Sketch Down the FLOPs:
Towards Efficient Networks for Human Sketch


Aneeshan Sain [1]


Subhajit Maity [2*]


Pinaki Nath Chowdhury [1]


Subhadeep Koley [1]


Ayan Kumar Bhunia [1]


Yi-Zhe Song [1]

1. SketchX, CVSSP, University of Surrey, United Kingdom
2. Department of Computer Science, University of Central Florida, United States

* Work done during an internship at SketchX before joining UCF

SketchX

CVSSP Centre for Vision, Speech and Signal Processing

CVPR Nashville JUNE 11-15, 2025

UNIVERSITY OF SURREY

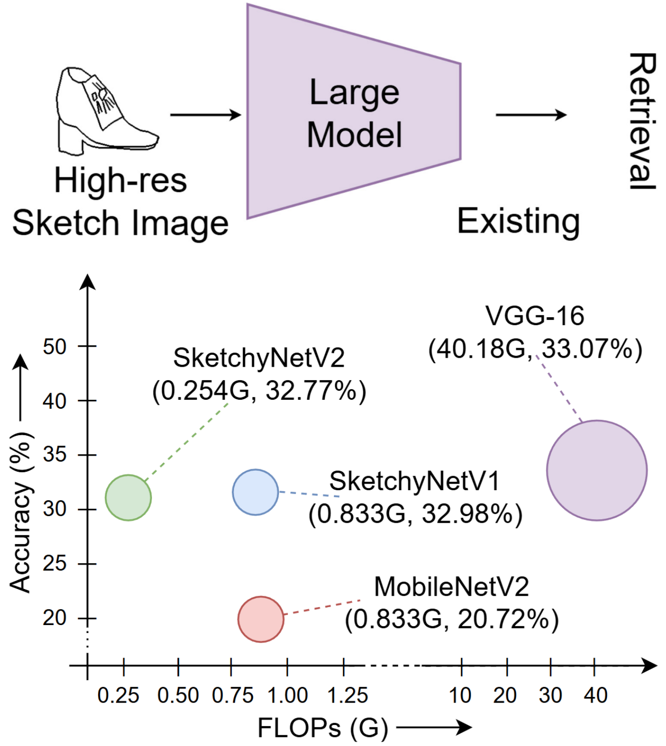
Motivation

- A practical challenge of sketch-based applications faced by the community till date – ***Inference Efficiency***.

- Existing sketch-models are **large** and **lacks designs catering to efficiency**, and thus, are **unfit for practical implementation**.

- **Efficient photo networks** with **limited sketch understanding** results in **lower accuracy** on sketch-based applications.

- We aim to produce **sketch-specific efficient models** to ensure **low computation overhead** while **retaining the task-level performance**.



Issues

- Sketch-specific designs.
- Performance *vs.* Efficiency.

Intuition

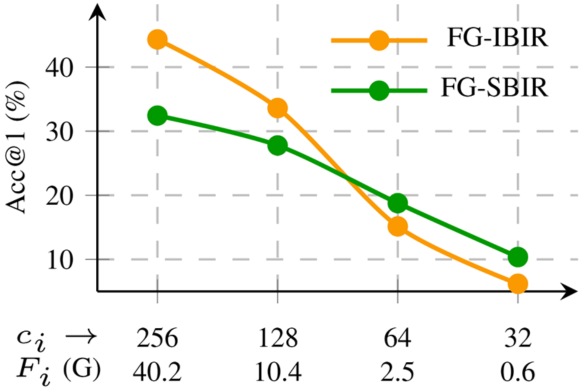
- Sketches are sparse and abstract – ideal for compressing information.

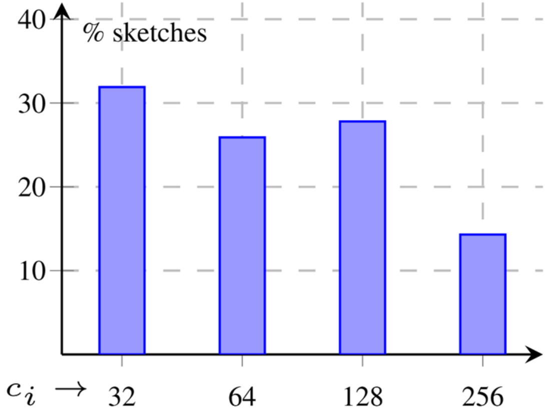
Proposed Approach

- Transfer performance from better models to efficient ones.
- Scaling input resolution.

