Contrastive Loss based on Contextual Similarity for Image Classification

Lucas Pascotti Valem[1][0000−0002−3833−9072], Daniel Carlos Guimarães Pedronette[1][0000−0002−2867−4838], and Mohand Said Allili[2][0000−0001−8736−6600]

[1] São Paulo State University (UNESP), Rio Claro, SP, Brazil
{lucas.valem,daniel.pedronette}@unesp.br
[2] Université du Quebec en Outaouais, Gatineau, QC, Canada
mohandsaid.allili@uqo.ca

Abstract. Contrastive learning has been extensively exploited in self-supervised and supervised learning due to its effectiveness in learning representations that distinguish between similar and dissimilar images. It offers a robust alternative to cross-entropy by yielding more semantically meaningful image embeddings. However, most contrastive losses rely on pairwise measures to assess the similarity between elements, ignoring more general neighborhood information that can be leveraged to enhance model robustness and generalization. In this paper, we propose the Contextual Contrastive Loss (CCL) to replace pairwise image comparison by introducing a new contextual similarity measure using neighboring elements. The CCL yields a more semantically meaningful image embedding ensuring better separability of classes in the latent space. Experimental evaluation on three datasets (Food101, MiniImageNet, and CIFAR-100) has shown that CCL yields superior results by achieving up to 10.76% relative gains in classification accuracy, particularly for fewer training epochs and limited training data. This demonstrates the potential of our approach, especially in resource-constrained scenarios.

Keywords: Contrastive Learning · Image Classification.

1 Introduction

The advent of advanced technologies for capturing and sharing images has significantly expanded the volume of visual data available [25]. As the volume of data increases, the demand for machine learning approaches capable of leveraging this information becomes indispensable [25]. Machine learning models rely on loss functions, which are essential as they measure prediction errors and guide the learning process. For classification, the cross-entropy loss is the most commonly used metric for training in supervised learning scenarios [10]. The idea behind cross-entropy loss is to quantify the difference between probability distributions. Despite its widespread use, it exhibits limitations, particularly in its ability to generalize effectively to unseen data. It also struggles with issues like class imbalance, noisy labels [29,23], and the potential for poor margins [6,13].

Metric learning and contrastive learning are proposed as solutions to the limitations of cross-entropy loss by focusing on learning effective feature representations that emphasize the relationships and distances between data points, rather than merely categorizing individual examples [10,3]. Metric learning focuses on learning a distance function over pairs of objects. This distance function aims to quantify how similar or dissimilar these objects are to each other. The primary goal is to ensure that similar objects are closer together while dissimilar objects are farther apart in the learned metric space [10,3].

One of the most well-known methods for self-supervised contrastive learning is the Simultaneous Contrastive Learning of Representations [3] (SimCLR), which is a pioneer in the field. However, since it does not consider labeled data, the Supervised Contrastive Learning [10] (SupCon) was proposed, which can be seen as a supervised version of SimCLR. Although significant progress has been made with contrastive losses, these methods rely solely on comparing the similarity between pairs of embeddings, ignoring contextual information.

In this research, the concept of *contextual information* refers to the process of exploiting the neighboring elements of a data sample to compute more semantically meaningful similarity measures. Some works exploit neighborhood analysis for different purposes, showing the relevance of this information in the context of learning. The Simple Siamese (SimSiam) [4] is compared to SimCLR by employing a kNN classifier on their latent features. The adaptive neighborhood metric learning (ANML) [22] identifies and removes inseparable similar and dissimilar samples in the training procedure. There is also an example of application [24] that integrates nearest-neighbor to enhance classification performance through a local-margin triplet loss and local mining strategy. Another approach employs neighborhood information in graphs to regularize learning [9], but without using a contrastive loss. However, few methods directly integrate contextual similarity information into the contrastive loss [30,5,12].

In this work, we propose a novel loss function, the Contextual Contrastive Loss (CCL), based on the supervised contrastive loss [10,3] and contextual information, successfully exploited for image retrieval [19,18]. The proposed CCL improves the learned similarity by taking advantage of contextual neighborhood information for comparing elements during the training process. Among the main contributions, we can mention: *(i)* A novel loss, named Contextual Contrastive Loss (CCL), based on the supervised contrastive loss [10,3] and contextual information by [19,18] is proposed; *(ii)* Different from other methods that demand constant feature updates, ours only requires updates once per epoch, utilizing those created during each iteration, causing no significant overhead during training; *(iii)* The neighborhood sets are computed once and do not need to be recomputed during the training process; *(iv)* A dynamic neighborhood size is proposed to initially enforce the regrouping of larger regions in space, and then progressively focuses on fine-grained regions as the training progresses, which smooths convergence; *(v)* Results reveal superior results compared to the original contrastive loss [10,3] on image classification datasets, especially in cases where there are few labeled data and a smaller number of epochs, which shows the potential of our approach in resource-constrained scenarios.

