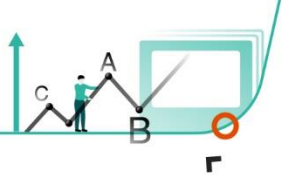
SUALAB

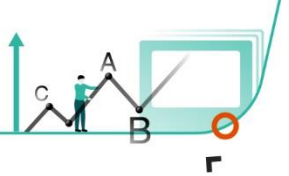
2019 CVPR paper overview

SUALAB

Ho Seong Lee

Contents





- What is CVPR?
  - Conference on Computer Vision and Pattern Recognition (CVPR)
  - CVPR was first held in 1983 and has been held annually
  - CVPR 2019: June 16th – June 20th in Long Beach, CA

Diamond Sponsors

Microsoft MEGVII 旷视

Platinum Sponsors

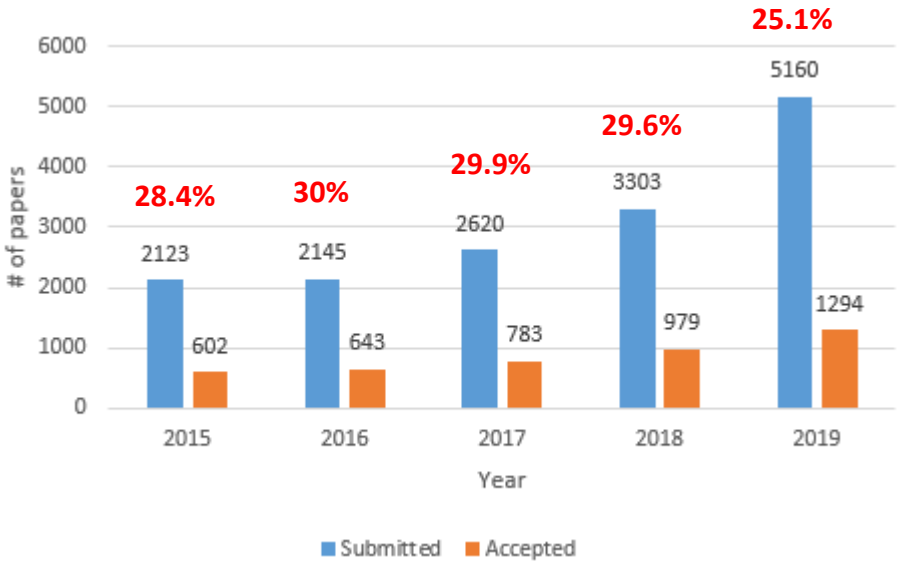
amazon anyVISION A BETTER TOMORROW ARGO Baidu 百度 ByteDance COGNEX

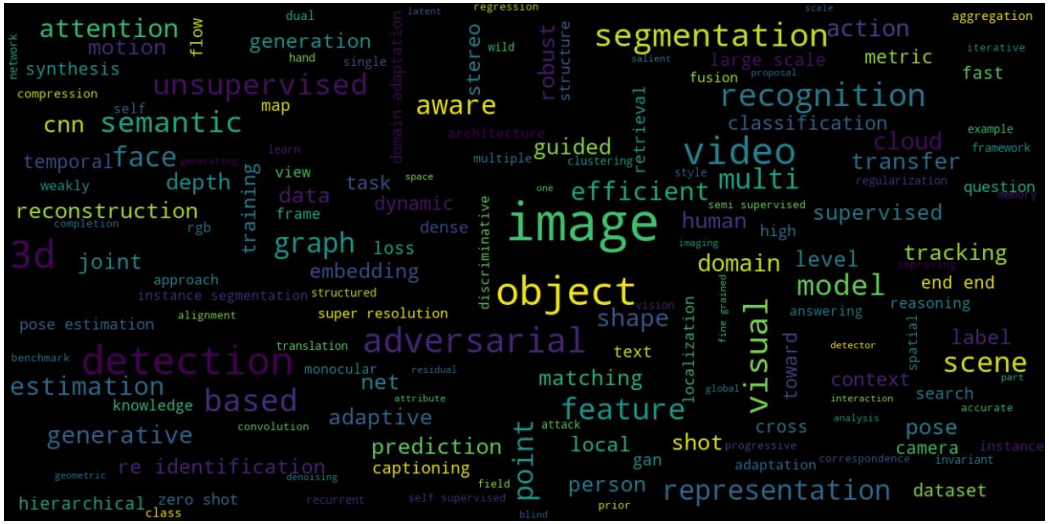
facebook Google HUAWEI IBM Research intel AI lyft

NAVER LINE NVIDIA pony.ai 快手 SAMSUNG 商汤 sensetime Tencent YouTu Lab

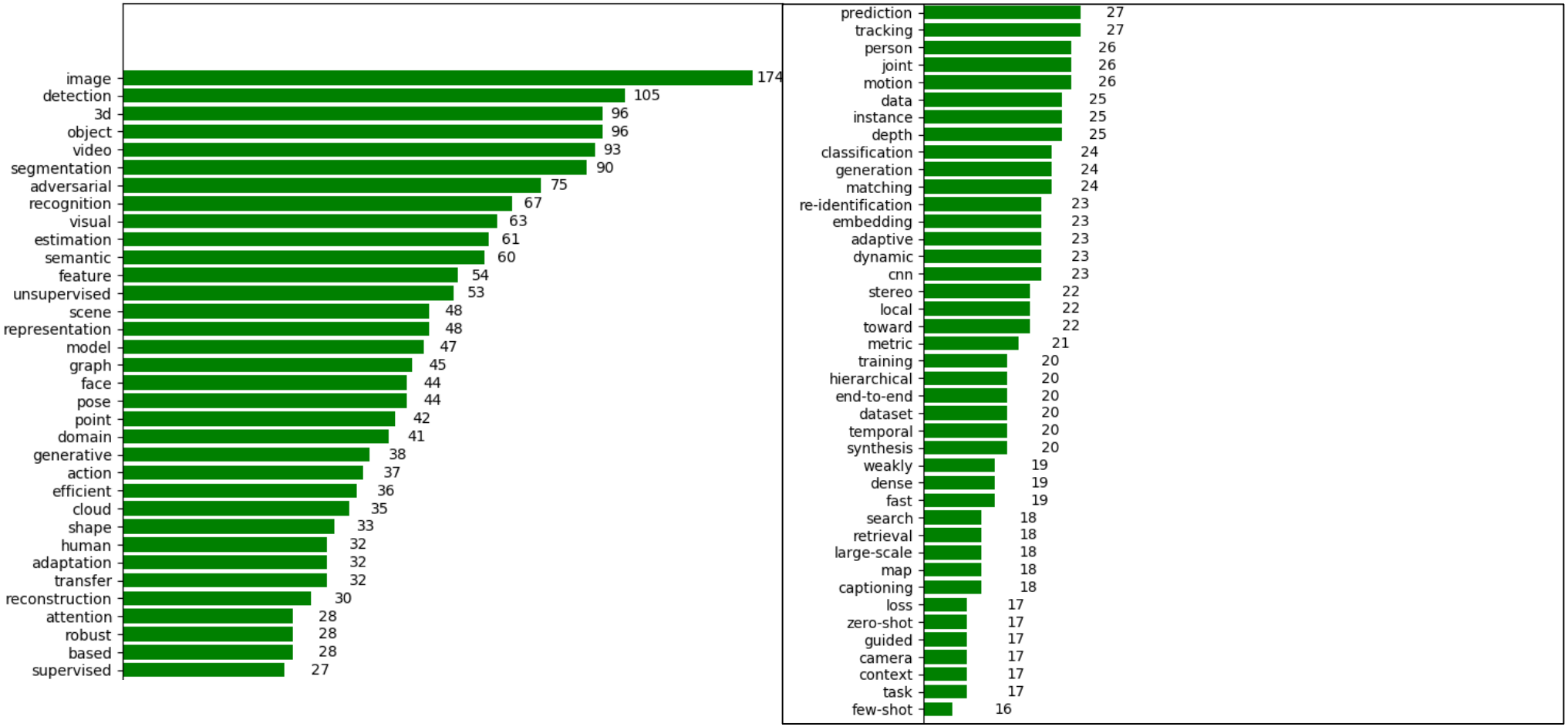
TOYOTA RESEARCH INSTITUTE Uber wayfair XILINX

- CVPR 2019 statistics
  - The total number of papers is increasing every year and this year has increased significantly!
  - We can visualize main topic using title of paper and simple python script!
    - https://github.com/hoya012/CVPR-Paper-Statistics



