Computer Vision for Embedded Systems

Yung-Hsiang Lu
Purdue University
yunglu@purdue.edu

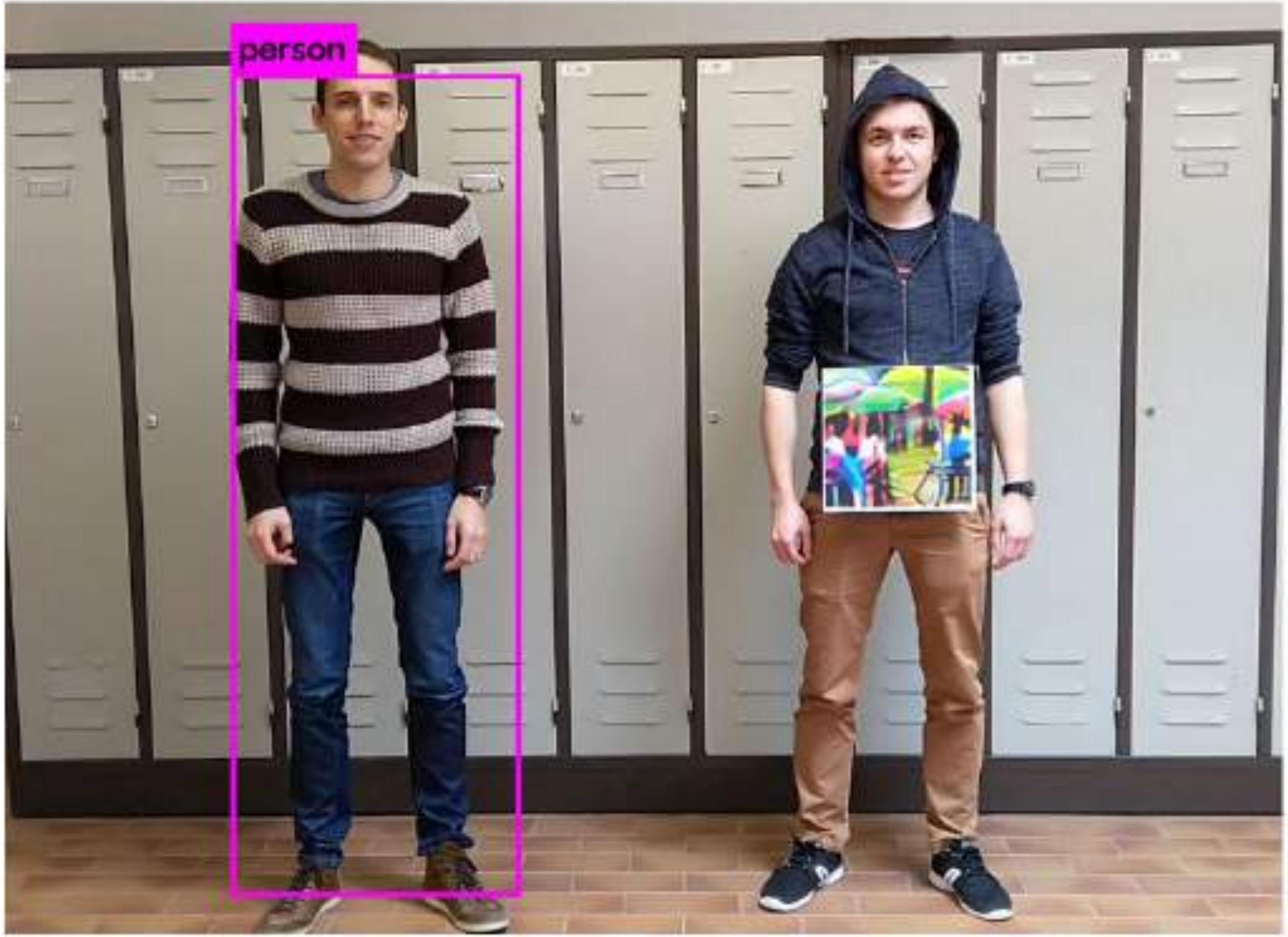
PURDUE UNIVERSITY

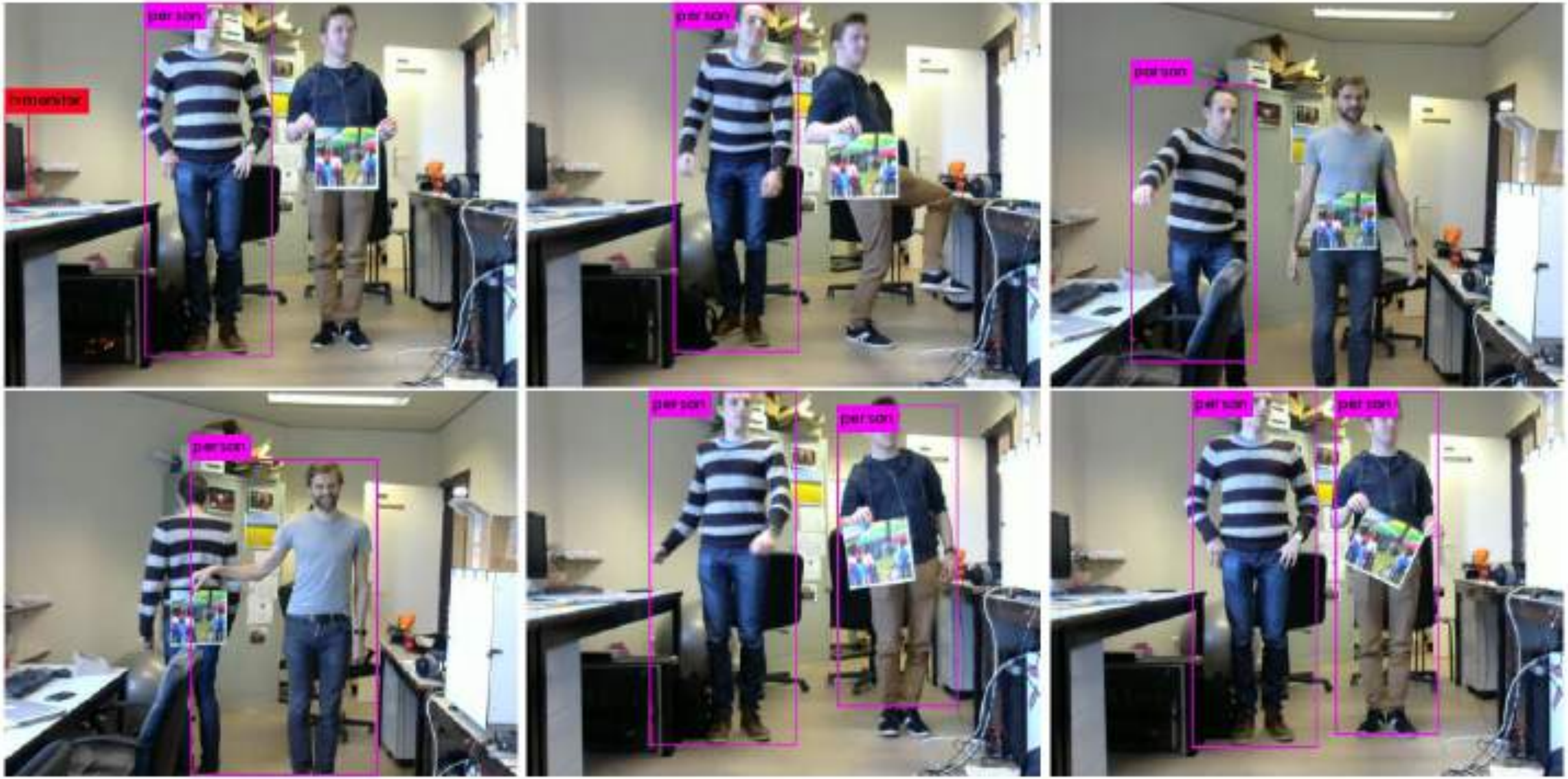
Fooling Neural Networks

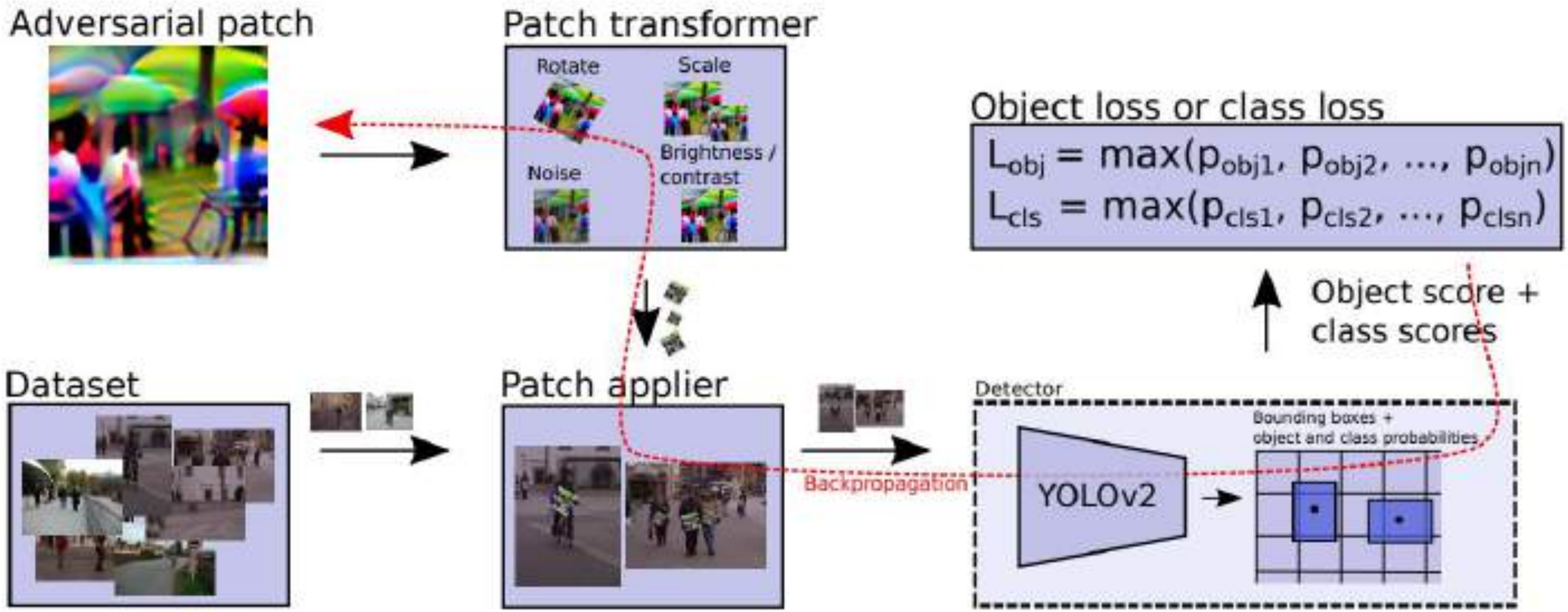
Fooling automated surveillance cameras: adversarial patches to attack person detection