The remainder of this paper is organized as follows: Section 2 reviews the related work about contrastive learning. Section 3 presents the Contextual Contrastive Loss (CCL), and the workflow used for training and testing. Section 4 presents the experimental results (**a link is added for supplementary material containing extensive results, illustrations, and the source code of our approach**). Finally, Section 5 discusses the conclusions and future work.

2 Related Work

Traditional methods, like cross-entropy loss, focus primarily on achieving correct classifications but may not always encourage the learning of robust, discriminative features that generalize well to new, unseen data, among other issues (e.g., lack of robustness to noisy labels [29,23], possibility of poor margins [6,13]). In light of this, the contrastive losses, that aim to differentiate between similar and dissimilar data points, are a promising solution [3,10]. Many recent works have been using contrastive loss for diverse applications: self-supervised facial expression recognition [21], blind video restoration [15], self-supervised vision transformers [16], and many others [8].

The Simultaneous Contrastive Learning of Representations [3] (SimCLR), a pioneer in the field of self-supervised learning, was proposed for learning visual representations by maximizing the agreement between differently augmented views of the same image through a contrastive loss in the latent space. This method significantly contributed to the field by facilitating the training of more robust and generalizable features without relying on labeled data. However, SimCLR is not capable of exploiting labeled data because the method is entirely unsupervised. Considering this issue, the Supervised Contrastive Learning [10] (SupCon) was proposed to extend the principles of SimCLR by incorporating labels for more discriminative learning in supervised tasks.

Both SimCLR [3] and SupCon [10] leverage pairwise comparisons for effective representation learning. However, this strategy may be limited since it does not consider contextual information [19]. Based on data augmentation, a recent work [1] proposed to enhance document ranking on small datasets of different text document types (news, finance, and science) through supervised contrastive learning. The approach involves augmenting training data by utilizing portions of relevant documents from query-document pairs. This augmented dataset is then used with a supervised contrastive learning objective, differing from traditional pairwise training objectives which did not show improvement with data augmentation.

There are different means of exploiting contextual similarity information, among them: employing graph approaches [28,27,14,20,9], data augmentation [1,7], and using kNN information in parts of the model framework [17,9,24,7]. However, very few incorporate some type of contextual similarity information in the contrastive loss. Some examples are the Nearest-Neighbor Contrastive Learning of Visual Representations [5] (NNCLR), the Contextual Loss [12], and the kNN Contrastive Loss [30]. The NNCLR [5] is unsupervised and based on SimCLR. It introduces a loss function that compares an augmentation not with the original

element, but with the closest neighbor of that element. Besides its contributions, it strictly uses only a single closest neighbor in the comparison, ignoring other elements present in the neighborhood. By contrast, the Contextual Loss [12] improves similarity prediction by counting the number of neighbors two samples have in common.

Despite sharing some similarities with our research, the kNN Contrastive Loss [30] is also distinctly different: *(i)*: It is designed for classification in dialogue systems, specifically considering out-of-domain (OOD) samples, as opposed to image classification; *(ii)*: The kNN Contrastive Loss computes the average contrastive loss for an element and its k neighbors. It iterates for the k neighbors before the contrastive loss logarithmic function. In contrast, our loss formulation is notably different, replacing the similarity function with the square of three components and featuring symmetry; *(iii)*: The methodologies diverge in managing neighborhood lists and features. Our method requires only occasional updates of certain features once per epoch and does not necessitate updating the neighborhood set throughout the training process.

3 Proposed Approach

3.1 Background

Supervised contrastive loss has been proposed in [10], which is an extension of the self-supervised [3] batch contrastive approaches to a fully supervised setting, enabling the model to leverage effectively label information. The general idea involves grouping data samples that belong to the same class closer together in the embedding space while pushing apart groups of samples from different classes. The objective is to enhance the model's ability to distinguish between different classes based on the learned representations (features).

