MIN-MAX ENTROPY FOR WEAKLY SUPERVISED POINTWISE LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pointwise localization allows more precise localization and accurate interpretability, compared to bounding box, in applications where objects are highly unstructured such as in medical domain. In this work, we focus on weakly supervised localization (WSL) where a model is trained to classify an image and localize regions of interest at pixel-level using only global image annotation. Typical convolutional attentions maps are prune to high false positive regions. To alleviate this issue, we propose a new deep learning method for WSL, composed of a localizer and a classifier, where the localizer is constrained to determine relevant and irrelevant regions using conditional entropy (CE) with the aim to reduce false positive regions. Experimental results on a public medical dataset and two natural datasets, using Dice index, show that, compared to state of the art WSL methods, our proposal can provide significant improvements in terms of image-level classification and pixel-level localization (low false positive) with robustness to overfitting. A public reproducible PyTorch implementation is provided.

1 INTRODUCTION

Pointwise localization is an important task for image understanding, as it provides crucial clues to challenging visual recognition problems, such as semantic segmentation, besides being an essential and precise visual interpretability tool. Deep learning methods, and particularly convolutional neural networks (CNNs), are driving recent progress in these tasks. Nevertheless, despite their remarkable performance, their training requires large amounts of labeled data, which is time consuming and prone to observer variability. To overcome this limitation, weakly supervised learning (WSL) has emerged recently as a surrogate for extensive annotations of training data (Zhou, 2017). WSL involves scenarios where training is performed with inexact or uncertain supervision. In the context of pointwise localization or semantic segmentation, weak supervision typically comes in the form of image level tags (Kervadec et al., 2019; Kim et al., 2017; Pathak et al., 2015; Teh et al., 2016; Wei et al., 2017), scribbles (Lin et al., 2016; Tang et al., 2018) or bounding boxes (Khoreva et al., 2017).

Current state-of-the-art WSL methods rely heavily on pixelwise activation maps produced by a CNN classifier at the image level, thereby localizing regions of interest (Zhou et al., 2016). Furthermore, this can be used as an *interpretation* of the model's decision (Zhang & Zhu, 2018). The recent literature abounds of WSL works that relax the need of dense and prohibitively time consuming pixel-level annotations (Rony et al., 2019). Bottom-up methods rely on the input signal to locate regions of interest, including spatial pooling techniques over activation maps (Durand et al., 2017; Oquab et al., 2015; Sun et al., 2016; Zhang et al., 2018b; Zhou et al., 2016), multi-instance learning (Ilse et al., 2018) and attend-and-erase based methods (Kim et al., 2017; Li et al., 2018; Pathak et al., 2015; Singh & Lee, 2017; Wei et al., 2017). While these methods provide pointwise localization, the models in (Bilen & Vedaldi, 2016; Kantorov et al., 2016; Shen et al., 2018; Tang et al., 2017; Wan et al., 2018) predict a bounding box instead, i.e., perform weakly supervised object detection. Inspired by human visual attention, top-down methods rely on the input signal and a selective backward signal to determine the corresponding region of interest. This includes special feedback layers (Cao et al., 2015), backpropagation error (Zhang et al., 2018a) and Grad-CAM (Chattopadhyay et al., 2018; Selvaraju et al., 2017).

In many applications, such as in medical imaging, region localization may require high precision such as cells, boundaries, and organs localization; regions that have an unstructured shape, and different scale that a bounding box may not be able to localize precisely. In such cases, a pointwise localization can be more suitable. The illustrative example in Fig.1 (bottom row) shows a typical case where using a bounding box to localize the glands is clearly problematic. This motivates us to consider predicting a mask instead of a bounding box. Consequently, our latter choice of evaluation datasets is constrained by the availability of both global image annotation for training and pixel-level annotation for evaluation. In this work, we focus on the case where there is one object of interest in the image.

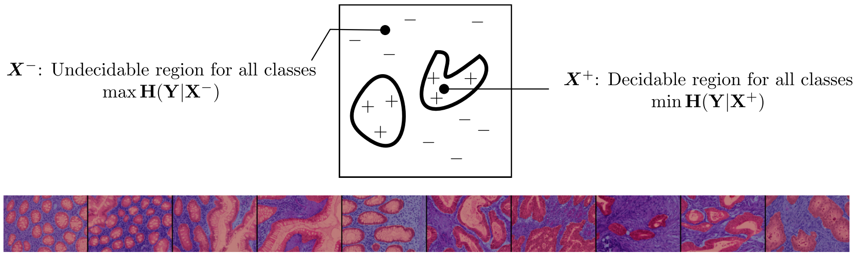
$\boldsymbol{X}^-$: Undecidable region for all classes
$\max \mathbf{H}(\mathbf{Y}|\mathbf{X}^-)$
$\boldsymbol{X}^+$: Decidable region for all classes
$\min \mathbf{H}(\mathbf{Y}|\mathbf{X}^+)$

Figure 1: **Top row**: Intuition of our proposal. A decidable region covers all the discriminative parts, while an undecidable region covers all the non-discriminative parts. (See Sec.3 for notation.) **Bottom row**: Example of test samples from different classes of the GlaS dataset, where the annotated glands are regions of interest, and the remaining tissue is noise/background. Note the glands' different shapes, size, context, and multiple instances aspect.

Often, within an agnostic-class setup, input image contains the object of interest among other irrelevant parts (noise, background). Most the aforementioned WSL methods do not consider such prior, and feed the entire image to the model. In such scenario, (Wan et al., 2018) argue that there is an *inconsistency* between the classification loss and the task of WSL; and that typically the optimization may reach sub-optimal solutions with considerable randomness in them, leading to high false positive localization. False positive localization is aggravated when a class appears in different and random shape/structure, or may have relatively similar texture/color to the irrelevant parts driving the model to confuse between both parts. False positive regions can be problematic in critical domains such as medical applications where interpretability plays a central role in trusting and understanding an algorithm's prediction. To address this important issue, and motivated by the importance of using prior knowledge in learning to alleviate overfitting when training using few samples (Belharbi et al., 2017; Krupka & Tishby, 2007; Mitchell, 1980; Yu et al., 2007), we propose to use the aforementioned prior in order to favorite models with low false positive localization. To this end, we constrain the model to learn to localize both relevant and irrelevant regions *simultaneously* in an end-to-end manner within a WSL scenario, where only image-level labels are used for training. We model the relevant (discriminative) regions as the *complement* of the irrelevant (non-discriminative) regions (Fig.1). Our model is composed of two sub-models: (1) a *localizer* that aims to localize both types of regions by predicting a *latent mask*, (2) and a *classifier* that aims to classify the visible content of the input image through the latent mask. The *localizer* is driven through CE (Cover & Thomas, 2006) to simultaneously identify (1) relevant regions where the *classifier* has high confidence with respect to the image label, (2) and irrelevant regions where the classifier is being unable to decide which image label to assign. This modeling allows the discriminative regions to pop out and be used to assign the corresponding image label, while suppressing non-discriminative areas, leading to more reliable predictions. In order to localize complete discriminative regions, we extend our proposal by training the localizer to recursively erase discriminative parts during *training* only. To this end, we propose a consistent recursive erasing algorithm that we incorporate within the backpropagation. At each recursion, and within the backpropagation, the algorithm localizes the most discriminative region; stores it; then erases it from the input image. At the end of the final recursion, the model has gathered a large extent of the object of interest that is fed next to the classifier. Thus, our model is driven to localize complete relevant regions while discarding irrelevant regions, resulting in more reliable region localization. Moreover, since the discriminative parts are allowed to be extended over different instances, our proposal handles multi-instances intrinsically.

The main contribution of this paper is a new deep learning framework for WSL at pixel level. The framework is composed of two sequential sub-networks where the first one localizes regions of interest, whereas the second classifies them. Based on CE, the end-to-end training of the framework allows to incorporate prior knowledge that, an image is more likely to contain relevant and irrelevant regions. Throughout the CE measured at the classifier level, the localizer is driven to localize relevant regions (with low CE) and irrelevant regions (with high CE). Such localization is achieved with the main goal of providing a more interpretable and reliable regions of interest with low false positive localization. This paper also contributes a consistent recursive erasing algorithm that is incorporated within backpropagation, along with a practical implementation in order to obtain complete discriminative regions. Finally, we conduct an extensive series of experiments on three public image datasets (medical and natural), where the results show the effectiveness of the proposed approach in terms of pointwise localization (measured with Dice index) while maintaining competitive accuracy for image-level classification.

2 Background on WSL

In this section, we briefly review state of the art of WSL methods, divided into two main categories, aiming at pointwise localization of regions of interest using only image-level labels as supervision. (1) Fully convolutional networks with spatial pooling have shown to be effective to obtain localization of discriminative regions (Durand et al., 2017; Oquab et al., 2015; Sun et al., 2016; Zhang et al., 2018b; Zhou et al., 2016). Multi-instance learning methods have been used within an attention framework to localize regions of interest (Ilse et al., 2018). (Singh & Lee, 2017) propose to hide randomly large patches in training image in order to force the network to seek other discriminative regions to recover large part of the object of interest, since neural networks often provide small and most discriminative regions of object of interest (Kim et al., 2017; Singh & Lee, 2017; Zhou et al., 2016). (Wei et al., 2017) use the attention map of a trained network to erase the most discriminative part of the original image. (Kim et al., 2017) use two-phase learning stage where the attention maps of two networks are combined to obtain a complete region of the object. (Li et al., 2018) propose a two-stage approach where the first network classifies the image, and provides an attention map of the most discriminative parts. Such attention is used to erase the corresponding parts over the input image, then feed the resulting erased image to a second network to make sure that there is no discriminative parts left. (2) Inspired by the human visual attention, top-down methods were proposed. In (Simonyan et al., 2014; Springenberg et al., 2015; Zeiler & Fergus, 2014), backpropagation error is used in order to visualize saliency maps over the image for the predicted class. In (Cao et al., 2015), an attention map is built to identify the class relevant regions using feedback layer. (Zhang et al., 2018a) propose Excitation backprop that allows to pass along top-down signals downwards in the network hierarchy through a probabilistic framework. Grad-CAM (Selvaraju et al., 2017) generalize CAM (Zhou et al., 2016) using the derivative of the class scores with respect to each location on the feature maps; it has been furthermore generalized in (Chattopadhyay et al., 2018). In practice, top-down methods are considered as visual explanatory tools, and they can be overwhelming in term of computation and memory usage even during inference.

While the aforementioned approaches have shown great success mostly with natural images, they still lack a mechanism for modeling what is relevant and irrelevant within an image which is important to reduce false positive localization. This is crucial for determining the reliability of the regions of interest. Erase-based methods (Kim et al., 2017; Li et al., 2018; Pathak et al., 2015; Singh & Lee, 2017; Wei et al., 2017) follow such concept where the non-discriminative parts are suppressed through constraints, allowing only the discriminative ones to emerge. Explicitly modeling negative evidence within the model has shown to be effective in WSL (Azizpour et al., 2015; Durand et al., 2017; 2016; Parizi et al., 2015).

Our proposal is related to (Behpour et al., 2019; Wan et al., 2018) in using entropy-measure to explore the input image. However, while (Wan et al., 2018) defines an entropy over the bounding boxes' position to minimize its variance, we define a CE over the classifier to be low over discriminative regions, while being high over non-discriminative ones. Our recursive erasing algorithm follows general erasing and mining techniques (Kim et al., 2017; Li et al., 2018; Singh & Lee, 2017; Wan et al., 2018; Wei et al., 2017), but places more emphasis on mining consistent regions, and being performed on the fly during backpropagation. For instance, compared to (Wan et al., 2018), our algorithm attempts to expand regions of interest, accumulate consistent regions while erasing, provide automatic mechanism to stop erasing over samples independently from each other. However (Wan et al., 2018) aims to locate multiple instances without erasing, and use manual/empirical threshold for assigning confidence to boxes. Our proposal can be seen as a *guided* dropout (Srivastava et al., 2014).

While standard dropout is applied over a given input image to *randomly* zero out pixels, our proposed approach *seeks* to zero out irrelevant pixels and keep only the discriminative ones that support the image label. From this perspective, our proposal mimics a *discriminative gate* that inhibits irrelevant and noisy regions while allowing only informative and discriminative regions to pass through the gate.

