Adaptive Select Loss Strategy for Semantic Segmentation of Agricultural Crop Images

Corneliu Florea, Laura Florea, and Mihai Ivanovici, *Senior Member, IEEE*

***Abstract*—We address the problem of agricultural image segmentation by introducing a novel loss formulation called Adaptive Select Loss (ASL), inspired by the Top-k loss strategy. While Top-k loss was originally designed for classification tasks, ASL is specifically tailored for semantic segmentation. It exploits the hierarchical structure of loss computation specific in semantic segmentation–aggregated first at the pixel level, then at the image level–while accounting for the imbalance between precise but scarce image-level annotations and noisy yet abundant pixel-level labels. ASL selectively aggregates loss from a "Few" (Top-k) of the most informative image-level instances and from "Almost all" (remove few) pixel-level data, thereby balancing robustness and sensitivity to noise. To ensure stability during training, we introduce a derivative smoothing mechanism that addresses the convergence issues introduced by the hard selection threshold, particularly when training with small number of images for loss aggregation. Empirically, the proposed approach improves boundary localization and segmentation quality in the presence of annotation noise. We evaluate ASL on three challenging semantic segmentation tasks–two agricultural and one mixed–using a visual transformer backbone, including hyperspectral data. ASL achieves consistent performance improvements, with gains of approximately 2.5% on hyperspectral and satellite imagery, and up to 6% on RGB-D data plant segmentation problem.**

***Index Terms*—Adaptive select loss (ASL), crop identification, unmanned aerial vechicle (UAV) hyperspectral image, visual transformer, image semantic segmentation.**

I. Introduction

ARGUABLY the most remarkable evolution of today's agriculture is the use of digitization and artificial intelligence as a cost-effective alternative to human decision-making [1]. Digital data from on-site, proximal, and satellite measurements provide valuable insights into crop health. Key approaches are based on visual data and agricultural imaging varies by camera placement (satellite, UAS, local, close-up) and light spectra (RGB to multi- and hyperspectral) [2], [3]. Quite often, the image is further processed and segmentation is among top necessities [4].

Semantic segmentation techniques are gaining importance in remote sensing and agriculture as they are helpful while delineating objects of interest, such as crop, or individual plants, or weeds. More precisely, remote sensed hyperspectral image segmentation has been approached by Ashraf et al. [5] by modifying a UNet module for the specific of data. In greenhouse farming [6], the segmentation technique is critical for monitoring crop growth, predicting canopy area and height, and assessing fruit ripeness. It can map the green house [7], monitor the sowing process [8], or it can enhance crop production quality by suggesting the optimal harvesting time; it can improve the overall efficiency. Directly crop segmentation in multispectral images have been approached by Song et al. [9] by developing and particularizing a Segment Anything Model. Furthermore, in remote monitoring of agricultural crops, due to the nonuniformity of ground and near-ground environments, there are challenges in acquiring macroscopic information and regularizing parameters as spatial scale increases. In agricultural applications, segmentation challenges are significant due to the nature of the environment and the data involved. These challenges include data acquisition, high annotation costs, and heterogeneity.

In the early stages of agricultural image segmentation development, classical techniques were commonly used. These methods included spectral texture features followed by thresholding [10], or using clustering algorithm such as mean shift [11]. Recently, however, deep learning-based solutions started to dominate the state of the art. Here, the initial convolutional based methods [12] have been slowly, but firmly, replaced with ones built upon Visual Transformers [13]. Nowadays most solutions–whether convolutional or transformer-based–adapt architectures derived from U-Net [14], [15], [16]. These architectures typically include an encoder to identify the class and a decoder to assign it to each pixel, while preserving image resolution.

However, the solutions are perfectible and various directions have been investigated. Niu et al. [17] used image pre-processing and augmentation to offer more views for training. Improved models have been proposed: Pastorino et al. [18] investigated the addition of hierarchical probabilistic graphical models over the initial convolutional models, while [13] proposed developing the TransUNet model [16] into a version adapted to hyperspectral imaging. Also, in the same work, better performance

Received 3 April 2025; revised 6 June 2025; accepted 7 July 2025. Date of publication 16 July 2025; date of current version 5 August 2025. This work was supported in part by the European Union and in part by AI4AGRI project entitled "Romanian Excellence Center on Artificial Intelligence on Earth Observation Data for Agriculture" received funding from the European Union's Horizon Europe research and innovation program under Grant 101079136. *(Corresponding author: Laura Florea.)*

Corneliu Florea is with the Image Processing and Analysis Laboratory, National University of Science and Technology Politehnica Bucharest, 060042 Bucharest, Romania, and also with the MIV Laboratory, Transilvania University of Brasov, 500024 Brasov, Romania (e-mail: corneliu.florea@upb.ro).

Laura Florea is with the Image Processing and Analysis Laboratory, National University of Science and Technology Politehnica Bucharest, 060042 Bucharest, Romania (e-mail: laura.florea@upb.ro).

Mihai Ivanovici is with the MIV Laboratory, Transilvania University of Brasov, 500024 Brasov, Romania (e-mail: mihai.ivanovici@unitbv.ro).

Digital Object Identifier 10.1109/JSTARS.2025.3589635

has been sought by using more adapted loss functions [13], such as combining the standard Cross-Entropy with the Dice Coefficient. Alternate architecture, named LodgeNet has been employed for rice lodging recognition [19]. Also, another architecture, namely EsaNet was used for segmentation of of RGB-D (depth) image multi-class plant species semantic segmentation in crop farming [20]. Other solution sought improvement in the better usage of the data such a synergy between channels in multispectral data [21], placed carefully engineered attention blocks in the transformer architecture [22], added local-global interaction modules to fully exploit local and global contexts for feature refinement [23], or optimized learning based on the local image frequency [24].

This article proposes a way to enhance semantic segmentation solutions. To justify the proposal, we will use a general machine learning principle of building loss functions, which we will develop for the agricultural image segmentation task.

To train a semantic segmentation model, one needs a large image dataset with corresponding pixel-level annotations (masks) indicating the class of each pixel. Several such datasets for agricultural crop images are public for research purposes, yet they suffer from noisy labels at pixel level. More precisely, the areas with crops are correctly labeled—meaning that the overall label is correct, which is the crop is precisely identified. But, often, there are errors in pixels labels at the borders of the regions—the regions are not very precisely delimited. These pixel labeling errors can cause issues during segmentation model training and can lead to incorrect results in testing. Based on these observations, we propose a new semantic segmentation method, that automatically eliminates from the training process those pixels that might be incorrectly labeled and also, in contrast, focuses on images and regions that are correctly labeled, but are more difficult and, thus, more informative.

Our proposal introduces a way of aggregating the loss values from individual data instances into the overall loss. The most common approach used today for aggregation is to average the individual loss values over all examples; this means to follow the principle of Empirical Risk Minimization (ERM) [25]. ERM is preferred due to its efficient optimization algorithms and robust theoretical foundation, as highlighted by Bartlett et al. [26].

On the other hand, selecting the largest (Top-k) individual losses, instead of using all of them, was shown to form a more stringent criterion [27]. This selection forces the model to focus on *hard examples*. Recently, variants of the Top-k loss have been explored in image classification. Those variants were proven to give good results, although there were some identified limitations in terms of convergence [28], [29], [30].

Starting from the Top-k loss and the observation that not all pixels are correctly labeled, we adopt a strategy that will allow selecting specific examples for specific stages in training: we will focus on hard examples when looking at image or large regions, but we will discard from training the pixels that are potentially noisy. While Top-k loss focuses on classification tasks by selecting the most difficult or informative samples across a batch, it operates at a coarse, instance-level granularity without considering spatial structure. In contrast, Adaptive Select Loss (ASL) extends this principle to semantic segmentation, where loss computation is inherently hierarchical—first at the pixel level, then aggregated at the image level. ASL introduces a dual-selection mechanism: It applies Top-k selection to image-level instances to focus on challenging examples, while simultaneously performing an "Almost-all" selection at the pixel level to retain broad spatial coverage and reduce the impact of noisy labels. This tailored approach allows ASL to maintain the robustness benefits that Top-k has in classification tasks while adapting to the fine-grained, spatially structured nature of segmentation tasks.

We will test the proposed strategy on two datasets from different tasks related to agricultural image segmentation and, to have a better evaluation, on an additional set with mixed classes but that include "agricultural." One purely agricultural task uses hyperspectral images remotely acquired and focuses on delineating crops. Applications include crop area estimation, which aids in yield prediction, as well as autonomous drone navigation. The second task uses RGB-D images acquired from close-by and focuses on individual plants. Applications encompass greenhouse farming, as well as traditional agriculture, serving as an intermediate step in automatic weed removal or crop yield estimation. For the third database, its application is in the early stages a terrain delineation application, where the general and heterogeneous classes are separated.

A. Motivation and Proposal

As previously mentioned, in this article we propose a new strategy for aggregation of the individual loss values, that is named ASL. This strategy is motivated by the following observations with respect to semantic segmentation: 1) at image label, the class of the object (e.g. type of crop in the used databases) is correct; 2) at pixel level, due to limitations in annotation tool, differences in resolution between the image used for annotation and the image in the actual database, natural limitation of human annotators, there are some noisy (pixel) labels. These ideas are illustrated with examples from the two agricultural datasets considered (UAV-HSI and WE3DS) in Fig. 1. One can see that the regions (images) have correct labels—e.g., Fig. 1(a) is correctly labeled as containing millet. Yet there are areas where the labels may be wrongly placed - marked with red arrows. These areas, having hard labels, force the model to learn poor data. It is better to remove the wrongly labeled data from the training process.

We recall also an important result [31], from neural networks, that is obviously inherited by deep networks [32]; this result states that in the iterative learning process, in the early stages, the clear and simple examples are addressed first (as they produce the largest changes in the derivative), while in the later stages, the focus should be on the harder, more difficult ones. By adopting such a strategy, the learning duration is decreased too. We build our proposal into this direction: the proposed ASL is a strategy or formalism which allows selecting specific examples for specific stages of training. In the particular case of semantic segmentation, the selection in late stages is contradictory: for images we have to focus on few hard examples, for pixels we need to use all, and discard few, which are potentially noisy.