https://arxiv.org/pdf/1904.08653.pdf

https://www.youtube.com/watch?v=MIbFvK2S9g8&ab_channel=AnonymousCVCOPS


person


person
person
person
person
person
person
person
person


Adversarial patch
Patch transformer
Rotate
Scale
Noise
Brightness / contrast
Object loss or class loss
$L_{obj} = \max(p_{obj1}, p_{obj2}, \ldots, p_{objn})$
$L_{cls} = \max(p_{cls1}, p_{cls2}, \ldots, p_{clsn})$
Object score + class scores
Dataset
Patch applier
Backpropagation
Detector
Bounding boxes + object and class probabilities
YOLOv2

Generative Adversarial Networks GAN

Communication of the ACM, November 2020

IEEE Signal Processing Magazine, January 2018

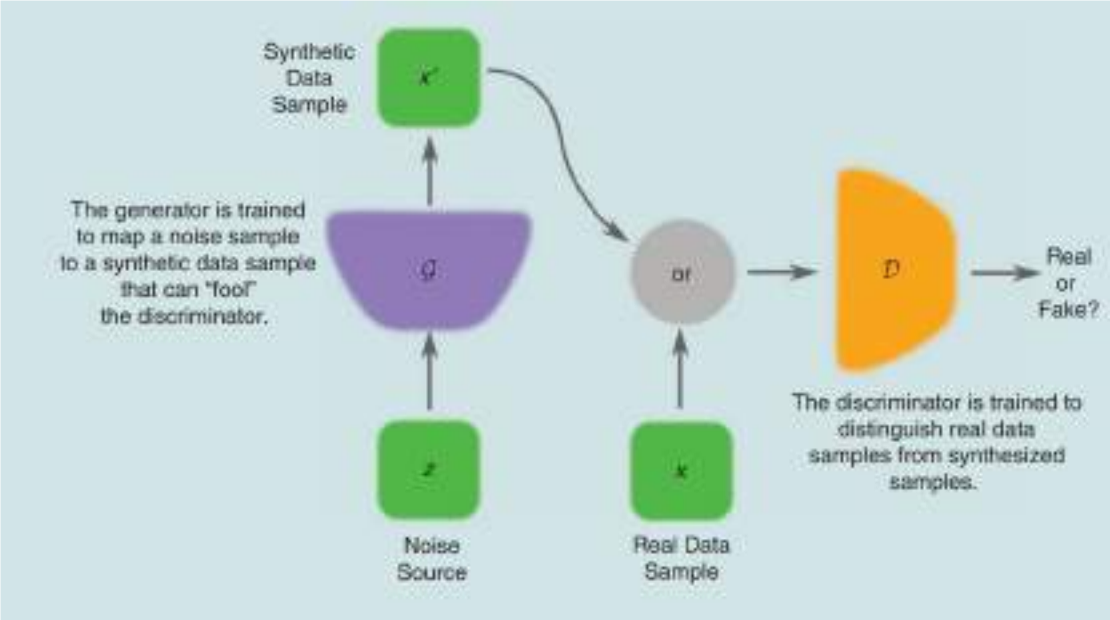
NIPS 2016 Tutorial: Generative Adversarial Networks

Why Generative Models?

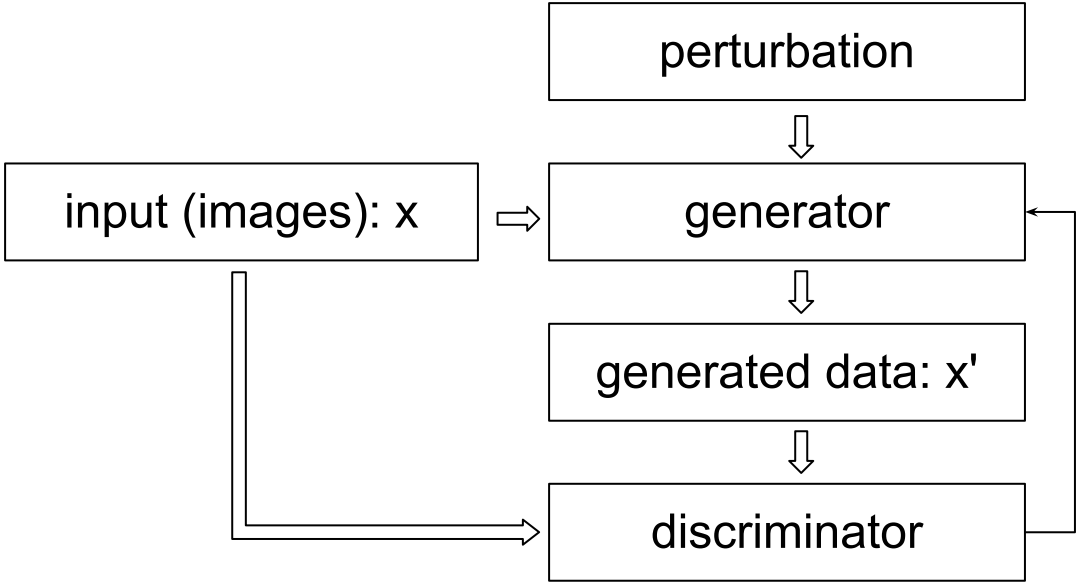
- supervised learning
  - goals are well-defined: map inputs to correct outputs
  - need data + answers
  - need human supervision
  - answers need to be generated by humans
- unsupervised learning
  - to find "patterns" but what is a pattern?
  - goals not clearly defined
  - clustering and dimension reduction are common
- Generative models: Generate data with specific properties

Supervised Learning vs Generative Model

input (images) ⇨ machine model ⇨ answer ⇨ ⊖ ⇦ truth



Generative Model based on Data



real or generated?

Progression of GAN

2014 2015 2016 2017

[redacted]

