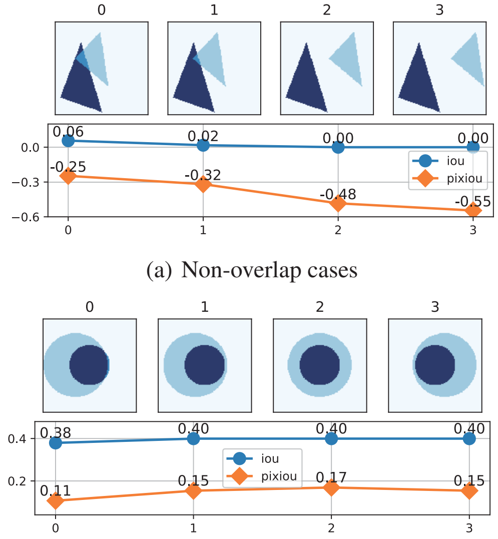
Learning Generalized Intersection Over Union for Dense Pixelwise Prediction

Jiaqian Yu [1] **Jingtao Xu** [1] **Yiwei Chen** [1] **Weiming Li** [1] **Qiang Wang** [1] **Byung In Yoo** [2] **Jae-Joon Han** [2]

Abstract

Intersection over union (IoU) score, also named Jaccard Index, is one of the most fundamental evaluation methods in machine learning. The original IoU computation cannot provide non-zero gradients and thus cannot be directly optimized by nowadays deep learning methods. Several recent works generalized IoU for bounding box regression, but they are not straightforward to adapt for pixelwise prediction. In particular, the original IoU fails to provide effective gradients for the non-overlapping and location-sensitive cases, which results in performance plateau. In this paper, we propose PixIoU, a generalized IoU for pixelwise prediction that is sensitive to the distance for non-overlapping cases and the locations in prediction. We provide proofs that PixIoU holds nice properties as the original IoU. To optimize the PixIoU, we also propose a loss function that is proved to be submodular, hence we can apply the Lovász functions, the efficient surrogates for submodular functions for learning this loss. Experimental results show consistent performance improvements by learning PixIoU over the original IoU for several different pixelwise prediction tasks on Pascal VOC, VOT-2020 and Cityscapes.



(a) Non-overlap cases

(b) Different locations

Figure 1: An illustration of the advantages of PixIoU over IoU: dark blue for groundtruth and light blue for predictions. In 1(a), when the prediction pixels share no overlap with the groundtruth, the original IoU stays always zero; PixIoU continuously decreases if the prediction is getting further (within the finite region). In 1(b), the original IoU stays unchanged if the numbers of pixels in intersection and in union stay unchanged; PixIoU achieves a higher score if the prediction is more centered to the groundtruth.

1. Introduction

Intersection over Union (IoU), also known as Jaccard Index in statistics, is one of the most fundamental methods to compare similarity between data samples in machine learning. In computer vision tasks, IoU is one of the core evaluation method in various benchmarks including object detection, object tracking, semantic segmentation, etc. In general, IoU is defined to calculate the overlap between two given set of elements A and B: IoU $= \frac{A \cap B}{A \cup B}$. For example, for a bounding box regression, A presents the ground-truth box and B presents the prediction box, then the IoU in this case can be formulated as a function of the box location which is often presented by the coordinates of the left-top corner, the width and the height of the bounding box.

For dense pixelwise prediction such as a semantic segmentation task, however, the evaluation is carried out on different variables. At test time, given an image with finite pixels and a finite set of classes $\mathcal{C}$ to predict, the IoU for a label $c \in \mathcal{C}$ measures the overlap between the set of ground-truth pixels $\mathbf{y}$ and the set of predicted pixels $\tilde{\mathbf{y}}$ as:

$$\text{IoU} = \frac{|\{\mathbf{y} = c\} \cap \{\tilde{\mathbf{y}} = c\}|}{|\{\mathbf{y} = c\} \cup \{\tilde{\mathbf{y}} = c\}|}. \quad (1)$$

Following the principle of empirical risk minimization (Vapnik, 1995), the optimal objective to learn for an evaluation

[1] Samsung Research China - Beijing, Beijing, China [2] Samsung Advanced Institute of Technology, Suwon, South Korea. Correspondence to: Jiaqian Yu <jiaqian.yu@samsung.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

is the evaluation itself. Learning a metric sensitive loss function has proven to be better than a default option e.g. a standard cross-entropy loss (Yu & Blaschko, 2015; Yu et al., 2016; Berman et al., 2018; Rezatofighi et al., 2019; Zheng et al., 2020). In general, a more structured and distance-based objective often provides a better interpretation than simply counting the number of mispredicted labels e.g. a Hamming distance (Gillenwater et al., 2015; Ye et al., 2016). To optimize IoU during training stage, the related loss function is usually denoted as the Jaccard loss:

$$\mathcal{L}_{\text{iou}} := 1 - \text{IoU}. \tag{2}$$

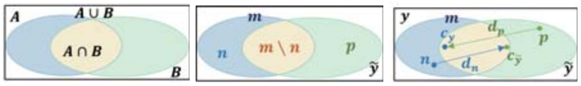
While learning IoU in training presents theoretical advantages, empirically it is often found to suffer from a performance plateau, e.g. in bounding box regression (Rezatofighi et al., 2019). In this work, we also observe that the original IoU fails to provide sufficient gradients to continuously drive optimization in two frequently encountered cases in pixelwise prediction. First, as shown in Figure 1, when the prediction pixels share no overlap with the groundtruth, the IoU stays zero no matter how far the prediction is. Second, with different locations, the IoU stays unchanged as long as the numbers of pixels in intersection and in union stay unchanged, where the best prediction should be the most centered prediction. For both the above cases, human cognition can clearly judge that the optimization should be further performed, while learning over the original IoU will only provide zero gradients in these cases, thus yields suboptimal performance and leads to slower convergence.

In this paper, we propose a generalized IoU for dense pixelwise prediction. The contributions of the work can be summarized as follows:

1. We propose PixIoU, a generalized IoU that is sensitive to the distance and the location of the mispredicted pixels, thus provides better interpretation and non-zero gradients for such cases (Section 3);
2. We demonstrate that PixIoU is invariant to the scale, maintains a lower bound of the standard IoU, and is well-bounded (Section 3.1);
3. We propose a loss function to optimize PixIo, prove that it is submodular w.r.t. the mispredictions, therefore, the Lovász surrogate is applicable (Section 3.2);
4. On several large-scale datasets of pixelwise prediction, experimental results verify that optimizing PixIoU provides efficient convergence rate and consistent improvements comparing to the original IoU (Section 4).

2. Related Work

IoU for bounding box regression Learning IoU during training has been actively investigated in recent years. Yu et al. (2016) proposed to directly learn the IoU during training where the objective is defined as the negative logarithm of the IoU score; the feasibility of the backpropagation is achieved by computing the partial derivative w.r.t. the coordinates i.e. the coordinates of the left-top corner, the width and the height of the box. Rezatofighi et al. (2019) proposed a generalized IoU namely the GIoU, where one need to find the smallest enclosing shape of A and B, which is feasible for two rectangles while not straightforward in practice for arbitrary shapes; GIoU generalizes the IoU so that non-overlapping cases of two boxes can be better evaluated. Zheng et al. (2020) further proposed a distance-based IoU where the distance between two boxes is explicitly computed, and a complete IoU where the ratios of the box is additionally involved to constraint on the optimization.