The learning process consists of the use of batches, which contain pairs of images. For each image, two augmentations (i.e., views) are generated. Given a set of N randomly sampled sample/label pairs, $\{\boldsymbol{x}_k, \boldsymbol{y}_k\}_{k=1\ldots N}$, the corresponding batch used for training consists of $2N$ pairs, $\{\tilde{\boldsymbol{x}}_\ell, \tilde{\boldsymbol{y}}_\ell\}_{\ell=1\ldots 2N}$, where $\tilde{\boldsymbol{x}}_{2k}$ and $\tilde{\boldsymbol{x}}_{2k-1}$ are two random augmentations of $\boldsymbol{x}_k$ ($k = 1 \ldots N$) and $\tilde{\boldsymbol{y}}_{2k-1} = \tilde{\boldsymbol{y}}_{2k} = \boldsymbol{y}_k$. Here, we consider only multiviewed batches (size $2N$), which present two augmentations for each image. Let $i \in I \equiv \{1 \ldots 2N\}$ be the index of an arbitrary augmented sample, and let $j(i)$ be the index of the other augmented sample originating from the same source sample. The set of indices of all positives in a batch distinct from i is defined as $P(i) \equiv \{p \in A(i) : \tilde{\boldsymbol{y}}_p = \tilde{\boldsymbol{y}}_i\}$, and $|P(i)|$ is its cardinality. $A(i)$ refers to the set of all elements in the batch except the image i called the anchor.

Based on these definitions, the work of [10] proposes an equation for the supervised contrastive loss (SupCon):

$$\mathcal{L}^{\text{sup}} = \sum_{i \in I} \mathcal{L}_i^{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \sigma\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \sigma\right)} \tag{1}$$

Here, $\boldsymbol{z}_i$ is the embedding generated by the model during the learning process for the data sample i. The index i is called the anchor. The similarity of

embeddings is computed using the dot product operation. The scalar parameter $\sigma \in \mathbb{R}^+$, known as temperature, controls how tightly or loosely the model should group embeddings of the same class versus those of different classes.

3.2 Contextual Contrastive Loss (CCL)

The proposed contextual loss is based on the supervised contrastive loss [10], more specifically the one defined by Eq. 1. Among the various factors that significantly impact the performance of a loss function, the similarity measurement is a crucial one. Accurately measuring the similarity between elements helps to quantify the difference between the predicted values and the actual values.

Contextual Similarity based on Neighborhood Information Pairwise measures have been widely employed in model training [10]. However, they are limited since they often ignore contextual similarity information [19]. The concept of *"contextual information"* is overly used in the literature with different meanings. In this work, it is used to describe the process of utilizing the closest neighboring elements of a given item to calculate a more semantically meaningful similarity metric.

To formalize our approach, let $\mathcal{C} = \{img_1, img_2, \ldots, img_n\}$ be an image collection. Let $\boldsymbol{z}_i$ denote an embedding for the image img_i in a metric space $\mathbb{R}^m$, where m is the size of the embedding (number of dimensions). Originally, the similarity between two elements of indexes i and j is computed using the dot product of their embeddings [10] denoted by $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$. This operation is equivalent to cosine similarity if both embeddings are normalized.

We formulate the contextual similarity between images by first defining sim: $\mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ as the cosine similarity $\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ between images img_i and img_j. Based on the comparison between embeddings, an ordered list of nearest neighbors can be computed for a given anchor $img_i \in \mathcal{C}$. The set of the k nearest neighbors (kNN) of img_i, denoted by $NN_k(img_i)$, contains the k most similar images to img_i in the collection $\mathcal{C}$. Let $|NN_k(x_i)| = k$, where $|\cdot|$ denotes the cardinality of the set. For every $x_j \in NN_k(x_i)$ and every $x_l \notin NN_k(x_i)$, it holds that $d(x_i, x_j) \leq d(x_i, x_l)$. Additionally, we define $NN_k^{\mathcal{Y}}(img_i)$ as the subset of $NN_k(img_i)$ where each image belongs to the same class $\mathcal{Y}$ as img_i. This subset can be expressed as: $NN_k^{\mathcal{Y}}(img_i) = \{x \in NN_k(img_i) \mid \text{class}(x) = \mathcal{Y}\}$. This definition ensures that $NN_k^{\mathcal{Y}}(img_i)$ exclusively contains images from class $\mathcal{Y}$. We define the contextual similarity measure as:

$$\text{sim}_{\text{ctx}}(\boldsymbol{z}_p, \boldsymbol{z}_i, k) = \frac{1}{|NN_k^{\mathcal{Y}}(i)|} \times \sum_{j \in NN_k(i)} \text{sim}(\boldsymbol{z}_p, \boldsymbol{z}_j), \tag{2}$$

