Understanding Places Using Ground-Level and Overhead Views

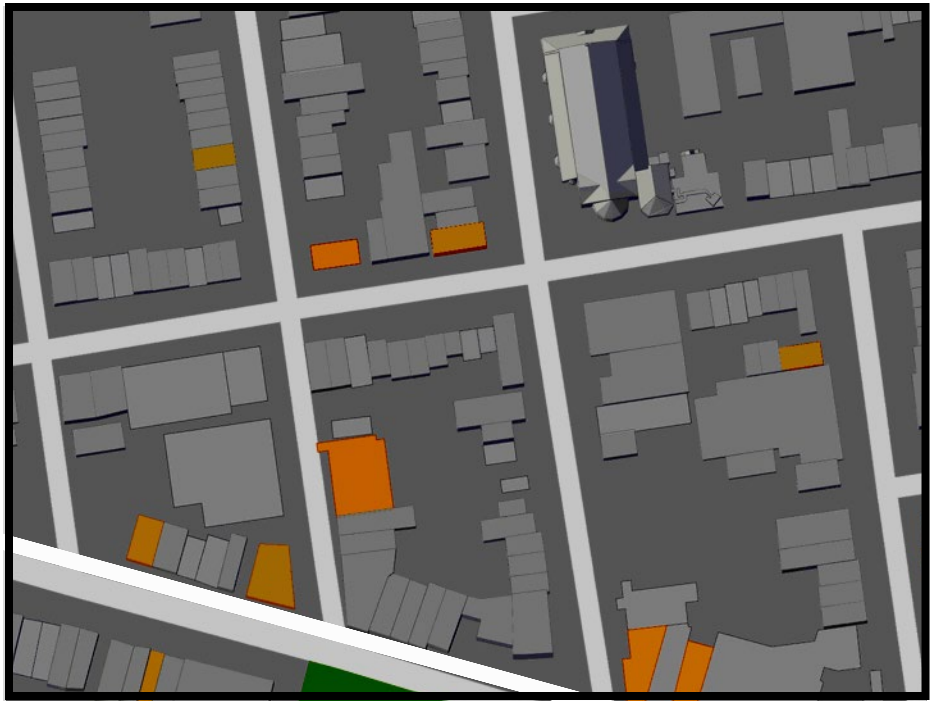
Nathan Jacobs

Department of Computer Science

University of Kentucky

Goal: Automatically describe the world in rich detail, using all available data.

Making maps using images



P(attribute | location, time)

Goal: Automatically describe the world in rich detail, using all available data.

Making maps using images

image = Camera(location, time)

P(attribute | image)

Two Parts

1. The ongoing revolution in automated perception.

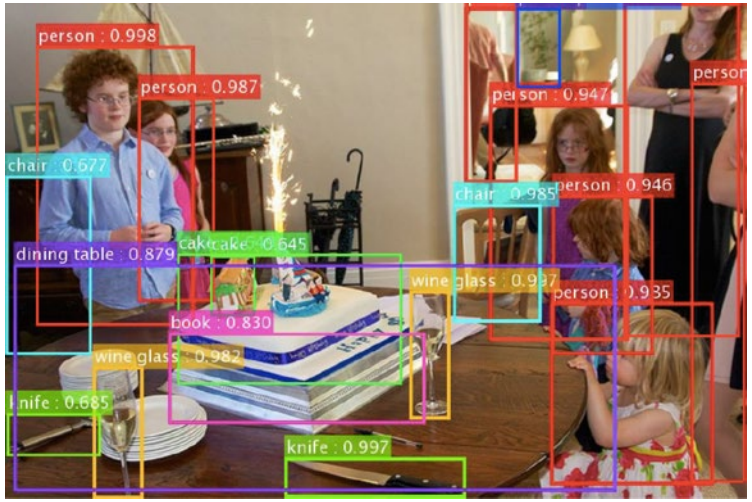
2. My work on image-driven mapping.

Computer Vision is finally useful!

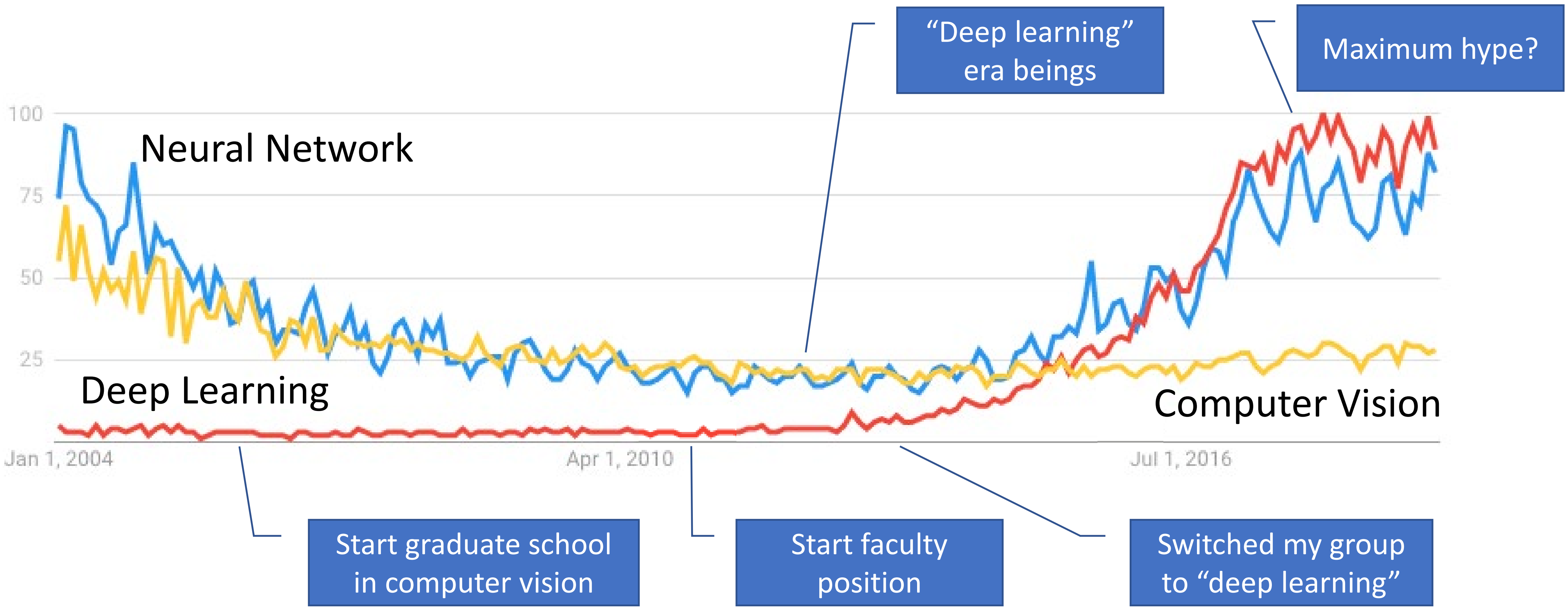
Image/Scene Classification

Image Segmentation

Object Detection



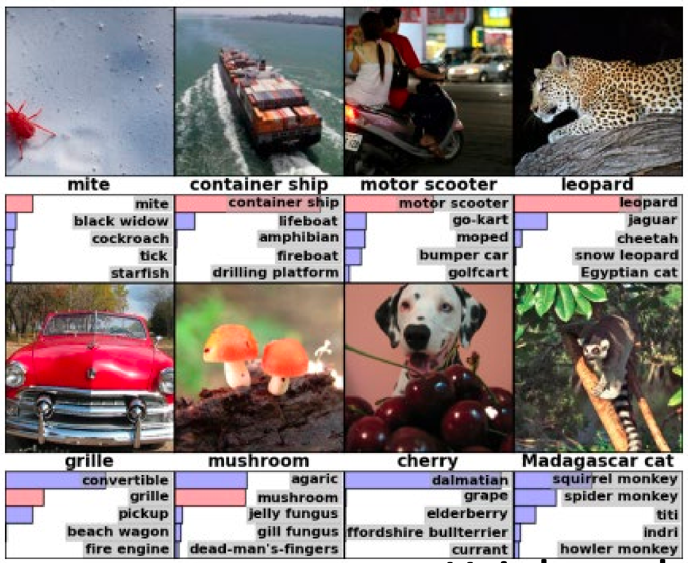
Deep (machine) learning is the reason.



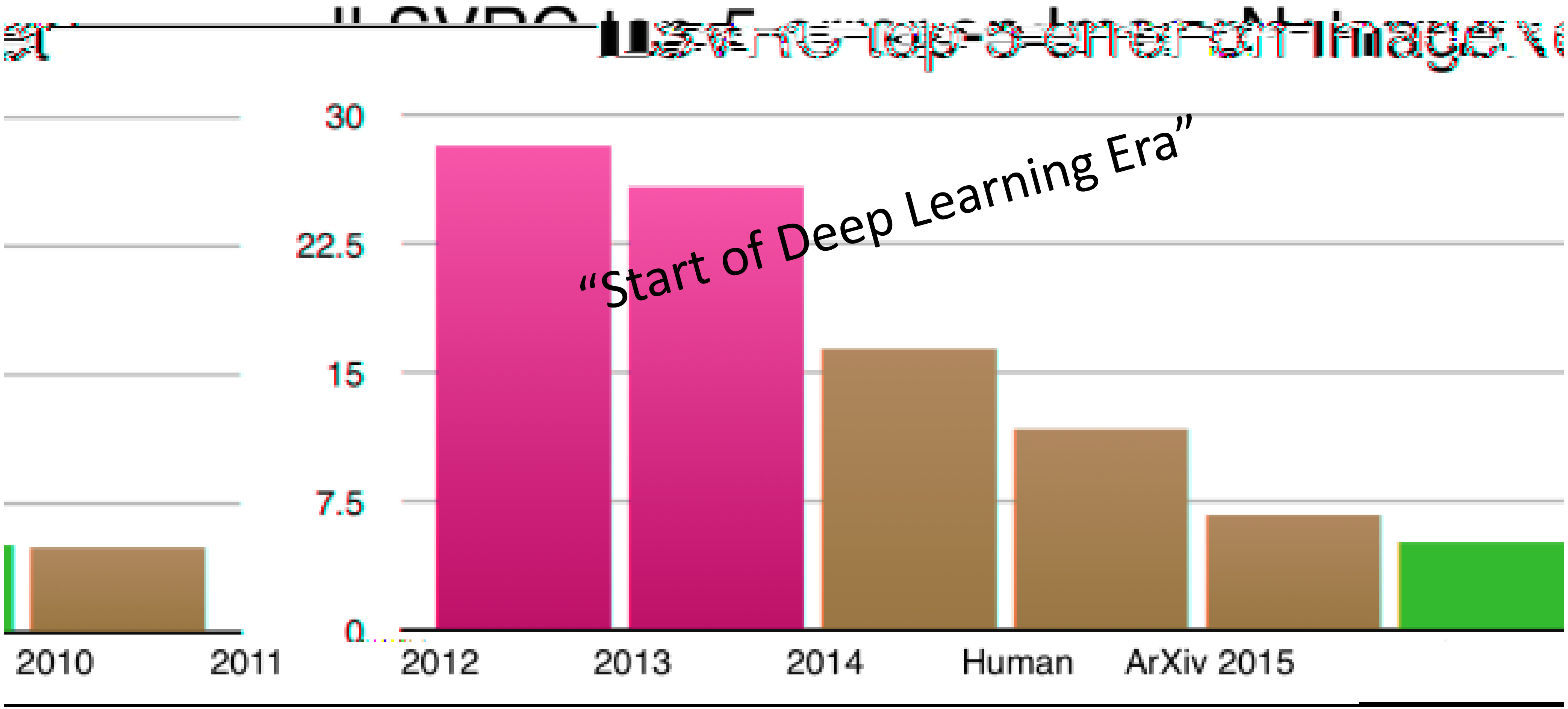
Source: Google Trends

ImageNet Large Scale Visual Recognition Challenge

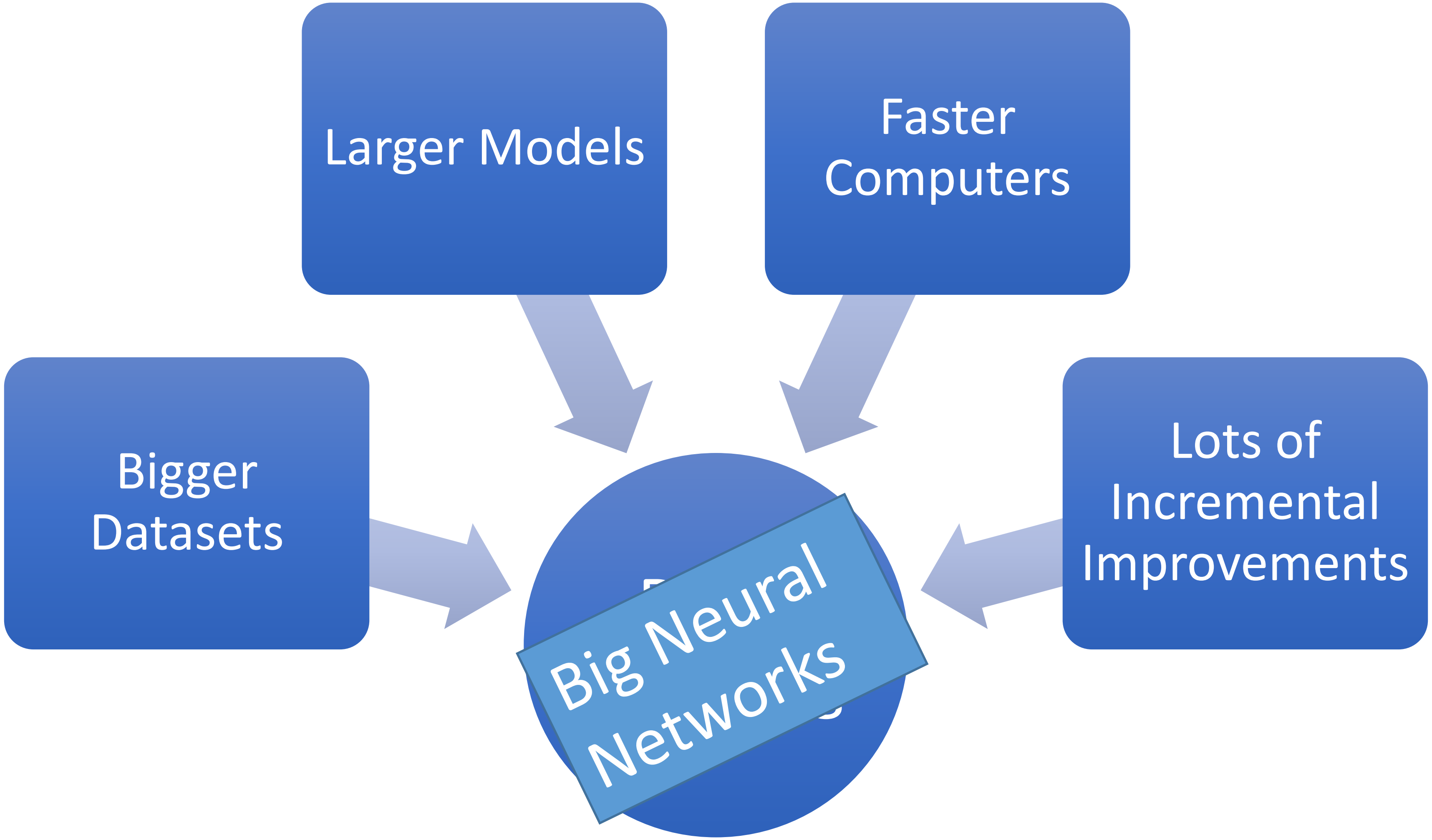
- Task: Classify an image into one of 1000 categories
  - guacamole
  - oxcart
  - cradle
  - australian terrier
  - trimaran
  - submarine
  - …
- 1,200,000 training images
- 100,000 test images



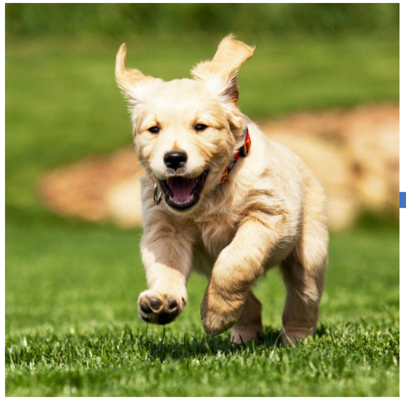
Krizhevsky



https://srconstantin.wordpress.com/2017/01/28/performance-trends-in-ai/


Larger Models
Faster Computers
Bigger Datasets
Lots of Incremental Improvements
Big Neural Networks

Non-ML

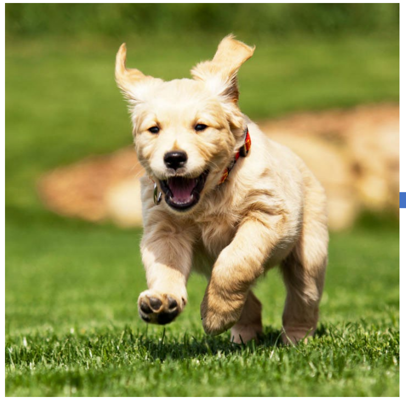


(very complicated) human written code

linear model

“animal”

Shallow ML

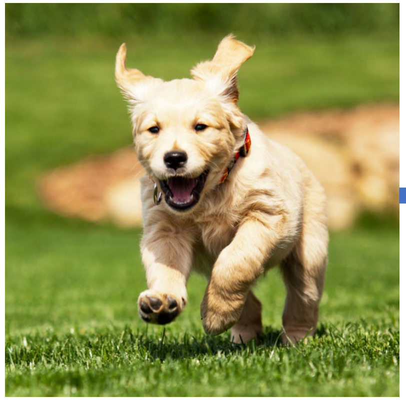


(simpler) human written code

machine learning model

“dog”

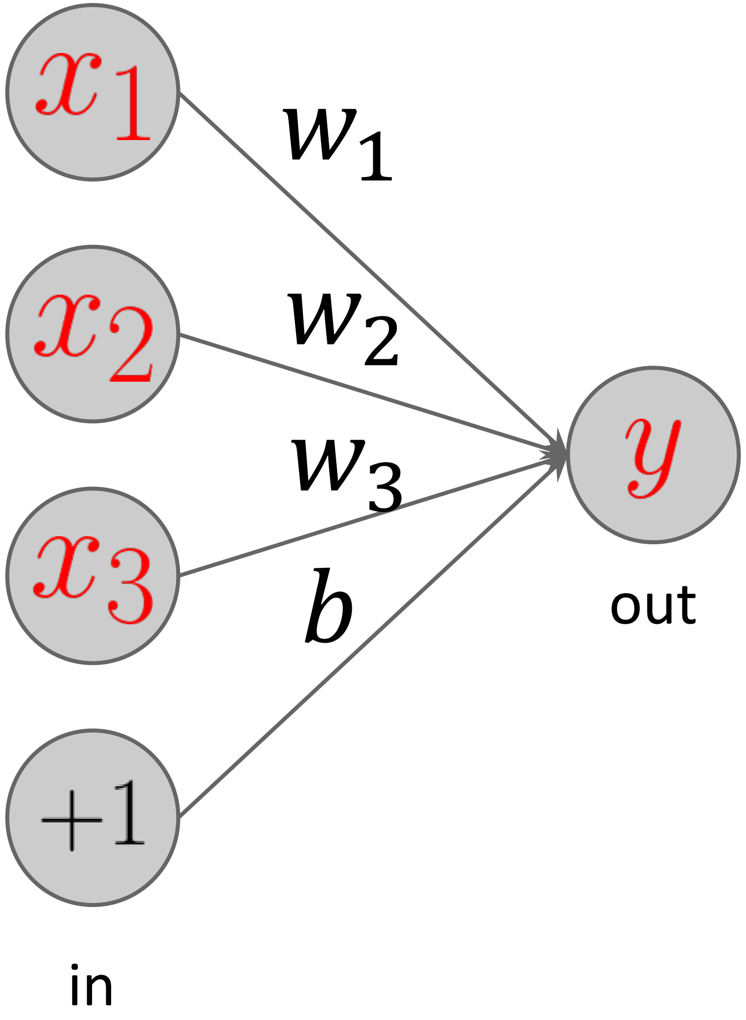
Deep ML

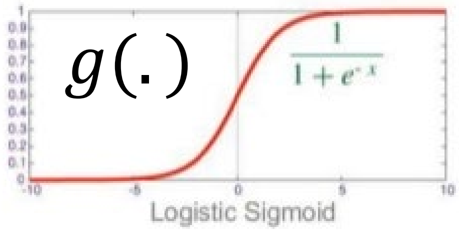


deep learning model (e.g., convolutional neural network)

“golden retriever puppy”

Deep Convolutional Neural Networks



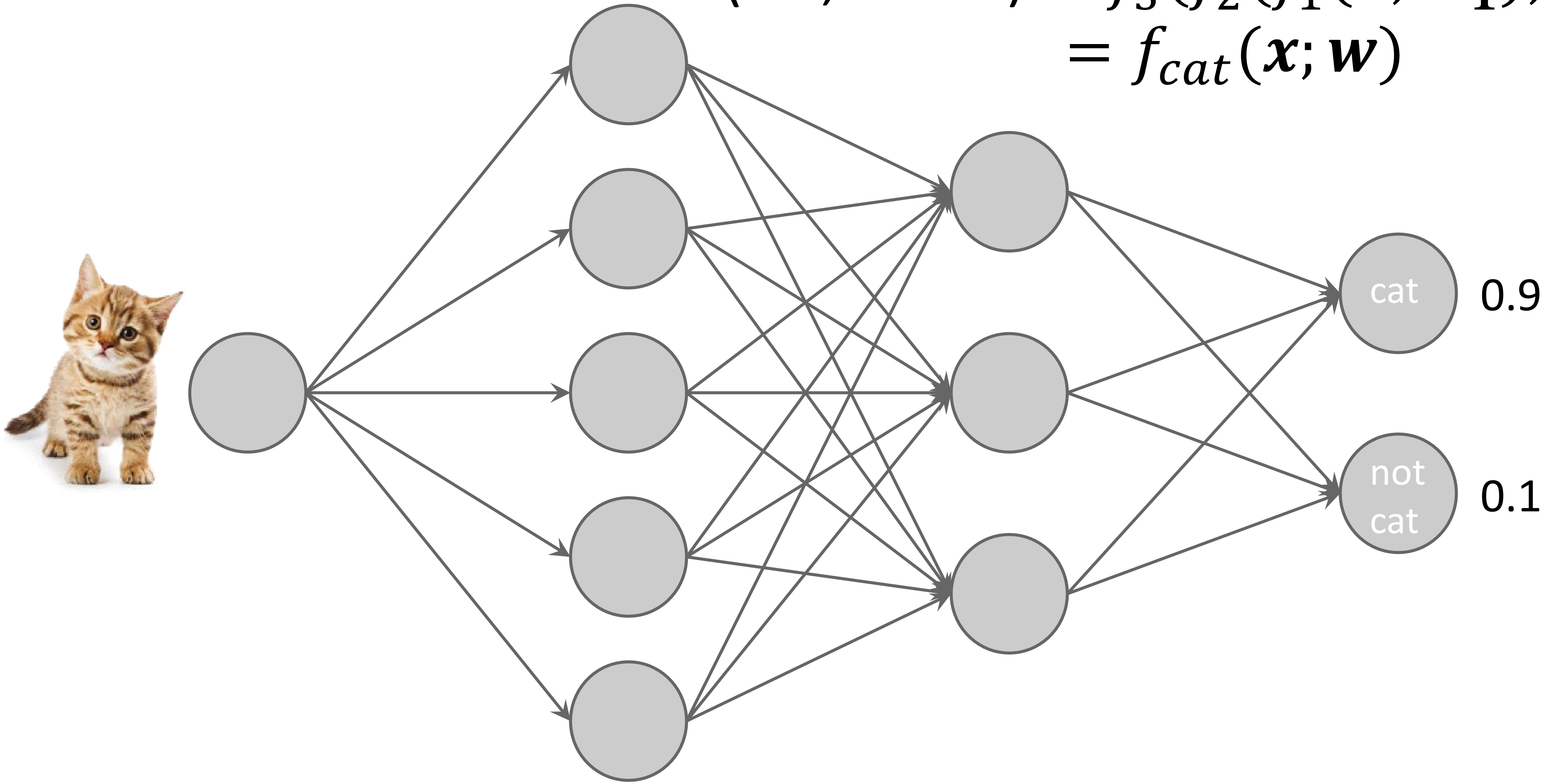


$$y = g(\sum_i w_i x_i + b)$$

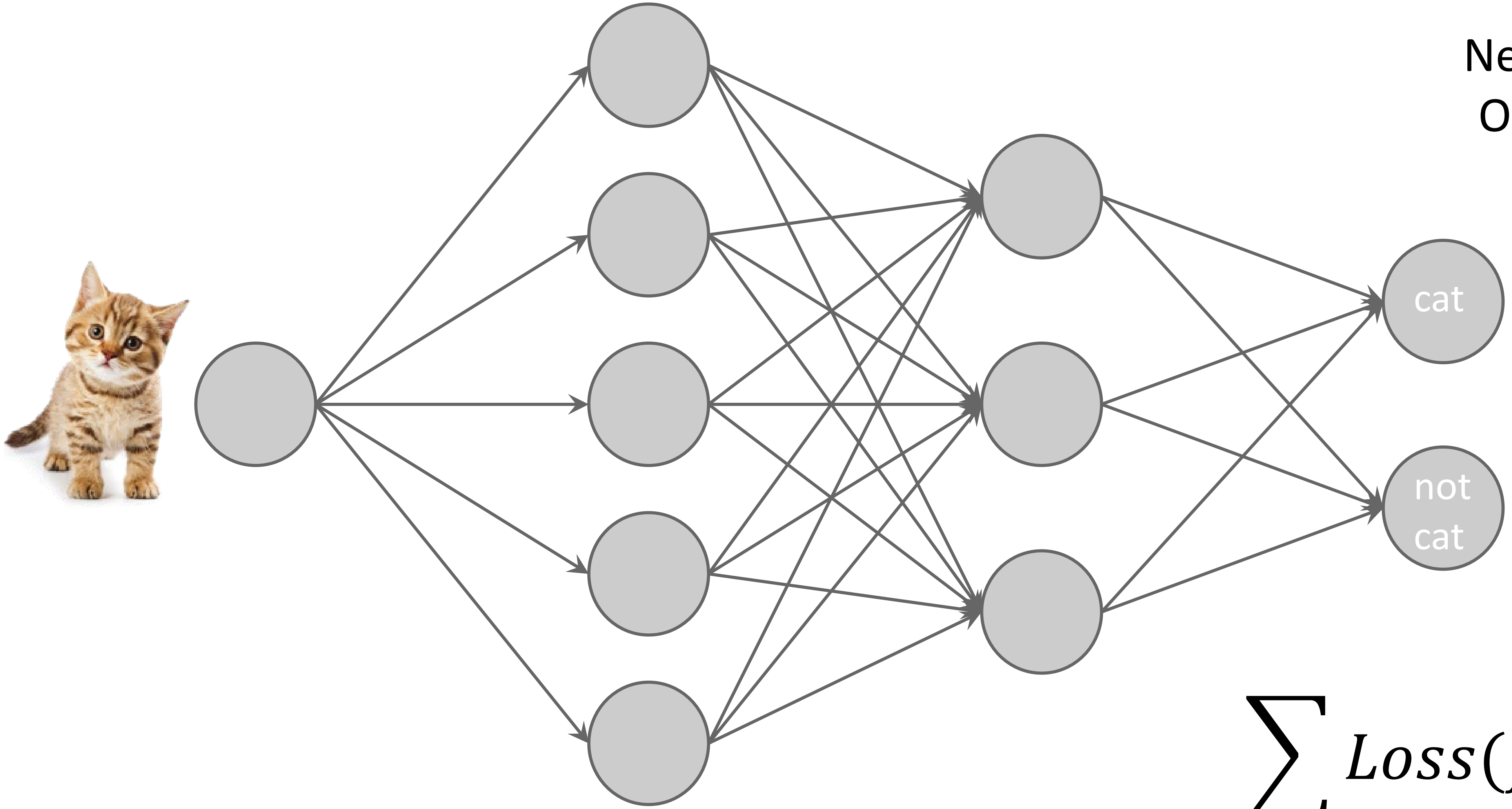
$$y = f(\boldsymbol{x}; \boldsymbol{w})$$

