Image Processing & Vision

Lecture 13: IPV Review

Hak Gu Kim

hakgukim@cau.ac.kr

Immersive Reality & Intelligent Systems Lab (IRIS LAB)

Graduate School of Advanced Imaging Science, Multimedia & Film (GSAIM)

Chung-Ang University (CAU)

12 June 2023

Roadmap: Image Processing & Vision

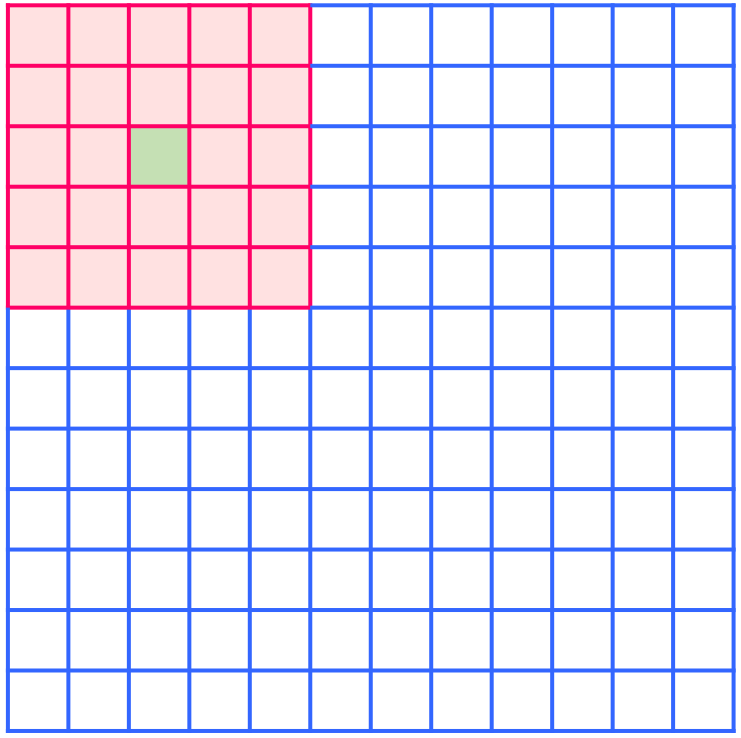
Pixels	Segments	Images	Videos	Advanced Topic
- Filtering	- Segmentation	- Recognition	- Motion	- Neural Networks
- Edges	- Clustering	- Detection	- Tracking	- CNNs for CV
- Features	- Resizing			

Roadmap: Image Processing & Vision

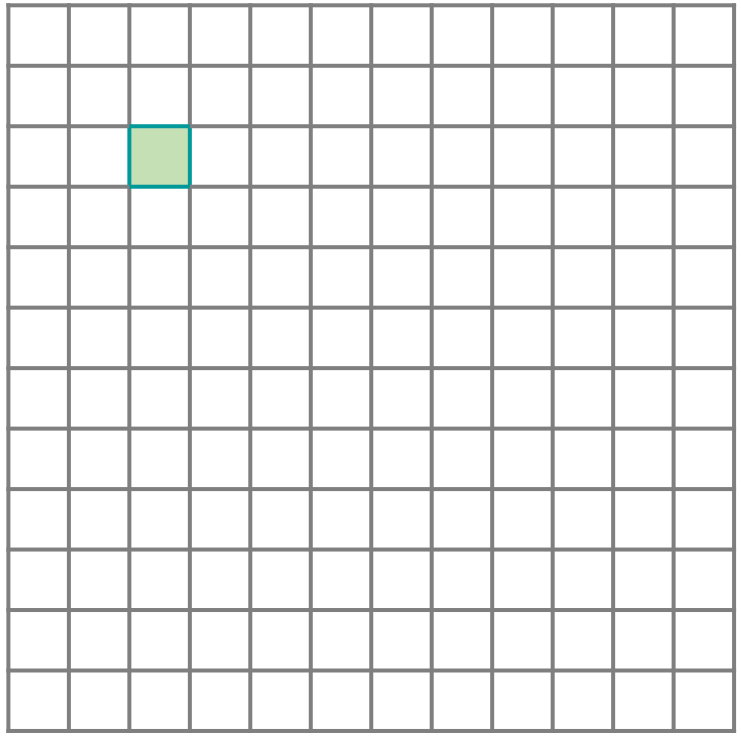
Pixels	Segments	Images	Videos	Advanced Topic
- **Filtering**	- Segmentation	- Recognition	- Motion	- Neural Networks
- Edges	- Clustering	- Detection	- Tracking	- CNNs for CV
- Features	- Resizing			

Linear Filters

- For a given x and y, superimpose the filter on the image centered at (x, y)
- Compute the new pixel value, $I'(x, y)$, as the sum of $m \times m$ values, where each value is the product of the original pixel value in $I(x, y)$ and the corresponding values in the filter



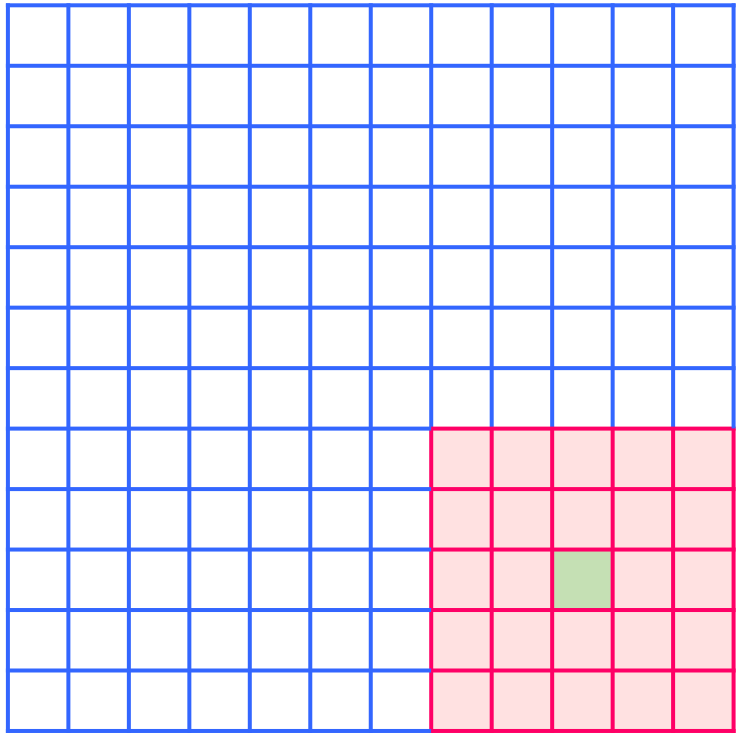
Input, $I(x, y)$



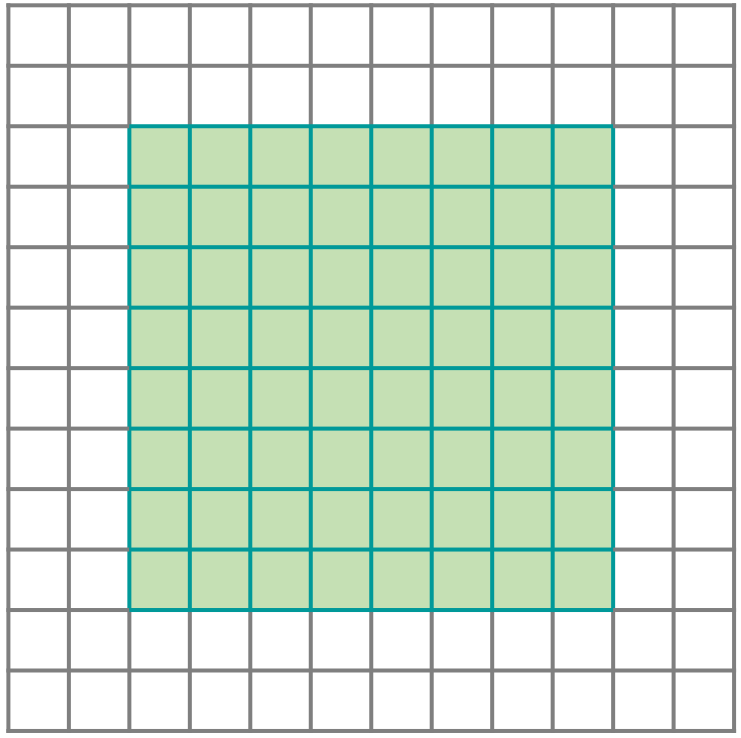
Output, $I'(x, y)$

Linear Filters

- For a given x and y, superimpose the filter on the image centered at (x, y)
- Compute the new pixel value, $I'(x, y)$, as the sum of $m \times m$ values, where each value is the product of the original pixel value in $I(x, y)$ and the corresponding values in the filter



Input, $I(x, y)$



Output, $I'(x, y)$

Example of **Linear Filters**

$$I'(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} F(i,j)\, I(x+i, y+j)$$

Output | Filter | Image

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	180	180	180	180	180	0	0
0	0	0	180	180	180	180	180	0	0
0	0	0	180	180	180	180	180	0	0
0	0	0	180	0	180	180	180	0	0
0	0	0	180	180	180	180	180	0	0
0	0	0	0	0	0	0	0	0	0
0	0	180	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Image, $I(x,y)$

$*$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Filter, $F(x,y)$

$=$

	0	20	40	60	60	60	40	20	
	0	40	80	120	120	120	80	40	
	0	60	120	180	180	180	120	60	
	0	60	100	160	160	180	120	60	
	0	60	100	160	160	180	120	60	
	0	40	60	100	100	120	80	40	
	20	40	60	60	60	60	40	20	
	20	20	20	0	0	0	0	0	

Output, $I'(x,y)$

Non-linear Filters: **Median Filter**

- **Median** filter takes the **median value** of the pixels under the filter
  ① Sorting the pixel values in the filter
  ② Pick the median value as an output pixel

5	13	5	221	7
4	16	7	34	98
24	54	34	23	21
23	75	90	123	89
54	25	67	12	24

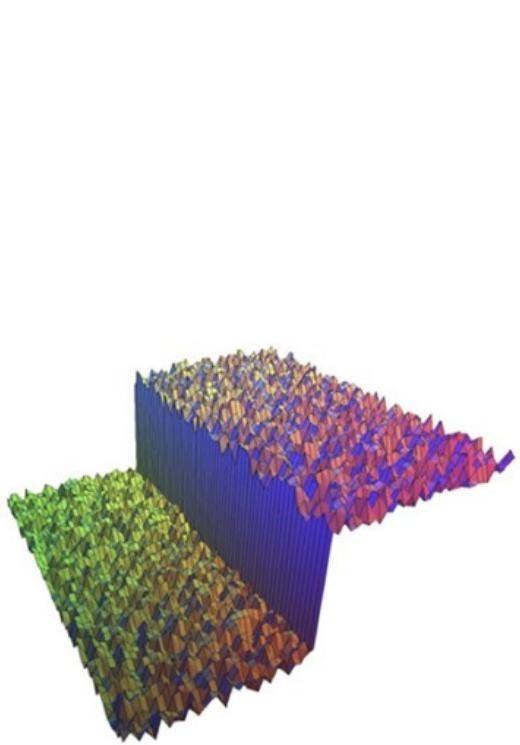
Image

12	21	23	24	34	67	89	90	123

	13	23	23	
	24	34	34	
	54	54	34	

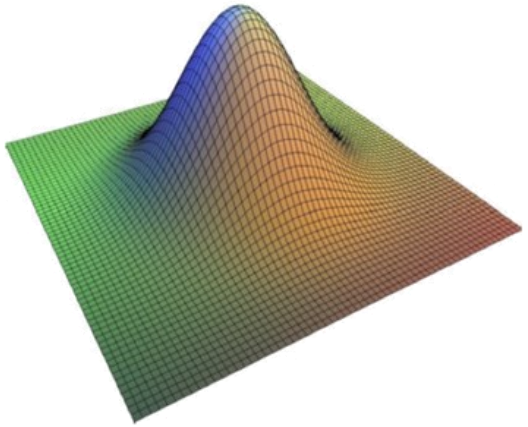
Output

Non-linear Filters: **Bilateral Filter**

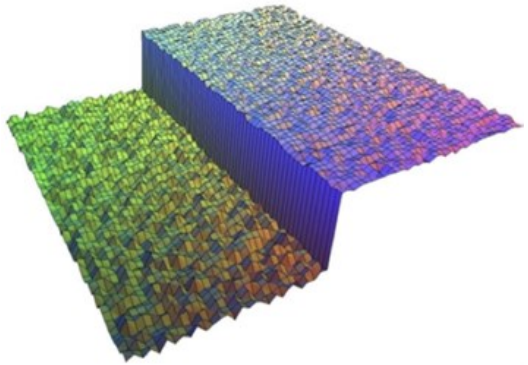


Input

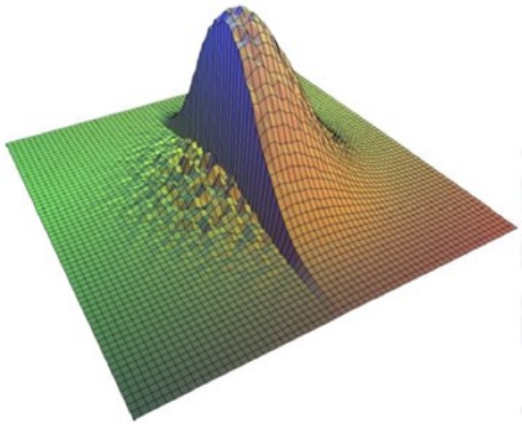
*



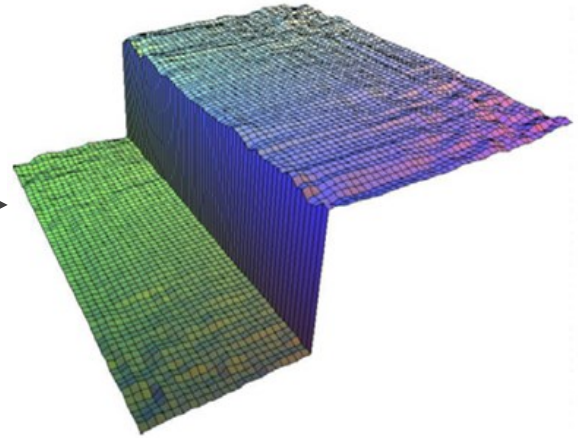
Domain (Spatial) Kernel



Range Kernel Influence



Bilateral Filter
(=Domain×Range)

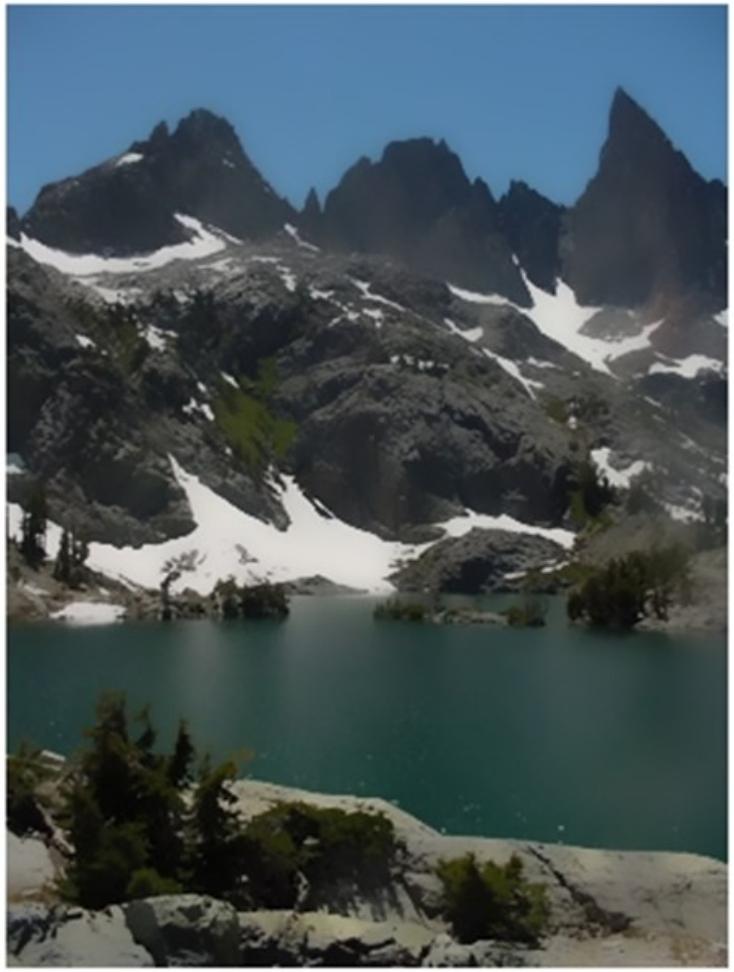


Output

Durand and Dorsey, Image from Fast Bilateral Filtering for the Display of High-Dynamic Range Images, **ACM Trans. Graphics (ToG), 2002**

Non-linear Filters: **Bilateral Filter**

Original image



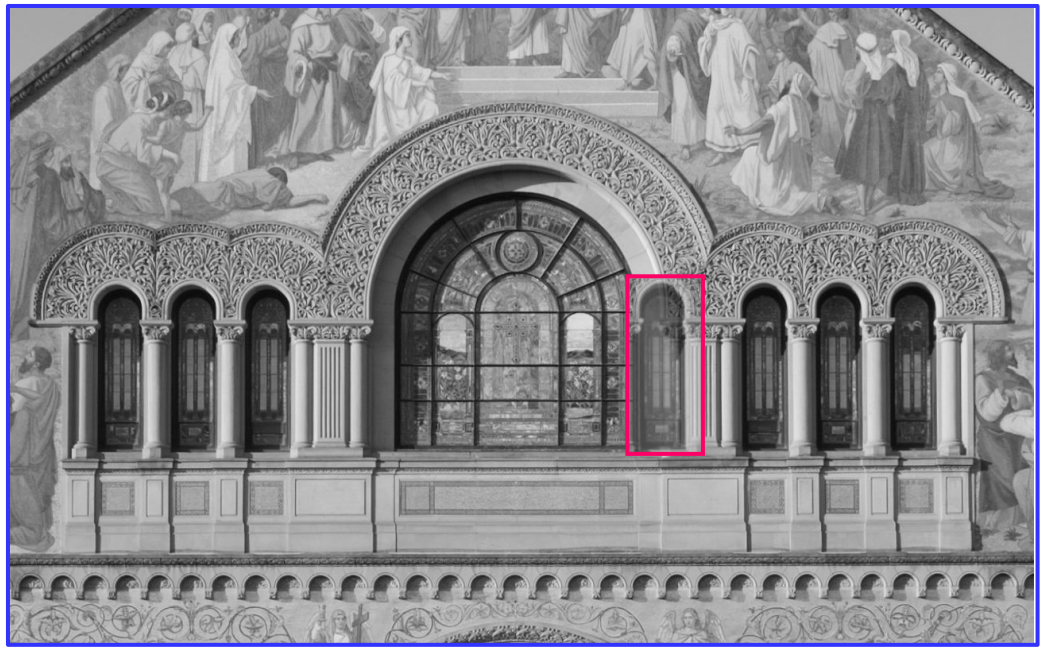
Result of **Bilateral** filter

https://en.wikipedia.org/wiki/Bilateral_filter

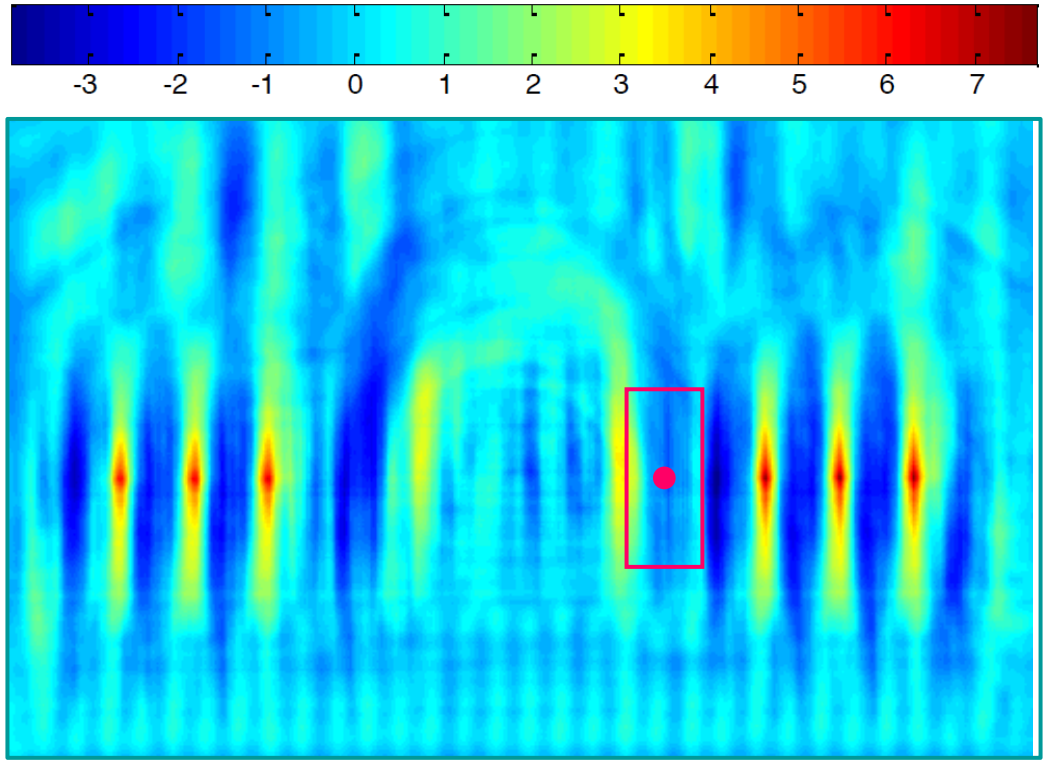
Correlation: Template Matching

- Template matching is a technique in digital image processing **for finding small parts of an image** which match a template image