Pilot Study

- Does sparsity help in lower resolution? – Let us consider the popular task, *Fine-Grained Image Retrieval*.
- Comparing the image-based retrieval task (**FG-IBIR**) *vs.* sketch-based retrieval task (**FG-SBIR**).
    - Photos are *pixel-dense*; retrieval **accuracy drops** rapidly **with input resolution**.
    - Sketches are *abstract* and *sparse*; more **tolerant to decreasing input resolution**.

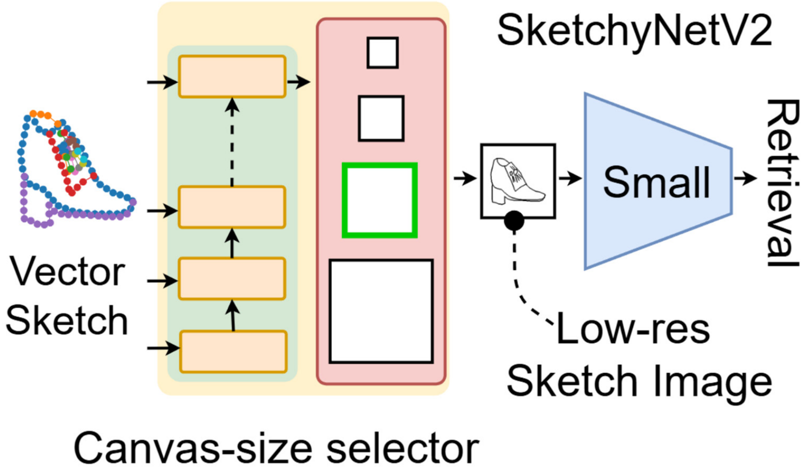




- The percentage of sketches needing a higher input resolution for a perfect retrieval decreases progressively.

Our Solution

- A *canvas-selector* to **predict input resolution**.
- Retrieval using a *small* and *efficient network* with task-level **knowledge transfer** from state-of-the-art models.



Our Framework

- **Baseline:** VGG16 backbone trained *sketch-photo triplet loss* as a **pre-trained teacher**.
- **SketchyNetV1:** A smaller network MobileNetV2 trained with **knowledge distillation**.
  - Supervision with traditional *sketch-photo triplet loss*.

$$\mathcal{L}_{\text{Tri}} = \max\{0, m + \delta(f_s, f_p) - \delta(f_s, f_n)\}$$

  - **Knowledge Distillation** of l_2 distance as *sketch-photo alignment* from **pre-trained teacher** to **student** using a *Huber Loss*.

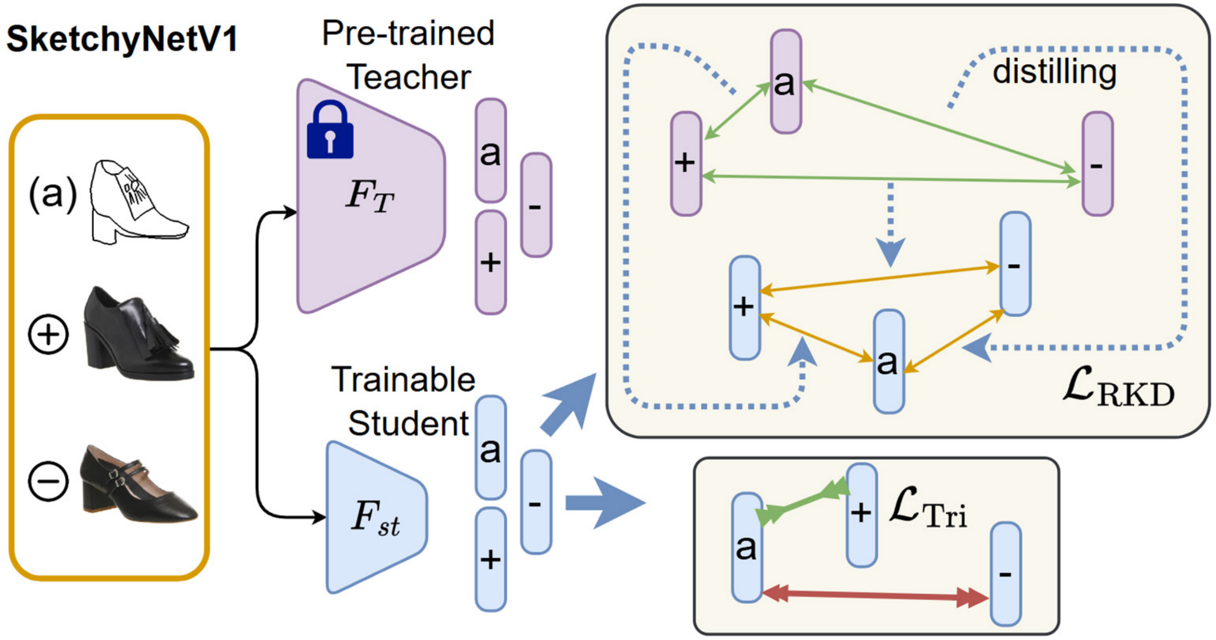
$$\mathcal{L}_{\text{RKD}}^{sp} = \mathcal{L}_{\delta}(d_{sp}^{T}, d_{sp}^{st})$$

