# Computer Vision for Embedded Systems

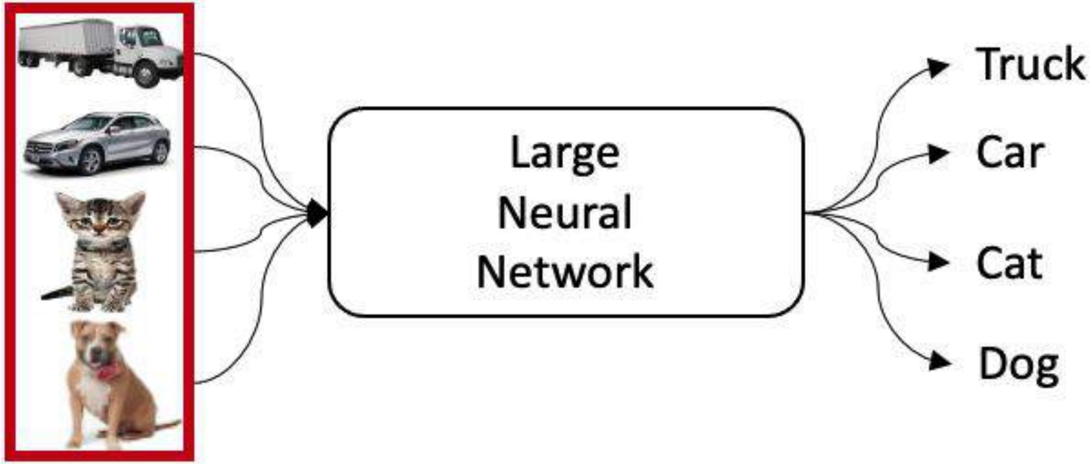
Yung-Hsiang Lu
Purdue University
yunglu@purdue.edu

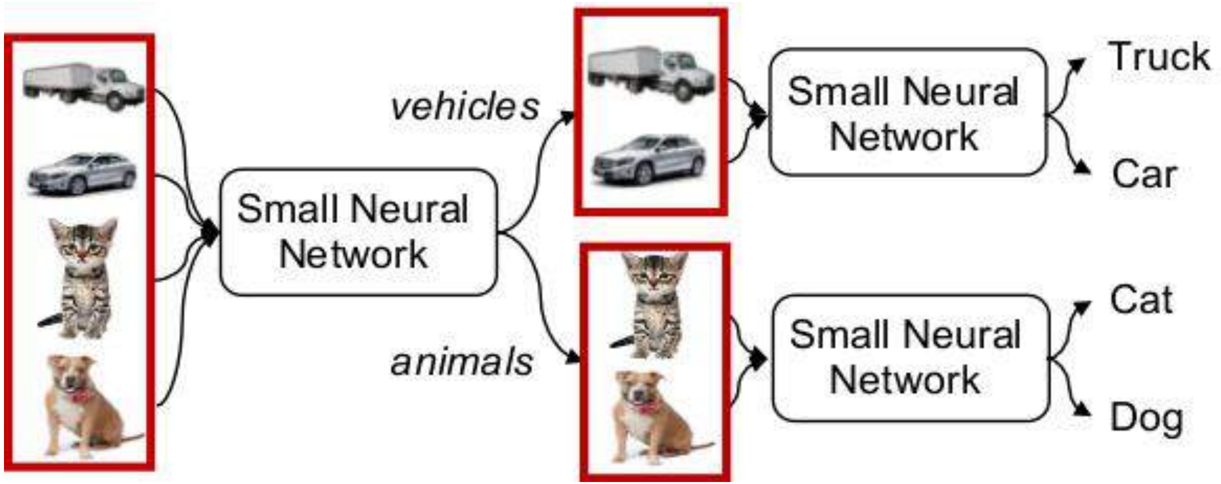
PURDUE UNIVERSITY

# References

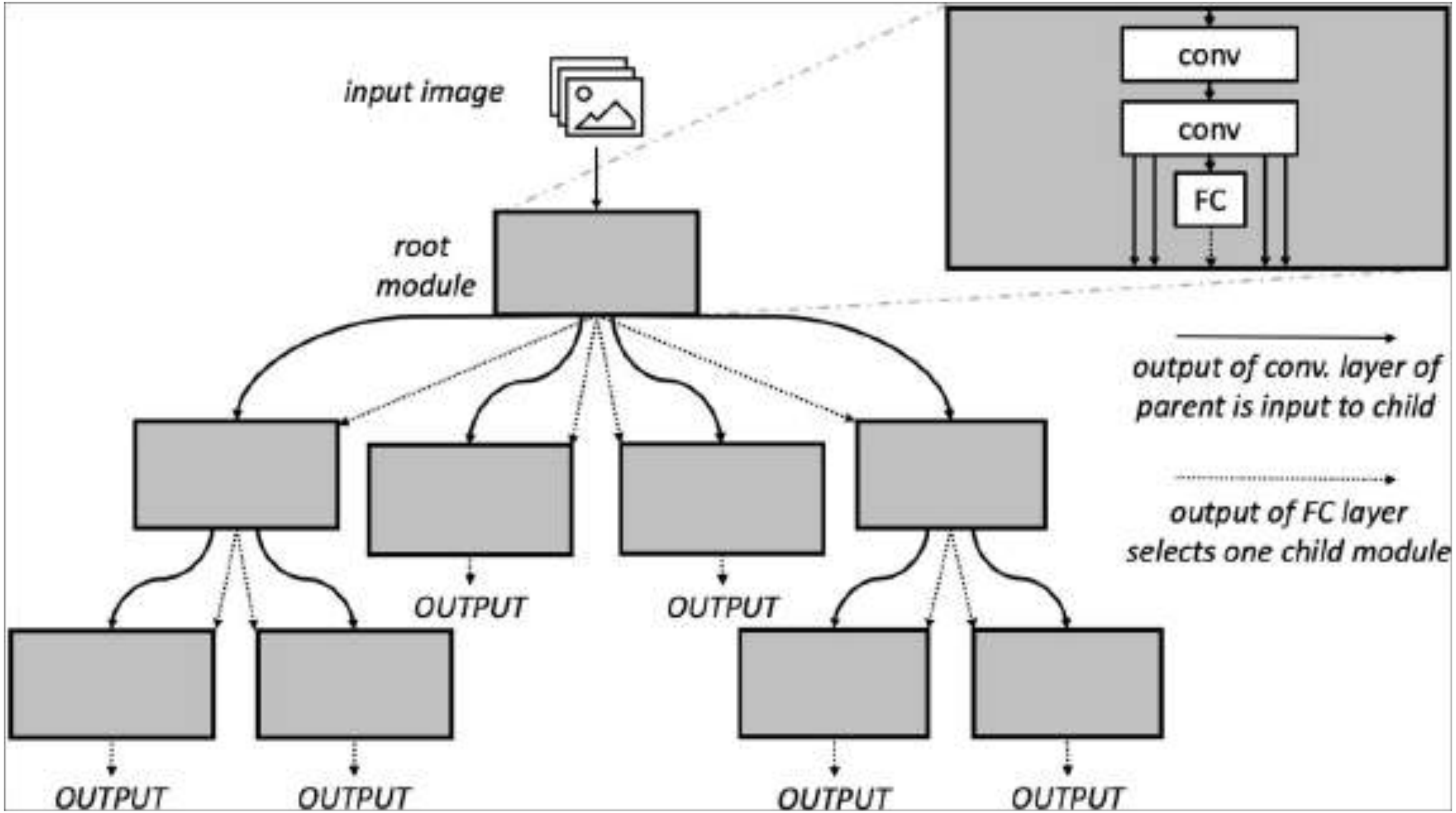

1. Goel, et al., “Modular Neural Networks for Low-Power Image Classification on Embedded Devices”, ACM Transactions on Design Automation of Electronic Systems, October 2020.
2. Goel, et al., “Low-Power Multi-Camera Object Re-Identification using Hierarchical Neural Networks”, ACM/IEEE International Symposium on Low Power Electronics and Design 2021
3. Goel, et al., “Low-Power Object Counting with Hierarchical Neural Networks”, ACM/IEEE International Symposium on Low Power Electronics and Design 2020. Pages 163-168.
4. Goel, et al., “Efficient Computer Vision on Edge Devices with Pipeline-Parallel Hierarchical Neural Networks”, Asia and South Pacific Design Automation Conference 2022


Abhinav Goel
Purdue PhD 2022
now at Nvidia


Large
Neural
Network
Truck
Car
Cat
Dog


Small Neural
Network
vehicles
animals
Small Neural
Network
Truck
Car
Small Neural
Network
Cat
Dog

