Article

Not peer-reviewed version

Auto-probabilistic Mining Method for Siamese Neural Networks Training

Arseniy Mokin *, Alexander Sheshkus *, Vladimir L. Arlazarov

Posted Date: 1 November 2024

doi: 10.20944/preprints202411.0001.v1

Keywords: deep metric learning; optical character recognition; pattern recognition

 doi:10.20944/preprints202411.0001.v1

Article

Auto-Probabilistic Mining Method for Siamese Neural Networks Training

Arseniy Mokin [1,2,*], Alexander Sheshkus [1,3,*] and Vladimir L. Arlazarov[1,3]

[1] Smart Engines Service LLC, Moscow, Russia
[2] Lomonosov Moscow State University, Moscow, Russia
[3] Federal Research Center "Computer Science and Control" of RAS, Moscow, Russia
* Correspondence: a.mokin@smartengines.com (A.M.); asheshkus@smartengines.com (A.S.)

Abstract: In this paper we present a novel auto-probabilistic mining method designed to enhance the training process of Siamese neural networks. The proposed method demonstrates significant efficiency, particularly in scenarios characterized by limited data and computational resource. The study incorporates the application of the newly developed mining method in conjunction with a previously introduced auto-clustering technique. Comparative experiments are conducted both with and without the integration of the proposed method, highlighting its impact on overall performance. Additionally, the research introduces a novel metric loss to further refine the training process taking into account the drawbacks of existing loss functions applying for metric learning. During the experimentation phase, validation is conducted on the printed Hangul character PHD08 database and Omniglot. This validation provides valuable insights into the performance and effectiveness of the proposed methods.

Keywords: deep metric learning; optical character recognition; pattern recognition

1. Introduction

Deep learning methods have been widely used in recent years, transforming many industries, including natural language processing and computer vision. Because they can efficiently build hierarchical representations from unprocessed input, convolutional neural networks, or CNNs, have become a mainstream in a wide range of applications. Optical Character Recognition (OCR) is one such application where CNNs have demonstrated outstanding performance. OCR systems are essential for digitizing and extracting text from documents, photos, and other scanned materials. They make activities easier, like text-based analysis and automated data entry.

The ability of CNNs to automatically learn discriminative features straight from raw pixel data accounts for their effectiveness in OCR tasks. CNNs are capable of capturing the details necessary for precise character identification by using convolutional layers to extract hierarchical structures and local patterns. For example, OCR is a task where neural networks typically have as many neurons in the last layer as there are distinguishable classes. But handling a huge alphabet (see Fig. 1) in OCR operations is a problem. Under such circumstances, the number of neurons in the last layer increases dramatically, increasing the volume of data and requiring a large amount of processing power and training data. As a result, training these networks becomes ineffective.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 November 2024 doi:10.20944/preprints202411.0001.v1

Figure 1. Part of Korean alphabet (Hangul).

Metric learning aids in avoiding this issue [1]. A network of this type works by matching a point in the metric space to the input vector in such a way that the point is as far from points of other classes and as close to points of the same class as feasible [2]. A few CNN (Convolutional Neural Network) branches with shared weights can be found in Siamese networks. The Euclidean norm is used to calculate the distances between the outputs of the branches, and the resulting values are then sent to the loss function.

In this work we propose an auto-probabilistic mining method applied to training Siamese Neural Networks (SNNs) in OCR tasks. Siamese networks, initially introduced by Bromley et al. [3], have attracted considerable attention for their capability to learn robust representations in scenarios with limited labeled data. Unlike traditional CNN architectures that aim to directly predict class labels, Siamese networks acquire a similarity metric between inputs, allowing them to distinguish subtle differences and similarities even without sufficient labeled data.

1.1. Contrastive Loss

Contrastive loss function [4,5] that was initially proposed by Chopra et al. for face verification tasks, has gained popularity in metric learning experiments as shown by its use in studies like [6–8].

