidani, and Associate Professor Sayaka Kamei. They provided the best environment for my research, supported my student life, and helped my research with many ideas. Especially, Professor Kurita always consulted me about my research and helped me with his accurate advice and sharp points. He guided me with a lot of ideas which always made me think new and more creative in the research. Also, Associate Professor Miyao and Assistant Professor Aizawa helped me with their opinions, and reference papers in our regular laboratory seminars. I am grateful to Dr. Jonathan Mojoo who guided me in my research with his accurate points and advice. He motivated me to study more and provided detailed information and knowledge in the research, which made me feel that I should do my best. My research was completed, thanks to their knowledge and advice. Finally, I would want to express my sincere thanks to my lab members who are always supportive and my family who helped me to accomplish my research in such a timely manner.

Contents

List of Figures

List of Tables

List of Abbreviations

CNN	**C**onvolutional **N**eural **N**etworks
SIMCLR	**Sim**ple framework for **C**ontrastive **L**earning of visual **R**epresentations
ReLU	**Re**ctified **L**inear **U**nit
SGD	**s**tochastic **g**radient **d**escent
PCA	**P**rincipal **C**omponent **A**nalysis
BYOL	**B**ootstrap **Y**our **O**wn **L**atent
SIMSIAM	**SIM**ple **SIAM**ese
t-SNE	**t**-distributed **S**tochastic **N**eighborhood **E**mbedding
GPU	**G**raphics **P**rocessing **U**nit
SwAV	**Sw**apping **A**ssignments between multiple **V**iews
SeLa	**Se**lf **La**belling

Chapter 1

Introduction

Supervised learning is a subcategory of machine learning algorithms that construct the model trained using labeled datasets. The trained model can accurately classify new samples or predict outcomes for the new samples. Usually, the accuracy of the model is measured through a loss function, and the best parameters are searched during the training until the loss has been sufficiently minimized.

Image classification is a computer vision task where the trained model must accurately assign test images into specific categories. Convolutional neural networks (CNNs) have proven to be an excellent performance for image classification. The cross-entropy loss is the most widely used loss function for supervised learning of deep classification models. A number of works have explored shortcomings of this loss, such as lack of robustness to noisy labels and the possibility of poor margins, leading to reduced generalization performance. However, in practice, most proposed alternatives have not worked better for large-scale datasets, such as ImageNet, as evidenced by the continued use of cross-entropy to achieve state-of-the-art results.

In recent years, a resurgence of work in contrastive learning has led to major advances in self-supervised representation learning. The common idea in these works is the following: pull together an anchor and a "positive" sample in embedding space, and push apart the anchor from many "negative" samples. Since no labels are available, a positive pair often consists of data augmentations of the sample, and negative pairs are formed by the anchor and randomly chosen samples from the minibatch.

Recently research on self-supervised learning [10] becomes popular in machine learning algorithms and has achieved almost similar accuracy to supervised learning. In self-supervised learning, the samples without labels are used in the training and the perturbation-invariant features are extracted by contrastive mechanism [14] which reduces the distance between the representations of differently augmented views of the same image (positive pairs) and increase the distance between the representations of the augmented views from different images (negative pairs).

One of the self-supervised learning methods, named Barlow Twins [12] uses an objective function that computes the cross-correlation matrix between the outputs of two identical networks fed with distorted versions of the same sample, and makes

it as close to the identity matrix as possible. This causes the embedding vectors of the distorted versions of the same sample to be similar while minimizing the redundancy between the components of these vectors. Barlow Twins does not require large batches nor asymmetry between the network twins such as a predictor network, gradient stopping, or a moving average on the weight updates.

In this work, we propose a supervised learning method for image classification with CNN by combining the loss function of the supervised learning (cross-entropy loss) with the loss of self-supervised learning (Barlow Twins loss). Perturbations (augmentations) are applied to the input data. The perturbed images are first fed into the two networks with the same parameters and then the Barlow Twins loss is measured at a small neural network projection heads that give the embedding features. The embedding features from the same class are pulled closer together than embedding features from different classes and are represented as 1's and 0's in cross-correlation matrix. The main goal is to encourage the network to learn the perturbation invariant embeddings. We experimentally confirmed that the cross entropy (CE) loss with Barlow Twins loss can achieve the best classification accuracy in the Supervised classification problems. Our main contributions are summarized below:

1. We propose a perturbation-invariant feature extraction mechanism in the Supervised Learning for Classification with CNN using the Barlow Twins loss function.

2. We show that the proposed approach provides consistent improvements in classification accuracy for different baseline CNNs, VGG-16 and ResNet-18, on different datasets, CIFAR-10 and STL-10.

Chapter 2

Related Works

Self-supervised learning aims to learn useful representations of the input data without relying on human annotations. Recent advances in self-supervised learning for visual data show that it is possible to learn self-supervised representations that are competitive with supervised representations. A common underlying theme that unites these methods is that they all aim to learn representations that are invariant under different distortions (also referred to as 'data augmentations'). This is typically achieved by maximizing the similarity of representations obtained from different distorted versions of a sample using a variant of Siamese networks.

Contrastive methods like SIMCLR [2] define 'positive' and 'negative' sample pairs which are treated differently in the loss function. Additionally, they can also use asymmetric learning updates wherein momentum encoders are updated separately from the main network. Clustering methods use one distorted sample to compute 'targets' for the loss, and another distorted version of the sample to predict these targets, followed by an alternate optimization scheme like k-means in DEEPCLUSTER or non-differentiable operators in SWAV and SELA.

In another recent line of works, BYOL [6] and SIMSIAM [13], both the network architecture and parameter updates are modified to introduce asymmetry. The network architecture is modified to be asymmetric using a special 'predictor' network and the parameter updates are asymmetric such that the model parameters are only updated using one distorted version of the input, while the representations from another distorted version are used as a fixed target. SIMSIAM concludes that the asymmetry of the learning update, 'stop-gradient', is critical to preventing trivial solutions.

Among discriminative methods, contrastive methods [9],[8] currently achieve the state-of-the-art performance in self-supervised learning [4],[3]. Contrastive approaches avoid a costly generation step in pixel space by bringing the representation of different views of the same image closer ('positive pairs') and spreading representations of views from different images ('negative pairs') apart [11]. Contrastive methods often require comparing each example with many other examples to work well [2],[7] prompting the question of whether using negative pairs is necessary.

DeepCluster [1] partially answers this question. It uses bootstrapping on previous versions of its representation to produce targets for the next representation; it

clusters data points using the prior representation and uses the clustered index of each sample as a classification target for the new representation. While avoiding the use of negative pairs, requires a costly clustering phase and specific precautions to avoid collapsing to trivial solutions.

In the new method, Barlow Twins[12], which applies redundancy reduction, a principle first proposed in neuroscience to self-supervised learning. In his influential article Possible Principles Underlying the Transformation of Sensory Messages (Barlow, 1961), neuroscientist H. Barlow hypothesized that the goal of sensory processing is to recode highly redundant sensory inputs into a factorial code (a code with statistically independent components). This principle has been fruitful in explaining the organization of the visual system, from the retina to cortical areas, and has led to a number of algorithms for supervised and unsupervised learning. Based on this principle, they propose the objective function which tries to make the cross-correlation matrix computed from twin embeddings as close to the identity matrix as possible. Barlow Twins is conceptually simple, easy to implement and, learns useful representations as opposed to trivial solutions. Intriguingly, Barlow Twins strongly benefits from the use of very high-dimensional embeddings. The network architecture of Barlow Twins is shown in Fig.2.1. Barlow Twins outperforms previous methods on ImageNet for semi-supervised classification in the low-data regime and is on par with the current state of the art for ImageNet classification with a linear classifier head, as well as for a number of transfer tasks of classification and object detection.

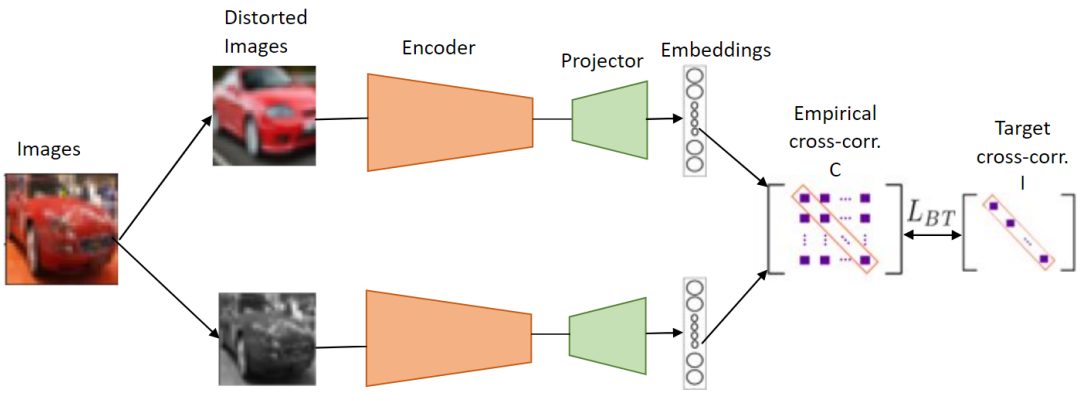


FIGURE 2.1: Barlow Twins's objective function computes the cross-correlation matrix between the embeddings of two identical networks fed with distorted versions of a batch of samples, and tries to make this matrix close to the identity matrix. Barlow Twins is competitive with state-of-the-art methods for self-supervised learning while being conceptually simpler, naturally avoiding trivial constant (i.e. collapsed) embeddings, and being robust to the training batch size.

Chapter 3

Proposed Method

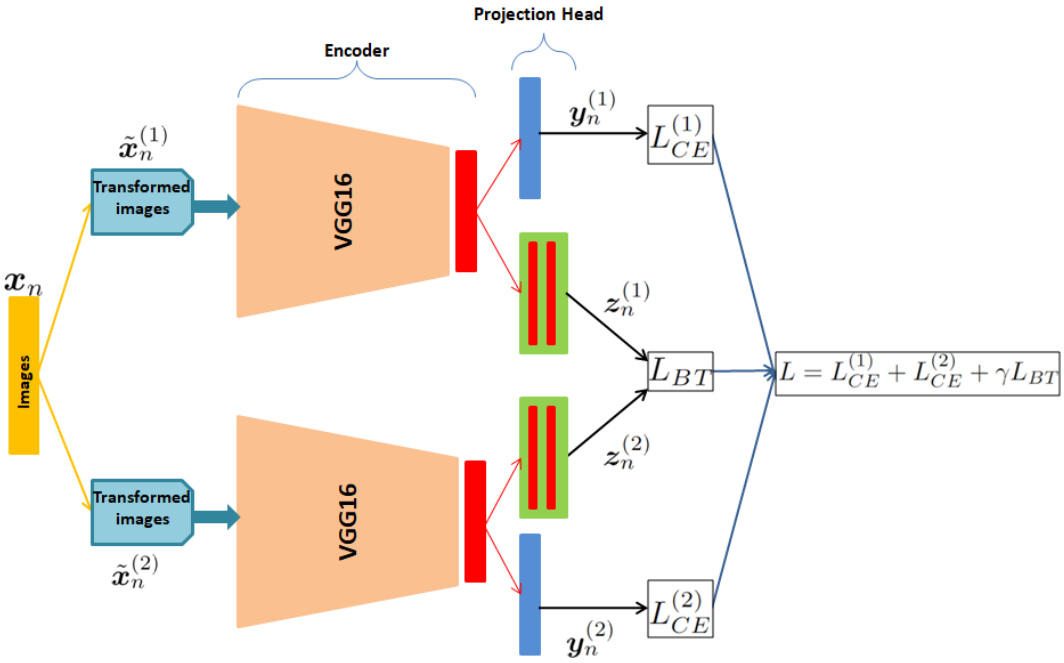


FIGURE 3.1: Supervised CNN Architecture, where Barlow Twins loss function and Cross Entropy Loss are calculated from the embeddings of projection head layers and are combine together to form a single loss function

3.1 Architecture for Supervised Learning with Barlow Twins

The architecture of the proposed supervised learning is shown in Fig. 3.1. There are two networks with the same parameters and two branches for classifiers and feature embedding for each network.

Let $\{(\boldsymbol{x}_n, \boldsymbol{t}_n)|n = 1, \ldots, N\}$ be the set of training samples, where $\boldsymbol{t}_n = \begin{bmatrix} t_{n1} & \cdots & t_{nK} \end{bmatrix}^T$ is the one-hot vector representation of the target class of the image $\boldsymbol{x}_n$. The number of classes is assumed to be K. Then we apply two different perturbations to each of the training sample $\boldsymbol{x}_n$. The perturbed images are denoted as $\tilde{\boldsymbol{x}}_n^{(1)}$ and $\tilde{\boldsymbol{x}}_n^{(2)}$ and they are fed into the upper and the lower networks in Fig. 3.1.

The outputs for two classifiers are denoted as $\boldsymbol{y}_n^{(1)}$ and $\boldsymbol{y}_n^{(2)}$. Also, the embedding features for two networks are denoted as $\boldsymbol{z}_n^{(1)}$ and $\boldsymbol{z}_n^{(2)}$ The transformed input images are fed into the architecture in the form of two branches using a variant of the Siamese network. We call the output of the projection heads the 'embeddings'. These embeddings from the first projection head network are used to calculate the cross entropy loss for each network and the embeddings from the second projection head network are used to calculate the Barlow Twins loss function as shown in Fig. 3.1.

Each input image is randomly transformed during training to produce the two distorted views $\tilde{\boldsymbol{x}}_n^{(1)}$ and $\tilde{\boldsymbol{x}}_n^{(2)}$ as shown in Fig. 3.1. The image augmentation pipeline consists of the following transformations: random resize crop, random horizontal flip, random grayscale, and random solarize. We investigate the performance of our framework when applying augmentations individually or in pairs. Specifically, we always first randomly crop images and resize them to the same resolution, and then we apply the targeted transformations only to one branch of the framework, while randomly applying transformations to the other branch. The first two transformations (cropping and flipping) are always applied, while the random grayscale and random solarize are applied randomly, with desired probability.