3 THE MIN-MAX ENTROPY FRAMEWORK FOR WSL

Notations and definitions: Let us consider a set of training samples $\mathbb{D} = \{(\boldsymbol{X}_i, y_i)\}_{i=1}^{n}$ where $\boldsymbol{X}_i$ is an input image with depth d, height h, and width w; a realization of the discrete random variable $\mathbf{X}$ with support set $\mathcal{X}$; y_i is the image-level label (i.e., image class), a realization of the discrete random variable $\mathbf{y}$ with support set $\mathcal{Y} = \{1, \cdots, c\}$. We define a *decidable* region[1] of an image as any informative part of the image that allows predicting the image label. An *undecidable* region is any noisy, uninformative, and irrelevant part of the image that does not provide any indication nor support for the image class. To model such definitions, we consider a binary mask $\boldsymbol{M}^+ \in \{0, 1\}^{h \times w}$ where a location (r, z) with value 1 indicates a decidable region, otherwise it is an undecidable region. We model the decidability of a given location (r, z) with a binary random variable $\mathbf{M}$. Its realization is $\boldsymbol{m}$, and its conditional probability $p_{\mathbf{m}}$ over the input image is defined as follows,

$$p_{\mathbf{M}}(\mathbf{m} = 1 | \boldsymbol{X}, (r, z)) = \begin{cases} 1 & \text{if } \boldsymbol{X}(r, z) \text{ is a decidable region },\\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

We note $\boldsymbol{M}^- \in \{0, 1\}^{h \times w} = \boldsymbol{U} - \boldsymbol{M}^+$ a binary mask indicating the undecidable region, where $\boldsymbol{U} = \{1\}^{h \times w}$. We consider the undecidable region as the *complement* of the decidable one. We can write: $\|\boldsymbol{M}^+\|_0 + \|\boldsymbol{M}^-\|_0 = h \times w$, where $\|\cdot\|_0$ is the l_0 norm. Following such definitions, an input image $\boldsymbol{X}$ can be decomposed into two images as $\boldsymbol{X} = \boldsymbol{X} \odot \boldsymbol{M}^+ + \boldsymbol{X} \odot \boldsymbol{M}^-$, where $(\cdot \odot \cdot)$ is the Hadamard product. We note $\boldsymbol{X}^+ = \boldsymbol{X} \odot \boldsymbol{M}^+$, and $\boldsymbol{X}^- = \boldsymbol{X} \odot \boldsymbol{M}^-$. $\boldsymbol{X}^+$ inherits the image-level label of $\boldsymbol{X}$. We can write the pair $(\boldsymbol{X}_i^+, y_i)$ in the same way as $(\boldsymbol{X}_i, y_i)$. We note by $\boldsymbol{R}_i^+$, and $\boldsymbol{R}_i^-$ as the respective approximation of $\boldsymbol{M}_i^+$, and $\boldsymbol{M}_i^-$. We are interested in modeling the true conditional distribution $p(\mathbf{Y}|\mathbf{X})$ where $p(\mathbf{Y} = y_i | \mathbf{X} = \boldsymbol{X}_i) = 1$. $\hat{p}(\mathbf{Y}|\mathbf{X})$ is its estimate. Following the previous discussion, predicting the image label depends only on the decidable region, i.e., $\boldsymbol{X}^+$. Thus, knowing $\boldsymbol{X}^-$ does not add any knowledge to the prediction, since $\boldsymbol{X}^-$ does not contain any information about the image label. This leads to: $p(\mathbf{Y}|\mathbf{X} = \boldsymbol{X}) = p(\mathbf{Y}|\mathbf{X} = \boldsymbol{X}^+)$. As a consequence, the image label is conditionally independent of $\boldsymbol{X}^-$ provided $\boldsymbol{X}^+$ (Koller & Friedman, 2009): $p \models \mathbf{Y} \perp \mathbf{X}^- | \mathbf{X}^+$, where $\mathbf{X}^+, \mathbf{X}^-$ are the random variables modeling the decidable and the undecidable regions, respectively. In the following, we provide more details on how to exploit such conditional independence property in order to estimate $\boldsymbol{R}^+$ and $\boldsymbol{R}^-$.

Min-max entropy: We consider modeling the uncertainty of the model prediction over decidable, or undecidable regions using conditional entropy (CE). Let us consider the CE of $\mathbf{Y}|\mathbf{X} = \boldsymbol{X}^+$, denoted $\mathbf{H}(\mathbf{Y}|\mathbf{X} = \boldsymbol{X}^+)$ and computed as (Cover & Thomas, 2006),

$$\mathbf{H}(\mathbf{Y}|\mathbf{X} = \boldsymbol{X}^+) = -\sum_{y \in \mathcal{Y}} \hat{p}(\mathbf{Y}|\mathbf{X} = \boldsymbol{X}^+) \ \log \hat{p}(\mathbf{Y}|\mathbf{X} = \boldsymbol{X}^+) \ . \tag{2}$$

Since the model is required to be certain about its prediction over $\boldsymbol{X}^+$, we constrain the model to have low entropy over $\boldsymbol{X}^+$. Eq.2 reaches its minimum when the probability of one of the classes is certain, i.e., $\hat{p}(\mathbf{Y} = y | \mathbf{X} = \boldsymbol{X}^+) = 1$ (Cover & Thomas, 2006). Instead of directly minimizing Eq.2, and in order to ensure that the model predicts the correct image label, we cast a supervised learning problem using the cross-entropy between p and $\hat{p}$ using the image-level label of $\boldsymbol{X}$ as a supervision,

$$\mathbf{H}(p_i, \hat{p}_i)^+ = -\sum_{y \in \mathcal{Y}} p(\mathbf{Y} = y | \mathbf{X} = \boldsymbol{X}_i^+) \ \log \hat{p}(\mathbf{Y} = y | \mathbf{X} = \boldsymbol{X}_i^+) = -\log \hat{p}(y_i | \boldsymbol{X}_i^+) \ . \tag{3}$$

Eq.3 reaches its minimum at the same conditions as Eq.2 with the true image label as a prediction. We note that Eq.3 is the negative log-likelihood of the sample $(\boldsymbol{X}_i, y_i)$. In the case of $\boldsymbol{X}^-$, we consider the CE of $\mathbf{Y}|\mathbf{X} = \boldsymbol{X}^-$, denoted $\mathbf{H}(\mathbf{Y}|\mathbf{X} = \boldsymbol{X}^-)$ and computed as,

$$\mathbf{H}(\mathbf{Y}|\mathbf{X} = \boldsymbol{X}^-) = -\sum_{y \in \mathcal{Y}} \hat{p}(\mathbf{Y}|\mathbf{X}^-) \log \hat{p}(\mathbf{Y}|\mathbf{X}^-) \ . \tag{4}$$

[1] In this context, the notion of *region* indicates one pixel.

Over irrelevant regions, the model is required to be *unable to decide* which image class to predict since there is no evidence to support any class. This can be seen as a high uncertainty in the model decision. Therefore, we consider maximizing the entropy of Eq.4. The later reaches its maximum at the uniform distribution (Cover & Thomas, 2006). Thus, the inability of the model to decide is reached since each class is *equiprobable*. An alternative to maximizing Eq.4 is to use a supervised target distribution since it is already known (i.e., uniform distribution). To this end, we consider q as a uniform distribution, $q(\mathbf{Y} = y|\mathbf{X} = \boldsymbol{X}_i^-) = 1/c \ , \forall y \in \mathcal{Y}$, and caste a supervised learning setup using a cross-entropy between q and $\hat{p}$ over $\boldsymbol{X}^-$,

$$\mathbf{H}(q_i, \hat{p}_i)^- = -\sum_{y \in \mathcal{Y}} q(\mathbf{Y} = y|\mathbf{X} = \boldsymbol{X}_i^-) \ \log \hat{p}(\mathbf{Y} = y|\mathbf{X} = \boldsymbol{X}_i^-) = -\frac{1}{c}\sum_{y \in \mathcal{Y}} \log \hat{p}(y|\boldsymbol{X}_i^-) \ . \quad (5)$$

The minimum of Eq.5 is reached when $\hat{p}(\mathbf{Y}|\mathbf{X} = \boldsymbol{X}_i^-)$ is uniform, thus, Eq.4 reaches its maximum. Now, we can write the total training loss to be minimized as,

$$\min \ \mathop{\mathbb{E}}_{(\boldsymbol{X}_i, y_i) \in \mathbb{D}} \left[\mathbf{H}(p_i, \hat{p}_i)^+ + \mathbf{H}(q_i, \hat{p}_i)^-\right] \ . \quad (6)$$

The posterior probability $\hat{p}$ is modeled using a classifier $\mathcal{C}(. \ , \boldsymbol{\theta}_{\mathcal{C}})$ with a set of parameters $\boldsymbol{\theta}_{\mathcal{C}}$; it can operate either on $\boldsymbol{X}_i^+$ or $\boldsymbol{X}_i^-$. The binary mask $\boldsymbol{R}_i^+$ (and $\boldsymbol{R}_i^-$) is learned using another model $\mathcal{M}(\boldsymbol{X}_i; \ \boldsymbol{\theta}_{\mathcal{M}})$ with a set of parameters $\boldsymbol{\theta}_{\mathcal{M}}$. In this work, both models are based on neural networks (fully convolutional networks (Long et al., 2015) in particular). The networks $\mathcal{M}$ and $\mathcal{C}$ can be seen as two parts of one single network $\mathcal{G}$ that localizes regions of interest using a binary mask, then classifies their content. Fig.2 illustrates the entire model. Due to the depth of $\mathcal{G}$, $\mathcal{M}$ receives its supervised gradient based only on the error made by $\mathcal{C}$. In order to boost the supervised gradient at $\mathcal{M}$, and provide it with more hints to select the most discriminative regions with respect to the image class, we consider using a secondary classification task at the output of $\mathcal{M}$ to classify the input $\boldsymbol{X}$, following (Lee et al., 2015). $\mathcal{M}$ computes the posterior probability $\hat{p}^s(\boldsymbol{Y}|\boldsymbol{X})$ which is another estimate of $p(\boldsymbol{Y}|\boldsymbol{X})$. To this end, $\mathcal{M}$ is trained to minimize the cross-entropy between p and $\hat{p}^s$,

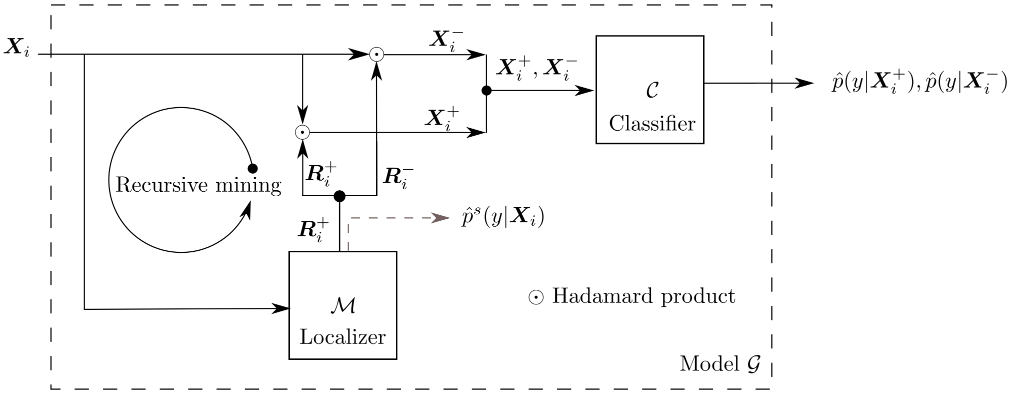

Figure 2: Our proposed method. The recursive mining is done only during training (See Sec.A.1).

$$\mathbf{H}(p_i, \hat{p}_i^s) = -\log \hat{p}^s(\mathbf{Y} = y_i|\mathbf{X} = \boldsymbol{X}_i) \ . \quad (7)$$

The total training loss to minimize is formulated as,

$$\min_{\{\boldsymbol{\theta}_{\mathcal{M}}, \boldsymbol{\theta}_{\mathcal{C}}\}} \ \mathop{\mathbb{E}}_{(\boldsymbol{X}_i, y_i) \in \mathbb{D}} \left[\mathbf{H}(p_i, \hat{p}_i)^+ + \mathbf{H}(q_i, \hat{p}_i)^- + \mathbf{H}(p_i, \hat{p}_i^s)\right] \ . \quad (8)$$

Mask computation and recursive erasing: The mask $\boldsymbol{R}^+$ is computed using the last feature maps of $\mathcal{M}$ which contains high abstract descriminative activations. We note such feature maps by a tensor $\boldsymbol{A}_i \in \mathbb{R}^{c \times h' \times w'}$ that contains a spatial map for each class. $\boldsymbol{R}_i^+$ is computed by aggregating the spatial activation of all the classes as, $\boldsymbol{T}_i = \sum_{k=1}^{c} \hat{p}^s(\mathbf{Y} = k|\mathbf{X} = \boldsymbol{X}_i) \ \boldsymbol{A}_i(k)$, where $\boldsymbol{T}_i \in \mathbb{R}^{h' \times w'}$ is the continuous downsampled version of $\boldsymbol{R}_i^+$, and $\boldsymbol{A}_i(k)$ is the feature map of the class k of the input $\boldsymbol{X}_i$. At convergence, the posterior probability of the winning class is pushed toward 1 while the rest is pushed down to 0. This leaves only the feature map of the winning classe. $\boldsymbol{T}_i$ is upscaled using *interpolation* (Sec.A.2) to $\boldsymbol{T}\!\uparrow_i \in \mathbb{R}^{h\times w}$ which has the same size as the input $\boldsymbol{X}$, then pseudo-thresholded using a sigmoid function to obtain a pseudo-binary $\boldsymbol{R}_i^+$,

$$p_{\mathbf{M}}(\mathbf{m}=1|\boldsymbol{X}_i,(r,z)) = 1/(1+\exp(-\omega \times (\boldsymbol{T}\!\uparrow_i(r,z)-\sigma'))) \ , \tag{9}$$

where ω is a constant scalar that ensures that the sigmoid approximately equals to 1 when $\boldsymbol{T}\!\uparrow_i(r,z)$ is larger than σ', and approximately equals to 0 otherwise. At this point, $\boldsymbol{R}^-$ may still contain discriminative regions. To alleviate this issue, we propose a learning incremental and recursive erasing approach that drives $\mathcal{M}$ to mine complete discriminative regions. The mining algorithm is consistent, sample dependent, it has a maximum recursion depth u, associates trust coefficients to each recursion, integrated within the backpropagation, operates only during training, and has a practical implementation. Due to space limitation, we left it in the supplementary material (Sec.A.1).