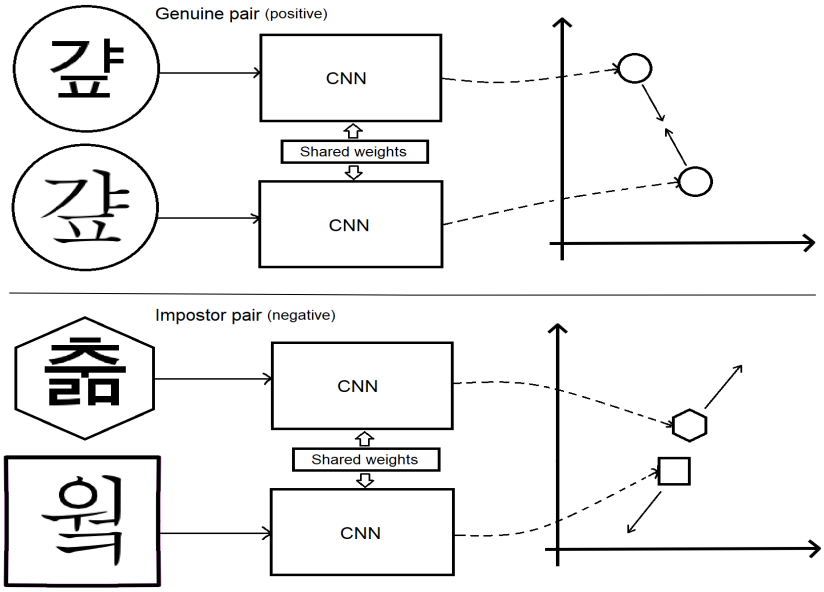

Figure 2. Contrastive loss.

 doi:10.20944/preprints202411.0001.v1

In summary, contrastive loss (Equation 1) uses pairwise distance and pulls positive pairs as close as possible and pushes negative pairs at a distance not less than the selected α value.

$$L^{\alpha}(x^i, x^j; f) = y_{ij} d_{ij}^2 + (1 - y_{ij}) max(0, \alpha - d_{ij})^2, \quad (1)$$

where $d_{ij} = \|f_i - f_j\|_2$ is the Euclidean distance between the pair, y_{ij} returns 1 if $y_i = y_j$ and 0 otherwise.

The absence of a natural stopping mechanism is one of the notable aspects in contrastive loss. This loss function encourages the minimization of distance between positive pair to zero. In practice, the neural network may use computational resources to refine pairs that are already close to convergence, preventing it from exploring and optimizing other examples.

The recent work [9] mentions a different interpretation of a contrastive loss function that has similar meaning but is formulated differently. In our work we refer to the classic version of contrastive loss, proposed in [4,5].

1.2. Triplet Loss

The triplet loss function (Equation 2), as proposed and described in [10], provides an alternative approach to learning embeddings in metric learning tasks. Unlike contrastive loss, which deals with pairs of samples, triplet loss operates on triplets comprising an anchor, a positive sample from the same class as the anchor, and a negative sample from a different class. The objective (Figure 3) of triplet loss is to train the network to reduce the distance between the anchor and positive samples while simultaneously increasing the distance between the anchor and negative samples by at least a specified margin (α).

