Improving Pixel-Level Contrastive Learning by Leveraging Exogenous Depth Information

Ahmed Ben Saad[1,2] Kristina Prokopetc[2] Josselin Kherroubi[2]
Axel Davy[1] Adrien Courtois[1] Gabriele Facciolo[1]

[1]ENS Paris-Saclay, Centre Borelli   [2]Schlumberger AI Lab
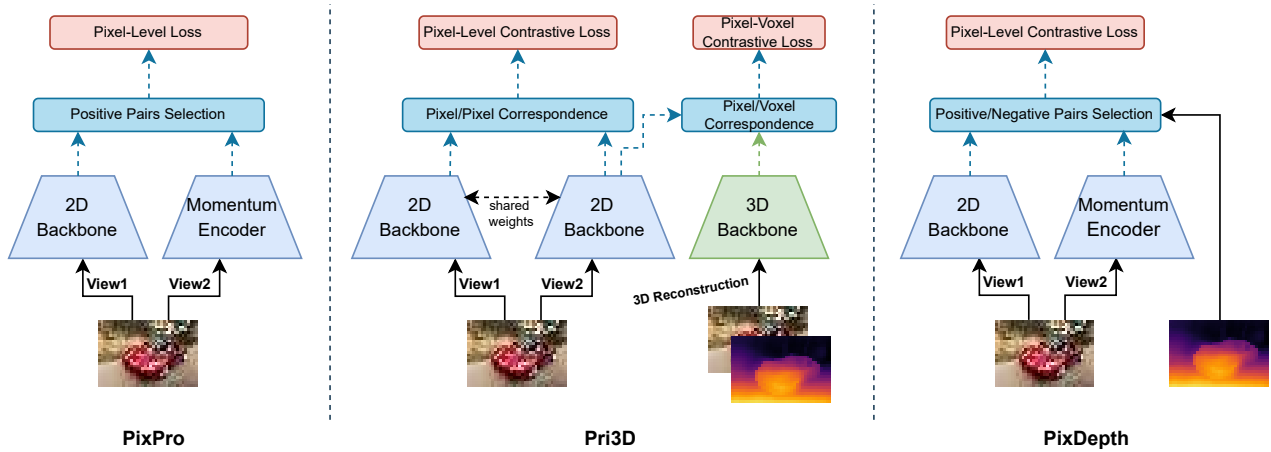
Figure 1. Visual comparison between existing methods of Contrastive Learning for Segmentation and our method: PixPro [1] (left) does not use any 3D priors and only relies on 2D distance between image patches to distinguish between positive and negative samples. Pri3D [2] (middle) uses a 3D prior by reconstructing 3D scenes from stereo data and then leverages pixel to pixel and pixel to voxel correspondences. We propose to use 3D prior knowledge in a simpler way (right): A full 3D representation is not required. Instead depth maps are used in the positive/negative pairs selection, which makes it easier to train and improves results on several datasets. The flow of data, 2D and 3D representations is represented by black arrows, blue arrows and green arrows respectively.

Abstract

Self-supervised representation learning based on Contrastive Learning (CL) has been the subject of much attention in recent years. This is due to the excellent results obtained on a variety of subsequent tasks (in particular classification), without requiring a large amount of labeled samples. However, most reference CL algorithms (such as SimCLR and MoCo, but also BYOL and Barlow Twins) are not adapted to pixel-level downstream tasks. One existing solution known as PixPro proposes a pixel-level approach that is based on filtering of pairs of positive/negative image crops of the same image using the distance between the crops in the whole image. We argue that this idea can be further enhanced by incorporating semantic information provided by exogenous data as an additional selection filter, which can be used (at training time) to improve the selection of the pixel-level positive/negative samples. In this paper we will focus on the depth information, which can be obtained by using a depth estimation network or measured from available data (stereovision, parallax motion, LiDAR, etc.). Scene depth can provide meaningful cues to distinguish pixels belonging to different objects based on their depth. We show that using this exogenous information in the contrastive loss leads to improved results and that the learned representations better follow the shapes of objects. In addition, we introduce a multi-scale loss that alleviates the issue of finding the training parameters adapted to different object sizes. We demonstrate the effectiveness of our ideas on the Breakout Segmentation on Borehole Images where we achieve an improvement of 1.9% over PixPro and nearly 5% over the supervised baseline. We further validate our technique on the indoor scene segmentation tasks with ScanNet and outdoor scenes with CityScapes (1.6% and 1.1% improvement over PixPro respectively).