  - Alignments between each pair of the sketch-photo triplet is considered.

$$\mathcal{L}_{\text{RKD}} = \mathcal{L}_{\text{RKD}}^{sp} + \mathcal{L}_{\text{RKD}}^{sn} + \mathcal{L}_{\text{RKD}}^{pn}$$

  - Total loss is weighted sum of both losses.

$$\mathcal{L}_{\text{trn}}^{st} = \lambda \mathcal{L}_{\text{Tri}} + (1-\lambda)\mathcal{L}_{\text{RKD}}$$



- Trained using sketches rendered with *varying canvas sizes*.

$$\mathcal{L}_{\text{trn}}^{st*} = \frac{1}{4}\sum_{i=1}^{4} \mathcal{L}_{\text{trn}}^{st(c_i)}$$ where, $c_i \in C$ and $C = \{32 \times 32, \cdots, 256 \times 256\}$

Our Framework

- **SketchyNetV2:** GRU or LSTM as an adaptive **canvas-size selector** and SketchyNetV1 student as critic.
  - Sketch-vector s_v for modeling $p(c|s_v)$ with **policy network** ψ_C followed by a linear layer (γ).

$$p(c|s_v) = \mathtt{softmax}(W_\gamma f_{s^T} + b_\gamma)$$

  - Sample **canvas-size** from $c_{pred} \sim \mathtt{categorical}([p(c_1|s_v), \cdots, p(c_K|s_v)])$.
  - *Lower FLOPs* refer to *higher rewards*.

$$R_{\text{comp}} = (-\mathcal{L}_{\text{F}}) \qquad \mathcal{L}_{\text{F}} = \frac{\sum_{j=1}^{K}(q_j \cdot \eta_j)}{q_{max} - q_{min}} \quad \text{where, } \eta_i = p(c_i|s_v)$$

  - *Lower* retrieval *rank* and *loss* refers to *higher rewards*.

