Precise Location Matching Improves Dense Contrastive Learning in Digital Pathology

Jingwei Zhang[1](✉), Saarthak Kapse[1], Ke Ma[2], Prateek Prasanna[1],
Maria Vakalopoulou[3], Joel Saltz[1], and Dimitris Samaras[1]

[1] Stony Brook University, Stony Brook, USA
{jingwezhang,samaras}@cs.stonybrook.edu,
{saarthak.kapse,prateek.prasanna}@stonybrook.edu,
Joel.Saltz@stonybrookmedicine.edu
[2] Snap Inc., New York, USA
kemma@cs.stonybrook.edu
[3] CentraleSupélec, University of Paris-Saclay, Paris, France
maria.vakalopoulou@centralesupelec.fr

Abstract. Dense prediction tasks such as segmentation and detection of pathological entities hold crucial clinical value in computational pathology workflows. However, obtaining dense annotations on large cohorts is usually tedious and expensive. Contrastive learning (CL) is thus often employed to leverage large volumes of unlabeled data to pre-train the backbone network. To boost CL for dense prediction, some studies have proposed variations of dense matching objectives in pre-training. However, our analysis shows that employing existing dense matching strategies on histopathology images enforces invariance among incorrect pairs of dense features and, thus, is *imprecise*. To address this, we propose a *precise location-based matching mechanism* that utilizes the overlapping information between geometric transformations to precisely match regions in two augmentations. Extensive experiments on two pretraining datasets (TCGA-BRCA, NCT-CRC-HE) and three downstream datasets (GlaS, CRAG, BCSS) highlight the superiority of our method in semantic and instance segmentation tasks. Our method outperforms previous dense matching methods by up to 7.2% in average precision for detection and 5.6% in average precision for instance segmentation tasks. Additionally, by using our matching mechanism in the three popular contrastive learning frameworks, MoCo-v2, VICRegL, and ConCL, the average precision in detection is improved by 0.7% to 5.2%, and the average precision in segmentation is improved by 0.7% to 4.0%, demonstrating generalizability. Our code is available at https://github.com/cvlab-stonybrook/PLM_SSL.

Keywords: Dense contrastive learning · Self-supervised learning · Segmentation · Detection · Computational Pathology

1 Introduction

In computational pathology, dense prediction tasks such as segmentation and detection are essential in analyzing digitized histology scans [24,25]. However, unlike classifica-

J. Zhang and S. Kapse—These authors contributed equally to this paper.

A. Frangi et al. (Eds.): IPMI 2023, LNCS 13939, pp. 783–794, 2023.
https://doi.org/10.1007/978-3-031-34048-2_60

tion, obtaining labels from pathologists for dense prediction tasks is very tedious and expensive.

Contrastive learning (CL) is being increasingly adopted as a self-supervised learning (SSL) strategy in computational pathology [4,5,17] to reduce the need for annotations. In standard CL, two augmented views are obtained from the input image, and the key idea is to pull the representations of these views closer while pushing apart representations from any other image. Popular CL methods such as SimCLR [6], MoCo [7,8,12], BYOL [11], and VICReg [2], generalize well to multiple computer vision and medical imaging tasks. In CL, traditionally, a spatial pooling operation is applied to the output feature map of the backbone network to encode each view into a global representation. These global CL approaches [6,7,11] work well when the downstream tasks involve classification problems; however, for dense prediction tasks, such representations are not optimal since they require detailed local descriptors of the images. Towards this direction, DenseCL [26] and VICRegL [3] propose incorporating local details in pre-training through leveraging dense matching objectives between the feature maps across both views. In particular, representations from local patches are extracted, and their correspondences across the different views are investigated through the dense matching operation. DenseCL utilizes feature space cosine similarity matching (denoted by M_{ft}) between local representations across the views to find the correspondence pairs. Whereas VICRegL employs spatial location of local patches of the feature maps (denoted by M_{loc}) to find the closest spatial distance between corresponding pairs across the views.