1. Introduction

Research in Deep Learning and Computer Vision took an important change in direction towards Unsupervised and Self-Supervised Learning [3, 4, 5]. This is mostly motivated by an abundance of unlabeled data in various applications and the high cost of labeling efforts. Specifically, a huge amount of unlabeled data is available from remote sensing. For example, in the petrophysical domain, the years of data acquisition with ultrasonic and other types of sensors have lead to a tremendous accumulation of data archives. Whilst the task of data labeling is not trivial, tedious and time consuming. Thus, the incentive of learning representations without the use of labels is very promising. Following this trend, we have seen an emergence of Contrastive Learning (CL) methods [6] as a promising set of algorithms that learns representations from data without the need for labels. These methods were adapted for Computer Vision [7, 8, 9, 10] and other methods were inspired by them [11, 12], which resulted in improvements in classification tasks on ImageNet [13] and CIFAR-100 [14]. The most successful CL based methods [8, 4, 11] rely on two important assumptions: two transformed versions of the same image must have their representations as "close" as possible. Similarly, different images should have their representations as "far" as possible (even though some methods such as BYOL [11] do not rely on this second assumption explicitly). This paradigm is called instance discrimination. According to this logic, learned representations are invariant to the set of transformations used during training. This results in instance-level features that perform well when transferred to instance-level tasks such as classification. This, however, does not encode positional information, which is crucial for pixel-level tasks [15, 16, 17, 18]. As a result, these representations are poorly transferable for semantic and instance segmentation.

One proposed solution to fix this issue is PixPro [1]. The idea behind this method is to modify BYOL by adding a pixel-level task that uses the 2D distance between pixel-level features (belonging to two transformed image crops) to determine the positive/negative pairs. More precisely, distances smaller than a threshold are considered as positive pairs and are pulled to each other. They also propose an additional module that is used to smooth the pixel representations. While this method yields improvements on segmentation tasks compared to supervised pre-training, we believe that it can be further improved for the following reason: the positive/negative pairs selection procedure described above is equivalent to adding a shift transformation in BYOL. In fact, forcing representations of close patches to be similar makes them invariant to displacements in all directions by any value below the chosen threshold, which comes down to adding small shifts to the data augmentation. This means that we do not take into account that nearby points might
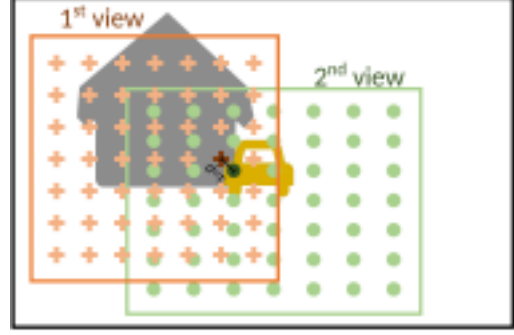

Figure 2. Example of a failure case for PixPro: the . and the + correspond to the learned representations of two views respectively. The two pixels separated by the black segment are close on the image ($d < \tau$) despite belonging to two different objects. This figure is based on Figure 1 in [1]

belong to different objects. For example, when some objects occlude parts of other objects. If the distance filtering threshold is not small enough or when crops are located on objects borders, this can lead to feature pairings that do not belong to the same object yet are treated as positive pairs.

To alleviate this problem, we propose PixDepth. The main idea is to leverage exogenous depth information in the positive/negative pairs selection. The image depth does not only provide very useful prior semantic information about the correct object boundaries, but also prevents learning fuzzy representations near the object boundaries. Specifically, when comparing pixel-level features, in addition to the 2d distance we also compute the difference in depth at the corresponding locations. If the depth difference is larger than a depth threshold we assume that the objects are distinct. This additional constraint refines the choice of positive pairs, thus injecting more semantic information in the training. We demonstrate our approach by pre-training on the ImageNet dataset [19] and using depth maps precomputed using MIDAS [20] and then fine-tuning on the CityScapes dataset [21]. Moreover, we show that we can drop the pre-trained monocular depth estimator and directly use the large amount of existing binocular stereo data to pretrain PixDepth. In addition, we will discuss how the threshold parameters may depend on the scale of the objects in the dataset and propose a multi-threshold loss that reduces this dependency.

We demonstrate the application of our ideas with experiments on natural images from ImageNet [19] and ScanNet [22] (using monocular depth estimations and measured depth data for pretraining respectively) as well as on an industrial use case with borehole images. In this case we will use the transit time of ultrasonic waves as an indicator of depth. We also present an experimental setup to qualitatively analyze the representations learned by PixPro and PixDepth. This work is motivated by the geological fea-
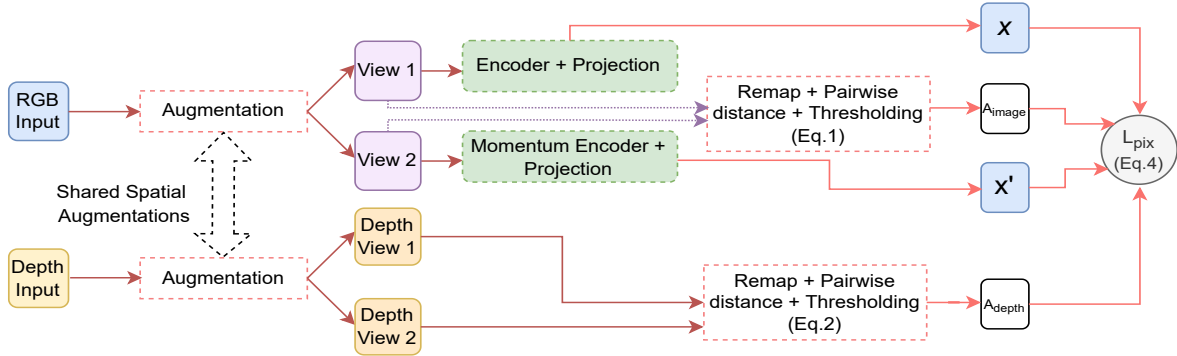
Figure 3. The detailed overview of our proposed framework (PixDepth): given an input image, we apply a set of augmentations and then crop two patches (as in [1]). We do the same thing for the depths views. Next, each view goes through the target or the momentum network. To build the positive and negative pairs, the depth views are remapped to the feature map sizes and then a pairwise absolute difference (Equation 3) is calculated to measure the distance in depth space between image patches. This newly obtained mask $\mathcal{A}_{depth}$ helps the network to better distinguish if the selected patches are of the same object or not.