Template



Image



Correlation map

Roadmap: Image Processing & Vision

Pixels	Segments	Images	Videos	Advanced Topic
- Filtering	- Segmentation	- Recognition	- Motion	- Neural Networks
- **Edges**	- Clustering	- Detection	- Tracking	- CNNs for CV
- Features	- Resizing			

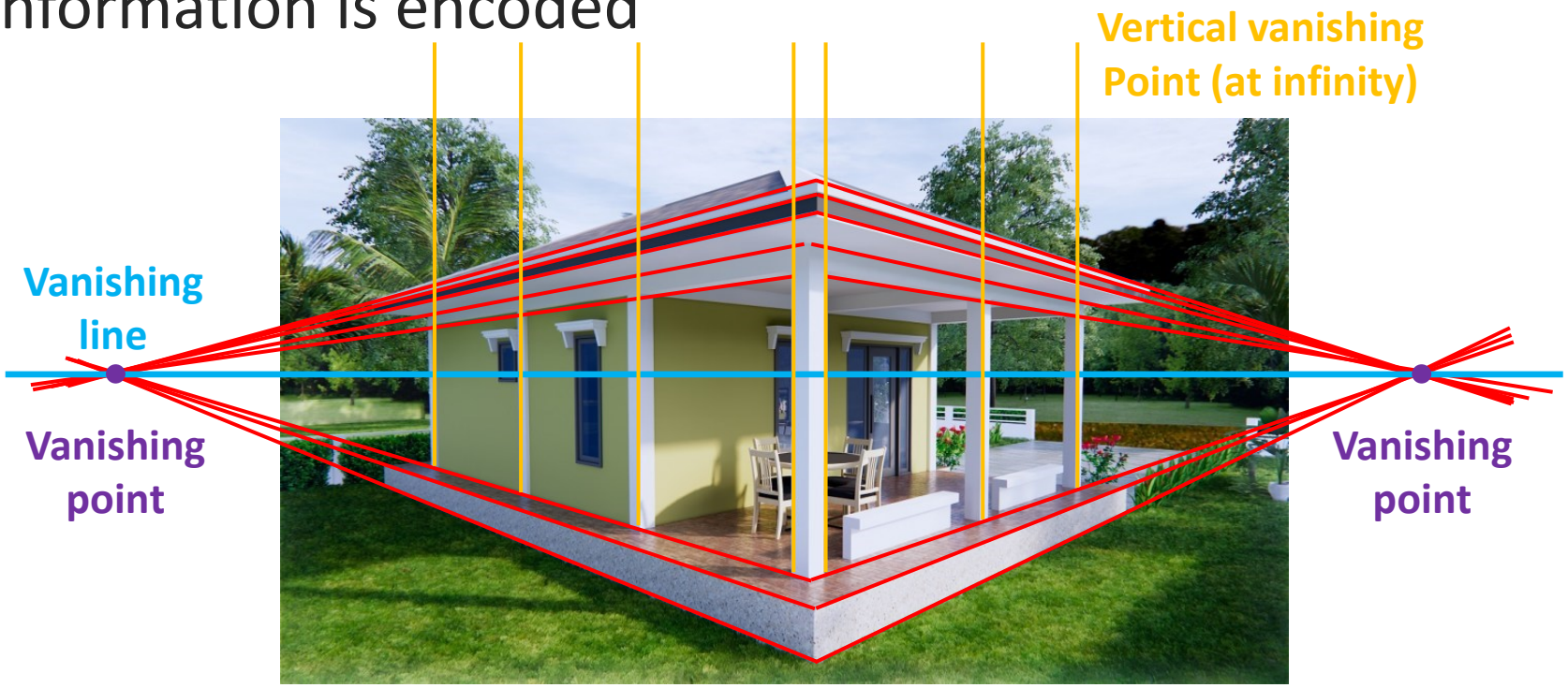
Edge Detection

- **Goal**

— Identify sudden changes in image intensity

— It is where most shape information is encoded

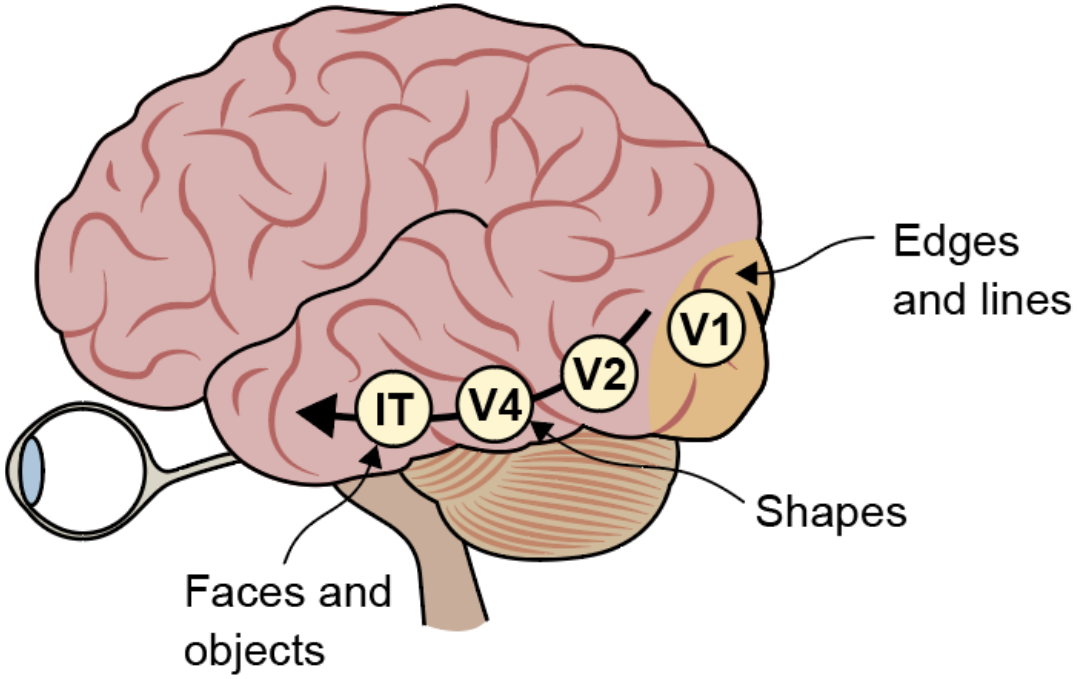
Information extraction
& Object recognition

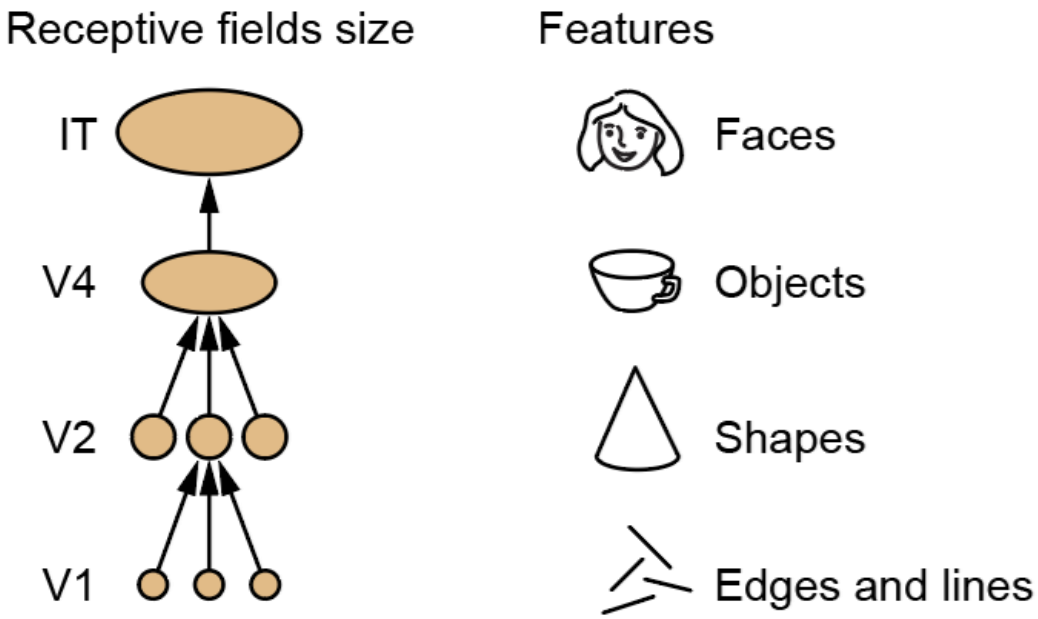


Geometry and viewpoint recover
& 3D reconstruction

Discoveries about **Visual System and Processing**

- Organization of the **Visual Cortex**
  - It is widely known that in V1 region, all neurons respond to rather simple forms of stimuli on different areas of the retina.
  - It is also known that neurons in other regions of the brain (called V2, V4, and IT) respond to increasingly complex visual stimuli like, for example, a whole face.





https://freecontent.manning.com/deep-learning-for-image-like-data/

Sobel Edge Detector

① Use **central differencing** to compute gradient image

② **Threshold** to obtain edges

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{I} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \left(\begin{bmatrix} -1 & 0 & +1 \end{bmatrix} * \mathbf{I} \right)$$

Sobel operator for horizontal changes

Smoothing

Derivative

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{I} = \begin{bmatrix} +1 \\ 0 \\ -1 \end{bmatrix} * \left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \mathbf{I} \right)$$

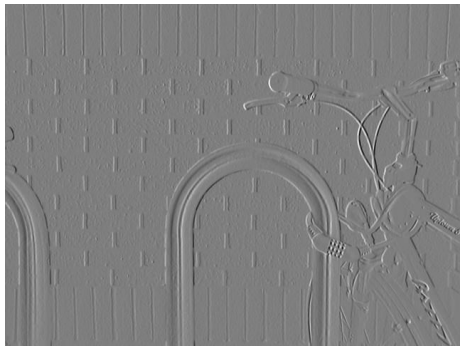
Sobel operator for vertical changes

Derivative

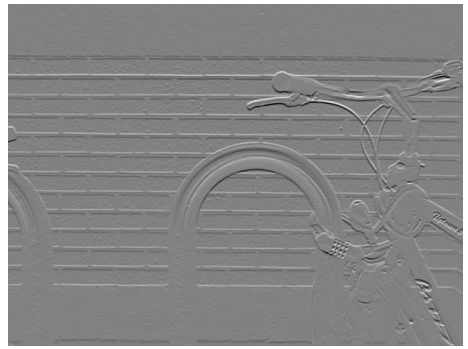
Smoothing

Original image

Gradient magnitude



x-gradient



y-gradient

Canny Edge Detector

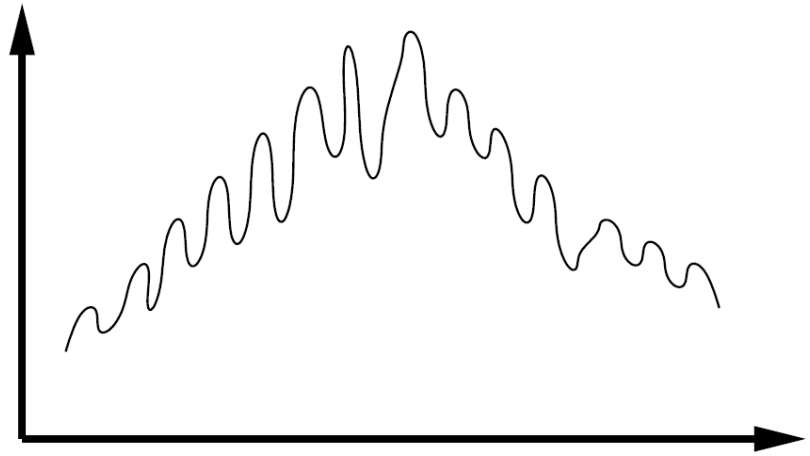
- **Canny edge detector** is based on a **local extrema of a first derivative** operator approach

- **Design Criteria:**

— Good detection
  - Low error rate for omissions (missed edges)
  - Low error rate for commissions (false positive)

— Good localization

— Single response to a given edge
  - Eliminate multiple responses to a single edge



Q: How many edges are?

Q: What is the position of each edge?

Canny Edge Detector

① Apply **directional derivatives of Gaussian**

② Compute **gradient magnitude** and **gradient direction**

③ **Non-maximum suppression (NMS)**

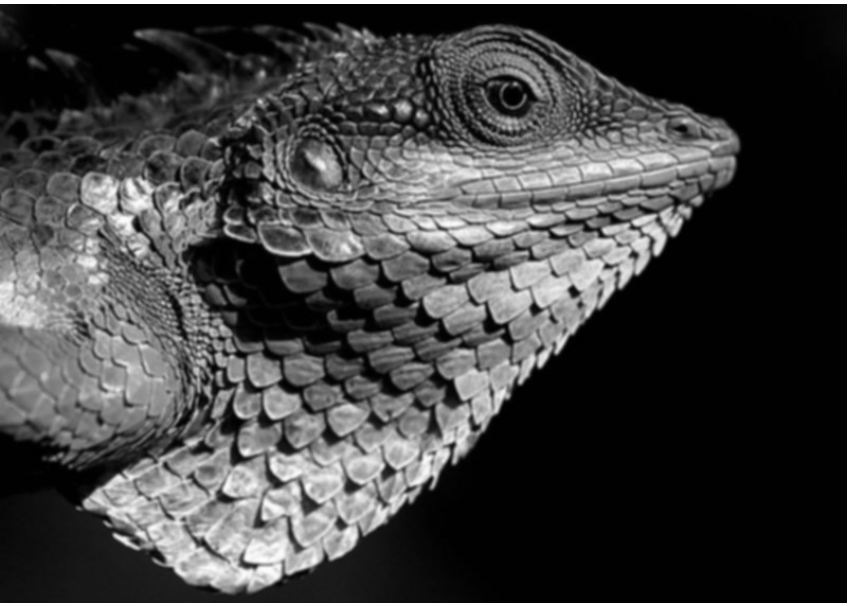
— Thin multi-pixel wide “ridges” down to single pixel width

④ **Link** and **threshold**

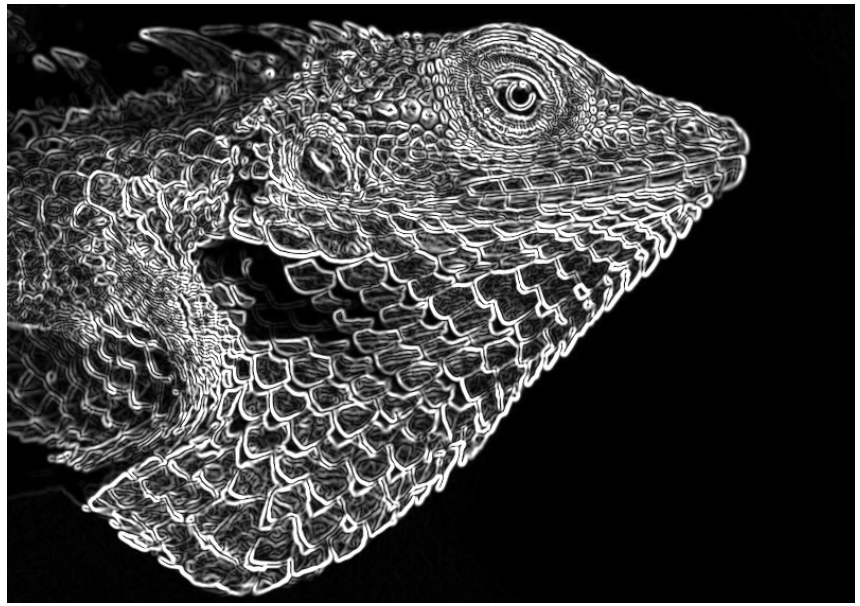
— Low and high edge-strength thresholds

— Accept all edges over low threshold that are connected to edge over high threshold

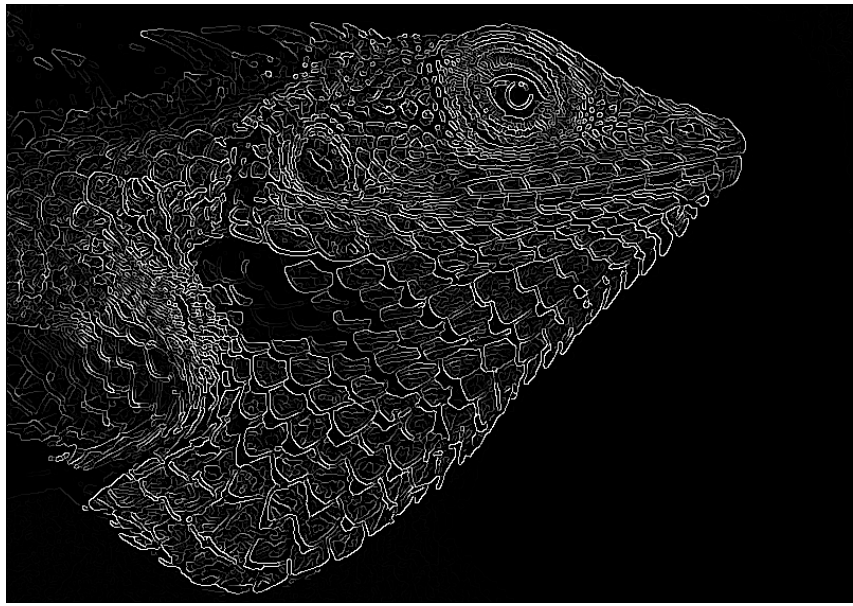
Canny Edge Detector: **Non-Maximum Suppression**



Original image



Gradient magnitude



Result of non-maximum suppression

https://en.wikipedia.org/wiki/Canny_edge_detector

Roadmap: Image Processing & Vision

Pixels	Segments	Images	Videos	Advanced Topic
- Filtering	- Segmentation	- Recognition	- Motion	- Neural Networks
- Edges	- Clustering	- Detection	- Tracking	- CNNs for CV
- **Features**	- Resizing			