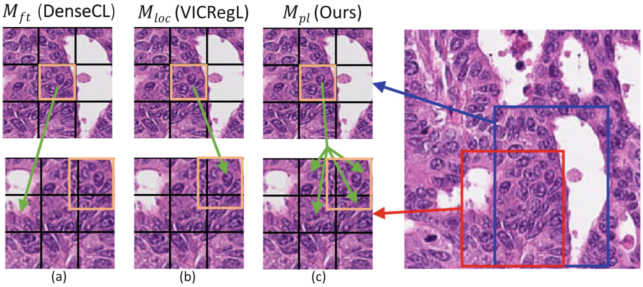

Fig. 1. Local feature matching using three methods for two random augmentations of the same image. Green arrows indicate the matching operation, and orange boxes indicate the same regions in the two views. (a) **Feature similarity-based matching** (M_{ft}), used by DenseCL [26]. The network erroneously pairs patches with multiple nuclei to a local patch containing mainly stroma and non-tissue. (b) **Location-based matching** (M_{loc}), proposed by VICRegL [3]. It matches a patch consisting of multiple nuclei to only one patch in another view containing fewer nuclei due to zooming augmentation. (c) **Precise location-based matching** (M_{pl}). We match a patch to multiple patches in the other view by incorporating exact overlapping weights between the orange boxes across the views. (Color figure online)

Need for Precise Matching in Pathology: The pitfalls of both M_{ft} and M_{loc} can be observed in Fig. 1(a) & (b) respectively. The feature-based matching, M_{ft}, erroneously matches the local patch consisting of multiple nuclei to a patch in another view predominantly consisting of stroma and non-tissue regions. This is because their similarity is defined on the features, which is significantly affected by the model. This could critically hamper the representation learning in histopathology as the invariance between these two patches could force the model to focus on stroma-based descriptors while ignoring other crucial information about cells and their morphology. Location-based matching M_{loc} avoids this error by storing the location information after the geometric transformations to find the pairs. However it can be observed in Fig. 1(b) that the local patch consisting of multiple nuclei is matched with a patch in the other view containing fewer nuclei. This is because they allow matching to strictly one patch. This invariance could potentially enforce the network to ignore crucial information regarding cell density. Due to the *zooming* and *cropping* augmentations, a local patch in a given view may overlap with multiple local patches in another. Formulation of M_{loc} thus has the undesired constraint that a local patch in a view can only match to one corresponding local patch in the other, which is not precise and sub-optimal. Since histopathology images consist of numerous fine-grained individual entities/objects, there is a need for more precise dense matching across the views to overcome the limitations encountered and provide better representations for dense prediction tasks.

To this end, we propose a precise location-based matching strategy, denoted by M_{pl}, which matches a local patch in a view to multiple corresponding overlapping patches in another, as shown in Fig. 1(c). By relaxing the previous constraint, M_{pl} enables *precise matching* between the views. We demonstrate the efficacy of our precise matching strategy in dense prediction tasks involving detection and segmentation on multiple datasets across colon and breast cancer. Experiment results show that our precise location-based matching outperforms previous local matching strategies, improving average precision by up to 7.2% for detection and 5.6% for instance segmentation. We further demonstrate the generalizability of our approach by adopting M_{pl} in three popular contrastive learning frameworks: MoCo-v2 [7], VICRegL [3], and ConCL [28]. M_{pl} shows a relative improvement in average precision by 5.2%, 1.5%, 0.7% for detection, and by 4.0%, 2.9%, 0.7% for instance segmentation with the three aforementioned CL frameworks, respectively.

2 Method

Our method consists of a global contrastive learning part similar to MoCo, which learns a global feature representation of an input image, and a local dense contrastive learning part that learns the local feature representations of small local patches in an input image. These two parts share the same backbone, while the projection heads are different. For the rest of the paper, we use x to represent the input image and x^q and x^k to represent the query and key images, respectively. When x^q and x^k are two randomly augmented views of the same image, we optimize the network to pull their feature representations closer. When x^q and x^k come from two different images, we optimize the network to push their feature representations apart.

