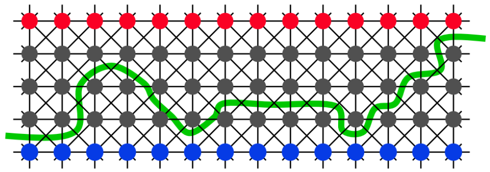
Devil is in the edges: Learning semantic boundaries from noisy annotations

4/7/19

Authors: David Acuna, Amlan Kar, Sanja Fidler
Presented by: Zilong Zhong

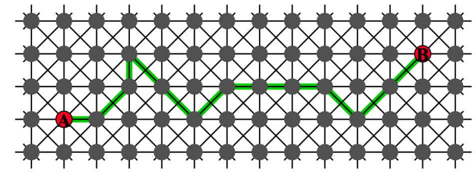
UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

What is the dual task of semantic segmentation?



A Cut separates regions

Graph Cut (Min Cut)

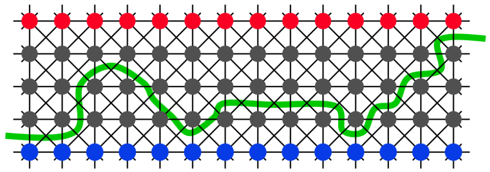


A Path connects points

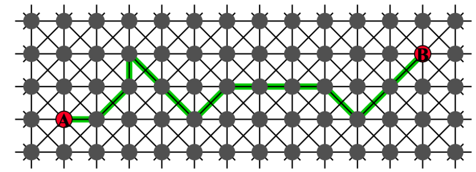
Path search (Max Flow)

Dual task

What is the dual task of semantic segmentation?



A Cut separates regions



A Path connects points

Graph Cut (Min Cut)

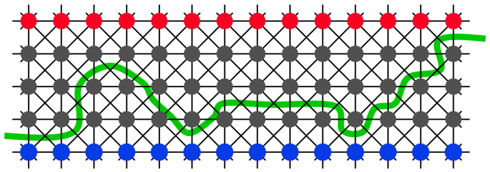
Dual task

Path search (Max Flow)

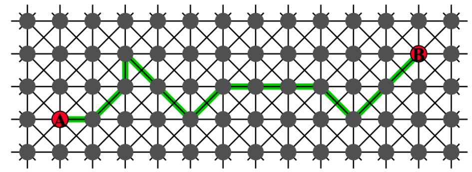
Multi-label

Semantic Segmentation

What is the dual task of semantic segmentation?



A Cut separates regions



A Path connects points

Graph Cut (Min Cut) ⟷ Dual task ⟷ Path search (Max Flow)

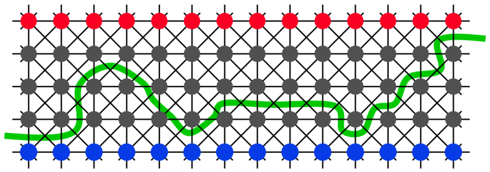
Multi-label ↓ Multi-label ↓

Semantic Segmentation ⟷ Dual task ⟷ ?

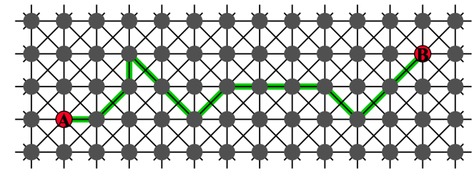
Devil is in the edges, Acuna et al, CVPR2019

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

Semantic-aware edge detection is the dual task of semantic segmentation



A Cut separates regions

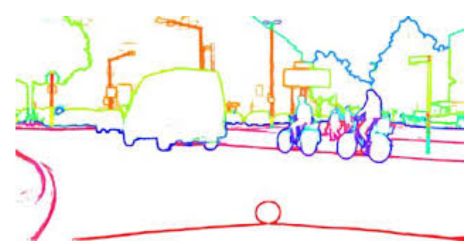


A Path connects points

Graph Cut (Min Cut) ⟷ Dual task ⟷ Path search (Max Flow)

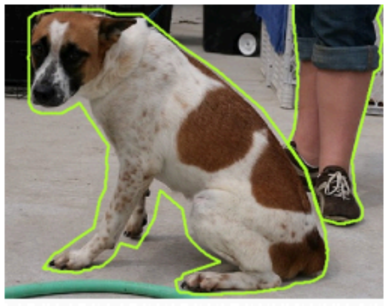
Multi-label ↓   Multi-label ↓

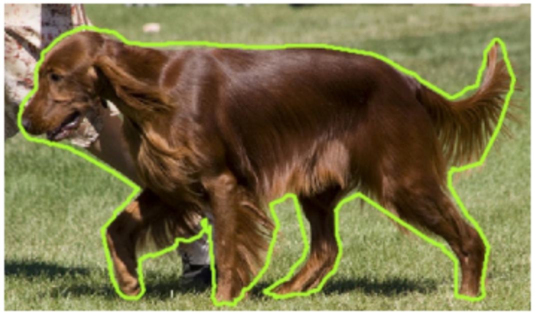
Semantic Segmentation ⟷ Dual task ⟷ Semantic edge detection



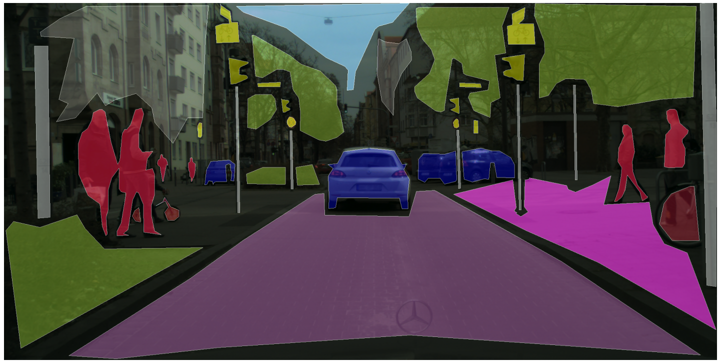
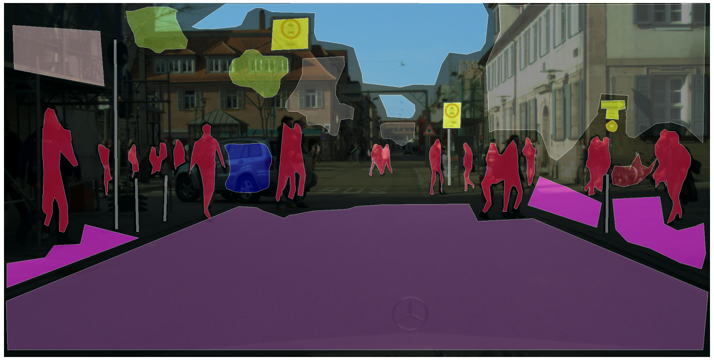
Noisy labels hinder precise semantic boundary detection and segmentation

Noisy labels (Semantic boundaries dataset)





Coarse labels (Cityscapes dataset)