3.2 Cross Entropy Loss

Cross-entropy loss measures the similarity between the given one-hot vector and the estimated output vector (probabilities) and it is used to calculate how accurate the trained deep learning model is. Since we have two networks the cross entropy losses for each network can be defined as

$$L_{CE}^{(1)} = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\log y_{nk}^{(1)} \tag{3.1}$$

$$L_{CE}^{(2)} = -\sum_{n}^{N}\sum_{k=1}^{K} t_{nk}\log y_{nk}^{(2)} \tag{3.2}$$

3.3 Barlow Twins Loss

Like other recent methods for self-supervised learning, Barlow Twins operates on a joint embedding of distorted images. In the proposed method, the Barlow Twins loss is defined by using the embedding feature vectors in mini-batch. Let

$$\mathbb{Z}^{(1)} = \left\{ \boldsymbol{z}_n^{(1)} = \begin{bmatrix} z_{n1}^{(1)} & \cdots z_{nM}^{(1)} \end{bmatrix}^T \middle| n = 1,\ldots,B \right\}$$
$$\mathbb{Z}^{(2)} = \left\{ \boldsymbol{z}_n^{(2)} = \begin{bmatrix} z_{n1}^{(2)} & \cdots z_{nM}^{(2)} \end{bmatrix}^T \middle| n = 1,\ldots,B \right\}$$

be the sets of embedding feature vectors in a given mini-batch, where M is the length of the feature vectors and B is the number of samples in mini-batch. Then the Barlow

Twins loss is defined as

$$L_{BT} = \sum_i (1 - C_{ii})^2 + \lambda \sum_i \sum_{j \neq i} (C_{ij})^2 \tag{3.3}$$

where λ is a positive constant trading off the importance of the first and second terms of the loss. The cross-correlation matrix $C = [C_{ij}]$ is computed between the sets of outputs of the two identical networks $\mathbb{Z}^{(1)}$ and $\mathbb{Z}^{(2)}$ along the batch dimension as

$$C_{ij} = \frac{\sum_{b=1}^{B} z_{bi}^{(1)} z_{bj}^{(2)}}{\sqrt{\sum_{b=1}^{B} \left(z_{bi}^{(1)}\right)^2} \sqrt{\sum_{b=1}^{B} \left(z_{bj}^{(2)}\right)^2}} \tag{3.4}$$

where b indexes batch samples and i, j index the vector dimension of the network's outputs. The cross-correlation matrix C is a square matrix with size $M \times M$. Each of the elements of the matrix C have the value between -1 (i.e. perfect anti-correlation) and 1 (i.e. perfect correlation).

Intuitively, the first term of Barlow Twins loss forces the diagonal elements of the cross-correlation matrix to be equal to 1. This makes the embedding features invariant to the perturbations applied to the input images. On the other hand, the second term forces the off-diagonal elements of the cross-correlation matrix to be equal to 0. This decorrelates the different elements of the embedding feature vectors. This decorrelation reduces the redundancy between the elements in the embedding feature vector, so that the output units contain non-redundant information.

3.4 Barlow Twins in Supervised Classification CNN

In the proposed methods, we combine the standard cross-entropy loss with the Barlow Twins loss. As shown in Fig. 3.1, we have two classifiers. Thus the total loss can be defind as

$$L = L_{CE}^{(1)} + L_{CE}^{(2)} + \gamma L_{BT}, \tag{3.5}$$

where γ is a hyper-parameter to control the strength of the Barlow Twins loss. By introducing the Barlow Twins loss, it is expected that each of the classifiers becomes robust to the perturbation and the generalization performance of the trained model is improved.

In test phase, we can use one of the backbone networks with classification head as one simple CNN.

Chapter 4

Experiments

To confirm the effectiveness of the proposed approach, we have performed experiments using different data sets (CIFAR-10 and STL-10) and different baseline CNNs (VGG-16 and ResNet-18).

4.1 Datasets

We evaluated our method on two datasets, CIFAR-10 and STL-10[5]. The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain exactly 5000 images from each class. The STL-10 dataset is inspired by the CIFAR-10 dataset but with some modifications. In particular, each class has fewer labeled training examples than in CIFAR-10, but a very large set of unlabeled examples is provided to learn image models prior to supervised training. This dataset consists of 10 classes with 96x96 pixels color images. There are 500 training images (10 pre-defined folds), 800 test images per class, and 100000 unlabeled images for unsupervised learning.

4.2 Experimental Setup

4.2.1 Network Architecture

In the encoder of the network architecture, we use VGG-16 and ResNet-18 networks without pre-trained weights. We change the output size of the last layer to 2048 output units, following the original work on Barlow Twins. The encoder network is followed by two parallel projection head networks, where the first projection head has one linear layer with 10 output units for classification and the second projection head has two linear layers with 2048 hidden units and 128 output units for the transformation-invariant embedding. The first layer is followed by a ReLU activation.

4.2.2 Siamese Neural Network

We use Siamese Neural Network which is one of the leading methods for deep learning and fine-tuning vector representations or even training a new model from scratch. This architecture contains two or more identical subnetworks. 'identical' here means, they have the same configuration with the same parameters and weights. Parameter updating is mirrored across both sub-networks. It is used to find the similarity of the inputs by comparing their feature vectors, so these networks are used in many applications. We learn to make these similarities closer if the categories of the pair are the same and to make the similarities apart if the categories are different.

4.2.3 Image Augmentations

Each input image are transformed randomly to produce the two distorted views shown in Fig. 3.1. The image augmentation pipeline consists of the following transformations: random resize crop, random horizontal flip, random grayscale, and random solarize. We investigate the performance of our framework when applying augmentations individually or in pairs. Specifically, we always first randomly crop images and resize them to the same resolution, and then we apply the targeted transformations only to one branch of the framework, while randomly applying transformations to the other branch. The first two transformations (cropping and flipping) are always applied, while the random grayscale and random solarize are applied randomly, with desired probability.

4.2.4 Optimization

We use the SGD optimizer with a momentum of 0.9 and train the model for 1000 epochs at a learning rate of 0.01. We also add some weight decay with a rate of 0.0001. We run experiments with the values of γ 0.0001, 0.001, and 0.01, and our results show that the optimal value for this parameter changes under different configurations. We use a batch size of 512 for VGG-16 and 128 for ResNet-18.

Training time taken for the VGG-16 network in GPU is approximately 22 hours for each experiment. For the ResNet-18 network, the training time takes approximately 5 hours for each experiment.

4.3 Evaluation on CIFAR-10 dataset

We train the VGG-16 network model without pre-trained weights with our proposed method on the CIFAR-10 dataset. The network is trained with different gamma parameters ranging from 0.0001 to 0.01. The gamma parameter value which results in the best accuracy is compared with the cross entropy loss (baseline model), where the baseline cross-entropy loss accuracies are calculated from the transformed train

and test data. Additionally, we show evaluations on augmented data to directly confirm our method's ability to learn transformation-invariant features. We perform the same process when training the ResNet-18 network model. The same experiments are conducted multiple times and the performance variance is reported with the mean accuracy and standard deviation to all tables of ResNet-18 architecture. The classification accuracies obtained on the CIFAR-10 dataset with the VGG-16 network for train data, test data, and augmented test data are reported in Table 4.1. The gamma parameter of 0.001 consistently gives the best accuracy, outperforming the baseline (standard classification training with cross-entropy loss). Fig.4.1 and Fig.4.2 shows the 2-dimensional PCA and t-SNE visualizations of augmented train embeddings for baseline cross entropy loss and different gamma parameter values. The visualizations of embeddings from our proposed method show cleaner class separation than the baseline method. The classification accuracies obtained on the CIFAR-10 dataset with ResNet-18 network for train data, test data, and augmented test data are reported in Table 4.2. The gamma parameter of 0.001 shows the best accuracy under all configurations. Fig.4.4 shows the 2-dimensional t-SNE visualizations of augmented train embeddings for baseline cross entropy loss and different gamma parameter values.

TABLE 4.1: Classification accuracy results on CIFAR-10 dataset with VGG-16 architecture

	Proposed Method (mean acc±std.deviation)			Cross Entropy Loss (baseline)
Gamma parameter(γ)	**0.0001**	**0.001**	**0.01**	
Train Acc	99.59±0.09	**99.61±0.23**	99.53±0.06	98.56±0.51
Test Acc-test data	88.90±0.20	**88.94±0.78**	88.59±0.67	86.15±2.28
Test Acc-Aug test data	87.88±0.41	**87.91±0.85**	87.59±0.49	86.37±1.03

TABLE 4.2: Classification accuracy results on CIFAR10 dataset with ResNet-18 architecture

	Proposed Method (mean acc±std.deviation)			Cross Entropy Loss (baseline)
Gamma parameter(γ)	**0.0001**	**0.001**	**0.01**	
Train Acc	98.29±0.25	**98.58±0.19**	98.53±0.5	97.39±0.25
Test Acc-test data	81.78±1.51	**83.63±0.38**	82.86±0.33	82.78±0.26
Test Acc-Aug test data	80.13±2.61	**81.3±0.14**	80.84±0.92	80.85±0.43

4.4 Evaluation on STL-10 dataset

For the STL-10 dataset, we consider 5000 labeled train samples and 5000 unlabelled train samples to train the VGG-16 network model without pre-trained weights with our proposed method. We used only the 5000 labeled train samples to train the baseline cross-entropy loss. The classification accuracies obtained on the STL-10 dataset

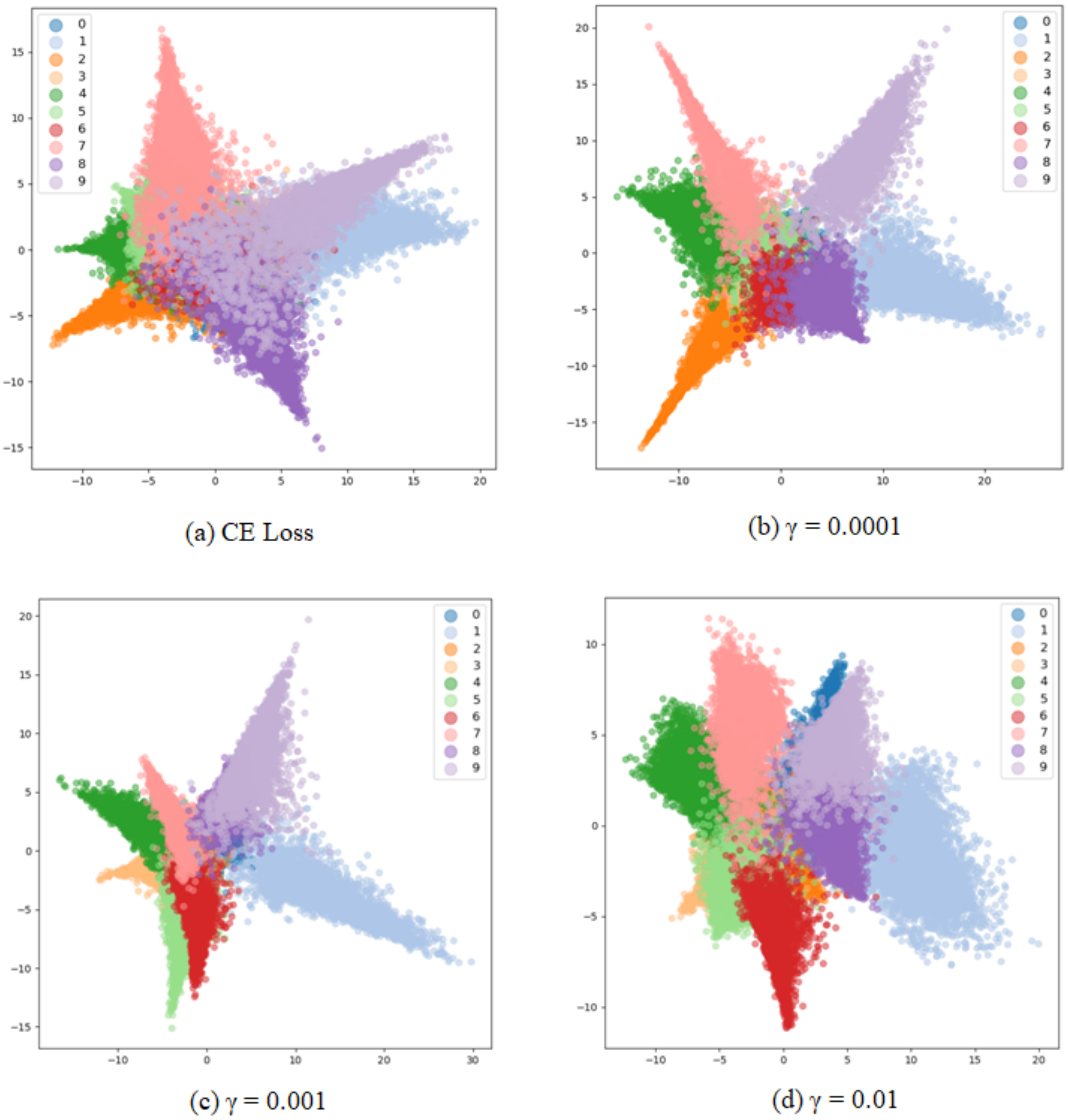
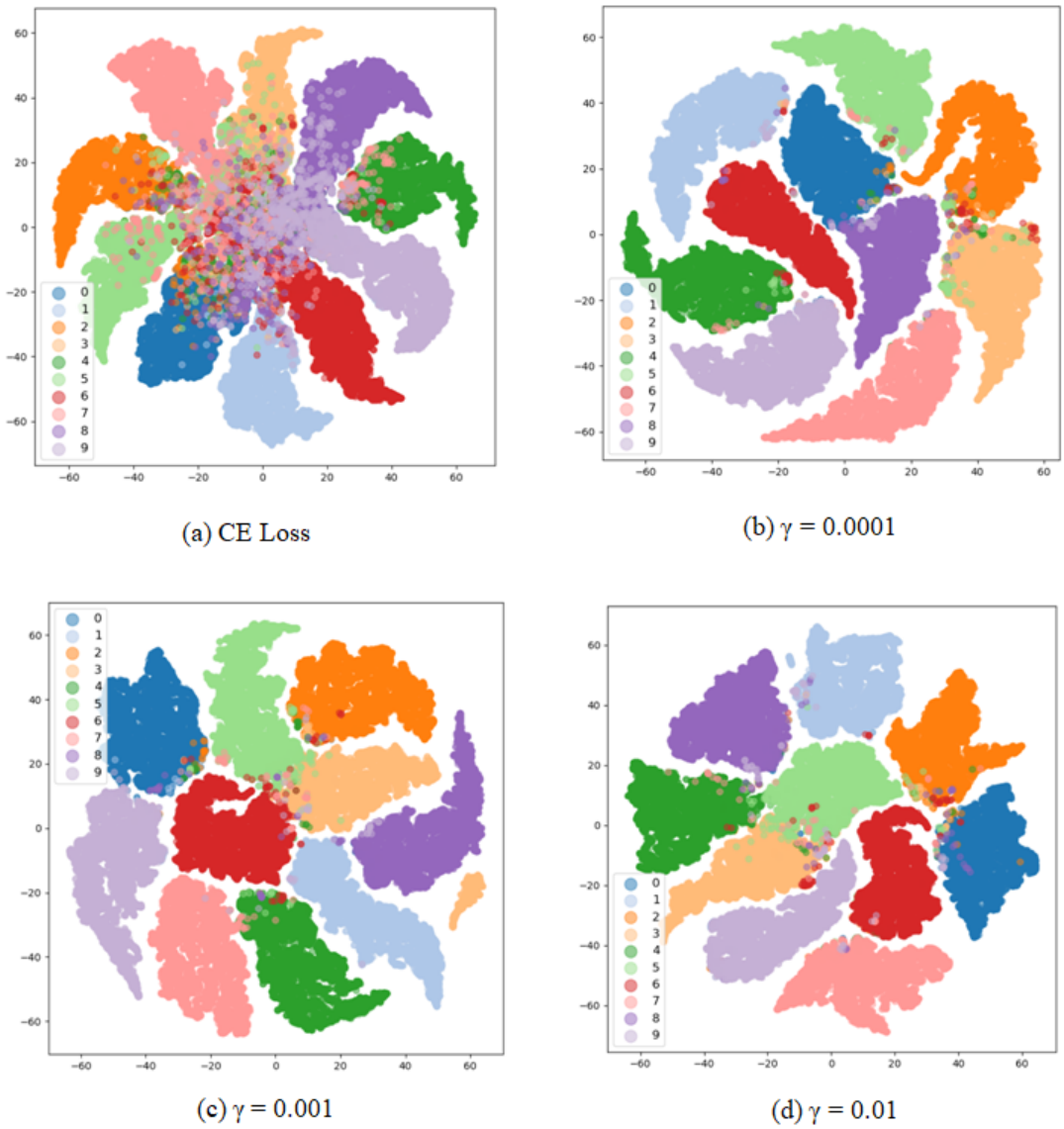