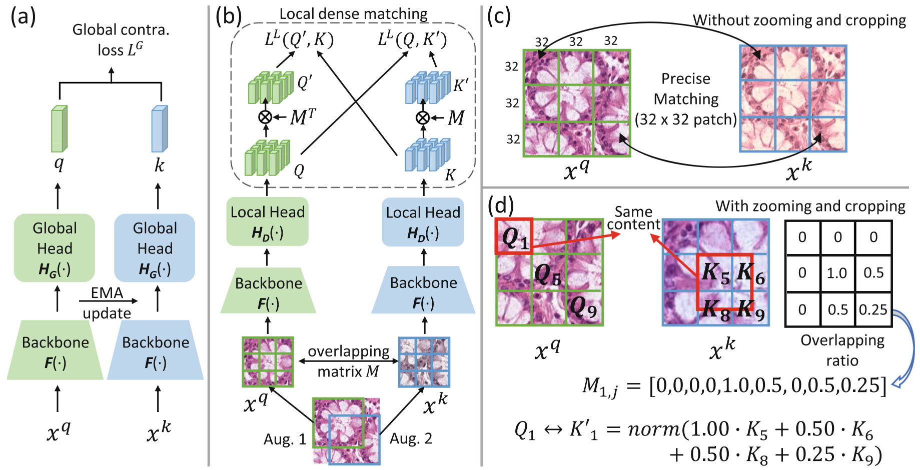

Fig. 2. Overview of the proposed method. (a) Overall structure of our global contrastive framework, the same as MoCo-v2 [7]. For each view of the image, we generate a single global feature representation that represents the entire image. (b) Overall structure of our local dense contrastive framework. For each view of the image, we generate several local feature representations. Each local feature represents a local patch in the image. (c) Without zooming and cropping, patches from two augmentations (for e.g., color jitter) precisely match with each other. (d) If the augmentation contains zooming and cropping, Q_1 matches the weighted sum of K_5, K_6, K_8 and K_9. The weights are calculated as the overlapping ratio of K_i and the red boxed region in x^k corresponding to Q_1. $M_{1,j}$ is the first row of overlapping matrix M.

2.1 Global Contrastive Learning

As shown in Fig. 2(a), in the left branch (query branch), image x^q is passed to a backbone $F(\cdot)$ and a global head $H_G(\cdot)$ to produce a global representation q as shown in Eq. 1. In the right branch (key branch), the same operation is performed on input image x^k to produce a global representation k, as shown in Eq. 1.

$$q = H_G(F(x^q)); \quad k = H_G(F(x^k)) \tag{1}$$

We then calculate the global contrastive loss $\mathcal{L}^G$ between the two global feature q and k as follows:

$$\mathcal{L}^G = \mathcal{L}_{con}(q, k) \tag{2}$$

where the $\mathcal{L}_{con}$ represents the vanilla MoCo-v2 loss [7].

2.2 Local Dense Contrastive Learning

Apart from the global head, we also have a local dense head similar to the DenseCL [26]. As shown in Fig. 2(b), in the left query branch, backbone $F(\cdot)$ and a dense head $H_D(\cdot)$ map the input image x^q to a set of local query features:

$$Q = H_D(F(x^q)) = \{Q_i\}, i = 1, \ldots, n \tag{3}$$

where n is the number of local features. Similarly, the right key branch produces a set of local key features:

$$K = H_D(F(x^k)) = \{K_i\}, i = 1, \ldots, n \tag{4}$$

Each feature Q_i or K_i in the two sets corresponds to a local patch in the original image x; we use P_i^q and P_i^k to represent such regions. As shown in Fig. 2(c), assuming x^q is a 96 × 96 patch and $H_D(F(x^q))$ outputs a 3 × 3 feature map $\{Q_i\}$, $i = 1, \ldots, 9$, each Q_i corresponds to a 32 × 32 P_i^q patch in x^q.

We calculate a local dense contrastive loss $\mathcal{L}^L$ between the two groups of local features Q and K as:

$$\mathcal{L}^L = \mathcal{L}^{pre}(Q, K) + \mathcal{L}^{pre}(K, Q) \tag{5}$$

where the loss is calculated by matching both Q to K and K to Q and $\mathcal{L}^{pre}(\cdot)$ is the precise location-based feature matching loss we will introduce in the next subsection.

2.3 Precise Location-Based Feature Matching

The key problem in the local dense branch is that random zooming and cropping augmentations lead to spatially mismatched features. For example, as shown in Fig. 2(c), if the augmentation operation does not contain any zooming or cropping, Q_i should precisely match to K_i, since they represent the same 32 × 32 patch context ($P_i^q = P_i^k, \forall i$).