# Tree Modular Neural Network Architecture

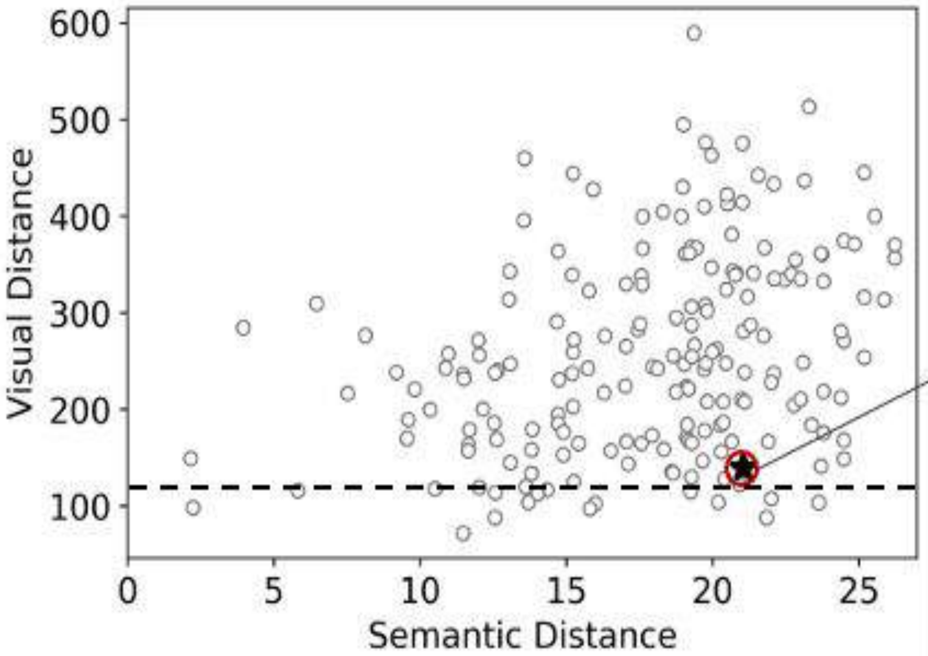


# Constructing The Hierarchy: Similarity Metrics

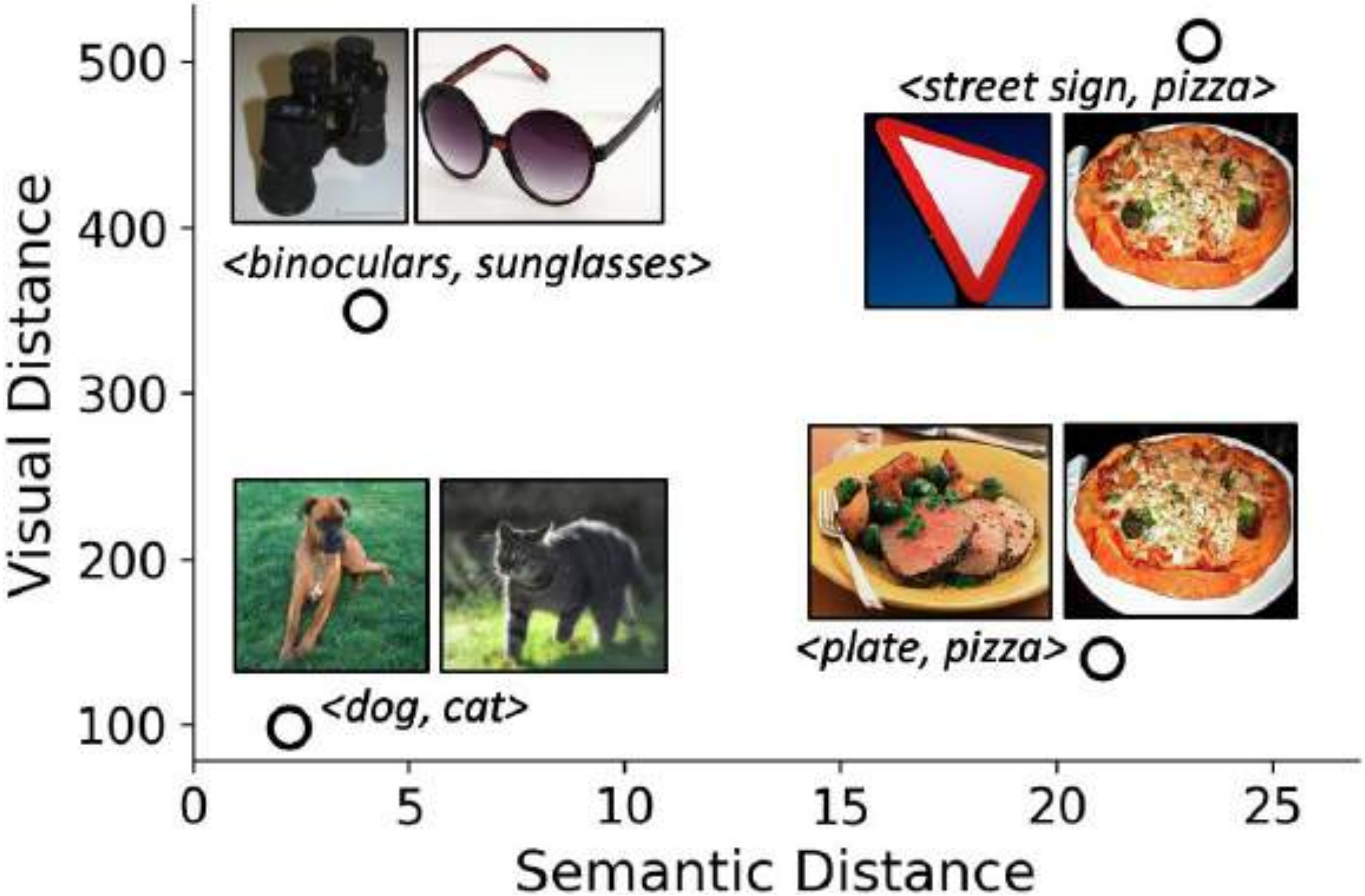



| Similarity | Example | | Properties |
|---|---|---|---|
| Visual | Birds | Airplanes | In sky, have wings |
| Semantic | Banana | Apple | Fruit |

# *Differences in Semantic and Visual Similarities*



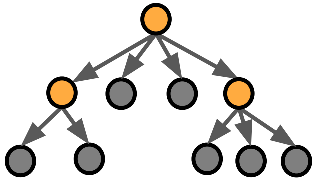

*Pizzas* and *Plates* are semantically distant, but are visually similar



Visual Distance
500
400
300
200
100
0
5
10
15
20
25
Semantic Distance
<binoculars, sunglasses>
<street sign, pizza>
<dog, cat>
<plate, pizza>

# Constructing the Hierarchy

Group Similar Categories

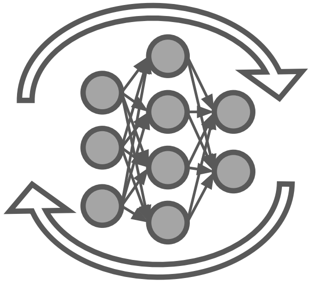


Select DNNs at Each Node

Large DNN

Small DNN

Train DNNs



# Finding Visual Similarities

Randomly initialized DNN: Softmax outputs are approximately $\frac{1}{10}$

| | Input | Airplane | Auto. | Bird | Cat | Deer | Dog | Frog | Horse | Ship | Truck |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Untrained | Cat | 0.094 | 0.108 | 0.097 | 0.098 | 0.097 | 0.103 | 0.092 | 0.092 | 0.110 | 0.105 |
| | Truck | 0.099 | 0.103 | 0.098 | 0.101 | 0.107 | 0.101 | 0.104 | 0.105 | 0.092 | 0.086 |
| Trained | Cat | 0.011 | 0.026 | **0.103** | **0.303** | **0.105** | **0.200** | **0.102** | **0.104** | 0.010 | 0.032 |
| | Truck | 0.060 | **0.193** | 0.018 | 0.024 | 0.017 | 0.022 | 0.012 | 0.041 | 0.060 | **0.550** |

10 categories in the CIFAR-10 dataset

Auto.

Truck

Cat



Yung-Hsiang Lu, Purdue University

# Finding Visual Similarities

Randomly initialized DNN: Softmax outputs are approximately $\frac{1}{10}$

| | Input | Airplane | Auto. | Bird | Cat | Deer | Dog | Frog | Horse | Ship | Truck |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Untrained | Cat | 0.094 | 0.108 | 0.097 | 0.098 | 0.097 | 0.103 | 0.092 | 0.092 | 0.110 | 0.105 |
| | Truck | 0.099 | 0.103 | 0.098 | 0.101 | 0.107 | 0.101 | 0.104 | 0.105 | 0.092 | 0.086 |
| Trained | Cat | 0.011 | 0.026 | **0.103** | **0.303** | **0.105** | **0.200** | **0.102** | **0.104** | 0.010 | 0.032 |
| | Truck | 0.060 | **0.193** | 0.018 | 0.024 | 0.017 | 0.022 | 0.012 | 0.041 | 0.060 | **0.550** |

10 categories in the CIFAR-10 dataset

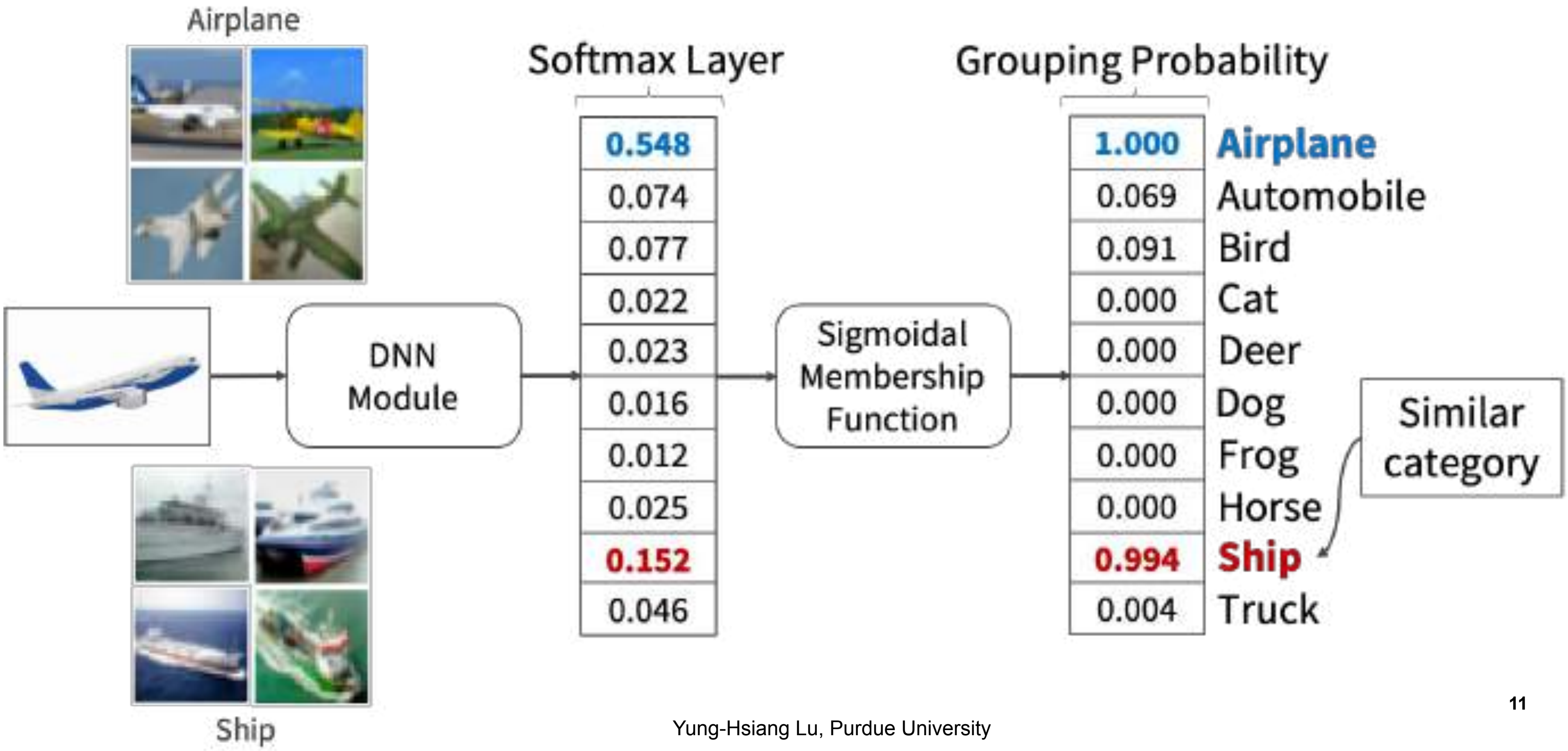
Trained DNN: Softmax outputs for certain categories increases after training

Auto.