(a) IoU (b) PixIoU

Figure 2: Illustration of the definitions of IoU and PixIoU.

IoU for pixelwise prediction Optimizing IoU for pixelwise prediction is not straightforward due to the fact that all pixels are taken into consideration. Therefore, some works proposed to use an approximation for learning IoU during training (Nowozin, 2014; Rahman & Wang, 2016). Li (2015) proposed an approach to estimate the Jaccard Index in the presence of incomplete samples. Berman et al. (2018) proposed a surrogate loss function called the Lovász Softmax for semantic segmentation task. Proven that Jaccard loss is submodular (Yu & Blaschko, 2020), the Lovász surrogate yields a convex surface and provide a polynomial computation complexity, thus a tractable surrogate for learning IoU during training. The surrogate have been actively applied in many pixelwise prediction problems including medical image segmentation (Bertels et al., 2019), video object segmentation (Bhat et al., 2020) and general segmentation tasks (Neven et al., 2019; Ying et al., 2019) for learning the Jaccard loss.

3. Method

We first discuss the original definition of IoU for dense pixelwise prediction. For a given set of groundtruth pixels $\mathbf{y} = \{y_i | i \in V\}$, where V is the set of all pixels of size N, and a set of predicted pixels $\tilde{\mathbf{y}}$, we note $\mathbf{n} = \{\mathbf{y} = c, \tilde{\mathbf{y}} \neq c\}$ the set of false negative pixels, $\mathbf{p} = \{\mathbf{y} \neq c, \tilde{\mathbf{y}} = c\}$ the set of false positive pixels for the class c. Denote $\mathbf{m} = \{\mathbf{y} = c\}$ the set of pixels that belongs to c in $\mathbf{y}$. Then $\mathbf{m} \backslash \mathbf{n}$ is the set of the true positives for the class c. We can reformulate the

IoU in Equation (1) as follows:

$$\mathrm{IoU} = \frac{|\mathbf{m} \backslash \mathbf{n}|}{|\mathbf{m} \cup \mathbf{p}|} = \frac{|\mathbf{m}| - |\mathbf{n}|}{|\mathbf{m}| + |\mathbf{p}|}, \tag{3}$$

where $|\cdot|$ presents the size of the set, namely $|\mathbf{m}|$ the number of pixels belongs to c in the groundtruth $\mathbf{y}$, $|\mathbf{n}|$ is the number of false negatives and $|\mathbf{p}|$ is the number of false positives. Thus, the IoU can be written as a function of two sample sets in general, a function of the set of groundtruth and predicted pixels, or a function of the false negatives and the false positives. In the sequel, we will omit the difference between $\mathrm{IoU}(A, B)$, $\mathrm{IoU}(\mathbf{y}, \tilde{\mathbf{y}})$, and $\mathrm{IoU}(\mathbf{n}, \mathbf{p})$. An illustration of IoU is shown in Figure 2(a).

From Equation (3), given a groundtruth $\mathbf{y}$ then $|\mathbf{m}|$ is fixed, the IoU depends only on the number of the false negatives and false positives, while ignoring the form of their existence. For instance, a false positive that is at further *distance* to the groundtruth is currently penalized equally to one that just locates next to the boundary; a set of predicted pixels with zero overlapping i.e. $|\mathbf{m}| - |\mathbf{n}| = 0$ will lead to $\mathrm{IoU} = 0$ no matter $\mathbf{p}$ is.

3.1. PixIoU: a generalized IoU for pixelwise prediction

Motivated by the fact that the weakness of the original IoU mainly comes from the lack of the distance or the location information for each false negative and false positive, we seek to integrate the coordinates information of each mispredictions into the IoU calculation. We propose to penalize each misprediction differently. Formally, we propose the **Pix**elwise **I**ntersection **o**ver **U**nion as follows:

Definition 1 (PixIoU)**.** *Given a set of groundtruth pixels* $\mathbf{y}$ *and a set of predicted pixels* $\tilde{\mathbf{y}}$*, the PixIoU is defined as:*

$$\mathrm{PixIoU} = \frac{|\mathbf{m}| - \langle \mathbf{d_n}, \mathbf{1_n} \rangle}{|\mathbf{m}| + \langle \mathbf{d_p}, \mathbf{1_p} \rangle} + \mathrm{IoU} - 1 \tag{4}$$

where $\mathbf{d_n}$ *is the set of the* distance *from* $\{\mathbf{y} = c\}$ *to* $\tilde{\mathbf{y}}$*,* $\mathbf{d_p}$ *is the set of the* distance *from* $\{\tilde{\mathbf{y}} = c\}$ *to* $\mathbf{y}$*;* $\mathbf{1_n}$ *and* $\mathbf{1_p}$ *are the indicator vectors, e.g.* $\mathbf{1_n} := (x_i)$ *with* $x_i = 1$ *if* $i \in \mathbf{n}$*,* $x = 0$ *otherwise;* $\langle \cdot, \cdot \rangle$ *is a dot product.*

Intuitively, the dot product is carried out in such a way that we compute a sum of the distance from all the false positives to the $\mathbf{y}$, and from all the false negatives to $\tilde{\mathbf{y}}$, respectively. The illustration is shown in Figure 2(b).

In this paper, we propose to calculate the distance as follows: $\mathbf{d_n}(n)$ is the L_2 distances in Euclidean space from a false negative pixel n to the *center* of $\tilde{\mathbf{y}}$, and $\mathbf{d_p}(p)$ is the one from a false positive p to the *center* of $\mathbf{y}$:

$$\mathbf{d_n}(n) = g \circ \mathrm{Euclidean}(n, \mathfrak{c}_{\tilde{\mathbf{y}}}), \tag{5a}$$

$$\mathbf{d_p}(p) = g \circ \mathrm{Euclidean}(p, \mathfrak{c}_{\mathbf{y}}). \tag{5b}$$

Algorithm 1 Computation of PixIoU in Equation (4)

```
Given the groundtruth $\mathbf{y}$, the prediction $\tilde{\mathbf{y}}$
Identify $\mathbf{m} = \{\mathbf{y} = c\}$, $\mathbf{n} = \{\mathbf{y} = c, \tilde{\mathbf{y}} \neq c\}, \mathbf{p} = \{\mathbf{y} \neq c, \tilde{\mathbf{y}} = c\}$, same as for the original IoU
Compute $\mathfrak{c}_{\mathbf{y}}$ and $\mathfrak{c}_{\tilde{\mathbf{y}}}$, the center of $\mathbf{y}$ and of $\tilde{\mathbf{y}}$
Compute $\mathbf{d_n}$ and $\mathbf{d_p}$ as in Equation (5)
$\mathrm{IoU} = \frac{|\mathbf{m}| - |\mathbf{n}|}{|\mathbf{m}| + |\mathbf{p}|}$, $\mathrm{PixIoU} = \frac{|\mathbf{m}| - \langle \mathbf{d_n}, \mathbf{1_n} \rangle}{|\mathbf{m}| + \langle \mathbf{d_p}, \mathbf{1_p} \rangle} + \mathrm{IoU} - 1$
```

where g is a normalization operator. For an arbitrary shape, we can calculate a box center of which the box is the minimal surrounding closure of the shape. We can also calculate a geometric center i.e. the gravity center. In our experiments, we compute the mean value of all pixel coordinates of the foreground pixels as the center. We did not observe significant difference by using different methods.