Related works include semantic boundary detection and level set segmentation

- Semantic Boundary Detection
  - Canny edge detector [1987, Canny]
  - Semantic boundaries dataset (SBD)[ICCV2011, Hariharan]
  - Deep catogory-aware semantic edge detection (CASENet) [CVPR2017, Yu]
  - Simultaneous edge alignment and learning (SEAL) [ECCV2018, Yu]
- Level set segmentation
  - Geodestic object proposal [ECCV2014, Krahenbuhl]
  - Deep level sets for salient object detection [CVPR2017, Hu]
  - Deep structured active contours [CVPR2018, Marcos]
  - Deep extreme level set [CVPR2019, Wang]

Semantic edge detection and active alignment are defined as an optimization problem

Semantic edge detection is to predict boundary maps for K object classes given an input image $\mathbf{x}$ through maximizing the likelihood of $P(\mathbf{y}_k|\mathbf{x};\theta)$, where $y_k^m \in \{0,1\}$ indicate whether pixel m belongs class k

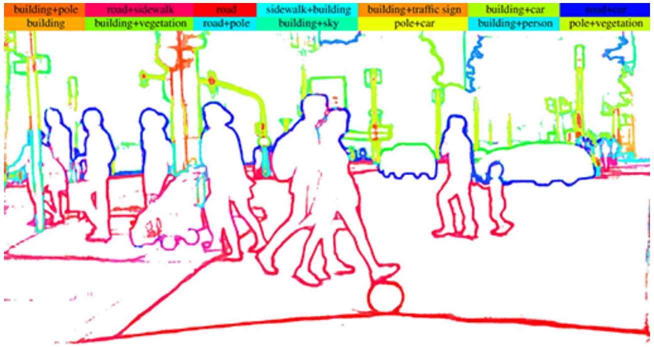
Active alignment is to find a more accurate version $\hat{\mathbf{y}}$ of ground-truth label y, where $\hat{\mathbf{y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_K\}$

Semantic edge detection and active alignment are defined as an optimization problem

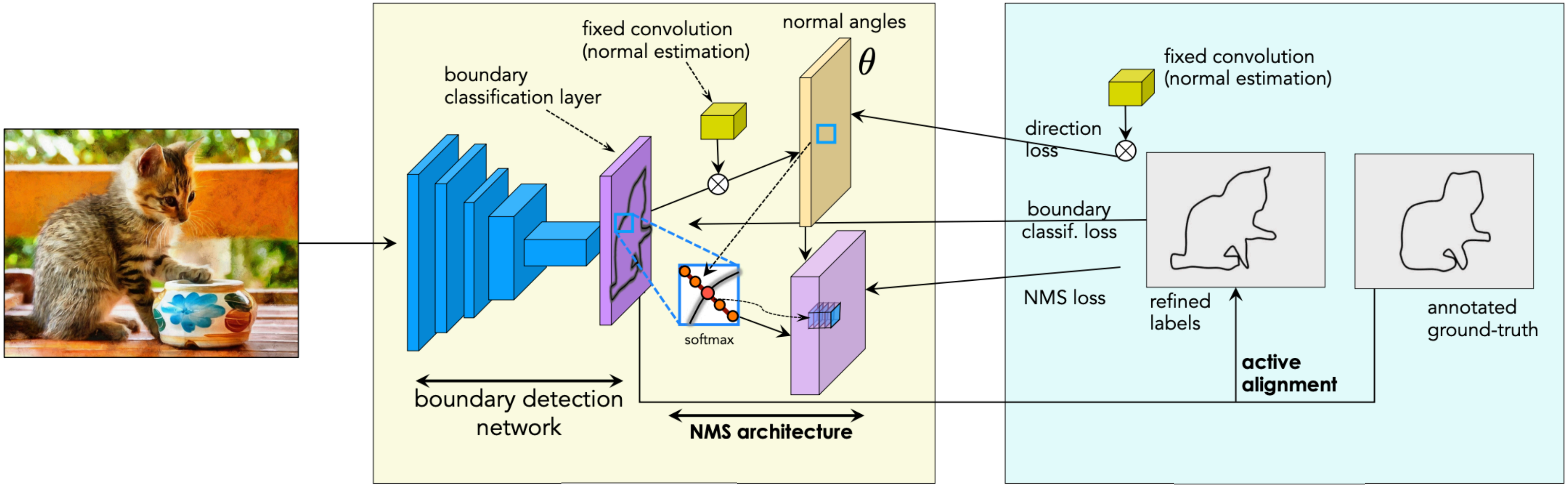
Semantic edge detection is to predict boundary maps for K object classes given an input image $\mathbf{x}$ through maximizing the likelihood of $P(\mathbf{y}_k|\mathbf{x};\theta)$, where $y_k^m \in \{0,1\}$ indicate whether pixel m belongs class k

Active alignment is to find a more accurate version $\hat{\mathbf{y}}$ of ground-truth label y, where $\hat{\mathbf{y}} = \{\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_K\}$

Objective: $\min_{\hat{\mathbf{y}},\theta} \mathcal{L}(\hat{\mathbf{y}},\theta) = \min_{\theta} \min_{\hat{\mathbf{y}}} \mathcal{L}(\hat{\mathbf{y}},\theta)$

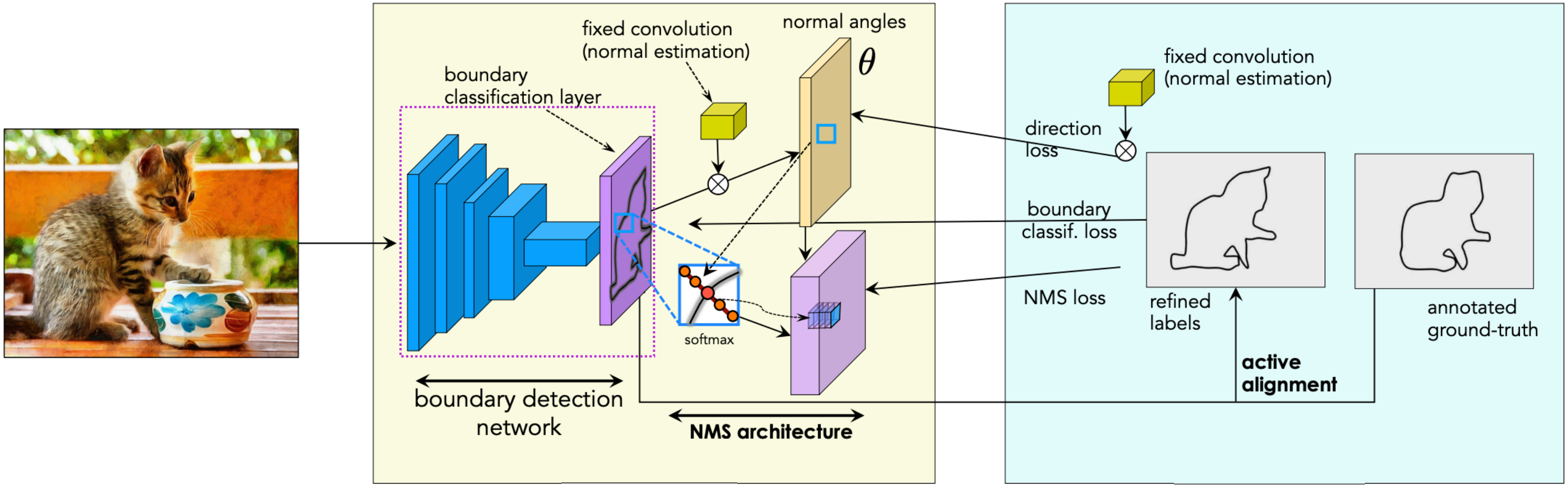


STEAL architecture consists of two parts: semantic edge detection and active alignment