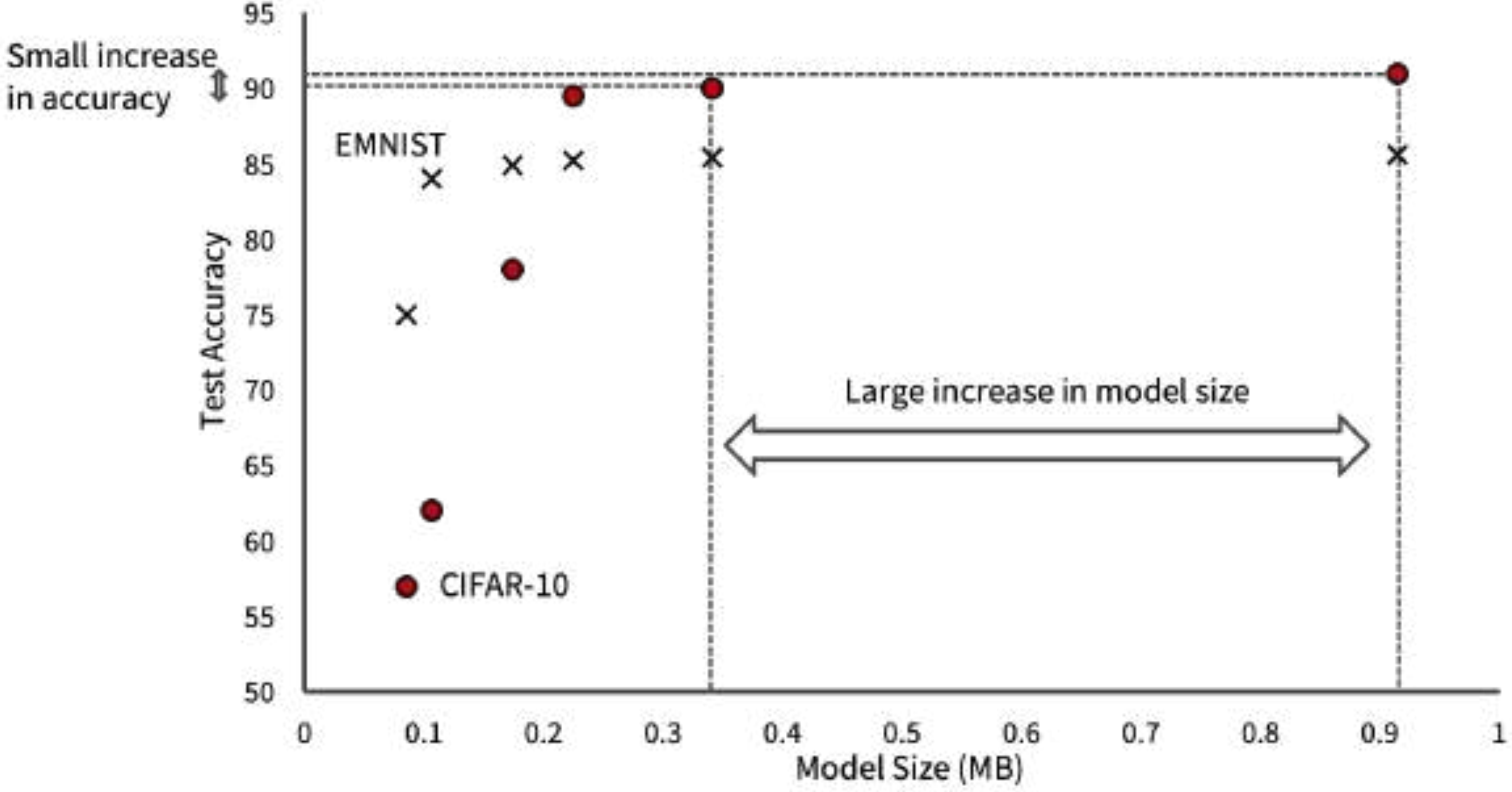
Truck

Cat

# Averaged Softmax Likelihood

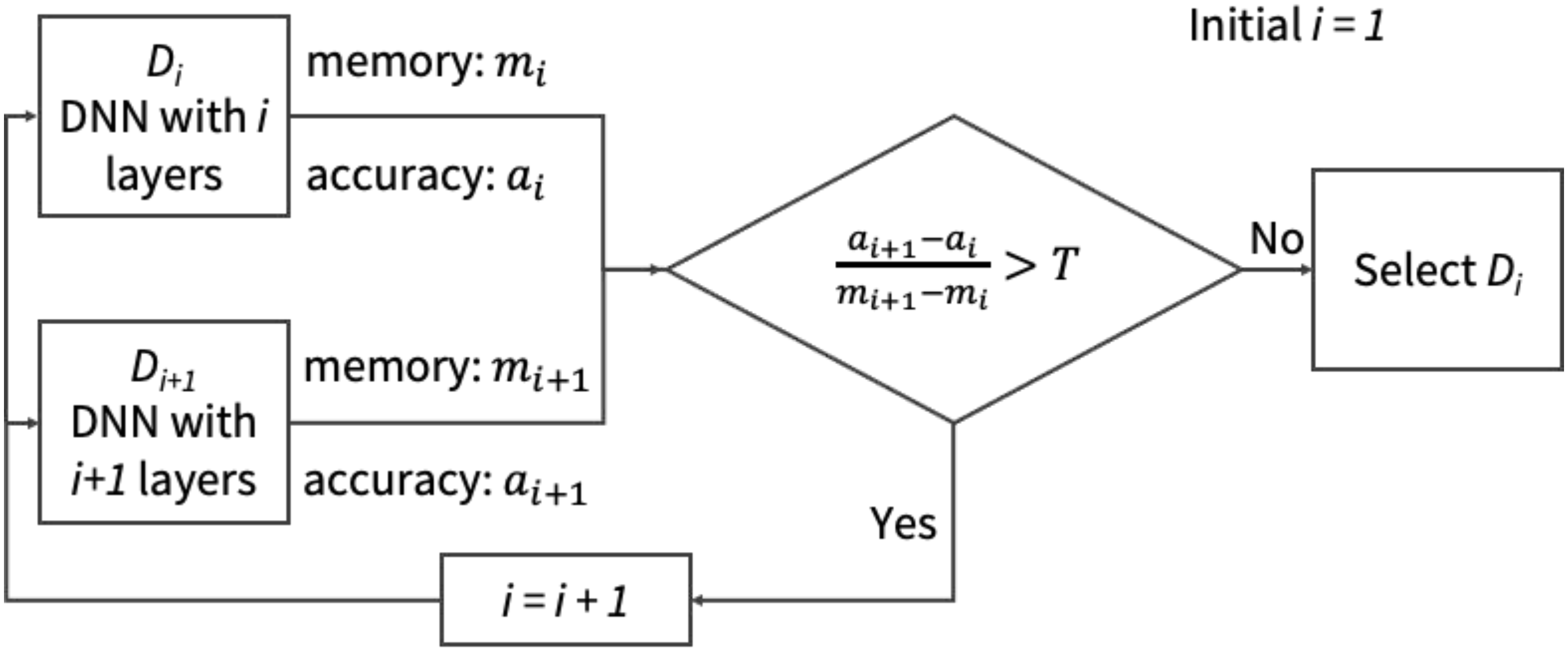


# Neural Architecture Search To Find Neural Network Size

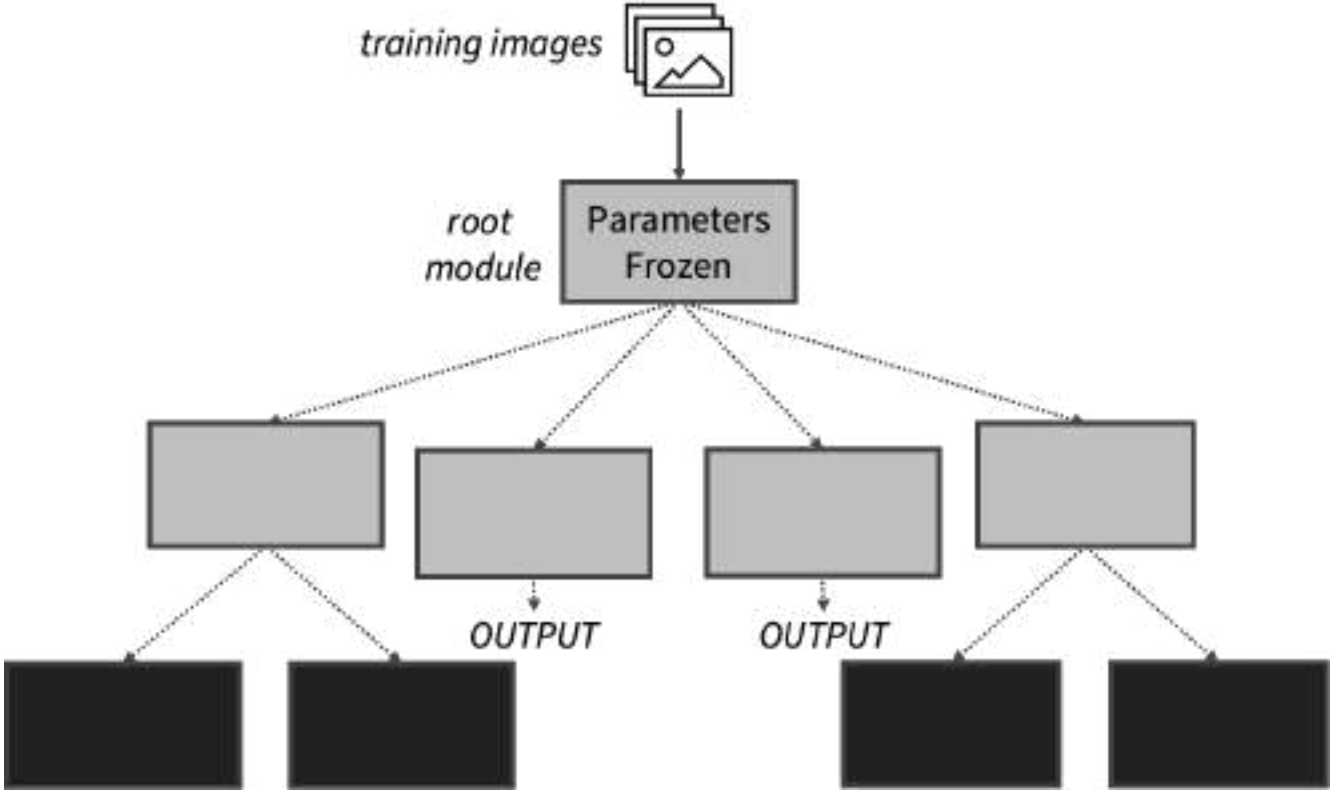



S. Bianco, et al. "Benchmark Analysis of Representative Deep Neural Network Architectures," in *IEEE Access*, vol. 6, 2018.