Applications of Features

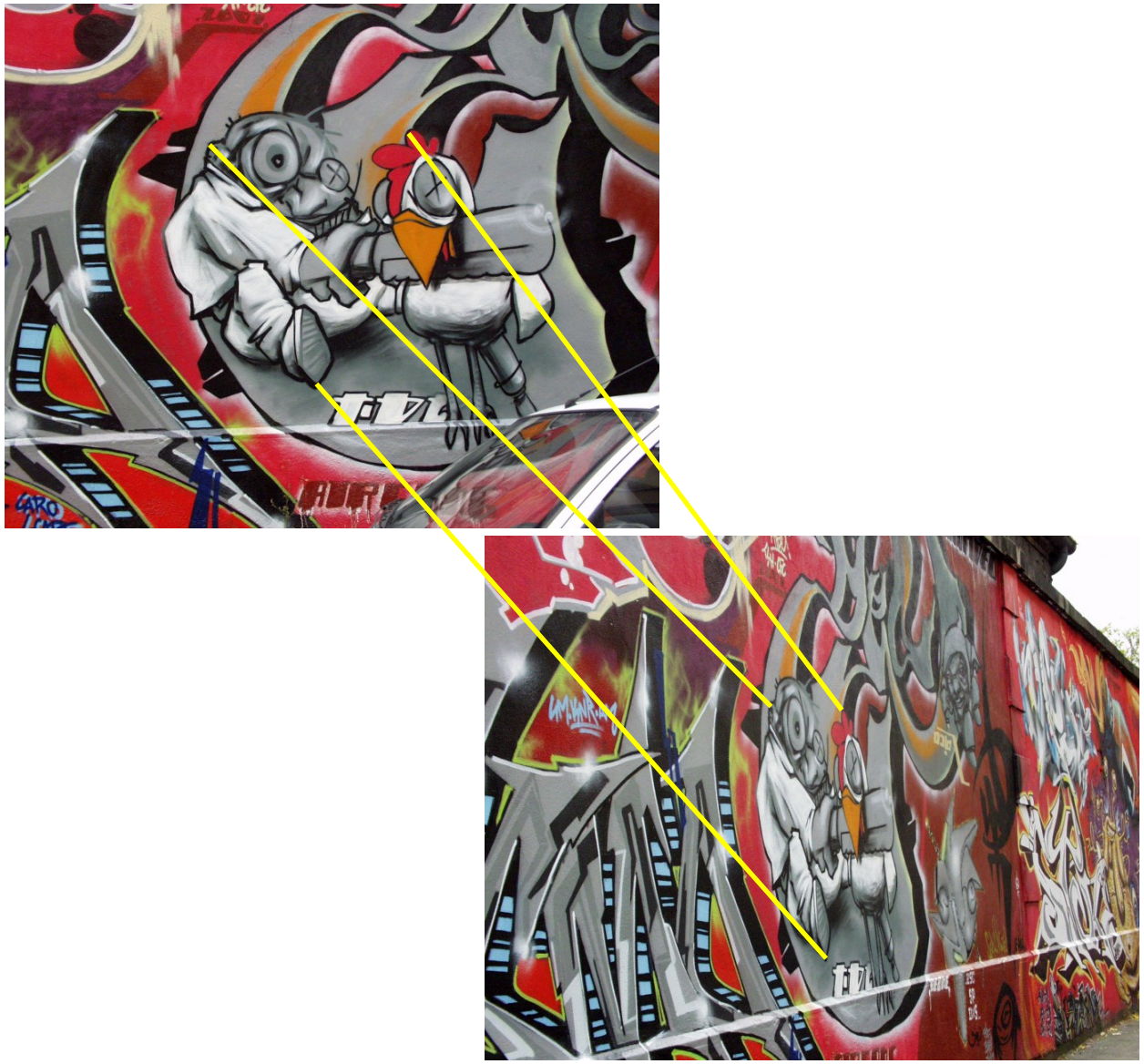


Image matching

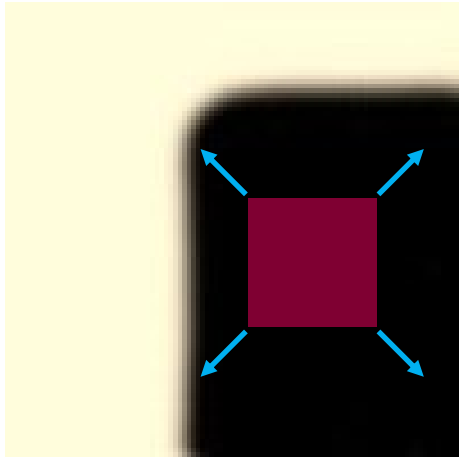
Corner

- **Key Property:**

— In the region around a corner, image gradient has two or more dominant directions

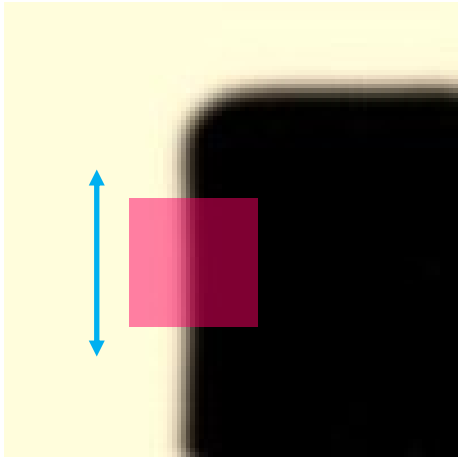
— Corners are robust and distinctive

Flat



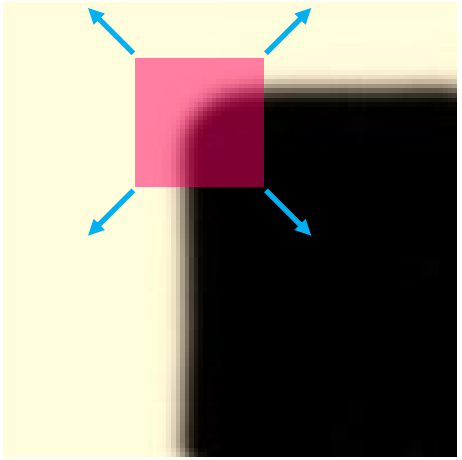
No changes
in all directions

Edge



No changes
along the edge direction

Corner



Significant changes
in all directions

C. Harris and M. Stephens, A combined Corner and Edge Detector, **Alvey Vision Conference**, 1988

Harris Corner Detection

① Compute **image gradient** over small region

② Compute the **covariance matrix**

③ Compute **eigenvectors** and **eigenvalues**

④ Use **threshold on eigenvalues** to detect corners

Harris Corner Detection: ④ **Eigenvalue Thresholding**

- Use **threshold on eigenvalues** to detect corners

- **Cornerness:**

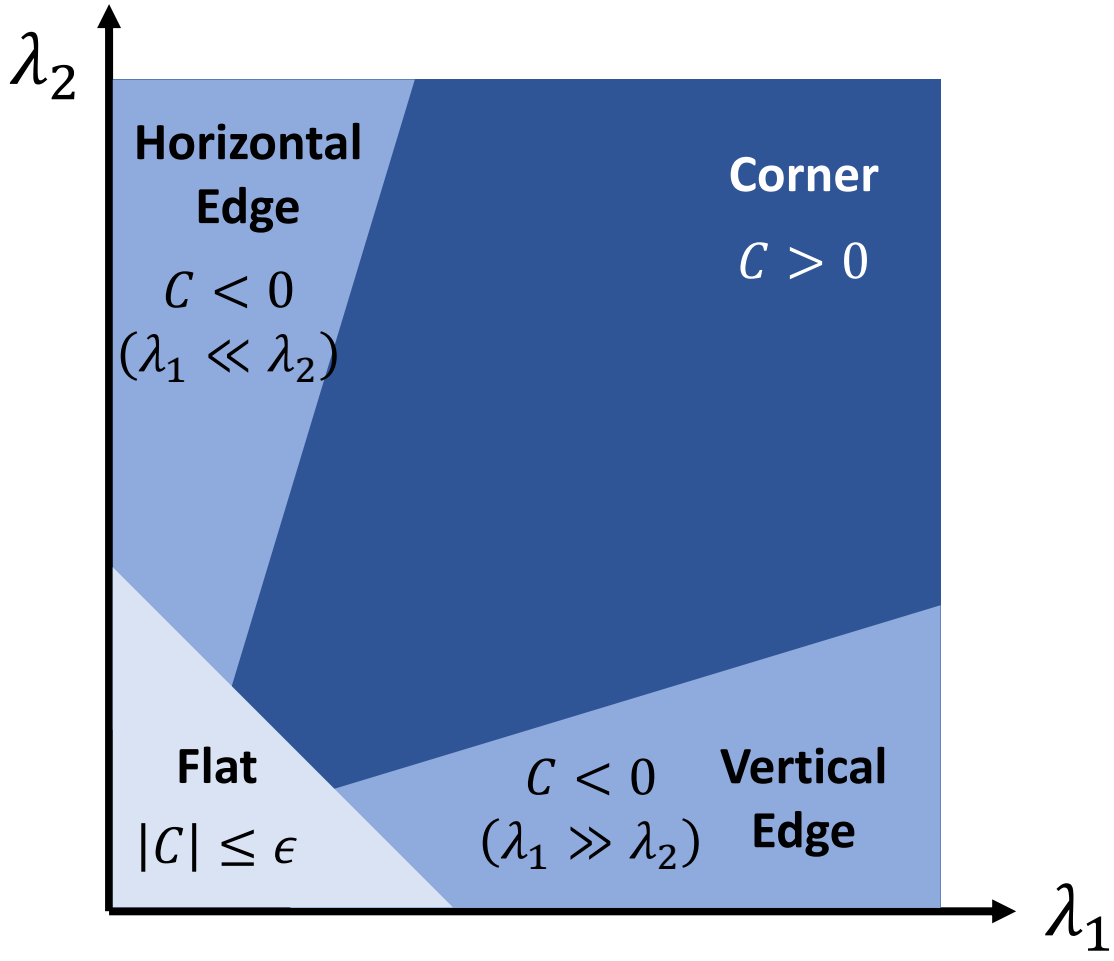
$$C = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$
$$= \det(\mathbf{M}) - \alpha \cdot \mathrm{tr}(\mathbf{M})^2$$

$0.04 \leq \alpha \leq 0.06$

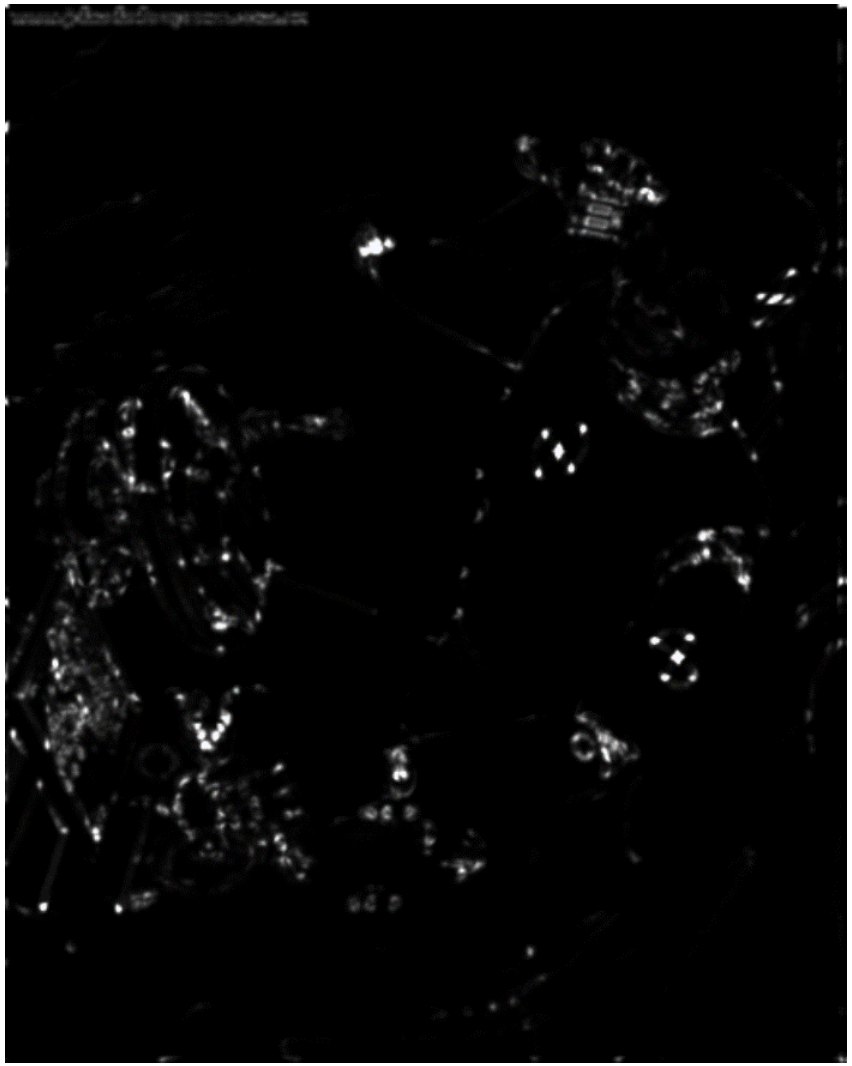
— Flat region: $|C| \leq \epsilon$

— Edge: $C < 0$

— Corner: $C > 0$



Example: Harris Corner Detection

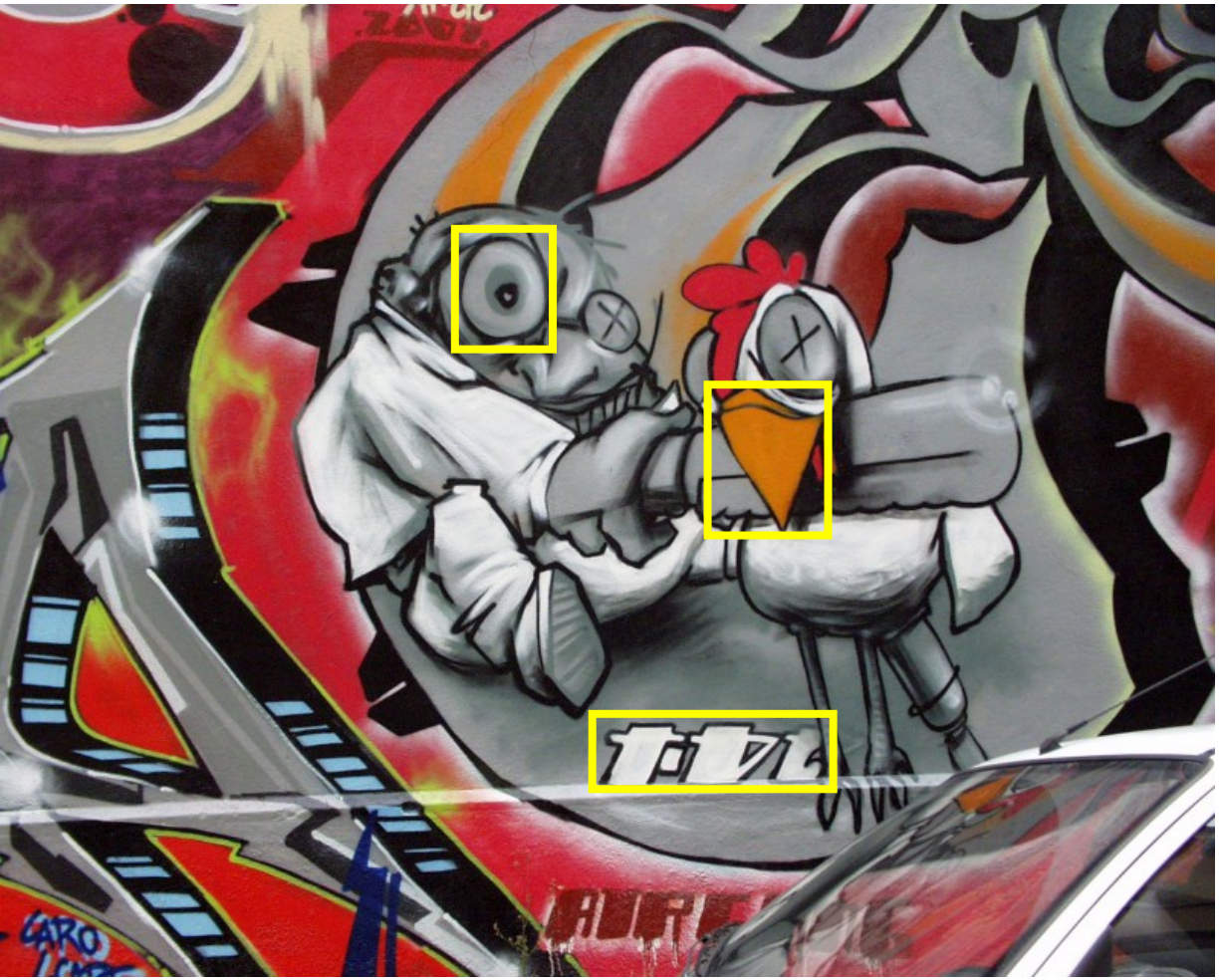


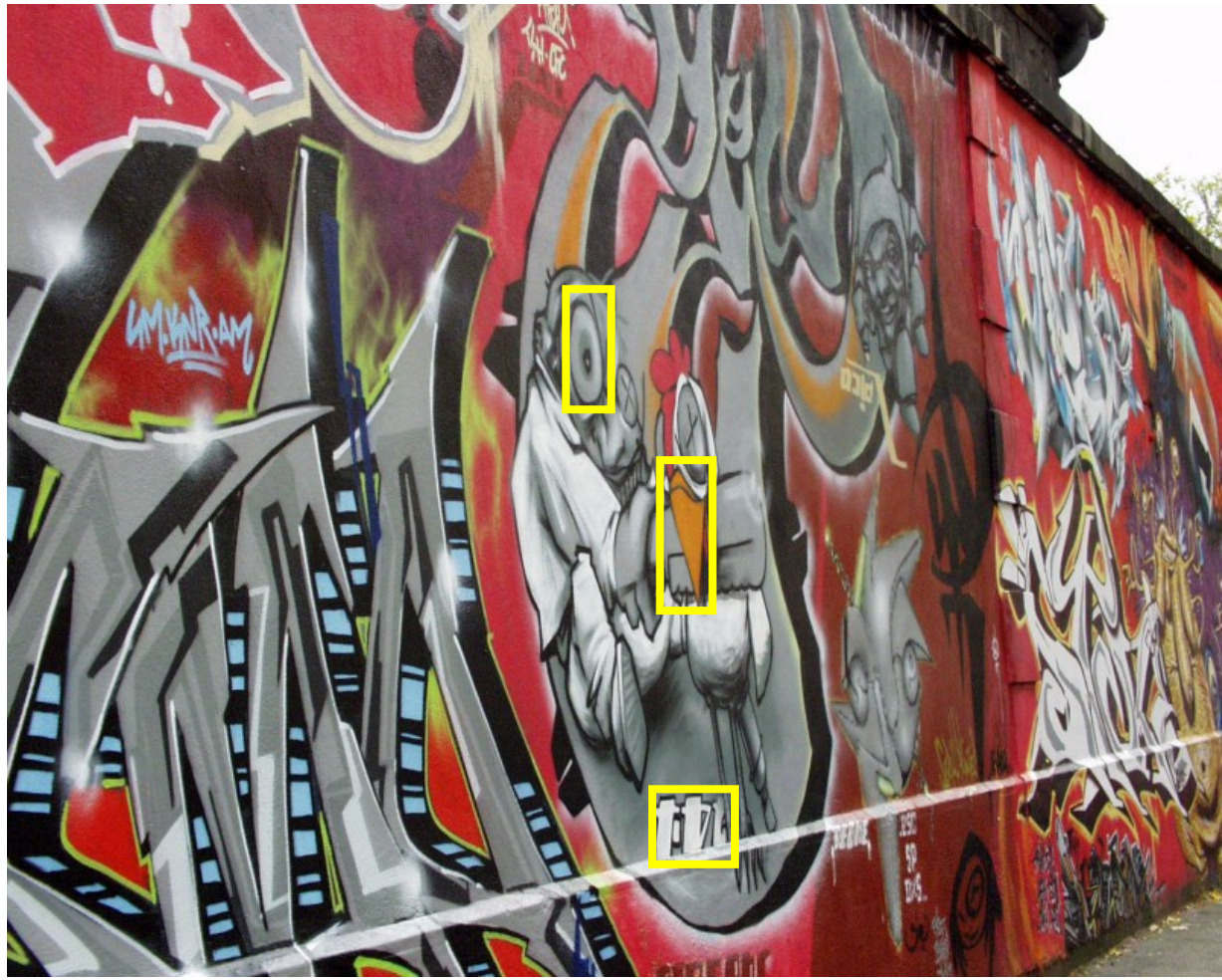
Corner response map

Corner detection result ($\sigma = 1$)