STEAL: Semantic Thinning Edge Alignment Learning
Devil is in the Edges

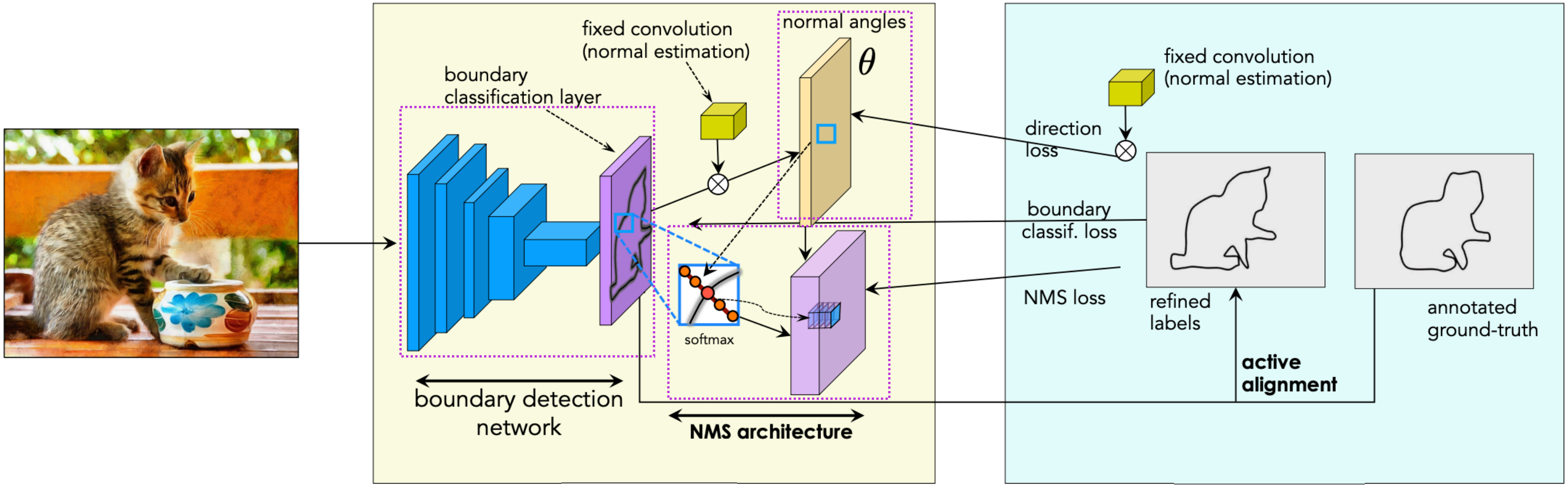
STEAL architecture consists of two parts: semantic edge detection and active alignment



1.1 Boundary loss

STEAL architecture consists of two parts: semantic edge detection and active alignment

1.3 Direction loss



1.1 Boundary loss

1.2 NMS loss

Non-maximum suppression

$$\mathcal{L} = \alpha_1 \, \mathcal{L}_{BCE} + \alpha_2 \, \mathcal{L}_{\text{nms}} + \alpha_3 \, \mathcal{L}_{dir}$$

Semantic boundary prediction involves thress losses

1.1 Boundary loss

$$\mathcal{L}_{BCE}(\theta) = -\sum_k \log P(\mathbf{y}_k|\mathbf{x};\theta)$$
$$= -\sum_k \sum_m \{\beta\, y_k^m \log f_k(m|\mathbf{x},\theta) +$$
$$+ (1-\beta)(1-y_k^m)\log(1-f_k(m|\mathbf{x},\theta))\}$$

1.2 NMS loss

Non-maximum suppression

$$\mathcal{L}_{nms}(\theta) = -\sum_k \sum_p \log h_k(p|\mathbf{x},\theta)$$

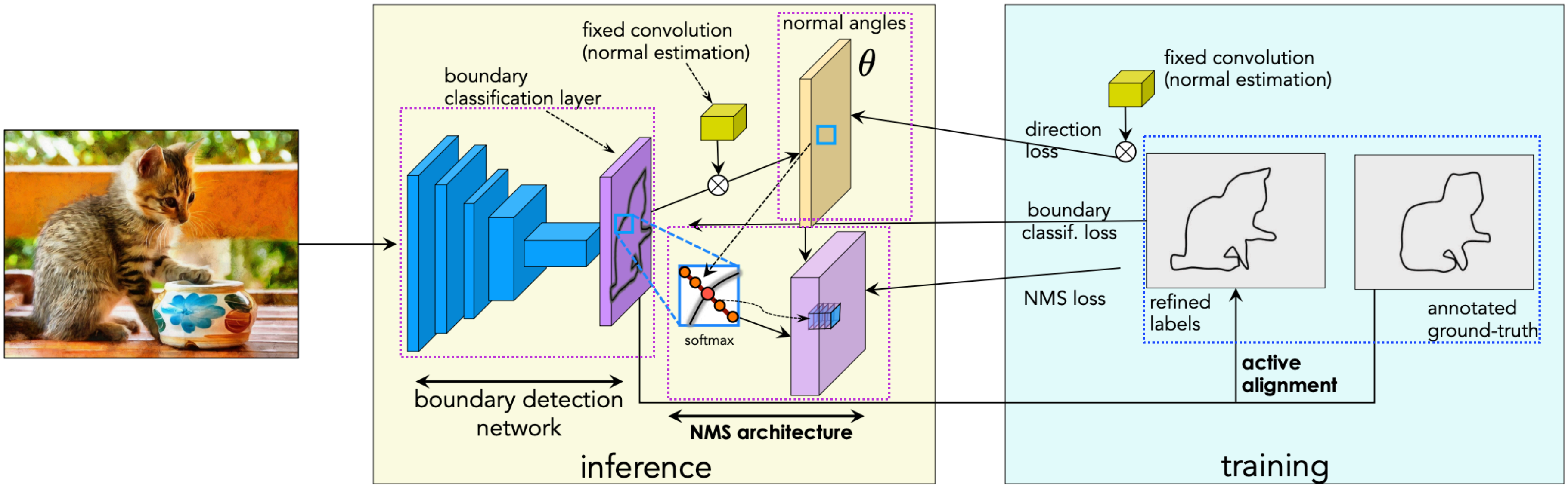
$$h_k(p|\mathbf{x},\theta) = \frac{\exp(f_k(p|\mathbf{x},\theta)/\tau)}{\sum_{t=-L}^{L}\exp(f_k(p_t|\mathbf{x},\theta)/\tau)} \qquad t \in \{-L, -L+1, \dots, L\}$$

