Yalong Pi (Texas A&M Institute of Data Science)

Address: Office 221C, John R. Blocker Building

Email: piyalong@tamu.edu

- B.S., Mechanical Engineering, 2007-2011
- M.S., Civil Engineering, 2011-2013
- Ph.D., Architecture Engineering, 2017-2020

- Assistant Research Scientist, 2020-present
- Architect, 2016-2017
- Project manager, 2013-2016

Machine-Learning-for-Computer-Vision

Intro 1:00-1:20

Syllabus

- *Day 1: Classification Fundamentals and Convolutional Neural Network (CNN)*
- *Day 2: Data Augmentation, Evaluation, and Transfer-learning*
- *Day 3: Object Detection and Tracking*
- *Day 4: Segmentation and Autocoder*
- *Day 5: Generative Adversarial Networks (GANs) and Beyond*

Syllabus

- *Everyday (1:00-4:00 pm)*
  - *Quiz (40%)*
  - *Lecture*
  - *Lab*
- *Final project (60%) due: End of Day5*

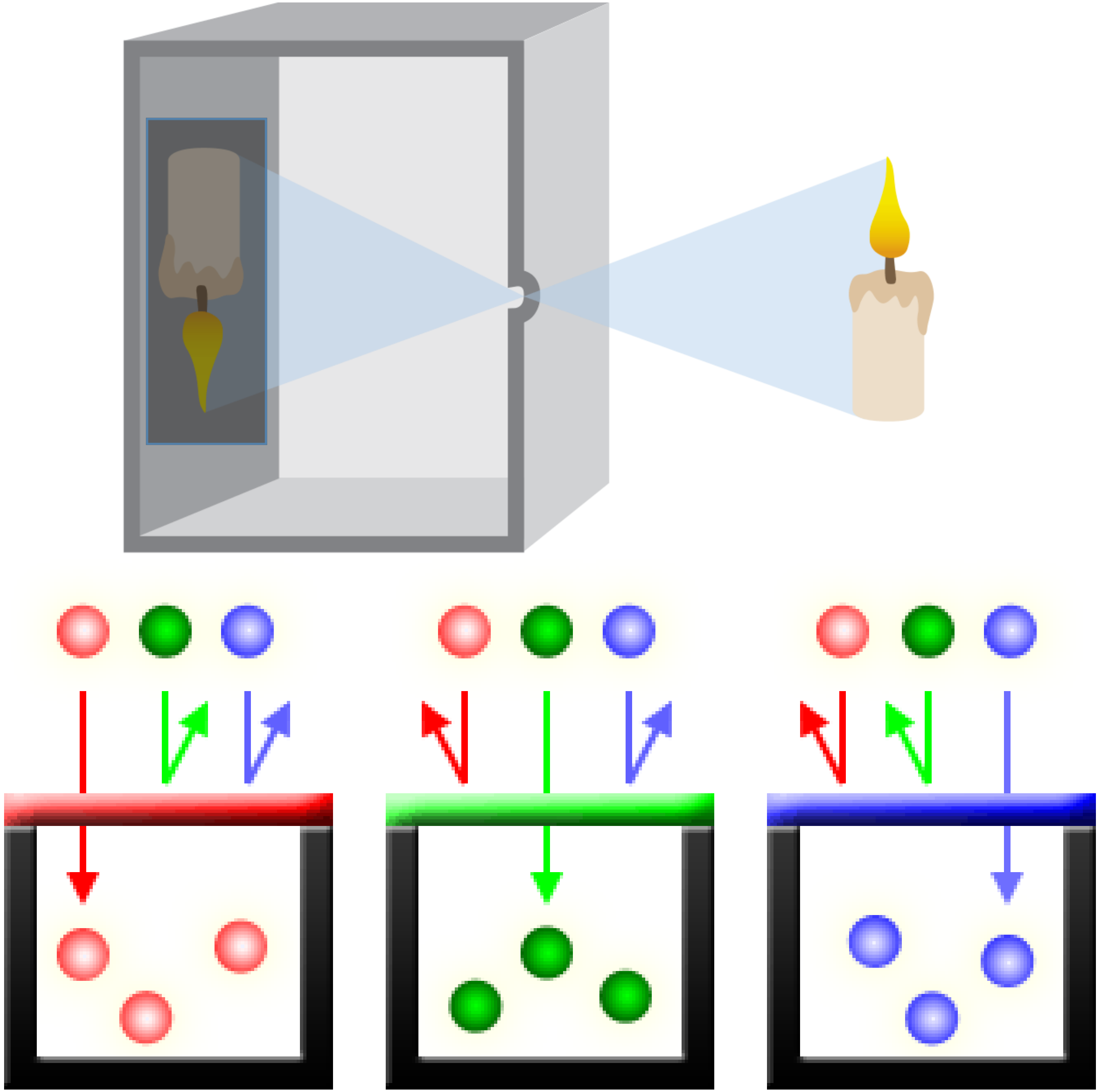
Course link

- https://github.com/TAMIDSpiyalong/Machine-Learning-for-Computer-Vision
- Canvas
- Howdy

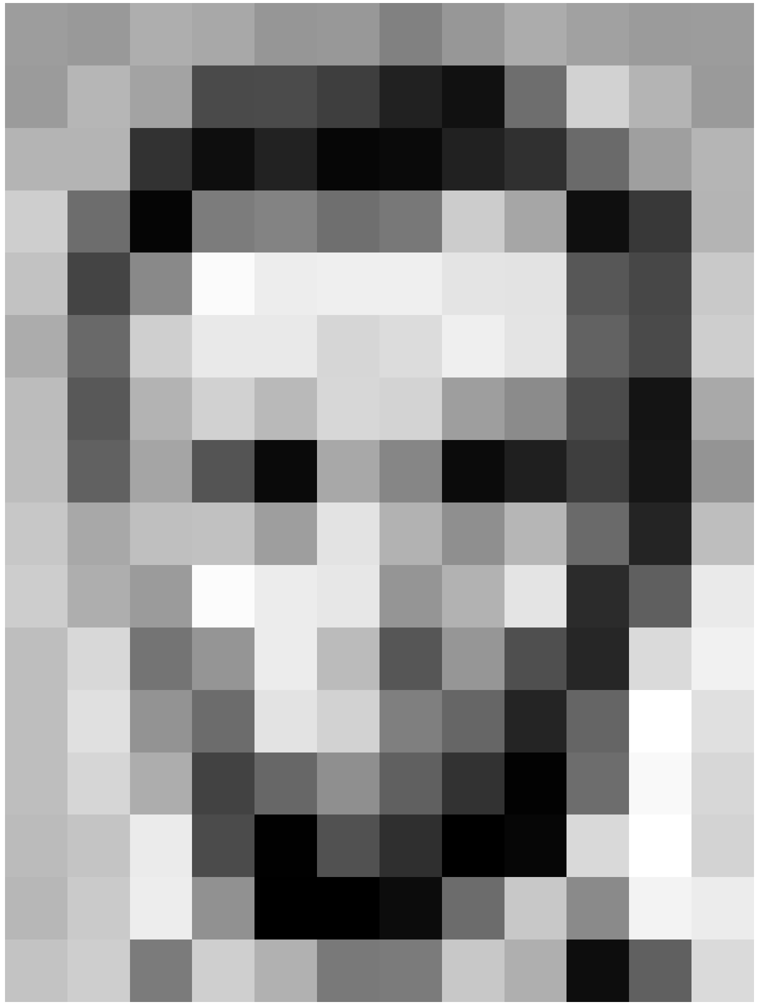
TensorFlow	Google Brain, 2015 (rewritten DistBelief)
Theano	University of Montréal, 2009
Keras	François Chollet, 2015 (now at Google)
Torch	Facebook AI Research, Twitter, Google DeepMind
Caffe	Berkeley Vision and Learning Center (BVLC), 2013


X000URJVYF
NEW
MADE IN CHINA

- Pinhole principle
- Traditional film
- Digital sensors (CCD and CMOS)
- Red Green Blue (RGB) channels



Grey Scale



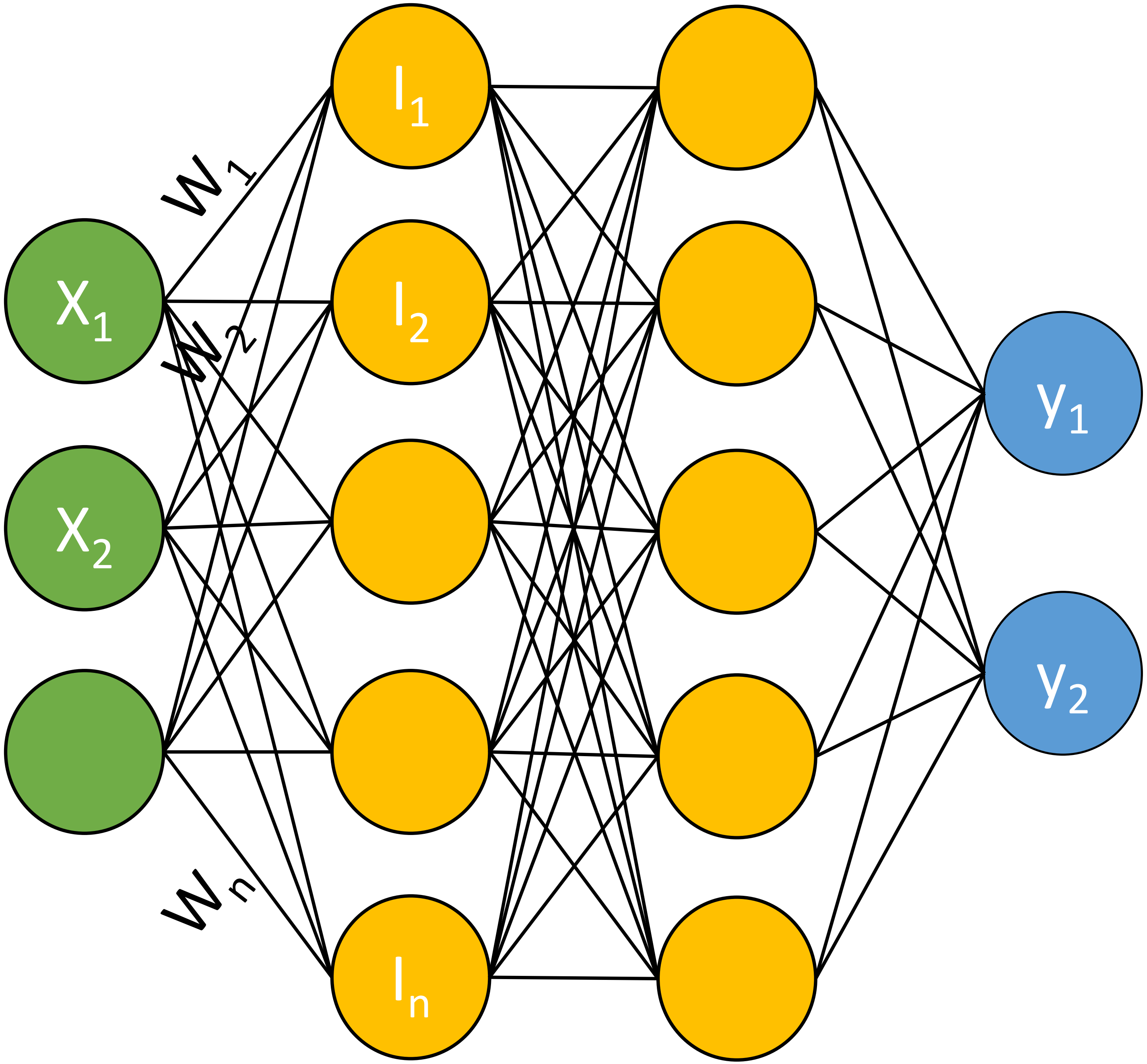
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

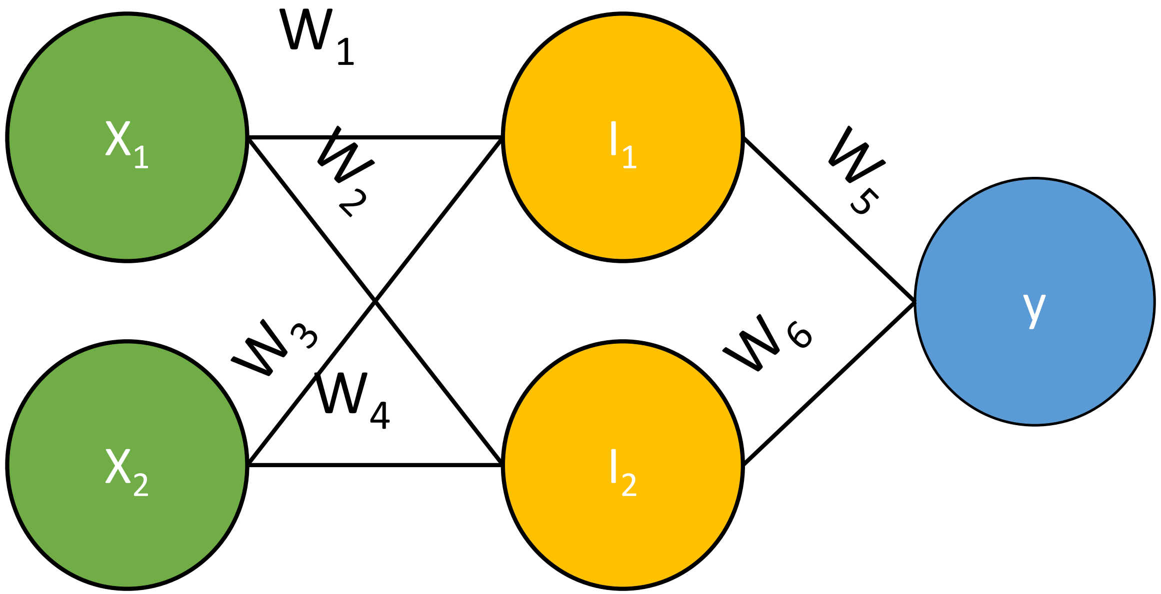
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

For each connection:

$$I_n = f(\sum_n X_n W_n + b)$$

- ❑ f is the activation function
- ❑ W_n is the weight
- ❑ b is the bias.
- ❑ A DNN has millions of weights and biases





$$y_{pred} = (X_1W_1 + X_2W_3)W_5 + (X_1W_2 + X_2W_4)W_6$$

$$Loss = 1/2(y_{pred} - y_{true})^2$$

$$W_n{}' = Wn - LR\ (\partial Loss/\partial W_n)$$

$$e.g.,\ W_6{}' = W_6 - LR\ (\partial Loss/\partial W_6)$$

- ❑Y_true are from the dataset labels
- ❑W_n are randomly initialized
- ❑Many types of loss function
- ❑Learning rate (LR) is very small (e.g., 0.0001)
- ❑Repeat in many epochs