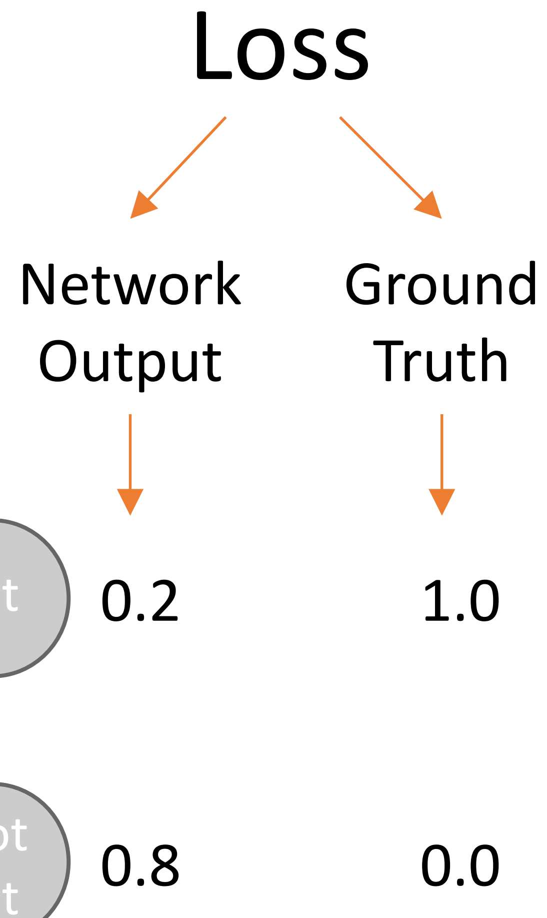
A (Shallow) Cat Detection Neural Network

$$(\text{cat,notcat}) = f_3(f_2(f_1(\boldsymbol{x};\boldsymbol{w_1});\boldsymbol{w_2}),\boldsymbol{w_3})$$
$$= f_{cat}(\boldsymbol{x};\boldsymbol{w})$$



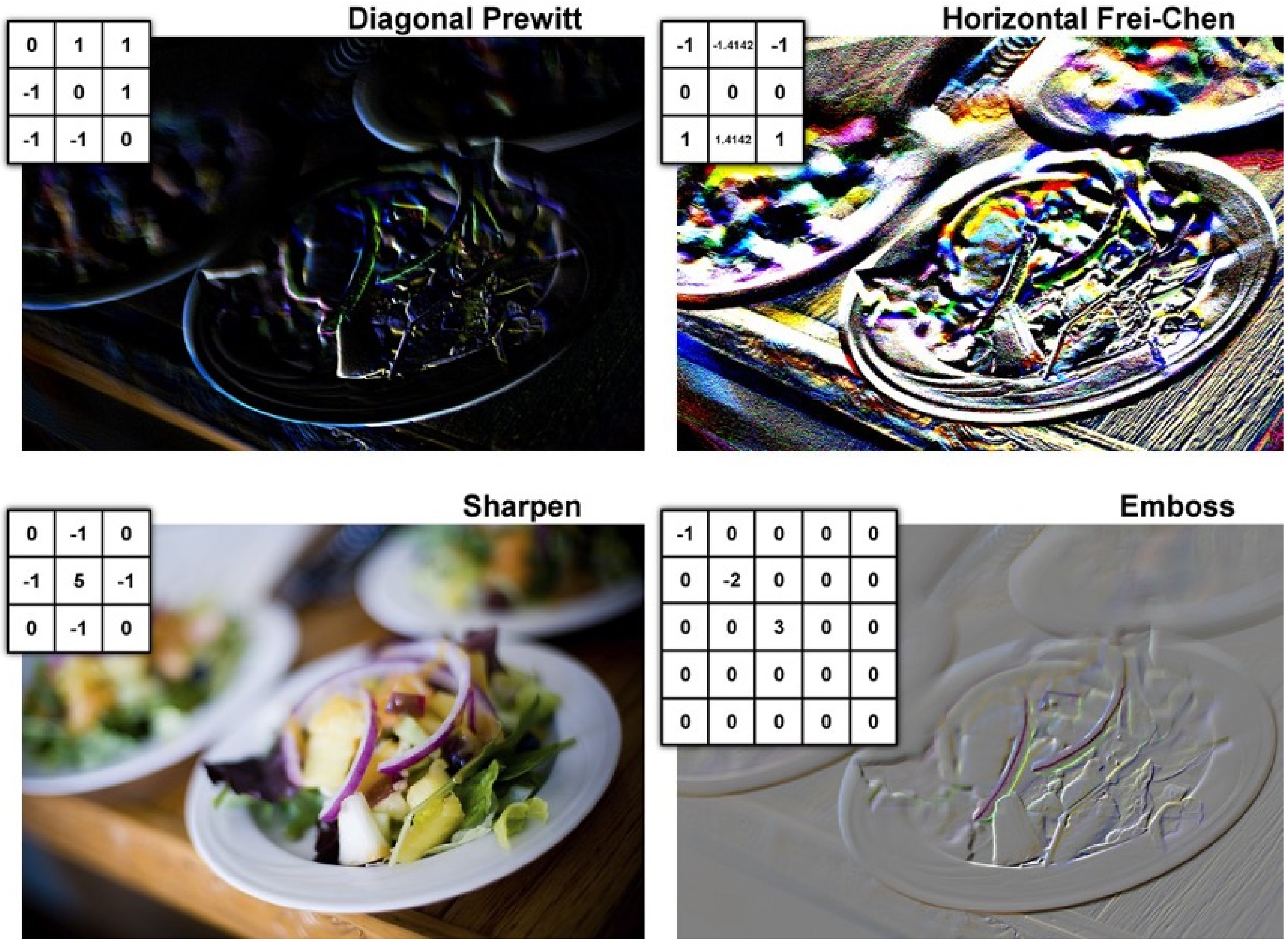
Minimize Loss to Solve for Weights



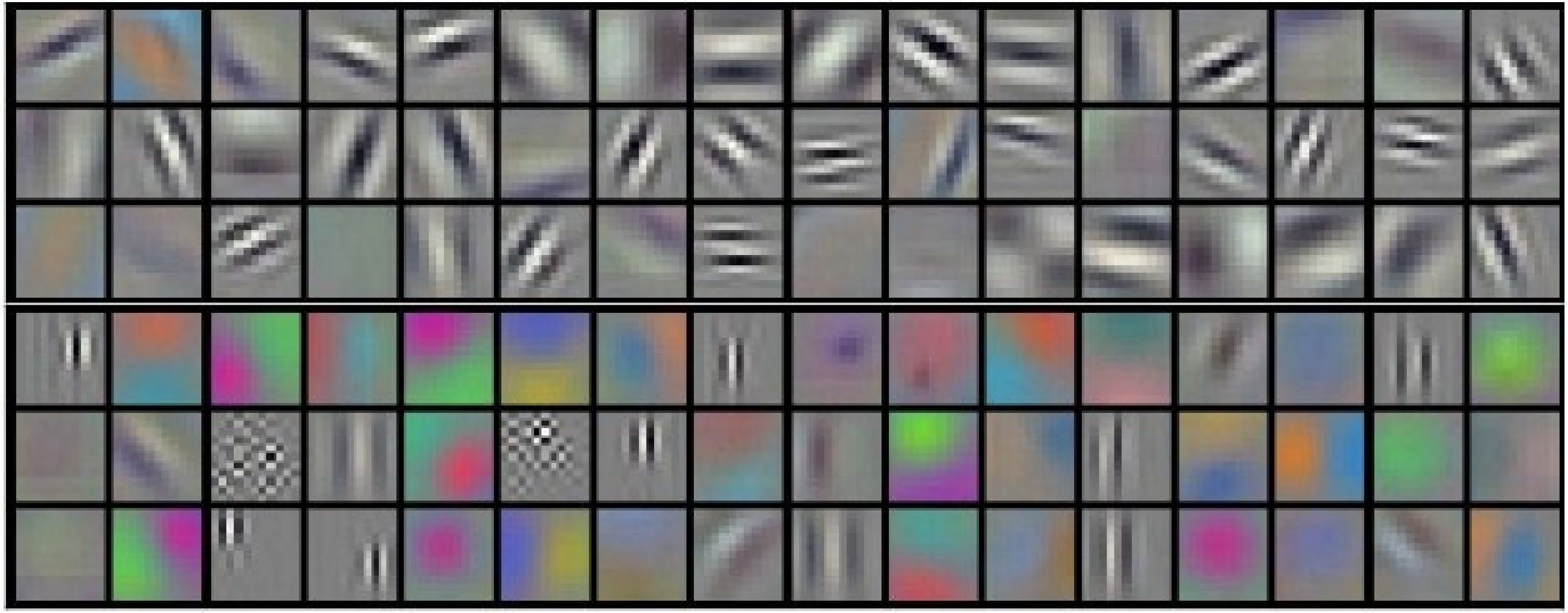


$$\sum_{i \in D} Loss(f_{cat}(\boldsymbol{x}^{\boldsymbol{i}}; \boldsymbol{w}), \boldsymbol{y}^{\boldsymbol{i}})$$

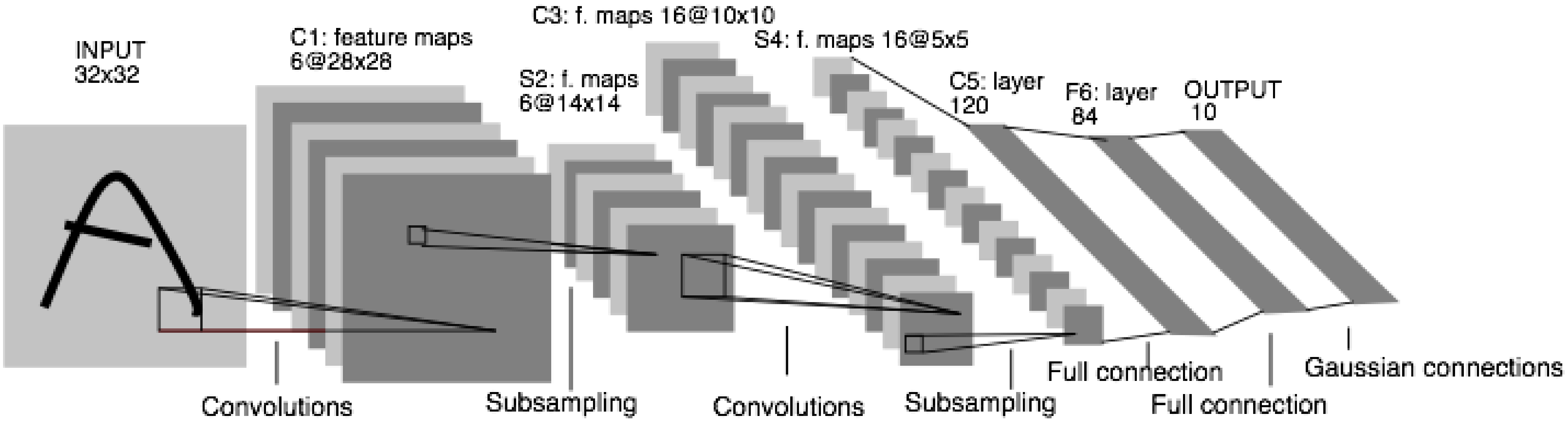
Deep Convolutional Neural Networks



Examples of Learned Convolutions

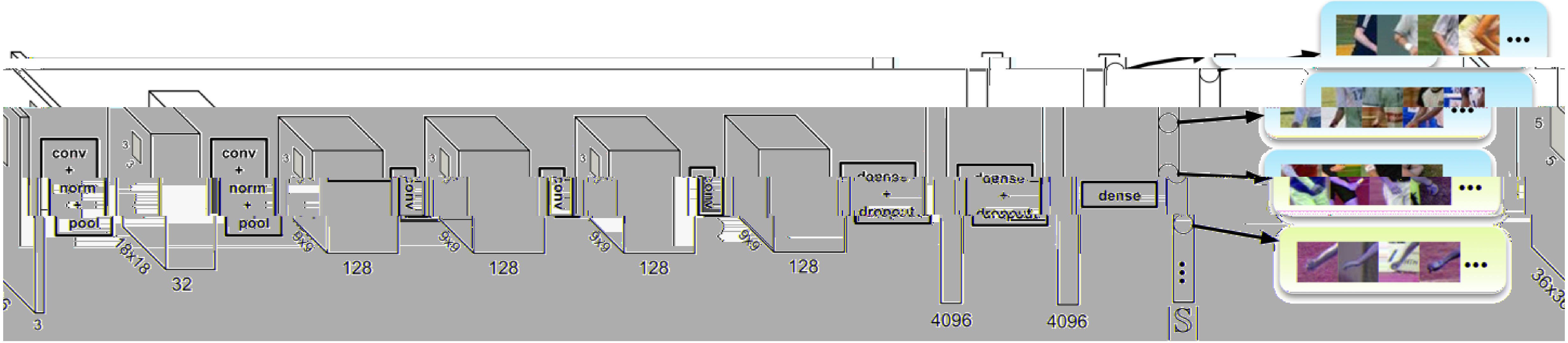


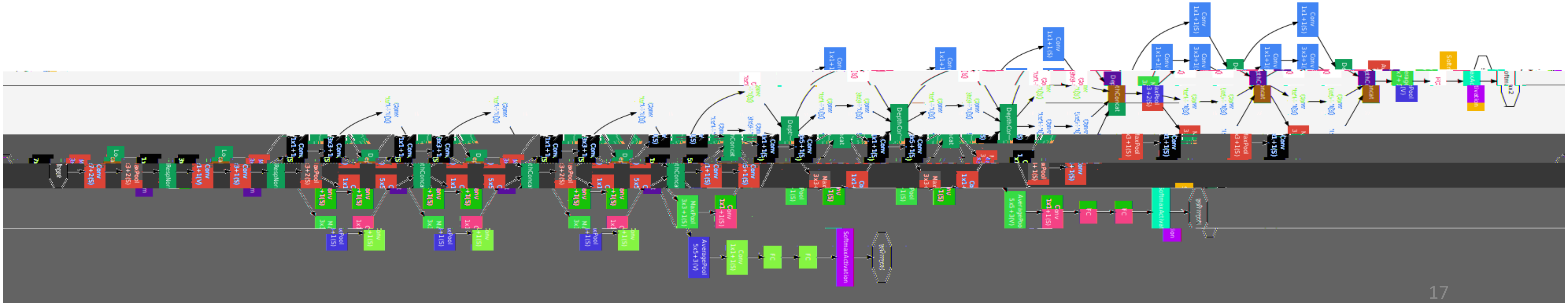
Shallow Convolutional Neural Network (1998)



Gradient-based learning applied to document recognition
Y LeCun, L Bottou, Y Bengio, P Haffner, **1998**

Deep Convolutional Neural Networks (2012-)

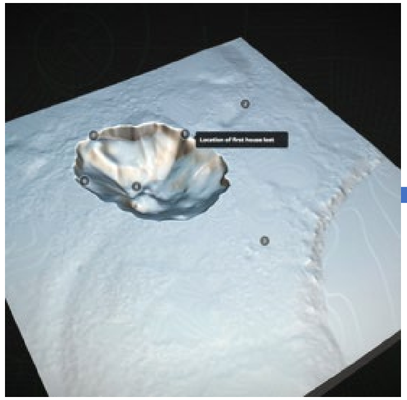




Applied to a Task from Geology

Junfeng Zhu, Nolte AM., Jacobs N., Ye M. 2019. Incorporating Machine Learning with LiDAR for Delineating Sinkholes. In: *Kentucky Water Resources Annual Symposium*.

ML

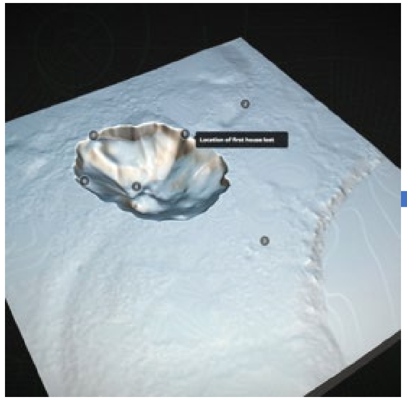


human written code (identify potential sinkholes, compute features)

shallow neural network

"sinkhole?"

Deep

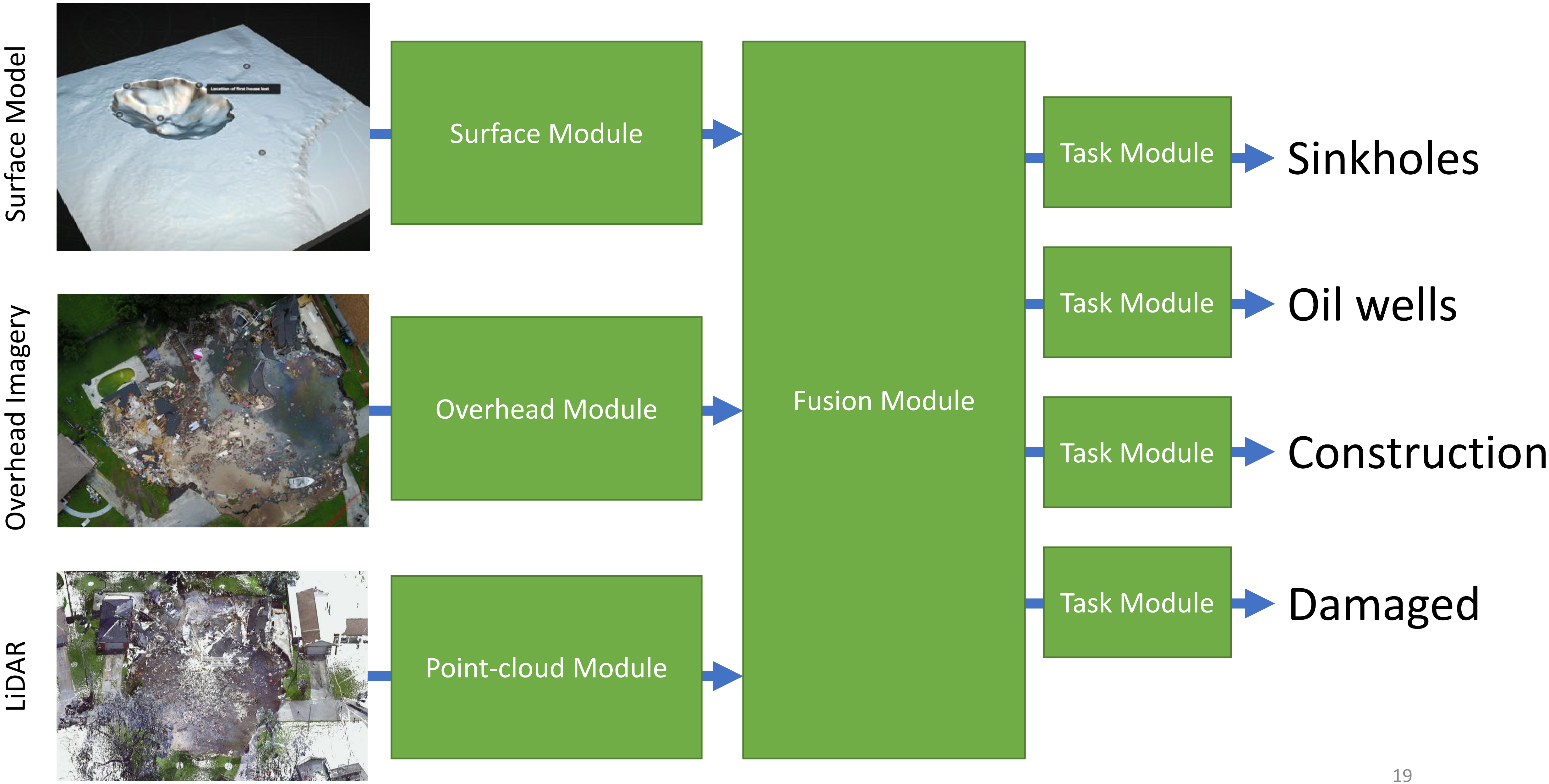


Deep neural network

?

"sinkhole!"

Going a few steps further…



Two Parts

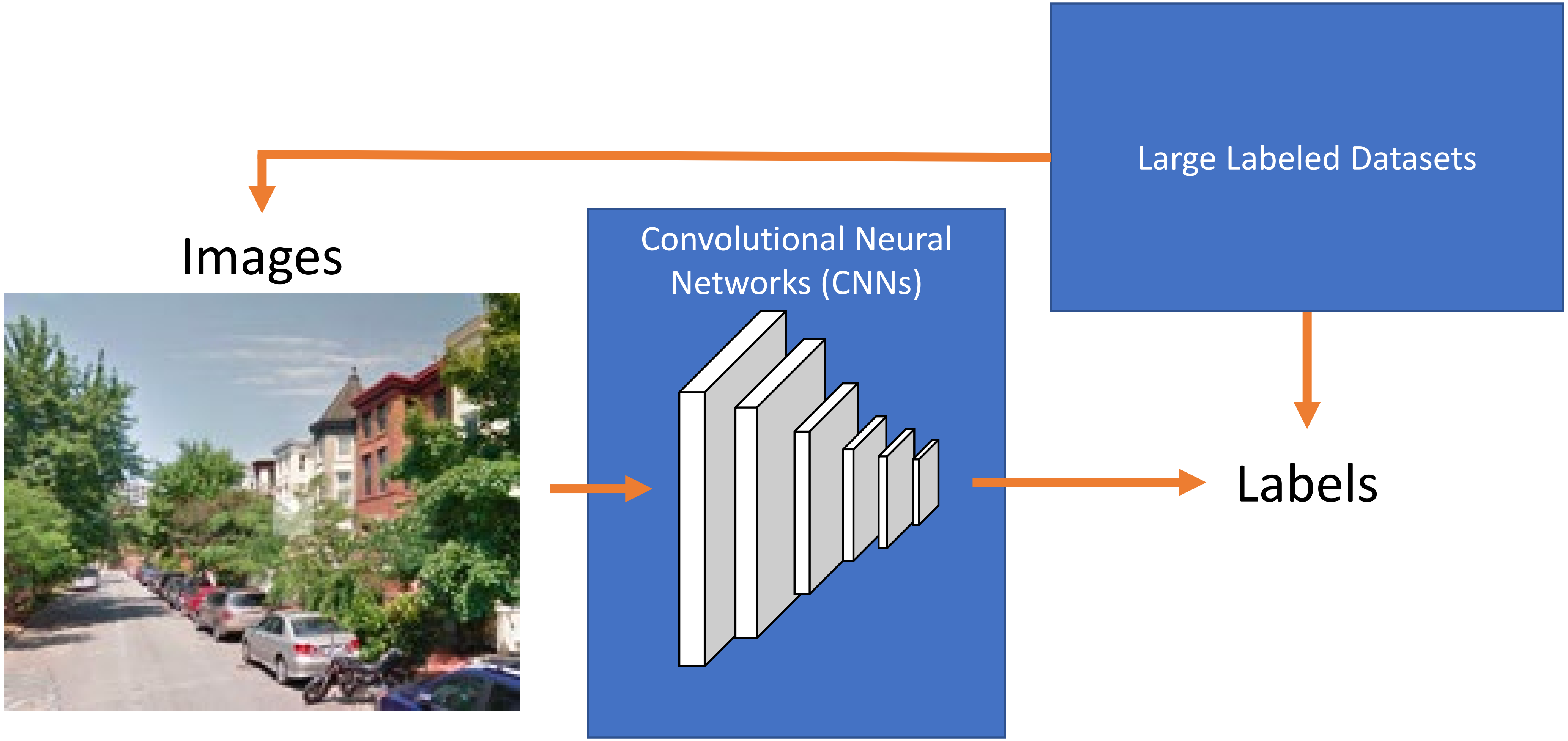
1. The ongoing revolution in automated perception.

