Received: 10 October 2023 | Revised: 21 February 2024 | Accepted: 24 June 2024

IET Computer Vision

DOI: 10.1049/cvi2.12302

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Centre-loss—A preferred class verification approach over sample-to-sample in self-checkout products datasets

Bernardas Ciapas | **Povilas Treigys**[†]

Institute of Data Science and Digital Technologies, Vilnius University, Vilnius, Lithuania

Correspondence

Bernardas Ciapas.
Email: bernardas.ciapas@mif.vu.lt

Funding information

Ministry of Education, Science and Sports of the Republic of Lithuania, Grant/Award Number: 12-001-01-01-01

Abstract

Siamese networks excel at comparing two images, serving as an effective class verification technique for a single-per-class reference image. However, when multiple reference images are present, Siamese verification necessitates multiple comparisons and aggregation, often unpractical at inference. The Centre-Loss approach, proposed in this research, solves a class verification task more efficiently, using a single forward-pass during inference, than sample-to-sample approaches. Optimising a Centre-Loss function learns class centres and minimises intra-class distances in latent space. The authors compared verification accuracy using Centre-Loss against aggregated Siamese when other hyper-parameters (such as neural network backbone and distance type) are the same. Experiments were performed to contrast the ubiquitous Euclidean against other distance types to discover the optimum Centre-Loss layer, its size, and Centre-Loss weight. In optimal architecture, the Centre-Loss layer is connected to the penultimate layer, calculates Euclidean distance, and its size depends on distance type. The Centre-Loss method was validated on the Self-Checkout products and Fruits 360 image datasets. Centre-Loss comparable accuracy and lesser complexity make it a preferred approach over sample-to-sample for the class verification task, when the number of reference image per class is high and inference speed is a factor, such as in self-checkouts.

KEYWORDS

computer vision, image classification, image matching, image recognition

1 | INTRODUCTION

Self-checkout systems were introduced to cut cashier labour costs in food retail stores, but they have led to new issues: slower product identification without barcodes and increased shoplifting. Studies indicate a 75% rise in theft in stores with self-checkouts compared to those without. Shoplifting occurs through selecting the wrong item, barcode switching, failing to scan items, or leaving without paying. Retailers try to tackle this with security scales, which work for consistent-weight products but not variable-weight ones such as fresh produce. Some use RFID tags on high-value items, but this is costly and impractical for many products, especially unpacked fruits and vegetables.

The process for checking out a product without a barcode involves these steps: (1) The customer selects a product from a menu. (2) The chosen product is placed on a scale for weighing. (3) Once the weight is confirmed, the customer's choice is recorded. (4) The product is then placed on security scales. To enhance this process, a computer vision system is necessary to verify if the product in the image matches the customer's selection after step 3. If there is a high likelihood of a product mismatch, an attendant is alerted to visually confirm the product.

There is a scarcity of representative datasets for real-world self-checkout barcodeless products. In contrast to synthetic counterparts, authentic self-checkout datasets comprise items enclosed in plastic bags, objects partially occluded by

[†]These authors contributed equally to this work.

anatomical features, and variable lighting conditions. It is imperative to underscore that only computer vision solutions validated against authentic self-checkout datasets apply to real-world self-checkout environments. Of all the publicly accessible image sets, Fruits 360 [1], although synthetic, is the closest match to a self-checkout barcodeless products image set. It contains 65 K images within 95 categories. Figure 1 illustrates sample images from our self-checkout dataset compared to Fruits 360 [1]. Several publicly accessible image collections feature packaged products typically equipped with barcodes. The "RPC: A Large-Scale Retail Product Dataset" [2] stands out as the most extensive retail dataset available to the public, comprising approximately 83,000 images within 200 categories. This dataset was artificially compiled in a controlled environment using cameras positioned at various degrees. The background is uniform, facilitating easy removal. Another noteworthy dataset is the MVTec Densely Segmented Supermarket Dataset (MVTec D2S), presented in ref. [3], encompassing around 21,000 images across 60 categories—a fraction of the categories found in supermarkets. This dataset features labels at the pixel level, allowing for extensive application of data augmentation techniques. The GroZi-120 image set [4] includes around 12,000 images depicting 120 products, sourced from a combination of web images and actual grocery store shelves. Smaller retail product image sets are also available to the public, such as those presented in refs. [5, 6], containing up to 10,000 images. However, since products with barcodes can be easily identified through barcode scanning, these datasets are less applicable to research focusing on computer vision-based product selection assistance at self-checkouts.

The authors utilised an authentic dataset of retail products collected in a self-checkout environment and removed empty images and images having unsatisfactory product visibility as in ref. [7]. Alternatively, depth estimators [8, 9] could have been used to segment the product area and then remove images having too-small product area. This dataset encompasses 194 diverse food retail items that typically lack barcodes. The dataset was split 64%–16%–20% into training-validation-test subsets. The training subset was balanced to contain approximately 10,000 distinct images per class through data augmentation. All the training, validation, and testing sets excluded out-of-distribution samples (samples of unknown classes). Negative samples, denoted as "Incorrect" selections, were generated by labelling an image with a class other than the correct one. The omission of out-of-distribution samples was due to challenges in their collection. The authors acknowledge the potential value of including out-of-distribution samples in future research endeavours.

The proposed computer vision solution aims to reduce theft through class verification. This task involves two inputs: an image and a claimed class, which must be among the known classes. The image can contain an object of the claimed class, another known class, an unknown class, or no object at all. The verification task's goal is to distinguish the claimed class object from the rest (excluding differentiation between other known classes, unknown classes, and no objects). Unlike verification, classification only uses the image as input, producing a probability vector across known classes. While verification tasks are well-studied in security with human face datasets, research using other datasets is limited.

Different research domains use various strategies to select negative [image; claimed class] pairs. In computer vision safety, the goal is to differentiate real images from those generated by Generative Adversarial Networks (GAN) [10, 11]. Negative samples for this task consist of GAN-generated images. In face verification, classes represent different individuals, with research often using faces from the same dataset paired with other individuals' identities as negative samples [12–18]. In the self-checkout domain, negative samples should include two types of images likely to be incorrectly chosen in a self-checkout picklist menu: (1) barcodeless images in the self-checkout dataset and (2) images of expensive barcode-containing products sold in the same stores. Self-checkout datasets often contain hundreds of images per class. In contrast, face datasets typically have fewer images per identity (Digi-Face 1M [19] has 11, CelebA [20] has 20. Wider Face [21] has nine images-to-identities ratio), sometimes even less in security applications. This research explores verification methods using a self-checkout dataset.

Research in the realm of class verification tasks spans multiple directions. One approach involves employing sample-to-sample comparison using neural networks, such as Siamese. Another avenue explores the learning of class prototypes during training. During inference, sample-to-sample methods assess the image being verified by comparing it to one (or possibly several) images of the same class from the training dataset. In contrast, class-prototype-based methods evaluate the image being verified by comparing it to the learnt prototypes.

Sample-to-sample methods yield pairwise distances but lack aggregation capability. Verification tasks require selecting which training samples to compare during inference and aggregating results across multiple samples. To address this, some researchers model intra-class distance distributions [22] and measure class probability based on distribution parameters. Others use Earth Mover's Distance (EMD) [23, 24] to quantify dissimilarity between the training set's intra-class distance distribution and the distance distribution of the test sample compared to same-class training samples.

Certain class verification applications need to run inference on hardware with limited storage, processing power, and without GPUs, such as retail self-checkout computers. Previous research found that on machines with an Intel iCore3 CPU, the inference for a single image takes about 0.75 s. However, attempting to select random subsets of training samples and varying the number of samples during inference can lead to unpredictable results. Moreover, when using sample-to-sample methods and comparing against multiple training images during inference, the demands on storage space and computation time increase proportionally with the number of training images. Consequently, conducting inference for multiple images on low-powered self-checkout machines makes sample-to-sample methods impractical.