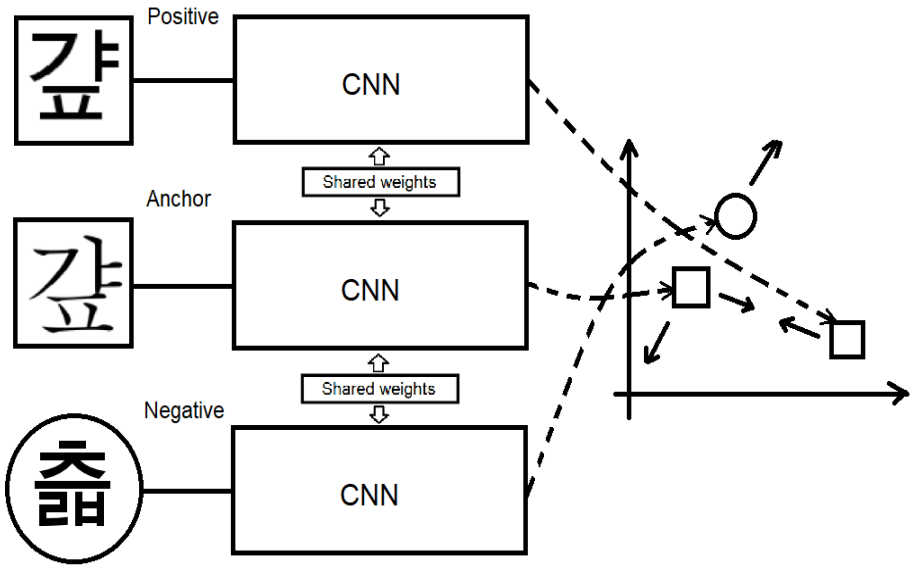

Figure 3. Triplet loss.

However, it's important to note that triplet loss introduces cubic complexity [11], meaning that there are significantly more possible triplets than pairs. Not all triplets are equally informative for training, and including all possible triplets can lead to slower convergence and computational inefficiency. Therefore, selecting "hard" triplets that are particularly challenging [12] and informative is crucial for effectively training the model and achieving faster convergence and improved performance. Apart from that, proposed methods in this work also introduce the strategy how to solve "hard" triplets problem.

Another challenge associated with triplet loss is related to the absence of a direct requirement for the closeness between the anchor and positive instances. Rather, the emphasis lies on ensuring that the

 doi:10.20944/preprints202411.0001.v1

distance between the anchor and positive samples is smaller than the distance between the anchor and negative samples by a predefined margin.

The main purpose of triplet loss is to train the network so that the distance between the anchor and positive becomes less than the distance between the anchor and negative by a selected value equal to or greater than α value.

$$L^{\alpha}(x_a, x_p, x_n, f) = max(0, \|f(x_a) - f(x_p)\|_2 - \|f(x_a) - f(x_n)\|_2 + \alpha) \quad (2)$$

2. Related Work

Numerous researchers in the field of computer vision have delved into Optical Character Recognition (OCR) [13,14]. This task involves classifying and associating characters with their respective classes. While some languages, like English, have achieved high recognition rates [15], the task becomes increasingly challenging as the size of the alphabet exceeds $10,000$, necessitating novel training and recognition methodologies.

Moreover, certain languages permit character decomposition, enabling segmentation into individual components for recognition and subsequent composition of the final result [6,16]. This approach obviates the need for neural networks with tens and even hundreds of thousands outputs. However, it suffers from drawbacks such as vulnerability to image distortions and heavy reliance on segmentation quality. Some researchers [17] choose for deep neural networks with numerous trainable parameters, offering exceptional quality but demanding significant computational resources and unsuitable for mobile applications and so on.

Furthermore, training metric networks presents challenges in managing the sequence of pair/triplet mining. While random data selection is common [18], some works prioritize addressing this issue by focusing on similarity. For instance, in [19], an aggressive hard-mining strategy is employed, mining pairs with the highest error for backpropagation to train the network on challenging examples exclusively. The method's ability to tolerate noise in data can make it difficult to reach the local minimum.

Additionally, distance-based pairs generation methods, such as the approach proposed by [7], demonstrate effective results by generating pairs based on creating a vector of distances for all possible impostor pairs. While promising in terms of training quality, this method is significantly time-consuming.

Auto-clustering [20] method partially solves the mining problem in Siamese network. This method is based on creating clusters, i.e. groups consisting of classes similar from a network's point of view. The usage of clusters allows the network to pay more attention to classes that are difficult to differ. According to the auto-clustering method the negative class was selecting from the same cluster with positive cluster. Let us consider how to effectively choose a positive class.