2. My work on image-driven mapping.

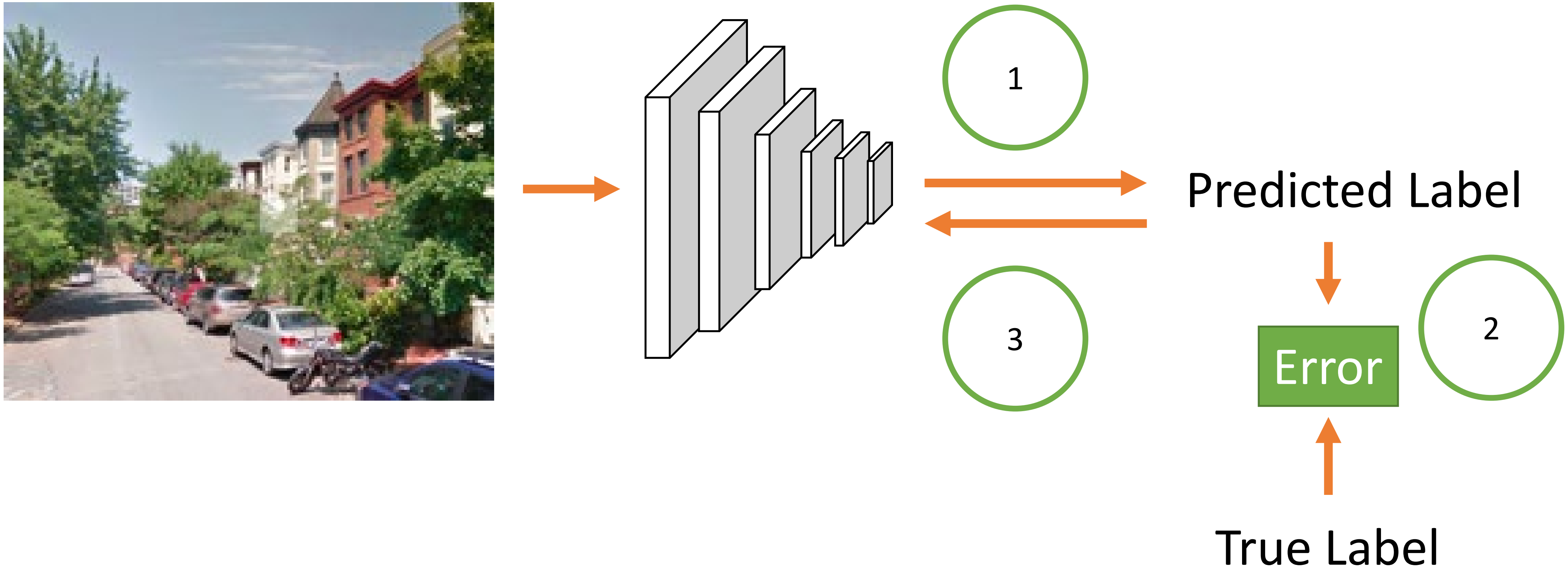
Research Theme # 1

Ground-level images as a supervisory signal for overhead image interpretation.

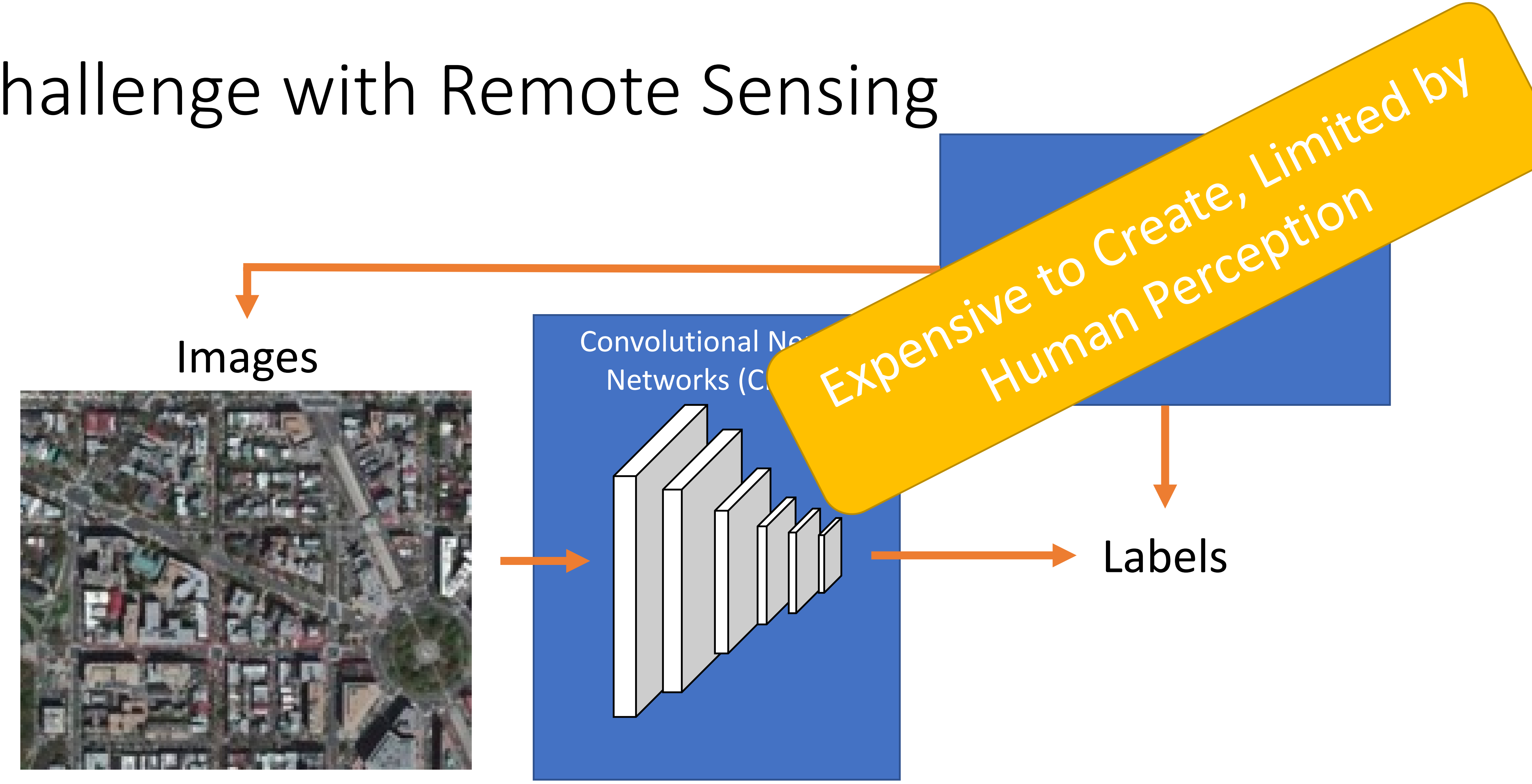
Essential Building Blocks



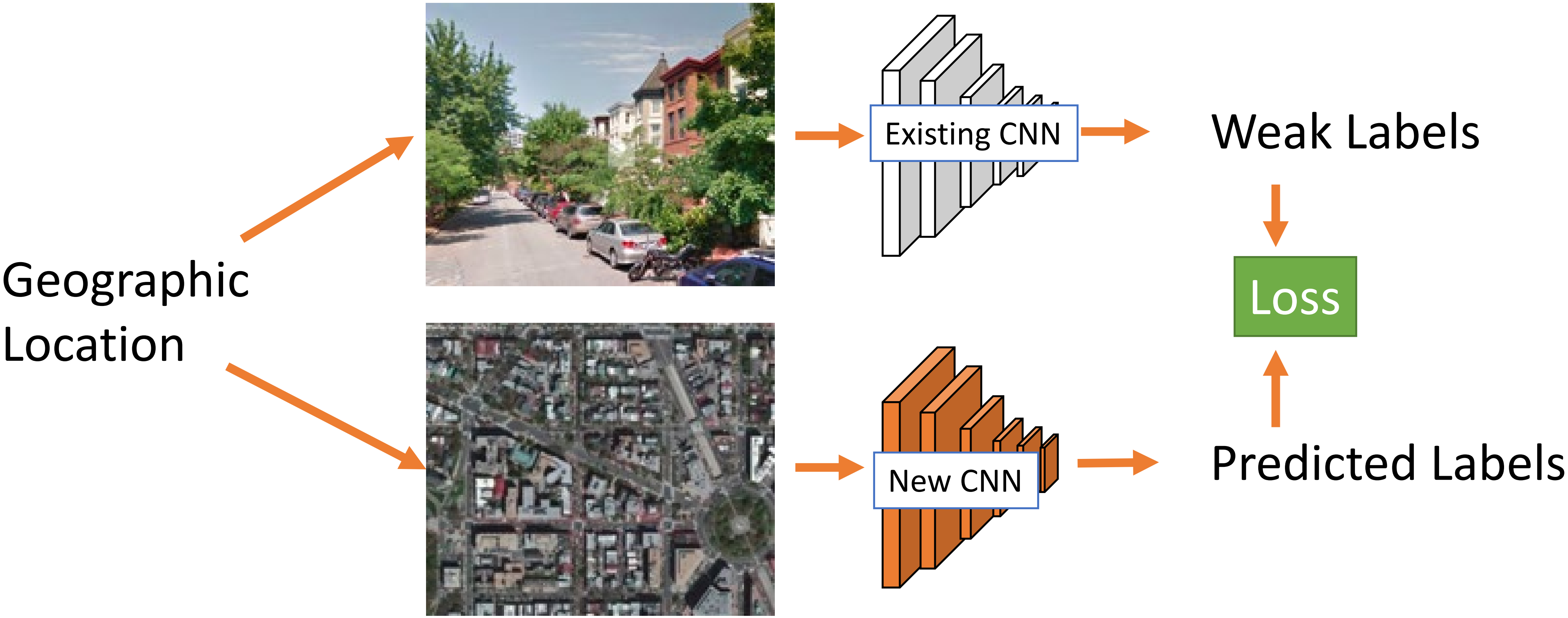
Training a CNN



Challenge with Remote Sensing



Ground-Level Images as a Supervisory Signal



Three Examples

- Scene classification
- Semantic segmentation
- Object detection

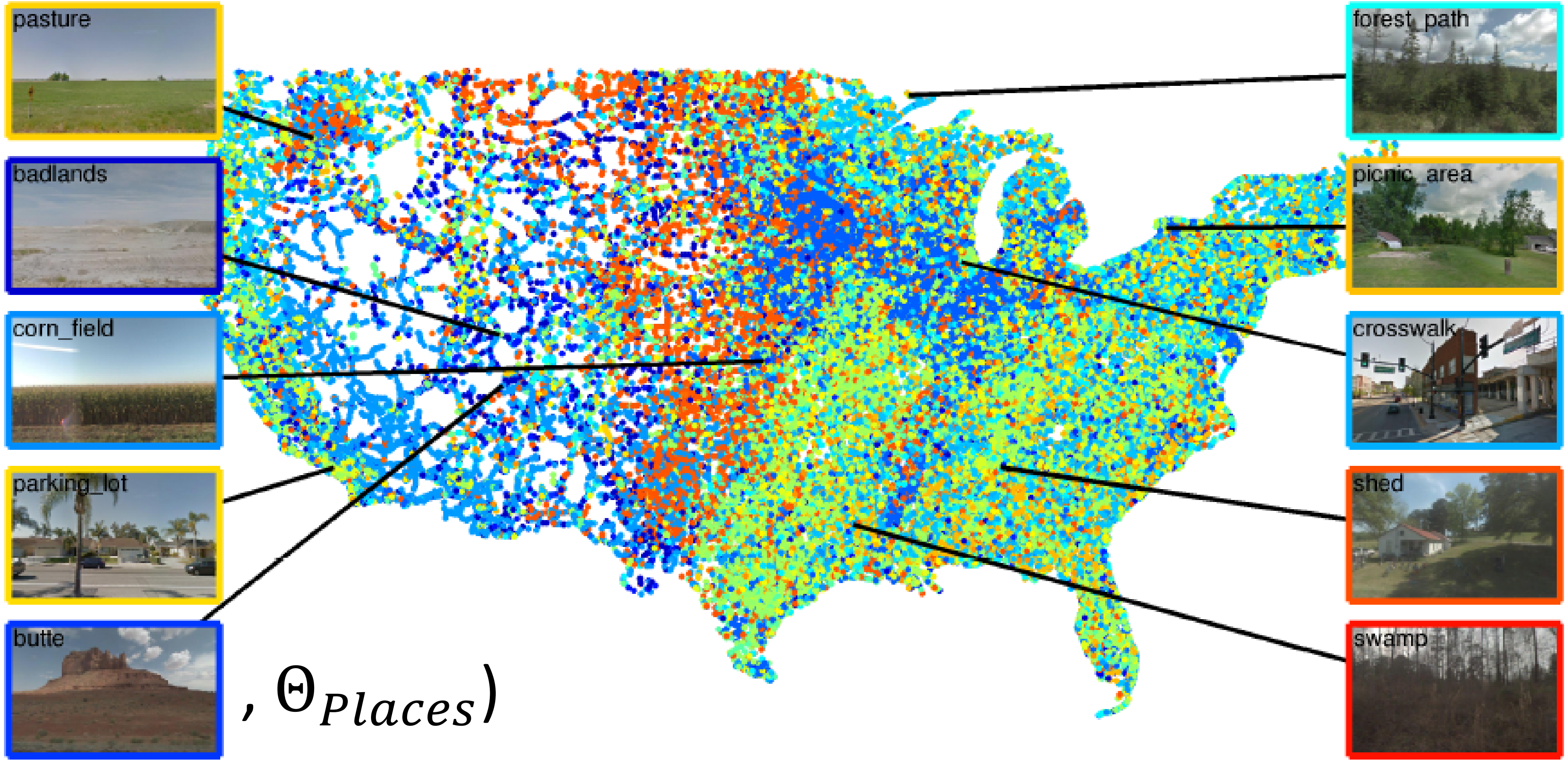
CVUSA: A Large Training Database of Ground-Level and Aerial Image Pairs

ground-level image | high-res overhead | med-res overhead | low-res overhead

⋮

ICCV 2015

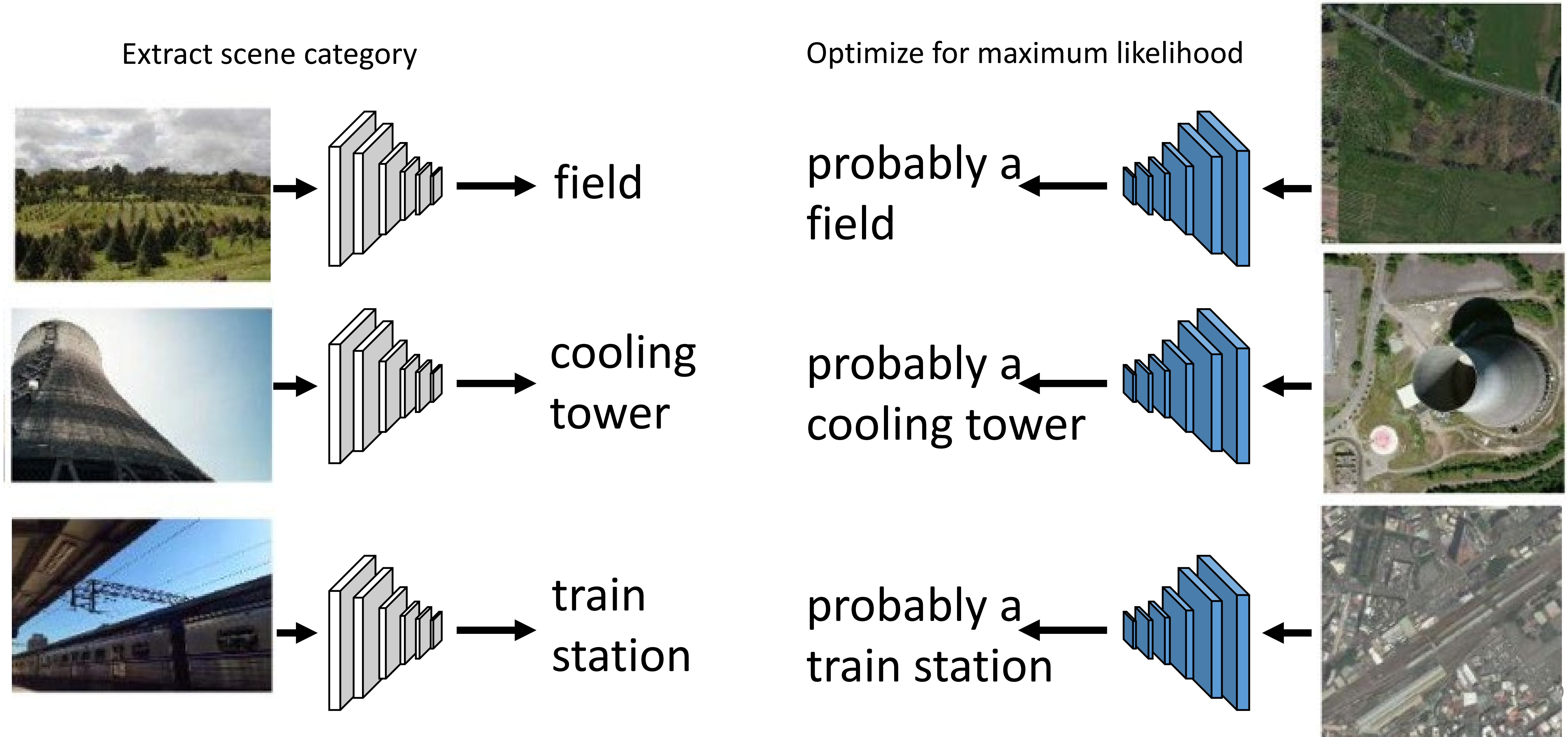
Scene Categories are Location Dependent



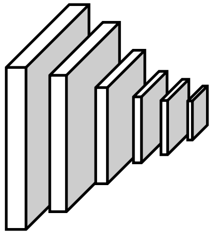
$$f_g(\;\cdot\;, \Theta_{Places})$$

B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, 2014.

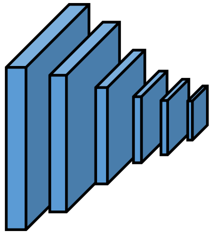
Learning to Predict Ground-Level Scene Categories from Overhead Imagery



Ad-Hoc Mapping Using a Single Query Image

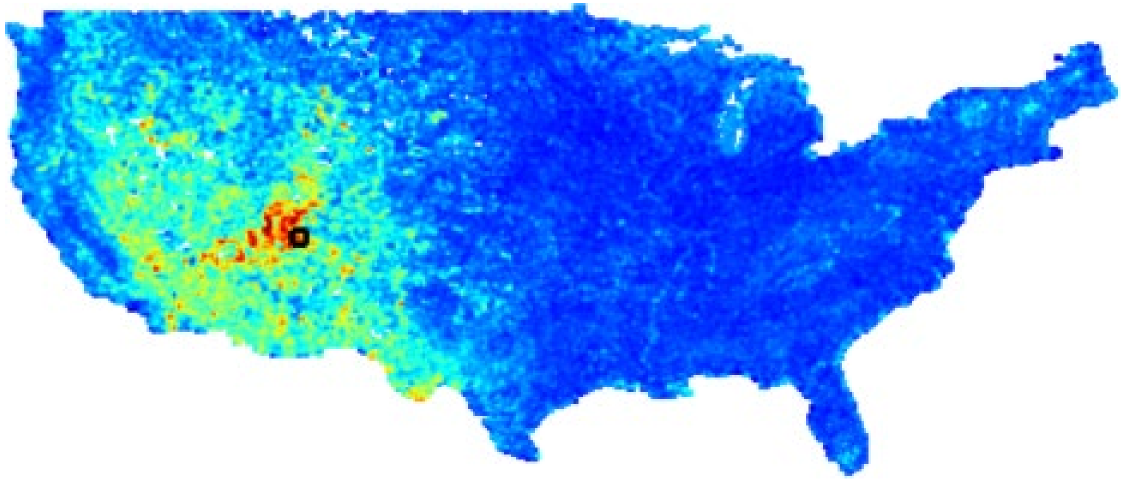
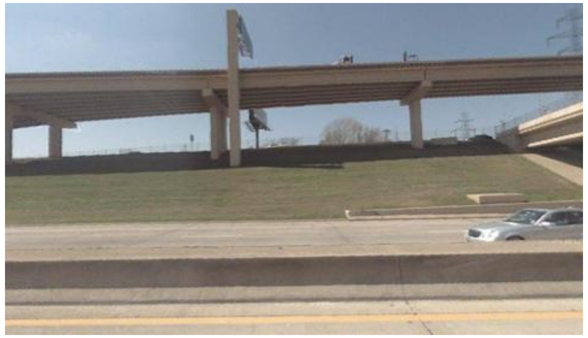
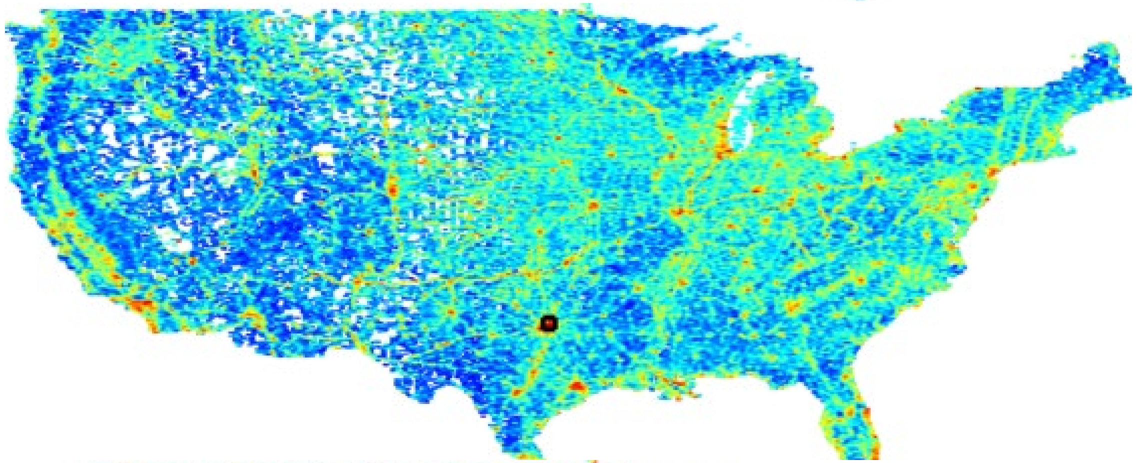


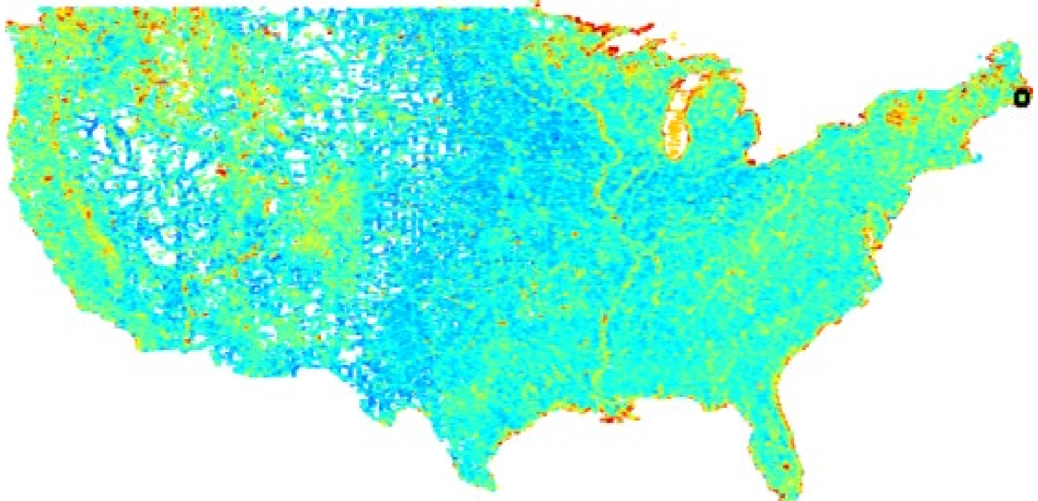
Description of
query image



Description of
location

Examples of Ad-hoc Maps



Three Examples

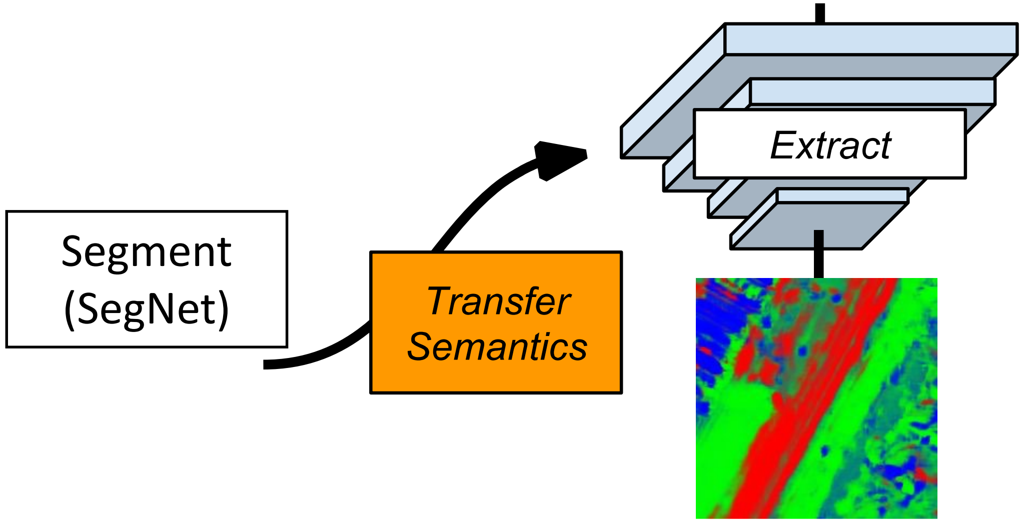
- Scene classification
- Semantic segmentation
- Object detection

