# On Learning Density Aware Embeddings

Soumyadeep Ghosh, Richa Singh, Mayank Vatsa
IIIT-Delhi, India
{soumyadeepg, rsingh, mayank}@iiitd.ac.in

## Abstract

*Deep metric learning algorithms have been utilized to learn discriminative and generalizable models which are effective for classifying unseen classes. In this paper, a novel noise tolerant deep metric learning algorithm is proposed. The proposed method, termed as Density Aware Metric Learning, enforces the model to learn embeddings that are pulled towards the most dense region of the clusters for each class. It is achieved by iteratively shifting the estimate of the center towards the dense region of the cluster thereby leading to faster convergence and higher generalizability. In addition to this, the approach is robust to noisy samples in the training data, often present as outliers. Detailed experiments and analysis on two challenging cross-modal face recognition databases and two popular object recognition databases exhibit the efficacy of the proposed approach. It has superior convergence, requires lesser training time, and yields better accuracies than several popular deep metric learning methods.*

## 1. Introduction

Classification models such as Convolutional Neural Networks (CNN) utilize deep metric learning based loss function for learning discriminative embeddings. The loss function attempts to bring the embeddings of the same classes close to each other in the output manifold. In this embedding space, a direct computation of the distance gives the dissimilarity score between the two images. Several different applications have investigated the use of deep metric learning algorithms such as person re-identification [4, 13, 21], 3D object retrieval [12], biometric recognition [7, 20, 23, 24], robot perception [15], patch matching [11, 31], and object recognition [18, 27].

In the literature, very efficient deep metric learning methods have been proposed such as triplet loss [20] and quadruplet loss [3]. However, a major limitation of these loss functions is their heavy dependence on mining of hard samples for training [13, 20, 21, 30]. In the triplet Loss [20], for $N$ training classes and $K$ samples in each
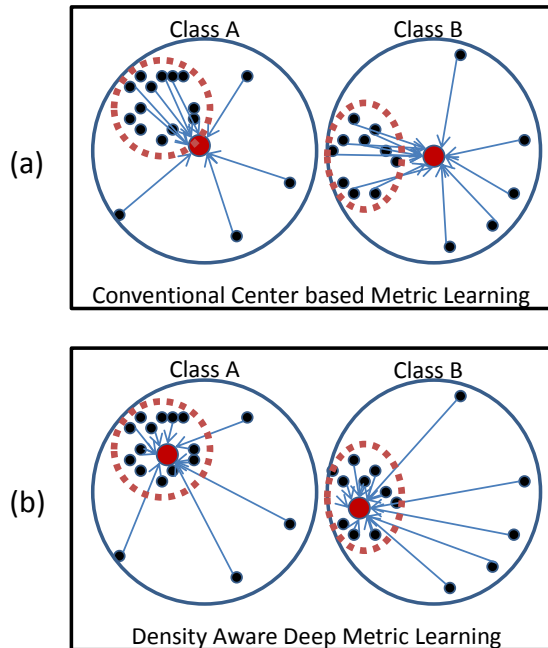


Figure 1: Illustrating the difference in the conventional and proposed metric learning techniques. (a) Conventional center loss based deep metric learning algorithms pull the data of a class towards the centroid of that class. (b) The proposed density aware deep metric learning algorithm pulls the samples of every class towards the most dense region of the respective clusters.

class, the total number of triplets for training can be as high as $N(N-1)K^2(K-1)$, which increases the training time on large datasets significantly. Another limitation of these methods is slow convergence, this heavily depends on the appropriate choice of the training curriculum. Further, the presence of outliers (noisy/poor-quality samples) in the training data, and their participation in triplets may hurt the training process. To the best of our knowledge there has been no study to understand the effect of outliers and density distribution of the training data on the performance of deep metric learning algorithms.

As seen in Figure 1(a), conventional center loss based

deep metric learning methods [12, 26] generate embeddings of each class that lie closer to the centroid of the samples of that particular class. However, they do not take into account the distribution of the training data. In cases where outliers are present, the convergence of such methods on large databases can be slow and the outliers/noisy training samples can adversely affect the training of a discriminative model. In order to mitigate this challenge, the proposed algorithm minimizes the effect of outliers by calculating the center, taking into account the most dense region of the respective clusters for each class (Figure 1(b)). Using the philosophy of the classical mean-shift algorithm [6], the estimate of the mean is shifted to a denser region from the initial estimate of the centroid. This shifted center embedding is used for learning a discriminative model. The research contributions of the paper can be summarized as follows:

- The proposed density aware deep metric learning algorithm provides a generalized framework which can be augmented with any deep metric learning method for effective training especially with noisy data.

- Detailed analysis and comparison with other popular deep metric learning methods on four challenging databases pertaining to face and object images show that the proposed approach gives better recognition accuracies, exhibits superior convergence with reduced training time and is resilient to noisy training data.