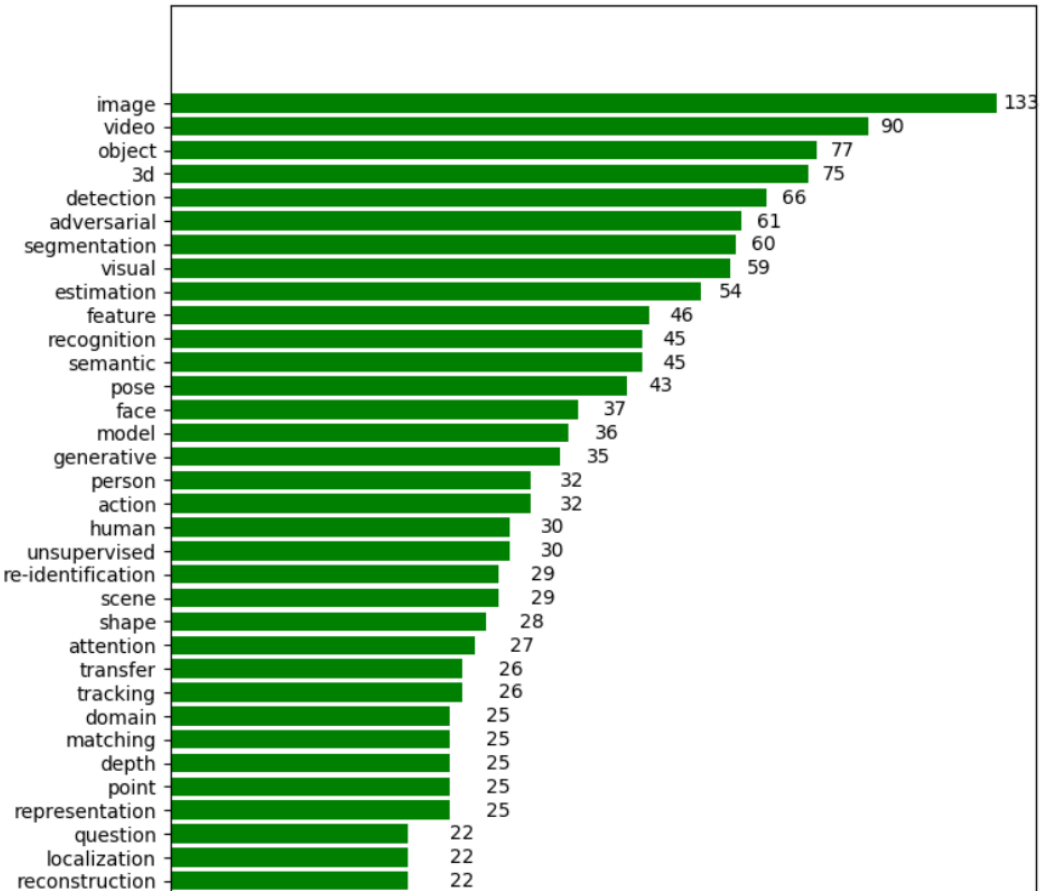


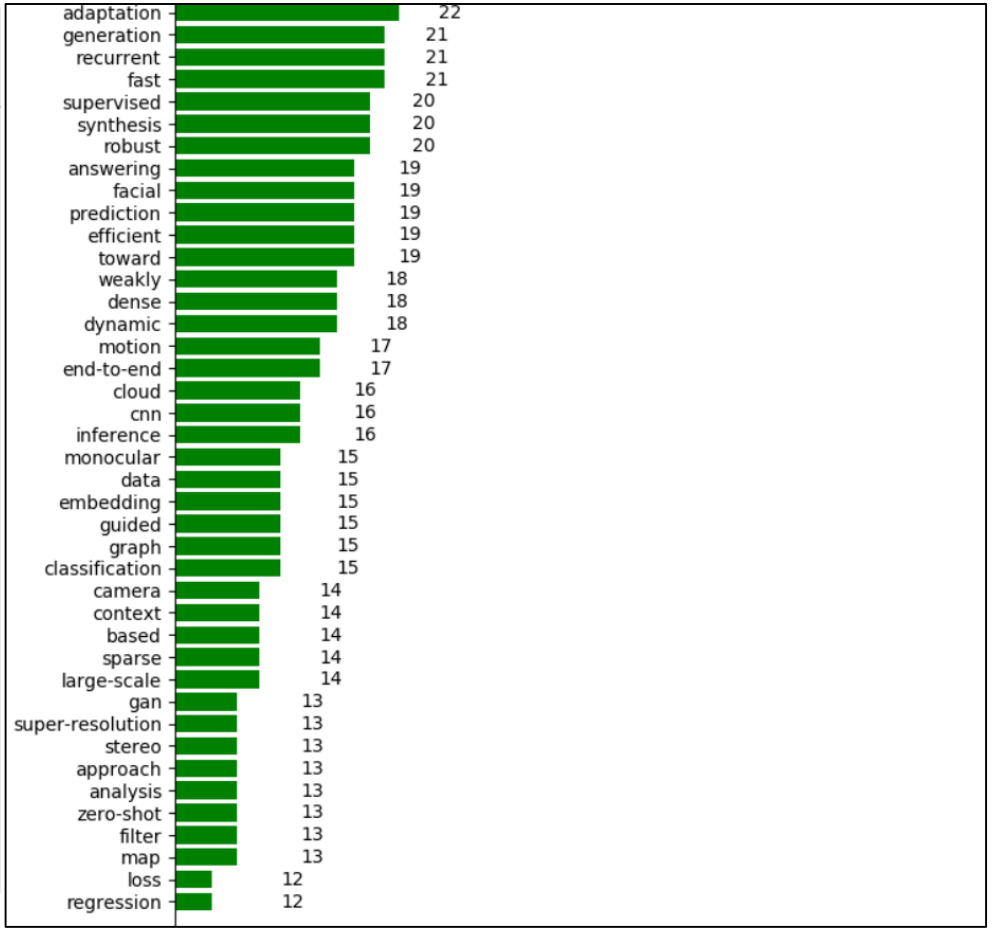
CVPR 2019 Statistics



2019 CVPR paper statistics

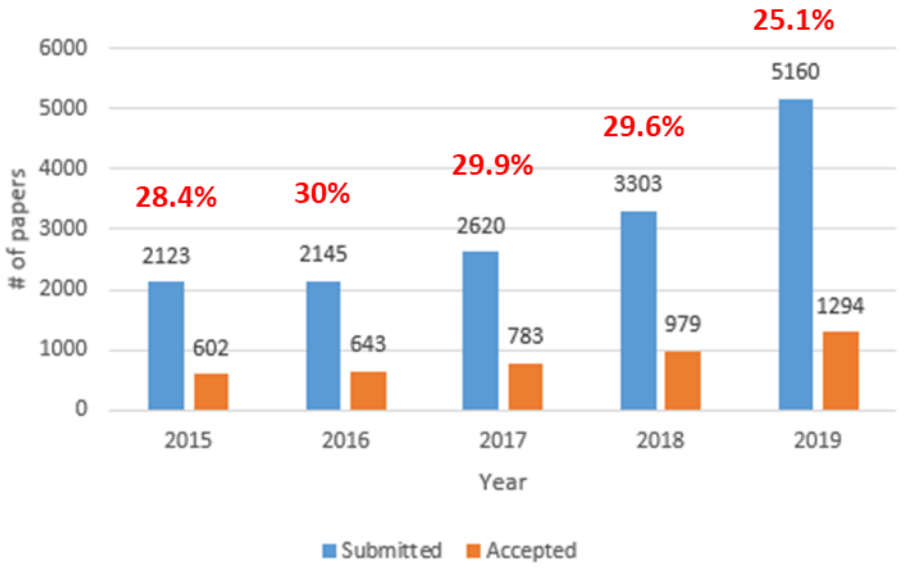
Compared to 2018 Statistics..

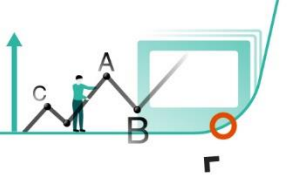




2018 CVPR paper statistics

- Most of the top keywords were maintained
  - Image, detection, 3d, object, video, segmentation, adversarial, recognition, visual …
- “graph”, “cloud”, “representation” are about twice as frequent
  - graph : 15 → 45
  - representation: 25 → 48
  - cloud: 16 → 35



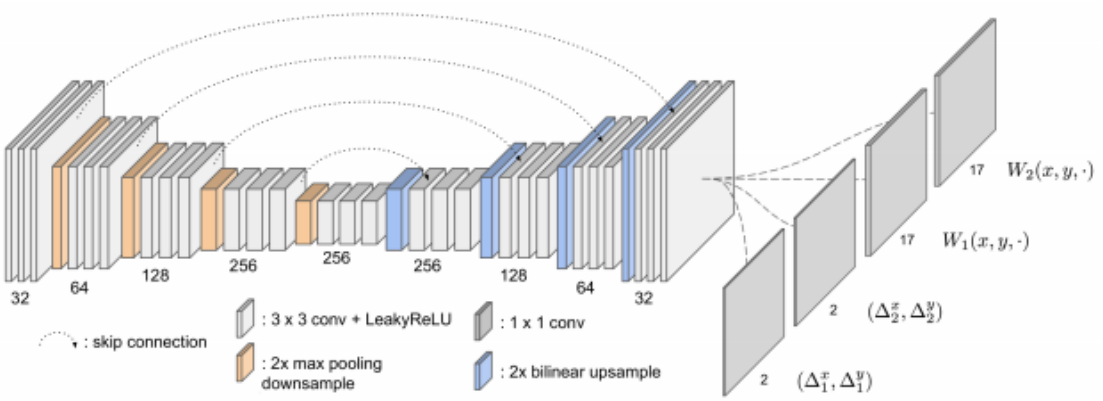


- **It does not mean that it is not an interesting article because it is not in the list.**
- Since I mainly studied Computer Vision, most papers that I will discuss today are Computer Vision papers..
- **Topics not covered today**
  - Natural Language Processing
  - Reinforcement Learning
  - Robotics
  - Etc..?

- Synthesizing a motion blurred image **from a pair of unblurred sequential images**
  - Motion blur is important in cinematography, and artful photo
  - Generate a large-scale synthetic training dataset of motion blurred images

(a) A pair of input images.

(b) Our model's output.

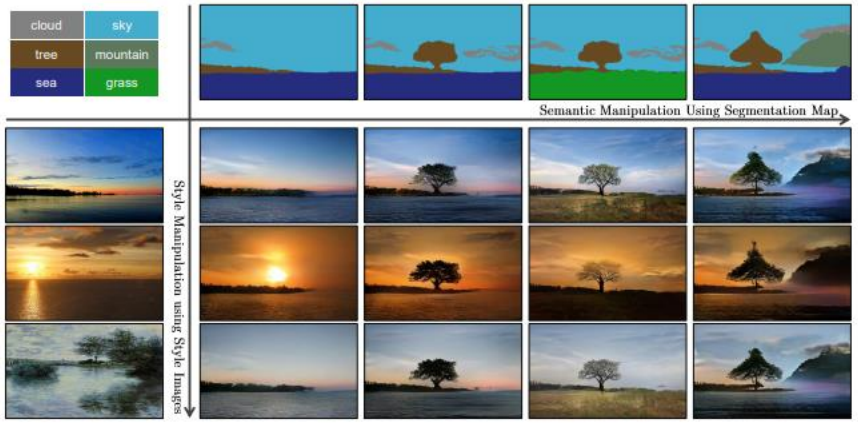


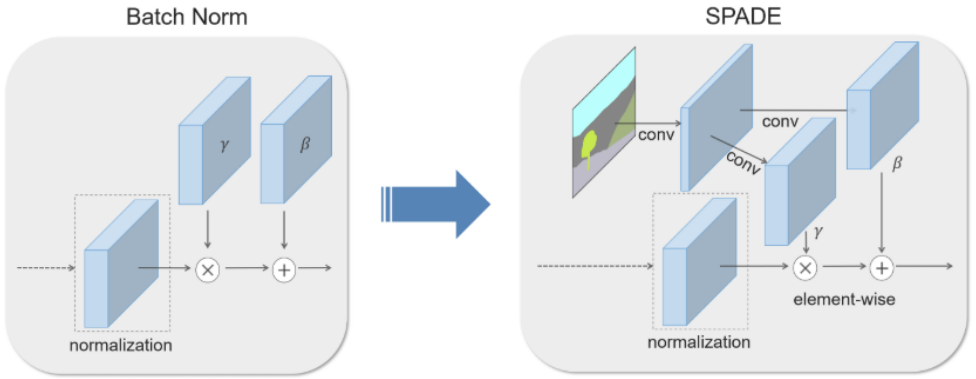
Algorithm	PSNR	SSIM	Runtime (ms)
Naive Baseline	28.06 ± 4.05	0.888 ± 0.087	-
PWC-Net [29]	29.93 ± 3.47	0.938 ± 0.057	39.5
EpicFlow [27]	30.07 ± 3.49	0.940 ± 0.057	96.3×10^6
SepConv [26]	32.91 ± 4.60	0.954 ± 0.054	10.9×10^4
Super SloMo [14]	33.64 ± 4.66	0.958 ± 0.048	13.7×10^6
Ours (direct pred.)	33.97 ± 4.53	0.961 ± 0.044	34.7
Ours (uniform weight)	33.88 ± 4.68	0.959 ± 0.050	42.8
Ours (kernel pred.)	33.73 ± 4.31	0.961 ± 0.045	65.5
Our Model	34.14 ± 4.65	0.963 ± 0.045	43.7

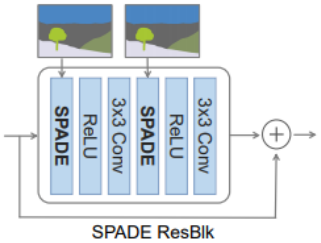
Table 1. Performance on our real test dataset, in which we compare our model to three of its ablated variants and five baseline algorithms.