1.3 Direction loss

$$\mathcal{L}_{\mathrm{dir}}(\theta) = \sum_k \sum_p ||\cos^{-1}\langle \vec{d_p}, \vec{e_p}(\theta)\rangle||$$

STEAL architecture consists of two parts: level set formulation and regularization loss

1.3 Direction loss



1.1 Boundary loss

1.2 NMS loss

2. Level set formulation

$$\mathcal{L} = \alpha_1 \, \mathcal{L}_{BCE} + \alpha_2 \, \mathcal{L}_{\text{nms}} + \alpha_3 \, \mathcal{L}_{dir}$$

Level set formulation is built for active edge alignment

2. Level set formulation

$$\hat{\mathbf{y}}_k = \{\Gamma : \phi(\Gamma, t) = 0\}\ \forall t$$

Geodestic active contours, Caselles et al., IJCV1997

Level set formulation is built for active edge alignment

2. Level set formulation

$$\hat{\mathbf{y}}_k = \{\Gamma : \phi(\Gamma, t) = 0\} \; \forall t$$

$$\begin{aligned}\min_{\hat{\mathbf{y}},\theta} \mathcal{L}(\hat{\mathbf{y}}, \theta) &= -\sum_k \log P(\mathbf{y}_k, \hat{\mathbf{y}}_k | \mathbf{x}; \theta) \\ &= -\sum_k \big(\underbrace{\log P(\mathbf{y}_k | \hat{\mathbf{y}}_k)}_{\text{Prior}} + \underbrace{\log P(\hat{\mathbf{y}}_k | \mathbf{x}; \theta)}_{\text{Edge detector}} \big)\end{aligned}$$

Geodestic active contours, Caselles et al., IJCV1997

Level set formulation is built for active edge alignment

2. Level set formulation

$$\hat{\mathbf{y}}_k = \{\Gamma : \phi(\Gamma, t) = 0\} \; \forall t$$

$$\begin{aligned}\min_{\hat{\mathbf{y}}, \theta} \mathcal{L}(\hat{\mathbf{y}}, \theta) &= -\sum_k \log P(\mathbf{y}_k, \hat{\mathbf{y}}_k | \mathbf{x}; \theta) \\ &= -\sum_k \big(\underbrace{\log P(\mathbf{y}_k | \hat{\mathbf{y}}_k)}_{\text{Prior}} + \underbrace{\log P(\hat{\mathbf{y}}_k | \mathbf{x}; \theta)}_{\text{Edge detector}} \big)\end{aligned}$$

Prior energe: $E(\mathbf{y}_k | \hat{\mathbf{y}}_k; \lambda; f_k) = \int_p g(f_k, \mathbf{y}_k, \lambda) \, \hat{\mathbf{y}}_k(p) \, |\hat{\mathbf{y}}'_k(p)| \, \partial p$

$$g(f_k, \mathbf{y}_k, \lambda) = \frac{1}{\sqrt{1 + |f_k|}} + \frac{\lambda}{\sqrt{1 + |\mathbf{y}_k|}}$$

Geodestic active contours, Caselles et al., IJCV1997

Level set formulation is built for active edge alignment

2. Level set formulation

Prior energe: $$E(\mathbf{y}_k|\hat{\mathbf{y}}_k;\lambda;f_k) = \int_p g(f_k,\mathbf{y}_k,\lambda)\;\hat{\mathbf{y}}_k(p)\;|\hat{\mathbf{y}}'_k(p)|\;\partial p$$

Gradient descent: $$\frac{\partial \hat{\mathbf{y}}(t)}{\partial t} = \kappa\, g(f,\mathbf{y},\lambda)\vec{\mathbf{n}} - (\nabla g(f,\mathbf{y},\lambda)\vec{\mathbf{n}})\vec{\mathbf{n}}$$

Geodestic active contours, Caselles et al., IJCV1997

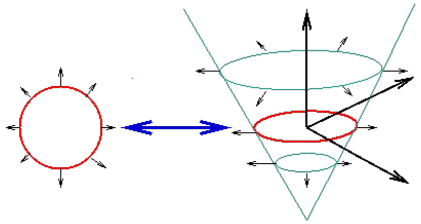
Level set formulation is built for active edge alignment

2. Level set formulation

Prior energe: $E(\mathbf{y}_k|\hat{\mathbf{y}}_k;\lambda;f_k) = \int_p g(f_k,\mathbf{y}_k,\lambda)\,\hat{\mathbf{y}}_k(p)\,|\hat{\mathbf{y}}'_k(p)|\,\partial p$

Gradient descent: $\frac{\partial \hat{\mathbf{y}}(t)}{\partial t} = \kappa\, g(f,\mathbf{y},\lambda)\vec{\mathbf{n}} - (\nabla g(f,\mathbf{y},\lambda)\vec{\mathbf{n}})\vec{\mathbf{n}}$

Level set trick: $\frac{\partial \hat{\mathbf{y}}_k(t)}{\partial t} = \beta\vec{\mathcal{N}} \Rightarrow \frac{\partial \phi(t)}{\partial t} = \beta|\vec{\nabla}\phi|$



Geodestic active contours, Caselles et al., IJCV1997

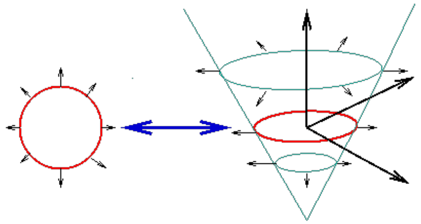
Level set formulation is built for active edge alignment

2. Level set formulation

Prior energe: $E(\mathbf{y}_k|\hat{\mathbf{y}}_k;\lambda;f_k) = \int_p g(f_k,\mathbf{y}_k,\lambda)\,\hat{\mathbf{y}}_k(p)\,|\hat{\mathbf{y}}'_k(p)|\,\partial p$

Gradient descent: $\dfrac{\partial \hat{\mathbf{y}}(t)}{\partial t} = \kappa\, g(f,\mathbf{y},\lambda)\vec{\mathbf{n}} - (\nabla g(f,\mathbf{y},\lambda)\vec{\mathbf{n}})\vec{\mathbf{n}}$

Level set trick: $\dfrac{\partial \hat{\mathbf{y}}_k(t)}{\partial t} = \beta\vec{\mathcal{N}} \Rightarrow \dfrac{\partial \phi(t)}{\partial t} = \beta|\vec{\nabla}\phi|$