Similar Idea;
Richer Supervision

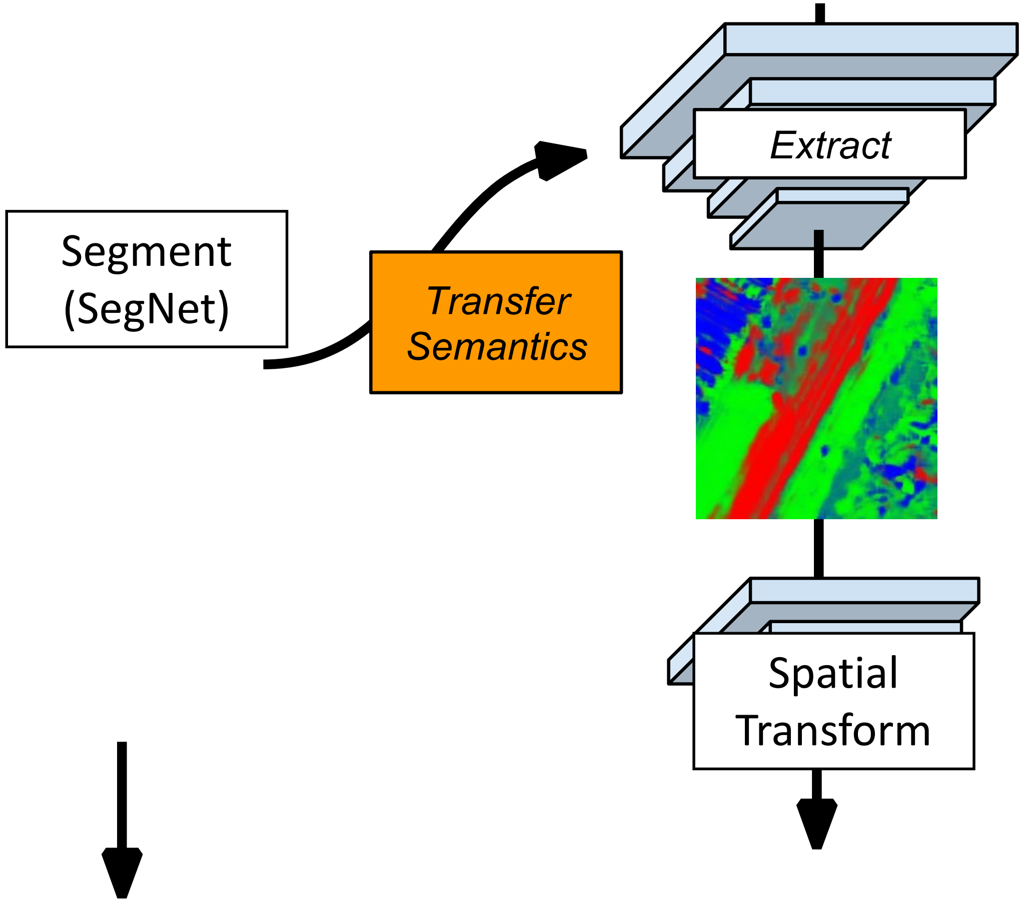
Segment
(SegNet)

CVPR 2017

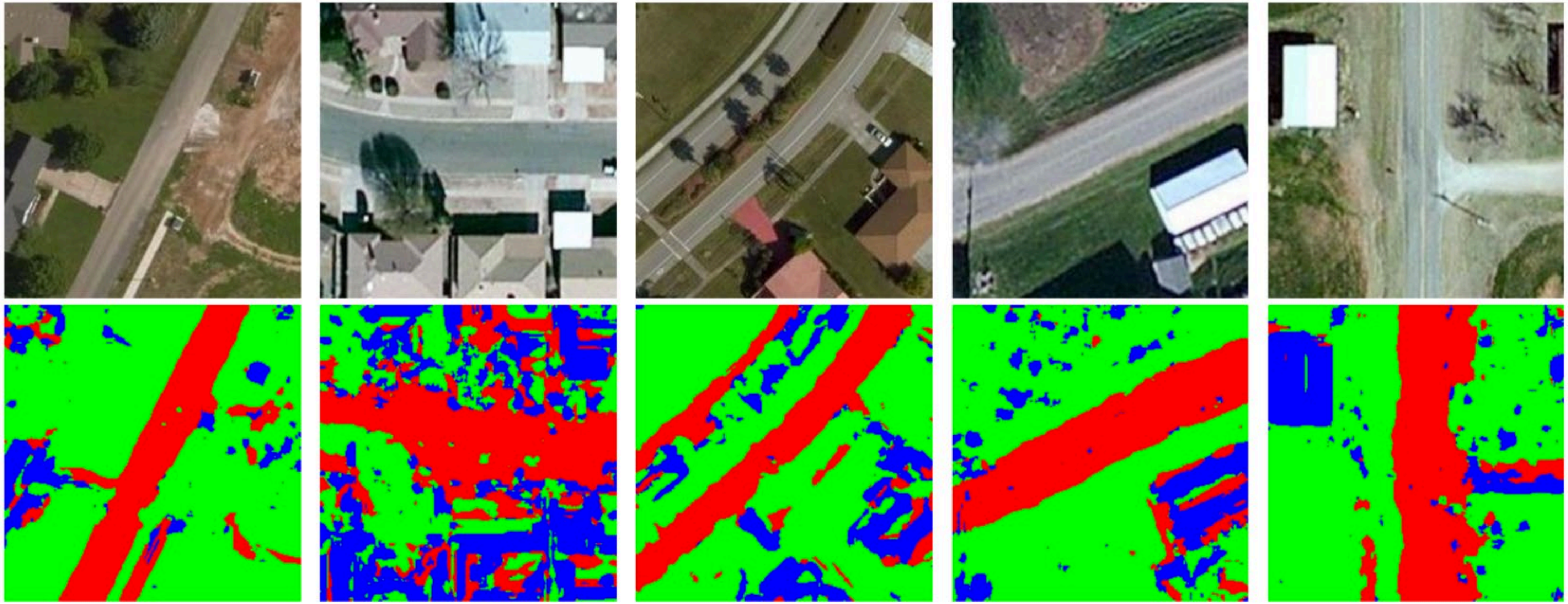
Similar Idea; Richer Supervision



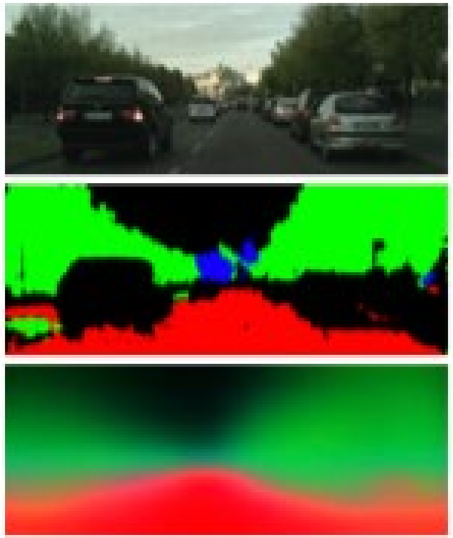
Similar Idea; Richer Supervision

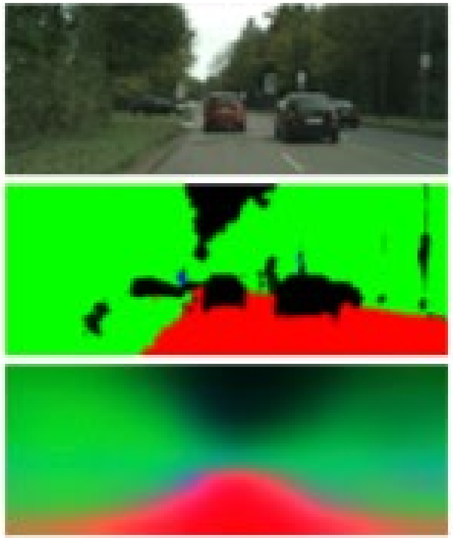


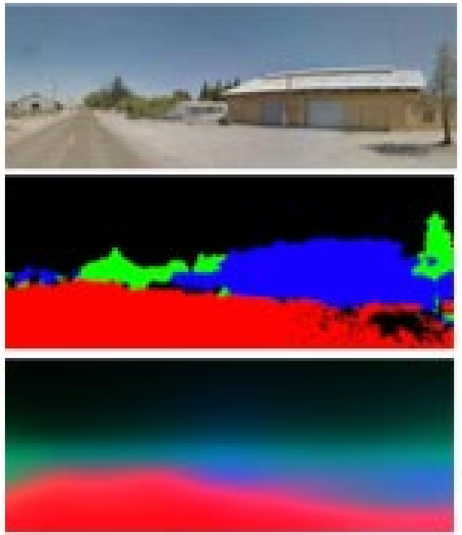
Segmentation without Labeled Satellite Imagery

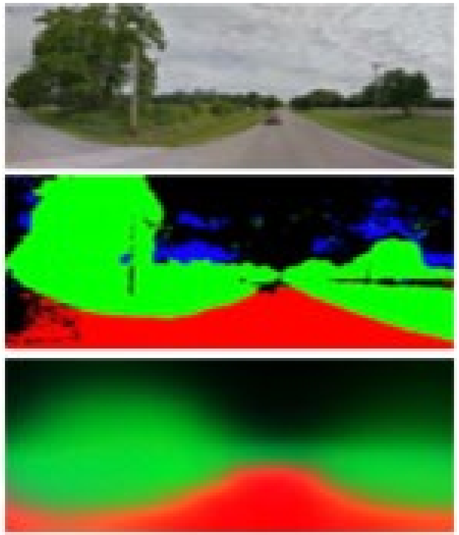


Application: Panorama Orientation Estimation

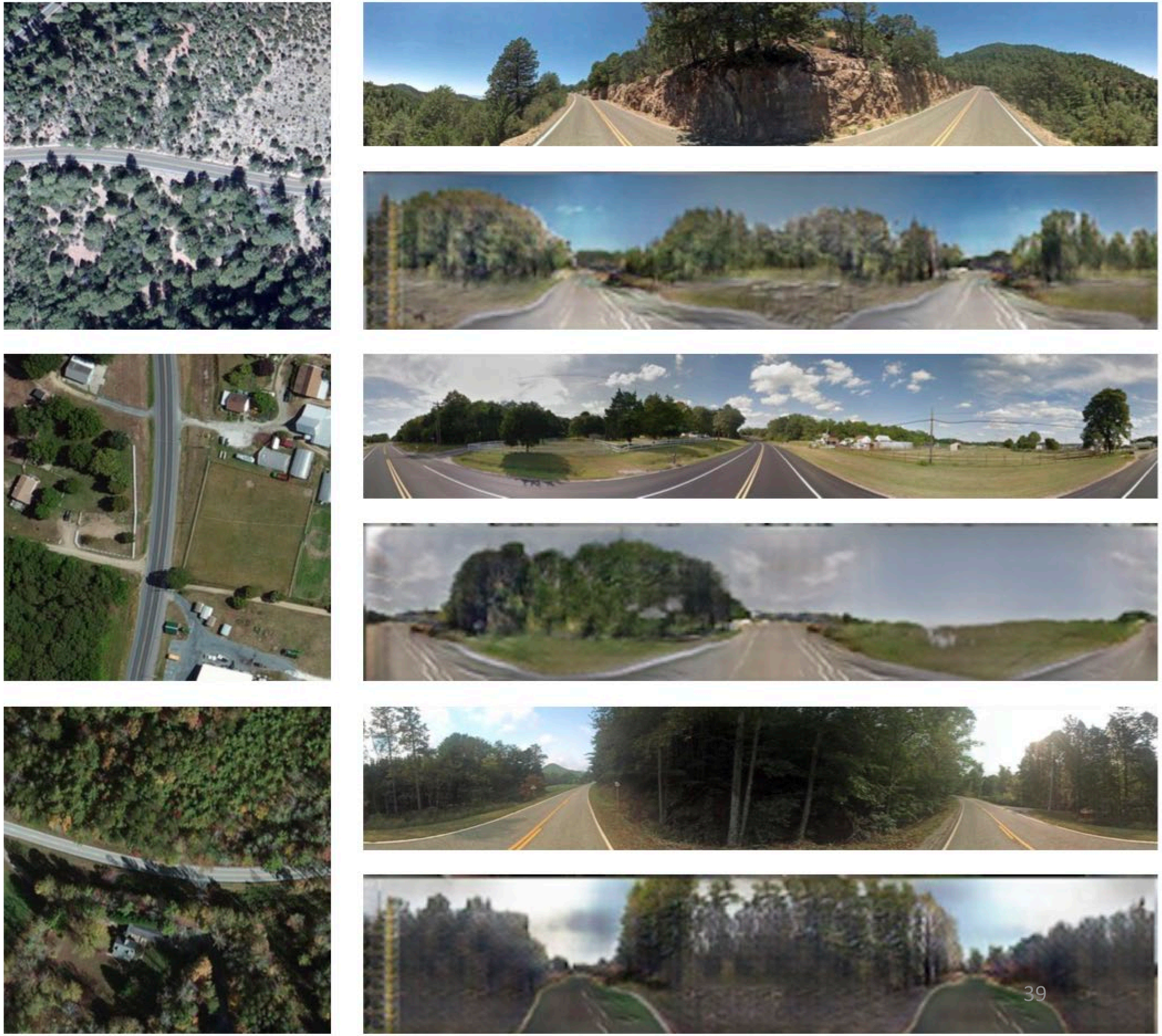








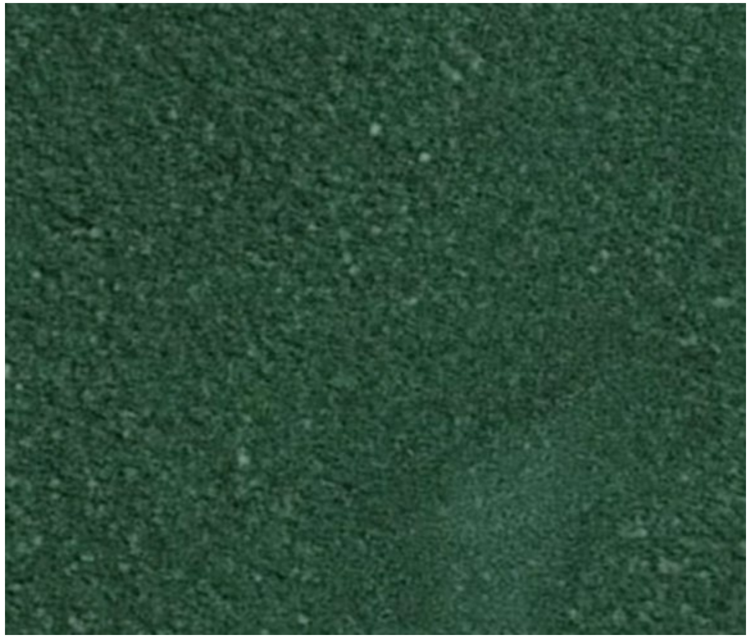
Application: Synthesizing Ground-Level Images



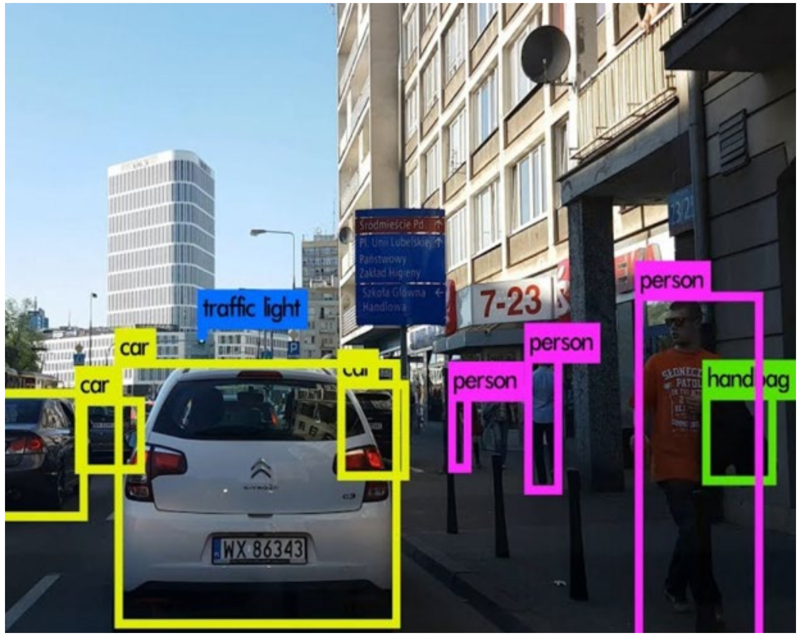
Three Examples

- Scene classification
- Semantic segmentation
- Object detection

IGARSS 2018



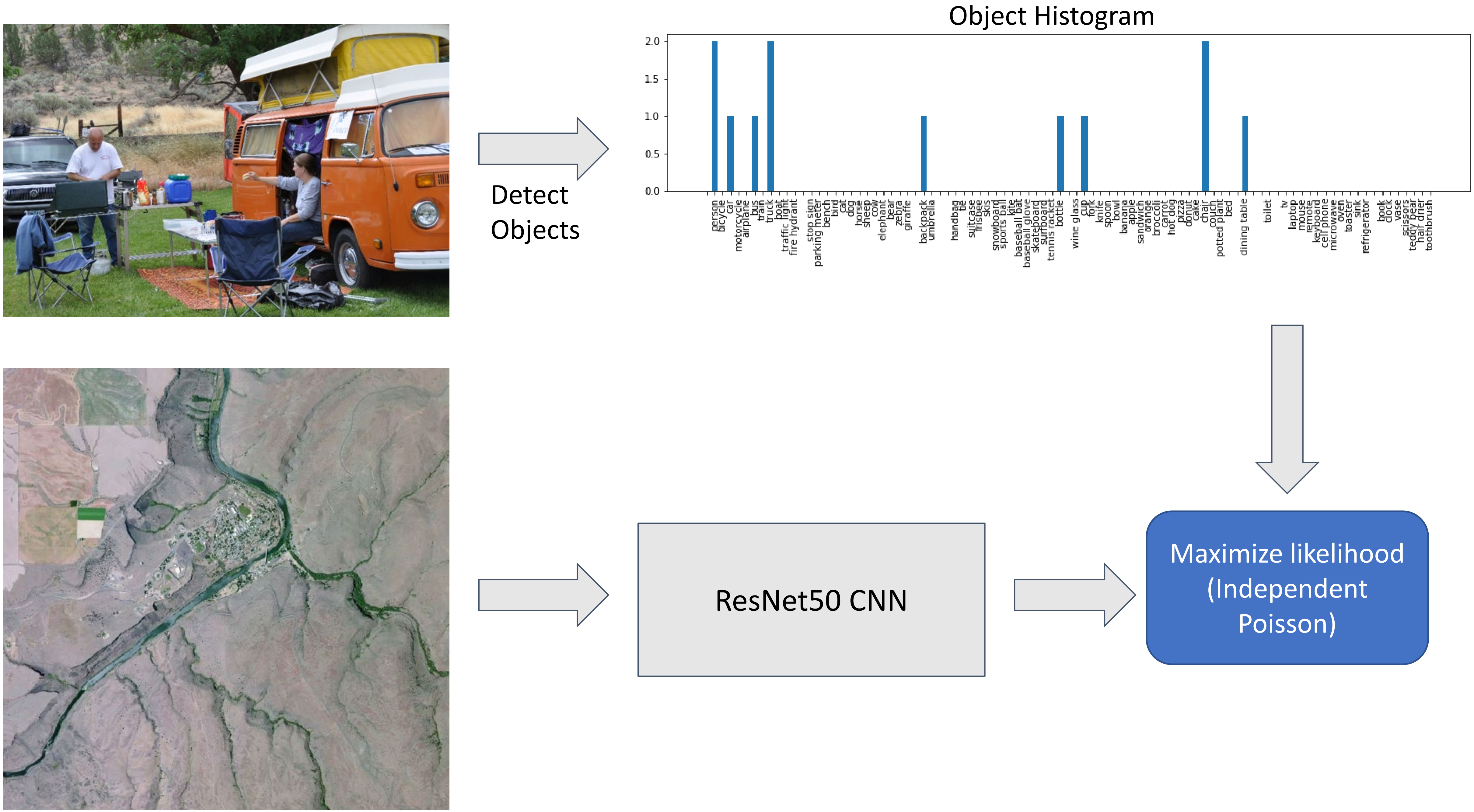
Tree: 35%
Monkey: 97%



traffic light
car
car
person
person
person
7-23
WX 86343

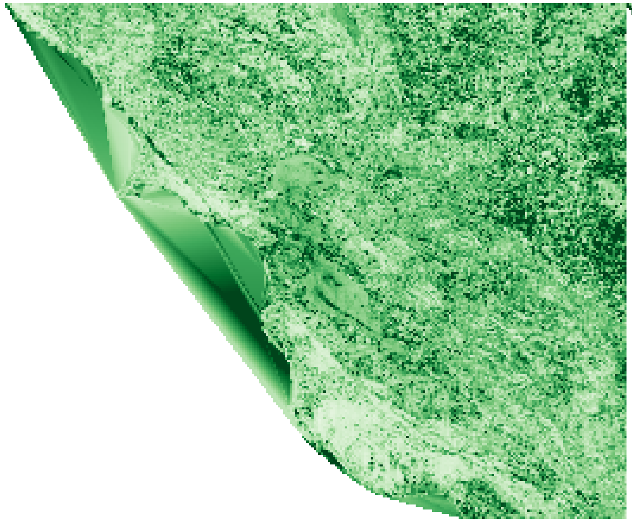
Dataset and Pre-Processing

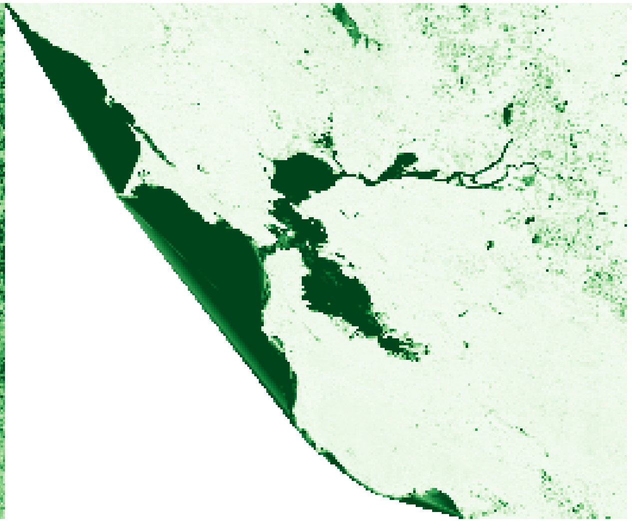
- 551,851 Geotagged Flickr Images (from CVUSA Dataset)
- Use Faster R-CNN to detect 91 Object Classes (from MSCOCO)

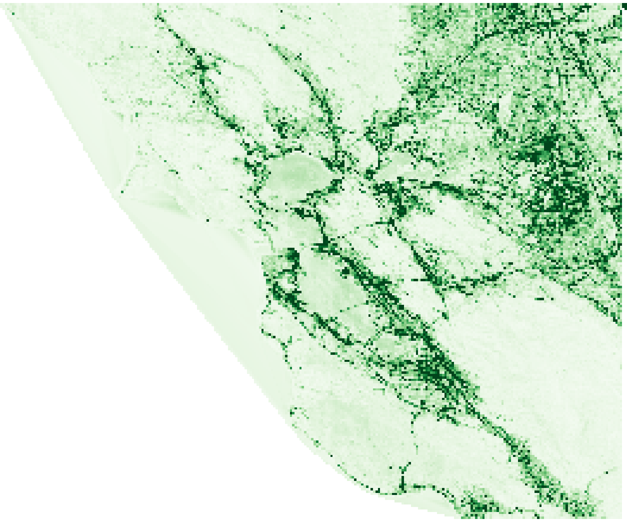


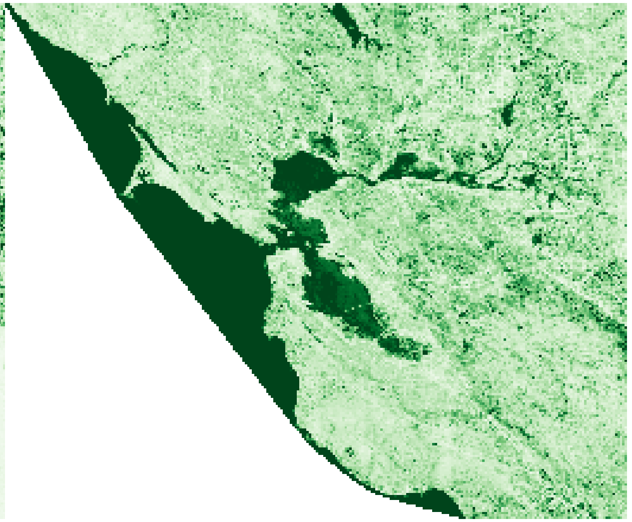
Object Histogram
2.0
1.5
1.0
0.5
0.0
person
bicycle
car
motorcycle
airplane
bus
train
truck
boat
traffic light
fire hydrant
stop sign
parking meter
bench
bird
cat
dog
horse
sheep
cow
elephant
bear
zebra
giraffe
backpack
umbrella
handbag
tie
suitcase
frisbee
skis
snowboard
sports ball
kite
baseball bat
baseball glove
skateboard
surfboard
tennis racket
bottle
wine glass
cup
fork
knife
spoon
bowl
banana
apple
sandwich
orange
broccoli
carrot
hot dog
pizza
donut
cake
chair
couch
potted plant
bed
dining table
toilet
tv
laptop
mouse
remote
keyboard
cell phone
microwave
oven
toaster
sink
refrigerator
book
clock
vase
scissors
teddy bear
hair drier
toothbrush
Detect Objects
ResNet50 CNN
Maximize likelihood (Independent Poisson)