FIGURE 4.1: 2-dimensional PCA visualizations of augmented train embeddings from CIFAR-10 dataset with VGG-16 architecture

with the VGG-16 network for train data, test data, and augmented test data are reported in Table 4.3. The gamma value of 0.01 shows the best accuracy under all configurations, outperforming the baseline by over 2% on the test data. Fig.4.5 and Fig.4.6 shows the 2-dimensional PCA and t-SNE visualizations of augmented train embeddings for baseline cross entropy loss and different γ parameters values. The visualizations of embeddings for γ value 0.01 is more clearly clustered than the baseline cross-entropy. The classification accuracies obtained on the STL-10 dataset with the ResNet-18 network for train data, test data, and augmented test data are reported in Table 4.4. The gamma parameter of 0.001 gives the best accuracy, again outperforming the standard classification baseline. Fig.4.8 shows the 2-dimensional t-SNE visualizations of augmented train embeddings for baseline cross entropy loss and different gamma parameter values.



(a) CE Loss (b) $\gamma = 0.0001$

(c) $\gamma = 0.001$ (d) $\gamma = 0.01$

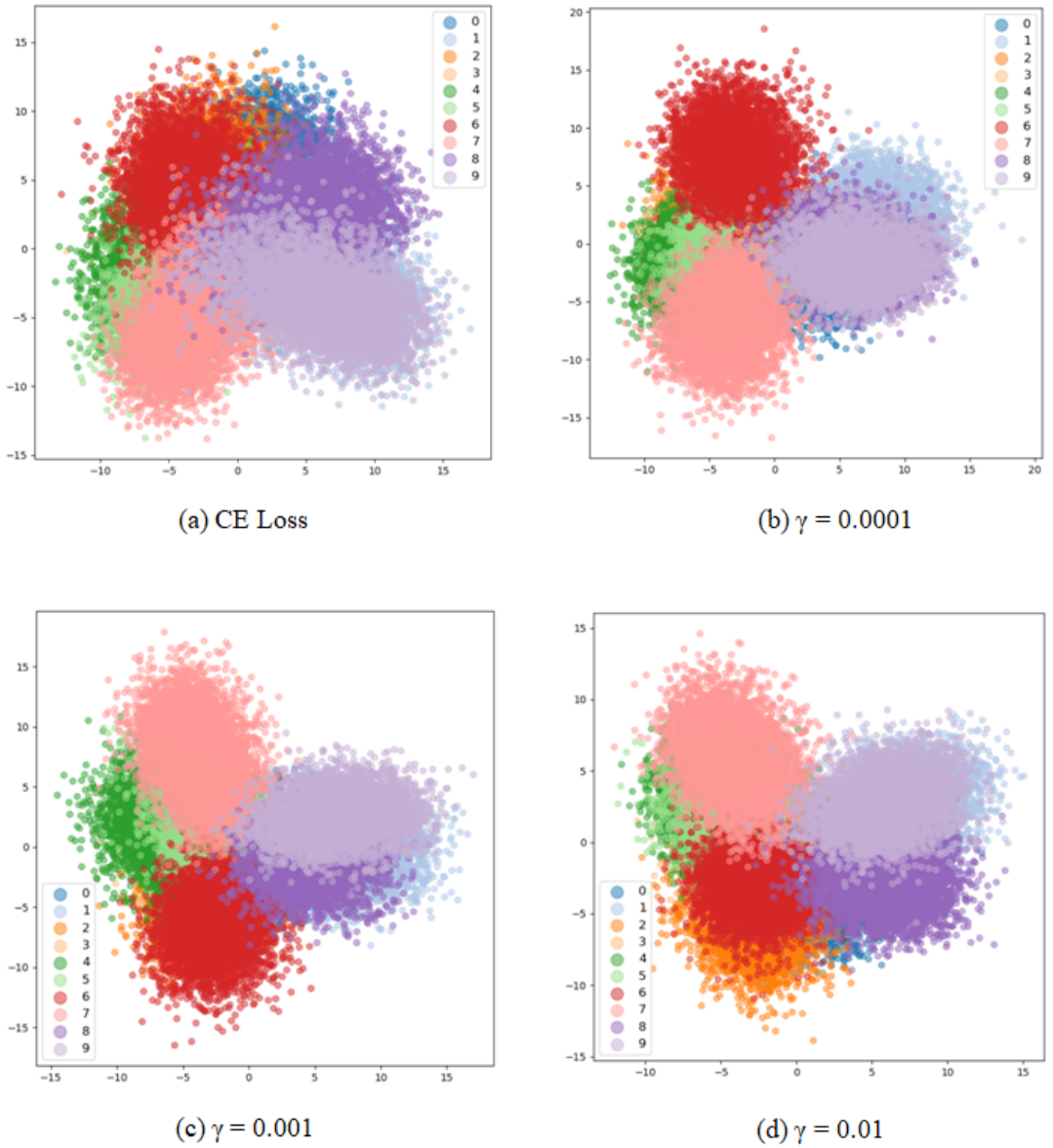
FIGURE 4.2: 2-dimensional t-SNE visualizations of augmented train embeddings from CIFAR-10 dataset with VGG-16 architecture

TABLE 4.3: Classification accuracy results on STL-10 dataset with VGG-16 architecture

	Proposed Method (mean acc±std.deviation)			Cross Entropy Loss (baseline)
Gamma parameter(γ)	**0.0001**	**0.001**	**0.01**	
Train Acc	99.74±0.26	99.85±0.12	**99.88±0.13**	99.82±0.18
Test Acc-test data	72.81±0.42	74.45±0.48	**76.87±2.22**	70.97±4.63
Test Acc-Aug test data	72.44±0.45	74.19±0.57	**76.64±2.52**	69.88±3.19

4.5 Evaluation of different augmentations on CIFAR-10 dataset

We train the ResNet-18 network model without pre-trained weights with our proposed method on the CIFAR-10 dataset. The network is trained with different gamma parameters ranging from 0.0001 to 0.01. The gamma parameter value which results



(a) CE Loss (b) $\gamma = 0.0001$

(c) $\gamma = 0.001$ (d) $\gamma = 0.01$

FIGURE 4.3: 2-dimensional PCA visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture

TABLE 4.4: Classification accuracy results on STL-10 dataset with ResNet-18 architecture

	Proposed Method (mean acc±std.deviation)			Cross Entropy Loss (baseline)
Gamma parameter(γ)	**0.0001**	**0.001**	**0.01**	
Train Acc	99.86±0.02	**99.88±0.13**	99.17±1.13	99.36±0.33
Test Acc-test data	72.78±1.06	**73.66±0.37**	71.74±3.45	70.46±0.87
Test Acc-Aug test data	72.34±0.33	**73.51±0.46**	71.33±2.49	71.28±0.27

in the best accuracy is compared with the cross entropy loss (baseline model), where the baseline cross-entropy loss accuracies are calculated from the transformed train and test data. Additionally, we show evaluations on augmented data to directly confirm our method's ability to learn transformation-invariant features.

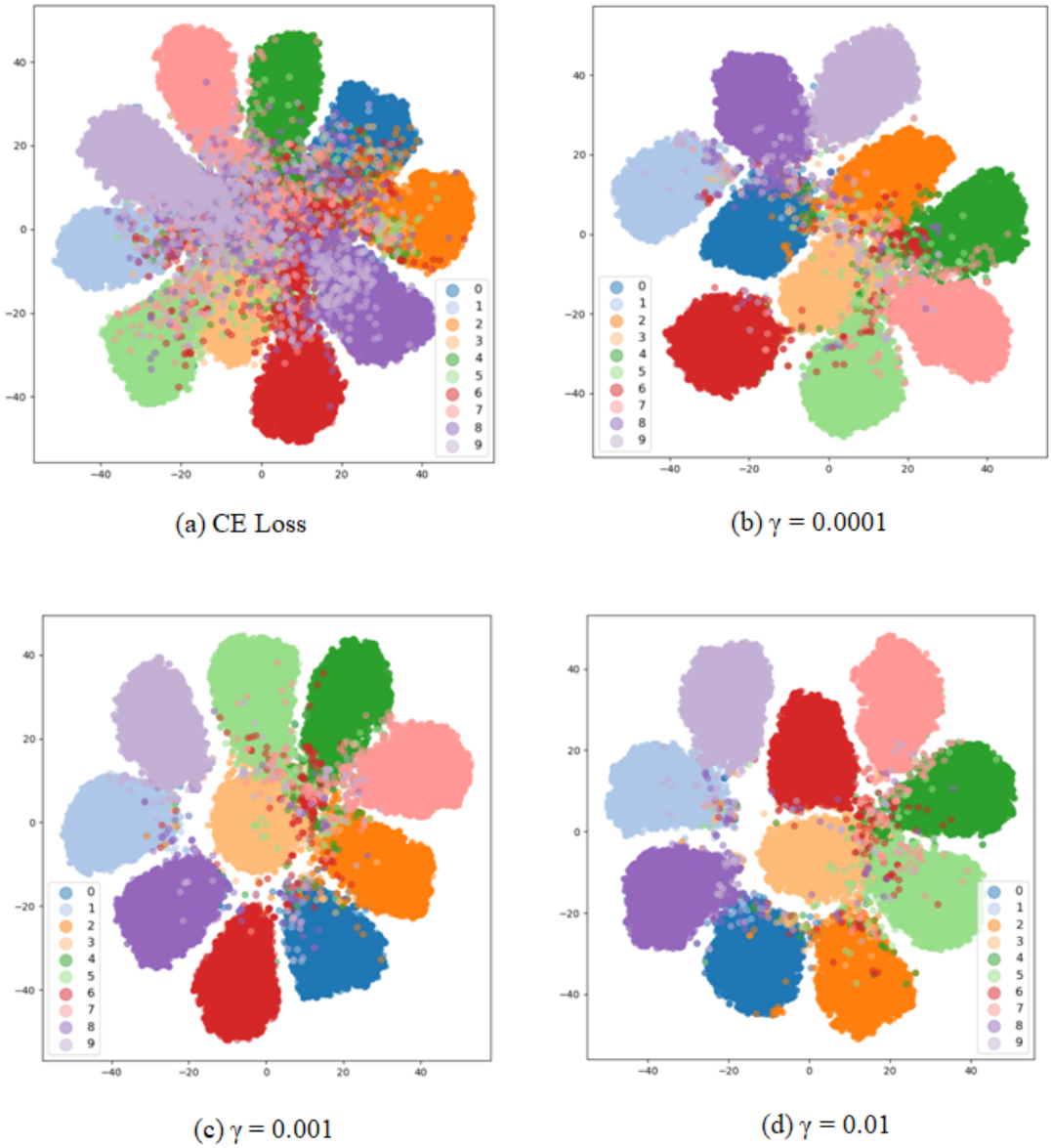


FIGURE 4.4: 2-dimensional t-SNE visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture

4.5.1 Evaluation of Random Resize Crop augmentation on CIFAR-10 dataset

We train ResNet-18 network model with only random resize crop augmentation on the CIFAR-10 dataset. The classification accuracies obtained on the CIFAR-10 dataset with ResNet-18 network for train data, test data, and augmented test data are reported in Table 4.5. The gamma parameter of 0.01 consistently gives the best accuracy, outperforming the baseline (standard classification training with cross entropy loss). Fig.4.9 and Fig.4.10 show the 2-dimensional PCA and t-SNE visualizations of augmented train embeddings for baseline cross entropy loss and different gamma parameter values. The visualizations of embeddings from our proposed method show cleaner class separation than the baseline method.