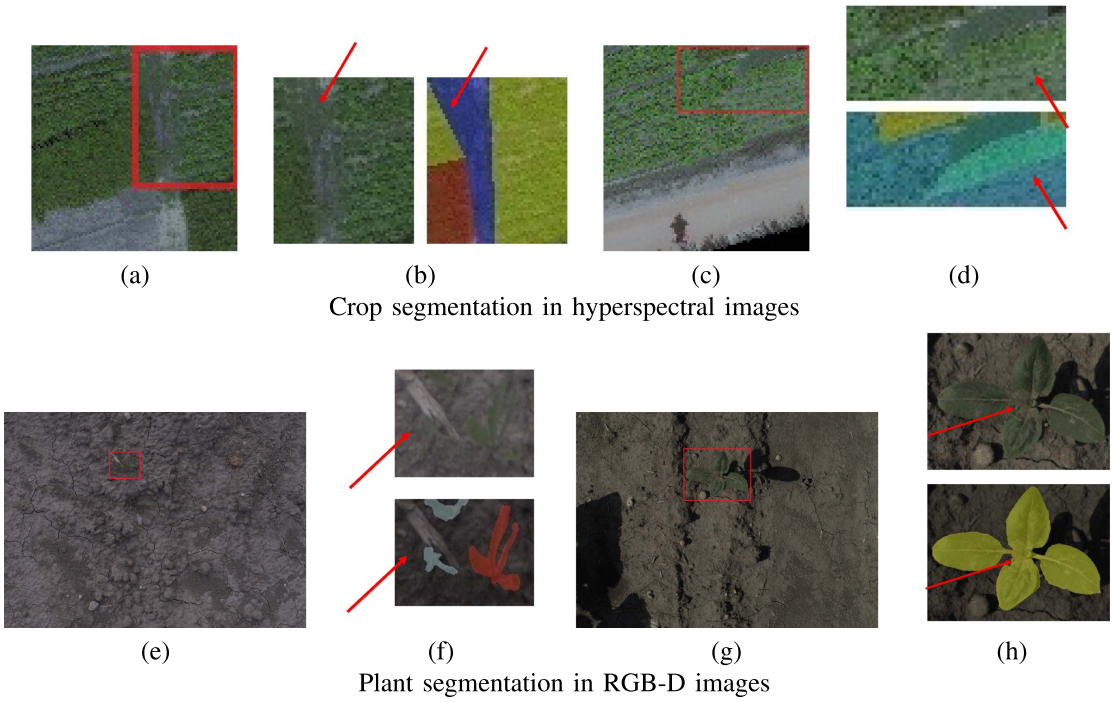


Fig. 1. Examples of images from two agricultural datasets, with differences between classes and noisy annotations at pixels level. On the top row, there are hyperspectral images from the UAV-HSI datasets rendered in RGB color space: (a) image with millet with a detail marked with a red rectangle, (b) detail of the image and of the annotation (millet is labeled with yellow); red arrow points to a boundary area where the distinction between the crops is not clear and those are noisy labeled pixels, (c) image with corn, (d) detail (corn is labeled with olive—low left corner). On the bottom row are RGB-D images used for plant phenomics: (e) image with "small-flower geranium," (f) detail showing noisy pixel labels, (g) image with "sunflower," (h) detail. Note the differences between the relative size of the object of interest among the two agricultural databases. Red arrows point to areas with noisy pixel annotations, where the annotated border does not match the visual perception.

B. Prior Work

In the previous paragraphs, the applicability of semantic segmentation in wide plethora of agricultural applications has been emphasized. We have pointed that, nowadays, the dominant solutions use a deep learning framework, typically inspired from the structure of the U-Net (encoder–decoder) and are trained using an iterative, stochastic-gradient based, algorithm. In this context, in this subsection we review deep learning methods that are using rank losses strategies.

Rank Losses: Alternate ways to aggregate loss function were perceived as a way to implement the well known *hard negative mining* concept [32] in a structured way. One of the first results was based on analyzing and developing the hinge loss in SVMs [33]. Further development was based on the analysis from the studies by Shalev et al. [27], and by Huan et al. [34], respectively, which highlighted specific shortcomings of the average aggregate loss, especially when working with imbalanced datasets. We note that imbalanced sets are the norm in agricultural image datasets and similar, but less, in general remote sensing databases and this case is definitely met in the studied cases.

These developments sparked interest in exploring alternative aggregate loss formulations, such as using the maximum individual loss as a measure (named "Maximum Loss"). Building on this concept, Lyu et al. [29] introduced the "ATk Loss" or Average Top-k Loss, which averages the largest top k individual losses. This type of loss demonstrated superior performance compared to both "Average Loss" and "Maximum Loss" across various benchmarks in image classification. However, it encounters challenges with derivatives during the learning process. To address these issues, one approach involved smoothing the overall loss using polynomial approximations [28]. These methods emphasize the importance of data instances that produce the largest errors. They have been applied exclusively in image classification tasks.

Top-k Losses in Segmentation: The Top-k loss has been applied in segmentation tasks only at the pixel level, as demonstrated by Ma et al. [35]. They focused on medical imaging, and found that strategies selecting the worst top-k pixels, using various methods, can be beneficial. However, the results and effectiveness vary depending on the problem, with optimal outcomes ranging from selecting the top $k = 90\%$ to $k = 10\%$ in their evaluations. In practical segmentation, Wu et al. [36] ranked pixel values and, in later learning iterations, focused on the largest errors, i.e., the hardest pixels to classify. Their approach was tailored to problems with highly accurate pixel-level annotations, which is not always the case (as pointed in Fig. 1). Unlike prior works, our target are tasks that have errors in labels at pixel level and we build our proposed method based on this prior assumption.

To the best of our knowledge, rank loss in general, and Top-k loss in particular, have not been explored in the area of agricultural image segmentation. Moreover, strategies that

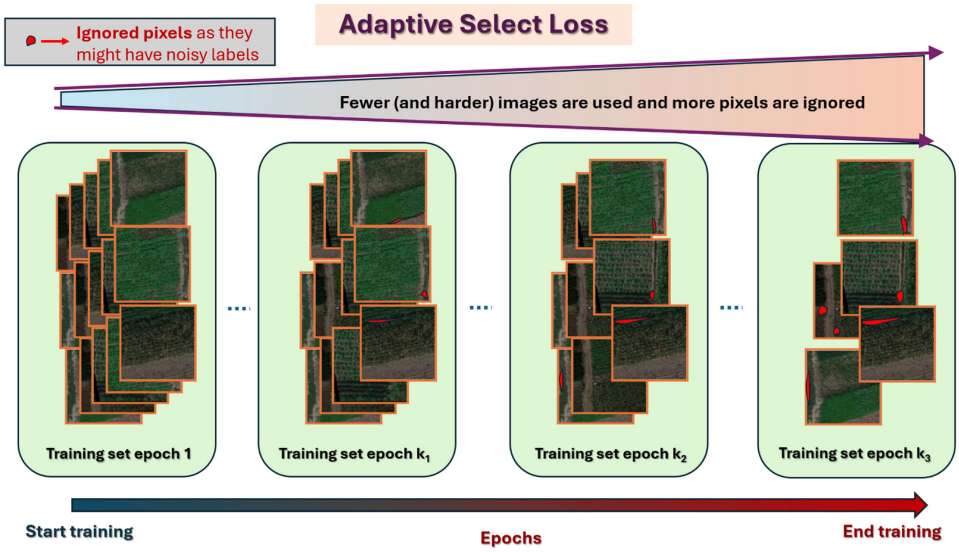


Fig. 2. Overview of the proposed method. As training progresses, the adaptive select loss focuses on the difficult images, considering image-level labels to be reliable. At the same time, it disregards the most challenging pixels (marked in red) as they tend to be noisy, particularly near region boundaries.

accommodate both "few" at image level and "almost all" values at pixel level have yet to be proposed.

II. Method

The proposed method focuses on the strategy of combining the loss function values. An overview of the proposed method may be seen in Fig. 2.

The problem is approached within a deep learning framework, and while the strategy does not make any explicit request about the model and only some general ones about the loss function (to be ascending with magnitude of the error), we will provide details about those too. Thus, in this section we will first describe the select loss strategy, to follow with the preferred loss function and, respectively, model. However, before delving into the proposed procedure, let us formalize the notations.

Formalization: Our problem is assumed to be supervised. Thus, the training data is from the domain $\mathcal{X}$: $\mathbf{x}_i \in X = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ which has associated the target set given as $y_i \in \mathcal{Y}$. Both, $\mathbf{x}$ and y come in different forms, with respect to the problem nature: in image classification $\mathbf{x}_i$ is an image, while y_i is its class; in object localization $\mathbf{x}_i$ is a bounding box in an image, while $\mathbf{y}_i$ contains the geometrical coordinates and the class; in image segmentation $\mathbf{x}_i$ is a pixel from an image, y_i is its class, while N is the total number of pixels within all images. The goal of supervised learning is to identify a predictor function $h : \mathcal{X} \rightarrow \mathcal{Y}$ having the parameters $\mathbf{w}$, that aims to predict target y; the actual prediction is $h_{\mathbf{w}}(\mathbf{x}) = \widehat{y}$. The quality of prediction for the ith sample is measured by $\varepsilon_i = \varepsilon(\widehat{y}_i, y_i)$.

The ε is named the loss function and practical examples include cross-entropy loss, mean square loss, Hubert loss, cosine loss, or logistic loss.

A. Select Loss Strategy

The core idea of supervised machine learning is to find the set of (model) parameters that minimizes a cost function over the training data. In this process, each of the training data instances is used as input and the loss function is build from the difference between the prediction and the label. Initially, the loss function is available and is associated with each individual data instance. Subsequently, an aggregation takes place. The most popular aggregation is by averaging all over the data

$$\mathcal{L}_{avg} = \frac{1}{N}\sum_{i=1}^{N}\varepsilon(h_{\mathbf{w}}(\mathbf{x}_i), y_i) = \frac{1}{N}\sum_{i=1}^{N}\varepsilon_i. \tag{1}$$

The aggregation by averaging forms the so-called Empirical Risk Minimization principle [25]. It is popular because it is accompanied by an efficient optimization algorithm and has a solid theoretical base, as highlighted by Bartlett et al. [26]. Due to the nature of the aggregation, it has also been named Average Loss [27], [29]. The Average Loss is applied on a wide plethora of deep learning tasks tasks including the semantic segmentation problems.

Due to the fact that in deep learning the number of samples N is large enough and corroborated with the amount of parameter space $\mathbf{w}$, the learning with ERM is split over subsets of data and model corrections are applied sequentially onto these partial accumulations. If the entire data set is separated using random choice, the learning follows the stochastic gradient principles. Yet, recent studies [33] suggested that the ERM principle is sub-optimal for many tasks; such an example is the Top-k error in image classification. An alternate aggregation is the "Maximum Loss"

$$\mathcal{L}_{max} = \max_{i\in\{1\ldots N\}}\varepsilon(h_{\mathbf{w}}(\mathbf{x}_i), y_i). \tag{2}$$

The Maximum Loss has been showed to lead to improved performance in Top-k image classification [27]. The key observation is that is *more stringent than average loss*, being always larger. However it has drawbacks: for an entire epoch, a single instance matter, which may be a mere outlier; choosing only one

data, may also create problems with the derivation process in the back–propagation.