4 Results and analysis

Our experiments focus simultaneously on classification and pointwise localization tasks. Thus, we consider datasets that provide both image and pixel-level labels for evaluation. Particularly, the following three datasets are considered: GlaS in medical domain, and CUB-200-2011 and Oxford flower 102 on natural scene images. (1) GlaS dataset, one of the rare medical datasets that fits our scenario (Rony et al., 2019), was provided in the 2015 Gland Segmentation in Colon Histology Images Challenge Contest[2] (Sirinukunwattana et al., 2017). The main task of the challenge is gland segmentation of microscopic images. However, image-level labels were provided as well. The dataset is composed of 165 images derived from 16 Hematoxylin and Eosin (H&E) histology sections of two grades (classes): benign, and malignant. It is divided into 84 samples for training, and 80 samples for test. Images have a high variation in term of gland shape/size, and overall H&E stain. In this dataset, the glandes are the regions of interest that the pathologists use to prognosis the image grading of being benign or malignant. (2) CUB-200-2011 dataset[3] (Wah et al., 2011) is a dataset for bird species with $11,788$ samples and 200 species. Preliminary experiments were conducted on small version of this datatset where we selected randomly 5 species and build a small dataset with 150 samples for training, and 111 for test; referred to in this work as CUB5. The entire dataset is referred to as CUB. In this dataset, the regions of interest are the birds. (3) Oxford flower 102[4] (Nilsback & Zisserman, 2007) datatset is collection of 102 species (classes) of flowers commonly occurring in United Kingdom; referred to here as OxF. It contains a total of $8,189$ samples. We used the provided splits for training ($1,020$ samples), validation ($1,020$ samples) and test ($6,149$ samples) sets. Regions of interest are the flowers which were segmented automatically. In GlaS, CUB5 and CUB datasets, we randomly select 80% of training samples for effective training, and 20% for validation to perform early stopping. We provide in our public code the used splits and the deterministic code that generated them for the different datasets.

In all the experiments, image-level labels are used during training/evaluation, while pixel-level labels are used exclusively during evaluation. The evaluation is conducted at two levels: at image-level where the classification error is reported, and at the pixel-level where we report F1 score (Dice index) over the foreground (region of interest), referred to as F1$^+$. When dealing with binary data, F1 score is equivalent to Dice index. We report as well the F1 score over the background, referred to as F1$^-$, in order to measure how well the model is able to identify irrelevant regions. We compare our method to different methods of WSL. Such methods use similar pre-trained backbone (resent18 (He et al., 2016)) for feature extraction and differ mainly in the final pooling layer: CAM-Avg uses average pooling (Zhou et al., 2016), CAM-Max uses max-pooling (Oquab et al., 2015), CAM-LSE uses an approximation to maximum (Pinheiro & Collobert, 2015; Sun et al., 2016), Wildcat uses the pooling in (Durand et al., 2017), Grad-CAM (Selvaraju et al., 2017), and Deep MIL is the work of (Ilse et al., 2018) with adaptation to multi-class. We use supervised segmentation using U-Net (Ronneberger et al., 2015) as an upper bound of the performance for pixel-level evaluation (Full sup.). As a simple baseline, we use a mask full of 1 with the same size of the image as a constant prediction of regions of interest to show that F1$^+$ alone is not an efficient metric to evaluate pixel-level localization particularly over GlaS set (All-ones, Tab.2). In our method, $\mathcal{M}$ and $\mathcal{C}$ *share* the same pre-trained backbone (resnet101 (He et al., 2016)) to avoid *overfitting* while using (Durand et al., 2017) as a

[2] GlaS: warwick.ac.uk/fac/sci/dcs/research/tia/glascontest.

[3] CUB-200-2011: www.vision.caltech.edu/visipedia/CUB-200-2011.html

[4] Oxford flower 102: http://www.robots.ox.ac.uk/ vgg/data/flowers/102/

Table 1: Image level performance over GlaS, CUB5, CUB, and OxF test sets.

Method	Image level Classification error (%)			
	GlaS	CUB5	CUB	OxF
CAM-Avg (Zhou et al., 2016)	**0.00**	13.79	24.62	13.04
CAM-Max (Oquab et al., 2015)	1.25	69.65	30.30	28.60
CAM-LSE (Pinheiro & Collobert, 2015; Sun et al., 2016)	1.25	84.13	28.44	27.35
Wildcat (Durand et al., 2017)	1.25	22.75	**22.12**	13.01
Deep MIL (Ilse et al., 2018)	2.50	12.41	24.74	**12.14**
Grad-CAM (Selvaraju et al., 2017)	**0.00**	11.03	24.62	13.04
Ours ($u = 4$)	**0.00**	**10.34**	26.73	19.98

pooling function. All methods are trained using stochastic gradient descent using momentum. In our approach, we use the same hyper-parameters over all datasets, while other methods require adaptation to each dataset. We provide the datasets splits, more experimental details, and visual results in the supplementary material (Sec.B). Our reproducible code is publicly available.

A comparison of the obtained results of different methods, over all datasets, is presented in Tab.1 and Tab.2 with visual results illustrated in Fig.3. In Tab.2, and compared to other WSL methods, our method obtains relatively similar F1$^+$ score; while it obtains large F1$^-$ over GlaS where it may be easy to obtain high F1$^+$ by predicting a mask full of 1 (Fig.3). However, a model needs to be very selective in order to obtain high F1$^-$ score in order to localize tissues (irrelevant regions) where our model seems to excel at. Cub5 set seems to be more challenging due to the variable size (from small to big) of the birds, their view, the context/surrounding environment, and the few training samples. Our model outperforms all the WSL methods in both F1$^+$ and F1$^-$ with a large gap due mainly to its ability to discard non-discriminative regions which leaves it only with the region of interest, the bird in this case. While our model shows improvements in pointwise localization, it is still far behind full supervision.

Similar improvements are observed on CUB data. In the case of OxF dataset, our approach provides low F1$^+$ values compared to other WSL methods. However, the latter are not far from the performance of the All-ones that predicts a constant mask. Given the large size of flowers, predicting a mask that is active over all the image will easily lead to 56.10% of F$^+$. The best WSL methods for OxF are only better than All-ones by $\sim 2\%$, suggesting that such methods have predicted a full mask in many cases. In term of F1$^-$, our approach is better than all the WSL techniques. All methods achieve low classification error on GlaS which implies that it represents an easy classification problem. Surprisingly, the other methods seem to overfit on CUB5, while our model shows a robustness. The other methods outperform our approach on CUB and OxF, although ours is still in a competitive range to half WSL methods. Results obtained on both these datasets indicate that, compared to WSL methods, our approach is effective in terms of image classification and pointwise localization with more reliability in the latter.

Visual quality of our approach (Fig.3) shows that the predicted regions of interest on GlaS agree with the doctor methodology of colon cancer diagnostics where the glands are used as diagnostic tool. Additionally, it deals well with multi-instances when there are multiple glands within the image. On CUB5/CUB, our model succeeds to locate birds in order to predict its category which one may do in such task. We notice that the head, chest, tail, or body particular spots are often parts that are used by our model to decide a bird's species, which seems a reasonable strategy as well. On OxF dataset, we observe that our approach mainly locates the central part of pistil. When it is not enough, the model relies on the petals or on unique discriminative parts of the flower. In term of time complexity, the inference time of our model is the same as a standard fully convolutional network since the recursive algorithm is disabled during inference. However, one may expect a moderate increase in training time that depends mainly on the depth of the recursion (see Sec.B.3.2).

Table 2: Pointwise localization performance over GlaS, CUB5, CUB, and OxF test sets.

Method	Pixel level							
	$F1^+$ (%)				$F1^-$ (%)			
	GlaS	CUB5	CUB	OxF	GlaS	CUB5	CUB	OxF
All-ones	66.01	23.72	22.16	56.10	00.00	00.00	00.00	00.00
CAM-Avg (Zhou et al., 2016)	66.90	35.25	32.24	**58.37**	17.88	68.44	82.87	68.70
CAM-Max (Oquab et al., 2015)	66.00	5.46	35.41	40.03	26.32	75.52	91.62	73.81
CAM-LSE (Pinheiro & Collobert, 2015; Sun et al., 2016)	66.05	8.00	35.79	40.65	27.93	77.21	91.62	73.07
Wildcat (Durand et al., 2017)	67.21	36.05	37.91	52.33	22.96	75.62	89.09	74.15
Deep MIL (Ilse et al., 2018)	68.52	29.70	22.19	56.10	41.34	37.59	0.31	0.0
Grad-CAM (Selvaraju et al., 2017)	66.30	36.91	32.24	**58.37**	21.30	69.55	82.87	68.70
Ours ($u = 4$)	**72.54**	**52.97**	**51.05**	43.35	**66.51**	**90.69**	**91.86**	**75.77**
Full sup.: U-Net (Ronneberger et al., 2015)	90.19	60.06	92.09	88.81	88.52	93.73	98.97	92.37

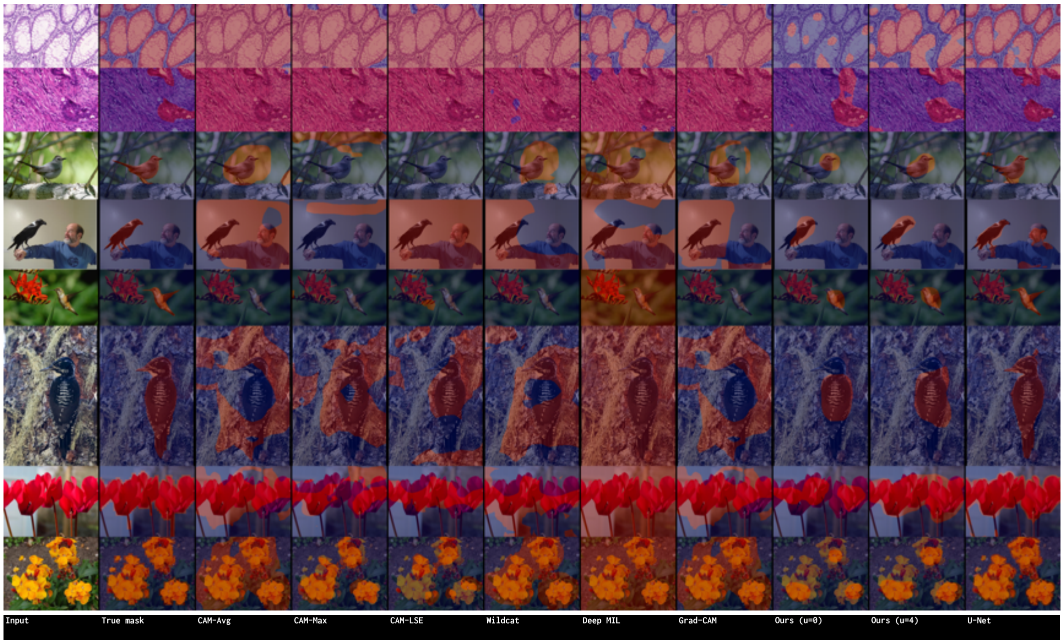

Figure 3: Visual comparison of the predicted binary mask of each method over GlaS, CUB5, CUB, and OxF test sets. (Best visualized in color.) (See supplementary material for more samples.)

5 Conclusion

In this work, we present a novel approach for WSL at pixel-level where we impose learning relevant and irrelevant regions within the model with the aim to reduce false positive localization. Evaluated on three datasets, and compared to state of the art WSL methods, our approach shows its effectiveness in accurately localizing regions of interest with low false positive while maintaining a competitive classification error. This makes our approach more reliable in term of interpetability. As future work, we consider extending our approach to handle multiple classes within the image. Different constraints can be applied over the predicted mask, such as texture properties, shape, or other region constraints. Predicting bounding boxes instead of heat maps is considered as well since they can be more suitable in some applications where pixel-level accuracy is not required. Our recursive erasing algorithm can be further improved by using a memory-like mechanism that provides spatial information to prevent forgetting the previously spotted regions and promote localizing the entire region (Sec.B.3).

References

H. Azizpour, M. Arefiyan, S. Naderi Parizi, and S. Carlsson. Spotlight the negatives: A generalized discriminative latent model. In *BMVC*, 2015.

S. Behpour, K. Kitani, and B. Ziebart. Ada: Adversarial data augmentation for object detection. In *WACV*, 2019.

S. Belharbi, R.Hérault, C. Chatelain, and S. Adam. Deep multi-task learning with evolving weights. In *ESANN*, 2016.

S. Belharbi, C. Chatelain, R. Hérault, and S. Adam. Neural networks regularization through class-wise invariant representation learning. *arXiv preprint arXiv:1709.01867*, 2017.

H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016.

C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *ICCV*, 2015.

A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, 2018.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

T. Durand, T. Mordan, N. Thome, and M. Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *CVPR*, 2017.

Thibaut Durand, Nicolas Thome, and Matthieu Cord. Weldon: Weakly supervised learning of deep convolutional neural networks. In *CVPR*, 2016.

G. Ghiasi, T.-Y. Lin, and Q. V. Le. Dropblock: A regularization method for convolutional networks. In *NIPS*. 2018.

K. He, X. Zhang, S.g Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

M. Ilse, J. M. Tomczak, and M. Welling. Attention-based deep multiple instance learning. *arXiv preprint arXiv:1802.04712*, 2018.

V. Kantorov, M. Oquab, M. Cho, and I. Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *ECCV*, 2016.

H. Kervadec, J. Dolz, M. Tang, E. Granger, Y. Boykov, and I. Ben Ayed. Constrained-CNN losses for weakly supervised segmentation. *MedIA*, 2019.

A. Khoreva, R. Benenson, J.H. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017.

D. Kim, D. Cho, D. Yoo, and I. So Kweon. Two-phase learning for weakly supervised object localization. In *ICCV*, 2017.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

E. Krupka and N. Tishby. Incorporating prior knowledge on features into learning. In *Artificial Intelligence and Statistics*, 2007.

C. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-Supervised Nets. In *ICAIS*, 2015.

K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu. Tell me where to look: Guided attention inference network. In *CVPR*, 2018.

D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

T.M. Mitchell. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research, 1980.

M.-E. Nilsback and A. Zisserman. Delving into the whorl of flower segmentation. In *BMVC*, 2007.

M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*, 2015.

S. Naderi Parizi, A. Vedaldi, A.w Zisserman, and P. F. Felzenszwalb. Automatic discovery and optimization of parts for image classification. In *ICLR*, 2015.

D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, 2015.