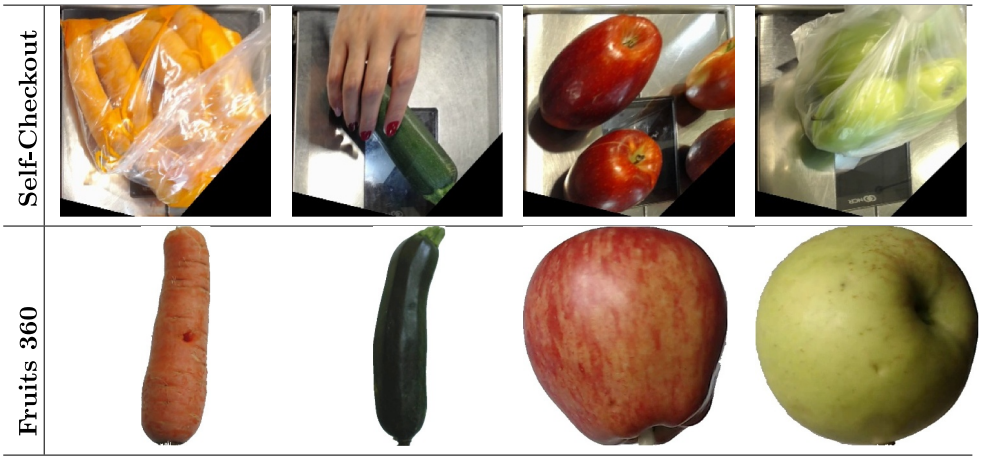


FIGURE 1 Sample images from self-checkout dataset (ours) versus Fruits 360.

Unlike sample-to-sample methods, class-prototype-based approaches compare the image being verified only against the class prototype. This results in significantly faster computation times (1 × 0.75 s compared to N x 0.75 s with sample-to-sample methods, where N is the number of images). However, there is a research gap in comparing the verification accuracy between these two approaches. This study aims to fill that gap by evaluating their accuracy. If the class-prototype-based approach demonstrates comparable or superior accuracy, it becomes the preferred choice for low-powered inference machines.

Many researchers typically follow a two-step process when quantifying image differences: first, they extract higher-level features, and then they calculate the Euclidean distance. However, the choice of distance metric can impact verification results. When measuring the distance between neural network embeddings (i.e. higher-level image features), alternatives such as Cosine and various Minkowski distances (including Manhattan, Euclidean, Chebyshev) are valid options. In contrast, distance types such as Hamming, Jaccard, and Dice are more suitable for comparing binary data. It is worth noting that Minkowski distances are scale-variant, while Cosine distance is not. The p parameter in Minkowski distances allows for flexibility, with special cases such as Manhattan ($p = 1$), Euclidean ($p = 2$), and Chebyshev ($p = +\infty$). Despite the prevalence of Euclidean distance in research, there is a dearth of studies comparing it to other distance metrics in class verification tasks. In this research, the authors aim to investigate how the choice of distance metric between embeddings affects verification accuracy.

Given a classifier with a Softmax function, the authors have optimised a class verifier encompassing a Centre-Loss function. The architecture was validated against an authentic self-checkout dataset as well as a public dataset Fruits 360. The accuracy of the Centre-Loss verifier was compared against that of sample-to-sample verifiers (Siamese and Triplet) using various distance types.

2 | RELATED WORKS

Computer vision classifiers such as [25, 26] and [27] assign probabilities to known classes. However, many real-world applications, such as self-checkout product verification, require identifying out-of-distribution samples, a task classifiers are not designed for. Verification, in contrast, distinguishes between in-distribution and out-of-distribution samples. Efforts to address this issue include [28], which sets a lower threshold for the Top 1 classifier's prediction to consider a sample as in-distribution and [29], which modifies the classifier architecture by adding a confidence branch. However, both approaches may struggle with datasets containing similar classes, such as self-checkout products with multiple similar-looking tomato types, leading to complex probability distribution issues.

In a class verification task, one approach is to treat it as outlier detection, identifying outliers as incorrect [image; claimed-class] pairs. Outlier detectors create a boundary to separate in-distribution from out-of-distribution samples, experimenting with various boundary shapes such as hyperplanes in Support Vector Machines (SVM) [30], ellipsoids in robust covariance models [31], and any shape in isolation forests [32]. In this research, the choice of boundary shape depends on the distance type used: Manhattan leads to a hypercube boundary, Euclidean results in a hypersphere, and Cosine forms a cone. Traditional outlier detection methods do not explicitly minimise intra-class distances, but they enhance separability using techniques such as Gaussian Radial Basis Function (RBF) kernels [33]. This study aims to minimise intra-class distances for all latent layer embeddings, a distinction from typical outlier detection methods.

A class prototype is a generalised representation of a class used in class verification tasks. In ref. [34], they create prototypes based on activations and use Earth Mover's Distance (EMD). However, these prototypes often have high intra-class variation. In Centre-Loss [13], class centres are learnt by averaging class samples in the same embedding space, pushing

embeddings towards them with a two-fold loss function: In addition to Softmax, the other summand Centre-Loss pushes samples towards their respective class centres. We adopt this method in our experiments, extending it to various distance types. Proxy-NCA [35] learns class centres using a loss function that pushes samples not only towards their own but also away from other class centres. It measures cosine distance, which, by definition, loses the scale component. Proxy-NCA's spin-offs SoftTriple [36] uses several single class centres, and Proxy-Anchor [37] trains more efficiently by minimising both sample-to-centre and sample-to-sample distances. Artificial Immune Networks, as in ref. [38], form high-density clusters for each class but do not explicitly minimise cluster size, resulting in expected high intra-class variance. Some researchers use text data to build class prototypes, as in ref. [39]. However, obtaining such data for self-checkout products is challenging.

In class verification research, face identity verification takes the spotlight. The common approach involves comparing image features through sample-to-sample methods. Siamese networks, pioneered by [17], learn to distinguish between pairs of images, labelling them as 0 for the same person and 1 for different individuals. An extension, the Triplet network [12], uses three images: an anchor, a positive (from the same person), and a negative (from a different person). Both Siamese and Triplet networks excel at comparing multiple samples, but they require storing samples or their embeddings in an inference machine. This is feasible with a small number of images per class but becomes impractical in low-powered machines like self-checkouts. The sample-to-sample approach for verification also involves principles of selecting reference images for inference, an area where research is lacking. Considering the proven accuracy of Siamese and Triplet networks, our experiments contrast them with class-prototype-based methods.

While numerous research papers delve into class-prototype-based class verification and sample-to-sample-based verification separately, there is a research gap when it comes to comparing these two approaches. Surprisingly, our investigation did not uncover any articles focused on a verification task that directly compares these two approaches while maintaining identical hyperparameters, including the neural network architecture and dataset.

Class verification research is distinguished by the type of negative samples used. In face verification, negative samples, as in refs. [12–18], typically consist of images associated with a different person's identity. In the context of AI safety, like [40], negative samples are generated by GANs, while positive samples are real images. However, in self-checkout product verification, negative samples should be either images from a different in-distribution class or any out-of-distribution product not in the training data. Unfortunately, there is a shortage of datasets for retail barcodeless products, so we used the same dataset as in refs. [41, 42].