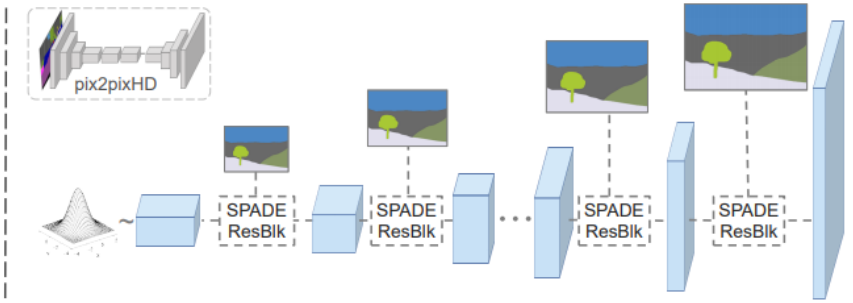
Recommended reference: "Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation", 2018 CVPR

- Synthesizing photorealistic images given an input semantic layout
  - Spatially-adaptive normalization can keep semantic information
  - This model allows user control over both semantic and style as synthesizing images









Demo Code: https://github.com/NVlabs/SPADE

- Reconstruct a complete and textured 3D model of a person from a single image
  - Use 2D Silhouettes and 3D joints of a body pose to reconstruct 3D mesh
  - An effective two-stage 3D shape reconstruction pipeline
    - Predicting multi-view 2D silhouettes from single input segmentation
    - Deep visual hull based mesh reconstruction technique

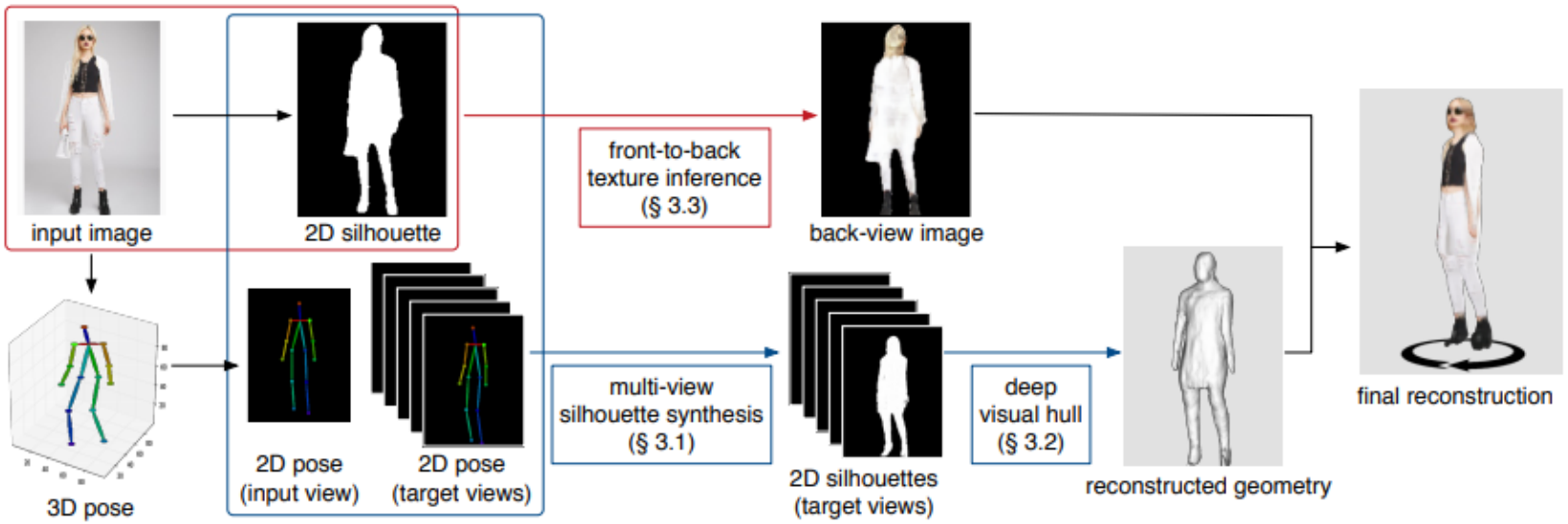


Figure 2: Overview of our framework.

Recommended reference: "BodyNet: Volumetric Inference of 3D Human Body Shapes", 2018 ECCV

- Propose **controllable photo-to-pencil translation** method
  - Modeling pencil outline(rough, clean), pencil shading(4 types)
  - Create training data pairs from online websites(e.g., Pinterest) and use image filtering techniques

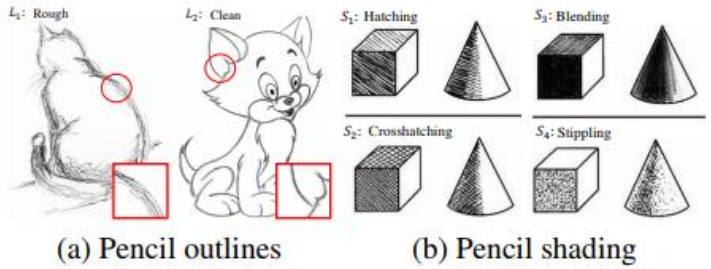
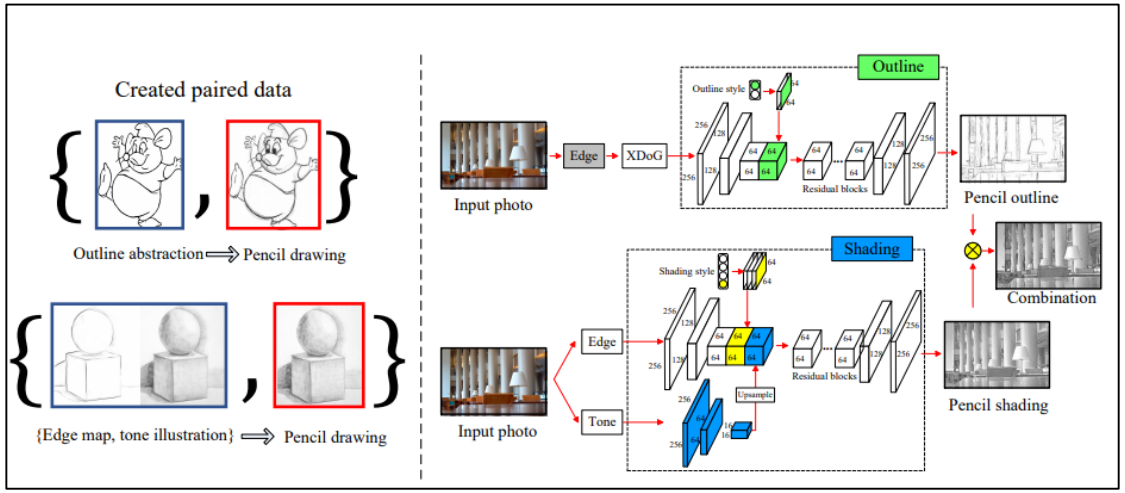


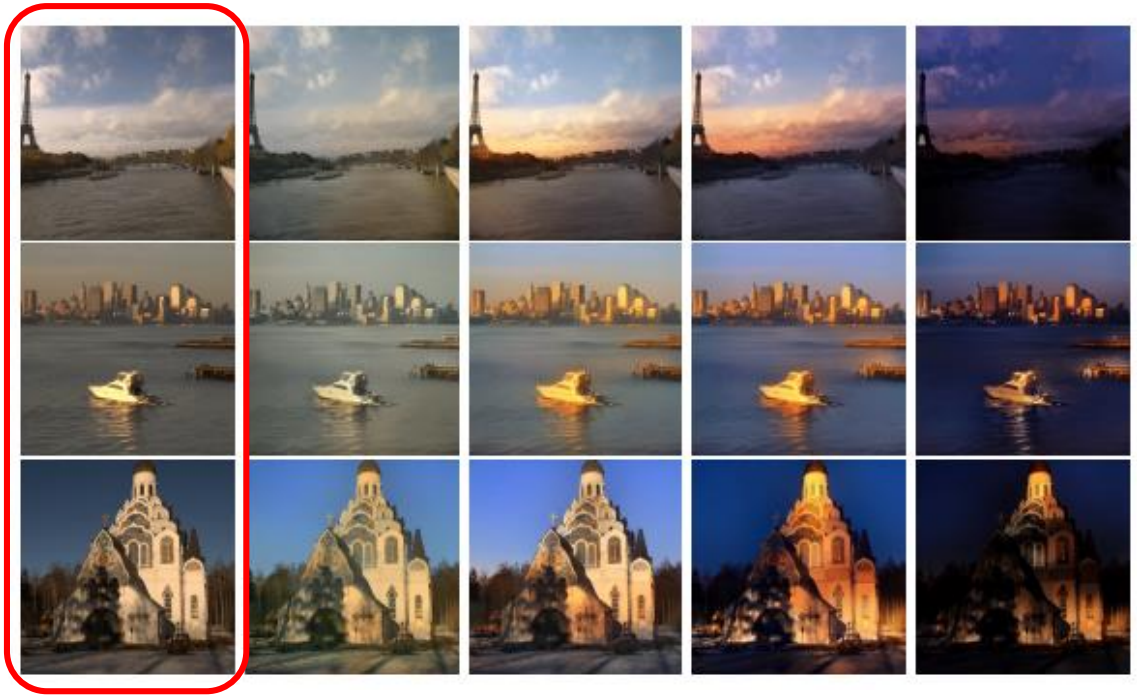
Figure 1. Examples of real pencil drawings in the outline ($L_1 \sim L_2$) and shading ($S_1 \sim S_4$) styles that we train on.



