Penn Engineering
University of Pennsylvania

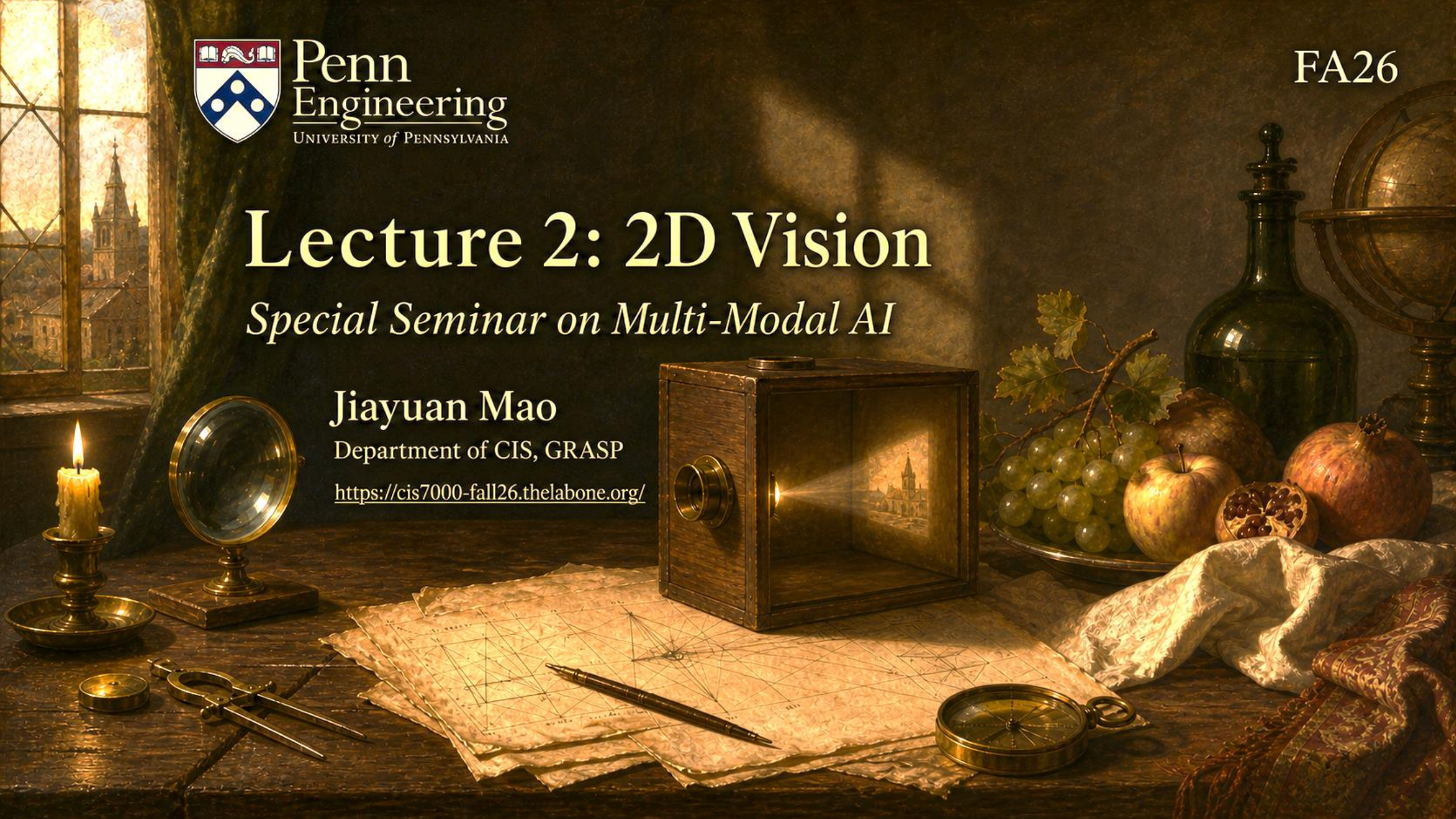
FA26

Special Seminar on
Multi-Modal AI

Jiayuan Mao

Department of CIS, GRASP

https://cis7000-fall26.thelabone.org/


Penn
Engineering
University of Pennsylvania
FA26
Lecture 2: 2D Vision
Special Seminar on Multi-Modal AI
Jiayuan Mao
Department of CIS, GRASP
https://cis7000-fall26.thelabone.org/

Logistics – Page 1

Still a lot of people in the waitlist
After today's lecture, please try to make your final decision

This is not an "accelerated" CV+NLP+Robotics+CompBio class
We expect students to have background in all these and we dive deeper

The course is project-based

Logistics – Page 2

Still a lot of people in the waitlist
After today's lecture, please try to make your final decision

Since we have a lot more people, we will do group discussions
After each lecture we will pair in groups of 4 to 6 students and discuss
Each final project should have <=4 students

Please read the papers yourself before coming to the class

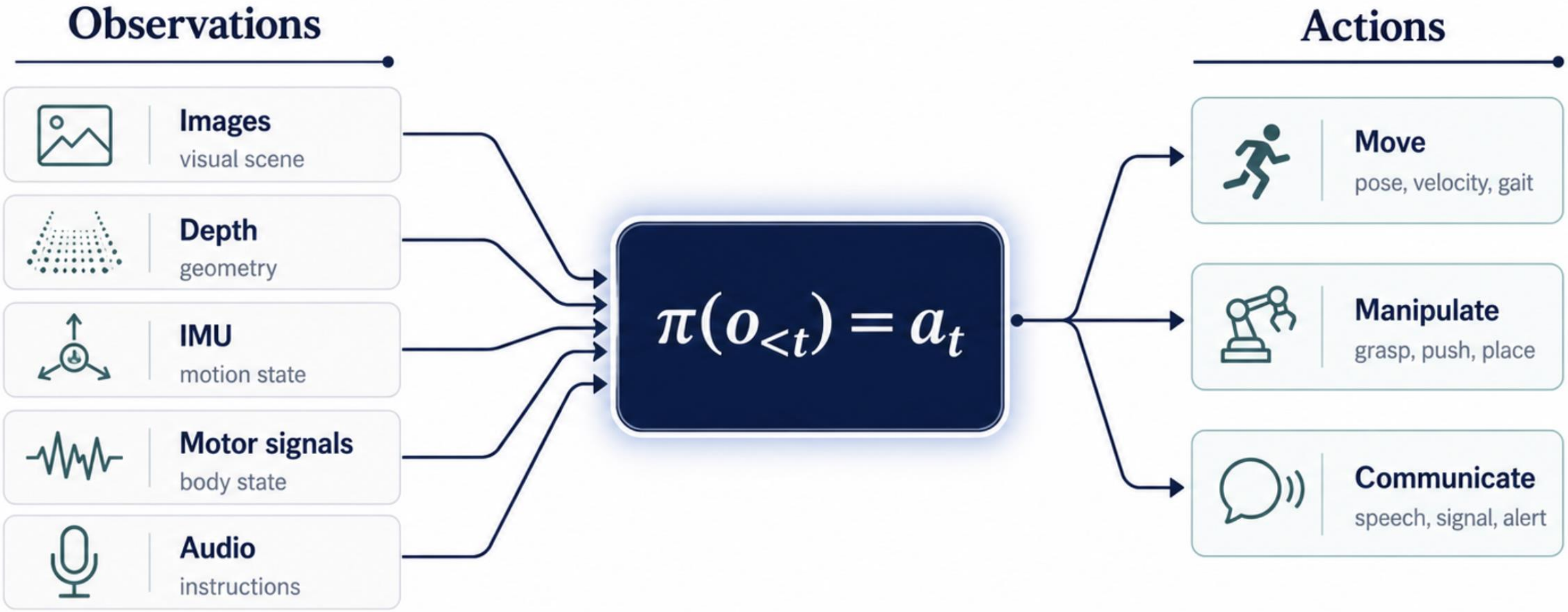
- We will prioritize discussing the principles behind these papers and how they extend to other problems
- Instead of focusing on the technical content

Goal of This Course

Use Machine Learning to Build Intelligent Agents

Abstract View of An Agent: Multimodal AI

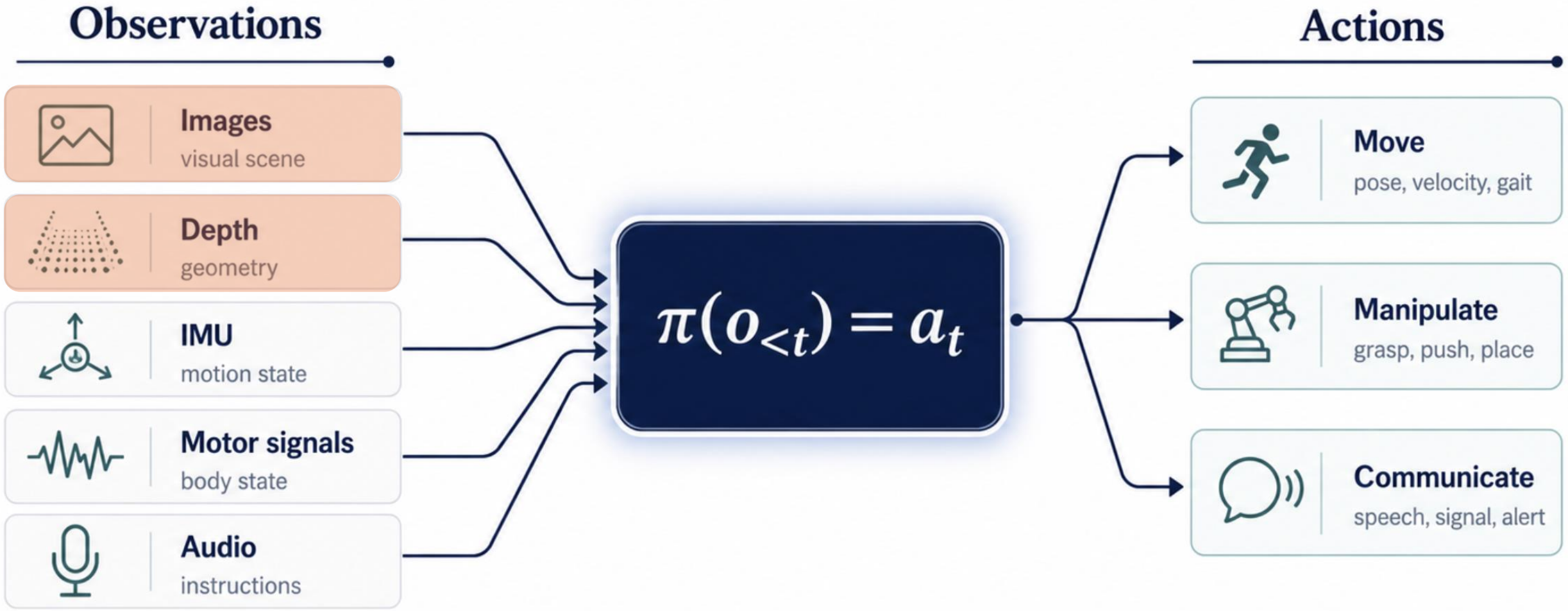
$$\pi: \text{observations} \rightarrow \text{action}$$



Notation: t – time. $o_{<t}$ - all observations before t. a_t - action at t.

Abstract View of An Agent: Multimodal AI

$$\pi: \text{observations} \rightarrow \text{action}$$



Notation: t – time. $o_{<t}$ - all observations before t. a_t - action at t.

Goal of This Lecture

Use Machine Learning to Sense and Understand Images

Recurring Questions in this Class

- How to formulate a multi-modal AI problem
  - What's the inputs, outputs, objectives, and evaluation metrics?
- What are the principles for solving these problems
  - What's the principles, advantages, and disadvantages of the representation and the computation models?

Outline for Today

Part 1: Principles for (Inverse) Image Formation

- Camera models (intrinsics and extrinsics)
- Using camera models as a natural supervision

Part 2: Principles for Image Recognition

- From image classification to object recognition
- Feature pyramids, iterative refinement, and relational computation

Part 3: Principles for Image Feature Extraction

- Relational features and contextual features

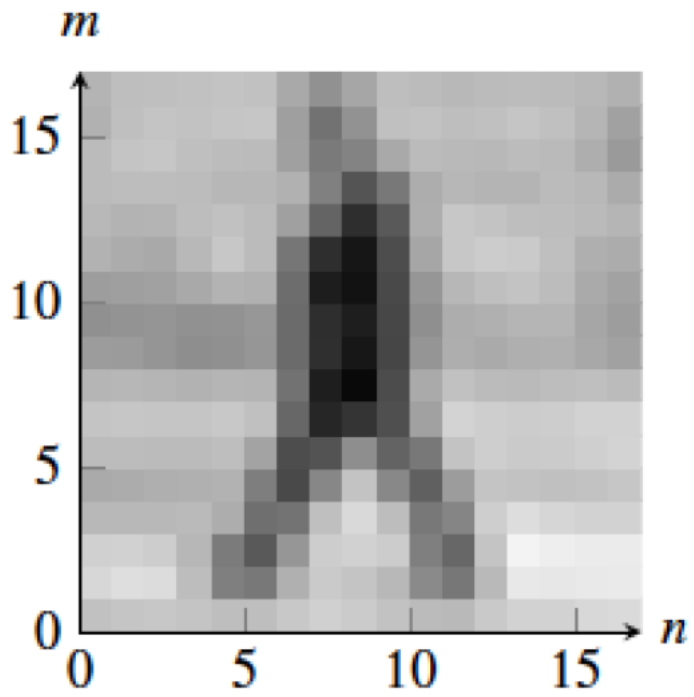
Outline for Today

Part 1: Principles for (Inverse) Image Formation
- Camera models (intrinsics and extrinsics)
- Using camera models as a natural supervision

Part 2: Principles for Image Recognition
- From image classification to object recognition
- Feature pyramids, iterative refinement, and relational computation

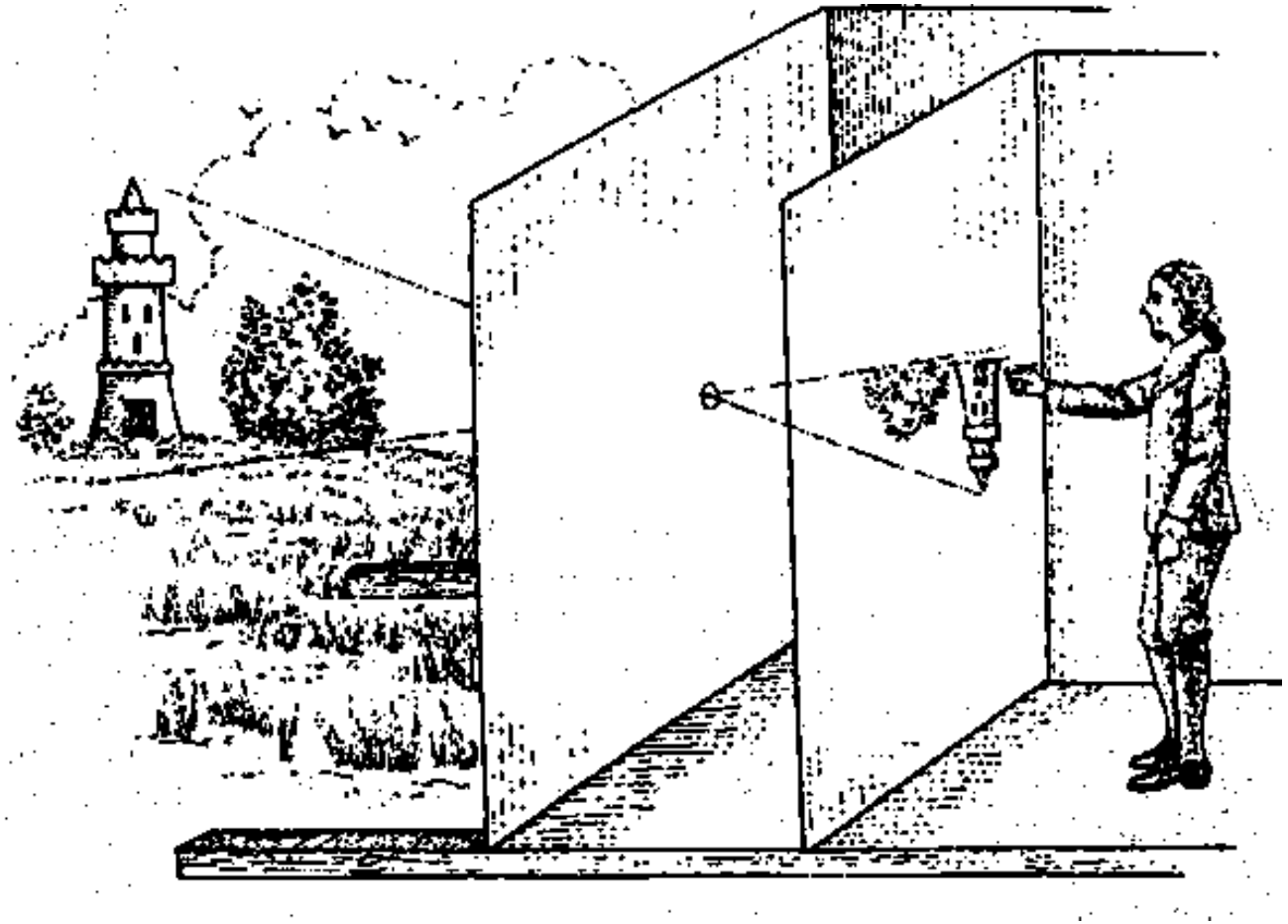
Part 3: Principles for Image Feature Extraction
- Relational features and contextual features

Camera Is a Measurement Device



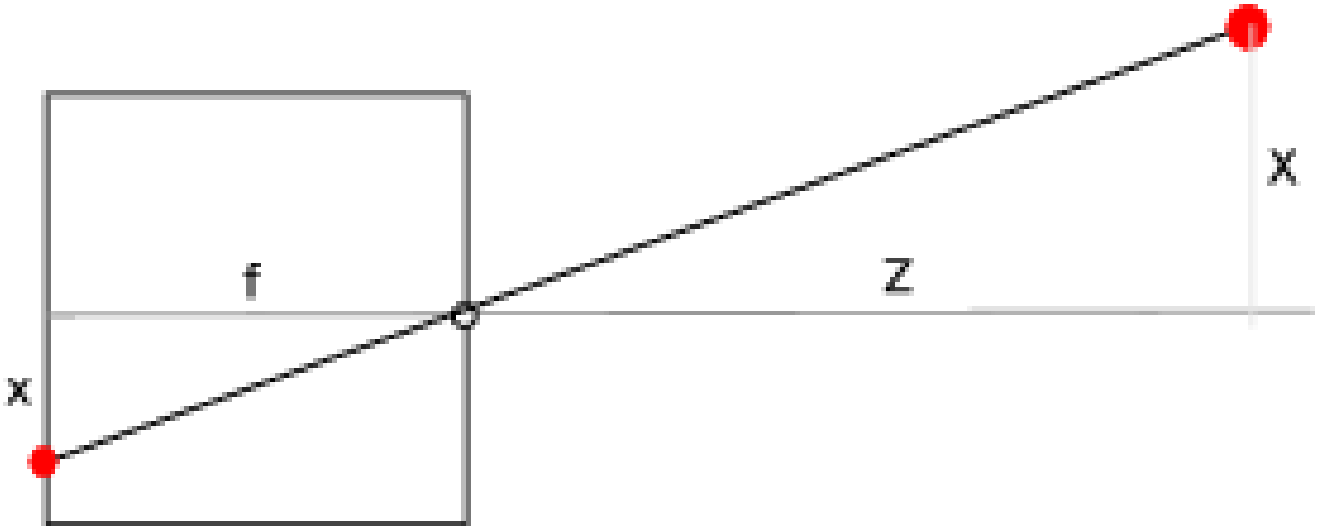
$$\mathbf{I} = \begin{bmatrix}
160 & 175 & 171 & 168 & 168 & 172 & 164 & 158 & 167 & 173 & 167 & 163 & 162 & 164 & 160 & 159 & 163 & 162 \\
149 & 164 & 172 & 175 & 178 & 179 & 176 & 118 & 97 & 168 & 175 & 171 & 169 & 175 & 176 & 177 & 165 & 152 \\
161 & 166 & 182 & 171 & 170 & 177 & 175 & 116 & 109 & 169 & 177 & 173 & 168 & 175 & 175 & 159 & 153 & 123 \\
171 & 174 & 177 & 175 & 167 & 161 & 157 & 138 & 103 & 112 & 157 & 164 & 159 & 160 & 165 & 169 & 148 & 144 \\
163 & 163 & 162 & 165 & 167 & 164 & 178 & 167 & 77 & 55 & 134 & 170 & 167 & 162 & 164 & 175 & 168 & 160 \\
173 & 164 & 158 & 165 & 180 & 180 & 150 & 89 & 61 & 34 & 137 & 186 & 186 & 182 & 175 & 165 & 160 & 164 \\
152 & 155 & 146 & 147 & 169 & 180 & 163 & 51 & 24 & 32 & 119 & 163 & 175 & 182 & 181 & 162 & 148 & 153 \\
134 & 135 & 147 & 149 & 150 & 147 & 148 & 62 & 36 & 46 & 114 & 157 & 163 & 167 & 169 & 163 & 146 & 147 \\
135 & 132 & 131 & 125 & 115 & 129 & 132 & 74 & 54 & 41 & 104 & 156 & 152 & 156 & 164 & 156 & 141 & 144 \\
151 & 155 & 151 & 145 & 144 & 149 & 143 & 71 & 31 & 29 & 129 & 164 & 157 & 155 & 159 & 158 & 156 & 148 \\
172 & 174 & 178 & 177 & 177 & 181 & 174 & 54 & 21 & 29 & 136 & 190 & 180 & 179 & 176 & 184 & 187 & 182 \\
177 & 178 & 176 & 173 & 174 & 180 & 150 & 27 & 101 & 94 & 74 & 189 & 188 & 186 & 183 & 186 & 188 & 187 \\
160 & 160 & 163 & 163 & 161 & 167 & 100 & 45 & 169 & 166 & 59 & 136 & 184 & 176 & 175 & 177 & 185 & 186 \\
147 & 150 & 153 & 155 & 160 & 155 & 56 & 111 & 182 & 180 & 104 & 84 & 168 & 172 & 171 & 164 & 168 & 167 \\
184 & 182 & 178 & 175 & 179 & 133 & 86 & 191 & 201 & 204 & 191 & 79 & 172 & 220 & 217 & 205 & 209 & 200 \\
184 & 187 & 192 & 182 & 124 & 32 & 109 & 168 & 171 & 167 & 163 & 51 & 105 & 203 & 209 & 203 & 210 & 205 \\
191 & 198 & 203 & 197 & 175 & 149 & 169 & 189 & 190 & 173 & 160 & 145 & 156 & 202 & 199 & 201 & 205 & 202 \\
153 & 149 & 153 & 155 & 173 & 182 & 179 & 177 & 182 & 177 & 182 & 185 & 179 & 177 & 167 & 176 & 182 & 180
\end{bmatrix}$$

Camera Is a Measurement of Lights: From 3D to 2D



“Pinhole Camera”

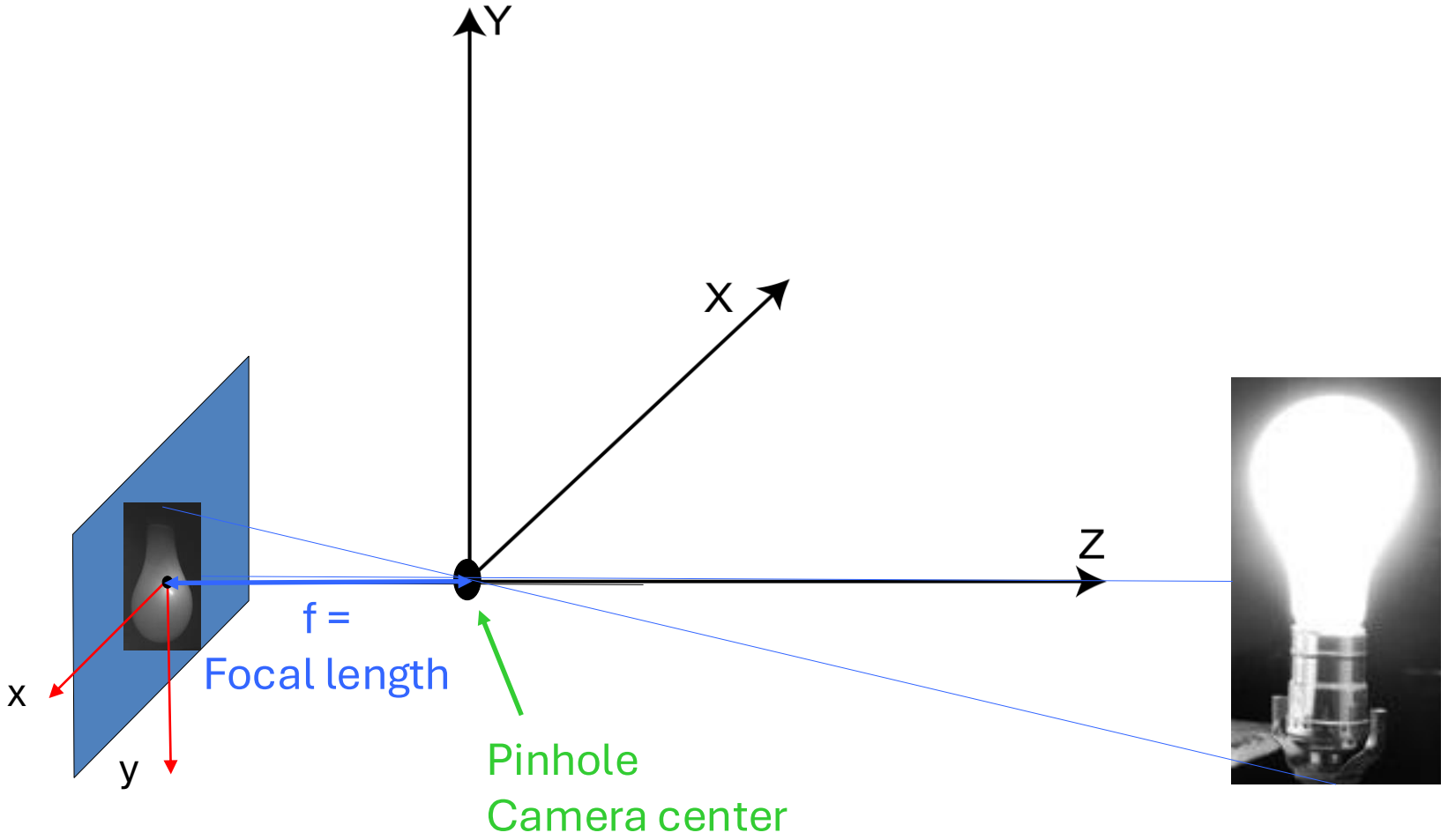
Camera Is a Measurement of Lights: From 3D to 2D



A mathematical model of the pinhole camera: $\frac{x_{real}}{x_{image}} = \frac{z}{f}$

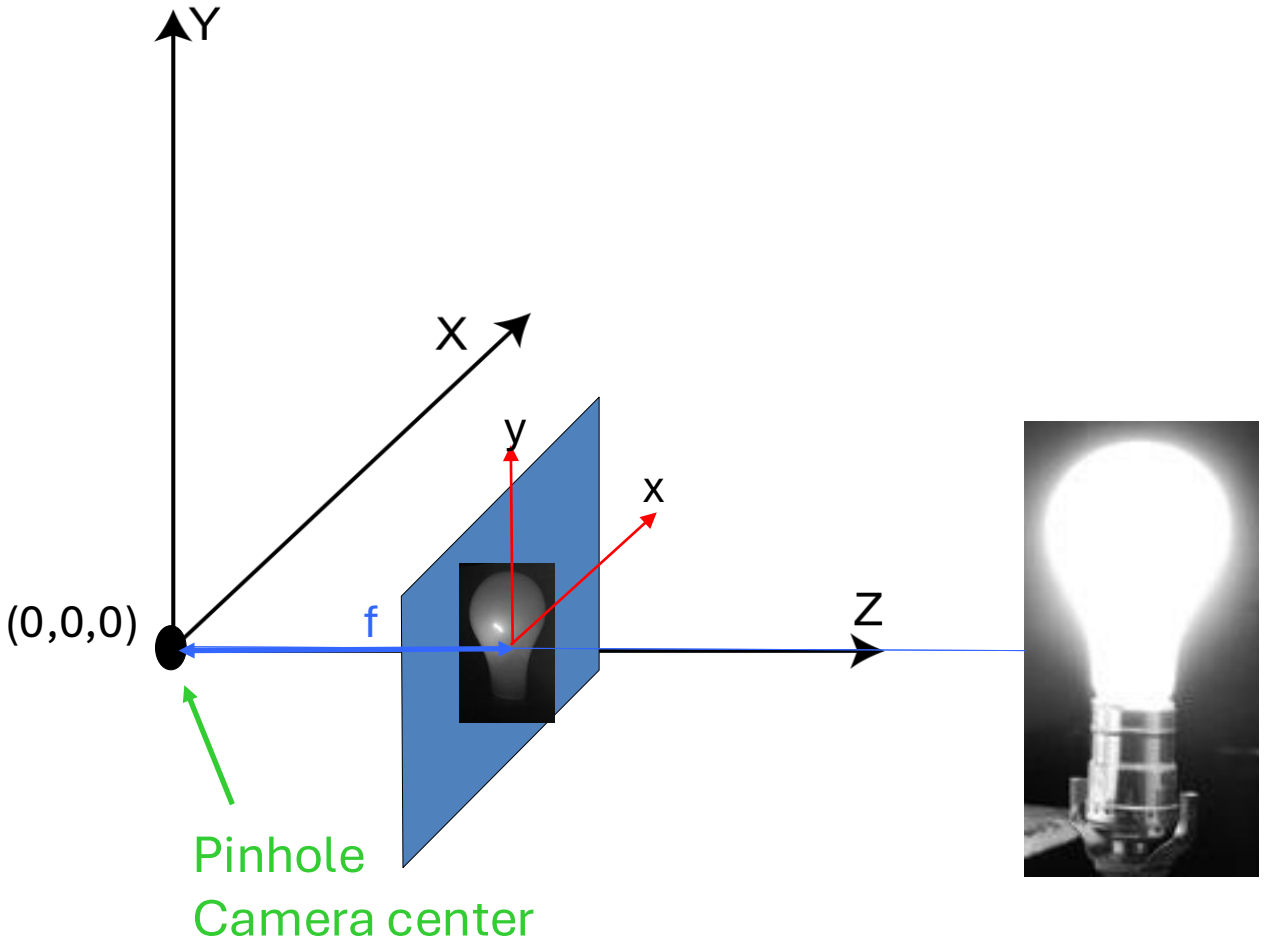
How can we generalize this model to 3D?