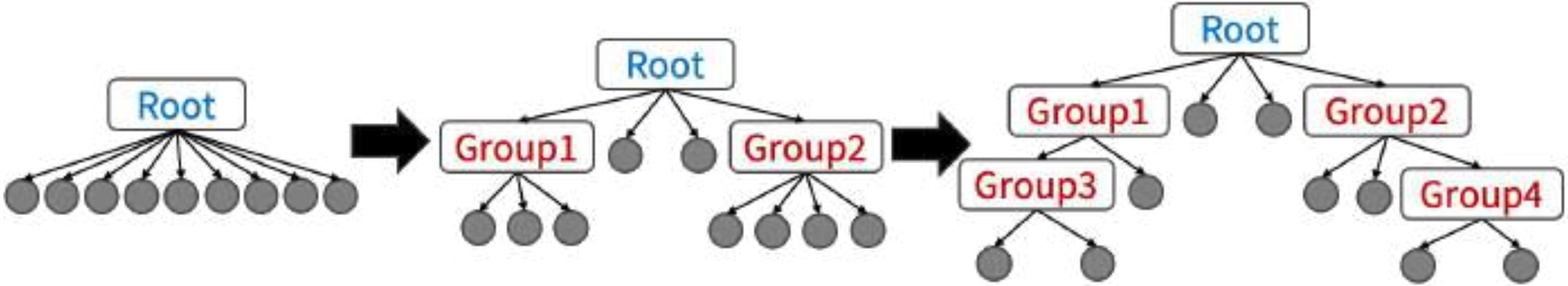
# *Selecting DNN Sizes*

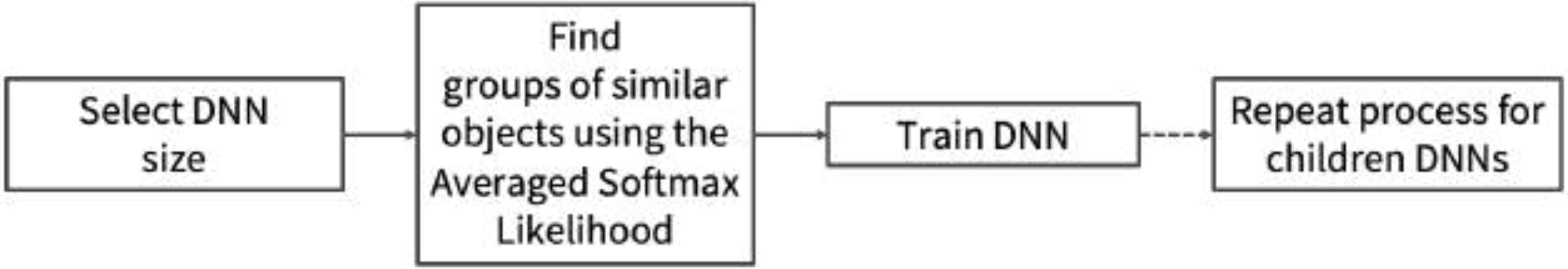


# Training Tree Modular Neural Networks

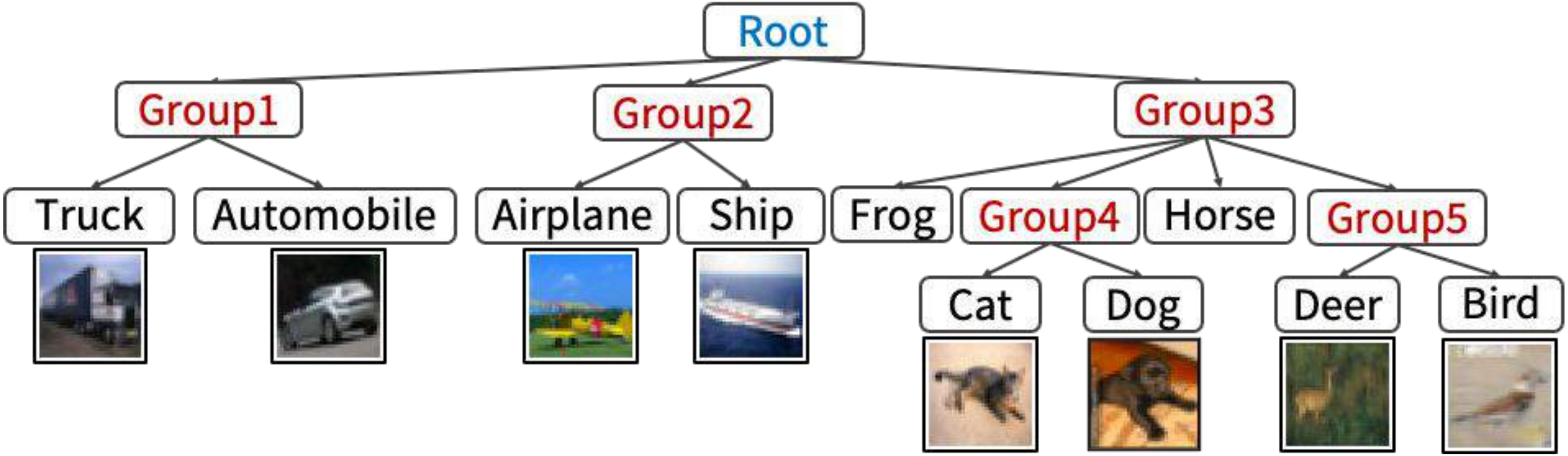


# Tree Modular Neural Network Construction



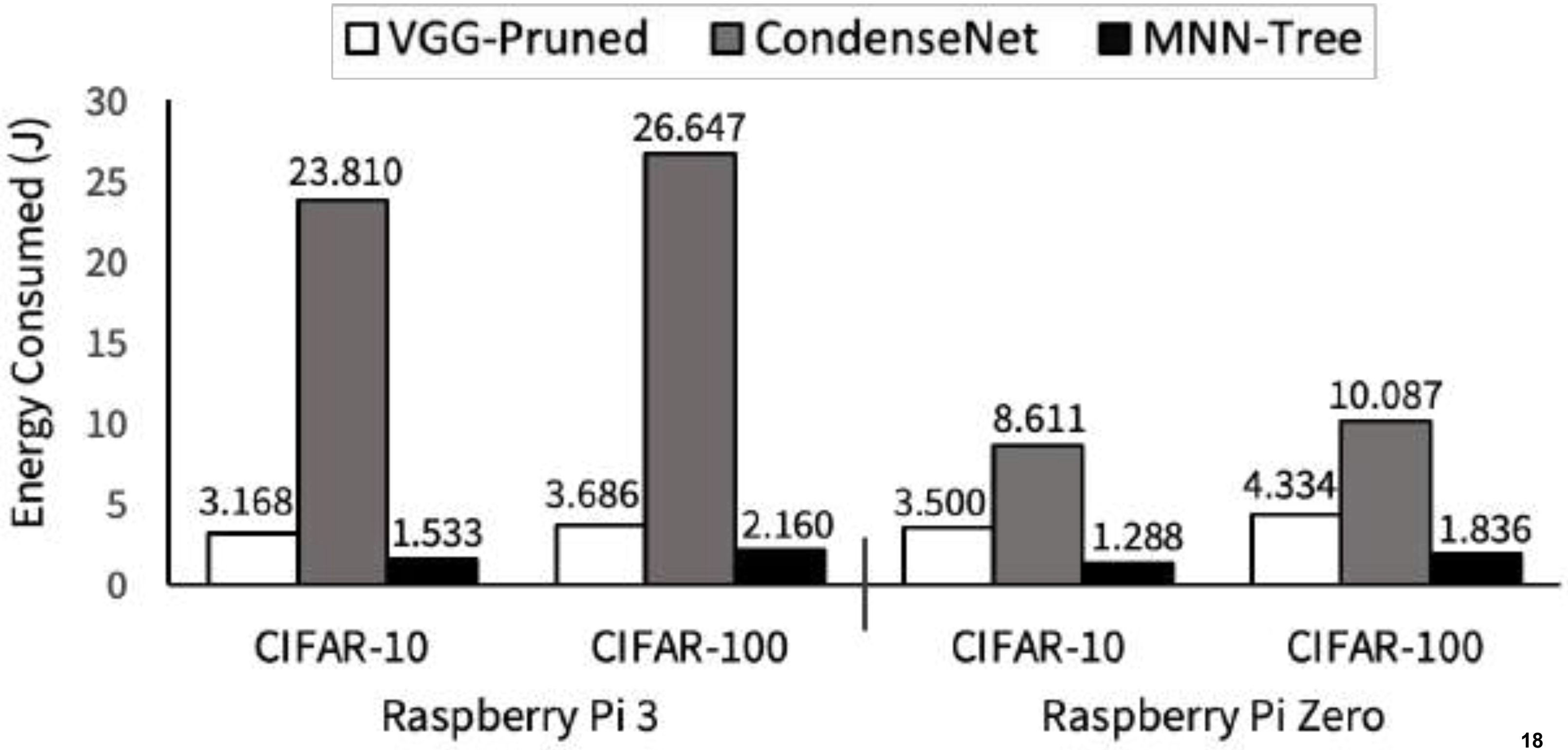



# Hierarchy Constructed For CIFAR-10 Dataset

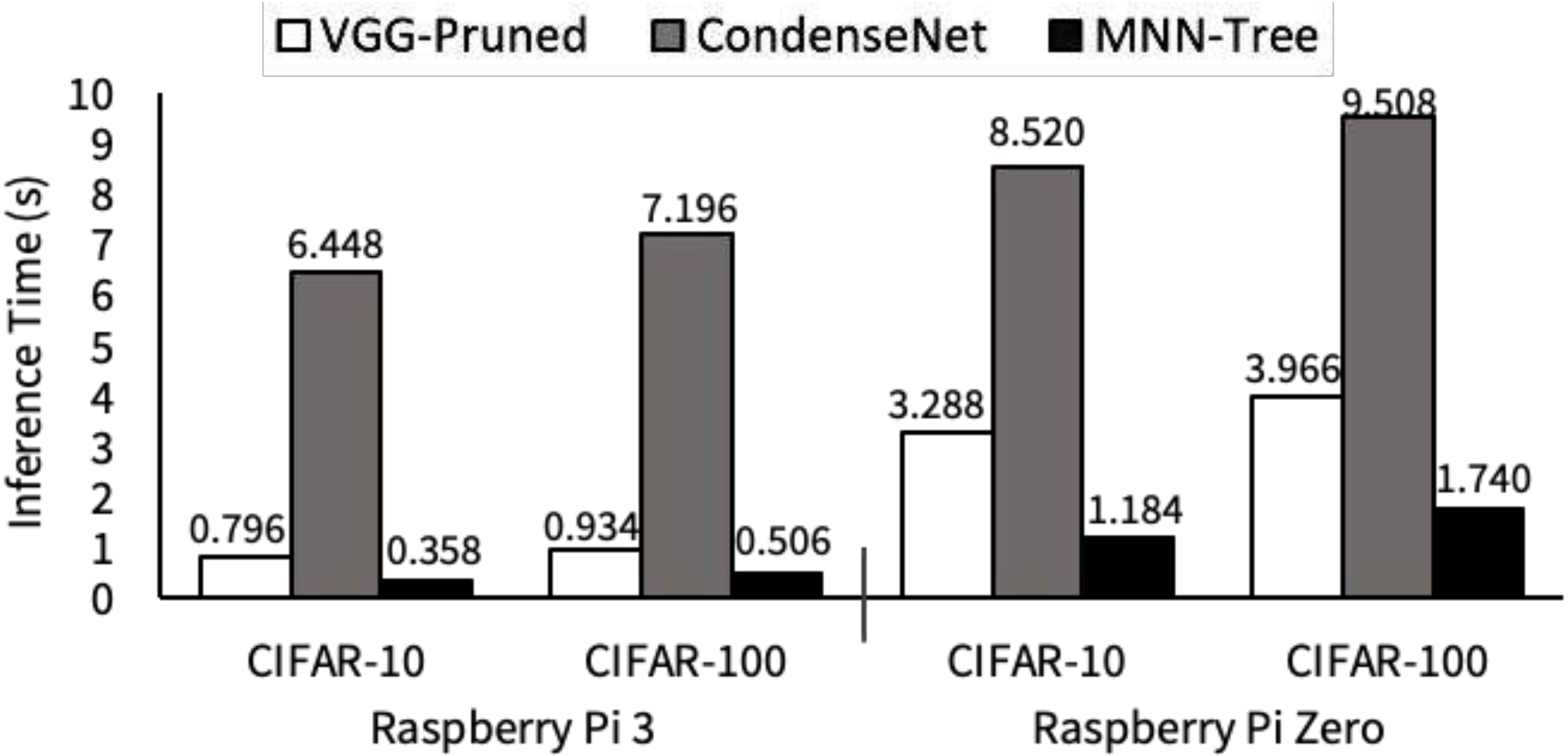


| Dataset | Technique | Model Size (KB) | FLOPs | Test Error |
|---|---|---|---|---|
| CIFAR-10 | VGG 16 | 28,200 | 206 M | 0.066 |
| | DenseNet | 102,000 | 9,388 M | 0.070 |
| | CondenseNet | 43,000 | 1,024 M | 0.034 |
| | MobileNet v2 | 8,800 | 100 M | 0.060 |
| | **This Method** | **806** | **28 M** | **0.079** |
| ImageNet | VGG 16 | 528,120 | 15,300 M | 0.295 |
| | ResNet-34 | 84,100 | 3,640 M | 0.276 |
| | DenseNet | 32,400 | 3,000 M | 0.230 |
| | SqueezeNet | 5,330 | 837 M | 0.425 |
| | **This Method** | **2,515** | **713 M** | **0.313** |