where $\boldsymbol{z}_p$ and $\boldsymbol{z}_i$ are the embeddings being compared and $k \in \mathbb{R}^+$ is a scalar value that defines the neighborhood size. The function *sim* is the dot product operation between the two embeddings, defined by $\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_p) = \boldsymbol{z}_i \cdot \boldsymbol{z}_p$. However, the result of sim_{ctx} for the pairs $(\boldsymbol{z}_p, \boldsymbol{z}_i)$ and $(\boldsymbol{z}_i, \boldsymbol{z}_p)$ is not symmetric, which is an important aspect in this scenario. Therefore, to ensure symmetry, we propose to sum the symmetric pairs, each raised to the power of 2:

$$\mathrm{sim}_{\mathrm{ctx}}^{\mathrm{sym}}\left(\boldsymbol{z}_p, \boldsymbol{z}_i, k\right) = \mathrm{sim}_{\mathrm{ctx}}\left(\boldsymbol{z}_p, \boldsymbol{z}_i, k\right)^2 + \mathrm{sim}_{\mathrm{ctx}}\left(\boldsymbol{z}_i, \boldsymbol{z}_p, k\right)^2. \tag{3}$$

The importance of squaring is further discussed in the next subsections.

Dynamic Neighborhood Size The neighborhood size defined by the scalar $k \in \mathbb{Z}^+$, is of fundamental importance in our approach. It defines the number of elements to be considered by the contextual similarity in Eqs. 2 and 3. However, the optimal value of k tends to vary throughout the training process. At the beginning of the training, the model uses larger values of k to contract larger chunks of data in the embedding space. This manifests in larger adjustments of the network weights since each image is pushed toward a considerable number of neighbors. At the end of the training, the model should use smaller neighborhood sizes to ensure smaller adjustments of the model weights and a smoother convergence thereof.

Let k_{start} be the initial value of k for the first epoch, $\epsilon \in \mathbb{Z}^+$ be the current epoch, and ϵ_{total} the total number of epochs to run. The value of k is computed according to a logarithmic decay across epochs, defined as follows: $k = \max\left(1,\ \mathrm{round}((1 - \log_{\epsilon_{total}}(\epsilon)) \cdot k_{\mathrm{start}})\right)$, where round is a function that returns the nearest integer to a given real number.

Proposed Contextual Contrastive Loss (CCL) The length of a vector (a, b, c) in a 3D space is given by $\sqrt{a^2 + b^2 + c^2}$. In our proposal, we use this equation to define the contextual contrastive similarity between i and p as follows: $\mathrm{sim}_{\mathrm{ccl}}\left(\boldsymbol{z}_i, \boldsymbol{z}_p, k\right) = \sqrt{\mathrm{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_p\right)^2 + \mathrm{sim}_{\mathrm{ctx}}\left(\boldsymbol{z}_i, \boldsymbol{z}_p, k\right)^2 + \mathrm{sim}_{\mathrm{ctx}}\left(\boldsymbol{z}_p, \boldsymbol{z}_i, k\right)^2}$. Using all the previous definitions, this can be simplified as:

$$\mathrm{sim}_{\mathrm{ccl}}\left(\boldsymbol{z}_i, \boldsymbol{z}_p, k\right) = \sqrt{\mathrm{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_p\right)^2 + \mathrm{sim}_{\mathrm{ctx}}^{\mathrm{sym}}\left(\boldsymbol{z}_p, \boldsymbol{z}_i, k\right)}, \tag{4}$$

where the result of $\mathrm{sim}_{\mathrm{ccl}}$ is the same for symmetric pairs.

With $\mathrm{sim}_{\mathrm{ccl}}$, the complete equation of our proposed contextual contrastive loss (CCL) is:

$$\mathcal{L}^{\mathrm{ccl}} = \sum_{i \in I} \mathcal{L}_i^{\mathrm{ccl}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\mathrm{sim}_{\mathrm{ccl}}\left(\boldsymbol{z}_i, \boldsymbol{z}_p\right)/\sigma\right)}{\sum_{a \in A(i)} \exp\left(\mathrm{sim}_{\mathrm{ccl}}\left(\boldsymbol{z}_i, \boldsymbol{z}_a\right)/\sigma\right)}, \tag{5}$$

where the variable k is omitted for readability proposes.

