A GAN framework for Instance Segmentation using the Mutex Watershed Algorithm

Mandikal Vikram, Steffen Wolf
15it217.vikram@nitk.edu.in, steffen.wolf@iwr.hedielberg.de

Abstract

The Mutex Watershed algorithm is a fast graph clustering algorithm of linearithmic complexity for graphs with both positive and negative weight edges[1]. The objective is to train a neural network to produce these weights when presented with an image. We propose a GAN loss function along with the supervised loss for this task since the objective is to have structural similarity with the ground truth. We show that this method and additional auxiliary task losses improve the quality of the Adjusted Rand Score over the score reported in [1]. We present an ablation study to show that when the images in the target domain are constrained to be discrete, adding an auxiliary task with smooth target images significantly improves the training performance and stability. We also use the discriminator to train on unlabeled images which further improves our results.

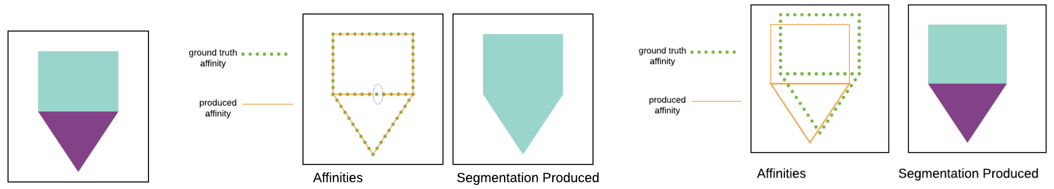
1 Introduction

The Mutex Watershed is a greedy graph partitioning algorithm which when presented with the attractive and repulsive affinities between pixels, efficiently finds an image segmentation. In its current formulation it is not differentiable, and hence a learnable loss function cannot be defined for it. Using a point-to-point loss function between the output of the neural network and the ground truth affinity maps is a fair approximation but has its shortcomings which are illustrated in Figure 1. Figure 1b shows a case when the network output has a high point-to-point similarity with the ground truth, but a small hole in the boundary merges the two instances. We ideally want to this result to be highly penalized, but the point-to-point loss levies only a minimal penalty in this case. Figure 1c shows the case when the network output is slightly translated from the ground truth. This would result in a reasonable quality segmentation, and we expect the loss function to penalize this minimally, but the point-to-point loss function imposes a massive loss now. These shortcomings of the supervised point-to-point loss indicate that the quality of the affinities from the neural network can be improved by improving the loss function, which motivates us to supplement this supervised loss with a GAN loss.

2 Proposed Approach

We model this as an image translation problem from the image space to the affinity map space and use a conditional GAN [2] based approach for this. The generator is trained to learn a deterministic transition from a given image to its corresponding ground truth affinities. It is trained to learn a mapping from the observed image x to its corresponding affinity map y, $G : x \rightarrow y$. We do not include any random noise in the input to the generator as we intend to learn a deterministic transition, and also the networks tend to ignore the random noise vector in the input in practice [2]. The discriminator seeks to distinguish between the ground truth affinities and the “fake” affinities produced by the generator.

Extended abstract. Smooth Games Optimization and Machine Learning Workshop (NIPS 2018), Montréal, Canada.



(a) A synthetic image consisting two instances

(b) A small hole in an otherwise pixel-perfect affinity leads to the merging of the two instances

(c) A small translation w.r.t. to the groundtruth affinities, leads to a reasonable segmentation but attracts a massive loss

Figure 1: Shortcomings of supervised point-to-point loss functions

The objective of the conditional GAN is the loss mentioned in [2], however we drop the z (noise vector) which does not affect the training as reported in [2] itself.

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) =& \mathbb{E}_{x,y}[\log D(x, y)]+ \\ & \mathbb{E}_{x}[\log(1 - D(x, G(x))]\end{aligned} \tag{1}$$

The generator G seeks to minimize the above loss while the adversary D seeks to maximize it. The generator also minimizes an additional dice loss[3] $\mathcal{L}_{dice}(G)$ between the generated affinities and the ground truth affinities. Thus the overall objective is the following -

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{dice}(G) \tag{2}$$

The architecture of the generator is the U-Net [4] which takes in an image as the input and produces a n channel output, where n is the number of affinity maps we intend to produce. The discriminator architecture is the patchgan proposed in [2]. The discriminator can capture patterns well beyond the simple point-to-point similarity which the dice loss enforces. One such pattern is that the ground truth affinities are always closed contours and do not have any holes in them, the discriminator could use this to differentiate between the "real" and "fake" affinity maps, this would then force the generator to close the holes and thus improve the quality of the affinity maps.