For the normalization function, a linear normalization e.g. min-max would brings theoretically benefits (c.f. Proposition 1). Empirically, we observe a slight improvement by using the log normalization. We expect that it provides a practical balance for different distances, namely that will not largely reduce effect the of the pixels close to the boundary.

The computation complexity of PixIoU involves marginal additions comparing to the original IoU. Explicitly, given N the number of pixels, the original IoU need $\mathcal{O}(N)$ to identify each misprediction, while the computation of PixIoU is $\mathcal{O}(kN)$ with k additionally involves a mean operation to compute the center coordinates and an Euclidean distance calculation, followed by a dot product between the misprections and the distance vector. The calculation of PixIoU is summarized in Algorithm 1 (c.f. the supplementary materials for a pseudo code).

Lemma 1. *The distance* $\mathbf{d_p}$ *defined in Equation* (5) *has the following properties : (i)* $\mathbf{d_p}(p) \in [0, 1]$*,* $\forall p \in \mathbf{p}$*; (ii)* $\mathbf{d_p}(p) = 0 \iff p = \mathfrak{c}_{\mathbf{y}}$*, which only happens when the center* $\mathfrak{c}_{\mathbf{y}}$ *is not on the area of* $\{\mathbf{y} = c\}$ *and the prediction for this center pixel is wrong. Similarly with* $\mathbf{d_n}$*.*

Proof. (i) can be directly proved by Equation (5) that we design the distance to be normalized. (ii) Zero-distance only occurs when the center pixel $\mathfrak{c}_{\mathbf{y}}$ and one p overlap, which means $\mathfrak{c}_{\mathbf{y}}$ is not on the area of $\{\mathbf{y} = c\}$ and is incorrectly predicted by the prediction action. □

With Definition 1 and Lemma 1, the PixIoU holds the following properties:

Proposition 1. PixIoU *is invariant to the scale of the problem if* $\mathbf{d_n}$ *and* $\mathbf{d_p}$ *are the Euclidean distance in general, or, if* g *is a linear normalization in Equation* (5).

Proof. For an arbitrary set of pixels which forms an area/volume A, if it is scaled by a factor γ, given the distances are defined as linear, any pairwise distance within A will be also scaled by γ, which gives us:

$$\begin{aligned}\mathrm{IoU}(\gamma A,\gamma B) &= \frac{\gamma A\cap\gamma B}{\gamma A\cup\gamma B}=\frac{\gamma\,(A\cap B)}{\gamma\,(A\cup B)}=\frac{A\cap B}{A\cup B}\\ &= \mathrm{IoU}(A,B), \end{aligned}\tag{6a}$$

$$\begin{aligned}\mathrm{PixIoU}(\gamma A,\gamma B) &= \frac{|\gamma A|-\langle\gamma\mathbf{d_n},\mathbf{1_n}\rangle}{|\gamma A|+\langle\gamma\mathbf{d_p},\mathbf{1_p}\rangle}+\mathrm{IoU}(\gamma A,\gamma B)-1\\ &= \frac{\gamma|A|-\gamma\langle\mathbf{d_n},\mathbf{1_n}\rangle}{\gamma|A|+\gamma\langle\mathbf{d_p},\mathbf{1_p}\rangle}+\mathrm{IoU}(A,B)-1\\ &= \mathrm{PixIoU}(A,B).\end{aligned}\tag{6b}$$

This proves the scale-invariance of PixIoU. □

Proposition 2. PixIoU *is always a lower bound of IoU:* $\mathrm{PixIoU}(A,B)\ \le\ \mathrm{IoU}(A,B), \forall A,B\ \subseteq\ V$*; it becomes tighter when the predictions get better and* $\lim_{B\to A}$ *PixIoU*$(A,B)=$ *IoU*(A,B).

Proof. By definition, $\mathbf{n}$ is the set of false negatives which is actually one subset of $\mathbf{m}=\{\mathbf{y}=c\}$, therefore $|\mathbf{n}|\le|\mathbf{m}|$. By Lemma 1, we have $\mathbf{d_n}\in[0,1]$, then $\langle\mathbf{d_n},\mathbf{1_n}\rangle\ge 0$ leads to $|\mathbf{m}|-\langle\mathbf{d_n},\mathbf{1_n}\rangle\le|\mathbf{m}|$, therefore:

$$\frac{|\mathbf{m}|-\langle\mathbf{d_n},\mathbf{1_n}\rangle}{|\mathbf{m}|+\langle\mathbf{d_p},\mathbf{1_p}\rangle}\le\frac{|\mathbf{m}|}{|\mathbf{m}|+\langle\mathbf{d_p},\mathbf{1_p}\rangle}\le 1,\tag{7}$$

which proves that PixIoU $\le$ IoU always holds.

When two sets of predicted pixels getting closer i.e. $B\to A$, which means $|\mathbf{n}|\ \to 0$ and/or $\mathbf{d_n}\ \to 0,\forall n$, as well as $|\mathbf{p}|\to 0$ and/or $\mathbf{d_p}\to 0,\forall p$. Therefore:

$$\begin{aligned}\lim_{B\to A}\mathrm{PixIoU}(A,B) &= \frac{|\mathbf{m}|-\overbrace{\langle\mathbf{d_n},\mathbf{1_n}\rangle}^{\approx 0}}{|\mathbf{m}|+\underbrace{\langle\mathbf{d_p},\mathbf{1_p}\rangle}_{\approx 0}}+\mathrm{IoU}(A,B)-1\\ &= 1+\mathrm{IoU}(A,B)-1=\mathrm{IoU}(A,B).\end{aligned}\tag{8}$$

□

Proposition 3. PixIoU *is well-bounded:* PixIoU $\in[\alpha-1,1]$ *where* $\alpha=\frac{|\mathbf{m}|-\langle\mathbf{d_m},\mathbf{m}\rangle}{|\mathbf{m}|+\langle\mathbf{d_{V\backslash m}},\mathbf{V\backslash m}\rangle}$.

Proof. By Proposition 2, we already know the upper bound of PixIoU $\le$ IoU ≤ 1. The maximal is achieved if and only if when A and B overlap perfectly, that is to say:

$$\begin{aligned}\mathrm{PixIoU} &= \frac{|\mathbf{m}|-\overbrace{\langle\mathbf{d_n},\mathbf{1_n}\rangle}^{=0}}{|\mathbf{m}|+\underbrace{\langle\mathbf{d_p},\mathbf{1_p}\rangle}_{=0}}+\overbrace{\mathrm{IoU}(A,B)}^{=1}-1\\ &= 1+1-1=1\iff A=B.\end{aligned}\tag{9}$$

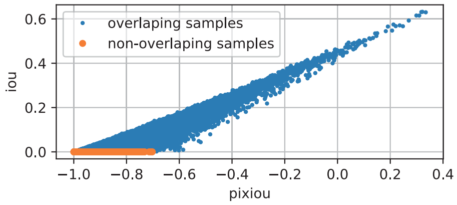


Figure 3: Correlation between IoU and PixIoU over 10k random samples.