tures detection in petrophysical domain and can be easily transferred to other applications where this external information is widely available in contrast to labeled data. To summarize, our contributions are:

- Improving PixPro by making use of the prior knowledge of depth to improve the learned representations.
- A multi-scale loss and training with multiple distance and depth thresholds that helps encoding objects with different scales.
- An extensive experimental evaluation to support our findings on different datasets for semantic segmentation on CityScapes and ScanNet and geological features identification on a BoreholeImage dataset.

2. Related Work

Contrastive Learning and Computer Vision   There exist many works which apply CL to visual tasks [7, 8, 9, 10, 4]. Most notably SimCLR [8] proposes a simple CL framework that surpassed the supervised pre-training in ImageNet classification but required a very large batch size to ensure good performances. BYOL [11], on the other hand, is not an explicit CL method but was inspired from SimCLR and uses a momentum encoder to remove the need of large batch sizes. The main limitation of these methods is inability to generate instance-level features that transfer well for pixel-level tasks. When it comes to image segmentation, [1] proposes a modification to BYOL by adding a pixel-level pretext task that separates positive and negative pairs of two crops in one image by computing the distance between these crops. If this distance is smaller than a given threshold, the pair of crops is considered as positive example and a negative example otherwise. Our method extends this idea in two ways: we improve the selection of pairs

by leveraging depth information and we incorporate different thresholds at the same time to remove the dependence on thresholds as hyperparameters by taking multiple scales into account.

Pri3D [2] is based on a similar intuition of using prior 3D information to improve CL. The authors rely on a 3D reconstruction of indoor scenes to build another proposed pixel-level pretext task. This creates pixel-to-pixel correspondences from different views of the same scene and pixel-to-voxel correspondences between representations of 2D views and 3D meshes. While this idea is comparable to ours, we believe that our framework is simpler because we do not rely on a 3D reconstruction and a neural network to encode this 3D structure. Moreover, depth maps are more readily available (from RGB-D sensors, stereovision, monocular estimation) than the full 3D scenes used by Pri3D.

Recently, Point-level Region Contrast [23] has proposed a CL approach at region level by directly sampling individual point pairs from different regions that are defined as a fixed grid at the beginning of the training and are updated periodically during training. In contrast, we believe that the depth information provides a strong semantic hint for segmenting objects as they have distinct depths from their background.

Depth Estimation   Visual ability to perceive the world in 3D and the ability to estimate the distance/depth of an object from the source is paramount in numerous tasks including scene understanding. The details in the environment that allow to perceive depth are often called depth cues. They can be categorised as *binocular* cues, when viewing a scene with both eyes, as *motion* cues, resulting from the motion of the observer or in the scene, or as *monocular* cues, when

viewing a static scene with one eye. Binocular stereo-based depth estimation is a go-to solution when more than one observation of the scene is available. To achieve this, matching pixels from the left and right camera images are identified and the disparity (or difference) in corresponding pixel locations can then be used to infer the depth component. However, in many application scenarios only one observation of a scene is available at each time. To this end, monocular depth estimation methods such as MRF based formulations [24], methods based on geometry assumptions [25, 26] or non-parametric methods [27, 28]. More recent deep learning based techniques [20, 29] showcase the great progress in ability to estimate disparities and depth maps from a single image. In our work we use MIDAS [20] a recently proposed method based on a transformer architecture, which was trained on a very large selection of diverse datasets.

3. Contrastive Learning with PixDepth

Our method builds upon the pixel contrast loss introduced in [1]. This loss is defined on pixel-features extracted from two augmented views of the same image. More precisely, given an input image, two color-augmented crops are generated. The two crops are further resized to a fixed resolution and passed through a regular encoder network and a momentum encoder network [4, 11]. The obtained feature maps (e.g. of size $1024 \times 8 \times 8$) are then projected using a pixel level projector (two 1×1 convolution layers). The output of the pixel projector already contains 2D positional information.

Figure 2 illustrates the features for two crops. The plus signs and circles correspond to the vectors of the first and second representation vectors of the respective first and second views. The idea of [1] is to use all the pairs of pixel-features from the two crops to generate the positive and negative pairs for contrastive learning. Intuitively, overlapping or spatially close locations are used for positive pairs. This is achieved by constructing a positive/negative mask

$$A_{image}(i,j) = \begin{cases} 1, & \text{if } dist(i,j) \leq \mathcal{T} \\ 0, & \text{if } dist(i,j) > \mathcal{T} \end{cases}, \quad (1)$$

where $\mathcal{T}$ is the distance threshold, $dist$ is the normalized (from 0 to 1) euclidean distance between 2D point coordinates, i and j are the indexes of vectors in the first and second view respectively.