3.3 Proposed Training Workflow

This section discusses the workflow of the proposed approach and all its steps from training to testing, including how the proposed CCL is used by the metric learning model. Figure 1 presents an overview of the four steps that compose our framework, which is divided into two main categories: *(i)* metric learning: given image data, it learns new embedding representations based on the contrastive

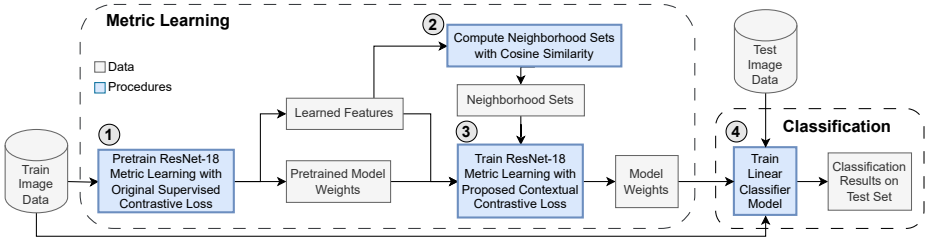

Fig. 1: Workflow of the steps of the proposed approach.

loss; and *(ii)* classification: where a linear model is trained using the binary cross-entropy loss to classify the embeddings according to their classes.

The procedures are marked in blue color and the data, that flows (input/output) between procedures are marked in gray. The steps of the workflow, marked in blue, are the following:

1) Metric Learning Pretraining: A pretraining is conducted using the metric learning model and the original supervised contrastive loss. The weights of this training are later used to generate the neighborhood set and for training the metric learning model in step (3). For a fair comparison, this step is included for both the baseline and ours.

2) Compute Neighborhood Sets: The neighborhood sets are computed based on the features (i.e., embeddings) extracted by the pretrained model. The neighborhood sets are computed according to the formulation in Section 3.2. Our approach is efficient since the neighborhood sets are computed only once and do not need to be recomputed.

3) Train Metric Learning with CCL: The metric learning receives RGB images as input and learns features in a space with $m = 128$ dimensions. The metric learning step uses the proposed CCL for learning more accurate representations. To calculate the similarity with the nearest neighbors, a set of features is considered. This feature set is updated each epoch with the features generated for the batches in every iteration within that epoch. If an image appears more than once, only the most recent feature from it is used to update the feature set.

4) Classification: A linear classification model is trained using the embeddings learned by the metric learning model. This model is used to predict the labels for the test set. The accuracy is computed and reported on the test set.

4 Experimental Evaluation

In this section, we describe the experimental protocol and present both the quantitative and qualitative results obtained. Our proposed CCL loss[1] is frequently compared with SupCon [10], once CCL is based on this approach. Additionally, we include comparisons with SimCLR [3], which, although unsupervised, was also compared to SupCon [10] in its original publication [10].

[1] **Supplementary files and source code:** ccl.lucasvalem.com

To conduct our experiments, we considered three datasets: ***(i)*** **Food101 [2]:** a food categorization dataset with 101 food types, containing 1,000 images each of different resolutions, totaling 101,000 images; ***(ii)*** **MiniImageNet [26]:** a subset of the ImageNet dataset originally proposed for few-shot learning, which contains a balanced set of images across 100 classes, totaling 60,000 images of varied resolutions; ***(iii)*** **CIFAR-100 [11]:** a traditional dataset of 60,000 32x32 color images in 100 classes, with 600 images per class.

Table 1 presents the default hyperparameters used for the metric learning model and linear classifier model. Most of them were adopted according to the Supervised Contrastive Loss (SupCon) implementation. We used the same parameters for CCL and SupCon loss to ensure a fair and consistent comparison. The parameters specific to our approach are marked with a star symbol*.

Table 1: Neural network architecture and default hyperparameters used.

Parameter	Metric Learning	Downstream Classifier
Architecture	ResNet-18	Linear Classifier
Loss Function	Contrastive	Cross-entropy
Batch Size	128	128
Epochs (ϵ)	100	20
Pre-Training Epochs*	10	—
Neighborhood Size (k)*	70	—
Image Resolution	Augmented $32x32$ Crop	Resized to $64x64$
Temperature (σ)	0.1	—
Output Feature Size (m)	128	—
Learning Rate	0.5	5
Cosine Learning Rate Decay	True	True
Learning Rate Warmup	True	True
Weight Decay	10^{-4}	0
Momentum	0.9	0.9
Optimizer	SGD	SGD

Among the parameters, experiments were conducted to evaluate two crucial ones: the batch size and the neighborhood size (k). These experiments were performed on the Food101 dataset, which is the largest one, with a random split of 20% of images for training. Batch size plays a crucial role in contrastive learning, which hinges on comparing different data samples to learn distinctive features. A larger batch size provides more diverse sample pairs, enhancing the model's ability to generalize and distinguish between features. However, it must be carefully chosen to balance the quality of the learned representations. Table 2 presents the accuracy for different batch sizes for both SupCon [10] and CCL. Notably, there is a significant increase in accuracy when the batch size changes from 64 to 128; beyond this point, the accuracy begins to stabilize. Also, our CCL presented gains in all cases. These results are plotted in Figure 2, where the dashed line indicates the default batch chosen (i.e., 128).