Good Local Features

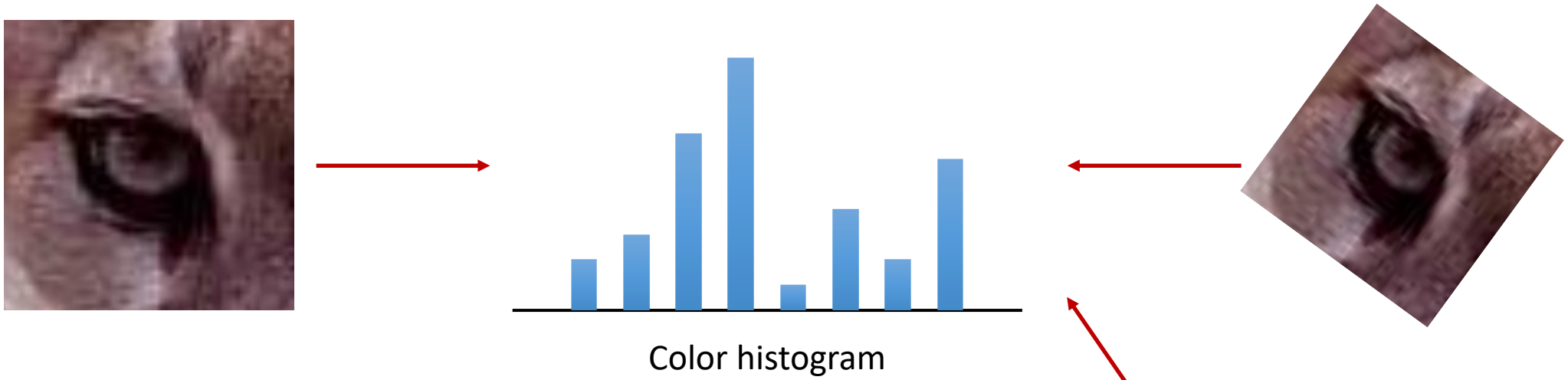
- **Patch** around the local feature is much more **informative**





Color Histogram

- Count the colors in the image using a histogram

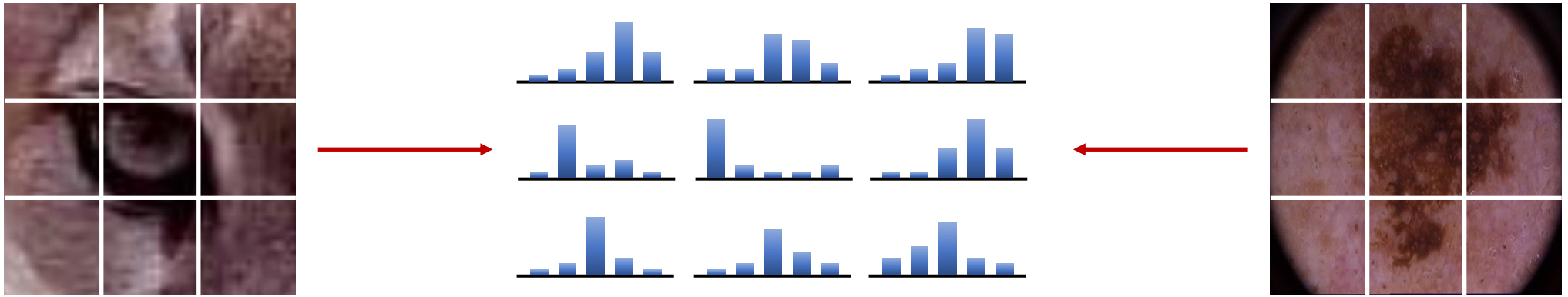


— **Pros:** Invariant to changes in scale and rotation

— **Cons:** Ignoring its shape and texture (i.e., spatial layout)

Spatial Histograms

- Count the colors in the image using a histogram



Spatial histograms

— **Pros:** Invariant to some deformations and retains rough spatial layout

— **Cons:** Sensitive to rotation

Scale Invariant Feature Transform (SIFT)

- The scale-invariant feature transform (SIFT) is an algorithm to **detect**, **describe**, and **match** local features in images
- SIFT describes both a **detector** and **descriptor**

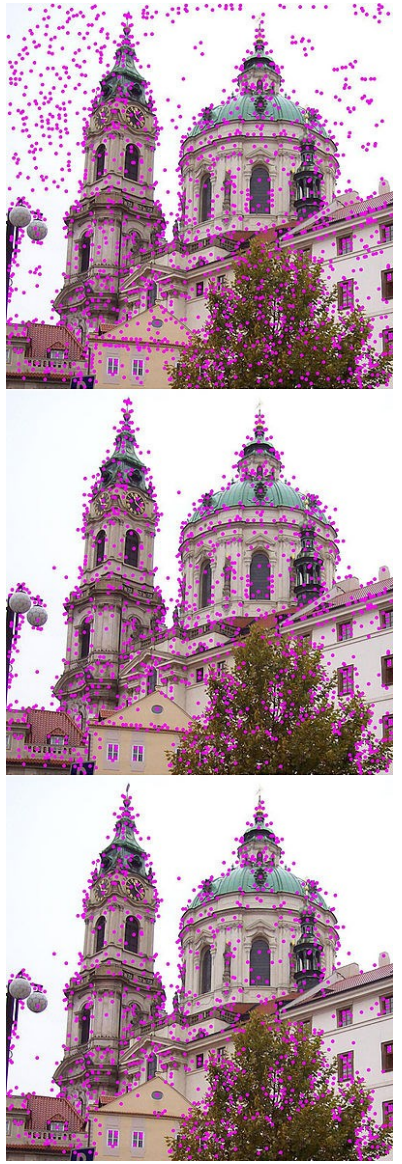
— Applications: object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, *etc*.

① Multi-scale extrema detection

② Keypoint localization

③ Orientation assignment

④ Keypoint descriptor

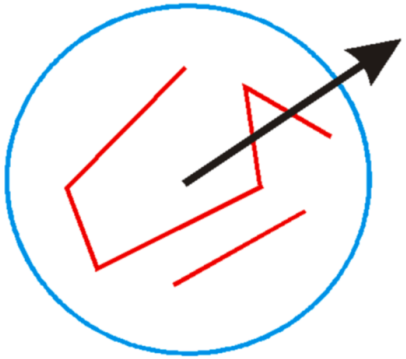


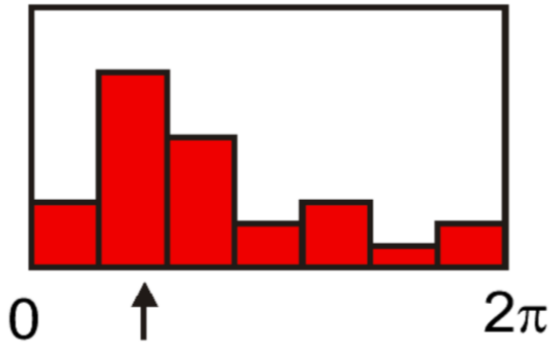
SIFT: **Orientation Assignment**

- Each keypoint is assigned one or more orientations based on local image gradient directions
- This is the key step in achieving invariance to rotation on the Gaussian-smoothed image, L

$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2}$$

$$\theta(x,y) = \tan^{-1}\big[\big(L(x+1,y) - L(x-1,y)\big)/\big(L(x,y+1) - L(x,y-1)\big)\big]$$





SIFT: **Keypoint Descriptor**

- Thresholded image gradients are sampled over 16 x 16 array of locations in scale space (weighted by a Gaussian with sigma half the size of the window)
- Create array of orientation histograms
- 8 Orientations x (4 x 4) histogram array

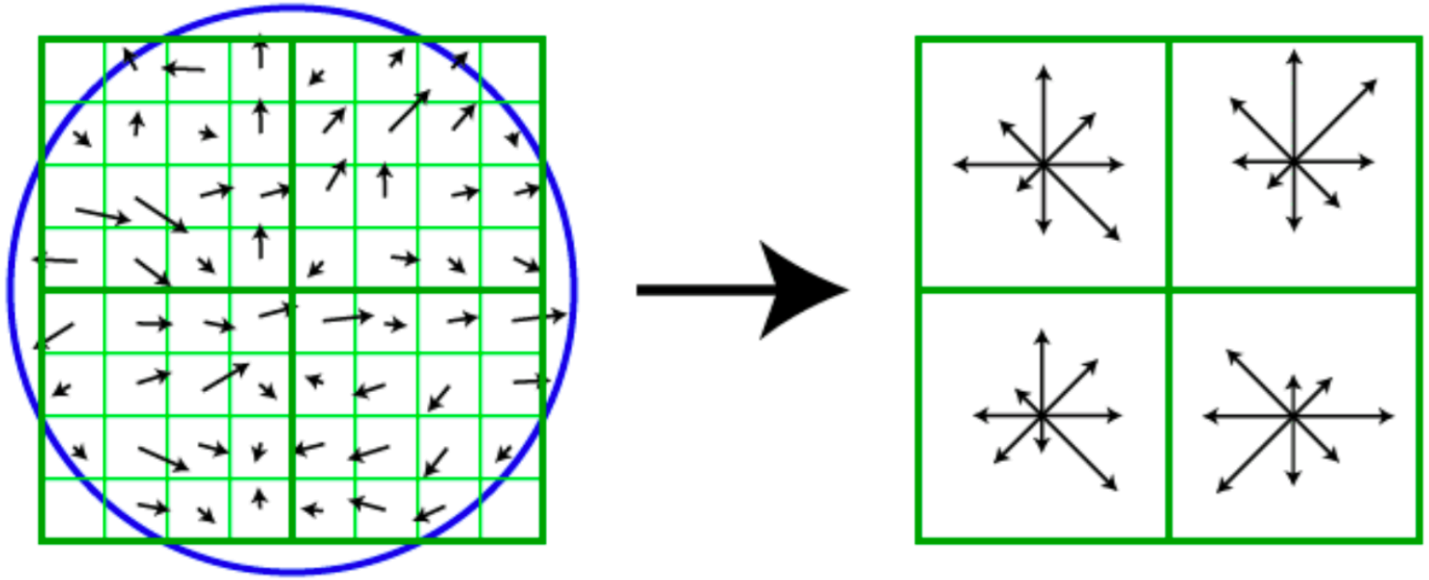


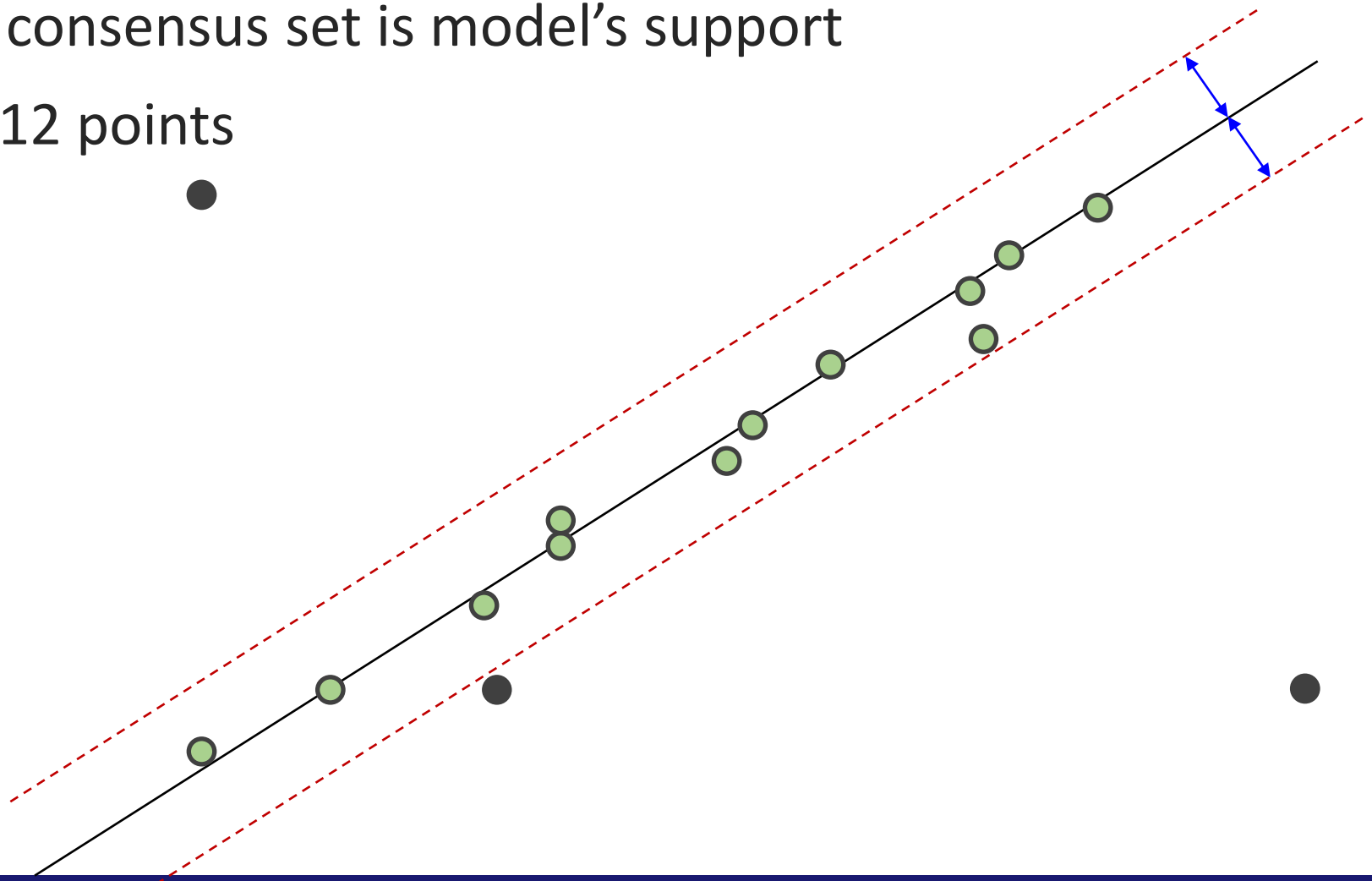
Image gradients

Keypoint descriptor

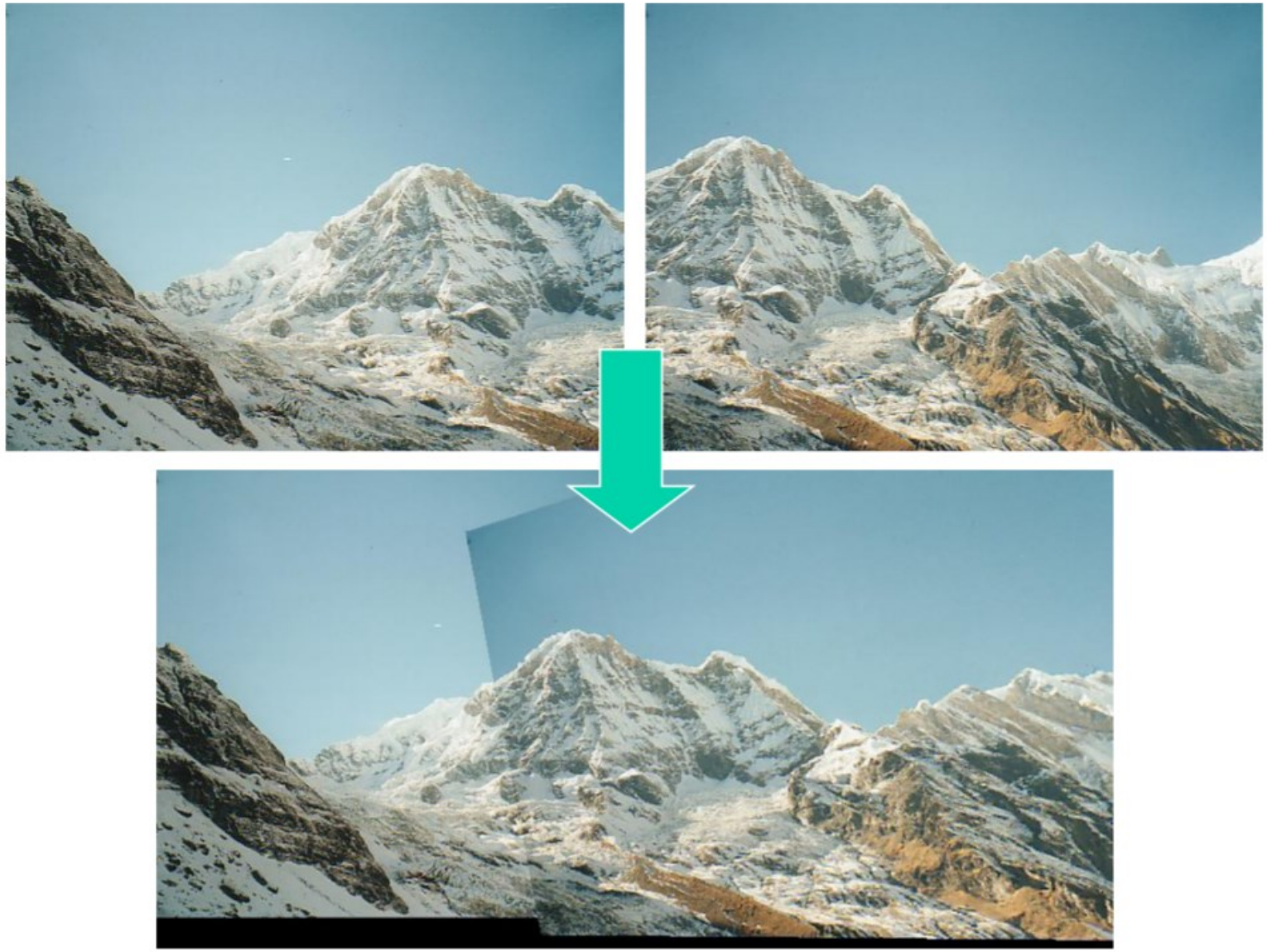
RANSAC (RANdom SAmple Consensus)

② **Points within some distance threshold**, t, of model are a consensus set. Size of consensus set is model's support

— **Inliers:** 12 points



Panorama Image Stitching: SIFT & RANSAC



Roadmap: Image Processing & Vision

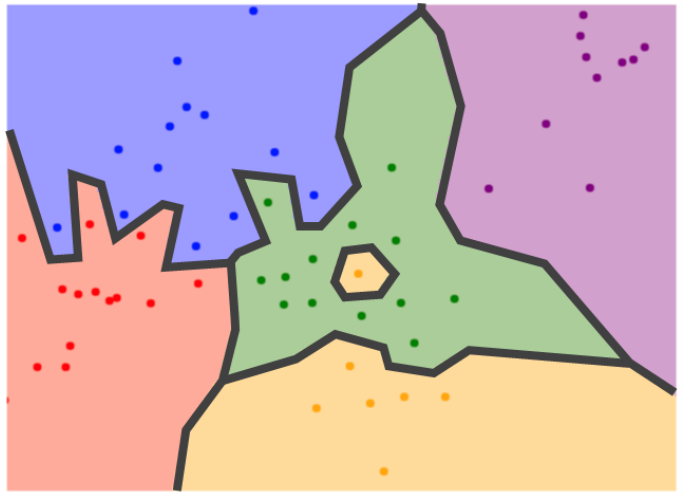
Pixels	Segments	Images	Videos	Advanced Topic
- Filtering	- Segmentation	- Recognition	- Motion	- Neural Networks
- Edges	- **Clustering**	- Detection	- Tracking	- CNNs for CV
- Features	- Resizing			

