Hyper-DINO: Efficient Hyperbolic Embeddings for Histopathological Content-Based Image Retrieval

Stanisław Łażewski ✉[0009−0009−0542−0198]
and Bogusław Cyganek[0000−0001−5185−1145]

AGH University of Krakow, Faculty of Electronics, Computer Science and Telecommunication, ul. Czarnowiejska 78, 30-059 Krakow, Poland
✉ slazewsk@agh.edu.pl

Abstract. In this paper, we propose new hyperbolic embeddings, Hyper-DINO, for use in histopathological content-based image retrieval. First, we used a DINO pretrained ViT Small vision transformer to extract the base features. The resulting [CLS] token is dimensionally reduced using PCA or NCA techniques. These features are then mapped onto a hyperbolic space, so that features from positive pairs are closer together, while features from negative pairs are further apart. For this mapping, we use a specially designed Hyperbolic Mapper model with Hyperbolic Contrastive Loss function. This approach improved MAP@20 results on the Kather and BreakHis histopathology datasets at separate magnifications of 40X, 100X, 200X, and 400X, respectively, to 94.61%, 86.64%, 79.15%, 78.91%, and 75.17%. An additional advantage of the Hyper-DINO features is their compact size. The representation of a single image is a vector of 32 float16 numbers, so it takes up only 64 bytes.

Keywords: HCBIR · Hyperbolic embeddings · ViT · DINO

1 Introduction

In recent years, we have clearly observed the simultaneous occurrence of two phenomena in hospitals: a significant increase in available data and a growing demand for histopathologists. Hospitals have so much data that their analysis must be delegated to algorithms and AI systems, as humans are unable to cope. Properly filtered data can help doctors better diagnose diseases. A system that identifies similar cancer cases in other patients based on histopathological images can accelerate and support their work. To address these challenges, we propose new hyperbolic features Hyper-DINO, which are effective in histopathological content-based image retrieval (HCBIR). There are many ways to extract features from histopathological images. A popular feature extraction method used various CNNs [1][2][4][5]. However, CNNs are limited in the size of convolution kernels and stride settings and struggle to capture global features. The ViT architecture [3], proposed by Dosovitskiy et al., which divides an image into

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

several fixed-size patches and apply the attention mechanism, and its modifications also enable the generation of efficient representations [6][7]. In [8] a DINO approach was proposed. It is a self-supervised learning method forViT where the model learns unlabeled image representations. Our idea is a continuation of the PCA/NCA-DINO feature research [9], which uses the ViT Small vision transformer pre-trained with the DINO approach for feature extraction and previous research [10][11]. The token image representation [CLS] is subjected to dimensionality reduction by PCA [12] or NCA [13]. Inspired by [14], in this work we propose to transform ViT based features to the hyperbolic space using a specially trained mapping model. Hyperbolic embeddings [15][16][17][18][19] often outperform Euclidean representations in computer vision tasks. Our main contributions are as follows:

- The Hyperbolic Mapper (HM) model for mapping compressed ViT embeddings to hyperbolic space, with additional mechanisms to improve the stability and numerical correctness of the calculations.
- The Hyperbolic Contrastive Loss function (HCL) with a dynamic margin and a regularization mechanism that prevents embeddings from escaping to the boundary, where distances increase very rapidly.
- An HCBIR system operating in the hyperbolic space.

2 Method

A method for extracting hyperbolic features is presented. We first extract previously developed PCA/NCA-DINO features from images and then train an HM model using the HCL function, which spreads out negative pairs and brings together positive pairs in hyperbolic space. This makes these features more informative in the HCBIR task. Architecture of our system is shown in Fig. 1.

2.1 Euclidean Embeddings Preparation

The Euclidean embedding preparation process consists of two main steps:

- **Extraction of image representations using the ViT Small.** Raw images are preprocessed and then passed through the ViT Small, according to the extraction pipeline described in [9]. For each image, a [CLS] token of length 384 is extracted.
- **Dimensionality Reduction.** We perform dimensionality reduction using PCA and NCA. The resulting PCA/NCA-DINO features have different lengths, depending on the number of components used: [5, 10, 20, 30, 40, 50, 60, 80, 100, 120, 160, 200, 230, 270, 310, 350, 384]. To further reduce image embeddings, we convert these features to the float16 representation.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

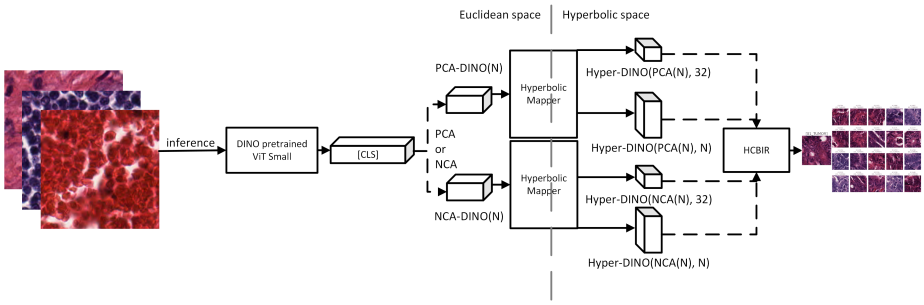


Fig. 1: Architecture diagram.