Demo Code: https://github.com/Yijunmaverick/Im2Pencil

- End-to-end solution to **synthesize a time-lapse video from single image**
  - Use time-lapse videos and image sequences during training
  - Use only single image during inference

Input image(single)



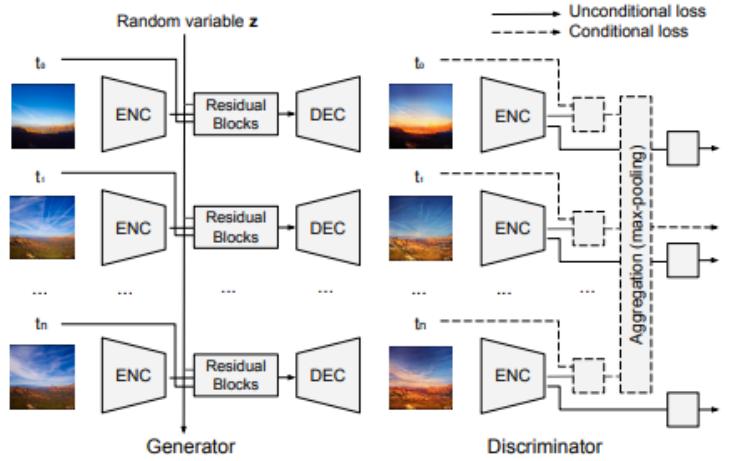
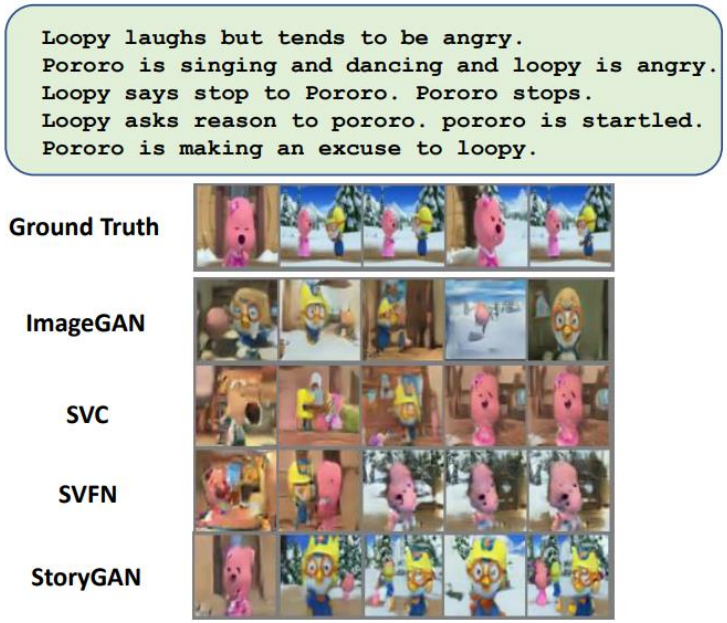
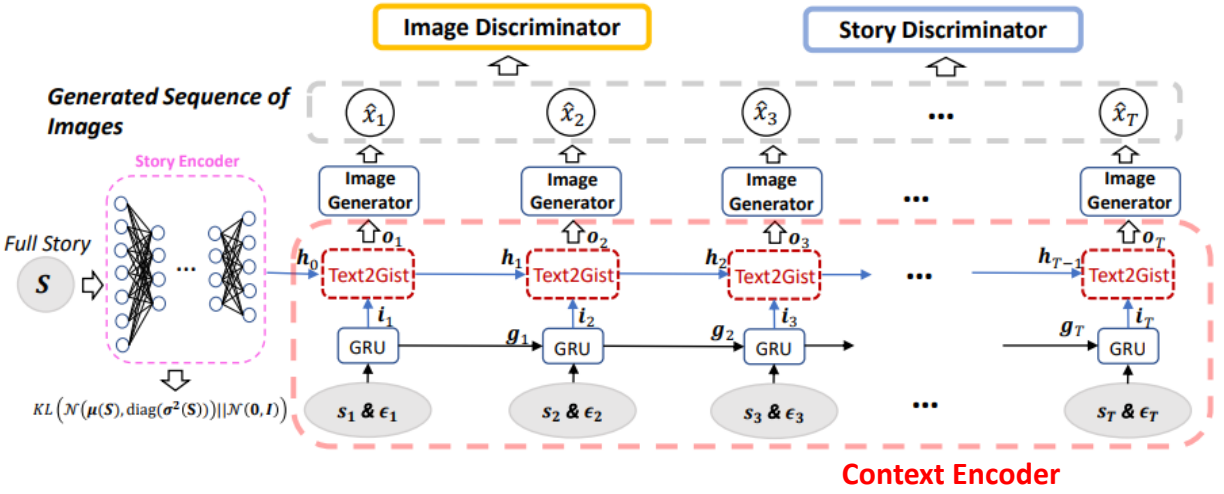


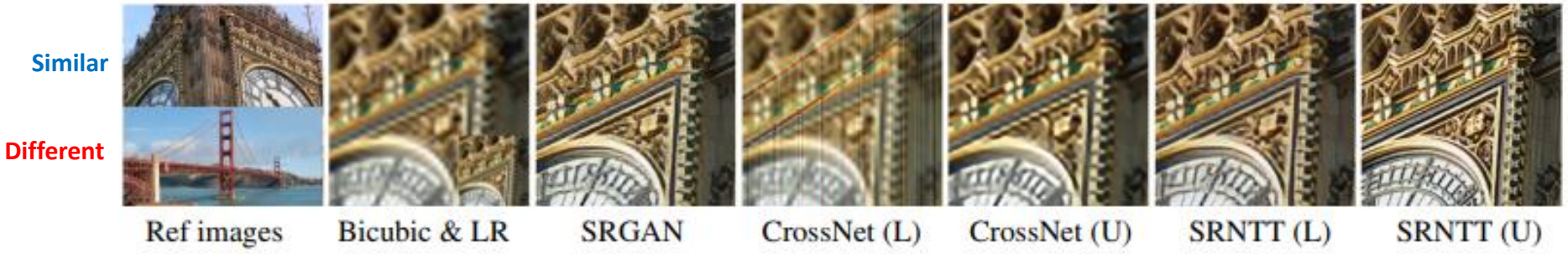
Figure 3: Illustration of our multi-frame joint conditional GAN method. For our discriminator, the encoded images are concatenated with the timestamps (dashed rectangles) before being aggregated to compute the conditional loss, while each image is directly used as an input to compute the unconditional loss (solid rectangles).

- Propose a new task called Story Visualization using GAN
  - Sequential conditional GAN based StoryGAN
  - Story Encoder – stochastic mapping from story to an low-dimensional embedding vector
  - Context Encoder – capture contextual information during sequential image generation
  - Two Discriminator – Image Discriminator & Story Discriminator



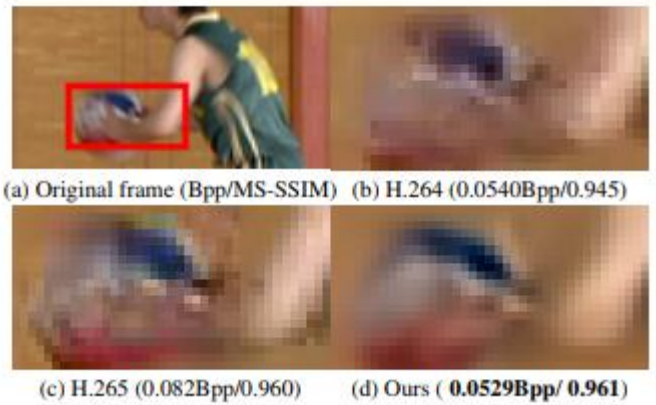


- Improve "*RefSR*" even **when irrelevant reference images are provided**
  - Traditional Single Image Super-Resolution is extremely challenging (ill-posed problem)
  - Reference-based(*RefSR*) utilizes rich texture from HR references .. but.. only similar Ref images
  - Adaptively transferring the texture from Ref Images according to their texture similarity



Recommended reference: "CrossNet: An end-to-end reference-based super resolution network using cross-scale warping", 2018 ECCV

- Propose the first **end-to-end video compression** deep model
  - Conventional video compression use predictive coding architecture and encode corresponding **motion information** and **residual information**
  - Taking advantage of both classical compression and neural network
  - Use learning based optical flow estimation



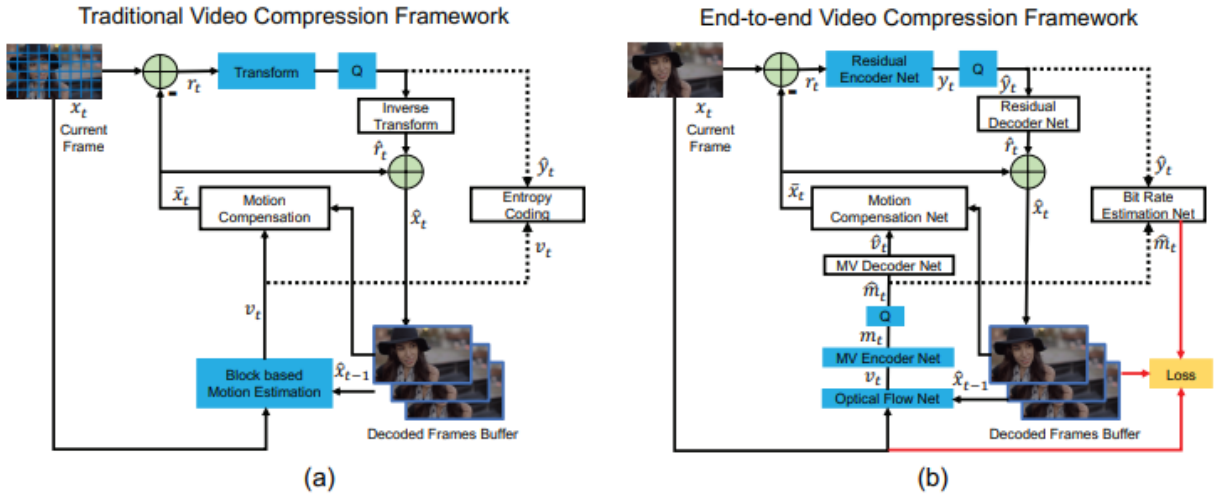
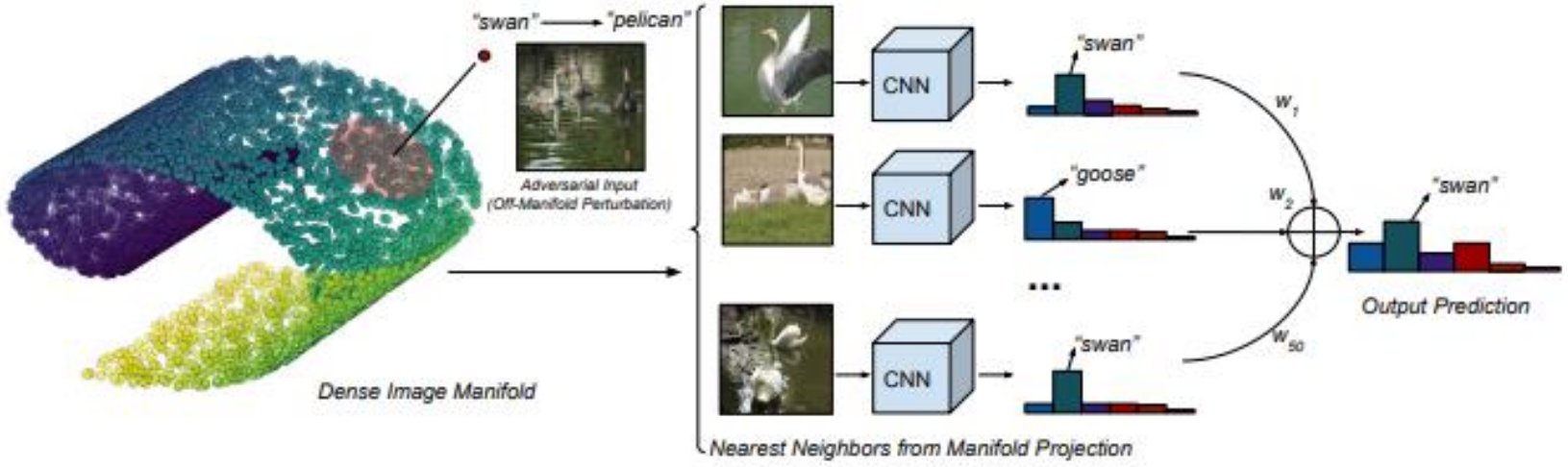
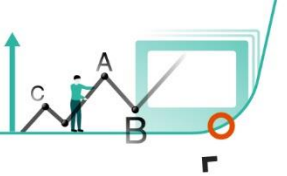


Figure 2: (a): The predictive coding architecture used by the traditional video codec H.264 [36] or H.265 [29]. (b): The proposed end-to-end video compression network. The modules with blue color are not included in the decoder side.