K-Nearest Neighbors (K-NN)

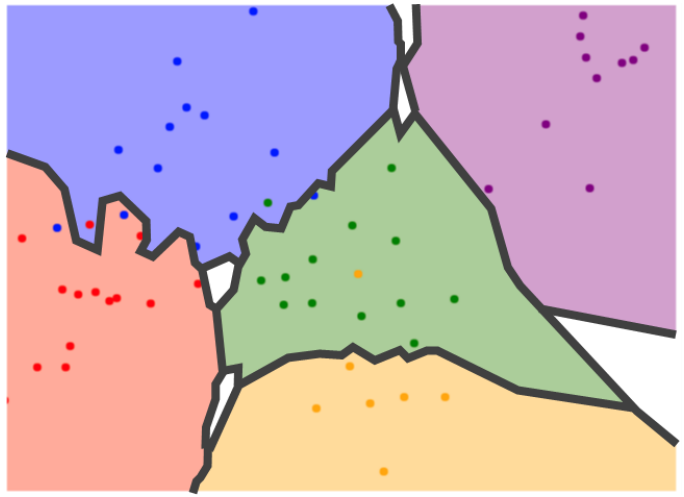
- Instead of finding the single closest image in the training set, we will **find the top k closest images**, and have them vote on the label of the test image

— NN method is the K-NN method where $K = 1$

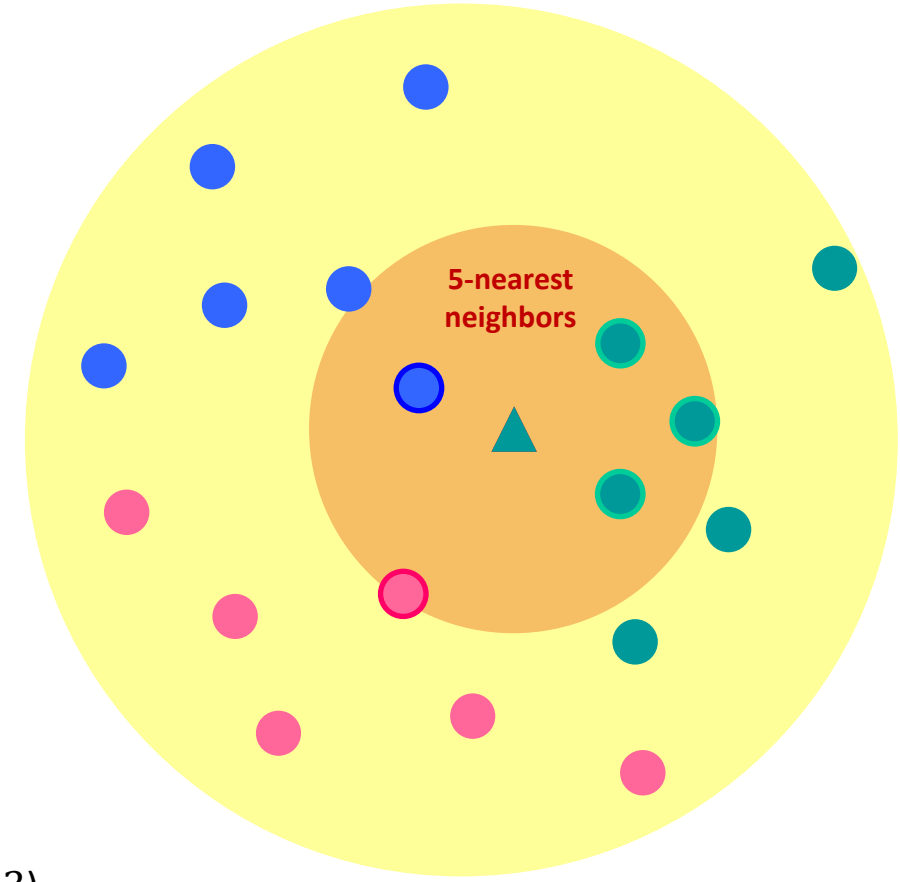
- **Higher values of k** have a smoothing effect that makes **the classifier more resistant** to outliers



Decision boundaries of NN classifier



Decision boundaries of K-NN classifier ($K = 3$)



Decision Tree: **Random Forest**

- A **random forest** is an ensemble of decision trees

— Randomness is incorporated via training set sampling and/or generation of the candidate binary tests

— The prediction of the random forest is obtained by averaging over all decision trees

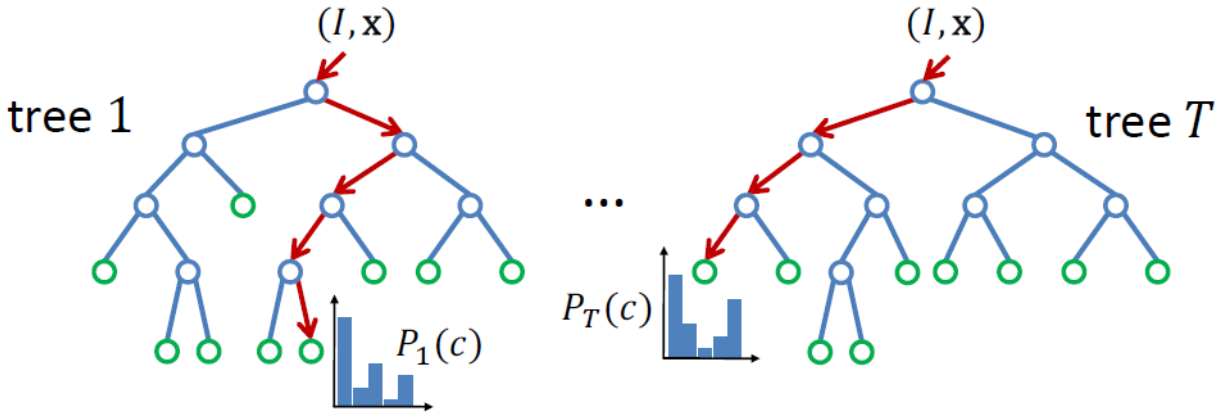


Figure 4. **Randomized Decision Forests.** A forest is an ensemble of trees. Each tree consists of split nodes (blue) and leaf nodes (green). The red arrows indicate the different paths that might be taken by different trees for a particular input.

J. Shotton et al., Real-Time Human Pose Recognition in Parts from Single Depth Images, **CVPR 2011**

Example of Random Forest: **Human Pose Recognition**

- **Kinect** allows users of Microsoft's Xbox 360 console to interact with games using natural body motions instead of a traditional handheld controller
- The pose (joint positions) of the user is predicted using a random forest trained on depth features

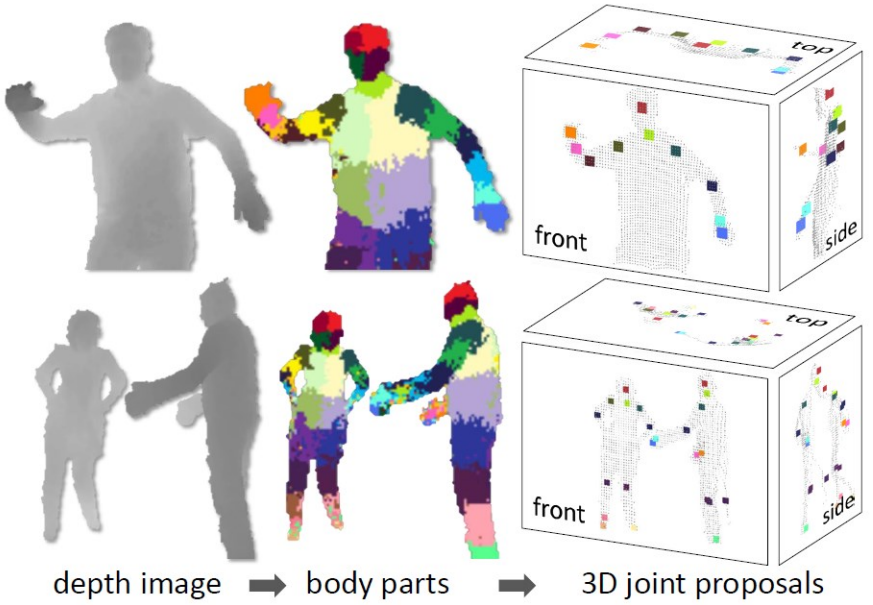


Figure 1. **Overview.** From an single input depth image, a per-pixel body part distribution is inferred. (Colors indicate the most likely part labels at each pixel, and correspond in the joint proposals).

J. Shotton et al., Real-Time Human Pose Recognition in Parts from Single Depth Images, **CVPR 2011**

Roadmap: Image Processing & Vision

Pixels	Segments	Images	Videos	Advanced Topic
- Filtering	- Segmentation	- **Recognition**	- Motion	- Neural Networks
- Edges	- Clustering	- **Detection**	- Tracking	- CNNs for CV
- Features	- Resizing			

Object Detection: **Sliding Window**

- **Slide** a fixed-sized detection window across the image and evaluate the classifier on each window

— We have to search over **scale** as well

— We may also have to search to search over **aspect ratios**

Is there a car?

An example on KITTI dataset benchmark

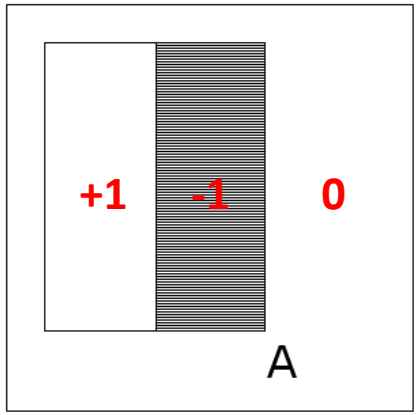
Face Detection: Viola-Jones – ① **Haar Feature**

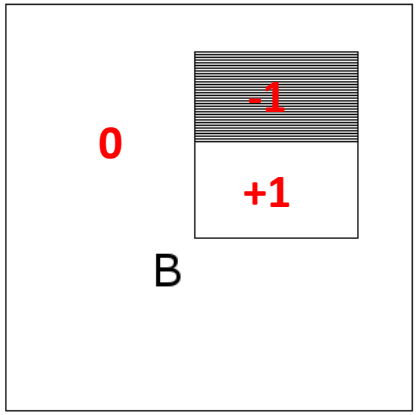
- A **rectangular feature** is computed by summing up pixel values within rectangular regions and then differencing those region sums

- **All human faces** share **similar** properties. These regularities may be matched using Haar features

— The eye region is darker than the upper-cheeks

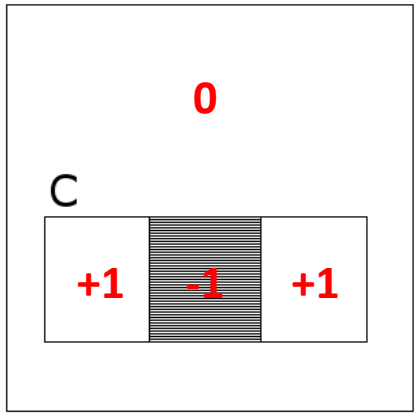
— The nose bridge region is brighter than the eyes

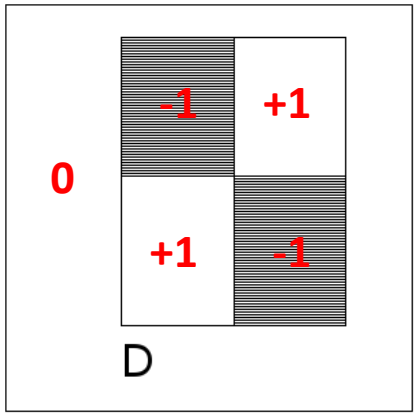




Haar Feature that looks similar to the eye region which is darker than the upper cheeks is applied onto a face

Haar Feature that looks similar to the bridge of the nose is applied onto the face



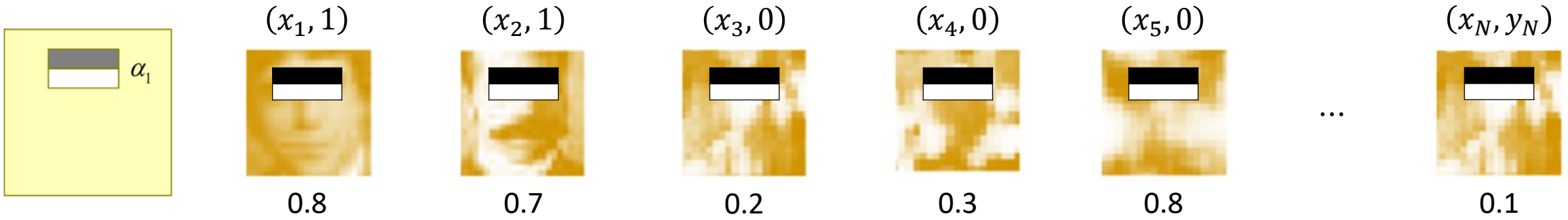


P. Viola and M. Jones, Rapid Object Detection using A Boosted Cascade of Simple Features, **CVPR 2001**

Face Detection: Viola-Jones – ③ **AdaBoost**

- Object detection framework employs AdaBoost to both select the best features and to train classifiers that use them

— **AdaBoost:** It constructs a strong classifier as a linear combination of weighted simple weak classifiers



Weak classifier: $h_j = \begin{cases} 1, & if\ f_j(x) > \theta_j \\ 0, & otherwise \end{cases}$

P. Viola and M. Jones, Rapid Object Detection using A Boosted Cascade of Simple Features, **CVPR 2001**

Detecting **A Single Object**

- Treat the localization as a regression problem

Object Detection

Feature Extraction

4096-d
feature vector

Classification

Bbox
Regression

What

Class Scores
- *Cat*: 0.9
- *Dog*: 0.05
- *Car*: 0.01
- ⋮

Minimize
the difference

Correct Label
- *Cat*

Where

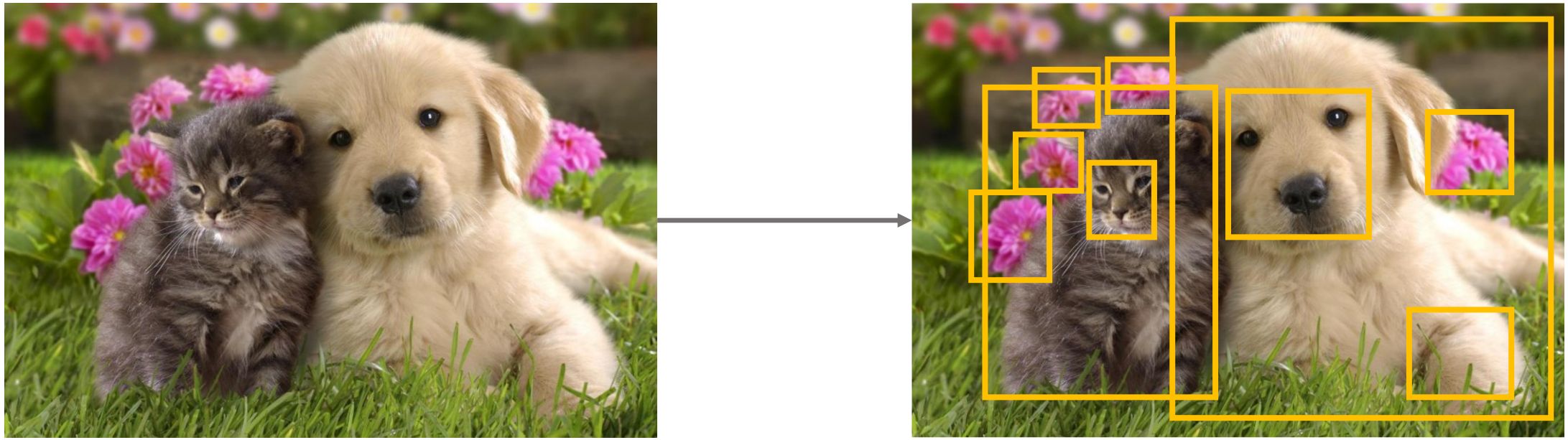
Minimize
the difference

Box Coordinates
(x, y, w, h)

Correct Box
(x', y', w', h')

Region Proposals

- Object region proposal algorithms generate **a short list of regions that have generic object-like properties**
- The object detector then considers **a small set of candidate regions only**, instead of exhaustive sliding window search



B. Alexe et al., Measuring the Objectness of Image Windows, **IEEE TPAMI 2012**
J. R. R. Uijlings et al., Selective Search for Object Recognition**, IJCV 2013**
M. M. Cheng et al., BING: Binarized Normed Gradients for Objectness Estimation at 300fps, **CVPR 2014**
C. L. Zitnick and P. Dollar, Edge Boxes: Locating Object Proposals from Edges, **ECCV 2014**

Roadmap: Image Processing & Vision

Pixels	Segments	Images	Videos	Advanced Topic
- Filtering	- Segmentation	- Recognition	- **Motion**	- Neural Networks
- Edges	- Clustering	- Detection	- Tracking	- CNNs for CV
- Features	- Resizing			

Optical Flow

- **Optical flow** is the apparent motion of brightness patterns in the image

- **Applications**

— Image and video stabilization in digital cameras, camcorders

— Motion-compensated video compression

— Motion segmentation

— Image registration

— Action recognition

Optical flow (motion vector)

Optical flow visualization

Optical Flow: **Key Assumptions**

- **Brightness Constancy**

— Brightness constancy for intensity images

— It allows for pixel to pixel comparison (not image feature comparison)

- **Small Motion**

— Pixels only move a little bit

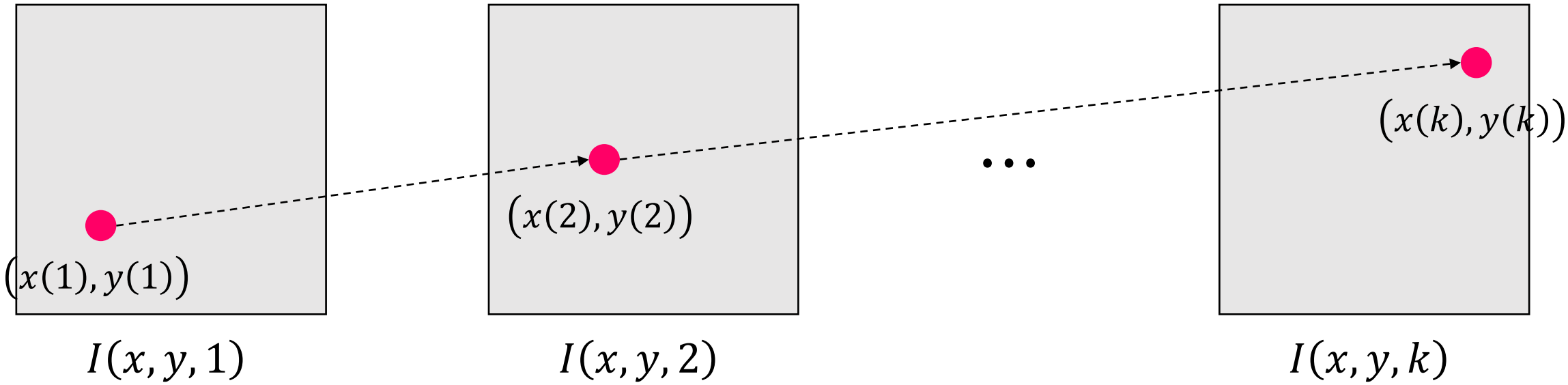
— It can be formulated as a linearization of the brightness constancy constraint

- **Approach:** Look for nearby pixels with the same color

Key Assumptions: ① Brightness Constancy

- Scene point moves through image sequence
- Brightness of the pixel point remains the same (constant, C)

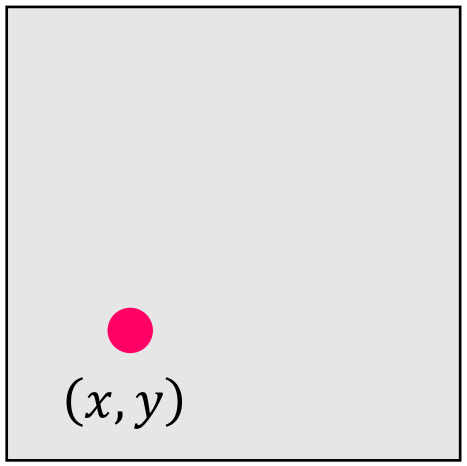
$$I(x(t), y(t), t) = C \qquad where \;\; t = 1, \cdots, k$$