However, zooming and cropping are among the key augmentations in contrastive schemes [6,12]. To solve this problem, in this study we propose a method to address this limitation. As demonstrated in Fig. 2(d), when augmentation operations include zooming and cropping, Q_1 and K_1 are spatially mismatched, since the represented regions are different. Instead, Q_1 should match entire K_5 and part of K_6, K_8 and K_9 in the example presented in Fig. 2(d). Observing this, we use a weighted sum of $K_{5,6,8,9}$ to match Q_1, where the weights are calculated from the extent of the overlapping areas between Q_1 and $K_{5,6,8,9}$. To achieve this, we define a $n \times n$ overlapping matrix M between two augmentations, as shown in Fig. 2(b). The elements of M are defined as:

$$M_{i,j} = A(P_i^q \cap P_j^k) \tag{6}$$

where P_i^q represents the i^{th} patch in the query augmentation and $A(x)$ is an area function that calculates the area of x. $M_{i,j}$ can be easily calculated using the bounding boxes of P^q and P^k generated during data augmentation. $M_{i,j}$ represents the overlapped area in the original image x between i^{th} patch in the left query augmentation and j^{th} patch in the right key augmentation. $M_{i,j}$ can be easily calculated from the position and size of the patches P_i^q and P_j^k.

To match the local features Q_i to features K, we need to find out the overlapping area between Q_i and all possible K. This overlapping area is the i^{th} row of the overlapping matrix M, thus the multiplication $M_{i,*} \cdot m_K$ represents the weighted sum of all K overlapped with Q_i. Considering all i, it is (in matrix format):

$$K' = M \cdot m_K / ||M \cdot m_K||, \quad m_K = [K_1, K_2, \ldots, K_n]^\top \tag{7}$$

where $|| \cdot ||$ represents the row-wise L^2 norms. For simplicity, K' is also viewed as a set of its rows $\{K'_{i,*} | i = 1, 2, \ldots, n\}$. The same process is repeated for matching K_i to all possible Q.

$$Q' = M^\top \cdot m_Q / ||M^\top \cdot m_Q||, \quad m_Q = [Q_1, Q_2, \ldots, Q_n]^\top \tag{8}$$

We then define the weights of each pair matching. Matches are not equally important since they have different overlaps. For example, as shown in Fig. 2(d), Q_1 is covered by $K_{5,6,8,9}$. However, Q_5 only overlaps with K_9 by a small area and Q_9 does not have any overlapping with the K. We thus define the weight of matching as follows:

$$w_i^q = \sum_j M_{i,j} / A(P_i^q); \quad w_i^k = \sum_j (M^T)_{i,j} / A(P_i^k) \tag{9}$$

where $A(P_i^q)$ is the area of the i^{th} patch in x^q in the original image x, and $A(P_i^q)$ is the area of the i^{th} patch in x^k in the original image x, and $w_i^q, w_i^k \in [0, 1]$. The final local loss between Q and K can then be formalized as:

$$\mathcal{L}^L = \mathcal{L}^{pre}(Q, K) + \mathcal{L}^{pre}(K, Q) \tag{10}$$

$$= \frac{1}{\sum_i (w_i^q + w_i^k)} \sum_i (w_i^q \cdot \mathcal{L}_{con}(Q_i, K'_i) + w_i^k \cdot \mathcal{L}_{con}(K_i, Q'_i)) \tag{11}$$

where the $\mathcal{L}_{con}$ represents the contrastive loss function which can be any contrastive loss applicable to the problem.

2.4 Optimization

The joint loss $\mathcal{L}$ is defined as the sum of global and local losses as follows:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}^G + \lambda \mathcal{L}^L \tag{12}$$

where $\lambda \in [0, 1]$ is a weight hyper-parameter. The parameters in the left query branch θ_q are optimized end-to-end using the gradients calculated by loss $\mathcal{L}$. The parameters in the right key branch θ_k are optimized using exponential moving average (EMA) as follows:

$$\theta_k = m\theta_k + (1 - m)\theta_q \tag{13}$$

where $m \in [0, 1)$ is a momentum coefficient. We use the MoCo-v2 contrastive learning loss, InfoNCE [16], for $\mathcal{L}_{con}$ in our experiments, given by:

$$\mathcal{L}_{con}^{q,k^+,k^-} = -\log \frac{\exp(q \cdot k^+ / \tau)}{\exp(q \cdot k^+ / \tau) + \sum_{k^-} \exp(q \cdot k^- / \tau)} \tag{14}$$

where q is a query representation, k^+ represents the positive (similar) key samples, and k^- represents the negative (dissimilar) key samples. τ is a temperature hyper-parameter. A query and a key form a positive pair if they are augmented from the same image, and otherwise form a negative pair. Our method does not have any requirement on the choice of contrastive loss for $\mathcal{L}_{con}$, making our framework generalizable to other contrastive learning frameworks.