P. H. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

Belharbi S. Rony, J., J. Dolz, I. Ben Ayed, L. McCaffrey, and E. Granger. Deep weakly-supervised learning methods for classification and localization in histology images: a survey. *coRR*, abs/1909.03354, 2019.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

Y. Shen, R. Ji, S. Zhang, W. Zuo, Y. Wang, and F. Huang. Generative adversarial learning towards fast weakly supervised detection. In *CVPR*, 2018.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLRw*, 2014.

K. K. Singh and Y. J. Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017.

K. Sirinukunwattana, J. P. Pluim, H. Chen, X. Qi, P.-A. Heng, Y. B. Guo, L. Y. Wang, B. J. Matuszewski, E. Bruni, U. Sanchez, et al. Gland segmentation in colon histology images: The glas challenge contest. *MIA*, 2017.

J.T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLRw*, 2015.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014.

C. Sun, M. Paluri, R. Collobert, R. Nevatia, and L. Bourdev. Pronet: Learning to propose object-specific boxes for cascaded neural networks. In *CVPR*, 2016.

M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers. Normalized Cut Loss for Weakly-supervised CNN Segmentation. In *CVPR*, 2018.

P. Tang, X. Wang, X. Bai, and W. Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, 2017.

E. W. Teh, M. Rochan, and Y. Wang. Attention networks for weakly supervised object localization. In *BMVC*, 2016.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

F. Wan, P. Wei, J. Jiao, Z. Han, and Q. Ye. Min-entropy latent model for weakly supervised object detection. In *CVPR*, 2018.

Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017.

T. Yu, T. Jan, S. Simoff, and J. Debenham. Incorporating prior domain knowledge into inductive machine learning. *Unpublished doctoral dissertation Computer Sciences*, 2007.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. *IJCV*, 2018a.

Q.-s. Zhang and S.-c. Zhu. Visual interpretability for deep learning: a survey. *FITEE*, 2018.

X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 2018b.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

Z.-H. Zhou. A brief introduction to weakly supervised learning. *NSR*, 2017.

A The min-max entropy framework for WSL

A.1 Region completeness using incremental recursive erasing and trust coefficients

Deep classification models tend to rely on small discriminative regions (Kim et al., 2017; Singh & Lee, 2017; Zhou et al., 2016). Thus, in our proposal, $\boldsymbol{R}^{-}$ may still contain discriminative parts. Following (Kim et al., 2017; Li et al., 2018; Pathak et al., 2015; Singh & Lee, 2017), and in particular (Wei et al., 2017), we propose a learning incremental and recursive erasing approach that drives $\mathcal{M}$ to seek complete discriminative regions. However, in the opposite of (Wei et al., 2017) where such mining is done offline, we propose to incorporate the erasing within the backpropagation using an efficient and practical implementation. This allows $\mathcal{M}$ to *learn* to seek discriminative parts. Therefore, erasing during inference is unnecessary. Our approach consists in applying $\mathcal{M}$ recursively before applying $\mathcal{C}$ within the same forward. The aim of the recursion, with maximum depth u, is to mine more discriminative parts within the non-discriminative regions of the image masked by $\boldsymbol{R}^{-}$. We accumulate all discriminative parts in a temporal mask $\boldsymbol{R}^{+,\star}$. At each recursion, we mine the most discriminative part, that has been correctly classified by $\mathcal{M}$, and accumulate it in $\boldsymbol{R}^{+,\star}$. However, with the increase of u, the image may run out of discriminative parts. Thus, $\mathcal{M}$ is forced, unintentionally, to consider non-discriminative parts as discriminative. To alleviate this risk, we introduce *trust coefficients* that control how much we trust a mined discriminative region at each step t of the recursion for each sample i as follows,

$$\boldsymbol{R}_i^{+,\star} := \max(\boldsymbol{R}_i^{+,\star}, \Psi(t,i)\ \boldsymbol{R}_i^{+,t}) \ , \tag{10}$$

where $\Psi(t,i) \in \mathbb{R}^{+}$ computes the trust of the current mask of the sample i at the step t as follows,

$$\forall t \geq 0, \quad \Psi(t,i) = \exp^{\frac{-t}{\sigma}}\ \Gamma(t,i) \ , \tag{11}$$

where $\exp^{\frac{-t}{\sigma}}$ encodes the overall trust with respect to the current step of the recursion. Such trust is expected to decrease with the depth of the recursion (Belharbi et al., 2016). σ controls the slop of the trust function. The second part of Eq.11 is computed with respect to each sample. It quantifies how much we trust the estimated mask for the current sample i,

$$\Gamma(t,i) = \begin{cases} \hat{p}^s(\mathbf{Y}=y_i|\mathbf{X}=\boldsymbol{X}_i \odot \boldsymbol{R}_i^{-,\star}) & \text{if } \hat{y}_i = y_i \text{ and } \mathbf{H}(p_i, \hat{p}_i^s)_t \leq \mathbf{H}(p_i, \hat{p}_i^s)_0 \ \ , \\ 0 & \text{otherwise} \ \ . \end{cases} \tag{12}$$

In Eq.12, $\mathbf{H}(p_i, \hat{p}_i^s)_t$ is computed over $(\boldsymbol{X}_i \odot \boldsymbol{R}_i^{-,\star})$. Eq.12 ensures that at a step t, for a sample i, the current mask is trusted only if $\mathcal{M}$ correctly classifies the erased image, and does not increase the loss. The first condition ensures that the accumulated discriminative regions belong to the same class, and more importantly, the true class. Moreover, it ensures that $\mathcal{M}$ does not change its class prediction through the erasing process. This introduces a *consistency* between the mined regions across the steps and avoids mixing discriminative regions of different classes. The second condition ensures maintaining, at least, the same confidence in the predicted class compared to the first forward without erasing ($t=0$). The given trust in this case is equal to the probability of the true class. The regions accumulator is initialized to zero at $t=0$, $\boldsymbol{R}_i^{+,\star} = \{0\}^{h\times w}$ at each forward in $\mathcal{G}$. $\boldsymbol{R}_i^{+,\star}$ is not maintained through epoches; $\mathcal{M}$ starts over each time processing the sample i. This prevents accumulating incorrect regions that may occur at the beginning of the training. In order to automatize when to stop erasing, we consider a maximum depth of the recursion u. For a mini-batch, we keep erasing as along as we do not reach u steps of erasing, and there is at least one sample with a trust coefficient non-zero (Eq.12). Once a sample is assigned a zero trust coefficient, it is maintained zero all along the erasing (Eq.10)(Fig.4). Direct implementation of Eq.10 is not practical since performing a recursive computation on a large model $\mathcal{M}$ requires a large memory that increases with the depth u. To avoid such issue, we propose a practical implementation using gradient accumulation at $\mathcal{M}$ through the loss Eq.7; such implementation requires the same memory size as in the case without erasing. An illustration of our proposed recursive erasing algorithm is provided in Fig.4. Alg.1 illustrates our implementation using accumulated gradient through the backpropagation within the localizer $\mathcal{M}$. We note that this erasing algorithm is performed only during training.

A.2 Note on interpolation (Eq.9)

In most neural networks libraries (Pytorch (pytorch.org), Chainer (chainer.org)), the upsacling operations using interpolation/upsamling have a non-deterministic backward. This makes training

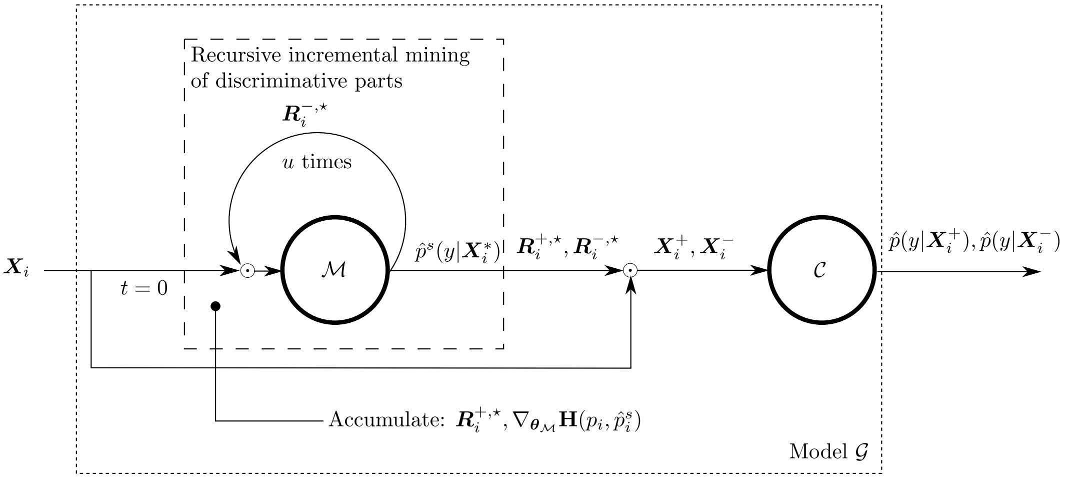
Recursive incremental mining of discriminative parts
$\boldsymbol{R}_i^{-,\star}$
u times
$\boldsymbol{X}_i$
$t = 0$
$\odot$
$\mathcal{M}$
$\hat{p}^s(y|\boldsymbol{X}_i^{*})$
$\boldsymbol{R}_i^{+,\star}, \boldsymbol{R}_i^{-,\star}$
$\odot$
$\boldsymbol{X}_i^{+}, \boldsymbol{X}_i^{-}$
$\mathcal{C}$
$\hat{p}(y|\boldsymbol{X}_i^{+}), \hat{p}(y|\boldsymbol{X}_i^{-})$
Accumulate: $\boldsymbol{R}_i^{+,\star}, \nabla_{\boldsymbol{\theta}_\mathcal{M}} \mathbf{H}(p_i, \hat{p}_i^s)$
Model $\mathcal{G}$

Figure 4: Illustration of the implementation of the proposed recursive incremental mining of discimi-native parts within the backpropagation. The recursive mining is performed only during training.

Algorithm 1 Practical implementation of our incremental recursive erasing approach during training for one epoch (or one mini-batch) using gradient accumulation.

```
Input: $\mathcal{G}, \mathbb{D}, u, \sigma$.
Initialization: $\nabla_{\{\boldsymbol{\theta}_\mathcal{M}, \boldsymbol{\theta}_\mathcal{C}\}} \big[\mathbf{H}(p_i, \hat{p}_i)^{+} + \mathbf{H}(q_i, \hat{p}_i)^{-} + \mathbf{H}(p_i, \hat{p}_i^s)\big] = \nabla^{\mathcal{G}}_{\{\boldsymbol{\theta}_\mathcal{M}, \boldsymbol{\theta}_\mathcal{C}\}} = \mathbf{0}$.
for $(\boldsymbol{X}_i, y_i) \in \mathbb{D}$ do
    Initialization: $\boldsymbol{R}_i^{+,\star} = \mathbf{0}, \nabla_{\boldsymbol{\theta}_\mathcal{M}} \mathbf{H}(p_i, \hat{p}_i^s) = \mathbf{0}, t = 0$, stop = False.
    Make a copy of $\boldsymbol{X}_i$: $\boldsymbol{X}_i^{\star}$.
    # Perform the recursion.  Accumulate gradients, and masks.
    while $t \leq u$ and stop is False do
        Forward $\boldsymbol{X}_i^{\star}$ in $\mathcal{M}$.
        Compute $\boldsymbol{R}_i^{+,t}, \boldsymbol{R}_i^{-,t}, \hat{p}^s(y|\boldsymbol{X}_i^{\star}), \mathbf{H}(p_i, \hat{p}_i^s)_t, \Psi(t, i)$.
        if $\Psi(t, i) \neq 0$ then
            Update accumulative mask $\boldsymbol{R}_i^{+,\star}$. (Eq.10)
            Accumulate gradient: $\nabla_{\boldsymbol{\theta}_\mathcal{M}} \mathbf{H}(p_i, \hat{p}_i^s) \mathrel{+}= \nabla_{\boldsymbol{\theta}_\mathcal{M}} \mathbf{H}(p_i, \hat{p}_i^s)_t$
            Erase the discriminative parts: $\boldsymbol{X}_i^{\star} := \boldsymbol{X}_i^{\star} \odot \boldsymbol{R}_i^{-,\star}$.
        else
            stop = True.
        end if
    end while
    Compute: $\boldsymbol{X}_i^{+} = \boldsymbol{X}_i \odot \boldsymbol{R}_i^{+,\star}, \boldsymbol{X}_i^{-} = \boldsymbol{X}_i \odot \boldsymbol{R}_i^{-,\star}$.
    Forward $\boldsymbol{X}_i^{+}, \boldsymbol{X}_i^{-}$ in $\mathcal{C}$.
    Compute: $\mathbf{H}(p_i, \hat{p}_i)^{+}, \mathbf{H}(q_i, \hat{p}_i)^{-}, \nabla_{\boldsymbol{\theta}_\mathcal{C}} \Big[\mathbf{H}(p_i, \hat{p}_i)^{+}, \mathbf{H}(q_i, \hat{p}_i)^{-}\Big]$.
    Update the total gradient: $\nabla^{\mathcal{G}}_{\{\boldsymbol{\theta}_\mathcal{M}, \boldsymbol{\theta}_\mathcal{C}\}} \mathrel{+}= \nabla_{\boldsymbol{\theta}_\mathcal{C}} \Big[\mathbf{H}(p_i, \hat{p}_i)^{+}, \mathbf{H}(q_i, \hat{p}_i)^{-}\Big] + \nabla_{\boldsymbol{\theta}_\mathcal{M}} \mathbf{H}(p_i, \hat{p}_i^s)$.
end for
Normalize total gradient: $\nabla^{\mathcal{G}}_{\{\boldsymbol{\theta}_\mathcal{M}, \boldsymbol{\theta}_\mathcal{C}\}} \mathrel{/}= n$. Update $\boldsymbol{\theta}_\mathcal{M}, \boldsymbol{\theta}_\mathcal{C}$ using $\nabla^{\mathcal{G}}_{\{\boldsymbol{\theta}_\mathcal{M}, \boldsymbol{\theta}_\mathcal{C}\}}$.
Output: $\mathcal{G}$ updated.
```

unstable due to the non-deterministic gradient; and makes reproducibility impossible as well. To avoid such issues, we detach the upsacling operation, in Eq.9, from the training graph and consider it as input data for $\mathcal{C}$.