Learning From Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2 = \frac{1}{n}\sum_{i=1}^{n}(y-(mx+b))^2$$

$$MSE = \frac{1}{2}((3 - (m(1)+b))^2 + (5 - (m(2)+b))^2)$$

$$\frac{\partial MSE}{\partial m} = 5m + 3b - 13 \qquad \frac{\partial MSE}{\partial b} = 3m + 2b - 8$$

$$\frac{\partial MSE}{\partial m} = -3 \qquad \frac{\partial MSE}{\partial b} = -1$$

$m = -1$

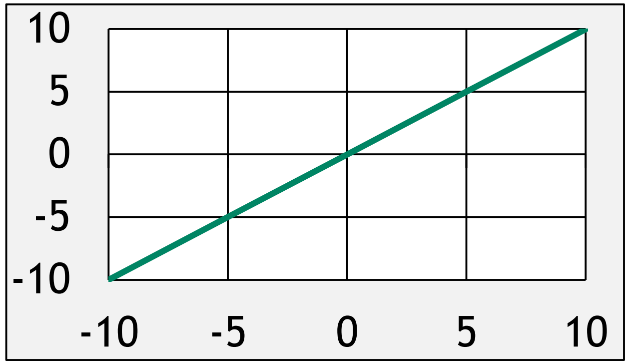
b = 5

ACTIVATION FUNCTIONS

Linear

$$\hat{y} = wx + b$$

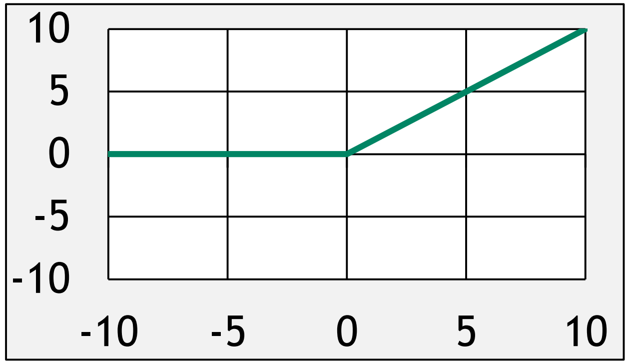
```
# Multiply each input
# with a weight (w) and
# add intercept (b)
y_hat = wx+b
```



ReLU

$$\hat{y} = \begin{cases} wx + b \; if \; wx + b > 0 \\ 0 \;\; otherwise \end{cases}$$

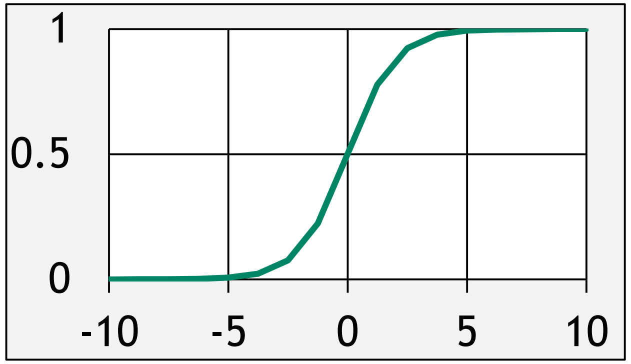
```
# Only return result
# if total is positive
linear = wx+b
y_hat = linear * (linear > 0)
```

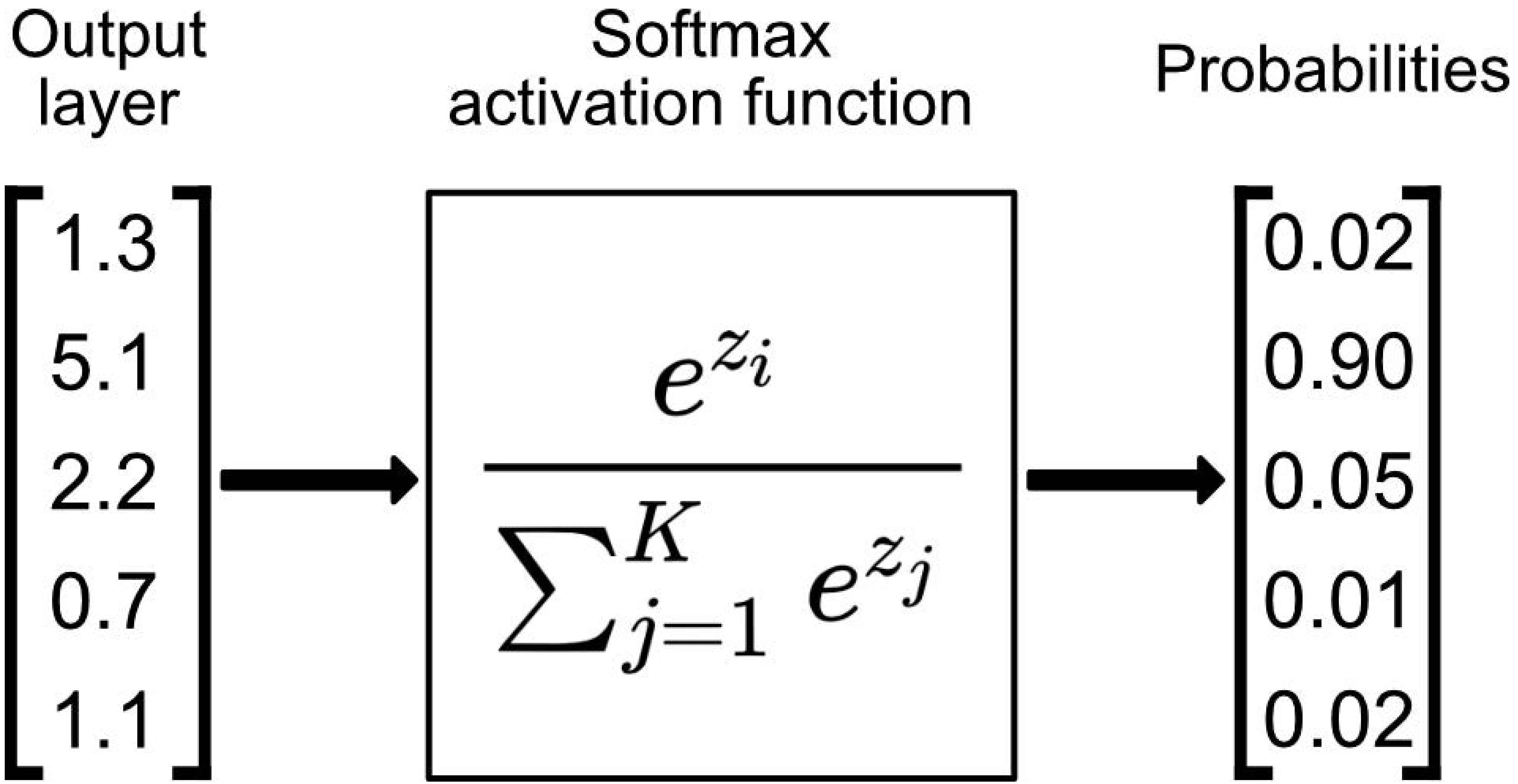


Sigmoid

$$\hat{y} = \frac{1}{1 + e^{-(wx+b)}}$$

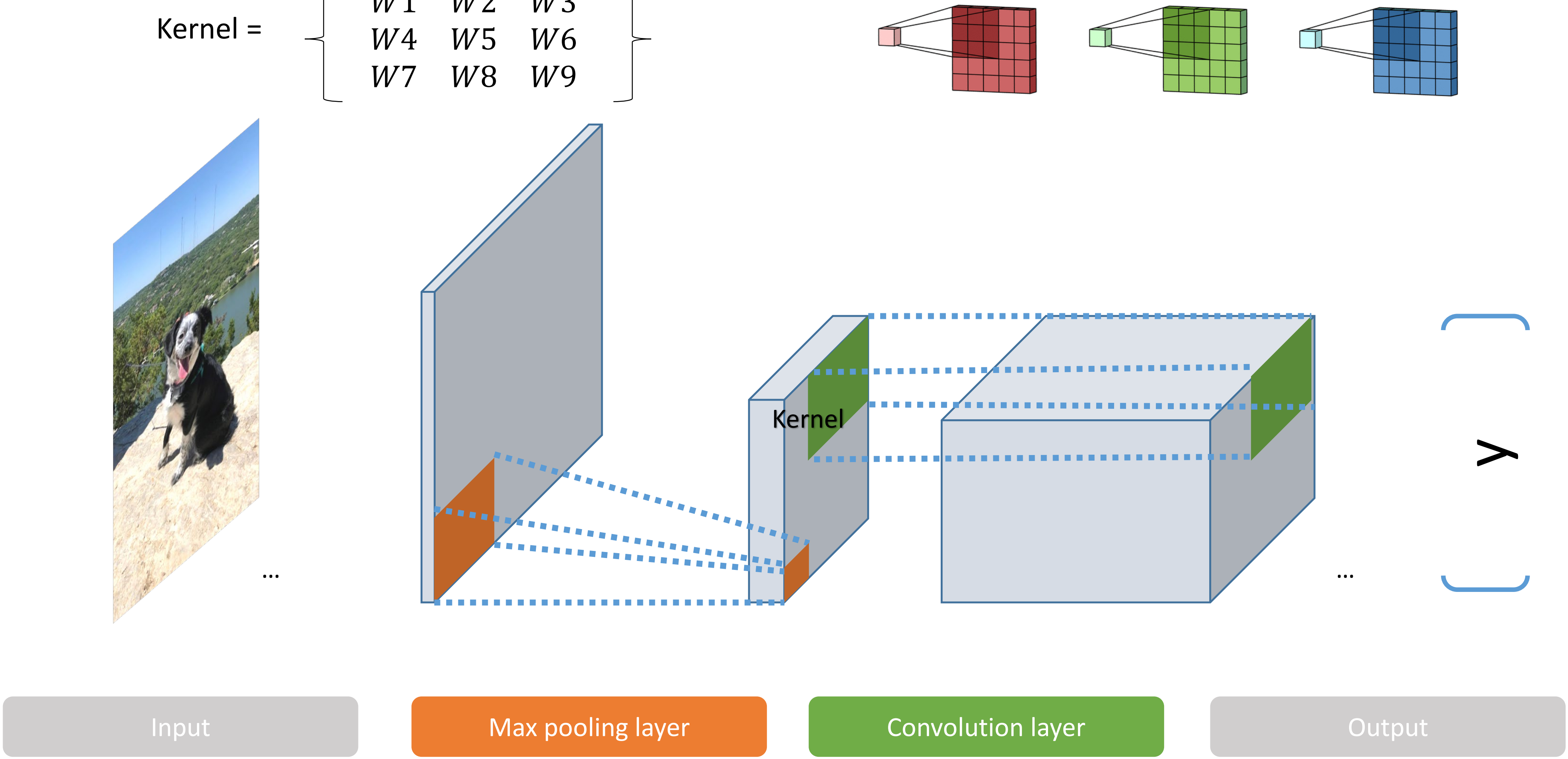
```
# Start with line
linear = wx + b
# Warp to - inf to 0
inf_to_zero = np.exp(-1 * linear)
# Squish to -1 to 1
y_hat = 1 / (1 + inf_to_zero)
```




Output layer
Softmax activation function
Probabilities
1.3
5.1
2.2
0.7
1.1
$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$
0.02
0.90
0.05
0.01
0.02

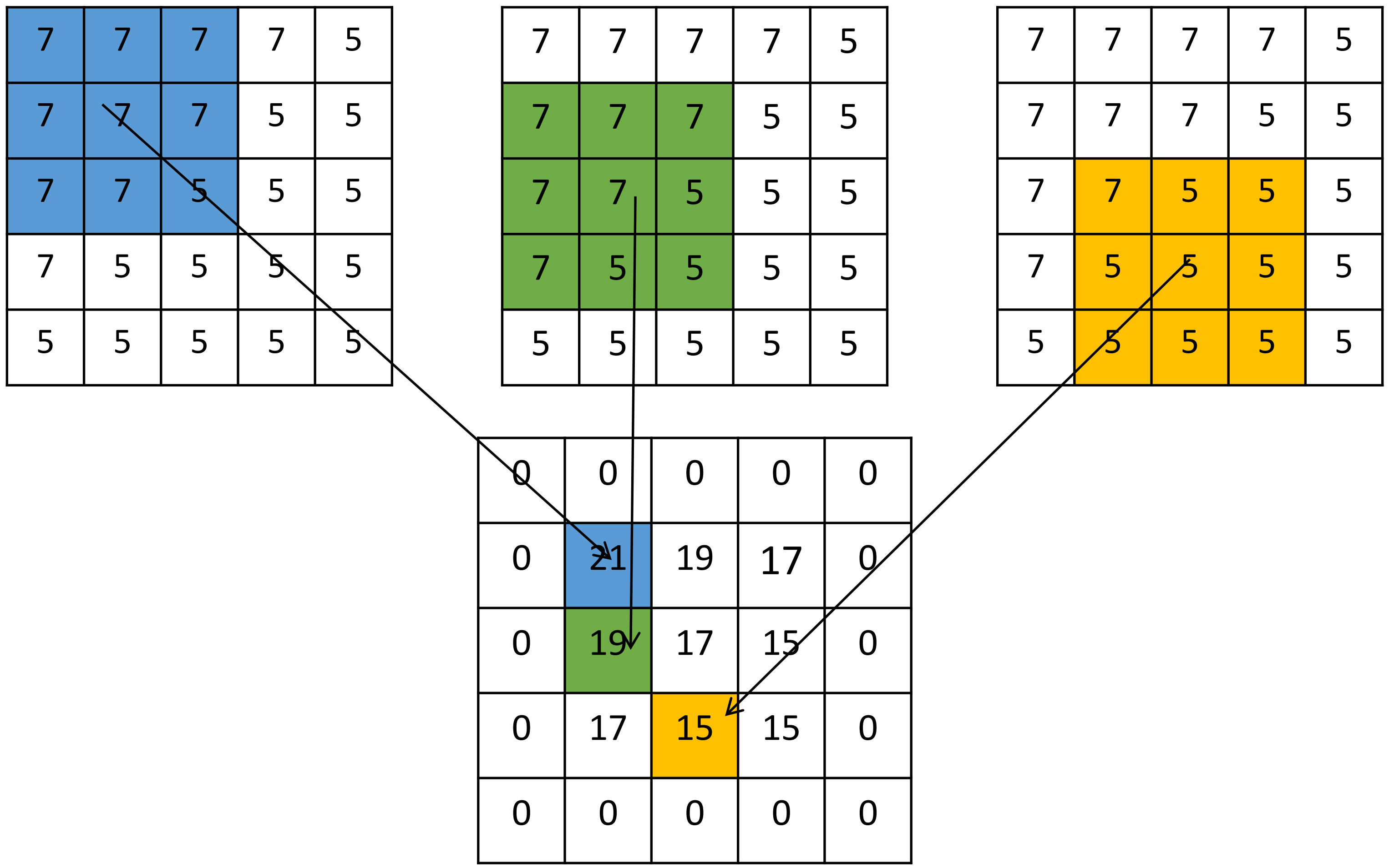
Convolutional Neural Network (CNN)