2.2 Mapping to hyperbolic space

We propose to transfer the prepared features onto the Poincaré ball. We want to learn a mapping of features so that positive image pairs are closer together and negative pairs are farther apart. At the same time, we ensure that embeddings are not placed too close to the boundary of the Poincaré ball, as this will significantly increase the distance between embeddings. For this task, we train the HM model. For an input vector $x \in \mathbb{R}^{dim\_in}$, the network generates a representation $h \in \mathbb{D}^{dim\_out}$, where

$$\mathbb{D}^{dim\_out} = \{z \in \mathbb{R}^{dim\_out} : \|z\| < 1\}$$

is the dim_out-dimensional Poincaré ball. The network consists of two stages:
Euclidean part (neural network). The input is processed by a two-layer fully connected network:

$$e = W_2\,\sigma(W_1 x + b_1) + b_2,$$

where $W_1 \in \mathbb{R}^{256\times dim\_in}$, $b_1 \in \mathbb{R}^{256}$, $W_2 \in \mathbb{R}^{dim\_out\times 256}$, $b_2 \in \mathbb{R}^{dim\_out}$, and $\sigma(\cdot)$ is the ReLU activation function. The vector $e \in \mathbb{R}^{dim\_out}$ is an intermediate Euclidean representation and is interpreted as an element of the tangent space $T_0\mathbb{H}^{dim\_out}$ of the hyperbolic manifold at the reference point 0 (in the Poincaré model). The tangent space at this point is isomorphic to $\mathbb{R}^{dim\_out}$, which allows the use of standard linear operations and classical optimization methods.
Mapping to Hyperbolic Space. The vector e is then mapped to the Poincaré ball using an exponential mapping at the point 0 and a projection onto a ball with radius less than 1. The exponential mapping at point 0 is defined as:

$$exp_0(e) = \tanh(\|e\|)\,\frac{e}{\|e\|},$$

where the norm of $\|e\|$ is lower-bounded by $\varepsilon > 0$ to avoid division by zero. The result is then projected onto a ball of radius $r = 1 - \delta$, where $\delta \ll 1$:

$$h = \beta\, exp_0(e), \quad \beta = \min\left(1, \frac{r}{\|exp_0(e)\|}\right).$$

Finally, we obtain the embedding $h \in \mathbb{D}^{dim\_out}$, which is numerically stable and offset from the boundary.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

2.3 Geometric Operations in the Poincaré Ball Model

The basic operation necessary to define the distance between the hyperbolic embeddings is the Möbius sum of two vectors $x, y \in \mathbb{D}^{dim\_out}$:

$$x \oplus y = \frac{(1 + 2\langle x, y\rangle + \|y\|^2)x + (1 - \|x\|^2)y}{1 + 2\langle x, y\rangle + \|x\|^2\|y\|^2},$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product and $\| \cdot \|$ denotes the Euclidean norm. The Poincaré distance between two points $x, y \in \mathbb{D}^{dim\_out}$ is calculated by:

$$d_{\mathbb{D}}(x, y) = 2 \cdot artanh\big(\|\Delta(x, y)\|\big),$$

where

$$\Delta(x, y) = (-x) \oplus y.$$

In the implementation, the norm of $\|\Delta(x, y)\|$ is truncated to the interval $[\varepsilon, 1-\varepsilon]$.

Hyperbolic Contrastive Loss Function. Embeddings are learned using a contrastive loss function defined in a hyperbolic space. For a batch of B embeddings $h \in \mathbb{R}^{B \times dim\_out}$ and their corresponding C classes labels $y \in \{1, \ldots, C\}^B$, a hyperbolic distance matrix is defined as follows:

$$D = d_{\mathbb{D}}(h, h) \in \mathbb{R}^{B \times B}, \quad D_{ij} = d_{\mathbb{D}}(h_i, h_j), \quad i, j \in 1, ..., B.$$

Masks of positive and negative pairs are defined as follows:

$$S_{ij} = \mathbb{I}[y_i = y_j], \quad N_{ij} = 1 - S_{ij},$$

where S denotes positive pairs (same class), and N denotes negative pairs (different classes). For positive pairs, the hyperbolic distance is minimized:

$$\mathcal{L}_{pos} = \frac{\sum_{i,j} S_{ij} D_{ij}}{\sum_{i,j} S_{ij} + \varepsilon}.$$

This component is responsible for ensuring that embeddings belonging to the same class approach each other in hyperbolic space. For negative pairs, a dynamic margin is applied, depending on the average distance in the batch:

$$m = m_0 + \alpha \bar{d}, \quad \bar{d} = \frac{1}{B^2} \sum_{i,j} D_{ij},$$

where m_0 is the base margin ($m_0 = 0.5$) and α is the scaling factor ($\alpha = 0.1$). The loss for negative pairs is:

$$\mathcal{L}_{neg} = \frac{\sum_{i,j} N_{ij} \max(0, m - D_{ij})}{\sum_{i,j} N_{ij} + \varepsilon}.$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

If the distance D_{ij} is smaller than the margin m, the negative pair contributes positively to the loss, which results in "pushing" embeddings of different classes to larger distances.

We also propose to apply a regularization of the embedding norm to prevent them from getting too close to the boundary of the ball:

$$\mathcal{R}_{norm} = \lambda \cdot \frac{1}{B} \sum_{i=1}^{B} \|h_i\|,$$

where $\lambda = 10^{-3}$. This term encourages embeddings to stay closer to the center of the ball, which improves numerical stability and optimization quality.