For the lower bound, we know that IoU ≥ 0 and IoU $=0$, $\forall$ A and B s.t. $A\cap B=0$. The lower bound of the first part of PixIoU, however, exits when $\langle\mathbf{d_n},\mathbf{1_n}\rangle$ achieves the maximal, and $\langle\mathbf{d_p},\mathbf{1_p}\rangle$ achieves its maximal at the same time. This happens when the prediction B is entirely inverse to A, namely the prediction action causes all possible false negatives and all possible false positives in the given finite region V:

$$\mathbf{n}=\mathbf{m},\quad \mathbf{p}=V\backslash\mathbf{m},\tag{10}$$

which proves the lower bound of the PixIoU exists at $\frac{|\mathbf{m}|-\langle\mathbf{d_m},\mathbf{m}\rangle}{|\mathbf{m}|+\langle\mathbf{d_{V\backslash m}},\mathbf{V\backslash m}\rangle}-1$. □

We illustrate the aforementioned properties as well as the advantages of PixIoU to IoU in Figure 1 and in Figure 3.

3.2. Learning PixIoU for pixelwise prediction

We propose to learn an objective function that directly defined on the PixIoU:

$$\mathcal{L}_{\text{pix}}:=1-\mathrm{PixIoU}.\tag{11}$$

Corollary 1. $\mathcal{L}_{\text{pix}}$ *holds that: (i)* $\mathcal{L}_{\text{pix}}\ge 0$*;(ii) the identity of indiscernible:* $\mathcal{L}_{\text{pix}}(A,B)=0\iff A=B$.

Proof. By Proposition 3, PixIoU ≤ 1 always holds, thus $\mathcal{L}_{\text{pix}}=1-\mathrm{PixIoU}\ge 0$ is true. By Equation 9, $\mathcal{L}_{\text{pix}}(A,B)=0\iff\mathrm{PixIoU}=1\iff A=B$. □

We notice that $\mathcal{L}_{\text{pix}}$ and PixIoU do not hold the symmetry. In general, it does not hold that $\mathrm{PixIoU}(A,B)\approx\mathrm{PixIoU}(B,A),\forall A,B\subseteq V$. Other asymmetry distance measure can be found such as the KullbackLeibler divergence for probability distribution. Regarding the triangle inequality, for special cases, it can be easily checked for special overlapping status between A, B and C. For example, $A\cap C=\emptyset\iff\mathbf{n}_{CA}=\mathbf{m}_A,\mathbf{p}_{CA}=\mathbf{m}_C$. For general case, following the proof of GIoU (Rezatofighi et al., 2019), we randomly generate 10^6 triples of arbitrary polygons (A,B,C) and compute $\mathcal{L}_{\text{pix}}(A,C)$, $\mathcal{L}_{\text{pix}}(A,B)$, $\mathcal{L}_{\text{pix}}(B,C)$. Throughout all samples, it all holds that $\mathcal{L}_{\text{pix}}(A,C)\le\mathcal{L}_{\text{pix}}(A,B)+\mathcal{L}_{\text{pix}}(B,C)$ which empirically validates the triangle inequality.

In brief, PixIoU extends IoU to a more general aspect for measuring the predictions given a groundtruth. Empirically, we observe in our experiments that optimizing PixIoU benefits the performance even when evaluating with IoU.

Motivated by the efficient Lovász surrogates for learning Jaccard loss (Yu & Blaschko, 2020; Berman et al., 2018), we here study the submodularity of $\mathcal{L}_{\text{pix}}$.

Proposition 4. *Given a groundtruth, $\mathcal{L}_{\text{pix}}(\cdot, A)$ is submodular w.r.t. the set of mispredictions of A to the groundtruth.*

Proof. By definition (Lovász, 1983), a function $\mathcal{L}$ is submodular if and only if $\forall B \subseteq A \subset V$, and $\forall x \in V \backslash A$:

$$\mathcal{L}\left(B \cup \{x\}\right) - \mathcal{L}(B) \geq \mathcal{L}\left(A \cup \{x\}\right) - \mathcal{L}(A). \tag{12}$$

In our case, given a groundtruth and two sets of predicted pixels $\forall B \subseteq A \in V$, it holds that:

$$\mathbf{n}_B \subseteq \mathbf{n}_A \subseteq \mathbf{m} \implies \langle \mathbf{d}_{\mathbf{n}_B}, \mathbf{1}_{\mathbf{n}_B} \rangle \leq \langle \mathbf{d}_{\mathbf{n}_A}, \mathbf{1}_{\mathbf{n}_A} \rangle \leq |\mathbf{m}| \tag{13a}$$

$$\mathbf{p}_B \subseteq \mathbf{p}_A \subseteq \mathbf{m} \implies \langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle \leq \langle \mathbf{d}_{\mathbf{p}_A}, \mathbf{1}_{\mathbf{p}_A} \rangle \leq |\mathbf{m}| \tag{13b}$$

We have $\mathcal{L}_{\text{pix}} = 1 - \text{PixIoU} = \mathcal{L}_{\text{iou}} + 1 - \frac{|\mathbf{m}| - \langle \mathbf{d}_\mathbf{n}, \mathbf{1}_\mathbf{n} \rangle}{|\mathbf{m}| + \langle \mathbf{d}_\mathbf{p}, \mathbf{1}_\mathbf{p} \rangle}$. It has been already demonstrated that $\mathcal{L}_{\text{iou}}$ is submodular w.r.t. the set of mispredictions (Yu & Blaschko, 2020). As the sum of two submodular functions is submodular (Bach, 2013; Fujishige, 1991), we now prove that $\mathcal{L}'_{\text{pix}} = 1 - \frac{|\mathbf{m}| - \langle \mathbf{d}_\mathbf{n}, \mathbf{1}_\mathbf{n} \rangle}{|\mathbf{m}| + \langle \mathbf{d}_\mathbf{p}, \mathbf{1}_\mathbf{p} \rangle} = \frac{\langle \mathbf{d}_\mathbf{p}, \mathbf{1}_\mathbf{p} \rangle + \langle \mathbf{d}_\mathbf{n}, \mathbf{1}_\mathbf{n} \rangle}{|\mathbf{m}| + \langle \mathbf{d}_\mathbf{p}, \mathbf{1}_\mathbf{p} \rangle}$ is also submodular.

We first prove that $\mathcal{L}'_{\text{pix}}$ is submodular w.r.t. false negatives. For $\forall x \in V \backslash A$ and x is a false negative,

$$\begin{aligned}
&\mathcal{L}\left(B \cup \{x\}\right) - \mathcal{L}(B) \\
&= \frac{\langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle + \langle \mathbf{d}_{\mathbf{n}_B}, \mathbf{1}_{\mathbf{n}_B} \rangle + \mathbf{d}_x x}{|\mathbf{m}| + \langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle} - \frac{\langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle + \langle \mathbf{d}_{\mathbf{n}_B}, \mathbf{1}_{\mathbf{n}_B} \rangle}{|\mathbf{m}| + \langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle} \\
&= \frac{\mathbf{d}_x x}{|\mathbf{m}| + \langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle} \geq \frac{\mathbf{d}_x x}{|\mathbf{m}| + \langle \mathbf{d}_{\mathbf{p}_A}, \mathbf{1}_{\mathbf{p}_A} \rangle} \\
&= \mathcal{L}\left(A \cup \{x\}\right) - \mathcal{L}(A).
\end{aligned} \tag{14}$$

The inequality holds owe to Equation (13) and $\mathbf{d}_x \in [0, 1]$.