Real and Virtual Imaging Planes



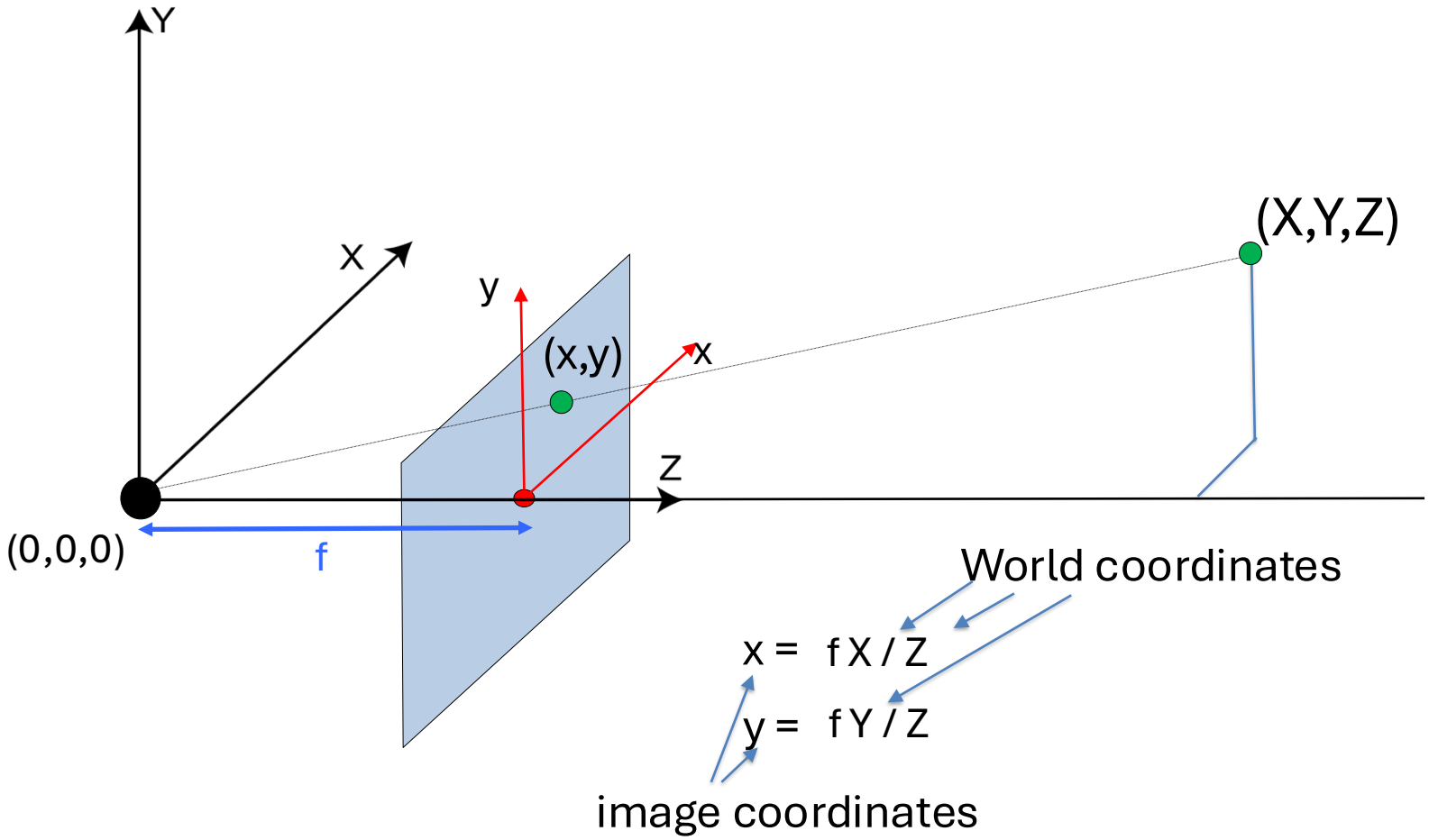
Slides from Bill Freeman 6.S058 https://introtocv.github.io/

Real and Virtual Imaging Planes



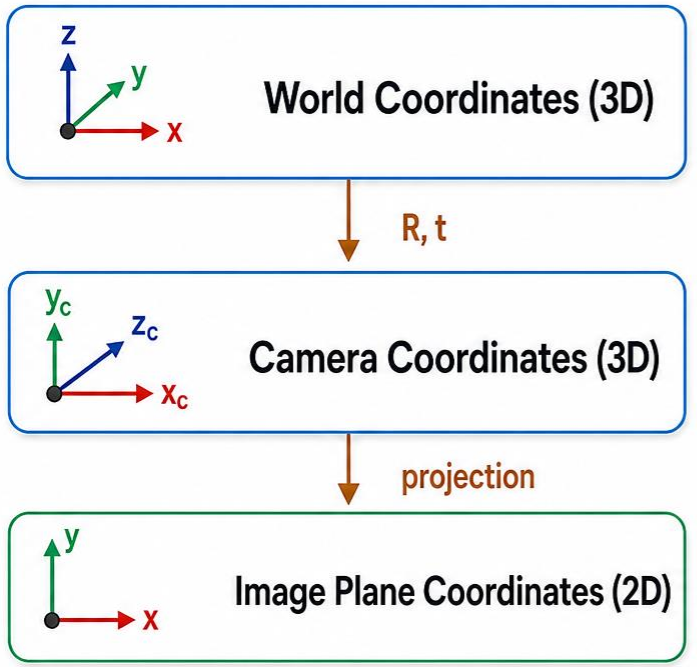
Slides from Bill Freeman 6.S058 https://introtocv.github.io/

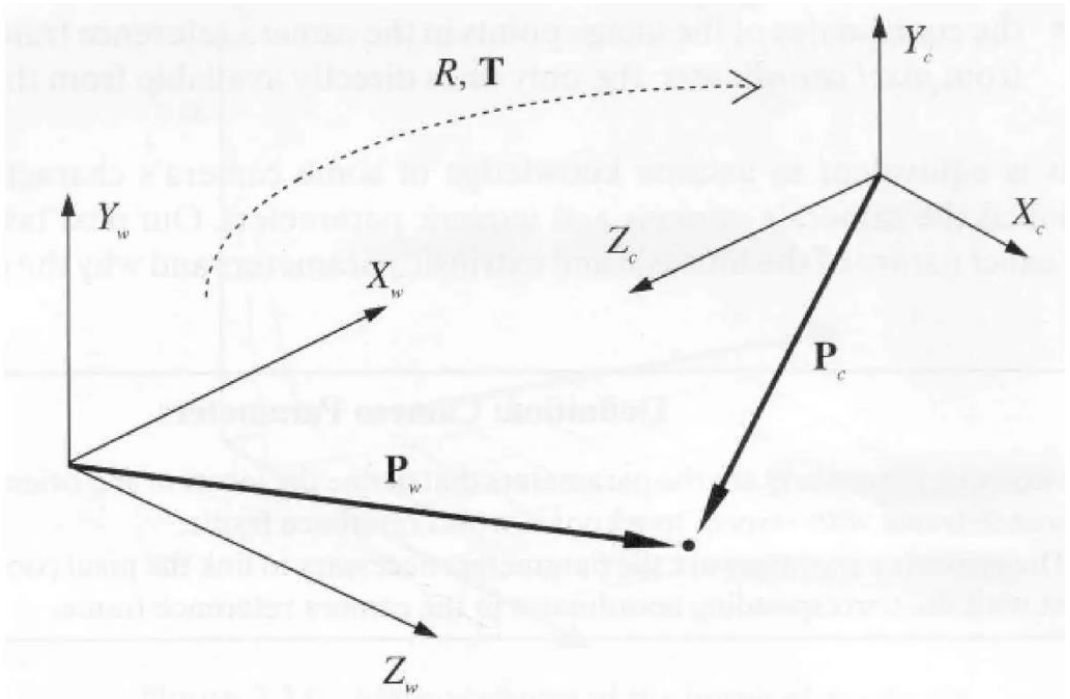
Real and Virtual Imaging Planes



This model assumes the coordinate system is centered at the pinhole

The Full Pinhole Camera Model

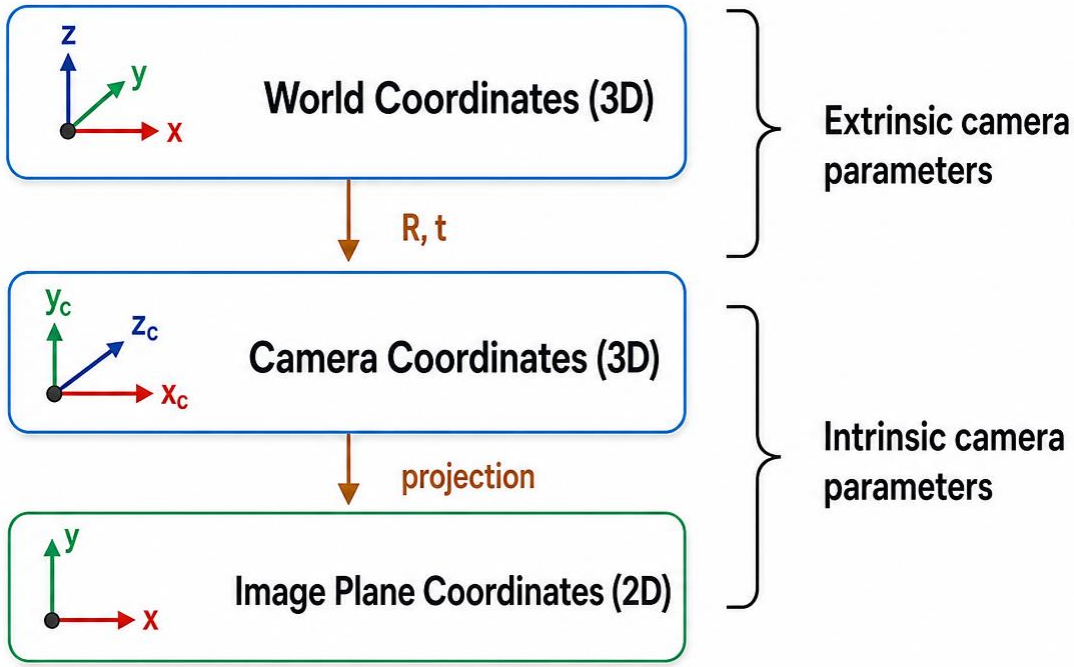


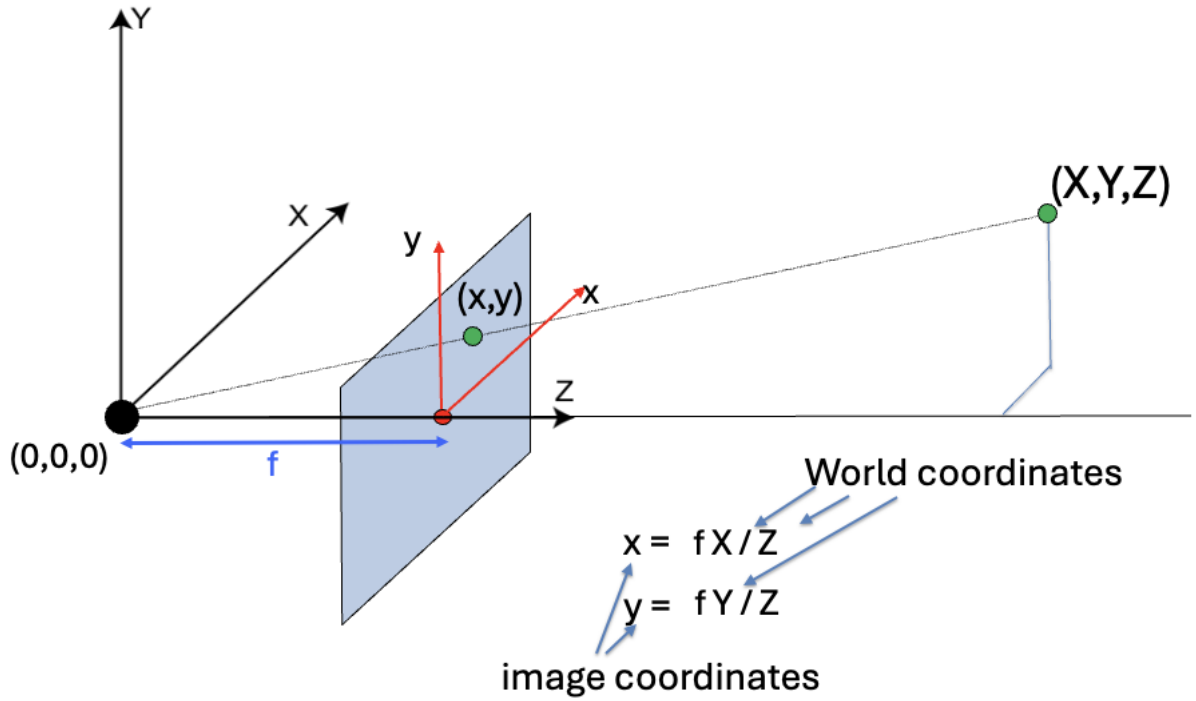


$$P_c = R(P_w - T) \text{ where } R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

https://www.slideshare.net/slideshow/camera-parameters/17046795#5

The Full Pinhole Camera Model





$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \\ 1 \end{bmatrix} \quad \text{where } K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Usually cx=W/2, cy=H/2, fx=fy

Important Notes

- Camera matrix is not “linear” (because of the “divide by z” term)
- Pinhole camera is just one model, and a simplified model
  - e.g., orthographical projection
  - e.g., distortions
- But most 2D/3D methods are based on this model

Using Camera Models as A Supervision Signal

Unsupervised Learning of Depth and Ego-Motion from Video

Tinghui Zhou*
UC Berkeley

Matthew Brown
Google

Noah Snavely
Google

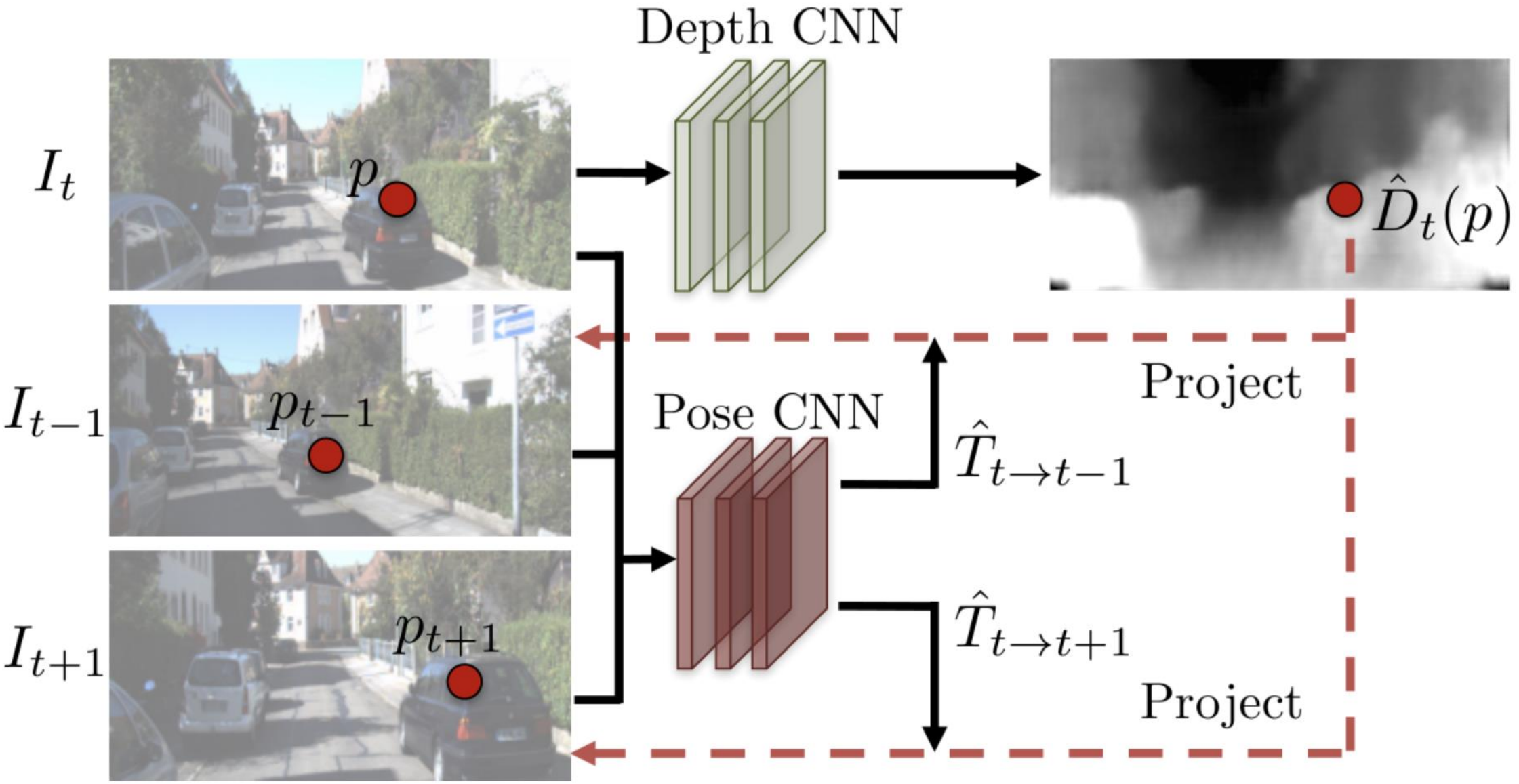
David G. Lowe
Google

Using Camera Models as A Supervision Signal

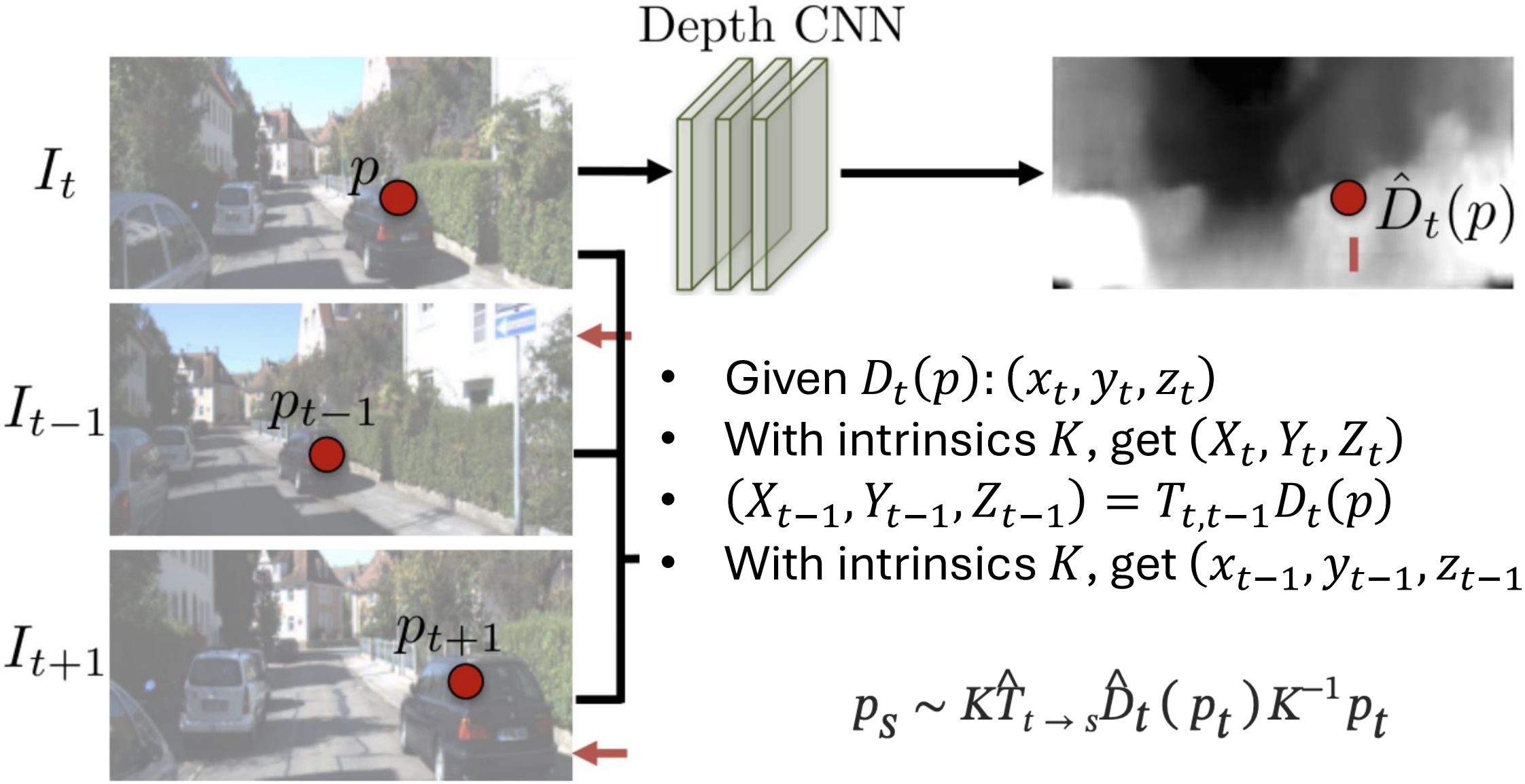
(a) Training: unlabeled video clips.

- Evaluation goal: predicting depths per image and camera poses

Using Camera Models as A Supervision Signal



Using Camera Models as A Supervision Signal



- Given $D_t(p)$: (x_t, y_t, z_t)
- With intrinsics K, get (X_t, Y_t, Z_t)
- $(X_{t-1}, Y_{t-1}, Z_{t-1}) = T_{t,t-1} D_t(p)$
- With intrinsics K, get $(x_{t-1}, y_{t-1}, z_{t-1})$

$$p_s \sim K\hat{T}_{t\to s}\hat{D}_t(p_t)K^{-1}p_t$$

Differentiable Rendering

- Given $D_t(p)$: (x_t, y_t, z_t)
- With intrinsics K, get (X_t, Y_t, Z_t)
- $(X_{t-1}, Y_{t-1}, Z_{t-1}) = T_{t,t-1} D_t(p)$
- With intrinsics K, get $(x_{t-1}, y_{t-1}, z_{t-1})$

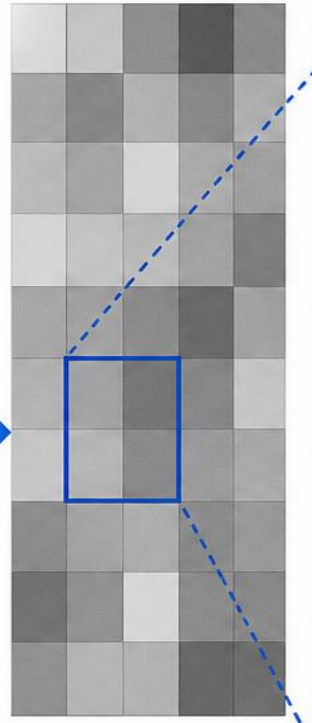
$$p_s \sim K\hat{T}_{t\to s}\hat{D}_t(p_t)K^{-1}p_t$$

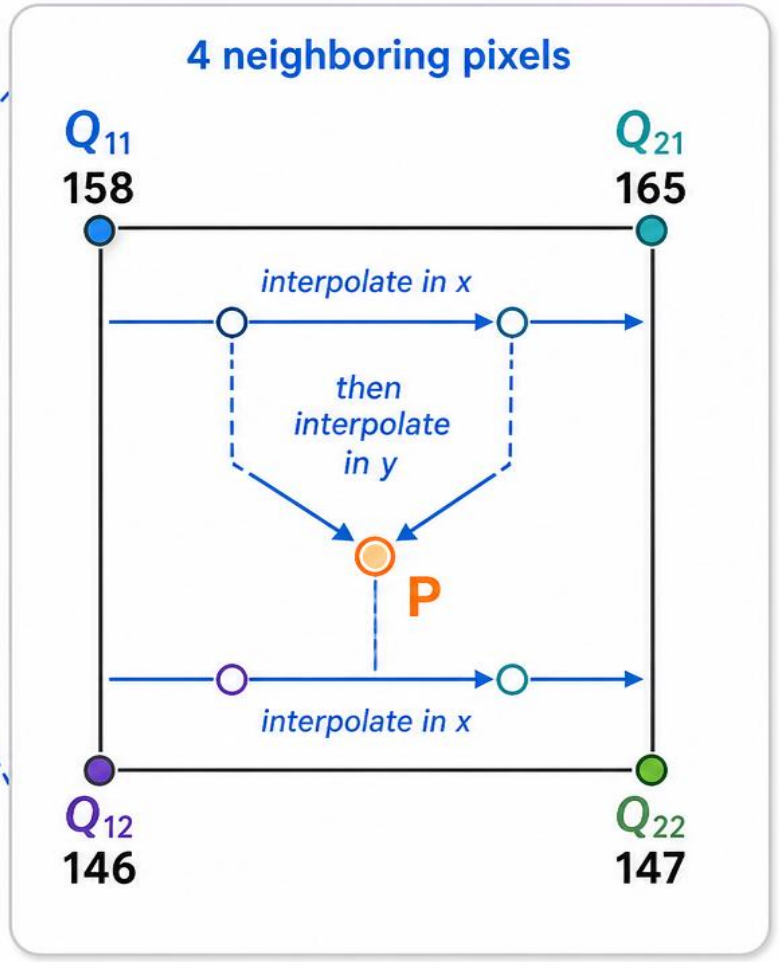
- $\hat{I}(p_s) = I(p_t)$
- How to “get” a value of a real-valued coordinate?

Important Trick: Differentiable Projection

This is a trick very commonly used in computer vision: for pixels and features

160	175	171	168	172
149	164	172	175	178
161	166	182	171	170
171	174	177	175	167
163	163	162	165	167
173	164	158	165	180
152	155	146	147	169
134	135	147	149	150
135	132	131	125	115
151	155	151	145	144





1) Interpolate in x

$$R_1 = (1 - x)\, Q_{11} + x\, Q_{21}$$

$$R_2 = (1 - x)\, Q_{12} + x\, Q_{22}$$

2) Interpolate in y

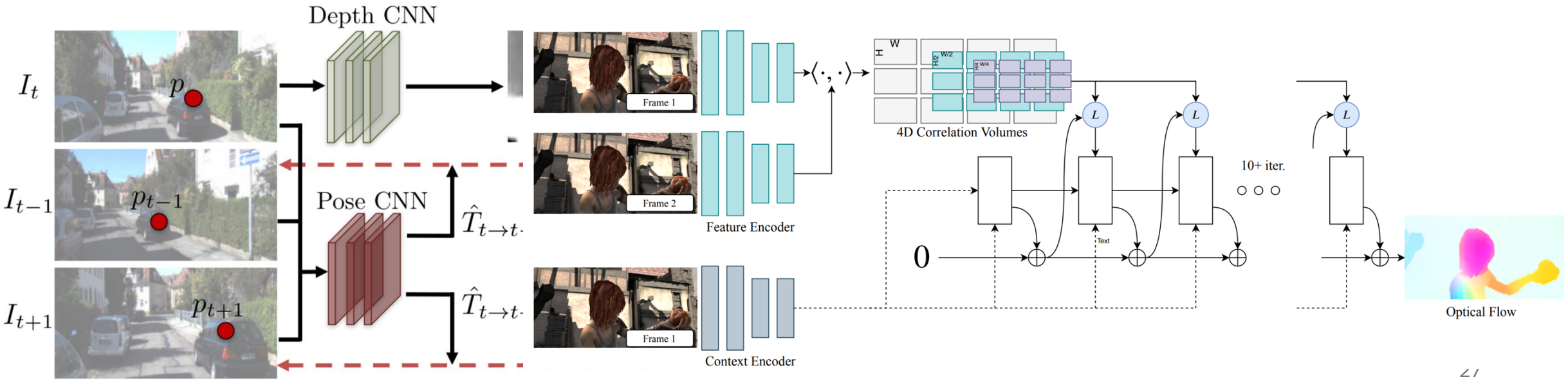
$$P = (1 - y)\, R_1 + y\, R_2$$

Why Can We Do This?