To address these issues, Lyu et al. [29] developed the "ATk Loss" or Average Top-k Loss. To define it, one requires rearranging the loss values ε_i into a sorted array, $\varepsilon^{[i]}$; here, an ascending order is assumed

$$\varepsilon_i \to \varepsilon^{[i]} : \varepsilon^{[0]} \leq \varepsilon^{[1]} \leq \cdots \leq \varepsilon^{[k]} \leq \cdots \leq \varepsilon^{[N]}. \quad (3)$$

Using this sorted set, $\varepsilon^{[i]}$ the average from the largest k individual losses is calculated. Lyu et al. [29] demonstrated superior performance over both "Average Loss" and "Maximum Loss" across various benchmarks of image classification. However, this average Top-k Loss function is not differentiable with respect to k.

The Maximum Loss [27] has the advantage of being a harder condition than the Average Loss, because it is always larger; if an optimization process drives it towards zero, it guarantees, also, that the Average Loss is nullified. As said, the disadvantage is related to its uniqueness (i.e computed on a single data instance) and the fact that, without particular domain knowledge, there is no practical guarantee that the data instance which enables the maximal loss is not an outlier resulted due to a wrong label.

In general, the maximal loss (and Atk loss, in particular) values are caused by outlying data instances. These can be either: 1) *hard examples* when the annotation process is accurate and when is desirable to anchor the training on these outliers, especially in the later stages, forcing the model to predict them correctly (i.e. "to master the domain"); 2) *noisy labels*, when the annotation process is loose or inexact and when the outliers should be avoided, since forcing the model to predict them correctly means pushing towards wrong directions and, thus, hurting the overall performance. Concluding, prior knowledge about *the quality of annotation* should be the driving force in how these outlier instances are used during the training process. Prior information about annotation quality is one aspect which is typically part of the so-called domain knowledge.

We seek the formulation of an envelope of the individual loss functions that is able to accept specific domain knowledge and address either direction, because we have to accommodate both cases, but at different levels.

The maximum approach often encounters issues with convergence due zero derivatives if none of the instances in the current batch is the largest. To mitigate this, one strategy [29] involves smoothing the loss function, such as by considering the average of the largest k losses in the average Top-k (Atk)

$$\mathcal{L}_{atk} = \frac{1}{k} \sum_{i=N-k+1}^{N} \varepsilon^{[i]}. \quad (4)$$

Lyu et al. [29] highlighted that this equation corresponds to the empirical risk minimization (ERM) approach if $N - k + 1 = 0$; also the Maximum Loss is found if $k = 1$. From a practical perspective, they have demonstrated that by using $k = 0.2N$ the performance is improved in ImageNet classification: marginally when quantified by Top-1 error and significantly with Top-5 error. A point of emphasis is that the performance is dependent on choosing k.

Instead of the last k, if a specific group of losses, between m and M is to be selected from the ordered series, the overall loss will be named ASL and can be expressed as

$$\mathcal{L}_{ASL} = \frac{1}{k} \sum_{i=m}^{M} \varepsilon^{[i]}. \quad (5)$$

Here, by setting $m = N - k + 1$ and $M = N$, the "Average Top-k" from the earlier equation is obtained. Conversely, if $m = 0$ and $M = k$, the smallest k losses are selected, leading to the "Average Bottom-k" (ABk) variant.

Using m and M, we formulate the ASL, our core proposal. This formulation can be adjusted to be sequentially a top followed by bottom selection. Either of them can address different practical cases and the choice is dependent on prior assumptions/knowledge about the problem. Key ideas with respect to these assumptions are: 1) when the labels y_i are accurate, "Average Top-k" (Few) is preferable as it focuses on hard examples; these establish more clear borders between different categories; 2) when the labels y_i are noisy, "Almost All but k" ("Average Bottom-k") is preferable as it ignores k examples, marked by the largest loss values and which have potentially incorrect labels; if one would force the learner to focus on these noisy examples, the learner would construct incorrect borders that would decrease the performance in the testing; by removing these examples from the overall loss, we limit the incorrect borders and the decrease in performance. Adaptively aggregating the two, we obtain a method that can be adjusted to various situations.

Precise practical values will be shown in Section III, where, for each database, an ablation study will be presented.

B. Discontinuity and Smoothing

While intuitively and algorithmically, one can implement ordering and selecting according to hyperparameters m and M in the Adaptive Select Loss, there are some limitations. In its precursor form, Top-k loss strategy, Lyu et al. [29] used gradient descent for optimization; yet such an approach is functional only for very large $M - m$.

However when the number of instances is limited (e.g. images in segmentation problems), in parallel to choosing very small $M - m$, the abrupt change in selecting instances, creates problems in convergence of the gradient descent. To alleviate this issue, additional smoothing of the loss function and thus, alleviating discontinuities in the derivatives is required.

Further development of the method is based on the indicator function. This is denoted by $\mathbb{1}(u)$ and is defined as 1 when u is true, and 0 otherwise. Using this, the summation in (5) can be written over the entire data, while it selects only those terms in the $[m,\ M]$ range

$$\mathcal{L}_{ASL} = \frac{1}{M-m} \sum_{i=1}^{N} \mathbb{1}(m < i) \cdot \mathbb{1}(i \leq M) \cdot \varepsilon^{[i]}. \quad (6)$$

The product between two indicator functions creates a boxcar representation, $b_{m,M}(i) = \mathbb{1}(m < i) \cdot \mathbb{1}(i < M)$, where the nonzero is used to select the aimed examples. On other hand, the indicator is a generalized version of the Heaviside step function,

$H(x)$, which is

$$H(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{7}$$

thus, $H(x) = \mathbb{1}(x \geq 0)$.

To construct the smoothing, we start form the approximation of the Heaviside step function based on a hyperbolic tangent (logistic function)

$$H(x) \approx \frac{1}{2} + \frac{1}{2}\tanh \alpha x \tag{8}$$

where α controls the transition: a larger α leads to a sharper transition.

The boxcar approximation is, thus

$$\begin{aligned} b_{m,M}(i) &= \mathbb{1}(m < i) \cdot \mathbb{1}(i < M) \\ &= H(i-m) \cdot (1 - H(i-M)) \\ &\approx \left(\frac{1}{2} + \frac{1}{2}\tanh \alpha(i-m)\right) \cdot \\ &\quad \cdot \left(\frac{1}{2} - \frac{1}{2}\tanh \alpha(i-M)\right) \approx \mathbb{b}_{\alpha,m,M}(i). \end{aligned} \tag{9}$$

In this case the ASL becomes:

$$\mathcal{L}_{ASL} = \frac{1}{M-m}\sum_{i=1}^{N} \mathbb{b}_{\alpha,m,M}(i) \cdot \varepsilon^{[i]}. \tag{10}$$

C. ASL in Segmentation. Specific Loss Functions

In semantic segmentation, where the number of instances N equals the total number of pixels across all images, the (average) loss strategy can be rewritten as a nested sum

$$\mathcal{L}_{avg} = \frac{1}{I \cdot P}\sum_{i=1}^{I}\sum_{p=1}^{P} \varepsilon_{ip} \tag{11}$$

where I is the number of images and each of them is assumed to have the same number of pixels, P. The loss, here, is computed first at pixel level, p and, than, aggregated at image level, i. Thus, ε_{ip} is the loss value of pixel p from image i.

In similar manner, the ASL for semantic segmentation may be rewritten as

$$\begin{aligned} \mathcal{L}_{ASL} &= \frac{1}{(M_1 - m_1)(M_2 - m_2)} \cdot \\ &\quad \cdot \sum_{i=1}^{I}\left(\mathbb{b}^{I}_{\alpha_1,m_1,M_1}(i)\sum_{p=1}^{P}\mathbb{b}^{P}_{\alpha_2,m_2,M_2}(j)\varepsilon^{[ip]}\right). \end{aligned} \tag{12}$$

Here, the boxcar continuous approximation $\mathbb{b}^{I}_{\alpha_1,m_1,M_1}(i)$ controls the selection at image level, while $\mathbb{b}^{P}_{\alpha_2,m_2,M_2}(j)$ controls the selection at pixel level.

To successfully apply this strategy in semantic segmentation, one needs loss functions that are computable at pixel level. While the standard cross entropy meets this constraint, the popular Dice Coefficient (used, for instance, by Niu et al. [13]) is computable at image level only. Thus a continuous form of the Dice Coefficient [37] is preferable for its ability to rank and select values at the pixel level

$$cDC = \frac{A \cap B}{\psi|A| + |B|}, \quad \psi = \frac{\sum_i a_i b_i}{\sum_i a_i sign(b_i)}. \tag{13}$$

For the continuous Dice Coefficient y_i is assumed to be in one-hot encoding form: a vector of C (classes) length. Thus y_{ci} is the label which indicate that pixel i is from class c and $\widehat{y_{ci}}$ is the prediction in the same location. The continuous Dice Coefficient for the entire image becomes

$$cDC = \frac{1}{C}\sum_{c=1}^{C}\left(1 - \sum_{i=1}^{P}\rho(\widehat{y_{ci}}, y_{ci})\right) \tag{14}$$

where

$$\rho(\widehat{y_{ci}}, y_{ci}) = \frac{1}{\sum_{i=1}^{P}\widehat{y_{ci}}^2 + \sum_{j=1}^{P} y_{ci}^2}\widehat{y_{ci}}y_{ci}$$

a) Choosing parameters: One important point to highlight is that previous studies that implied rank-based losses have primarily dealt with datasets containing hundreds of thousands or even millions of training examples, where sorting such large arrays can be time-consuming. However, in the context of agricultural image segmentation, datasets are typically smaller (with image in the order of thousands), making it feasible to sort and select values based on the chosen m and M. In this case, sorting a few thousand values is less demanding. Furthermore the loss function vector can be maintained with only a linear update required after each forward pass. If the resulting loss value is sufficiently small, the weights from the sigmoid functions can be disregarded, eliminating the need for a backward step to adjust weights.

In previous works, such as by Lyu et al. [29], the optimization algorithms like block coordinate descent were recommended for finding the optimal hyperparameter k in the Average Top-k Loss. However, when applied to agricultural image segmentation, convergence issues may arise, at image level aggregation, due to the limited number of instances. The previously discussed smoothing procedure addresses this issue.

Unlike the behavior at image level, at pixel level, an image contains (tens of) thousands of pixels, making it cumbersome to sort all of them. To address this, and since most of the pixels will be utilized, we employ a dispersion-based approach (denoted by σ). In this approach, the threshold $M - m$ is determined not by the number of instances, but by a specific percentage or statistical measure σ. The precise value will be found empirically as further detailed in the evaluation section.

With respect to α, practical evaluation shows the best choice. Yet, intuitively, the usage should be that of smoothing. In the first epoch of training, the loss should contain all individual losses, as one has to have them before being able to sort. The selection process takes place in the later stages. Best behavior is a "smooth" one: Choose α, m, M such that the transition between stages is gradual. Thus, initially large α (e.g., 20) with very large $M - m$ ($\approx N$) should be used and, while iterating, both α and $M - m$ decreases, with small increments, until the desired values.