B Results and analysis

In this section, we provide more details on our experiments, analysis, and discuss some of the drawbacks of our approach. We took many precautions to make the code reproducible for our model up to Pytorch's terms of reproducibility. Please see the `README.md` file for the concerned section in the code. We checked reproducibility up to a precision of 10^{-16}. All our experiments were conducted using the seed 0. We run all our experiments over one GPU with 12GB[5], and an environment with 10 to 64 GB of RAM (depending on the size of the dataset). Finally, this section shows more visual results, analysis, training time, and drawbacks.

B.1 Datasets

We provide in Fig.5 some samples from each dataset's test set along with their mask that indicates the region of interest. As we mentioned in Sec.4, we consider a subset from the original CUB-200-2011 dataset for preliminary experiments, and we referred to it as CUB5. To build it, we select, randomly, 5 classes from the original dataset. Then, pick all the corresponding samples of each class in the provided train and test set to build our train and test set (CUB5). Then, we build the effective train set, and validation set by taking randomly 80%, and the left 20% from the train set of CUB5, respectively. We provide the splits, and the code used to generate them. Our code generates the following classes:

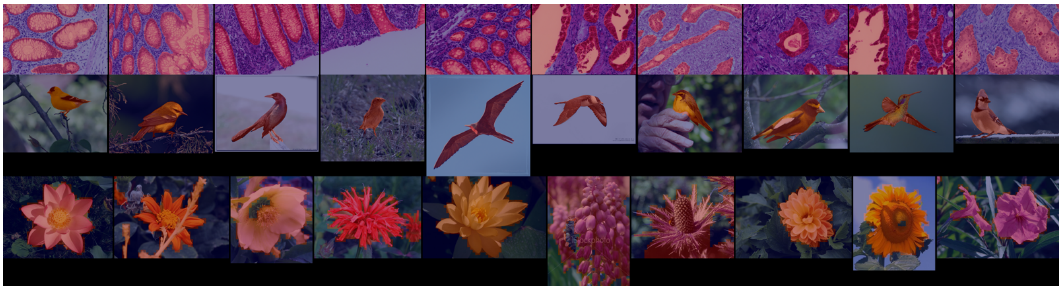

Figure 5: **Top row**: GlaS dataset: test set examples of different classes with the gland segmentation. The decidable regions are the glands while the undecidable regions are the leftover tissues. Glands have different shapes, size, context. They can be multi-instance. Images have variable H&E stain. (Sirinukunwattana et al., 2017). **Middle row**: CUB dataset: test set examples of randomly selected classes. The decidable regions are the birds while the undecidable regions are the leftover surrounding environment. Birds have different sizes, position/view, appearance, context. (Wah et al., 2011). **Bottom row**: Oxford flower 102 dataset: test samples of randomly selected classes. The decidable regions are the flowers while the undecidable ones are the surrounding environment. (Nilsback & Zisserman, 2007) (Best visualized in color.)

1. `019.Gray_Catbird`
2. `099.Ovenbird`
3. `108.White_necked_Raven`
4. `171.Myrtle_Warbler`
5. `178.Swainson_Warbler`

B.2 Experiments setup

The following is the configuration we used for our model over all the datasets:

Data 1. Patch size (hxw): 480×480. (for training sample patches, however, for evaluation, use the entire input image). 2. Augment patch using random rotation, horizontal/vertical flipping. (for CUB5 only horizontal flipping is performed). 3. Channels are normalized using 0.5

[5] Our code supports multiGPU, and Batchnorm synchronization with our own support to reproducibility.

mean and 0.5 standard deviation. 4. For GlaS: patches are jittered using brightness=0.5, contrast=0.5, saturation=0.5, hue=0.05.

Model Pretrained resnet101 (He et al., 2016) as a backbone with (Durand et al., 2017) as a pooling score with our adaptation, using 5 modalities per class. We consider using dropout (Srivastava et al., 2014) (with value 0.75 over GlaS and 0.85 over CUB5, CUB, OxF over the final map of the pooling function right before computing the score). High dropout is motivated by (Ghiasi et al., 2018; Singh & Lee, 2017). This allows to drop most discriminative parts at features with most abstract representation. The dropout is not performed over the final mask, but only on the internal mask of the pooling function. As for the parameters of (Durand et al., 2017), we consider their $\alpha = 0$ since most negative evidence is dropped, and use $kmax = kmin = 0.09$. $u = 0, u = 4, \sigma = 10, \sigma' = 0.5, \omega = 8$. For evaluation, our predicted mask is binarized using a 0.5 threshold to obtain exactly a binary mask. All our presented masks in this work follows this thresholding. Our F1$^+$, and F1$^-$ are computed over this binary mask.

Optimization 1. Stochastic gradient descent, with momentum 0.9, with Nesterov. 2. Weight decay of $1e-5$ over the weights. 3. Learning rate of 0.001 decayed by 0.1 each 40 epochs with minimum value of $1e-7$. 4. Maximum epochs of 400. 5. Batch size of 8. 6. Early stopping over validation set using classification error as a stopping criterion.

Other WSL methods use the following setup with respect to each dataset:

GlaS:

Data 1. Patch size (hxw): 416×416. 2. Augment patch using random horizontal flip. 3. Random rotation of one of: $0, 90, 180, 270$ (degrees). 4. Patches are jittered using brightness=0.5, contrast=0.5, saturation=0.5, hue=0.05.

Model 1. Pretrained resnet18 (He et al., 2016) as a backbone.

Optimization 1. Stochastic gradient descent, with momentum 0.9, with Nesterov. 2. Weight decay of $1e-4$ over the weights. 3. 160 epochs 4. Learning rate of 0.01 for the first 80, and of 0.001 for the last 80 epochs. 5. Batch size of 32. 6. Early stopping over validation set using classification error/loss as a stopping criterion.

CUB5:

Data 1. Patch size (hxw): 448×448. (resized while maintaining the ratio). 2. Augment patch using random horizontal flip. 3. Random rotation of one of: $0, 90, 180, 270$ (degrees). 4. Random affine transformation with degrees 10, shear 10, scale $(0.3, 1.5)$.

Model Pretrained resnet18 (He et al., 2016) as a backbone.

Optimization 1. Stochastic gradient descent, with momentum 0.9, with Nesterov. 2. Weight decay of $1e-4$ over the weights. 3. 90 epochs. 4. Learning rate of 0.01 decayed every 30 with 0.1. 5. Batch size of 8. 6. Early stopping over validation set using classification error/loss as a stopping criterion.

CUB/OxF:

Data 1. Patch size (hxw): 448×448. (resized while maintaining the ratio). 2. Augment patch using random horizontal flip. 3. Random rotation of one of: $0, 90, 180, 270$ (degrees). 4. Random affine transformation with degrees 10, shear 10, scale $(0.3, 1.5)$.

Model Pretrained resnet18 (He et al., 2016) as a backbone.

Optimization 1. Stochastic gradient descent, with momentum 0.9, with Nesterov. 2. Weight decay of $1e-4$ over the weights. 3. 90 epochs. 4. Learning rate of 0.01 decayed every 30 with 0.1. 5. Batch size of 64. 6. Early stopping over validation set using classification error/loss as a stopping criterion.

B.3 Results

In this section, we provide more visual results over the test set of each dataset.

Over GlaS dataset (Fig.7, 8), the visual results show clearly how our model, with and without erasing, can handle multi-instance. Adding the erasing feature allows recovering more discriminative regions. The results over CUB5 (Fig.9, 10, 11, 12, 13) while are interesting, they show a fundamental limitation to the concept of erasing in the case of one-instance. In the case of multi-instance, when the model spots one instance, then, erases it, it is more likely that the model will seek another instance which is the expected behavior. However, in the case of one instance, and where the discriminative parts are small, the first forward allows mainly to spot such small part and erase it. Then, the leftover may not be sufficient to discriminate. For instance, in CUB5, in many cases, the model spots only the head. Once it is hidden, the model is unable to find other discriminative parts. A clear illustration to this issue is in Fig.9, row 13. The model spots correctly the head, but was unable to spot the body while the body has similar texture, and it is located right near to the found head. We believe that the main cause of this issue is that the erasing concept *forgets* where discriminative parts are located since the mining iterations are done independently from each other in a sens that the next mining iteration is unaware of what was already mined. Erasing algorithms seem to be missing this feature that can be helpful to localize the entire region of interest by seeking *around* all the previously mined disciminative regions. In our erasing algorithm, once a region is erased, the model forgets about its location. Adding a memory-like, or constraints over the spatial distribution of the mined discriminative regions may potentially alleviate this issue. Another parallel issue of erasing algorithms is that once the most discriminative regions are erased it may not be possible to discriminate using the leftover regions. This may explain why our model was unable to spot other parts of the bird once its head is erased. Probably using soft-erasing (blur the pixel for example) can be more helpful than hard-erasing (set pixel to zero).

It is interesting to notice the strategy used by our model to localize some types of birds. In the case of the `099.Ovenbird`, it relies on the texture of the chest (white doted with black), while it localizes the white spot on the bird neck in the case of `108.White_necked_Raven`. One can notice as well that our model seems to be robust to small/occluded regions. In many cases, it was able to spot small birds in a difficult context where the bird is not salient.

Visual results over CUB and OxF are presented in Fig.14, and Fig.15, respectively.

B.3.1 Impact of our recursive erasing algorithm on the performance

Tab.3 and Tab.4 show the boosting impact of our erasing recursive algorithm in both classification and pointwise localization performance. From Tab.4, we can observe that using our recursive algorithm adds a large improvement in F1^+ without degrading F1^-. This means that the recursion allows the model to correctly localize larger portions of the region of interest *without* including false positive parts. The observed improvement in localization allows better classification error as observed in Tab.3. The localization improvement can be seen as well in the precision-recall curves in Fig.6.

Table 3: Impact of our incremental recursive erasing algorithm over the classification error of our approach over GlaS, CUB5, CUB, and OxF test sets.

Ours	Image level Classification error (%)			
	GlaS	CUB5	CUB	OxF
$u = 0$	1.25	19.31	**26.54**	25.15
$u = 4$	**0.00**	**10.34**	26.73	**19.98**

B.3.2 Running time of our recursive erasing algorithm

Adding recursive computation in the backpropagation loop is expected to add an extra computation time. Tab.5 shows the training time (of 1 run) of our model with and without recursion over identical computation resource. The observed extra computation time is mainly due to gradient accumulation (line 12. Alg.1) which takes the same amount of time as parameters' update (which is expensive to compute). The forward and the backward are practically fast, and take less time compared to gradient update. We do not compare the running between the datasets since they have different number/size of

Table 4: Impact of our incremental recursive erasing algorithm over the pointwise localization performance of our approach over GlaS, CUB5, CUB, and OxF test sets.

Ours	Pixel level							
	F1$^+$ (%)				F1$^-$ (%)			
	GlaS	CUB5	CUB	OxF	GlaS	CUB5	CUB	OxF
$u = 0$	39.99	40.07	46.39	23.43	65.30	89.99	85.94	73.65
$u = 4$	**72.54**	**52.97**	**51.05**	**43.35**	**66.51**	**90.69**	**91.86**	**75.77**

samples, and different pre-processing that it is included in the reported time. Moreover, the size of samples has an impact over the total time during the training over the validation set.

Table 5: Comparison of training time, of 1 run, over 400 epochs over GlaS and CUB5 of our model using identical computation resources (NVIDIA Tesla V100 with 12GB memory) when using our erasing algorithm ($u = 4$) and without using it ($u = 0$).

Model	GlaS	CUB5
Ours ($u = 0$)	49min	65min
Ours ($u = 4$)	90min ($\sim \times 1.83$)	141min ($\sim \times 2.16$)

B.3.3 POST-PROCESSING USING CONDITIONAL RANDOM FIELD (CRF)

Post-processing the output of fully convolutional networks using a CRF often leads to smooth and better aligned mask with the region of interest (Chen et al., 2015). To this end, we use the CRF implementation of (Krähenbühl & Koltun, 2011)[6]. The results are presented in Tab.6. Following the notation in (Krähenbühl & Koltun, 2011), we set $w^{(1)} = w^{(2)} = 1$. We set, over all the methods, $\theta_\alpha = 13, \theta_\beta = 3, \theta_\gamma = 3$ for 2 iterations, over GlaS, and $\theta_\alpha = 19, \theta_\beta = 11, \theta_\gamma = 5$ for 5 iterations, over CUB5, CUB, and OxF. Tab.6 shows a slight improvement in term of F1$^+$ and slight degradation in term of F1$^-$. When investigating the processed masks, we found that the CRF helps in improving the mask only when the mask covers precisely large part of the region of interest. In this case, the CRF helps spreading the mask over the region. In the case where there is high false positive, or the mask misses largely the region, the CRF does not help. We can see as well that the CRF increases slightly the false positive by spreading the mask out of the region of interest. Since our method has small false positive –i.e., the produced mask covers mostly the region of interest and avoids stepping outside it– using CRF helps in improving both F1$^+$ and F1$^-$ in most cases.