Table 3 presents the analysis of the parameter k. It is observed that $k = 70$ is the best setting in most cases. However, the variation in results across different k values is small, suggesting that CCL is robust to different choices of k. Also, for 300 epochs, an even smaller k can be considered. Therefore, we adopted $k = 70$ for all cases and $k = 30$ for 300 epochs in the remaining experiments.

With all the parameters and protocol set, an evaluation was conducted with various training splits (20%, 40%, 60%, and 80%) to assess the robustness of

Table 2: Impact of batch size on accuracy (%) for Food101 dataset (20% training split).

Batch Analysis: Acc. (%) on Food101			
Batch Size	**SupCon [10]**	**CCL (ours)**	**Relative Gain**
64	42.07	44.02	**+4.635%**
128	49.05	53.34	**+8.746%**
192	51.33	54.66	**+6.487%**
256	52.41	52.87	**+0.878%**
Avg.	48.71	51.22	**+5.190%**

Table 3: Impact of parameter k (neighborhood size) on accuracy (%). Results in gray deviate 0.20 or less from the best value in bold.

k Analysis: Accuracies (%) on Food101 dataset					
Train Epochs		**SupCon [10]**	**k=30**	**k=50**	**k=70**	**k=90**
20%	**100**	48.32	51.19	53.14	**54.10**	53.78
	200	56.50	58.59	**58.96**	58.80	58.69
	300	58.11	**59.86**	59.40	58.87	58.44
40%	**100**	62.47	64.68	65.65	65.86	**65.95**
	200	67.30	68.27	68.66	**68.72**	68.30
	300	68.02	68.95	**68.97**	68.80	68.59
Average		60.12	61.92	62.46	**62.53**	62.29

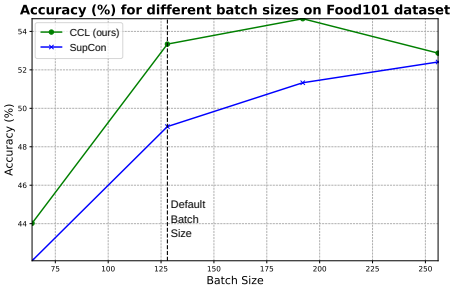

Fig. 2: Accuracy (%) on the test set for different batch sizes.

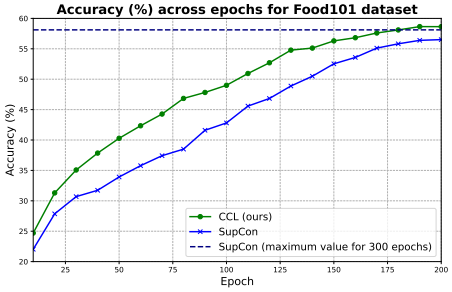

Fig. 3: Test set accuracy (%) across epochs: SupCon vs. CCL (ours).

CCL for 100 training epochs when compared to SimCLR [3] and SupCon [10]. In our protocol, a percentage of the entire dataset is selected for training, while the remaining portion is allocated for testing. For each training percentage, three different splits were randomly generated and used to compare our loss function to others. Table 4 presents the mean accuracy and a 95% confidence interval across the three splits for three evaluated datasets. The results reveal gains in all cases, especially with fewer training data which is a more challenging scenario.

For the Food101 dataset, the most extensive dataset included in our evaluation, we conducted experiments for 100, 200, and 300 epochs. Table 5 shows improvements across all scenarios. These results reveal a significant benefit of our method: it achieves superior performance in situations with limited training data and fewer epochs, which reveals the potential of our method in resource-constrained scenarios. Additionally, CCL with 200 epochs surpasses SupCon with 300 epochs in all cases. To better illustrate the advantages of CCL compared to SupCon [10], Figure 3 displays the accuracies on the test set during training. For 185 epochs, CCL reaches the accuracy that SupCon achieves in 300.

5 Conclusion

In this work, we introduced the Contextual Contrastive Loss (CCL) which leverages the contextual information from neighboring elements to conduct similarity

Table 4: Accuracies (%) achieved for 100 training epochs, comparing the proposed CCL with other contrastive losses, across four training set sizes on three datasets. The relative gains compare CCL with SupCon [10].