# Improvement in Energy Consumption

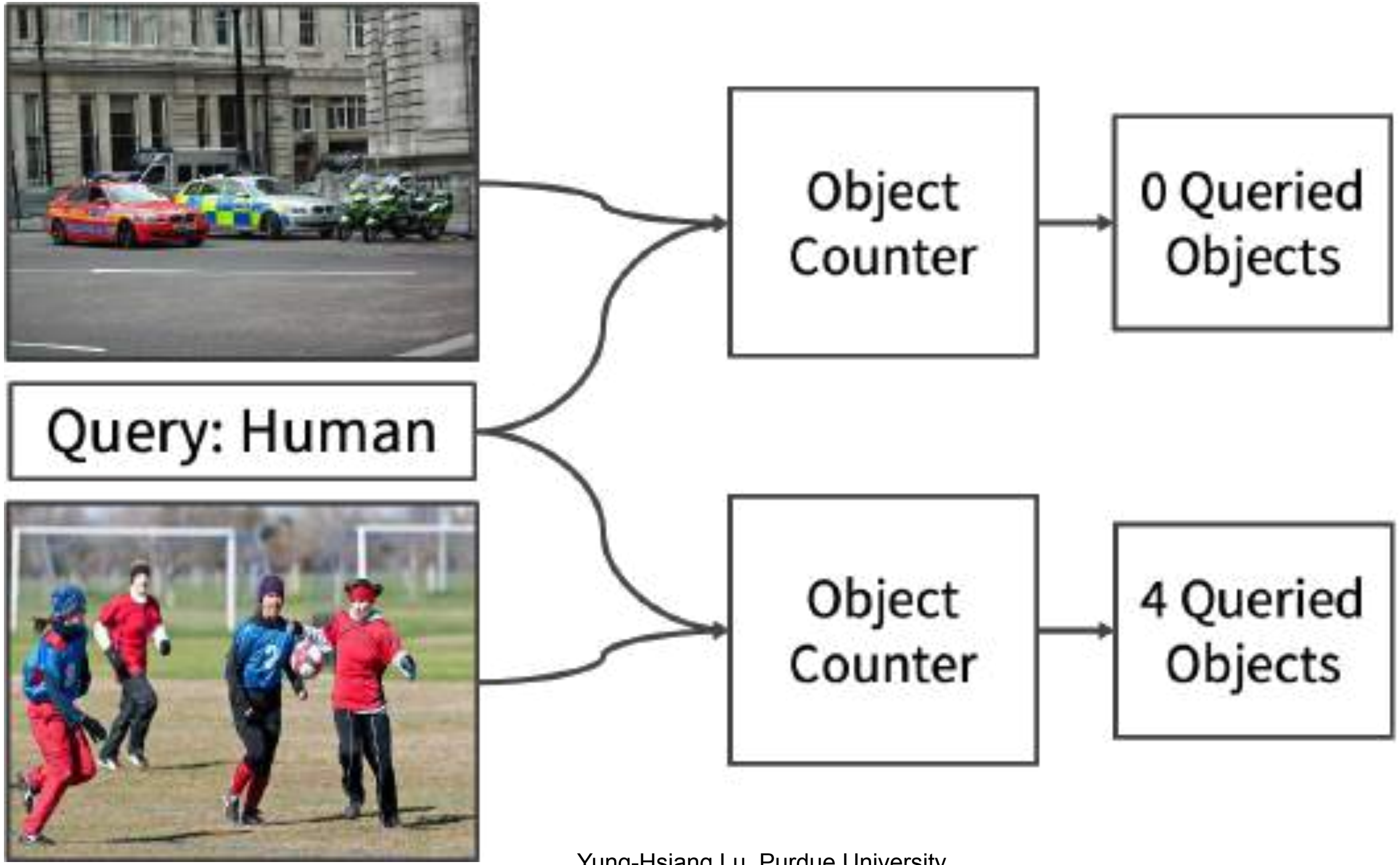


# Improvement in Inference Time

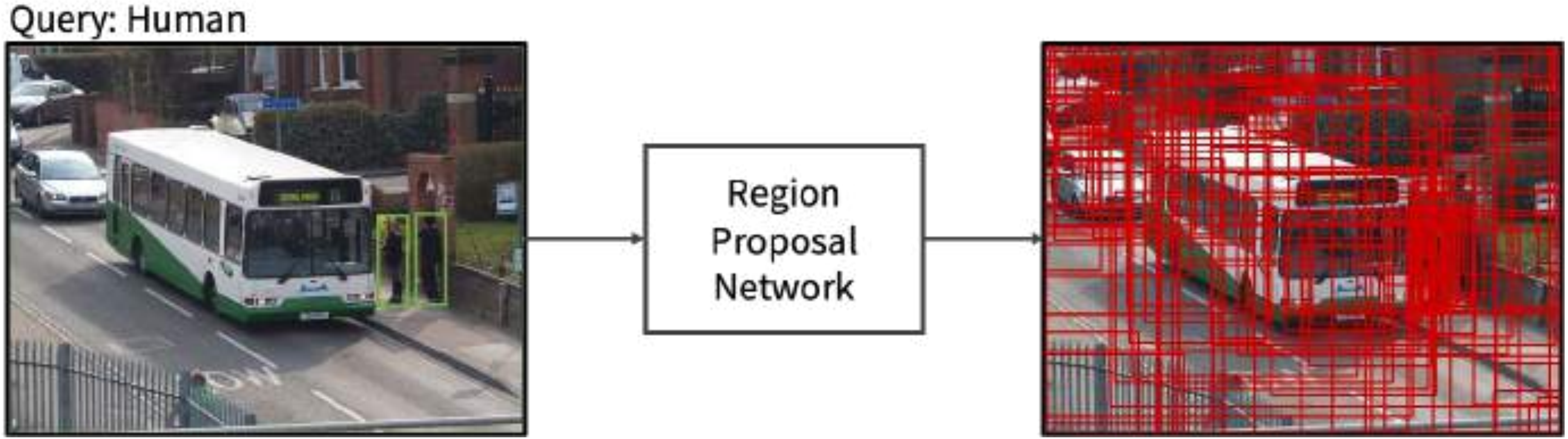


# Tree Modular Networks for Object Counting

# *Object Counting*



# *Existing Object Counting Techniques*

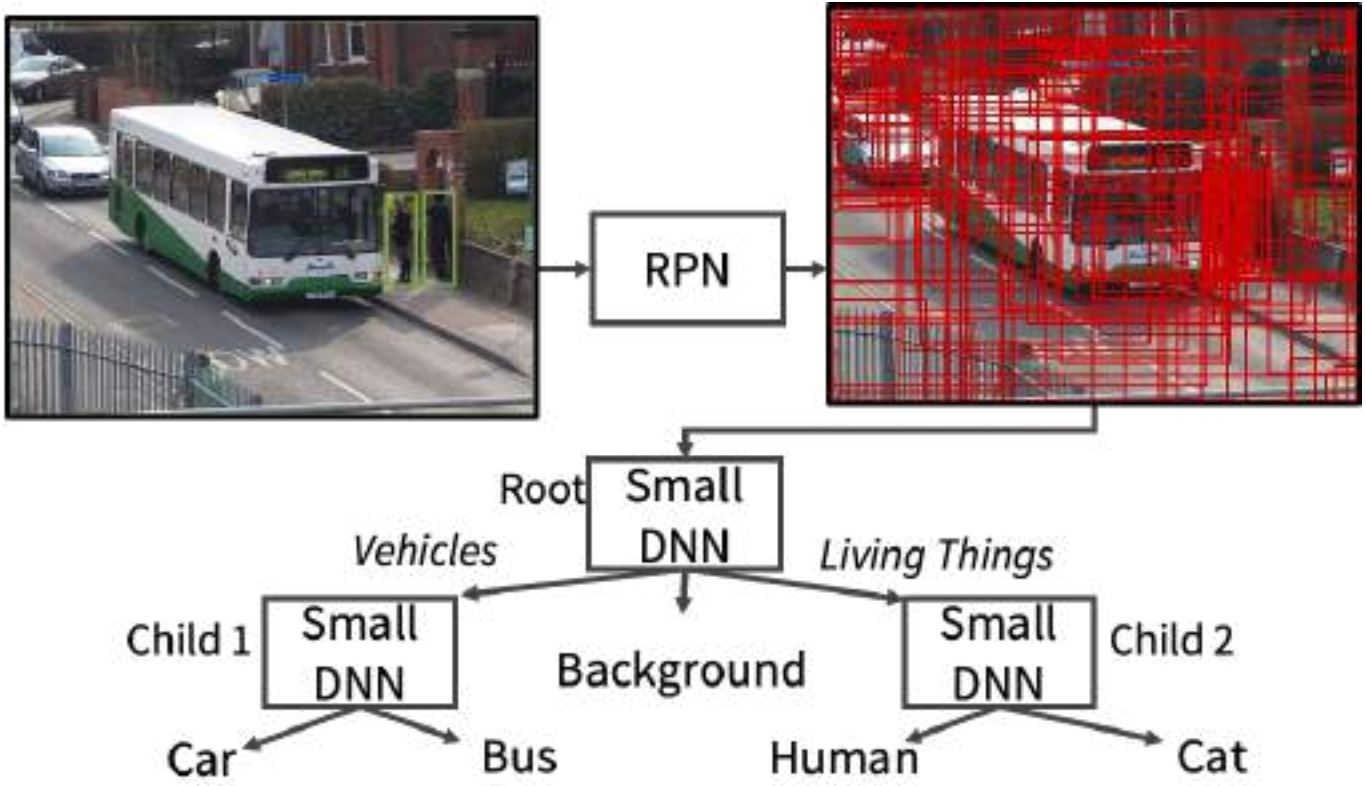



Each region (red box) is processed by a large DNN.

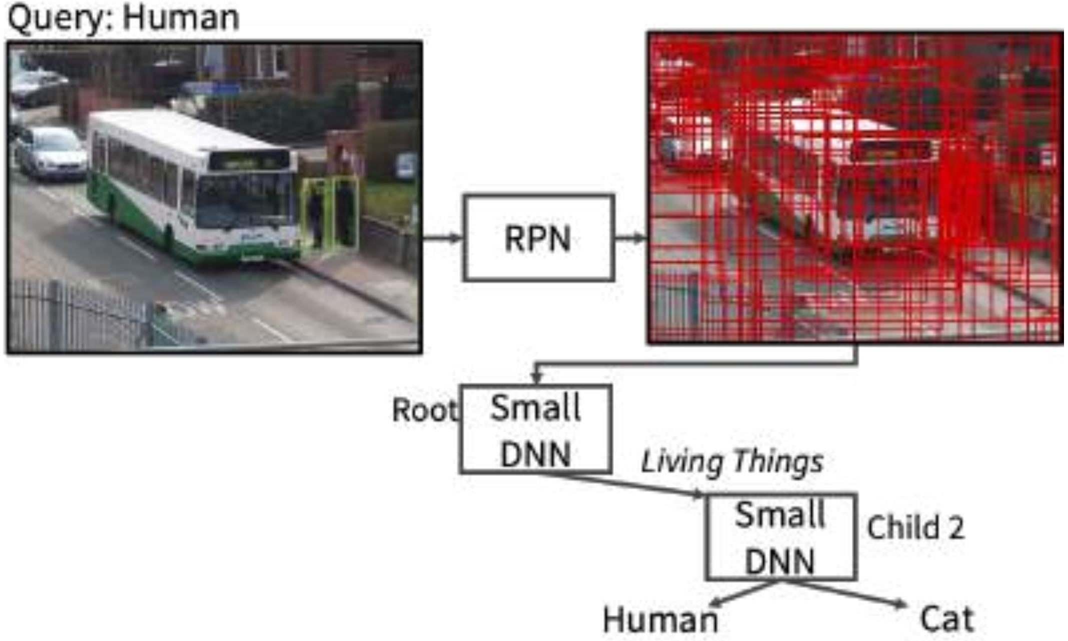
Identify the object in each region.

Return the number of regions that contain humans.