The pixel contrast loss functions [1] is then defined as

$$\mathcal{L}_{\text{Pix}}(i) = -\log \frac{\sum\limits_{j \in \Omega_p^i} e^{\cos(\mathbf{x}_i, \mathbf{x}'_j)/\tau}}{\sum\limits_{j \in \Omega_p^i} e^{\cos(\mathbf{x}_i, \mathbf{x}'_j)/\tau} + \sum\limits_{k \in \Omega_n^i} e^{\cos(\mathbf{x}_i, \mathbf{x}'_k)/\tau}}, \quad (2)$$

where i is a first-view pixel that is also present in the second view; Ω_p^i and Ω_n^i are groups of pixels in the second

view that have been designated as being positive (when $A_{image}(i,j) = 1$) and negative (when $A_{image}(i,j) = 0$), respectively, in relation to pixel i. The pixel feature vectors in two views are $\mathbf{x}_i$ and $\mathbf{x}'_j$, and τ is a scalar temperature hyper-parameter. All pixels from the first view that are located at the intersection of the two views are averaged to determine the loss. Similar to the first view, the second view computes and averages the contrastive loss for each pixel j. The average of all image pairs in a mini-batch represents the final loss. The overall framework is illustrated in the upper half of Figure 3.

Note that this contrastive loss forces feature representations corresponding to the same spatial location (or close depending on the choice of $\mathcal{T}$) to be close.

We believe, however, that this training process has two flaws: 1) The authors use a single pixel distance threshold $\mathcal{T}$ in the pairs selection which is a dataset-specific hyper-parameter and is based on a strong assumption that nearby pixels belong to the same object; 2) This logic does not reflect a structural similarity/difference in cropped patches, which may potentially produce contradictory signals, especially on the frontier between objects in one image. The second problem is particularly important and is illustrated in Figure 2. If we look at the vectors distinguished by dark colors, we notice that they belong to two different objects, despite their proximity in the image. PixPro will tend to consider this pair of vectors as positive and this is detrimental to the quality of the representations.

To resolve this, we propose to modify the procedure for assigning pairs of positive or negative vectors so that it can incorporate some additional semantic information. Depth maps are a good solution because changes in depth can be an indicator of object change. We have therefore considered depth maps as an additional assignment filter.

3.1. Incorporating Depth Information

Figure 3 shows a general overview of our method. We enhance [1] by taking crops from the depth maps at the same positions of our two views. After that, we map them to the size of the feature maps (e.g. 7×7) and we calculate the pairwise difference between the pixels of the resized depth crops. In order to prevent crops that are distant in 2D and have similar depth (e.g. two different objects in foreground). We apply this new selection on top of the previous one. Thus, we build a new positive/negative depth mask as:

$$A_{depth}(i,j) = \begin{cases} 1, & \text{if } |depth(i) - depth(j)| \leq \mathcal{T}' \\ 0, & \text{if } |depth(i) - depth(j)| > \mathcal{T}' \end{cases}, \quad (3)$$

where $\mathcal{T}'$ is the threshold on the depth maps and the final selection mask is then obtained by element-wise multiplication of both masks

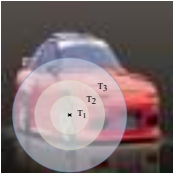$$A_{final}(i,j) = A_{image}(i,j) \times A_{depth}(i,j). \quad (4)$$

Figure 4. An overview of the multi-thresholding setup to account for different scale. We want our representations to be aware of the relative context at different scales. In the left figure, we show how each threshold $\{\mathcal{T}_i\}_{1\leq i\leq n}$ (we arbitrarily chose $n = 3$ and $\mathcal{T}_1 \leq \mathcal{T}_2 \leq \mathcal{T}_3$ for visualization purposes) affects the pairs selection procedure: If we take a crop centered at the X mark, each disk represents the centers of the crops that will form a positive pair with the reference crop. In practice (right), we divide the output of our encoder and projector channel-wise into n equal sized feature maps and then we independently apply Eq.1 on each one using its corresponding threshold. We than average all the contributions for the final loss computations.

The mask A_{final} is then used to define the groups of positive/negative pixels Ω_p^i and Ω_n^i to be used in the loss (2).

In summary, this is a way of incorporating depth prior information into the learning procedure without the need for the encoder to "see" the depth maps at any level of the training. It means that no extra computing power nor time is needed. Simultaneously, the selection of positive and negative pairs is now based on semantic information given by these maps and not only a distance in a projected 2D image, which might end up mixing different objects. As we will see in the next section, the depth information can be found in different sources such as depth maps, disparity maps, etc. Thus, we chose to normalize their values between 0 and 1. It is important to note that we are only interested in differences between depth values and not the actual depth measurements/estimations. Consequently, converting disparity maps to depth maps, for example, is not required.