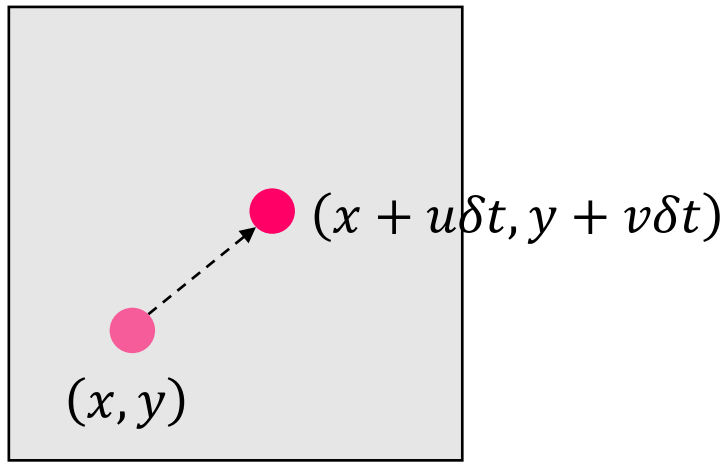
Key Assumptions: ② Small Motion

- Optical flow (velocities): (u, v)
- Displacement: $(\delta x, \delta y) = (u\delta t, v\delta t)$

$$I(x + u\delta t, y + v\delta t) = I(x, y, t)$$



$I(x, y, t)$



$I(x, y, t + \delta t)$

- For small space-time step, brightness of a point is the same

Lucas-Kanade Method

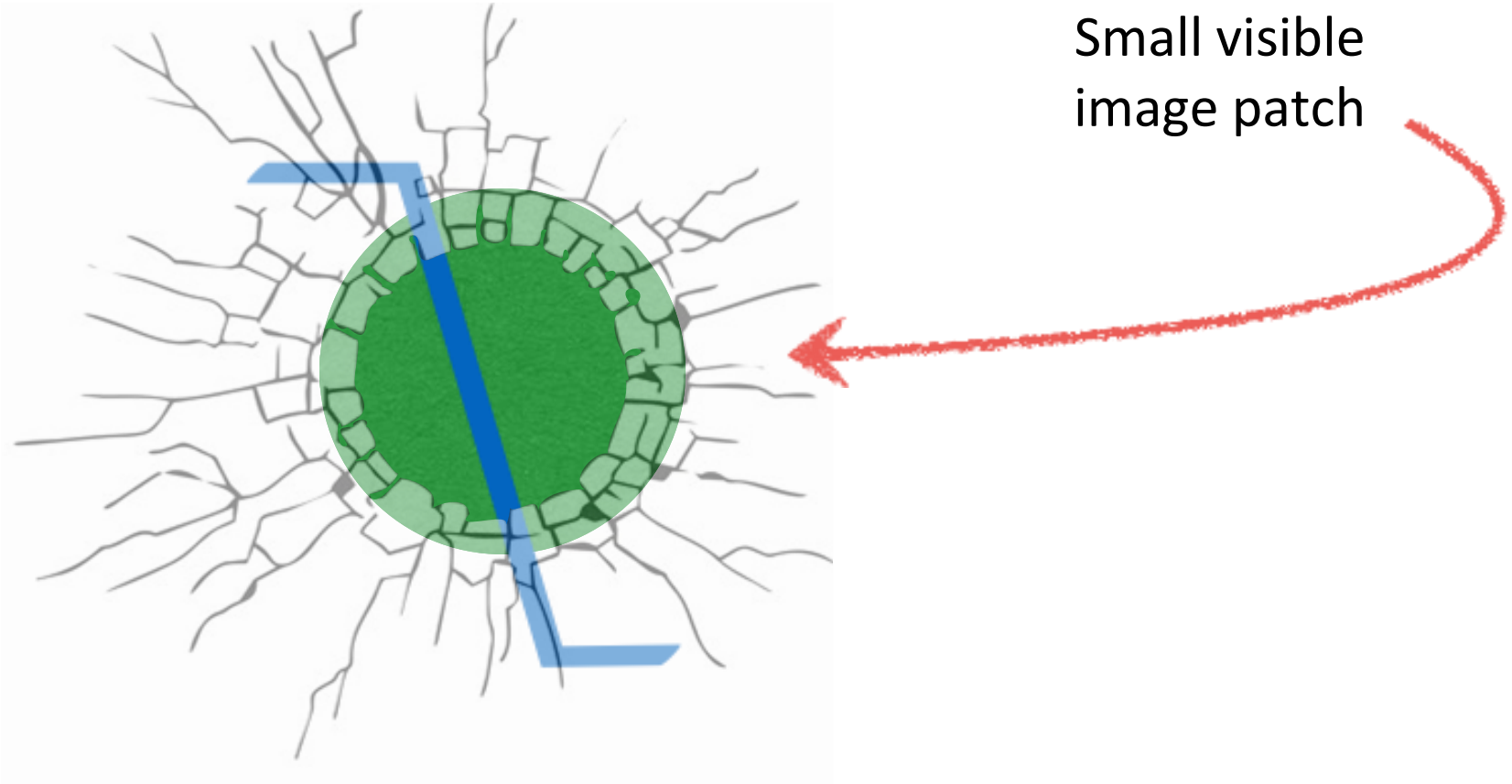
- Considering all n points in the window

$$I_{x_1}u + I_{y_1}v = -I_{t_1}$$
$$I_{x_2}u + I_{y_2}v = -I_{t_2}$$
$$\vdots$$
$$I_{x_n}u + I_{y_n}v = -I_{t_n}$$

- It can be written as the matrix equation form, $\mathbf{Ax} = \mathbf{b}$:

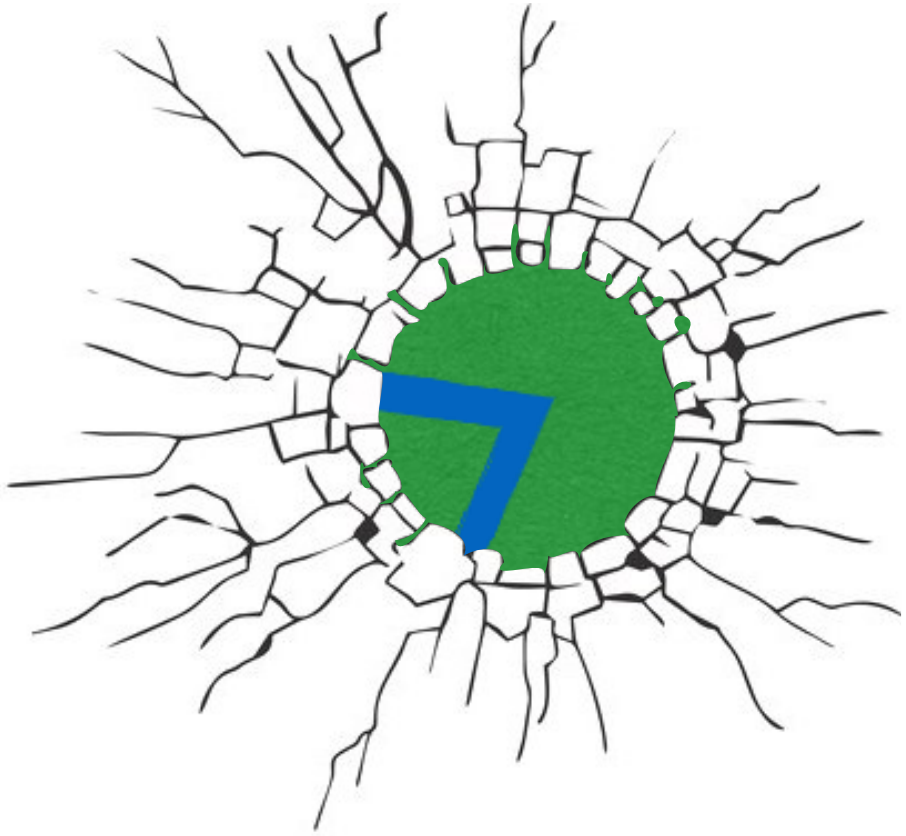
$$\mathbf{A} = \begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \vdots \\ I_{x_n} & I_{y_n} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} u \\ v \end{bmatrix} \qquad \mathbf{b} = -\begin{bmatrix} I_{t_1} \\ I_{t_2} \\ \vdots \\ I_{t_n} \end{bmatrix}$$

Aperture Problem

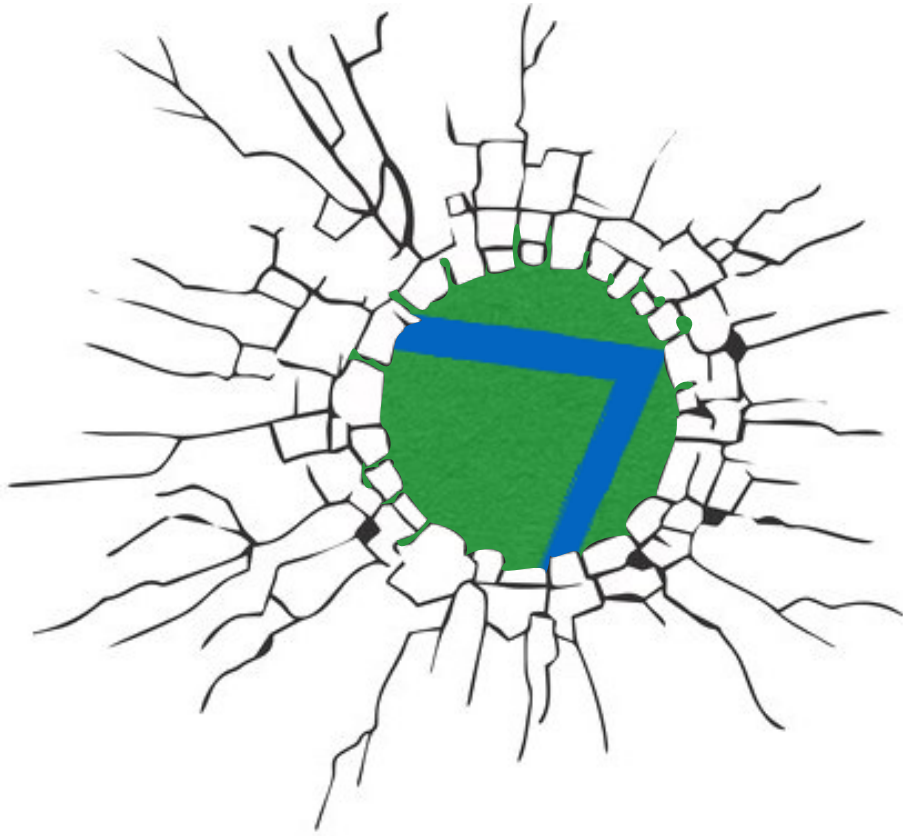


Solution: Aperture Problem

- Patches with different gradients to the avoid aperture problem

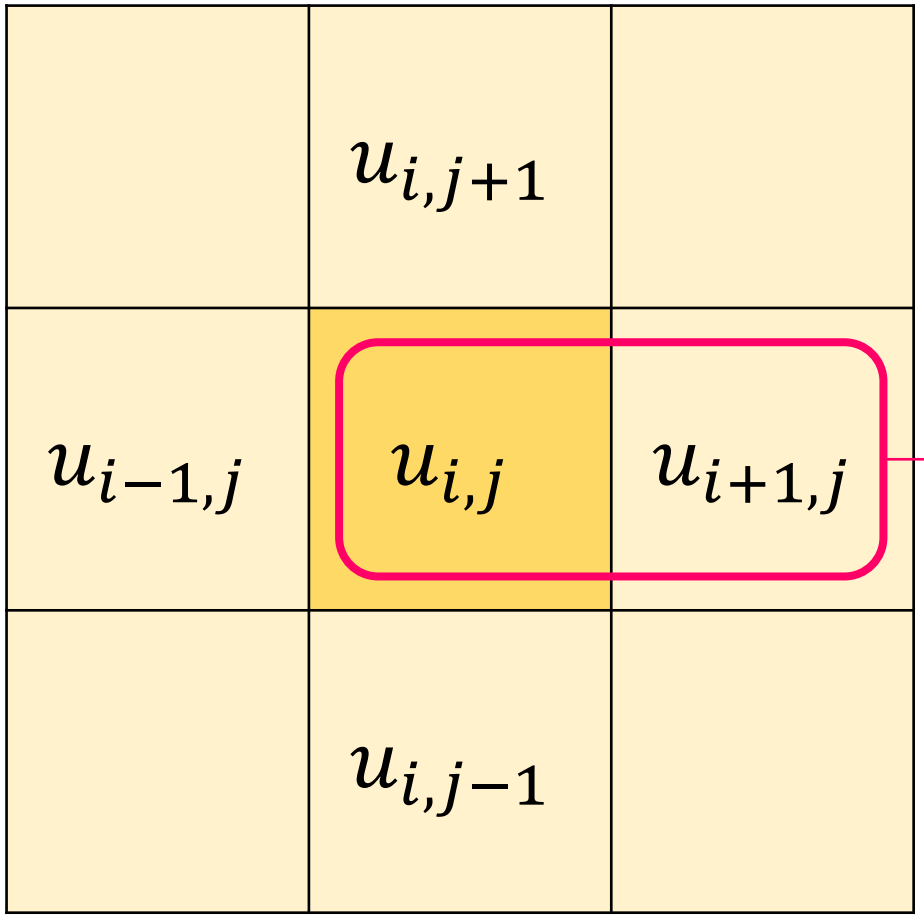


t



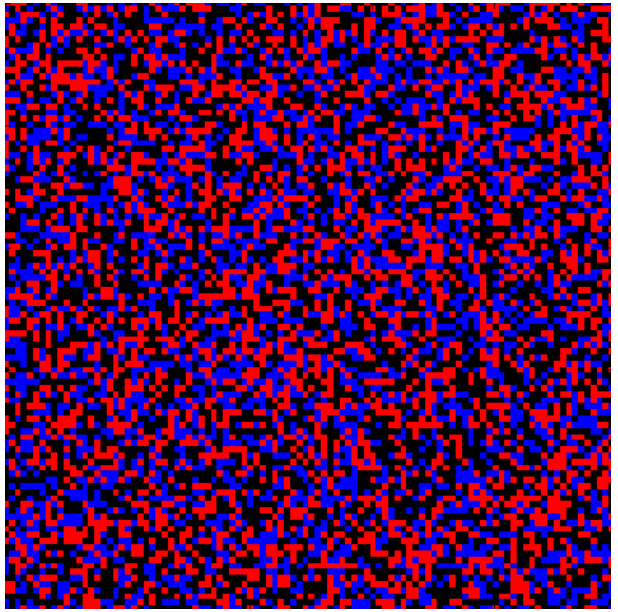
$t + 1$

Horn-Schunck: **Smooth Flow Field**



Optical flow in x-direction (u-component)

$$\min_{u}\left[u_{i,j} - u_{i+1,j}\right]^2$$



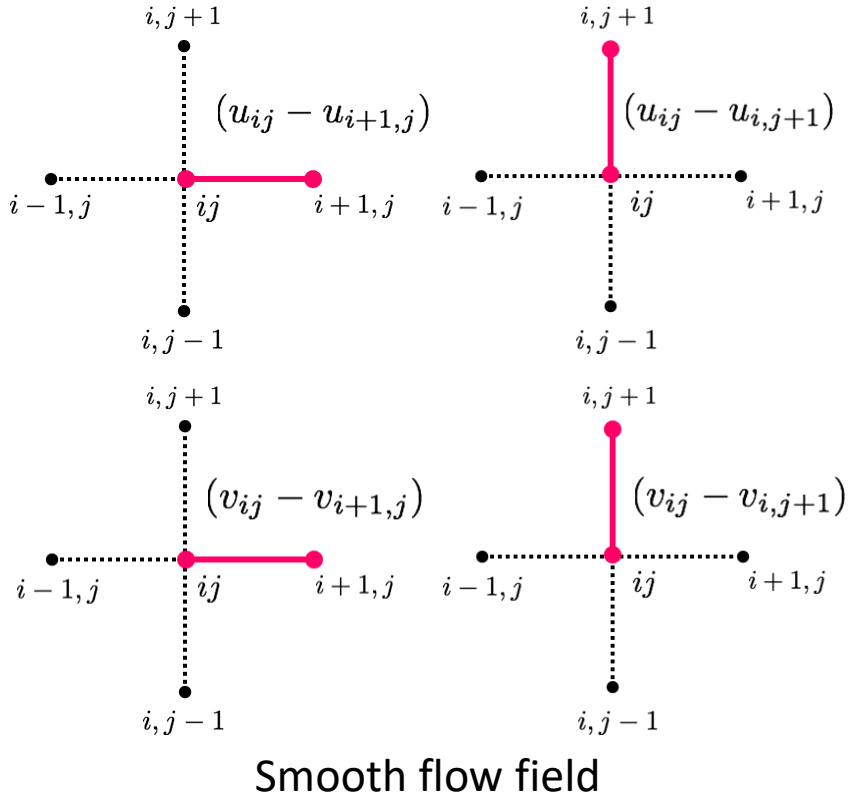
Large differences
between neighboring flows

Small differences
between neighboring flows

Horn-Schunck

- **Horn-Schunck Optical Flow:**

$$\min_{u,v} \sum_{i,j} E_s(i,j) + \lambda E_b(i,j)$$



Smooth flow field

— Brightness Constancy:

$$E_b(i,j) = \left[I_x(i,j)u_{i,j} + I_y(i,j)v_{i,j} + I_t(i,j)\right]^2$$

— Smooth Flow Field:

$$E_s(i,j) = \frac{1}{4}\left[\left(u_{i,j} - u_{i+1,j}\right)^2 + \left(u_{i,j} - u_{i,j+1}\right)^2 + \left(v_{i,j} - v_{i+1,j}\right)^2 + \left(v_{i,j} - v_{i,j+1}\right)^2\right]$$

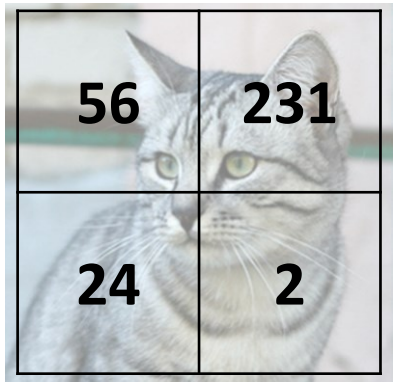
Roadmap: Image Processing & Vision

Pixels	Segments	Images	Videos	Advanced Topic
- Filtering	- Segmentation	- Recognition	- Motion	- **Neural Networks**
- Edges	- Clustering	- Detection	- Tracking	- CNNs for CV
- Features	- Resizing			

Example: 2x2 Image Classification (3 Classes)