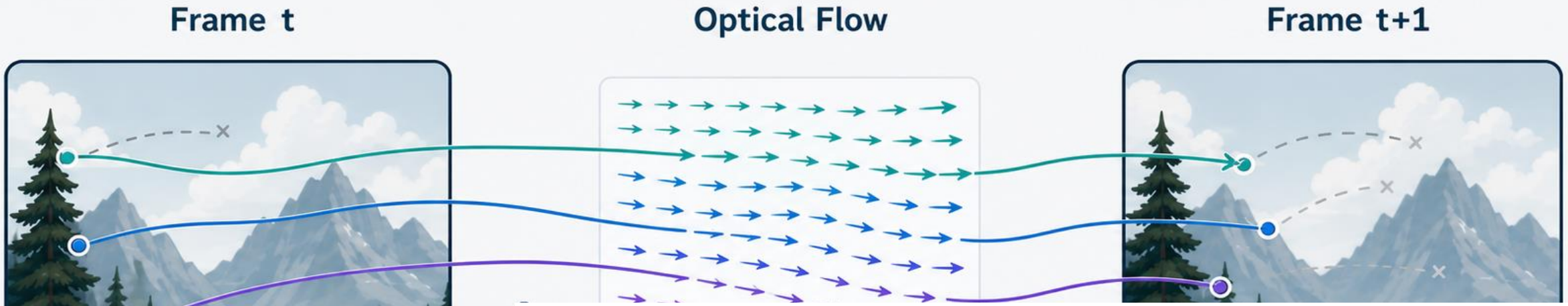
The key assumption: nearby videos have large overlaps in contents

Many computer vision tasks have “synergies”:

- ViewSynthesis(image, depth, new_pose)
- ViewSynthesis(image, optical_flow)
- and more 3D examples in the next lecture



Another Example: $Apply(I, Flow) = I'$



RAFT: Recurrent All-Pairs Field Transforms for Optical Flow

Zachary Teed and Jia Deng

Princeton University
{zteed,jiadeng}@cs.princeton.edu

Principles for (Inverse) Image Formation

Use camera models to reconstruct 3D information (depth, pose) from only 2D inputs
Some times called “Vision as Inverse Graphics” or “Analysis by Synthesis”
Use natural supervisions for such “inverse” tasks

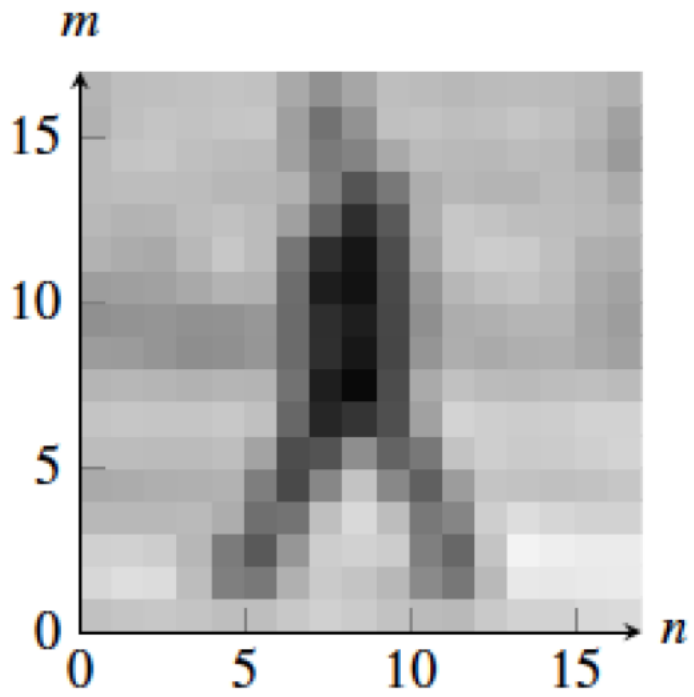
Outline for Today

Part 1: Principles for (Inverse) Image Formation
- Camera models (intrinsics and extrinsics)
- Using camera models as a natural supervision

Part 2: Principles for Image Recognition
- From image classification to object recognition
- Feature pyramids, iterative refinement, and relational computation

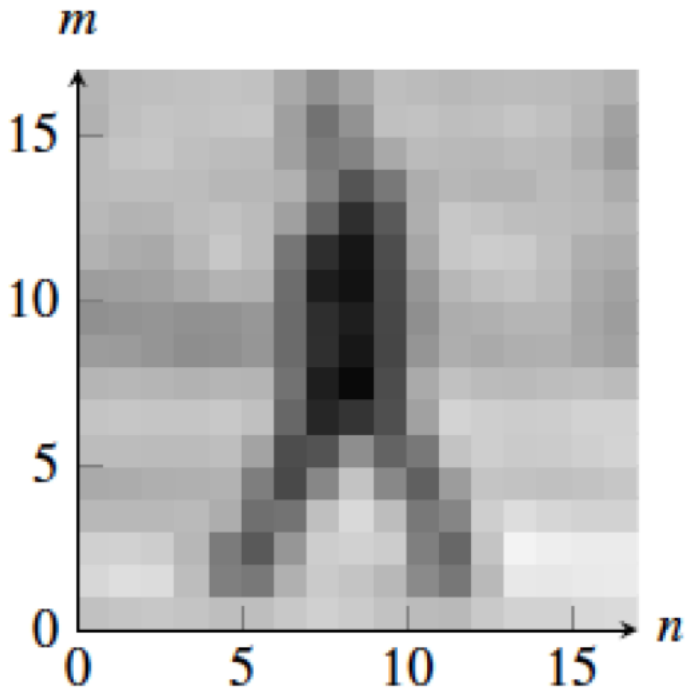
Part 3: Principles for Image Feature Extraction
- Relational features and contextual features

Processing



$$\mathbf{I} = \begin{bmatrix}
160 & 175 & 171 & 168 & 168 & 172 & 164 & 158 & 167 & 173 & 167 & 163 & 162 & 164 & 160 & 159 & 163 & 162 \\
149 & 164 & 172 & 175 & 178 & 179 & 176 & 118 & 97 & 168 & 175 & 171 & 169 & 175 & 176 & 177 & 165 & 152 \\
161 & 166 & 182 & 171 & 170 & 177 & 175 & 116 & 109 & 169 & 177 & 173 & 168 & 175 & 175 & 159 & 153 & 123 \\
171 & 174 & 177 & 175 & 167 & 161 & 157 & 138 & 103 & 112 & 157 & 164 & 159 & 160 & 165 & 169 & 148 & 144 \\
163 & 163 & 162 & 165 & 167 & 164 & 178 & 167 & 77 & 55 & 134 & 170 & 167 & 162 & 164 & 175 & 168 & 160 \\
173 & 164 & 158 & 165 & 180 & 180 & 150 & 89 & 61 & 34 & 137 & 186 & 186 & 182 & 175 & 165 & 160 & 164 \\
152 & 155 & 146 & 147 & 169 & 180 & 163 & 51 & 24 & 32 & 119 & 163 & 175 & 182 & 181 & 162 & 148 & 153 \\
134 & 135 & 147 & 149 & 150 & 147 & 148 & 62 & 36 & 46 & 114 & 157 & 163 & 167 & 169 & 163 & 146 & 147 \\
135 & 132 & 131 & 125 & 115 & 129 & 132 & 74 & 54 & 41 & 104 & 156 & 152 & 156 & 164 & 156 & 141 & 144 \\
151 & 155 & 151 & 145 & 144 & 149 & 143 & 71 & 31 & 29 & 129 & 164 & 157 & 155 & 159 & 158 & 156 & 148 \\
172 & 174 & 178 & 177 & 177 & 181 & 174 & 54 & 21 & 29 & 136 & 190 & 180 & 179 & 176 & 184 & 187 & 182 \\
177 & 178 & 176 & 173 & 174 & 180 & 150 & 27 & 101 & 94 & 74 & 189 & 188 & 186 & 183 & 186 & 188 & 187 \\
160 & 160 & 163 & 163 & 161 & 167 & 100 & 45 & 169 & 166 & 59 & 136 & 184 & 176 & 175 & 177 & 185 & 186 \\
147 & 150 & 153 & 155 & 160 & 155 & 56 & 111 & 182 & 180 & 104 & 84 & 168 & 172 & 171 & 164 & 168 & 167 \\
184 & 182 & 178 & 175 & 179 & 133 & 86 & 191 & 201 & 204 & 191 & 79 & 172 & 220 & 217 & 205 & 209 & 200 \\
184 & 187 & 192 & 182 & 124 & 32 & 109 & 168 & 171 & 167 & 163 & 51 & 105 & 203 & 209 & 203 & 210 & 205 \\
191 & 198 & 203 & 197 & 175 & 149 & 169 & 189 & 190 & 173 & 160 & 145 & 156 & 202 & 199 & 201 & 205 & 202 \\
153 & 149 & 153 & 155 & 173 & 182 & 179 & 177 & 182 & 177 & 182 & 185 & 179 & 177 & 167 & 176 & 182 & 180
\end{bmatrix}$$

Classification: Is this an image of a human?



$\longrightarrow$ $p = 0.8$

Input

Output

Classification: Is this an image of a human?

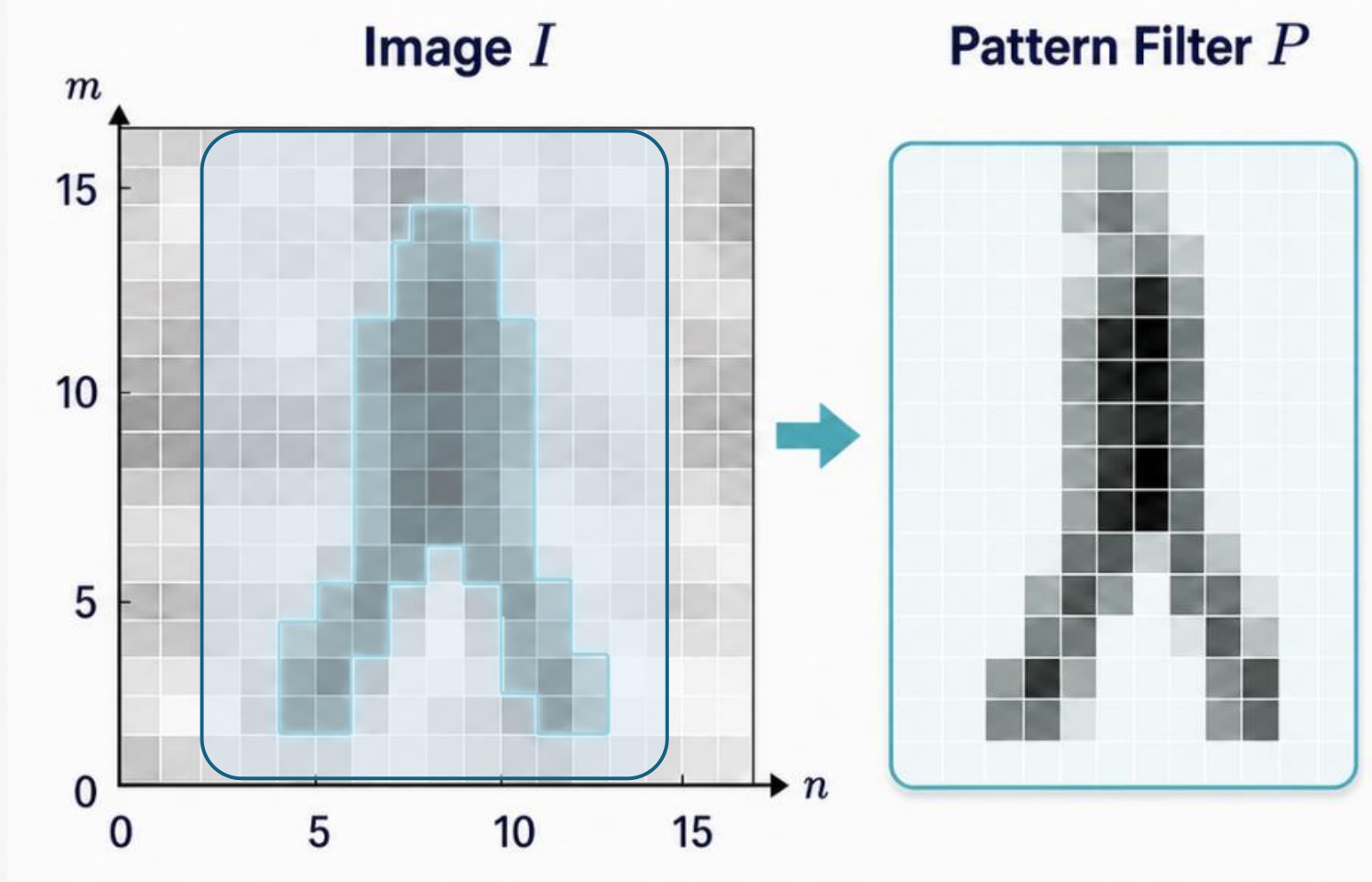
$$\mathbf{I} = \begin{bmatrix}
160 & 175 & 171 & 168 & 168 & 172 & 164 & 158 & 167 & 173 & 167 & 163 & 162 & 164 & 160 & 159 & 163 & 162 \\
149 & 164 & 172 & 175 & 178 & 179 & 176 & 118 & 97 & 168 & 175 & 171 & 169 & 175 & 176 & 177 & 165 & 152 \\
161 & 166 & 182 & 171 & 170 & 177 & 175 & 116 & 109 & 169 & 177 & 173 & 168 & 175 & 175 & 159 & 153 & 123 \\
171 & 174 & 177 & 175 & 167 & 161 & 157 & 138 & 103 & 112 & 157 & 164 & 159 & 160 & 165 & 169 & 148 & 144 \\
163 & 163 & 162 & 165 & 167 & 164 & 178 & 167 & 77 & 55 & 134 & 170 & 167 & 162 & 164 & 175 & 168 & 160 \\
173 & 164 & 158 & 165 & 180 & 180 & 150 & 89 & 61 & 34 & 137 & 186 & 186 & 182 & 175 & 165 & 160 & 164 \\
152 & 155 & 146 & 147 & 169 & 180 & 163 & 51 & 24 & 32 & 119 & 163 & 175 & 182 & 181 & 162 & 148 & 153 \\
134 & 135 & 147 & 149 & 150 & 147 & 148 & 62 & 36 & 46 & 114 & 157 & 163 & 167 & 169 & 163 & 146 & 147 \\
135 & 132 & 131 & 125 & 115 & 129 & 132 & 74 & 54 & 41 & 104 & 156 & 152 & 156 & 164 & 156 & 141 & 144 \\
151 & 155 & 151 & 145 & 144 & 149 & 143 & 71 & 31 & 29 & 129 & 164 & 157 & 155 & 159 & 158 & 156 & 148 \\
172 & 174 & 178 & 177 & 177 & 181 & 174 & 54 & 21 & 29 & 136 & 190 & 180 & 179 & 176 & 184 & 187 & 182 \\
177 & 178 & 176 & 173 & 174 & 180 & 150 & 27 & 101 & 94 & 74 & 189 & 188 & 186 & 183 & 186 & 188 & 187 \\
160 & 160 & 163 & 163 & 161 & 167 & 100 & 45 & 169 & 166 & 59 & 136 & 184 & 176 & 175 & 177 & 185 & 186 \\
147 & 150 & 153 & 155 & 160 & 155 & 56 & 111 & 182 & 180 & 104 & 84 & 168 & 172 & 171 & 164 & 168 & 167 \\
184 & 182 & 178 & 175 & 179 & 133 & 86 & 191 & 201 & 204 & 191 & 79 & 172 & 220 & 217 & 205 & 209 & 200 \\
184 & 187 & 192 & 182 & 124 & 32 & 109 & 168 & 171 & 167 & 163 & 51 & 105 & 203 & 209 & 203 & 210 & 205 \\
191 & 198 & 203 & 197 & 175 & 149 & 169 & 189 & 190 & 173 & 160 & 145 & 156 & 202 & 199 & 201 & 205 & 202 \\
153 & 149 & 153 & 155 & 173 & 182 & 179 & 177 & 182 & 177 & 182 & 185 & 179 & 177 & 167 & 176 & 182 & 180
\end{bmatrix} \longrightarrow p = 0.8$$

Input

Output

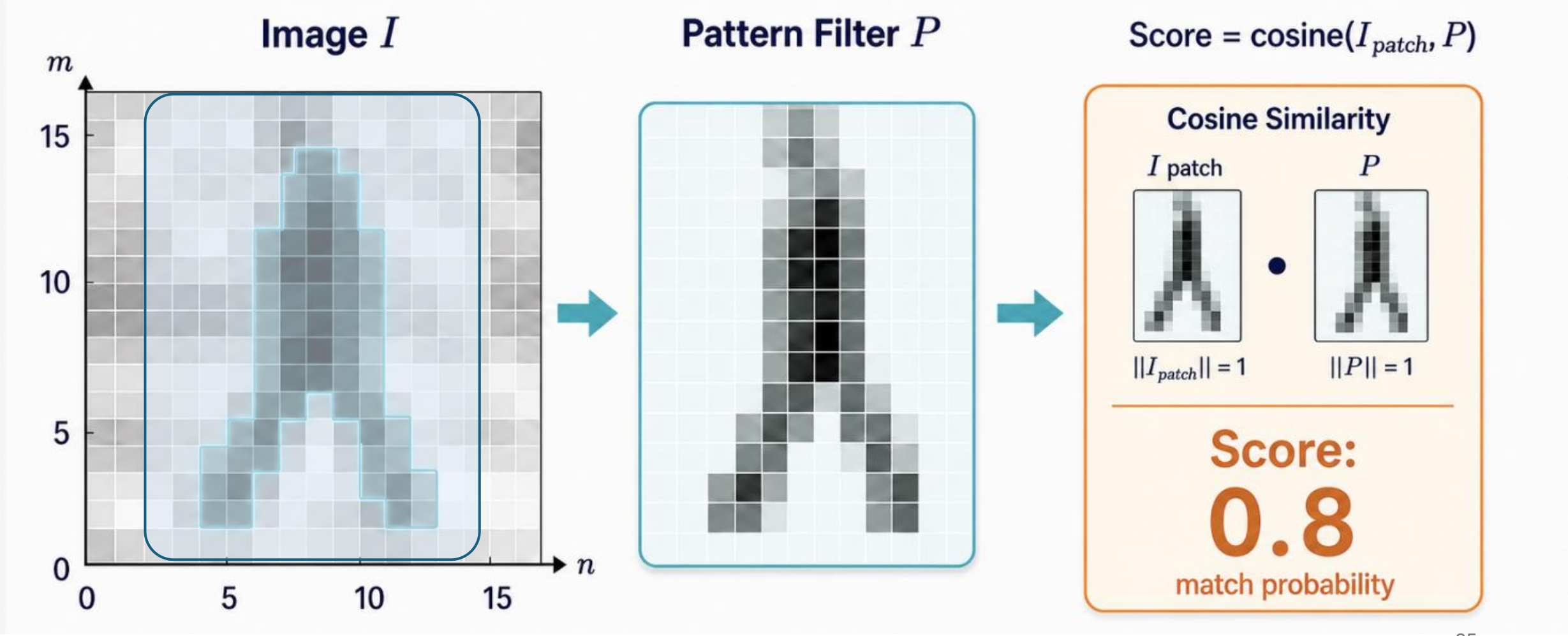
Classification: Is this an image of a human?

Simplest idea: Pattern matching



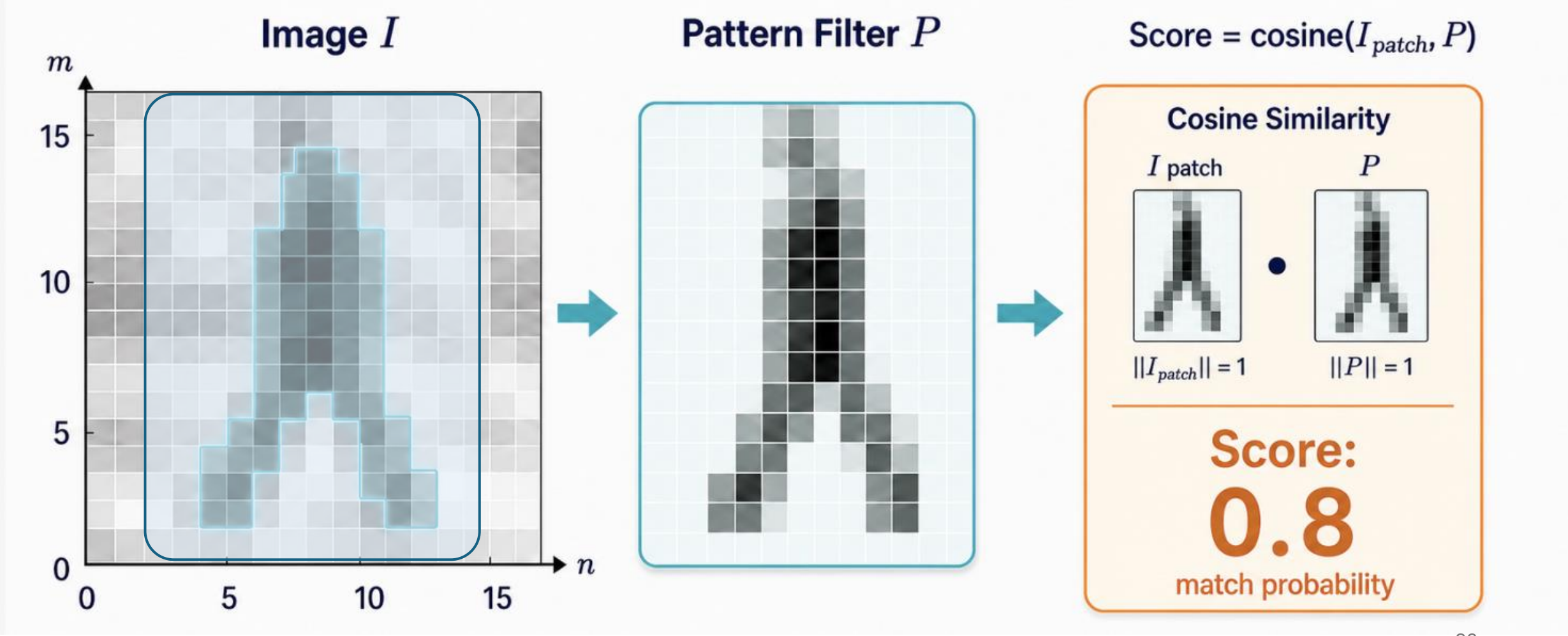
Classification: Is this an image of a human?

Simplest idea: Pattern matching



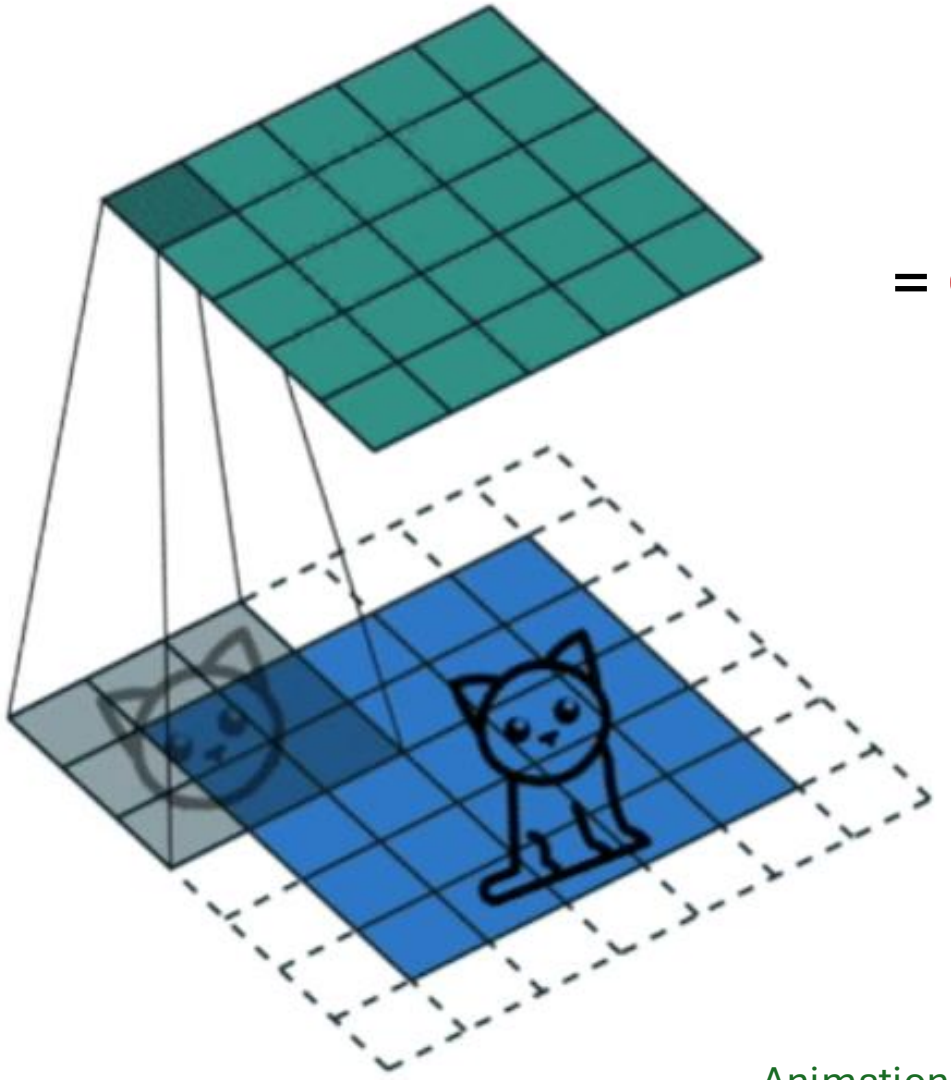
Classification: Is this an image of a human?

Simplest idea: Pattern matching = pixel-wise product



How to Extend This to Object Detection?

Simplest idea: using a sliding window on the image to match patterns



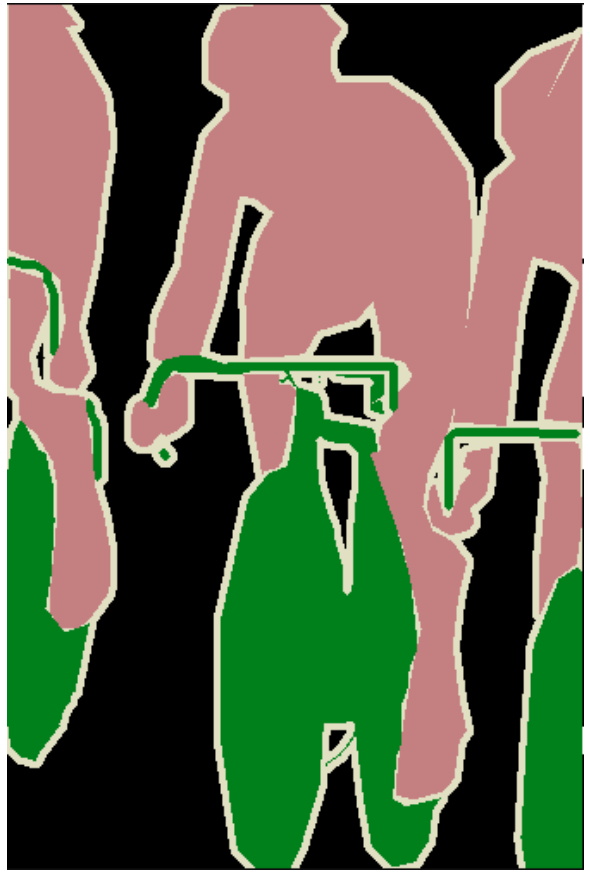
Applying a pattern at each location

= Convolution with a “kernel”

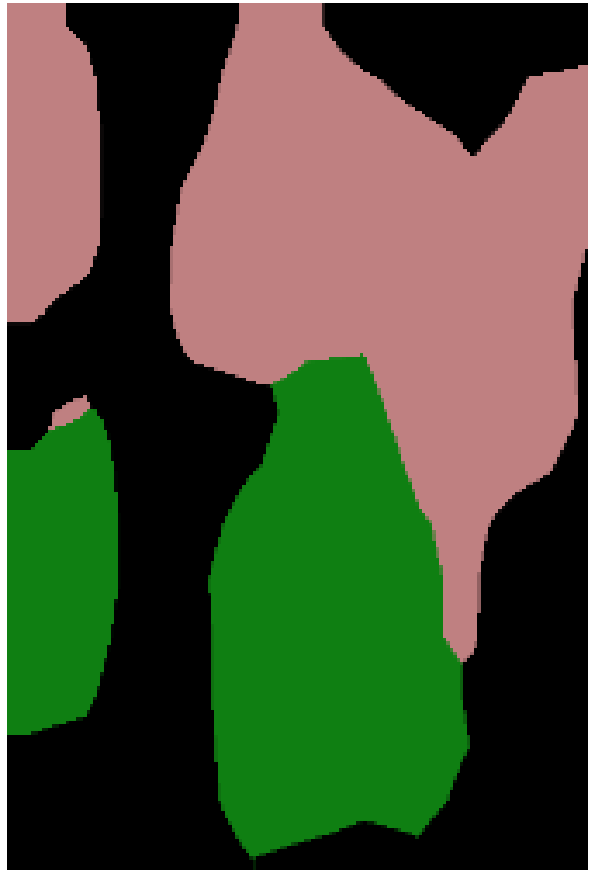
Animation adapted from: https://lena-voita.github.io/nlp_course/models/convolutional.html

Let’s Formulate Some Tasks

Pixel-Level Prediction Task: Semantic Segmentation



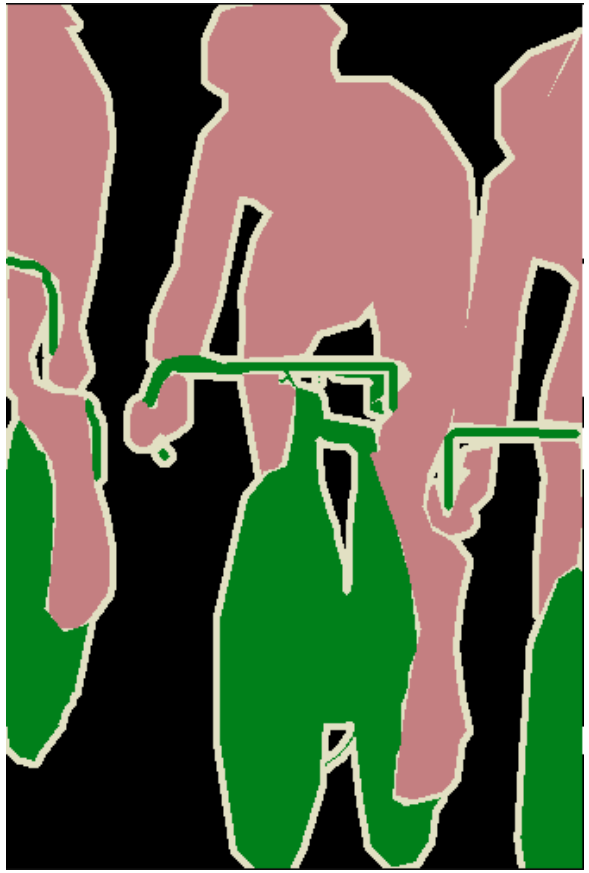
ground truth



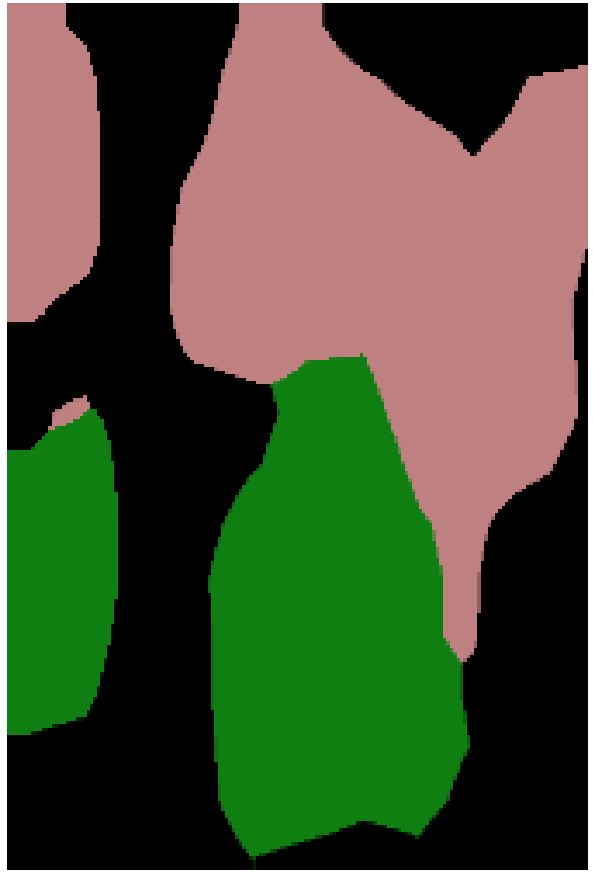
prediction

Slides from Kaiming He 6.S058 https://introtocv.github.io/

Pixel-Level Prediction Task: Semantic Segmentation



ground truth



prediction

$$\text{IoU} = \frac{\text{intersection}}{\text{union}}$$

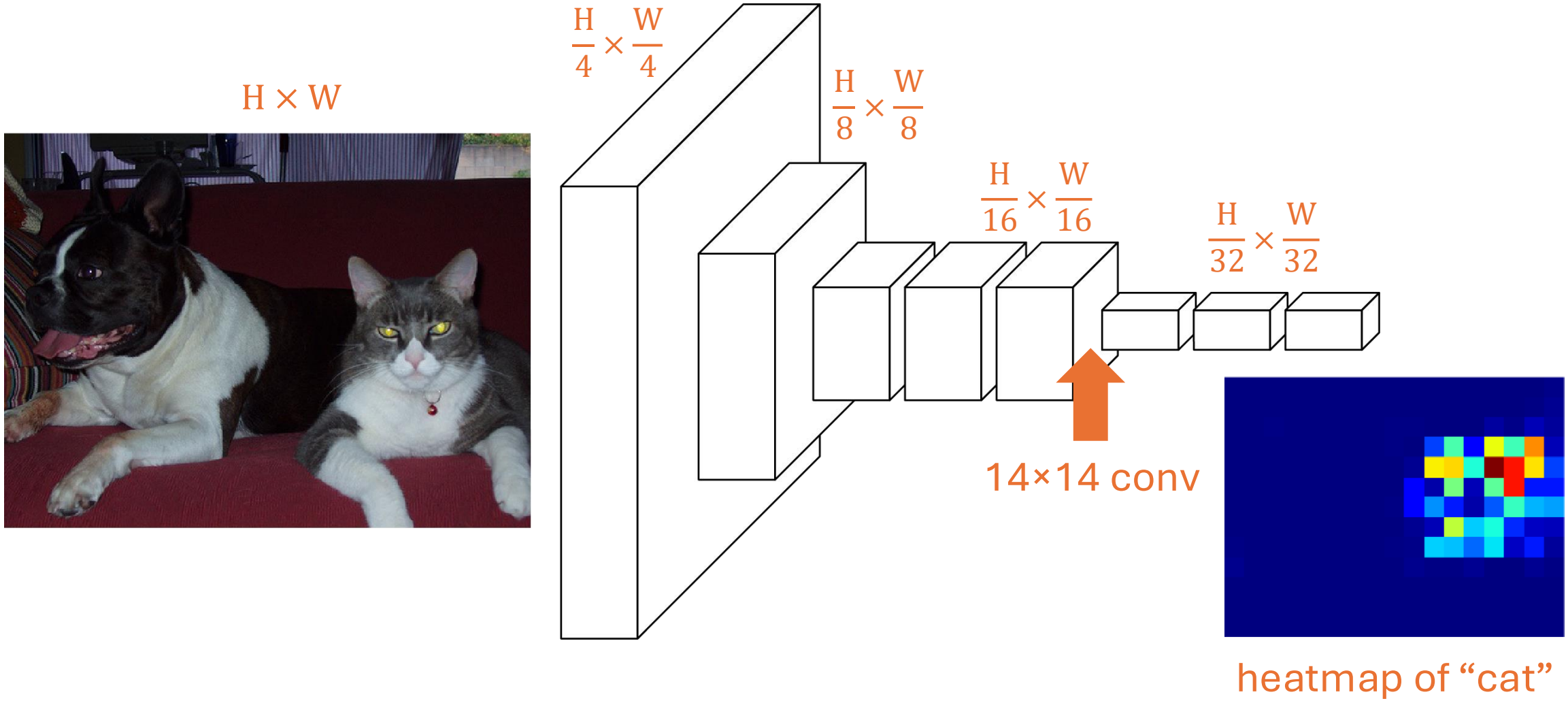
Pixel-Level Prediction Task: Semantic Segmentation