- Defense adversarial attack using **Big-data and image manifold**
  - Assume that adversarial attack move the image away from the image manifold
  - A successful defense mechanism should aim to project the images back on the image manifold
  - For tens of billions of images, search a nearest-neighbor images (***K=50***) and use them
  - Also propose two novel attack methods to break nearest neighbor defenses





- Examine a collection of some refinements and empirically evaluate their impact
  - Improve ResNet-50's accuracy from 75.3% to 79.29% on ImageNet with some refinements
    - **Efficient Training**
      - FP32 with BS=256 → FP16 with BS=1024 with some techniques

        Linear scaling LR
        LR warmup
        Zero γ initialization in BN
        No bias decay
    - **Training Refinements**:
      - Cosine Learning Rate Decay / Label Smoothing / Knowledge Distillation / Mixup Training
  - Transfer from classification to Object Detection, Semantic Segmentation

Model	Efficient			Baseline		
	Time/epoch	Top-1	Top-5	Time/epoch	Top-1	Top-5
ResNet-50	**4.4 min**	**76.21**	**92.97**	13.3 min	75.87	92.70
Inception-V3	**8 min**	**77.50**	**93.60**	19.8 min	77.32	93.43
MobileNet	**3.7 min**	**71.90**	**90.47**	6.2 min	69.03	88.71

Heuristic	BS=256		BS=1024	
	Top-1	Top-5	Top-1	Top-5
Linear scaling	75.87	92.70	75.17	92.54
+ LR warmup	76.03	92.81	75.93	92.84
+ Zero γ	76.19	93.03	76.37	92.96
+ No bias decay	76.16	92.97	76.03	92.86
+ FP16	76.15	93.09	76.21	92.97

Result of Efficient Training

Model	#params	FLOPs	Top-1	Top-5
ResNet-50	25 M	**3.8 G**	76.21	92.97
ResNet-50-B	25 M	4.1 G	76.66	93.28
ResNet-50-C	25 M	4.3 G	76.87	93.48
ResNet-50-D	25 M	4.3 G	**77.16**	**93.52**

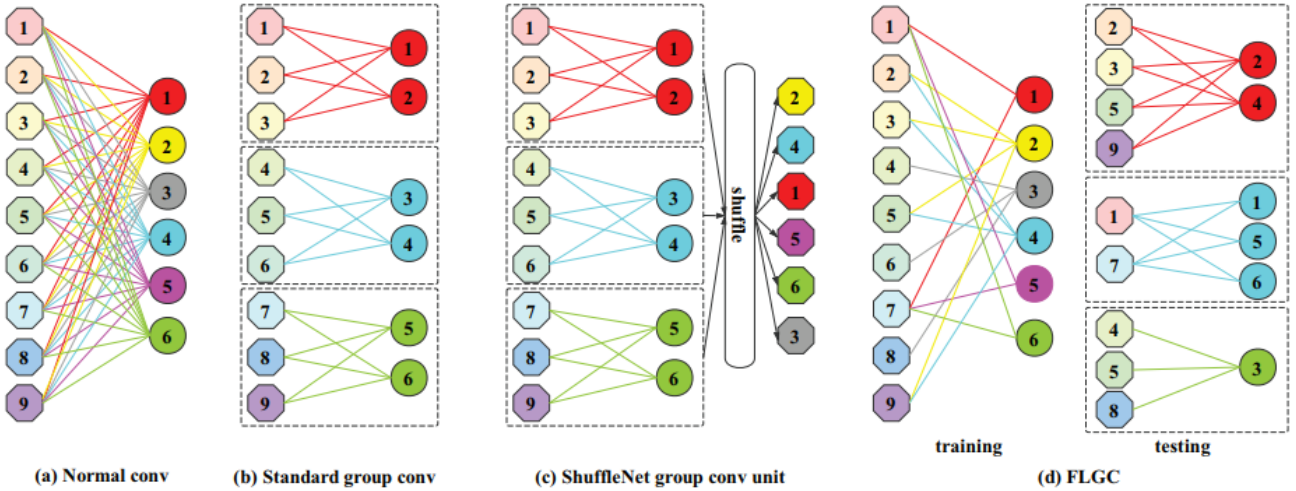
Table 5: Compare ResNet-50 with three model tweaks on model size, FLOPs and ImageNet validation accuracy.

ResNet tweaks

Refinements	ResNet-50-D		Inception-V3		MobileNet	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Efficient	77.16	93.52	77.50	93.60	71.90	90.53
+ cosine decay	77.91	93.81	78.19	94.06	72.83	91.00
+ label smoothing	78.31	94.09	78.40	94.13	72.93	91.14
+ distill w/o mixup	78.67	94.36	78.26	94.01	71.97	90.89
+ mixup w/o distill	79.15	94.58	**78.77**	**94.39**	**73.28**	**91.30**
+ distill w/ mixup	**79.29**	**94.63**	78.34	94.16	72.51	91.02

Result of Training refinements

- **Automatically learn the group structure** in training stage with end-to-end manner
  - Outperform standard group convolution
  - Propose an efficient strategy for index re-ordering



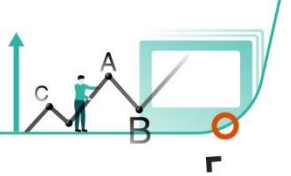
(a) Normal conv (b) Standard group conv (c) ShuffleNet group conv unit (d) FLGC

Table 2. Image classification error(%) and time complexity of different methods on CIFAR-10.(G:group number)

Model	MAdds	Params	Err
ResNet56-pruned [14]	62M	—	8.2
ResNet50-FLGC1(ours)	**23M**	**0.22M**	**7.95**
ResNet56-pruned [25]	90M	0.73M	6.94
ResNet50-FLGC2(ours)	**44M**	**0.68M**	**6.77**
MobileNetV2-SGC(G=2)	158M	1.18M	6.04
MobileNetV2-FLGC(G=2)	158M	1.18M	**5.89**
MobileNetV2-FLGC(G=3)	122M	0.85M	**5.80**
MobileNetV2-SGC(G=4)	103M	0.68M	6.64
MobileNetV2-FLGC(G=4)	103M	0.68M	**5.84**
MobileNetV2-FLGC(G=5)	92M	0.58M	**6.12**
MobileNetV2-FLGC(G=6)	85M	0.51M	**6.33**
MobileNetV2-FLGC(G=7)	80M	0.46M	**6.34**
MobileNetV2-SGC(G=8)	76M	0.43M	7.51
MobileNetV2-FLGC(G=8)	76M	0.43M	**6.91**

Table 3. Comparison of Top-1 and Top-5 classification error rate (%) with other state-of-the-art compact models on ImageNet.

Model	MAdds	Params	Top1	Top5
Inception V1[36]	1448M	6.6M	30.2	10.1
1.0 MobileNet-224[16]	569M	4.2M	29.4	10.5
ShuffleNet 2x[46]	524M	5.3M	26.3	—
NASNet-A (N=4)[48]	564M	5.3M	26.0	8.4
NASNet-B (N=4)[48]	488M	5.3M	27.2	8.7
NASNet-C (N=4)[48]	558M	4.9M	27.5	9.0
CondenseNet (G=4)[17]	529M	4.8M	26.2	8.3
CondenseNet-SGC	529M	4.8M	29.0	9.9
CondenseNet-FLGC	**529M**	**4.8M**	**25.3**	**7.9**



- Explore to train object detectors from scratch robustly
  - Almost SOTA detectors are fine-tuned from pretrained CNN (e.g., ImageNet)
    - The classification and detection have different degrees of sensitivity to translation
    - The architecture is limited by the classification network(backbone) → inconvenience!
  - Find that one of the overlooked points is BatchNorm!

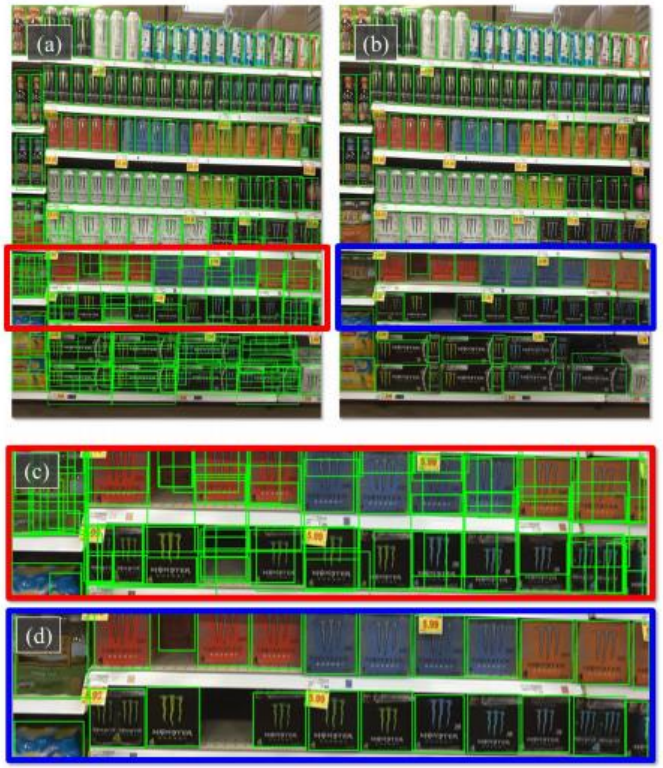
Table 3. Detection results on the PASCAL VOC datasets. For VOC 2007, all methods are trained on the VOC 2007 and 2012 `trainval` sets and tested on the VOC 2007 `test` set. For VOC 2012, all methods are trained on the VOC 2007 and 2012 `trainval` sets plus the VOC 2007 `test` set, and tested on the VOC 2012 `test` set. †: `http://host.robots.ox.ac.uk:8080/anonymous/0HPCHC.html` ‡: `http://host.robots.ox.ac.uk:8080/anonymous/JSL6ZY.html`