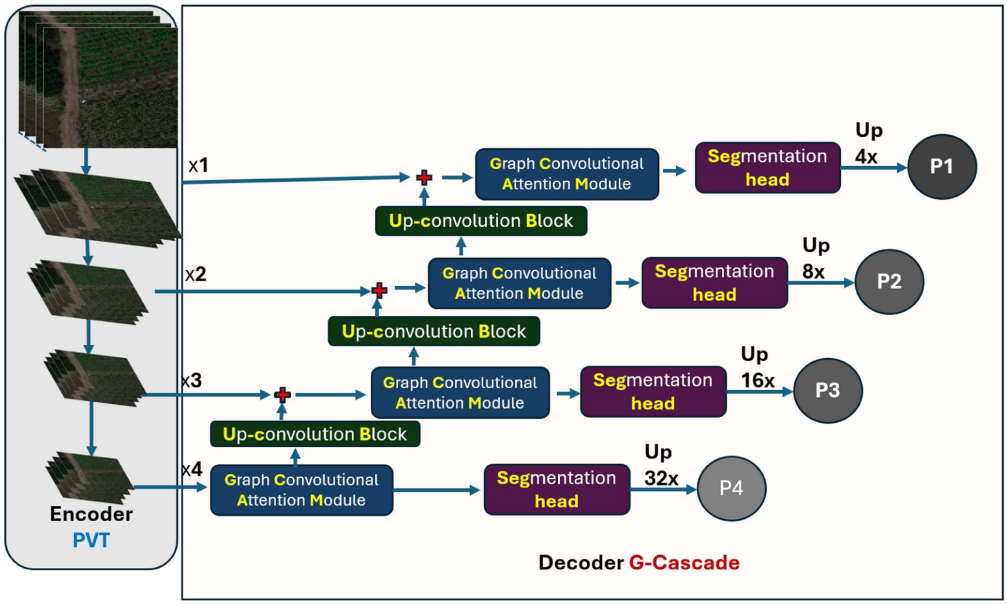


Fig. 3. G-Cascade model [38] using a PVT encoder [39], illustrated over an hyperspectral image from UAV-HSI dataset.

In summary, compared to the Average Top-k Loss, which was designed solely for image classification, the ASL extends the concept to segmentation, a much more complex problem. ASL improves performance by addressing several key aspects as follows.

1) The number of training images is significantly smaller, so the method introduces smoothing to avoid convergence issues caused by limited data.
2) It adapts ranking loss theory to operate at both the image and pixel levels: selecting the "largest few" losses at the image level, and "almost all except the largest" at the pixel level.

In contrast, Average Top-k Loss selects "almost all, but few" at the image level, resulting in less focus on the hardest examples.

D. Segmentation Model Architecture

As emphasized earlier, the "select loss" strategy is general and can be applied within conjunction of many learning models. Yet, to show its potency we focus on a strong model that, by itself is more than capable of being competitive.

The hereby choice is the Cascaded Graph Convolutional Attention Decoder (G-Cascade) architecture [38]. Similar to the seminal U-Net model [14], this architecture employs the encoder-decoder duality. However, the G-Cascade model advances previous alternatives by progressively refining multi-stage feature maps generated by hierarchical transformer encoders with an efficient graph convolution block.

In the G-Cascade scheme, illustrated in Fig. 3, the encoder is based on the Pyramid Vision Transformer(PVT) architecture [39]. This encoder uses a self-attention mechanism to capture long-range dependencies, while the decoder refines the feature maps and maintains long-range information through the global receptive fields of the graph convolution block. The core principle is derived from the Visual Transformers (ViT) [40] family, processing each image as a series of fixed-length tokens (patches) that serve as inputs to multiple Transformer layers for classification.

With respect to the encoder, PVT, the input image $\mathbf{x}_i$ of size $H \times W \times D$ is divided into $\frac{HW}{4^2}$ patches, each measuring $4 \times 4 \times D$. These flattened patches are then fed into a linear projection to produce embedded patches of size $\frac{HW}{4^2} \times C_1$. Next, these embedded patches, combined with positional embeddings, are passed through a Transformer encoder having L_1 layers, resulting in a reshaped feature map $\mathcal{F}_1$ of size $\frac{H}{4} \times \frac{W}{4} \times C_1$.

Similarly, by working on the feature map from the previous stage, subsequent feature maps $\mathcal{F}_2$, $\mathcal{F}_3$, and $\mathcal{F}_4$ are generated with strides of 8, 16, and 32 pixels relative to the input image. This feature pyramid $\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4\}$ acts as encoder according to the traditional models of encoder-decoder employed in image semantic segmentation tasks.

With respect to the decoder, the G-Cascade model comprises efficient upconvolution blocks (UCBs) for upsampling features, graph convolutional attention modules (GCAMs) to robustly enhance feature maps, and segmentation heads (SegHeads) to generate the segmentation output. There are four GCAMs, each corresponding to one of the four stages of pyramid features from the encoder. To combine the multiscale features, the upsampled features from the previous decoder block are first aggregated with the features from the skip connections through addition or concatenation. The concatenated features are then processed with the GCAM module to enhance the semantic information. The output from each GCAM is sent to a prediction head, and finally, the four different prediction maps are aggregated to produce the final segmentation output.

Key elements of G-Cascade decoders include the UCB, the GCAMs, and the segmentation head. The GCAM consists of a Graph Convolution Block (GCB) and a spatial attention (SPA)

block [41], which captures local contextual information

$$GCAM(\mathbf{u}) = SPA(GCB(\mathbf{u})) \tag{15}$$

where $\mathbf{u}$ is the input feature/activation map.

The GCB comprises a graph convolution layer and two 1×1 convolution layers, each followed by a batch normalization layer and a ReLU activation layer [38]. The SPA determines where to focus in a feature map and is formulated as

$$SPA(\mathbf{u}) = Sigmoid(Conv([C_{\max}(\mathbf{u}), C_{avg}(\mathbf{u})])) \circledast \mathbf{u} \tag{16}$$

where $Sigmoid()$ is the standard Sigmoid activation function, $C_{\max}$ and C_{avg} represent the maximum and average values obtained along the channel dimension, $Conv$ is a 7×7 convolution layer with padding, and $\circledast$ is the Hadamard product.

The upconvolution block aims to restore resolution and upsample the features of the current layer to match the dimension of the next skip connection. It consists of an UpSampling Up with a scale factor of 2, a 3×3 depth-wise convolution DWC with a batch normalization BN, a $ReLU$ activation, and a 1×1 convolution $Conv$, formulated as:

$$UCB(\mathbf{u}) = Conv(ReLU(BN(DWC(Up(\mathbf{u}))))). \tag{17}$$

The Segmentation head (SegHead) takes as input refined feature maps from the four stages of the decoder and outputs four segmentation maps, formulated as a 1×1 convolution over the feature maps, with the number of output channels equal to the number of target classes for classification.

The output segmentation maps are aggregated by summing all four output segmentation maps $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, and $\mathbf{P}_4$from the four prediction heads of the decoder

$$seg\_predict = \mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3 + \mathbf{P}_4. \tag{18}$$

III. Implementation and Results

The method has been implemented in Python using PyTorch capabilities for deep learning. On a computer having an A4000 GPU, the inference requires 1.10 s for one image with depth from WE3DS dataset (at full resolution) and 2.4 seconds for a hyperspectral image from UAV-HSI dataset.

Following previous works in segmentation [38], [13], we set the pixel loss, at location i, as a linear combination of cross entropy, CE_i and continuous Dice Coefficient, cDC_i

$$\varepsilon_i = 0.3 \cdot CE_i + 0.7(1 - \rho(\widehat{y_{ci}}, y_{ci})) \tag{19}$$

where 0.3 and 0.7 have been empirically determined and are consistent with findings in prior works.

Once the Dice Coefficient in the loss function has been reformulated using (14) to allow pixel-level computation, the overall process proceeds as follows.

1) After the forward pass, the predictions are obtained. At each pixel, the loss is computed by combining the cross-entropy and Dice components as defined in (19).
2) For each image, the image-level loss is computed by averaging only those pixel losses that fall below a predefined threshold (set for each experiment, with further details provided in the ablation studies).
3) The resulting image loss is added to an ordered list based on magnitude.
4) The total loss is then computed according to (12), where the parameters α_1, m_1, and M_1 are discussed in detail in the corresponding ablation experiments subsection.

A. Databases

We evaluate the proposed method on two rather different databases: UAV-HSI [13] and respectively WE3DS [20].

UAV-HSI: This dataset [13], is publicly available.[1] It contains hyperspectral data photographed from an Unmanned Aerial Vechicle (UAV) over two subregions, Makialou Village (MJK_N, MJK_S) and Xijingmeng Villiange (XJM). The hyperspectral data was obtained on September 18th, 2019 with an electric hexacopter. The UAV carried the sensor Pika L hyperspectral image made by Resonon company, which has a spectral range of 385 nm to 1034 nm with a total of 200 bands. The database is annotated, at pixels level with 30 classes with uneven distribution (which is showed in Fig. 4). The database is delivered as being prestructured into training and testing subsets. Each image consists of 96×96 pixels with 200 spectral bands. The training contain 312 hyperimages, 34 are in validation, and 87 in testing set.

WE3DS dataset[2] has been introduced by Kitzler et al. [20]. Two industrial RGB cameras (XIMEA MC023CGSY equipped with 2.3MP Sony IMX174 LLJ-C sensor) in a stereo camera setup were used. The cameras were placed at 4–5 cm one to other and both were set to focus at 12 mm, to obtain the stereo image and reconstruct the depth (D) plane. The equipment was placed at a working height of 90 cm, reached a spatial ground resolution of 0.4 mm/pixel and a ground depth accuracy of 1.6 mm. The depth images were reconstructed using HALCON image analysis tool. The entire ensemble was mounted on a trolley and moved over crop rows; meanwhile the cameras captured images at a frame rate of one frame per second. The acquisition trials were performed in 2020 and 2021, respectively, on the experimental farm of the University of Natural Resources and Life Sciences, Vienna (BOKU) in Groß–Enzersdorf (48^0200 N, 16^0560 E, 154 m above sea level). There, the soil is silty loam chernozem of alluvial origin which is rich in calcareous sediments. Each repetition contained small parcels (2.5 m $\times$ 9 m in 2020, 1.5 m $\times$ 5 m in 2021). Image recording started right after the emergence of the plants and until the plants reached a height of 30cm, resulting in variation. A total of 6224 image pairs on 25 measurement dates from 84 different parcels have been collected. The average plant cover in images is 2.09% (crops 1.37%, weeds 0.72%). A total of 4038 crop and 7506 weed instances were annotated. On average, an image contains 4.5 individual plants (1.6 crop and 2.9 weed instances) or 38,234 plant pixels (25,131 crop and 13,103 weed pixels). The distribution of pixels w.r.t crops in the WE3DS database may be seen in Fig. 5. We emphasize that the y-axis is logarithmic as the "soil" class vastly outnumbers the rest.