# Tree Modular Neural Networks Increase Efficiency



# *Identify Root-Leaf Path for Query*



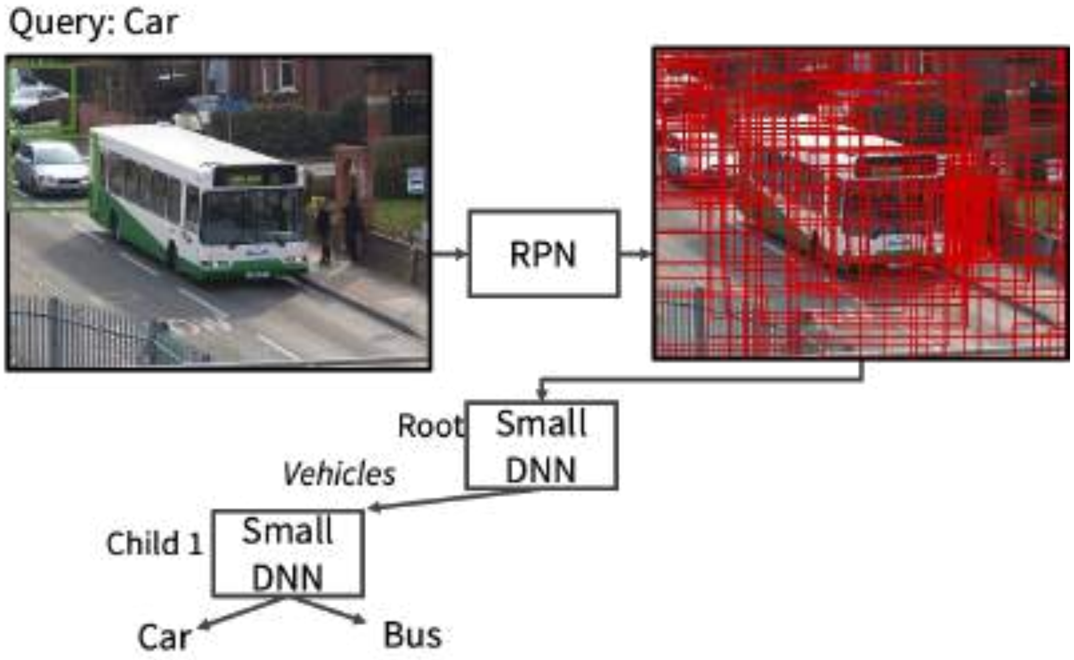

Each region (red box) is processed by a small root DNN.

If region contains a living thing, process further; otherwise discard region.



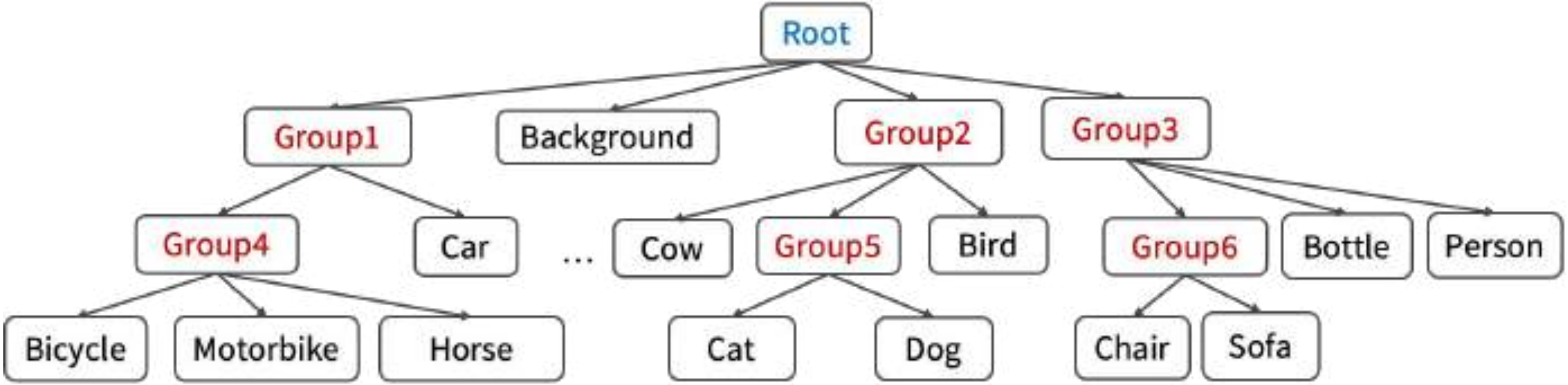
Yung-Hsiang Lu, Purdue University

# Different Paths for Different Queries




If region contains a vehicle, process further; otherwise discard region.