Researchers addressing verification tasks employ various methods to model distributions of class samples. For instance, open set deep networks, as in ref. [22], create a Weibull distribution for each class, enabling varying variance levels among different classes. Similarly, the authors in ref. [43] learn per-class distributions and establishes a fixed Mahalanobis distance from the class centres to determine a sample's class membership. Both of these approaches are more flexible than ours, as they accommodate different variance levels per class. However, none of these studies train models to reduce intra-class variance. We propose that training models that minimise intra-class variance can reduce the necessity to model distinct per-class distributions.

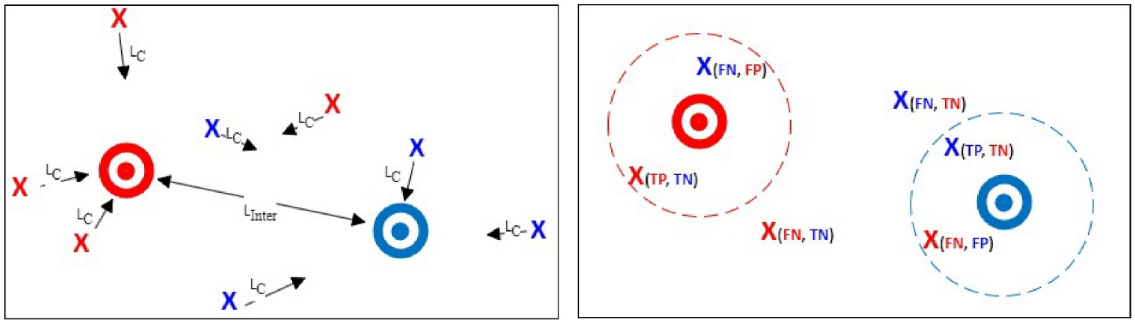
The use of non-Euclidean distance types remains relatively underexplored in research. Some language-focused researchers opt for metrics such as Manhattan, as evident in refs. [44, 45] and [46] or Chebyshev distance, as in ref. [47]. In the realm of computer vision, the authors in ref. [48] conduct a comparison involving Manhattan, Euclidean, and Chebyshev distances using an emotion-labelled dataset. We expand upon their work by including Minkowski distances with varying p values (3 and 4) as well as Cosine distance, conducting experiments on our focus-of-interest self-checkout products dataset. While the authors in ref. [49] examine architectures using Manhattan and Chebyshev distances, it is important to note that their two architectures differ in other aspects, making direct distance type comparison inconclusive. On the other hand, the author in ref. [50] undertakes a comprehensive comparison of various distance types (Manhattan, Euclidean, Minkowski, Chebyshev, and Cosine) in an image retrieval task. We aim to conduct a similar comparison, albeit in a different context—the verification task.

3 | METHODS

This research aimed to assess the effectiveness of class prototype-based verification technique Centre-Loss [13] compared to widely-used sample-to-sample Siamese [17] and Triplet [12]. Additionally, it explored whether alternative distance metrics, beyond Euclidean, could enhance similarity measurement between image embeddings.

Centre-Loss, a departure from traditional sample-to-sample verification methods, focuses on learning virtual class centres (Figure 2). These class centres represent the mean point within the same space as image embeddings for a given neural network layer. The Centre-Loss network's loss function incorporates a distance measurement between an image's embeddings and its respective class centre. Optimising this loss function encourages embeddings to move closer to their corresponding class centres (L_C in Figure 2a). Class centres are continually updated to reflect changes in embeddings due to weight updates, effectively learning class prototypes and bringing image embeddings closer to their respective centres.

During the evaluation, the distance between each data point and every class centre is computed. A maximum-distance-from-centre threshold (dashed line in Figure 2b) is established. Data points are classified as positive (inside the circles) or negative (outside the circles). Correct predictions



(a) Training. (b) Evaluation.

FIGURE 2 Centre-Loss, training and evaluation. The "targets" represent class centres (one per class). The "X"s represent data points (activations of a selected neural network layer). Different classes are represented by different colours. The arrows in (a) represent data point movement upon optimising a loss function summand L_C in Eq. 2 and centre movement—upon optimising L_{Inter} in Eq. 4. The dashed circles around class centres in (b) represent thresholds of verifying a datapoint's belonging to a class. Values in subscript at data points mark their verification predictions and correctness, for example, the left-most point $X_{(TP,TN)}$ is correctly verified as the member of the red class (TP) and as the non-member of the blue class (TN).

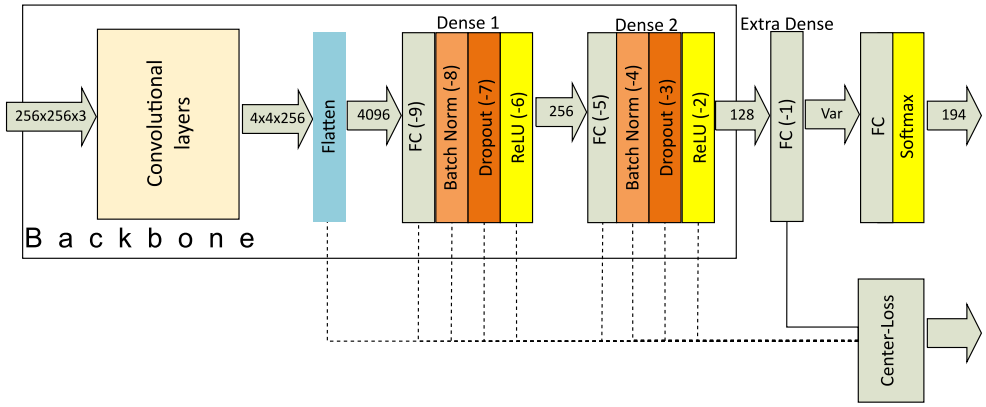


FIGURE 3 Centre-Loss Architecture. Negative numbers in parenthesis signify layer indexes (referred to in pre-Centre-Loss layer experiments). Lines connecting the Centre-Loss layer to its predecessor are marked in one solid line (best result) and nine alternative dashed lines (experiments performed on). The "backbone" refers to the architecture described in detail in ref. [41].

involve data points inside the same-class circle and outside the circle of another class. Receiver Operating Characteristics (ROC) are derived by incrementally adjusting the threshold.

The referenced methods, including Siamese, Triplet, and Centre-Loss, commonly employ Euclidean distance to calculate differences between sample-to-sample embeddings or sample-to-class embeddings. However, as one of the research goals was to compare various distance types, experiments were conducted, and results are presented using not just Euclidean but also other distance metrics: Manhattan, Minkowski, and Cosine. Minkowski distance, which encompasses both Euclidean ($p = 2$) and Manhattan ($p = 1$) distances, offers numerous versions based on different integer values of p. For practical reasons, this research focused on p values within the range [1, 4], considering resource and time constraints. The results section illustrates how varying the p value impacts performance. In contrast, Cosine distance is scale-invariant, limiting its values when measuring differences between data points with unknown scales. Consequently, Cosine distance was included in the research. However, other distance types like Hamming, Jaccard, and Dice were excluded due to their inability to quantify distances for non-binary values.

$$L = L_S + \lambda_1 * L_C \tag{1}$$

where: L - total loss; L_S - cross entropy loss of softmax; λ_1 - centre loss weight; L_C - centre loss.