$H \times W$

Slides from Kaiming He 6.S058 https://introtocv.github.io/

Jonathan Long, Evan Shelhamer, Trevor Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

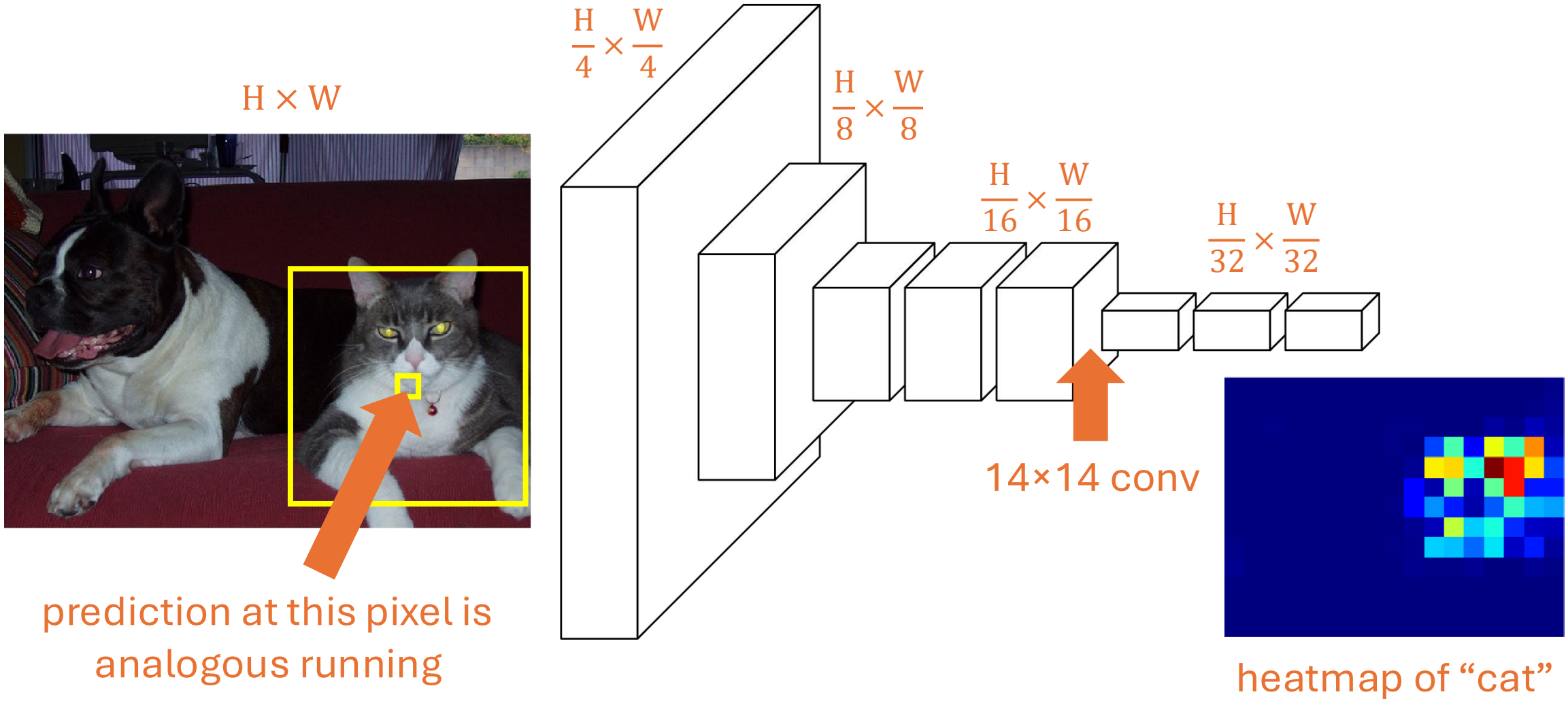
Pixel-Level Prediction Task: Semantic Segmentation



Slides from Kaiming He 6.S058 https://introtocv.github.io/

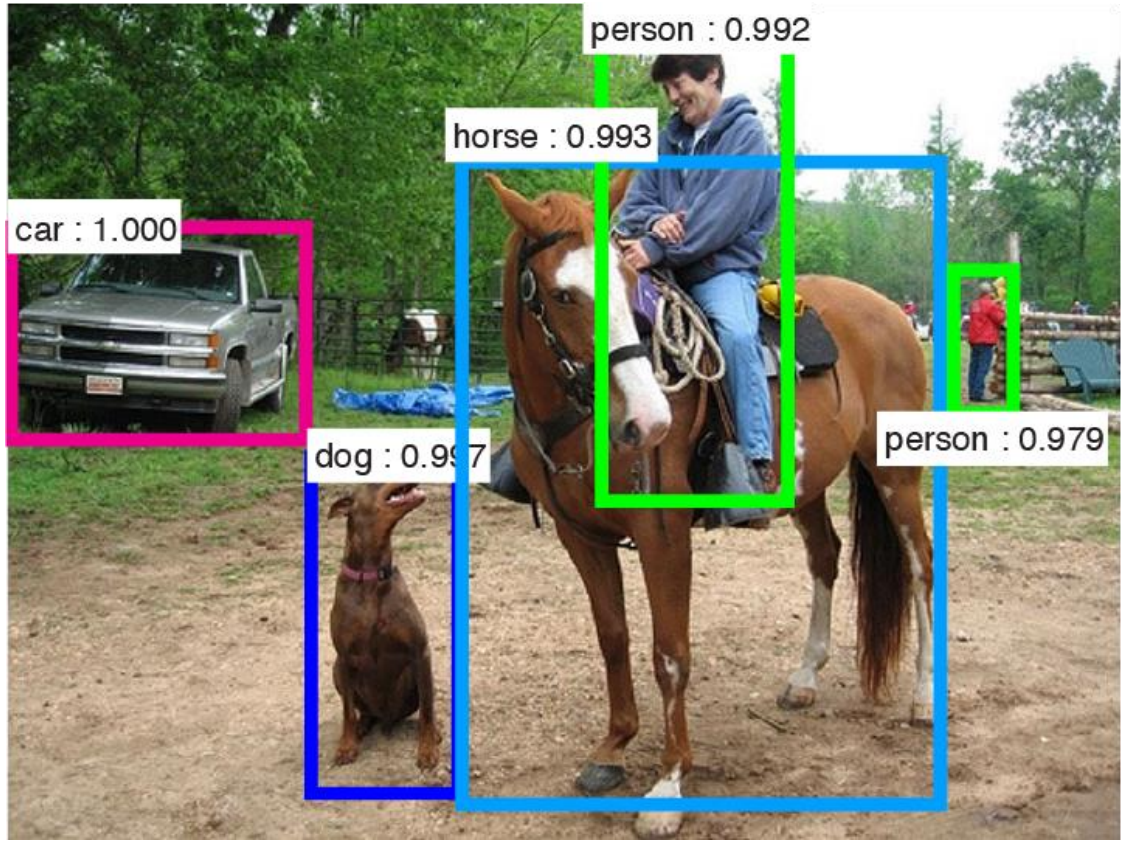
Jonathan Long, Evan Shelhamer, Trevor Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

Pixel-Level Prediction Task: Semantic Segmentation



Jonathan Long, Evan Shelhamer, Trevor Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

The “Full” Object Detection Task



Identify:

Are there objects presented?

Locate:

Where is each object?

(x, y, h, w)

Recognize:

What is each object?

Class label & confidence

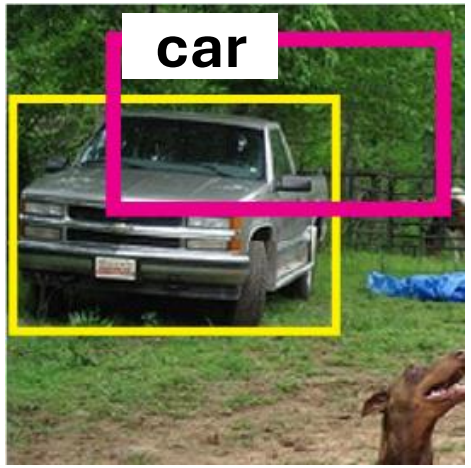
Slides from Kaiming He 6.S058 https://introtocv.github.io/

Jonathan Long, Evan Shelhamer, Trevor Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

Evaluation: Different Types of Errors

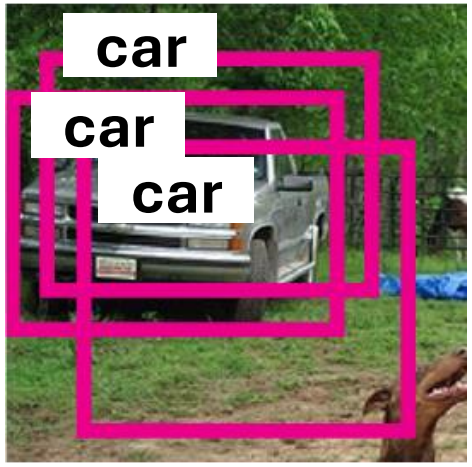
correct
(true positive)

missed
(false negative)



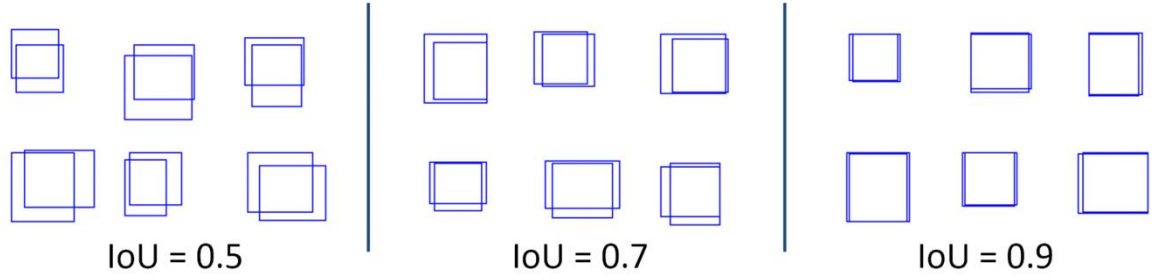
mislocalized
(false positive)

misrecognized
(false positive)



duplicated
(false positive)

define by an IoU threshold



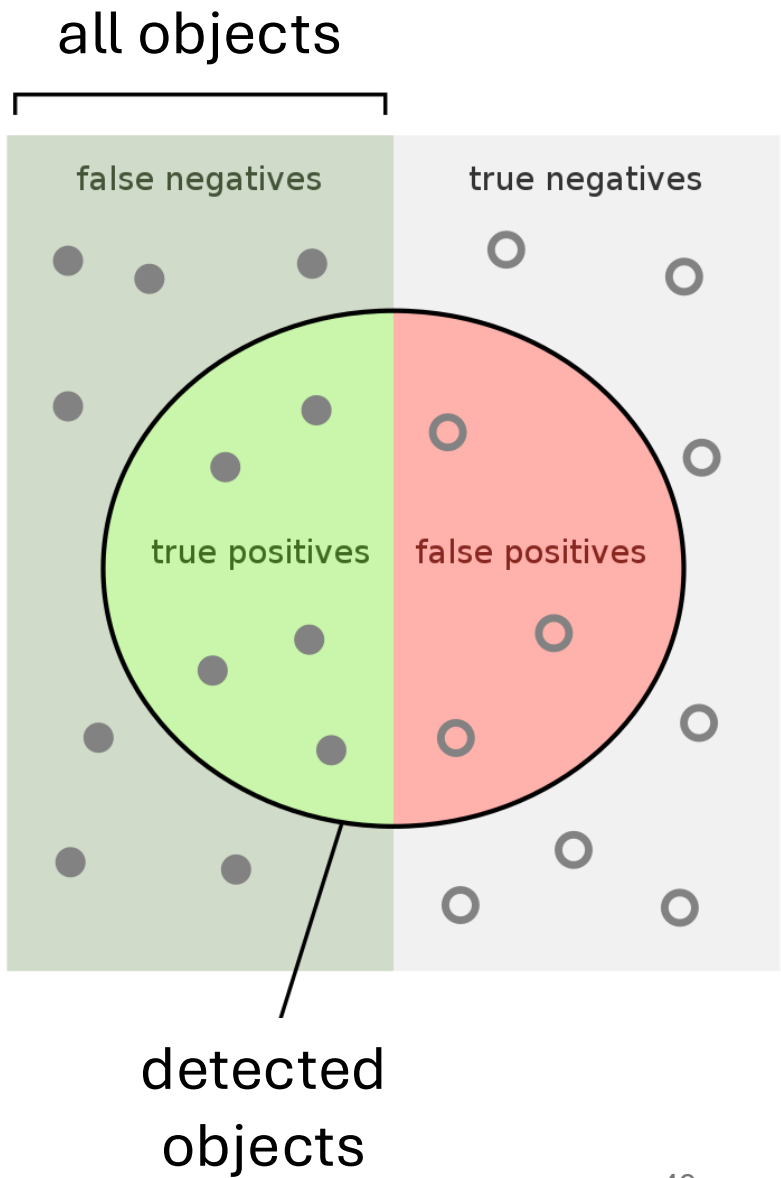
Evaluation: Different Types of Errors

- **Precision**: What % of detected objects are correct?

$$\textbf{Precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

- **Recall**: What % of objects are detected?

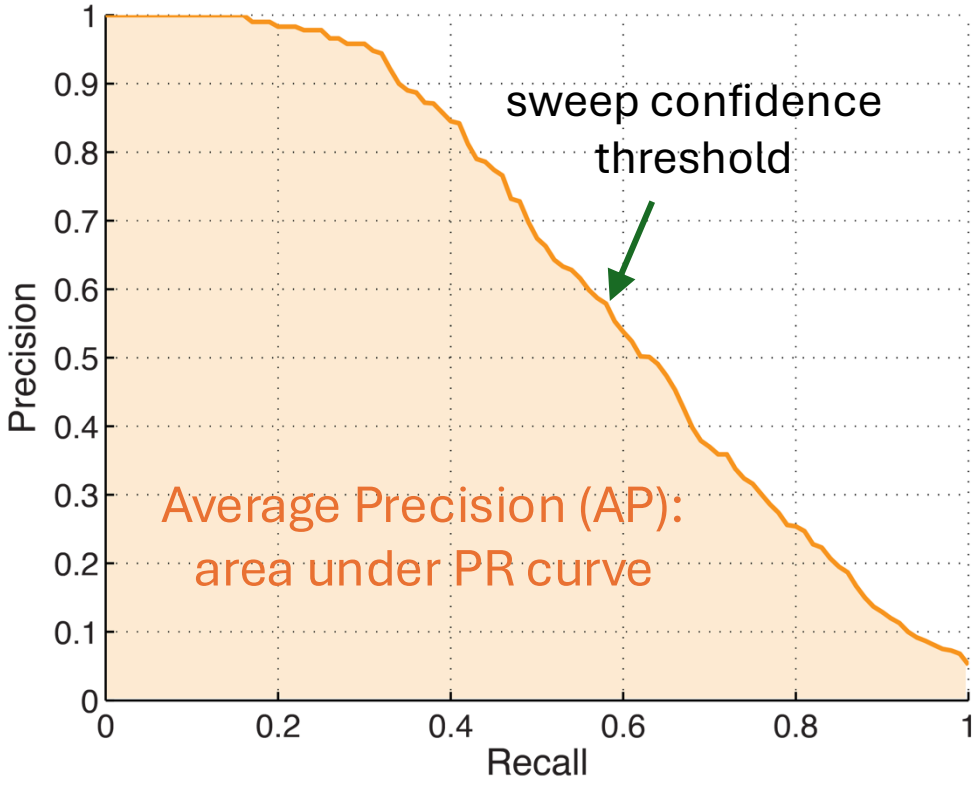
$$\textbf{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$



Slides from Kaiming He 6.S058 https://introtocv.github.io/

Evaluation: Different Types of Errors

- Precision-Recall Tradeoff



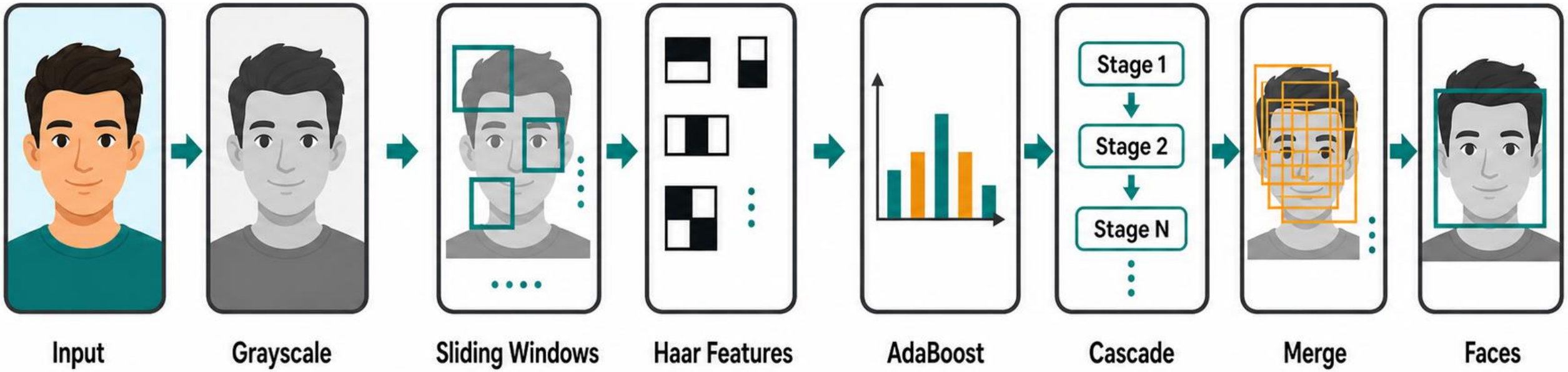
threshold = **0.4**: Recall ↑, Precision ↓

threshold = **0.7**: Recall ↓, Precision ↑

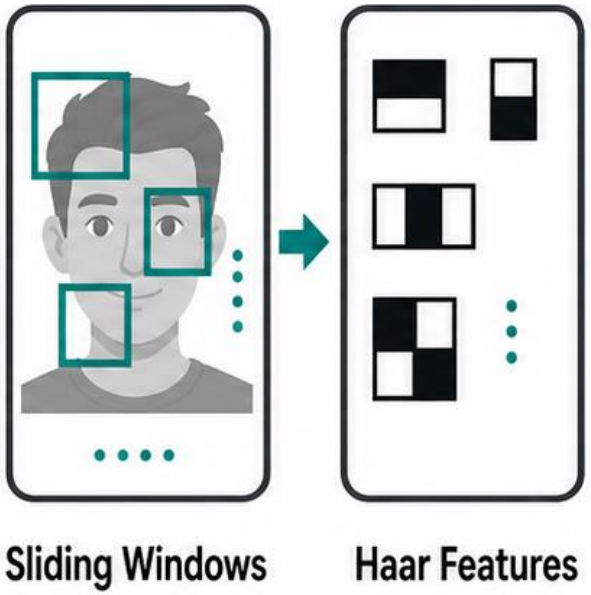
Slides from Kaiming He 6.S058 https://introtocv.github.io/

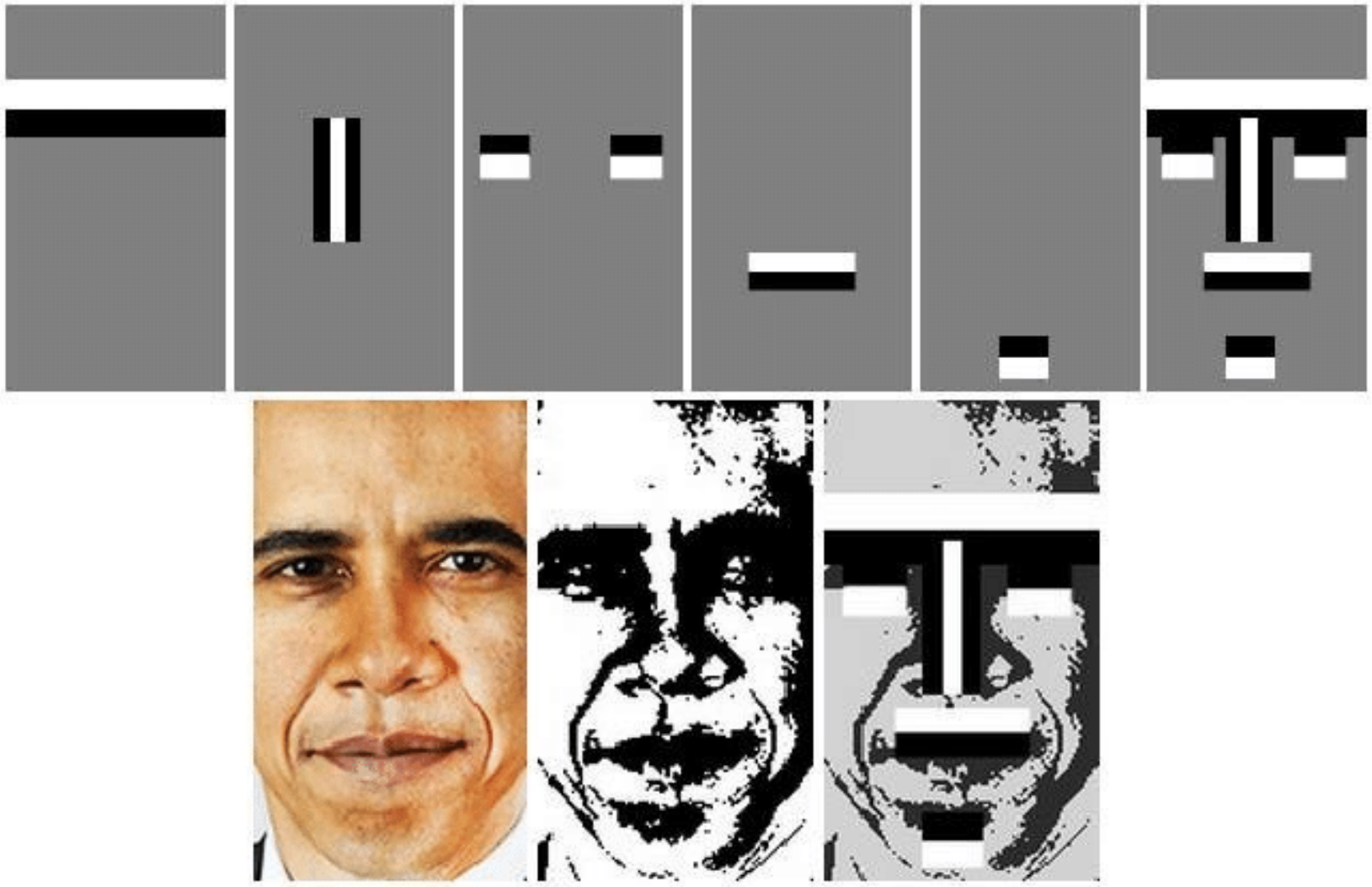
How to Solve This? Classical Model: V-J Face Detection

Classical Model: Viola-Jones Face Detection



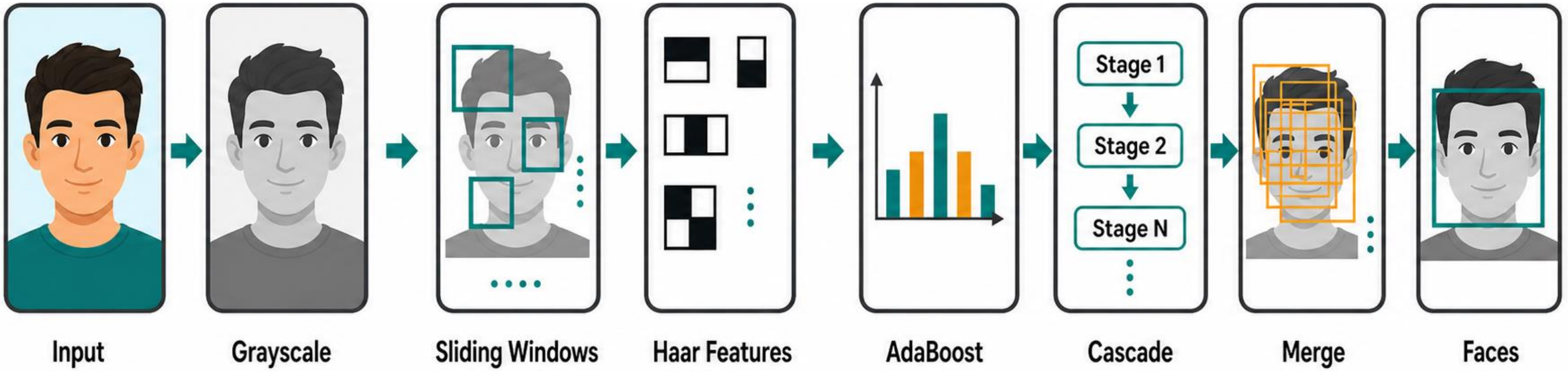
Classical Model: Viola-Jones Face Detection





https://www.semanticscholar.org/paper/A-comparative-study-between-LBP-and-Haar-like-for-Kadir-Kamaruddin/3ec98dabbcf440db746d4baf6550e44e12c5ee76

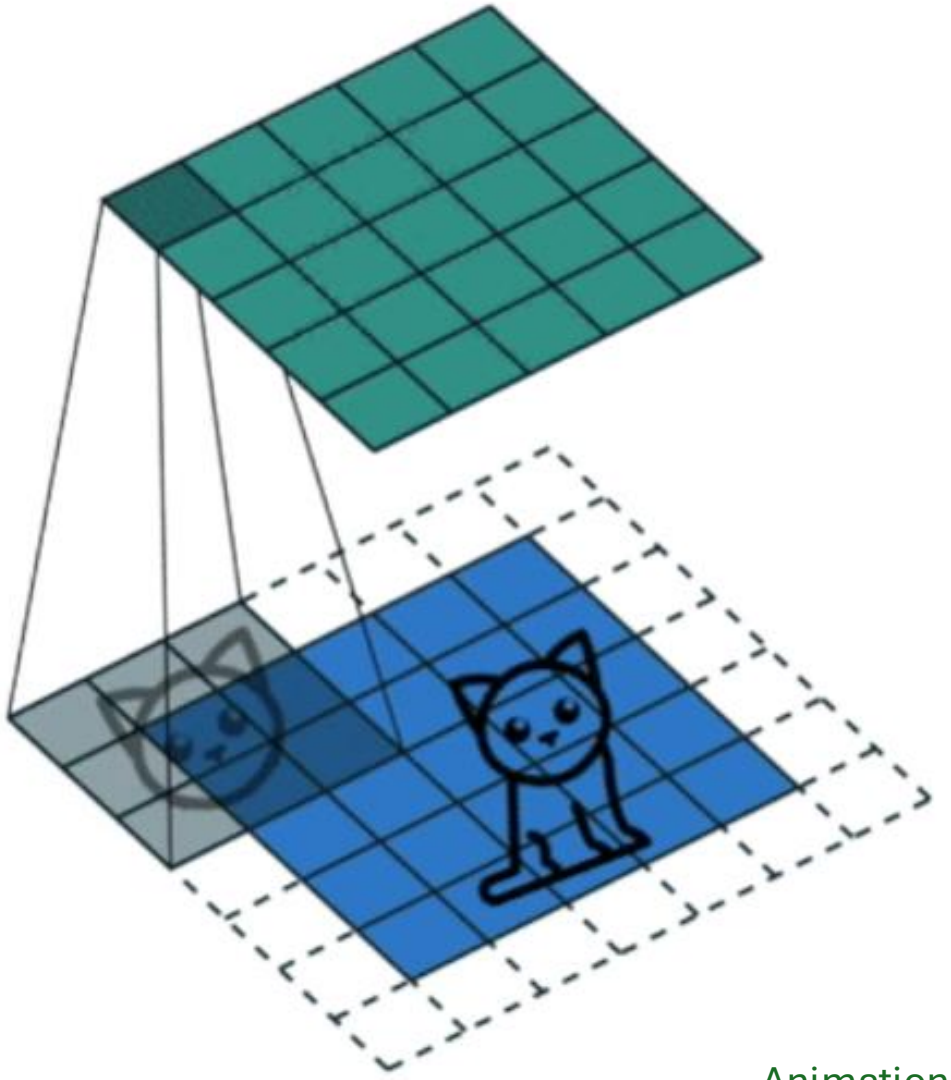
Classical Model: Viola-Jones Face Detection



Key ingredients: sliding window, feature extraction, multi-stage prediction, and bounding box merging

Modern Frameworks for Object Detection

Simplest idea: using a sliding window on the image to match patterns



Animation adapted from: https://lena-voita.github.io/nlp_course/models/convolutional.html