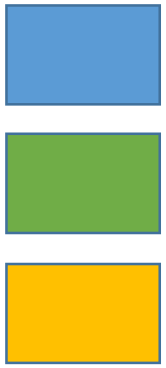
Kernel = $\begin{Bmatrix} W1 & W2 & W3 \\ W4 & W5 & W6 \\ W7 & W8 & W9 \end{Bmatrix}$



Convolution Computation

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$



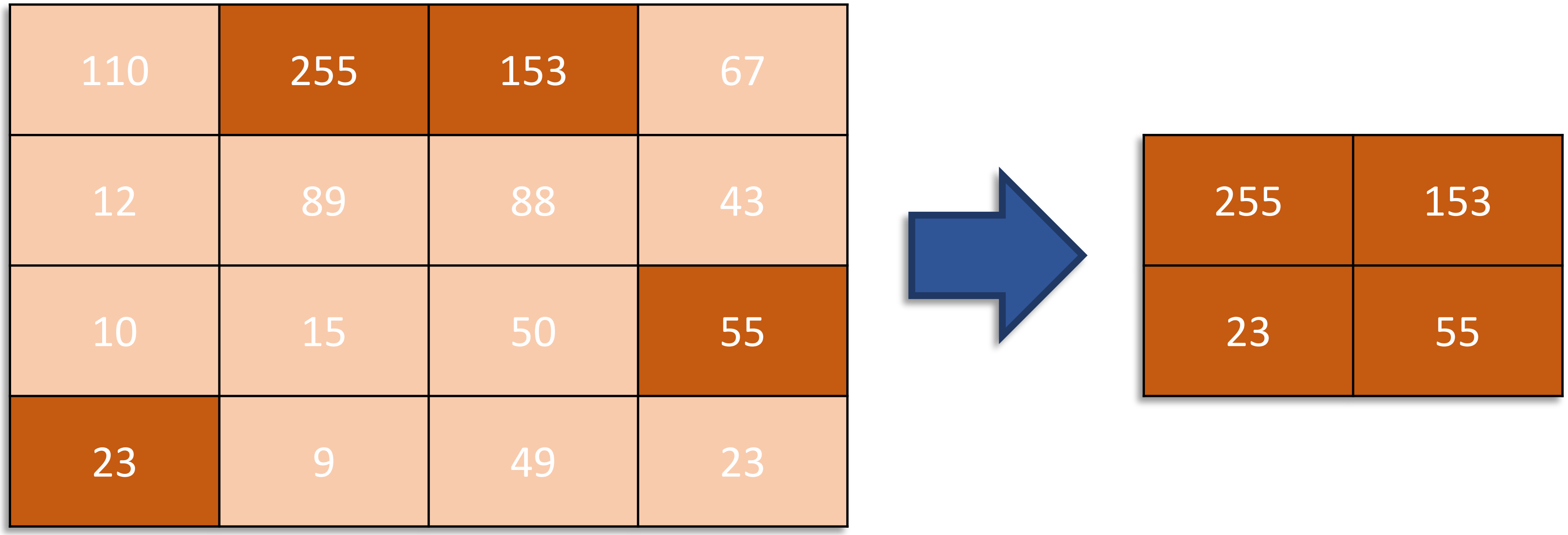


Kernel at position 1

Kernel at position 2

Kernel at position n

Max Pooling



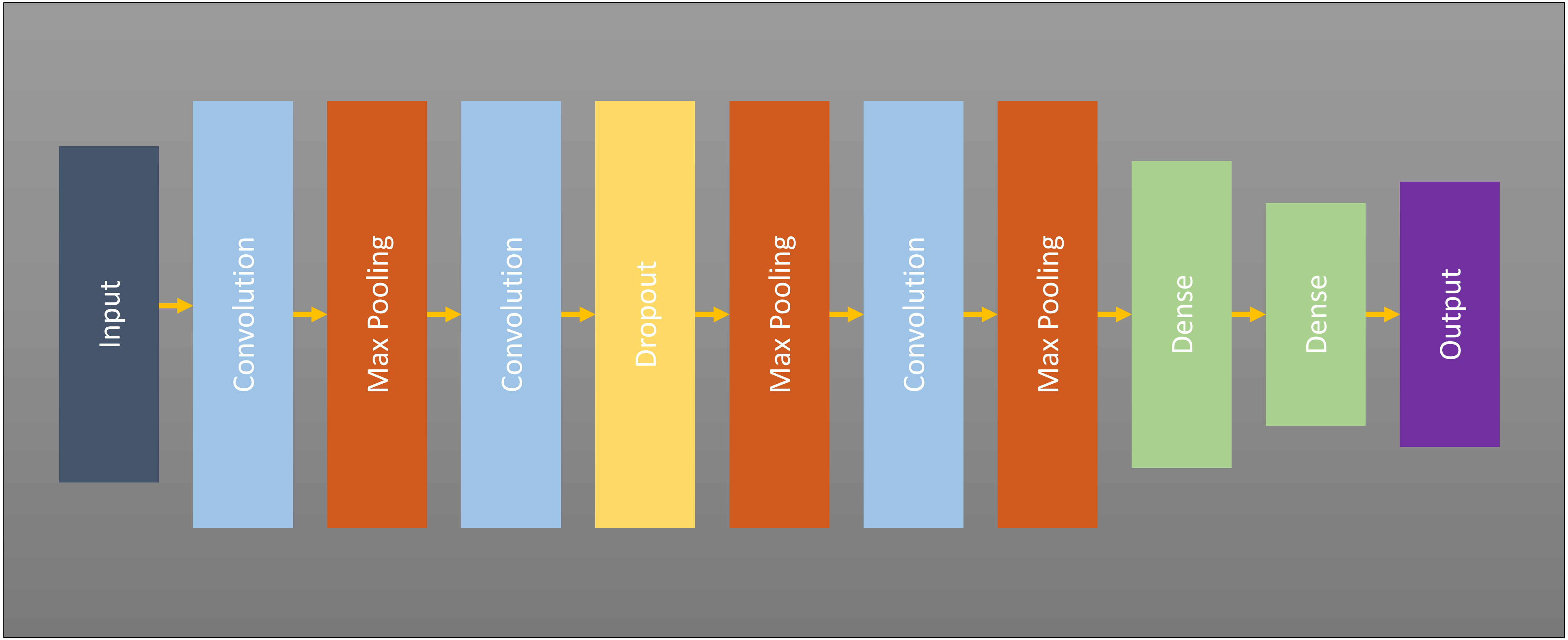
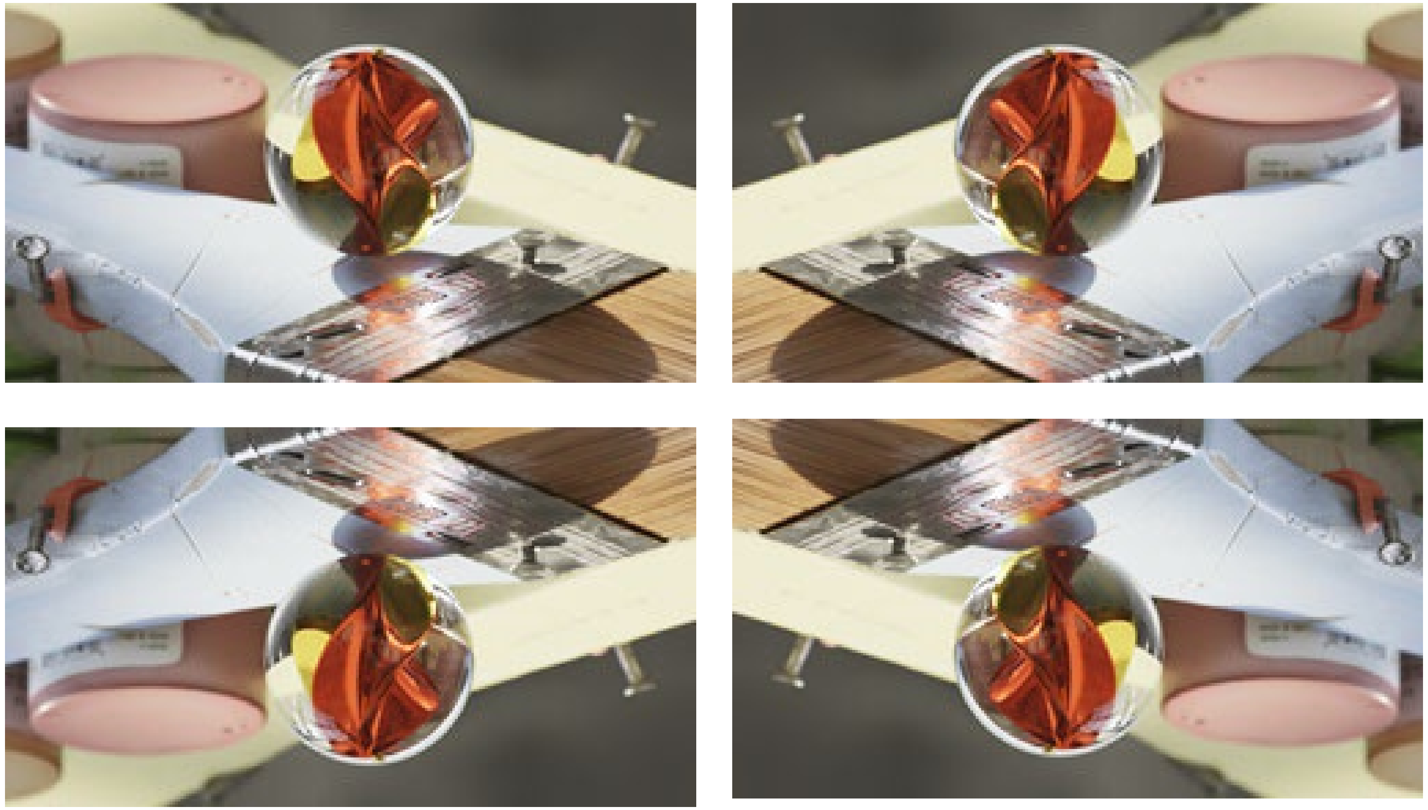

Input
Convolution
Max Pooling
Convolution
Dropout
Max Pooling
Convolution
Max Pooling
Dense
Dense
Output