We then prove that $\mathcal{L}'_{\text{pix}}$ is submodular w.r.t. false positives. For $\forall x \in V \backslash A$ and x is a false positive,

$$\begin{aligned}
&\mathcal{L}\left(B \cup \{x\}\right) - \mathcal{L}(B) \\
&= \frac{\langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle + \langle \mathbf{d}_{\mathbf{n}_B}, \mathbf{1}_{\mathbf{n}_B} \rangle + \mathbf{d}_x x}{|\mathbf{m}| + \langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle + \mathbf{d}_x x} - \frac{\langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle + \langle \mathbf{d}_{\mathbf{n}_B}, \mathbf{1}_{\mathbf{n}_B} \rangle}{|\mathbf{m}| + \langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle} \\
&= \frac{\mathbf{d}_x x \left(|\mathbf{m}| - \langle \mathbf{d}_{\mathbf{n}_B}, \mathbf{1}_{\mathbf{n}_B} \rangle\right)}{\left(|\mathbf{m}| + \langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle + \mathbf{d}_x x\right)\left(|\mathbf{m}| + \langle \mathbf{d}_{\mathbf{p}_B}, \mathbf{1}_{\mathbf{p}_B} \rangle\right)} \\
&\geq \frac{\mathbf{d}_x x \left(|\mathbf{m}| - \langle \mathbf{d}_{\mathbf{n}_A}, \mathbf{1}_{\mathbf{n}_A} \rangle\right)}{\left(|\mathbf{m}| + \langle \mathbf{d}_{\mathbf{p}_A}, \mathbf{1}_{\mathbf{p}_A} \rangle + \mathbf{d}_x x\right)\left(|\mathbf{m}| + \langle \mathbf{d}_{\mathbf{p}_A}, \mathbf{1}_{\mathbf{p}_A} \rangle\right)} \\
&= \mathcal{L}\left(A \cup \{x\}\right) - \mathcal{L}(A).
\end{aligned} \tag{15}$$

The inequality holds again owe to Equation (13) and $\mathbf{d}_x \in [0, 1]$. □

Algorithm 2 Gradient computation of the Lovász PixIoU

Inputs: vector of errors $\mathbf{s}(c) \in \mathbb{R}_+^N$, for a class label c, distance vectors $\mathbf{d_n}$ and $\mathbf{d_p}$ as calculated in Algorithm 1, class of foreground pixels $\mathbf{m} = \{\mathbf{y} = c\}$

Output: the gradients $\mathbf{g}$

```
$\pi \leftarrow$ permutation s.t. $\mathbf{s}$ in decreasing order
$\mathbf{m}_\pi \leftarrow (m_{\pi_i})_{i\in[1,N]}$; $\mathbf{d}_{n_\pi} \leftarrow (n_{\pi_i})_{i\in[i,N]}$; $\mathbf{d}_{p_\pi} \leftarrow (p_{\pi_i})_{i\in[i,N]}$
$\mathbf{n} \leftarrow \textbf{cumsum}(\mathbf{m}_\pi)$; $\mathbf{p} \leftarrow \textbf{cumsum}(1 - \mathbf{m}_\pi)$
$\textbf{iou} \leftarrow \frac{\textbf{sum}(\mathbf{m}_\pi) - \mathbf{n}}{\textbf{sum}(\mathbf{m}) + \mathbf{p}}$
$\textbf{piou} \leftarrow \frac{\textbf{sum}(\mathbf{m}_\pi) - \textbf{cumsum}(\mathbf{d}_{n_\pi} \odot \mathbf{m}_\pi)}{\textbf{sum}(\mathbf{m}) + \textbf{cumsum}(\mathbf{d}_{p_\pi} \odot (1 - \mathbf{m}_\pi))}$
$\mathbf{g} \leftarrow 1 - \textbf{iou} + 1 - \textbf{piou}$
if $N > 1$ then
    $\mathbf{g}[2:N] \leftarrow \mathbf{g}[2:N] - \mathbf{g}[1:N-1]$
end if
return $\mathbf{g}$
```

As discussed in prior works, the convex closure of submodular set functions is tight and computable in polynomial time (Lovász, 1983), it corresponds to its Lovász extension (Bach, 2013):

$$\begin{aligned}
&\bar{\mathcal{L}} : \mathbf{s} \in \mathbb{R}^p \to \sum_{i=1}^{p} s_i g_i(\mathbf{s}), \\
&\text{with } g_i(\mathbf{s}) = \mathcal{L}\left(\{\pi_1, \ldots, \pi_i\}\right) - \mathcal{L}\left(\{\pi_1, \ldots, \pi_{i-1}\}\right),
\end{aligned} \tag{16}$$

where $\pi = \{\pi_1, \ldots, \pi_p\}$ is a permutation s.t. $\mathbf{s}$ is in decreasing order $s_{\pi_1} \geq s_{\pi_2} \geq \ldots \geq s_{\pi_p}$. The vector $\mathbf{g}(\mathbf{s})$ directly corresponds to the derivative of $\bar{\mathcal{L}}$ w.r.t. $\mathbf{s}$ (Lovász, 1983). The Lovász surrogates apply $\mathbf{s}$ as the vector of all pixel errors. Specifically, the Lovász Softmax (Berman et al., 2018) is proposed based on the logistic output using a softmax unit. We refer to (Berman et al., 2018) regarding the details of Lovász surrogates for submodular functions.

On the other hand, Lovász Softmax applies $\mathcal{L}$ in Equation (16) only as the original Jaccard loss $\mathcal{L}_{\text{iou}}$ namely on IoU (c.f. Algorithm 1 in (Berman et al., 2018)):

$$\mathcal{L}_{\text{iou}}\left(\{\pi_1, \ldots, \pi_i\}\right) = 1 - \frac{|\mathbf{m}| - |\{\pi_{n_1}, \ldots, \pi_{n_i}\}|}{|\mathbf{m}| + |\{\pi_{p_1}, \ldots, \pi_{p_i}\}|}, \tag{17}$$

In this paper, we propose to apply $\mathcal{L}$ in Equation (16) following the definition of $\mathcal{L}_{\text{pix}}$, namely we have:

$$\begin{aligned}
\mathcal{L}_{\text{pix}}\left(\{\pi_1, \ldots, \pi_i\}\right) = 1 &- \frac{|\mathbf{m}| - \langle \mathbf{d_n}, \mathbf{1}_{\{\pi_{n_1}, \ldots, \pi_{n_i}\}} \rangle}{|\mathbf{m}| + \langle \mathbf{d_p}, \mathbf{1}_{\{\pi_{p_1}, \ldots, \pi_{p_i}\}} \rangle} \\
&- \frac{|\mathbf{m}| - |\{\pi_{n_1}, \ldots, \pi_{n_i}\}|}{|\mathbf{m}| + |\{\pi_{p_1}, \ldots, \pi_{p_i}\}|} + 1.
\end{aligned} \tag{18}$$

Compare to the Lovász Softmax in (Berman et al., 2018), the Lovász surrogate applying to $\mathcal{L}_{\text{pix}}$ only additionally involves

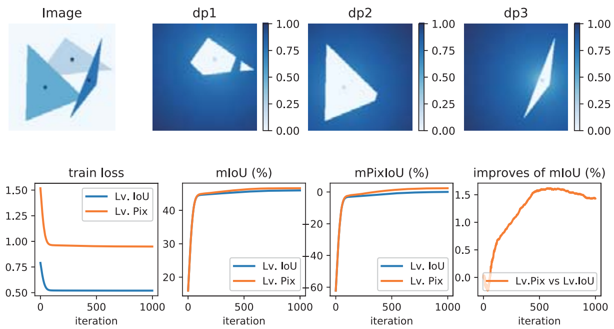


Figure 4: Synthetic experiment. Different classes are shown in different color with the center in the darkest blue and the distance maps for false positives in each case. Training Lovász PixIoU can achieve better PixIoU and IoU scores.