Ultimately, the loss function is as follows:

$$\mathcal{L}(h, y) = \mathcal{L}_{pos} + \mathcal{L}_{neg} + \mathcal{R}_{norm}.$$

2.4 Histopathological Content-Based Image Retrieval

The mapped PCA/NCA-DINO features to hyperbolic space using the trained HM model are coined **Hyper-DINO**. For each query image, we calculate the Poincaré distance between its hyperbolic representation and the hyperbolic representations of all reference images. The reference image embeddings closest to the query image embeddings are selected as those that represent the most similar content to the query.

3 Experiments & Results

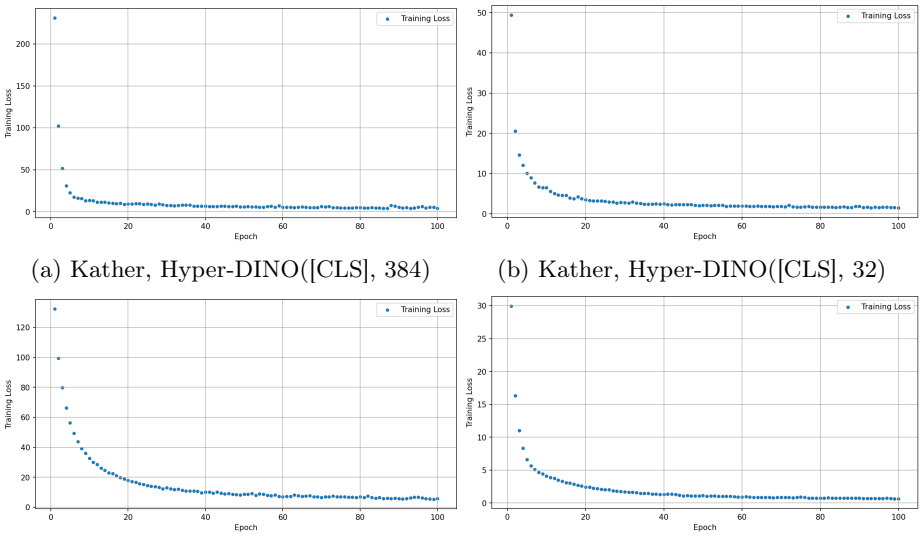
In this chapter, we describe the datasets used for the evaluation, the setup of the experiments, the metrics to testing the quality of the approach, and the results. We use the notation Hyper-DINO($base\_features$, dim_out), where $base\_features$ denotes the base features on which HM mapping is performed, and dim_out denotes the output size of these features. For example, the variant Hyper-DINO(PCA(384), 32) means that the hyperbolic features are derived from the PCA-DINO(384) basis vector (i.e., from the [CLS] ViT Small token, on which, after standard scaling, PCA with 384 components was applied) and that the vector of these features in the hyperbolic space has size 32.

3.1 Datasets & Metrics

To evaluate our proposed method, we used two histopathological image datasets: Kather [20] (colorectal cancer) and BreakHis [21] (breast cancer), at all four available magnifications: 40X (BH40X), 100X (BH100X), 200X (BH200X) and 400X (BH400X). All magnifications of the BreakHis dataset were treated as separate datasets. All of the datasets have 8 classes. Because Hyper-DINO features are based on PCA/NCA-DINO features, the selection of these histopathological datasets allows for a precise comparison of the effectiveness of both methods in

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

the domain of histopathological images. The experiments also reproduced the exact same division of the datasets into folds in 5-fold cross-validation. The results obtained independently at 5 different folds were averaged. To evaluate the proposed approach in the HCBIR task, we again used the $MAP@k$ [9] metric. The symbol $@k$ next to the metric means that the k representations with the shortest distance from the query were used to calculate it. The query images always came from the test set, while the reference database contains images from the training set.



(a) Kather, Hyper-DINO([CLS], 384) (b) Kather, Hyper-DINO([CLS], 32)

(c) BH40X, Hyper-DINO([CLS], 384) (d) BH40X, Hyper-DINO([CLS], 32)

Fig. 2: HM training loss.

3.2 Hyperbolic Mapper Training

Experiments were conducted for two families of hyperbolic features: one with a fixed size of 32 – Hyper-DINO(·, 32), independent of the input vector size, and one with a size corresponding to the input vector size – Hyper-DINO(·, dim_in). Both variants involved training separate HMs for all PCA/NCA-DINO input feature combinations. Experiments with fixed hyperbolic embedding sizes allowed us to assess at which stage of feature generation it is worthwhile to apply dimensionality reduction: either before mapping to hyperbolic space by applying the PCA or NCA transformation on a vector $[CLS]$ with fewer components, as was done in PCA/NCA-DINO, or using HMs, in which we restrict the output to a 32-dimensional vector. HM training lasted 100 epochs, with a batch size of 128. We used the Adam optimizer with a learning rate of $lr = 10^{-3}$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

and $weight\_decay = 10^{-5}$. For all tested datasets, a significant decrease in the loss on the training set occurred during training. Fig. 2 presents graphs of the loss values on the training set during subsequent training iterations. All compared methods used the same training and image preprocessing methods. Table 1 presents the collected setup of all experiments with Hyper-DINO features obtained from Hyperbolic Mapper.