Table 6: Pointwise localization performance of different WSL models over the different test sets when post-processing the predicted masks using CRF (Krähenbühl & Koltun, 2011). $*^+$ indicates improvement while $*^-$ indicates degradation of performance compared to Tab.2. Deep MIL (Ilse et al., 2018) is discarded since the produced plans do not form probability over the classes axe at pixel level which is required for the CRF input (Krähenbühl & Koltun, 2011). To preserve horizontal space, we rename the methods CAM-Avg, CAM-Max, CAM-LSE, Grad-CAM to Avg, Max, LSE, G-C, respectively.

Method	Pixel level							
	F1$^+$ (%)				F1$^-$ (%)			
	GlaS	CUB5	CUB	OxF	GlaS	CUB5	CUB	OxF
Avg (Zhou et al., 2016)	66.90	34.90$^+$	30.86$^-$	57.66$^-$	17.65$^-$	66.85$^-$	80.63$^-$	65.84$^-$
Max (Oquab et al., 2015)	66.02$^+$	5.22$^-$	35.59$^+$	41.04$^+$	26.22$^-$	75.23$^-$	91.14$^+$	73.36$^-$
LSE (Pinheiro & Collobert, 2015; Sun et al., 2016)	66.06$^+$	7.85$^+$	36.32$^-$	41.49$^+$	27.75$^-$	77.09$^-$	91.26$^+$	72.57$^-$
Wildcat (Durand et al., 2017)	67.22$^+$	36.41$^+$	33.03$^-$	54.47$^+$	22.74$^-$	74.91$^-$	84.97$^+$	72.92$^-$
G-C (Selvaraju et al., 2017)	66.33$^+$	36.44$^+$	30.86$^-$	57.65$^-$	20.76$^-$	67.91$^-$	80.63$^-$	65.83$^-$
Ours ($u = 4$)	**72.58**$^+$	**54.69**$^+$	**53.35**$^+$	42.69$^-$	**66.49**$^-$	**91.06**$^+$	**92.30**$^+$	**75.84**$^+$

[6] https://github.com/lucasb-eyer/pydensecrf

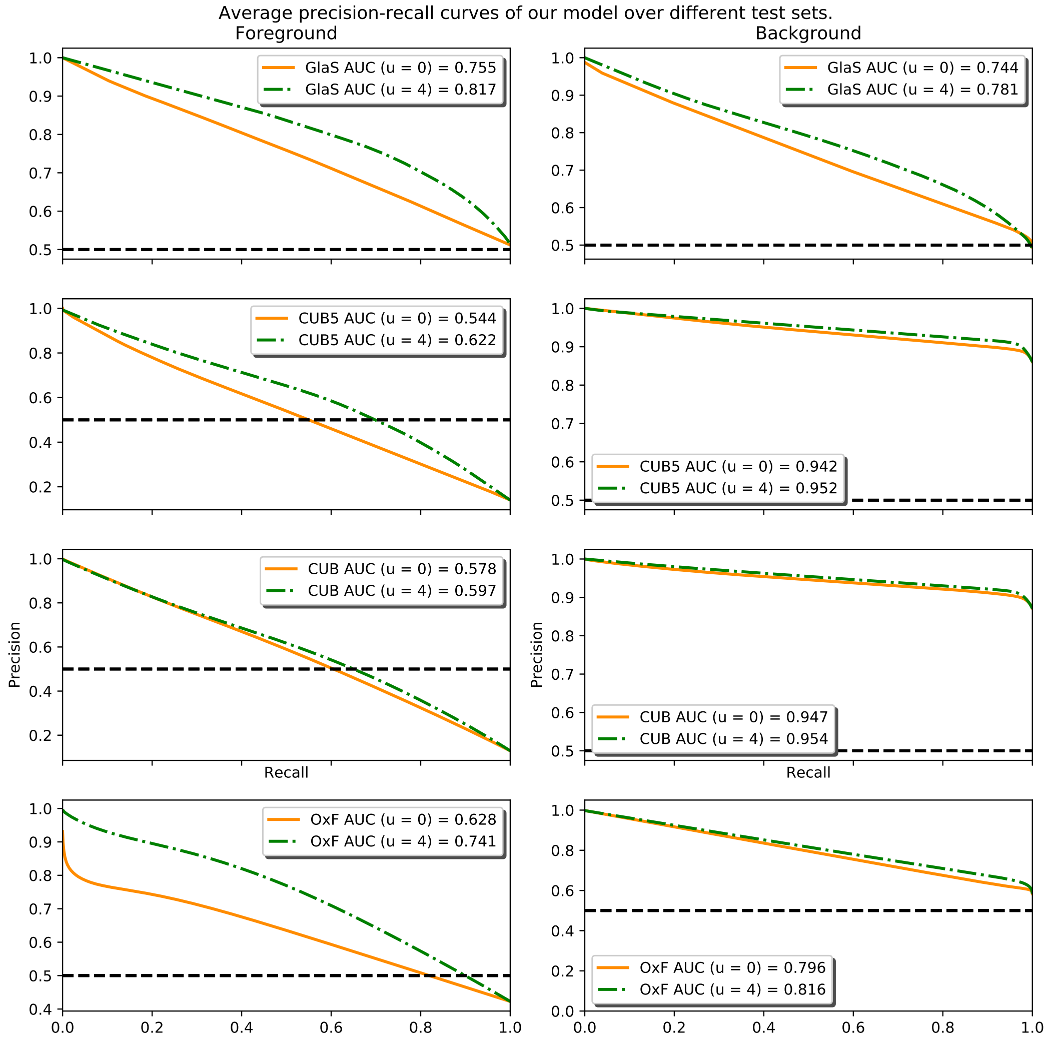

Figure 6: **Average** precision-recall curve of the foreground and the background of our proposal using $u = 0, u = 4$ over each test set. To be able to compute an average curve, the recall axis is unified for all the images to the axis $[0, 1]$ with a step $1e-3$. Then, the precision axis is interpolated with respect to the recall axis.

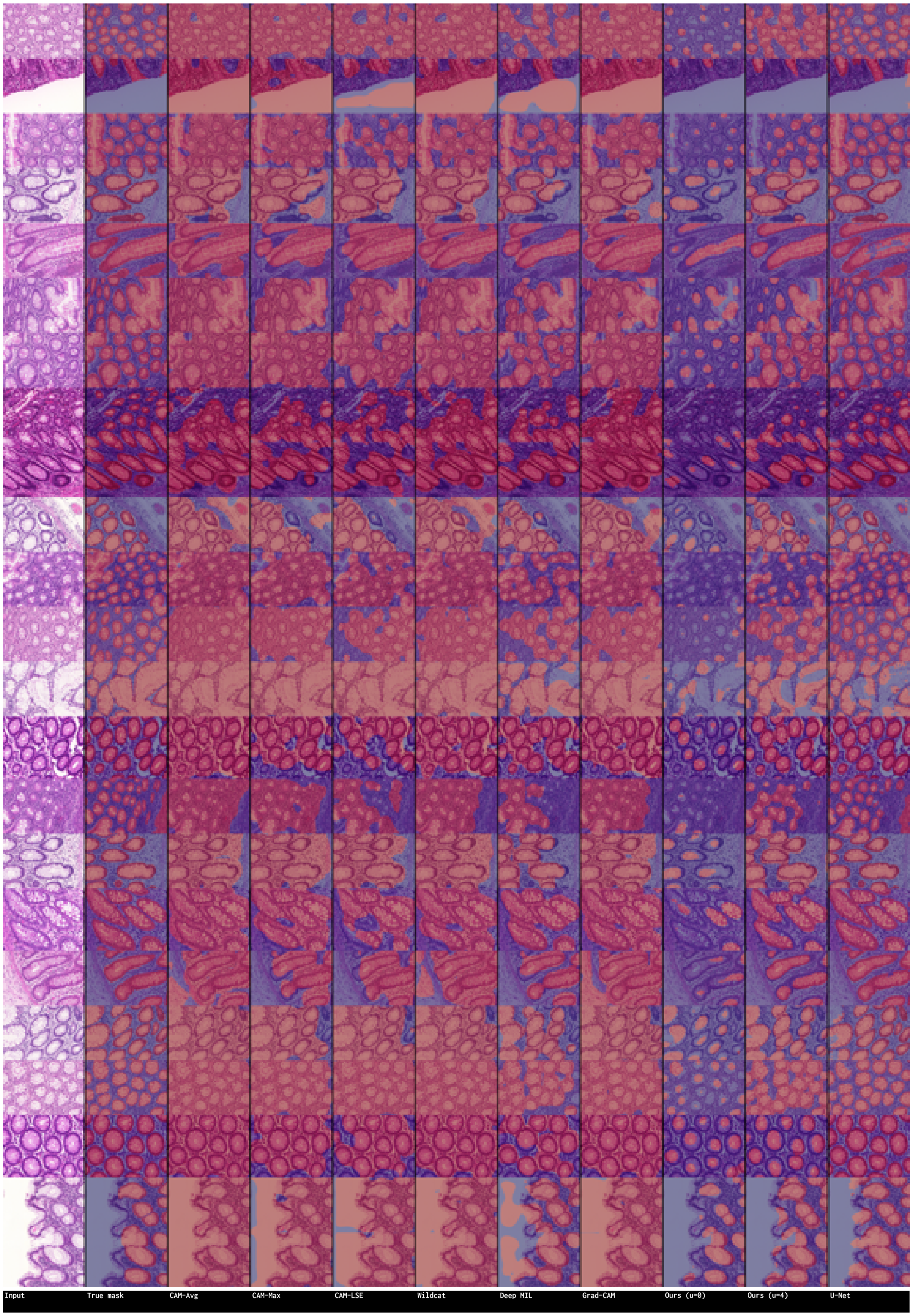

Figure 7: Visual comparison of the predicted binary mask of each method over GlaS test set. Class: `benign` (Best visualized in color.)

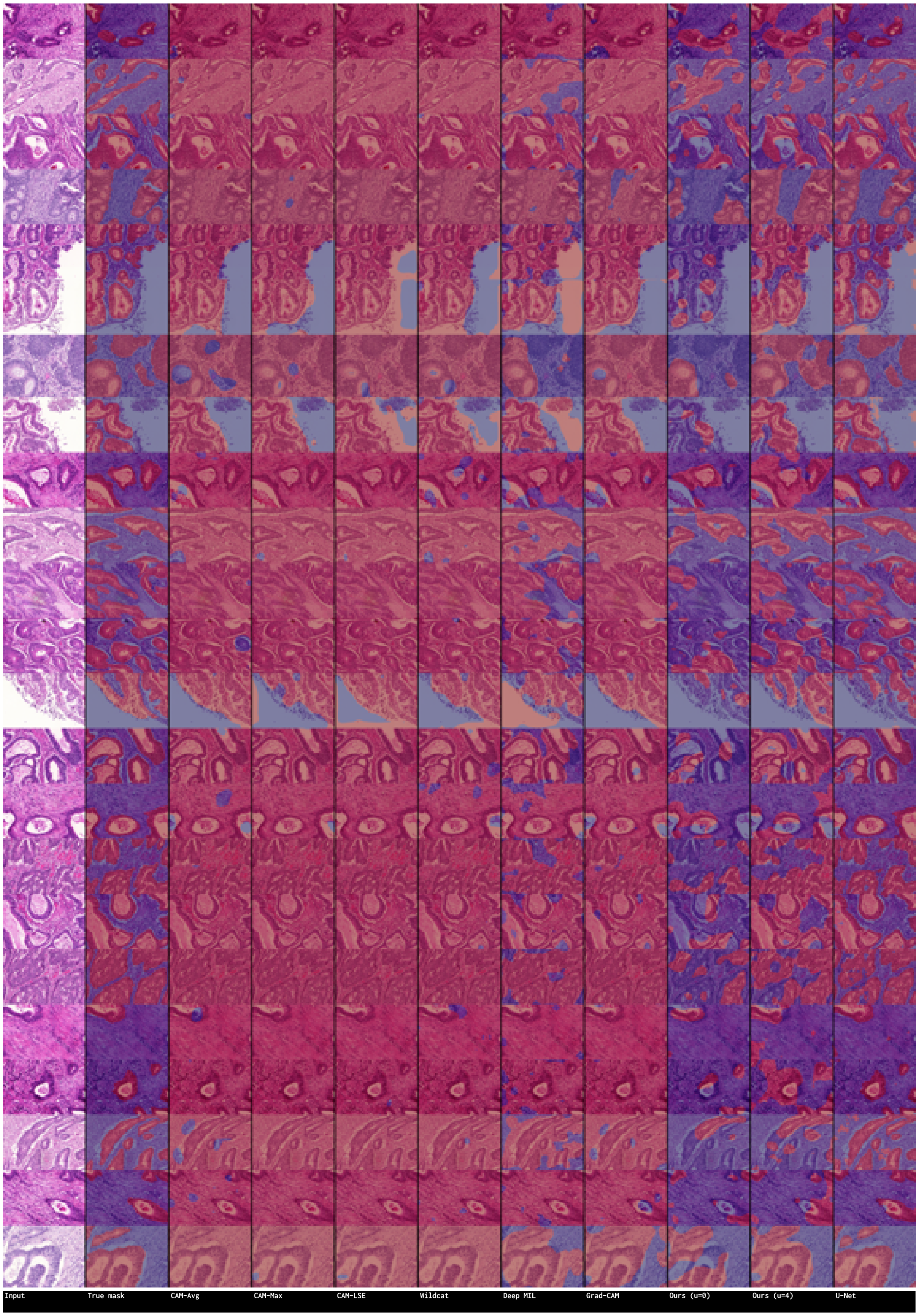

Figure 8: Visual comparison of the predicted binary mask of each method over GlaS test set. Class: `malignant` (Best visualized in color.)