In the Centre-Loss part of this research, the overall loss function (Eq. 2) comprises two components. The first component, denoted as L_S, utilises the familiar cross-entropy loss with the softmax function. Its role is to prevent all class centres from collapsing into a single point. The second component, L_C (Centre-Loss), imposes penalties on data samples based on their distances from class centres. The absence of cross-entropy loss would likely lead to all data centres collapsing to a single point; the absence of centre-loss leads to a classifier that provides class separability but not sample concentration in the embedding space.

$$L_C = \frac{1}{m}\sum_{i=1}^{m} \|x_i - c_{y_i}\|_p \quad (2)$$

where: L_C - centre loss; m - number of samples; x_i - ith sample's activations of the extra dense layer; y_i - ith sample's label; c_{y_i} - centre of the y_i-th class; $\|\ldots\|_p$ - pth Norm (distance).

$$L_C = \frac{1}{m}\sum_{i=1}^{m}\left(1 - \frac{x_i * c_{y_i}}{\|x_i\| * \|c_{y_i}\|}\right) \quad (3)$$

where: L_C - centre loss; m - number of samples; x_i - ith sample's activations of the extra dense layer; y_i - ith sample's label; c_{y_i} - centre of the y_i-th class.

The Centre-Loss component, denoted as L_C, varies based on the chosen distance type. For Minkowski distance types, including Manhattan ($p = 1$) and Euclidean ($p = 2$), the formula is detailed in Eq. 2. In contrast, for Cosine distance, the formula is specified in Eq. 3. Optimising the Centre-Loss involves two main steps: (1) moving sample embeddings closer to their respective class centres and (2) shifting class centres towards sample embeddings within a mini-batch. The formula for Minkowski-distance-based Centre-Loss (Eq. 2) generalises the approach used in ref. [13] to accommodate any value of the parameter p for Minkowski distance, whereas [13] is limited to $p = 2$ (Euclidean). The Cosine-distance-based Centre-Loss (Eq. 3) computes negative cosine similarity within the range of 0–2.

The weights in the neural network backbones were initiated from the pre-trained classifier on the same dataset. At first, training was performed with no fixed weights. However, a decline in performance during the initial epochs was observed, prompting to consider weight fixation. To make this choice, the authors drew from common practices in transfer learning tasks, where researchers often fix weights in shallower layers while training deeper ones [51, 52].

The Centre-Loss approach defines the loss function but does not define a neural network architecture (such as pre-Centre-Loss layer selection and neuron count in pre-Centre-Loss layer) and requires tuning hyperparameters specific to Centre-Loss (such as centre-loss weight). Tuning the architecture and hyperparameters is described in this section below.

The Centre-Loss architecture has two outputs: Softmax and Centre-Loss that are used to calculate L_S and L_C in Eq. 2, respectively. Initially in ref. [13] connected to the last layer before Softmax (solid line in Figure 3), experiments were conducted to assess if better results could be achieved by connecting the Centre-Loss layer to different shallower layers. Ten experiments were performed, connecting the Centre-Loss layer to various layers beyond the Convolutional backbone (dashed lines in Figure 3).

Optimising the Centre-Loss architecture involved selecting the appropriate pre-Centre-Loss layer size, represented by the Dense 3 block's FC layer in Figure 3. The authors began with a small 2-neuron layer and incrementally doubled its size. They continued this process until either metrics reached saturation or hardware limitations were encountered, with the maximum size being 2048 neurons.

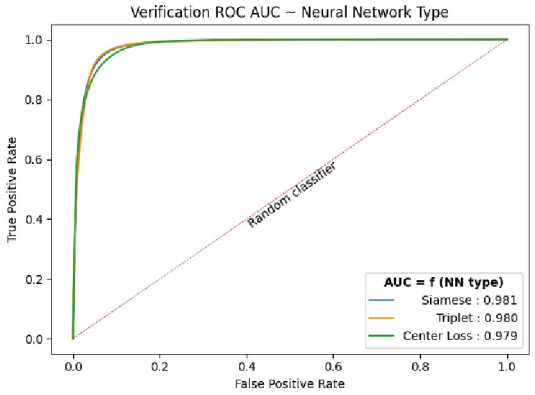


FIGURE 4 Verification ROC by type of the neural network.

TABLE 1 Verification ROC AUC and Accuracy @EER by type of the neural network.

Neural Network	ROC AUC	Accuracy @EER
Siamese	0.981	0.937
Triplet	0.980	0.940
Centre loss	0.979	0.927

The Centre-Loss function relies on a hyperparameter λ_1 in Eq. 2. A low λ_1 value prioritises the softmax's cross-entropy loss, undermining the goal of bringing class embeddings closer to their respective centres. Conversely, a high λ_1 value risks collapsing all class centres into a single point, rendering class differentiation impossible. In the original Centre-Loss paper [13], λ_1 was empirically determined to be 3e-3, with similar results achieved in the range of 1e-4–5e-2. In this research, the authors started within this range and systematically expanded it by a factor of 3.0 until metric saturation was observed.

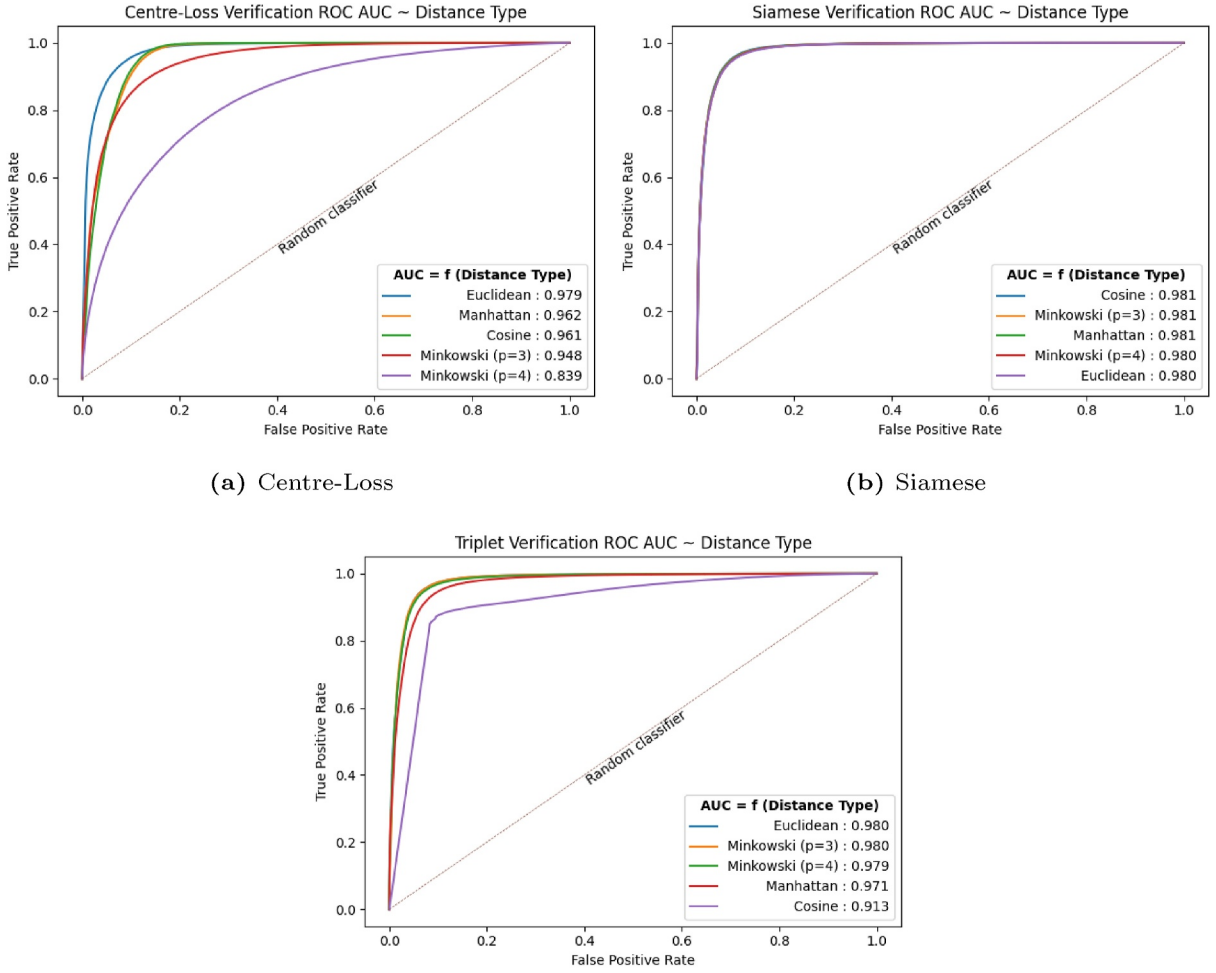
The training complexity of the suggested Centre-Loss approach is $O(MC)$ (where M—the overall number of samples, and C—the number of classes): every sample's distance is calculated to every class' centre. The Siamese' training complexity is $O(M^2/B)$ (where B—number of batches), as every pair of samples is compared, but pairs are limited to the samples within a batch. The Triplet's complexity is $O(M^3/B^2)$, as every sample/anchor is compared to every positive sample and every negative sample. Still, triplets are limited to sample combinations within a batch, and every pass through data places every sample as anchor once. The actual training ranged between 48 and 53 min/epoch for Centre-Loss, 68–83 for Siamese, 125–138 for Triplet networks.