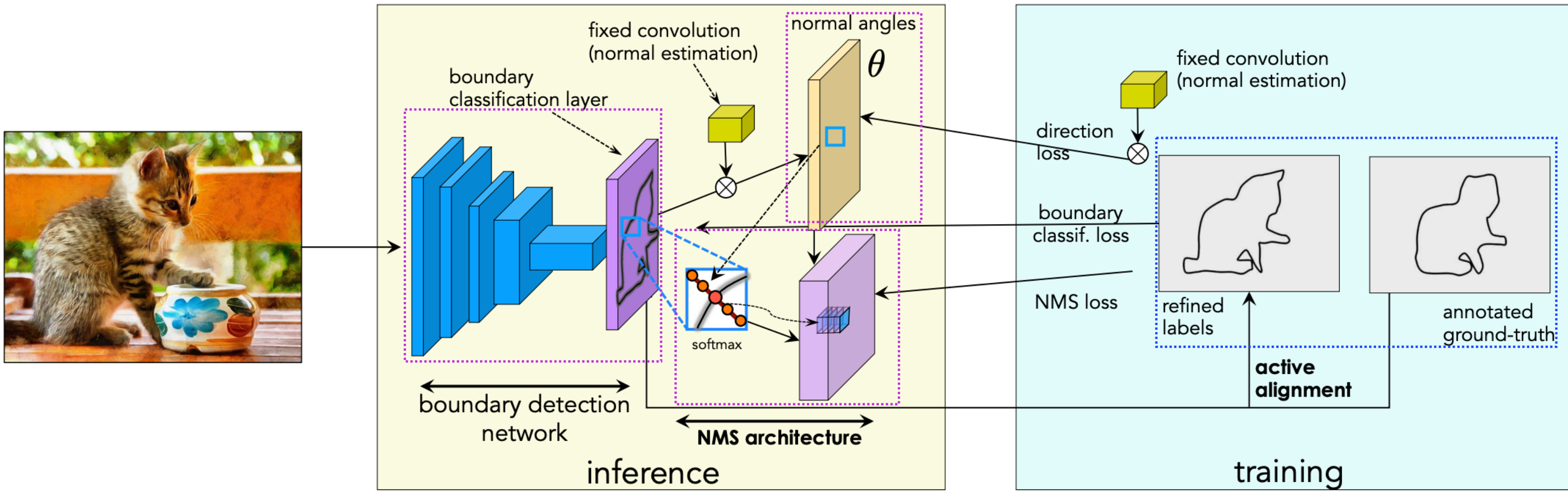
Rewrite evolution: $\dfrac{\partial \phi(t)}{\partial t} = g(f_k,\mathbf{y}_k,\lambda)(\kappa + c)|\nabla\phi| + \nabla g \nabla\phi$

Geodestic active contours, Caselles et al., IJCV1997

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

STEAL architecture consists of two parts: level set formulation and regularization loss

1.3 Direction loss



1.1 Boundary loss

1.2 NMS loss

2. Level set formulation

$$\mathcal{L} = \alpha_1\, \mathcal{L}_{BCE} + \alpha_2\, \mathcal{L}_{\text{nms}} + \alpha_3\, \mathcal{L}_{dir}$$

$$\frac{\partial \phi(t)}{\partial t} = g(f_k, \mathbf{y}_k, \lambda)(\kappa + c)|\nabla \phi| + \nabla g \nabla \phi$$

Minimizing the objective function is performed with an iterative two step optimization process

Objective function:

$$\min_{\hat{\mathbf{y}},\theta} \mathcal{L}(\hat{\mathbf{y}},\theta) = \min_{\theta} \min_{\hat{\mathbf{y}}} \mathcal{L}(\hat{\mathbf{y}},\theta)$$

Minimizing the objective function is performed with an iterative two step optimization process

Objective function:

$$\min_{\hat{\mathbf{y}},\theta} \mathcal{L}(\hat{\mathbf{y}},\theta) = \min_{\theta} \min_{\hat{\mathbf{y}}} \mathcal{L}(\hat{\mathbf{y}},\theta)$$

Step 1: Fixed θ, optimize ŷ

$$\min_{\hat{\mathbf{y}}_k} \mathcal{L}(\hat{\mathbf{y}}_k,\theta) = \min_{t}\{-\log P(\hat{\mathbf{y}}_k^t|\mathbf{x};\theta) - C\}$$

Step 2: Fixed ŷ, optimize θ

$$\min_{\theta} \mathcal{L}(\hat{\mathbf{y}}_k,\theta) = \min_{\theta} \alpha_1 \mathcal{L}_{BCE} + \alpha_2 \mathcal{L}_{\text{nms}} + \alpha_3 \mathcal{L}_{dir}$$

Experimental results show the effectiveness of STEAL in producing crisp semantic edges

