Semi Supervised Semantic Segmentation Using Generative Adversarial Network
(2017)

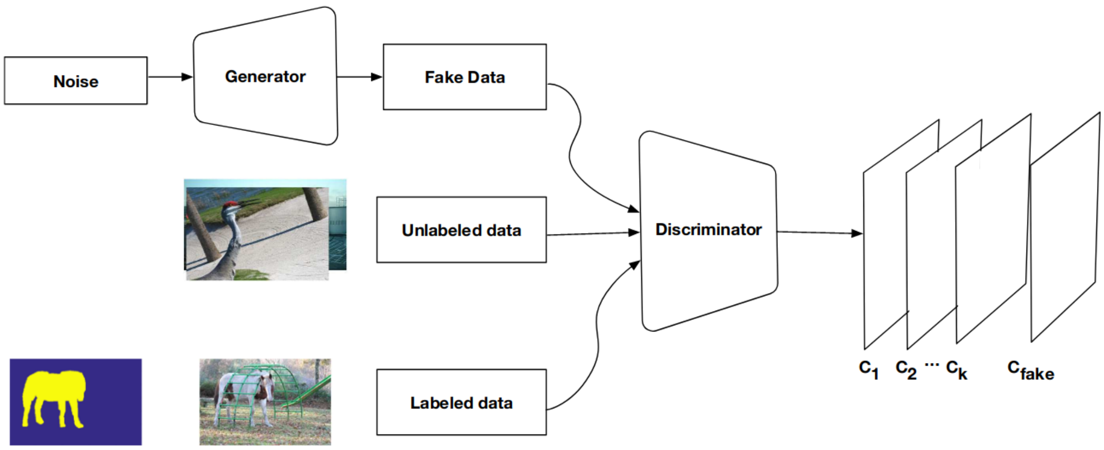
Nasim Souly, Concetto Spampinato, Mubarak Shah

Summary

Contributions

The authors propose to adapt Generative Adversarial Networks for semi-supervised segmentation, by extending the typical GaN framework to pixel level prediction. The discriminator this time is a fully convolutional network, predicting the class for each pixel if its a real image, or a fake class for generated images. They also use conditional GaNs to use additional weak labels.

Method



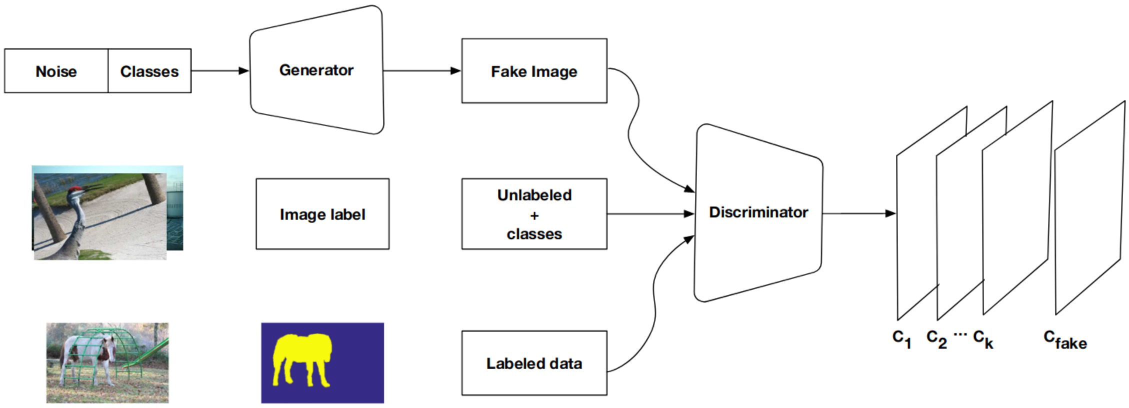
The authors propose to use use a segmentation network as a discriminator in GaN setting to exploit the unlabeled examples. The the discriminator is trained using the following loss:

$$\mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{data}}\ (x)} \log(D(x)) - \mathbb{E}_{z \sim p_z(z)} \log(1 - D(G(z)) + \gamma \mathbb{E}_{x,y \sim p(y,x)}[\text{CE}(y, P(y|x, D))]$$

The first two terms train the discriminator to distinguish between the real and fake images. In this case, all of the examples are used, if the provided real image comes from the labeled set where we have the labels, the third terms is also computed where the discriminator is trained not only to tell of the image is real or not, but also predict pixel-level classes, the outputs are have $K + 1$ classes, with K real classes and a one class for fake images. On the other hand, the generator is trained to trick the discriminator to predict wrong real / fake classes for the generated images.