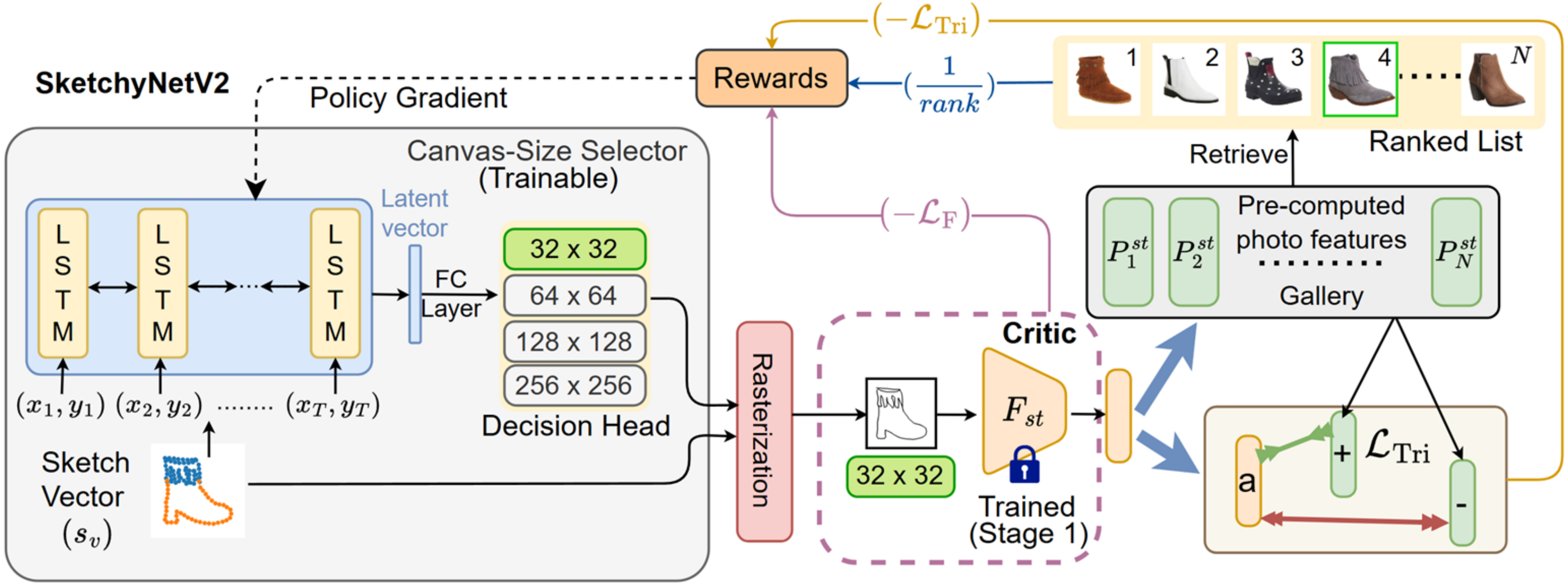
$$R_{\text{acc}} = \lambda_r(1/r) + \lambda_{\text{Tri}}(-\mathcal{L}_{\text{Tri}})$$

  - Total Reward:

$$R_{\text{Tot}} = \lambda_{\text{F}} R_{\text{comp}} + (1 - \lambda_{\text{F}}) R_{\text{acc}}$$

  - **Policy Gradient** optimization

$$\mathcal{L}_{PG}(\theta) = -\frac{1}{B}\sum_{i=1}^{B} \log p(c|s_v^i) \cdot R_{\text{Tot}}^i$$



Experiments

- **Datasets:** QMUL ShoeV2 [1], QMUL ChairV2 [1], Sketchy Extended [2], FS-COCO [3]
- **Competitors:** Triplet-SN [1]
  HOLEF-SN [4]
  Triplet-RL [5]
  StyleVAE [6]
  Partial-OT [7]
- **KD Baselines:** B-Regress – uses a l_2 regression [8]
  B-AKD – uses spatial attention maps [9]
  B-PKT – uses conditional probability [10]
- **Pruning Baselines:** B-Thinet – uses filter level strategy [11]
  B-Prune – uses convolutional pruning [12]
- **Other Baselines:** B-VGG16-SN – VGG16 with Triplet-SN [1]
- **Evaluation:** Top-q accuracy for task *performance* and FLOPs for *computation*.

[1] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In CVPR, 2016.
[2] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. ACM TOG, 2016.
[3] Pinaki Nath Chowdhury, Aneeshan Sain, Ayan Kumar Bhunia, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. FSCOCO: Towards understanding of freehand sketches of common objects in context. In ECCV, 2022.
[4] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In ICCV, 2017.
[5] Ayan Kumar Bhunia, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Sketch less for more: On-the-fly fine-grained sketch based image retrieval. In CVPR, 2020.
[6] Aneeshan Sain, Ayan Kumar Bhunia, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Stylemeup: Towards style-agnostic sketch-based image retrieval. In CVPR, 2021.
[7] Pinaki Nath Chowdhury, Ayan Kumar Bhunia, Viswanatha Reddy Gajjala, Aneeshan Sain, Tao Xiang, and Yi-Zhe Song. Partially does it: Towards scene-level FG-SBIR with partial input. In CVPR, 2022.
[8] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antonie Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In ICLR, 2015.
[9] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In ICLR, 2017.
[10] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In ECCV, 2018.
[11] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In ICCV, 2017.
[12] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. Preprint arXiv:1611.06440, 2016.

