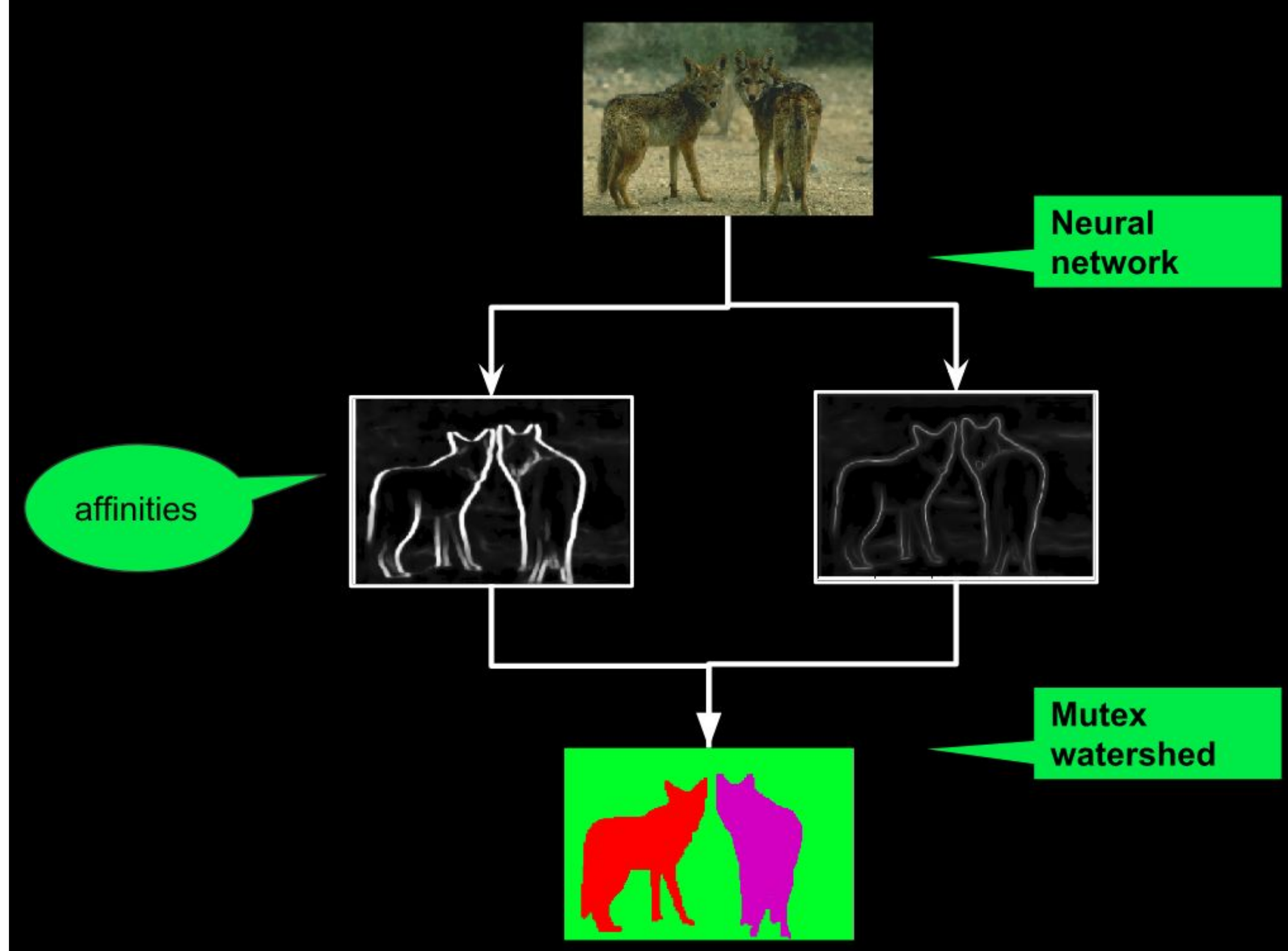
A GAN framework for Instance Segmentation using the Mutex Watershed Algorithm