$$\mathcal{L}_G = \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))]$$

Using weak labels



To use the image levels labels, the authors use conditional GaNs, in which the inputs for both the generator and discriminator are conditioned on a given class. For a given unlabeled image, this time we condition the generator over the same classes to generate fake images with the corresponding classes, and the same of the real image. The discriminator loss in this case is:

$$\mathcal{L}_D = -\mathbb{E}_{x,l\sim p_{data(x,l)}} \log\left[p\left(y \in K_i \subset 1 \ldots K|x\right)\right] - \mathbb{E}_{x,l\sim p_{z,l}(x,l)} \log[p(y = fake|x)]$$
$$+\gamma \mathbb{E}_{x,y\sim p(y,x)}[\mathbf{CE}(y, P(y|x, D))]$$

To condition on a given class, one hot vectors are concatenated to the noise sample for the generator, the same of the discriminator in which the one hot vector is concatenated with the input image.

Results

Table 1. The results on val set of VOC 2012 using all fully labeled and unlabeled data in train set.

method	pixel acc	mean acc	mean IU
Full - our baseline	89.9	69.2	59.5
Semi Supervised	90.5	80.7	64.1
Weak Supervised	91.3	80.0	65.8
FCN [14]	90.3	75.9	62.7
EM-Fixed [19]	-	-	64.6

Table 2. The results on VOC 2012 validation set using 30% of fully labeled data and all unlabeled data in training set.

method	pixel acc	mean acc	mean IU
Fully supervised	83.15	53.1	38.9
Semi supervised	83.6	60.0	42.2
Weak Supervised	84.6	58.6	44.6

Table 3. The results on SiftFlow using fully labeled data and 2000 unlabeled images from SUN2012

method	pixel acc	mean acc	mean IU
Fully supervised	83.4	46.7	34.4
Semi supervised	86.3	50.8	35.1
50% Fully Labeled	79.0	28.3	21.0
50% Full + Unlabeled	81.0	33.0	23.2

Table 6. The results on CamVid using fully labeled training data and 11k unlabeled frames from its videos.

method	pixel acc	mean acc	mean IU
Segnet-Basic [1]	82.2	62.3	46.3
SegNet (Pretrained) [1]	88.6	65.9	50.2
Ours Fully supervised	88.4	66.7	57.0
Ours Semi supervised	87.0	72.4	58.2

Table 4. The results using different percentages of fully labeled data and all unlabeled data in train set.

method	pixel acc	mean acc	mean IU
VOC 20% Full	73.15	23.2	16.0
VOC 20% Semi	79.6	27.1	19.8
VOC 50% Full	88.5	63.6	51.6
VOC 50% Semi	88.4	66.6	54.0

Table 5. The results on StanfordBG using fully labeled data and 10k unlabeled images from PASCAL dataset

method	pixel acc	mean acc	mean IU
Sem Seg Standard [15]	73.3	66.5	51.3
Sem Seg Adv [15]	75.2	68.7	54.3
Fully supervised	77.5	65.1	53.1
Semi supervised	82.3	77.6	63.3

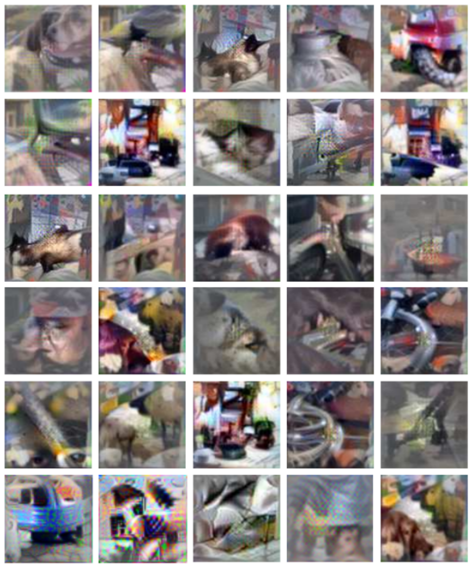


Figure 7. Images generated by the generator of our conditional GAN on the Pascal dataset. Interestingly, patterns related to dogs, cars, plants and cats have been automatically discovered. This highlights the effectiveness of our approach.