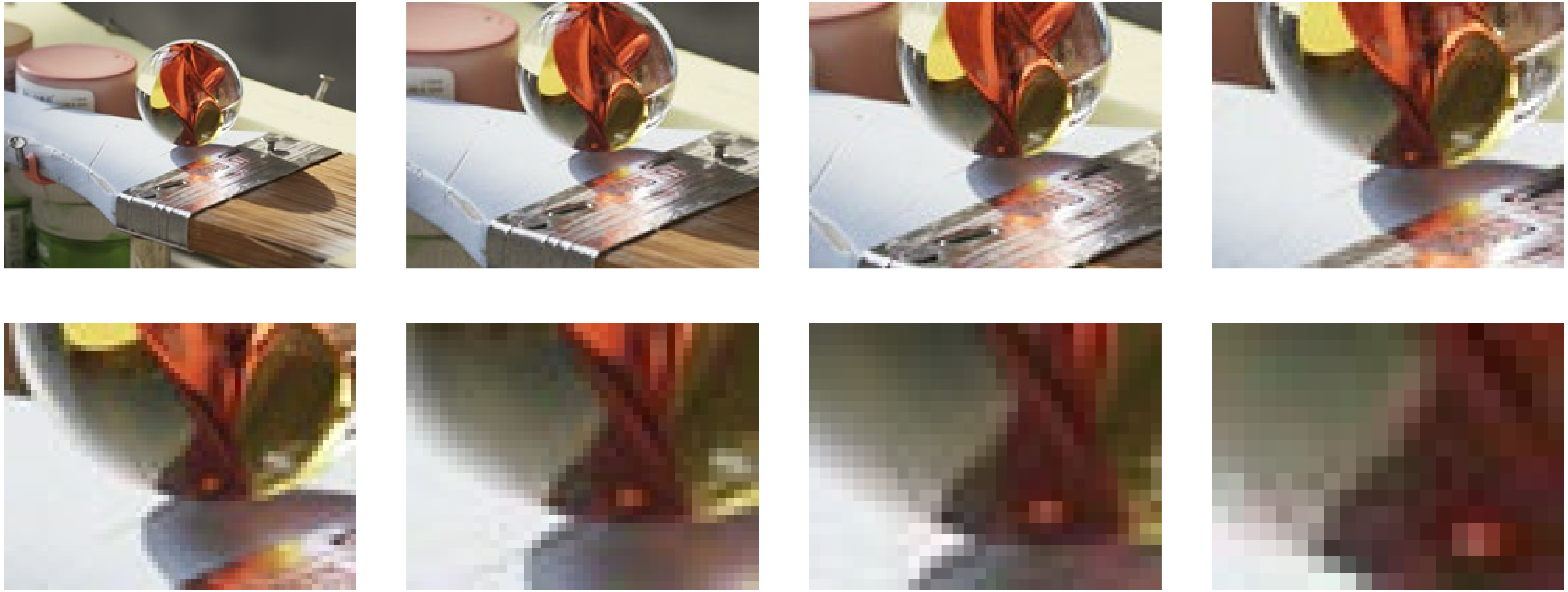
Image Flipping

Horizontal Flip



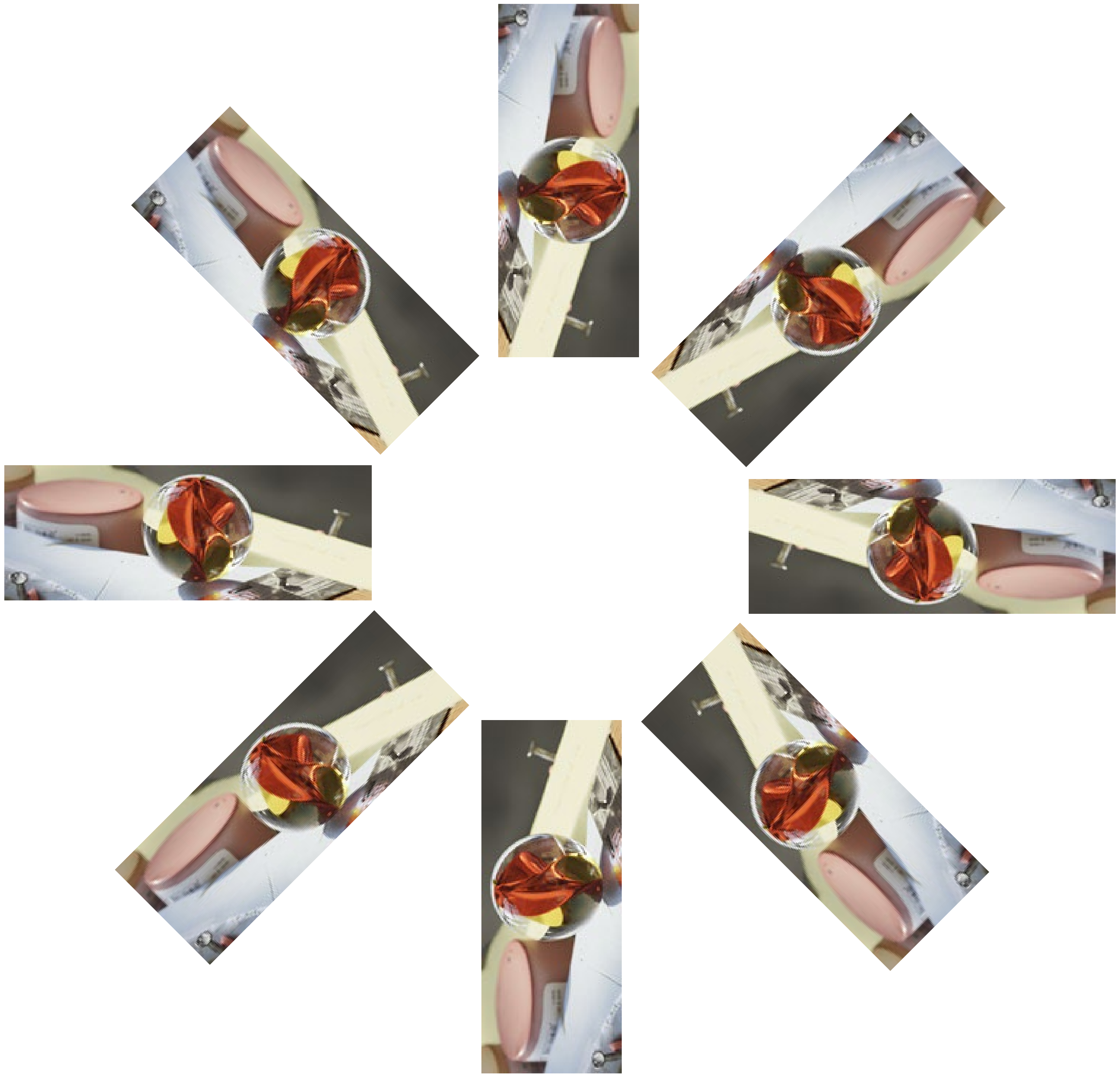
Vertical Flip

Zooming

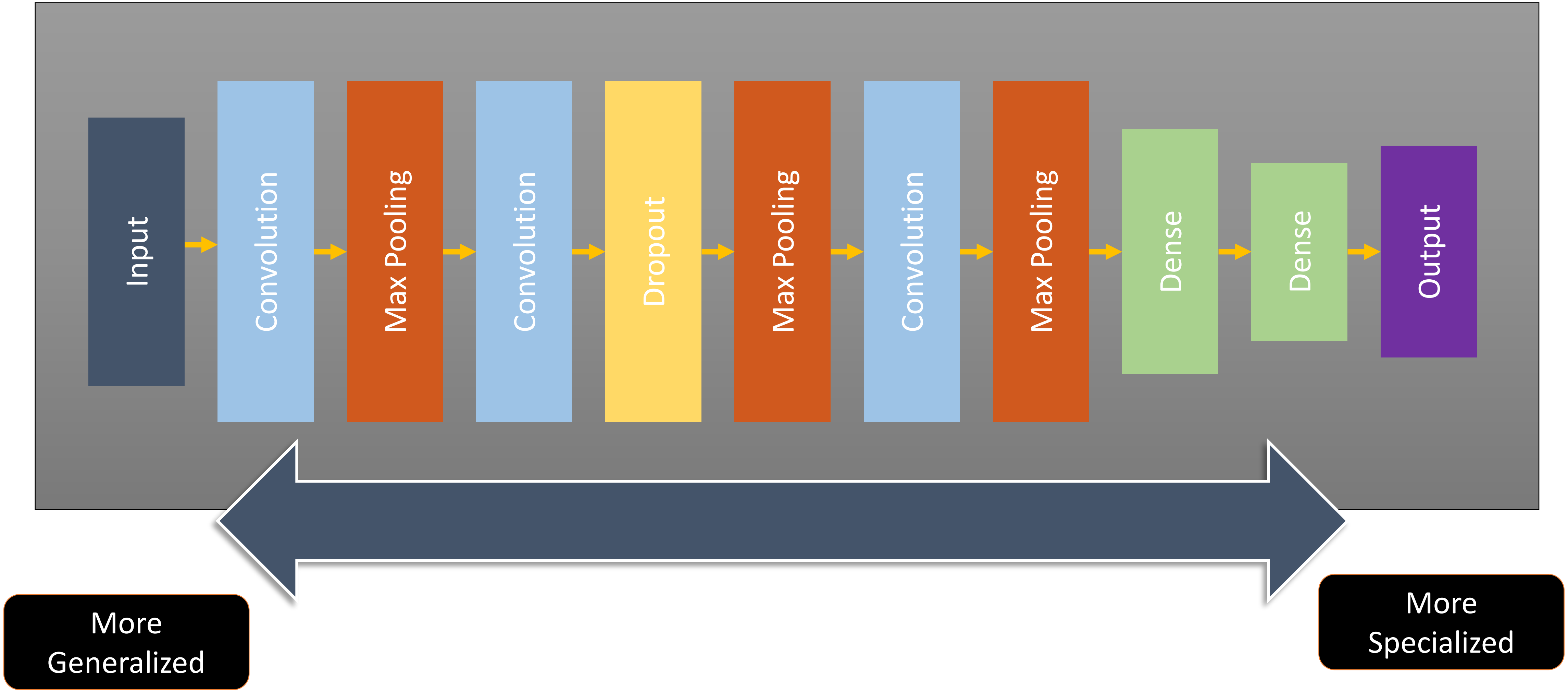


Brightness

Rotation

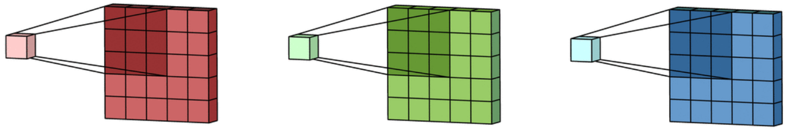


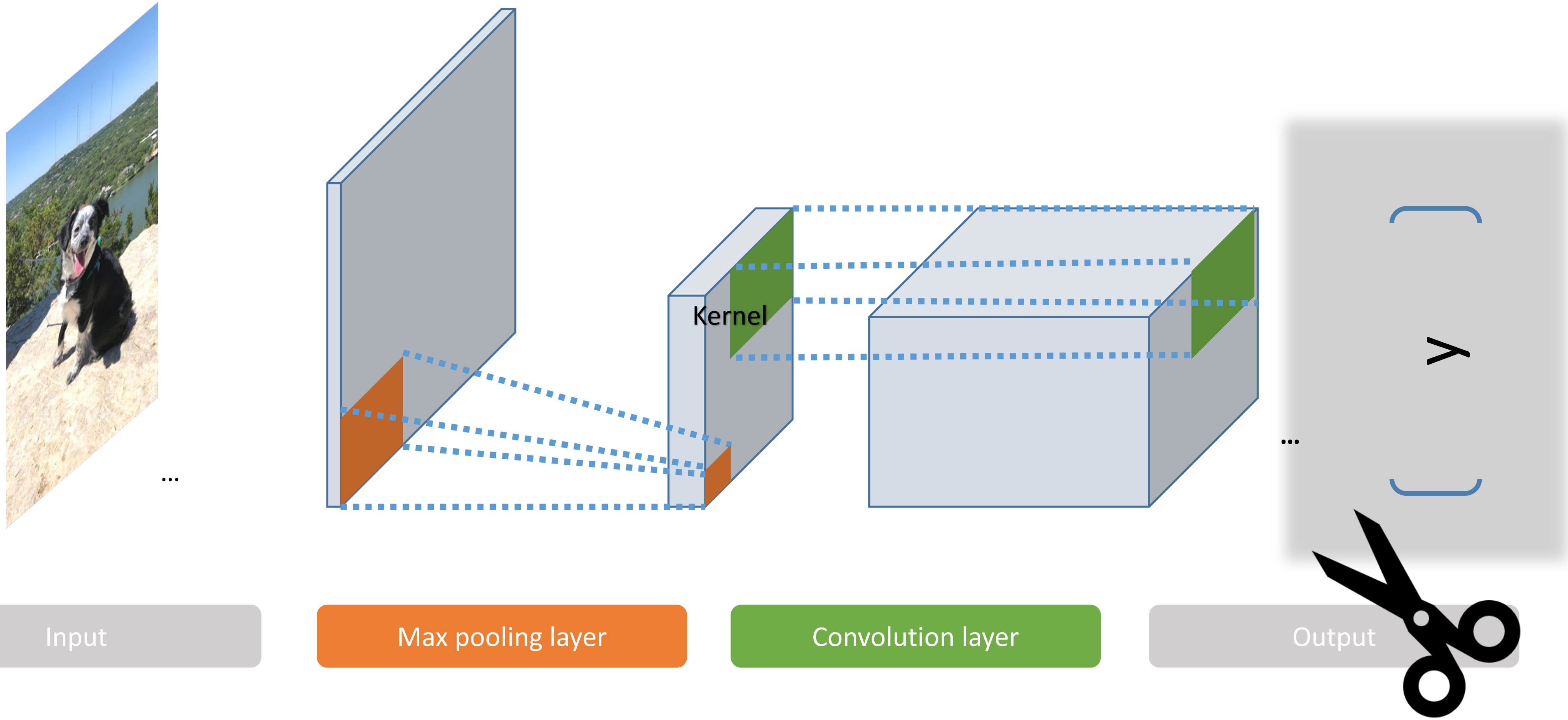
Transfer Learning



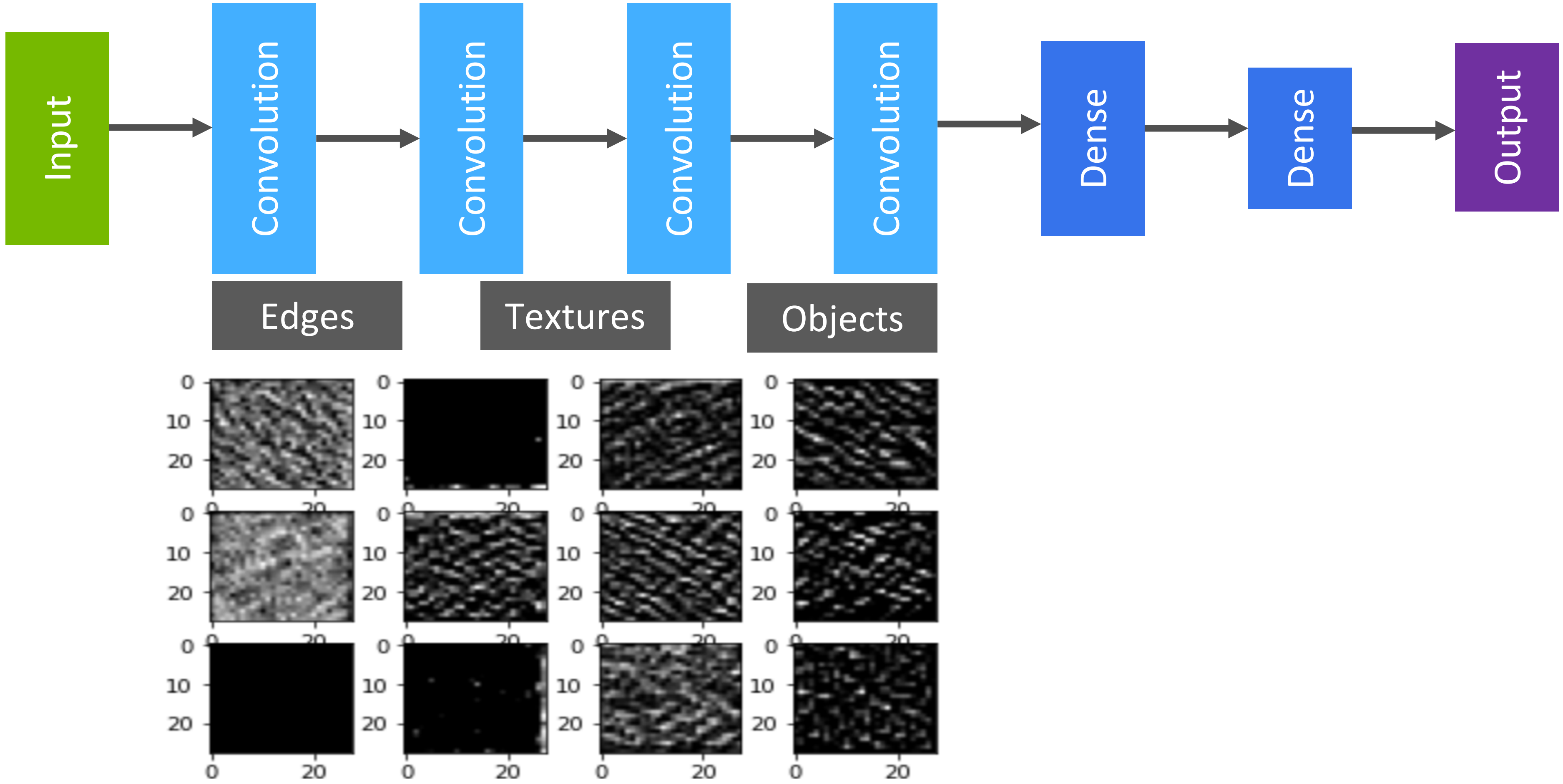
Transfer Learning

$$\text{Kernel} = \begin{bmatrix} W1 & W2 & W3 \\ W4 & W5 & W6 \\ W7 & W8 & W9 \end{bmatrix}$$