In the process of training the neural network, we can figure out which classes are far from their cluster in the metric space and try to manually increase the probability of their choice during generating positive class. However, automatic methods recommended itself as the impactful and convenient solution for such tasks.

Due to known bottlenecks in contrastive and triplet losses, a new metric loss is required. Our purpose is to introduce a new loss function that surpasses the commonly used losses.

And according to the existing approaches described above and their disadvantages, we propose:

- auto-probabilistic mining method
- new metric loss

In our experiments we show that the proposed contributions allow neural network to achieve greater accuracy using the same training data and architecture.

 doi:10.20944/preprints202411.0001.v1

3. Suggested Method

3.1. Description

Assigning specific probabilities to classes during neural network training can, in fact, improve the model's capacity to identify certain classes. Prioritizing some classes with greater probability encourages the network to produce them more frequently, which enhances the network's ability to recognize these symbols.

Manually assigning probability raises practical issues in a variety of situations. Firstly, the cognitive strain of keeping comprehensive knowledge of several varied classes is beyond human capacity when their number is in thousands. Furthermore, while automatic probability assignment is possible for some organized data, such as languages with symbol decompositions based on keys, many real-world settings lack such inherent structure, making hand assignment problematic. An automated technique can be applied to a wider range of items, regardless of their type or complexity.

As a result auto-probabilistic method provides a more flexible approach. The network gains the capacity to dynamically adapt and prioritize classes according to their significance for learning by automatically computing the probabilities for each class during training. This makes it possible for the network to determine which classes need more attention and to produce them more frequently, which improves recognition performance overall and optimizes the training process. Therefore, the auto-probabilistic method offers a promising way to boost neural network training efficiency.

Using computed probabilities, the algorithm in this method determines whether each pair belongs to a positive or negative class. This approach differs from uniform distribution techniques and it looks for classes with the greatest average distances between each example of class and corresponding cluster.

3.2. Automatic Calculation of Class Appearance Probabilities

According to the method, during the training process we should calculate average distances for each class. Let $f(x_i)$ output of the model, where x_i is input data, f is model, then let d vector keeping average distances $(f(x_i), c_i)$, where c_i is an ideal vector corresponding to certain class (average vector of the samples of the given class).

$$P = d_i^\gamma / \sum_{j=1}^{N} d_j^\gamma, i = 1, \ldots, N \tag{3}$$

$$P_k = ((1 - w) * P + w * P_{k-1})^\gamma, \tag{4}$$

where

- k - current epoch number
- N - number of classes
- P_k - vector of class appearance probabilities on k-th epoch
- w - weight factor
- γ - amplification

After calculation we should normalize P_k and set it as the discrete distribution, which is used for generation of positive pairs during the training.

3.3. Auto-Clustering Method Improvements

Base Auto-clustering method contains the following steps: calculate all norms between ideal vectors of every class, sort them and since there can be many of them select some part for analyzing.

In the previous version of the Auto-clustering method there were lots of confusing hyper parameters required for network training. And such approach is not convenient for experiments. On the current stage it was decided simplify but retain the essence and leave just two parameters:

 doi:10.20944/preprints202411.0001.v1

- Probability of selecting a class from a cluster for negative pair (θ)
- Number of the lowest norms considered for clusters generation (η)

This method is used for generation of negative sample in triplets during the training.

3.4. Metric Loss

We propose a loss that combines the advantages of contrastive and triplet loss functions. New metric loss is based on triplets and uses f (Equation 5) activation function.

$$a(x) = \log(1 + e^x) \tag{5}$$

First of all we calculate the distances (Equation 6).

$$d_{AP} = \left\| a(f(x_a)) - a(f(x_p)) \right\|_2, d_{AN} = \left\| a(f(x_a)) - a(f(x_n)) \right\|_2 \tag{6}$$

Let us consider different parts of the proposed metric loss.