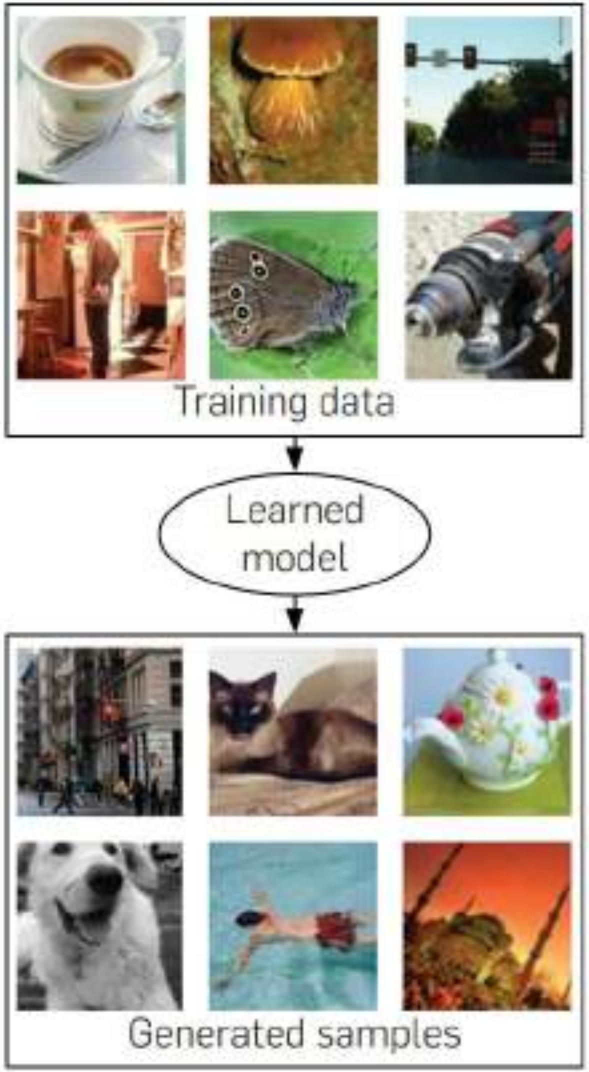
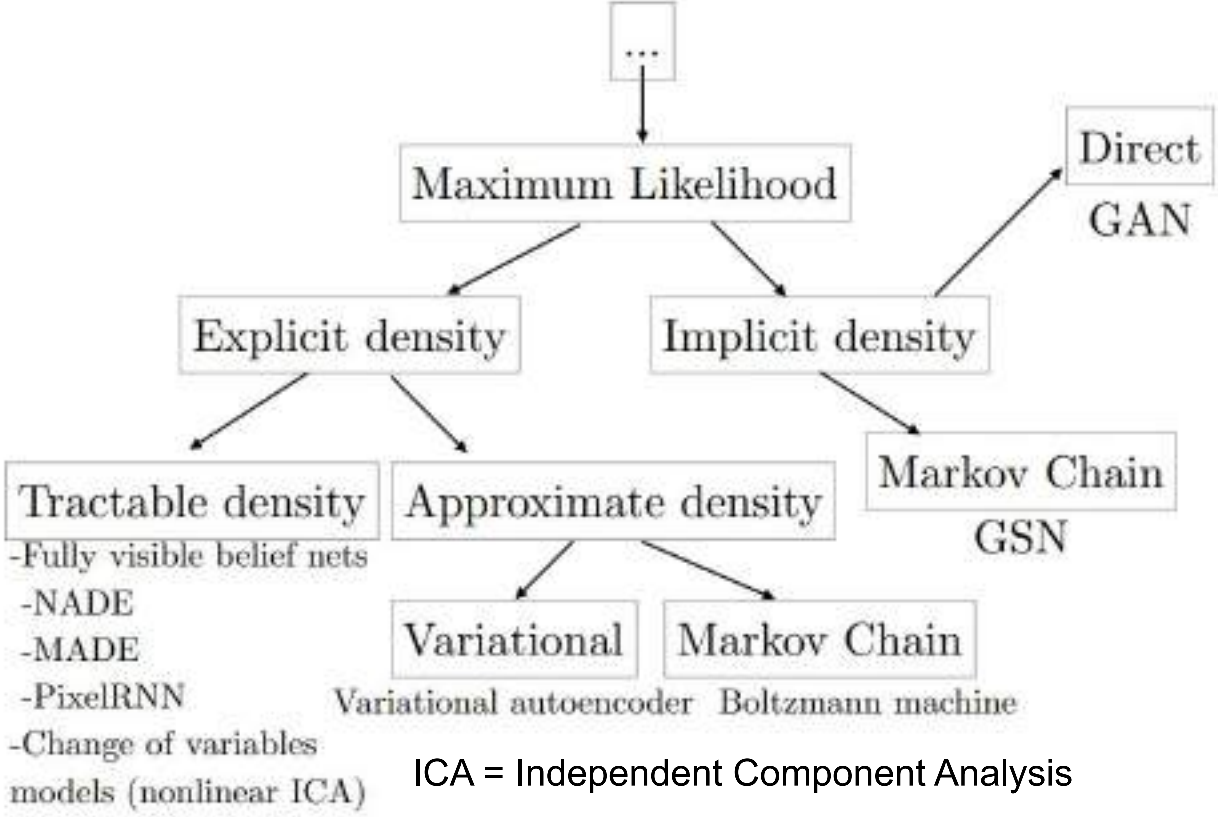


Figure 18: Samples of images of bedrooms generated by a DCGAN trained on the LSUN dataset.

Advantages of Generative Models

- Test the generality of the trained machine models
- Conduct reinforcement learning with data, without model
- Enhance supervised learning with data without labels
- Improve data quality (from low resolution to high)
- Create artwork
- Translate images

Taxonomy of Generative Models

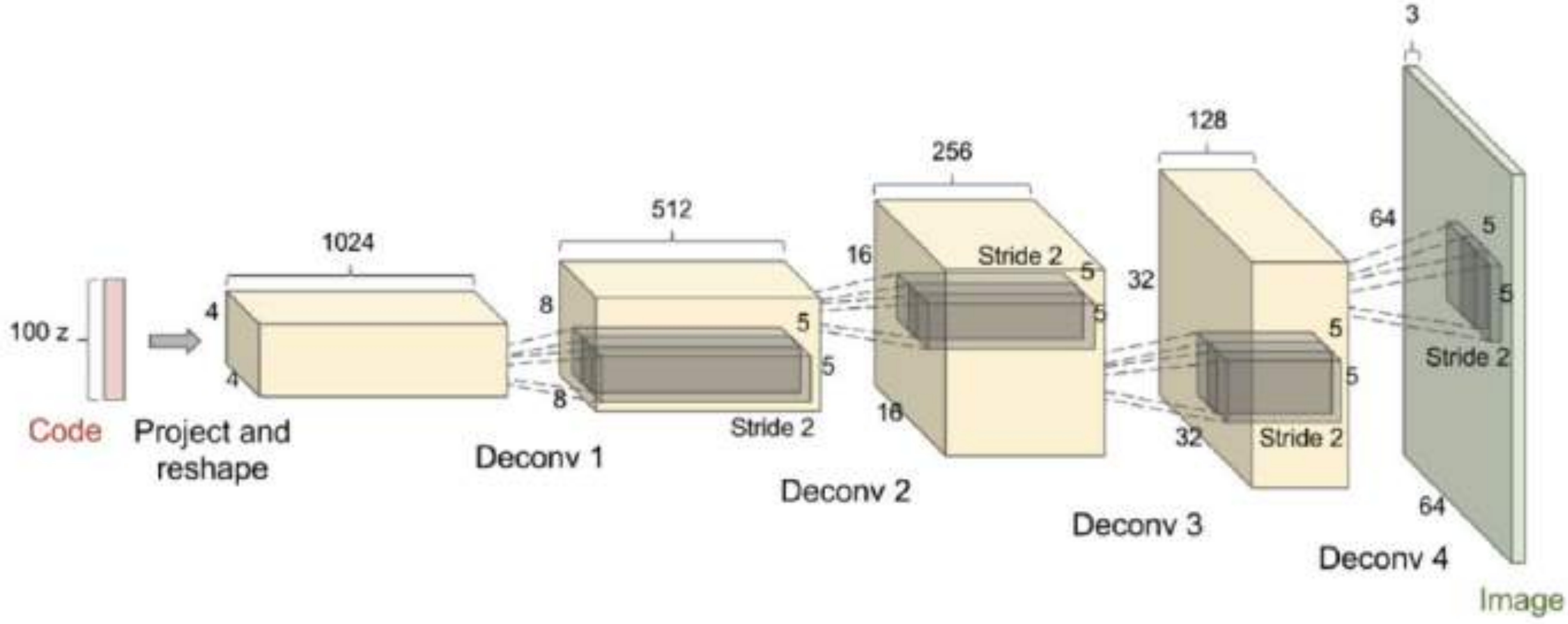


ICA = Independent Component Analysis

Advantage of GAN over other generative models

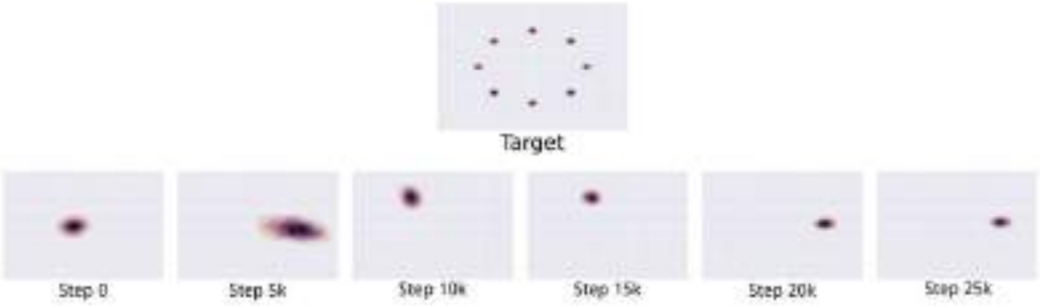
- GAN can generate data in parallel
- fewer restrictions
- No need of Markov chains
- Use game theory for strategies

DCGAN (deep convolution GAN)



Research Questions

- convergence: no theory about the conditions
- mode collapse
- systematic evaluation
- discrete outputs



Consistency vs. Accuracy

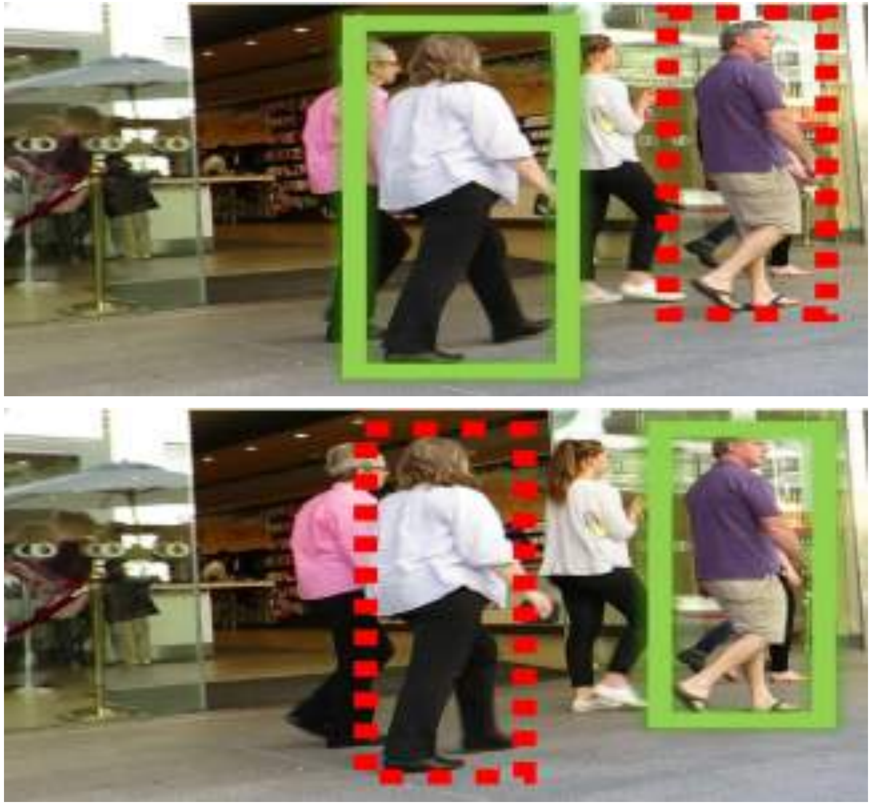
IEEE Multimedia (to appear)