Train loss	CE	Lv. IoU	Lv. Pix
mIoU(%)	71.6	74.2	**75.6**

Table 1: Performance evaluated by mIoU(%) on VOT-ST2020 at test time with different train losses.

applying a dot product between the set of false negatives, the false positives and the distance vectors corresponding to each of them, respectively. The computation complexity for the gradient of the loss surface thus stays the same which is $\mathcal{O}(N \log N)$. The gradient computation of the Lovász PixIoU is summarized in Algorithm 2.

4. Experimental Results

In this section, we show experimental results on several pixelwise prediction tasks. Denote Lovász IoU (Lv. IoU) for the existing surrogate on optimizing IoU i.e. the Lovász Softmax, and Lovász PixIoU (Lv. Pix) for the method we proposed in this paper.

4.1. Synthetic experiments

We first carry out a synthetic experiment. We generate 50 images of size $200 * 200$ which contain 3 foreground classes in each image, labelled from 1 to 3 in arbitrary triangle-like shapes. Objects can overlap each other which results in non-convex shapes that simulates the real-world scenario (c.f. Figure 4). We train a simple a linear model of a $3 * 3$ convolution layer with features that are synthetically generated using different Gaussian perturbations.

We test the models trained by cross-entropy, Lovász IoU and Lovász PixIoU during training. From the results in Figure 4: (i) training with Lovász PixIoU achieves reasonable convergence rate; (ii) training with Lovász PixIoU achieves better mean PixIoU scores, which validates the effectiveness of the Lovász PixIoU for learning $\mathcal{L}_{\text{pix}}$; (iii) training Lovász PixIoU achieves better mean IoU scores than training with Lovász IoU, which validates the benefits of PixIoU to IoU as discussed in the previous sections.

Train loss	CE	Lv. IoU	Lv. Pix
mIoU(%)	71.6	74.2	**75.6**

Table 2: Performance evaluated by mIoU(%) on VOT-ST2020 at test time with different train losses.

	CE	Lv. IoU	Lv. Pix
Resnet-50	80.1	80.5	**80.8**
Resnet-101	82.9	82.8	**83.1**

Table 3: Performance of Deeplabv3 models evaluated by mIoU(%) on Pascal VOC *val* set with different backbones.

4.2. Pixelwise Object Tracking on VOT2020

For real-world datasets, we first experiment on the VOT2020 [1], a pixelwise object tracking benchmark. This dataset consists not only the bounding box annotations of the target, but also segmentation mask on each frame. Therefore, it forms a binary segmentation problem (target vs. background) for each frame.

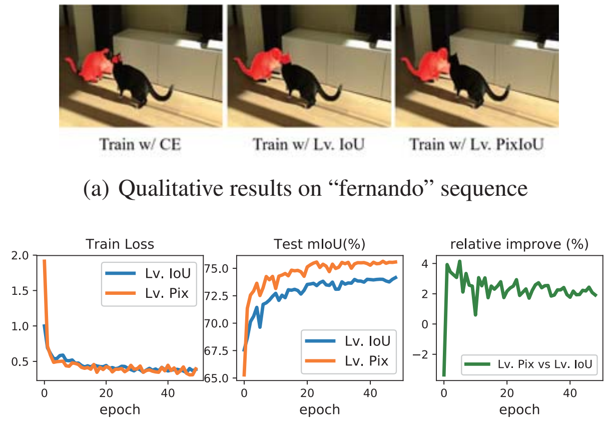
We compare with the method AFOD (Chen et al., 2020) where the Lovász IoU is used for training, and we swap it by the Lovász PixIoU. Models are trained on the Youtube-VOS-18 (Xu et al., 2018) and DAVIS-16 (Perazzi et al., 2016) dataset, with the backbone Resnet-50 pretrained on ImageNet (Krizhevsky et al., 2017). For evaluation, we fix the online updating module and only evaluate the segmentation performance using mIoU on all testing frames.

Shown in Figure 5, with comparable training convergence, training with Lovász PixIoU achieves better IoU scores at test time than training with the Lovász IoU within the same training epoch. Quantitative results are shown in Table 2. Training with cross-entropy only provide a suboptimal approximation to optimize IoU and the performance degrades. On the contrast, training with Lovász PixIoU provides a significant gain on the IoU score at test time.

4.3. Semantic segmentation on Pascal VOC

We then perform a semantic segmentation task on Pascal VOC 2012. We use the Deeplabv3 models (Chen et al., 2017) implemented by PyTorch framework with Resnet-50 and Resnet-101 (He et al., 2016) as backbones. Models are pre-trained on COCO (Lin et al., 2014), then trained by using the cross-entropy loss, the Lovász IoU and Lovász PixIoU. SGD is used for the optimization with a polynomial learning rate policy

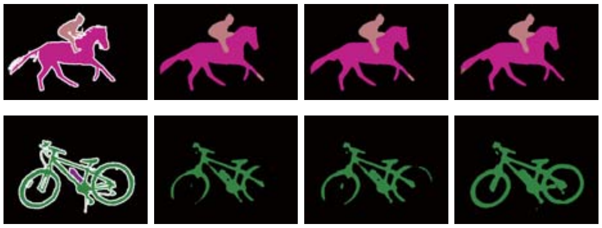
[1] `https://votchallenge.net/vot2020/`



(a) Qualitative results on “fernando” sequence

(b) Train/Test performance with different losses for training

Figure 5: Performance on VOT-2020. Training with Lovász PixIoU provides a comparable convergence rate as well as a significant gain on the IoU score at test time.



(a) GT (b) CE (c) Lv. IoU (d) Lv. Pix

Figure 6: Qualitative results on Pascal VOC 2012 of the model Deeplabv3-Resnet101.

$2.5 * 10^{-4}\,(1 - \text{iter/max\_iter})^{0.9}$, with momuntum 0.9 and weight decay $1 * 10^{-4}$. We train 50 epochs on 2 GPUs with a batch size of 16. Worth to mention, the results are reported *without* using particular data strategy such as the *equibatch* in previous work (Berman et al., 2018) for Lovász IoU and Lovász PixIoU.

As shown in Table 3 and Figure 6, training with Lovász PixIoU achieves consistently better performance at test time (see more in supplementary materials). In Table 4 we show the per-class mIoU for each category.

4.4. Semantic segmentation on Cityscapes

Next, we train and test on the Cityscapes (Cordts et al., 2016), a large-scale dataset contains 30 classes, 8 categories in total, with 5k fine annotations and 2k coarse annotations on urban street scenes. We now perform a semantic segmentation task using the DeeplabV3+ model (Chen et al., 2018) implemented the Detectron2 system (Wu et al., 2019). To compare with its released model [2], we use a modified version of the ResNet-101 pretrained on ImageNet. We compare the performance by training on Cityscapes trainset with the original loss (a weighted cross-entropy, in which the hard pixels are kicked off and the loss is only mining on the top K percent pixels, denoted as WCE), the Lovász IoU and the Lovász PixIoU.

	mIoU	iIoU	IoU_sup	iIoU_sup	mPixIoU
released	79.98	62.04	90.09	79.65	63.76
WCE	79.98	62.31	90.90	79.68	63.91
Lv. IoU	79.80	62.54	90.87	79.85	63.53
Lv. Pix	**80.62**	**63.19**	**91.04**	**80.42**	**65.00**

Table 5: Performance of Deeplabv3+ models evaluated on Cityscapes *val* set, numbers in percentage (%).