- $g_1 = \rho * d_{AP}$ (in this work we let $\rho = 0.1$). The output of this block is g_1^2.
- $g_2 = \tau * f(d_{AP} - d_{AN} + \alpha)$ (in this work we let $\tau = 1.0$). The output of this block is g_2^2.
- $g_3 = \xi * d(x_i, c_i)$ (in this work we let $\xi = 1.0$), where $x_i \in (A, P, N)$, c_i is an ideal vector corresponding to certain class (average vector of the samples of the given class).

Parameters ρ, τ, ξ in metric loss give weight to each block correspondingly.

- ρ is responsible for reducing the distance between anchor and positive (contrastive loss contribution),
- τ regulates principle that the distance between the anchor and positive should be less than the distance between the anchor and negative by a selected value equal to or greater than α (triplet loss contribution),
- ξ is responsible for stability so that the clusters created during training do not fall apart and receive a gradient to stay at the same place.

4. Datasets

For all experiments image size was set 37x37 with grayscale. Each epoch of training consists of 50 iterations including 10240 triplets.

4.1. Korean Hangul Recognition

4.1.1. PHD08 Dataset

To evaluate suggested methods the printed Hangul character database called the PHD08 [21] was chosen because the same dataset had been used in the previous works [6,7,20].

The dataset contains 2350 classes, each class has 2187 images of Korean characters. A total of 5139450 binary images have different sizes, rotations, and distortions. Examples of dataset images are shown in Figure 4.

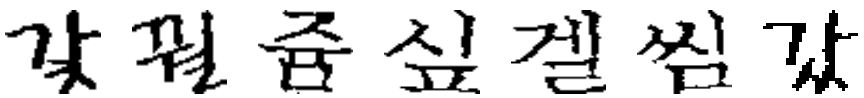

Figure 4. PHD08 dataset examples.

4.1.2. Synthetic Training Data

To assess of objectivity of the proposed approach synthetic data was generated using the same approach as in [20] for training the network in the same way as in the works [6,7,20] using an average 8 fonts per each class. The total number of classes is 11172 characters of the Korean language. Moreover,

 doi:10.20944/preprints202411.0001.v1

in our experiments we also considered not to generate all possible classes and make it equal to the evaluation dataset size. That decision was made to more accurately evaluate auto-probabilistic method, because the network could train on some classes, that don't exist in evaluation dataset. And this fact in combination with the considered approach could significantly increase such character probabilities.

4.2. Omniglot Dataset

The Omniglot dataset was collected by Brenden Lake and his collaborators at MIT via Amazon's Mechanical Turk to produce a standard benchmark for learning from few examples in the handwritten character recognition domain [22].

Figure 5. Omniglot dataset examples.

Omniglot contains 1623 characters from 50 different alphabets (see Fig. 5). It consists of not only international languages like Korean or Latin but also lesser known local dialects and fictional character sets such as Futurama and Klingon. Each of these was hand drawn by 20 different people. The number of letters in each alphabet varies considerably from about 15 to upwards of 40 characters. All characters across these alphabets are produced a single time by each of 20 drawers.

To train our verification network we set aside 60% of the total data for training. Test and validation parts were set in the same way as in [23] both equal to 20%. We fixed a uniform number of training examples per alphabet so that each alphabet receives equal representation during optimization, although this is not guaranteed to the individual character classes within each alphabet.

4.3. Augmentation

Data augmentation [20] was used with a probability of 0.7 for each sample and the following distortions:

- projective transformation - each point of the image is transformed and the values of the shift on the axes x and y are chosen randomly in the range $[0.0, 1.0]$, where presented minimum and maximum shares of offset on the x-axis and y-axis from the width of the image.
- rotation - rotating an image by an angle from the range $[-5, 5]$ in degrees,
- scale - scaling the image to a given size and then scaling the result to the original with $\min = 0.7$ and $\max = 0.9$ scaling coefficients of the original image in width and height to make the image pixelated.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 November 2024 doi:10.20944/preprints202411.0001.v1