Quantitative Results

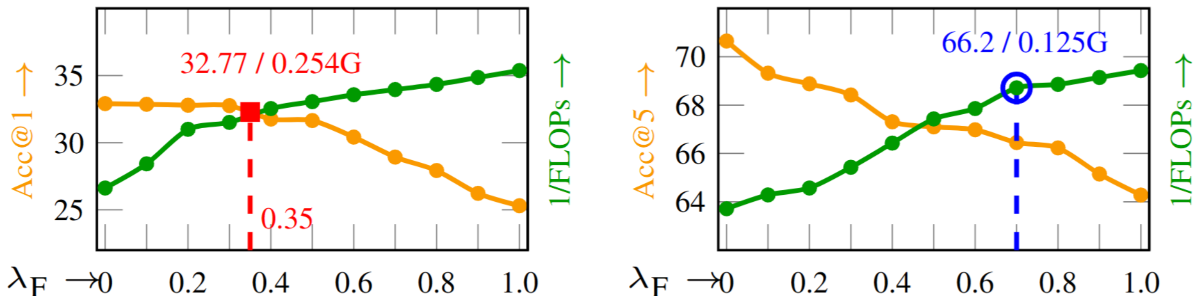
- Competitors and baselines adapted to our framework.

	Methods	Canvas-size c×c	Params (mil.)	ShoeV2 [1] Top1 (%)	ShoeV2 [1] Top10 (%)	ShoeV2 [1] FLOPS (G)	ChairV2 [1] Top1 (%)	ChairV2 [1] Top10 (%)	ChairV2 [1] FLOPS (G)	Sketchy [2] Top1 (%)	Sketchy [2] Top10 (%)	Sketchy [2] FLOPs (G)	FSCOCO [3] Top1 (%)	FSCOCO [3] Top10 (%)	FSCOCO [3] FLOPS (G)
State-of-the-Arts	Triplet-SN [1]	32x32	8.75	09.77 (↓18.94)	24.82 (↓46.74)	0.083	15.61 (↓32.04)	29.86 (↓54.38)	0.083	4.23 (↓11.12)	12.58 (↓23.13)	0.083	0.76 (↓3.94)	6.22 (↓14.8)	0.083
		64x64	8.75	17.63 (↓11.08)	44.96 (↓26.60)	0.338	30.25 (↓17.4)	54.21 (↓30.03)	0.338	9.68 (↓5.67)	22.45 (↓13.26)	0.338	1.85 (↓2.85)	7.33 (↓13.69)	0.338
		128x128	8.75	25.01 (↓3.70)	62.35 (↓9.21)	1.397	40.61 (↓7.04)	72.68 (↓11.56)	1.397	12.98 (↓2.37)	29.63 (↓6.08)	1.397	3.28 (↓1.42)	15.92 (↓5.1)	1.397
		256x256	8.75	28.71	71.56	5.280	47.65	84.24	5.280	15.35	35.71	5.280	4.7	21.02	5.280
		SketchyNetV1	2.22	28.46 (↓0.25)	69.01 (↓2.55)	0.833	46.28 (↓0.37)	82.33 (↓1.91)	0.833	14.85 (↓0.5)	34.68 (↓1.03)	0.833	4.22 (↓0.48)	18.33 (↓2.69)	0.833
		SketchyNetV2	2.27	27.89 (↓0.82)	68.76 (↓2.80)	0.264	45.98 (↓1.67)	80.14 (↓4.10)	0.294	14.21 (↓1.14)	34.16 (↓1.55)	0.321	3.98 (↓0.72)	17.64 (↓3.38)	0.423
	HOLEF-SN [4]	32x32	9.31	10.84 (↓20.9)	27.28 (↓48.5)	0.096	18.72 (↓34.69)	31.68 (↓55.88)	0.096	6.01 (↓10.69)	14.12 (↓24.78)	0.096	0.86 (↓4.04)	6.93 (↓14.78)	0.096
		64x64	9.31	19.83 (↓11.91)	51.72 (↓24.06)	0.387	34.32 (↓19.09)	56.46 (↓31.1)	0.387	9.92 (↓6.78)	24.38 (↓14.52)	0.387	2.11 (↓2.79)	7.86 (↓13.85)	0.387
		128x128	9.31	26.78 (↓4.96)	66.28 (↓9.5)	1.545	45.74 (↓7.67)	75.59 (↓11.97)	1.545	13.81 (↓2.89)	33.68 (↓5.22)	1.545	3.65 (↓1.25)	16.32 (↓5.39)	1.545
		256x256	9.31	31.74	75.78	5.758	53.41	87.56	5.758	16.70	38.90	5.758	4.9	21.71	5.758
		SketchyNetV1	2.22	31.59 (↓0.15)	73.96 (↓1.82)	0.833	53.27 (↓0.14)	86.93 (↓0.63)	0.833	16.25 (↓0.45)	38.21 (↓0.69)	0.833	4.56 (↓0.34)	19.92 (↓1.79)	0.833
		SketchyNetV2	2.27	30.86 (↓0.88)	72.56 (↓3.22)	0.259	52.74 (↓0.67)	84.02 (↓3.54)	0.289	15.91 (↓0.79)	37.88 (↓1.02)	0.315	4.04 (↓0.86)	18.63 (↓3.08)	0.417
	Triplet-RL [5]	32x32	22.1	11.98 (↓22.12)	28.11 (↓50.71)	0.142	20.53 (↓36.01)	32.59 (↓57.02)	0.142	1.76 (↓2.94)	3.88 (↓6.47)	0.142	–	–	–
		64x64	22.1	21.28 (↓12.82)	54.71 (↓24.11)	0.577	36.09 (↓20.45)	57.33 (↓32.28)	0.577	3.04 (↓1.66)	6.79 (↓3.56)	0.577	–	–	–
		128x128	22.1	28.83 (↓5.27)	67.84 (↓10.98)	2.299	48.38 (↓8.16)	77.25 (↓12.36)	2.299	4.12 (↓0.58)	9.05 (↓1.30)	2.299	–	–	–
		256x256	22.1	34.10	78.82	6.041	56.54	89.61	6.041	4.70	10.35	6.041	–	–	–