Caleb Tung
Purdue Doctoral Student (2022)

Consistency

Mask-RCNN

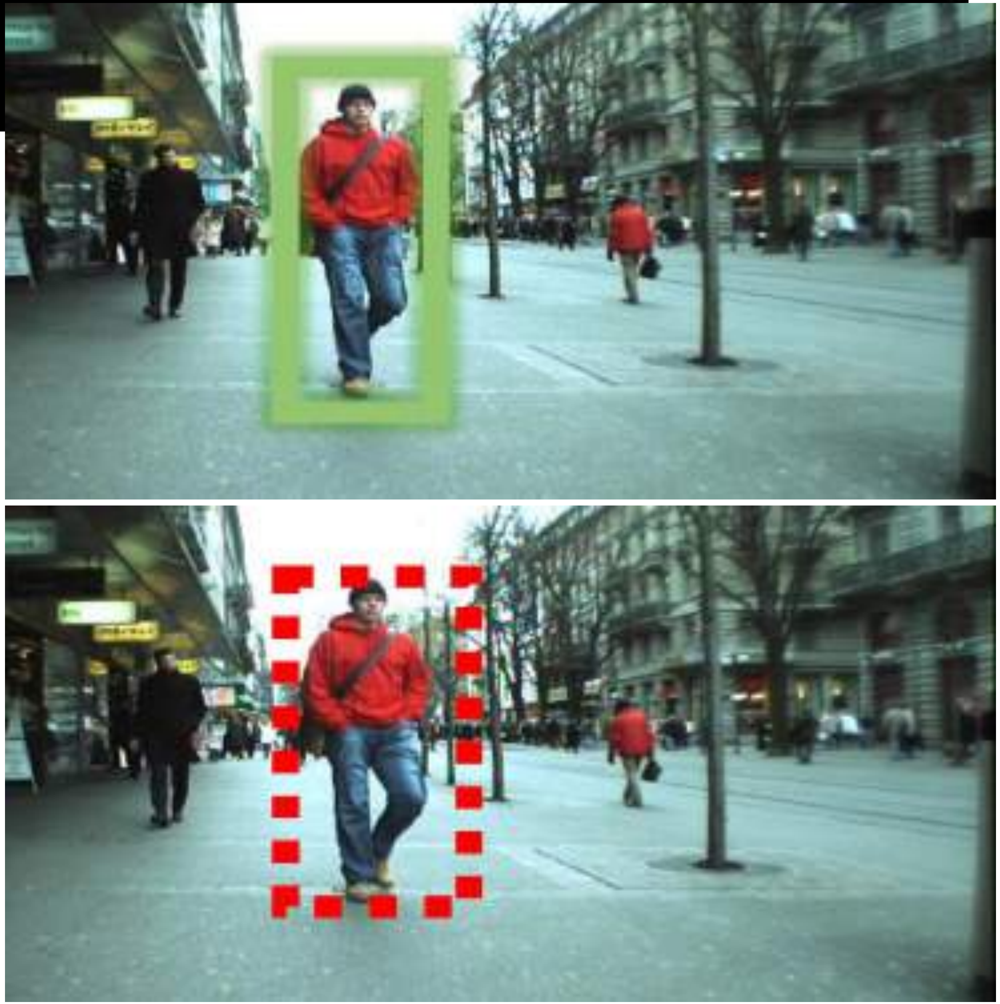
green: detected
red: missed



Faster RCNN

RetinaNet

Single Shot Detector



Consistency vs Accuracy

50% accuracy, consistent

50% accuracy, inconsistent

Define Consistency

$$C_{i,j} = \frac{|G_i \cap G_j| - |M_{i,j}| - |M_{j,i}|}{|G_i \cap G_j|}$$

ground truth of image i

object detected in image i,
missed in image j

consistency of images i and j

Example

Green box: detected

$$G_i \cap G_j = \{A, B\}$$

$$M_{i,j} = \{B\}$$

$$M_{j,i} = \{\}$$

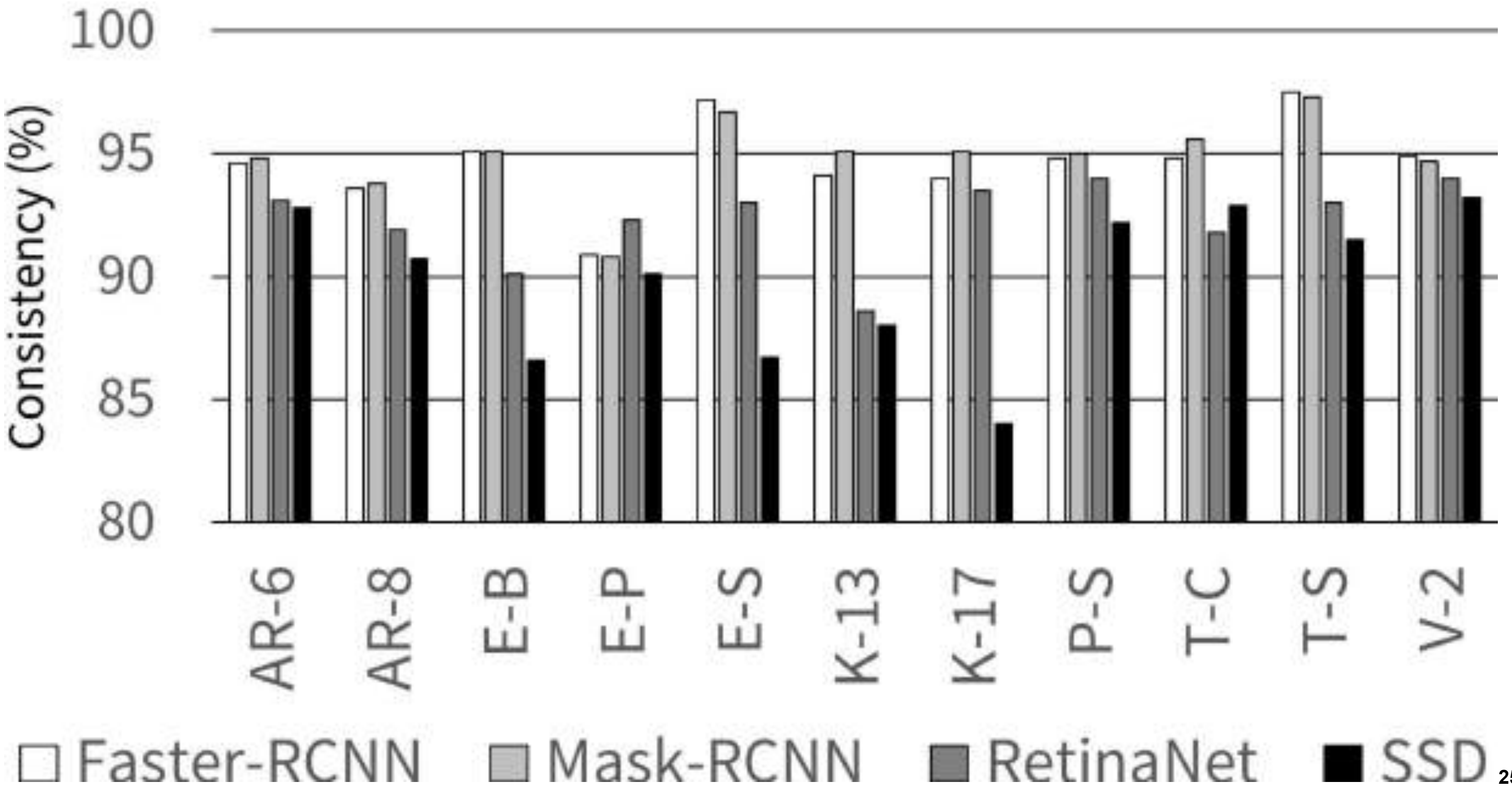
$$C_{i,j} = \frac{2-1}{2} = 0.5$$

image i: A and B detected, C missed

image j: A and D detected, B missed

Do you agree with this definition?