NEURAL NETWORK PERCEPTION



Pre-Trained Models

TensorFlow Hub

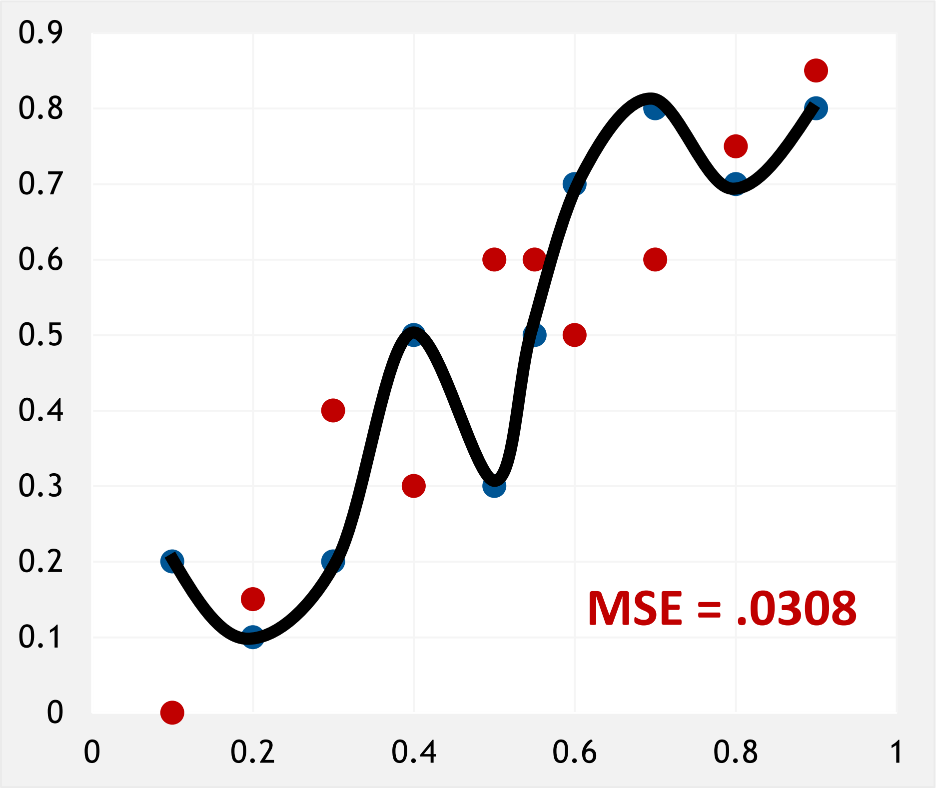
Keras

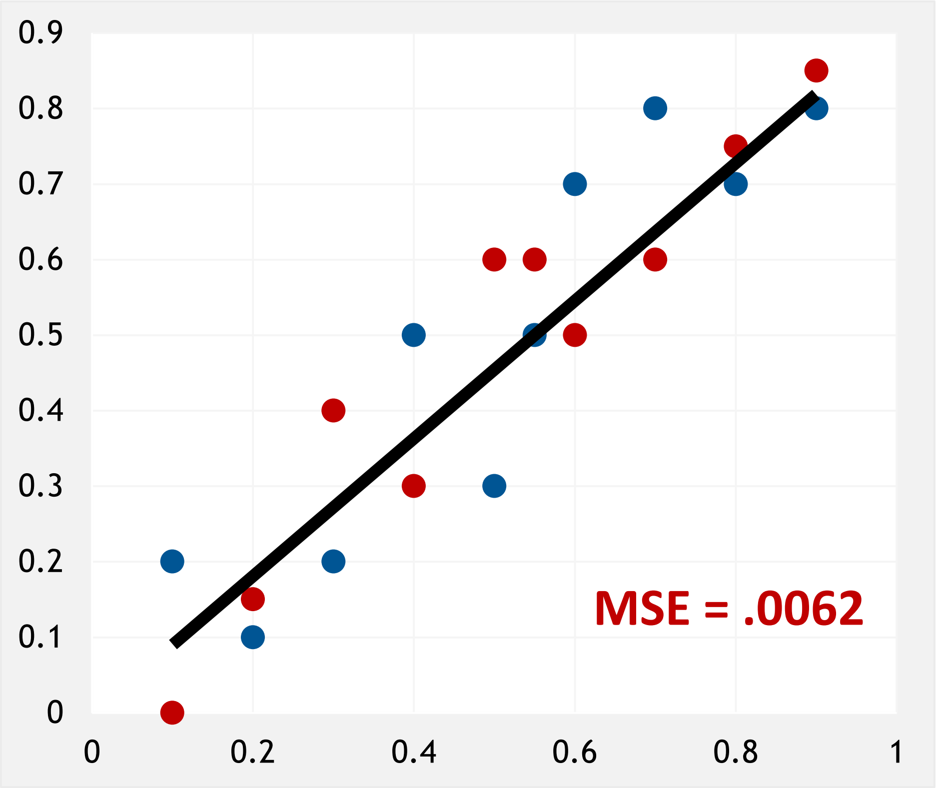
NVIDIA NGC

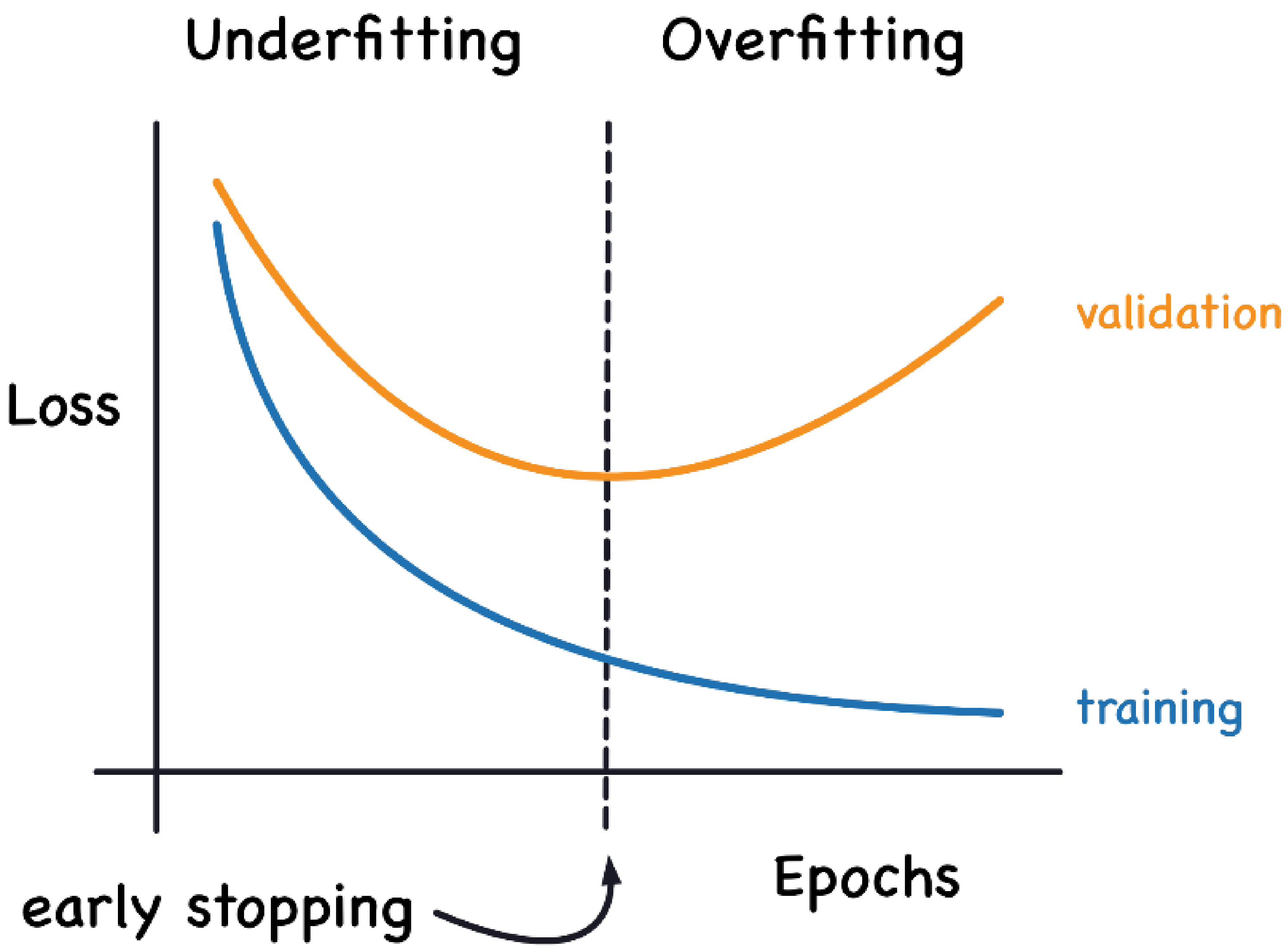
PYTORCH HUB

OVERFITTING

Which Trendline is Better?





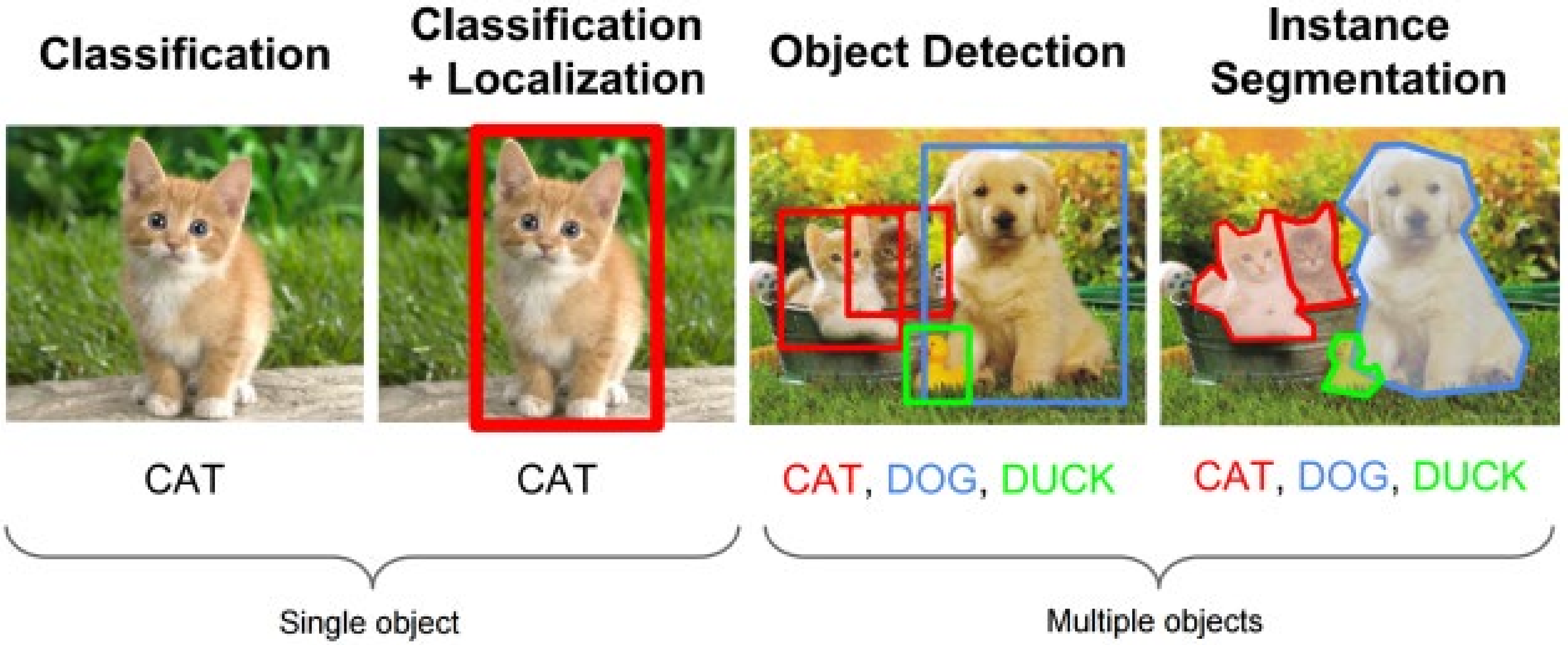

Underfitting
Overfitting
validation
Loss
training
Epochs
early stopping

Prediction\Ground Truth	Positive	Negative
Positive	TP	FP
Negative	FN	TN

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

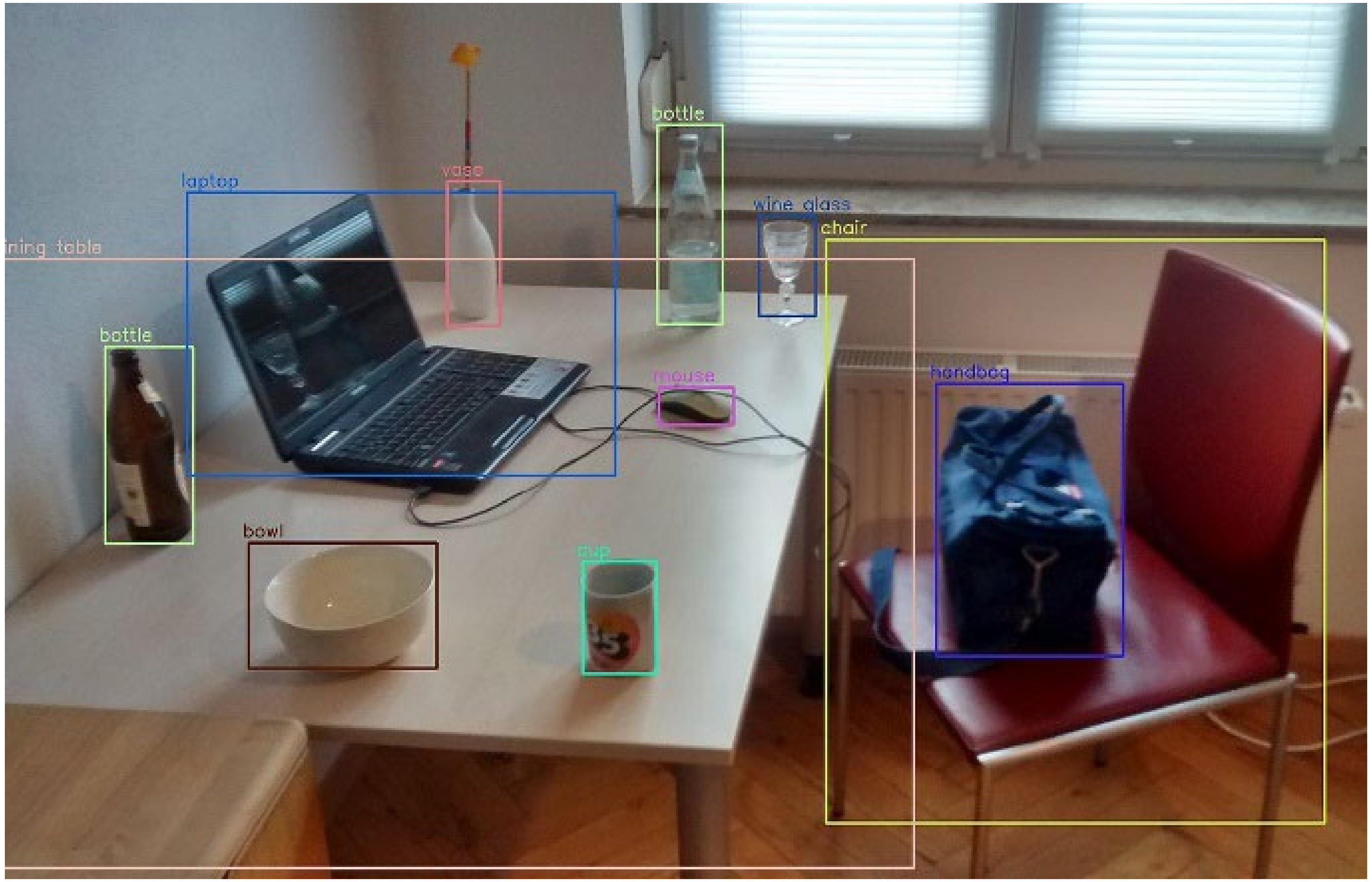
$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$


Classification
Classification + Localization
Object Detection
Instance Segmentation
CAT
CAT
CAT, DOG, DUCK
CAT, DOG, DUCK
Single object
Multiple objects