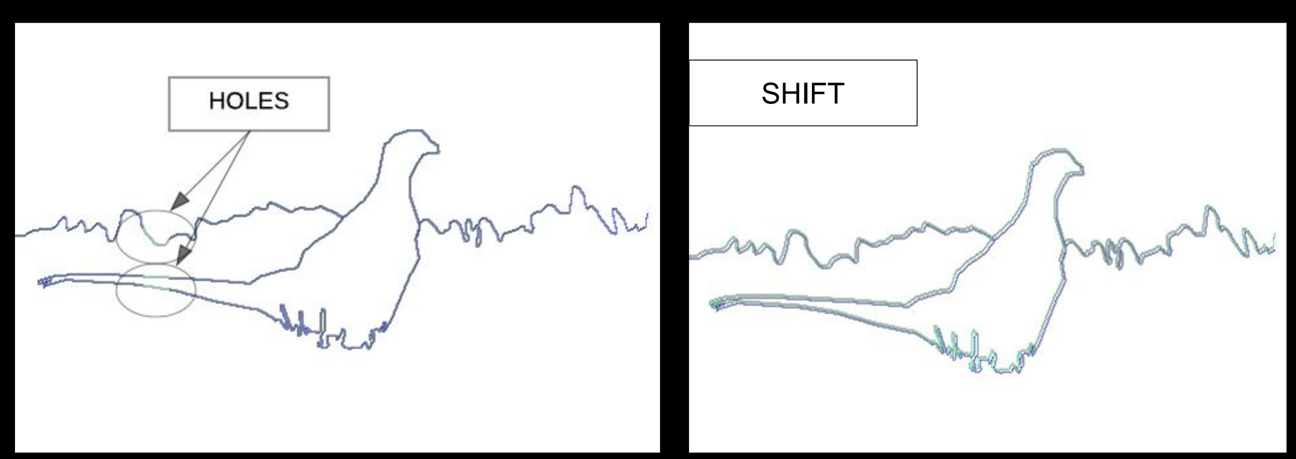
Mandikal Vikram[1] and Steffen Wolf[2]

Email: 15it217.vikram@nitk.edu.in, steffen.wolf@iwr.uni-heidelberg.de [1]National Institute of Technology Karnataka [2]HCI/IWR Heidelberg University

Segmentation Pipeline



Why GAN?

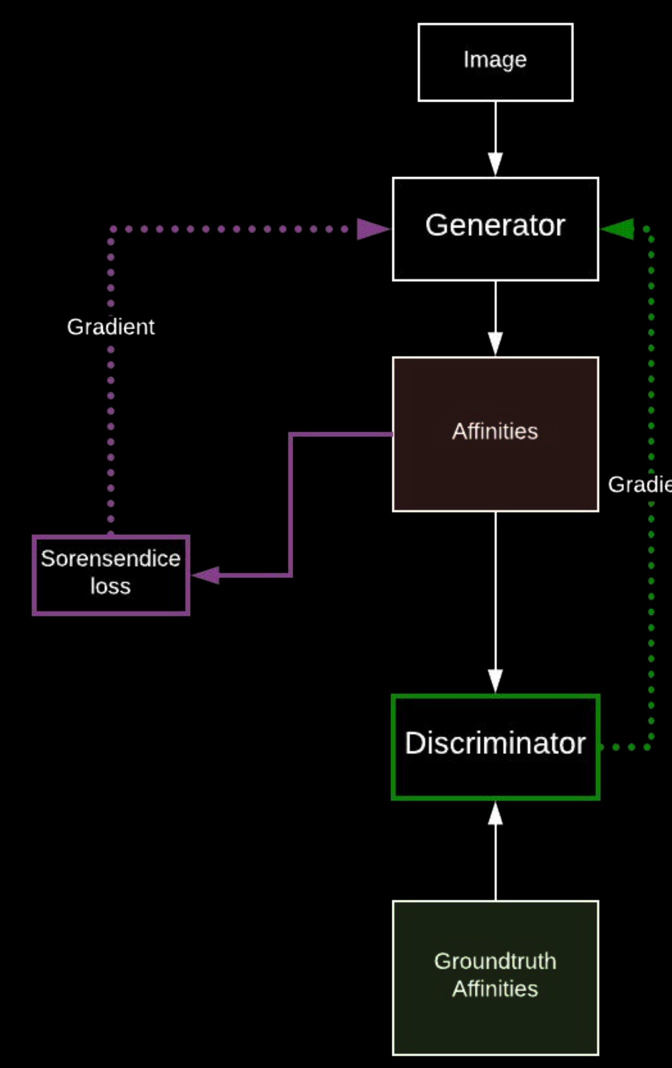


Pixelwise loss $\mathcal{J}$ is insensitive to holes and sensitive to shifts of the boundary.