Consistency of Popular Object Detectors MOT (Multiple Object Tracking) Dataset



Methods to Improve Consistency

GD: Gaussian Denoise; HF: Horizontal Flip
WC: WEBP compression (for websites)
UM: Unsharp mask (to remove motion blur)
GC: Gamma correction (enhance contrast)

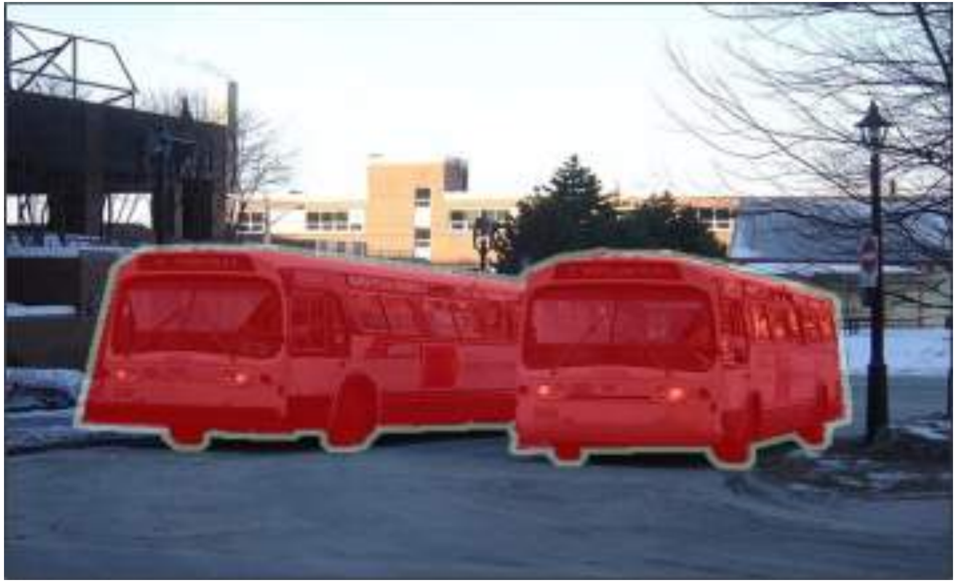
	Faster-RCNN	Mask-RCNN	RetinaNet	SSD
GD	0%	-0.3%	0%	-0.6%
HF	-5.3%	-5.4%	-7.3%	-10.1%
WC	**0.6%**	**0.5%**	**0.7%**	**0.4%**
UM	**3.6%**	**2.6%**	**3.0%**	**1.1%**
WC+UM	**5.1%**	**3.0%**	**3.2%**	**1.3%**
GC	0.1%	0.1%	0.4%	0.1%

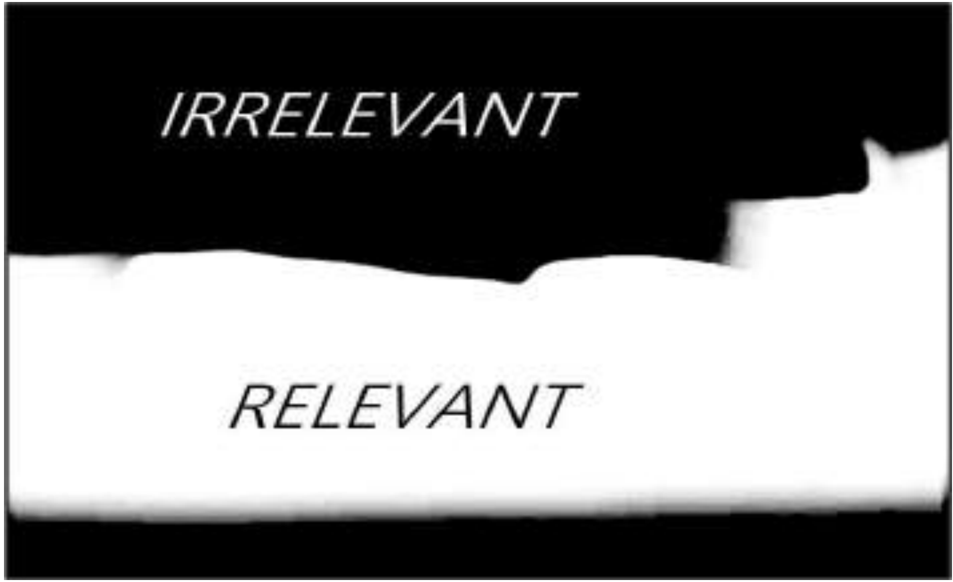
Faster-RCNN	Mask-RCNN	RetinaNet	SSD
2.1%	2.4%	-0.6%	-1.1%
-19.3%	-19.4%	-25.5%	-28.4%
1.5%	**1.8%**	**0.5%**	**0.5%**
2.0%	**3.2%**	**8.3%**	**3.6%**
3.2%	**4.1%**	**8.6%**	**3.9%**
0.1%	-0.5%	-0.7%	-0.1%

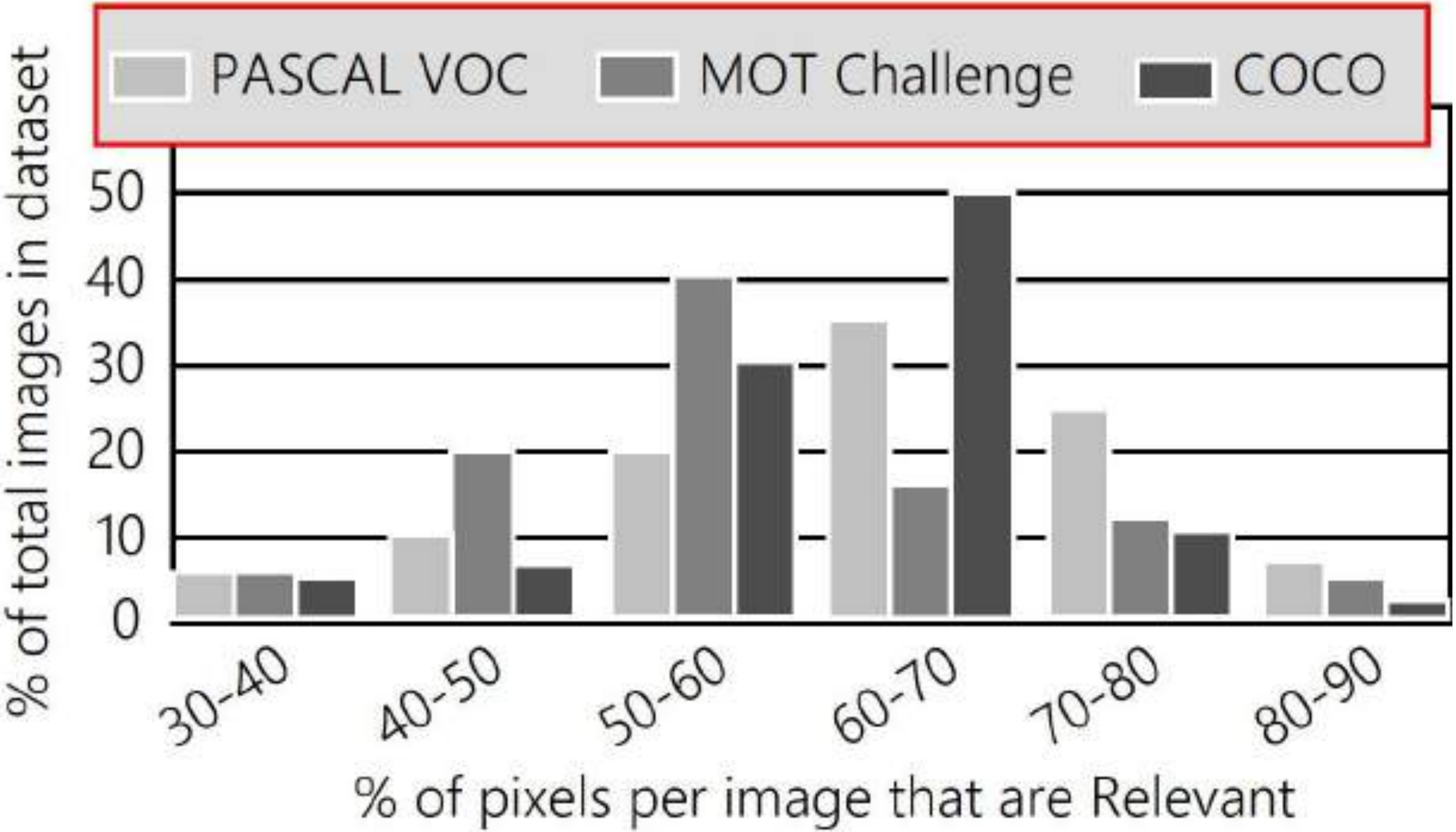
improvement in consistency   improvement in accuracy

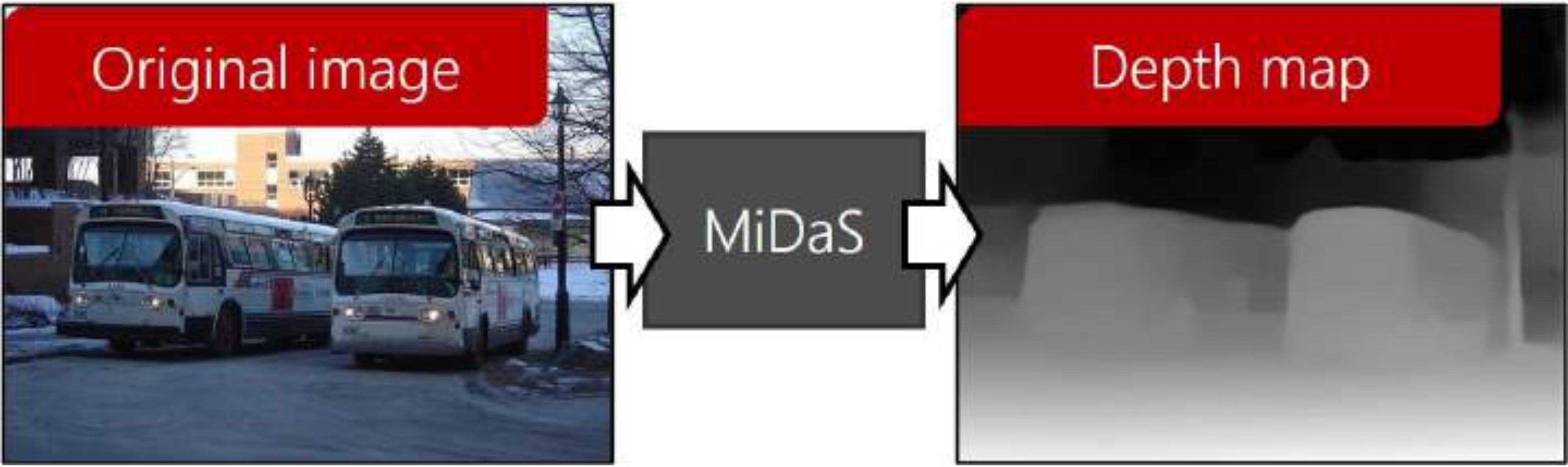
Irrelevant Pixels are Everywhere: Find and Exclude Them for More Efficient Computer Vision