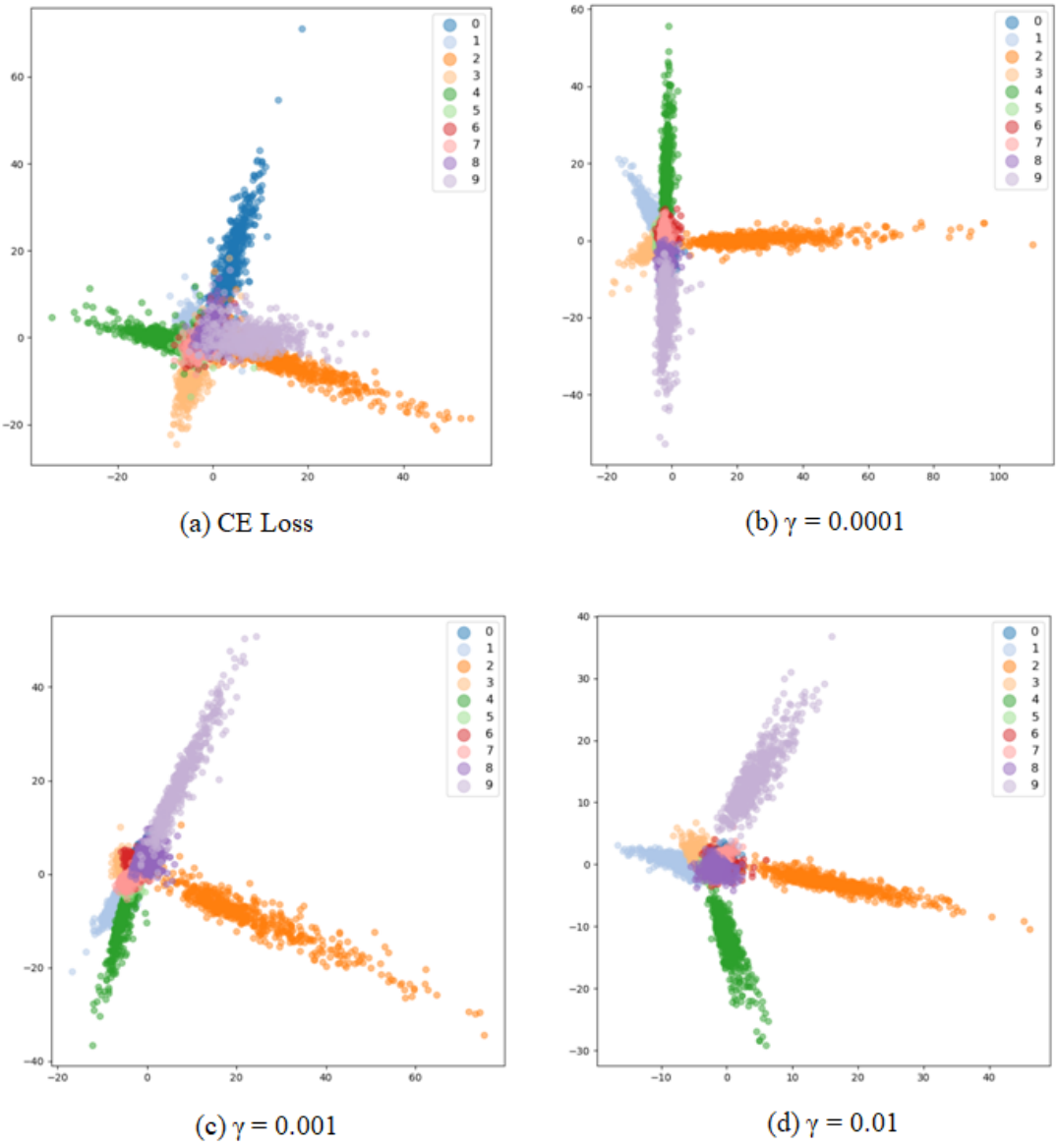


FIGURE 4.5: 2-dimensional PCA visualizations of augmented train embeddings from STL-10 dataset with VGG-16 architecture

TABLE 4.5: Classification accuracy results for Random Resize Crop augmentation on CIFAR-10 dataset with ResNet-18 architecture

	Proposed Method			Cross Entropy Loss (baseline)
Gamma parameter(γ)	**0.0001**	**0.001**	**0.01**	
Train Acc	99.17	98.81	**99.4**	98.39
Test Acc-test data	81.53	81.23	**82.11**	81.36
Test Acc-Aug test data	80.98	80.2	**82**	80.56

4.5.2 Evaluation of Random Horizontal Flip augmentation on CIFAR-10 dataset

We train the ResNet-18 network model with only random horizontal flip augmentation on the CIFAR-10 dataset. The classification accuracies obtained on the CIFAR-10

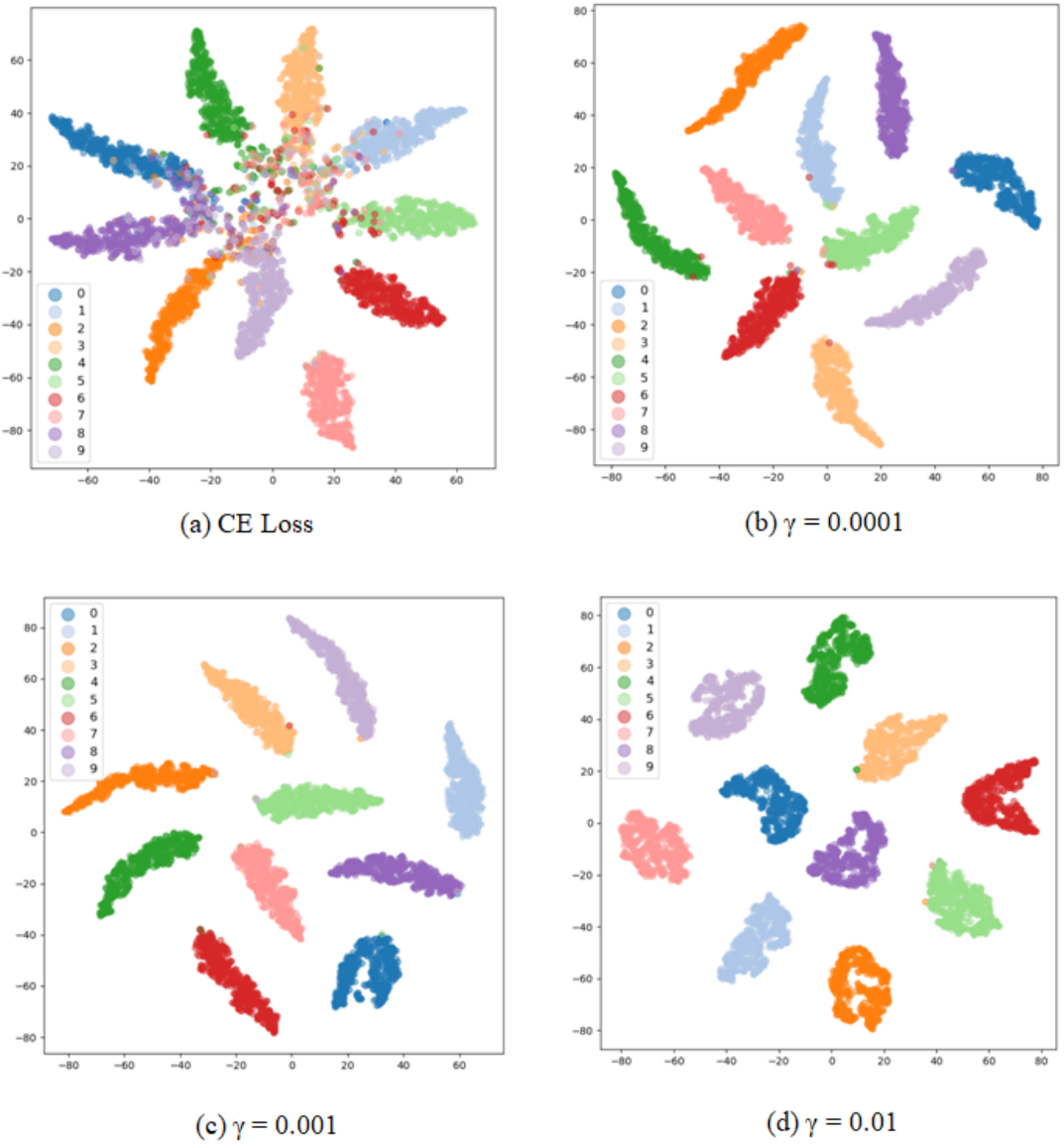
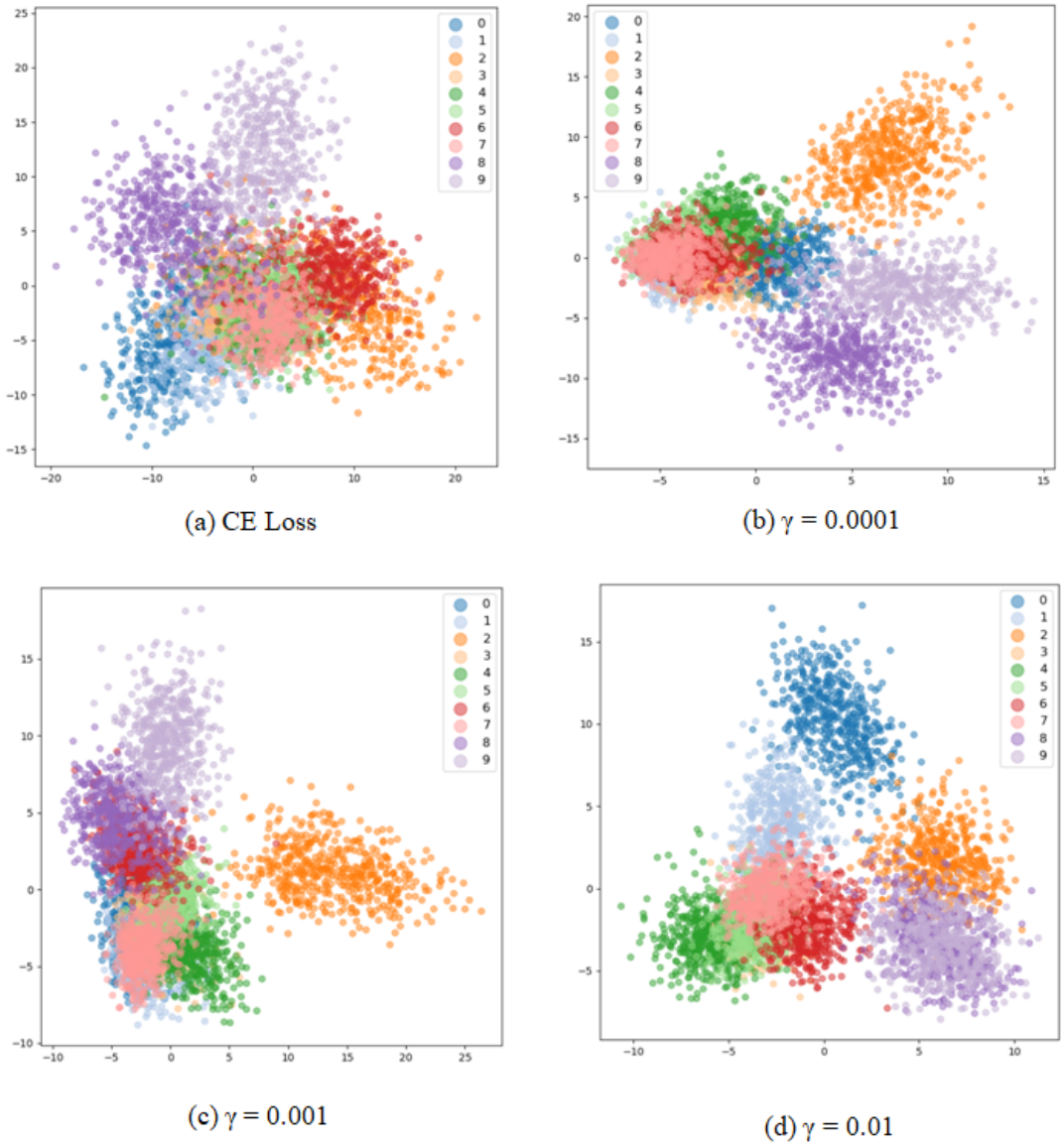


FIGURE 4.6: 2-dimensional t-SNE visualizations of augmented train embeddings from STL-10 dataset with VGG-16 architecture

dataset with ResNet-18 network for train data, test data, and augmented test data are reported in Table 4.6. The gamma parameter of 0.01 consistently gives the best accuracy, outperforming the baseline (standard classification training with cross-entropy loss). Fig.4.11 and Fig.4.12 show the 2-dimensional PCA and t-SNE visualizations of augmented train embeddings for baseline cross entropy loss and different gamma parameter values. The visualizations of embeddings from our proposed method show cleaner class separation than the baseline method.