5. Experiments

5.1. Model

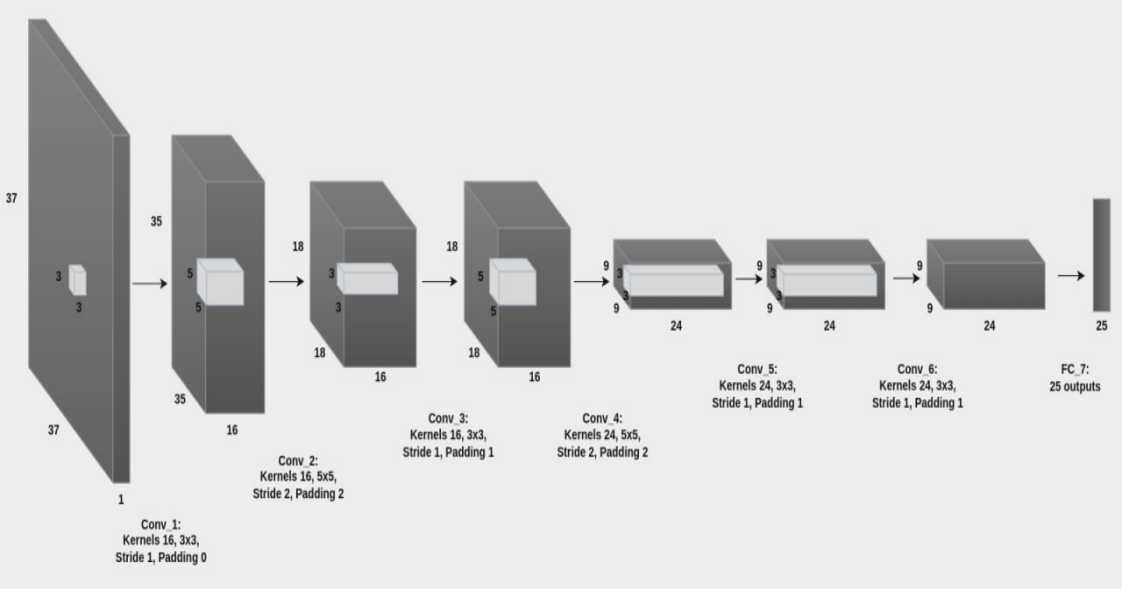

Figure 6. Metric network scheme.

We chose such architecture (Tab. 1) which was also used in the works [6,7,20]. Margin value $\alpha = 10$ was chosen for the metric loss function. A more illustrative scheme is represented on the Figure 6.

Table 1. Architecture of metric network.

Layers				
N	Layer's type	Parameters	Output size	Activation function
1	conv	16 filters 3×3, stride 1×1, no padding	$35 \times 35 \times 16$	softsign
2	conv	16 filters 5×5, stride 2×2, padding 2×2	$18 \times 18 \times 16$	softsign
3	conv	16 filters 3×3, stride 1×1, padding 1×1	$18 \times 18 \times 16$	softsign
4	conv	24 filters 5×5, stride 2×2, padding 2×2	$9 \times 9 \times 24$	softsign
5	conv	24 filters 3×3, stride 1×1, padding 1×1	$9 \times 9 \times 24$	softsign
6	conv	24 filters 3×3, stride 1×1, padding 1×1	$9 \times 9 \times 24$	softsign
7	fully connected	25 outputs	$1 \times 1 \times 25$	-

As an activation function $softsign$ (Equation 7) is used because this function is described in detail in works [24,25] and recommended as the most suitable for the current experiments. So, $softsign$

 doi:10.20944/preprints202411.0001.v1

is a bounded activation function that promotes stable convergence and eliminates any numerical inconsistencies. This function was also used in the works [6,7,20].

$$softsign(x) = \frac{x}{1+|x|} \tag{7}$$

5.2. Training

We performed experiments where the network was trained on synthetic images of Korean characters with the new metric loss.

Moreover, for the objectivity we considered two sizes of training alphabet: large (11172 classes) and small (2350 classes). Such approach provides more objective impact of auto-probabilistic method.