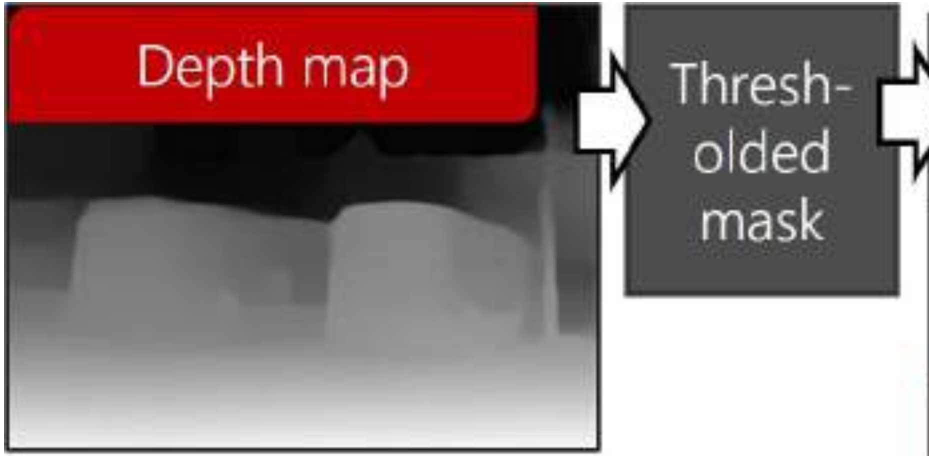
Artificial Intelligence Circuits and Systems 2022



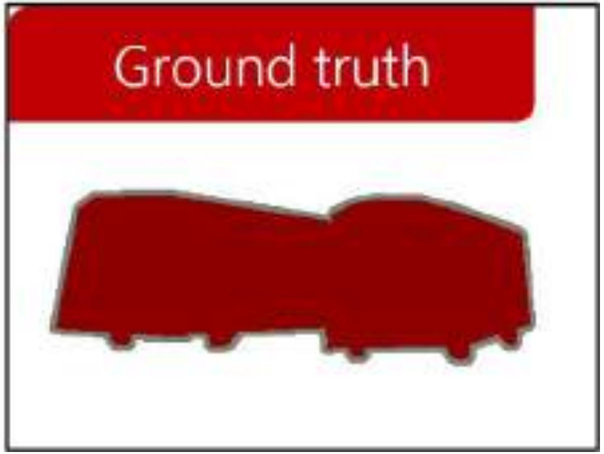

IRRELEVANT
RELEVANT


PASCAL VOC
MOT Challenge
COCO
% of total images in dataset
0
10
20
30
40
50
30-40
40-50
50-60
60-70
70-80
80-90
% of pixels per image that are Relevant

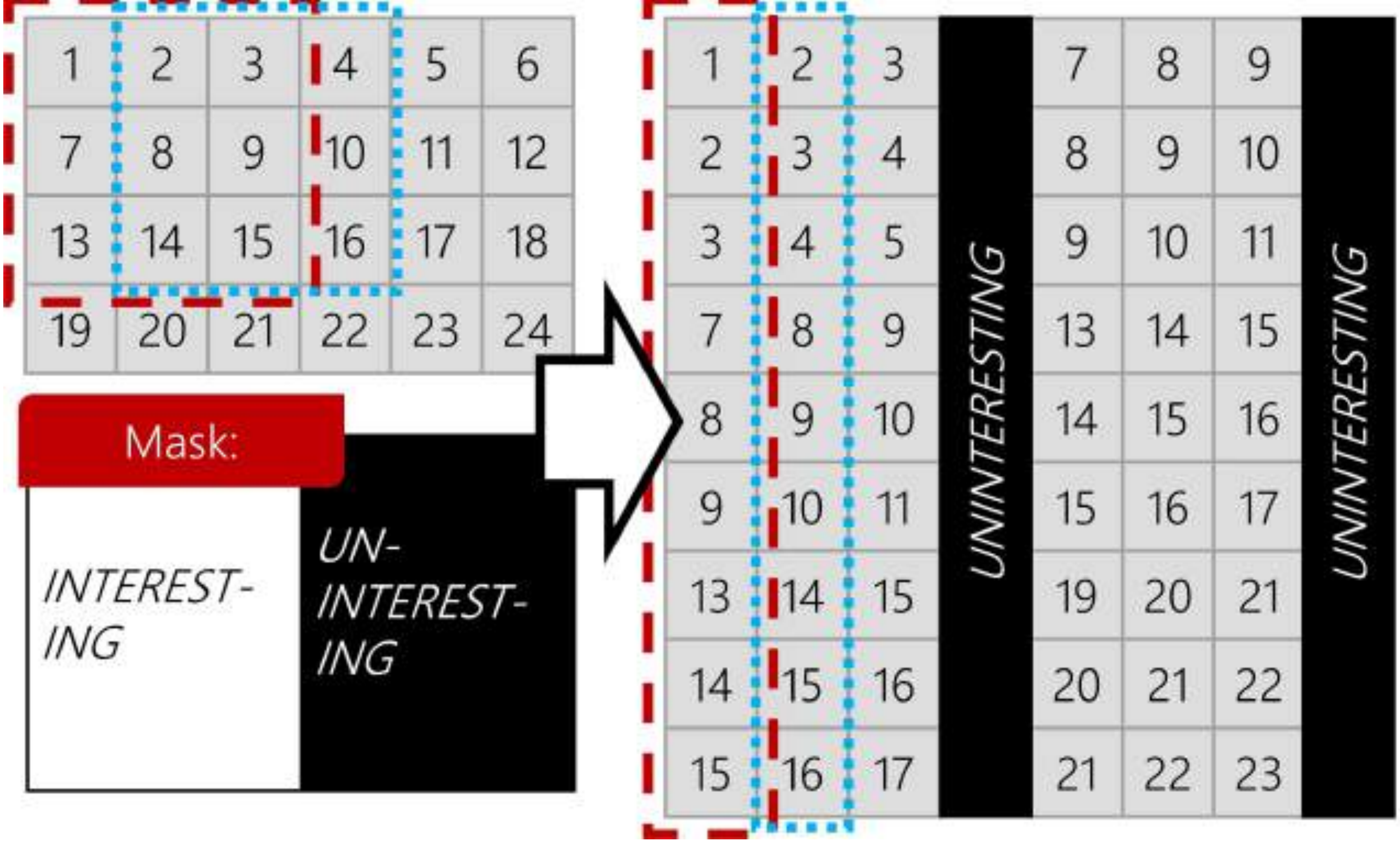

Original image
MiDaS
Depth map



All pixels at Depth Level > thresh are marked as Relevant. Others are Irrelevant.



Verify RELEVANT pixels contain ground truth to ensure no data is missed


Mask:
INTEREST-ING
UN-INTEREST-ING
UNINTERESTING
UNINTERESTING

		MOT2015		*COCO*		*PASCAL VOC*	
		ED	*SL*	*ED*	*SL*	*ED*	*SL*
Number of Mult-Add Operations (M/inference)							
	Normal	384.5	483.6	384.5	483.6	384.5	483.6
	Focused	**196.1**	**246.8**	**211.4**	**266.0**	**223.0**	**280.4**
Inference Latency (s/inference)							
RPi (5W)	Normal	2.10	2.26	2.00	2.33	2.06	2.29
	Focused	**1.11**	**1.30**	**1.33**	**1.51**	**1.47**	**1.56**
Intel (28W)	Normal	0.25	0.28	0.25	0.29	0.25	0.28
	MKL	0.18	0.19	0.18	0.20	0.18	0.20
	Focused	**0.17**	**0.18**	**0.18**	**0.20**	**0.19**	**0.20**
Energy Consumption (J/inference)							
RPi (5W)	Normal	10.22	11.80	10.15	11.81	10.20	10.90
	Focused	**5.60**	**6.11**	**6.71**	**7.50**	**7.44**	**7.80**
Intel (28W)	Normal	6.61	7.39	6.45	7.42	6.69	7.81
	MKL	5.18	5.09	5.09	5.61	5.23	5.60
	Focused	**4.76**	**5.04**	**5.10**	**5.60**	**5.29**	**5.62**