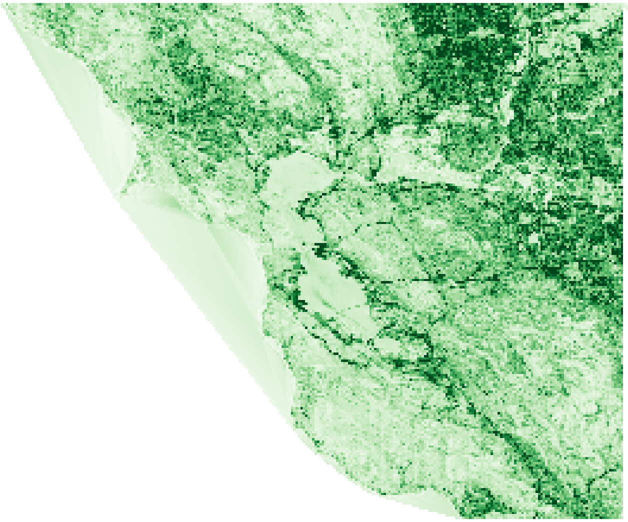
Satellite-Based Expectation of “Objects Per Image”

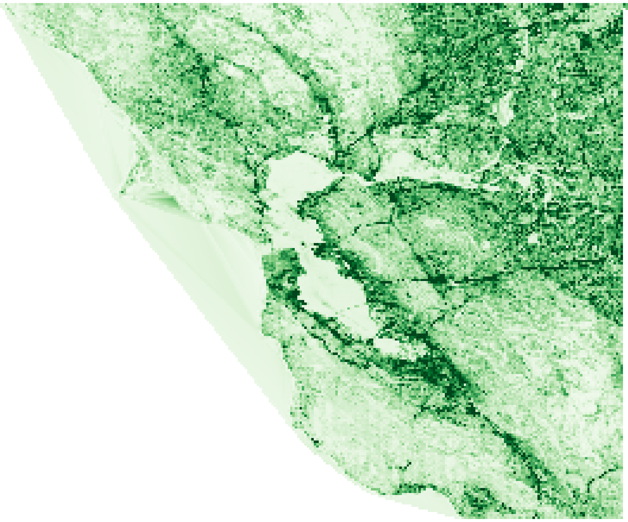

Person

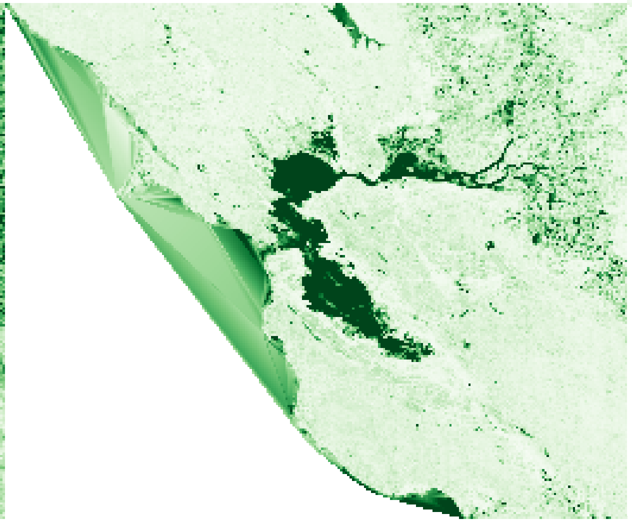

Boat

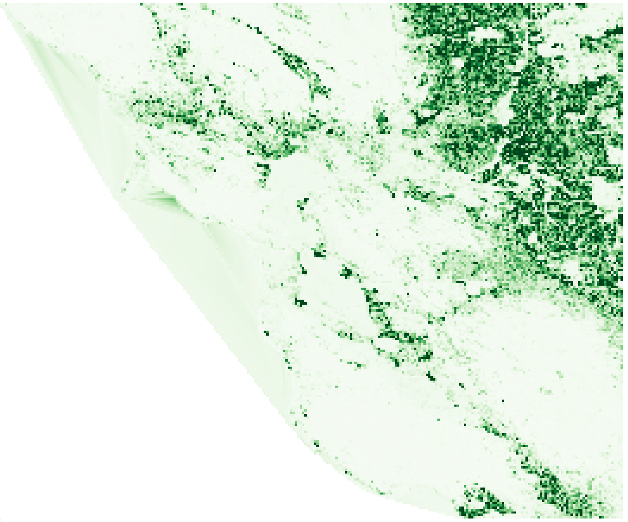

Train


Surfboard


Truck


Car


Bird

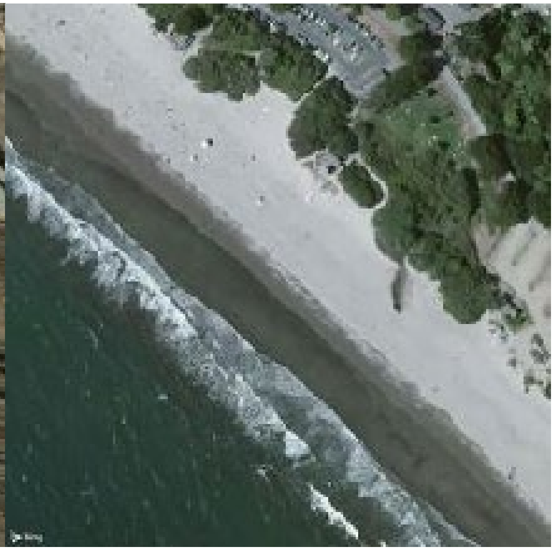

Airplane

Maximal Expectation Images

Person

Boat

Train



Surfboard

Truck

Car

Bird

Airplane

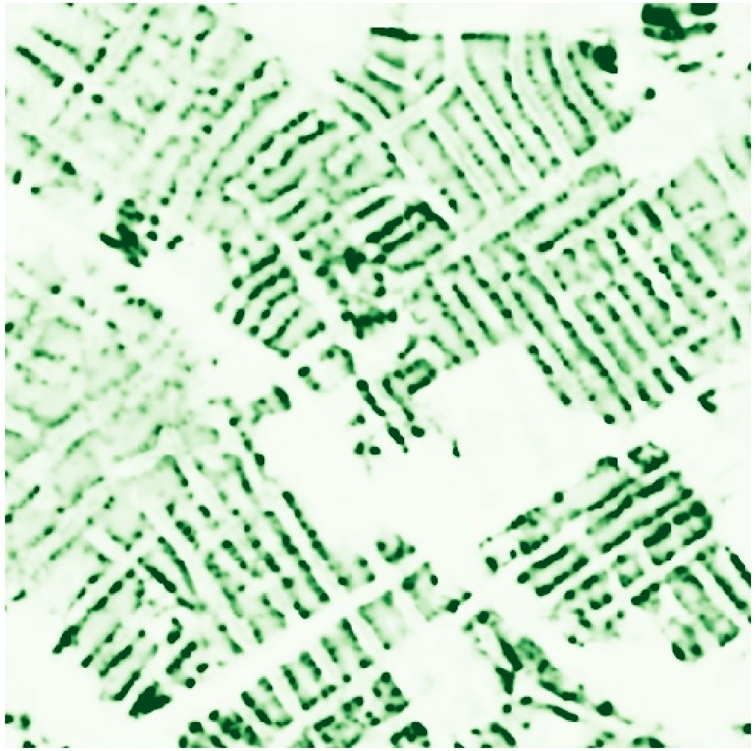
Research Theme # 2

Include differentiable domain approximations in the network.

Objective: Estimate Spatial Distribution of Some Object Type

Satellite Image

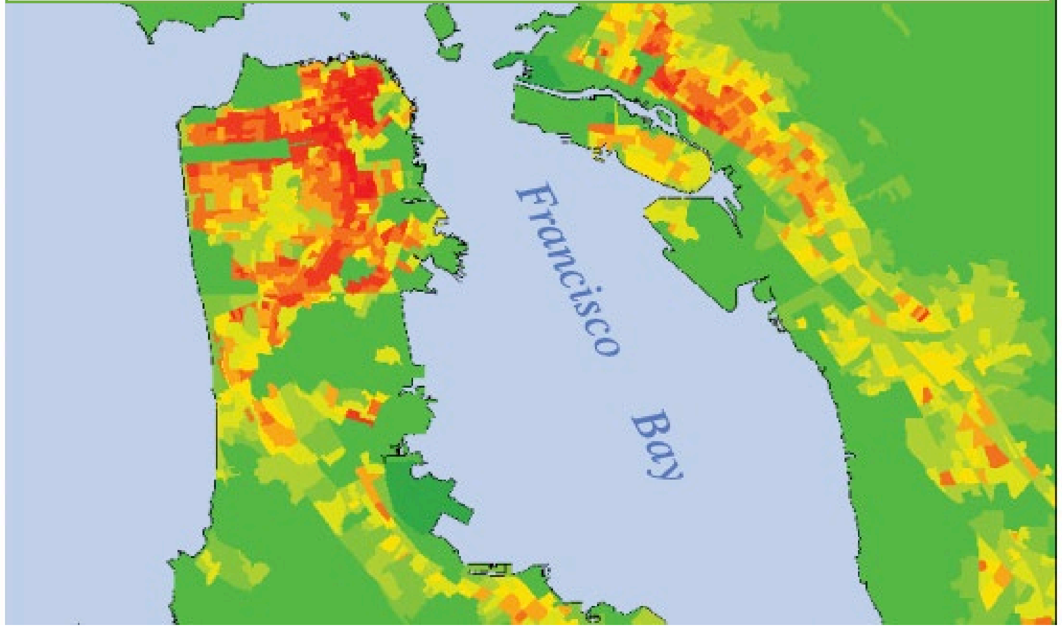
Population Density

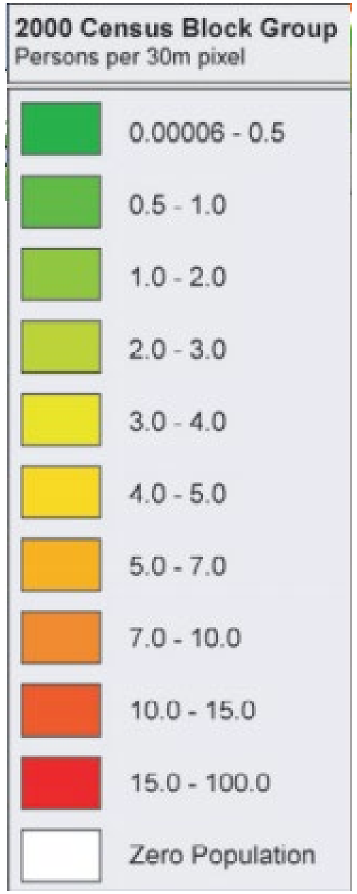


Traditional Approaches

Manual Census + Choropleth

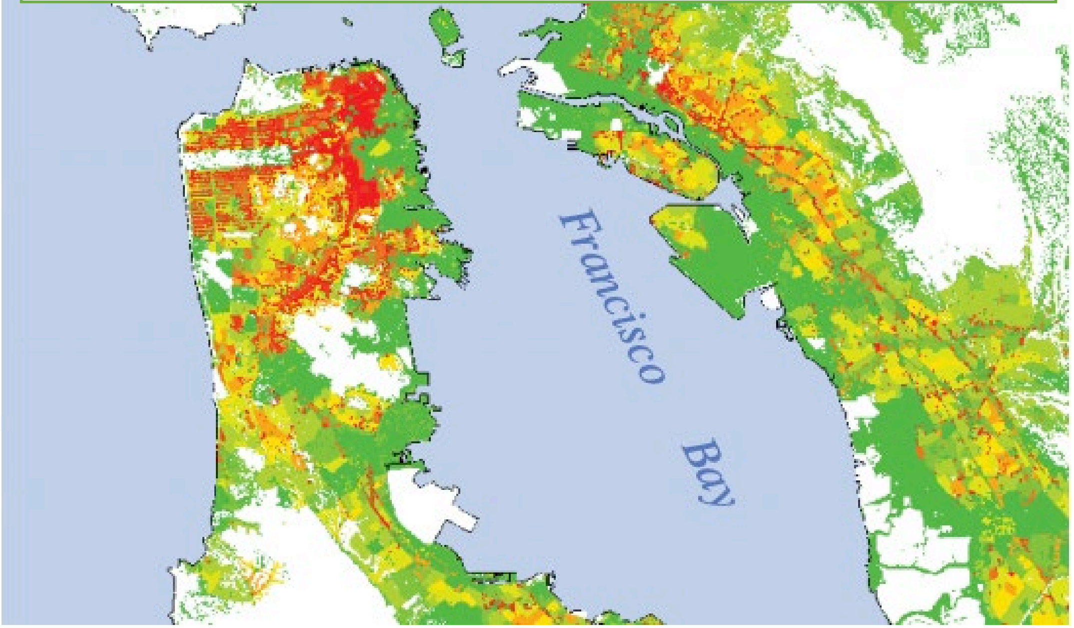
- Expensive data collection
- Low temporal frequency
- Low spatial resolution
- Shows people living in unlikely places (e.g., SFO)





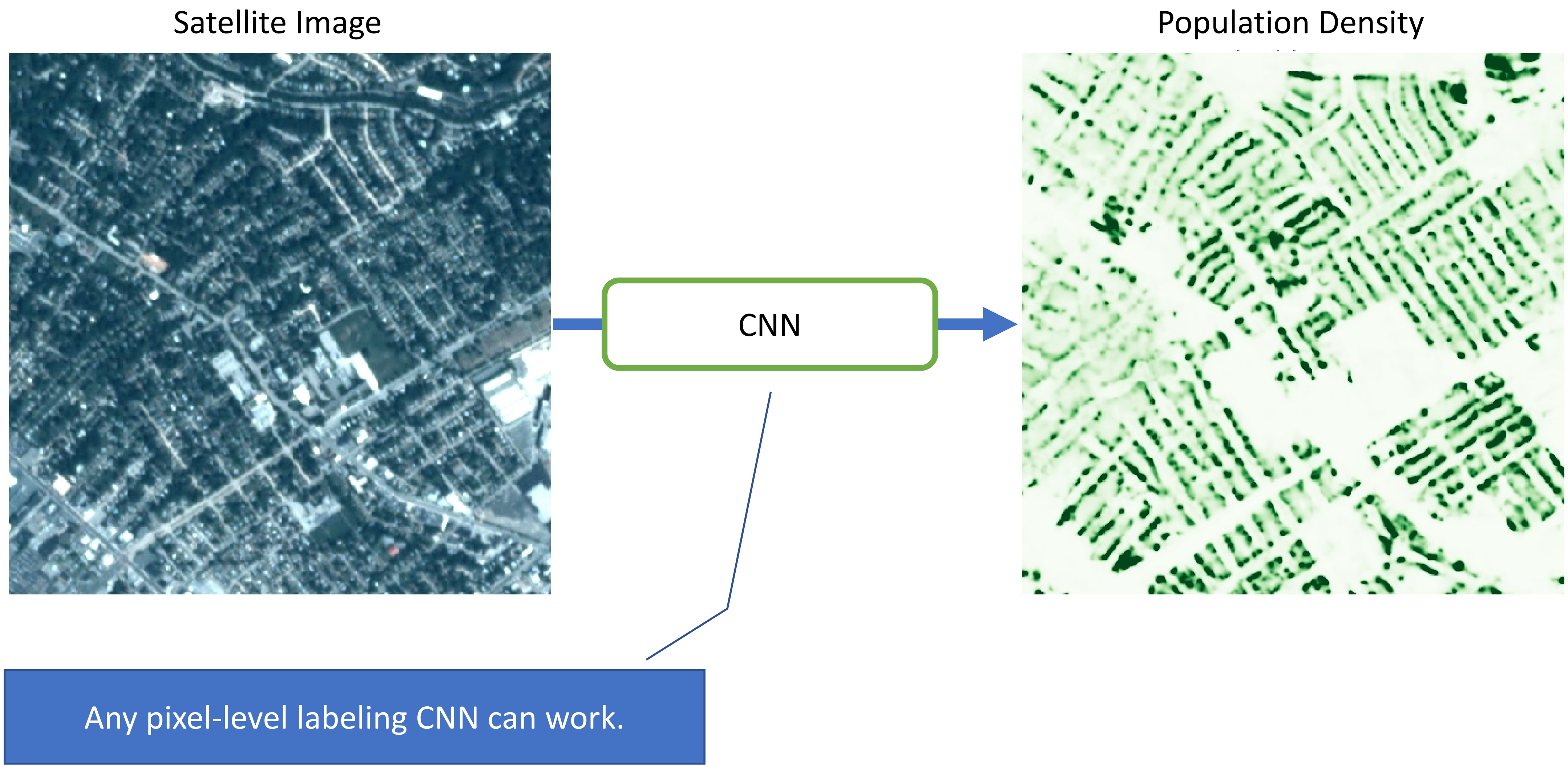
Manual Census + Dasymetric Mapping

- Improves spatial distribution (usually)
- Only redistributes densities
- Requires accurate foundation data
- Requires object-type specific assumptions

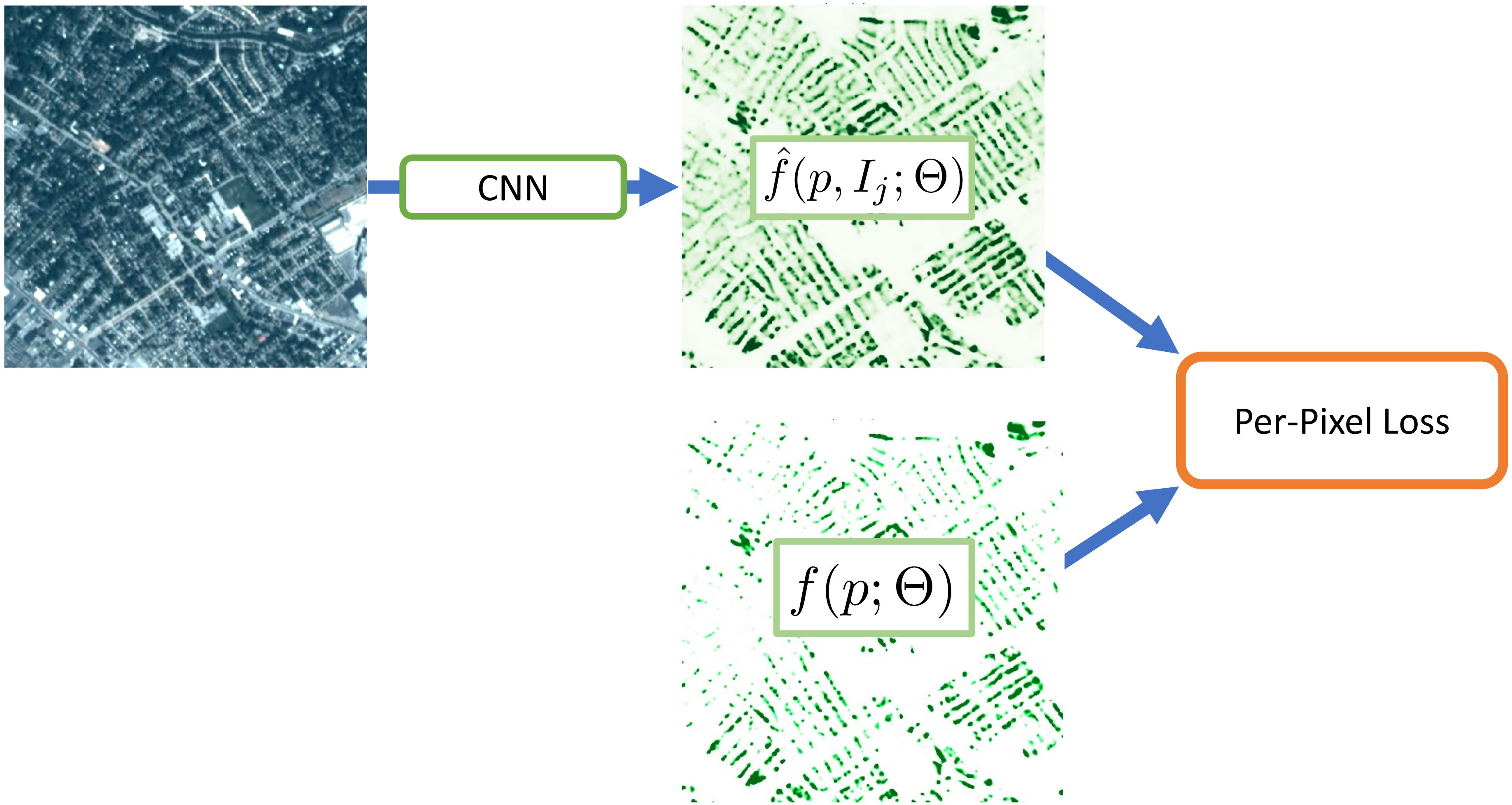


https://pubs.usgs.gov/fs/2008/3010/

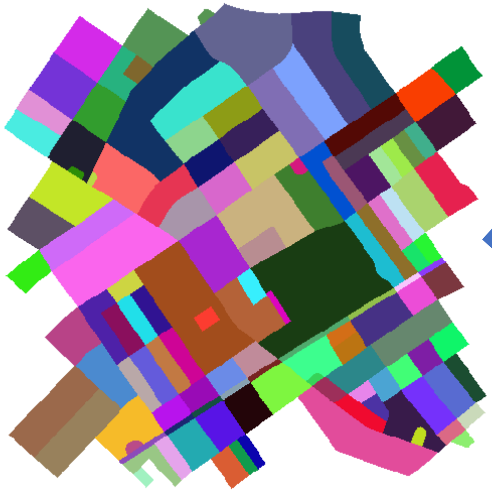
Our Approach: Predict Spatial Distributions Directly from Satellite Imagery