$$L = L_S + \lambda_1 * L_C + \lambda_2 * L_{Inter} \quad (4)$$

TABLE 2 Verification ROC AUC and Accuracy @EER by distance-between-embeddings type.

	ROC AUC			Accuracy @EER		
Distance type	**Centre-Loss**	**Siamese**	**Triplet**	**Centre-Loss**	**Siamese**	**Triplet**
Manhattan	0.962	**0.981**	0.971	0.901	0.937	0.922
Euclidean	**0.979**	0.980	**0.980**	**0.927**	0.935	**0.940**
Minkowski ($p = 3$)	0.948	**0.981**	**0.980**	0.879	0.937	0.939
Minkowski ($p = 4$)	0.839	0.980	0.979	0.759	0.936	0.936
Cosine	0.961	**0.981**	0.913	0.905	**0.938**	0.884

Note: Bold values indicate the best distance type for each method, i.e. the maximum values in each column.



(a) Centre-Loss

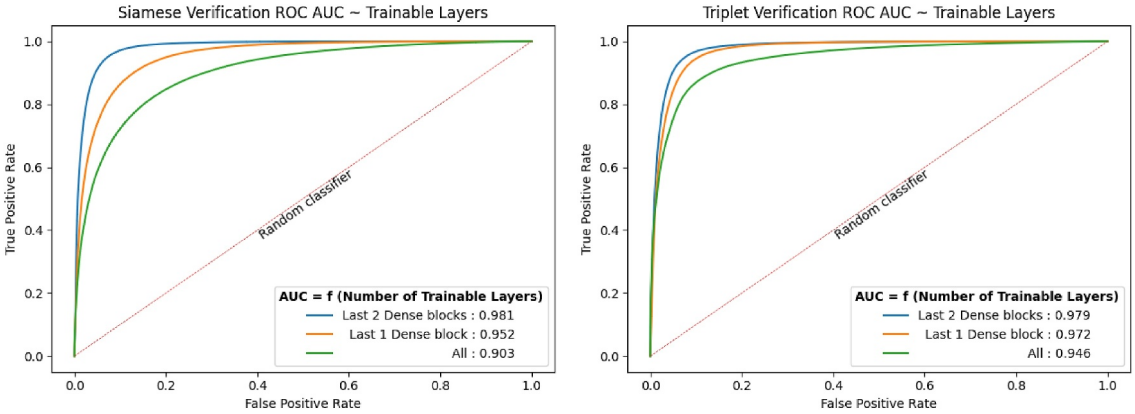
(b) Siamese

(c) Triplet

FIGURE 5 Verification ROC by distance-between-embeddings type.

where: L - total loss; L_S - cross entropy loss of softmax; λ_1 - centre loss weight; L_C - centre loss; λ_2 - inter-centre loss weight; L_{Inter} - inter-centre Loss.

While Centre-Loss aims to minimise distances between samples and their centres, it does not attempt to increase the distance between centres of different classes. This article investigated a possible enhancement to the Centre-Loss loss function to push class centres apart during training (L_{Inter} in Figure 2a), denoted in Eq. 4. The third component, referred to as L_{Inter} (Inter-Centre loss), applies penalties to class centres according to their cosine similarity within the range of 0–2 and is outlined in Eq. 5. The optimisation of this particular loss component drives class centres to move away from each other.



(a) Siamese (b) Triplet

FIGURE 6 Verification ROC by number of trainable layers.

TABLE 3 Verification ROC AUC and Accuracy @EER by the number of trainable layers.

	ROC AUC		Accuracy @EER	
Trainable layers	**Siamese**	**Triplet**	**Siamese**	**Triplet**
Last 2 dense blocks	**0.981**	**0.979**	**0.937**	**0.935**
Last 1 dense block	0.952	0.972	0.884	0.921
All	0.903	0.946	0.825	0.885

Note: Bold values indicate the highest value in the column, which indicates that the row of the bold value is the optimal hyper-parameter.

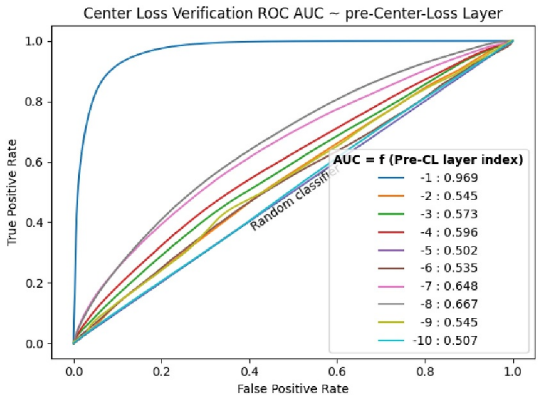


FIGURE 7 Centre-Loss Verification ROC by the pre-Centre Loss layer.

$$L_{Inter} = \frac{1}{m * (n-1)} \sum_{i=1}^{m} \sum_{\substack{j=1 \\ j \neq y_i}}^{n} \left(1 + \frac{c_{y_i} * c_j}{\|c_{y_i}\| * \|c_j\|}\right) \quad (5)$$

where L_{Inter} - inter-centre loss; m - number of samples; n - number of classes; y_i - ith sample's label; c_{y_i}, c_j - centres of the y_i-th, jth class.

The Inter-Centre loss component necessitates a relative weight hyperparameter, denoted as λ_2 in Eq. 4. To determine

TABLE 4 Centre-Loss Verification ROC AUC and Accuracy @EER by the pre-Centre Loss layer.

Pre-centre Loss layer	ROC AUC	Accuracy @ EER
−1	**0.969**	**0.910**
−2	0.545	0.535
−3	0.573	0.551
−4	0.596	0.569
−5	0.502	0.501
−6	0.535	0.530
−7	0.648	0.607
−8	0.667	0.620
−9	0.545	0.533
−10	0.507	0.505

Note: Bold values indicate the highest value in the column, which indicates that the row of the bold value is the optimal hyper-parameter.