Dataset	Loss	Dataset percentages used for (training, testing) (20%, 80%)	(40%, 60%)	(60%, 40%)	(80%, 20%)	Average Values
Food101	SimCLR [3]	31.889 ± 1.974	39.920 ± 0.149	44.246 ± 0.724	47.108 ± 0.937	40.791
	SupCon [10]	48.369 ± 0.515	62.346 ± 0.504	68.649 ± 0.300	71.998 ± 0.459	62.841
	CCL (ours)	53.573 ± 0.347	65.672 ± 0.368	71.074 ± 1.010	73.920 ± 0.935	66.060
	R. Gain	**+10.759%**	**+5.335%**	**+3.532%**	**+2.670%**	**+5.574%**
MiniImageNet	SimCLR [3]	37.909 ± 0.393	48.197 ± 0.244	54.148 ± 1.735	58.427 ± 1.121	49.670
	SupCon [10]	53.466 ± 1.133	67.269 ± 0.537	73.429 ± 0.949	77.454 ± 0.793	67.905
	CCL (ours)	57.231 ± 1.194	69.263 ± 0.104	74.787 ± 0.645	78.217 ± 0.982	69.875
	R. Gain	**+7.042%**	**+2.964%**	**+1.849%**	**+0.985%**	**+3.210%**
CIFAR-100	SimCLR [3]	36.595 ± 2.503	46.018 ± 0.324	51.427 ± 0.426	54.740 ± 1.502	47.195
	SupCon [10]	56.133 ± 1.614	68.089 ± 0.758	73.347 ± 0.545	76.383 ± 0.562	68.488
	CCL (ours)	58.813 ± 0.116	69.748 ± 0.124	74.753 ± 0.496	77.613 ± 1.283	70.232
	R. Gain	**+4.774%**	**+2.437%**	**+1.917%**	**+1.610%**	**+2.685%**
Average Gain		**+7.525%**	**+3.579%**	**+2.433%**	**+1.755%**	**+3.823%**

Table 5: Accuracies (%) achieved on the Food101 dataset when comparing the proposed CCL against SupCon [10], for different training epochs.

Analysis of the number of epochs on the Food101 dataset						
Epochs	**Loss**	**Dataset percentages used for (training, testing) (20%, 80%)**	**(40%, 60%)**	**(60%, 40%)**	**(80%, 20%)**	**Average Values**
100	SupCon [10]	48.369 ± 0.515	62.346 ± 0.504	68.649 ± 0.300	71.998 ± 0.459	62.841
	CCL (ours)	53.573 ± 0.347	65.672 ± 0.368	71.074 ± 1.010	73.920 ± 0.935	66.060
	R. Gain	**+10.759%**	**+5.335%**	**+3.532%**	**+2.670%**	**+5.574%**
200	SupCon [10]	56.116 ± 0.836	67.164 ± 0.656	72.102 ± 1.082	74.787 ± 1.418	67.542
	CCL (ours)	58.392 ± 0.636	68.657 ± 0.324	73.113 ± 0.709	75.748 ± 0.707	68.978
	R. Gain	**+4.056%**	**+2.223%**	**+1.402%**	**+1.285%**	**+2.242%**
300	SupCon [10]	57.981 ± 0.285	68.093 ± 0.345	72.738 ± 1.023	75.498 ± 0.200	68.578
	CCL (ours)	59.589 ± 0.626	69.094 ± 0.228	73.253 ± 1.023	75.691 ± 0.806	69.406
	R. Gain	**+2.773%**	**+1.470%**	**+0.708%**	**+0.256%**	**+1.302%**
Average Gain		**+5.863%**	**+3.009%**	**+1.881%**	**+1.404%**	**+3.039%**

comparisons between images. The experiments demonstrated that our approach achieved significantly improved accuracy compared to the traditional loss in many scenarios, especially in a limited number of labeled data. For future work, we intend to apply the proposed loss for image retrieval by replacing the downstream classification model with a ranking process. We also plan to further expand the proposed CCL for semi-supervised and self-supervised scenarios.

Acknowledgments. The authors are grateful to the São Paulo Research Foundation - FAPESP (grant #2018/15597-6), the Brazilian National Council for Scientific and Technological Development - CNPq (grants #313193/2023-1 and #422667/2021-8), and Petrobras (grant #2023/00095-3) for their financial support.