3.2. Multi-thresholds for Different Scales

The process of pairs selection described above depends on two hyperparameters $\mathcal{T}$ and $\mathcal{T}'$. The choice of these two parameters is tricky and may negatively affect the quality of the learned representations. In fact, objects come in different shapes and sizes depending on datasets and may vary greatly within a single dataset. This leads to the optimal values of $\mathcal{T}$ and $\mathcal{T}'$ being dependent on the pre-training dataset (see Table 1). To alleviate this problem without the need of extra processing, we propose the following procedure (illustrated in Figure 4) that allows different parts of the feature representations to be discriminating at different scales.

Given an output feature map $\mathcal{F}$ of size $C \times H \times W$, we divide it into n disjoint features maps $\{\mathcal{F}_i\}_{1\leq i\leq n}$ of size $\frac{C}{n} \times H \times W$. For each $\mathcal{F}_i$ we apply a different threshold

Pre-traning dataset	$\mathcal{T}'$	mIoU
Imagenet	0.3	74.3
+ Depth with MIDAS	0.5	**76.9**
	0.7	73.9
BoreholeImage	0.3	**74.0**
	0.5	72.8
	0.7	70.7

Table 1. The effect of varying $\mathcal{T}'$ on the fine-tuning performances. The fine-tuning was done on CityScapes for the first 3 rows and The labeled BoreholeImage dataset for the rest.

$\{\mathcal{T}_i\}_{1\leq i\leq n}$ and $\{\mathcal{T}'_i\}_{1\leq i\leq n}$. Every $\mathcal{F}_i$ is then treated as an independent feature vector. This leads to computing n loss terms as in Equation 2:

$$\mathcal{L}_{\text{Pix}}^k(i) = -\log \frac{\sum\limits_{j\in\Omega_p^i} e^{\cos\left(\mathbf{x^k}_i,\mathbf{x^{k'}}_j\right)/\tau}}{\sum\limits_{j\in\Omega_p^{i,k}} e^{\cos\left(\mathbf{x^k}_i,\mathbf{x^{k'}}_j\right)/\tau} + \sum\limits_{m\in\Omega_n^{i,k}} e^{\cos\left(\mathbf{x^k}_i,\mathbf{x^{k'}}_m\right)/\tau}}, \quad (5)$$

where $x_{i/j/m}^k$ represents the kth portion of the feature vector $x_{i/j/m}^k$ (the portion of x that lies in $\mathcal{F}_k$). The groups of pixels $\Omega_{n/p}^{i,k}$ are obtained differently for each k using the threshold $\mathcal{T}_k$. The remaining parameters are the same as in Equation 2 and the total loss is the average of all these contributions

$$\mathcal{L}_{\text{PixMulti}}(i) = \frac{1}{n}\sum_{k=1}^{n} \mathcal{L}_{\text{Pix}}^k(i). \quad (6)$$

In summary, each part of our new feature vector will encode a different scale centered around the object of interest. By selecting thresholds at different scales, we remove the dependency of our method on this hyperparameter. In practice, we choose 3 values for $\mathcal{T}_i$ and $\mathcal{T}'_i$: 0.3, 0.5, 0.7. the values are chosen between 0 and 1 because both the depth maps and the 2D distance defined in [1] are normalized and thus all distances are in the same interval. We believe that choosing 3 values that account for 3 different scales on the image is sufficient for multi-thresholding (see next Section). When we use this method on both hyperparameters in the same time, we use $\{\mathcal{T}_i, \mathcal{T}'_i\}_{1\leq i\leq n}$ for each $\mathcal{F}_\rangle$. This limits the possible combinations of threshold pairs (and thus the feature map division).

4. Experiments

The main motivation for this work is driven by the geological feature segmentation in remote sensing and more specifically petrophysical domain, which prompts a set of experiments on ultrasonic borehole image dataset that we have collected on the course of this study. This is complemented by a set of experiments on datasets from other applications such as semantic segmentation on CityScapes and
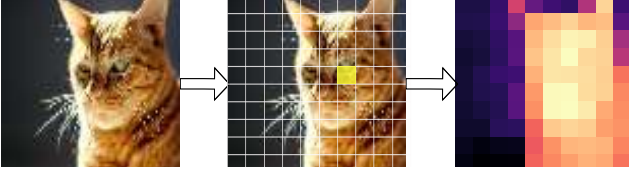
Figure 5. An overview of the representation quality check: an input image (left) is fed into the encoder to produce a feature map (middle dots). Each vector of this feature approximately encodes a patch of the image (a cell of the grid). We select one of the vectors (in yellow) and we measure its similarity to the rest of the vectors. We then plot a similarity heatmap (right).

ScanNet, which helps us validate the generality of our contributions and facilitates the reproducibility of our results. In the following sections we explain the experimental setup and describe in detail the datasets used in our study, configuration of the training set up as well as our approach for visually inspecting the quality of learned representations.