## 2. Related Work

Hadsell *et al.* [10] proposed the contrastive loss, which was one of the first deep metric learning methods for training a discriminative model with a deep neural network. They used a single loss function to pull positive pairs and push negative pairs in the output embedding space of the model. This method of training a discriminative neural network, popularly known as the Siamese Network, resulted in several extensions [18, 23, 24] which produced excellent results on a variety of image recognition problems. Recently, one of the most popular methods for deep metric learning is the triplet loss [20]. The triplet loss enforces the model to learn an embedding space where samples of similar classes are mapped closer to each other and that of other classes are pushed away. Wen *et al.* [26] used a combination of the softmax and the center loss for face recognition. Later, Chen *et al.* [3] proposed the quadruplet loss which used an extra negative sample in addition to the anchor, positive and the negative sample that were utilized by the triplet loss. They showed that the extra negative term helps to train a more generalizable model. Thereafter, several methods have attempted to improve upon the triplet and the quadruplet loss based methods. Yuan *et al.* [30] proposed an ensemble based technique for mining hard examples which are used for training a deep network using the

contrastive loss. Hermans *et al.* [13] proposed a triplet mining technique, by selecting the $k$ hardest positive samples and $k$ hardest negative samples for each anchor image in a batch of $N$ randomly sampled images from the training set. Recently, He *et al.* [12] proposed the triplet center loss where the center of the set of anchors and the center of the nearest negative cluster were utilized in the loss function of the triplet loss, for person re-identification.

## 3. Density Aware Metric Learning

The proposed method presents a novel contribution to the deep metric learning paradigm by incorporating the density of data in the clusters during training. Before delving into the detailed formulation, a brief illustration of the background is discussed.

### 3.1. Background

In a classical pattern classification scenario, data $\vec{Z}$ from $n$ different classes is available, $\vec{Z} = \{z_1, z'_1, z_2, ...., z_i, z'_i, ...z_n\}$, where $z_i$ and $z'_i$ are two images of the same class $i$. Let the $i^{th}$ class contain $n_i$ number of training samples. The goal of a deep metric learning algorithm is to learn a function $g_\theta(z) : \mathbb{R}^S \longrightarrow \mathbb{R}^T$ where $S$ is the dimensionality of the source data manifold, $T$ is the dimensionality of the output embedding space of the model $g$, and $\theta$ represents the trainable parameters of the model. For illustration, let $\{x, y\}$ be the pair of points on the embedding manifold of the model $g$. The distance metric function is defined as:

$$D\{x, y\} : \mathbb{R}^T \times \mathbb{R}^T \longrightarrow \mathbb{R} \qquad (1)$$

In this paper, Euclidean distance is used as the distance metric, which can be defined as:

$$D\{x, y\} = \|g_\theta(x) - g_\theta(y)\|_2^2 \qquad (2)$$

Inspired from Large Margin Nearest Neighbor Classification [25], a typical deep metric learning loss $L$ can be used which is minimized by pulling intra-class embeddings together into one cluster and pushing the inter-class embeddings. From the training set $Z$, a 3-tuple is formed using three images, $z_a$, which is a sample of the class $a$, a positive sample $z'_a$, which is another image of the same class $a$, and a negative sample $z_b$ which is an image of another class $b$. The loss function can be expressed as:

$$L = \left[ D\{\vec{Z}_a, \vec{Z}'_a\} - D\{\vec{Z}_a, \vec{Z}_b\} + \alpha \right]_+ \qquad (3)$$

$$\forall (\vec{Z}_a, \vec{Z}'_a, \vec{Z}_b) \in \tau$$

where, $\tau$ is the set of all 3-tuples in the training data, $[f]_+ = max(f, 0)$, $\alpha$ is the margin parameter and $\vec{Z}_a$, $\vec{Z}'_a$ and $\vec{Z}_b$ are sets of all the anchors, positive and negative samples, prepared from the training set.
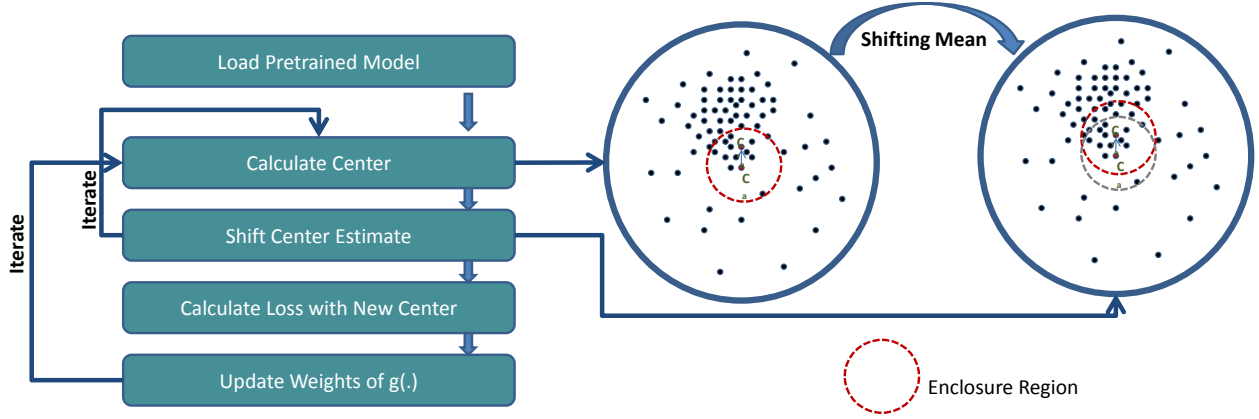
Figure 2: The proposed algorithm iteratively finds the estimate for the center in the most dense region of the cluster. This center, when used with a deep metric learning algorithm, is expected to provide effective training and better convergence. (best viewed in color).

## 3.2. Proposed Formulation

In order to present the proposed approach, the standard loss metric (Equation 3) is re-formulated where the anchor $z_a$ is replaced with the center of class $a$. The center embedding $C_a$ is calculated as the mean of all the embeddings of class $a$. Thus, the loss function may be expressed as:

$$L = \left[ D\{C_a, \vec{Z'}_a\} - D\{C_a, \vec{Z}_b\} + \alpha \right]_+ \quad (4)$$

$$where \quad C_a = \frac{\sum_{n_a} g(z_a)}{n_a}$$

$C_a$ represents the centroid of the cluster corresponding to class $a$, containing $n_a$ training samples. However, depending on the density of the cluster (as shown in Figure 2), the mean-shift algorithm [6] may be applied to iteratively arrive at the mean driven by non-parametric density estimation of the cluster.