- **Pixelwise loss:** $\frac{\mathcal{J}(y, y_{hole}) - \mathcal{J}(y,y)}{\mathcal{J}(y, y_{shifted}) - \mathcal{J}(y,y)} \approx 0.16$
- **Discriminator:** $\frac{\mathcal{D}(y, y_{hole}) - \mathcal{D}(y,y)}{\mathcal{D}(y, y_{shifted}) - \mathcal{D}(y,y)} \approx 3.6$

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

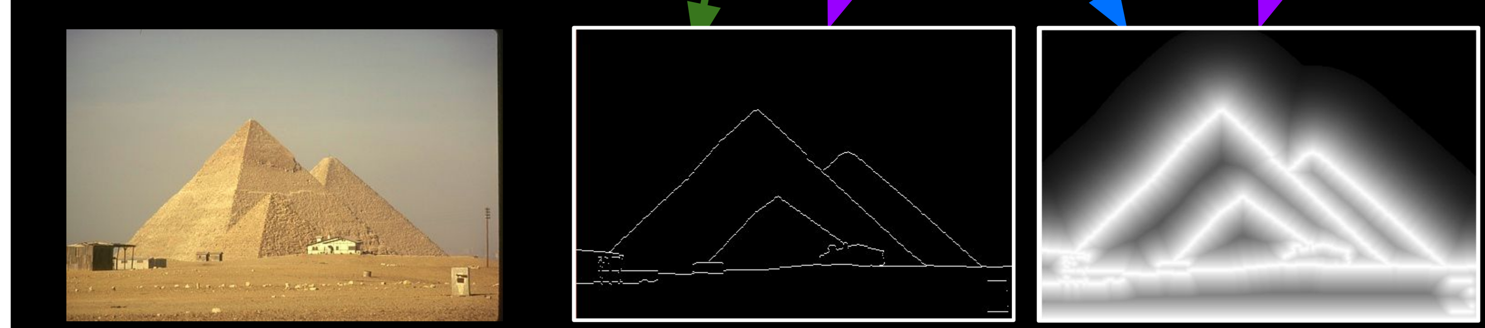
GAN as a Structured Loss



- Strongly penalizes holes in the affinities when compared to small shifts in the affinities.
- The GAN loss improves the qualities of the affinities.

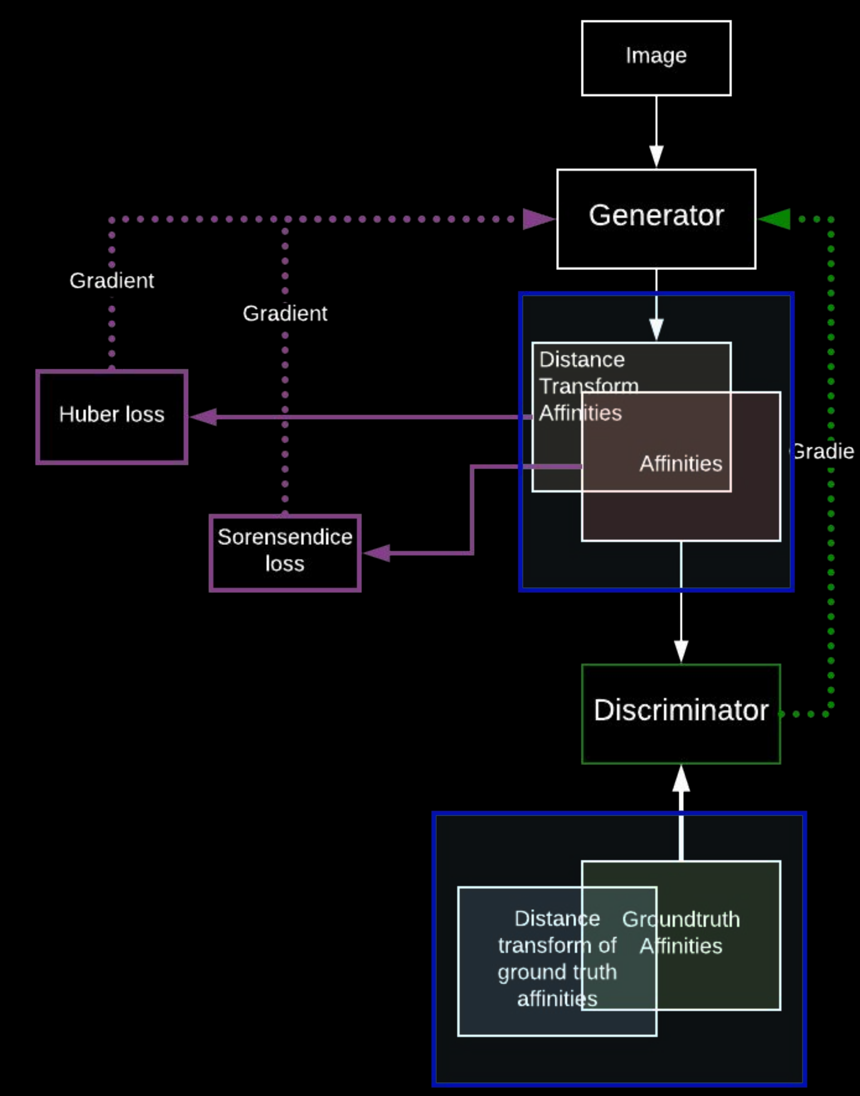
Stabilizing GAN Training

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G_a, D_a) + \mathcal{L}_{cGAN}(G_d, D_d) + \lambda_1 \tilde{\mathcal{L}}_{dice}(G) + \lambda_2 \tilde{\mathcal{L}}_{huber}(G)$$