4.1. Data

BoreholeImage dataset used in our study is a basis for geo-mechanical and geological interpretation of hydrocarbon reservoirs based on ultrasonic (US) imaging and plays an important role in Oil and Gas industry context. Borehole US images show meaningful information about petrophysical properties such as breakouts - a type of stress-induced geological features. Thus, the dataset consists of "unwrapped" measures of the amplitude of an ultrasonic wave sent towards a geological formation and received back by an ultrasonic sensor which is then represented in 2D format. This dataset is partially based on images obtained from a public repository UK NDR[1] and on data provided by industrial operators. Figure 6 demonstrates several examples of borehole images that correspond to the UK NDR portion of the dataset. We have used a diverse selection of nearly 27k images collected from 140 different wells. This UK NDR portion of the dataset does not provide labels whilst pixel-level annotations for the breakouts are coming from the remaining portion.

CityScapes is a large-scale dataset that contains a diverse set of binocular stereo-vision sequences recorded in street scenes from 50 different cities, with high quality pixel-level annotations of 5000 frames in addition to a larger set of 20000 weakly annotated frames [30].

ScanNet is an RGB-D video dataset containing 2.5 million views in more than 1500 scans, annotated with 3D camera poses, surface reconstructions, and instance-level semantic segmentations. [22].

4.2. Instance and Semantic Segmentation

We start by pre-training a ResNet [31] encoder on one of the unlabeled datasets depending on the subsequent task
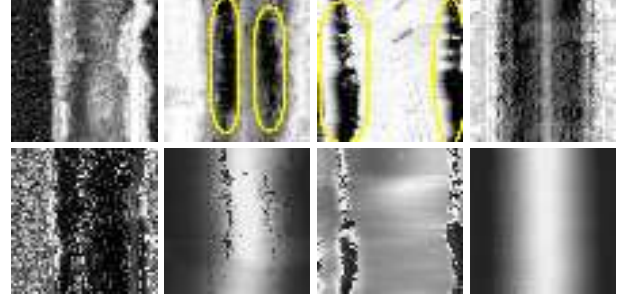

Figure 6. Examples of borehole images from UK NDR dataset used in our studies. First row shows examples of the images based on measured amplitudes of ultrasonic waves. Second row shows corresponding images based on transit times estimation. Breakouts identified on amplitude images are highlighted in yellow.

Pre-training Dataset	Pre-training Method	IoU
Imagenet	supervised	69.1
BoreholeImage	PixPro	72.1
BoreholeImage + Transit Time	PixDepth	**74.0**

Table 2. Breakout segmentation task with BoreholeImage dataset

Pre-training Dataset	Pre-training Method	mIoU
Imagenet	supervised	71.6
Imagenet	PixPro	75.8
Imagenet + depths maps	PixDepth	**76.9**
35% of Midas training set	PixDepth	76.3

Table 3. Indoor segmentation task with CityScapes dataset.

(i.e. indoor scene segmentation on CityScapes and ScanNet or breakouts identification on BoreholeImage dataset). Then, we integrate this pre-trained encoder into a semantic segmentation architecture where we freeze its weights. In this work we employ DeepLabV3 [32] as a segmentation model. Next, we proceed by training the segmentation model on the corresponding labeled portion of the previously used dataset. To demonstrate the improvements due to our approach and for fair comparison we report quantitative results for fully supervised trainings, compared with the original PixPro and our PixDepth pre-trainings.

For the Breakouts segmentation on BoreholeImage dataset we used a ResNet-34 encoder and two splits of the dataset for pre-training and fine-tuning. The results are shown in Table 2. For the experiments on the CityScapes data we also used a ResNet-34 encoder but we perform two variations of the experiment. In the first variation we used ImageNet-1K as a pre-training dataset with monocular depths maps computed using MIDAS [20]. In the second variation we remove the dependency on a pre-trained monocular depth estimation network (MIDAS) and instead directly use a subset of the disparity maps that were used to train it. The corresponding results are shown in Table 3. For the experiments on the ScanNet dataset we used a ResNet-18 encoder and the same dataset for pre-training and fine-tuning. The results are shown in Table 4.