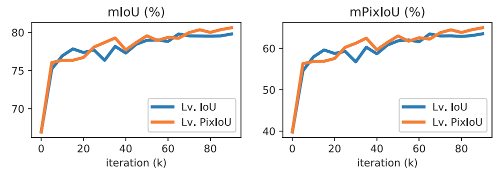


Figure 7: Performance of evaluating Deeplabv3+ models on *val* set during training.

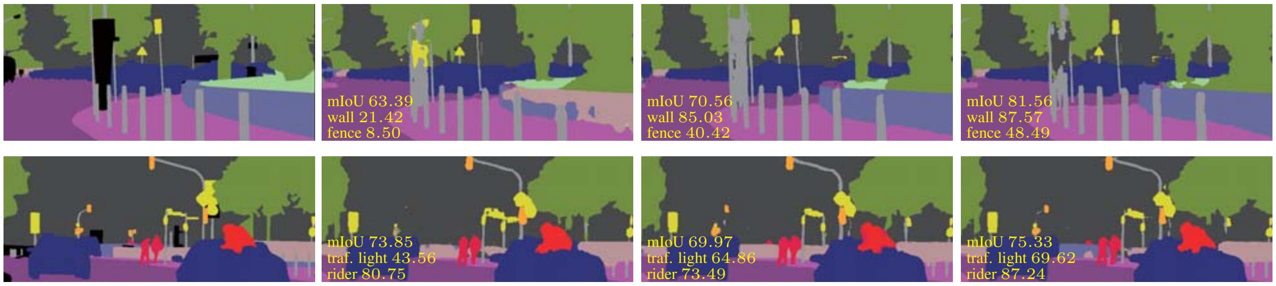
For training with the IoU-related losses, in prior works (Berman et al., 2018), Lovász IoU is mostly used for fine-tuning, which involves additional training iterations in total (90k+20k). Otherwise, training Lovász IoU from scratch only provides suboptimal performance (76.64 mIoU in our experiments). In this work, we here propose a pseudo-pretrain strategy. We first train a model with the WCE for only 5k iterations, then we train as normal for 90k iterations using different loss functions. This strategy provides a balance between the number of iterations (95k vs. 90k+20k) and the performance (79.8 mIoU vs. 76.64 mIoU). All training procedures are carried on with batchsize 16 on 8 P40, with a polynomial learning rate policy $0.008 * (1 - \text{iter}/\text{maxiter})^{0.9}$ and a warming-up by 1k iterations for the 90k-iteration training.

Shown in Table 5 are the experimental results. The released model (denoted “released”) is trained from scratch by WCE for just 90k iterations. Other rows (WCE, Lv. IoU, Lv. Pix) are all carried out with the pseudo-pretrain strategy. We can see that applying the pseudo-pretrain strategy using WCE, which equivalents to train for 95k iterations, does not significantly improve the performance. However, using Lovász PixIoU for training improves the performance for 0.6 mIoU and 1.15 iIoU scores, as well as achieves the best PixIoU score. It provides a consistent improvements over all IoU-related evaluation. Especially, in Table 6 shows the per-class performance, that training with Lovász PixIoU significantly improves the performance on the classes that contain more meaningful values on distance or shapes, such as *wall*, *traffic light/sign*, *person*, *rider*, etc, which empir-

[2] https://github.com/facebookresearch/detectron2/tree/master/projects/DeepLab

Res50	all	airpl.	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	pers.	plant	sheep	sofa	train	tv
CE	80.1	93.5	58.3	89.4	72.0	77.7	94.9	85.0	90.9	47.8	88.1	68.1	86.8	85.1	87.0	90.2	66.4	90.1	55.5	87.7	72.4
Lv. IoU	80.5	94.0	57.0	88.9	72.2	77.5	95.3	86.6	91.4	46.3	88.7	68.8	87.2	86.6	88.8	90.2	66.2	91.7	56.2	87.7	72.8
Lv. Pix	80.8	94.1	62.3	90.0	74.5	76.0	94.7	87.9	90.7	49.7	87.5	65.2	87.9	86.1	87.2	90.1	64.8	90.2	59.7	87.3	75.5
Res101	all	airpl.	cycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	pers.	plant	sheep	sofa	train	tv
CE	82.9	93.3	61.0	90.6	77.5	80.3	93.9	87.4	94.4	49.7	93.5	68.7	91.7	93.7	91.2	91.5	67.3	90.2	59.0	89.3	80.7
Lv. IoU	82.8	93.1	62.4	90.0	77.1	80.7	93.6	88.7	93.9	50.6	92.5	70.4	90.9	92.4	91.5	91.7	69.7	90.0	59.5	89.5	75.2
Lv. Pix	83.1	92.8	71.1	90.1	78.9	78.6	93.3	88.5	90.9	50.3	93.5	71.1	87.2	93.3	90.3	91.3	67.5	90.0	58.9	90.1	80.9

Table 4: Per-class mIoU (%) on Pascal VOC 2012 *val* set by training with different losses



(a) groundtruth (b) WCE (c) Lv.IoU (d) Lv.PixIoU

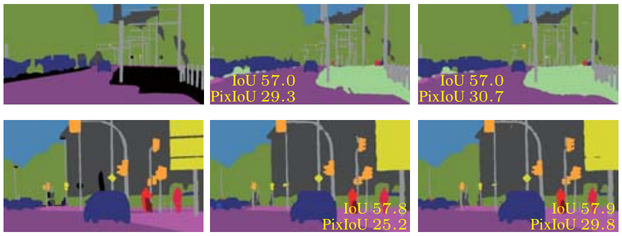
Figure 8: Qualitative results on Cityscapes of the model Deeplabv3+ trained with different loss functions.

ically validates that PixIoU constraints more on the shape and the distance of the set of the pixels.

Another widely used related function is the Dice loss (Milletari et al., 2016). The comparison between Dice loss and Lovász IoU has been carried on in prior works (Li et al., 2021), shows that optimizing Dice loss provides superior performance on Cityscapes (79.30 vs. 77.67) while inferior on Pascal VOC (77.78 vs. 79.72).

Some qualitative results are shown in Figure 8 and 9 (see more in supplementary). Particularly, in Figure 9 shown some examples, for which by using Lovász PixIoU for learning, we achieve only similar IoU scores but better PixIoU scores. However, the one with larger PixIoU score provides better visualization and steeper gradients for learning, which validates the effectiveness of optimizing on PixIoU.

While we look into the worst labels in Table 6 and observe qualitatively that some false negatives are produced by semantically and geographically nearing labels, such as predicting "train" pixels as "bus", "sidewalk" as "road". We speculate that PixIoU does not emphasis for intertwining shapes. While the overall performance is increased, it is still challenging for specific labels.



(a) groundtruth (b) Lv. IoU (c) Lv. PixIoU

Figure 9: Examples of the cases that PixIoU provides larger gradients than those of IoU, and the predictions with larger PixIoU provides better qualitative results.