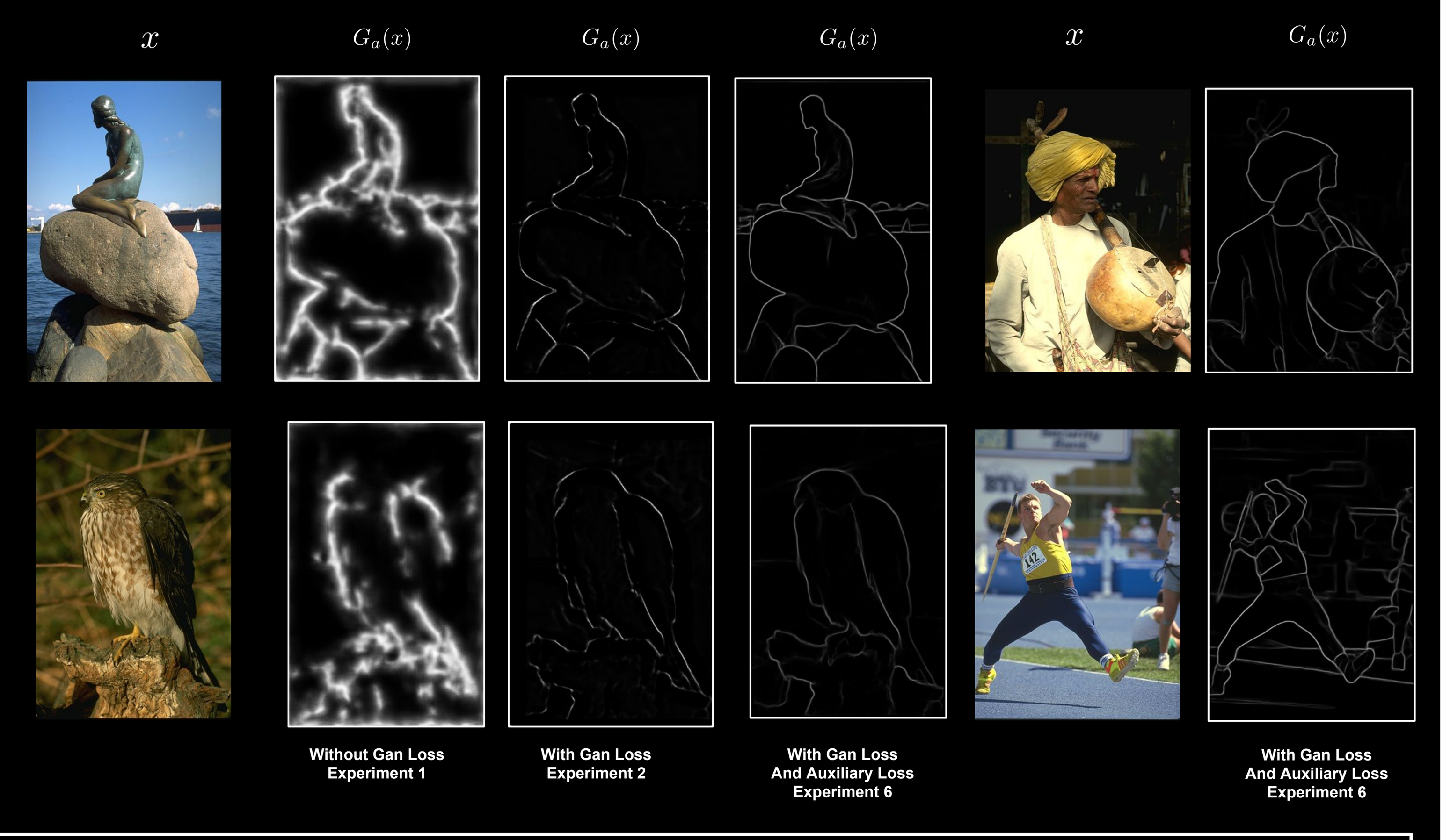
x $\quad G_a(x)$ $\quad G_d(x)$

Smooth Auxiliary Loss



- A smooth auxiliary-task loss which involves the <u>distance transform</u> of the affinities.
- This empirically stabilizes the training and further improves the quality of the affinities.

Results



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

Dataset: BSD500