		SketchyNetV1	2.22	33.88 (↓0.22)	77.15 (↓1.67)	0.833	55.92 (↓0.62)	88.20 (↓1.41)	0.833	4.59 (↓0.11)	10.21 (↓0.14)	0.833	–	–	–
		SketchyNetV2	2.27	33.26 (↓0.84)	76.84 (↓1.98)	0.255	55.14 (↓1.40)	87.31 (↓2.30)	0.285	4.51 (↓0.19)	9.89 (↓0.46)	0.308	–	–	–
	StyleVAE [6]	32x32	25.37	12.68 (↓23.79)	28.69 (↓53.14)	0.125	22.48 (↓40.38)	32.10 (↓59.04)	0.125	06.92 (↓12.7)	16.25 (↓29.53)	0.125	–	–	–
		64x64	25.37	22.93 (↓13.54)	57.71 (↓24.12)	0.508	40.26 (↓22.6)	58.23 (↓32.91)	0.508	12.46 (↓7.16)	29.62 (↓16.16)	0.508	–	–	–
		128x128	25.37	30.91 (↓5.56)	70.06 (↓11.77)	2.024	53.88 (↓8.98)	78.64 (↓12.5)	2.024	17.15 (↓2.47)	38.84 (↓6.94)	2.024	–	–	–
		256x256	25.37	36.47	81.83	5.642	62.86	91.14	5.642	19.62	45.78	5.642	–	–	–
		SketchyNetV1	2.22	36.11 (↓0.36)	79.63 (↓2.20)	0.833	62.23 (↓0.63)	88.95 (↓2.19)	0.833	19.48 (↓0.12)	44.02 (↓1.76)	0.833	–	–	–
		SketchyNetV2	2.27	35.88 (↓0.59)	78.46 (↓3.37)	0.251	61.95 (↓0.91)	87.68 (↓3.46)	0.281	19.10 (↓0.50)	43.57 (↓2.21)	0.304	–	–	–
	Partial-OT [7]	32x32	22.1	–	–	–	–	–	–	–	–	–	6.11 (↓18.05)	24.18 (↓29.74)	0.196
		64x64	22.1	–	–	–	–	–	–	–	–	–	12.32 (↓11.84)	37.65 (↓16.27)	0.687
		128x128	22.1	–	–	–	–	–	–	–	–	–	19.61 (↓4.55)	45.12 (↓8.8)	2.876
		256x256	22.1	–	–	–	–	–	–	–	–	–	24.16	53.92	6.512
		SketchyNetV1	2.22	–	–	–	–	–	–	–	–	–	23.89 (↓0.27)	51.29 (↓2.63)	0.833
		SketchyNetV2	2.27	–	–	–	–	–	–	–	–	–	23.77 (↓0.39)	50.68 (↓3.24)	0.325
Baselines	B-BFR	224x224	4.61	24.32	61.23	2.602	43.39	76.33	2.602	10.47	24.34	2.602	2.15	11.61	2.602
	B-DRS	Dynamic	9.36	25.12	68.84	5.493	44.78	78.14	6.187	11.87	26.58	6.245	3.86	17.63	6.291
	B-Crop	Dynamic	4.92	20.32	52.12	6.562	35.07	57.68	6.597	08.27	20.19	6.629	1.98	09.67	6.688
	B-Regress [8]	Dynamic	2.27	22.82	61.33	0.261	38.42	64.65	0.291	08.69	21.98	0.323	2.07	10.15	0.375
	B-AKD [9]	Dynamic	2.27	23.67	64.26	0.268	38.61	71.88	0.295	12.67	27.71	0.331	2.91	14.01	0.392
	B-PKT [10]	Dynamic	2.27	24.31	65.91	0.259	42.68	73.66	0.288	12.27	28.31	0.317	3.45	16.22	0.372
	B-VGG16-SN	256x256	14.71	33.03	78.51	40.18	52.16	84.08	40.18	18.61	41.25	40.18	5.16	23.62	40.18
	B-ThiNet [11]	256x256	8.32	7.51 (↓25.52)	22.34 (↓56.17)	9.342	12.31 (↓39.85)	26.63 (↓57.45)	9.342	3.16 (↓15.45)	12.73 (↓28.52)	9.342	0.65 (↓4.51)	5.35 (↓18.27)	9.342
	B-Prune [12]	256x256	6.17	3.92 (↓29.11)	17.32 (↓61.19)	8.138	7.98 (↓44.18)	19.82 (↓64.26)	8.138	1.45 (↓17.16)	8.22 (↓33.03)	8.138	0.53 (↓4.63)	4.97 (↓18.65)	8.138
	B-VGG16-SN	SketchyNetV2	2.27	32.77 (↓0.26)	78.21 (↓0.3)	0.254	50.75 (↓1.41)	82.21 (↓1.87)	0.283	16.55 (↓2.06)	38.66 (↓2.59)	0.332	4.69 (↓0.47)	21.76 (↓1.86)	0.407