- Classification: Cat / Dog / Bird

$$f(\mathbf{x}, \mathbf{W}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$



Input image,
$\mathbf{x} \in \mathbb{R}^{2\times 2}$

1.2	1.5	1.1	2.0
-0.5	0.3	0.1	0.0
0.1	0.05	-0.1	-0.2

$\mathbf{W} \in \mathbb{R}^{3\times 4}$

$\times$

56
231
24
2

$\mathbf{x} \in \mathbb{R}^{4\times 1}$

$+$

3.1
1.2
-1.5

$\mathbf{b} \in \mathbb{R}^{3\times 1}$

$=$

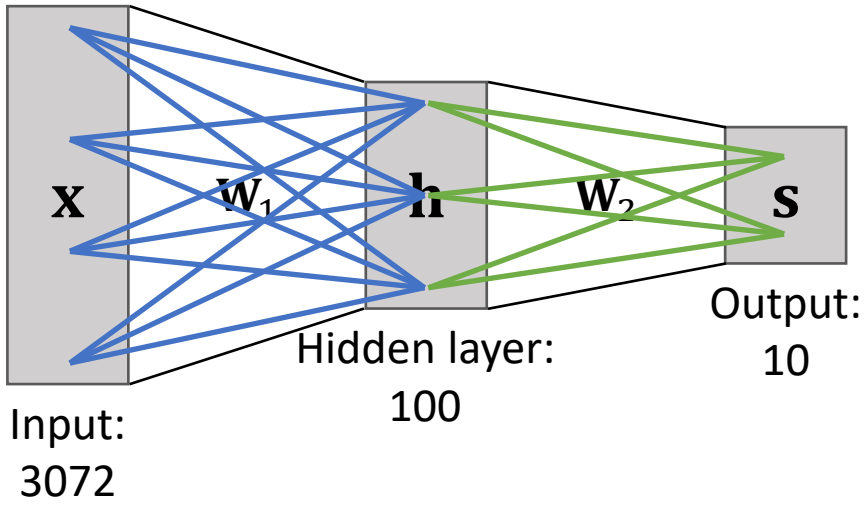
447.2	Cat
44.9	Dog
12.85	Bird

Score,
$f(\mathbf{x}; \mathbf{W}, \mathbf{b})$

Neural Networks: **Multi-Layer Perceptron (MLP)**

- Input: $\mathbf{x} \in \mathbb{R}^{D \times 1}$ $where$ $D = 3072$
- Output: $f(\mathbf{x}) \in \mathbb{R}^{C \times 1}$ $where$ $C = 10$
- **Linear Classifier:** $f(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$
- **2-layer Neural Network:** $f(\mathbf{x}) = \mathbf{W}_2 \max(0, \mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2$

All elements of $\mathbf{x}$ affect all elements of $\mathbf{h}$

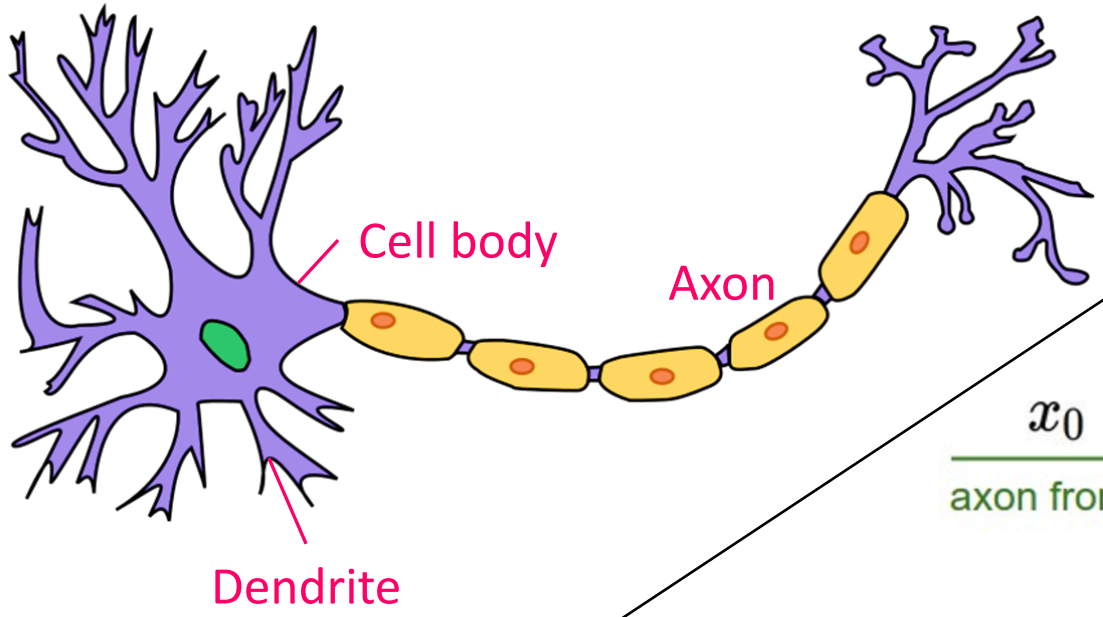


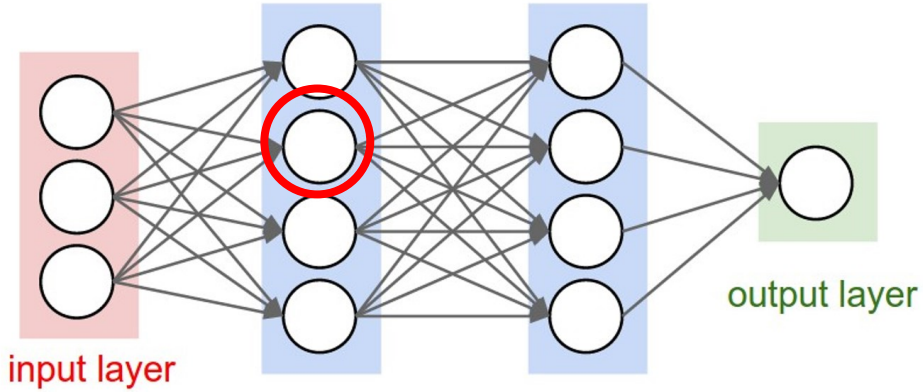
All elements of $\mathbf{h}$ affect all elements of $\mathbf{s}$

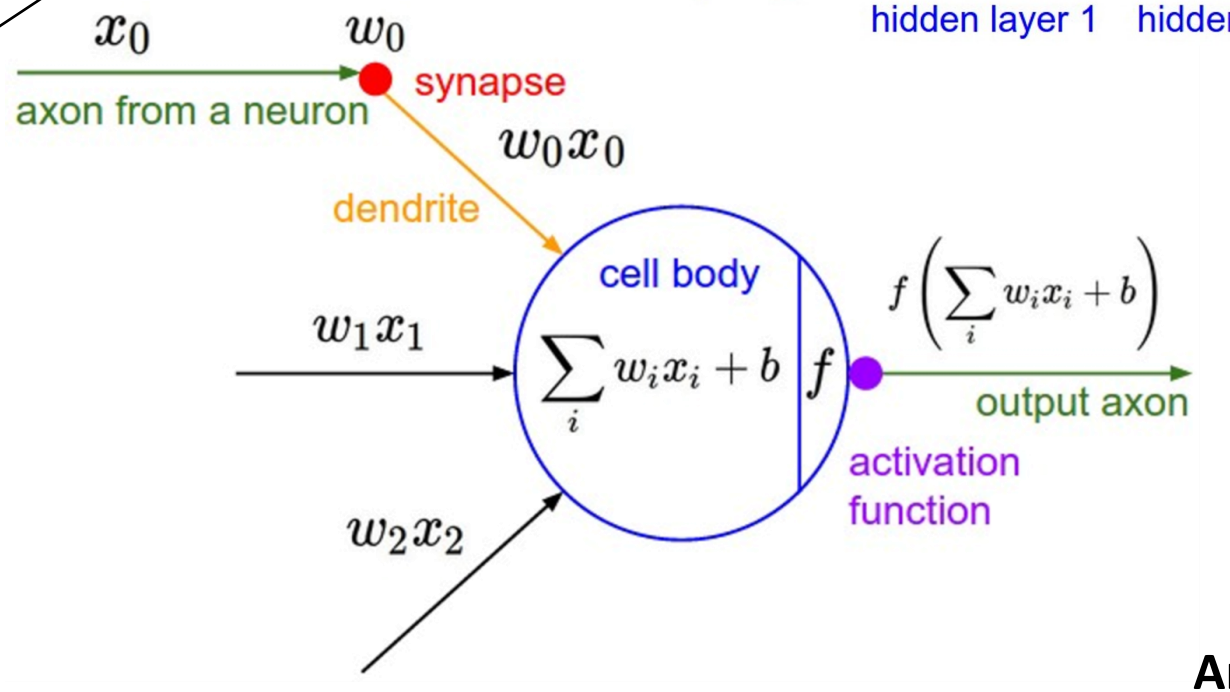
Fully-connected neural network (MLP)

Biological Neuron vs. Artificial Neuron

Biological Neuron







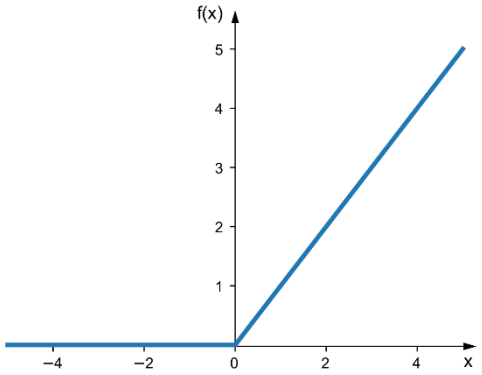
Artificial Neuron

Activation Functions in ANNs

- In ANNs, **the activation function** of a node defines **the output of that node** given an input or set of inputs. It is called **nonlinearities**

— The function ReLU (Rectified Linear Unit): $ReLU(z) = \max(0, z)$

$$f(\mathbf{x}) = \mathbf{W}_2 \max(0, \mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2$$

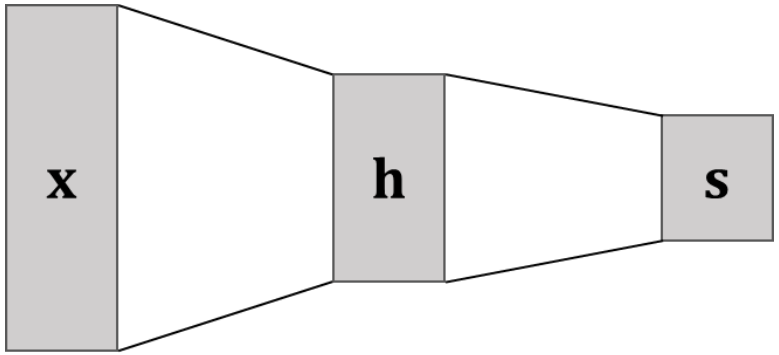


- If we build a neural network without activation function, it ends up with a linear classifier

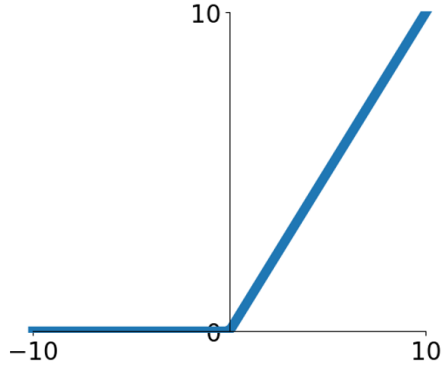
$$\begin{aligned} f(\mathbf{x}) &= \mathbf{W}_2(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2 \\ &= \mathbf{W}_2 \mathbf{W}_1 \mathbf{x} + (\mathbf{W}_2 \mathbf{b}_1 + \mathbf{b}_2) \end{aligned}$$

Components of **Neural Networks (MLP)**

Fully-Connected Layers (FC layers)



Activation Function

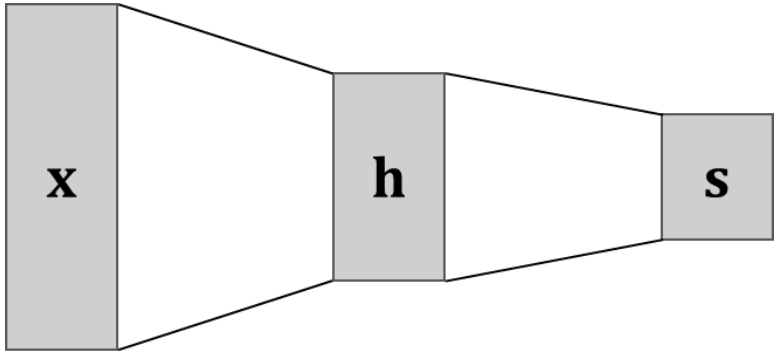


Roadmap: Image Processing & Vision

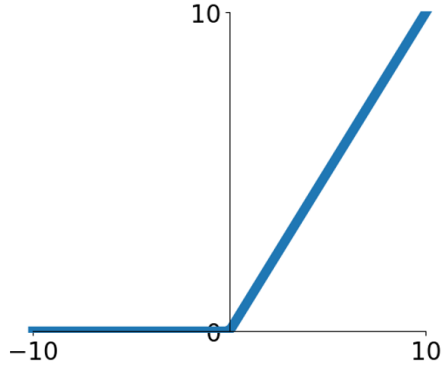
Pixels	Segments	Images	Videos	Advanced Topic
- Filtering	- Segmentation	- Recognition	- Motion	- Neural Networks
- Edges	- Clustering	- Detection	- Tracking	- **CNNs for CV**
- Features	- Resizing			

Components of **Convolutional Neural Networks** (CNNs)

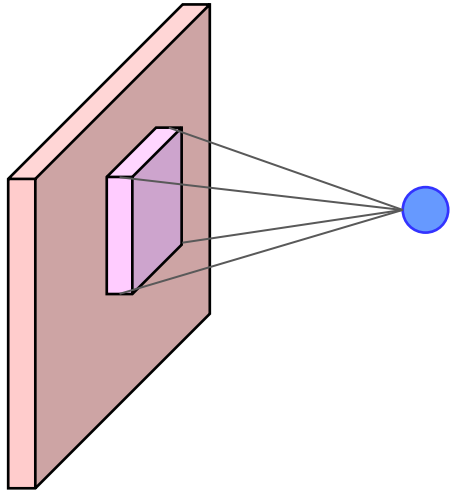
Fully-Connected Layers (FC layers)