The Ideal Scenario: Pixel-Level Training Data



The Problem

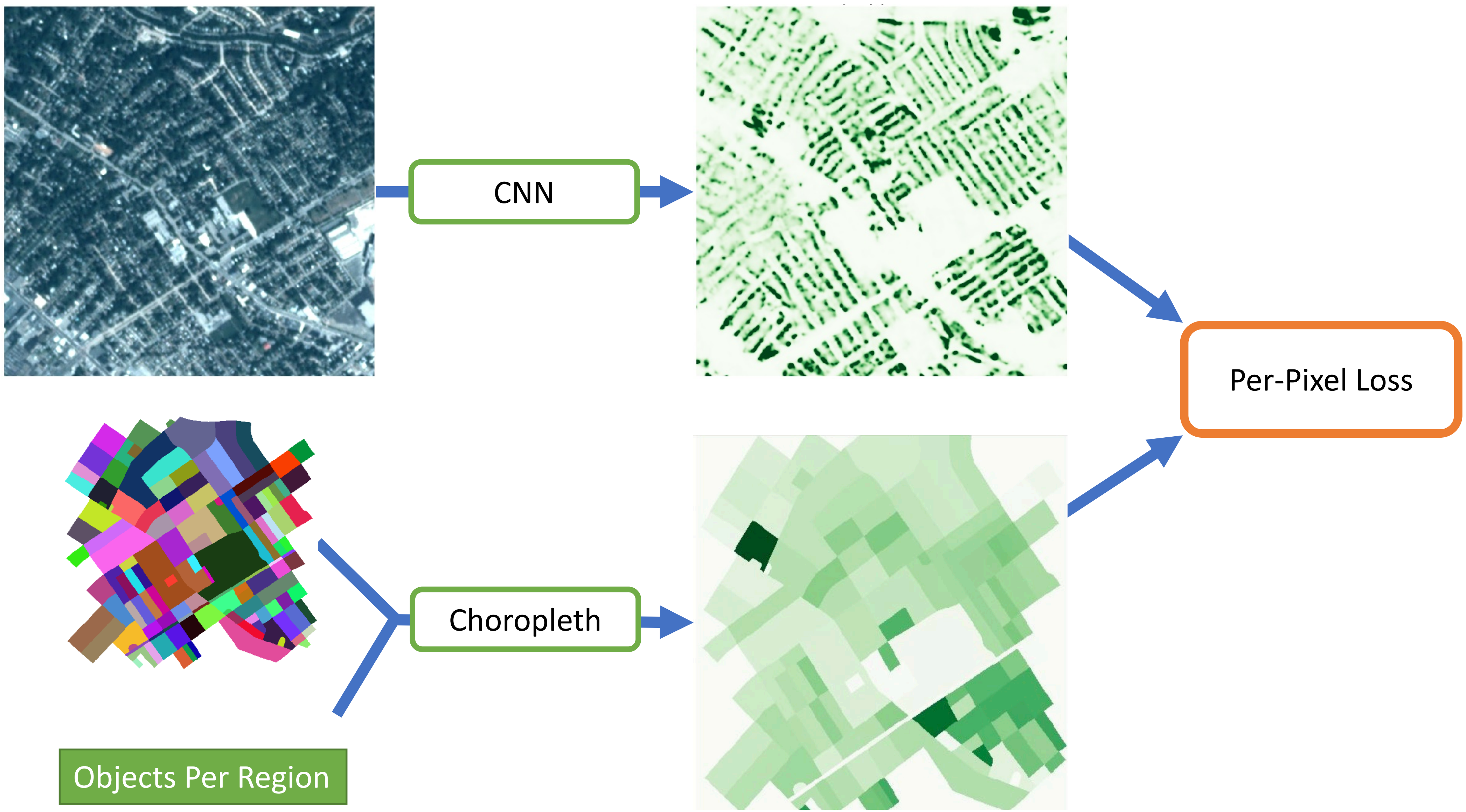


Census

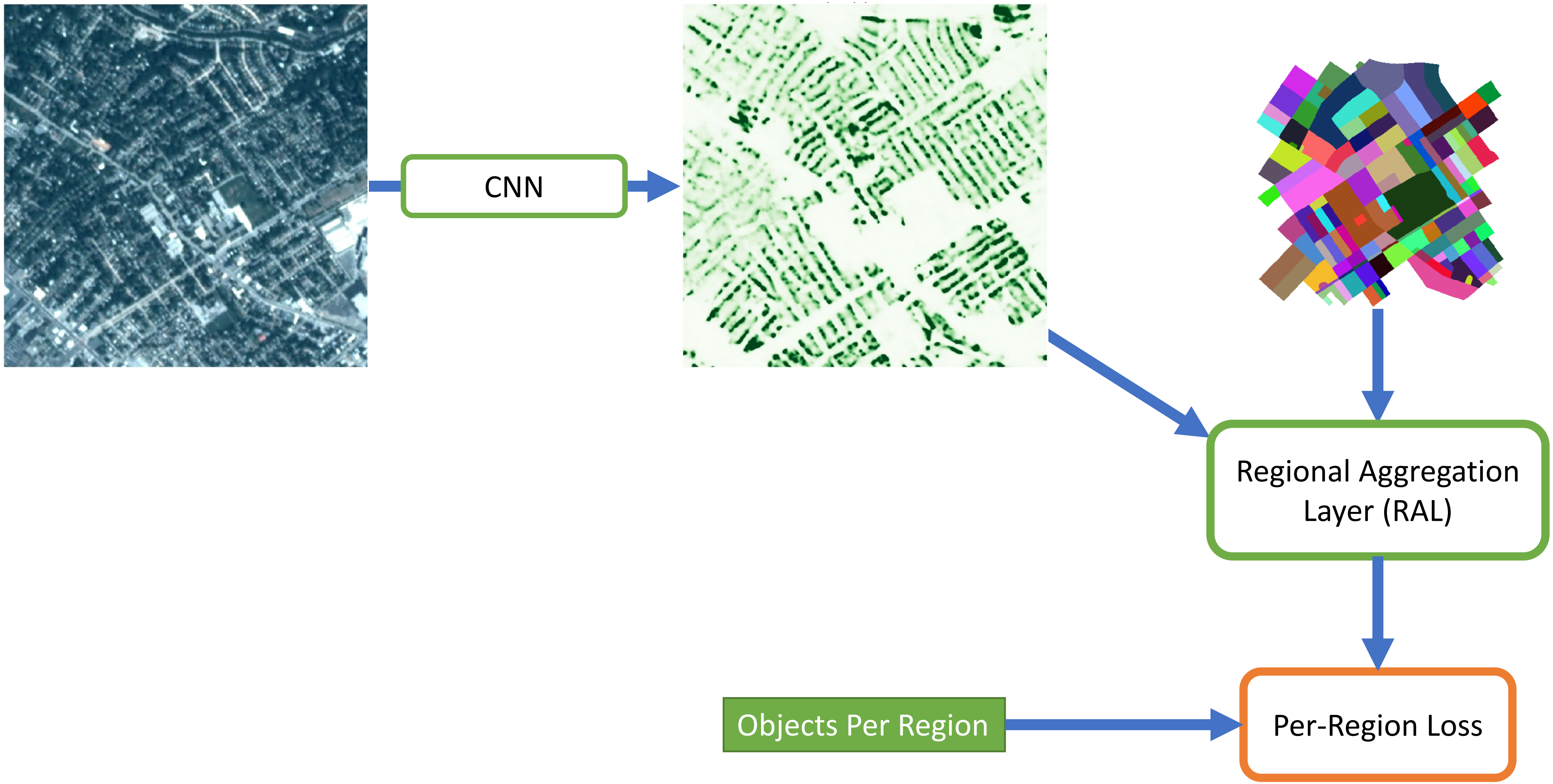
Objects Per Region

$$\begin{bmatrix} \sum_{p \in r_1} f(p;\Theta) \\ \sum_{p \in r_2} f(p;\Theta) \\ \vdots \\ \sum_{p \in r_N} f(p;\Theta) \end{bmatrix}$$

Naïve Approach: Assume Uniform Distribution (*unif*)

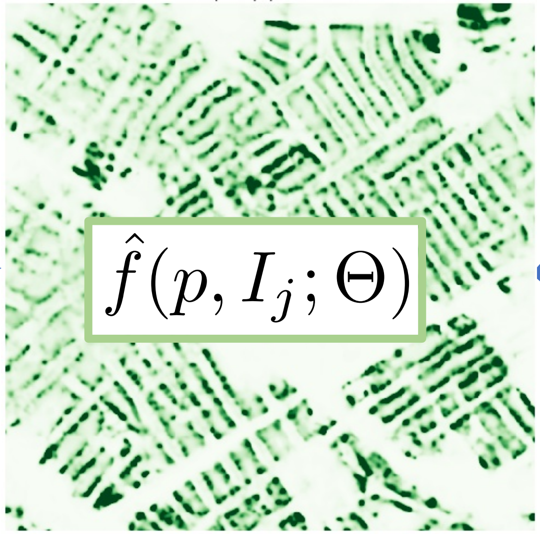


Our Approach: Regional Aggregation Layer (*RAL*)

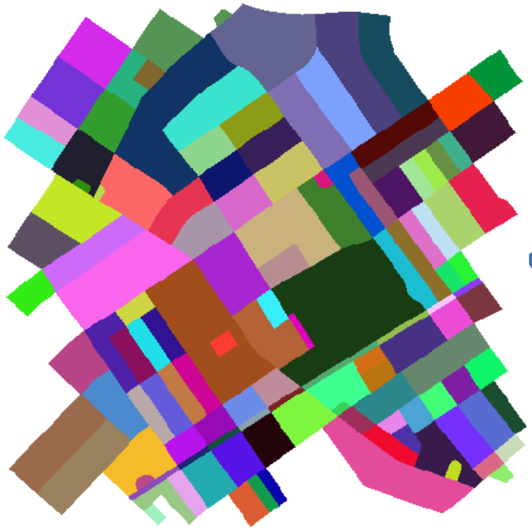


Our Approach: Regional Aggregation Layer (*RAL*)

CNN



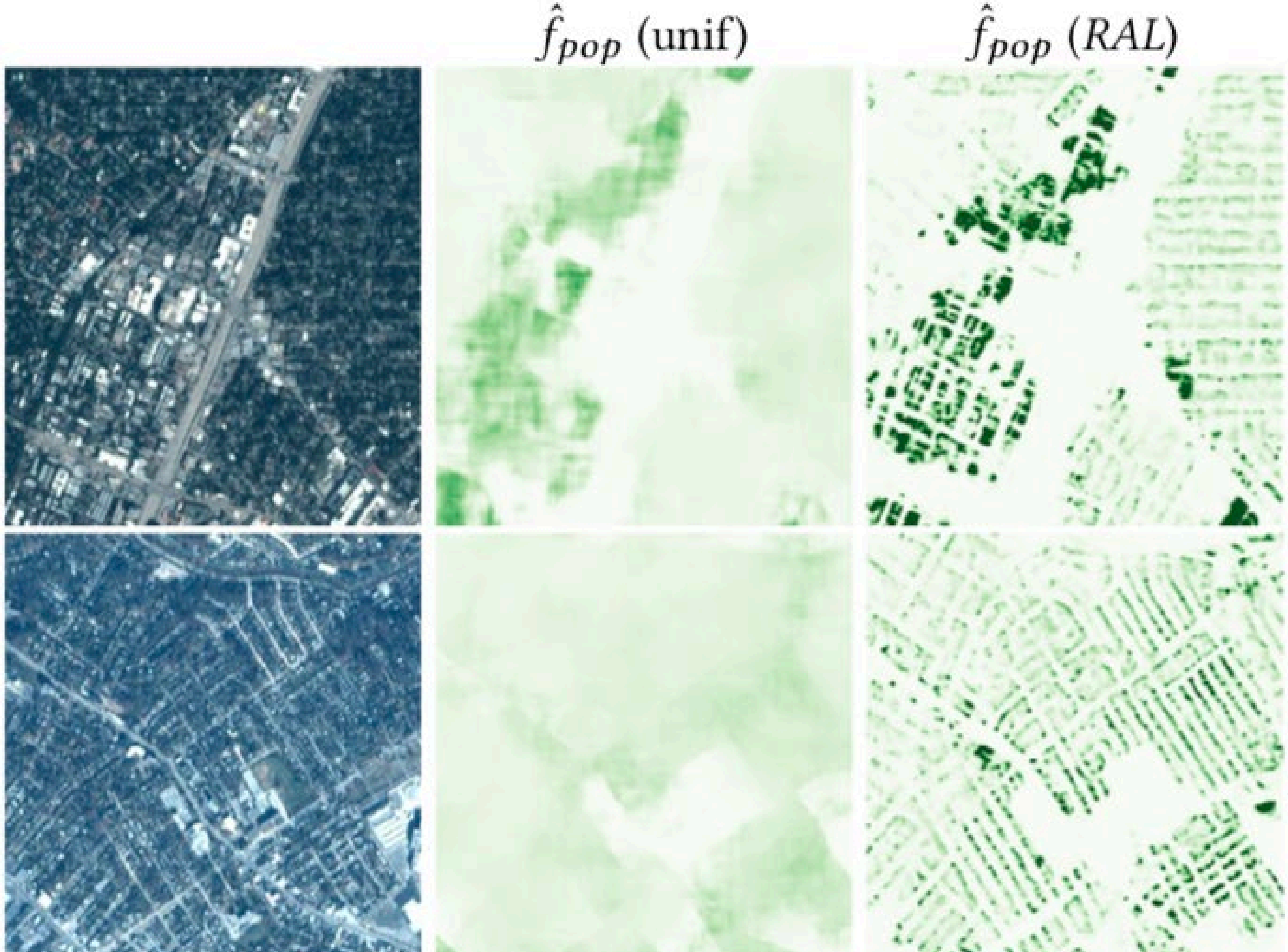
$\hat{f}(p, I_j; \Theta)$

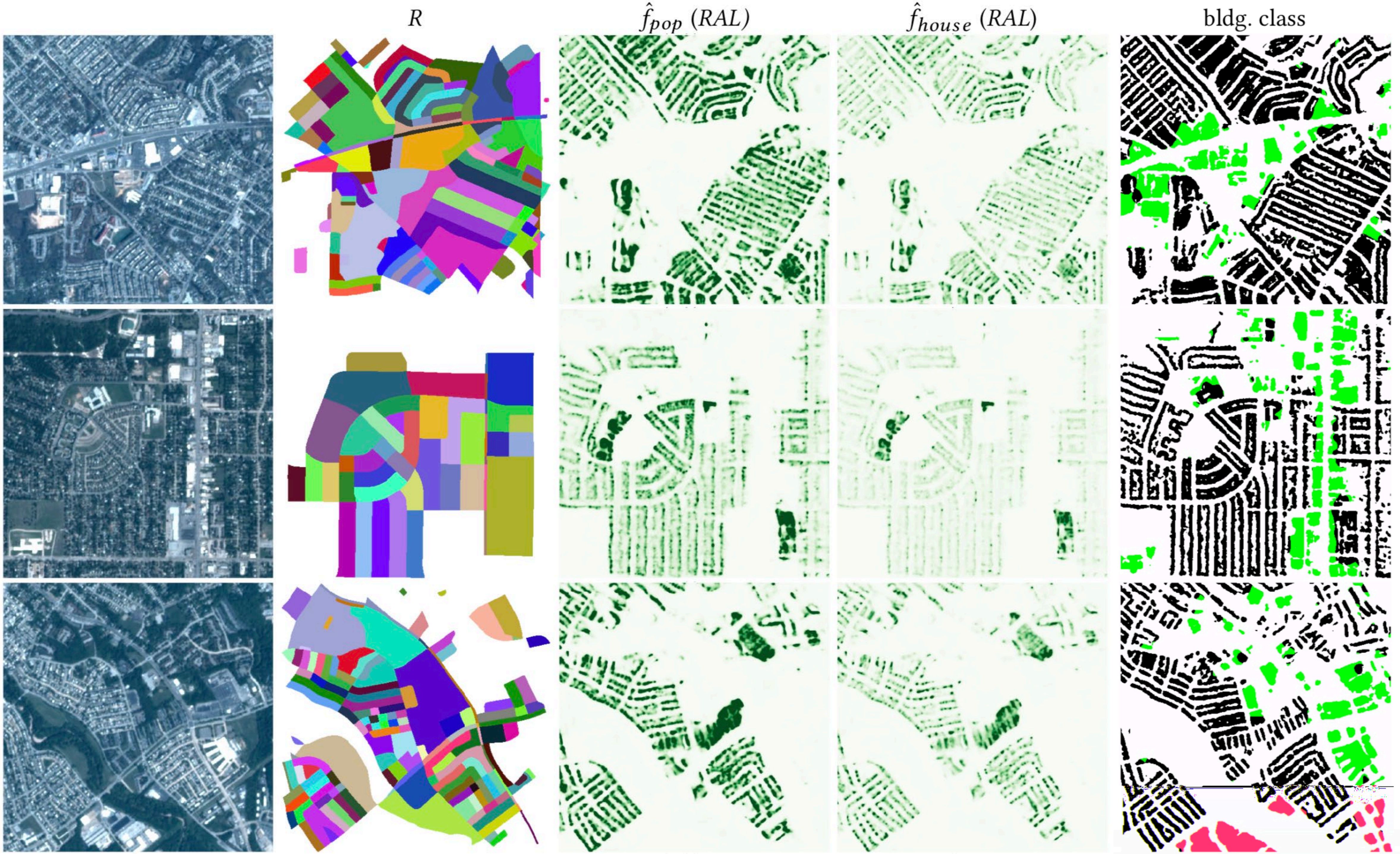


Regional Aggregation Layer (RAL)

(dense matrix multiply or a sparse summation)

$$\begin{bmatrix} \sum_{p \in r_1} \hat{f}(p, I_j; \Theta) \\ \sum_{p \in r_2} \hat{f}(p, I_j; \Theta) \\ \vdots \\ \sum_{p \in r_N} \hat{f}(p, I_j; \Theta) \end{bmatrix}$$


$\hat{f}_{pop}$ (unif)
$\hat{f}_{pop}$ (RAL)



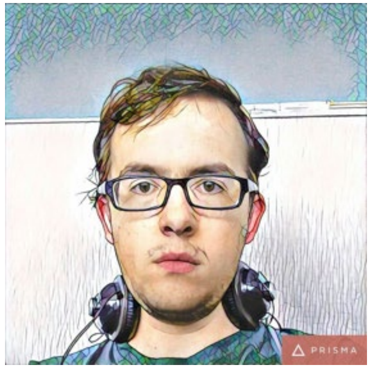
R
$\hat{f}_{pop}$ (RAL)
$\hat{f}_{house}$ (RAL)
bldg. class

Conclusions

- In the midst of a revolution
- Driven by deep learning (and availability of digital data)
- Practical tools for many domains, but requires teamwork

Thank You

Menghua



Connor

Scott

Tawfiq

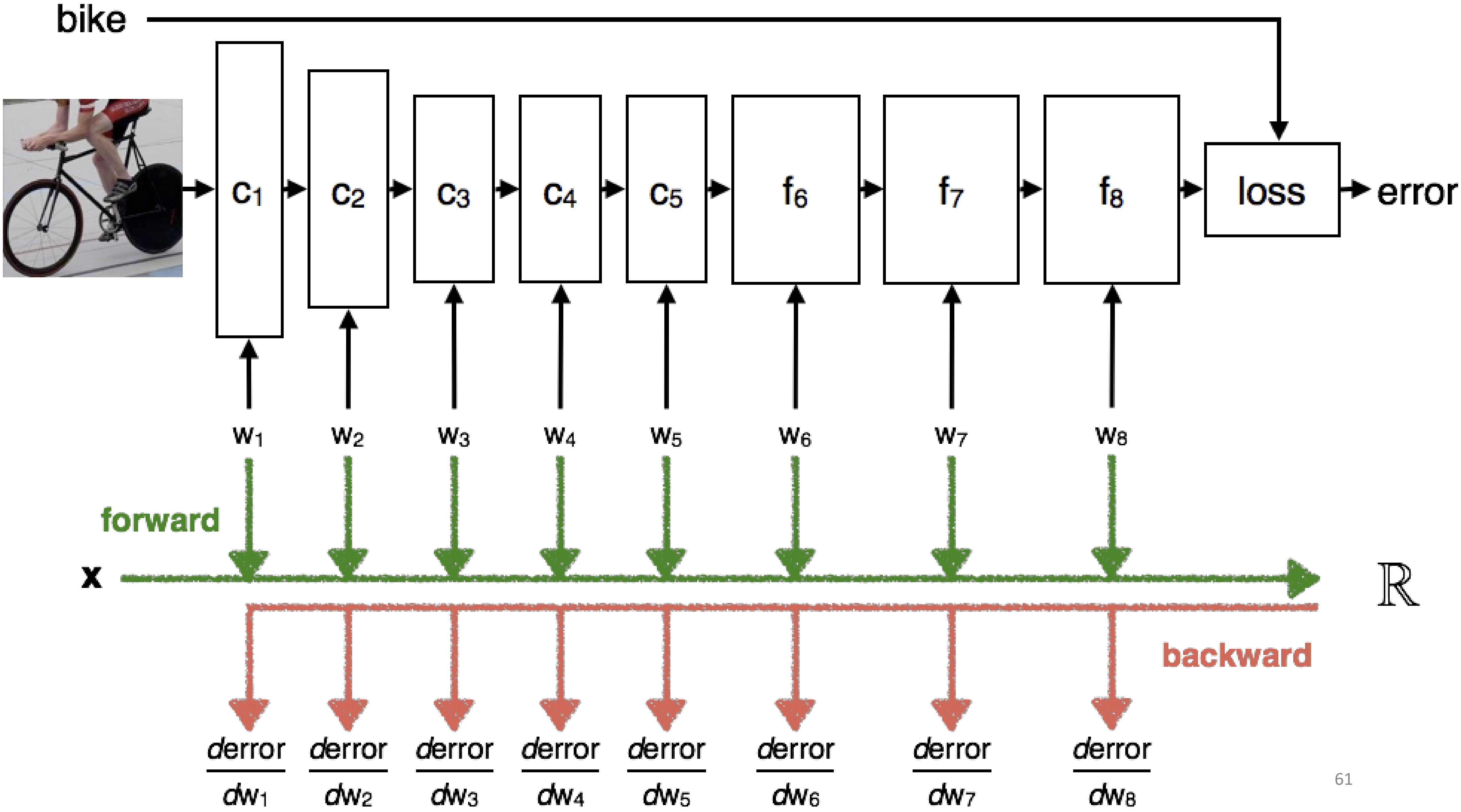
Zach

Usman

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1553116. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
Additional Funding Acknowledgements: Google Faculty Research Award, IARPA (Finder), AWS Research Education Grant, NVIDIA Hardware Donation

Questions?

Backup Slides



bike
c1
c2
c3
c4
c5
f6
f7
f8
loss
error
w1
w2
w3
w4
w5
w6
w7
w8
forward
x
ℝ
backward
derror/dw1
derror/dw2
derror/dw3
derror/dw4
derror/dw5
derror/dw6
derror/dw7
derror/dw8

Easy to Use Pre-Trained Neural Networks

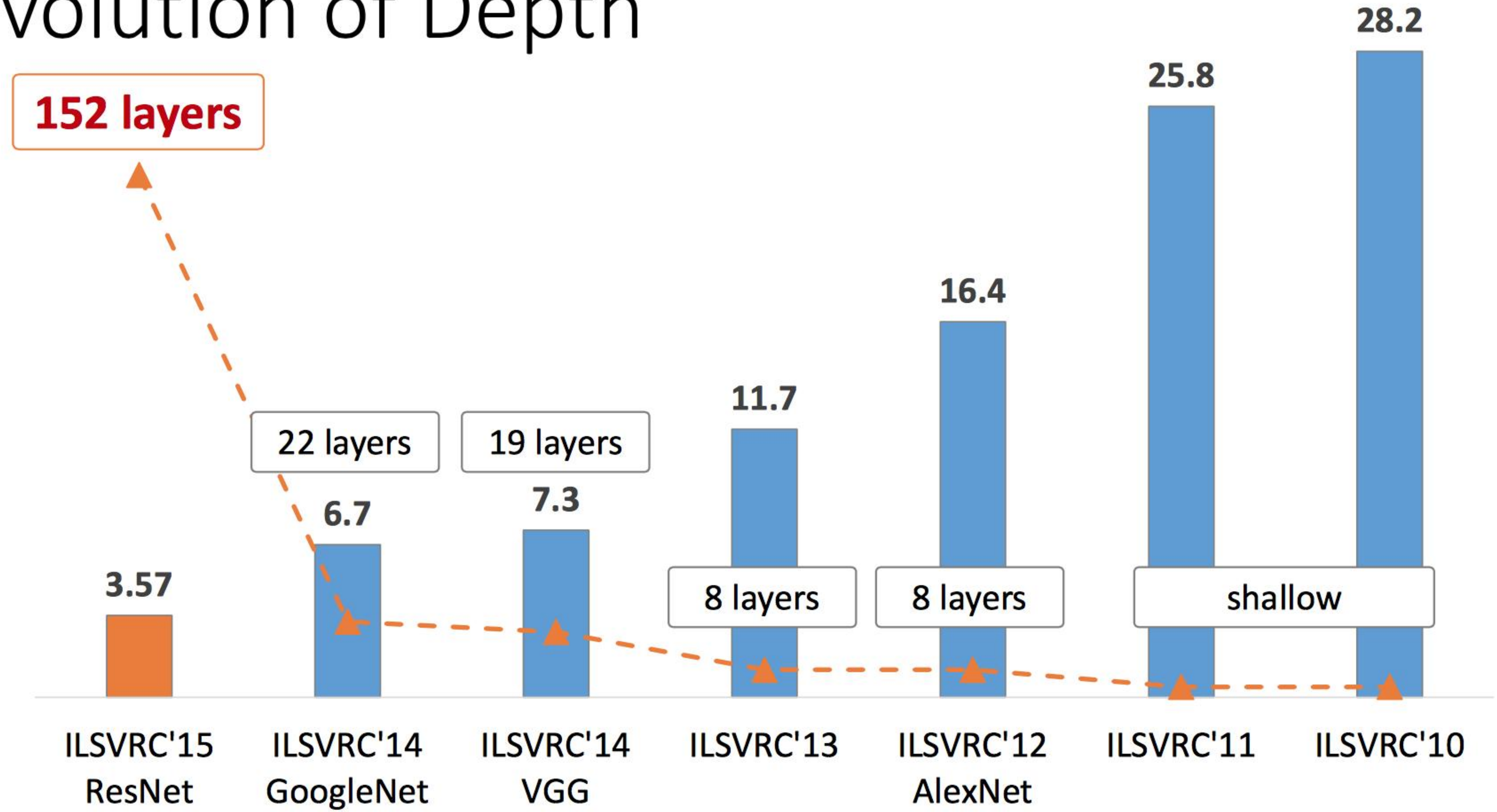
Python Code:

```
# load the trained model
model = ResNet50(weights='imagenet’)

# load the image
img = load_image(“image.jpg”)

# make the prediction
preds = model.predict(x)
```

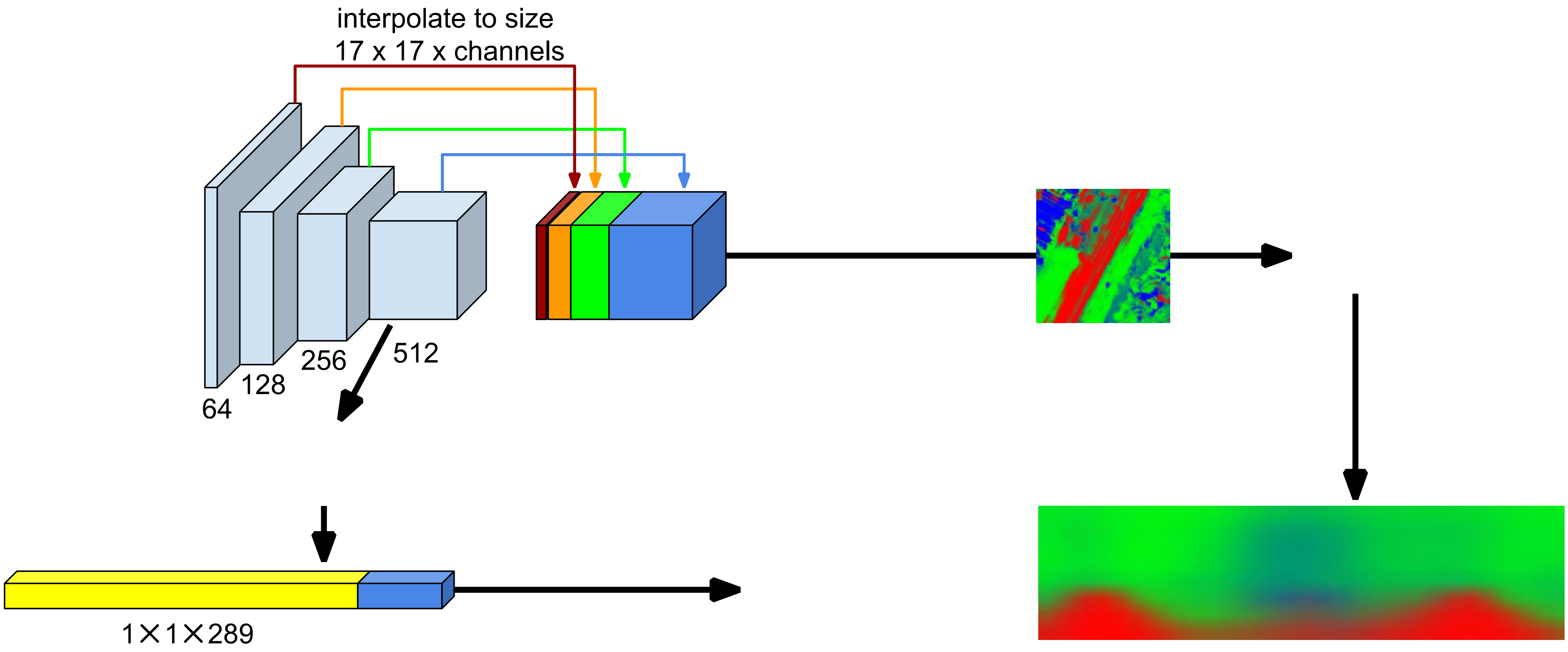
Predictions:

- 48%: sorrel
- 38%: worm fence
- 6%: ox
- …

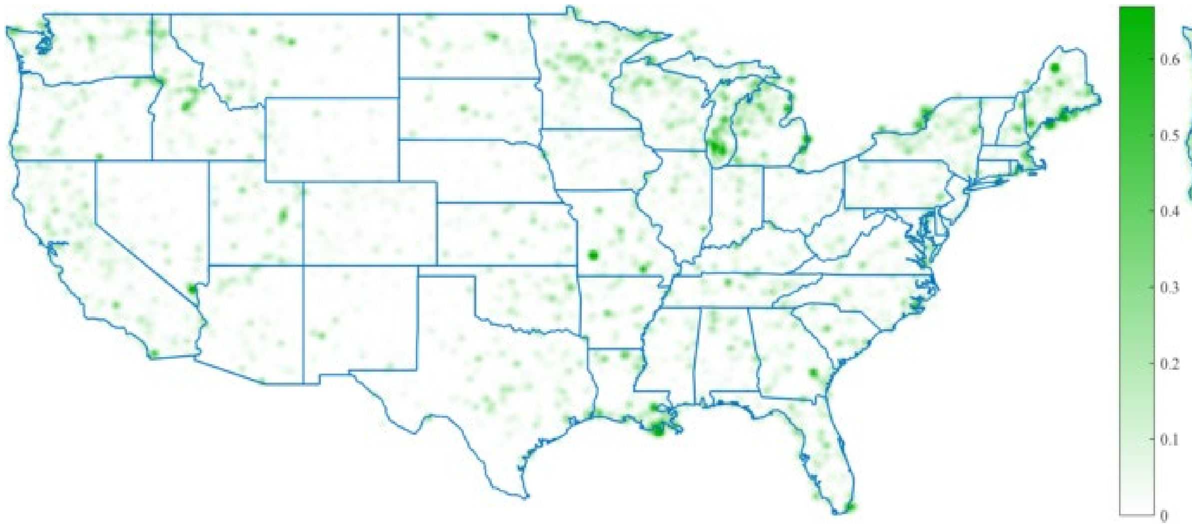
Revolution of Depth



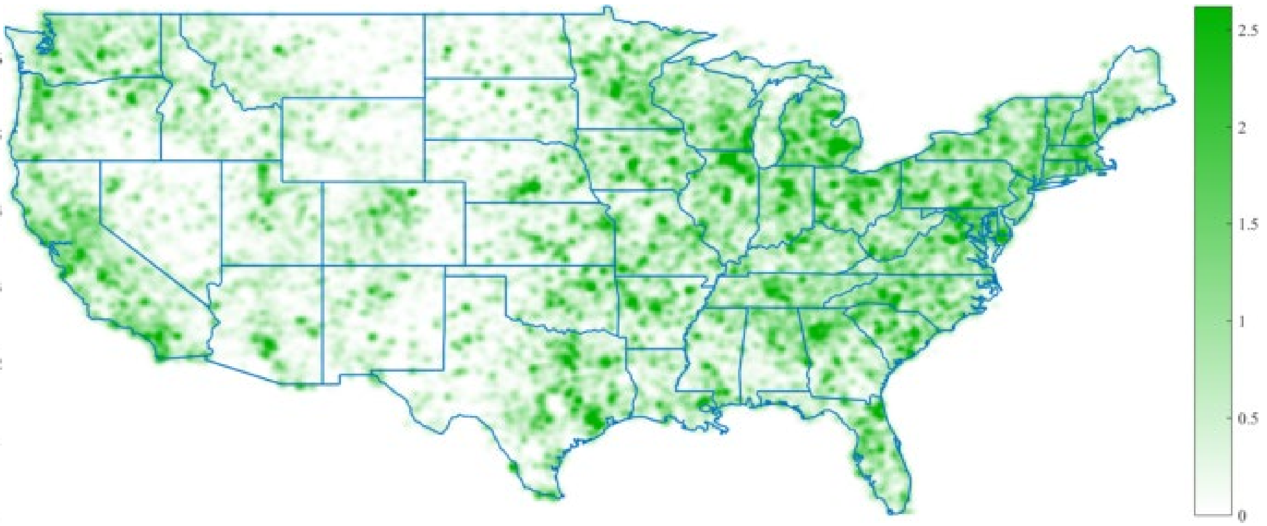
ImageNet Classification top-5 error (%)


interpolate to size
17 x 17 x channels
64
128
256
512
1×1×289

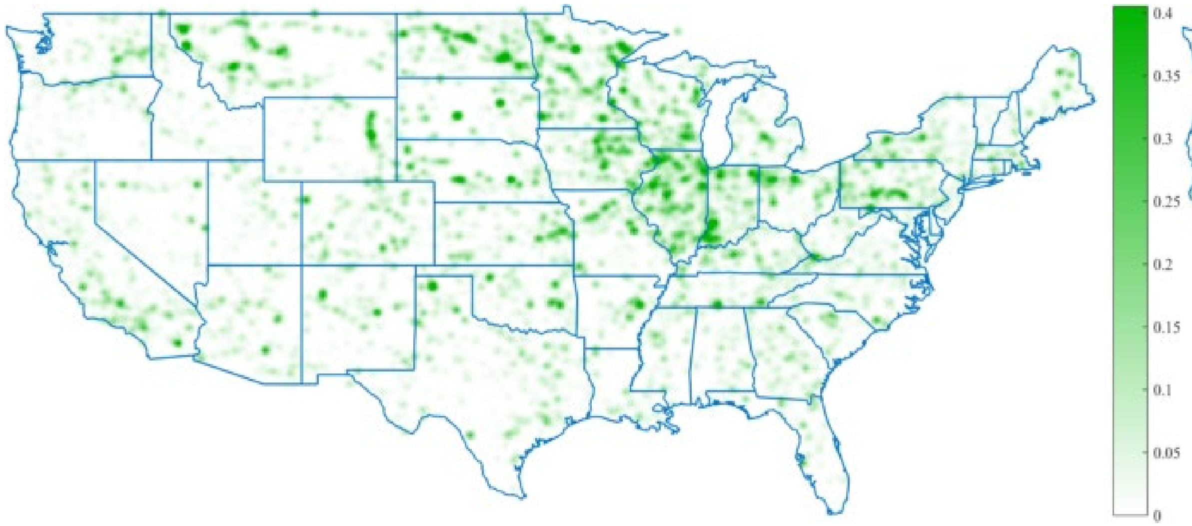
Class-Conditional Expectation of “Objects Per Image”



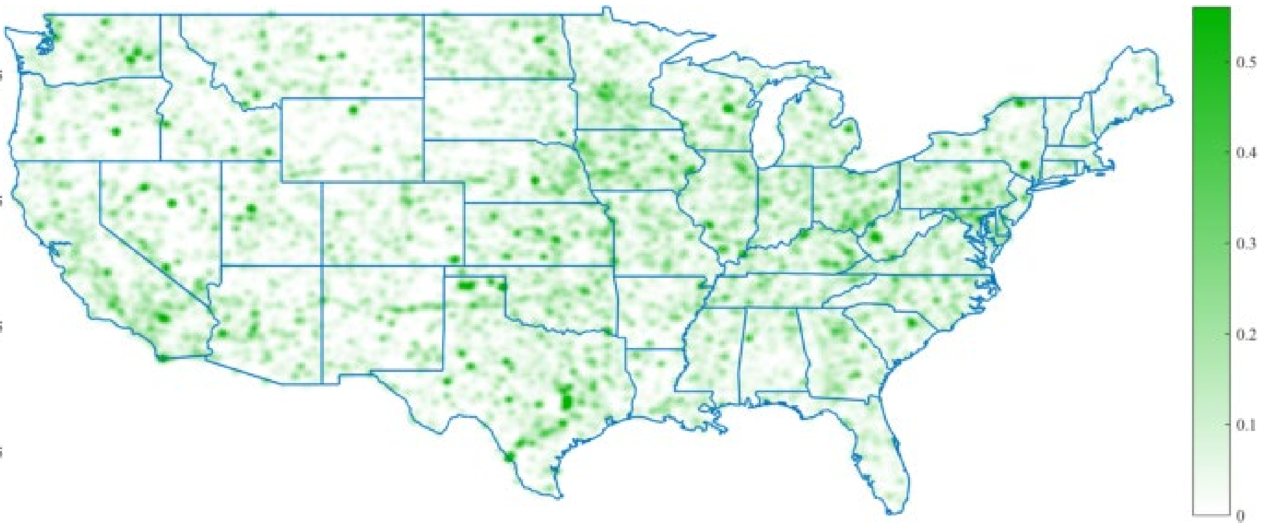
Boat



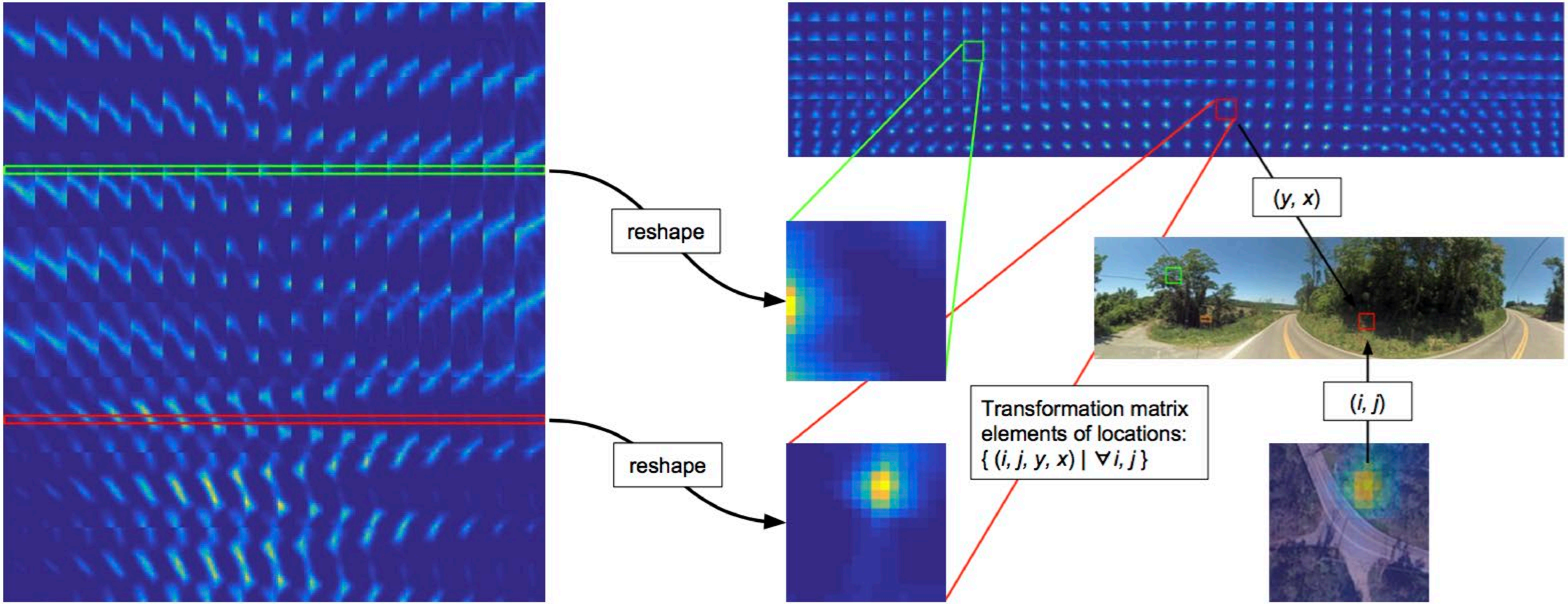
Person



Train

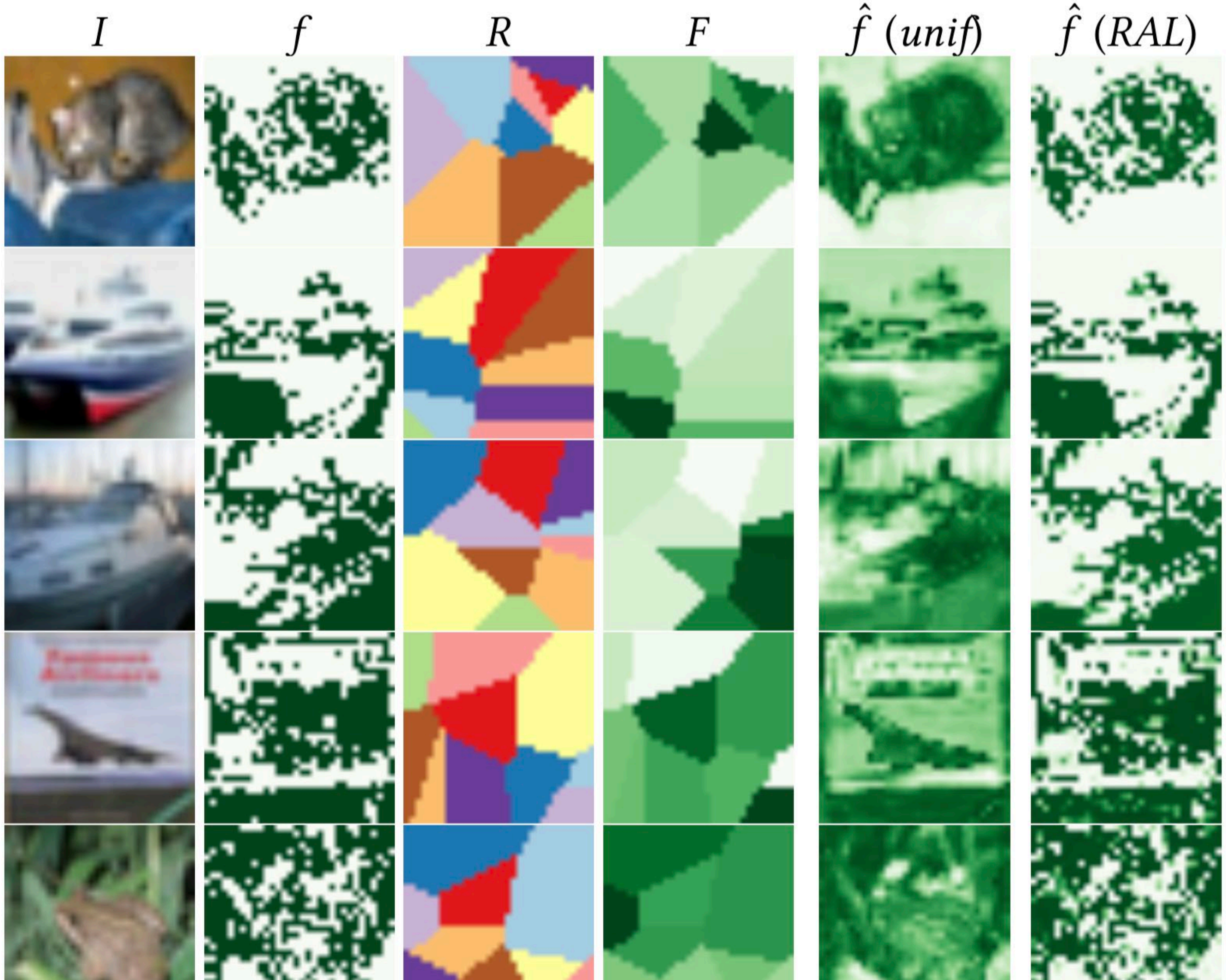


Truck

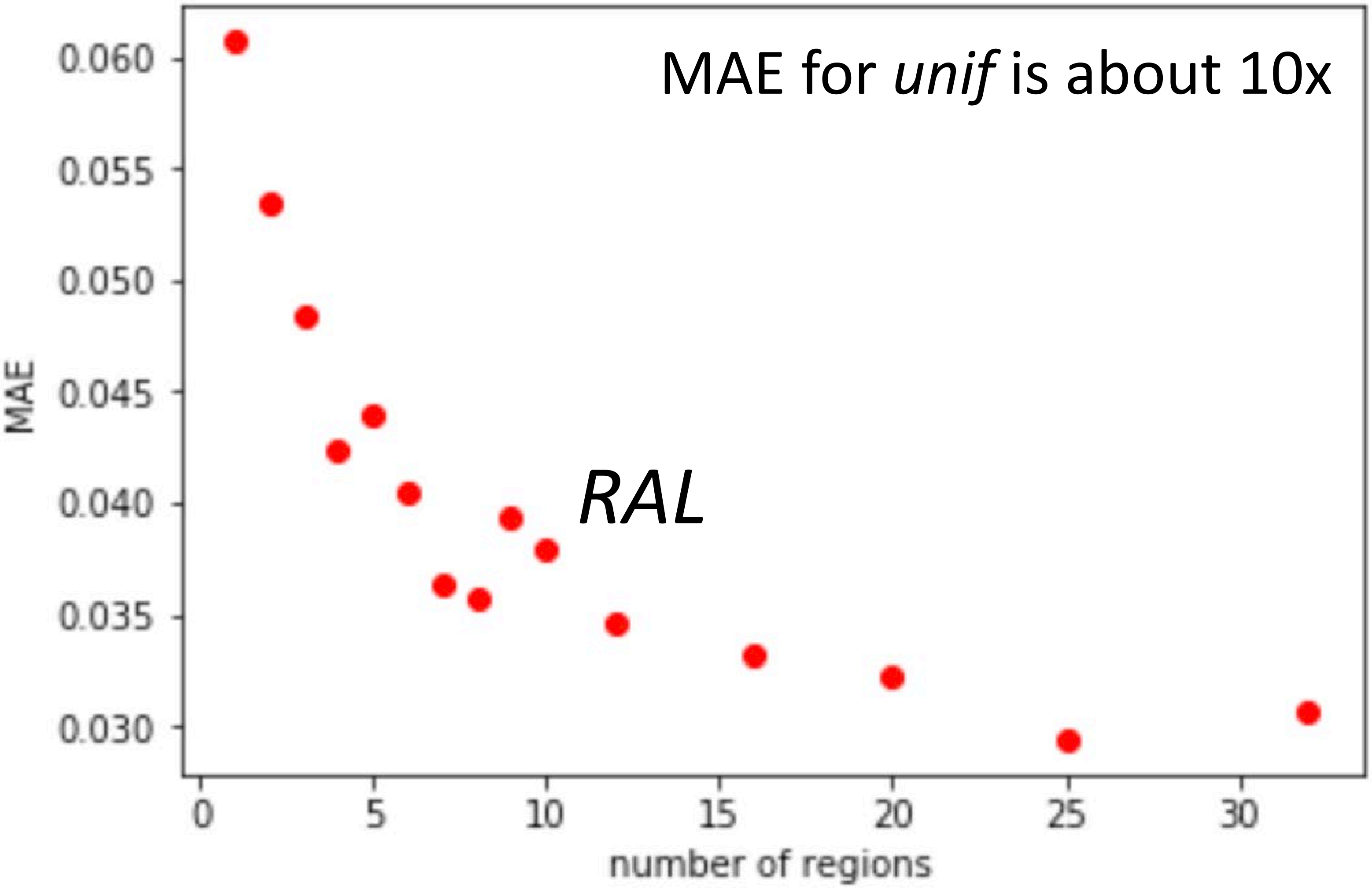

reshape
reshape
(y, x)
Transformation matrix
elements of locations:
{ (i, j, y, x) | ∀ i, j }
(i, j)

Evaluation (Synthetic Data)

- Setup:
  - **Imagery:** CIFAR (~85/15% split)
  - **Density:** random, binary, based on pixel values
  - **Regions:** 10 random Voronoi cells.
- Network:
  - **Architecture:**
    - Shallow CNN w/ 1x1 convolutions
    - “Softplus” activation on output
  - **Training:**
    - **Loss:** mean average error (MAE)
    - Standard optimization method

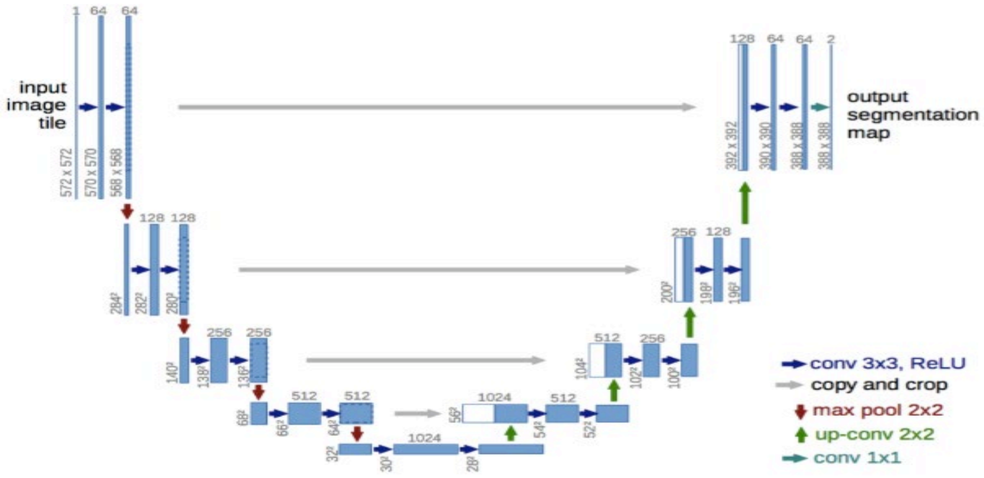


Quantitative Results

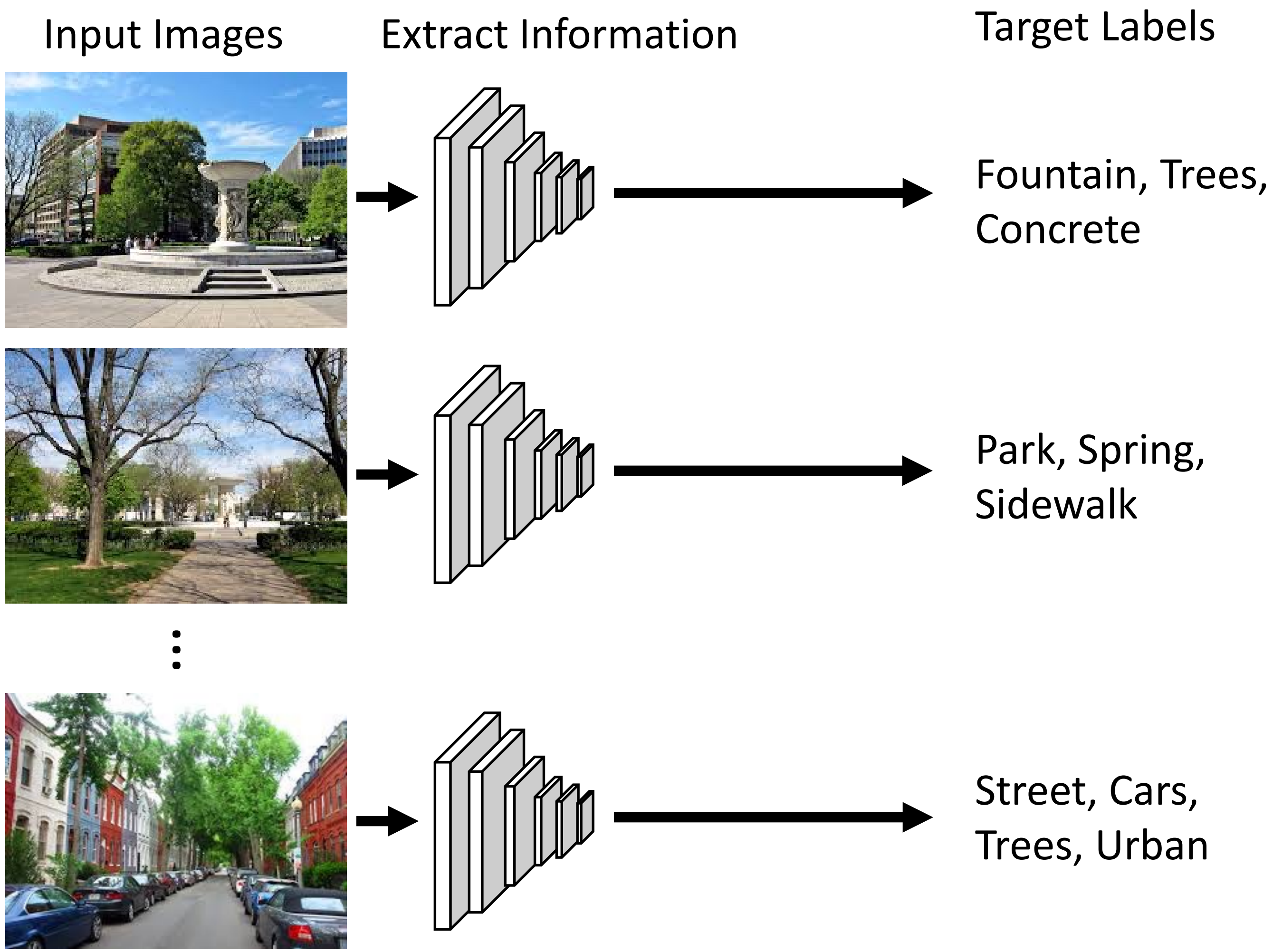


Evaluation (Census Data)