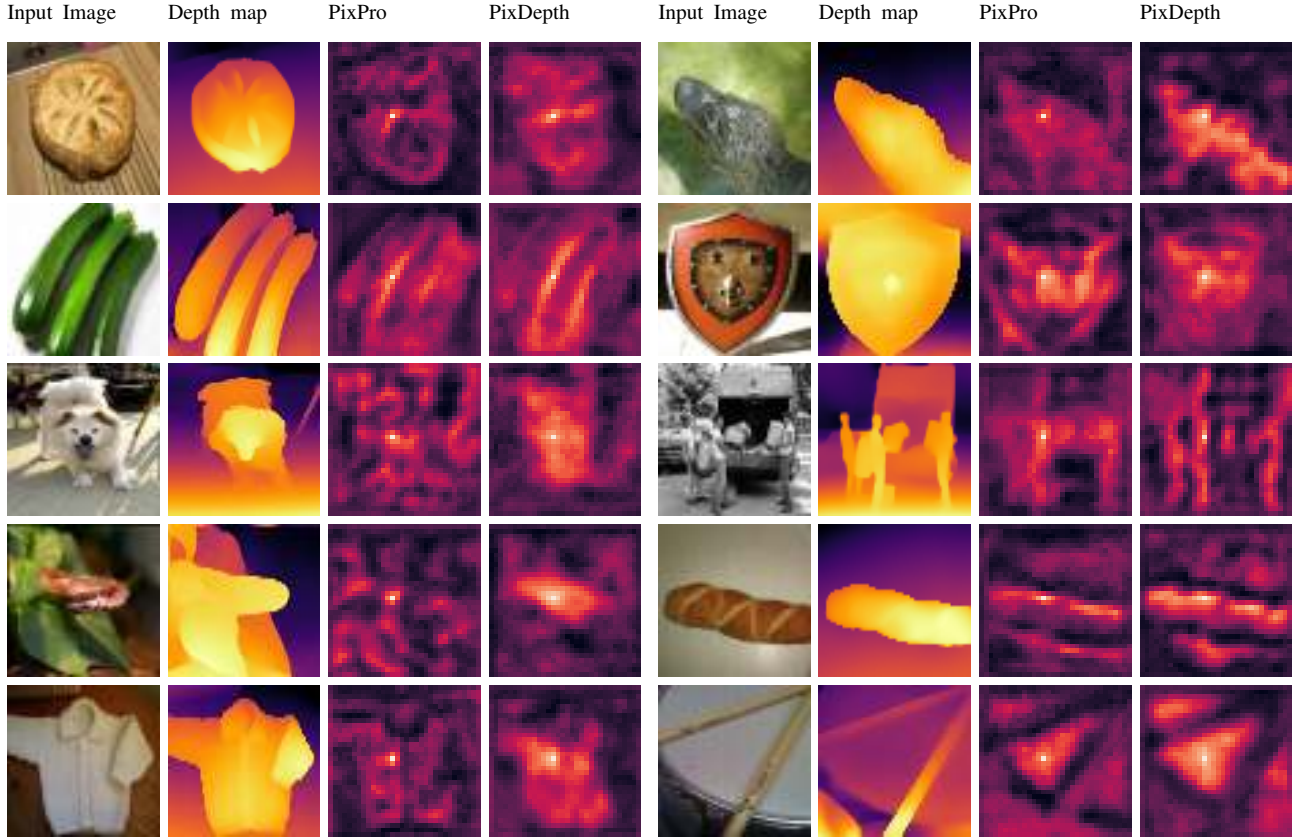
[1] https://ndr.ogauthority.co.uk

Figure 7. Comparison of the quality of the learned representations. The pixels in the similarity maps correspond to the reference feature vector, which is compared to all the rest of the vectors forming the features map. Note that these images are part of the validation set (unseen during training), and that the depth map is given for comparison purposes, it is not fed into the encoder.

Pre-training Method	IoU
Supervised	44.7
PixPro	46.3
Pri3D	48.1
PixDepth	**49.7**

Table 4. Indoor segmentation task with ScanNet dataset.

Implementation details In the pre-training stage we used the SGD optimizer with a momentum of 0.9, a linear warm-up of 20 epochs and a cosine annealing scheduler for the rest of the training. We also fixed the maximum learning rate at 0.1 and trained for a total 500 epochs and a batch size of 256 using 8 NVIDIA V100 GPUs for ImageNet and ScanNet, and 800 epochs and a batch size of 128 using 2 NVIDIA V100s for US BoreholeImage. In the fine-tuning stage we trained for 80 epochs with an early stopping using the Adam optimizer with a base learning rate of 0.001 and we used a cosine annealing scheduler. We fixed the batch size at 32 for all the segmentation tasks.

Representation Quality Evaluation To visually evaluate and compare the quality of the learned representations with PixDepth and PixPro, we propose the setup shown in Figure 5. We take an input image and we resize it to a large

size (*e.g.* 896×896) so that the output feature map $\mathcal{F}$ can be large enough (*e.g.* 32×32). Each vector of this map represents a patch in the original image (the size of this patch being approximately the size of the network's receptive field). We then select a feature vector $\mathcal{F}_{i,j}$ corresponding to a distinct area in the image (*e.g.* the head of a the cat) and we measure the cosine similarity of this vector with all the other vectors in $\mathcal{F}$ and we build a similarity map. Some examples from ImageNet are shown in Figure 7 and in the Supplementary Material. For PixDepth, we can see that the features belonging to the same object are more spatially coherent and more contrasted with the background. This indicates that feature vectors that constitute the objects are similar to each other and different from those representing the background. This experiment provides a qualitative view of the effect of using the depth prior in our method.

4.3. Results and Discussion

Table 3 shows the results obtained on the breakout segmentation task. We observe that using the transit time with PixDepth improves the results over PixPro by 1.9% and by 4.9% over the supervised baseline. The transit time in-
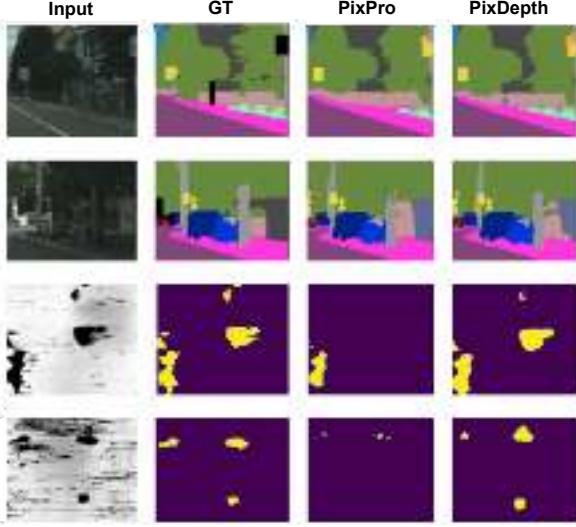
Figure 8. Some qualitative examples for segmentation results on CityScapes (first and second rows) and Breakout Segmentation (third and fourth rows).