4.5.3 Evaluation of Random Gray Scale augmentation on CIFAR-10 dataset

We train the ResNet-18 network model with only random grayscale augmentation on the CIFAR-10 dataset. The classification accuracies obtained on the CIFAR-10 dataset with ResNet-18 network for train data, test data, and augmented test data



(a) CE Loss (b) $\gamma = 0.0001$

(c) $\gamma = 0.001$ (d) $\gamma = 0.01$

FIGURE 4.7: 2-dimensional PCA visualizations of augmented train embeddings from STL-10 dataset with ResNet-18 architecture

TABLE 4.6: Classification accuracy results for Random Horizontal Flip augmentation on CIFAR-10 dataset with ResNet-18 architecture

	Proposed Method			Cross Entropy Loss (baseline)
Gamma parameter(γ)	**0.0001**	**0.001**	**0.01**	
Train Acc	99.28	99.42	**99.45**	99.44
Test Acc-test data	79.18	79.31	**80.51**	79.82
Test Acc-Aug test data	66	63.96	**69.2**	64

are reported in Table 4.7. The gamma parameter of 0.0001 consistently gives the best

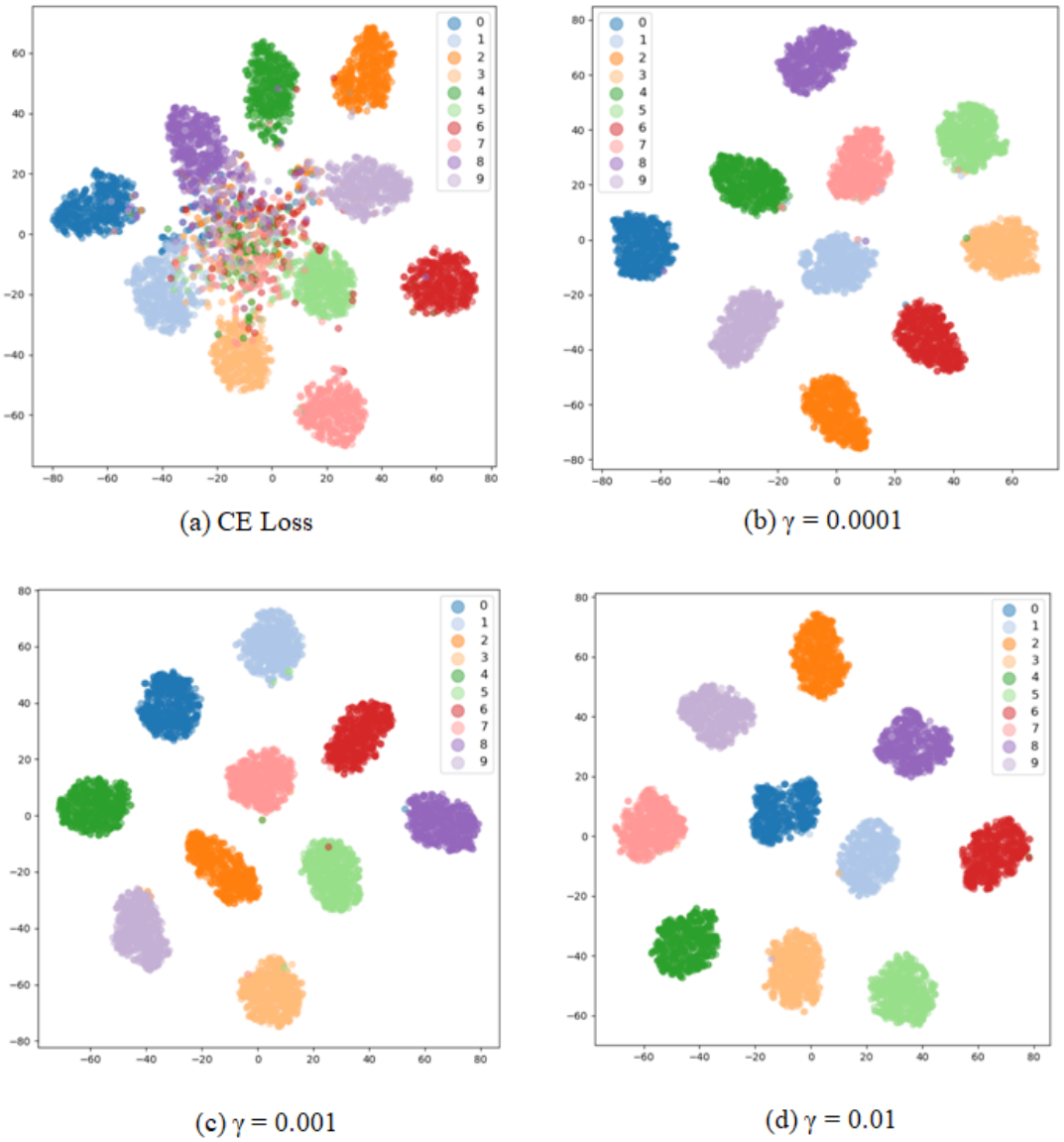


FIGURE 4.8: 2-dimensional t-SNE visualizations of augmented train embeddings from STL-10 dataset with ResNet-18 architecture

accuracy, outperforming the baseline (standard classification training with cross-entropy loss). Fig.4.13 and Fig.4.14 show the 2-dimensional PCA and t-SNE visualizations of augmented train embeddings for baseline cross entropy loss and different gamma parameter values. The visualizations of embeddings from our proposed method show cleaner class separation than the baseline method.

4.5.4 Evaluation of Random Solarize augmentation on CIFAR-10 dataset

We train the ResNet-18 network model with only random solarize augmentation on the CIFAR-10 dataset. The classification accuracies obtained on the CIFAR-10 dataset with ResNet-18 network for train data, test data, and augmented test data are reported in Table 4.8. The gamma parameter of 0.001 consistently gives the best accuracy, outperforming the baseline (standard classification training with cross-entropy

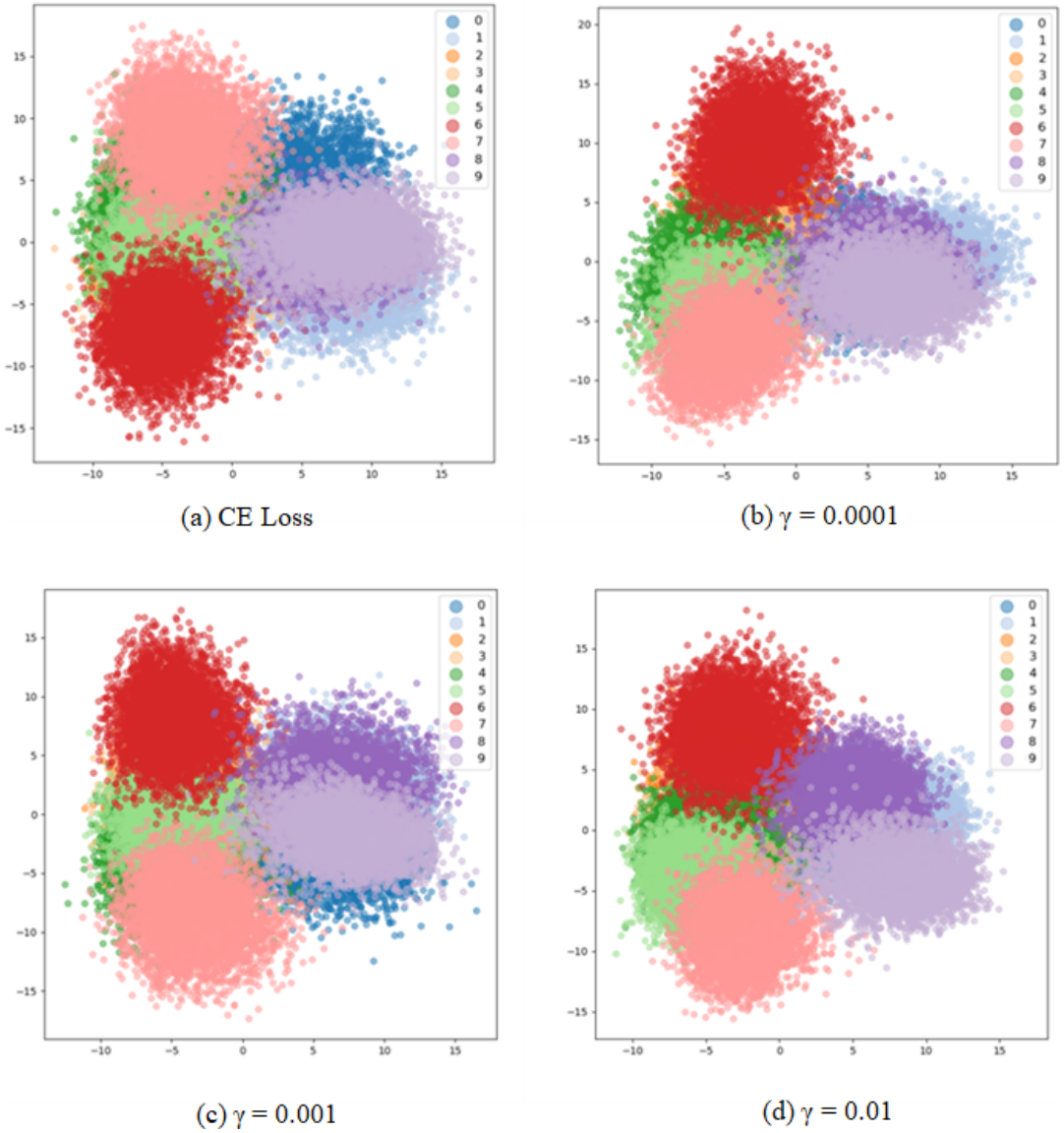


FIGURE 4.9: Random Resize Crop: 2-dimensional PCA visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture

TABLE 4.7: Classification accuracy results for Random Gray Scale augmentation on CIFAR-10 dataset with ResNet-18 architecture

	Proposed Method			Cross Entropy Loss (baseline)
Gamma parameter(γ)	**0.0001**	**0.001**	**0.01**	
Train Acc	**100**	99.91	100	99.6
Test Acc-test data	**77**	74.96	75.97	75.09
Test Acc-Aug test data	**63.58**	60.2	62.49	58.2

loss). Fig.4.15 and Fig.4.16 show the 2-dimensional PCA and t-SNE visualizations of augmented train embeddings for baseline cross entropy loss and different gamma parameter values. The visualizations of embeddings from our proposed method show cleaner class separation than the baseline method.

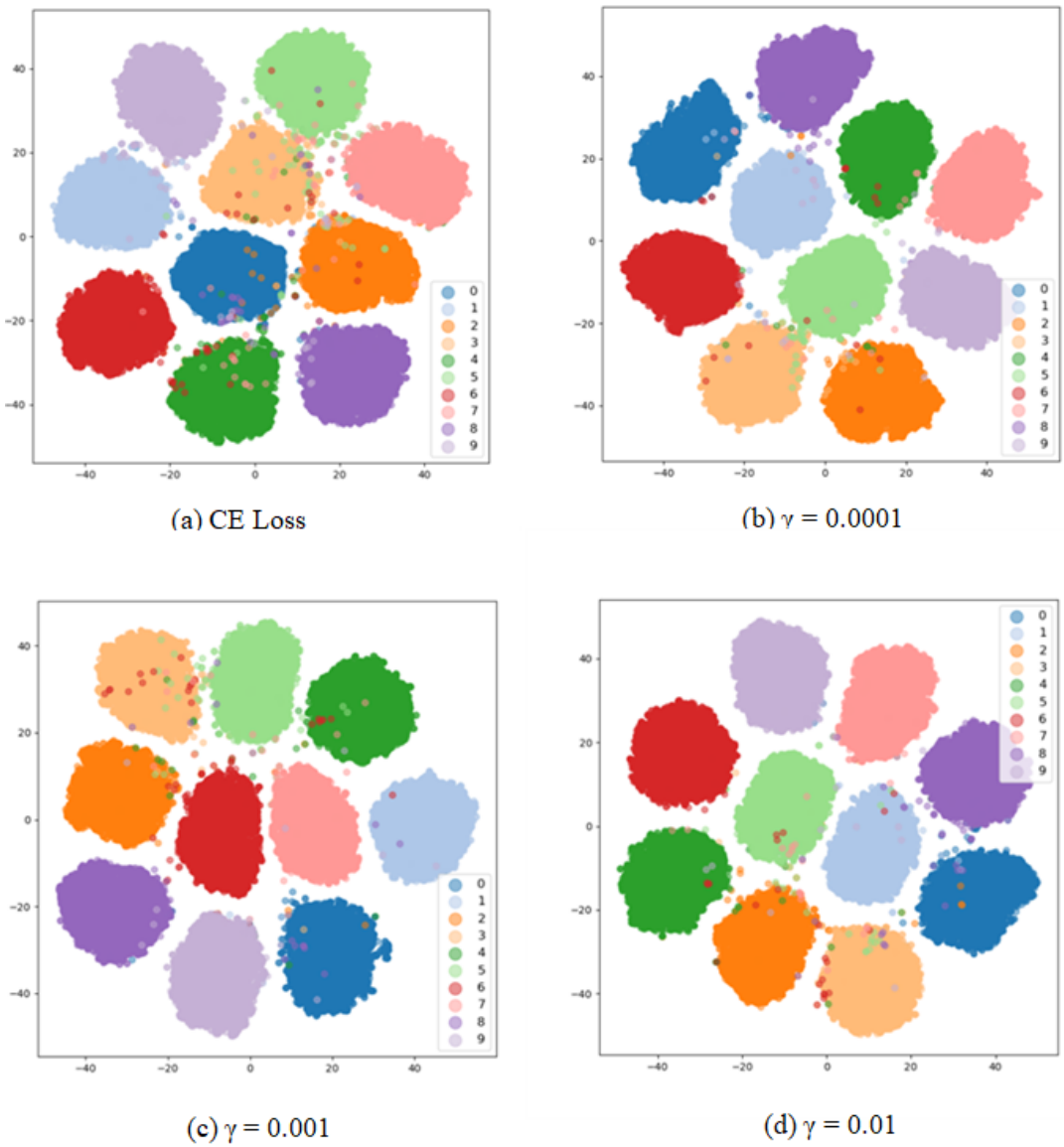


FIGURE 4.10: Random Resize Crop: 2-dimensional T-SNE visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture

TABLE 4.8: Classification accuracy results for Random Solarize augmentation on CIFAR-10 dataset with ResNet-18 architecture

	Proposed Method			Cross Entropy Loss (baseline)
Gamma parameter(γ)	**0.0001**	**0.001**	**0.01**	
Train Acc	98.73	**100**	100	99.64
Test Acc-test data	74.58	**79.6**	75.43	75.4
Test Acc-Aug test data	59.4	**64.2**	61	59.57

4.6 Evaluation of combinations of augmentation on CIFAR-10 dataset

4.6.1 Evaluation of Random Resize Crop and Random Horizontal Flip augmentations on CIFAR-10 dataset

We train ResNet-18 network model with random resize crop and random horizontal flip augmentation on CIFAR-10 dataset. The classification accuracies obtained on the CIFAR-10 dataset with ResNet-18 network for train data, test data and augmented test data are reported in Table 4.9. The gamma parameter of 0.01 consistently gives the best accuracy, outperforming the baseline (standard classification training with cross entropy loss). Fig.4.17 and Fig.4.18 shows the 2-dimensional PCA and t-SNE visualizations of augmented train embeddings for baseline cross entropy loss and different gamma parameter values. The visualizations of embeddings from our proposed method show cleaner class separation than the baseline method.