[1]The database can be downloaded at: `https://www.scidb.cn/en/detail?dataSetId=6de15e4ec9b74dacab12e29cb557f041`

[2]The dataset is available at https://zenodo.org/records/7457983.

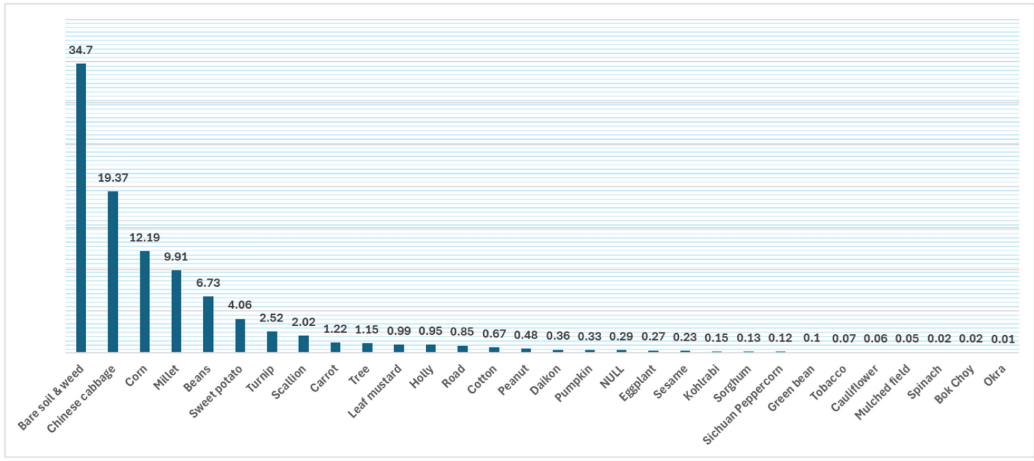


Fig. 4. UAV-HSI database: Crop relative pixel distribution. One might note the discrepancies between classes, although the vertical axis is linear.

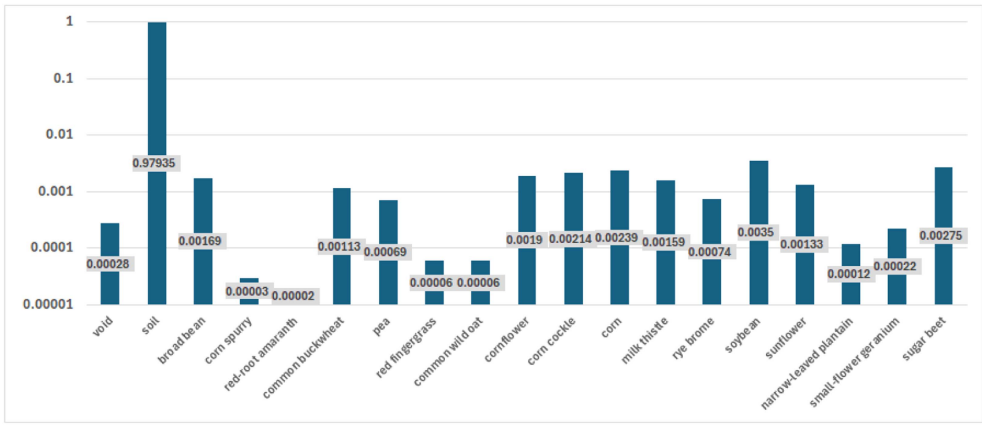


Fig. 5. WE3DS database: plant relative pixel distribution. We emphasize that the vertical axis is *logarithmic* as there is huge discrepancy between the "soil" class and the rest.

LoveDA dataset [42] was acquired from the Google Earth Platform. The images have a spatial resolution of 0.3 meters and cover a total area of over 536 square kilometers. The dataset includes data from three urban areas (Nanjing, Changzhou, and Wuhan), as well as so called rural image that contain agricultural areas. Each image has a resolution of 1024×1024 pixels. To ensure a fair and consistent evaluation, we followed the same data partitioning protocol as previous works [23], [24], enabling direct comparison. Specifically, the dataset is divided into 2522 images for training, 834 for validation, and 835 for testing. The annotated classes include: "background", "building", "road", "water", "barren", "forest" and "agriculture". Unlike the previous mentioned datasets, in this case various crop types are grouped under a single "agriculture" label. It is worth noting that this dataset also exhibits an imbalanced class distribution (as illustrated in Fig. 6). Combined with significant variation in scale, object size, and surface appearance, these factors make LoveDA a particularly challenging benchmark.

The datasets have been chosen to be different and one might note the highly uneven distribution between classes in all sets, and thus increasing the difficulty.

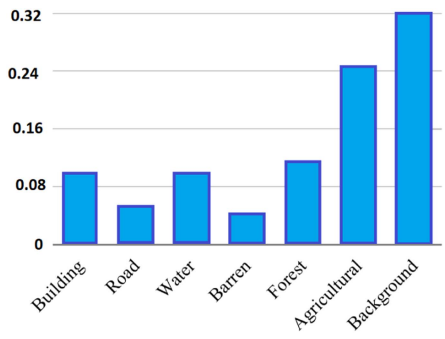


Fig. 6. LoveDA database: class relative pixel distribution.

B. Convergence of the Training

Previous studies on Top-k loss, such as those by Shalev et al. [27] and Lyu et al. [29], were built upon the foundational work of Ogryczak and Tamir [43] when discussing convergence. However, Ogryczak and Tamir's approach assumes that the functions $\varepsilon^{[i]}$ are either *linear* or can be effectively *approximated by*

piecewise linear functions. This assumption is quite restrictive and difficult to satisfy in practice. When the aggregation involves more than just a few terms, and several of these terms do not meet the linearity condition, the summation not only deviates from the theoretical framework but also fails to converge. This issue becomes more pronounced when the number of instances is small, which is often the case in agricultural image segmentation. Alleviation by using many of the largest, such the one used by Lyu et al. [29] helps, but it diminishes the practical relevance in the context of hard negative mining.

Also Lyu et al. [29] demonstrated that there exists an optimal solution, provided that all individual loss functions are convex. The same principle applies to hereby proposed method: if all *individual losses are convex*, the boxcar approximator $\mathbb{b}(\cdot)$ from (10) results in nonzero scaling factors. Since the weighted sum of convex functions is also convex, the proposed method can converge. However, the limitation remains: when the sum involves only a few terms, any deviation from the convexity assumption becomes significant.

Another important point is that the primary motivation for using Top-k rank losses is to focus on the hard examples, as learning these is more challenging than optimizing over the average. Intuitively, the more difficult an example is, the more it resembles an outlier, making it less likely that optimizing the loss over such examples will follow the same convergence path as optimizing over the entire dataset. Nonetheless, practical evaluations have shown that even when $M - m$ is sufficiently larger than 1 (e.g., $M - m > 10$) and a smoothing process with α reasonable (e.g., smaller than 8) is applied, the learning does converge.

The "almost all" strategy employed at pixel level, which uses the vast majority of pixels and discards only a small portion, has minimal impact on convergence. In fact, by removing instances with potentially incorrect labels, convergence is often improved.

C. Evaluation Metrics

To objectively asses the quality of various semantic segmentation solutions the following metrics have been used:

1) Overall accuracy (OA) and Kappa coefficient are based on the confusion matrix, having elements c_{ii} :

$$OA = \frac{1}{N}\sum_{i}^{C} c_{ii} \tag{20}$$

$$Kappa = \frac{N\sum_i^C c_{ii} - N\sum_i^C (c_{i+}\cdot c_{+i})}{N^2 - \sum_i^C (c_{i+}\cdot c_{+i})} \tag{21}$$

where C is the number of classes and thus the row length in the confusion matrix, c_{ii} are the diagonal (correct) elements, c_{i+} and c_{+i} represents the sum of the elements in the row and respectively the columns where c_i is located.

2) Intersection over Union (IoU), per class

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c} \tag{22}$$

where TP_c is the number of true positives, FP_c is the number of false positives, and FN_c is the number of false negatives for a class c. IoU is also known as the Jaccardi index. For the entire database, the average of IoU over class is considered.

3) Dice score, defined as

$$DSC(Y,\widehat{Y}) = \frac{2\times|Y\cap\widehat{Y}|}{|Y|+|\widehat{Y}|}\times 100 \tag{23}$$

where Y and $\widehat{Y}$ are the ground truth and, respectively, the predicted segmentation map.

4) 95% Hausdorff Distance (HD95) which is the greatest of all the distances from a point in the ground truth set Y to the closest point in predicted segmentation map $\widehat{Y}$ and formally defined as

$$D_{H95}(Y,\widehat{Y}) = \max\{\max_{y\in Y}\min_{\widehat{y}\in\widehat{Y}} d(y,\widehat{y}), \max_{\widehat{y}\in\widehat{Y}}\min_{y\in Y} d(y,\widehat{y})\}. \tag{24}$$

In this case, smaller values are better.

5) F1 score, which first is computed per-class and than averaged for all classes. It is defined as

$$F_1 = \frac{2\cdot \text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}} \tag{25}$$

where *precision* for a given class measures the proportion of pixels predicted as belonging to that class that are correctly classified, while *recall* for a given class measures the proportion of ground truth pixels of that class that are correctly identified by the model.

The first metrics have been used in the introductory articles: [13] for the UAV-HSI and, respectively, [20] for the WE3DS database. We add the rest for a more thorough evaluation. F1 score was used for LoveDA dataset.

D. Experiments

Experiments on UAV-HSI: Input images on this dataset have 200 planes and therefore in the G-Cascade model, $D = 200$. Visual examples of segmentation on this dataset may be seen in Fig. 7.

The experiments have been divided into two phases. First, we present the **ablation study** which are experiments regarding the influence of various parameter choices of Adaptive Select Loss over the final performance. These results are presented in Table I. There, σ is the percentage of the pixels discarded from each image when computing the loss function. As one can see, a relatively large portion of the pixels is discardable and thus, potentially, noisy; this is explainable by the relatively large percentage of pixels near the border, when compared to crop (region) size and image size. With respect other parameters, when taking as reference the ERM strategy, we manage to hurt the performance only in two situations: first when too few images (e.g. M = 320, m = 0.5% means only 6 images) were used for the final stages of the training and a decrease was produced and, respectively, when the smoothing was nonexistent (e.g., α is kept constant at 30) and the network did not converged (produced a single class). Otherwise the ASL strategy always improved.