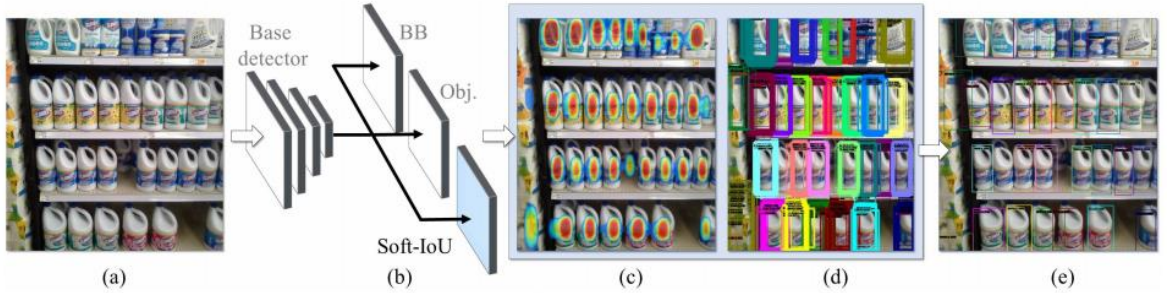
Method	Backbone	Input size	FPS	mAP (%)	
				VOC 2007	VOC 2012
pretrained two-stage:					
HyperNet [19]	VGG-16	~ 1000 × 600	0.88	76.3	71.4
Faster R-CNN[28]	ResNet-101	~ 1000 × 600	2.4	76.4	73.8
ION[1]	VGG-16	~ 1000 × 600	1.25	76.5	76.4
MR-CNN[9]	VGG-16	~ 1000 × 600	0.03	78.2	73.9
R-FCN[4]	ResNet-101	~ 1000 × 600	9	80.5	77.6
CoupleNet[42]	ResNet-101	~ 1000 × 600	8.2	82.7	80.4
pretrained one-stage:					
RON384[18]	VGG-16	384 × 384	15	74.2	71.7
SSD321[8]	ResNet-101	321 × 321	11.2	77.1	75.4
SSD300*[24]	VGG16	300 × 300	46	77.2	75.8
YOLOv2[27]	Darknet-19	544 × 544	40	78.6	73.4
DSSD321[8]	ResNet-101	321 × 321	9.5	78.6	76.3
DES300[41]	VGG-16	300 × 300	29.9	79.7	77.1
RefineDet320[39]	VGG-16	320 × 320	40.3	80.0	78.1
trained from scratch:					
DSOD300[32]	DS/64-192-48-1	300 × 300	17.4	77.7	76.3
GRP-DSOD320[33]	DS/64-192-48-1	300 × 300	16.7	78.7	77.0
ScratchDet300	Root-ResNet-34	300 × 300	17.8	80.4	78.5†
ScratchDet300+	Root-ResNet-34	-	-	**84.1**	**83.6**‡

Recommended reference: "DSOD: Learning Deeply Supervised Object Detectors from Scratch", 2017 ICCV

- Propose precise detection in **densely packed scenes**
  - In real-world, there are many applications of object detection (ex, detection and count # of object)
  - In densely packed scenes, SOTA detector can't detect accurately

(1) layer for estimating the Jaccard index (2) a novel EM merging unit (3) release SKU-110K dataset

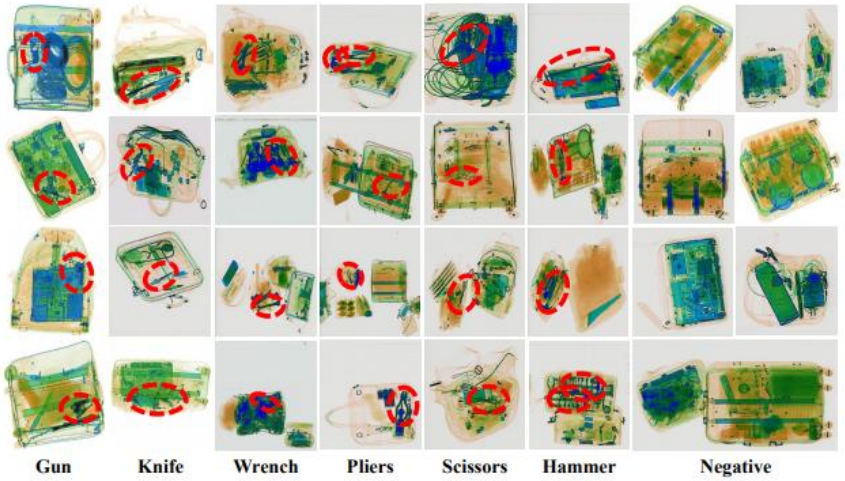


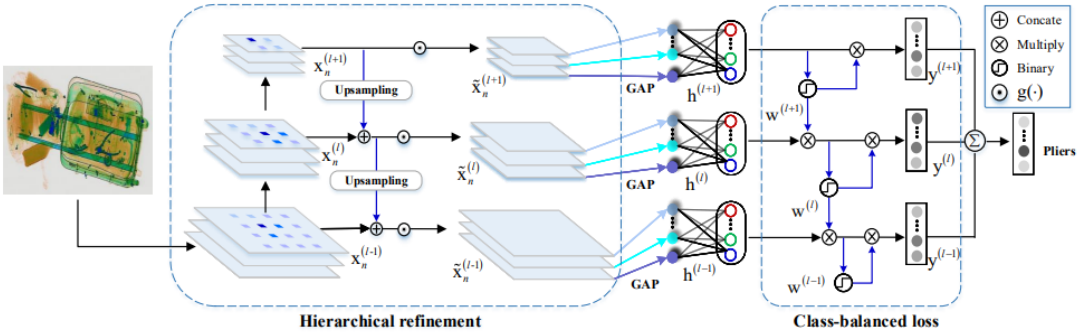


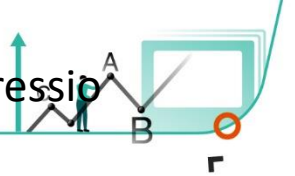
Name	#Img.	#Obj./img.	#Cls.	#Cls./img.	Dense.	Idnt.	BB
UCSD (2008) [8]	2000	24.9	1	1	✓	✗	✗
PACAL VOC (2012) [13]	22,531	2.71	20	2	✗	✗	✓
ILSVRC Detection (2014) [12]	516,840	1.12	200	2	✗	✗	✓
COCO (2015) [28]	328,000	7.7	91	3.5	✗	✗	✓
Penguins (2016) [2]	82,000	25	1	1	✓	✗	✗
TRANCOS (2016) [34]	1,244	37.61	1	1	✓	✓	✗
WIDER FACE (2016) [49]	32,203	12	1	1	✗	✗	✓
CityPersons (2017) [51]	5000	6	1	1	✗	✗	✓
PUCPR+ (2017) [22]	125	135	1	1	✓	✓	✓
CARPK (2018) [22]	1448	61	1	1	✓	✓	✓
Open Images V4 (2018) [25]	**1,910,098**	8.4	600	2.3	✗	✓	✓
Our SKU-110K	11,762	**147.4**	**110,712**	**86**	✓	✓	✓

- Present a large-scale dataset and establish a baseline for **security inspection X-ray**
  - Total 1,059,231 X-ray images in which 6 classes of 8,929 prohibited items
  - Propose an approach named class-balanced hierarchical refinement(CHR) and class-balanced loss function

The SIXray Dataset (1,059,231)						
Positive (8,929)						Negative
Gun	Knife	Wrench	Pliers	Scissors	Hammer	
3,131	1,943	2,199	3,961	983	60	1,050,302







- Address the weaknesses of IoU and **introduce generalized version(GIoU)**
  - Intersection over Union(IoU) is the most popular evaluation metric used in object detection
  - But, there is a gap between optimizing distance losses and maximizing IoU
  - Introducing generalized IoU as both a new loss and a new metric

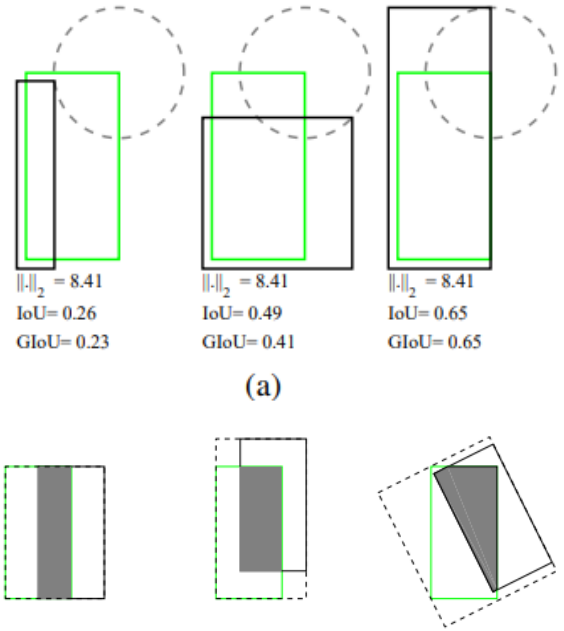


Figure 2. Three different ways of overlap between two rectangles with the exactly same IoU values, *i.e.* $IoU = 0.33$, but different $GIoU$ values, *i.e.* from the left to right $GIoU = 0.33, 0.24$ and -0.1 respectively. $GIoU$ value will be higher for the cases with better aligned orientation.

Algorithm 1: Generalized Intersection over Union

input : Two arbitrary convex shapes: $A, B \subseteq \mathbb{S} \in \mathbb{R}^n$

output: $GIoU$

1 For A and B, find the smallest enclosing convex object C, where $C \subseteq \mathbb{S} \in \mathbb{R}^n$

2 $IoU = \dfrac{|A \cap B|}{|A \cup B|}$

3 $GIoU = IoU - \dfrac{|C \backslash (A \cup B)|}{|C|}$

Algorithm 2: IoU and $GIoU$ as bounding box losses

input : Predicted B^p and ground truth B^g bounding box coordinates:
$B^p = (x_1^p, y_1^p, x_2^p, y_2^p), \quad B^g = (x_1^g, y_1^g, x_2^g, y_2^g)$.

output: $\mathcal{L}_{IoU}, \mathcal{L}_{GIoU}$.