Two Ideas: Pixel-Based and Region-Based Prediction

Two Ideas: Pixel-Based and Region-Based Prediction

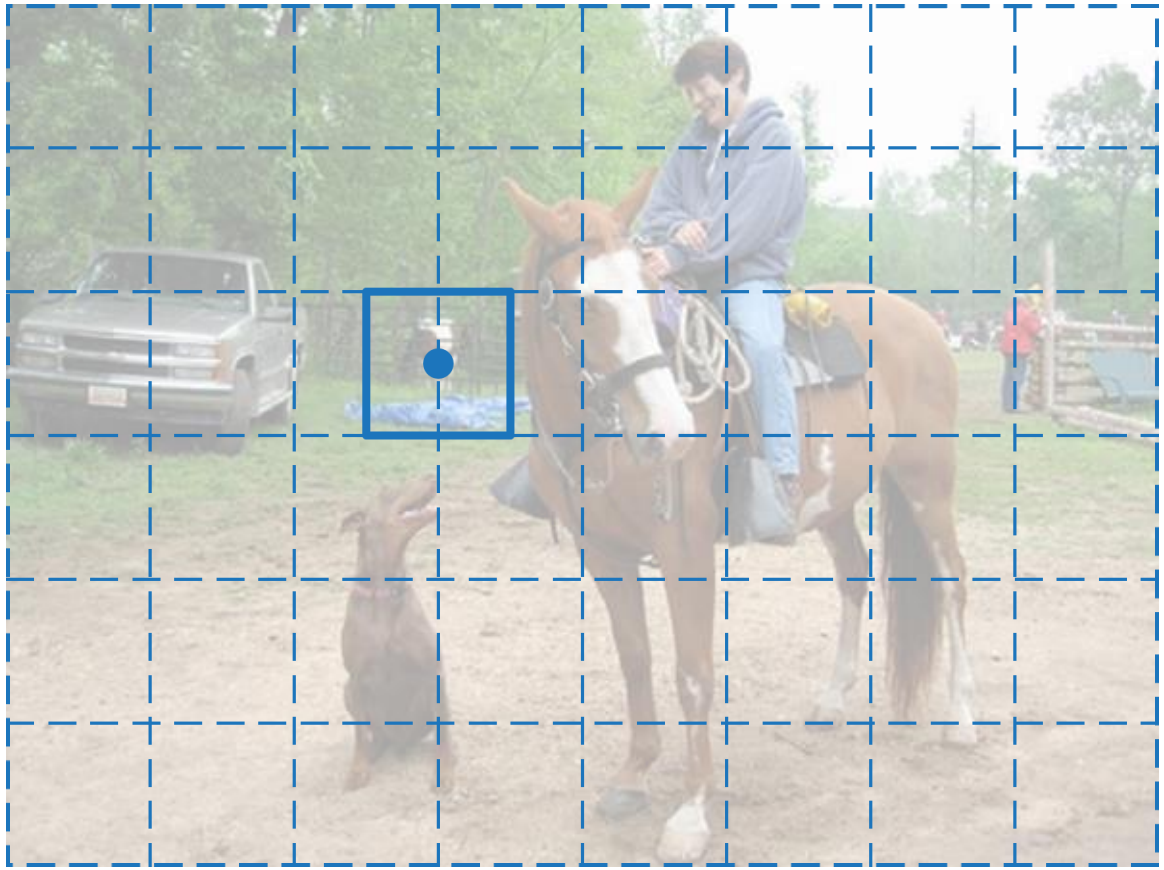
Two Ideas: Pixel-Based and Region-Based Prediction

- **ConvNets are Fully Convolutional!**
- Search objects w/ sliding windows (a.k.a., convolutions)

Slides from Kaiming He 6.S058 https://introtocv.github.io/

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NeurIPS 2015

Two Ideas: Pixel-Based and Region-Based Prediction

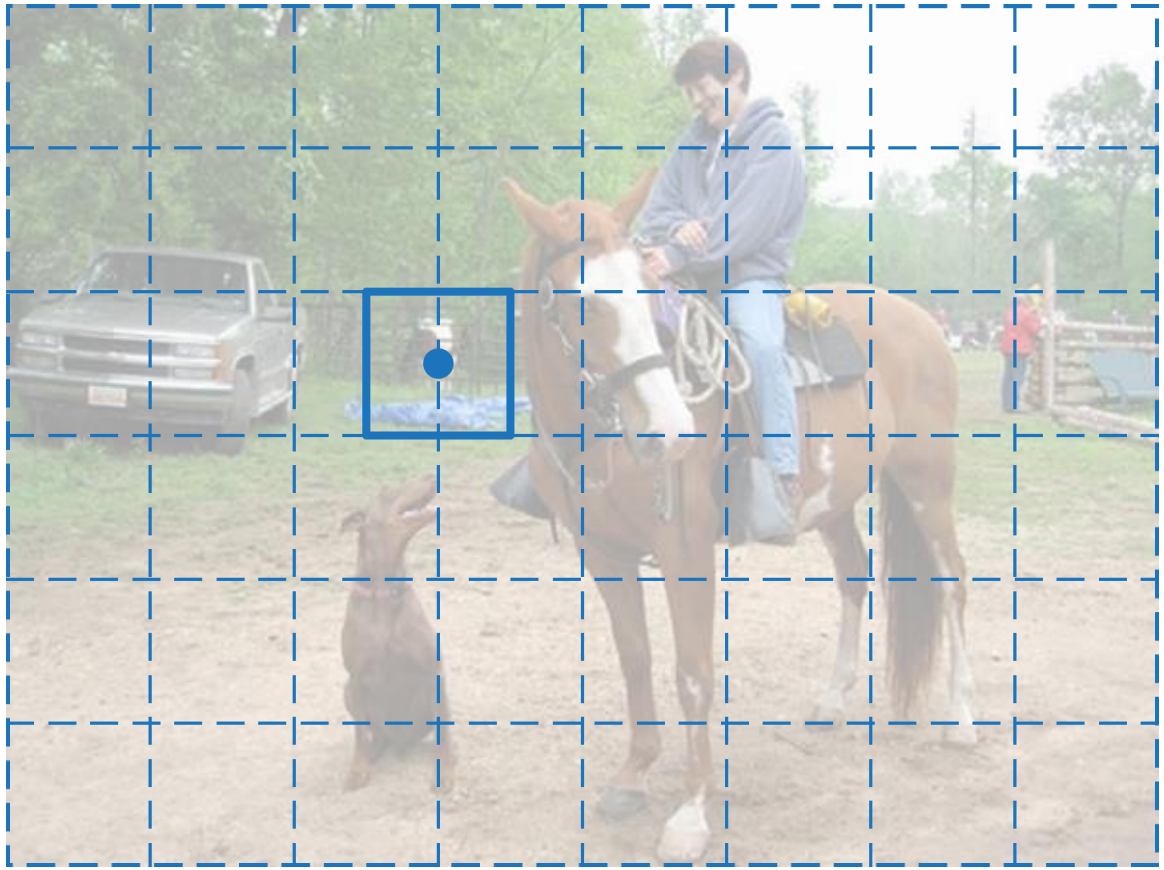


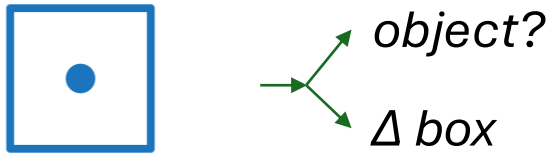
- **ConvNets are Fully Convolutional!**
- Search objects w/ sliding windows (a.k.a., convolutions)

Slides from Kaiming He 6.S058 https://introtocv.github.io/

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NeurIPS 2015

Two Ideas: Pixel-Based and Region-Based Prediction





For each window, predict:

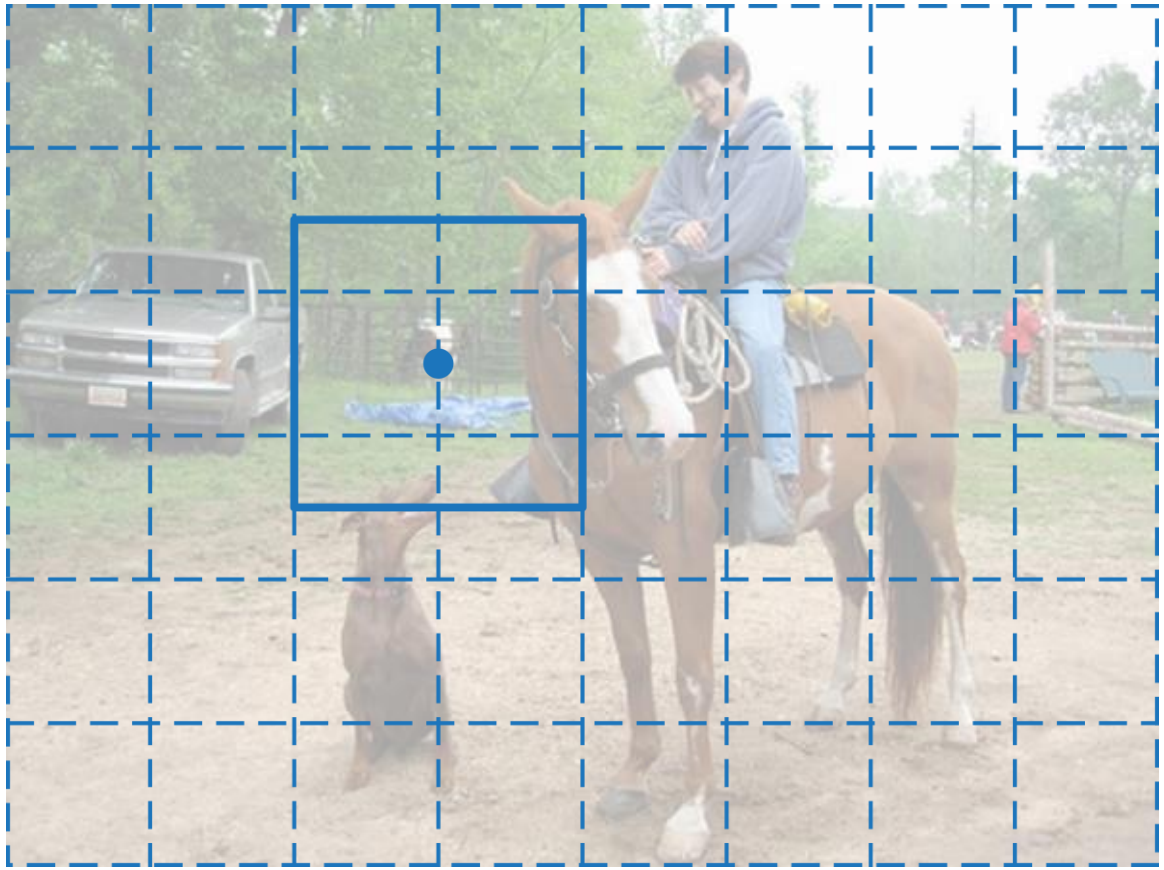
- **Is it an object?**
  - binary classification
- **Object location w.r.t. window?**
  - bbox regression

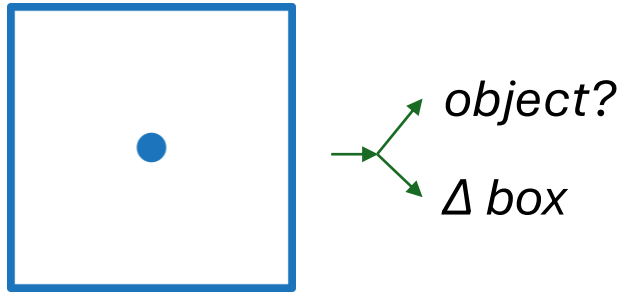
Slides from Kaiming He 6.S058 https://introtocv.github.io/

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NeurIPS 2015

Two Ideas: Pixel-Based and Region-Based Prediction

- **How can neural nets produce regions?**





Anchor

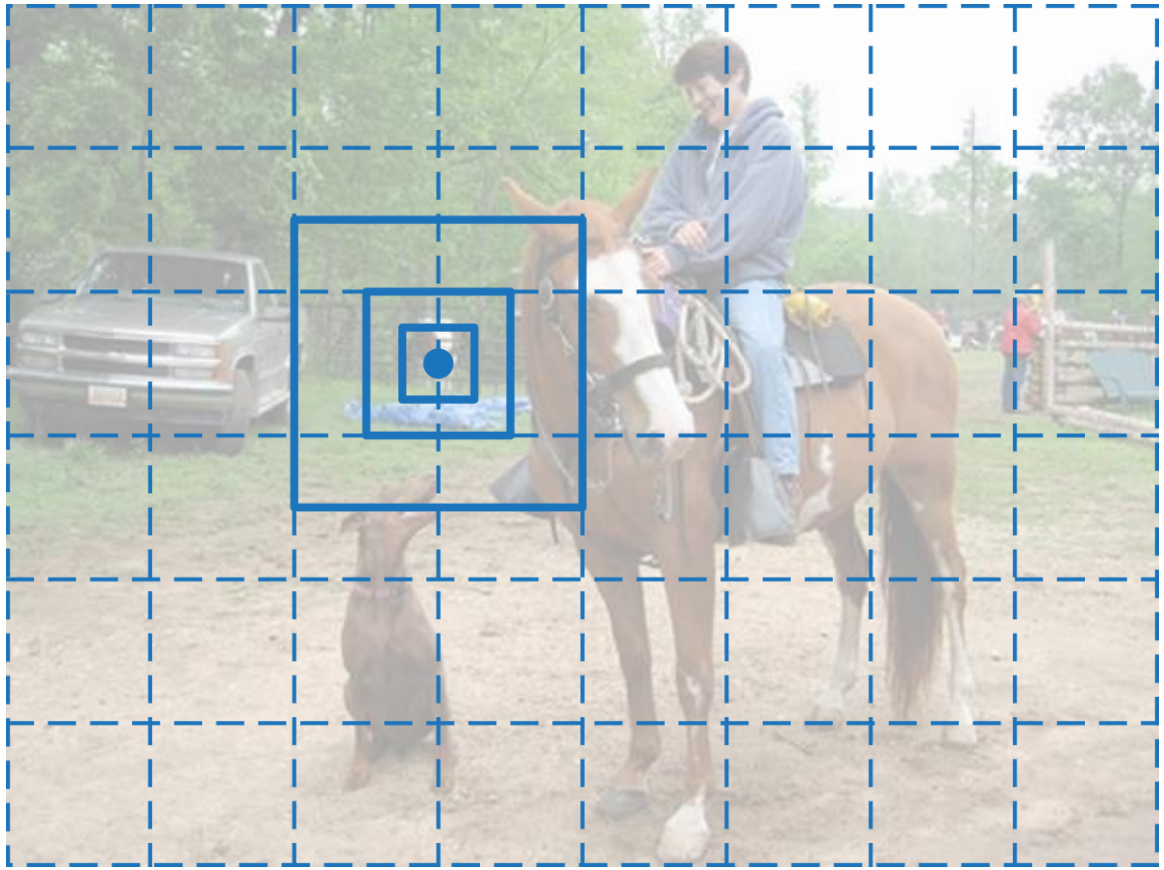
- pred w.r.t. a reference box
- reference box: called “anchor”
- can be different from the input sliding window

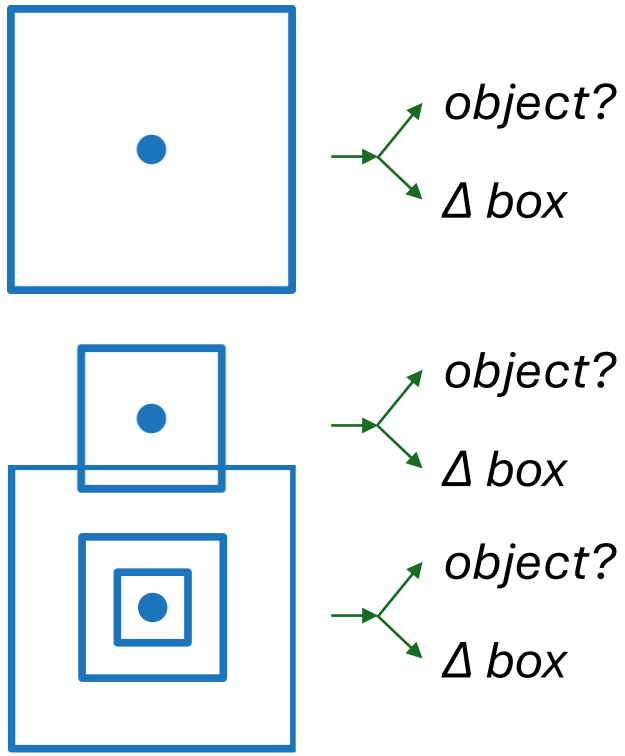
Slides from Kaiming He 6.S058 https://introtocv.github.io/

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NeurIPS 2015

Two Ideas: Pixel-Based and Region-Based Prediction

- **How can neural nets produce regions?**





Anchors

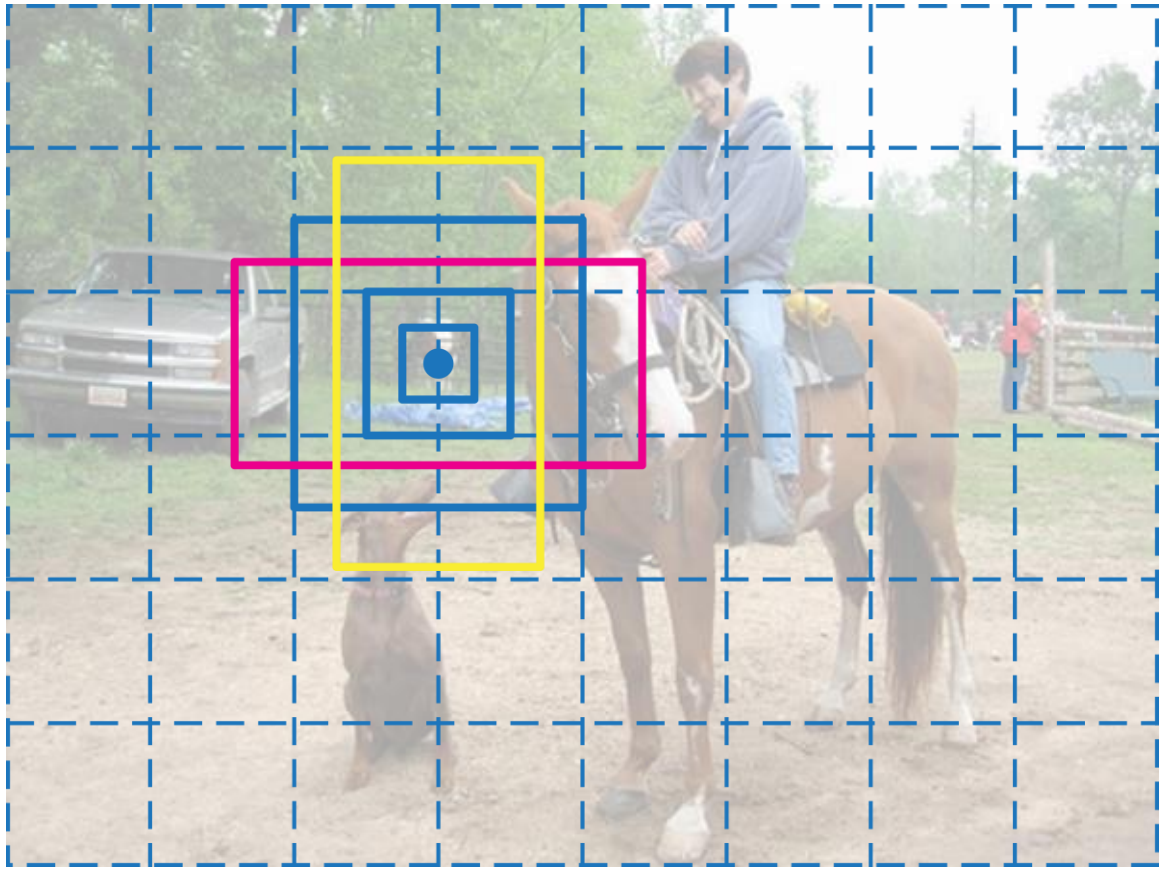
- one sliding window, many anchors
- multiple scales

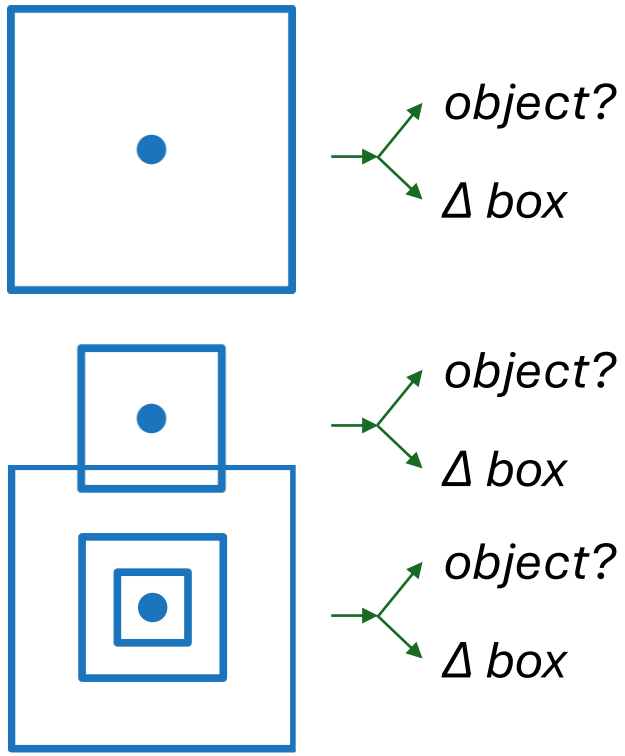
Slides from Kaiming He 6.S058 https://introtocv.github.io/

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NeurIPS 2015

Two Ideas: Pixel-Based and Region-Based Prediction

- **How can neural nets produce regions?**



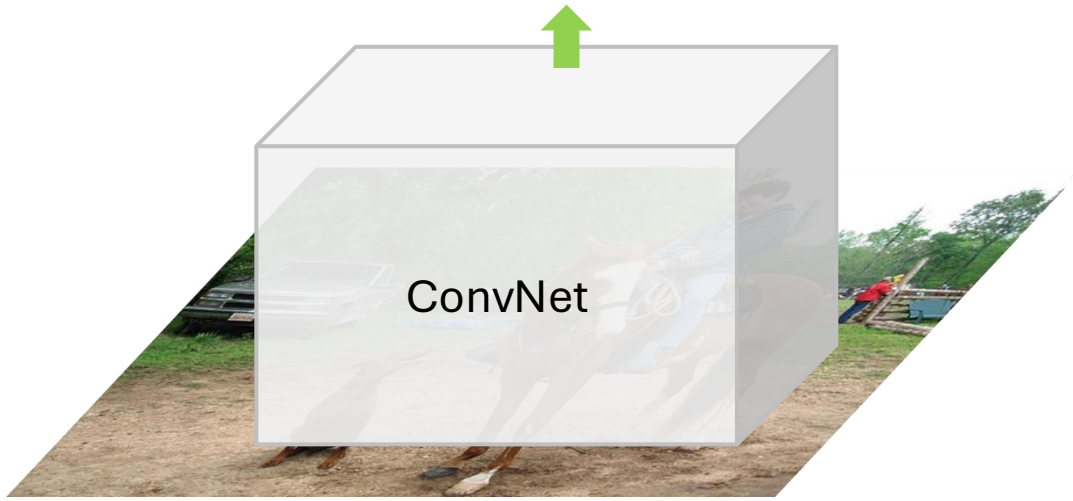


Anchors

- one sliding window, many anchors
- multiple scales
- multiple aspect ratios

Slides from Kaiming He 6.S058 https://introtocv.github.io/

Two Ideas: Pixel-Based and Region-Based Prediction



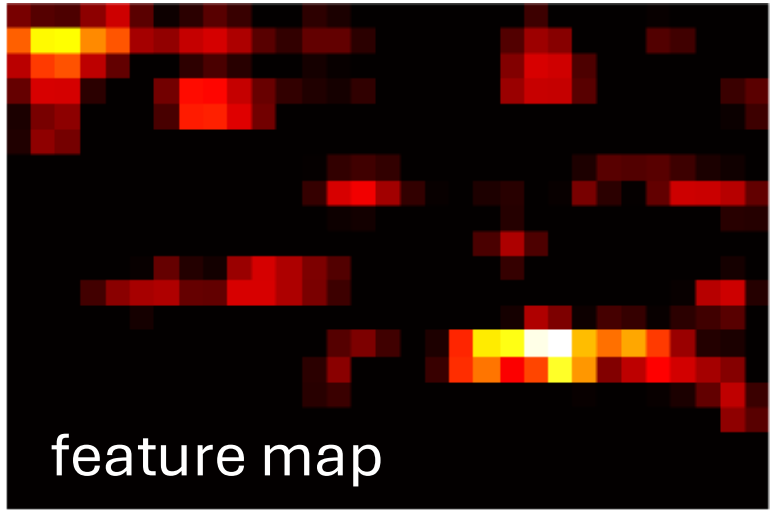
- Apply ConvNet to **whole** image

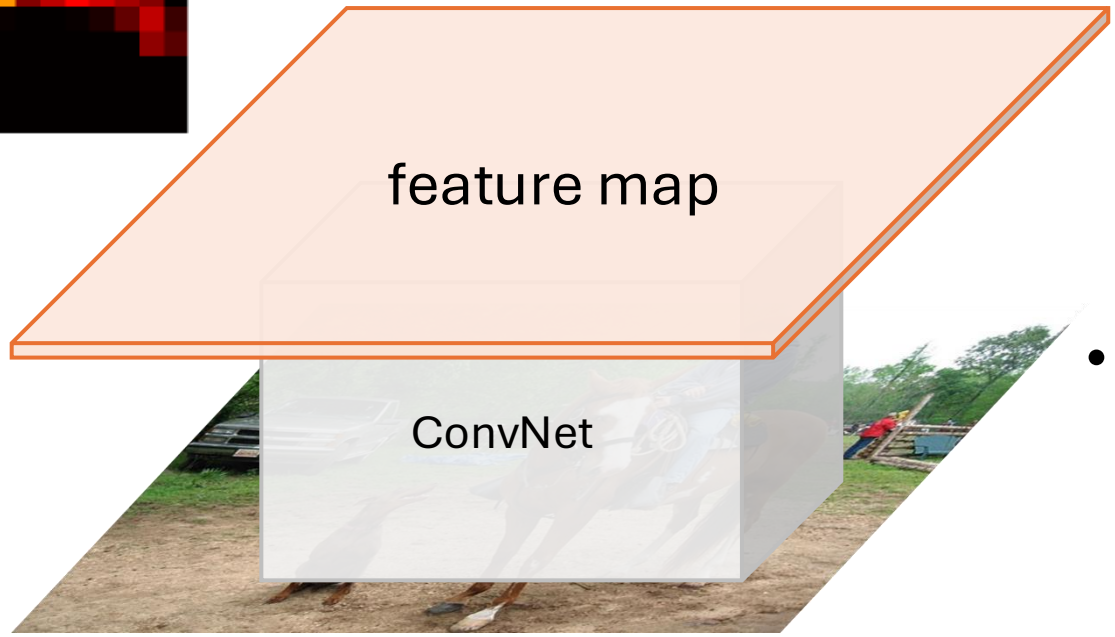
Slides from Kaiming He 6.S058 https://introtocv.github.io/

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NeurIPS 2015

Two Ideas: Pixel-Based and Region-Based Prediction

- **How can neural nets produce regions?**





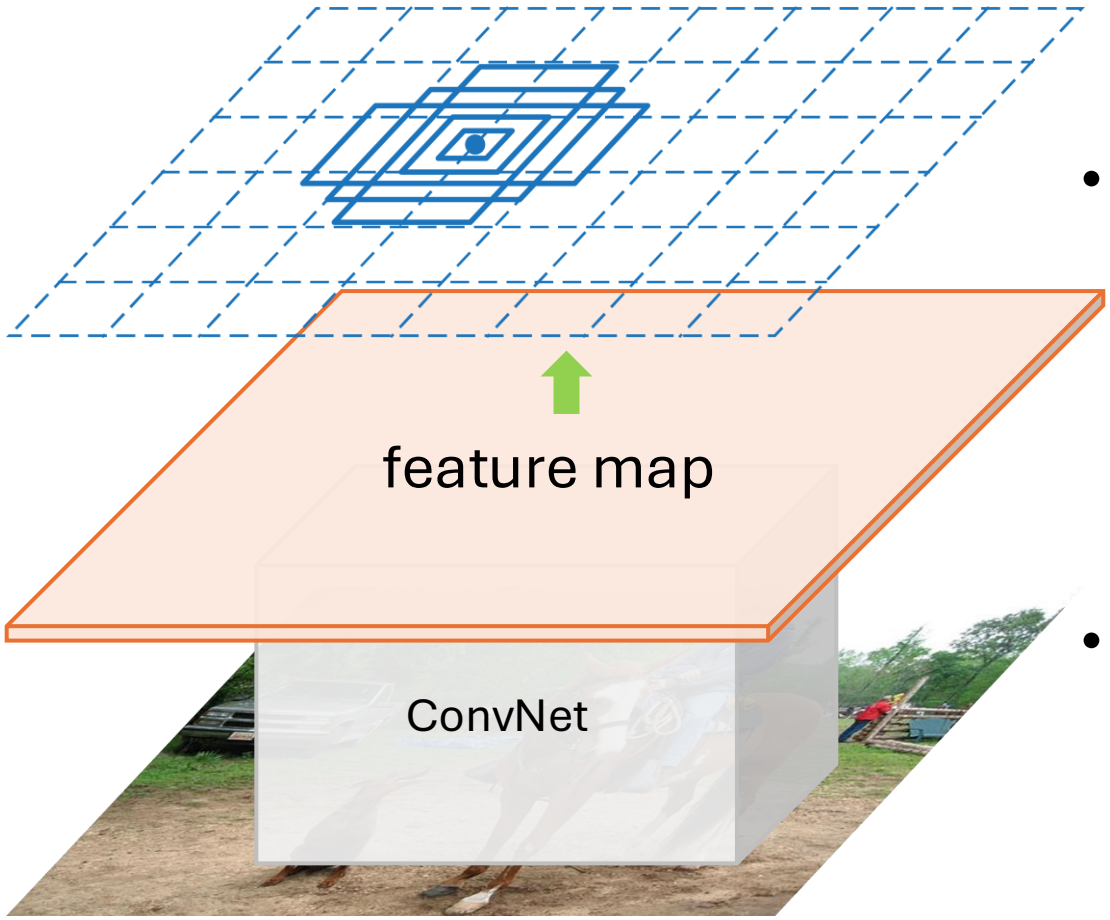
- Apply ConvNet to **whole** image

Slides from Kaiming He 6.S058 https://introtocv.github.io/

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NeurIPS 2015

Two Ideas: Pixel-Based and Region-Based Prediction

- **How can neural nets produce regions?**



- Convs for *k* anchors
  - *k*-d: binary classification
  - *4k*-d: bbox regression

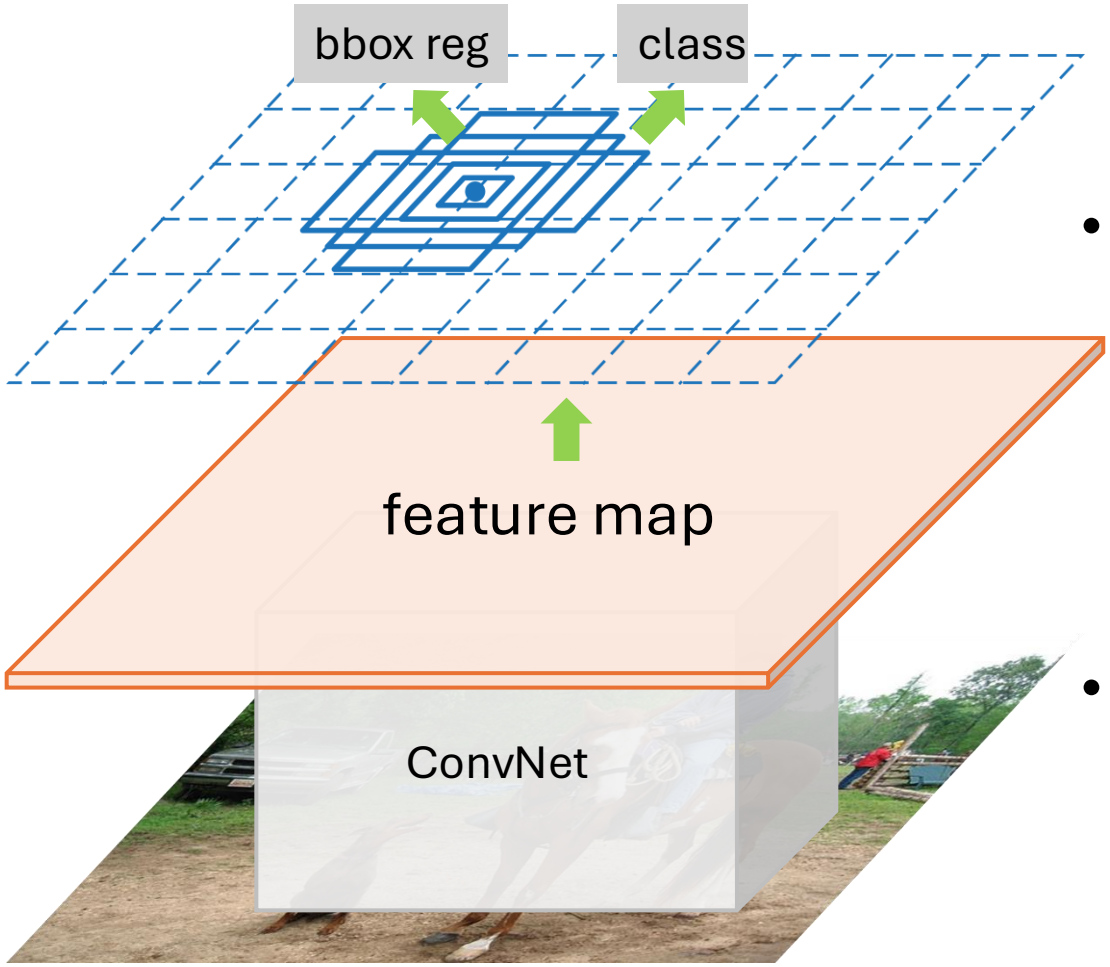
- Apply ConvNet to **whole** image

Slides from Kaiming He 6.S058 https://introtocv.github.io/

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NeurIPS 2015

Two Ideas: Pixel-Based and Region-Based Prediction

- **How can neural nets produce regions?**



- Loss: reg + class
  - over all locations
  - over all anchors
- Convs for *k* anchors
  - *k*-d: binary classification
  - *4k*-d: bbox regression
- Apply ConvNet to **whole** image