Secondly, the best solution has been compared to previous works and alternate architectures; these results may be followed

TABLE I
INFLUENCE OF VARIOUS PARAMETERS OF THE ADAPTIVE SELECT LOSS EVALUATED ON THE UAV-HSI DATABASE

Image Strategy M=312	**Pixel Strategy**	**Avg OA**	**Avg Kappa**	D_{H95}	**Dice**
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 5\%)$	σ=8%	87.28	0.856	3.37	92.17
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 25\%)$	σ=8%	86.52	0.840	4.10	90.95
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 0.5\%)$	σ=8%	85.12	0.811	6.47	88.95
$\alpha = 8$, $m = (100\% \downarrow 5\%)$	σ=8%	87.15	0.846	3.51	92.01
$\alpha = 30$, $m = (100\% \downarrow, 0.5\%)$	σ=8%	nc	nc	nc	nc
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 5\%)$	σ=1%	86.31	0.836	4.01	90.61
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 5\%)$	σ=5%	87.12	0.839	3.79	91.67
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 5\%)$	σ=8%	**87.28**	**0.856**	**3.37**	**92.17**
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 5\%)$	σ=12%	87.09	0.841	3.81	91.77

The architecture is the G-Cascade. All parameter transitions are linear. $(a \downarrow b)$ - refers to decreasing from a to b. We recall that 312 images are in the training set. For the D_{H95} metric better is lower, while for the rest better is higher. "nc" stands for not converged. For the ease of comprehension, we repeat the first line into the last part.

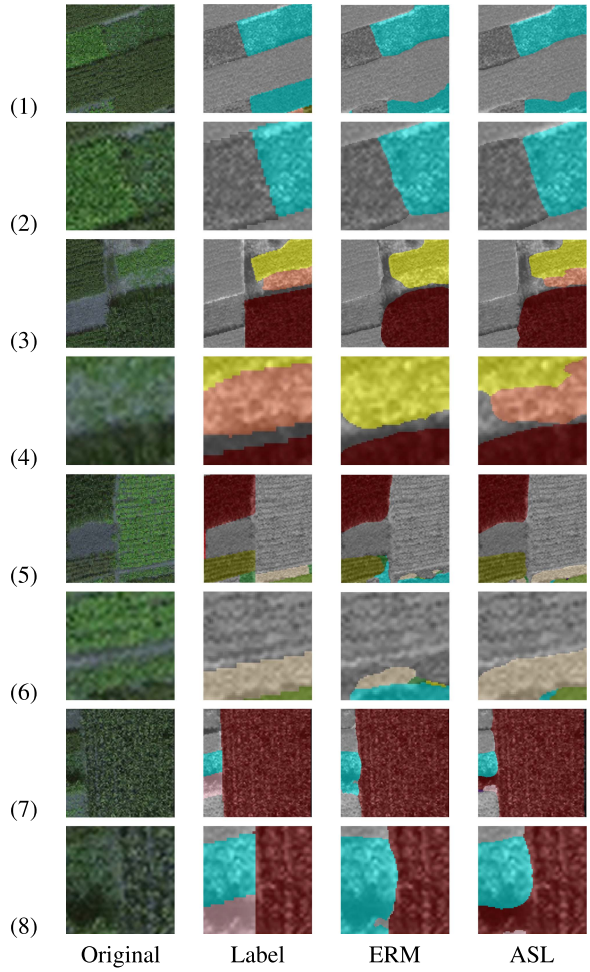


Fig. 7. Examples of segmentation from the UAV-HSI database. Odd rows (1, 3, 5, 7) are the original images, while in the even rows (2, 4, 6, 8) are zoom-in cuts from the previous rows that enhance the differences.

TABLE II
COMPARATIVE PERFORMANCE ON THE UAV-HSI DATABASE

Architecture(Loss)	**AvgOA**	**Kappa**	D_{H95}	**Dice**
Average Loss - ERM				
SegFormer [45]	76.18	0.731	8.15	85.65
Swin-UNet [44]	77.15	0.728	8.24	85.28
TransUNet (CE) [13]	78.64	0.7456	n/a	n/a
HSI-TransUNet (CE+Dice)[13]	86.05	0.8347	n/a	n/a
TransUNet (CE+Dice)-us	83.25	0.812	7.12	87.15
G-Cascade (CE)	84.15	0.825	6.32	88.17
G-Cascade (CE+Dice)	86.56	0.8399	4.15	90.87
Adaptive Select Loss - ASL				
SegFormer (CE+cDice)	79.24	0.748	6.23	89.15
Swin-UNet (CE+cDice)	79.03	0.741	6.55	88.28
G-Cascade (CE+cDice)	**87.28**	**0.856**	**3.37**	**92.17**

For D_{H95} metric better is lower, while fore the rest better is higher.

in Table II. With respect to prior works, we have found only reports by Niu et al. [13] that have relied on a strong Visual transformer based architecture; we used only models that are at least comparable. We report several baselines where the aggregation is done by ERM. We start by our implementation of TransUNet having Cross Entropy and Discrete Dice Coefficient as loss function–denoted by TransUNet–CE+Dice (us) in Table II. To enhance the comparison we have included two popular transformer based models, namely Swin-UNet [44] and SegFormer [45] (variant MiT-B2); public author code have been used. We also report two versions with G-Cascade, one using only Cross Entropy and one using both Cross Entropy and Discrete Dice Coefficient. We emphasize that the architecture used here (G-Cascade) is slightly stronger than the one used in the prior work, TransUNet. Furthermore, we report multiple metrics to offer a better view of the problem. We note that the proposed solution improves by a noticeable margin, an already high result.

The confusion matrix of the solution based on ASL is showed in Fig. 8 and is able to show more details about the results.

Experiments on WE3DS: Input images on this dataset have four planes and therefore in the G-Cascade model, $D = 4$. Characteristic to this database is that most of the images is covered by the "soil" class, while objects are rather small in the center. Visual results from the segmentation alternatives, on this dataset, may be seen in Fig. 9. Again, the experimentation was separated in two phases.

Initially, the influence of the parameters has been analyzed and results are showed in Table III. Due to the smaller objects (which also means that the vast majority of pixels are "soil"), the amount of arguable pixel annotations is proportionally smaller. Consequently, the percentage of discardable pixels by the ASL strategy, this time, is only 0.5%. In the same time, at image level the successfully strategy and the overall behavior replicates,

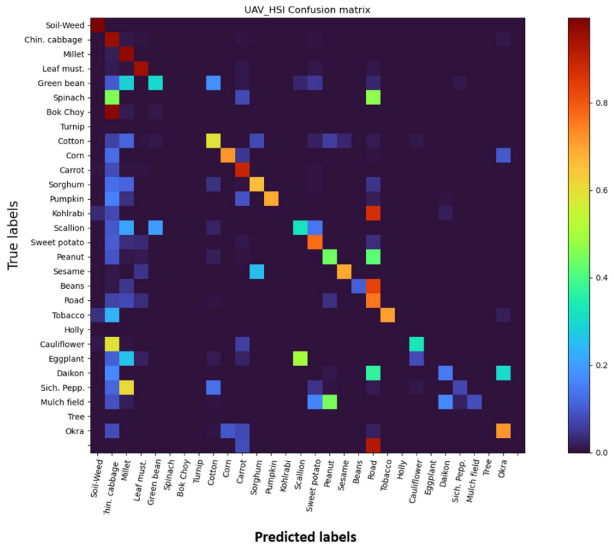


Fig. 8. Confusion matrix of the G-Cascade with ASL aggregation over the testing set from UAV-HSI database.

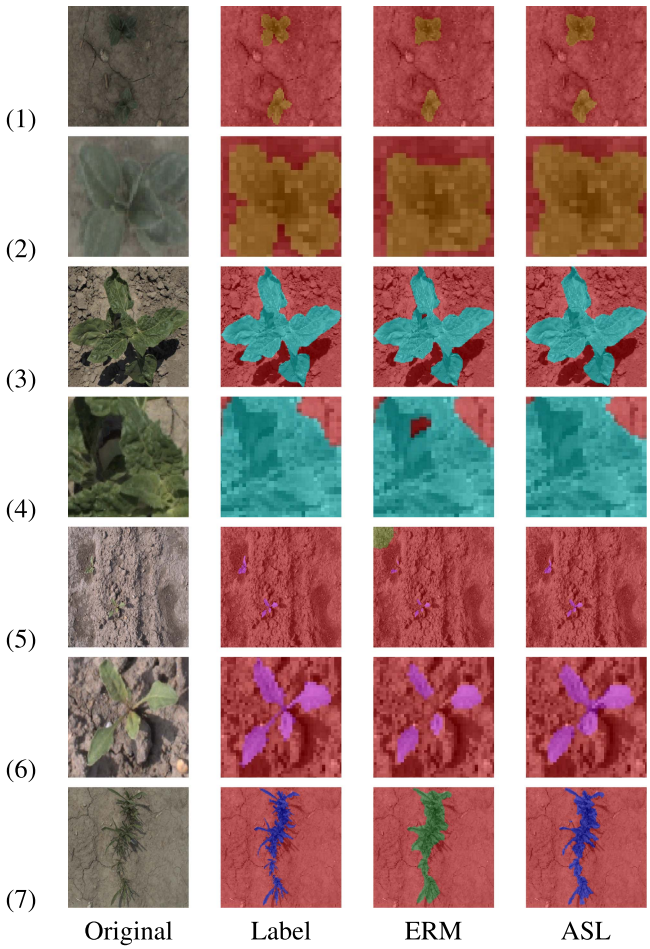


Fig. 9. Examples of segmentation from the WE3DS database. In this case, as the majority of the image is covered by the soil class, we focus on the plant of interest. Odd rows (1, 3, 5, 7) - original images, while in the even ones (2, 4, 6) are zoom-in crops that enhance the differences.

TABLE III
INFLUENCE OF VARIOUS PARAMETERS OF THE ADAPTIVE SELECT LOSS EVALUATED ON THE WE3DS DATABASE

Image Strategy M=1540	Pixel Strategy	mIoU	D_{H95}	Dice
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 5\%)$	σ=0.5%	95.57	1.09	96.62
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 25\%)$	σ=0.5%	93.52	2.10	93.45
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 0.5\%)$	σ=0.5%	89.17	3.47	89.15
$\alpha = 8$, $m = (100\% \downarrow 5\%)$, M=1540	σ=0.5%	87.15	3.51	92.01
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 5\%)$	σ=10%	84.31	4.54	85.07
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 5\%)$	σ=5%	88.89	2.45	89.67
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 5\%)$	σ=1%	93.27	1.45	95.28
$\alpha = (20 \downarrow 8)$, $m = (100\% \downarrow 5\%)$	σ=0.5%	**95.57**	**1.09**	**96.62**

The architecture is the G-Cascade. We recall that 1540 images are in the training set. For the ease of comprehension, we repeat the first line into the last part.

TABLE IV
COMPARATIVE PERFORMANCE ON THE WE3DS DATABASE

Architectures - Loss	mIoU	D_{H95}	Dice
Average Loss - ERM			
ESANet - CE[20]	70.7	n/a	n/a
UNet - CE	62.5	17.3	63.17
DeepLab-v3 -CE	70.7	12.34	71.35
DeepLab-v3 -CE +Dice	75.7	9.18	74.85
SegFormer [45]	84.17	4.61	86.13
Swin-UNet [44]	83.65	4.81	85.44
G-Cascade - CE+Dice	89.77	2.20	90.44
Adaptive Select Loss - ASL			
DeepLab-v3 -CE +cDice	80.11	6.32	79.85
Swin-UNet -CE +cDice	87.12	3.09	90.44
SegFormer -CE +cDice	87.44	2.99	91.03
G-Cascade - CE+cDice	**95.57**	**1.09**	**96.62**

accurately, the behavior from the previous database. Again, the majority of combinations that are using the ASL strategy improve the performance.