References

1. Anand, A., Leonhardt, J., Rudra, K., Anand, A.: Supervised contrastive learning approach for contextual ranking. In: Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval. p. 61–71. ICTIR '22, Association for Computing Machinery, New York, NY, USA (2022)
2. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. pp. 446–461. Springer International Publishing, Cham (2014)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20, JMLR.org (2020)
4. Chen, X., He, K.: Exploring simple siamese representation learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15745–15753 (2021)
5. Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A.: With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. IEEE/CVF Int. Conference on Computer Vision (ICCV) pp. 9568–9577 (2021)
6. Elsayed, G., Krishnan, D., Mobahi, H., Regan, K., Bengio, S.: Large margin deep networks for classification. In: Advances in neural information processing systems. pp. 842–852 (2018)
7. Fu, Z., Li, Y., Mao, Z., Wang, Q., Zhang, Y.: Deep metric learning with self-supervised ranking. Proceedings of the AAAI Conference on Artificial Intelligence **35**(2), 1370–1378 (May 2021)
8. Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., Tao, D.: A survey on self-supervised learning: Algorithms, applications, and future trends (2023)
9. Juan, D., Lu, C., Li, Z., Peng, F., Timofeev, A., Chen, Y., Gao, Y., Duerig, T., Tomkins, A., Ravi, S.: Graph-rise: Graph-regularized image semantic embedding. arXiv CoRR **abs/1902.10814** (2019)
10. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 18661–18673. Curran Associates, Inc. (2020)
11. Krizhevsky, A., Nair, V., Hinton, G.E.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009), technical Report TR-2009
12. Liao, C., Tsiligkaridis, T., Kulis, B.: Supervised metric learning to rank for retrieval via contextual similarity optimization. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023)
13. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: Proceedings of the International Conference on Machine Learning (ICML). vol. 2 (2016)
14. Luo, X., Ju, W., Gu, Y., Mao, Z., Liu, L., Yuan, Y., Zhang, M.: Self-supervised graph-level representation learning with adversarial contrastive learning. ACM Trans. Knowl. Discov. Data **18**(2) (nov 2023)
15. Meishvili, G., Djelouah, A., Hattori, S., Schroers, C.: Contrastive learning for controllable blind video restoration. In: 33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022. BMVA Press (2022)
16. Mo, S., Sun, Z., Li, C.: Multi-level contrastive learning for self-supervised vision transformers. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2777–2786 (2023)

17. Mou, Y., He, K., Wang, P., Wu, Y., Wang, J., Wu, W., Xu, W.: Watch the neighbors: A unified k-nearest neighbor contrastive learning framework for OOD intent discovery. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proc. of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 1517–1529. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (2022)
18. Pedronette, D.C.G., da Silva Torres, R.: Exploiting contextual spaces for image re-ranking and rank aggregation. In: Proceedings of the 1st International Conference on Multimedia Retrieval, (ICMR). p. 13. ACM (2011)
19. Pedronette, D.C.G., da Silva Torres, R., Calumby, R.T.: Using contextual spaces for image re-ranking and rank aggregation. Multimedia Tools and Applications **69**(3), 689–716 (2014)
20. Shao, P., Tao, J.: Multi-level graph contrastive learning. Neurocomputing **570**, 127101 (2024)
21. Shu, Y., Gu, X., Yang, G.Z., Lo, B.P.L.: Revisiting self-supervised contrastive learning for facial expression recognition. In: 33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022. BMVA Press (2022)
22. Song, K., Han, J., Cheng, G., Lu, J., Nie, F.: Adaptive neighborhood metric learning. Trans. on Pattern Analysis and Machine Intelligence **44**(9), 4591–4604 (2022)
23. Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., Fergus, R.: Training convolutional networks with noisy labels. arXiv preprint arXiv:1406.2080 (2014)
24. Thammasorn, P., Hippe, D.S., Chaovalitwongse, W.A., Spraker, M., Wootton, L., Nyflot, M., Combs, S., Peeken, J., Ford, E.: Neighborhood watch: Representation learning with local-margin triplet loss and sampling strategy for k-nearest-neighbor image classification. arXiv CoRR **abs/1911.07940** (2019)
25. Tripathi, S., King, C.R.: Contrastive learning: Big data foundations and applications. In: Proc. of the 7th Joint Int. Conference on Data Science & Management of Data. p. 493–497. CODS-COMAD '24, ACM, New York, NY, USA (2024)
26. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., Others: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems. pp. 3630–3638 (2016)
27. Wu, C., Wang, C., Xu, J., Liu, Z., Zheng, K., Wang, X., Song, Y., Gai, K.: Graph contrastive learning with generative adversarial network. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 2721–2730. KDD '23, ACM, New York, NY, USA (2023)
28. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
29. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In: Advances in neural information processing systems. pp. 8778–8788 (2018)
30. Zhou, Y., Liu, P., Qiu, X.: KNN-contrastive learning for out-of-domain intent classification. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 5129–5141. Dublin, Ireland (May 2022)