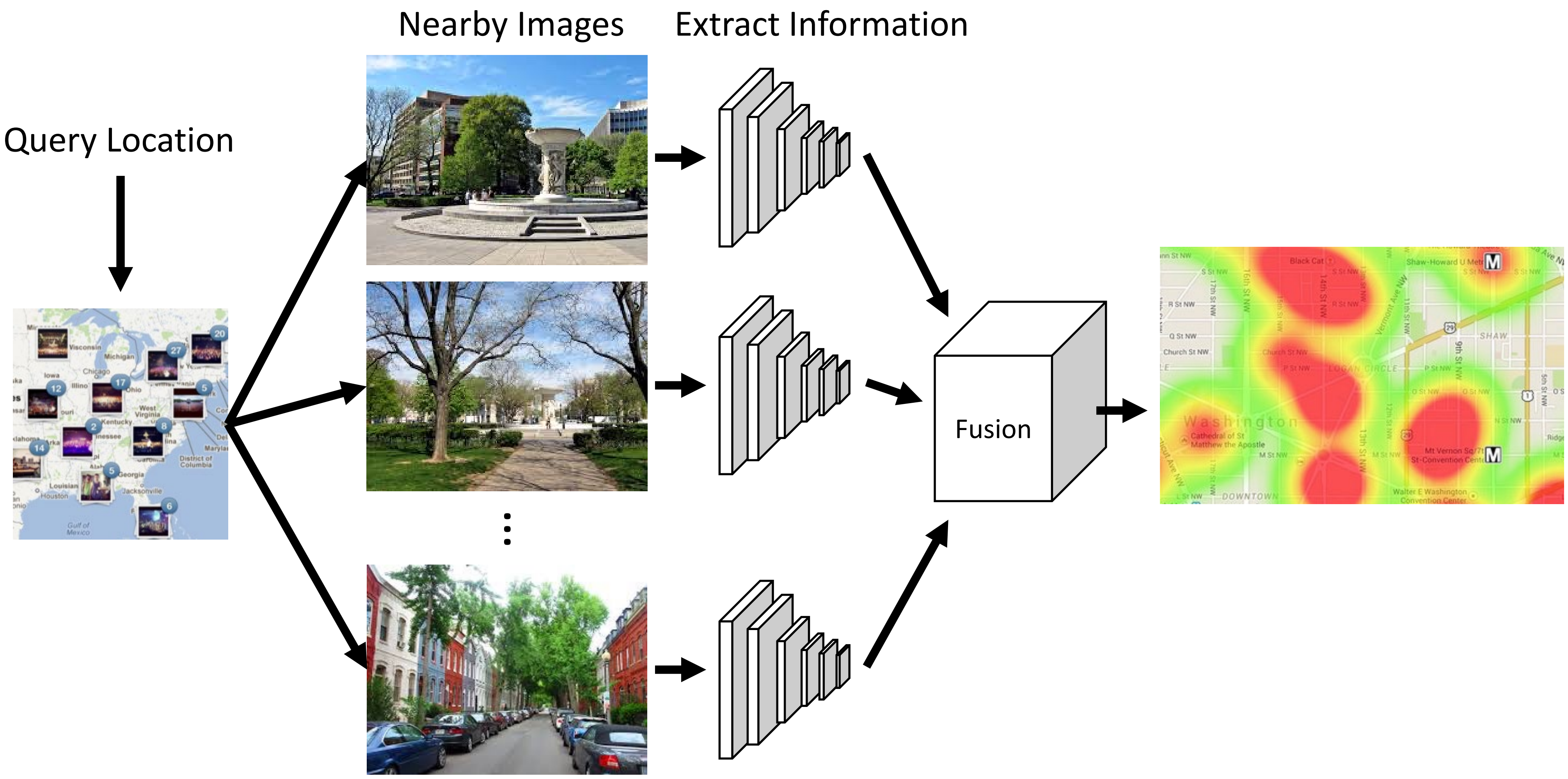
- Setup:
  - **Labels:** 2010 US Census
    - Housing and Population Counts (block group)
    - **Train**: 11 cities (~14,000 km^2)
    - **Test**: Dallas and Baltimore (~3,000 km^2)
  - **Imagery:**
    - 3m (GSD) RGB Imagery from PlanetScope
- Network:
  - **Architecture:**
    - Standard U-Net Architecture (Ronneberger; MICCAI 2015)
    - "Softplus" activation on output
    - Two heads: population and housing counts
  - **Training:**
    - **Loss:** mean average error (MAE)
    - Standard optimization method



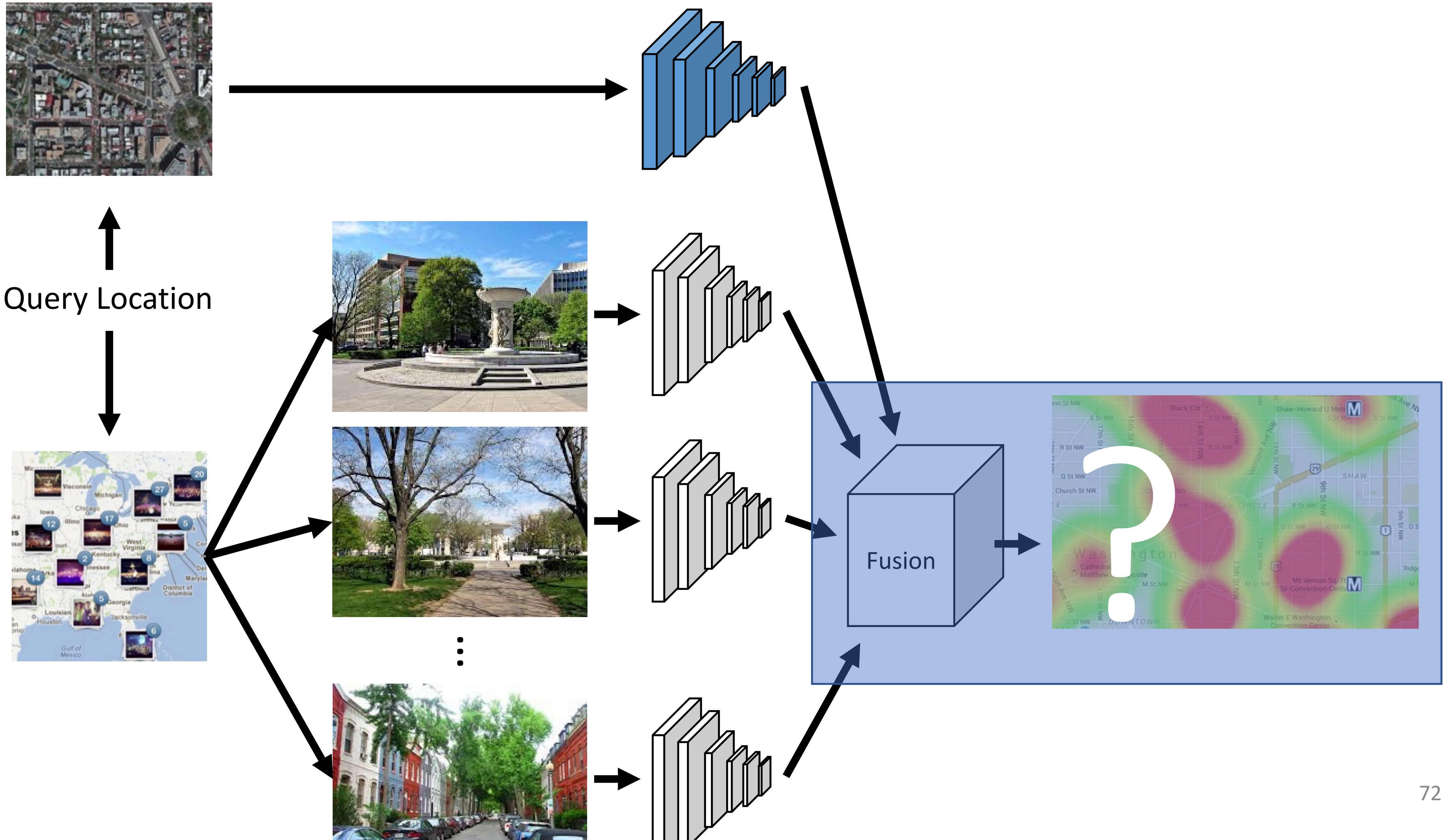
Standard Image-Driven Mapping



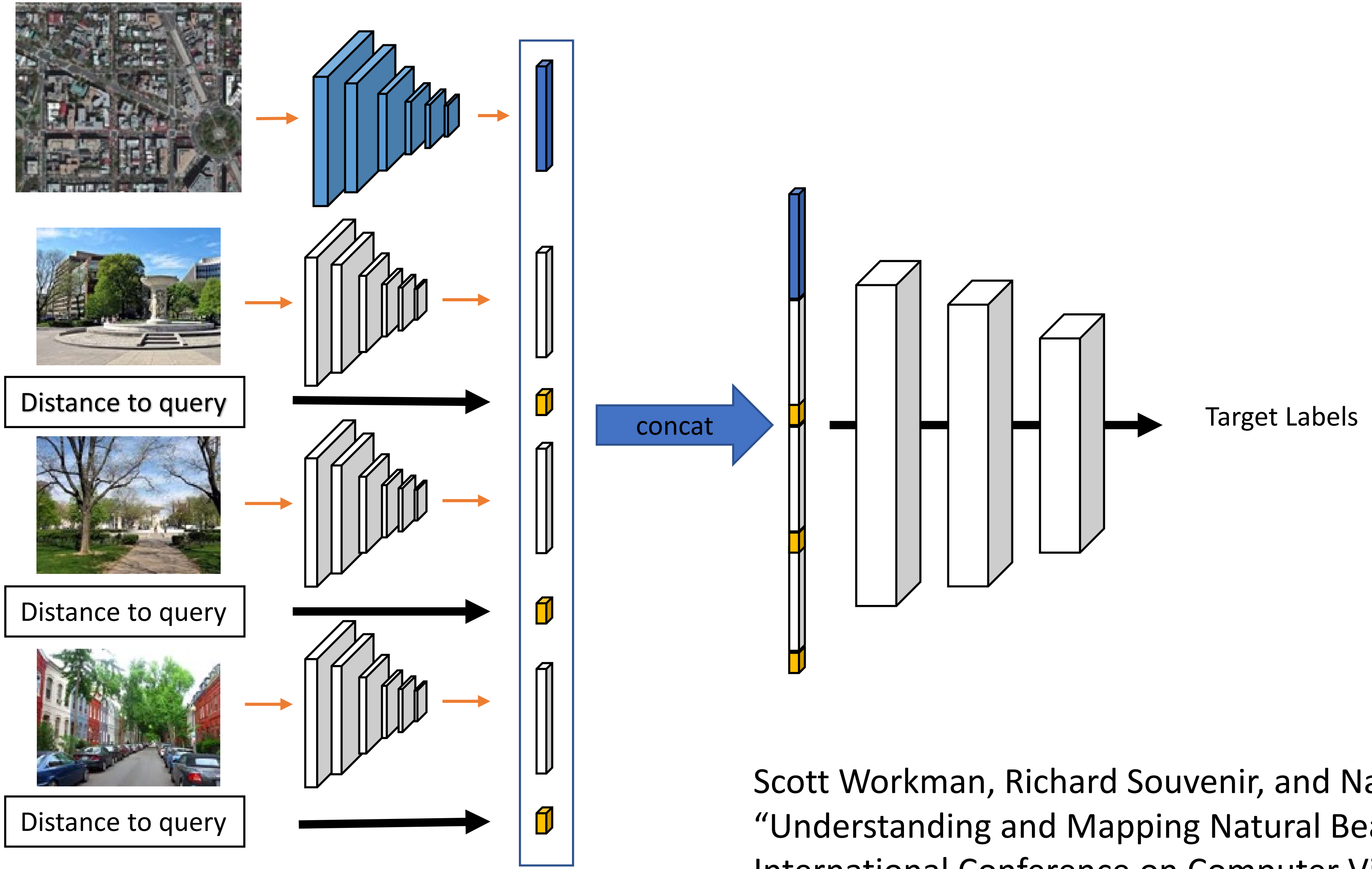
Standard Image-Driven Mapping



Crossview Image-Driven Mapping



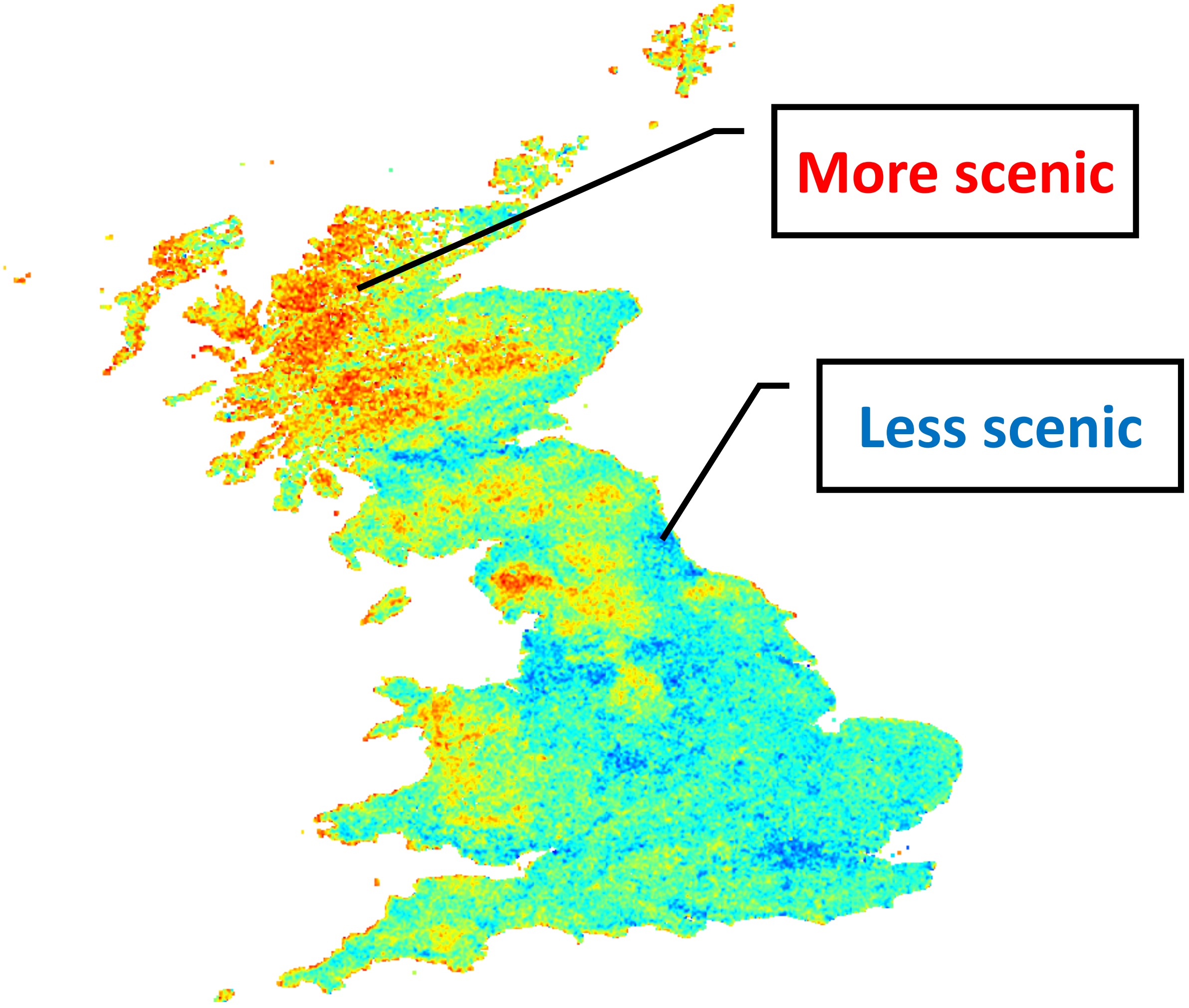
Architecture #1



Scott Workman, Richard Souvenir, and Nathan Jacobs, "Understanding and Mapping Natural Beauty," in IEEE International Conference on Computer Vision (ICCV), 2017.

Case Study: Mapping Natural Beauty

ScenicOrNot Dataset: 212,019 manually annotated geotagged ground-level images

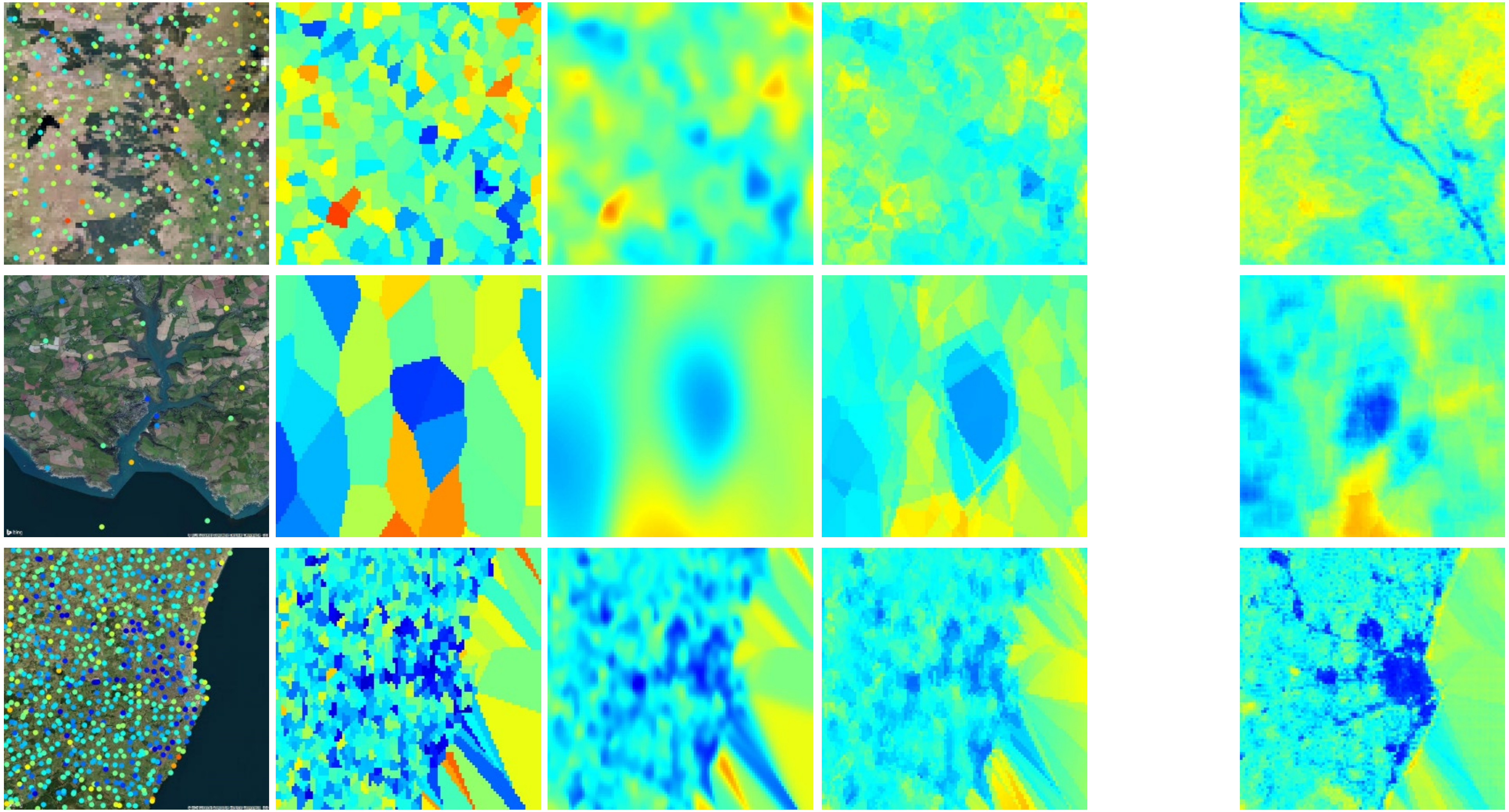


ROI

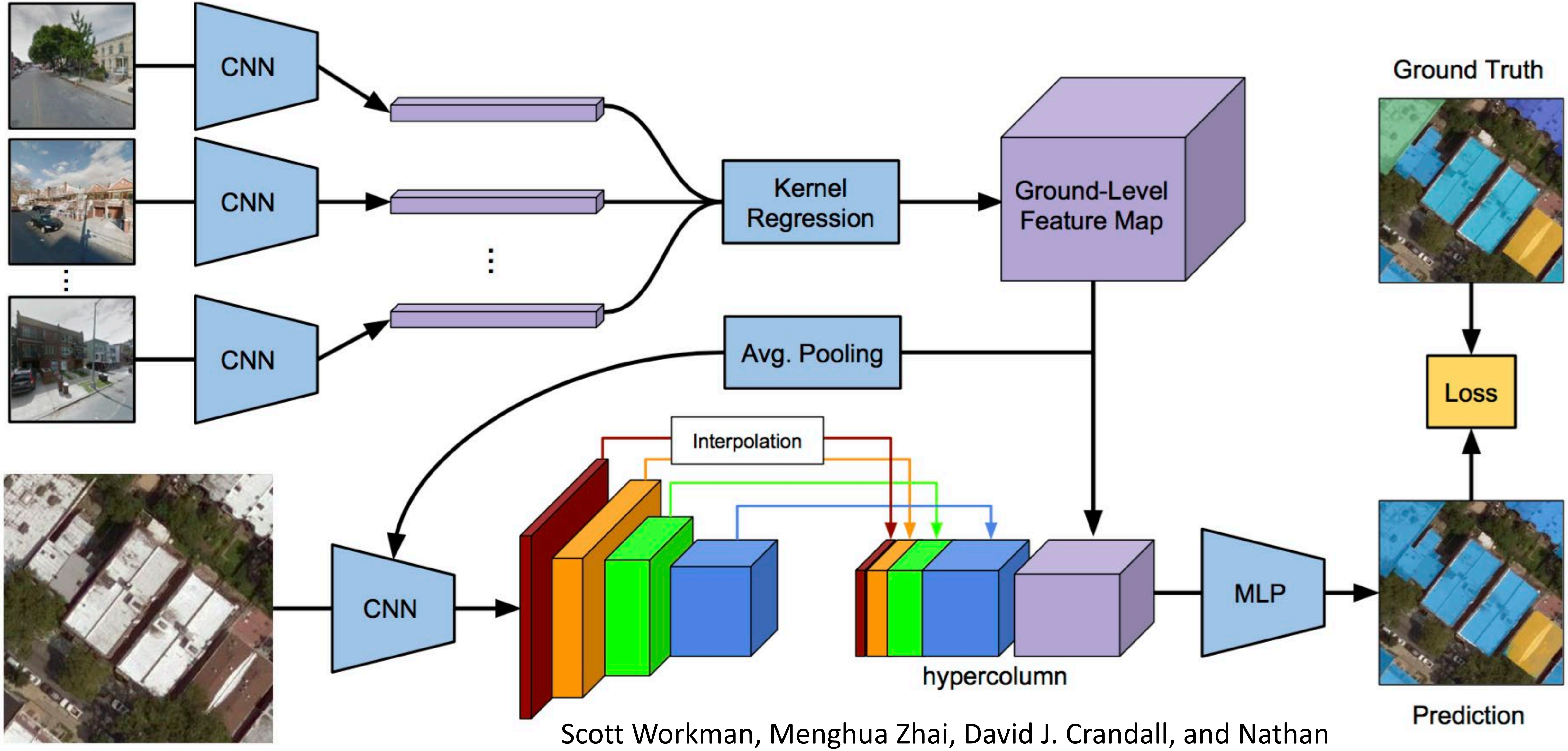
Using only ground-level images

Integrating Overhead

ROI 1NN LWA kNNet (ground-only) CVM (air+ground)



Architecture #2



Scott Workman, Menghua Zhai, David J. Crandall, and Nathan Jacobs, “A Unified Model for Near and Remote Sensing,” in IEEE International Conference on Computer Vision (ICCV), 2017.

Evaluation Dataset

Brooklyn:
- 73,921 non-overlapping overhead images (Bing Maps).
- 139,327 street-level panoramas (Google Street View).
- 4,361 overhead images held-out for testing.

Queens (held-out):
- 10,044 non-overlapping overhead images (Bing Maps).
- 38,603 street-level panoramas (Google Street View)

Pixel-Level Annotations (from NYC GIS)

Building Age

Building Function

Land Use

Results

Table 3: Queens evaluation results (top-1 accuracy).

	Age	Function	Landuse
random	06.80%	00.49%	08.41%
proximate	25.27%	22.50%	47.40%
grid	27.47%	26.62%	67.50%
remote	26.06%	29.85%	69.27%
unified (uniform)	29.68%	33.64%	68.08%
unified (adaptive)	**29.76%**	**34.13%**	**70.55%**