3 Experiments and Discussion

3.1 Datasets

In our experiments, we use 5 datasets. Two of them, NCT-CRC-HE-100K [15] and TCGA-BRCA-100K [19], are used as pretraining datasets. The other three, GlaS [23], CRAG [10], and BCSS [1] are used to evaluate downstream tasks.

NCT-CRC-HE-100K. The NCT-CRC-HE-100K dataset [15] has 100,000 patches of the size 224 × 224 cropped from 86 H&E stained colorectal adenocarcinoma cancer and normal tissue slides. All the patches are extracted at 20× magnification. The patches are annotated into nine classes. However, these patch-level labels are not utilized as we use this dataset for self-supervised pre-training. This dataset is used for pretraining followed by the downstream segmentation on the GlaS and CRAG dataset.

TCGA-BRCA-100K. The TCGA-BRCA dataset [19] has 1133 slides from patients diagnosed with either Invasive Ductal (IDC) or Invasive Lobular Breast Carcinoma (ILC). We create a dataset by randomly sampling 100,000 tissue patches at 20× magnification and denote this dataset as TCGA-BRCA-100K. Since this dataset is used for pre-training followed by the downstream segmentation on the BCSS [1] dataset, we ensure the slides for the pre-training and downstream tasks do not have any patient-level overlap.

GlaS. The Gland Segmentation in Colon Histology Images (GLaS) dataset [23] has 165 images of size 775 × 522 cropped from 16 H&E histological sections of stage T3 or T4 colorectal adenocarcinoma. The digitization of slides is done at 20× magnification. Each image contains object-instance-level annotations of both the benign and malignant glands.

CRAG. The Colorectal adenocarcinoma gland (CRAG) dataset [10] has 213 images of the size mostly around 1512 × 1516 collected from 38 H&E whole slide images (WSIs). The images are sampled at 20× magnification. The annotations include the instance-level segmentation masks of the adenocarcinoma and benign glands in colon cancer.

BCSS. The Breast Cancer Semantic Segmentation (BCSS) dataset [1] has over 20,000 semantic segmentation annotations of tissue regions sampled from 151 H × E stained breast cancer images at 40× magnification from TCGA-BRCA [19]. The annotations include the segmentation masks of 21 classes, such as Tumor, Stroma, Inflammatory, Necrosis, etc.

Table 1. Quantitative results of object detection, instance segmentation, and semantic segmentation. For GlaS and CRAG, model is pre-trained for 200 epochs on the NCT dataset. For BCSS, model is pre-trained for 200 epochs on randomly sampled patches from TCGA-BRCA. VICRegL^m corresponds to a dense matching extension of VICRegL [3] in MoCo-v2 framework.

Dataset	GlaS		CRAG		BCSS	
Metric	AP_{det}	AP_{seg}	AP_{det}	AP_{seg}	Jaccard	Dice
MoCo-v2	52.3	55.3	50.0	50.3	0.6529	0.7771
w/M_{ft} (DenseCL [26])	53.9	56.5	52.3	52.2	0.6547	0.7778
w/M_{ft} & M_{loc} (VICRegL^m [3])	51.3	56.0	53.5	51.1	0.6554	0.7783
Ours (M_{pl})	**55.0**	**57.5**	**54.5**	**54.0**	**0.6559**	**0.7787**

3.2 Implementation Details

In all the experiments, we use ResNet18 [14] as our backbone network and generate $n = 7 \times 7 = 49$ local features for Q and K. We compare pre-training ResNet18 with the multiple baseline methods and our precise location-based SSL method for 200 epochs with a batch size 256. We use the SGD optimizer with a learning rate of 0.03, weight decay of 0.0001, momentum of 0.9 and apply a cosine annealing learning rate decay policy. For downstream instance segmentation tasks on the GlaS and CRAG datasets, we use MaskRCNN [13] with Resnet18 [14] backbone. We train the network on the CRAG dataset for 15000 iterations and GlaS for 5000 iterations. We use a batch size of 16 and a base learning rate of 0.02. The other hyperparameters are the default ones in Detectron2 [27]. For downstream semantic segmentation on the BCSS dataset, we train a ResNet18 based UNet [22] using the AdamW [20] optimizer with a batch size of 32, a learning rate of $5e^{-4}$, and a cosine annealing learning rate decay. We use the PyTorch library [21], adopting the OpenSelfSup [9] code base. We train our models on NVIDIA Tesla A100 and Nvidia Quadro RTX 8000 GPUs.