Table 1: Comparison of Hyper-DINO configurations.

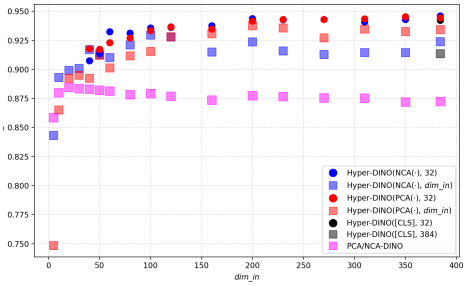
Hyper-DINO output features	**Output vector length** (dim_out)	**Input features**	**Input vector length** (dim_in)	**Compression rate**
Hyper-DINO([CLS], 32)	32	[CLS]	384	12
Hyper-DINO([CLS], 384)	384	[CLS]	384	1
for $\mathbf{N} \in \{\mathbf{5}, \mathbf{10}, \mathbf{20}, \mathbf{30}, \mathbf{40}, \mathbf{50}, \mathbf{60}, \mathbf{80}, \mathbf{100}, \mathbf{120}, \mathbf{160}, \mathbf{200}, \mathbf{230}, \mathbf{270}, \mathbf{310}, \mathbf{350}, \mathbf{384}\} \downarrow$				
Hyper-DINO(NCA(N), N)	N	NCA-DINO(N)	N	1
Hyper-DINO(PCA(N), N)	N	PCA-DINO(N)	N	1
for $\mathbf{M} \in \{\mathbf{40}, \mathbf{50}, \mathbf{60}, \mathbf{80}, \mathbf{100}, \mathbf{120}, \mathbf{160}, \mathbf{200}, \mathbf{230}, \mathbf{270}, \mathbf{310}, \mathbf{350}, \mathbf{384}\} \downarrow$				
Hyper-DINO(NCA(M), 32)	32	NCA-DINO(M)	M	M/32
Hyper-DINO(PCA(M), 32)	32	PCA-DINO(M)	M	M/32

3.3 Results

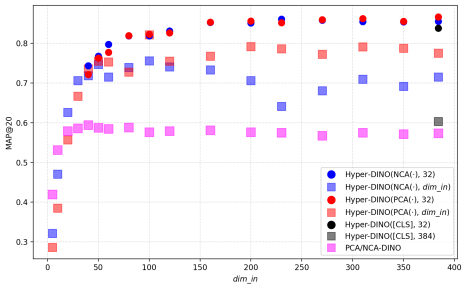
The general trend observed across all histopathology datasets shows that using Hyper-DINO(·, 32) features, consisting of only 32 float16 numbers, yields higher MAP@20 scores. Detailed results by input feature category (PCA/NCA-DINO) and different Hyper-DINO feature variants are presented in Fig. 3. Tab. 2, 3, 4, 5 and 6 provide detailed results of the highest $MAP@k$ scores for different k values and feature variant groups.

For the Kather dataset with 20 components, the MAP@20 for the best PCA/NCA-DINO features hovers around 88%. Using hyperbolic mapping yielded a MAP@20 of 94.61% for the Hyper-DINO(NCA(384), 32) feature, which has a length of only 32. This represents an improvement of over 6 percentage points. Also for smaller values of k, the best results for Hyper-DINO(·, 32) remain over 94%. For the BH40X dataset, the highest MAP@20 for PCA/NCA-DINO features is almost 60%. Our proposed Hyper-DINO(PCA(384), 32) features increase this result to 86.64%, which is an improvement of over 26pp. The best MAP@20 of PCA/NCA-DINO features for the BH100X was 53.01%. For the Hyper-DINO(PCA(350), 32) features, the value of this metric reached 79.15%. On images from the BH200X, the baseline PCA/NCA-DINO features had MAP@20 = 53.48%. The Hyper-DINO(NCA(384), 32) features obtained by hyperbolic mapping achieve the result of 78.91%. In the last of the tested datasets

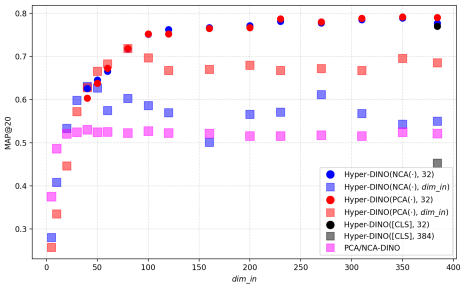
ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34



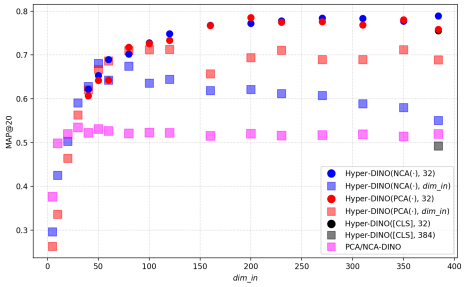
(a) Kather



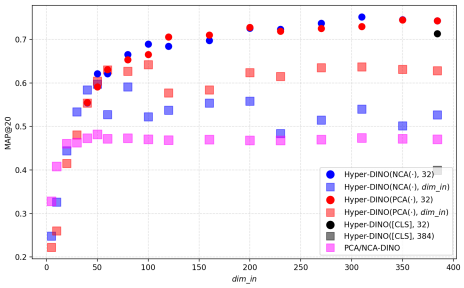
(b) BH40X



(c) BH100X



(d) BH200X



(e) BH400X

Fig. 3: Comparison of MAP@20 of feature variants.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