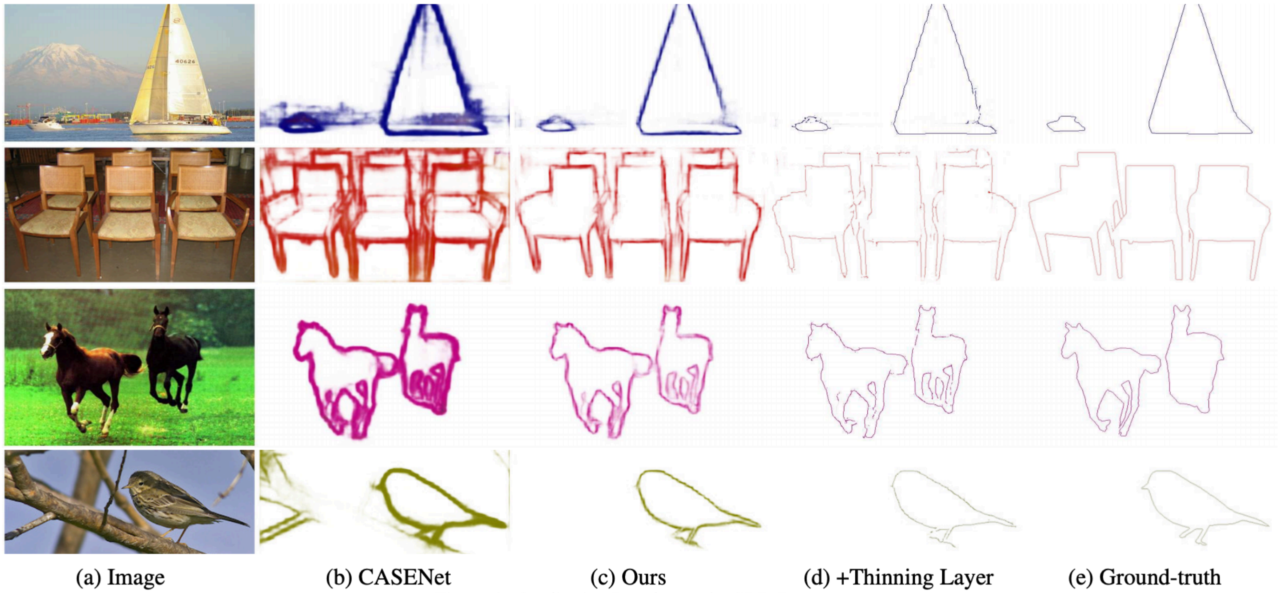
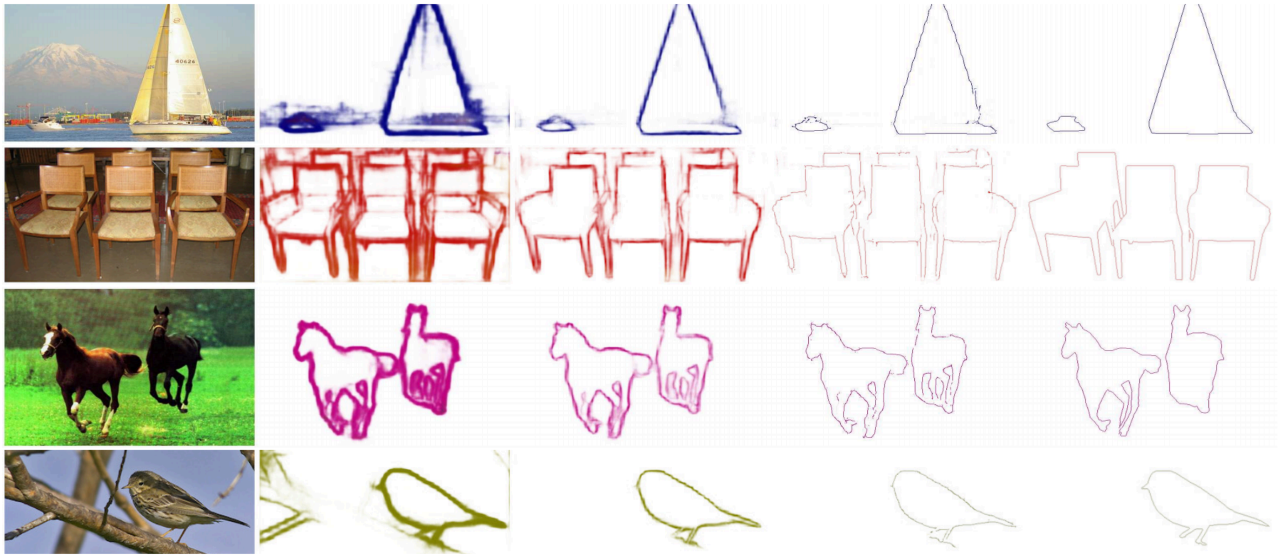


Figure 3: Qualitative Results on the SBD Dataset.

Experimental results show the effectiveness of STEAL in producing crisp semantic edges



(a) Image (b) CASENet (c) Ours (d) +Thinning Layer (e) Ground-truth

Figure 3: Qualitative Results on the SBD Dataset.

Metric	Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
MF (ODS)	CASENet	74.84	60.17	73.71	47.68	66.69	78.59	66.66	76.23	47.17	69.35	36.23	75.88	72.45	61.78	73.10	43.01	71.23	48.82	71.87	54.93	63.52
	CASENet-S	76.26	62.88	75.77	51.66	66.73	79.78	70.32	78.90	49.72	69.55	39.84	77.25	74.29	65.39	75.35	47.85	72.03	51.39	73.13	57.35	65.77
	SEAL	78.41	66.32	76.83	52.18	67.52	79.93	69.71	**79.37**	49.45	72.52	41.38	78.12	74.57	65.98	76.47	49.98	72.78	**52.10**	74.05	58.16	66.79
	Ours (NMS Loss)	78.96	66.20	77.53	**54.76**	69.42	**81.77**	**71.38**	78.28	**52.01**	**74.10**	42.79	79.18	76.57	66.71	77.71	49.70	74.99	50.54	**75.50**	59.32	67.87
	Ours (NMS Loss + AAlign)	**80.15**	**67.80**	**77.69**	54.26	**69.54**	81.48	71.34	78.97	51.76	73.61	**42.82**	**79.80**	**76.44**	**67.68**	**78.16**	**50.43**	**75.06**	50.99	75.31	**59.66**	**68.15**
AP	CASENet	50.53	44.88	41.69	28.92	42.97	54.46	47.39	58.28	35.53	45.61	25.22	56.39	48.45	42.79	55.38	27.31	48.69	39.88	45.05	34.77	43.71
	CASENet-S	67.64	53.10	69.79	40.51	62.52	73.49	63.10	75.26	39.96	60.74	30.43	72.28	65.15	56.57	70.80	33.91	61.92	45.09	67.87	48.93	57.95
	SEAL	74.24	57.45	72.72	42.52	65.39	74.50	65.52	**77.93**	40.92	65.76	**33.36**	76.31	68.85	58.31	73.76	38.87	66.31	**46.93**	69.40	**51.40**	61.02
	Ours (NMS Loss)	75.85	59.65	**74.29**	**43.68**	65.65	**77.63**	67.22	76.63	**42.33**	**70.67**	31.23	77.66	74.59	61.04	77.44	38.28	69.53	40.84	**71.69**	50.39	62.32
	Ours (NMS Loss + AAlign)	**76.74**	**60.94**	73.92	43.13	**66.48**	77.09	**67.80**	77.50	42.09	70.05	32.11	**78.42**	**74.77**	**61.28**	**77.52**	**39.02**	**68.51**	41.46	71.62	51.04	**62.57**

Table 1: Comparison of our method in the re-annotated SBD test set vs state-of-the-art. Scores are measured by %.

UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

Experimental results show the effectiveness of STEAL in tolenrence for noisy labels