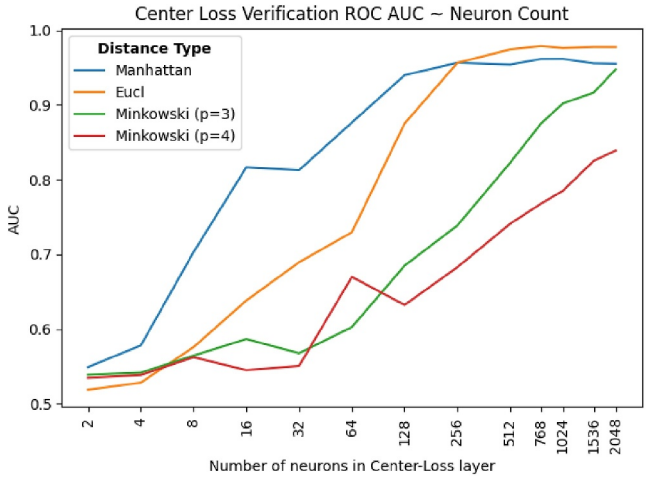
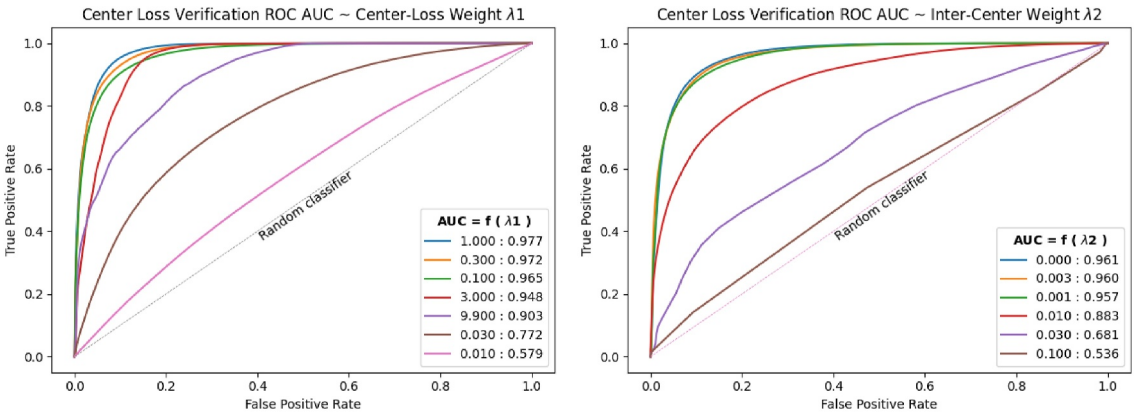


FIGURE 8 Centre-Loss Verification ROC AUC by Neuron Count in Centre-Loss layer.



(a) By Centre-Loss weight $\lambda 1$ **(b)** By Inter-Centre weight $\lambda 2$

FIGURE 9 Centre-Loss Verification ROC by Centre-Loss hyperparameters $\lambda 1$ and $\lambda 2$.

TABLE 5 Centre-Loss Verification by Centre-Loss weight $\lambda 1$.

Centre-loss Weight $\lambda 1$	ROC AUC	Accuracy @ EER
1.000	**0.977**	**0.925**
0.300	0.972	0.914
0.100	0.965	0.902
3.000	0.948	0.883
9.900	0.903	0.805
0.030	0.772	0.702
0.010	0.579	0.556

Note: Bold values indicate the highest value in the column, which indicates that the row of the bold value is the optimal hyper-parameter.

TABLE 6 Centre-Loss Verification by Inter-centre weight $\lambda 2$.

Inter-centre Weight $\lambda 2$	ROC AUC	Accuracy @ EER
0.000	**0.961**	**0.900**
0.003	0.960	0.895
0.001	0.957	0.890
0.010	0.883	0.799
0.030	0.681	0.619
0.100	0.536	0.533

Note: Bold values indicate the highest value in the column, which indicates that the row of the bold value is the optimal hyper-parameter.

its optimal value, the authors initially explored a range of λ_2 values akin to that of λ_1 (as both pertain to distance in the same dimensional space). Notably, the authors observed improved metrics with smaller λ_2 values. They continued to reduce λ_2 by a factor of 3.0 until results showed negligible differences from the case of λ_2 being set to 0.

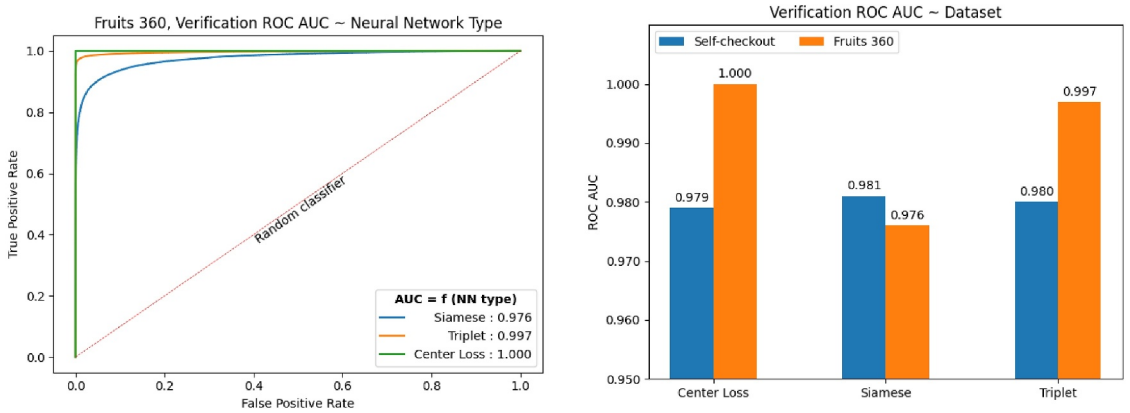
Throughout the experiments, the primary performance metric employed by the authors was the Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC). Furthermore, the authors provide the verification accuracy achieved at the Equal Error Rate (EER), which corresponds to the point on the ROC curve where the False Positive Rate (FPR) matches the False Negative Rate (FNR).

4 | RESULTS

The primary outcome of this study revolves around the performance evaluation of two distinct class verification approaches: sample-to-sample and class-prototype-based methods. To conduct this assessment, the authors employed three different neural network models in the context of barcodeless product verification using a self-checkout dataset: Centre-Loss (representing the class-prototype approach), Siamese, and Triplet (representing the sample-to-sample methods). Figure 4 and Table 1 showcase the Receiver Operating Characteristics (ROC) curve and provides detailed information regarding verification accuracy at the Equal Error Rate (EER) and the Area Under Curve (AUC) for each neural network type[1]. Remarkably, the differences in performance among all the network types are marginal. This suggests that sample-to-sample comparing networks (Siamese and Triplet) do not perform better than a class-prototype-based network (Centre-Loss).

The study investigated the impact of various distance types on accuracy metrics, which measure the dissimilarity between samples (Siamese, Triplet) or between a sample and a class centre (Centre-Loss). Experiments encompassed Manhattan, Euclidean, Minkowski ($p = 3, 4$), and Cosine distance types. Table 2 summarises accuracy results at an equal error rate (EER) and ROC AUC by the distance type, while detailed ROC curves for each distance type are in Figure 5a (Centre-Loss), Figure 5b (Siamese), and Figure 5c (Triplet). All distance types exhibited similar performance for Siamese and Triplet networks, except for a slight degradation observed with the

[1]In this and all the other experiments, the reported results utilise the optimal hyperparameter values, including distance type (for all network types), the number of trainable layers (Siamese and Triplet), pre-Centre-Loss layer index, neuron count in the pre-Centre-Loss layer, λ_1, and λ_2 values (all four for Centre-Loss).



(a) Fruits 360 dataset, verification ROC by type of neural network

(b) Self-Checkout (our) vs. Fruits 360 [1]

FIGURE 10 Comparing verification results on Fruits 360 [1] versus self-checkout (our) dataset.

FIGURE 11 Sample images and their distances from selected class centres.