Table 2: Kather: $MAP@k$ for the best PCA/NCA-DINO, Hyper-DINO(·, 32), and Hyper-DINO(·, *dim_in*) variants, respectively marked as [9], 32, *dim_in*.

method →	k=1			k=5			k=10			k=20		
dim_in ↓	[9]	32	dim_in	[9]	32	dim_in	[9]	32	dim_in	[9]	32	dim_in
384 [CLS]	–	94.22	91.30	–	94.18	91.28	–	94.17	91.24	–	94.24	91.37
384	94.22	94.36	93.32	91.60	94.48	93.42	89.67	94.59	93.45	87.24	94.61	93.44
350	–	94.20	93.18	–	94.44	93.11	–	94.49	93.19	–	94.54	93.27
310	–	94.00	93.02	–	94.26	93.38	–	94.32	93.44	–	94.36	93.47
270	–	94.22	92.60	–	94.22	92.59	–	94.28	92.70	–	94.31	92.72
230	94.22	94.10	93.58	91.83	94.16	93.55	90.12	94.25	93.54	87.66	94.31	93.57
200	–	94.06	93.38	–	94.30	93.58	–	94.33	93.68	–	94.38	93.77
160	94.08	93.58	92.52	91.41	93.61	92.85	89.64	93.68	92.99	87.36	93.76	93.07
120	–	93.46	92.62	–	93.53	92.65	–	93.62	92.82	–	93.67	92.83
100	–	93.06	92.88	–	93.32	93.05	–	93.45	92.96	–	93.57	92.97
80	94.34	92.64	92.30	91.96	92.96	92.06	90.30	93.04	92.07	87.84	93.13	92.11
60	–	92.62	91.26	–	93.04	91.22	–	93.11	91.13	–	93.24	91.03
50	93.94	91.28	91.38	92.06	91.60	91.38	90.31	91.65	91.29	88.21	91.73	91.30
40	93.76	90.98	92.54	91.87	91.43	91.90	90.46	91.62	91.77	88.31	91.79	91.71
30	–	–	89.94	–	–	90.12	–	–	90.04	–	–	90.09
20	93.10	–	89.90	91.50	–	89.91	90.34	–	89.89	88.46	–	89.93
10	91.68	–	89.26	90.53	–	89.50	89.44	–	89.40	87.99	–	89.31
5	–	–	85.08	–	–	85.04	–	–	84.72	–	–	84.32

Table 3: BH40X: $MAP@k$ for the best PCA/NCA-DINO, Hyper-DINO(·, 32), and Hyper-DINO(·, *dim_in*) variants, respectively marked as [9], 32, *dim_in*.

method →	k=1			k=5			k=10			k=20		
dim_in ↓	[9]	32	dim_in	[9]	32	dim_in	[9]	32	dim_in	[9]	32	dim_in
384 [CLS]	–	84.03	70.94	–	83.99	65.69	–	83.90	63.17	–	83.83	60.31
384	88.15	85.73	79.99	77.70	86.33	79.46	68.74	86.54	78.66	57.38	86.64	77.51
350	–	85.42	81.22	–	85.40	80.40	–	85.42	79.89	–	85.51	78.73
310	–	86.25	81.94	–	86.39	81.29	–	86.34	80.30	–	86.22	79.13
270	–	85.69	80.09	–	85.88	78.97	–	86.01	78.29	–	85.95	77.23
230	87.82	86.55	80.74	77.93	86.20	80.58	68.93	86.20	79.67	57.53	86.09	78.60
200	–	86.19	82.69	–	85.83	81.46	–	85.78	80.44	–	85.62	79.17
160	88.64	85.71	79.61	78.32	85.54	78.86	69.56	85.39	77.95	58.14	85.30	76.78
120	–	83.48	79.76	–	83.24	78.54	–	83.17	77.20	–	83.13	75.51
100	–	82.98	83.76	–	82.65	82.74	–	82.46	82.46	–	82.27	82.16
80	87.37	82.79	77.60	77.77	82.40	76.20	69.63	82.24	75.15	58.79	81.94	73.94
60	–	79.87	77.27	–	80.42	76.84	–	80.27	76.11	–	79.76	75.34
50	87.30	78.35	77.99	77.69	77.87	77.03	69.92	77.57	76.47	58.77	76.78	75.52
40	87.66	75.69	77.03	77.52	75.44	75.26	69.55	75.00	74.43	59.44	74.37	73.58
30	–	–	74.51	–	–	73.22	–	–	72.23	–	–	70.64
20	83.64	–	69.36	75.17	–	67.24	67.49	–	65.29	57.89	–	62.60
10	75.67	–	57.34	67.06	–	52.57	61.14	–	49.69	53.16	–	47.04
5	–	–	37.98	–	–	35.29	–	–	34.18	–	–	32.17

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

Table 4: BH100X: $MAP@k$ for the best Hyper-DINO(·, 32) and Hyper-DINO(·, dim_in) variants, respectively marked as 32 and dim_in.