2.1 Adding an auxiliary task to stabilize and improve the GAN training

We find that the above training is unstable and requires extensive tuning of the parameters λ in equation 2 and the learning rate to obtain the best affinities. We attempted to use the Wasserstein GAN [5] to stabilize our training, although the training was stabilized, it results in a significant drop in the quality of the affinities when compared to that of the conditional GAN. As noted earlier, the ground truth affinities are binary; we suppose that this is a hard target for the generator which leads to the instability of the training. Hence, we add an auxiliary task of producing smoother targets - we train the generator to produce the distance transform of the affinities along with the affinities. The distance transform of the affinities has continuous values and results in a smooth transition from the values of the pixels at the boundary (0) to the values away from the boundaries (1). Also, the discriminator is made to additionally differentiate between the distance transform produced by the generator and the distance transform of the ground truth affinities.

Hence, the generator is trained to produce a mapping G is $x \rightarrow \{y, \tilde{y}\}$ where y is the affinity map and $\tilde{y}$ is the distance transform. For the following definitions we will decompose the generator into the affinity generator $G_a(x)$ and the distance transform generator $G_d(x)$ and apply a cGAN loss to both

$$\tilde{\mathcal{L}}_{cGAN}(G, D) = \mathcal{L}_{cGAN}(G_a, D_a) + \mathcal{L}_{cGAN}(G_d, D_d) \tag{3}$$

For the distance transform regression we utilize the smooth L1 loss (Huber loss) $\mathcal{H}$ [6] between the distance transform affinities which it produces and the distance transform of the ground truth affinities along with dice loss $\mathcal{J}$.

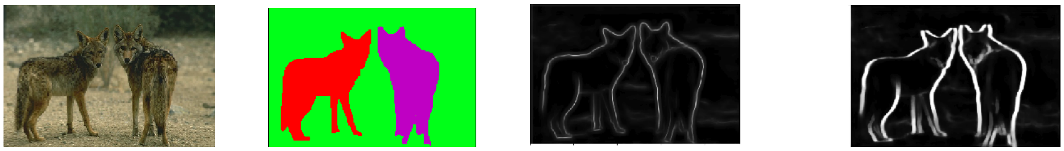
$$\tilde{\mathcal{L}}_{dice}(G) = \mathbb{E}_{x,y,z}\left[\mathcal{J}(y, G_a(x))\right] \qquad \tilde{\mathcal{L}}_{huber}(G) = \mathbb{E}_{x,\tilde{y},z}\left[\mathcal{H}(\tilde{y}, G_d(x))\right] \tag{4}$$

The final objective is

$$G^* = \arg \min_G \max_D \tilde{\mathcal{L}}_{cGAN}(G, D) + \lambda_1 \tilde{\mathcal{L}}_{dice}(G) + \lambda_2 \tilde{\mathcal{L}}_{huber}(G) \tag{5}$$

We empirically find that this not only stabilizes the training but also leads to a significant improvement in the quality of the affinities produced. We propose that this is due to the discriminator being able to capture the structural differences better when provided with smoother images and not due to the additional supervised loss (Huber loss on the distance transform) on the generator. We verify this through an ablation study presented in the next section.

3 Results and Experiments



(a) Image

(b) Segmentation produced by our final formulation

(c) Affinities produced for the vector (-1,0)

(d) Affinities produced for the vector (0,-9)

Figure 2: A sample segmentation and affinties produced on a BSD500 test image by our final formulation

We consider the BSD500 datatset[7] for our experiments. The formulation presented in equation 5 obtains an Adjusted Rand Index of 0.832 against the 0.826 reported in [1] (both use the Mutex Watershed algorithm). The fact that [1] used the edge detection output of an additional edge detector network and the image as an input to the network while we just give use the image as the input to our network, highlights the effectiveness and simplicity of our proposed approach. We further extend the GAN training on unlabelled natural images - after the losses reasonably saturate on the labelled BSD train data, we present external unlabelled images to the generator at periodic iterations along with labelled data. On the unlabelled data, we seek to minimize $\mathbb{E}_x[\log(1 - D(x, G_a(x))] + \mathbb{E}_x[\log(1 - D(x, G_d(x))]$. Hence, on the labelled data, both the discriminator and generator train, and on the unlabelled images the discriminator weights are frozen and only the generator updates by reducing the loss from the discriminator. Training on external unlabelled images and model averaging significantly improves the rand index to **0.845**. Figure 2 shows a sample segmentation and some of the affinity maps obtained on a BSD500 test image.