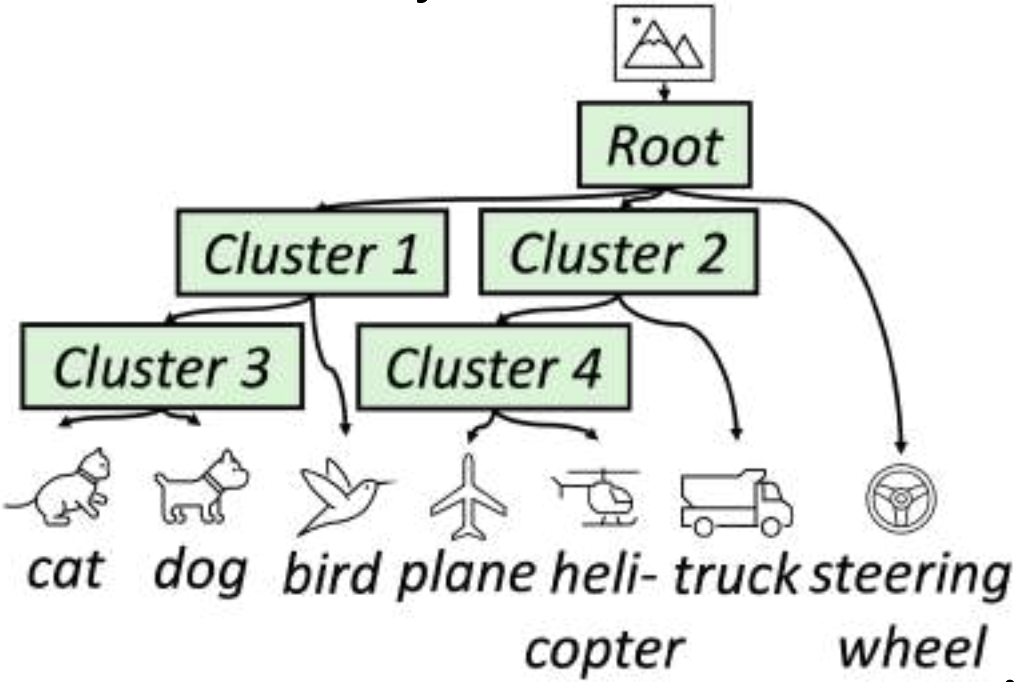
Directed Acyclic Graph-based Neural Networks for Tunable Low-Power Computer Vision

International Symposium on Low Power Electronics and Design 2022 (ISLPED)

Abhinav Goel
Purdue PhD 2022
now at Nvidia

What is the problem?

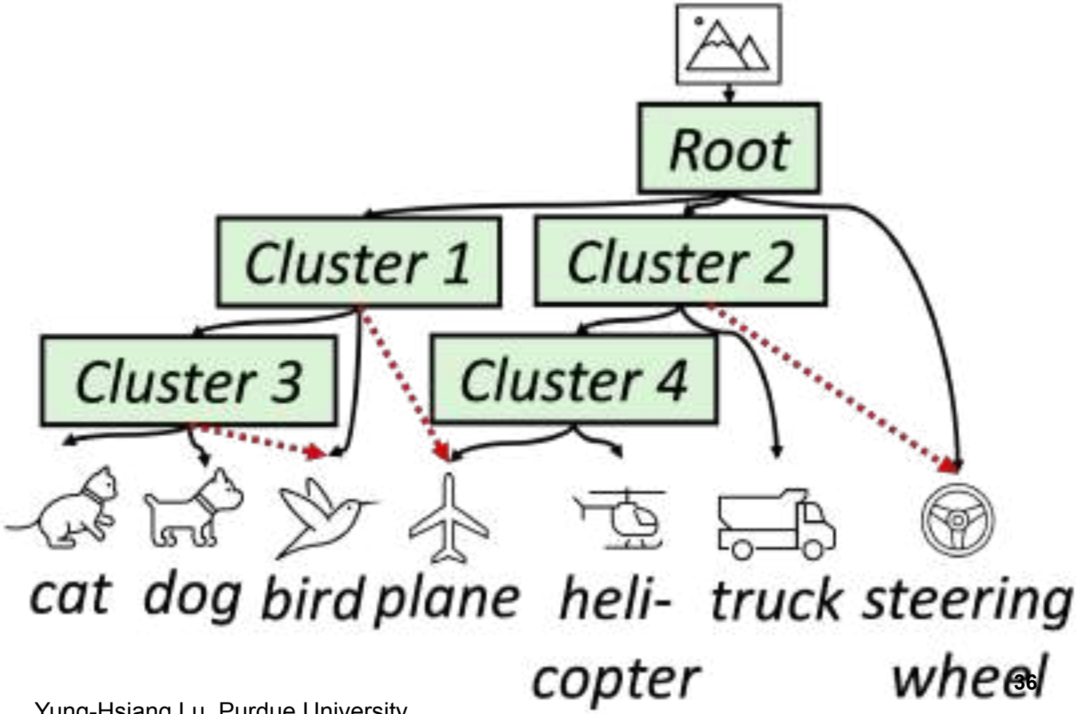
In a tree structure, there is only one path from the root to any leaf. If a mistake is made, there is no way to correct the mistake.



Solution: Directed Acyclic Graph-based (DAG)

Add paths to correct mistakes

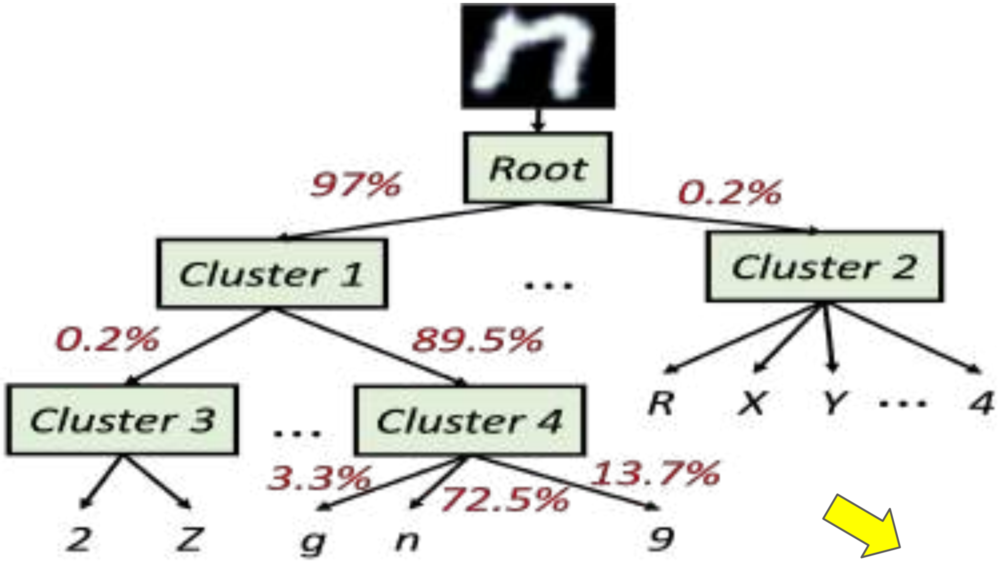
Questions: which paths to add? how much will the memory requirements increase?