1 For the predicted box B^p, ensuring $x_2^p > x_1^p$ and $y_2^p > y_1^p$:
$\hat{x}_1^p = \min(x_1^p, x_2^p), \quad \hat{x}_2^p = \max(x_1^p, x_2^p),$
$\hat{y}_1^p = \min(y_1^p, y_2^p), \quad \hat{y}_2^p = \max(y_1^p, y_2^p).$

2 Calculating area of B^g: $A^g = (x_2^g - x_1^g) \times (y_2^g - y_1^g)$.

3 Calculating area of B^p: $A^p = (\hat{x}_2^p - \hat{x}_1^p) \times (\hat{y}_2^p - \hat{y}_1^p)$.

4 Calculating intersection $\mathcal{I}$ between B^p and B^g:
$x_1^{\mathcal{I}} = \max(\hat{x}_1^p, x_1^g), \quad x_2^{\mathcal{I}} = \min(\hat{x}_2^p, x_2^g),$
$y_1^{\mathcal{I}} = \max(\hat{y}_1^p, y_1^g), \quad y_2^{\mathcal{I}} = \min(\hat{y}_2^p, y_2^g),$
$\mathcal{I} = \begin{cases} (x_2^{\mathcal{I}} - x_1^{\mathcal{I}}) \times (y_2^{\mathcal{I}} - y_1^{\mathcal{I}}) & \text{if } x_2^{\mathcal{I}} > x_1^{\mathcal{I}}, y_2^{\mathcal{I}} > y_1^{\mathcal{I}} \\ 0 & \text{otherwise.} \end{cases}$

5 Finding the coordinate of smallest enclosing box B^c:
$x_1^c = \min(\hat{x}_1^p, x_1^g), \quad x_2^c = \max(\hat{x}_2^p, x_2^g),$
$y_1^c = \min(\hat{y}_1^p, y_1^g), \quad y_2^c = \max(\hat{y}_2^p, y_2^g).$

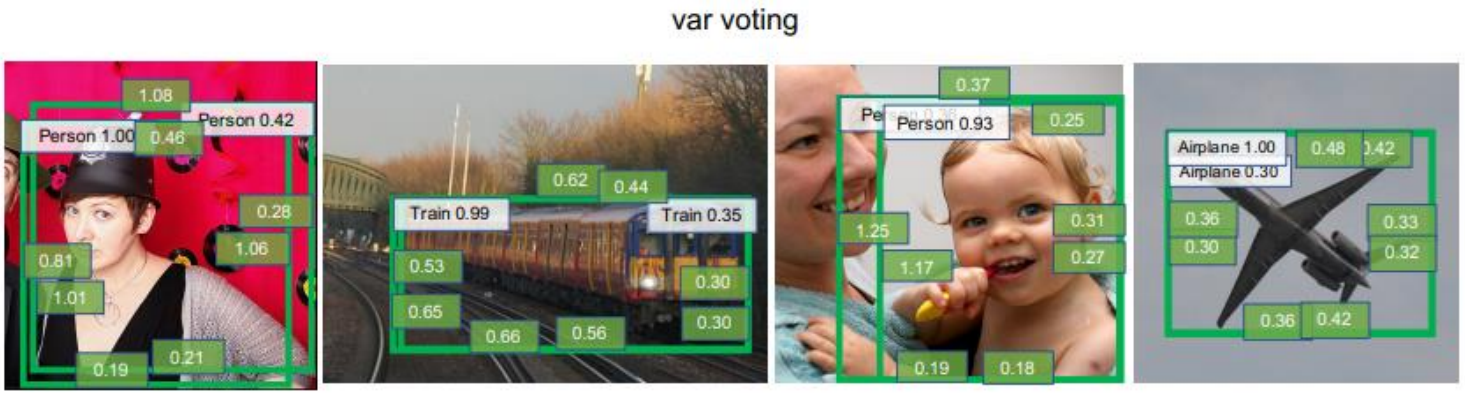
6 Calculating area of B^c: $A^c = (x_2^c - x_1^c) \times (y_2^c - y_1^c)$.

7 $IoU = \dfrac{\mathcal{I}}{\mathcal{U}}$, where $\mathcal{U} = A^p + A^g - \mathcal{I}$.

8 $GIoU = IoU - \dfrac{A^c - \mathcal{U}}{A^c}$.

9 $\mathcal{L}_{IoU} = 1 - IoU, \quad \mathcal{L}_{GIoU} = 1 - GIoU.$

- Propose novel bounding box regression loss with uncertainty
  - Most of datasets have ambiguities and labeling noise of bounding box coordinate
  - Network can learns to predict localization variance for each coordinate



	AP	AP^{50}	AP^{60}	AP^{70}	AP^{80}	AP^{90}
baseline [14]	38.6	**59.8**	55.3	47.7	34.4	11.3
MR-CNN [11]	38.9	**59.8**	55.5	48.1	$34.8^{+0.4}$	$11.9^{+0.6}$
soft-NMS [1]	39.3	59.7	**55.6**	**48.9**	$35.9^{+1.5}$	$12.0^{+0.7}$
IoU-NMS+Refine [27]	39.2	57.9	53.6	47.4	$36.5^{+2.1}$	$16.4^{+5.1}$
KL Loss	$39.5^{+0.9}$	58.9	54.4	47.6	$36.0^{+1.6}$	$15.8^{+4.5}$
KL Loss+var voting	$39.9^{+1.3}$	58.9	54.4	47.7	$36.4^{+2.0}$	$17.0^{+5.7}$
KL Loss+var voting+soft-NMS	$\mathbf{40.4}^{+1.8}$	58.7	54.6	48.5	$\mathbf{37.5}^{+3.3}$	$\mathbf{17.5}^{+6.2}$

Table 4: Comparisons of different methods for accurate object detection on MS-COCO. The baseline model is ResNet-50-FPN Mask R-CNN. We improve the baseline by ≈ 2% in AP

- Propose a **unified panoptic segmentation network**(UPSNet)
  - Semantic segmentation + Instance segmentation = panoptic segmentation
  - Semantic Head + Instance Head + Panoptic head → end-to-end manner

    Deformable Conv **Mask R-CNN** **Parameter-free**

Countable objects → things
Uncountable objects → stuff

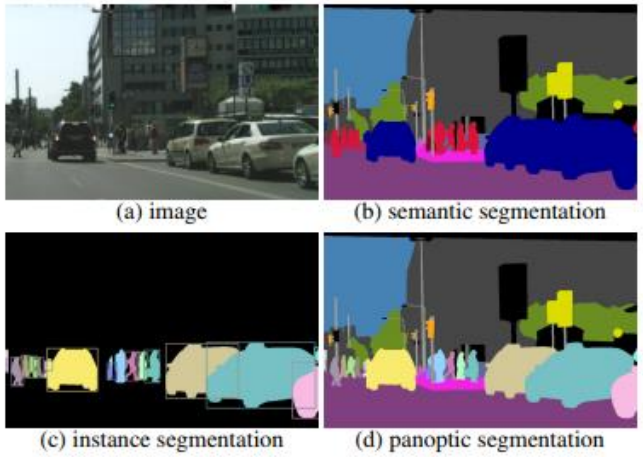


Figure 1: For a given (a) image, we show *ground truth* for: (b) semantic segmentation (per-pixel class labels), (c) instance segmentation (per-object mask and class label), and (d) the proposed *panoptic segmentation* task (per-pixel class+instance labels). The PS task: (1) encompasses both stuff and thing classes, (2) uses a simple but general format, and (3) introduces a uniform evaluation metric for all classes. Panoptic segmentation generalizes both semantic and instance segmentation and we expect the unified task will present novel challenges and enable innovative new methods.

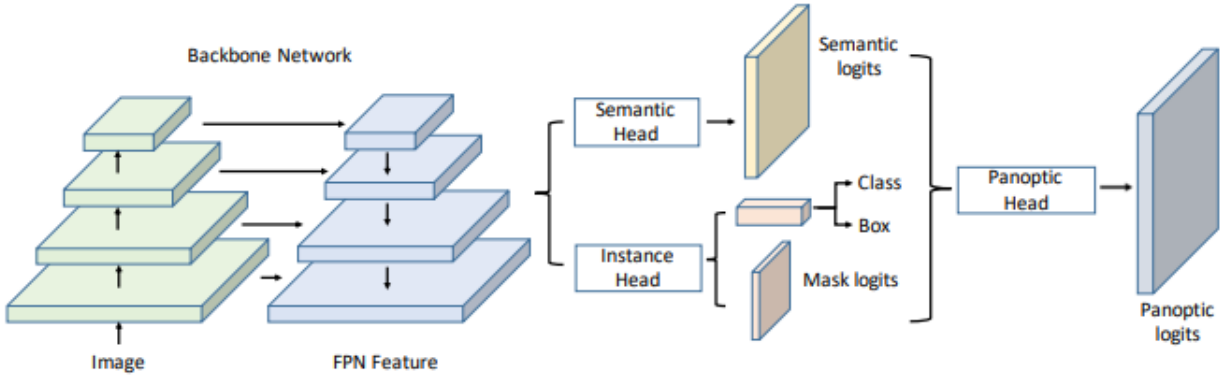


Figure 1: Overall architecture of our UPSNet.

Recommended reference: "Panoptic Segmentation", 2018 arXiv

- Propose SFNet for semantic correspondence problem
  - Propose to use images annotated with binary foreground masks and synthetic geometric deformations during training
    - Manually selecting point correspondences is so expensive!!
  - Outperform SOTA on standard benchmarks by a significant margin

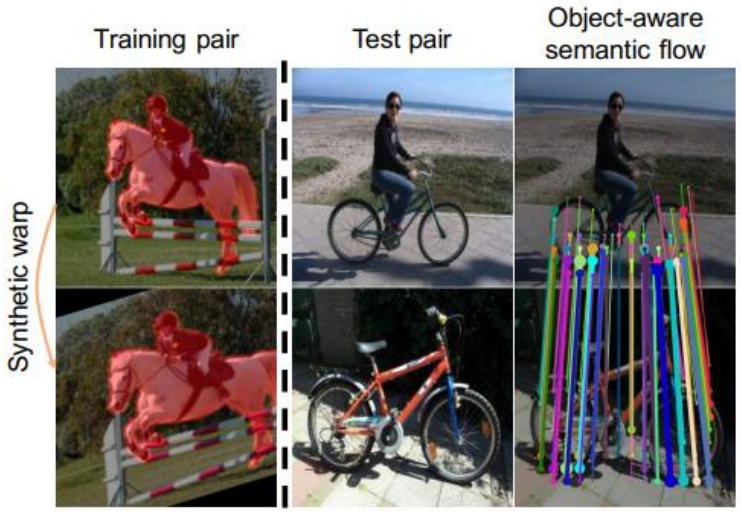
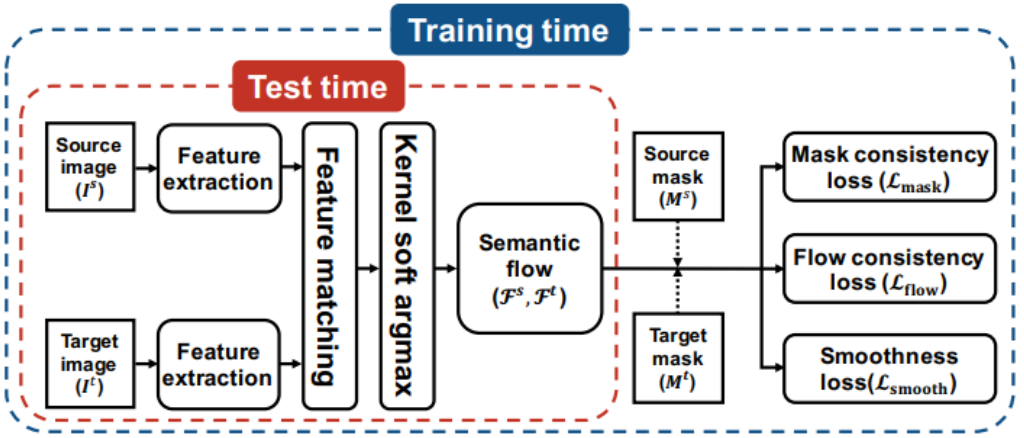
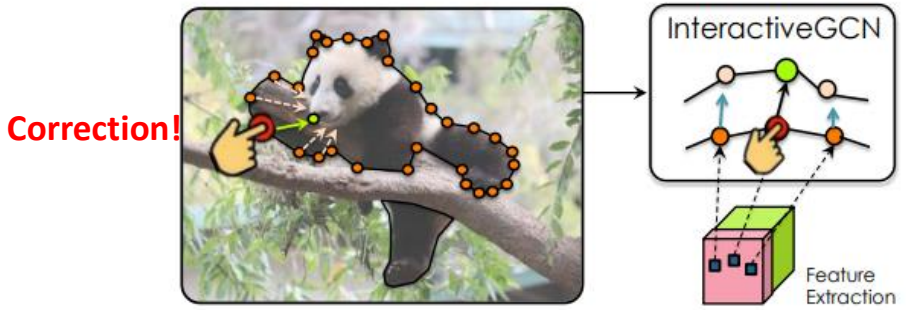


Figure 1: We use pairs of warped foreground masks obtained from a single image (left) as a supervisory signal to train our model. This allows us to establish object-aware semantic correspondences across images depicting different instances of the same object or scene category (right). No masks are required at test time. (Best viewed in color.)