Numerical comparisons with previous works may be followed in Table IV. For this database, the introductory work relied on CNN based module and the improvement brought by the use of transformer based models is significant. To emphasize this aspect we evaluate and report the performance achievable with several popular convolutional models, namely UNet and DeepLab-v3. Also two popular transformer based models have been included Swin-UNet [44] and SegFormer [45] (variant MiT-B2). One may notice that prior work performance is comparable with DeepLab-v3 that uses only Cross Entropy. To further show the efficiency of the proposed method, we have tested the ASL method in conjunction with DepLab-V3 architecture, which uses the convolutional paradigm and we show that in this case, the improvement is equally noticeable.

The confusion matrix for the best solution is reported in Fig. 10 and shows, as expected that the dominant class leads to confusion at pixel level. However, the overall accuracy is much improved.

Experiments on LoveDA: Input images on this dataset have three planes and therefore in the G-Cascade model, $D = 3$. Compared to the previous sets, this one is more even but contains smaller annotated objects. To approach this aspect we have used, in prediction, only the first two maps from the G-Cascade. Table V presents the comparative performance of the proposed ASL method in two variants in contrast to various state-of-the-art

TABLE V
COMPARATIVE PERFORMANCE ON THE LOVEDA DATABASE

Architectures/Method	Background	Building	Road	Water	Barren	Forest	Agriculture	AF	AvgOA	mIoU
Average Loss - ERM										
DeepLab-v3	52.29	54.99	57.16	77.96	16.11	48.18	67.79	53.50	52.30	47.62
RAANet [46]	55.02	62.19	65.58	81.03	29.25	54.11	74.07	60.18	58.95	53.93
Macu-Net [47]	59.16	64.08	66.73	81.01	32.23	55.81	75.79	62.12	59.65	54.16
SCAttnet [48]	65.95	71.88	77.04	86.61	50.79	61.19	82.00	70.78	67.31	61.09
ST-Unet [49]	66.68	71.07	75.44	88.68	51.36	64.20	81.42	71.26	69.78	63.04
FsaNet [50]	68.63	75.95	79.38	89.61	53.47	66.32	82.71	72.72	70.59	63.49
SSCBNet [23]	69.45	80.18	**82.25**	**91.86**	53.17	**65.87**	**84.84**	75.32	72.44	65.92
G-Cascade - CE+cDice	78.86	**83.73**	70.8	83.72	63.77	54.2	79.28	73.48	71.12	63.21
Adaptive Select Loss - ASL										
G-Cascade ATB	64.51	76.15	60.25	72.54	70.67	60.88	81.41	69.48	64.29	56.04
G-Cascade ASL	**80.09**	80.92	74.35	75.33	**79.29**	59.53	82.14	**75.95**	**74.25**	**66.17**

Per-class F_1-score, AF (average F_1), AvgOA, and mIoU are listed.

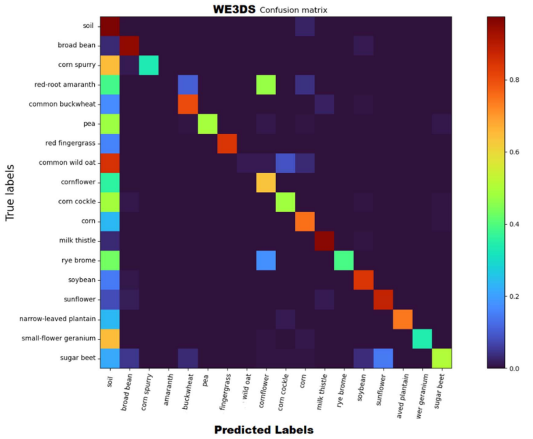


Fig. 10. Confusion matrix of the G-Cascade with ASL aggregation over the testing set from WE3DS database.

models on the LoveDA dataset. Illustrative images from this dataset are in Fig. 11.

Compared to the other two databases used in this evaluation, LovaDA has been acquired directly from satellite and has much lower spatial resolution, therefore presents a different challenge. As shown in Table V, the proposed backbone, while still very strong behaves quite different from previous solutions, by being stronger on different classes. Nevertheless, our solution surpasses the previous proposed models, SSCBNet, with improvements of approximately 2%, 3%, and 3% in AF, OA, and mIoU, respectively. This performance boost can be attributed to some factors: the better boundary detection due to selection of cases with poor region identification and ignoring noisy segment edge definition in annotation due to resolution. For "background" class (which is a heterogeneous, container class), G-Cascade with ASL achieves an improvement of over 10% compared to other best work which we argue is due a combination of high capacity of the deep model and emphasizing hard examples (that contain "background" and "barren") in the training. In Fig. 11 we present some particular examples, that are the same as in [23], to allow better comparison.

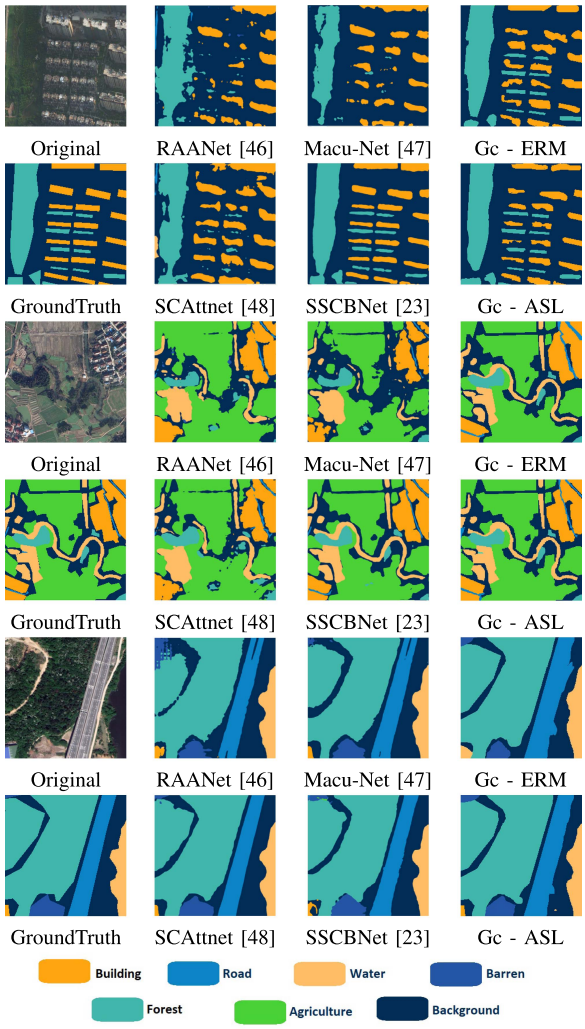


Fig. 11. Examples of segmentation from the LoveDA database. This database was more intensively used for evaluation by prior works, therefore is easier to show visual comparisons. Previous works have been reported in [23]. "Gc" stands for G-Cascade.

IV. DISCUSSION: LIMITATIONS

This article approached the problem of semantic image segmentation in agricultural images. We assumed a deep learning framework where a model is trained to separate pixels according to classes. The key proposal is a strategy to aggregate values for a loss function, based on ranking and selecting individual losses according to prior knowledge of the problem. We have developed and evaluated these on two tasks that are rather different: In

the first one, hyperspectral images acquired from an UAV (i.e., remotely) are used to identify crops, while in the second one, RGB-D images acquired from close-by are used also for crops, but focusing on individual plants. Both segmentation tasks are particularly challenging in a different manner. We have showed in both cases that the proposed framework is powerful, out–competing previous works and other state-of-the art solutions by noticeable margins.

On the UAV-HSI images, remoteness makes details that separate crops to be lost in resolution. For instance we would like to point the attention to Fig. 7, to the midvertical boundary on row (2) where the visual edge (based on region homogeneity) is slightly to left; the region homogeneity is also existing in the training set and is noticed by the segmentation model that learn incorrectly. Since the crops are similar, it is important to emphasize such images during training while placing less emphasis on precise edge details. This is precisely what ASL does: Ignores boundary pixel labels by focusing on hard images. Similar observation can be made on examples from Fig. 7, row (4)–top boundary, row (6) the middle one, or row (8)–edges from the left-hand side.

On the WE3DS images, the scarcity of "non-soil" class limits the examples, and thus outliers become more precious. We would like to point the attention to Fig. 9. Although the original images have high resolution, the plants are small and often slightly blurred. On row (2), the edges of the plant are not clear and annotation quality is poor, making the segmentation process unreliable. In row (4) there is a patch of shadowed soil surrounded by leaf that is wrongly marked as leaf. While these images are from the test set, they are characteristics of the database. The soil class dominates and therefore, important, plant classes have limited examples. Forcing the network to learn such small errors prevents a good generalization. The proposed ASL methodology allows the network to ignore such small details. At the same time, as the training progresses, classes with poor representations of certain plant species lead to the largest errors and ASL forces the network to focus on such images as selecting the top larges errors.

In the context of satellite date from LoveDA, the push for more details in the later stage of training lead to more accurate edges. We would like to point the attention to Fig. 11 to smaller objects as they are marked on the ground truth mask. Although the human observer easily distinguish between them, observing multiple images, one will note that objects that are perceptually similar (ignoring their nature) are set into different classes. Emphasizing their role, in the "few" stage of ASL, combined with the large capacity of the model, the solution is ablue to have better and more accurate object delineation. The most noticeable numeric improvement, compared to previous works is on rarely or heterogeneous represented classes. The ASL forces the network to focus on such cases as selecting the top largest errors.

A. Limitations

Our proposed method significantly enhances task performance by strategically exploiting poor pixel-level separation —primarily through ignoring hard pixels during learning. Moreover, noticeable improvements are observed in tasks requiring precise annotations, thanks to our selection of the hardest image instances in the later learning stages. These gains were demonstrated across visually diverse tasks, further highlighting the robustness and general effectiveness of our approach.

The "No Free Lunch" theorem reminds us that no single algorithm performs optimally across all problem types—success always depends on prior assumptions. Our method is no exception. If all data–both at the pixel and image level–are equally important and accurately annotated, then ranking data offers no clear benefit.

However, we argue that such ideal conditions are rare in agricultural and satellital image segmentation. In practice:

1) Often databases are rather small (up to thousands of instances), containing a mix of easy and hard examples.
2) Pixel-level annotations are often noisy, due to the following:
   a) limited expert labeling time;
   b) loss of resolution, especially in remotely sensed data.
3) The regions of interest (crops) are typically much smaller than the dominant background (e.g., soil).