4.5. Panoptic segmentation on Cityscapes

Last, on Cityscapes dataset, we additionally perform a panoptic segmentation task. This task aims at labeling all pixels in the scene, as well as distinguishing different instance for certain classes. Multiple metrics are involved for evaluating the panoptic task, while the IoU-related evaluations are mostly used for the semantic subtask. We train the state-of-the-art model Panoptic-DeepLab (Cheng et al., 2020) with backbone Resnet-50 in order to compare with the released model [3]. The original loss for the semantic head is a weighted bootstrapped cross-entropy loss which weights different pixels differently depending on the average size of the class instance. For the pixels in the same class, the weight is same. This differs with the PixIoU that different pixels in the same instance could be evaluated differently. Training is performed with batchsize 16 on 8 P40 for 90k iterations with a cosine learning rate policy and base learning rate 0.0008. Experimental results can be found in Table 7. We again observe an improvements on the IoU-related evaluation, with comparable performance

[3] https://github.com/facebookresearch/detectron2/tree/master/projects/Panoptic-DeepLab

class	WCE	Lv. IoU	Lv. Pix
road	98.34	98.22 (↓ 0.12)	98.14 (↓ 0.21)
sidewalk	86.36	85.80 (↓ 0.64)	85.37 (↓ 1.14)
building	93.21	93.05 (↓ 0.17)	93.26 (↑ 0.05)
wall	57.05	55.54 (↓ 2.65)	62.88 (↑ 10.21)
fence	64.01	63.94 (↓ 0.12)	63.51 (↓ 0.78)
pole	69.12	69.18 (↑ 0.09)	69.64 (↑ 0.76)
traf. light	74.00	75.35 (↑ 1.83)	76.15 (↑ 2.90)
traf. sign	81.54	82.36 (↑ 1.00)	83.18 (↑ 2.01)
vegetation	92.87	92.77 (↓ 0.11)	92.85 (↓ 0.03)
terrain	64.61	65.37 (↑ 1.17)	65.46 (↑ 1.32)
sky	95.40	95.36 (↓ 0.04)	95.31 (↓ 0.09)
person	84.09	84.45 (↑ 0.42)	84.60 (↑ 0.60)
rider	65.43	68.21 (↑ 4.26)	68.87 (↑ 5.27)
car	95.65	95.55 (↓ 0.10)	95.60 (↓ 0.05)
truck	79.63	75.32 (↓ 5.41)	78.65 (↓ 1.24)
bus	89.10	89.53 (↑ 0.48)	89.48 (↑ 0.43)
train	78.78	78.46 (↓ 0.40)	77.84 (↓ 1.19)
motorcycle	70.81	67.54 (↓ 4.62)	70.32 (↓ 0.70)
bicycle	79.63	80.15 (↑ 0.65)	80.68 (↑ 1.31)
all	79.98	79.80 (↓ 0.23)	80.62 (↑ 0.80)

Table 6: Per-class performance by mIoU (%) on Cityscapes *val* set. In parentheses shows the relative improvements (in percentage) compared to the performance with WCE.

	mIoU	iIoU	PQ	SQ	RQ	AP	AP50
WCE	78.72	62.83	60.28	81.02	**73.18**	**32.05**	54.53
Lv. IoU	78.45	63.18	59.47	80.71	72.57	30.12	53.87
Lv. Pix	**79.01**	**64.22**	60.26	**81.36**	72.93	31.88	**55.55**

Table 7: Performance of Panoptic-Deeplab models evaluated on Cityscapes *val* set, numbers in percentage (%).

by other metrics.

5. Conclusion

In this work, we propose a novel evaluation method named PixIoU for dense pixelwise prediction. Compared to the original IoU, PixIoU evaluates differently on the false negatives and false positives. By our definition, PixIoU provides steeper gradients for learning. We demonstrate in theory the feasibility and applicability using the Lovász surrogate for learning PixIoU, and breakthrough empirically the performance plateau by learning for the original IoU during training for various tasks. To the best of our knowledge, this is the first work that generalizes IoU computation to provides better interpretation and evaluation for pixelwise prediction, and provide feasible and efficient learning strategy that shows improvement over optimizing the original IoU. We expect this work to be an important step on reconsidering the evaluation methods in various benchmarks, and arouse, again, the attention of an objective-based learning in various computer vision tasks in general.

Acknowledgements

The authors would like to thank Pingjun Li, Hui Zhang, Yi Zhou, Chao Zhang for valuable comments and discussions.

References

Bach, F. *Learning with Submodular Functions: A Convex Optimization Perspective*. 2013.

Berman, M., Rannen Triki, A., and Blaschko, M. B. The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4413–4421, 2018.

Bertels, J., Eelbode, T., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., and Blaschko, M. B. Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice. *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 92–100, 2019.

Bhat, G., Lawin, F. J., Danelljan, M., Robinson, A., Felsberg, M., Gool, L. V., and Timofte, R. Learning what to learn for video object segmentation. *arXiv preprint arXiv:2003.11540*, 2020.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

Chen, Y., Xu, J., Yu, J., Wang, Q., Yoo, B., and Han, J.-J. AFOD: Adaptive focused discriminative segmentation tracker. In *The Eight Visual Object Tracking Challenge Workshop in ECCV*, 2020.

Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., and Chen, L.-C. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Fujishige, S. *Submodular functions and optimization*. 1991.

Gillenwater, J. A., Iyer, R. K., Lusch, B., Kidambi, R., and Bilmes, J. A. Submodular hamming metrics. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, pp. 3141–3149. 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of The ACM*, 60(6):84–90, 2017.

Li, H., Tao, C., Zhu, X., Wang, X., Huang, G., and Dai, J. Auto seg-loss: Searching metric surrogates for semantic segmentation. In *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.

Li, W. Estimating jaccard index with missing observations: A matrix calibration approach. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, pp. 2620–2628. 2015.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollr, P. Microsoft coco: Common objects in context. *arXiv preprint arXiv:1405.0312*, 2014.

Lovász, L. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pp. 235–257. Springer, 1983.

Milletari, F., Navab, N., and Ahmadi, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, 2016.

Neven, D., Brabandere, B. D., Proesmans, M., and Gool, L. V. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR))*, pp. 8837–8845, 2019.

Nowozin, S. Optimal decisions from probabilistic models: The intersection-over-union case. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 548–555, 2014.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Gool, L. V., Gross, M., and Sorkine-Hornung, A. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 724–732, 2016.

Rahman, A. and Wang, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International Symposium on Visual Computing*, pp. 234–244, 2016.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., and Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666, June 2019.

Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer, 1995.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., and Huang, T. S. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

Ye, H.-J., Zhan, D.-C., Si, X.-M., Jiang, Y., and Zhou, Z.-H. What makes objects similar: A unified multi-metric learning approach. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing System*, pp. 1235–1243. 2016.

Ying, H., Huang, Z., Liu, S., Shao, T., and Zhou, K. Embedmask: Embedding coupling for one-stage instance segmentation. *arXiv preprint arXiv:1912.01954*, 2019.

Yu, J. and Blaschko, M. B. Learning submodular losses with the Lovász hinge. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Journal of Machine Learning Research: W&CP*, pp. 1623–1631, 2015.

Yu, J. and Blaschko, M. B. The Lovász hinge: A novel convex surrogate for submodular losses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42: 735–748, 2020.

Yu, J., Jiang, Y., Wang, Z., Cao, Z., and Huang, T. Unitbox: An advanced object detection network. In *Proceedings of 24th ACM international conference on Multimedia*, pp. 516–520, 2016.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D. Distance-IoU Loss: Faster and better learning for bounding box regression. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pp. 12993–13000, 2020.