TABLE 4.9: Classification accuracy results for Random Resize Crop and Random Horizontal Flip augmentations on CIFAR-10 dataset with ResNet-18 architecture

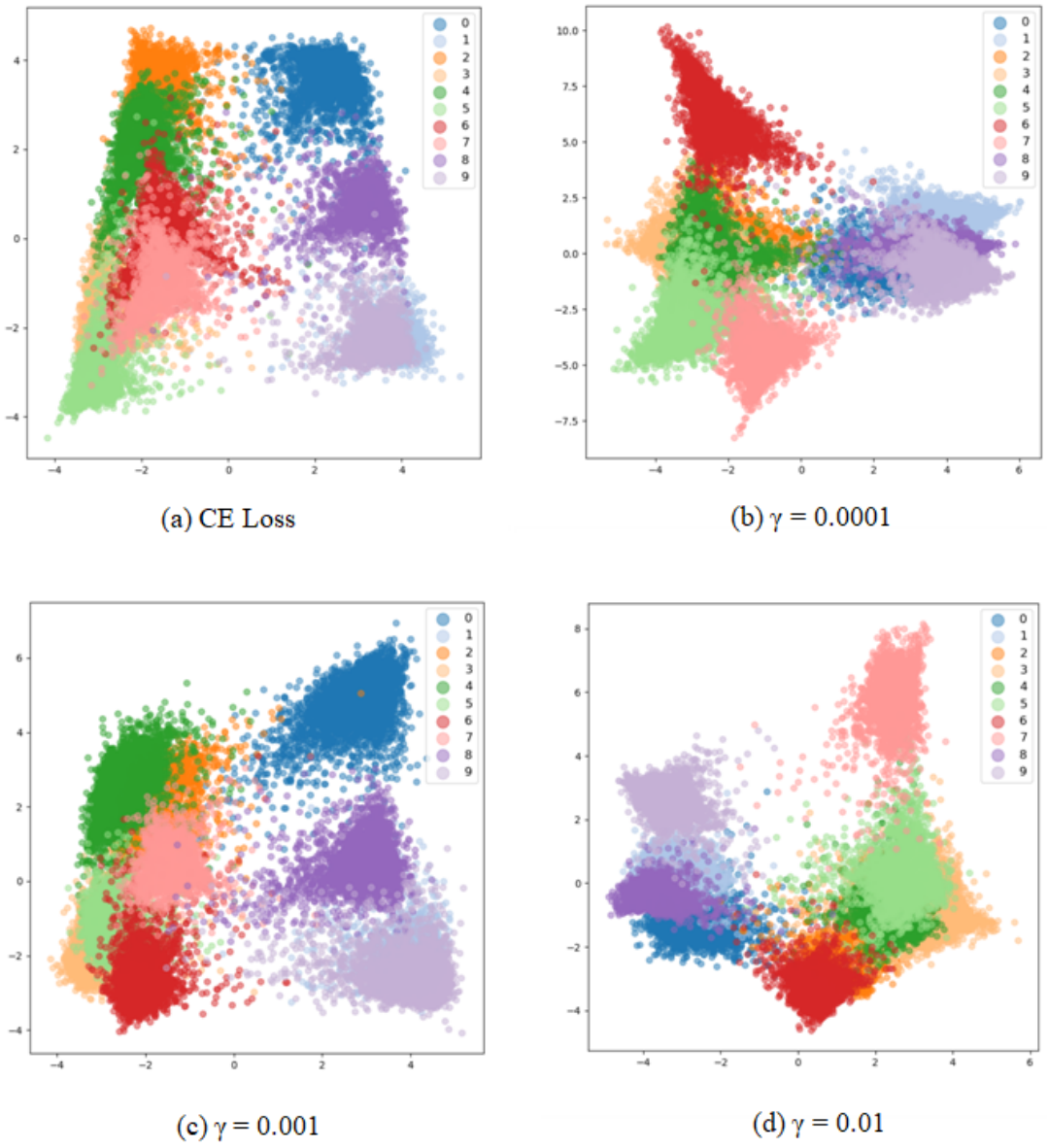
	Proposed Method			Cross Entropy Loss (baseline)
Gamma parameter(γ)	**0.0001**	**0.001**	**0.01**	
Train Acc	99.12	98.92	**99.30**	98.53
Test Acc-test data	83.42	83.46	**83.59**	83.17
Test Acc-Aug test data	81.04	81.31	**81.59**	80.77

4.6.2 Evaluation of Random Gray Scale and Random Solarize augmentations on CIFAR-10 dataset

We train ResNet-18 network model with random grayscale and random solarize augmentation on CIFAR-10 dataset. The classification accuracies obtained on the CIFAR-10 dataset with ResNet-18 network for train data, test data and augmented test data are reported in Table 4.10. The gamma parameter of 0.0001 consistently gives the best accuracy, outperforming the baseline (standard classification training with cross entropy loss). Fig.4.19 and Fig.4.20 shows the 2-dimensional PCA and t-SNE visualizations of augmented train embeddings for baseline cross entropy loss and different gamma parameter values. The visualizations of embeddings from our proposed method show cleaner class separation than the baseline method.

TABLE 4.10: Classification accuracy results for Random Gray Scale and Random Solarize augmentations on CIFAR-10 dataset with ResNet-18 architecture

	Proposed Method			Cross Entropy Loss (baseline)
Gamma parameter(γ)	**0.0001**	**0.001**	**0.01**	
Train Acc	**100**	98.42	100	99.64
Test Acc-test data	**77.82**	73.41	76.27	74.06
Test Acc-Aug test data	**63.06**	57.59	61.21	59.53



(a) CE Loss

(b) $\gamma = 0.0001$

(c) $\gamma = 0.001$

(d) $\gamma = 0.01$

FIGURE 4.11: Random Horizontal Flip: 2-dimensional PCA visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture

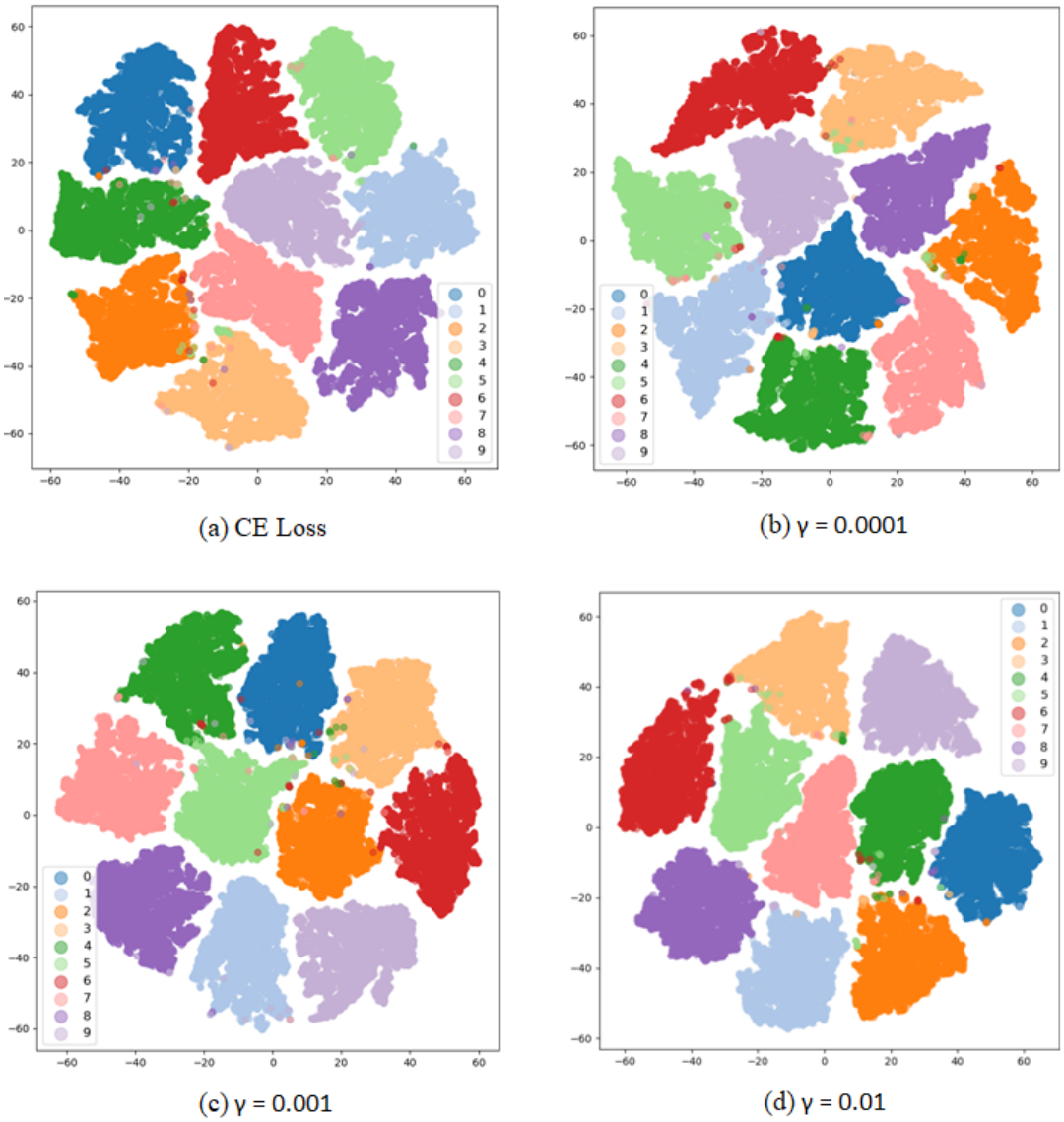
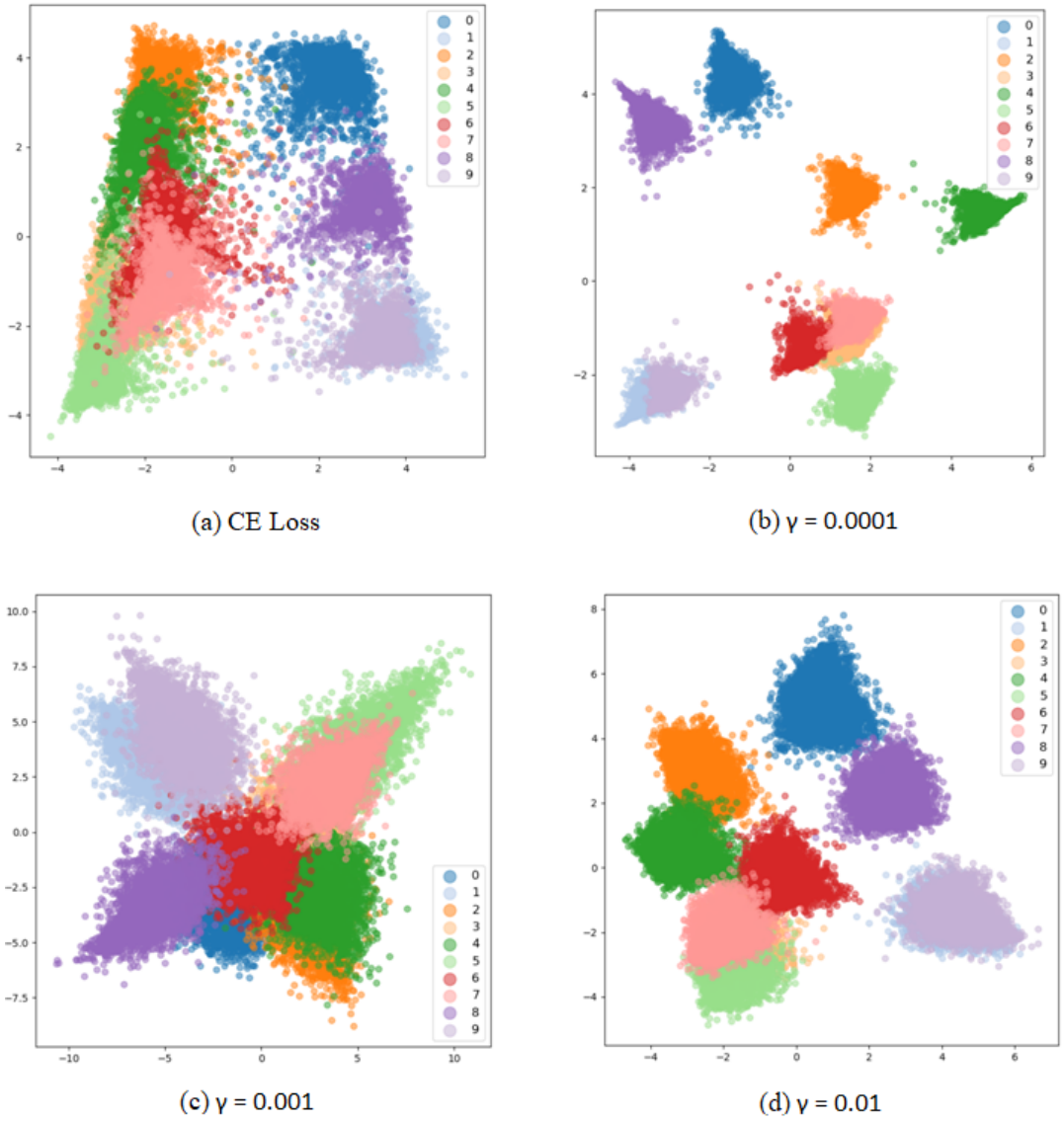