### 3.2.1 Shifting the Mean to a Denser Region

Reiterating, $C_a$ is the initial centroid of the cluster which is calculated by taking the mean of all the embeddings of class $a$. Now, from $C_a$, selecting the nearest $p$ points (in the embedding manifold of the model $g$) out of all the $n_a$ points in the cluster corresponding to class $a$, we take the mean of only these $p$ points, where $p < n_a$. We term the region of the embedding manifold containing these $p$ points around the centroid (the red dotted circle in Figure 2) as the **enclosure region**, and the set of these $p$ points as **enclosure points**. The estimate for the new center $C'_a$ can be calculated as,

$$C'_a = \frac{\sum_{i=1}^{p} g(z_a^i)}{p} \quad \forall z_a^i \in \{z_a^1, z_a^2...z_a^p\} \quad (5)$$

where, $z_a^i$ is the $i^{th}$ point inside the enclosure region. Figure 2 shows the new mean $C'_a$, which is expected to be in a denser region of the cluster. The difference of the new mean $C'_a$ and the old mean $C_a$ gives the mean shift vector which can be expressed as:

$$V_a = \left[ \frac{\sum_{i=1}^{p} g(z_a^i)}{p} - \frac{\sum_{n_a} g(z_a)}{n_a} \right] \quad (6)$$

This process is repeated iteratively until the mean shift is negligible, thus leading to convergence.

### 3.2.2 Weighted Mean Shift

The above calculation of the centroid does not take into account any weightage of the points around the mean that are considered. In order to give importance to the points nearer to the centroid, we can use a weight coefficient $W_i$ for every point $i$ in the *enclosure region*. The $k^{th}$ estimate of the center with respect to weights $W_i$ can be calculated as,

$$C_a^k = \frac{\sum_{i=1}^{p} W_i C_a^{k-1} g(z_a^i)}{\sum_{i=1}^{p} W_i C_a^{k-1}} \quad \forall z_a^i \in \{z_a^1, z_a^2...z_a^p\} \quad (7)$$

$C_a^{k-1}$ being the $(k-1)^{th}$ estimate of the mean. The corresponding mean shift vector may be expressed as,

$$V_a^k = \left[ \frac{\sum_{i=1}^{p} W_i C_a^{k-1} g(z_a^i)}{\sum_{i=1}^{p} W_i C_a^{k-1}} - C_a^{k-1} \right] \quad (8)$$

Here, $p$ is the number of *enclosure points* for the $k^{th}$ iteration, and $z_a^i$ is the $i^{th}$ data point for the class $a$.

### 3.3. Selecting weights using a Kernel Density Estimate (KDE)

In order to select weights $W_i$ for each point $i$ in the cluster represented by the centroid $C_a$ for a particular class $a$, we can use a kernel density estimate that are generally used by non-parametric density estimation techniques. A uniform kernel for selecting the weights can be expressed as:

$$W_i = \begin{cases} c & if \ \left\| C_a - z_a^i \right\| < f \\ 0 & otherwise \end{cases} \quad (9)$$

where, $\left\| C_a - z_a^i \right\|$ gives the distance of the point $z_a^i$ from the cluster centroid $C_a$ for class $a$. The uniform kernel assigns a weight $c$ to the point $z_a^i$ if it is within the *enclosure region*. The *enclosure region* has a radius of $f$, thus all the points which are at a distance of $f$ or less from the centroid $C_a$ are assigned the same weight $c$. Instead of directly using a parameter for the radius of the *enclosure region*, $p$ nearest *enclosure points* can also be considered out of all the points in the cluster for class $a$. Algorithm 1 outline the steps of the proposed approach using the triplet loss.

## 4. Density Aware Deep Metric Learning in Triplet and Quadruplet Loss

The proposed Density Aware Metric Learning is a generic formulation and can be incorporated into any deep metric learning loss function. Here, we present the formulations of triplet and quadruplet loss based density aware metric learning:

### 4.1. Density Aware Triplet Loss (DATL)

Schroff *et al.* [20] proposed the triplet loss based deep metric learning technique where the loss $L$ is minimized by the same philosophy as discussed in Section 3.1. From the training set $Z$, a triplet is formed using an anchor $z_a$, which is an image of the class $a$, a positive sample $z_a'$, which is another image of the same class $a$, and a negative sample $z_b$ which is an image of another class $b$. The loss function is expressed as,

$$L = \left[ \left\| g(\vec{Z}_a) - g(\vec{Z}'_a) \right\|_2^2 - \left\| g(\vec{Z}_a) - g(\vec{Z}_b) \right\|_2^2 + \alpha \right]_+ \quad (10)$$

$$\forall (\vec{Z}_a, \vec{Z}'_a, \vec{Z}_b) \in \tau$$

where $\vec{Z}_a$, $\vec{Z}'_a$ and $\vec{Z}_b$ are sets of all anchors, positive and negative samples, respectively, and $\tau$ is the set of all triplets in the training data. Using the proposed approach, the anchor is replaced with the center which is iteratively determined ($C_a$ or $C'_a$, and so on) with an appropriate kernel density estimate. The loss function for the Density Aware