Cosine distance type in the Triplet network. In Centre-Loss, most distance types (Cosine, Manhattan, Euclidean, and Minkowski) performed similarly, although degradation was noted for higher Minkowski p values ($p = 3, 4$).

Training all layers in Siamese and Triplet architectures led to declining metrics after the first epoch. To address this, the authors conducted experiments, fixing the weights of the last one and the last two dense blocks (see Figure 3) that were initialised from the original classifier. As depicted in Figure 6a (Siamese) and 6b (Triplet), the best ROC AUC was achieved when the last two dense blocks were fixed in both Siamese and Triplet networks. Conversely, performance deteriorated when no weights were fixed or when only the last dense block was fixed. Table 3 summarises Accuracy@EER and ROC AUC by the number of fixed blocks. In contrast, training all layers in the Centre-Loss architecture did not exhibit declining metrics; thus, all layers were trainable.

To optimise the Centre-Loss architecture, the authors conducted a series of experiments. Initially, they compared Centre-Loss architectures with the Centre-Loss layer attached to various layers of the original classifier. Figure 7 and Table 4 provide a summary of ROC AUC and Accuracy@EER for 10 different architectures, with the Centre-Loss layer attached from the last (-first) to the 10th from the end (−10th) layer in the original classifier. The results clearly highlight that the pre-last (-first) layer is optimal for Centre-Loss measurement.

Secondly, the authors empirically determined the optimal number of neurons in the pre-Centre-Loss layer. Figure 8 illustrates the growth and saturation of verification ROC AUC with respect to the neuron count in the pre-Centre-Loss layer. ROC AUC demonstrates a positive correlation with the neuron count for all distance types until it reaches saturation. The saturation point for ROC AUC occurs with a smaller number of neurons when the Minkowski coefficient p is low: Manhattan ($p = 1$) saturates at 256 neurons, Euclidean ($p = 2$) at 768 neurons, while higher p values ($p = 3$ and 4) do not saturate even at 2048 neurons.

The authors conducted experiments to determine the optimal hyperparameter values for Centre-Loss: the Centre-Loss relative weight (λ_1 in Eq.2) and the Inter-Centre Loss relative weight (λ_2 in Eq.2). Figure 9a displays the verification ROC, while Table 5 provides detailed ROC AUC and Accuracy@EER figures for various λ_1 values. The results indicate that λ_1 values between 0.1 and 3.0 produce similar metrics, while values outside this range lead to degraded performance. This underscores the importance of both summands in the Centre-Loss function: the Cross-Entropy summand for class separation and the Centre-Loss summand for reducing intra-class distances. The robust performance across a wide range of λ_1 values suggests that the Centre-Loss function is not highly sensitive to variations within this range. However, including the Inter-Centre summand in the loss function did not yield positive effects. Table 6 and Figure 9b show that the

optimal value for λ_2 is 0; increasing λ_2 led to lower ROC AUC and Accuracy@EER values.

In addition to the authentic self-checkout dataset, the authors also assessed the method's performance using the Fruits 360 dataset [1]: Models of each network type in question (Centre-Loss, Siamese, and Triplet) were trained on Fruits 360 training subset and evaluated on its test subset. The verification ROC curves for Fruits 360 are illustrated in Figure 10a, and a comparison of ROC AUC between Fruits 360 and our self-checkout dataset is presented in Figure 10b. With the exception of Siamese, most methods exhibited improved performance on Fruits 360, primarily due to the dataset's clean and synthetic images.

Figure 11 illustrates a set of sample images alongside their corresponding distances from selected class centres. Correct selections are denoted by green, while red dashes indicate incorrect ones. The blue dashed line represents the equal error rate (EER) threshold, delineating the boundary between correct (below the threshold) and incorrect (above the threshold) selections.

5 | CONCLUSIONS

Almost identical verification accuracy metrics between class-prototype-based (ROC AUC Centre-Loss: 0.979) and sample-to-sample (ROC AUC Siamese: 0.981, Triplet: 0.980) approaches were shown on the retail self-checkout barcodeless products dataset. This fact makes class centre a preferred approach in low-computing-power inference machines.

Authors experimentally showed that using Euclidean distance in loss functions to measure sample-to-sample or class-centre-to-sample distances always results in equal or better accuracy over other distance types (Manhattan, Minkowski, and Cosine), although using other nearby Minkowski p values ($p = 1$ Manhattan, $p = 3$) performs similarly. Using higher Minkowski p values requires more neurons to achieve saturation. In the Centre-Loss approach, using Euclidean distance achieved 0.979 ROC AUC, whereas nearby Minkowski p values resulted in 0.962 ($p = 1$ Manhattan) and 0.948 ($p = 3$).

Experiments with Centre-Loss architecture revealed the penultimate layer as the layer of choice to minimise intra-class distances upon training. The optimum size of the penultimate layer depends on the Minkowski p value.

The value of the Centre-Loss summand was experimentally proven. However, a suggested update in the Centre-Loss function to increase Inter-Class distance did not give a positive result.

The future research spans several directions. First, other class-prototype-based approaches, such as Proxy-NCA, SoftTriple, and Proxy-Anchor, should be applied to verify the product selection on the self-checkout products dataset. Second, the class verification accuracy of class-prototype-based approaches should be compared against sample-to-sample approaches on wider image sets, such as ImageNet.

AUTHOR CONTRIBUTIONS

Bernardas Ciapas: data acquisition, investigation, programming, visualisation, and original draft.

Prof. Povilas Treigys: review and editing, funding acquisition, resources, and supervision.

ACKNOWLEDGEMENTS

The authors are grateful to Vilnius University Mathematics and Informatics department Information Technologies Open Access Centre for providing high-performance computing (HPC) resources used in this research.

Publication/Research funded under the Programme "University Excellence Initiatives" of the Ministry of Education, Science and Sports of the Republic of Lithuania (Measure No. 12-001-01-01-01 "Improving the Research and Study Environment").

CONFLICT OF INTEREST STATEMENT

Authors have no financial or non-financial interests to disclose.

DATA AVAILABILITY STATEMENT

Data available on request from the authors.

ORCID

Bernardas Ciapas https://orcid.org/0000-0002-7156-5995
Povilas Treigys https://orcid.org/0000-0002-6608-5508

REFERENCES

1. Oltean, M.: Fruits 360 dataset: new research directions. URL https://www.researchgate.net/publication/354535752_Fruits_360_dataset_new_research_directions (2021)
2. Wei, X.-S., et al.: RPC: A Large-Scale Retail Product Checkout Dataset (2019). *arXiv preprint arXiv:1901.07249*
3. Follmann, P., et al. (eds.) MVTec D2S: Densely Segmented Supermarket Dataset. *Computer Vision – ECCV*, vol. 2018, pp. 581–597. Springer International Publishing, Cham (2018)
4. Merler, M., Galleguillos, C., Belongie, S.: Recognizing groceries in situ using in vitro training data. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
5. George, M., Floerkemeier, C.: Recognizing products: a per-exemplar multi-label image classification approach. In: European Conference on Computer Vision, 440–455 (2014). Springer. https://doi.org/10.1007/978-3-319-10605-2_29
6. Jund, P., et al.: The Freiburg Groceries Dataset (2016). *arXiv preprint arXiv:1611.05799*
7. Ciapas, B., Treigys, P.: Automated barcodeless product classifier for food retail self-checkout images. Vis. Comput. 1–15 (2023). https://doi.org/10.1007/s00371-023-03163-8
8. Fu, C., et al.: TMSO-Net: texture adaptive multi-scale observation for light field image depth estimation. J. Vis. Commun. Image Represent. 90, 103731 (2023). https://doi.org/10.1016/j.jvcir.2022.103731
9. Fan, W., Yang, L., Bouguila, N.: Unsupervised grouped axial data modeling via hierarchical Bayesian nonparametric models with Watson distributions. IEEE Trans. Pattern Anal. Mach. Intell. 44(12), 9654–9668 (2021). https://doi.org/10.1109/tpami.2021.3128271
10. Huang, X., et al.: Safety verification of deep neural networks. In: International Conference on Computer Aided Verification, pp. 3–29. Springer (2017)
11. Katz, S.M., et al.: Verification of image-based neural network controllers using generative models. J. Aero. Inf. Syst. 19(9), 574–584 (2022). https://doi.org/10.2514/1.i011071

12. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
13. Wen, Y., et al.: A discriminative feature learning approach for deep face recognition. In: European Conference on Computer Vision, pp. 499–515. Springer (2016)
14. Deng, J., et al.: Arcface: additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4690–4699 (2019)
15. Alfarsi, G., et al.: Techniques for face verification: literature review. In: 2019 International Arab Conference on Information Technology (ACIT), pp. 107–112. IEEE (2019)
16. Sleit, A., Abu-Hurra, R., Almobaideen, W.: Lower-quarter-based face verification using correlation filter. Imag. Sci. J. 59(1), 41–48 (2011). https://doi.org/10.1179/136821910x12863757400286
17. Taigman, Y., et al.: DeepFace: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1701–1708 (2014)
18. Liu, W., et al.: Sphereface: deep hypersphere embedding for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 212–220 (2017)
19. Bae, G., et al.: DigiFace-1M: 1 million digital face images for face recognition. In: 2023 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE (2023)
20. Liu, Z., et al.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision 2015(Inter), 3730–3738 (2015)
21. Yang, S., et al.: Wider face: a face detection benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5525–5533 (2016)
22. Bendale, A., Boult, T.E.: Towards open set deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1563–1572 (2016)
23. Pabian, M., Rzepka, D., Pawlak, M.: Supervised training of siamese spiking neural networks with earth mover's distance. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4233–4237 (2022)
24. Kumar, S., Chakrabarti, S., Roy, S.: Earth Mover's Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading, vol. 2046–2052. IJCAI (2017)
25. Chen, X., et al.: Symbolic Discovery of Optimization Algorithms (2023). *arXiv preprint arXiv:2302.06675*
26. Yu, J., et al.: Coca: Contrastive Captioners Are Image-Text Foundation Models (2022). *2022. arXiv preprint arXiv:2205.01917*
27. Wortsman, M., et al.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In: International Conference on Machine Learning, pp. 23965–23998. PMLR (2022)
28. Liu, Z., et al.: Large-scale long-tailed recognition in an open world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2537–2546 (2019)
29. DeVries, T., Taylor, G.W.: Learning Confidence for Out-Of-Distribution Detection in Neural Networks (2018). arXiv preprint arXiv:1802.04865
30. Erfani, S.M., et al.: High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. Pattern Recogn. 58, 121–134 (2016). https://doi.org/10.1016/j.patcog.2016.03.028
31. Sun, X., et al.: Conditional Gaussian distribution learning for open set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
32. Ding, Z., Fei, M.: An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. IFAC Proc. Vol. 46(20), 12–17 (2013). https://doi.org/10.3182/20130902-3-cn-3020.00044
33. Er, M.J., et al.: Face recognition with radial basis function (RBF) neural networks. IEEE Trans. Neural Network. 13(3), 697–710 (2002). https://doi.org/10.1109/tnn.2002.1000134
34. Zhang, C., et al.: Deepemd: few-shot image classification with differentiable earth mover's distance and structured classifiers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12203–12213 (2020)
35. Movshovitz-Attias, Y., et al.: No fuss distance metric learning using proxies. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 360–368 (2017)
36. Qian, Q., et al.: Softtriple loss: deep metric learning without triplet sampling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6450–6458 (2019)
37. Kim, S., et al.: Proxy anchor loss for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3238–3247 (2020)
38. Sadeghi, M., Maghooli, K., Moein, M.S.: Using artificial immunity network for face verification. Int. Arab J. Inf. Technol. 11, 354–361 (2014)
39. Jiang, H., et al.: Learning class prototypes via structure alignment for zero-shot recognition. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
40. Che, T. et al.: Deep verifier networks: verification of deep discriminative models with deep generative models. In: Proceedings of the AAAI Conference on Artificial Intelligence, Proceedings of the AAAI Conference on Artificial Intelligence, 35(8), pp. 7002–7010 (2021) https://doi.org/10.1609/aaai.v35i8.16862
41. Čiapas, B., Treigys, P.: High F-score model for recognizing object visibility in images with occluded objects of interest. Baltic J. Modern Comput. 9(1), 35–48 (2021). https://doi.org/10.22364/bjmc.2021.9.1.03
42. Ciapas, B., Treigys, P.: Retail self-checkout image classification performance: similar class grouping or individual class classification approach. In: International Baltic Conference on Digital Business and Intelligent Systems, pp. 167–182. Springer (2022)
43. Wang, Z., Hu, Y., Chia, L.-T.: Image-to-class distance metric learning for image classification. In: European Conference on Computer Vision, pp. 706–719. Springer (2010)
44. Othman, N., Faiz, R., Smaili, K.: Manhattan siamese LSTM for question retrieval in community question answering. In: On the Move to Meaningful Internet Systems: OTM 2019 Conferences: Confederated International Conferences: CoopIS, ODBASE, C&TC 2019, Rhodes, Greece, October 21–25, 2019, Proceedings, pp. 661–677. Springer (2019)
45. Amin, K., et al.: Advanced similarity measures using word embeddings and siamese networks in CBR. In: Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys), 2, pp. 449–462. Springer (2020). https://doi.org/10.1007/978-3-030-29513-4_32
46. Imtiaz, Z., et al.: Duplicate questions pair detection using siamese malstm. IEEE Access 8, 21932–21942 (2020). https://doi.org/10.1109/access.2020.2969041
47. Wang, Z., Zhang, B.: Chinese text similarity calculation model based on multi-attention siamese Bi-LSTM. In: Proceedings of the 4th International Conference on Computer Science and Software Engineering, pp. 93–98 (2021)
48. Zeng, H., et al.: Siam-GCAN: a siamese graph convolutional attention network for EEG emotion recognition. IEEE Trans. Instrum. Meas. 71, 1–9 (2022). https://doi.org/10.1109/tim.2022.3216829
49. Bühler, A., et al.: Deep unsupervised common representation learning for LiDAR and camera data using double siamese networks. *arXiv preprint arXiv:2001.00762* (2020)
50. Öztürk, S.: Comparison of pairwise similarity distance methods for effective hashing. IOP Conference Series: Materials Science and Engineering, IOP Conference Series: Materials Science and Engineering, 1099(1), 012072 (2021) https://doi.org/10.1088/1757-899x/1099/1/012072(IOP Publishing)

51. Tercan, H., Guajardo, A., Meisen, T.: Industrial transfer learning: boosting machine learning in production. In: 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), vol. 1, pp. 274–279. IEEE (2019). https://doi.org/10.1109/indin41052.2019.8972099
52. Huh, M., Agrawal, P., Efros, A.A.: What Makes ImageNet Good for Transfer Learning? arXiv Preprint arXiv:1608.08614 (2016)

How to cite this article: Ciapas, B., Treigys, P.: Centre-loss—A preferred class verification approach over sample-to-sample in self-checkout products datasets. IET Comput. Vis. 1–13 (2024). https://doi.org/10.1049/cvi2.12302