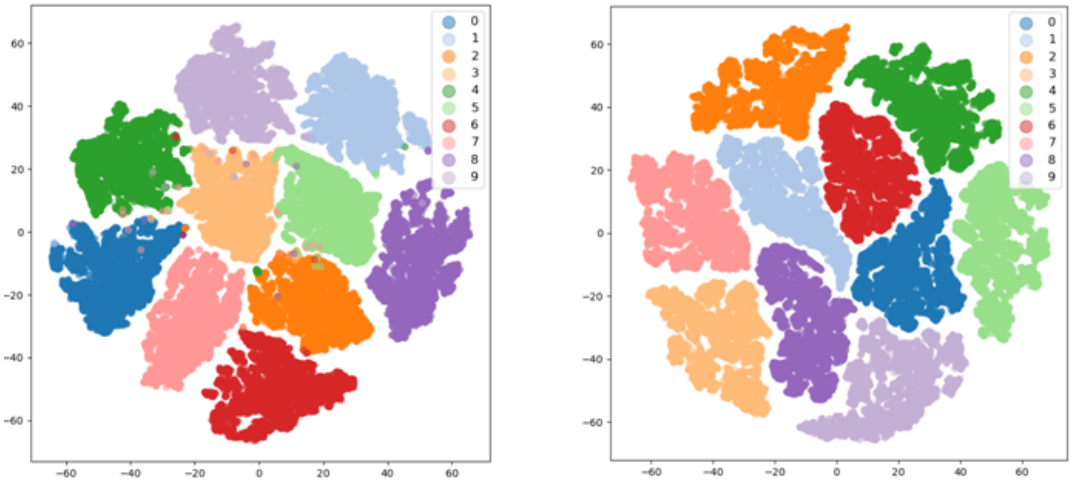
FIGURE 4.12: Random Horizontal Flip: 2-dimensional T-SNE visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture



(a) CE Loss (b) γ = 0.0001

(c) γ = 0.001 (d) γ = 0.01

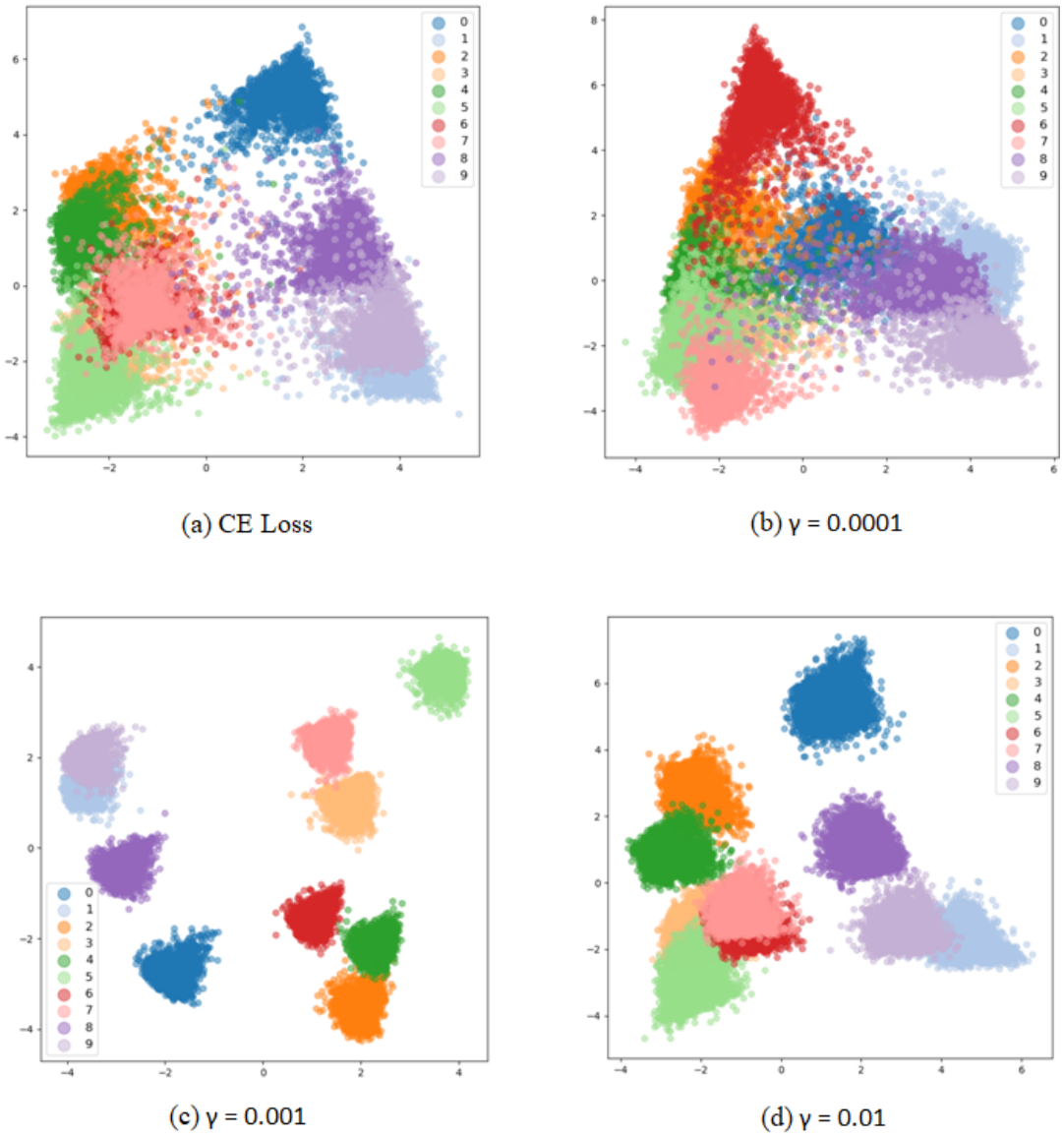
FIGURE 4.13: Random Gray Scale: 2-dimensional PCA visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture



(a) CE Loss (b) γ = 0.0001

(c) γ = 0.001 (d) γ = 0.01

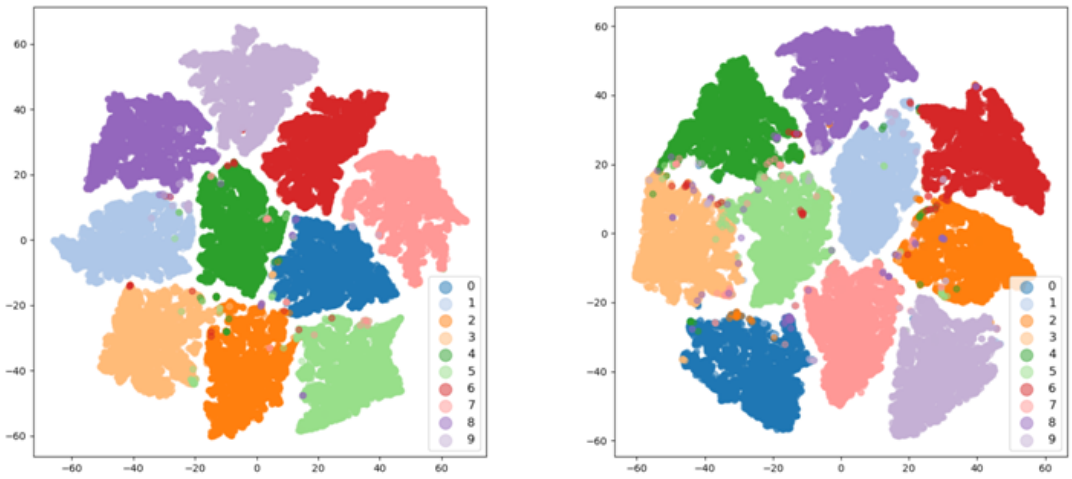
FIGURE 4.14: Random Gray Scale: 2-dimensional T-SNE visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture



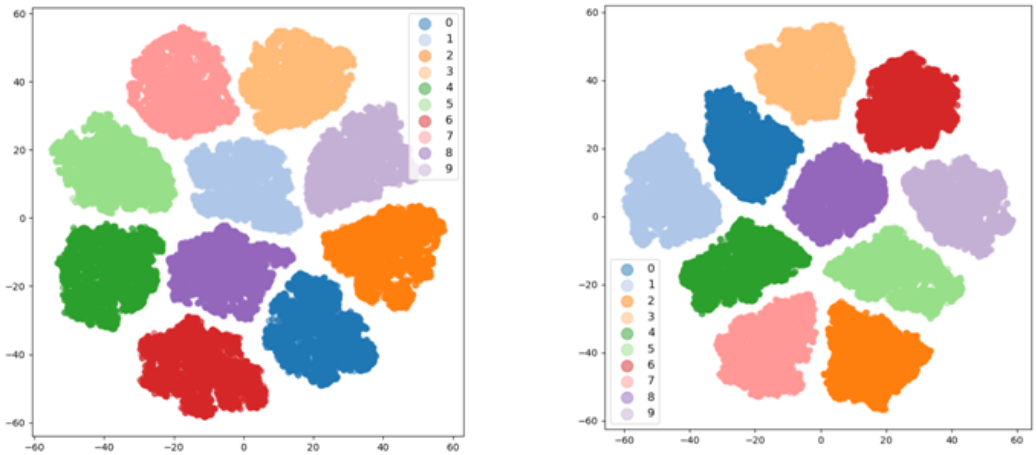
(a) CE Loss (b) γ = 0.0001

(c) γ = 0.001 (d) γ = 0.01

FIGURE 4.15: Random Solarize: 2-dimensional PCA visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture

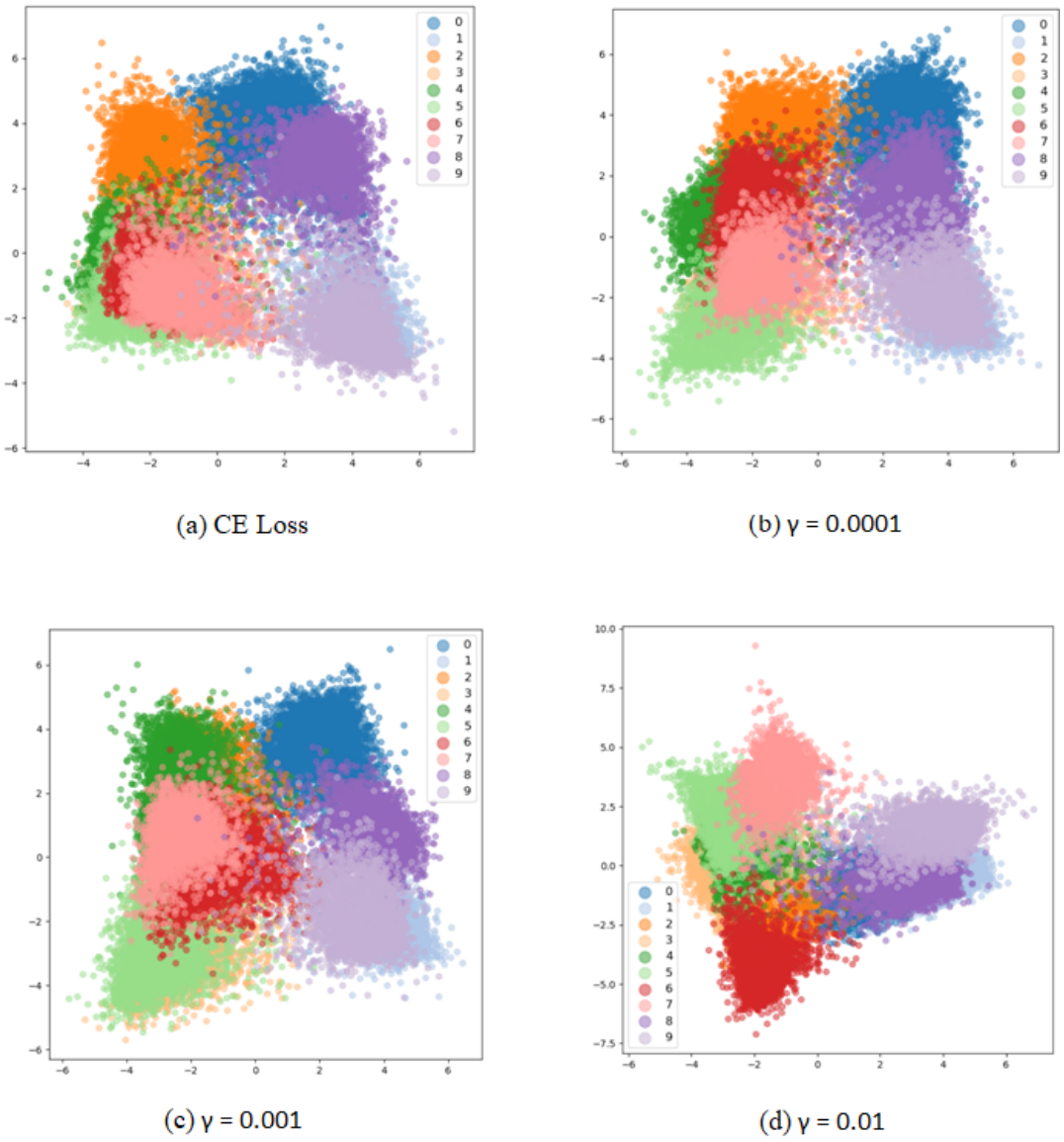


(a) CE Loss (b) γ = 0.0001



(c) γ = 0.001 (d) γ = 0.01

FIGURE 4.16: Random Solarize: 2-dimensional T-SNE visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture



(a) CE Loss

(b) γ = 0.0001

(c) γ = 0.001

(d) γ = 0.01

FIGURE 4.17: Random Resize Crop and Random Horizontal Flip: 2-dimensional PCA visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture

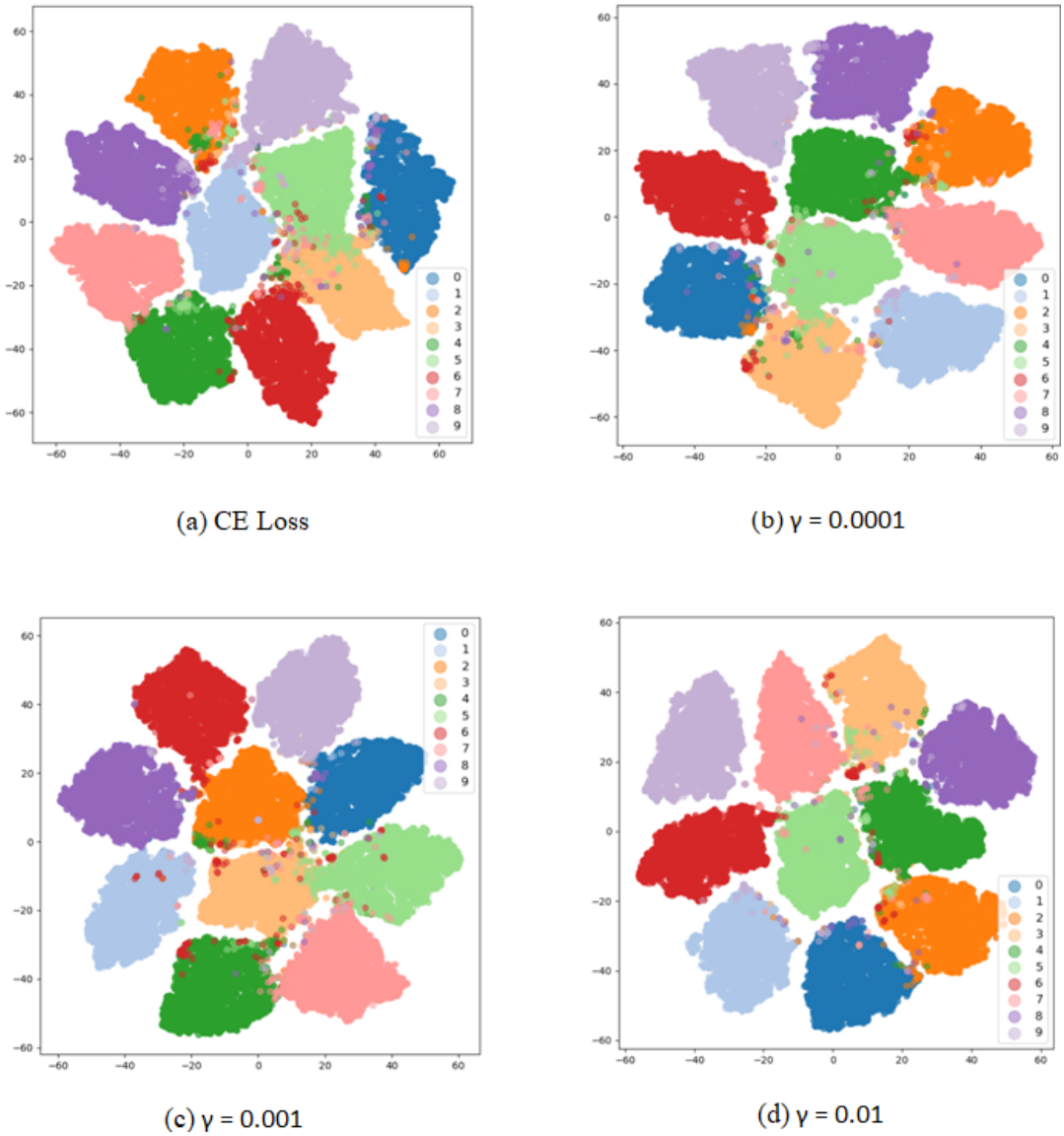
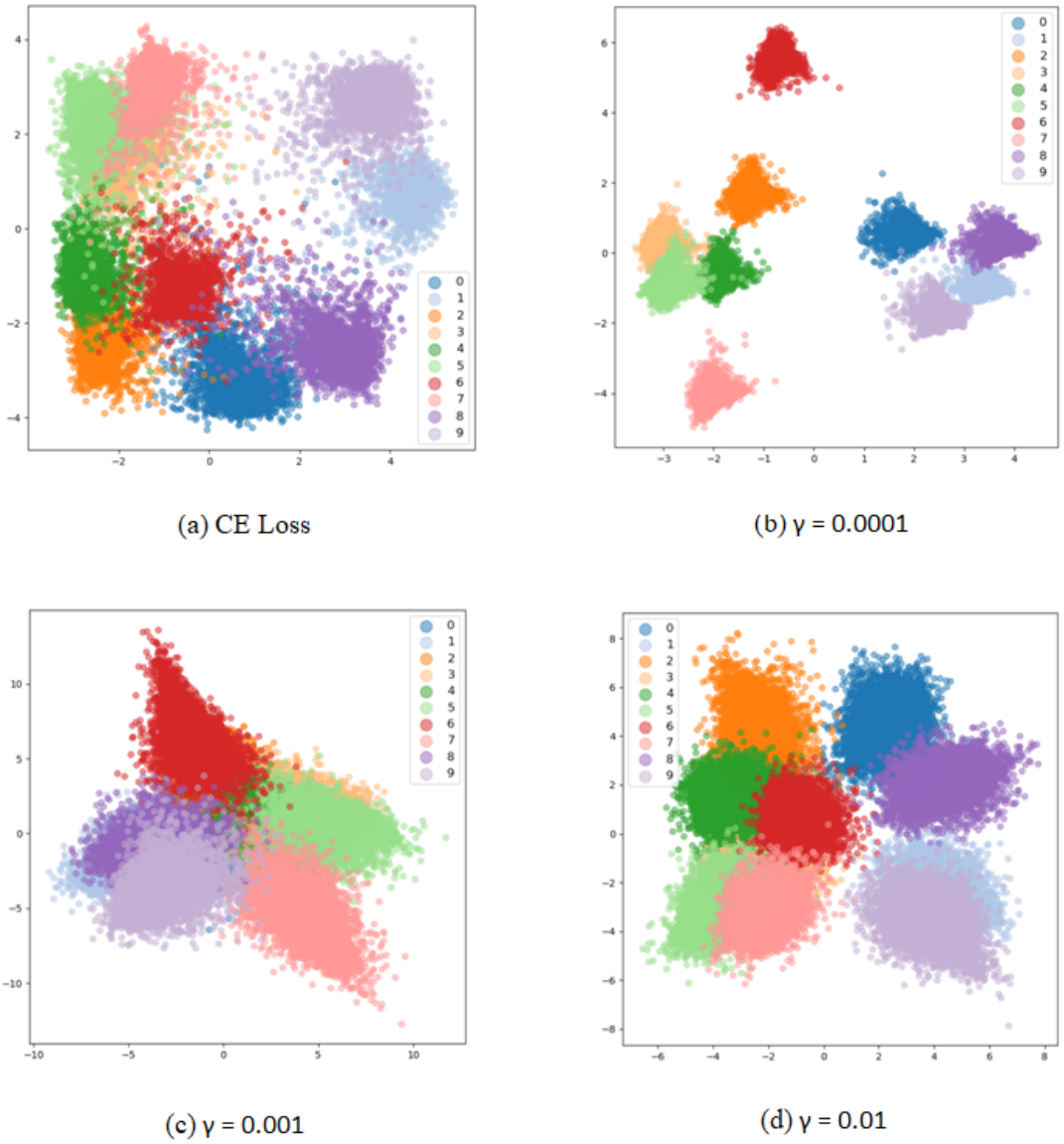


FIGURE 4.18: Random Resize Crop and Random Horizontal Flip: 2-dimensional T-SNE visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture



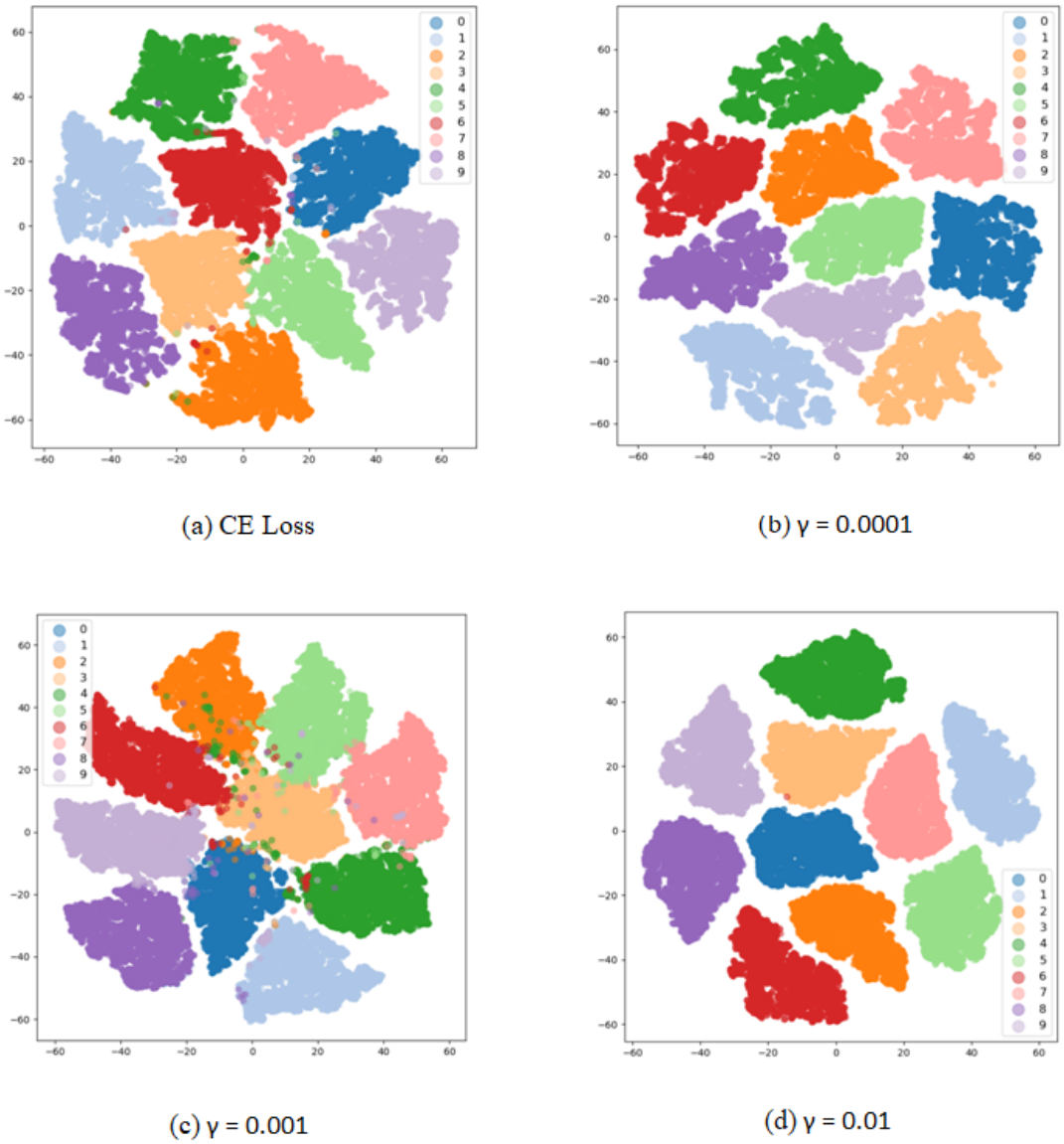
(a) CE Loss

(b) γ = 0.0001

(c) γ = 0.001

(d) γ = 0.01

FIGURE 4.19: Random Gray Scale and Random Solarize: 2-dimensional PCA visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture



(a) CE Loss (b) γ = 0.0001

(c) γ = 0.001 (d) γ = 0.01

FIGURE 4.20: Random Gray Scale and Random Solarize: 2-dimensional T-SNE visualizations of augmented train embeddings from CIFAR-10 dataset with ResNet-18 architecture

Chapter 5

Conclusion

We propose a method of supervised learning for Convolutional Neural Network with Barlow Twins to learn useful features that are invariant to distortions applied to the input data. This method uses a cross-entropy loss in a siamese setting, where the Barlow Twins function is used as an additional loss term with a weighting hyper-parameter γ, and can recover better learnable features from the transformed input image samples. Through our experiments, We show that the proposed method makes it possible to achieve higher classification accuracy and learns a better feature space (proven by visualization of the features) by tuning the weighting hyper-parameter. The proposed method achieves better results in classification accuracy than the baseline method, which uses a cross-entropy loss with augmented input.

Bibliography

[1] Mathilde Caron et al. "Deep Clustering for Unsupervised Learning of Visual Features". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018, pp. 139–156.

[2] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1597–1607. URL: https://proceedings.mlr.press/v119/chen20j.html.

[3] Ting Chen et al. "Big Self-Supervised Models are Strong Semi-Supervised Learners". In: *ArXiv* abs/2006.10029 (2020).

[4] Xinlei Chen et al. "Improved Baselines with Momentum Contrastive Learning". In: *CoRR* abs/2003.04297 (2020). arXiv: 2003.04297. URL: https://arxiv.org/abs/2003.04297.

[5] Adam Coates, Andrew Ng, and Honglak Lee. "An Analysis of Single-Layer Networks in Unsupervised Feature Learning". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, 2011, pp. 215–223. URL: https://proceedings.mlr.press/v15/coates11a.html.

[6] Jean-Bastien Grill et al. "Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 21271–21284.

[7] Kaiming He et al. "Momentum Contrast for Unsupervised Visual Representation Learning". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 9726–9735.

[8] Junnan Li et al. "Prototypical Contrastive Learning of Unsupervised Representations". In: *ArXiv* abs/2005.04966 (2021).

[9] Ishan Misra and Laurens van der Maaten. "Self-Supervised Learning of Pretext-Invariant Representations". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 6706–6716.

[10] Kriti Ohri and Mukesh Kumar. "Review on self-supervised image recognition using deep neural networks". In: *Knowledge-Based Systems* 224 (2021), p. 107090.

[11] Zhirong Wu et al. “Unsupervised Feature Learning via Non-parametric Instance Discrimination”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 3733–3742.

[12] Jure Zbontar et al. “Barlow twins: Self-supervised learning via redundancy reduction”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12310–12320.

[13] Chaoning Zhang et al. “How Does SimSiam Avoid Collapse Without Negative Samples? A Unified Understanding with Self-supervised Contrastive Learning”. In: *CoRR* abs/2203.16262 (2022). DOI: 10.48550/arXiv.2203.16262. arXiv: 2203.16262. URL: https://doi.org/10.48550/arXiv.2203.16262.

[14] Tong Zhang et al. “Leverage Your Local and Global Representations: A New Self-Supervised Learning Strategy”. In: *ArXiv* abs/2203.17205 (2022).