---

**Algorithm 1:** Density Aware Triplet Loss.

**Input:** CNN model $g_\theta$, training data $\{\vec{Z}\}$
**Output:** Trained model $g_\theta$
**Parameters:** $e$ (epochs), $\theta$ (parameters of $g$), $m$ (batch size), $k$ (number of batches) $p$ (number of enclosure points) $s$ (mean shift iterations), $t^p$ (threshold for hard positive selection), $t^n$ (threshold for hard negative selection), $f$ (radius of enclosure region)

1 **for** *Epoch=1 to e* **do**
   **Generate Triplets:**
2    **Initialize:** $X = \{\}$ (empty set of selected Hard Triplets)
3    **Initialize:** $Pool = \{\}$ (empty pool of samples)
4    **for** *every class $a = 1$ to $n$* **do**
5       Select b images randomly from class $a$
6       $Pool = Pool \cup$ selected images
   **end**
7    **for** *each image $z_a^i$ of each class $a$ in $Pool$* **do**
8       Select $z_a^i$ as the anchor image
9       **for** *each image $z_l^y$ in Pool such that $z_l^y \neq z_a^i$* **do**
10         if $a = l$ and $D\{z_a^i, z_l^y\} > t^p$ then $X = X \cup z_l^y$
11         if $a \neq l$ and $D\{z_a^i, z_l^y\} < t^n$ then $X = X \cup z_l^y$
      **end**
   **end**
   **Calculate Center:**
12    $C_a = \frac{\sum_{n_a} g(z_a)}{n_a}$
   **Shift Center:**
13    **for** *every class $a$* **do**
14       **for** *k=1 to s* **do**
15         $W_i = K_u(C_a^{k-1} - z_a^i) =$
        $\begin{cases} c & if \ \left\| C_a^{k-1} - z_a^i \right\| < f \\ 0 & otherwise \end{cases}$
16         $C_a^k = \frac{\sum_{i=1}^p W_i C_a^{k-1} g(z_a^i)}{\sum_{i=1}^p W_i C_a^{k-1}} \quad \forall z_a^i \in$
        $\{z_a^1, z_a^2 ... z_a^p\}$
17         $V_a^k = \left[ \frac{\sum_{i=1}^p W_i C_a^{k-1} g(z_a^i)}{\sum_{i=1}^p W_i C_a^{k-1}} - C_a^{k-1} \right]$
      **end**
   **end**
   **Generate embeddings**
18    **for** *every batch of size $m$* **do**
19       Forward pass through $g$ to find
      $g_\theta(Z_a), g_\theta(Z_a'), f_\theta(Z_b)$
      **Calculate loss $L$**
20       $L =$
      $\sum_m \left[ \left\| C_a^s - g(\vec{Z}'_a) \right\|_2^2 - \left\| C_a^s - g(\vec{Z}_b) \right\|_2^2 + \alpha \right]$
      **Calculate gradient**
21       $\triangle W = \nabla_\theta \frac{1}{m} \sum_m L$
22       **Update weights** of $g_\theta$ using $\triangle W$
   **end**
**end**

---

Triplet Loss (DATL) is as follows:

$$L = \left[ \left\| C_a - g(\vec{Z}'_a) \right\|_2^2 - \left\| C_a - g(\vec{Z}_b) \right\|_2^2 + \alpha \right]_+ \quad (11)$$

## 4.2. Density Aware Quadruplet Loss (DAQL)

The triplet loss is extended by Chen *et al.* [3] as the quadruplet loss where a second negative image $z_c$ is introduced. The loss function for the same in the proposed density aware paradigm can be expressed as,

$$L = \left[ \left\| C_a - g(\vec{Z}'_a) \right\|_2^2 - \left\| C_a - g(\vec{Z}_b) \right\|_2^2 + \alpha_1 \right]_+$$
$$+ \left[ \left\| C_a - g(\vec{Z}'_a) \right\|_2^2 - \left\| C_a - g(\vec{Z}_c) \right\|_2^2 + \alpha_2 \right]_+ \quad (12)$$

$$\forall (\vec{Z}_a, \vec{Z}'_a, \vec{Z}_b, \vec{Z}_c) \in \vartheta$$

where $\vartheta$ is the set of all the quadruplets prepared from the training set.

## 4.3. Experimental Setup and Implementation

The deep CNN architecture by Wu *et al.* [28] is utilized to learn a discriminative model with the proposed loss function. The weights are initialized from a network that is pre-trained on the MS-Celeb 1M dataset. The model has 17 convolutional layers, along with 10 Max-Feature-Map layers. The network has two fully connected layers at the end, producing embeddings of dimensionality 256. Training is performed using the Adam optimizer. The batch size is kept at 60 and the learning rate of $10^{-3}$ is used which is decreased gradually till $10^{-7}$. Hard mining is performed (steps 7-11 of Algorithm 1) for all the variants of triplet and quadruplet losses according to the *Batch Hard* scheme proposed by Hermans *et al.* [13]. The hard mining is only performed at the end of training (once the learning plateaus) to accelerate the training process. All the codes are implemented using the Pytorch platform on a machine with Intel Core i7 CPU, 64GB RAM and NVIDIA GTX 1080Ti GPU.

## 5. Experiments

The proposed algorithm is evaluated on the SCface [8] and FaceSurv [9] datasets for cross-modal face matching, and on the CIFAR10 [14] and STL-10 [5] datasets for object recognition.

## 5.1. Datasets

Details of the databases and experimental protocols are described in this section.