3.3 Results

We evaluate the performance of our method on the three downstream datasets involving colorectal and breast cancers. After pre-training, the model is used as the backbone for the downstream segmentation tasks. Since the tasks in the GlaS and CRAG datasets involve instance segmentation, we use the COCO-style [18] metrics to evaluate the model: mean average precision for detection and segmentation, denoted by AP_{det} and AP_{seg}, respectively. For the BCSS dataset, we use the Jaccard index, and the Dice score to evaluate the quality of predictions.

Segmentation Performance Evaluation. *Pre-training:* We use vanilla MoCo-v2 as the base SSL framework. To evaluate the dense contrastive learning baselines, we use the feature-based matching DenseCL and location-based matching VICRegL in MoCo-v2. DenseCL corresponds to MoCo-v2 w/ M_{ft}, whereas MoCo-v2 w/ M_{ft} & M_{loc} corresponds to adoption of VICRegL [3] in the MoCo-v2 framework (denoted as

Table 2. Experiments on the generalizability of our proposed method on the GlaS dataset: 1) MoCo-v2 (global contrastive learning), 2) VICRegL (global + local dense contrastive learning), 3) ConCL (global + clustering-based contrastive learning).

CL method	MoCo-v2 [7]		VICRegL [3]		ConCL [28]	
Metric	AP_{det}	AP_{seg}	AP_{det}	AP_{seg}	AP_{det}	AP_{seg}
vanilla method	52.3	55.3	48.3	52.3	56.8	58.7
vanilla method w/M_{pl}	**55.0**	**57.5**	**49.0**	**53.8**	**57.2**	**59.1**

VICRegL^m). Our precise location-based matching M_{pl} in MoCo-v2 is denoted in the rest of the paper as MoCo-v2 w/ M_{pl}. In Table 1, we observe that pre-training MoCo-v2 with dense matching techniques such as M_{ft} or M_{loc} results in better performance. Our proposed dense matching method M_{pl}, unlike M_{loc}, uses better zooming and cropping augmentations and thus is a more reliable as a complete dense matching strategy. In Table 1, we empirically verify this claim by showing a consistent improvement across multiple datasets and downstream tasks. Compared to vanilla MoCo-v2, our method achieves a relative improvement in average precision of 5.2% and 9% in detection, 4.0% and 7.3% in instance segmentation on GlaS and CRAG, respectively. Compared to DenseCL, we see a consistent relative improvement of 2.0% and 4.2% in detection, 1.8% and 3.5% in instance segmentation; compared to VICRegL^m, we see a relative improvement of 7.2% and 1.9% in detection, 2.7% and 5.7% in instance segmentation on GlaS and CRAG respectively. For semantic segmentation, compared to vanilla MoCo-v2, relative improvement in Jaccard index is up to 0.45%.

Evaluation of Generalizability. We demonstrate the generalizability of our precise location-based matching by incorporating M_{pl} into popular SSL frameworks, including vanilla MoCo-v2 [7], VICRegL [3], and ConCL [28]. All experiments are performed on the GlaS dataset in this study. In Table 2, we observe that our matching method consistently boosts the performance of all the SSL frameworks. This shows that our precise location-based matching can be easily adopted by a diverse set of SSL frameworks to boost the representation learning abilities for the dense prediction tasks such as detection and segmentation.

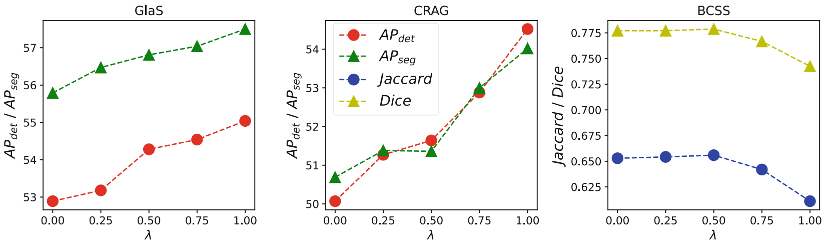

Fig. 3. Illustration of the effect of loss weight λ on model performance The optimal λ for the GlaS and the CRAG dataset is 1.0, and the optimal λ for the BCSS dataset is 0.5.