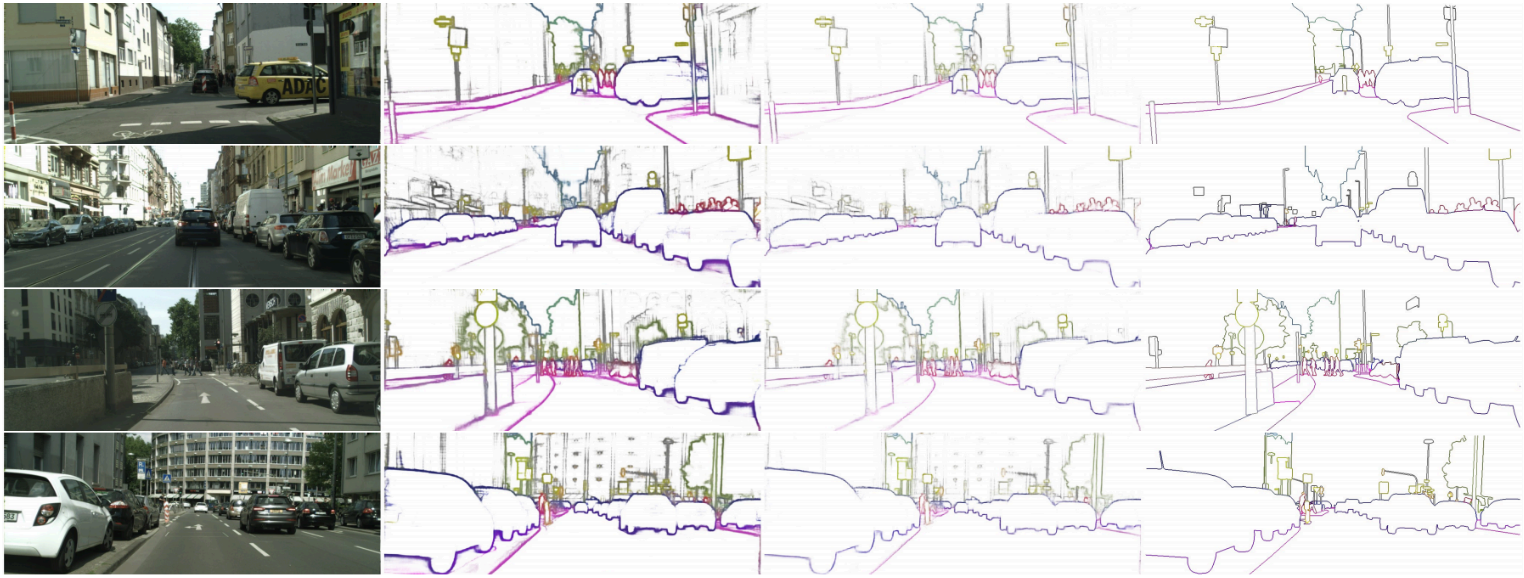
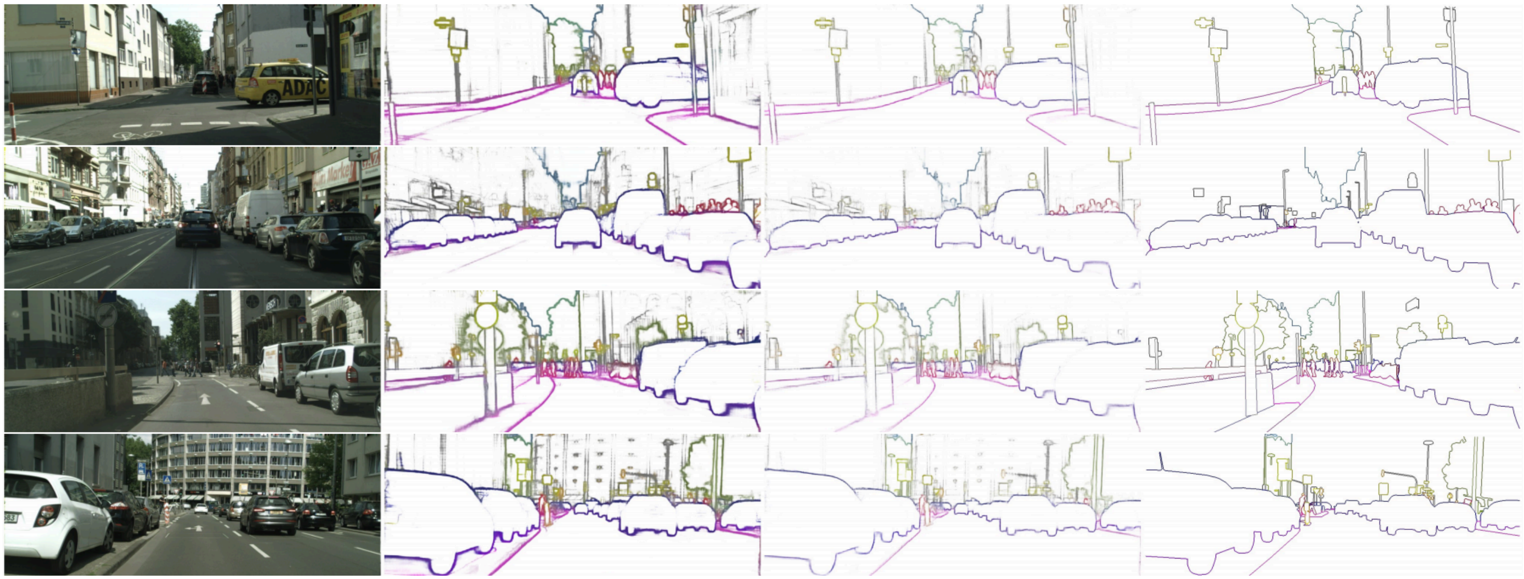


Figure 7: Qualitative Results on the Cityscapes Dataset.

Devil is in the edges, Acuna et al, CVPR2019

Experimental results show the effectiveness of STEAL in tolenrence for noisy labels



(a) Image (b) CASENet (c) Ours (d) Ground-truth

Figure 7: Qualitative Results on the Cityscapes Dataset.

Metric	Method	Test NMS	road	s.walk	build.	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mean
MF (ODS)	CASENet		87.06	75.95	75.74	46.87	47.74	73.23	72.70	75.65	80.42	57.77	86.69	81.02	67.93	89.10	45.92	68.05	49.63	54.21	73.74	68.92
	Ours(CASENet)		87.23	76.08	75.73	47.86	47.57	73.67	71.77	75.19	80.58	58.39	86.78	81.00	68.18	89.31	48.99	67.82	50.84	55.30	74.16	69.29
	Ours(CASENet)	✓	88.13	76.53	76.75	48.70	48.60	74.21	74.54	76.38	81.32	58.98	87.26	81.90	69.05	90.27	50.93	68.41	52.11	56.23	75.66	70.31
	+ NMS LOSS		88.08	77.62	77.08	50.02	49.62	75.48	74.01	76.66	81.51	59.41	87.24	81.90	69.87	89.50	52.15	67.80	**53.60**	55.93	75.17	70.67
	+ NMS LOSS	✓	**88.94**	**78.21**	**77.75**	**50.59**	**50.39**	**75.54**	**76.31**	**77.45**	**82.28**	**60.19**	**87.99**	**82.48**	**70.18**	**90.40**	**53.31**	**68.50**	53.39	**56.99**	**76.14**	**71.42**
AP	CASENet		54.58	65.44	67.75	37.97	39.93	57.28	64.65	69.38	71.27	50.28	73.99	72.56	59.92	66.84	35.91	56.04	41.19	46.88	63.54	57.65
	Ours(CASENet)		68.38	69.61	70.28	40.00	39.26	61.74	62.74	73.02	72.77	50.91	80.72	76.06	60.49	79.43	40.86	62.27	42.87	48.84	64.42	61.30
	Ours(CASENet)	✓	88.83	73.94	76.86	42.06	41.75	69.81	74.50	76.98	79.67	56.48	**87.73**	83.21	**68.10**	91.20	44.17	66.69	44.77	52.04	**75.65**	68.13
	+NMS LOSS		89.54	75.72	74.95	42.72	41.53	65.86	67.55	75.84	77.85	52.72	82.70	79.89	62.59	91.07	45.26	**67.73**	**47.08**	50.91	70.78	66.44
	+NMS LOSS	✓	**90.86**	**78.94**	**77.36**	**43.01**	**42.33**	**71.13**	**75.57**	**77.60**	**81.60**	**56.98**	87.30	**83.21**	66.79	**91.59**	**45.33**	66.64	46.25	**52.07**	74.41	**68.89**

Table 5: Results on the val set on the Cityscapes dataset. Training is done using the finely annotated train set. Scores are measured by %.