# Hierarchy Constructed for PASCAL-VOC Dataset



# Comparison with Existing Techniques

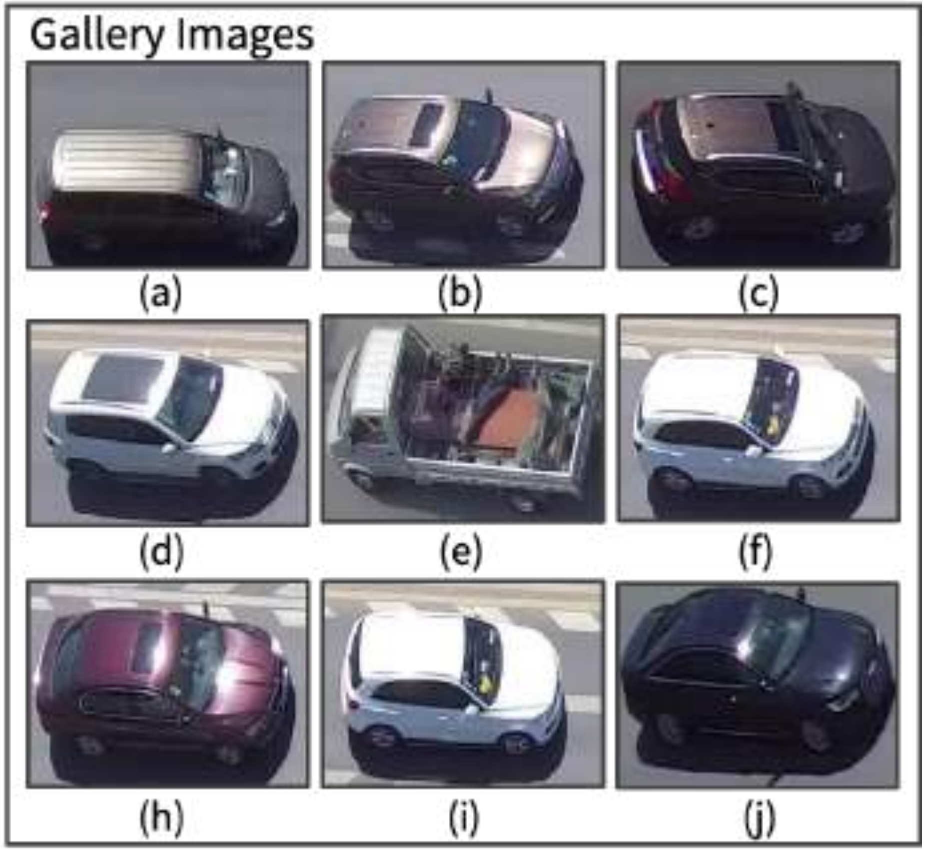
| Dataset | Technique | Model Size (MB) | FLOPs | Test RMSE |
|---|---|---|---|---|
| Pascal VOC | Faster RCNN | 1,100 | 440 B | 1.38 |
| | YOLO v3 | 248 | 141 B | 1.61 |
| | Tiny YOLO | 55 | 5 B | 2.32 |
| | LC-FCN | 194 | 156 B | 1.20 |
| | Semantic Tree | 16 | 41 B | 2.56 |
| | This Method | 16 | 42 B | 1.80 |
| COCO | Faster RCNN | 1,100 | 440 B | 1.99 |
| | YOLO v3 | 249 | 141 B | 2.07 |
| | Tiny YOLO | 56 | 5 B | 3.01 |
| | Semantic Tree | 20 | 44 B | 3.51 |
| | This Method | 19 | 44 B | 2.24 |

| Metric | YOLOv3 | Tiny YOLO | Proposed Method |
|---|---|---|---|
| Inference Time (sec/img) | 22.33 | 1.25 | 1.02 |
| Energy Consumption (J/img) | 162.00 | 9.90 | 8.10 |

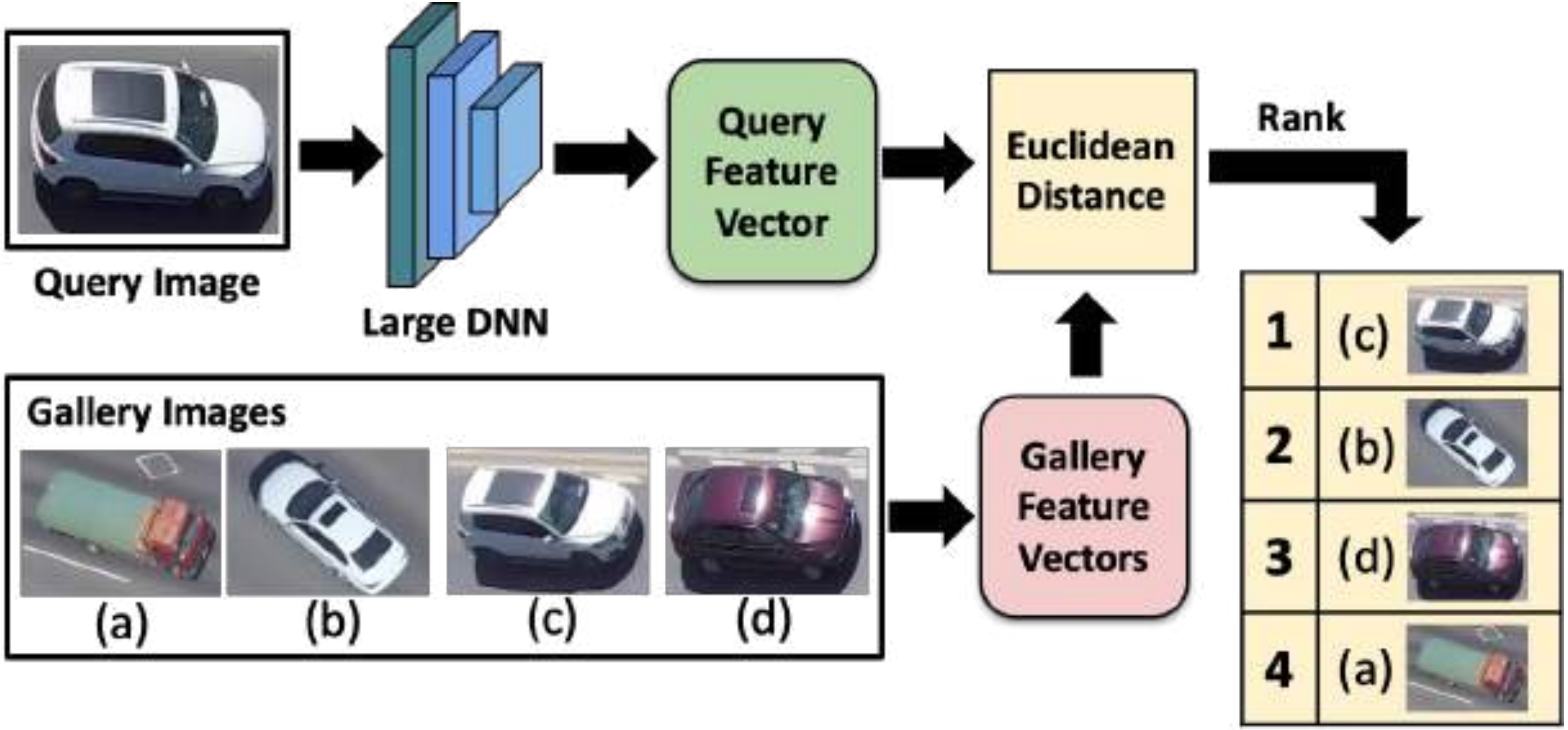
RMSE: Root Mean Squared Error

# Object Re-Identification

# Object Re-Identification

# Existing Object ReID Techniques



# Efficient Object ReID with Attribute Labels

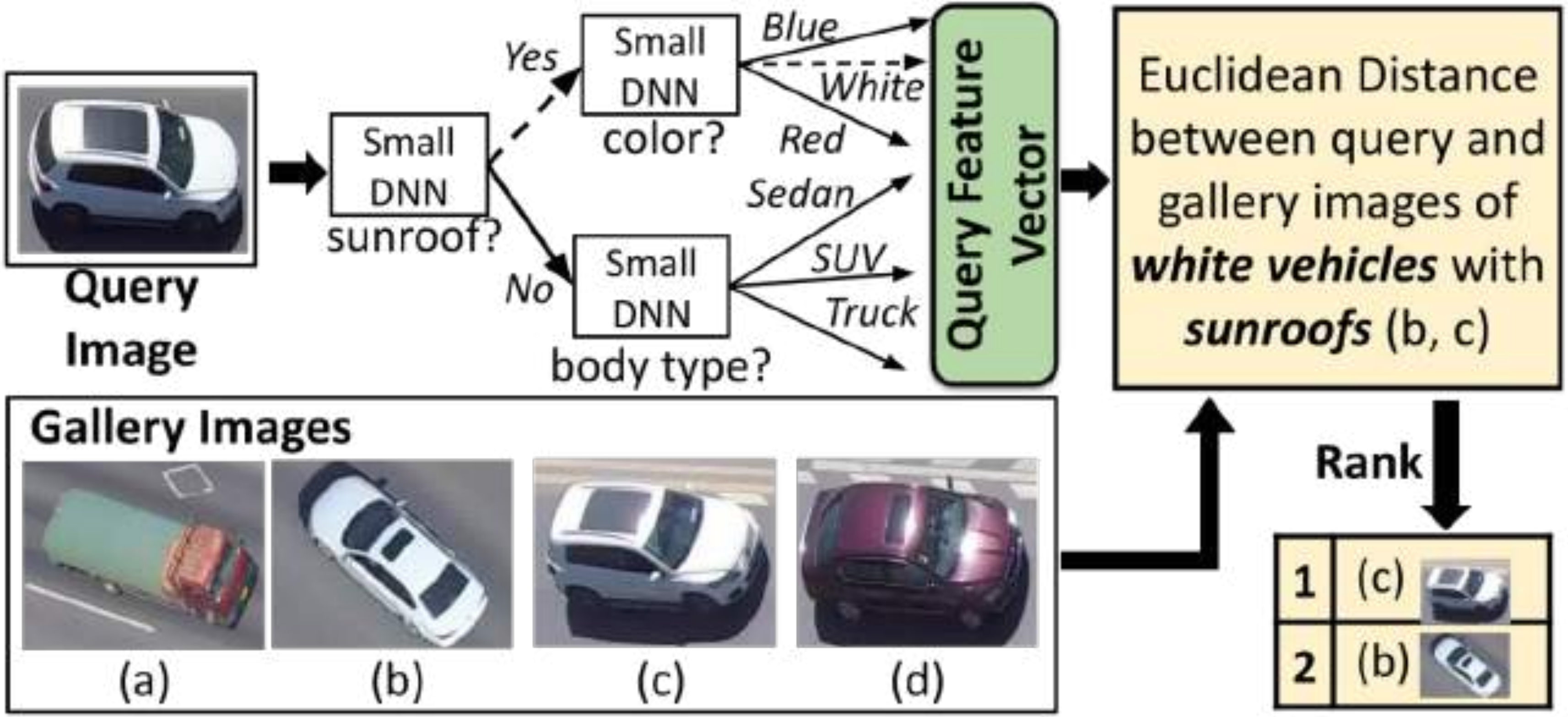
Query Image

A **white** car with a **sunroof**

Gallery Images

(a) (b) (c) (d)