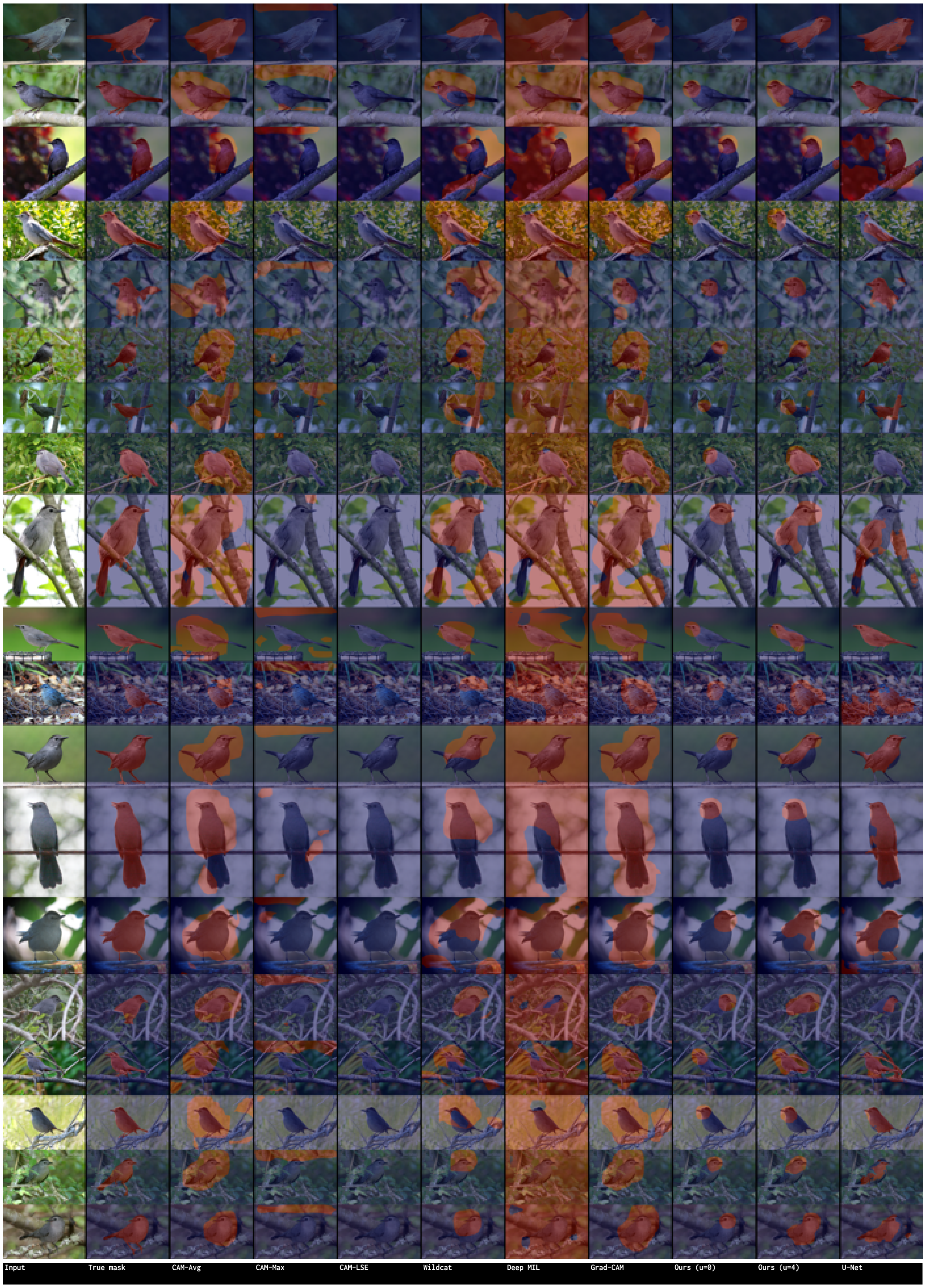

Figure 9: Visual comparison of the predicted binary mask of each method over CUB-200-2011 (CUB5) test sets. Species: `019.Gray_Catbird` (Best visualized in color.)

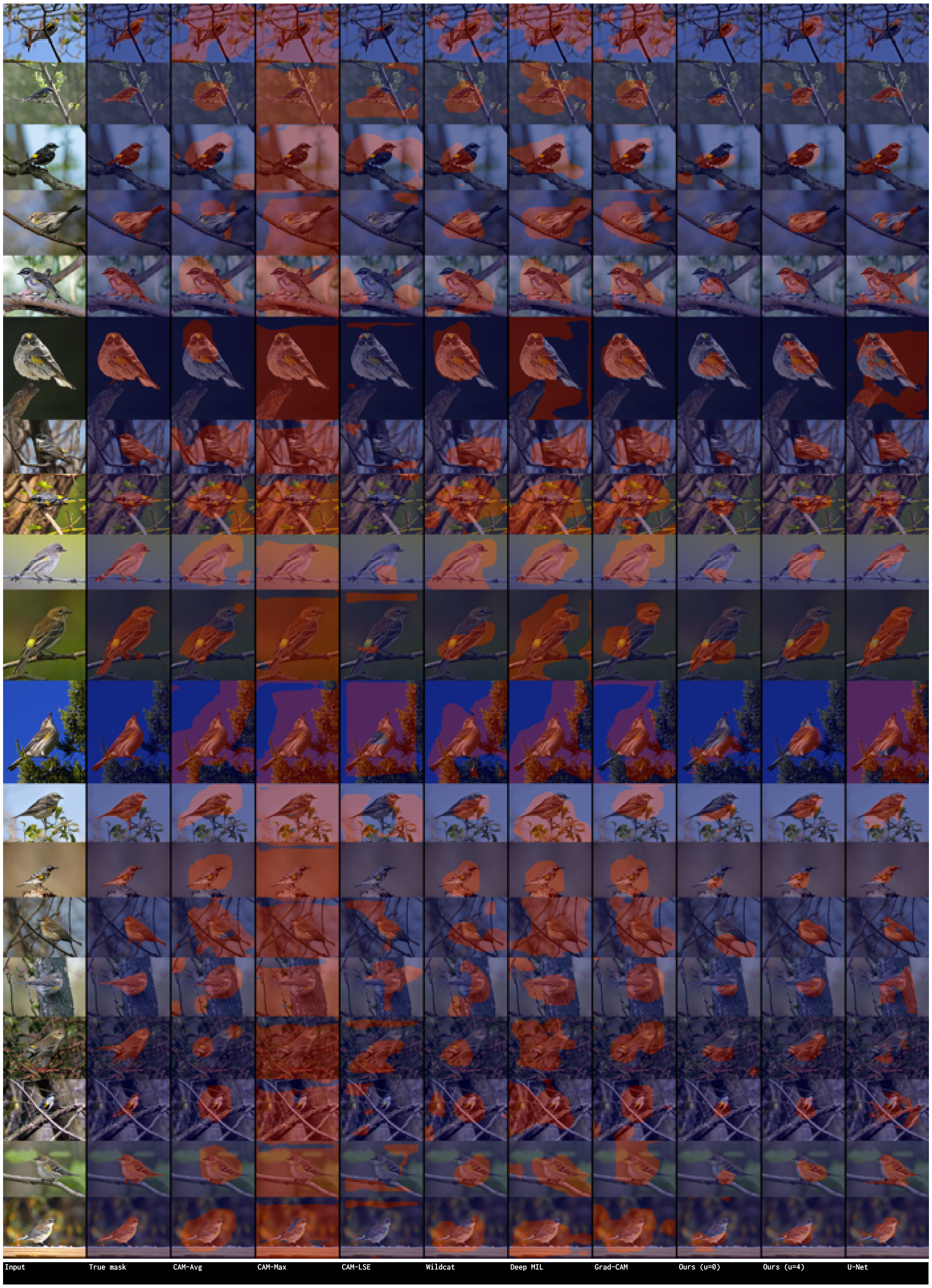

Figure 10: Visual comparison of the predicted binary mask of each method over CUB-200-2011 (CUB5) test sets. Species: `171.Myrtle_Warbler` (Best visualized in color.)

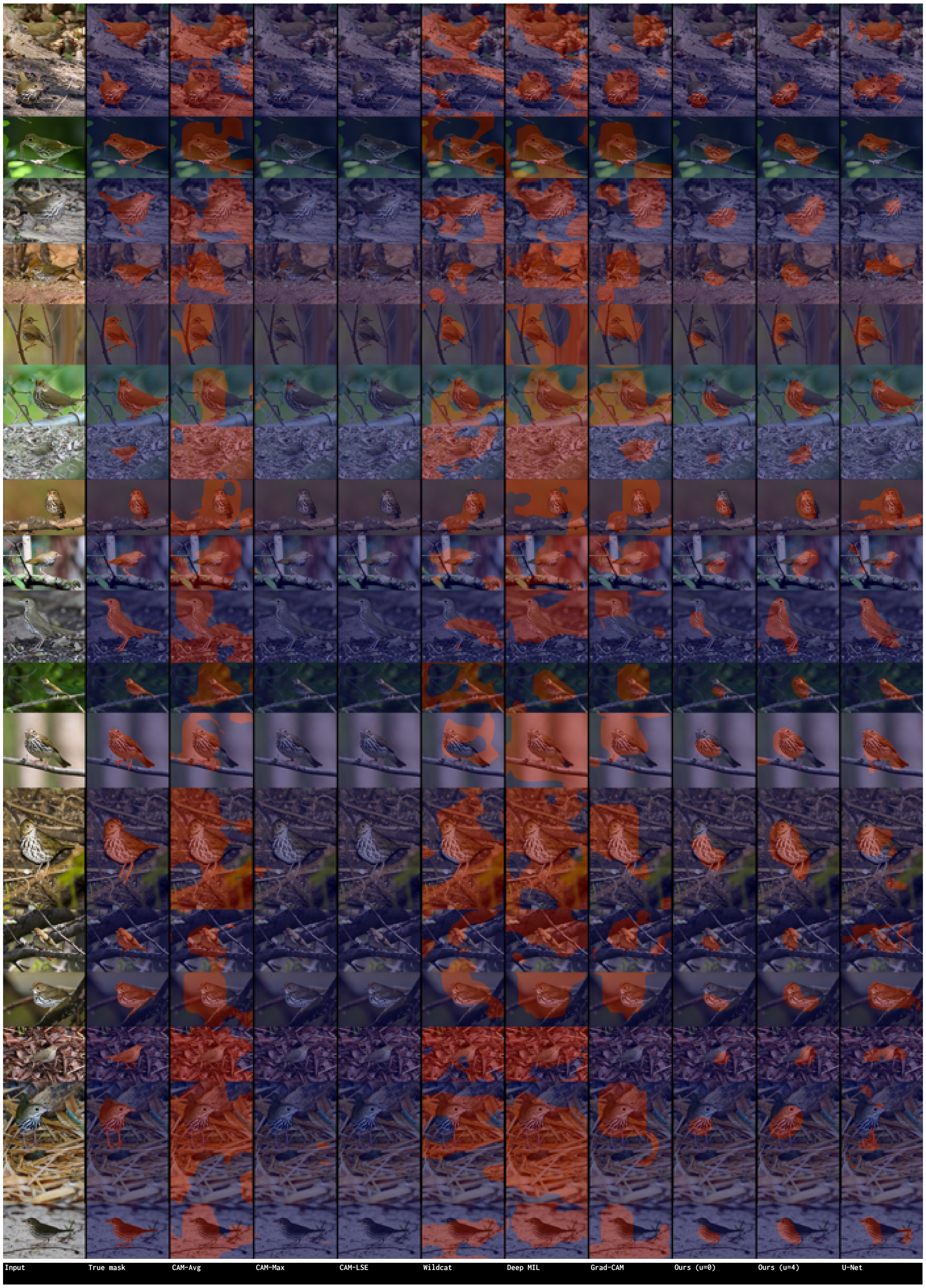

Figure 11: Visual comparison of the predicted binary mask of each method over CUB-200-2011 (CUB5) test sets. Species: `099.Ovenbird` (Best visualized in color.)

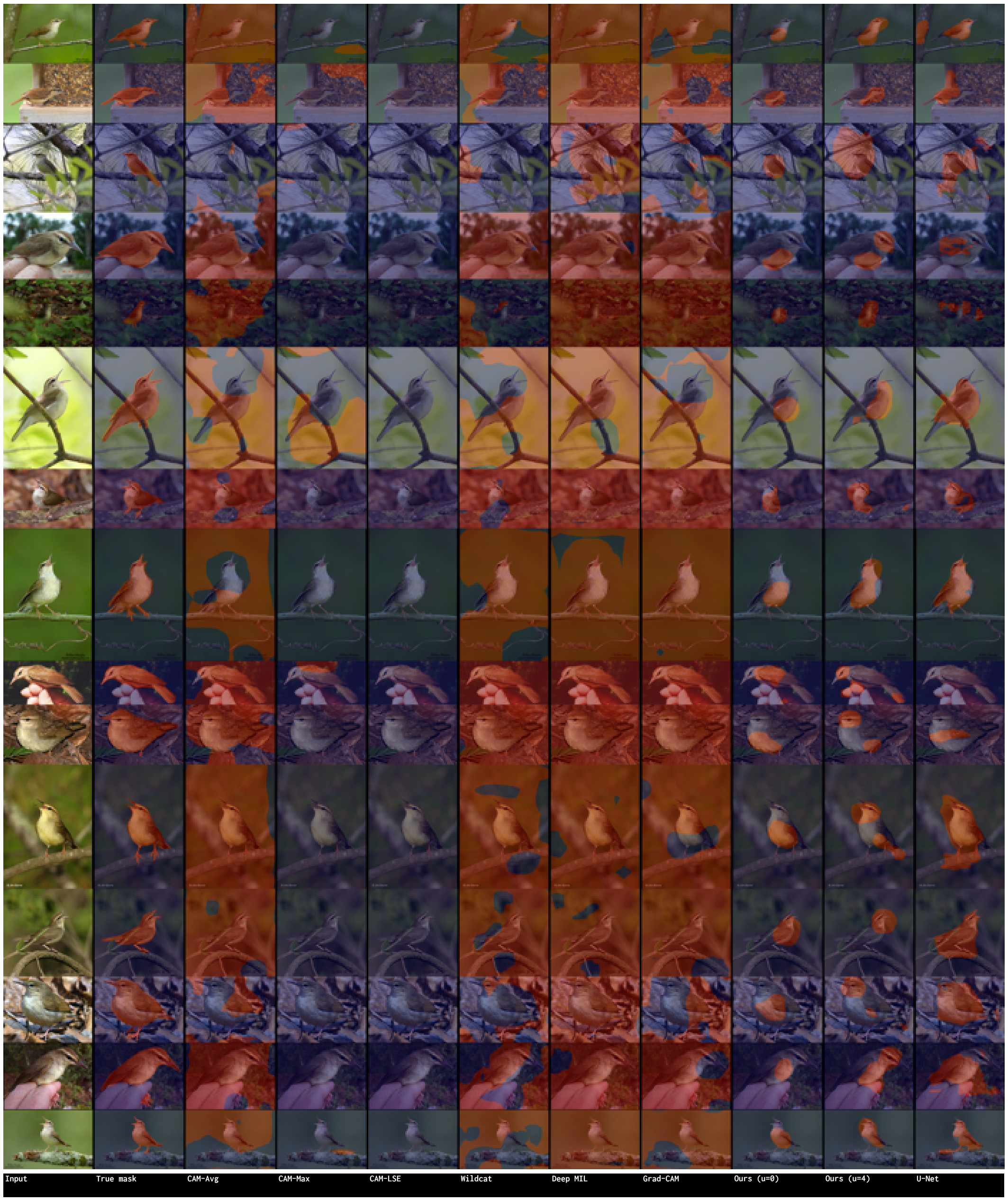

Figure 12: Visual comparison of the predicted binary mask of each method over CUB-200-2011 (CUB5) test sets. Species: `178.Swainson_Warbler` (Best visualized in color.)