**SCface [8]** is a face dataset containing poor quality face images captured from surveillance cameras in indoor environment. The database contains 4160 images of 130 subjects. The images are captured with eight different cameras, out of which two cameras operated in night-vision mode and one camera is operated in Near-Infrared mode. The images are taken from three different stand-off distances namely 4.2 mts, 2.6 mts, and 1 mt. Out of the 130 subjects, images of

50 subjects are used for training and the remaining are used for testing. The classes/subjects in the train and test set are non overlapping.

**FaceSurv [9]** is a video face database where the subjects walk towards the camera from a distance of about 10 mts. It has 396 daytime and 365 nighttime videos of 240 subjects. The nighttime videos are captured in complete darkness with an NIR illuminator. Each video has about 200 frames on an average. Each subject has three gallery images which are captured in controlled scenarios from a standoff distance of 1 mt. Videos of only 39 subjects are used for training and the remaining are used for testing. The classes/subjects in train and test sets are disjoint.

**CIFAR-10 [14]** is a popular object recognition dataset consisting 60,000 images of 10 classes. The resolution of the images is $32 \times 32$. The training set contains 50,000 images (5,000 images of each class) and 10,000 images (1,000 per class) comprise the testing set.

**STL-10 [5]** contains 113,000 images of 10 different objects. The resolution of the images is $96 \times 96$. The total number of images for training and testing are 5,000 (500 per class) and 8,000 (800 per class), respectively. The remaining images are unlabeled and have not been used for the experiments.

## 5.2. Evaluation Criteria

In this work, two different kinds of experiments have been performed: cross-modal face recognition (identification) and object recognition (retrieval). For face identification, the test set is partitioned into probe (query images) and gallery (reference set/database) sets. For every image of the probe set, matching is performed with each image of the gallery by a forward pass through the learned model ($g_\theta$), followed by computing Euclidean distance between the embeddings of the probe and the gallery images to calculate the match score. Rank $\mathcal{K}$ accuracy is the ratio (multiplied by 100 to get a percentage) of the number of times the correct class is among the top $\mathcal{K}$ matches to the number of matching attempts (once for each probe image). Recall @ $\mathcal{K}$ is the average recall score for all the query images. Following the definition by Song *et al.* [22], the recall score is one if the relevant class is retrieved in the top $\mathcal{K}$ matches with the gallery/database set, and is zero otherwise.

## 6. Results

The experiments have been performed by partitioning each database into the train and test sets. The CNN model is trained on the train set using the proposed density aware deep metric learning, i.e. the Density Aware Triplet Loss (DATL) and the Density Aware Quadruplet Loss (DAQL). Comparisons have been performed with the vanilla triplet and quadruplet losses, (their variants for cross-modal matching are implemented for the SCface and FaceSurv databases). In addition, the proposed algorithm

Table 1: Summarizing the results of face identification on the SCface [8] Database.

| Method | Identification (%) (Rank 1) | | |
|---|---|---|---|
| | 24 x 24 | 32 x 32 | 48 x 48 |
| MDS [2] | 64.87 | 70.48 | 76.14 |
| Co-Transfer Learning [1] | 70.14 | 76.29 | 83.47 |
| Res-Net [26] | 36.30 | 81.80 | 94.30 |
| Coupled Res-Net [17] | 73.30 | 93.50 | 98.00 |
| VGGFace [17] | 41.30 | 75.50 | 88.80 |
| Coupled VGGFace [17] | 62.30 | 91.00 | 94.80 |
| Coupled Light-CNN [17] | 50.50 | 85.00 | 94.00 |
| Triplet loss [16] | 70.69 | 95.42 | 97.02 |
| Quadruplet loss [3] | 74.00 | 96.57 | 98.41 |
| Hard triplet loss [13] | 72.65 | 96.12 | 98.05 |
| Triplet Center Loss [12] | 75.45 | 96.10 | **98.50** |
| Discriminative MDS [29] | 62.70 | 65.50 | 70.70 |
| Proposed DATL | **76.24** | **96.87** | 98.09 |
| Proposed DAQL | **77.25** | **96.58** | 98.14 |

Table 2: Summarizing the results of face identification on the FaceSurv [9] database.

| Method | Identification (%) (Rank 1) | | |
|---|---|---|---|
| | 24 x 24 | 32 x 32 | 48 x 48 |
| Triplet loss [16] | 18.0 | 38.5 | 72.5 |
| Quadruplet loss [3] | 16.4 | 38.5 | 77.9 |
| Hard triplet loss [13] | 17.8 | 40.8 | 78.9 |
| Triplet Center Loss [12] | 18.4 | 41.5 | 82.7 |
| Proposed DATL | **20.4** | **42.8** | **85.9** |
| Proposed DAQL | **21.3** | **48.6** | **85.6** |

Table 3: Summarizing the results of object retrieval on the CIFAR-10 [14] database.

| Method | Recall @ $\mathcal{K}$ (%) | | |
|---|---|---|---|
| | $\mathcal{K} = 1$ | $\mathcal{K} = 10$ | $\mathcal{K} = 100$ |
| Triplet loss [16] | 76.24 | 94.78 | 97.21 |
| Quadruplet loss [3] | 78.35 | 95.40 | **98.99** |
| Siamese+Triplet [15] | 78.62 | 92.57 | 97.19 |
| Hard triplet loss [13] | 78.51 | 93.87 | 97.41 |
| Triplet Center Loss [12] | 79.41 | 96.10 | 95.78 |
| Proposed DATL | **80.34** | **96.68** | 97.84 |
| Proposed DAQL | **80.81** | **96.12** | 97.58 |