- Propose end-to-end **fast interactive object annotation tool** (Curve-GCN)
  - Predict all vertices **simultaneously** using a Graph Convolutional Network, (→ Polygon-RNN X)
  - Human annotator can correct any wrong point and only the neighboring points are affected



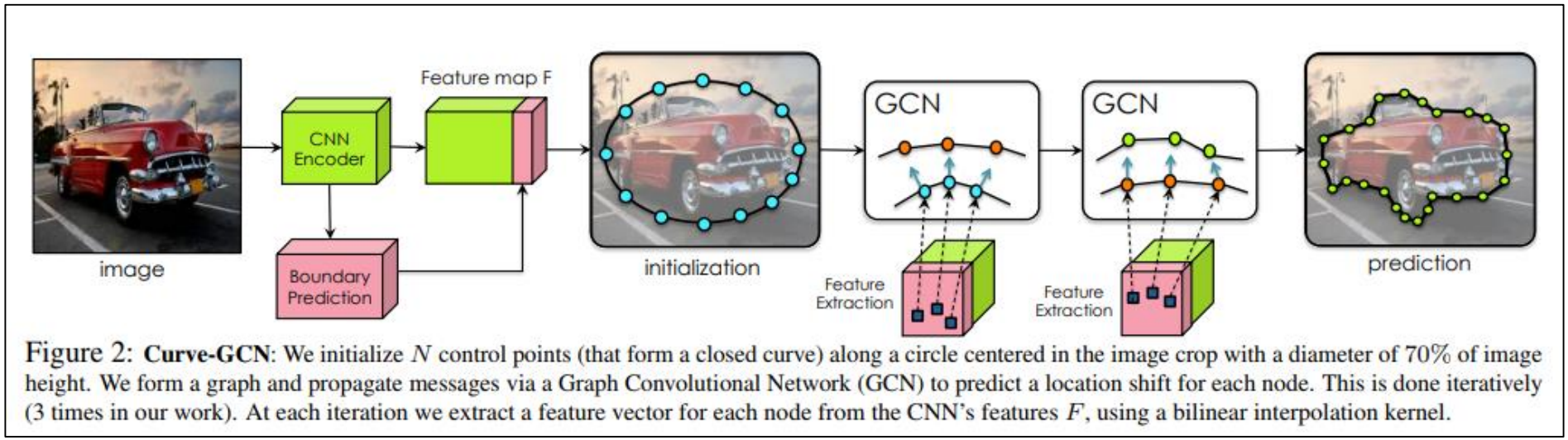
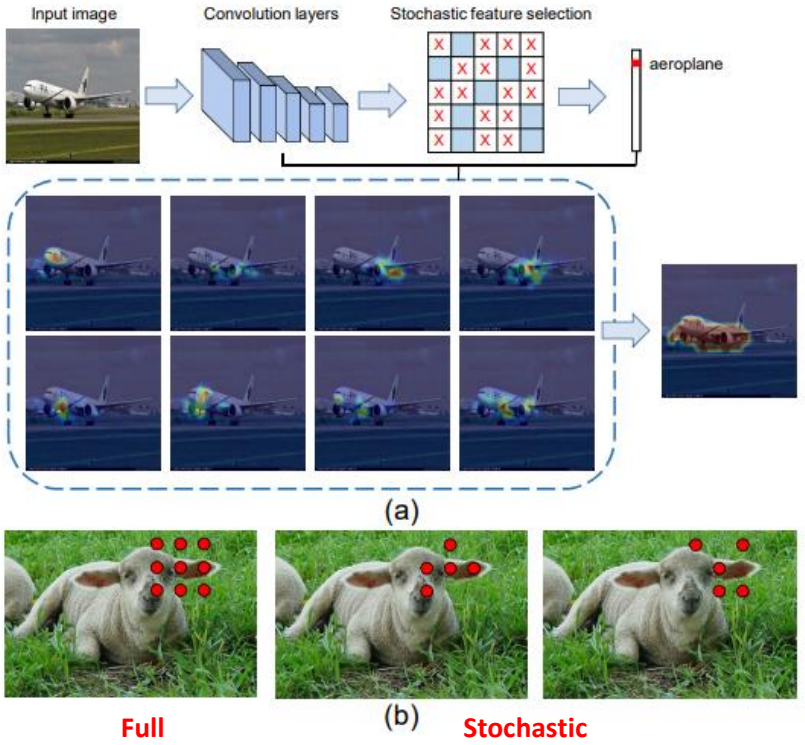


Figure 2: **Curve-GCN**: We initialize N control points (that form a closed curve) along a circle centered in the image crop with a diameter of 70% of image height. We form a graph and propagate messages via a Graph Convolutional Network (GCN) to predict a location shift for each node. This is done iteratively (3 times in our work). At each iteration we extract a feature vector for each node from the CNN's features F, using a bilinear interpolation kernel.

Code: https://github.com/fidler-lab/curve-gcn

Recommended reference: "Efficient interactive annotation of segmentation datasets with polygon-rnn++ ", 2018 CVPR

- Propose **image-level WSSS** method using **stochastic inference (dropout)**
  - Localization maps(CAM) only focus on the small parts of objects → Problem
  - FickleNet allows a single network to generate multiple CAM from a single image
  - Does not require any additional training steps and only adds a simple layer



Both Training and Inference

Algorithm 1: Training and Inference Procedure

Input: Image I, ground-truth label c, dropout rate p
Output: Classification score S and localization maps M

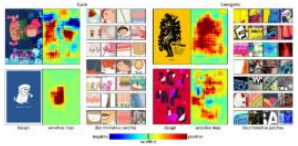
x = Forward(I) until conv5 layer;
Stochastic hidden unit selection: Sec. 3.1
  x^{expand} = Expand(x); Sec. 3.1.1
  x_p^{expand} = Center-fixed spatial dropout(x^{expand}, p); Sec. 3.1.2
  S = Classifier(x_p^{expand}); Sec. 3.1.3
Training Classifier:
  Update network by L=SigmoidCrossEntropy(S, c)
Inference CAMs: Sec. 3.2
  For different random selections i ($1 \leq i \leq N$):
    $M^c[i]$ = Grad-CAM(x, S^c); Sec. 3.2.1
  M^c = Aggregate($M^c[i]$); Sec. 3.2.2

- In my personal blog, there are similar works
  - SIGGRAPH 2018
  - NeurIPS 2018
  - ICLR 2019

https://hoya012.github.io/

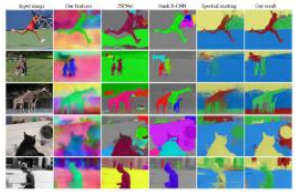
1. What Characterizes Personalities of Graphic Designs?

이 논문은 아래의 그림에서 알 수 있듯이 그래픽 디자인 이미지가 있으면 해당 이미지의 personality(성격)을 예측하는 딥러닝 모델을 제안하고 있습니다. 또한 이미지의 어떤 부분이 해당 성격에 긍정적인, 혹은 부정적인 영향을 주고 있는지를 나타내는 sensitive map까지 얻을 수 있습니다.



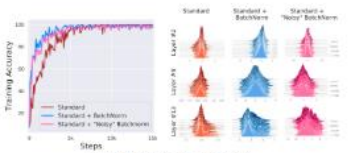
포스터 데이터셋을 이용하였으며 총 15가지의 Personality에 대해 분류를 하는 문제를 해결하였고, 선행 연구들에 비해 높은 정확도를 달성한 논문입니다. 개인적으로 연구를 하다가 직접 데이터셋을 취득해야 할 때 포스터 이미지들, 책 표지 이미지들을 활용하였었는데 비슷한 관점에서 다룬 논문이라 흥미로웠습니다. 또한 sensitive map도 비교적 정교하게 얻을 수 있어서 이를 활용할 수 있는 여러 방식이 있을 것 같습니다.

2. Semantic Soft Segmentation



1. How Does Batch Normalization Help Optimization? (Oral)

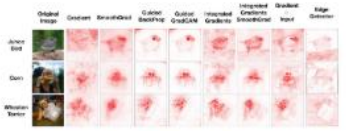
- Batch normalization이 optimization을 잘 되게 해주는 이유를 분석한 논문. 일반적으로 알려져 있는 "internal covariate shift" 효과는 실제로는 미미하며, optimization landscape를 smooth하게 해주는 효과가 optimization에 큰 도움을 주고 있음을 증명함.



[본 논문의 성능 표, 그림 예시]

2. Sanity Checks for Saliency Maps (Spotlight)

- Saliency method들은 학습 결과와 관련이 있는 입력의 feature를 강조하는데 사용이 됨. 이러한 방식들이 좋은 지 나쁜 지 판단할 기준이 애매한데, 실험을 통해 saliency map의 온전성 검사를 하는 방식을 제안함.



[다양한 saliency map 예시]

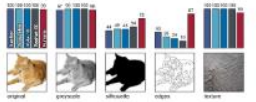
1. Large Scale GAN Training for High Fidelity Natural Image Synthesis (Oral)

- Rating: 7 / 10 / 9 , avg. 8.67
- 512x512 크기의 이미지와 같이 high resolution 이미지를 생성하는 generative model **BigGAN** 제안. ICLR paper중 가장 높은 rating을 받았으며 실제로 결과 이미지들을 보면 해상도가 큰 이미지인데도 그럴싸하게 생성해내는 것을 확인할 수 있음.

[본 논문의 결과 그림 예시]

2. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness (Oral)

- Rating: 7 / 8 / 8 , avg. 7.67
- ImageNet으로 pretrain된 CNN은 object의 texture에 bias 되어있음을 보이며, global object shape 정보를 이용하면 robust한 CNN을 만들 수 있음을 보임. 또한 실험을 위해 Style Transfer 알고리즘을 이용하여 ImageNet으로부터 Stylized-ImageNet 이라는 데이터셋을 생성한 점이 인상 깊음. 해당 데이터셋은 해당 링크에서 확인할 수 있음.



[본 논문의 그림 예시 1]

Thank you

SUALAB