First of all, it was trained on the large alphabet with random-mining method. Secondly, it was trained on the small alphabet in turn with random-mining, auto-clustering and auto-probabilistic. Moreover specific of two last methods allows us to use them both, where auto-probabilistic method is used for positive sample generation and auto-clustering is responsible for negative.

In our experiments auto-probabilistic method showed the best accuracy with $\gamma = 1$, but with auto-clustering parameter $\gamma = 2$ showed the highest result. Auto-clustering method demonstrated high accuracy with $\theta = 0.5$ and $\eta = 1000$, auto-probabilistic method got the highest result with w equals to 0 that means there was no dependency on the previous probability distribution, but this weight factor also can be considered for the future experiments.

5.3. Evaluation

The accuracy(Acc) for classification on validation dataset was defined as:

$$Acc = \frac{N_{correct}}{N_{total}} \cdot 100\%, \tag{8}$$

where N_{total} is the size of the dataset and $N_{correct}$ is the number of images correctly recognized by the network.

6. Results

6.1. PHD08

The results of the experiments with large number of classes are presented in Table 2. The results of random-mining, hard-mining, distance-based mining were taken from work [7] and the result of auto-clustering was demonstrated in [20]. New metric loss function showed outstanding accuracy comparing to all previous results.

Table 2. Results for the alphabet with 11172 classes.

Loss	Method	*Acc*
Contrastive loss	Random-mining [18]	63.7%
	Hard-mining [19]	64.5%
	Distance-based mining [7]	69.7%
	Auto-clustering [20]	76.1%
Metric loss	**Random-mining**	**82.6%**

We can also see in the Table 3 that the proposed auto-probabilistic method improves recognition accuracy. Moreover, this method is supposed to be an efficient and convenient solution regardless of the type of object being recognized. Also, it can be successfully used with and without the integration of previously suggested auto-clustering method.

 doi:10.20944/preprints202411.0001.v1

Table 3. Results for the alphabet with 2350 classes.

Loss	Method	*Acc*
Metric loss	Random-mining	88.6%
	Auto-probabilistic	90.6%
	Auto-clustering	91.1%
	Auto-probabilistic + Auto-clustering	**92.3**%

6.2. Omniglot

As we can see in the Table 4 using the proposed methods improves the recognition quality. Also, it is worth noting that the auto-cluster method has not improved quality significantly. This may be due to the fact that this dataset is mostly collected with different data than the Korean language.

Table 4. Results for the Omniglot dataset.

Loss	Method	*Acc*
Metric loss	Random-mining	91.25%
	Auto-clustering	91.6%
	Auto-probabilistic	92.32%
	Auto-probabilistic + Auto-clustering	**93.17**%

7. Conclusions

We have proposed the auto-probabilistic method, the new mining method which can improve the Siamese model accuracy suitable for recognition tasks as OCR. This method can be successfully used in combination with or without the auto-clustering method and enhance network training with fast convergence and high quality. Furthermore, unlike methods that require an understanding of the object of recognition, such as those that use key-decomposition of characters, the developed method does not require any knowledge of the object and can be used to train networks in any type of object, such as characters, feature point descriptors, faces, and so on. Apart from that the proposed new metric loss combines the benefits of contrastive and triplet loss functions while taking into account their disadvantages.

References

1. Hoffer, E.; Ailon, N. Deep metric learning using triplet network. International Workshop on Similarity-Based Pattern Recognition. Springer, 2015, pp. 84–92.
2. Oh Song, H.; Xiang, Y.; Jegelka, S.; Savarese, S. Deep metric learning via lifted structured feature embedding. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4004–4012.
3. Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems* **1993**, *6*.
4. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, Vol. 1, pp. 539–546.
5. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). IEEE, 2006, Vol. 2, pp. 1735–1742.
6. Ilyuhin, S.A.; Sheshkus, A.V.; Arlazarov, V.L. Recognition of images of Korean characters using embedded networks. Twelfth International Conference on Machine Vision (ICMV 2019). International Society for Optics and Photonics, 2020, Vol. 11433, p. 1143311.
7. Kondrashev, I.V.; Sheshkus, A.V.; Arlazarov, V.V. Distance-based online pairs generation method for metric networks training. Thirteenth International Conference on Machine Vision. International Society for Optics and Photonics, 2021, Vol. 11605, p. 1160508.