Table 4: Summarizing the results of object retrieval on the STL-10 [5] database.

| Method | Recall @ $\mathcal{K}$ (%) | | |
|---|---|---|---|
| | $\mathcal{K} = 1$ | $\mathcal{K} = 10$ | $\mathcal{K} = 100$ |
| Triplet loss [16] | 72.47 | 78.54 | 80.41 |
| Quadruplet loss [3] | 73.98 | 78.71 | 81.77 |
| Siamese+Triplet [15] | 73.62 | 77.15 | 81.34 |
| Hard triplet loss [13] | 72.95 | 76.08 | 81.90 |
| Triplet Center Loss [12] | 74.61 | 77.98 | 81.59 |
| Proposed DATL | **75.27** | **79.08** | **82.38** |
| Proposed DAQL | **75.84** | **80.17** | **83.74** |

results for the SCface database. It outperforms the vanilla triplet and the quadruplet losses and their variants on the FaceSurv database (Table 2) as well. Moreover, on the SC-face database, we report published results from different cross-modal face recognition methods. As shown in Table 1, it can be observed that the proposed algorithm outperforms these existing algorithms, specifically for lower resolution levels.

For the object retrieval task, experiments are performed on the CIFAR-10 and STL-10 datasets. As shown in Table 3, on the CIFAR-10 dataset, the proposed algorithm outperforms both the triplet and the quadruplet losses and their variants for recall @ 1 and recall @ 10. However, for recall @ 100 it produces competitive accuracy with respect to the other algorithms. As shown in Table 4, on the STL-10 dataset, the proposed algorithm outperforms the vanilla triplet and quadruplet losses along with their variants on recall @1, 10 and 100.

# 7. Analysis and Discussion

This section analyzes the performance of the proposed algorithm with respect to training with noisy data, convergence, training time, and parameters.

## 7.1. Effect of Noisy Data during Training

One of the primary properties of the proposed method is the ability to ignore outliers during training. Such outliers may often be represented by noisy data (low resolution/quality and poor illumination). As shown in Figure 3a, these noisy data samples affect the training process of conventional deep metric learning based algorithms. Since the proposed method computes the cluster center only by using the points inside the enclosure region, the outliers are effectively ignored. On the other hand, conventional deep metric learning algorithms would consider all the points (including outliers) which may lead to unnecessary jitter in the convergence during training. An experiment is performed on the STL-10 database by replacing 15% of samples from each class by low resolution variants ($32 \times 32$ and $24 \times 24$ as

is also compared with hard triplet loss [13] and recently proposed triplet center loss [12]. The former is a variant of the vanilla triplet loss using a moderate hard mining approach. Triplet center loss is a formulation which mimics the conventional center based triplet loss previously discussed (Equation 4).

For face recognition, Rank 1 accuracies for three different probe resolutions, namely $48 \times 48$, $32 \times 32$ and $24 \times 24$ are reported. As shown in table 1 on $32 \times 32$ and $24 \times 24$ resolutions, the proposed algorithm produces state-of-the-art
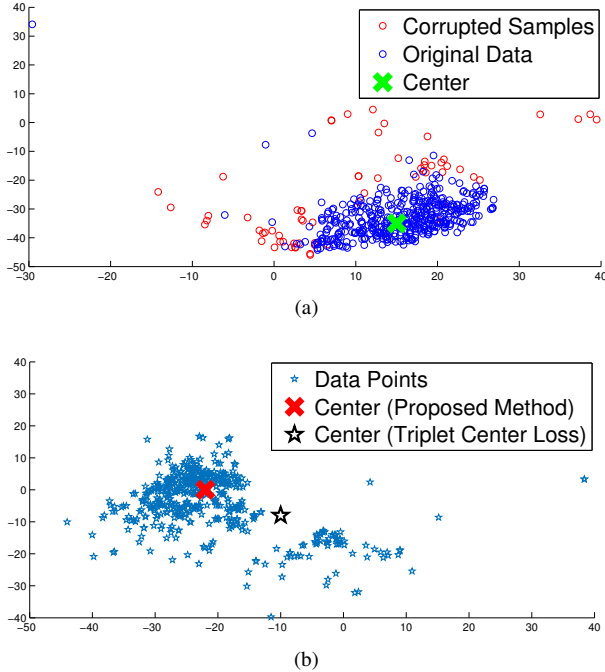
(a)



(b)

Figure 3: (a) tSNE Visualization of noisy samples (for a particular class), which shows that most of the noisy samples are outliers (b) Center computed by the proposed method is in the dense region of the class while the conventional center is away from the dense region. Visualization is on the STL-10 database, one particular class is shown for illustrative brevity (best viewed in color).

two separate experiments) of the same. Such training samples are expected to be outliers and thus may have potential to hurt the training process. We use a no reference image quality score (BRISQUE [19]) (a lower score implies better image quality) for the original training samples of STL-10 which is 33.90 (average for training set). For the noisy samples, the score is 45.14 (for $24 \times 24$) and 41.83 (for $32 \times 32$). This infers that the low resolution data are of lower quality. As shown in Table 5, the proposed methods perform better than conventional deep metric learning techniques when noisy data is introduced in the training process. It also exhibits that performance improvement is greater for the experiment where higher amount of data corruption (adding $24 \times 24$ images) is performed.