Slides from Kaiming He 6.S058 https://introtocv.github.io/

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NeurIPS 2015

Two Ideas: Pixel-Based and Region-Based Prediction

Slides adapted from: Ross Girshick, ICCV 2015

Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, CVPR 2014

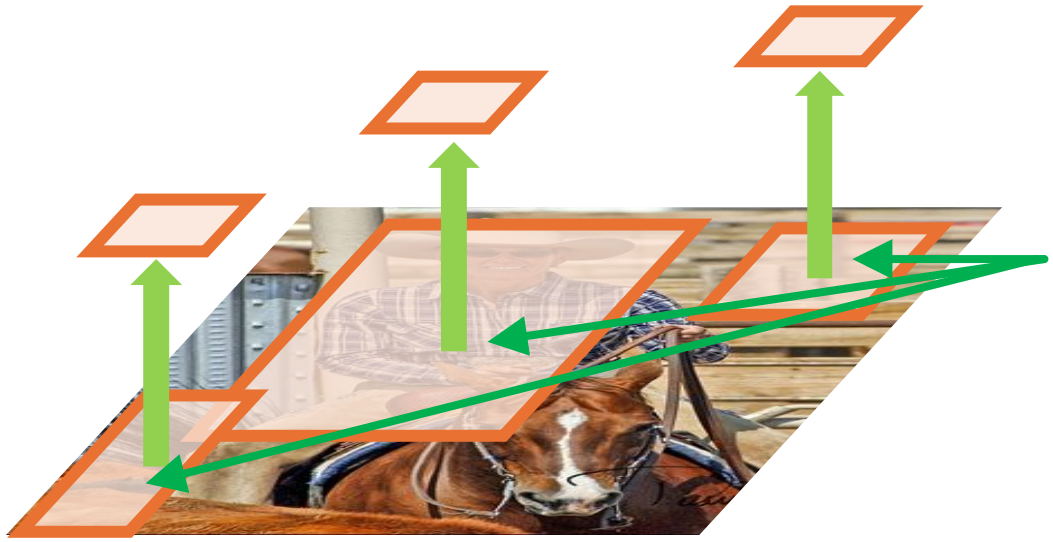
Two Ideas: Pixel-Based and Region-Based Prediction

- Propose
  Regions of Interest (RoI)

Slides adapted from: Ross Girshick, ICCV 2015

Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, CVPR 2014

Two Ideas: Pixel-Based and Region-Based Prediction

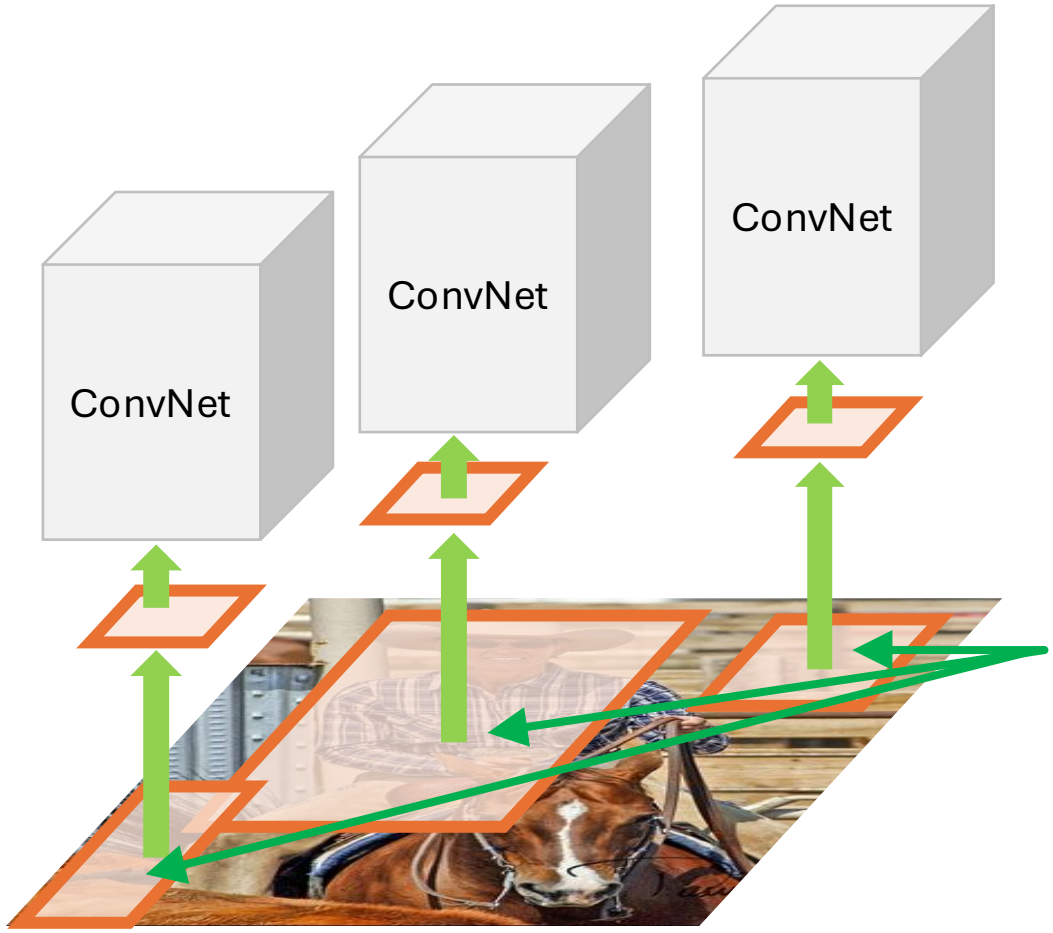


- Warp regions
- Propose
  Regions of Interest (RoI)

Slides adapted from: Ross Girshick, ICCV 2015

Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, CVPR 2014

Two Ideas: Pixel-Based and Region-Based Prediction

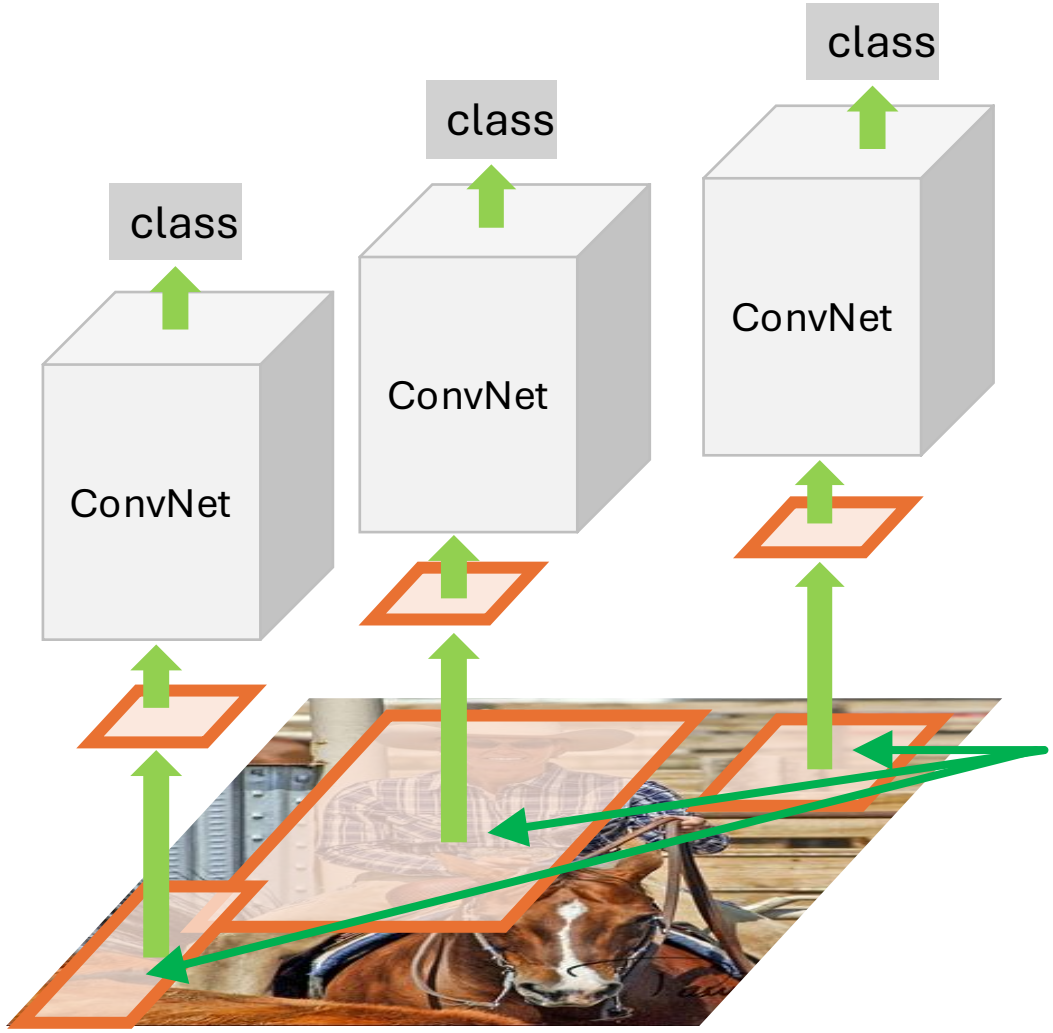


- Apply ConvNets to regions
- Warp regions
- Propose
  Regions of Interest (RoI)

Slides adapted from: Ross Girshick, ICCV 2015

Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, CVPR 2014

Two Ideas: Pixel-Based and Region-Based Prediction

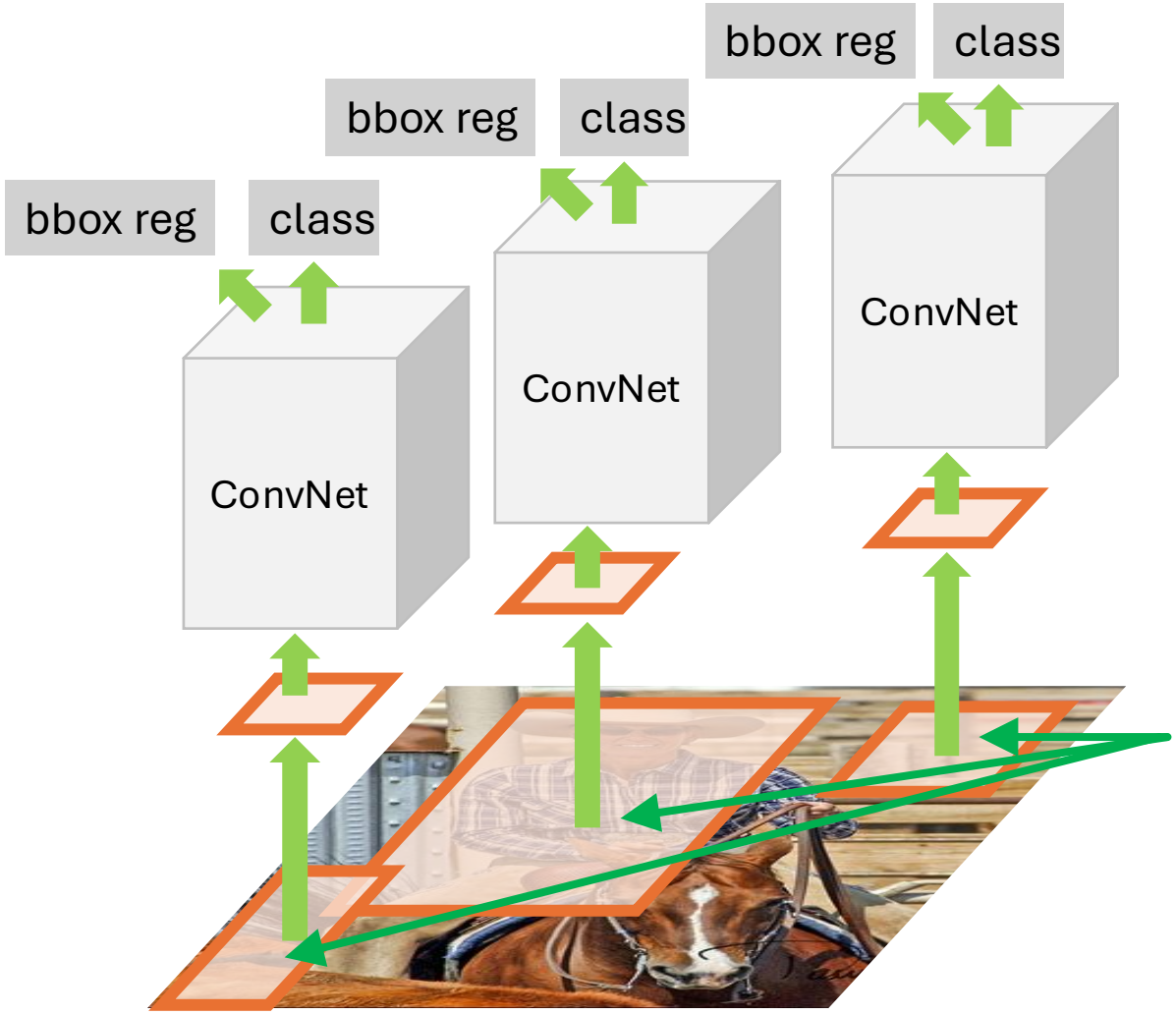


- Classify regions
- Apply ConvNets to regions
- Warp regions
- Propose Regions of Interest (RoI)

Slides adapted from: Ross Girshick, ICCV 2015

Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

Two Ideas: Pixel-Based and Region-Based Prediction



- Regress bounding-boxes
- Classify regions
- Apply ConvNets to regions
- Warp regions
- Propose Regions of Interest (RoI)

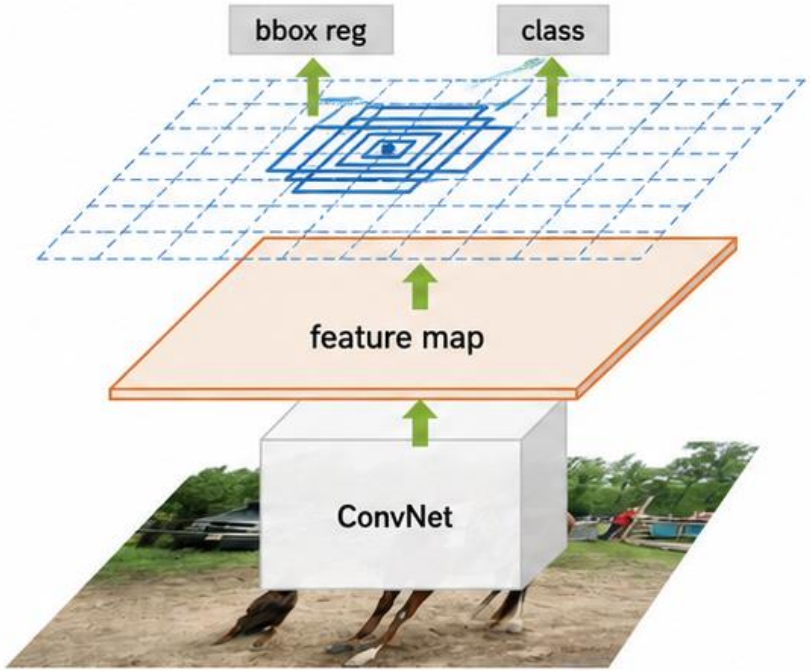
Slides adapted from: Ross Girshick, ICCV 2015

Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, CVPR 2014

Two Ideas: Pixel-Based and Region-Based Prediction

Pixel-Based Prediction

One ConvNet over the whole image;
predict class + box at many grid locations / anchors.



Pros

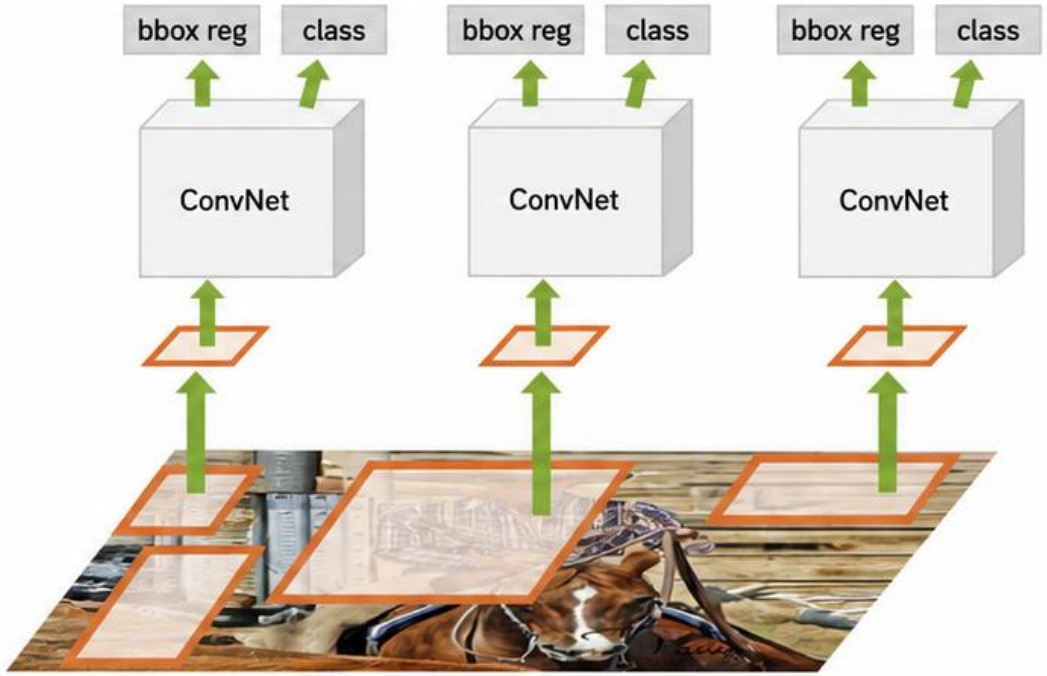
- Fast shared features
- Dense coverage
- End-to-end proposal generation

Cons

- Many anchors to score
- Class imbalance
- Anchor / NMS tuning

Region-Based Prediction

Propose RoIs, crop or warp each region,
then classify and refine boxes.



Pros

- Focuses on candidate regions
- Strong per-region features
- Simple classify + regress setup

Cons

- Slower for many regions
- Depends on proposal quality
- Warping can lose context

Two Ideas: Pixel-Based and Region-Based Prediction

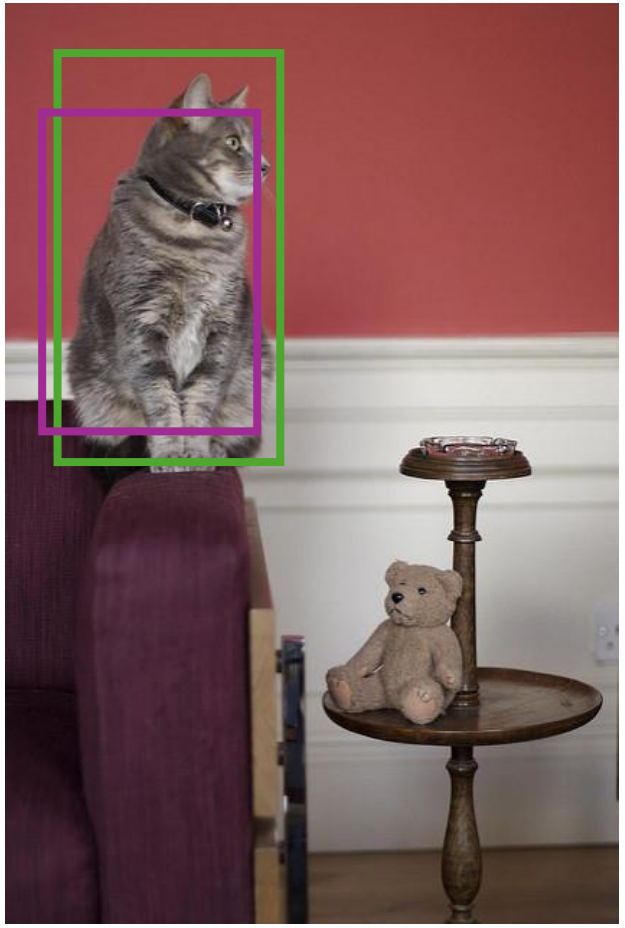
Faster-RCNN / Mask-RCNN are all two-stage frameworks

Two Ideas: Pixel-Based and Region-Based Prediction

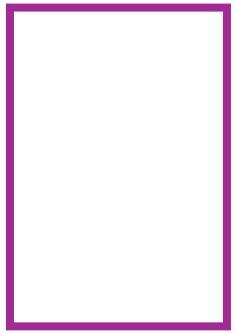
Faster-RCNN / Mask-RCNN are all two-stage frameworks

Can we add more stages, e.g., for bounding box refinement?

Bounding Box Regression: Recap



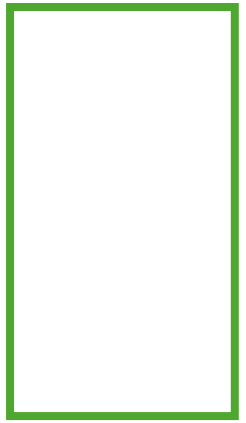
(x_1, y_1, w_1, h_1)



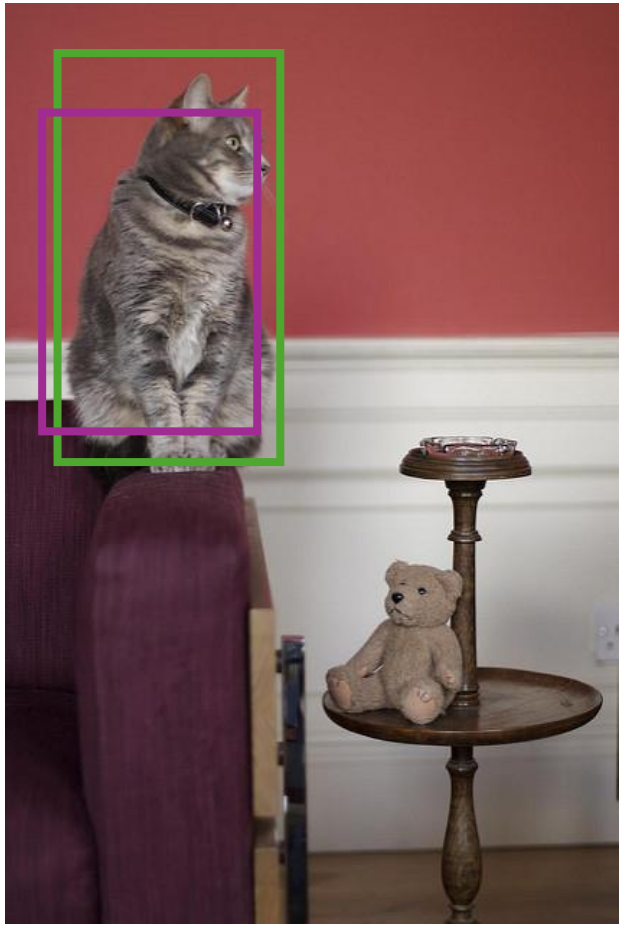
Transformation

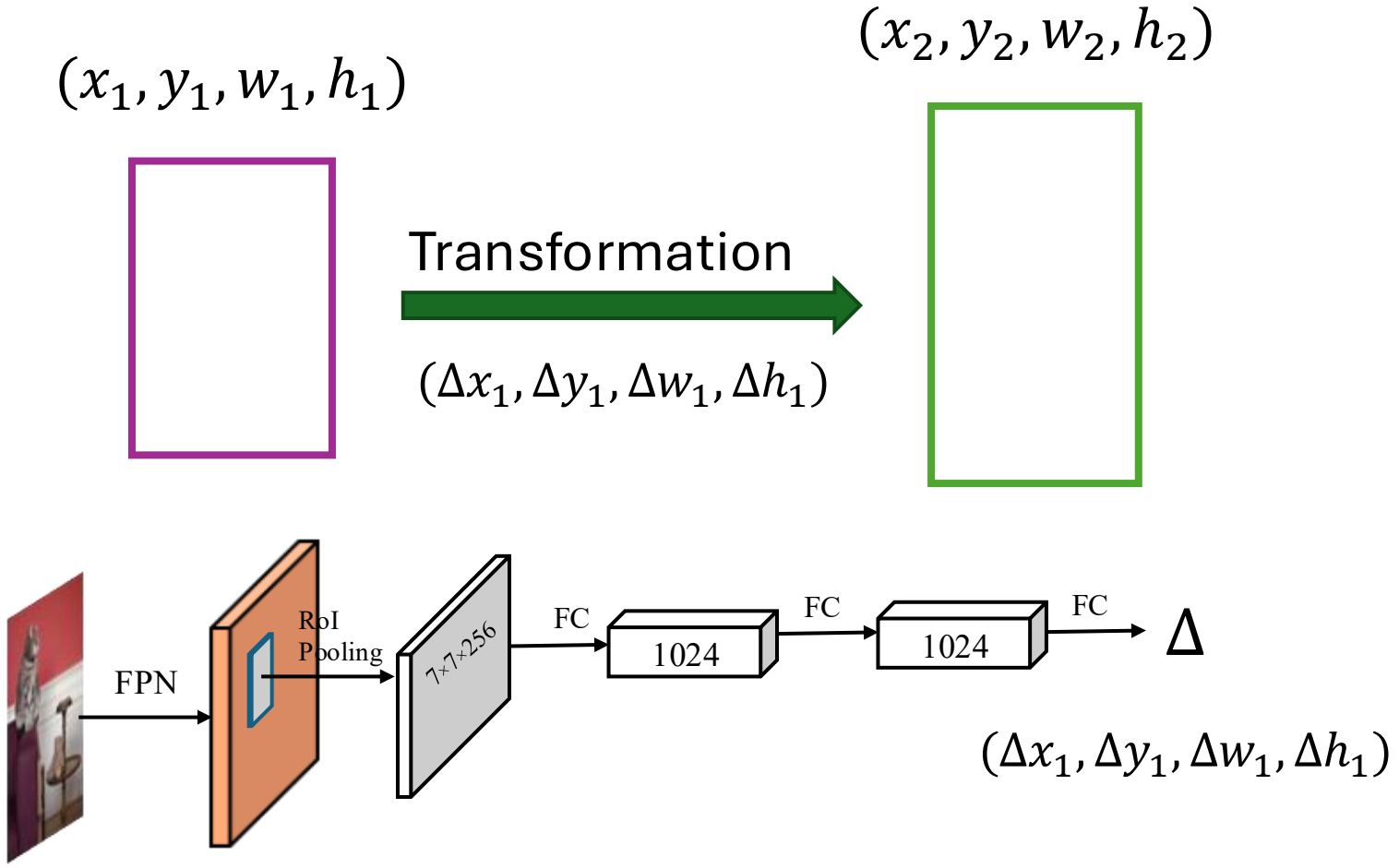
$(\Delta x_1, \Delta y_1, \Delta w_1, \Delta h_1)$

(x_2, y_2, w_2, h_2)