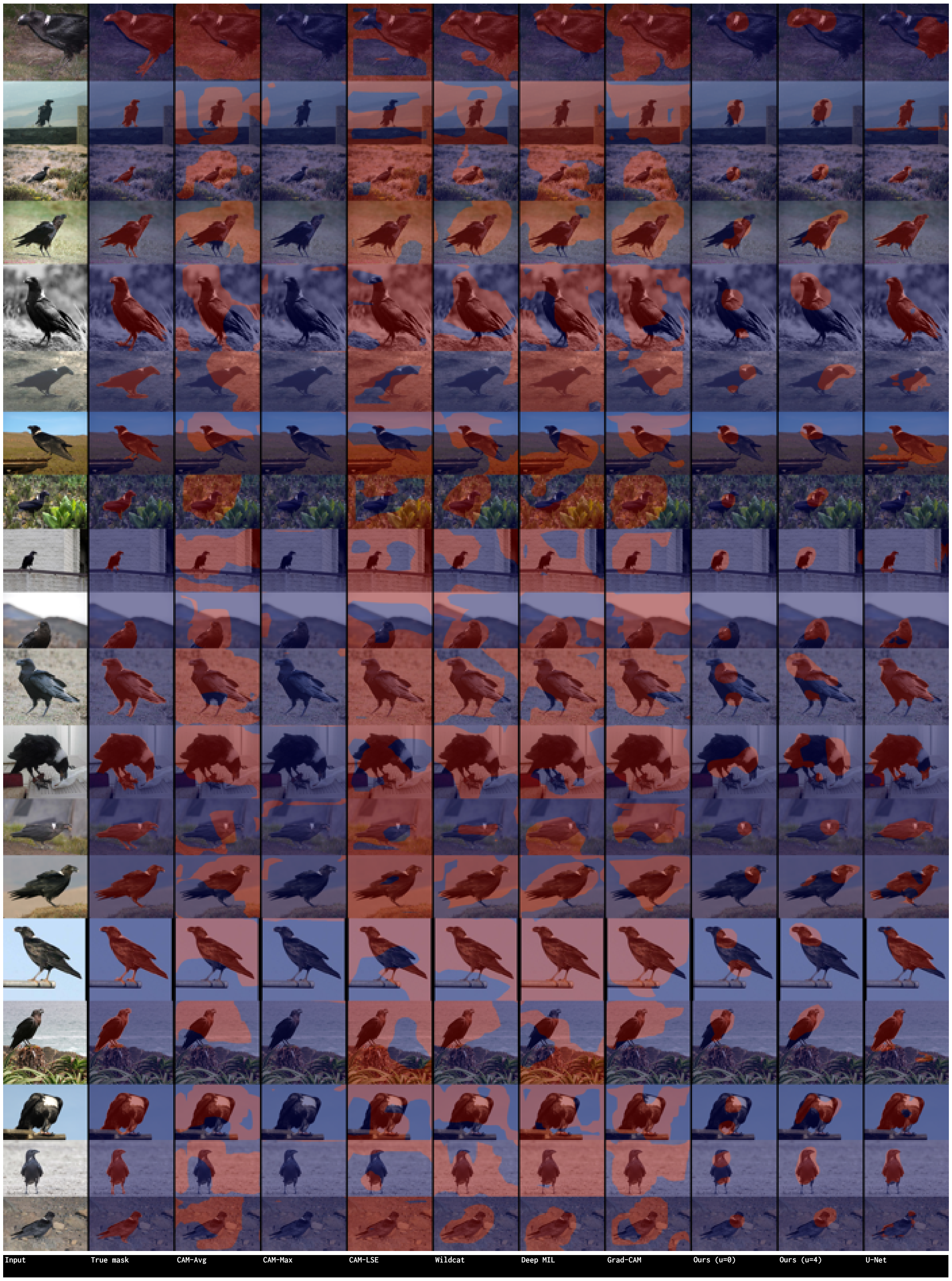

Figure 13: Visual comparison of the predicted binary mask of each method over CUB-200-2011 (CUB5) test sets. Species: `108.White_necked_Raven` (Best visualized in color.)

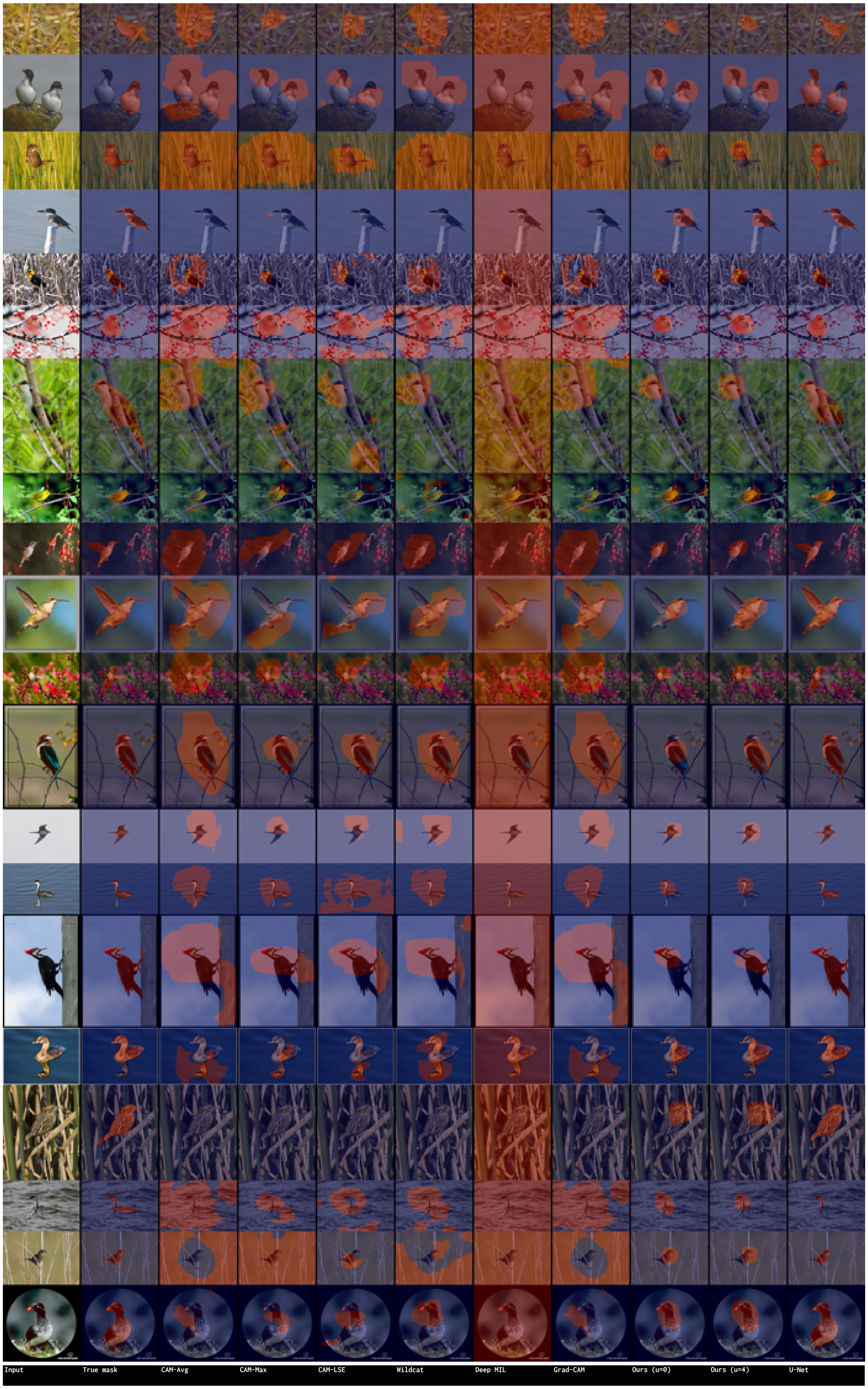

Figure 14: Visual comparison of the predicted binary mask of each method over CUB test sets. (Best visualized in color.)

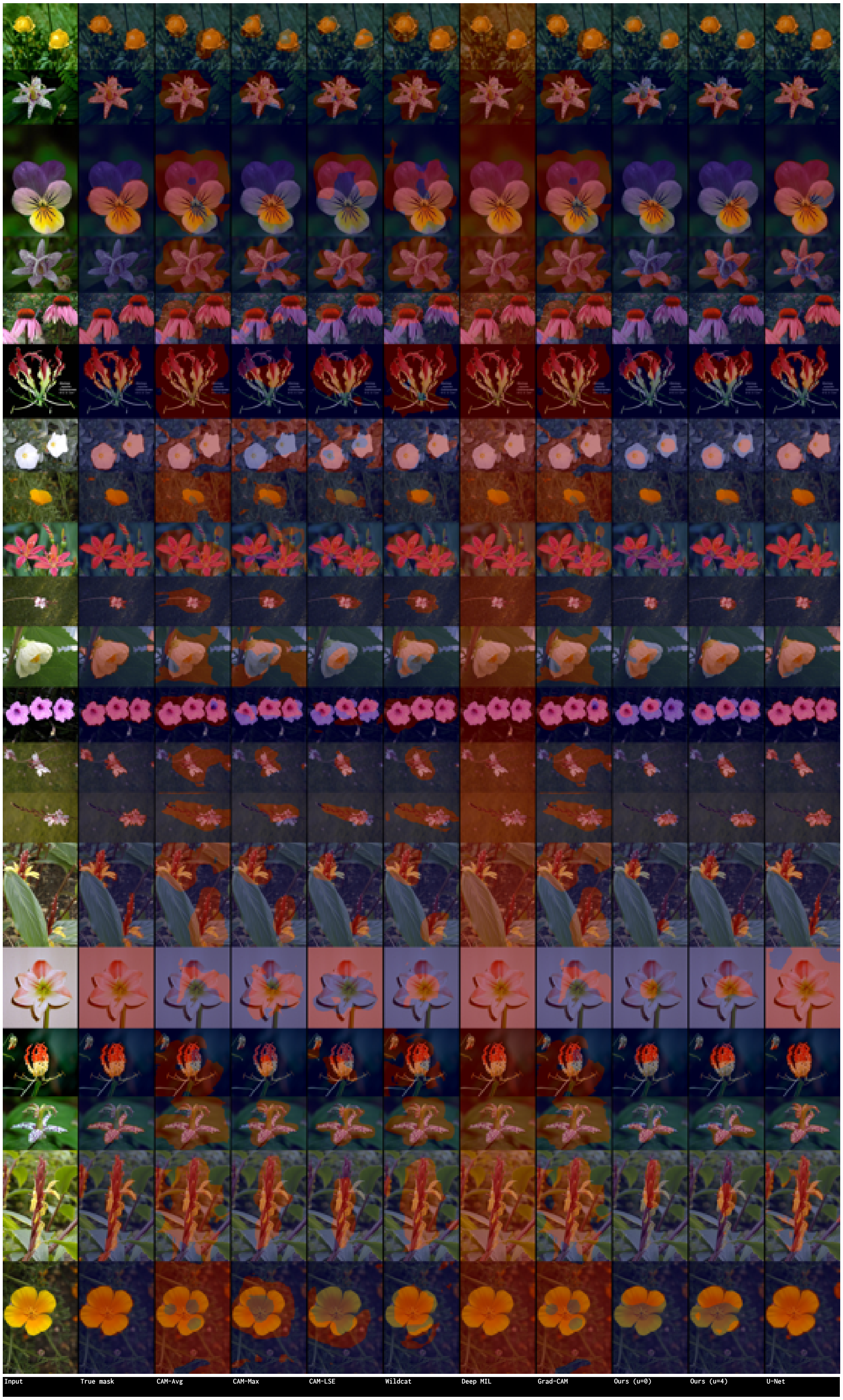

Figure 15: Visual comparison of the predicted binary mask of each method over oxF test sets. (Best visualized in color.)