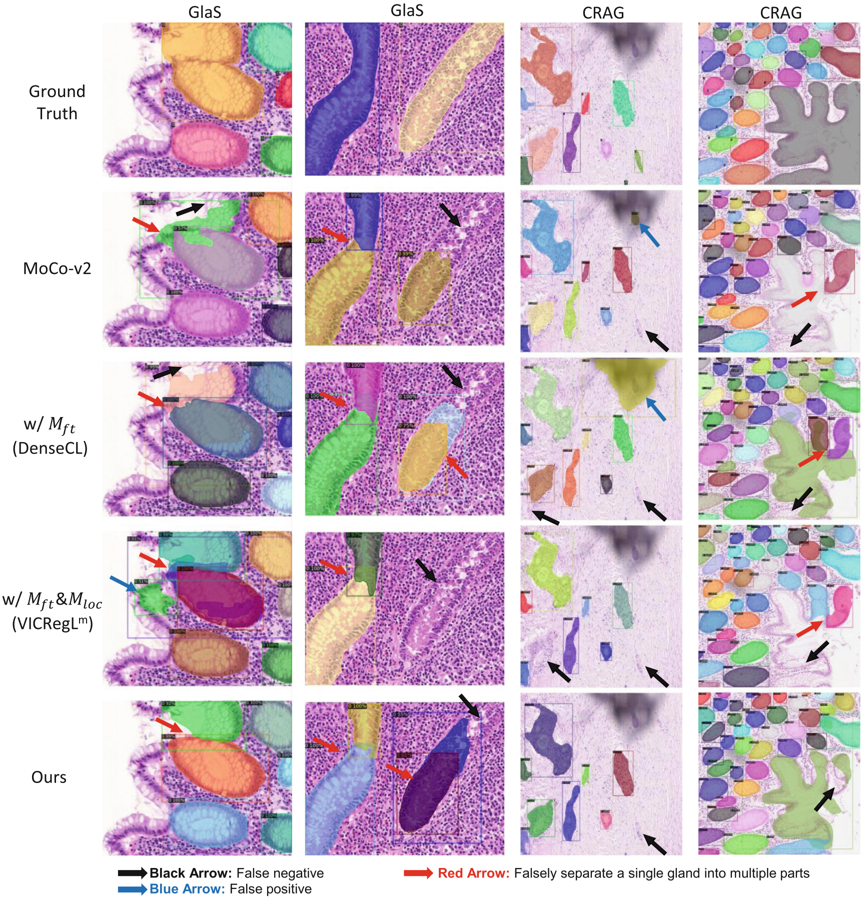
GlaS
GlaS
CRAG
CRAG
Ground Truth
MoCo-v2
w/ M_{ft} (DenseCL)
w/ M_{ft}&M_{loc} (VICRegL^{m})
Ours
Black Arrow: False negative
Blue Arrow: False positive
Red Arrow: Falsely separate a single gland into multiple parts

Fig. 4. Qualitative comparison on the GlaS and the CRAG datasets. Our method has fewer false negative and false positive segmentations, outperforming other methods. (Color figure online)

Ablation Study on the Hyperparameter λ. To study the optimal weight of the global and the local losses, we perform an ablation study on $\lambda \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. When $\lambda = 0.0$, the global contrastive loss is used alone for the training, whereas for $\lambda = 1.0$, only our proposed local dense loss is used. Figure 3 demonstrates that overall our proposed loss boosts the performance of the models on all three datasets. In particular, for instance segmentation tasks (GlaS and CRAG datasets), where we have multiple local objects to segment, our formulation alone provides the best performance and global contrastive schemes may not be so helpful. On the other hand, for semantic segmentation tasks (BCSS dataset), the best performance is achieved when both global and local loss components are combined. Indeed, for such tasks, global interactions of different regions are important to capture different structures.

Qualitative Comparison. To qualitatively compare the performance of our method against others, we visualize the segmentation masks and detection boxes in Fig. 4.

Different detection errors (false positives, false negatives and falsely separating a single gland into multiple parts) are indicated by arrows in different colors. Overall, our method has fewer false positive and false negative errors, outperforming previous methods and providing more robust segmentations.