Bounding Box Regression: Recap





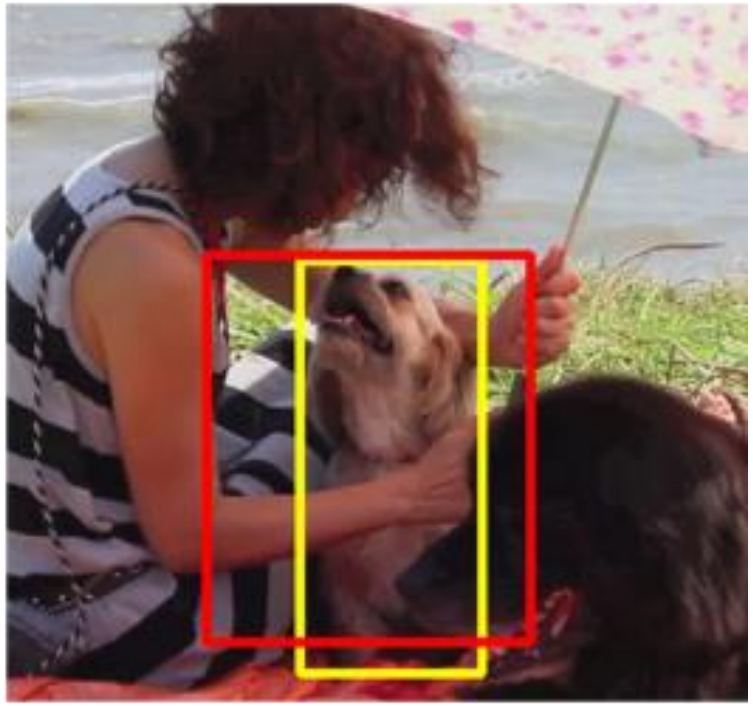
Iterative Bounding Box Regression

Original Proposal
IoU=0.62

Iterative Bounding Box Regression

Apply! Better
IoU=0.65

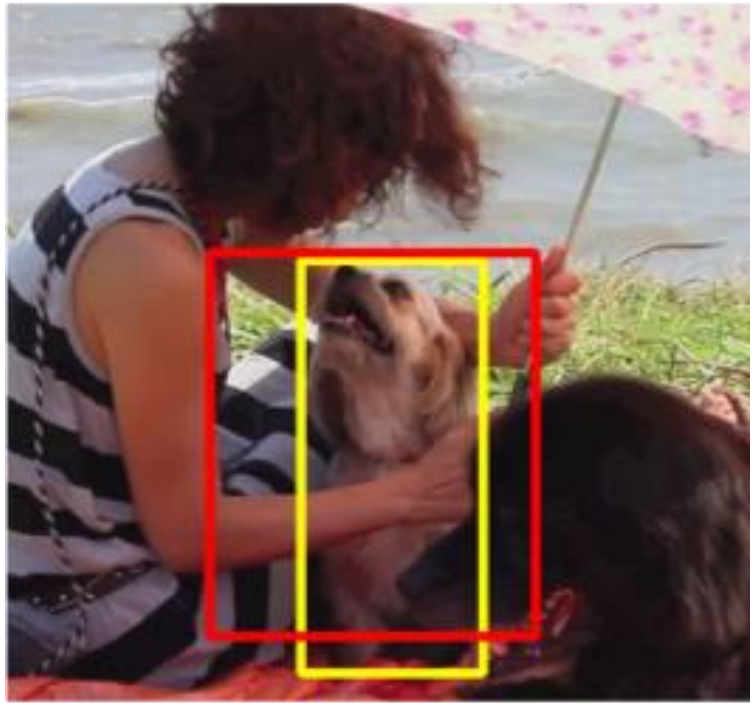
Iterative Bounding Box Regression



Apply again! Worse

IoU=0.49

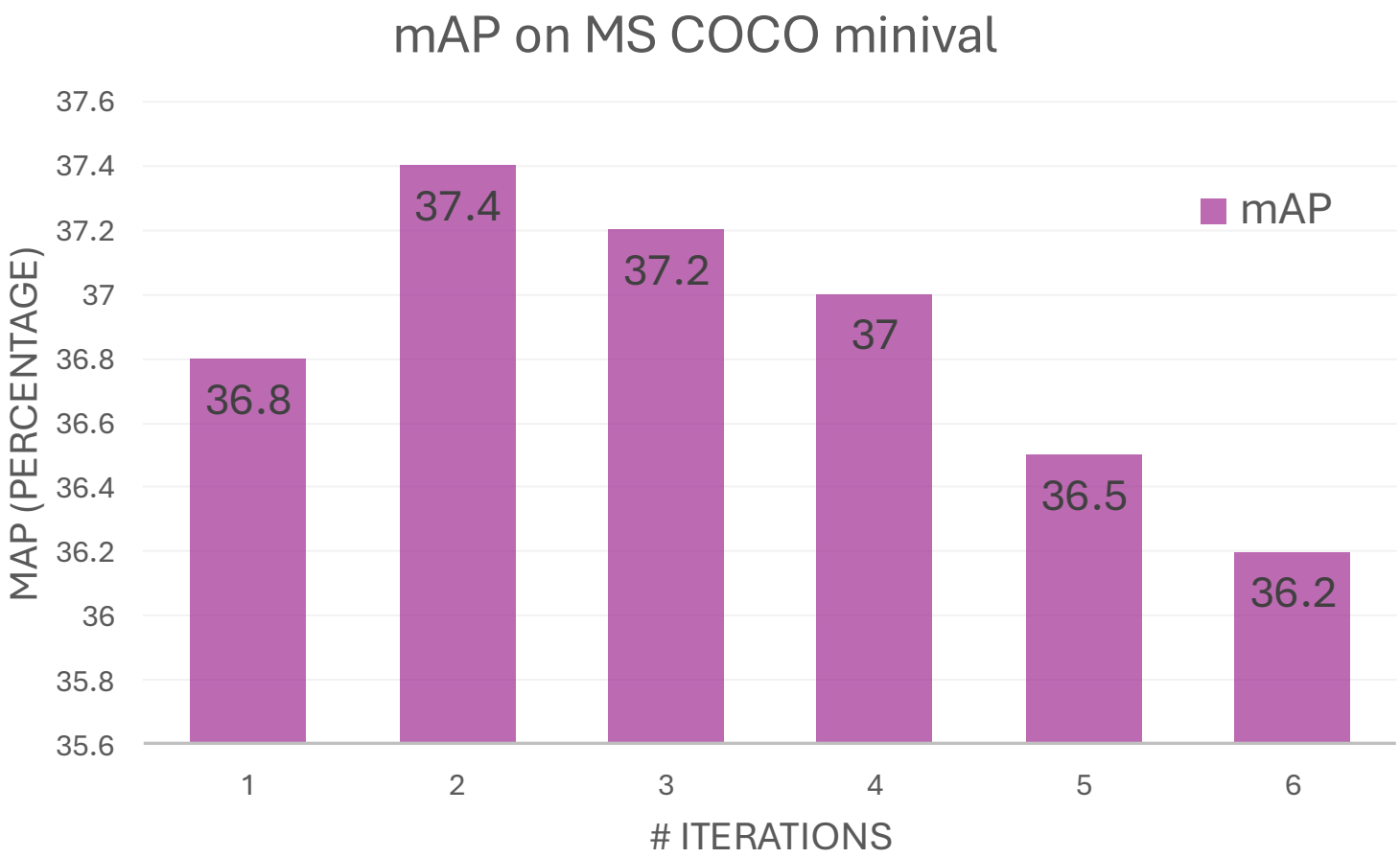
Iterative Bounding Box Regression



One more time? Even worse

IoU = 0.48

Iterative Bounding Box Regression



Cai, Z., & Vasconcelos, N. “Cascade R-CNN: Delving into High Quality Object Detection.” *in CVPR 2018*

Distribution Mismatch Yields Performance Degradation

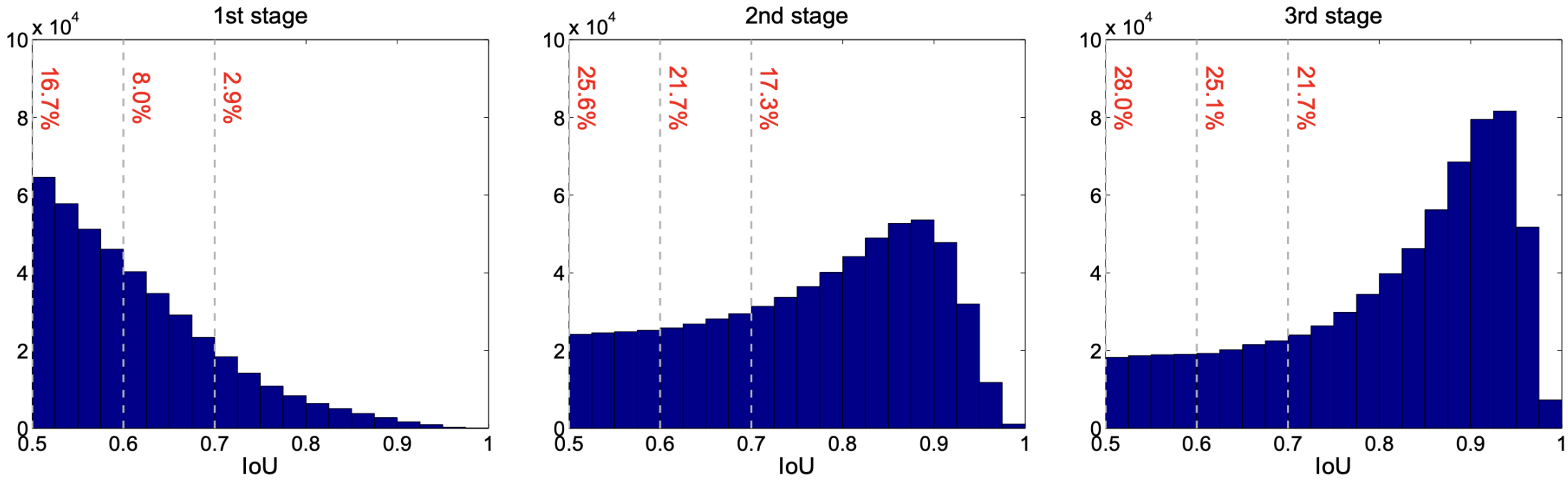
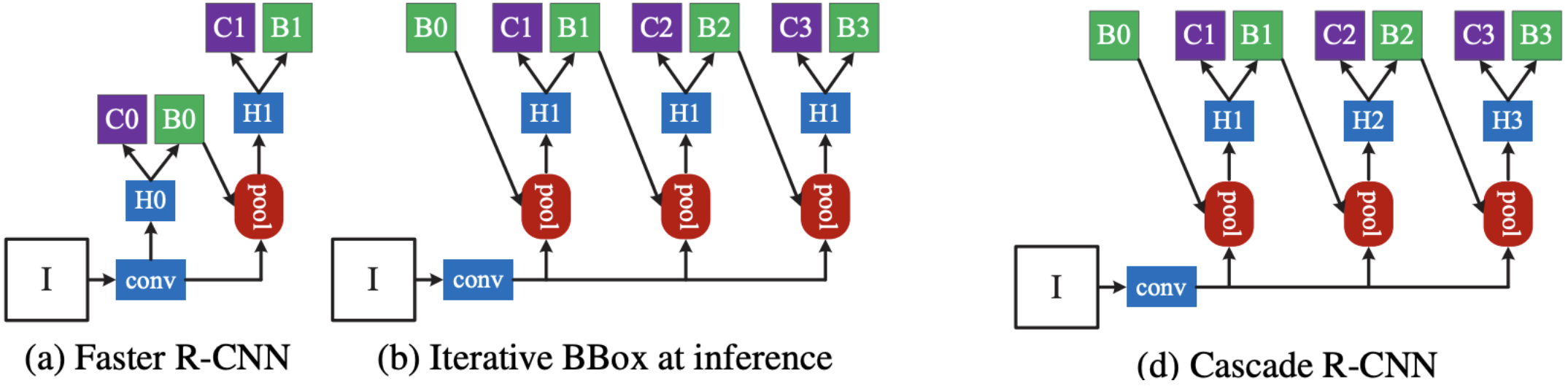


Figure 4. The IoU histogram of training samples. The distribution at 1st stage is the output of RPN. The red numbers are the positive percentage higher than the corresponding IoU threshold.

Cai, Z., & Vasconcelos, N. "Cascade R-CNN: Delving into High Quality Object Detection." *in CVPR 2018*

Solution 1: Using Different Network Weights

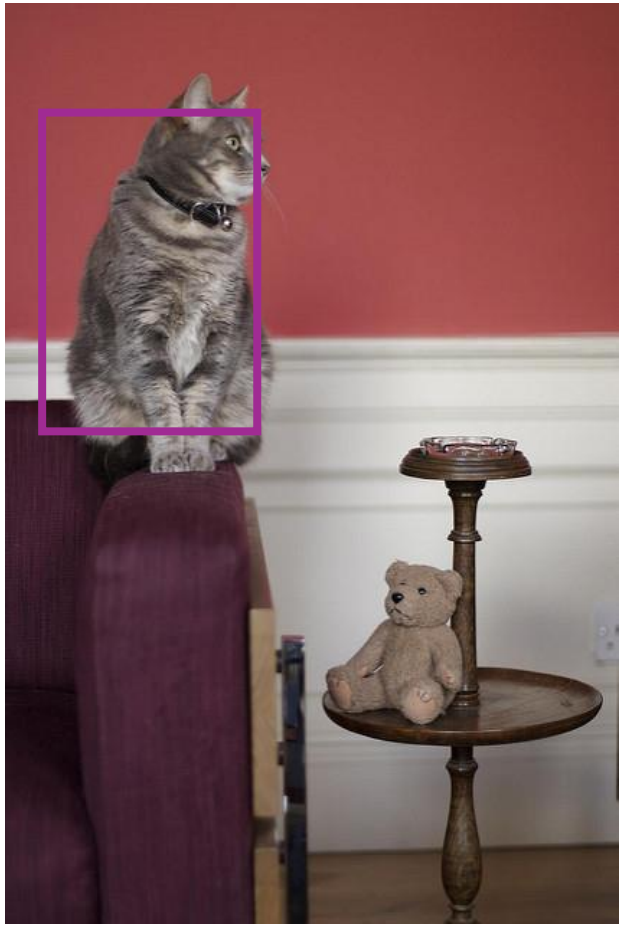


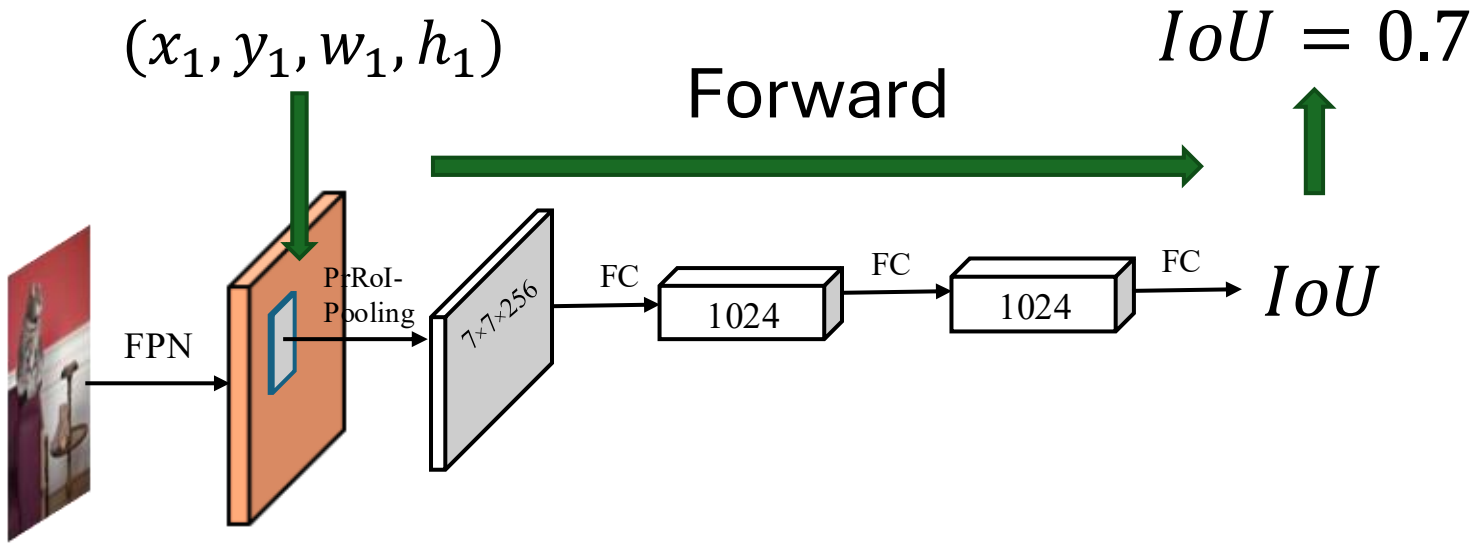
(a) Faster R-CNN (b) Iterative BBox at inference (d) Cascade R-CNN

A fix: use different fully-connected layers for different iterations

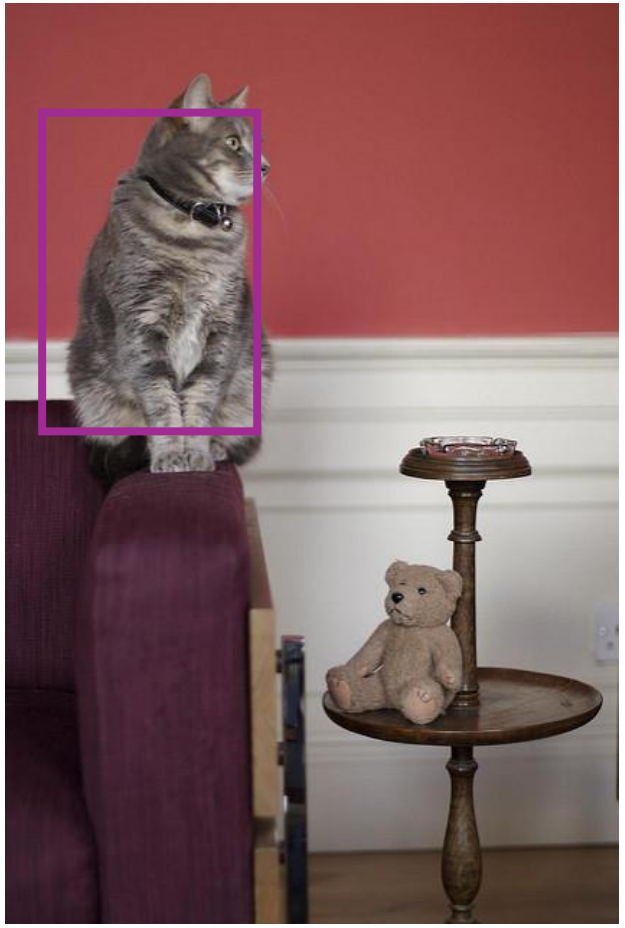
Cai, Z., & Vasconcelos, N. “Cascade R-CNN: Delving into High Quality Object Detection.” *in CVPR 2018*

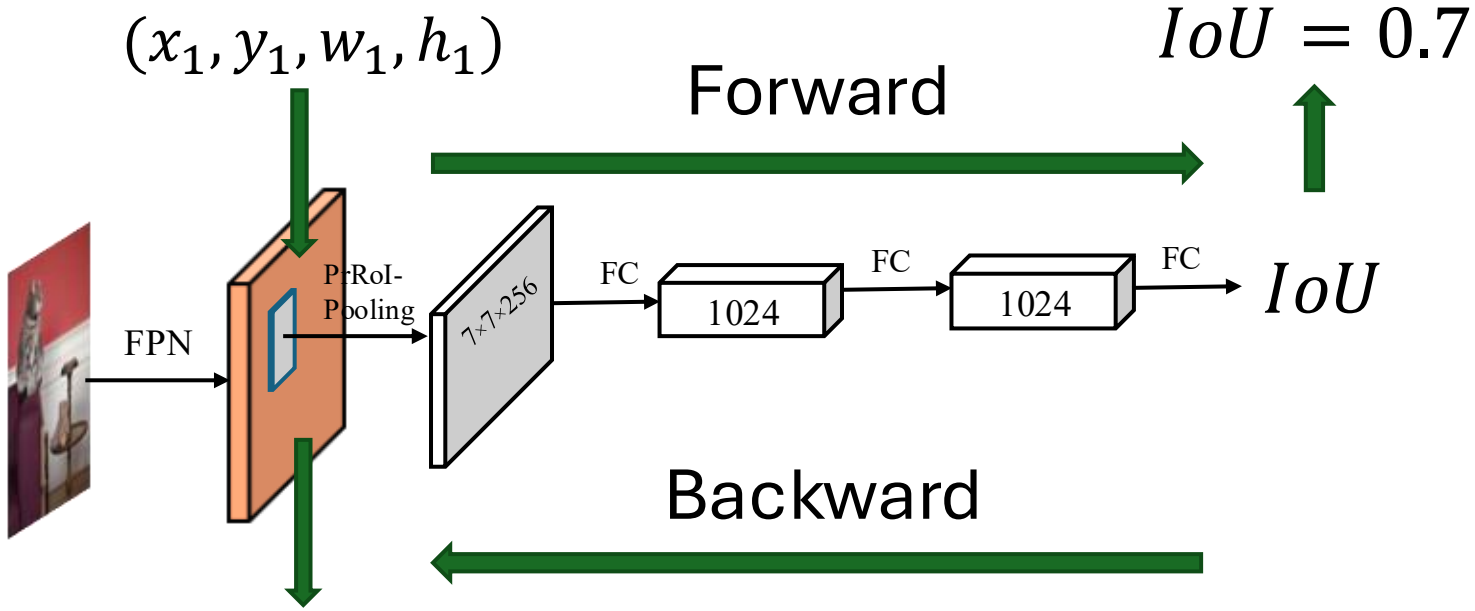
Solution 2: Optimization-Based Formulation





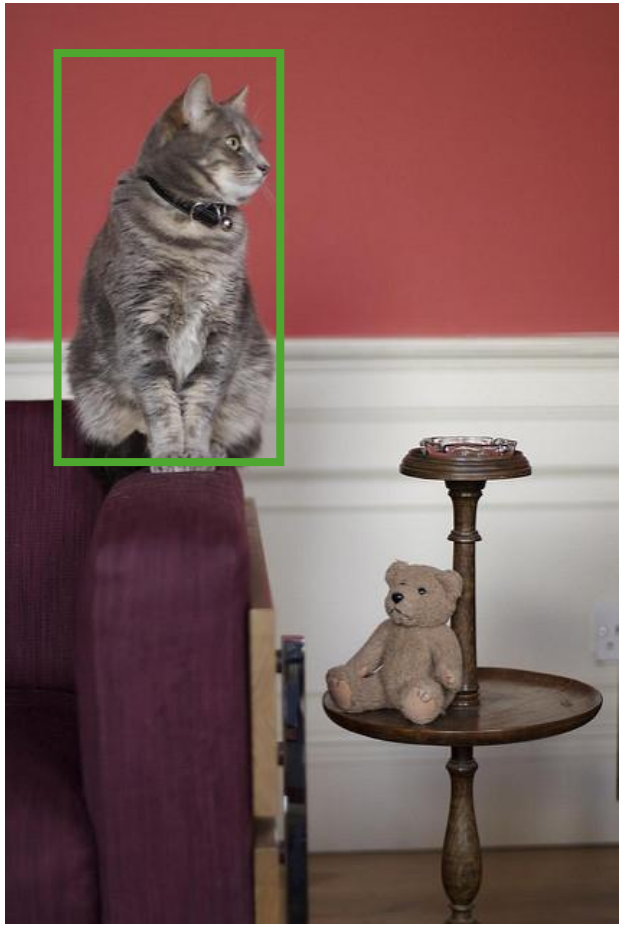
Solution 2: Optimization-Based Formulation



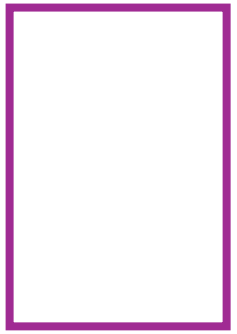


$$\left(\frac{\partial IoU}{\partial x_1}, \frac{\partial IoU}{\partial y_1}, \frac{\partial IoU}{\partial w_1}, \frac{\partial IoU}{\partial h_1}\right)$$

Solution 2: Optimization-Based Formulation



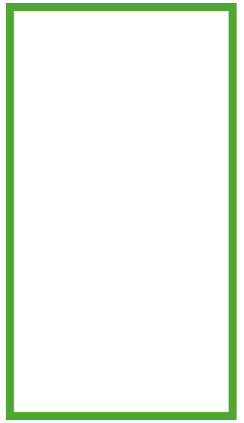
(x_1, y_1, w_1, h_1)



Transformation

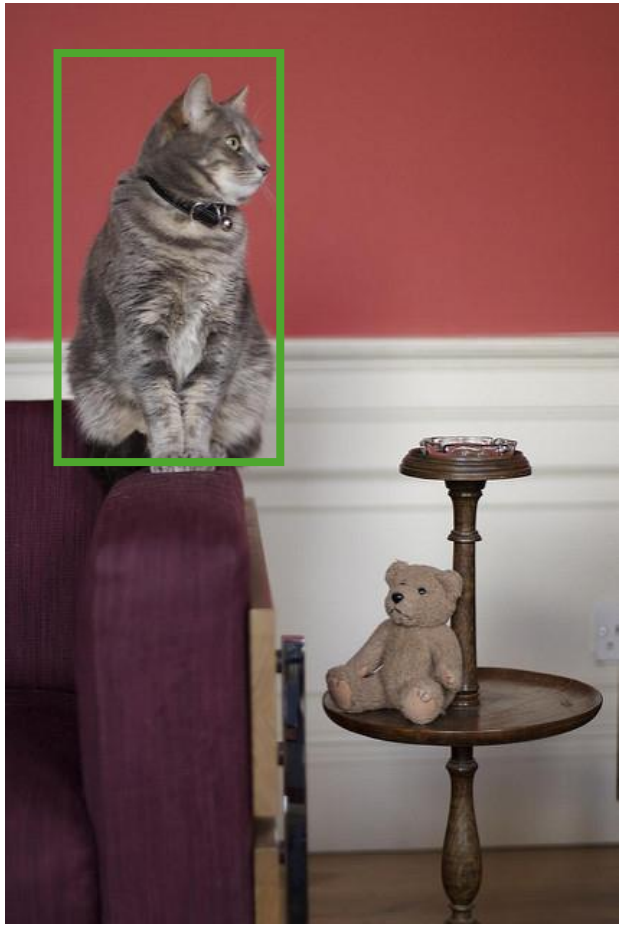
$$\left(\frac{\partial IoU}{\partial x_1}, \frac{\partial IoU}{\partial y_1}, \frac{\partial IoU}{\partial w_1}, \frac{\partial IoU}{\partial h_1}\right)$$

(x_2, y_2, w_2, h_2)



$$\left(x_1 + \gamma \cdot \frac{\partial IoU}{\partial x_1}, y_1 + \gamma \cdot \frac{\partial IoU}{\partial y_1}, w_1 + \gamma \cdot \frac{\partial IoU}{\partial w_1}, h_1 + \gamma \cdot \frac{\partial IoU}{\partial h_1}\right)$$

Solution 2: Optimization-Based Formulation



(x_1, y_1, w_1, h_1)

(x_2, y_2, w_2, h_2)

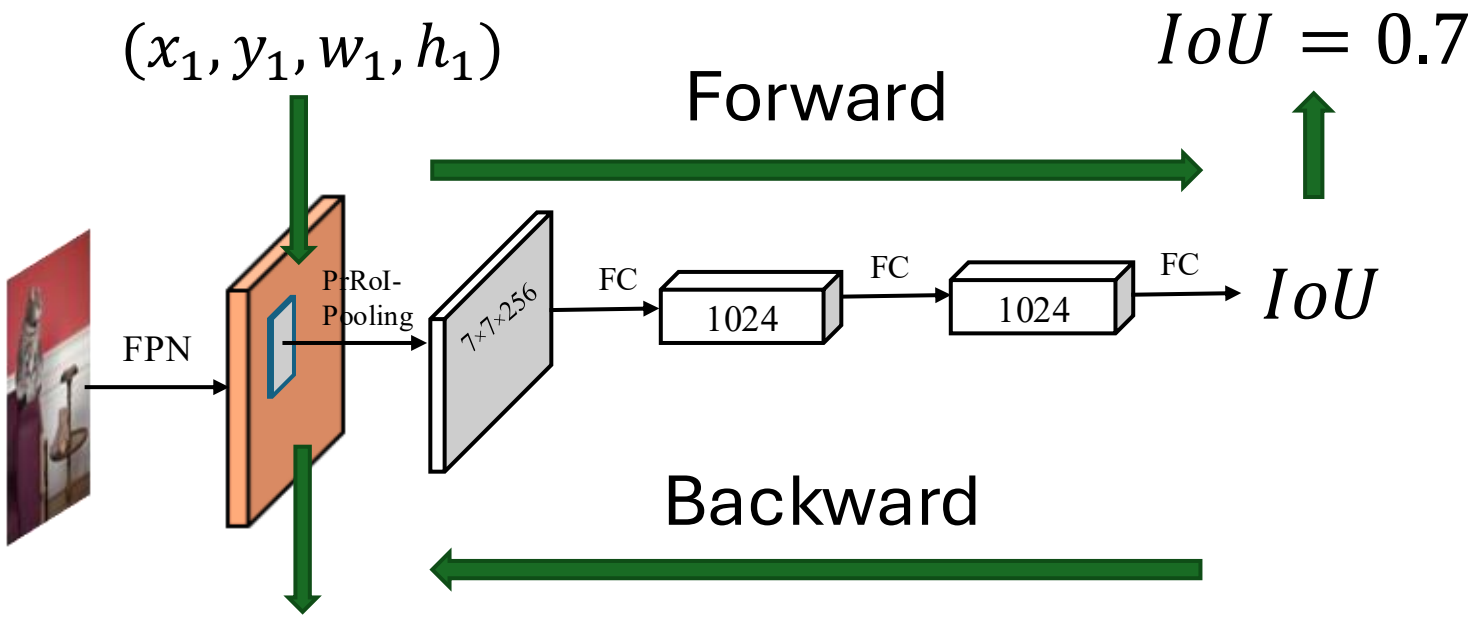
Transformation

$$\left(\frac{\partial IoU}{\partial x_1}, \frac{\partial IoU}{\partial y_1}, \frac{\partial IoU}{\partial w_1}, \frac{\partial IoU}{\partial h_1}\right)$$

Repeat!
Iterative refinement.

$$\left(x_1 + \gamma \cdot \frac{\partial IoU}{\partial x_1}, y_1 + \gamma \cdot \frac{\partial IoU}{\partial y_1}, w_1 + \gamma \cdot \frac{\partial IoU}{\partial w_1}, h_1 + \gamma \cdot \frac{\partial IoU}{\partial h_1}\right)$$

Optimization-Based Formulation



$$\left(\frac{\partial IoU}{\partial x_1}, \frac{\partial IoU}{\partial y_1}, \frac{\partial IoU}{\partial w_1}, \frac{\partial IoU}{\partial h_1}\right)$$

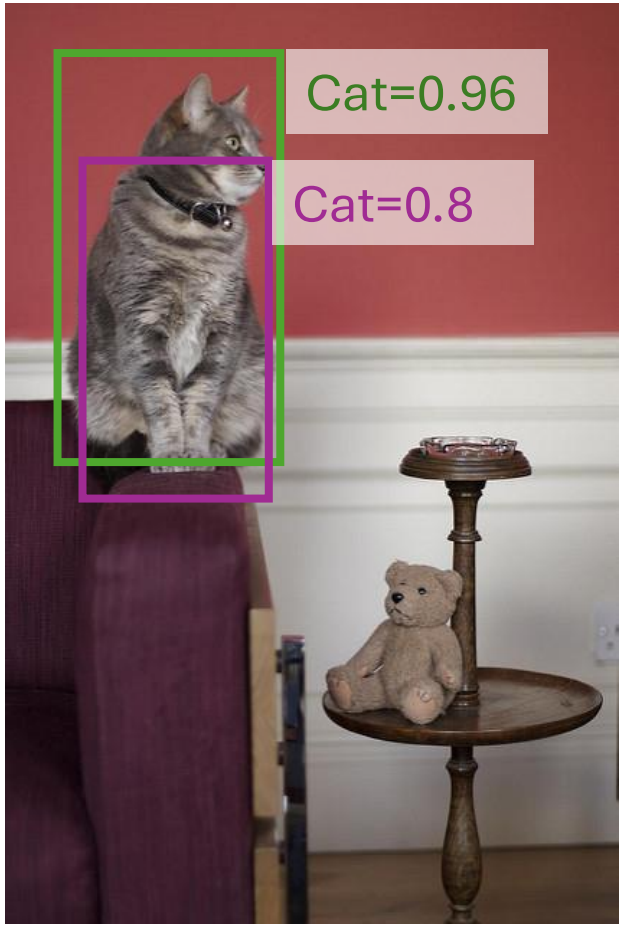
Acquisition of Localization Confidence for Accurate Object Detection

Borui Jiang*[1,3], Ruixuan Luo*[1,3], Jiayuan Mao*[2,4], Tete Xiao[1,3], and Yuning Jiang[4]

Bounding Box Merging

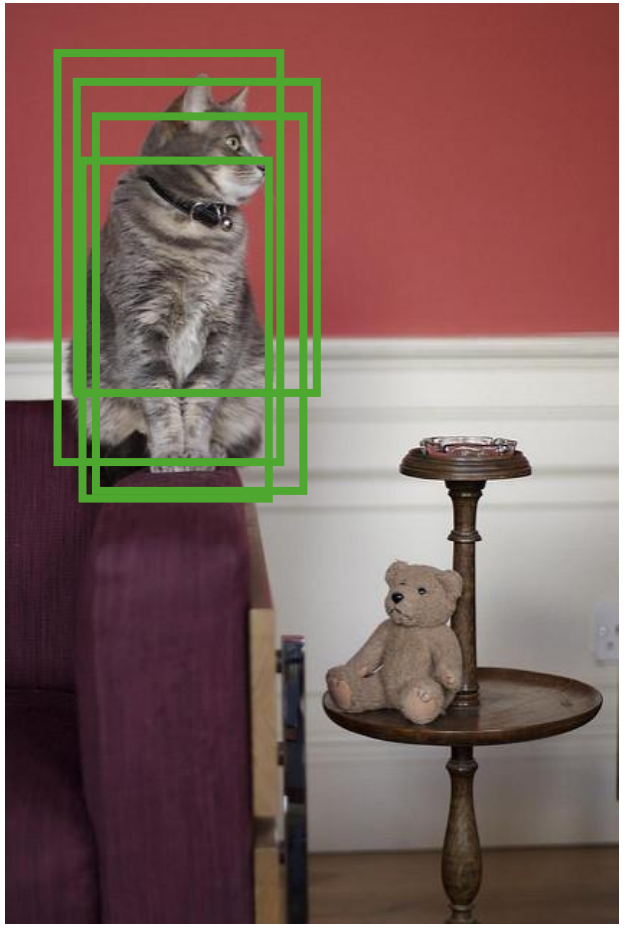
Classical solution: Non-Maximum Suppression

Bounding Box Merging: Non-Maximum Suppression



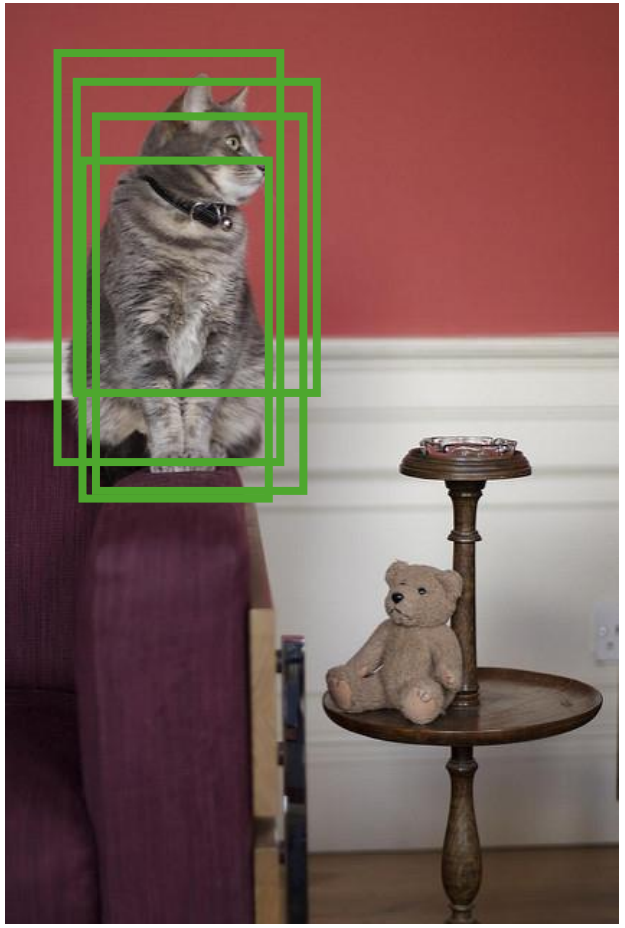
1. Sort all boxes by classification confidence in descending order.
2. If $IoU(\square, \square) > 0.5$, suppress the one with lower confidence(in this case) .
3. Repeat 2 until there is no such (,).