Dataset	Architecture
❑MNIST database (60,000 images) ❑ImageNet (1.2 M images for 1000 classes) ❑…	❑VGG 16 (19) series ❑ResNet (50, 101, 150) ❑…

Dataset	Architecture
❑ COCO (300K image in 80 classes) ❑ Open Images (9M images in 600 classes) ❑ …	❑ YOLO series ❑ RCNN series ❑ Retina Net ❑ …



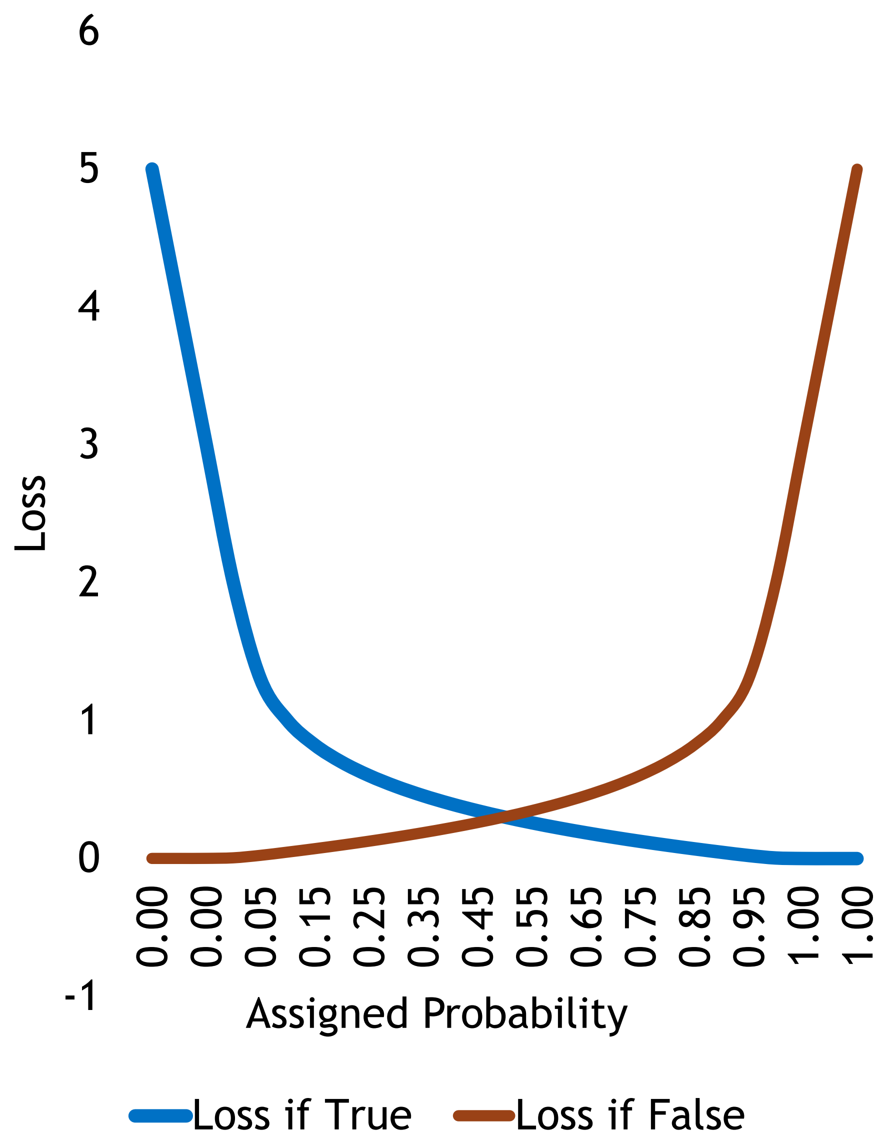
CROSS ENTROPY

```python
def cross_entropy(y_hat, y_actual):
    """Infinite error for misplaced confidence."""
    loss = log(y_hat) if y_actual else log(1-y_hat)
    return -1*loss
```

$$Loss = -\Big(\big(t(x) \cdot \log(p(x)) + \big(1 - t(x)\big) \cdot \log(1 - p(x)\big)\Big)$$

$$t(x) = target\ (0\ if\ False, 1\ if\ True)$$

$$p(x) = probability\ prediction\ of\ point\ x$$

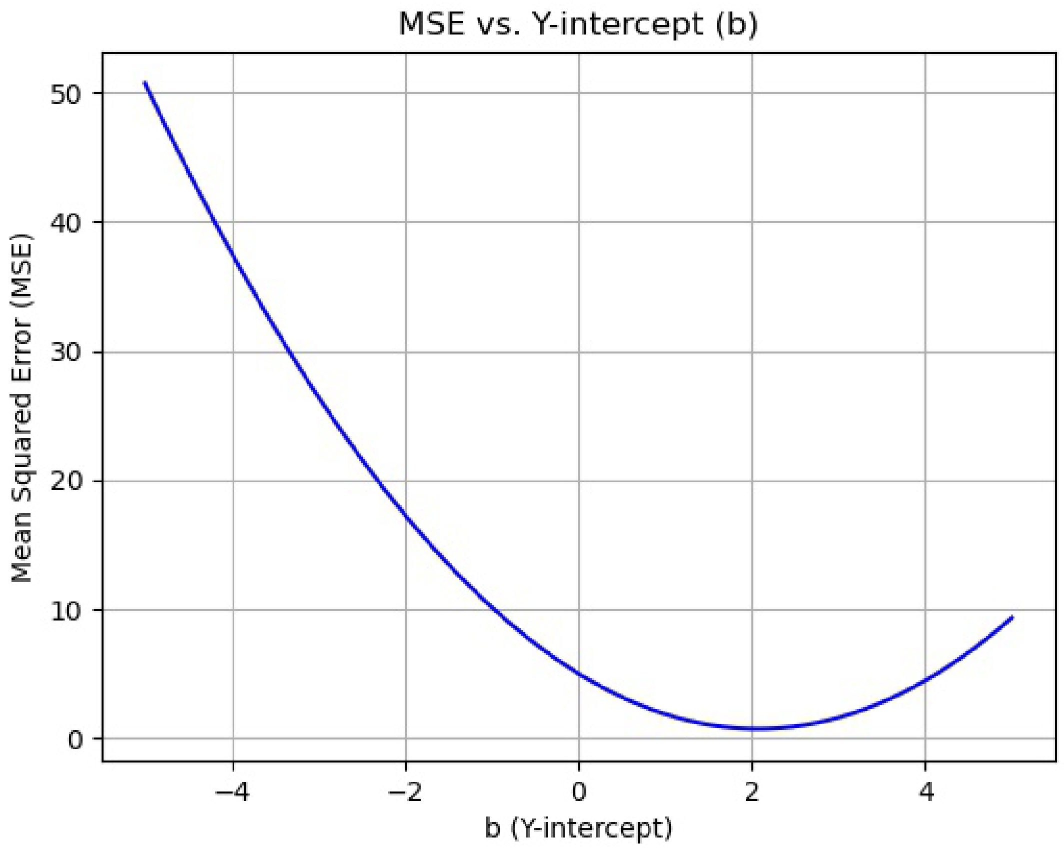


$$For\ y = ax + b$$

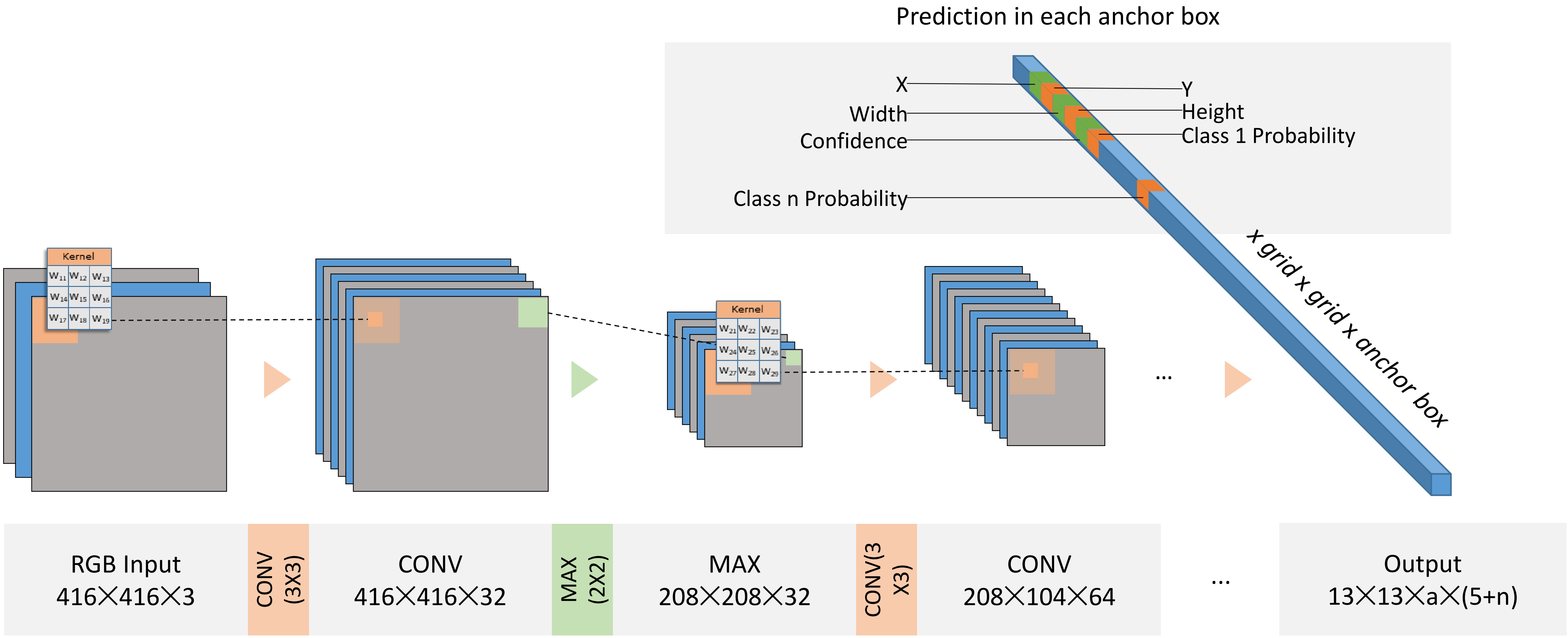
```python
# Calculate MSE for different "b" values
b_values = np.linspace(-5, 5, 100)
mse_values = []

for b in b_values:
    predicted_y = true_slope * x + b
    mse = np.mean((y - predicted_y)**2)
    mse_values.append(mse)
```

$$MSE = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} (y_{\mathrm{i}} - y_{\mathrm{pred}})^2$$

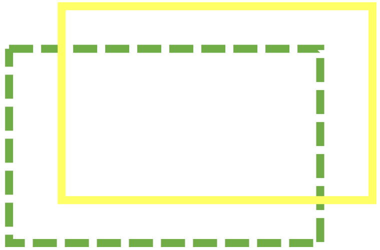


You Only Look Once (YOLO) Architecture

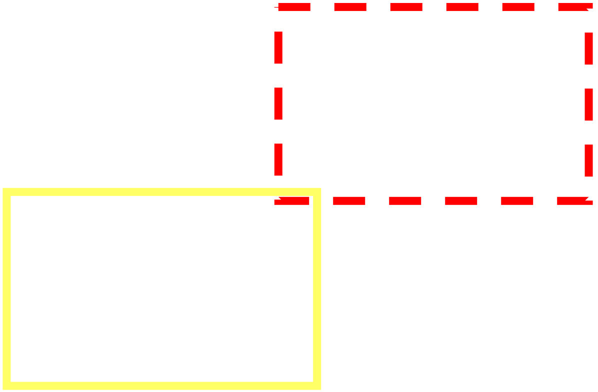


Performance metrics

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



True **P**ositive (**TP**): IoU ≥ 50%



False **P**ositive (**FP**): IoU < 50%

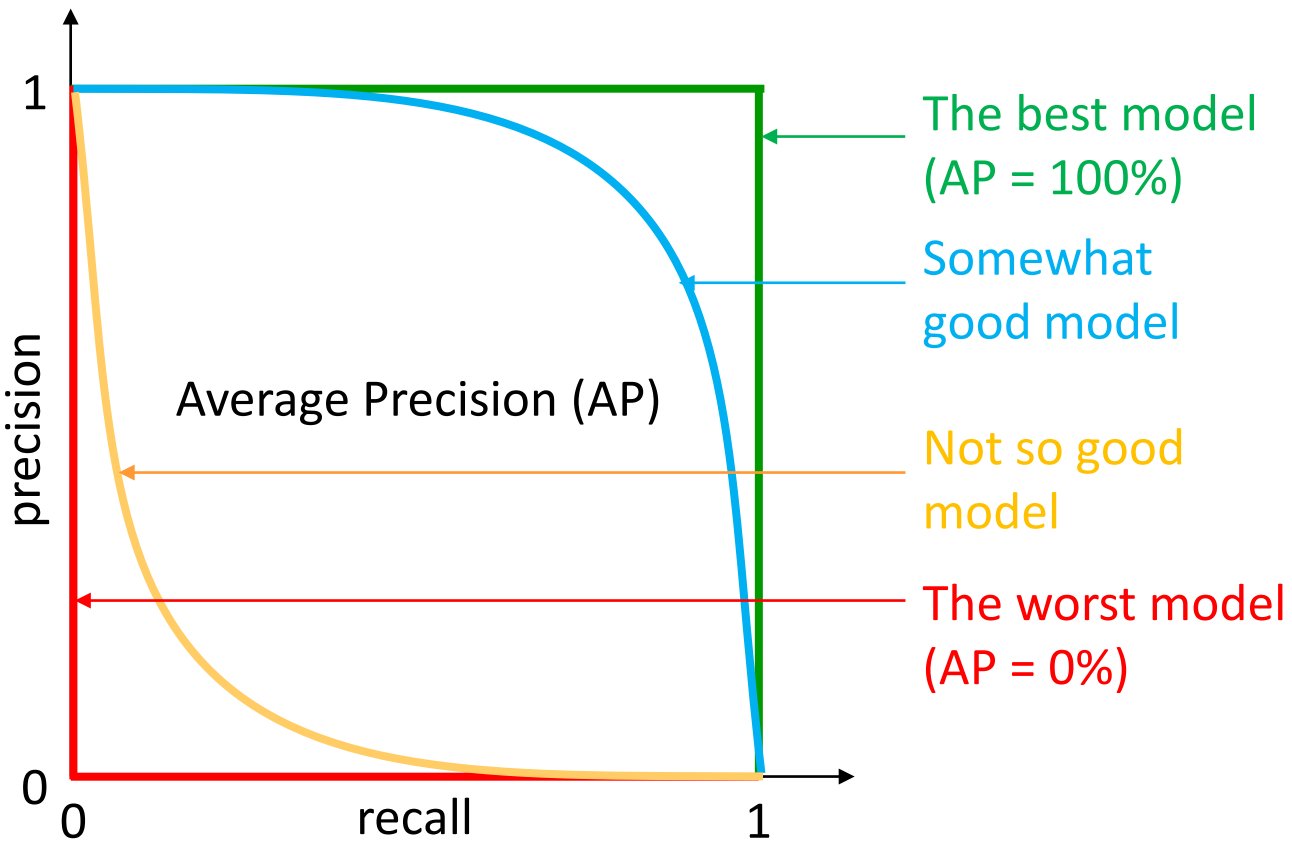
No True **N**egative (**TN**)