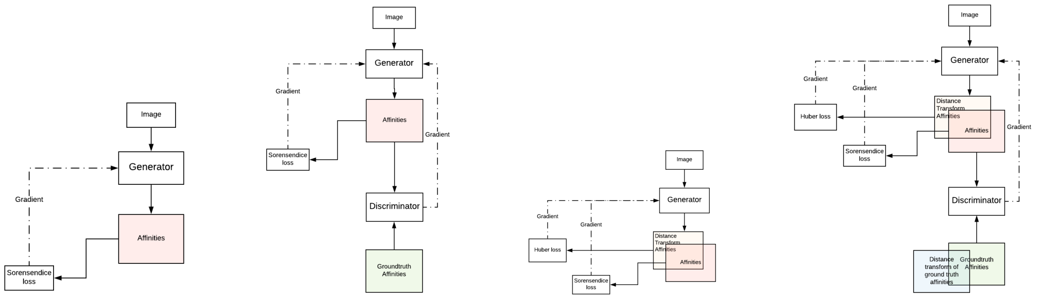
3.1 Ablation Study

We perform ablation experiments to provide a better insight to our proposed approach. These are illustrated in Figure 3. Experiment 1 (Figure 3a), is training the generator with just the supervised dice loss without any multi-tasking. Experiment 2 (Figure 3b), is training the generator and discriminator using the loss in equation 2. Experiment 3 (Figure 3c), is not having the discriminator and multi-tasking just the generator, the loss used is $\lambda_1 \tilde{\mathcal{L}}_{dice}(G) + \lambda_2 \tilde{\mathcal{L}}_{huber}(G)$. Experiment 4 , is when just the generator is multitasked and not the discriminator, i.e. the discriminator is not asked to differentiate between "fake" and "real" distance transforms but just the affinities. Experiment 5 (Figure 3d), is our final formulation given in equation 5. The rand index on the BSD500 dataset for these experiments are reported in table 1.

The significant improvement in experiment 2 (Dice Loss+ Discriminator) when compared to experiment 1 (Dice Loss) justifies our initial motivation that the supervised losses have certain drawbacks and hence improving the loss function would result in better quality affinities. Another observation is that there is a minimal improvement from experiment 1 (Dice Loss) to experiment 3 (Dice Loss+ Multi Task Generator) and from experiment 2 (Dice Loss + Discriminator) to experiment 4 (Dice Loss + Discriminator + Multi Task Generator), where the only change is that the generator is multitasked. However, there is a significant improvement in the rand-index from experiment 4 (Dice Loss + Discriminator + Multi Task Generator) to experiment 5 (Dice Loss + Multi Task Generator + Multi Task Discriminator) where the only difference is that the discriminator is multitasked. This bolsters our claim that the improvement from experiment 2 (Dice Loss + Discriminator) to experiment 5 (Dice Loss + Multi Task Generator + Multi Task Discriminator) is due to the improved capability of the discriminator and not the additional supervised loss (Huber loss) in the multitasking generator.

Table 1: Rand Index on BSD500 for experiments in the ablation study

Experiment		Rand Index
1	Dice Loss	0.726
3	Dice Loss + Multi Task Generator	0.734
2	Dice Loss + Discriminator	0.79
4	Dice Loss + Discriminator + Multi Task Generator	0.805
5	Dice Loss + Multi Task Generator + Multi Task Discriminator	0.832
6	Dice Loss + Multi Task Generator + Multi Task Discriminator + Transfer Learning	0.845



(a) Experiment 1 - Training just the generator with the supervised dice loss.

(b) Experiment 2 - Training the generator and discriminator according to the loss in equation 2.

(c) Experiment 3 - Training just the discriminator in a multitask manner.

(d) Experiment 5 - Our final formulation as mentioned in equation 5.

Figure 3: Ablation study

4 Conclusion

The greedy Mutex Watershed segmentation algorithm [1] is especially susceptible to undersegmentation, when the input graph weights, describing the merge and split costs have small holes. We find that the pixelwise loss (used in [1]) is not sensitive enough to these errors and the algorithms input can be improved for natural images by adding a GAN loss. We find that this approach significantly improves the segmentation results, but suffers from training instabilities. Introducing a smooth auxiliary loss, we stabilize the cGAN training and furhter improve the segmentation accuracy. We hope that the proposed idea of adding an auxiliary multi task loss on a smoother target where the actual task has a discrete target can be extended to other image translation problems using GANs such as from pictures to sketches.

References

[1] Steffen Wolf, Constantin Pape, Alberto Bailoni, Nasim Rahaman, Anna Kreshuk, Ullrich Kothe, and FredA. Hamprecht. The mutex watershed: Efficient, parameter-free image partitioning. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[3] Lucas Fidon, Wenqi Li, Luis C Garcia-Peraza-Herrera, Jinendra Ekanayake, Neil Kitchen, Sébastien Ourselin, and Tom Vercauteren. Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks. In *International MICCAI Brainlesion Workshop*, pages 64–76. Springer, 2017.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[6] Peter J Huber et al. Robust estimation of a location parameter. *The annals of mathematical statistics*, 35(1):73–101, 1964.

[7] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423. IEEE, 2001.