## 7.2. Size of the Enclosure Region

One important parameter of the proposed algorithm is the size of the enclosure region. For implementation, the enclosure region is determined by taking the nearest k% points from the current center embedding point. A region of 20% signifies that the nearest 20% points (with respect to all the points of the particular cluster) from the current center are considered to be inside the enclosure region. Figure 4 shows the results for the proposed Density Aware Triplet Loss on

Table 5: Results on the STL-10 database after adding noisy training data for every class.

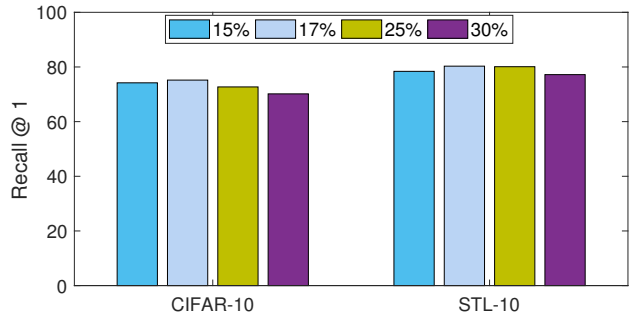| Method | Resolution of Noisy Samples | | | |
| | 24 x 24 | | 32 x 32 | |
| | Recall @ $\mathcal{K}$ (%) | | | |
| | $\mathcal{K}$=1 | $\mathcal{K}$=10 | $\mathcal{K}$=1 | $\mathcal{K}$=10 |
|---|---|---|---|---|
| Triplet Loss [20] | 54.12 | 58.00 | 68.45 | 72.58 |
| Quadruplet Loss [3] | 56.51 | 59.77 | 68.74 | 73.01 |
| Hard Triplet Loss [13] | 58.29 | 60.41 | 65.37 | 72.91 |
| Triplet Center Loss [12] | 62.76 | 65.40 | 68.76 | 73.16 |
| Proposed  DATL | **66.52** | 69.45 | **69.87** | 73.21 |
| Proposed  DAQL | 65.47 | **69.80** | 68.52 | **75.40** |



Figure 4: Effect of the size of the enclosure region on the proposed Density Aware Triplet Loss on the STL-10 and CIFAR-10 databases (best viewed in color).
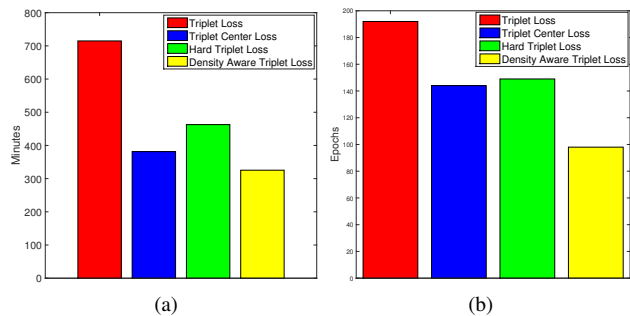


(a)                                    (b)

Figure 5: Total (a) training time and (b) epochs for training on the STL-10 database (best viewed in color).

the STL-10 and CIFAR-10 databases for four different enclosure regions. It can be seen that an enclosure region of 17% yields optimal results while larger or smaller enclosure region results in reduced accuracy on both the databases.

## 7.3. Training Time

Owing to better convergence properties of the proposed DATL, total training time required is much less as compared to the vanilla triplet loss and its variants. As shown
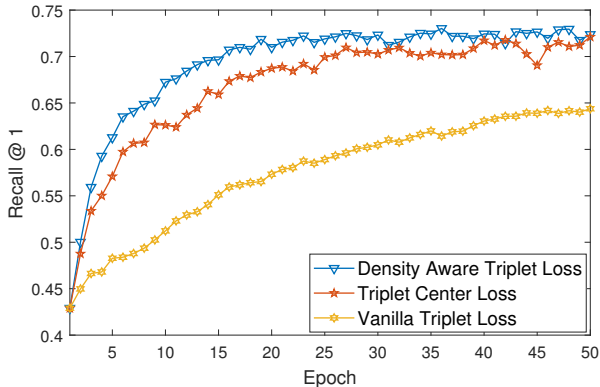
Figure 6: Convergence of the proposed DATL compared with the vanilla triplet loss and triplet center loss on the STL-10 dataset.



Figure 7: Normalized center shift epoch-wise for the CIFAR-10 and SCface database (best viewed in color).

in Figure 5(a), the total time needed to train the proposed algorithm is 325.4 minutes. On the other hand, the vanilla triplet loss, triplet center loss, and the hard triplet loss requires 714.9, 381.4, and 462.7 minutes, respectively on the STL-10 dataset. In terms of the number of epochs as well, the proposed density aware triplet loss requires much lesser number of epochs (98 epochs), whereas the triplet loss, triplet center loss, and the hard triplet loss takes 192, 144, and 149 epochs, respectively.

### 7.4. Convergence

The foremost advantage of the proposed density based deep metric learning approach is its ability to converge quickly as compared to the vanilla triplet and quadruplet loss methods. In addition, the proposed algorithm also converges much faster with respect to the triplet center loss [12] which uses the centroid of the cluster in the loss function (Equation 4). The proposed method avoids outliers and thus updates the weights of the model in such a way, so as to create embeddings in the most dense region of the clusters. This avoids large weight updates as the embeddings need not be shifted away from the dense region. As shown in Figure 6, the convergence of the proposed density aware triplet loss (on a validation set which is prepared by randomly selecting 10% samples from the training set) is superior than the vanilla triplet loss and the triplet center loss. For all the methods, convergence is defined as the stage when the validation accuracy does not improve for 50 epochs at a stretch.