# Tree Modular Neural Networks for Object ReID



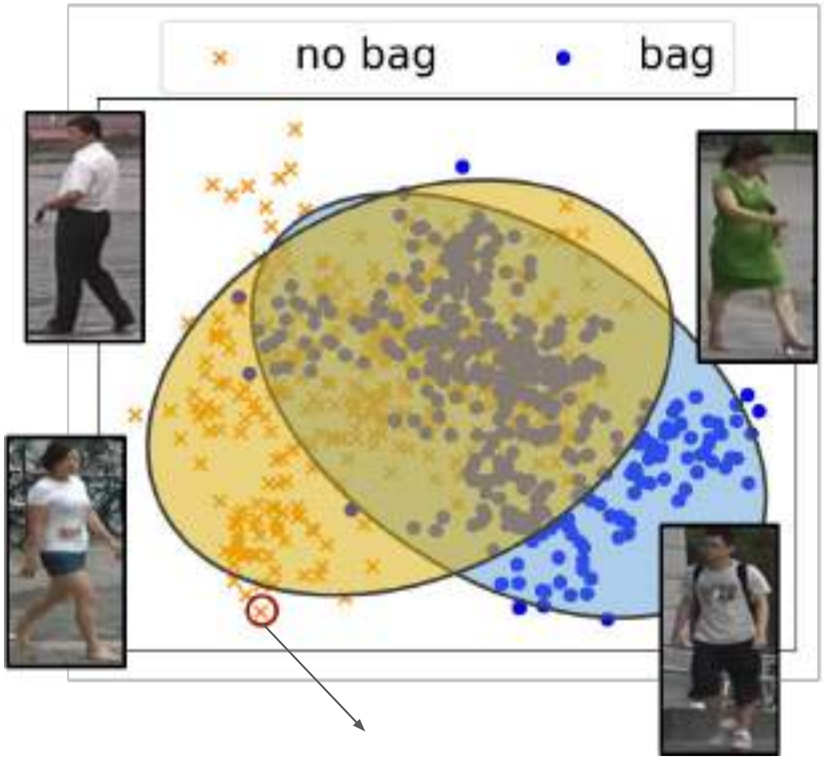
# *Neural Network Construction for ReID*

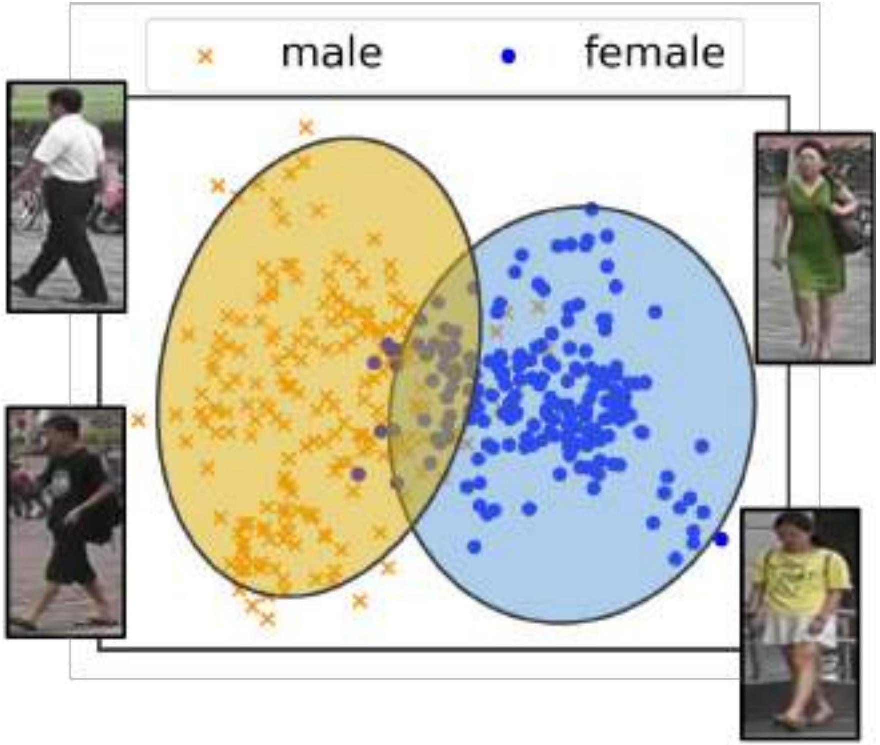
- Which attributes should be identified? Obtain attribute correlations.
- What order should they be identified? Quantify the difficulty of attribute identifications.



Yung-Hsiang Lu, Purdue University

# Difficulty of Identifications: In Market 1501 dataset



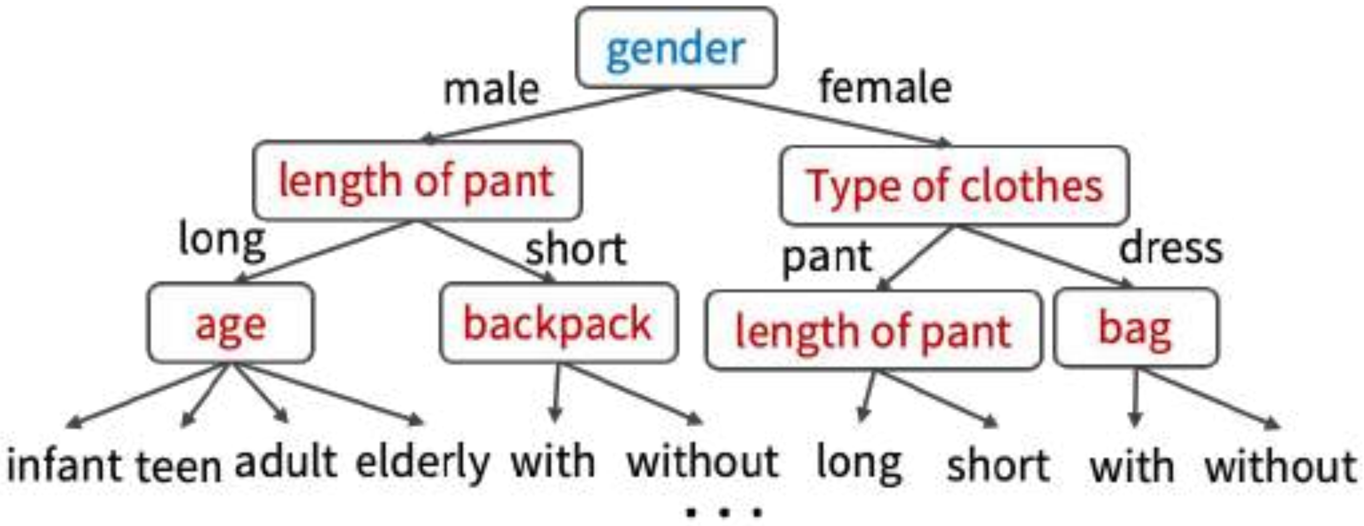




Feature vector of an image obtained
with a pre-trained DNN

# Attribute Correlations

| | Male | Female | Long hair | Backpack | Dress | Short sleeve | Teen | Shoulder bag | Long pant |
|---|---|---|---|---|---|---|---|---|---|
| Male | 1.00 | 0.00 | 0.01 | 0.30 | 0.00 | 0.97 | 0.73 | 0.10 | 0.44 |
| Female | 0.00 | 1.00 | 0.75 | 0.22 | 0.34 | 0.91 | 0.79 | 0.13 | 0.33 |
| Long hair | 0.02 | 0.98 | 1.00 | 0.23 | 0.37 | 0.90 | 0.85 | 0.13 | 0.31 |
| Backpack | 0.64 | 0.36 | 0.28 | 1.00 | 0.11 | 0.95 | 0.96 | 0.00 | 0.31 |
| Dress | 0.00 | 1.00 | 0.83 | 0.20 | 1.00 | 0.90 | 0.78 | 0.14 | 0.14 |
| Short sleeve | 0.59 | 0.41 | 0.31 | 0.27 | 0.14 | 1.00 | 0.76 | 0.09 | 0.39 |
| Teenager | 0.55 | 0.45 | 0.37 | 0.34 | 0.15 | 0.95 | 1.00 | 0.08 | 0.30 |
| Shoulder bag | 0.35 | 0.65 | 0.48 | 0.00 | 0.82 | 0.94 | 0.74 | 1.00 | 0.64 |
| Long pant | 0.65 | 0.35 | 0.26 | 0.21 | 0.05 | 0.94 | 0.57 | 0.09 | 1.00 |

# Hierarchy Constructed for Market-1501



# *Experimental Results*

| Dataset | Technique | Model Size (MB) | FLOPs | Rank-1 | mAP |
|---|---|---|---|---|---|
| VRAI | RAM-VGG | 528 | 15,483 M | 0.720 | 0.573 |
| | Multi-Task | 351 | 11,172 M | 0.803 | 0.786 |
| | DenseNet | 77 | 4,340 M | 0.671 | 0.700 |
| | Random Tree | 25 | 2,026 M | 0.631 | 0.585 |
| | **Proposed Method** | 14 | 1,082 M | 0.781 | 0.737 |
| Market-1501 | Pyramidal | 184 | 9,757 M | 0.928 | 0.821 |
| | Auto ReID | 55 | 2,050 M | 0.938 | 0.834 |
| | DG-Net | 101 | 4,029 M | 0.896 | 0.745 |
| | Random Tree | 257 | 1,736 | 0.788 | 0.535 |
| | **Proposed Method** | 14 | 808 M | 0.885 | 0.699 |

| | Raspberry Pi 3 B+ | | NVIDIA Jetson Nano | |
|---|---|---|---|---|
| | Query Time (s) | Energy (J) | Query Time (s) | Energy (J) |
| DenseNet | - | - | 2.55 | 18.18 |
| DARE | 11.41 | 55.83 | 1.09 | 7.92 |
| Random Tree | 4.39 | 19.83 | 0.77 | 5.72 |
| **Proposed Method** | 2.13 | 10.33 | 0.35 | 2.70 |

# Conclusions

- Tree Modular Neural Networks perform efficient image classification.
- Can be extended to object counting and re-identification applications.
- Accuracy-efficiency tradeoff exist.

SOURCE CODE:

https://github.com/abhinavgoel95/Modular_Neural_Networks

https://github.com/abhinavgoel95/Object-ReID-Hierarchical-Neural-Networks