formation helped the network to distinguish between real breakouts and imperfections in the images leading to better segmentation scores. Table 2 shows the results of Semantic Segmentation on CityScapes. Again, our method exhibits an improvement by 1.1% over PixPro and more than 5% over the supervised baseline. We also observe that the results with the change of the pre-training dataset and the exogenous data it comes with (a fraction of the MI-DAS training set with ground truth disparity/depth maps) remained better than PixPro by 0.5%. Table 4 compares our PixDepth pre-training with PixPro and Pri3D on Scan-Net. We observe an improvement of mean IoU by 3.4% over PixPro and 0.6% over Pri3D. Apart from confirming the improvement over PixPro on another task, these results show that our method, which uses less trainable parameters and a simpler procedure, achieves comparable results to Pri3D. We show also in Figure 8 some of the examples where we can qualitatively see that our method performs better than PixPro. This supports our claim that the prior depth information used for pairs selection boosts the quality of learned representations. Additional experiments with larger models (ResNet-50) were not possible in our study, but can be considered for future work.

4.4. Ablation Study

To demonstrate the impact of each part of the PixDepth, we conducted an ablation study on the multi-thresholding for $\mathcal{T}_i$ and $\mathcal{T}'_i$ on the BoreholeImage dataset. Results are shown in Table 5. This suggests that multi-thresholding does improve slightly the mIoU on the breakout segmentation task. But most importantly it gives a score closer to the ones obtained with the best parameters when set to a single threshold. However, using more than 3 thresholds gives

$\mathcal{T}'$	$\mathcal{T}$	mIoU
0.1	0.5	70.3
0.3		**74.0**
0.5		71.4
0.7		69.8
0.3	0.1	71.6
	0.3	72.1
	0.5	**74.0**
	0.7	72.4
	{0.3, 0.7}	73.7
	{0.3, 0.5, 0.7}	73.9
{0.3, 0.7}	0.5	72.6
{0.3, 0.5, 0.7}		**74.4**
{0.1, 0.3, 0.5, 0.7}		73.7
{0.3, 0.5, 0.7 }	{0.3, 0.5, 0.7}	<u>74.2</u>

Table 5. The effect of varying the thresholds $\mathcal{T}$ and $\mathcal{T}'$ on the segmentation performances on the breakout detection task. Multiple values indicate multi-thresholding.

worse results. This is likely due to the fact that using too many thresholds results in the division of the feature map into smaller fragments. These smaller features become less expressive and thus perform worse at the contrastive tasks.

5. Conclusion

We have introduced PixDepth, an improvement to an existing CL based method for pixel-level representation learning. The main goal is to leverage prior depth information, which can be inferred using a pre-trained monocular depth estimation network or calculated/collected along with the images. We show that this idea along with the use of multiple thresholds (to take into account different scales in learned representations) results in better segmentation scores and improves the coherence of learned representations. We have supported our findings by multiple experiments in different datasets such as segmentation of indoor scenes on CityScapes and ScanNet and geological features identification on BoreholeImage dataset. Future work will focus on exploring the use PixDepth with more modern backbone architectures (based on transformers) and its impact on the pixel-level downstream tasks.

References

[1] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *IEEE CVPR*, pages 16684–16693, 2021.

[2] Ji Hou, Saining Xie, Benjamin Graham, Angela Dai, and Matthias Nießner. Pri3d: Can 3d priors help 2d representation learning? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5693–5702, 2021.

[3] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE CVPR*, pages 9729–9738, 2020.

[5] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[6] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[7] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pages 4182–4192. PMLR, 2020.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020.

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33:21271–21284, 2020.

[12] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Toronto, ON, Canada, 2009.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[18] Shikun Liu, Shuaifeng Zhi, Edward Johns, and Andrew J Davison. Bootstrapping semantic segmentation with regional contrast. *arXiv preprint arXiv:2104.04465*, 2021.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[20] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[21] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[22] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[23] Yutong Bai, Xinlei Chen, Alexander Kirillov, Alan Yuille, and Alexander C Berg. Point-level region contrast for object detection pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16061–16070, 2022.

[24] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008.

[25] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, page 577–584, New York, NY, USA, 2005. Association for Computing Machinery.

[26] Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Learning monocular depth estimation with unsupervised trinocular assumptions. In *2018 International conference on 3d vision (3DV)*, pages 324–333. IEEE, 2018.

[27] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014.

[28] Huihui Xu and Mingyan Jiang. Non-parametric learning-based depth estimation from a single image in the fourier domain. In *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, pages 334–338, 2018.

[29] S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9685–9694, 2021.

[30] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *IEEE CVPR Workshop on the Future of Datasets in Vision*, volume 2, 2015.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[32] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.