 doi:10.20944/preprints202411.0001.v1

8. Wang, X.; Hua, Y.; Kodirov, E.; Hu, G.; Robertson, N.M. Deep metric learning by online soft mining and class-aware attention. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, Vol. 33, pp. 5361–5368.
9. Wang, F.; Liu, H. Understanding the behaviour of contrastive loss. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2495–2504.
10. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.
11. Yuan, Y.; Chen, W.; Yang, Y.; Wang, Z. In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 354–355.
12. Suloev, K.; Sheshkus, A.; Arlazarov, V. Spherical constraints in the triplet loss function. institute for systems analysis russian academy of sciences, 2023, Vol. 73, pp. 50–58. DOI: 10.14357/20790279230205.
13. Khaustov, P.A. Algorithms for handwritten character recognition based on constructing structural models. *Computer Optics* **2017**, *41*, 67–78.
14. Nikolaev, D.P.; Polevoy, D.V.; Tarasova, N.A. Training data synthesis in text recognition problem solved in three-dimensional space. *Informatsionnye Tekhnologii i Vychslitel'nye Sistemy* **2014**, pp. 82–88.
15. Tafti, A.P.; Baghaie, A.; Assefi, M.; Arabnia, H.R.; Yu, Z.; Peissig, P. OCR as a service: an experimental evaluation of Google Docs OCR, Tesseract, ABBYY FineReader, and Transym. International Symposium on Visual Computing. Springer, 2016, pp. 735–746.
16. Franken, M.; van Gemert, J.C. Automatic Egyptian hieroglyph recognition by retrieving images as texts. Proceedings of the 21st ACM international conference on Multimedia. ACM, 2013, pp. 765–768.
17. Kim, Y.g.; Cha, E.y. Learning of Large-Scale Korean Character Data through the Convolutional Neural Network. Proceedings of the Korean Institute of Information and Commucation Sciences Conference. The Korea Institute of Information and Commucation Engineering, 2016, pp. 97–100.
18. Bell, S.; Bala, K. Learning visual similarity for product design with convolutional neural networks. *ACM transactions on graphics (TOG)* **2015**, *34*, 1–10.
19. Simo-Serra, E.; Trulls, E.; Ferraz, L.; Kokkinos, I.; Fua, P.; Moreno-Noguer, F. Discriminative learning of deep convolutional feature point descriptors. Proceedings of the IEEE international conference on computer vision, 2015, pp. 118–126.
20. Mokin, A.K.; Gayer, A.V.; Sheshkus, A.V.; Arlazarov, V.L. Auto-clustering pairs generation method for Siamese neural networks training. Fourteenth International Conference on Machine Vision (ICMV 2021). SPIE, 2022, Vol. 12084, pp. 369–376.
21. Ham, D.S.; Lee, D.R.; Jung, I.S.; Oh, I.S. Construction of printed Hangul character database PHD08. *The Journal of the Korea Contents Association* **2008**, *8*, 33–40.
22. Lake, B.; Salakhutdinov, R.; Gross, J.; Tenenbaum, J. One shot learning of simple visual concepts. Proceedings of the annual meeting of the cognitive science society, 2011, Vol. 33.
23. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. ICML deep learning workshop, 2015, Vol. 2.
24. Bergstra, J.; Desjardins, G.; Lamblin, P.; Bengio, Y. Quadratic polynomials learn better image features. *Technical report, 1337* **2009**.
25. Lin, G.; Shen, W. Research on convolutional neural network based on improved Relu piecewise activation function. *Procedia computer science* **2018**, *131*, 977–984.