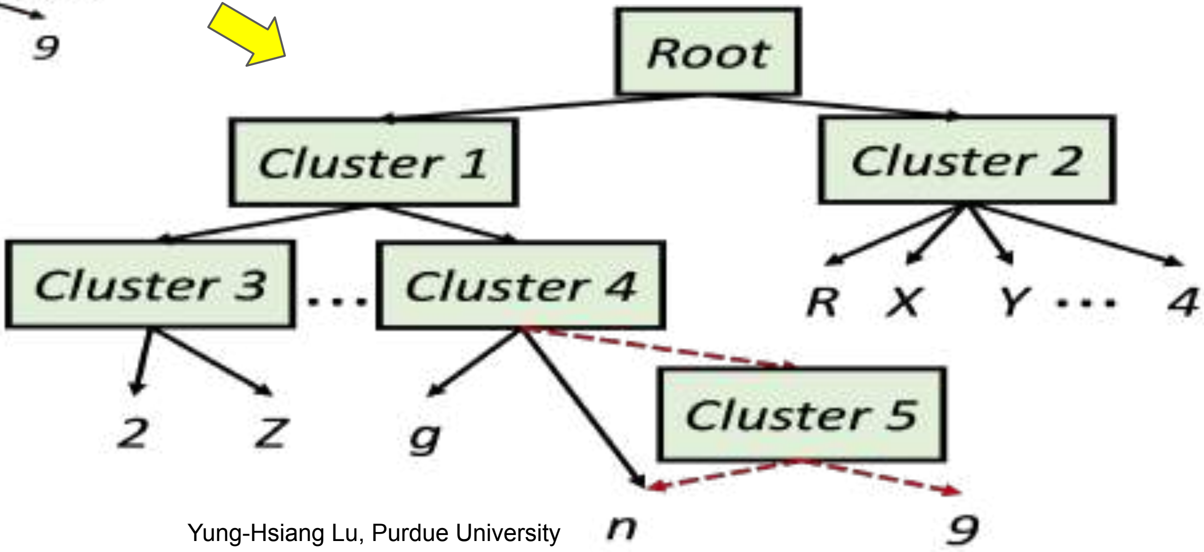


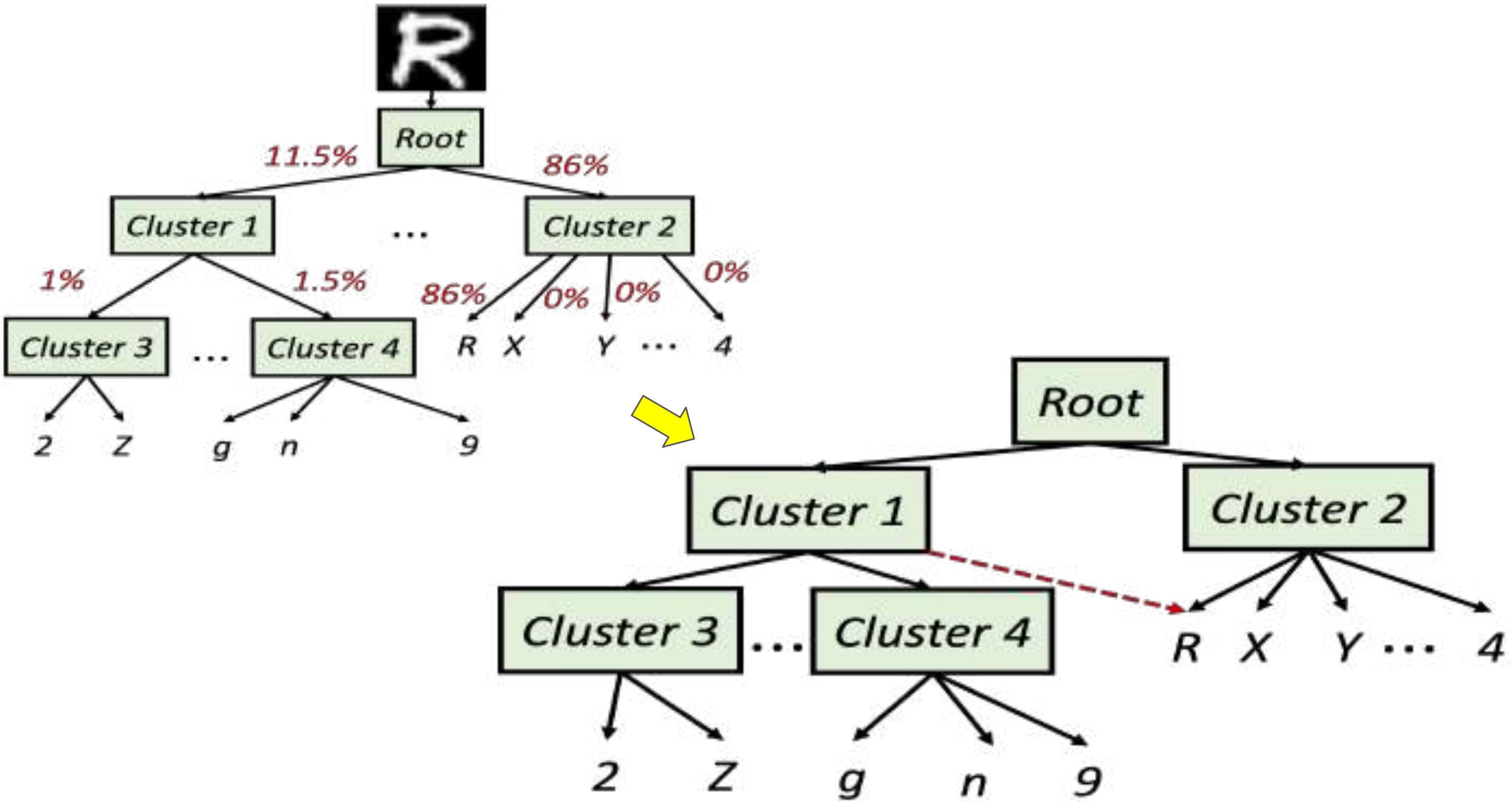
Trade-Off

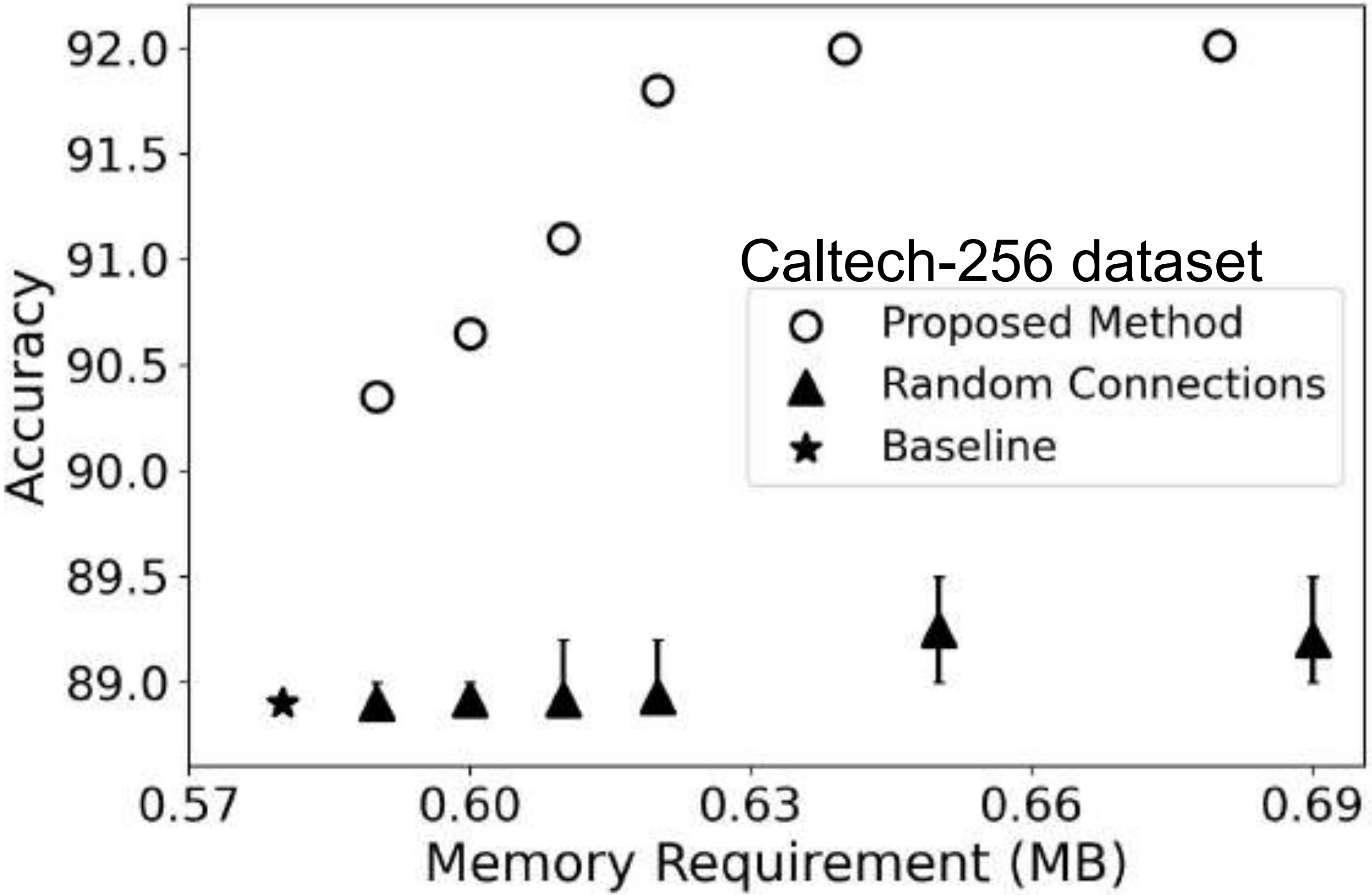
- adding none or too few ⇒ low accuracy
- adding too many ⇒ becomes a large and deep CNN (larger memory requirements)

Add only the most impactful paths


Root
97%
0.2%
Cluster 1
...
Cluster 2
0.2%
89.5%
Cluster 3
...
Cluster 4
R X Y ... 4
3.3%
72.5%
13.7%
2 Z g n 9


Root
Cluster 1
Cluster 2
Cluster 3
...
Cluster 4
R X Y ... 4
Cluster 5
2 Z g
n 9


Root
11.5%
86%
Cluster 1
...
Cluster 2
1%
1.5%
86%
0%
0%
0%
Cluster 3
...
Cluster 4
R X Y ... 4
2 Z g n 9
Root
Cluster 1
Cluster 2
Cluster 3
...
Cluster 4
R X Y ... 4
2 Z g n 9


Caltech-256 dataset
Proposed Method
Random Connections
Baseline
Accuracy
92.0
91.5
91.0
90.5
90.0
89.5
89.0
0.57
0.60
0.63
0.66
0.69
Memory Requirement (MB)

Dataset	Technique	Accuracy (%)	Model Size (MB)	FLOPs ($\times 10^6$)
	HDNN [5]	91.20	0.25	2.13
	DAG-Net 1	91.30	0.27	2.14
	DAG-Net 2	91.70	0.28	2.17
EMNIST	DAG-Net 3	92.00	0.29	2.79
	DAG-Net 4	92.14	0.32	3.21
	DAG-Net 5	92.15	0.37	3.45
	VGG-5 [19]	92.59	15.00	161.24
	ResNet9 [2]	92.00	26.00	636.71

Dataset	Technique	Raspberry Pi 4B Latency	Raspberry Pi 4B Energy	NVIDIA Jetson Nano Latency	NVIDIA Jetson Nano Energy
EMNIST	HDNN	0.053	0.28	0.320	2.46
	DAG-Net 1	0.057	0.30	0.320	2.47
	DAG-Net 3	0.062	0.32	0.322	2.49
	DAG-Net 5	0.066	0.35	0.322	2.49
	VGG-5	0.431	2.27	4.041	31.15