4 Conclusion

In this paper, we introduced a precise location-based matching for SSL frameworks that matches a local patch in a view to multiple corresponding overlapping patches in the other view. We applied our proposed matching on two pre-training datasets and evaluated on three downstream tasks. Our method consistently outperforms state-of-the-art local matching strategies, showing substantial improvement in average precision in both detection and instance segmentation. Moreover, by using our matching mechanism, the average precision in detection and segmentation was improved in the three popular contrastive learning frameworks, demonstrating the method's generalizability. Our proposed approach shows the promising potential of local matching in self supervised learning. In future work, we will perform extensive cross-validation on the current datasets and further explore better matching mechanisms and their application to a diverse set of computational pathology tasks.

Acknowledgements. This work was partially supported by the ANR Hagnodice ANR-21-CE45-0007, the NSF IIS-2212046, the NSF IIS-2123920, the NCI UH3CA225021, the NIH 1R21CA258493-01A1, the Stony Brook Provost Venture Fund (ProFund) and generous donor support from Bob Beals and Betsy Barton.

References

1. Amgad, M., et al.: Structured crowdsourcing enables convolutional segmentation of histology images. Bioinformatics **35**(18), 3461–3467 (2019)
2. Bardes, A., Ponce, J., LeCun, Y.: VICReg: variance-invariance-covariance regularization for self-supervised learning. In: International Conference on Learning Representations (2022)
3. Bardes, A., Ponce, J., LeCun, Y.: VICRegL: self-supervised learning of local visual features. In: NeurIPS (2022)
4. Boyd, J., Liashuha, M., Deutsch, E., Paragios, N., Christodoulidis, S., Vakalopoulou, M.: Self-supervised representation learning using visual field expansion on digital pathology. In: Proceedings of the IEEE/CVF ICCV, pp. 639–647 (2021)
5. Chen, R.J., et al.: Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In: Proceedings of the IEEE CVPR, pp. 16144–16155 (2022)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML, pp. 1597–1607. PMLR (2020)
7. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
8. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF ICCV, pp. 9640–9649 (2021)
9. Contributors, M.: MMSelfSup: OpenMMLab self-supervised learning toolbox and benchmark (2021). https://github.com/open-mmlab/mmselfsup

10. Graham, S., et al.: MILD-net: minimal information loss dilated network for gland instance segmentation in colon histology images. Med. Image Anal. **52**, 199–211 (2019)
11. Grill, J.B., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Adv. Neural. Inf. Process. Syst. **33**, 21271–21284 (2020)
12. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF CVPR, pp. 9729–9738 (2020)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV, pp. 2961–2969 (2017)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE CVPR, pp. 770–778 (2016)
15. Kather, J., Halama, N., Marx, A.: 100,000 histological images of human colorectal cancer and healthy tissue. 1214456 (2018) https://doi.org/10.5281/zenodo
16. Lai, C.I.: Contrastive predictive coding based feature for automatic speaker verification. arXiv preprint arXiv:1904.01575 (2019)
17. Li, B., Li, Y., Eliceiri, K.W.: Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In: CVPR (2021)
18. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
19. Lingle, W., et al.: Radiology data from the cancer genome atlas breast invasive carcinoma [TCGA-BRCA] collection. Cancer Imaging Arch. **10**, K9 (2016)
20. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2018)
21. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. Adv. Neural Inf. Process. Syst. **32** (2019). https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html
22. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
23. Sirinukunwattana, K., et al.: Gland segmentation in colon histology images: the GlaS challenge contest. Med. Image Anal. **35**, 489–502 (2017)
24. Tellez, D., Litjens, G., van der Laak, J., Ciompi, F.: Neural image compression for gigapixel histopathology image analysis. IEEE TPAMI **43**(2), 567–578 (2019)
25. Wang, S., Yang, D.M., Rong, R., Zhan, X., Xiao, G.: Pathology image analysis using segmentation deep learning algorithms. Am. J. Pathol. **189**(9), 1686–1698 (2019)
26. Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense contrastive learning for self-supervised visual pre-training. In: Proceedings of the IEEE CVPR, pp. 3024–3033 (2021)
27. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2 (2019)
28. Yang, J., Chen, H., Liang, Y., Huang, J., He, L., Yao, J.: ConCL: concept contrastive learning for dense prediction pre-training in pathology images. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13681, pp. 523–539. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19803-8_31