Experimental results show the effectiveness of STEAL in tolenrence for noisy labels

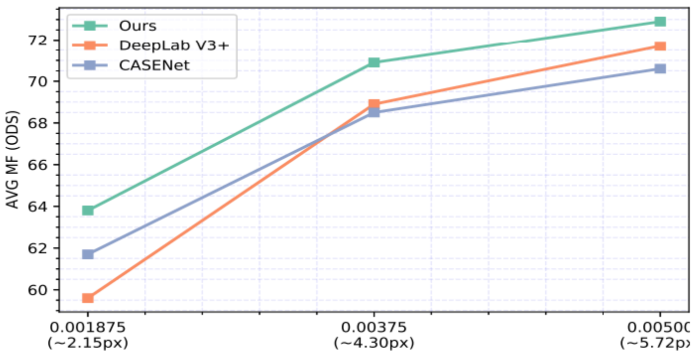


Figure 5: Comparison of our boundaries vs those obtained from DeepLab v3+'s segmentation masks. We perform 4.2% better at the strictest regime.

Experimental results show the effectiveness of STEAL in tolenrence for noisy labels

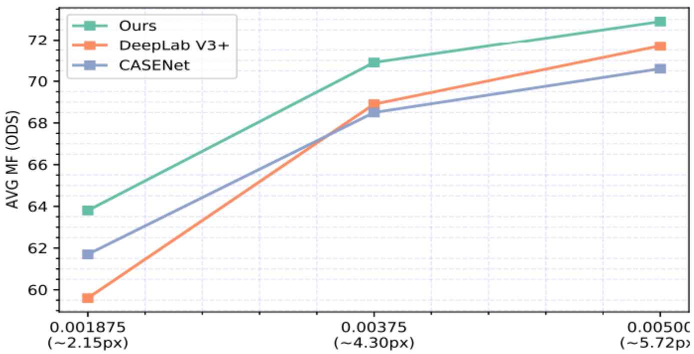


Figure 5: Comparison of our boundaries vs those obtained from DeepLab v3+'s segmentation masks. We perform 4.2% better at the strictest regime.

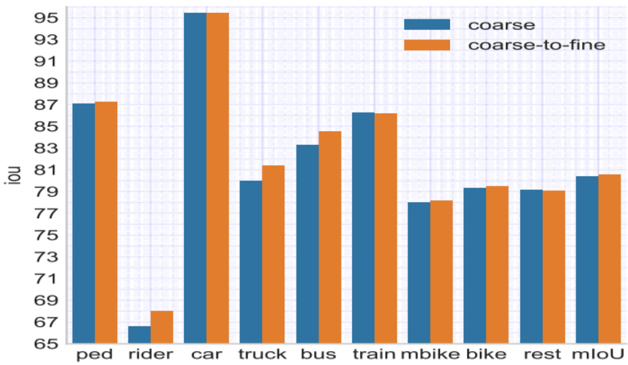


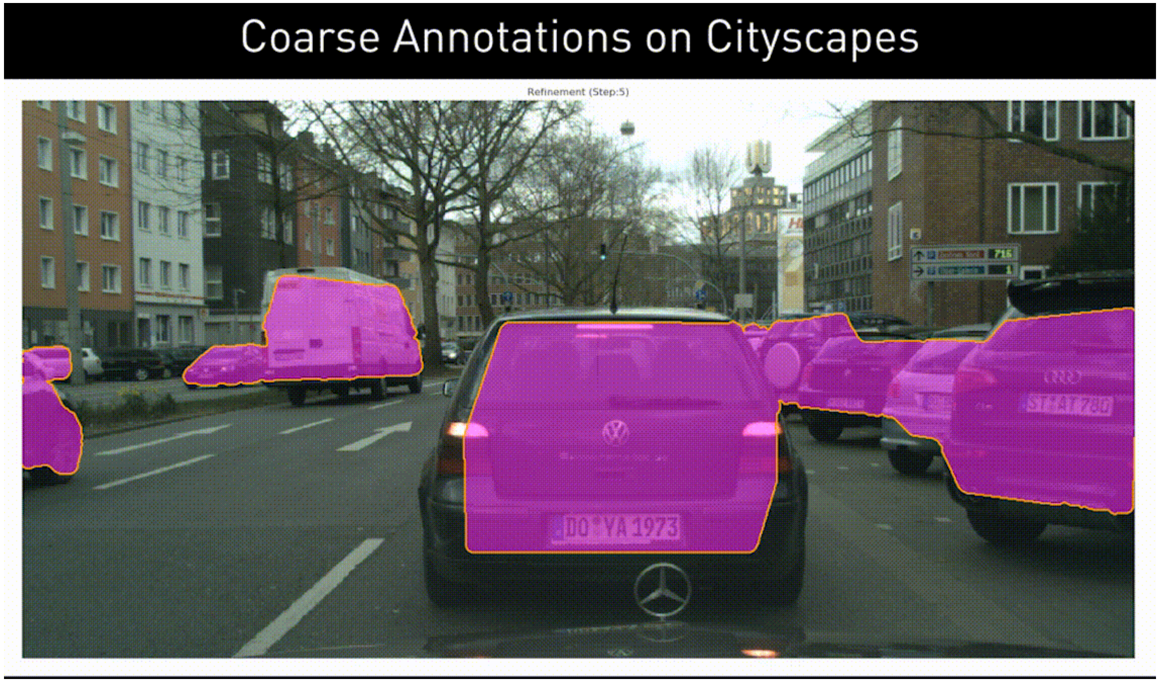
Figure 6: Semantic Segmentation on Cityscapes val: Performance of DeepLab V3+ when trained with fine data and (blue) vanilla train_extra set, (orange) our refined data (8 object classes) from train_extra. We see improvement of more than 1.2 IoU % in rider, truck and bus.

Conclusions

- Proposed a simple and effective Thinning Layer and NMS and direction loss that can be used in conjunction with existing boundary detectors
- Introduced a framework that reasons about true object boundaries during training to deal with the fact that most datasets have noisy annotations

Conclusions

- Proposed a simple and effective Thinning Layer and NMS and direction loss that can be used in conjunction with existing boundary detectors
- Introduced a framework that reasons about true object boundaries during training to deal with the fact that most datasets have noisy annotations



https://github.com/nv-tlabs/STEAL