False **N**egative (**FN**): IoU = 0

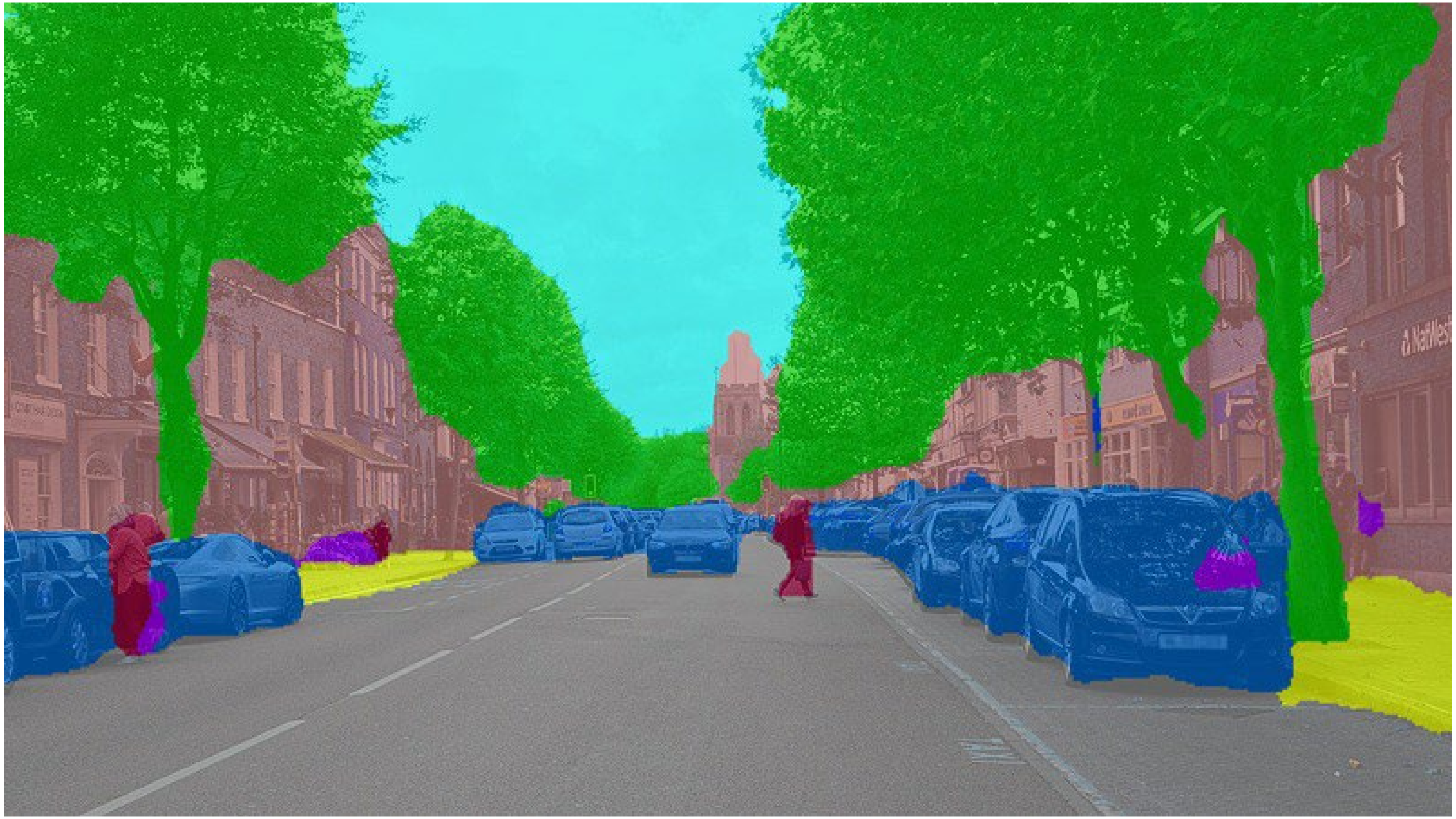
$$precision = \frac{TP}{TP + FP} = \frac{TP}{Predictions}$$

$$recall = \frac{TP}{TP + FN} = \frac{TP}{Ground\ Truth}$$

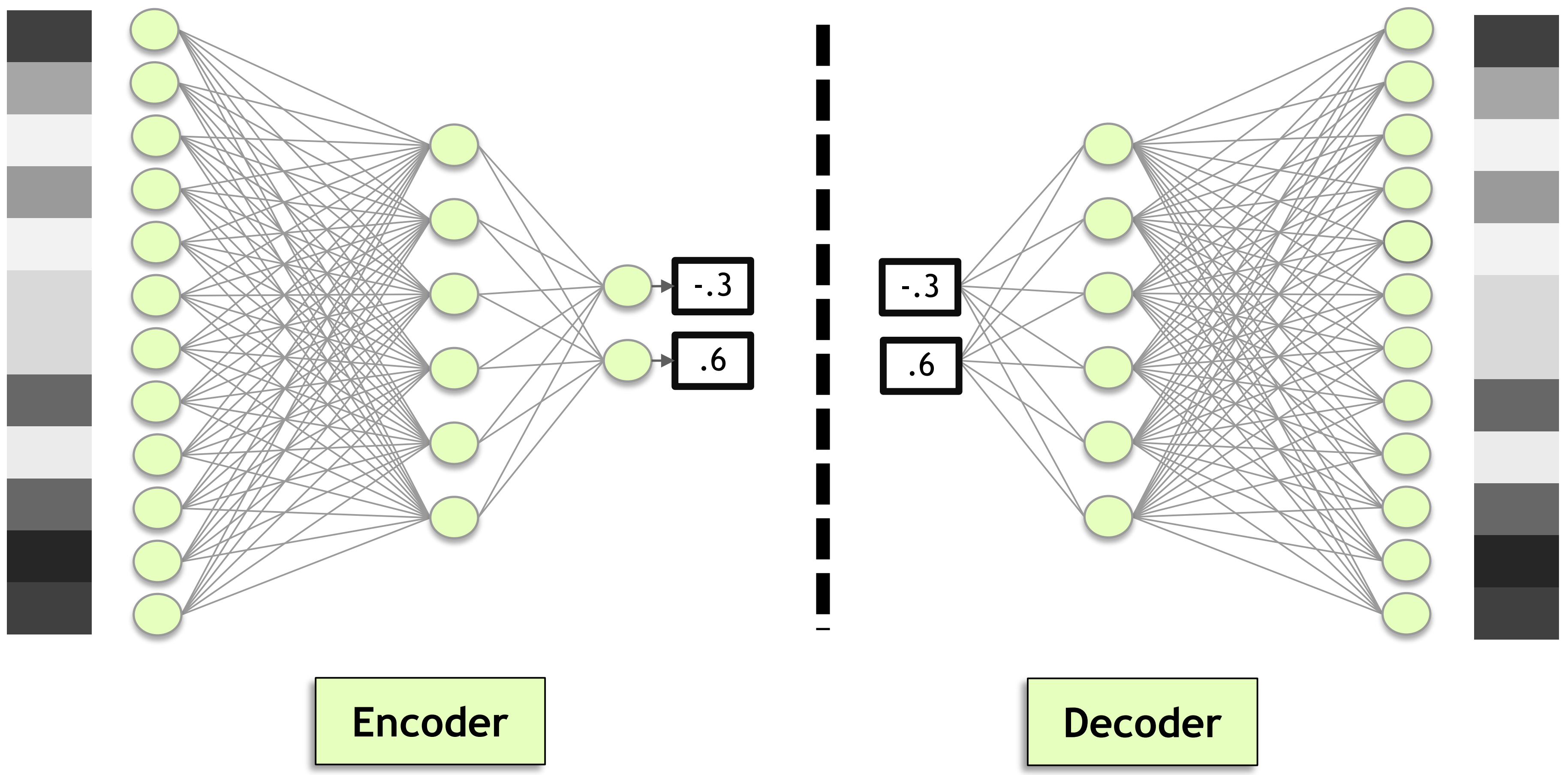
Mean average precision (mAP)



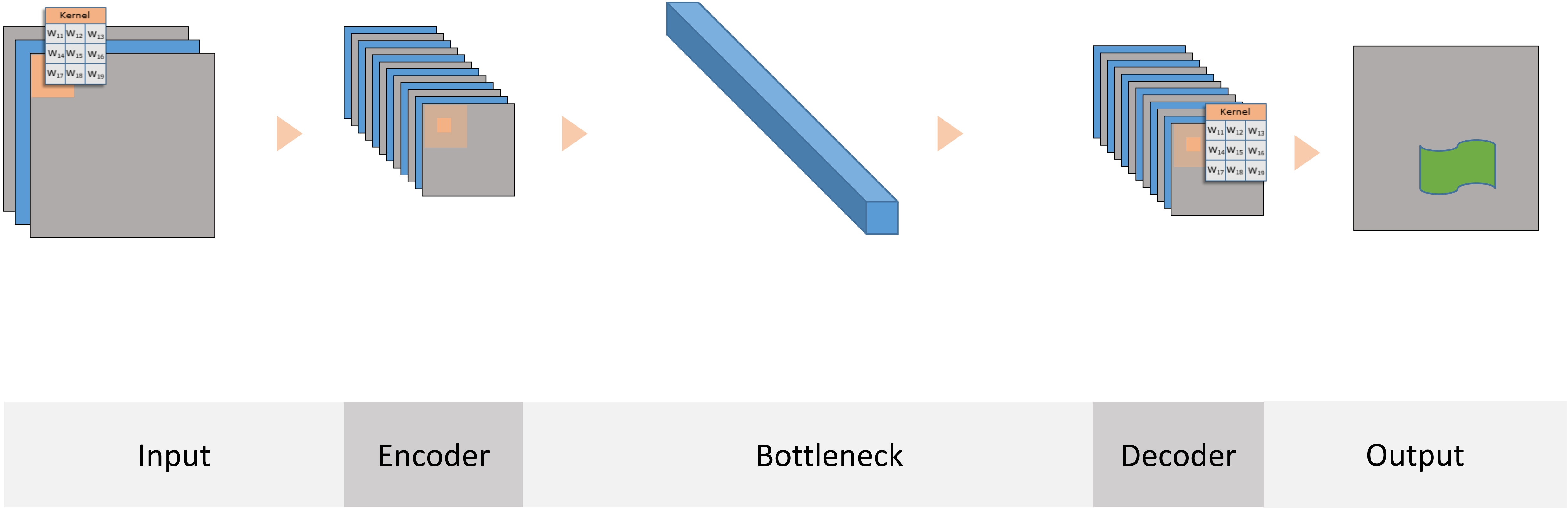
Dataset	Architecture
❑ ADE20k (150 classes) ❑ COCO&OpenImage ❑ …	❑ UNet ❑ SegNet ❑ …