Bounding Box Merging: Non-Maximum Suppression



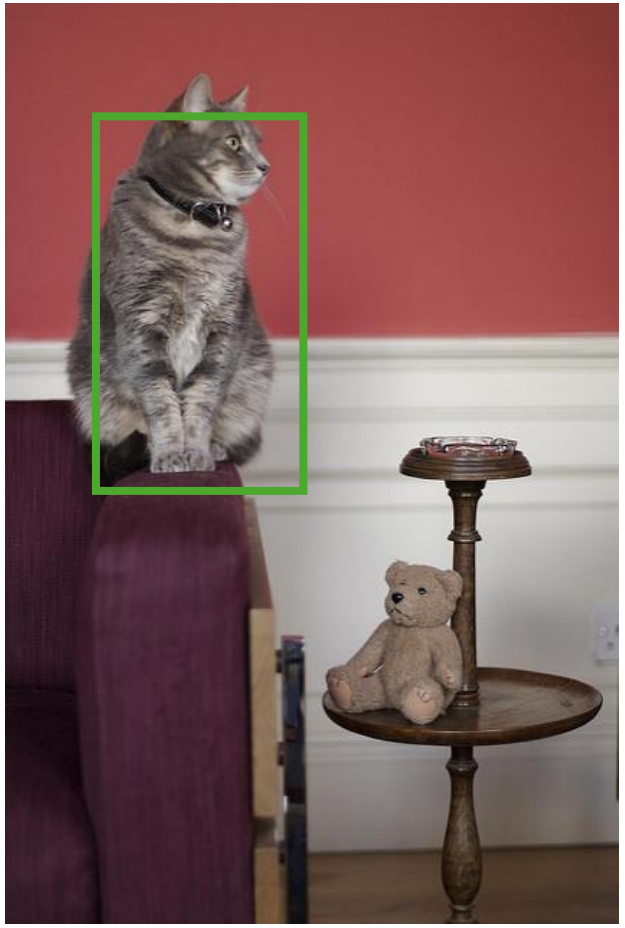
	Cls. Conf.
Box 1	0.85
Box 2	0.9
Box 3	0.96
Box 4	0.85
.....	

Bounding Box Merging: Non-Maximum Suppression



	Cls. Conf.
Box 1	0.85
Box 2	0.9
Box 3	0.96
Box 4	0.85
.....	

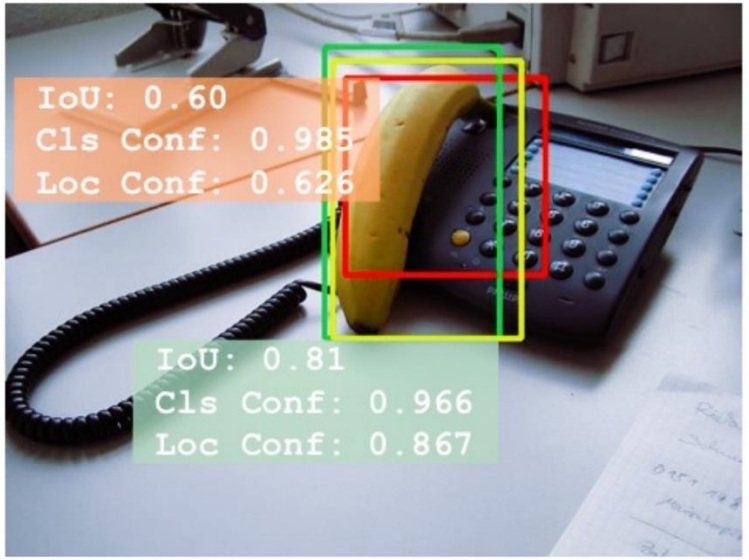
Bounding Box Merging: Non-Maximum Suppression

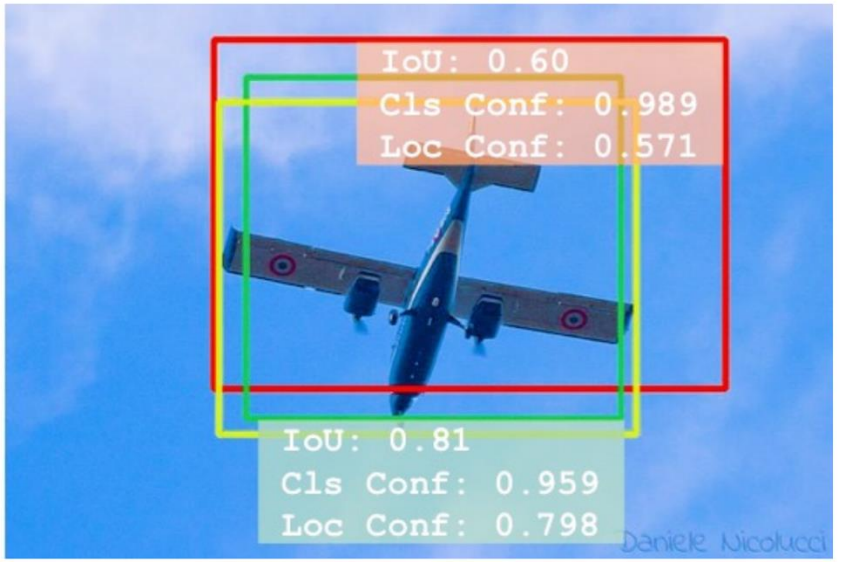


	Cls. Conf.
Box 1	0.85
Box 2	0.9
Box 3	0.96
Box 4	0.85
.....	

Keep this!

The Issue: Better Localized Ones may get Suppressed

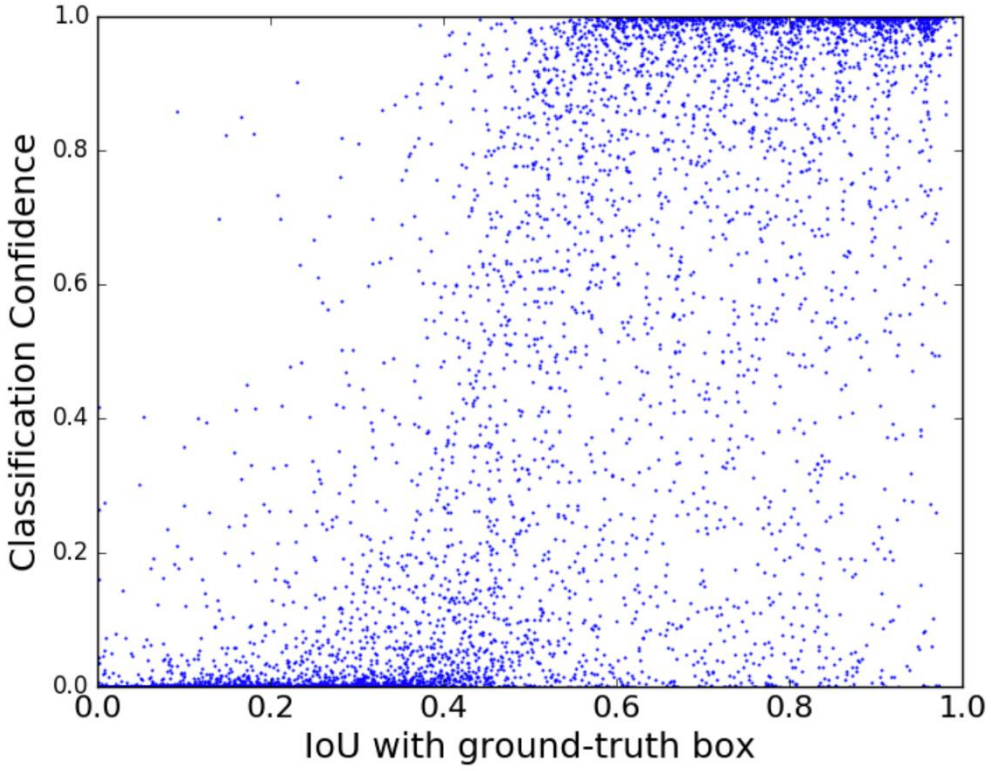




Core: The prediction target is misaligned with the evaluation metric

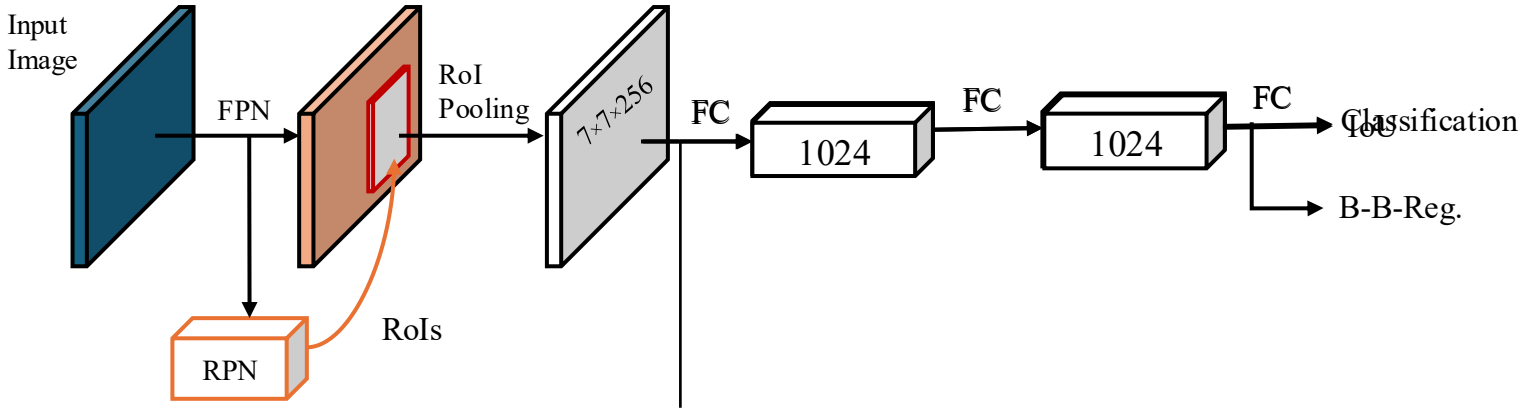
When we evaluate bounding boxes, we first check their IoU

The Issue: Classification Confidence != IoU



Classification Confidence *vs.* ***Localization Accuracy (IoU)***

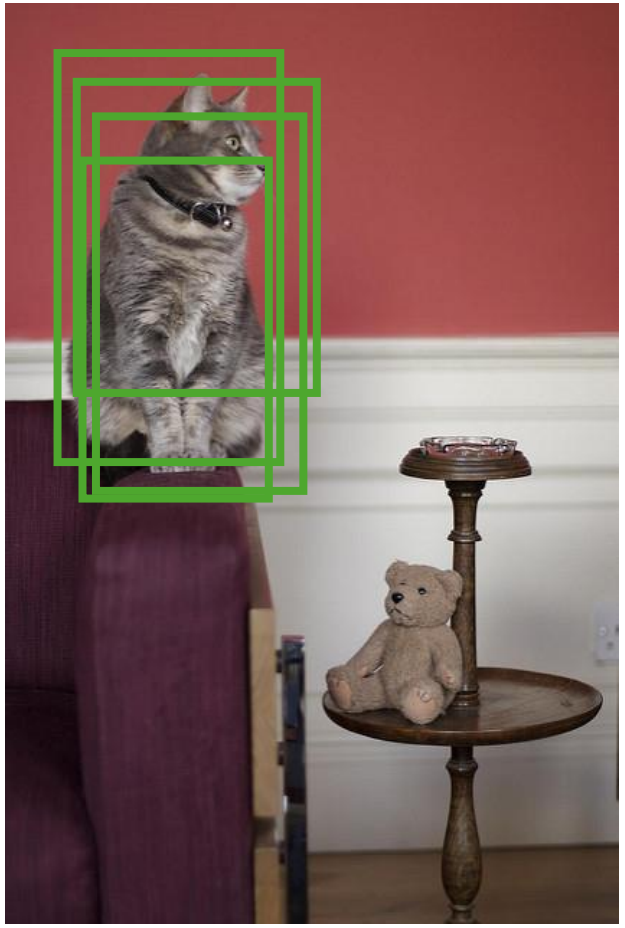
Solution: Predicting the IoU (Localization Confidence)



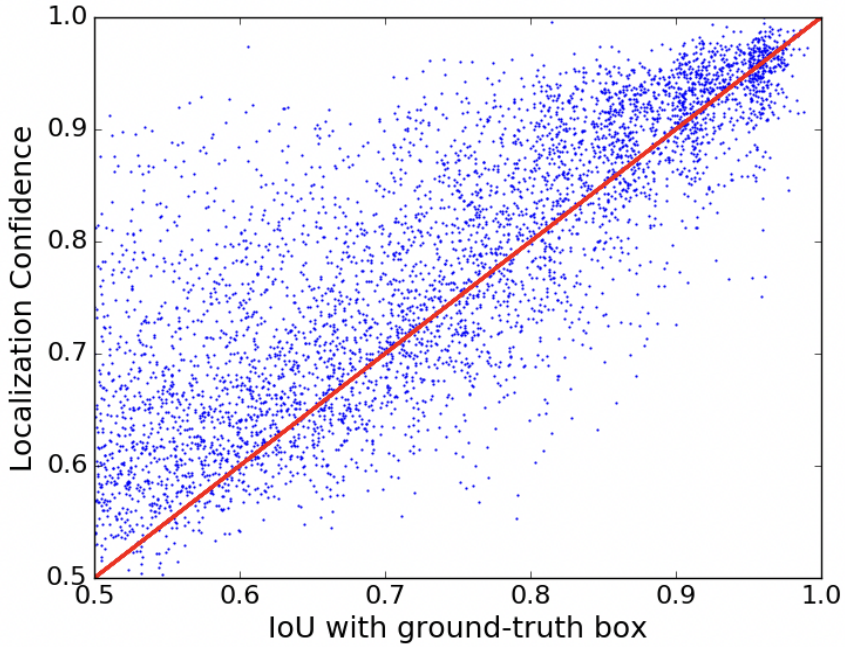
Learning to Predict IoU.

More precisely, predict the IoU with the best-matched groundtruth.

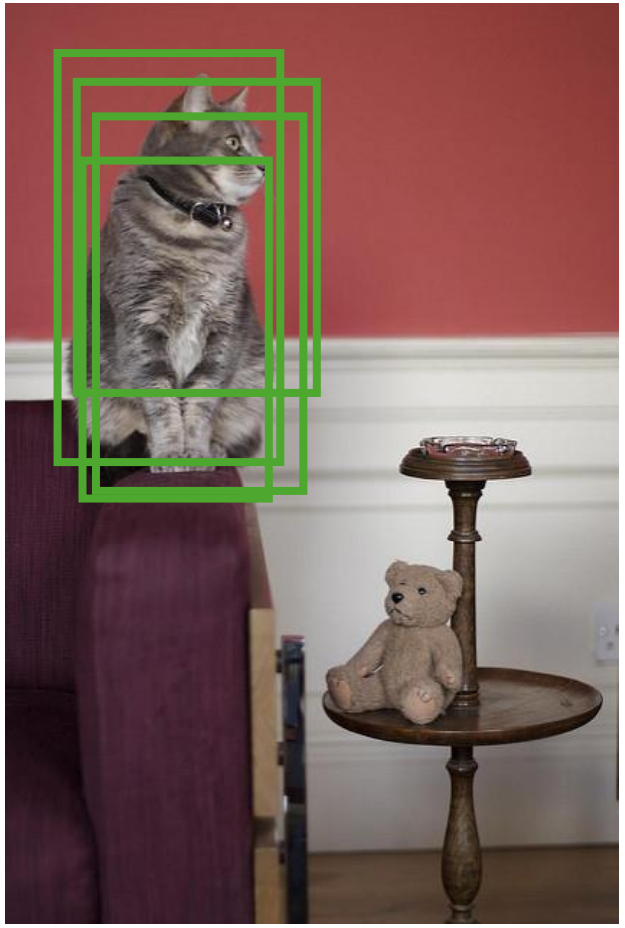
Improve NMS with Localization Confidence



	Cls. Conf.	Loc. Conf.
Box 1	0.85	0.95
Box 2	0.9	0.82
Box 3	0.96	0.74
Box 4	0.85	0.70
.....		



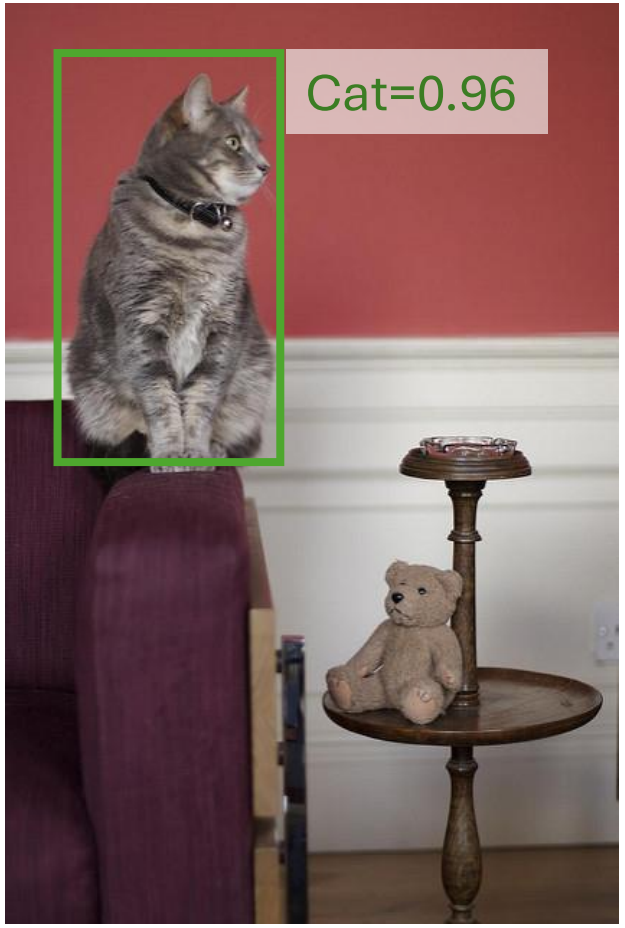
Improve NMS with Localization Confidence



	Cls. Conf.	Loc. Conf.
Box 1	0.85	**0.95**
Box 2	0.9	0.82
Box 3	0.96	0.74
Box 4	0.85	0.70
.....		

Best Localized!

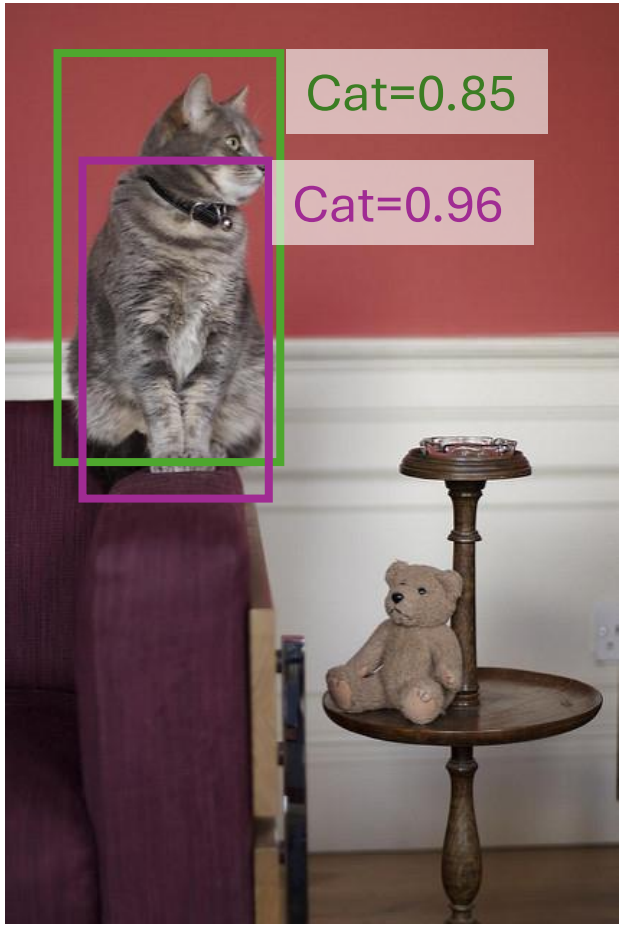
Improve NMS with Localization Confidence



	Cls. Conf.	Loc. Conf.
Box 1	0.85	0.95
Box 2	0.9	0.82
Box 3	**0.96**	0.74
Box 4	0.85	0.70
.....		

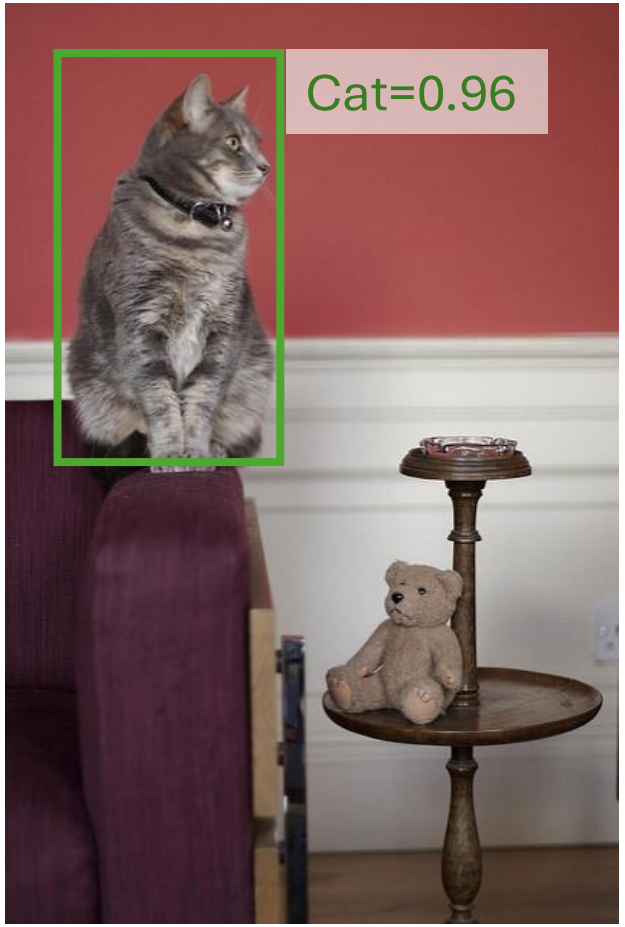
Most Confident Classification!

Improve NMS with Localization Confidence



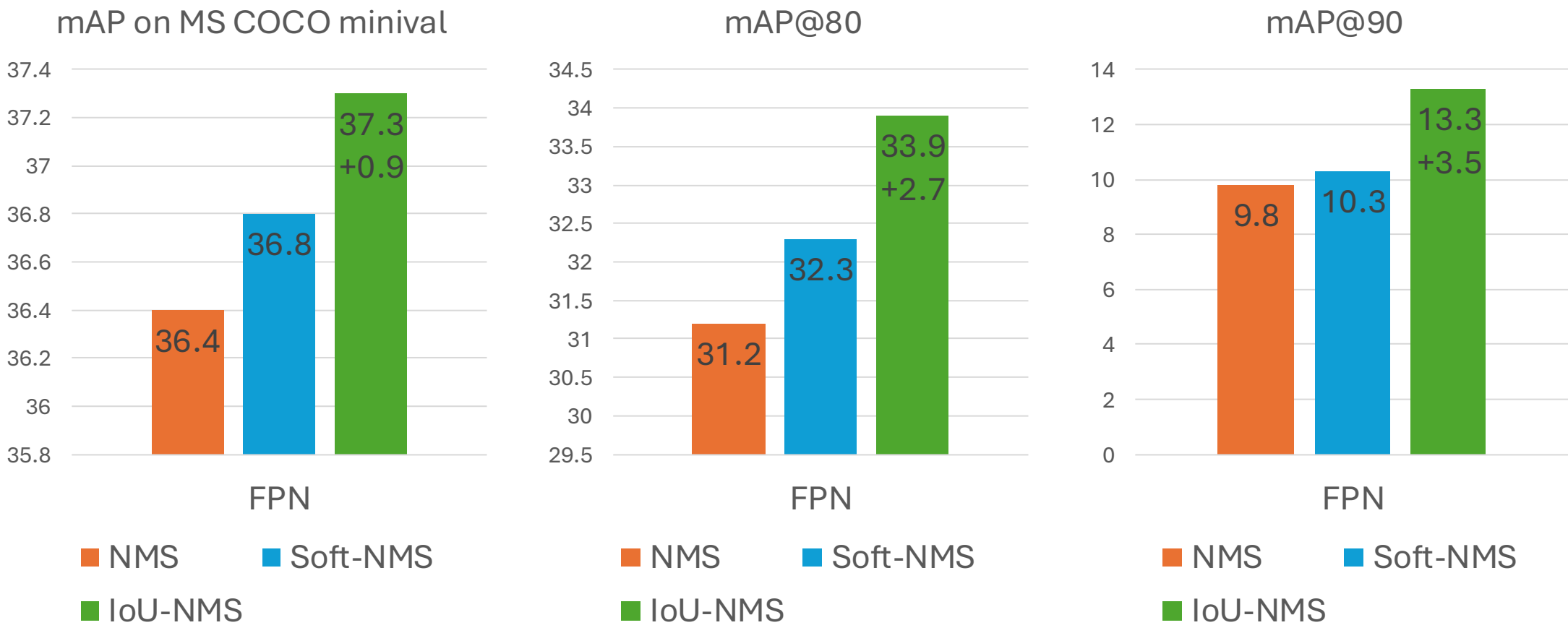
1. Sort all boxes by ***localization confidence*** in descending order.

Improve NMS with Localization Confidence



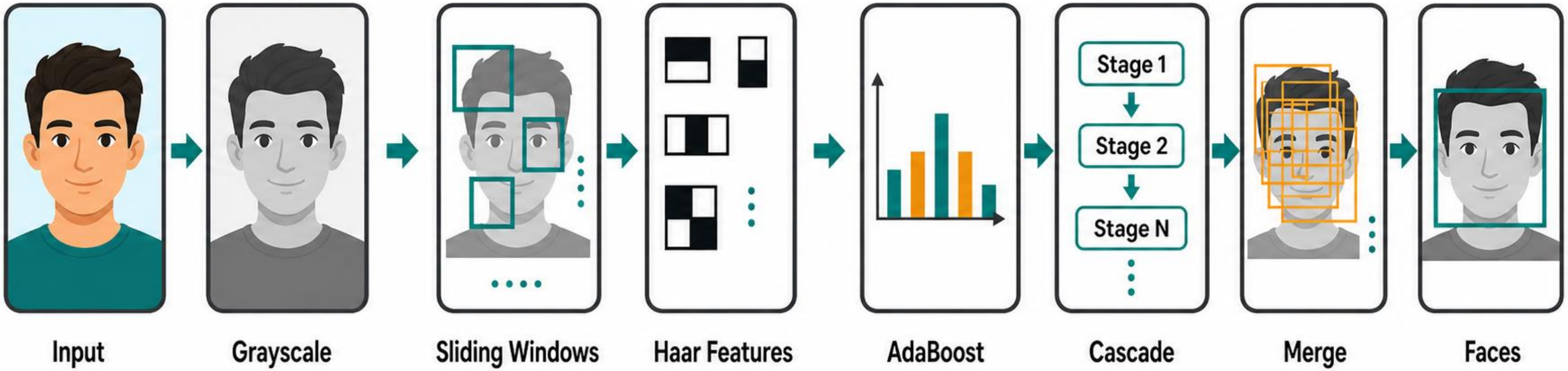
1. Sort all boxes by ***localization confidence*** in descending order.
2. If $IoU(\square, \square) > 0.5$, suppress the one with lower loc. confidence($\square$ in this case) , ***and update the cls. confidence for*** $\square$ **.**
3. Repeat 2 until there is no such ($\square$, $\square$).

Improve NMS with Localization Confidence



Principles for Image Recognition

Pattern matching ~= convolution filter
Sliding window ~= convolution
Multi-stage prediction can get fast and accurate prediction
Multi-scale (image pyramid) is important



Recurring Questions in this Class

- How to formulate a multi-modal AI problem
  - What's the inputs, outputs, objectives, and evaluation metrics?
- What are the principles for solving these problems
  - What's the principles, advantages, and disadvantages of the representation and the computation models?

Questions for Discussion

- Is depth prediction + camera pose estimation all you need for your application of interest? Explain

- What kind of issues do you see in the evaluation metrics for object detection (Average Precision)?

- If we want to detect objects in videos, what would be challenges?