method →	k=1		k=5		k=10		k=20	
length ↓	32	dim_in	32	dim_in	32	dim_in	32	dim_in
384 [CLS]	77.23	58.18	77.33	53.18	77.19	49.65	76.98	45.29
384	79.60	71.69	79.26	70.81	79.20	69.96	79.05	68.53
350	79.78	73.04	79.55	72.06	79.38	71.10	79.15	69.53
310	79.21	69.85	78.98	68.98	78.86	68.14	78.85	66.77
270	78.64	70.56	78.18	69.61	78.14	68.70	78.01	67.19
230	79.02	71.24	78.69	69.57	78.69	68.26	78.72	66.78
200	77.49	71.99	77.25	70.66	77.18	69.44	77.11	67.94
160	77.15	71.49	77.22	70.14	76.89	68.89	76.69	67.01
120	76.60	70.21	77.19	69.34	76.75	68.29	76.26	66.79
100	76.37	72.52	75.97	71.64	75.74	70.89	75.19	69.66
80	72.63	71.49	72.57	71.95	72.41	72.00	71.86	71.85
60	68.90	69.32	67.69	69.18	67.72	68.78	67.26	68.30
50	66.24	68.91	65.63	67.90	65.28	67.44	64.52	66.56
40	64.05	65.45	64.09	64.80	63.42	64.12	62.59	63.06
30	–	64.66	–	62.94	–	61.73	–	59.84
20	–	60.78	–	57.65	–	55.52	–	53.35
10	–	49.10	–	45.36	–	43.18	–	40.83
5	–	31.68	–	30.23	–	29.47	–	28.07

Table 5: BH200X: $MAP@k$ for the best Hyper-DINO(·, 32) and Hyper-DINO(·, dim_in) variants, respectively marked as 32 and dim_in.

method →	k=1		k=5		k=10		k=20	
length ↓	32	dim_in	32	dim_in	32	dim_in	32	dim_in
384 [CLS]	76.23	62.17	75.36	56.50	75.43	53.27	75.54	49.25
384	79.00	71.27	78.82	70.65	78.95	70.09	78.91	68.89
350	78.05	74.29	78.36	73.13	78.13	72.47	78.05	71.20
310	77.86	70.06	78.18	69.91	78.49	69.39	78.37	68.92
270	78.78	71.00	78.59	70.54	78.59	69.94	78.39	68.97
230	78.50	71.92	77.89	72.39	77.86	71.78	77.76	71.05
200	78.52	72.46	78.70	71.51	78.66	70.65	78.54	69.38
160	76.62	69.05	77.09	67.44	76.97	66.69	76.75	65.70
120	75.05	74.11	75.10	72.93	75.13	72.40	74.84	71.25
100	73.30	72.63	73.21	72.40	73.14	71.93	72.76	71.19
80	72.25	71.67	72.87	71.76	72.38	71.49	71.79	70.99
60	69.15	68.64	69.64	68.95	69.70	68.87	68.93	68.61
50	67.53	69.44	66.24	69.06	66.05	68.73	65.34	68.10
40	64.87	66.19	64.39	65.34	63.60	63.97	62.20	62.84
30	–	63.01	–	61.84	–	60.60	–	59.08
20	–	57.86	–	55.15	–	52.86	–	50.26
10	–	50.82	–	47.56	–	45.51	–	42.55
5	–	32.36	–	31.75	–	31.00	–	29.65

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

Table 6: BH400X: $MAP@k$ for the best Hyper-DINO(·, 32) and Hyper-DINO(·, dim_in) variants, respectively marked as 32 and dim_in.

method →	k=1		k=5		k=10		k=20	
length ↓	32	dim_in	32	dim_in	32	dim_in	32	dim_in
384 [CLS]	71.61	52.97	71.63	46.92	71.61	43.75	71.35	39.96
384	74.55	64.83	74.50	64.48	74.36	63.97	74.31	62.82
350	75.19	65.55	74.54	64.75	74.64	64.00	74.55	63.10
310	75.29	67.08	75.22	65.62	75.19	64.82	75.17	63.70
270	74.34	66.46	74.08	65.57	73.92	64.50	73.73	63.48
230	72.16	64.97	72.35	63.77	72.25	63.12	72.33	61.46
200	73.11	65.59	72.91	64.54	72.88	63.56	72.77	62.40
160	71.80	62.27	71.32	61.39	71.09	60.01	71.02	58.41
120	70.99	62.36	71.03	60.59	70.82	59.48	70.56	57.71
100	69.28	66.36	69.39	65.67	69.25	65.24	68.93	64.19
80	66.34	64.80	67.17	64.40	66.87	63.66	66.53	62.69
60	64.84	64.04	64.01	64.07	63.74	63.50	63.05	63.01
50	63.19	63.62	63.04	62.82	63.00	61.11	62.14	60.40
40	58.78	61.43	57.75	60.28	56.86	59.59	55.49	58.41
30	–	58.95	–	56.56	–	55.44	–	53.35
20	–	51.70	–	48.68	–	46.94	–	44.42
10	–	39.72	–	37.07	–	34.95	–	32.65
5	–	27.41	–	26.29	–	25.61	–	24.81

- BH400X $MAP@20$ of the PCA/NCA-DINO features was only 48.20%. For the proposed Hyper-DINO(NCA(310), 32) features, $MAP@20 = 75.17\%$.