Therefore, placing additional weight on select, difficult images is not only justified—it is essential for meaningful improvement in real-world conditions.

Another point to be mentioned is the diverse nature of the data used in this study. By incorporation hyperspectral, RGB and RBG-D and, respectively, satellite acquired, UAV recorded and close robot acquisition (with implications of image resolution and detail level), we have explored multiple cases and we have shown that, always, the proposed method leads to improved performance. However there always be other, maybe newer, types of data which may raise unexplored challenges.

V. Conclusion

The proposed strategy, named ASL, which takes into account only the largest *few* loss values at the *image level* and the *almost all* at the *pixel level*, along with a smoothing strategy, has demonstrated beneficial effects in three different image segmentation tasks: Two that focus exclusively on agricultural related problems and one more heterogeneous, but that includes separation of agricultural parcels from other type of terrain. The improvement is more pronounced in harder problems and cases where pixel-level annotations are noisy. The method has been validated in conjunction with a powerful backbone model based on visual transformers. The improvement is noticeable in all conditions and the proposed method also leads to better performance when compared to strong previous solutions.

References

[1] M. Ivanovici, G. Olteanu, C. Florea, R.-M. Coliban, M. Tefan, and K. Marandskiy, "Digital transformation in agriculture," in *Digital Transformation: Exploring the Impact of Digital Transformation on Organizational Processes*, Berlin, Germany: Springer, 2024, pp. 157–191.

[2] M. I. Sadiq, S. P. Rahman, S. Kayes, A. H. Sumaita, and N. A. Chisty, "A review on the imaging approaches in agriculture with crop and soil sensing methodologies," in *Proc. 5th Int. Conf. Intell. Comput. Data Sci.*, 2021, pp. 1–7.

[3] N. Victor et al., "Remote sensing for agriculture in the era of industry 5.0–a survey," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 5920–5945, 2024.

[4] Z. Luo, W. Yang, Y. Yuan, R. Gou, and X. Li, "Semantic segmentation of agricultural images: A survey," *Inf. Process. Agriculture*, vol. 11, no. 2, pp. 172–186, 2024.

[5] M. Ashraf et al., "Novel 3D deep neural network architecture for crop classification using remote sensing-based hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 12649–12665, 2024.

[6] J. U. M. Akbar, S. F. Kamarulzaman, A. J. M. Muzahid, M. A. Rahman, and M. Uddin, "A comprehensive review on deep learning assisted computer vision techniques for smart greenhouse agriculture," *IEEE Access*, vol. 12, pp. 4485–4522, 2024.

[7] P. Zhang et al., "Pixel–scene–pixel–object sample transferring: A labor-free approach for high-resolution plastic greenhouse mapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4401717.

[8] K. Tu et al., "A non-destructive and highly efficient model for detecting the genuineness of maize variety JINGKE 968 using machine vision combined with deep learning," *Comput. Electron. Agriculture*, vol. 182, 2021, Art. no. 106002.

[9] B. Song, H. Yang, Y. Wu, P. Zhang, B. Wang, and G. Han, "A multispectral remote sensing crop segmentation method based on segment anything model using multi-stage adaptation fine-tuning," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4408818.

[10] J. Yuan, D. Wang, and R. Li, "Remote sensing image segmentation by combining spectral and texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 16–24, Jan. 2014.

[11] J. Michel, D. Youssefi, and M. Grizonnet, "Stable mean-shift algorithm and its application to the segmentation of arbitrarily large remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 952–964, Feb. 2015.

[12] K. Yue, L. Yang, R. Li, W. Hu, F. Zhang, and W. Li, "Treeunet: Adaptive tree convolutional neural networks for subdecimeter aerial image segmentation," *ISPRS J. Photogrammetry Remote Sens.*, vol. 156, pp. 1–13, 2019.

[13] B. Niu, Q. Feng, B. Chen, C. Ou, Y. Liu, and J. Yang, "Hsi-transunet: A transformer based semantic segmentation model for crop mapping from UAV hyperspectral imagery," *Comput. Electron. Agriculture*, vol. 201, 2022, Art. no. 107297.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assisted Interv.–MICCAI 18th Int. Conf.*, Munich, Germany, Oct. 5–9, 2015, Proc., Part III 18, Springer, 2015, pp. 234–241.

[15] H. Huang et al., "UNet 3 : A full-scale connected unet for medical image segmentation," in *Proc. ICASSP 2020-2020 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 1055–1059.

[16] J. Chen et al., "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.

[17] Z. Niu, W. Liu, J. Zhao, and G. Jiang, "Deeplab-based spatial feature extraction for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 251–255, Feb. 2019.

[18] M. Pastorino, G. Moser, S. B. Serpico, and J. Zerubia, "Semantic segmentation of remote-sensing images through fully convolutional neural networks and hierarchical probabilistic graphical models," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5407116.

[19] Z. Su, Y. Wang, Q. Xu, R. Gao, and Q. Kong, "Lodgenet: Improved rice lodging recognition using semantic segmentation of UAV high-resolution remote sensing images," *Comput. Electron. Agriculture*, vol. 196, 2022, Art. no. 106873.

[20] F. Kitzler, N. Barta, R. W. Neugschwandtner, A. Gronauer, and V. Motsch, "We3ds: An RGB-D image dataset for semantic segmentation in agriculture," *Sensors*, vol. 23, no. 5, 2023, Art. no. 2713.

[21] X. Li et al., "A synergistical attention model for semantic segmentation of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5400916.

[22] X. Li et al., "AAFormer: Attention-attended transformer for semantic segmentation of remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, 2024, Art. no. 5002805.

[23] X. Li, F. Xu, F. Liu, Y. Tong, X. Lyu, and J. Zhou, "Semantic segmentation of remote sensing images by interactive representation refinement and geometric prior-guided inference," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5400318.

[24] X. Li, F. Xu, A. Yu, X. Lyu, H. Gao, and J. Zhou, "A frequency decoupling network for semantic segmentation of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 5607921.

[25] V. Vapnik, "Principles of risk minimization for learning theory," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 4, 1991, pp. 831–838.

[26] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *J. Amer. Stat. Assoc.*, vol. 101, no. 473, pp. 138–156, 2006.

[27] S. Shalev-Shwartz and Y. Wexler, "Minimizing the maximal loss: How and why," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2016, pp. 793–801.

[28] L. Berrada, A. Zisserman, and M. P. Kumar, "Smooth loss functions for deep top-k classification," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.

[29] S. Lyu, Y. Fan, Y. Ying, and B.-G. Hu, "Average top-k aggregate loss for supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 76–86, Jan. 2022.

[30] S. Hu, X. Wang, and S. Lyu, "Rank-based decomposable losses in machine learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13599–13620, Nov. 2023.

[31] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[32] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 761–769.

[33] M. Lapin, M. Hein, and B. Schiele, "Loss functions for top-k error: Analysis and insights," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1468–1477.

[34] L. Huang, C. Zhang, and H. Zhang, "Self-adaptive training: Beyond empirical risk minimization," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 19365–19376, 2020.

[35] J. Ma et al., "Loss odyssey in medical image segmentation," *Med. Image Anal.*, vol. 71, 2021, Art. no. 102035.

[36] Z. Wu, C. Shen, and A. v. d. Hengel, "Bridging category-level and instance-level semantic image segmentation," 2016, *arXiv:1605.06885*.

[37] R. R. Shamir, Y. Duchin, J. Kim, G. Sapiro, and N. Harel, "Continuous dice coefficient: A method for evaluating probabilistic segmentations," 2019, *arXiv:1906.11031*.

[38] M. M. Rahman and R. Marculescu, "G-CASCADE: Efficient cascaded graph convolutional decoding for 2D medical image segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2024, pp. 7728–7737.

[39] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 568–578.

[40] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–12.

[41] L. Chen et al., "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5659–5667.

[42] J. Wang, Z. Zheng, X. Lu, and Y. Zhong, "LoveDA: A remote sensing land-cover dataset for domain adaptive semantic segmentation," in *Proc. 34th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track (Round 2)*, 2021, pp. 1–12.

[43] W. Ogryczak and A. Tamir, "Minimizing the sum of the k largest functions in linear time," *Inf. Process. Lett.*, vol. 85, no. 3, pp. 117–122, 2003.

[44] H. Cao et al., "Swin-unet: UNet-like pure transformer for medical image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 205–218.

[45] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 12077–12090, 2021.

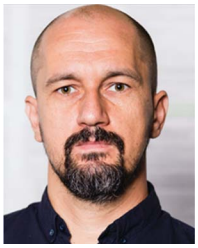
[46] R. Liu et al., "Raanet: A residual ASPP with attention framework for semantic segmentation of high-resolution remote sensing images," *Remote Sens.*, vol. 14, no. 13, 2022, Art. no. 3109.

[47] R. Li, S. Zheng, C. Duan, J. Su, and C. Zhang, "Multistage attention ResU-Net for semantic segmentation of fine-resolution remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 8009205.

[48] R. Niu, X. Sun, Y. Tian, W. Diao, K. Chen, and K. Fu, "Hybrid multiple attention network for semantic segmentation in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5603018.

[49] J. Long, M. Li, and X. Wang, "Integrating spatial details with long-range contexts for semantic segmentation of very high-resolution remote-sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, 2023, Art. no. 2501605.

[50] F. Zhang, A. Panahi, and G. Gao, "FSANet: Frequency self-attention for semantic segmentation," *IEEE Trans. Image Process.*, vol. 32, pp. 4757–4772, 2023.



Corneliu Florea was born in 1980 in Bucharest. He got the master's degree in information engineering and the Ph.D. degree in electronics, telecommunications, and information technology from the University "Politehnica" of Bucharest, Romania, in 2004 and 2009, respectively.

After a stint with digital still camera software industry, currently he is a Professor with University "Politehnica" of Bucharest, within Image Processing and Analysis group. There he lectures on statistical signal and image processing and has introductory courses in computational photography and machine learning. He has authored or coauthored more than 50 peer-reviewed papers and 25 US patents. His research interests include statistical approaches to machine leaning and computer vision.

Laura Florea received the M.Sc degree in information engineering and the Ph.D. degree in electronics, telecommunications, and information technology from the University "Politehnica" of Bucharest, Romania, in 2004 and 2009, respectively.

From 2004, she teaches classes with the University "Politehnica" of Bucharest, where she is currently an Associate Professor. Her interests include automatic understanding of human behavior by analysis of portrait images, image processing algorithms for digital still cameras, medical image processing, computer vision, and statistic signal processing theory.

Mihai Ivanovici (Senior Member, IEEE) received the Ph.D. degree in electronics, telecommunications, and information technology from the Politehnica University of Bucharest, Romania, in 2006.

He is currently a Full Professor with the Transilvania University of Brasov, Romania. His research interests include signal and image processing and analysis, as well as remote sensing and earth observation data analysis.