### 7.5. Shifting of the Center

The proposed approach iteratively evaluates the center towards the most dense region of each class (Figure 3b). An analysis is performed showing the magnitude of center shift for each epoch. The average of the center shift for all the classes is used to plot the graph in Figure 7 which shows that the magnitude of the center shift is much higher for the
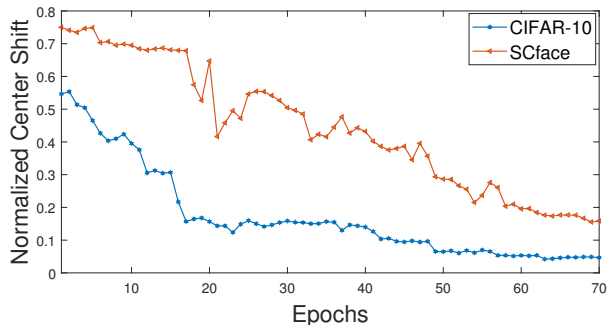
SCface database which has a very large number of noisy training samples. It can be also observed that the magnitude of the center shift decreases as the training progresses, thereby producing more compact clusters.

## 8. Conclusion

The paper presents an elegant approach for density aware deep metric learning. The proposed approach can be augmented with any deep metric learning technique such as triplet and quadruplet loss, and its variants. It results in superior convergence and accuracies, thus providing an important enhancement in current deep metric learning strategies. The proposed DATL and DAQL have also shown to be resilient to noisy training data compared to other deep metric learning methods. Extensive experiments on four datasets showcase the superiority of the proposed DATL and DAQL over existing deep metric learning techniques.

## 9. Acknowledgements

## References

[1] Himanshu S Bhatt, Richa Singh, Mayank Vatsa, and Nalini K Ratha. Improving cross-resolution face matching using ensemble-based co-transfer learning. *IEEE TIP*, 23(12):5654–5669, 2014. 6

[2] Soma Biswas, Gaurav Aggarwal, Patrick J Flynn, and Kevin W Bowyer. Pose-robust recognition of low-resolution face images. *IEEE TPAMI*, 35(12):3037–3049, 2013. 6

[3] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *IEEE CVPR*, 2017. 1, 2, 5, 6, 7

[4] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based CNN with improved triplet loss function. In *IEEE CVPR*, pages 1335–1344, 2016. 1

[5] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pages 215–223, 2011. 5, 6

[6] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI*, 24(5):603–619, 2002. 2, 3

[7] Rishabh Garg, Yashasvi Baweja, Richa Singh, Mayank Vatsa, and Nalini Ratha. Heterogeneity aware deep embedding for mobile periocular recognition. In *IEEE BTAS*, 2018. 1

[8] Mislav Grgic, Kresimir Delac, and Sonja Grgic. Scface–surveillance cameras face database. *Springer MTA*, 51(3):863–879, 2011. 5, 6

[9] Sanchit Gupta, Nikita Gupta, Soumyadeep Ghosh, Maneet Singh, Shruti Nagpal, Richa Singh, and Mayank Vatsa. FaceSurv: A benchmark video dataset for face detection and recognition across spectra and resolutions. In *IEEE FG*, 2019. 5, 6

[10] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE CVPR*, pages 1735–1742, 2006. 2

[11] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *IEEE CVPR*, pages 3279–3286, 2015. 1

[12] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3D object retrieval. In *IEEE CVPR*, 2018. 1, 2, 6, 7, 8

[13] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 1, 2, 5, 6, 7

[14] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 5, 6

[15] BG Kumar, Gustavo Carneiro, Ian Reid, et al. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *IEEE CVPR*, pages 5385–5394, 2016. 1, 6

[16] Xiaoxiang Liu, Lingxiao Song, Xiang Wu, and Tieniu Tan. Transferring deep representation for NIR-VIS heterogeneous face recognition. In *IAPR ICB*, pages 1–8, 2016. 6

[17] Ze Lu, Xudong Jiang, and Alex ChiChung Kot. Deep coupled ResNet for low-resolution face recognition. *IEEE SPL*, pages 2030–2035, 2018. 6

[18] Jonathan Masci, Davide Migliore, Michael M Bronstein, and Jürgen Schmidhuber. Descriptor learning for omnidirectional image matching. In *RRIV*, pages 49–62. Springer, 2014. 1, 2

[19] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE TIP*, 21(12):4695–4708, 2012. 7

[20] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *IEEE CVPR*, pages 815–823, 2015. 1, 2, 4, 7

[21] Hailin Shi, Yang Yang, Xiangyu Zhu, Shengcai Liao, Zhen Lei, Weishi Zheng, and Stan Z Li. Embedding deep metric for person re-identification: A study against large variations. In *ECCV*, pages 732–748, 2016. 1

[22] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *IEEE CVPR*, pages 1014–1023, 2017. 5

[23] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *NIPS*, pages 1988–1996, 2014. 1, 2

[24] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *IEEE CVPR*, pages 1891–1898, 2014. 1, 2

[25] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009. 2

[26] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515, 2016. 2, 6

[27] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3D pose estimation. In *IEEE CVPR*, pages 3109–3118, 2015. 1

[28] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE TPAMI*, 13(11):2884–2896, 2018. 5

[29] Fuwei Yang, Wenming Yang, Riqiang Gao, and Qingmin Liao. Discriminative multidimensional scaling for low-resolution face recognition. *IEEE SPL*, 25(3):388–392, 2018. 6

[30] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *IEEE ICCV*, pages 1539–1547, 2017. 1, 2

[31] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *IEEE CVPR*, pages 4353–4361, 2015. 1