3.4 Query examples

To complement the numerical results, we have prepared several examples of HCBIR performance using the Hyper-DINO(PCA(384), 32) feature in Fig. 4 for the Kather and Fig. 5 for the BH40X dataset, respectively. These examples reflect the high $MAP@k$ results presented in the previous section. For the query images from the test sets on the left, images from the reference sets were returned on the right, according to the shortest distance from the query representation (a number placed above each reference image). The classes from which the images originate are also placed above the images. In Fig. 5, we can see that the closest reference image has a distance of 0. In fact, these images are identical, but assigned to different classes. Similarly, images 5 and 6 are also identical and assigned to two different classes. Such cases can cause a significant drop in the $MAP@k$ value; excluding such duplicate images would probably further improve the values of this metric. To maintain consistency with previous studies on these datasets, we used exactly the same training and test datasets in this study. Despite variations in the colors and shapes of the returned closest reference images, classes were generally returned correctly for all test datasets.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

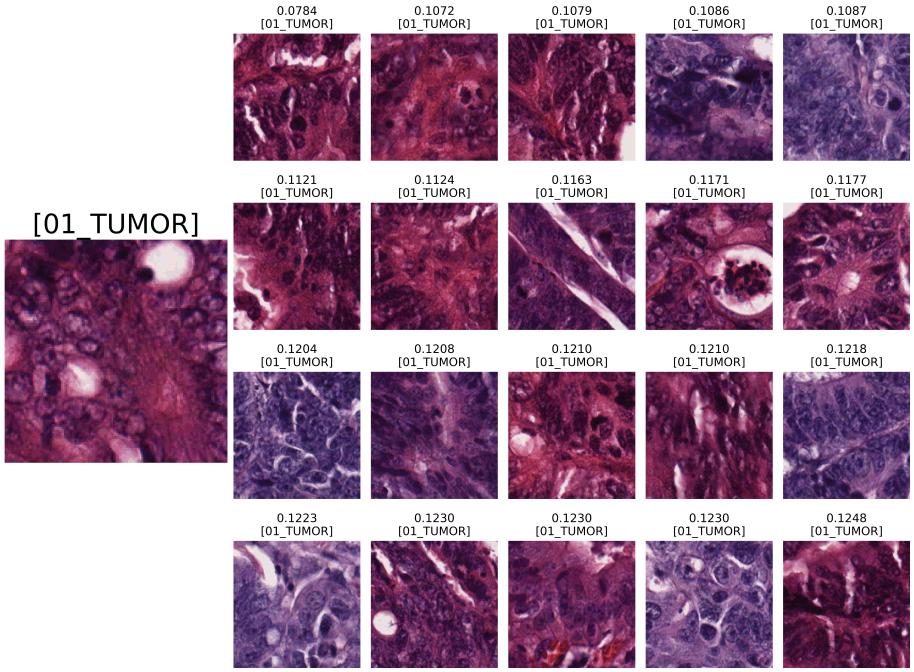


Fig. 4: Kather: On the left side the query, on the right side 20 HCBIR images.

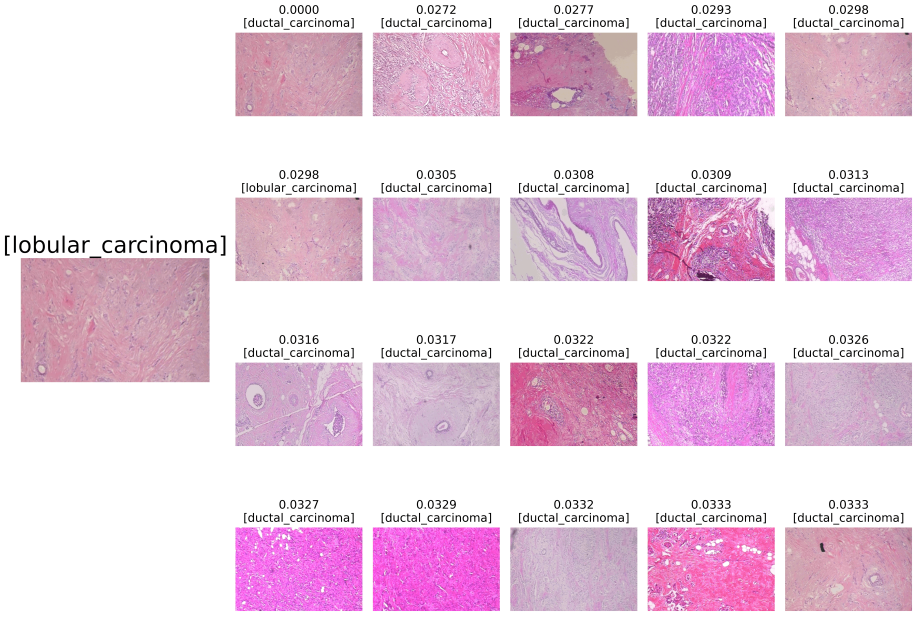


Fig. 5: BH40X: On the left side the query, on the right side 20 HCBIR images.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