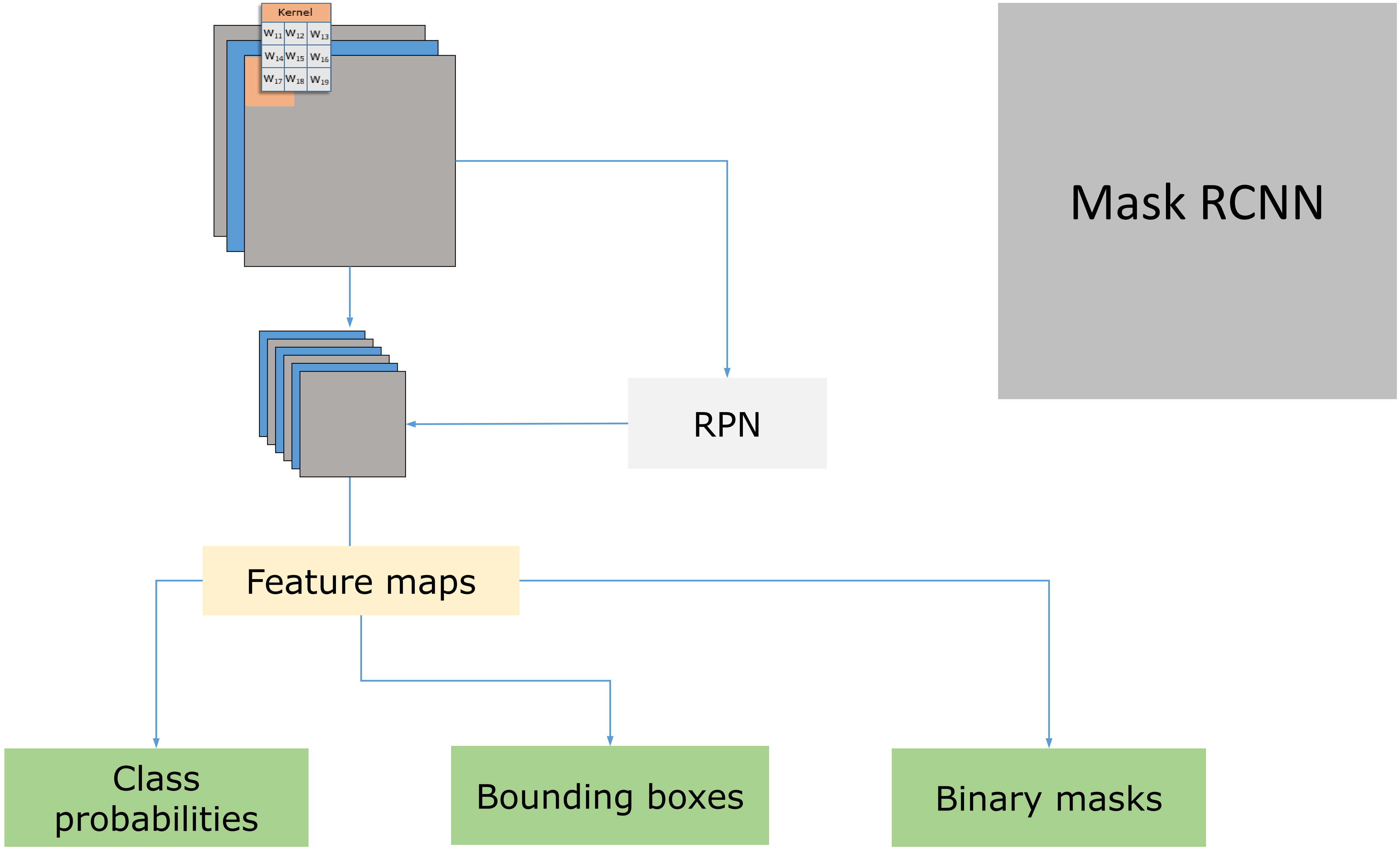


AUTOENCODERS

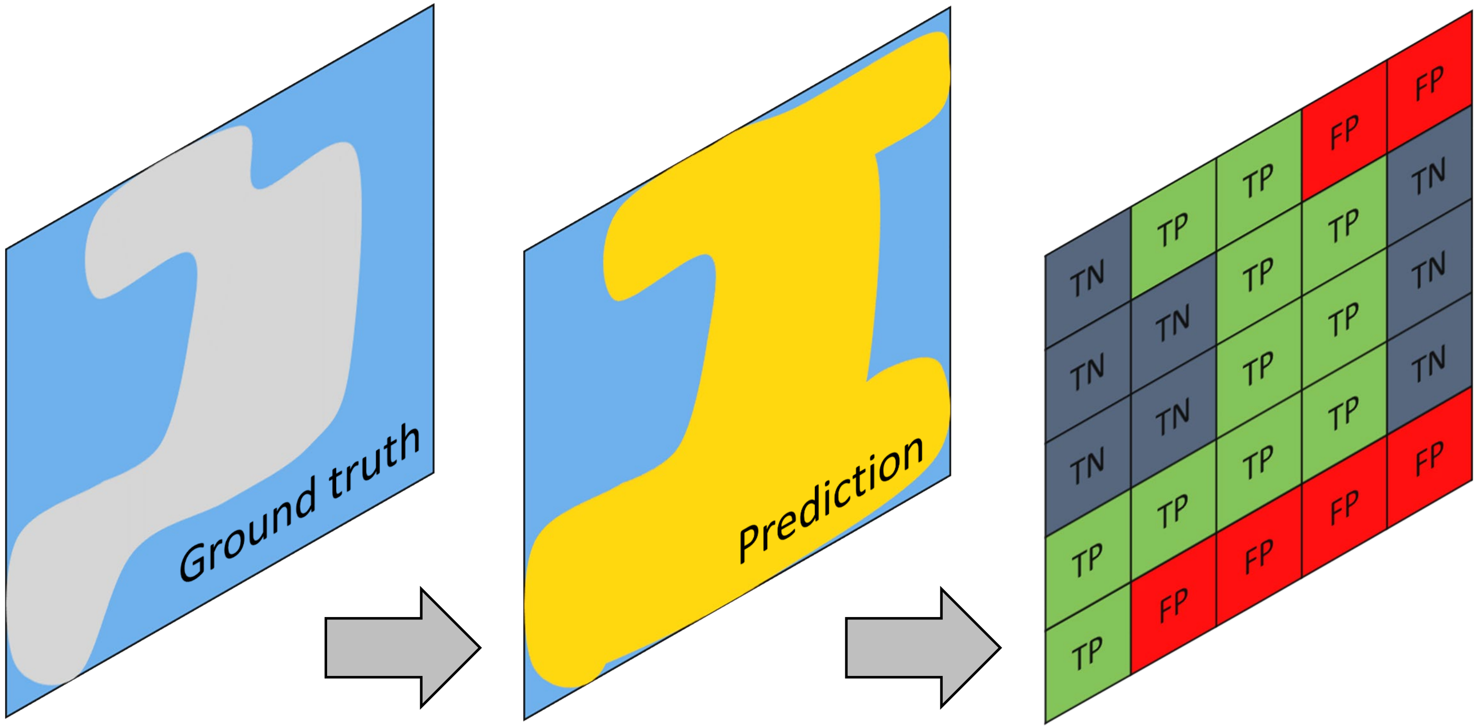


CNN Autocoder

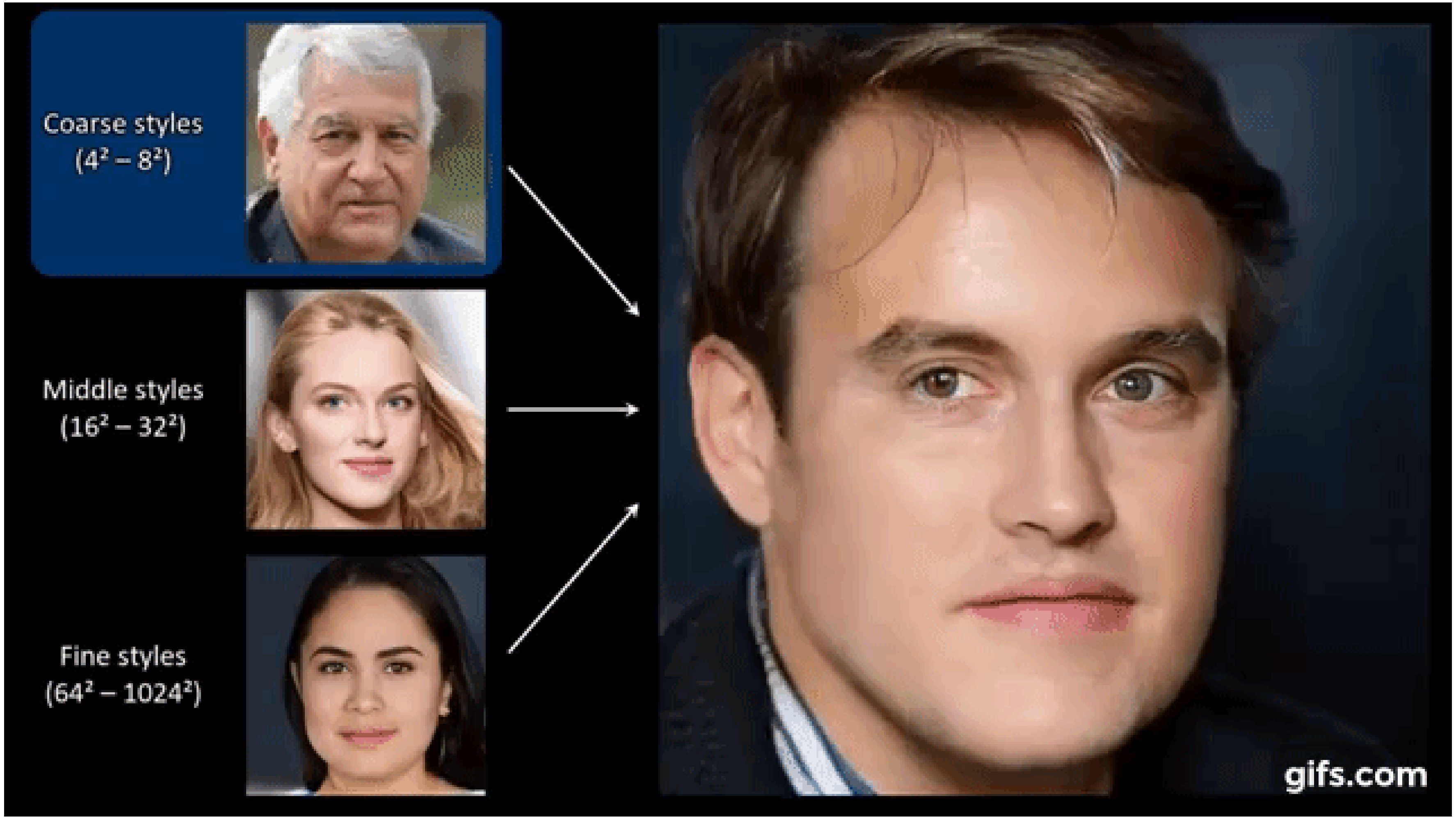




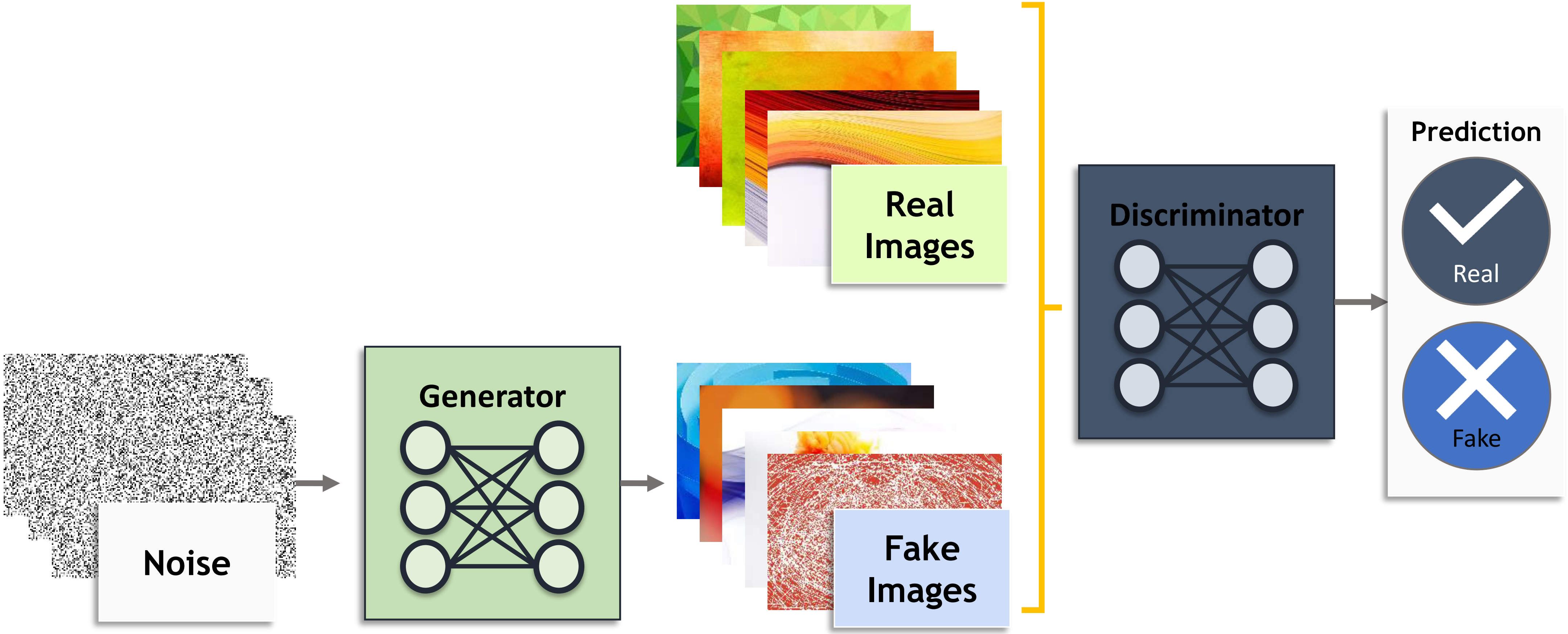
Kernel
w11 w12 w13
w14 w15 w16
w17 w18 w19
Mask RCNN
RPN
Feature maps
Class probabilities
Bounding boxes
Binary masks


Ground truth
Prediction
TN TP TP FP FP
TN TN TP TP TN
TN TN TP TP TN
TP TP TP TP TN
TP FP FP FP FP

Generative Adversarial Network (GAN)

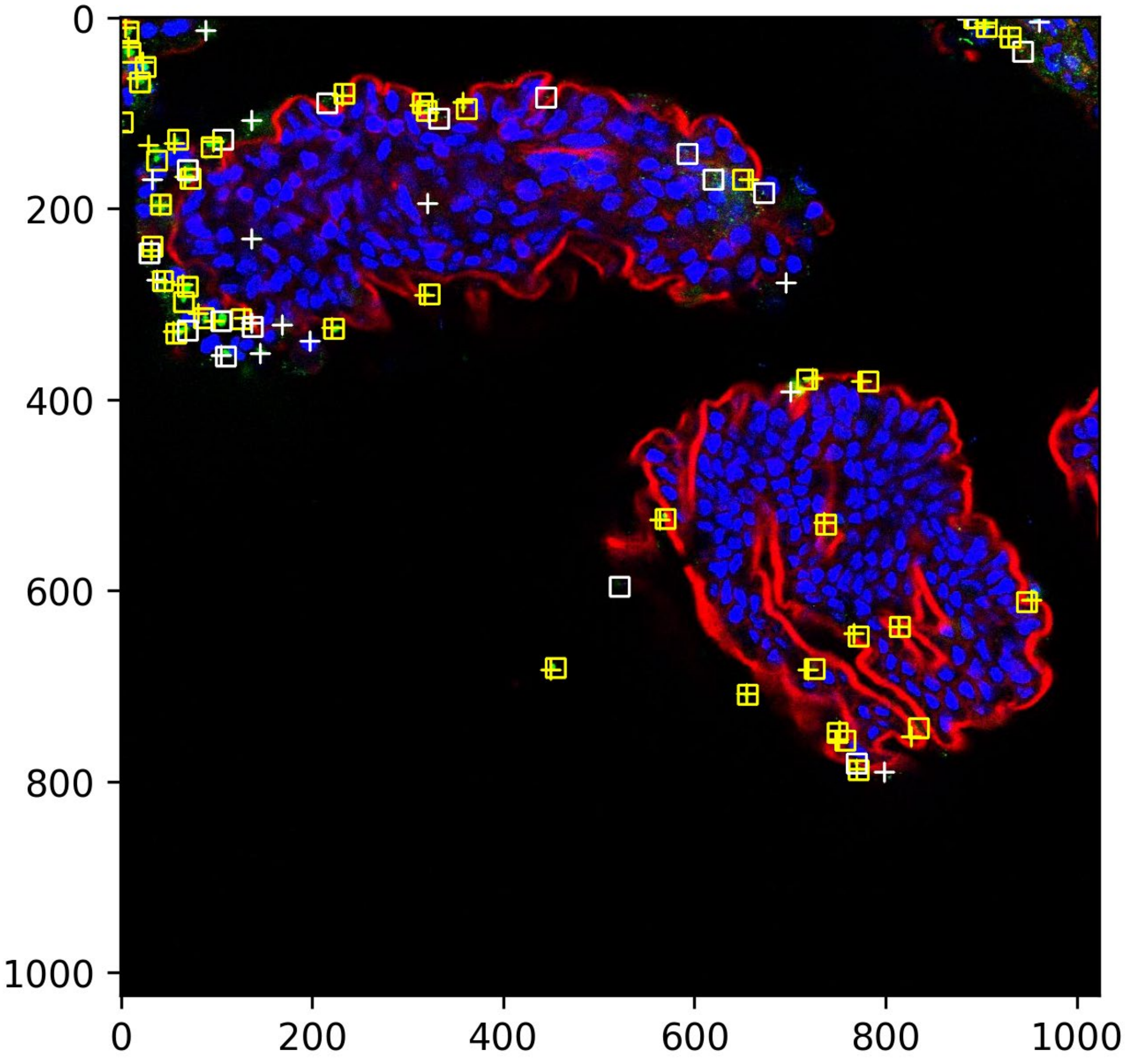


Generative Adversarial Networks (GANs)




Source: https://www.youtube.com/watch?v=2DiQUX11YaY
Source: https://www.youtube.com/watch?v=pW


frame


0
200
400
600
800
1000
0
200
400
600
800
1000