Analysis

- Exemplary sketch rendered in different input resolutions.

- Adaptive Canvas Selector provides a tradeoff between performance and computation.



Ablative Study

- Ablation on different objective functions.
- Ablation on different backbones.
- Ablation on different sketch encoders for canvas-size selection.

	Type	FLOPs (G)	Params (mil.)	ShoeV2 Top1 (%)	ShoeV2 Top10 (%)
∅	B-VGG16-SN	40.18	14.71	33.03	78.51
I	w/o rank^{-1}-reward	0.173	2.27	25.31	64.36
II	w/o $\mathcal{L}_{\mathrm{Tri}}^{R}$-reward	0.182		27.78	69.32
III	w/o $\mathcal{L}_{\mathrm{F}}$-reward	0.833		32.91	78.39
IV	ResNet18 backbone	1.451	11.18	26.72	68.18
V	EfficientNet backbone	0.459	4.01	29.47	72.04
VI	Image-based ψ_C	5.730	7.62	32.21	77.68
VII	Offset s_v	0.255	2.27	32.61	78.40
VIII	Decoder-LSTM	0.268	2.25	31.41	77.32
IX	Decoder-Tf	0.314	2.34	31.98	78.02
	SketchyNetV2	0.254	2.27	32.77	78.21

Ablative Studies

- Ablation on the performance of SOTA strategies with different input resolutions.

Canvas Size	Triplet-SN			Triplet-RL		
	Top1 (%)	Top10 (%)	FLOPs (G)	Top1 (%)	Top10 (%)	FLOPs (G)
32×32	10.91	26.65	0.083	12.97	31.08	0.142
64×64	19.04	48.12	0.338	22.96	56.23	0.577
128×128	26.07	63.87	1.397	30.04	72.95	2.299
256×256	28.74	71.68	5.280	34.21	77.24	6.041

- Ablation on the performance of different backbones as student-teacher pairs.

Teacher \ Student	MobileNetV2		EfficientNet		ResNet-18	
	Top-1	FLOPs	Top-1	FLOPs	Top-1	FLOPs
VGG-16	32.77%	0.254G	29.47%	0.459G	26.72%	1.451G
Inception-V3	27.89%	0.264G	27.16%	0.461G	25.31%	1.452G

Thank You

SketchX

http://sketchx.ai

For more such works, visit:
aneeshan95.github.io