4 Conclusions and Discussion

In this paper, we present Hyper-DINO – a new, compact feature set for the HCBIR task. In the first step, we extract the PCA/NCA-DINO features. We then pass them through an HCL-trained HM model, which distributes these features in hyperbolic space so that positive image pairs are close together and negative pairs are further apart. We also introduced mechanisms to prevent embeddings from escaping to the boundary, as this causes the distances to increase dramatically. Experiments confirmed the effectiveness of this approach on the Kather and BreakHis (at separate magnifications of 40X, 100X, 200X, and 400X) histopathology datasets. The $MAP@20$ metric used for evaluation achieved values of 94.61%, 86.64%, 79.15%, 78.91%, and 75.17% on these datasets, respectively. This represents an improvement of over 26 percentage points in some cases, compared to those obtained with PCA/NCA-DINO features. For all datasets, almost all Hyper-DINO(·, 32) feature variants outperformed Hyper-DINO(·, *dim_out*) feature variants. Although the highest results were achieved by variants whose base features were based on PCA or NCA, very similar results were also achieved by features based on the [CLS] vector (after standard scaling). Choosing such variants simplifies the feature creation process because it eliminates the calculation and application of PCA or NCA. Another undoubted advantage of Hyper-DINO features is their size. The feature vector consists of only 32 float16 numbers, meaning it occupies 64 bytes of memory per image. The small size of the features allows for faster computation and requires less memory resources. In future work, we will investigate the impact of transferring dimensionality reduction from the pre-mapping stage to the hyperbolic mapping stage by controlling the output size *dim_out* of the HM model. We also want to apply Hyper-DINO features to multimodal processing of large histopathological datasets such as DiagSet [22].

Acknowledgments. This research was funded in whole by the National Science Centre, Poland, grant number: UMO-2024/55/B/ST6/01681.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Zhang, K., et. al. Content-based image retrieval with a Convolutional Siamese Neural Network: Distinguishing lung cancer and tuberculosis in CT images. Computers in biology and medicine, 140, 105096 (2022)
2. Nuñez-Fernández, C., Farias, H., & Solar, M. (2025). Content-based histopathological image retrieval. Sensors, 25(5), 1350.
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., et al.. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929 (2020)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34

4. Rahaman, M. M., Millar, E. K., & Meijering, E. (2024, May). Histopathology image classification using supervised contrastive deep learning. In 2024 IEEE International Symposium on Biomedical Imaging (ISBI) (pp. 1-5). IEEE.
5. Li, Y., Wu, J., & Wu, Q. (2019). Classification of breast cancer histology images using multi-size and discriminative patches based on deep learning. IEEE Access, 7, 21400-21408.
6. Xu, H. et al. Vision transformers for computational histopathology. IEEE Reviews in Biomedical Engineering, 17, 63-79 (2023)
7. Ding, M., Qu, A., Zhong, H., Lai, Z., Xiao, S., & He, P. (2023). An enhanced vision transformer with wavelet position embedding for histopathological image classification. Pattern Recognition, 140, 109532.
8. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 9650-9660).
9. Łażewski, S., & Cyganek, B. (2025, June). Vision Transformer Representations for Efficient Content-Based Image Retrieval. In International Conference on Artificial Intelligence and Soft Computing (pp. 144-157). Cham: Springer Nature Switzerland.
10. Łażewski, S., & Cyganek, B. (2024, April). Evaluation of highly compressed semantic features for efficient image representation. In Computer Science On-line Conference (pp. 629-640). Cham: Springer Nature Switzerland.
11. Łażewski, S., & Cyganek, B. (2024). Highly compressed image representation for classification and content retrieval. Integrated Computer-Aided Engineering, 31(3), 267-284.
12. Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin philosophical magazine and journal of science, 2(11), 559-572.
13. Goldberger, J., Hinton, G. E., Roweis, S., & Salakhutdinov, R. R. Neighbourhood components analysis. Adv. in neural information processing systems, 17 (2004)
14. Zhang, M., Wang, E., Qin, Q., Hou, J., Zhang, W., & Huang, L. (2025). Deep binary hyperbolic embedding for large-scale image retrieval. Neurocomputing, 131411.
15. Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. Advances in neural information processing systems, 30.
16. Khrulkov, V., Mirvakhabova, L., Ustinova, E., Oseledets, I., & Lempitsky, V. (2020). Hyperbolic image embeddings. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6418-6428).
17. Atigh, M. G., Schoep, J., Acar, E., Van Noord, N., & Mettes, P. (2022). Hyperbolic image segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4453-4462).
18. Liu, S., Chen, J., Pan, L., Ngo, C. W., Chua, T. S., & Jiang, Y. G. (2020). Hyperbolic visual embedding learning for zero-shot recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9273-9281).
19. Fang, P., Harandi, M., & Petersson, L. (2021). Kernel methods in hyperbolic spaces. In Proceedings of the IEEE/CVF Int. Conf. on Computer Vision (pp. 10665-10674).
20. Kather, J. N., et al. Multi-class texture analysis in colorectal cancer histology. Scientific reports, 6(1), 27988 (2016)
21. Spanhol, F. A., et al. A dataset for breast cancer histopathological image classification. IEEE Tr. on Biomedical Engineering, 63(7), 1455-1462 (2015)
22. Koziarski, M., Cyganek, B., Niedziela, P., Olborski, B., Antosz, Z., Żydak, M., & Sitkowski, P. (2024). DiagSet: a dataset for prostate cancer histopathological image classification. Scientific Reports, 14(1), 6780.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_34