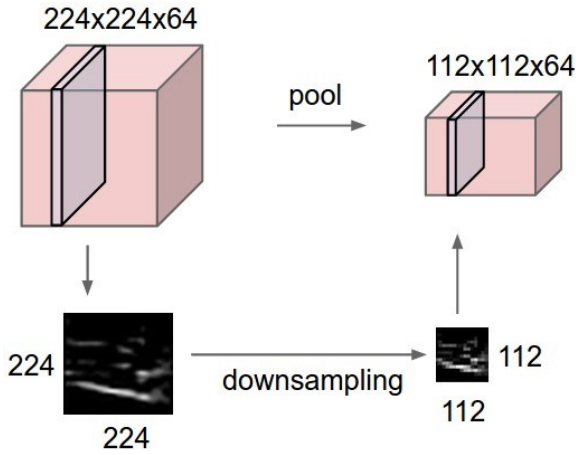
Activation Function



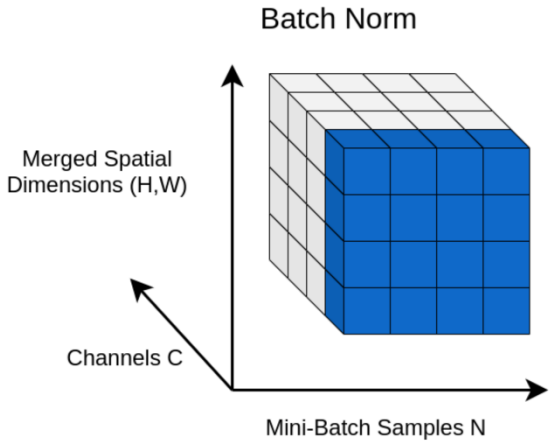
Convolution Layers



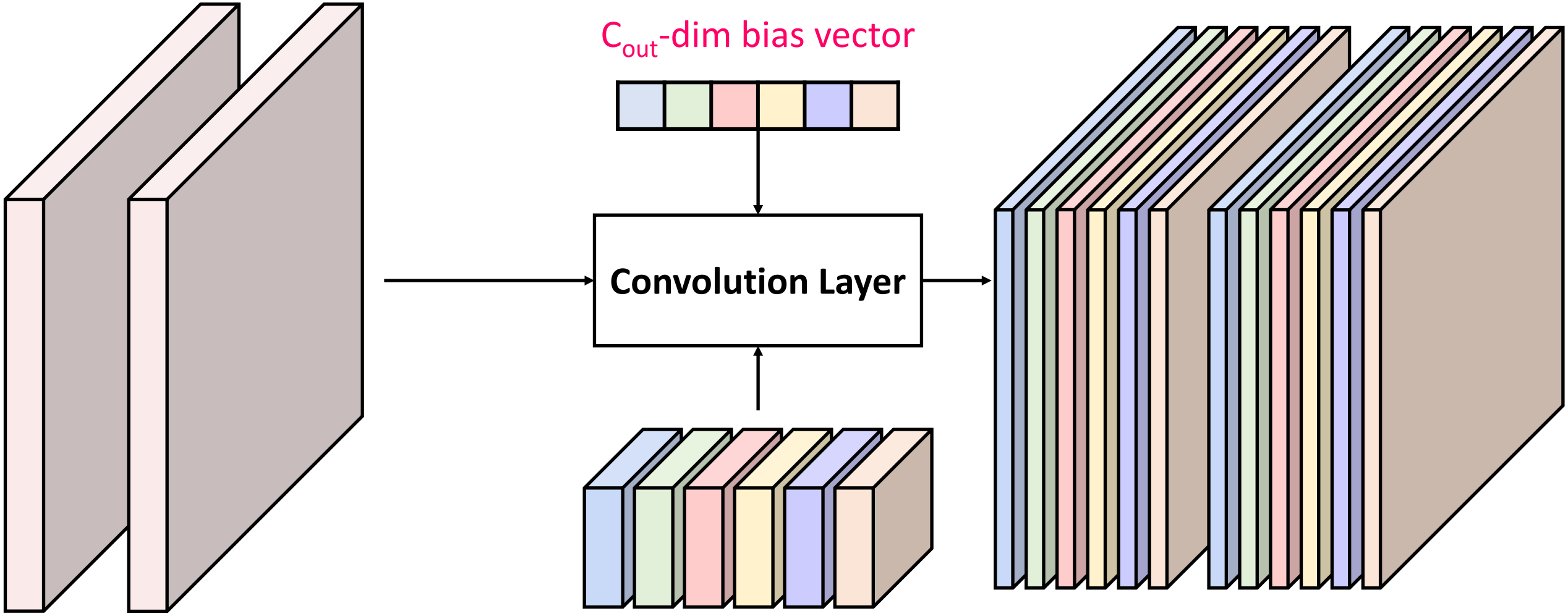
Pooling Layers



Normalization

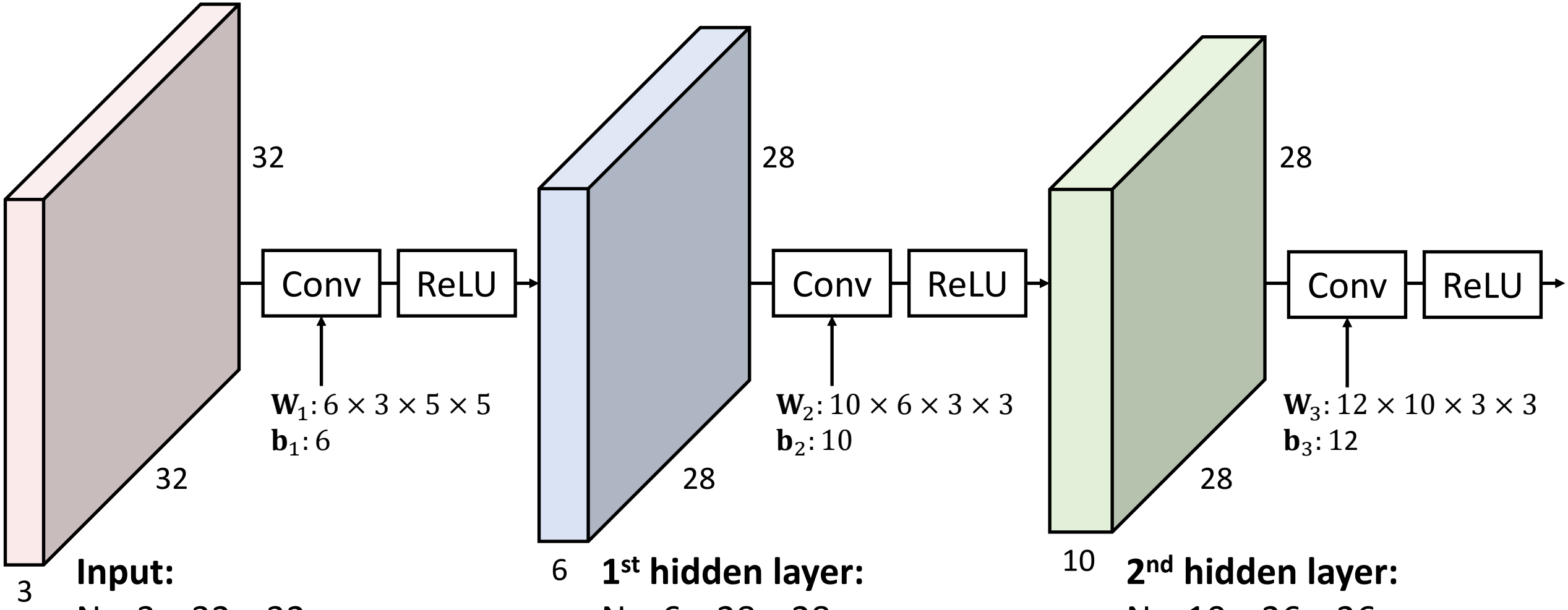


Architecture of CNNs: **Convolution Layer**



Architecture of CNNs: **Convolution Layer**

- Stacking Convolutions



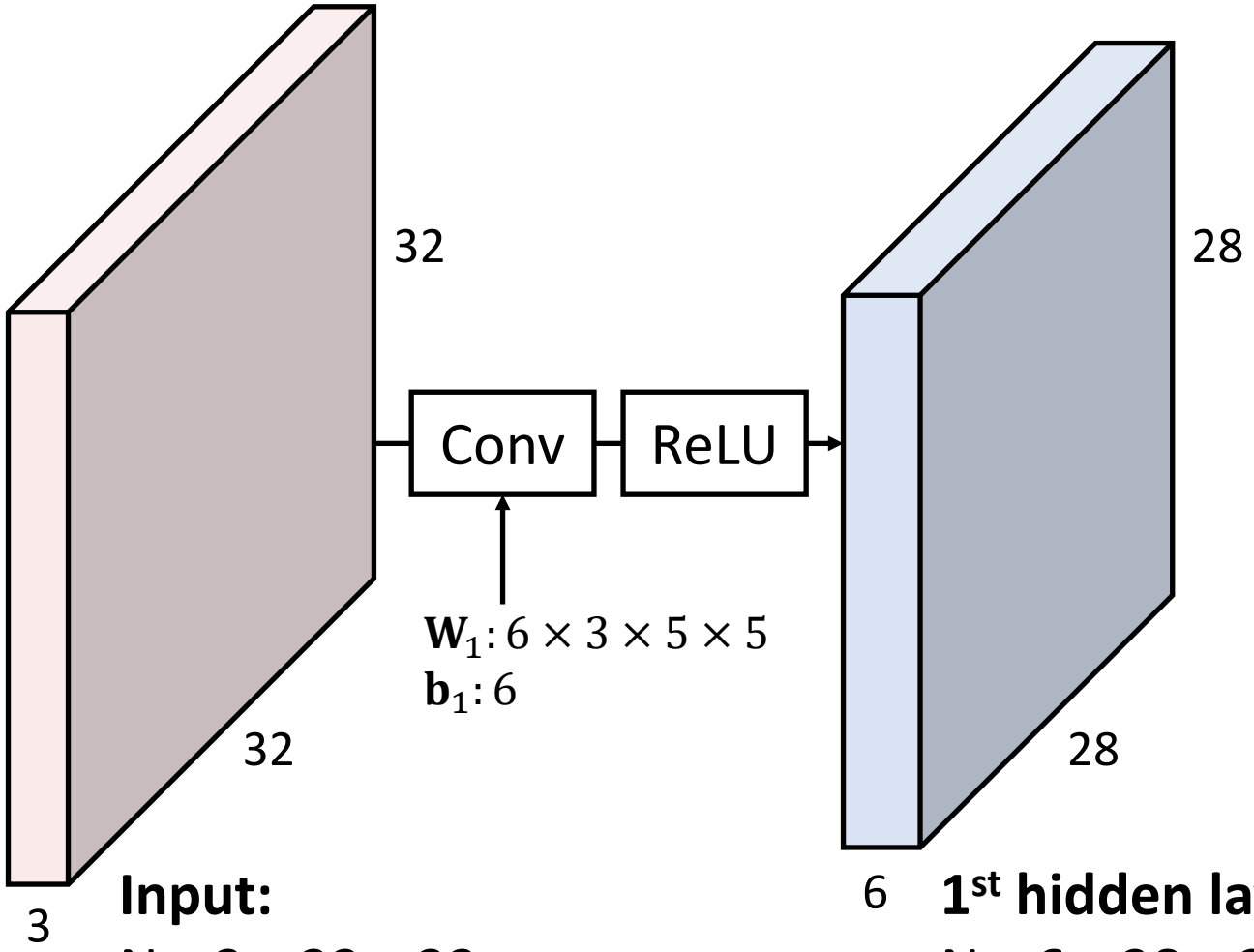
Input:
N x 3 x 32 x 32

1st hidden layer:
N x 6 x 28 x 28

2nd hidden layer:
N x 10 x 26 x 26

Architecture of CNNs: **Convolution Layer**

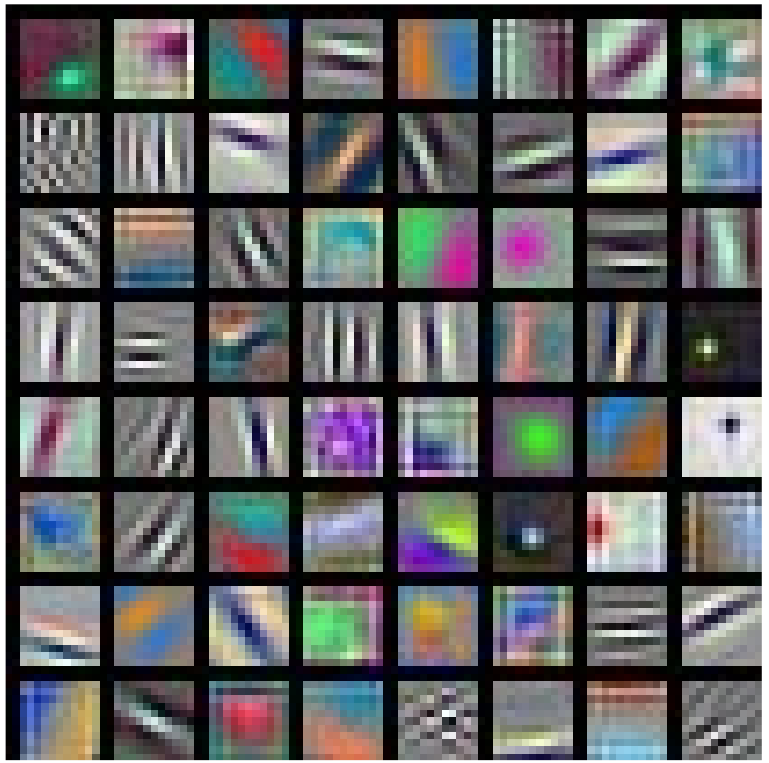
- What do convolution filters learn?



Input:
N x 3 x 32 x 32

1st hidden layer:
N x 6 x 28 x 28

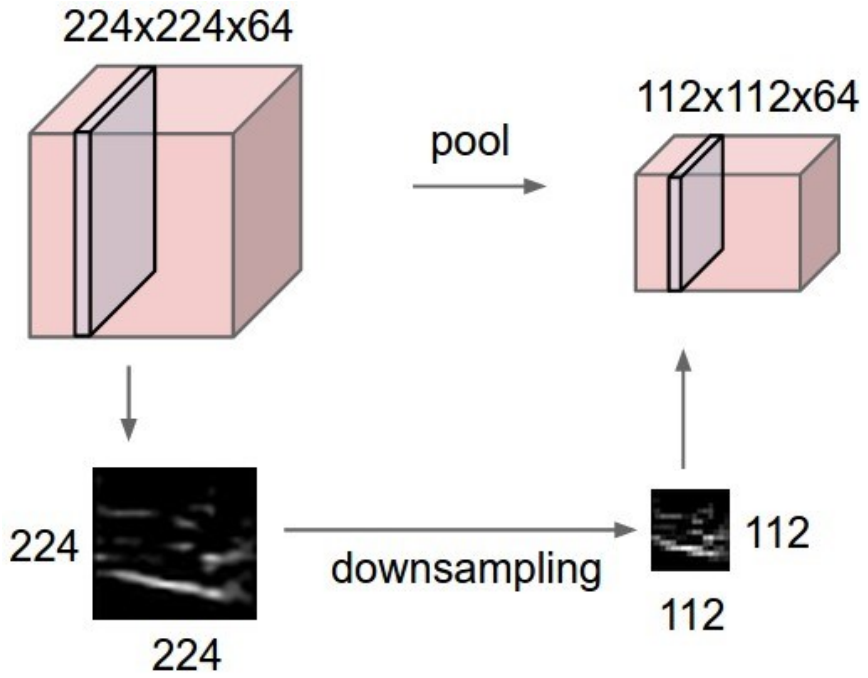
Convolution filter at 1st layer learns the local image templates
(e.g., oriented edges, opposite colors, etc.)



64 filters (3x11x11) in AlexNet

Architecture of CNNs: **Pooling Layer**

- Pooling layer is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network
- It operates independently on every depth slice of the input and resizes it spatially, using the MAX operation



Architecture of CNNs: **Batch Normalization**

- **Advantages of Batch Normalization:**

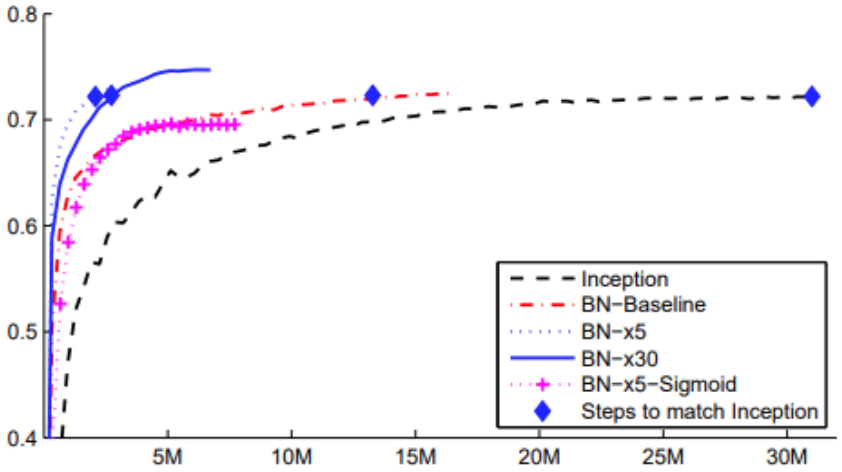
— Makes deep networks much easier to train

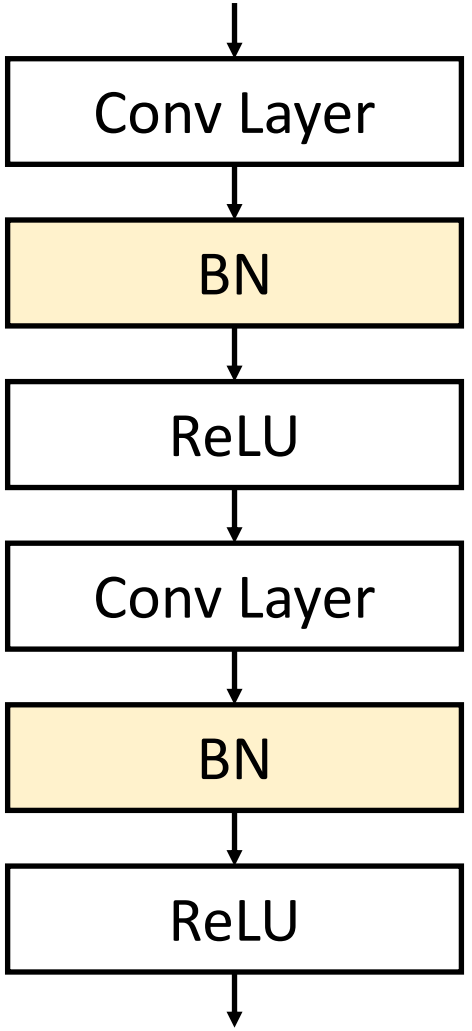
— Allows higher learning rates, faster convergence

— Networks become more robust to initialization

— Acts as regularization during training

— Zero overhead at test-time

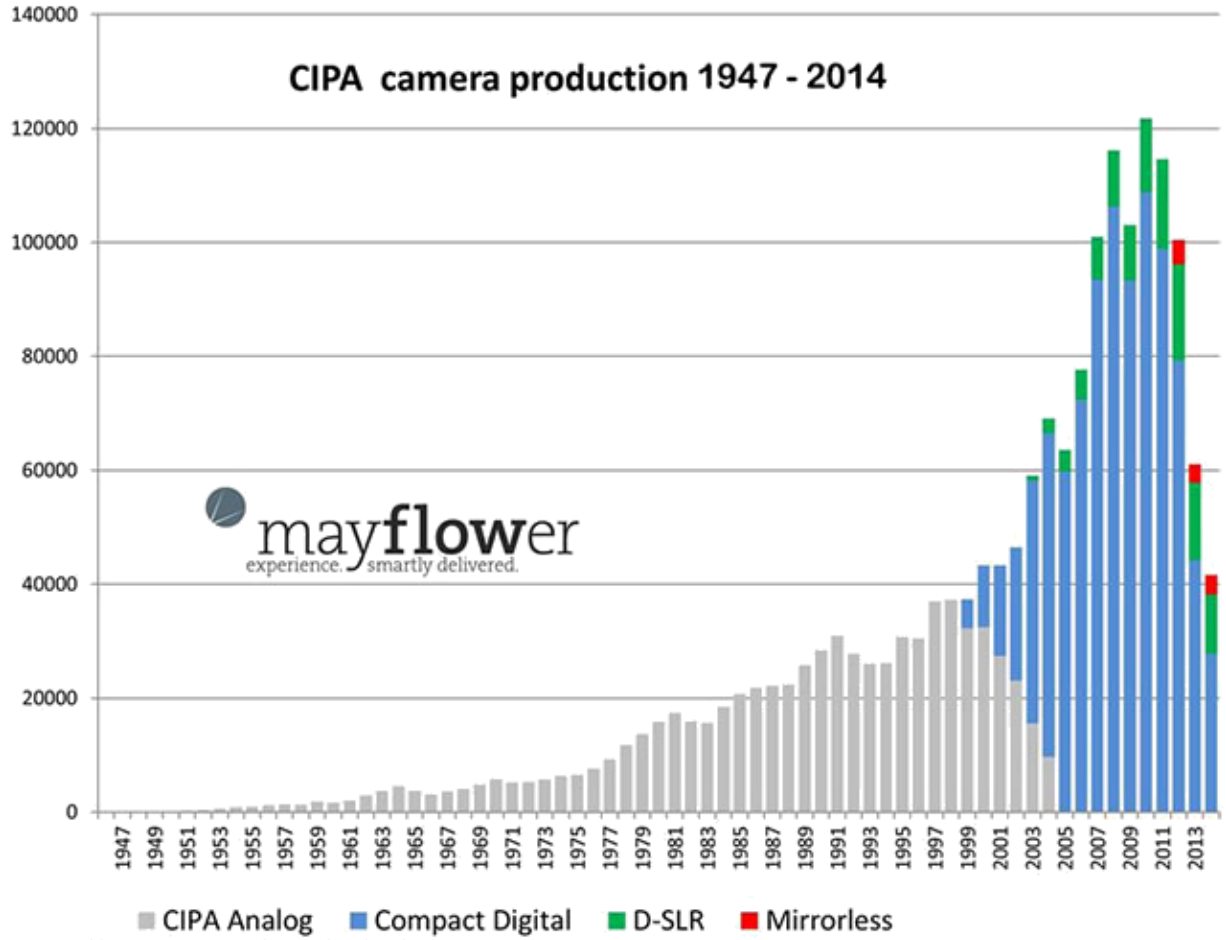


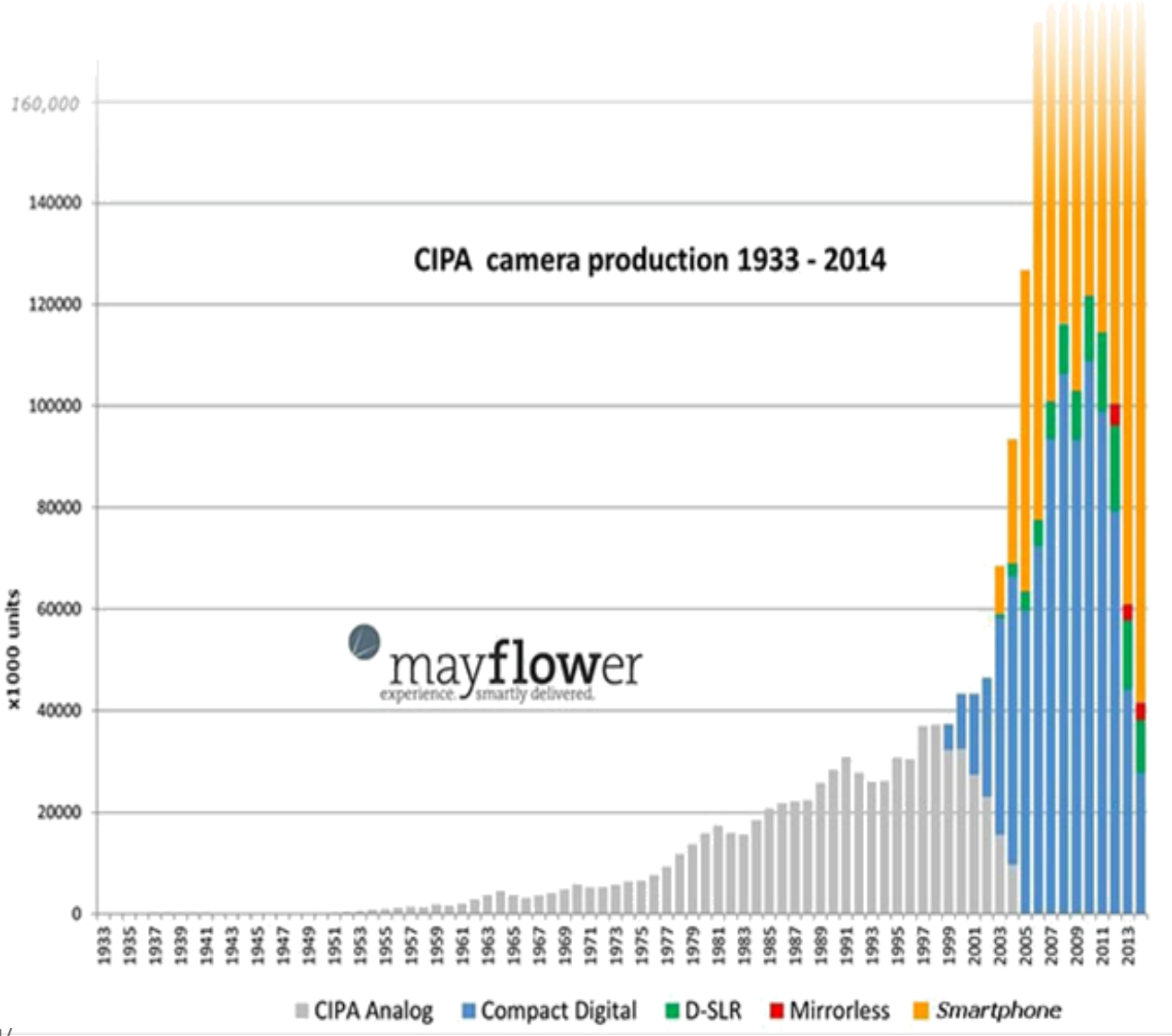


S. Ioffe and C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, **ICML 2015**

History of Camera Sales

- Camera Production



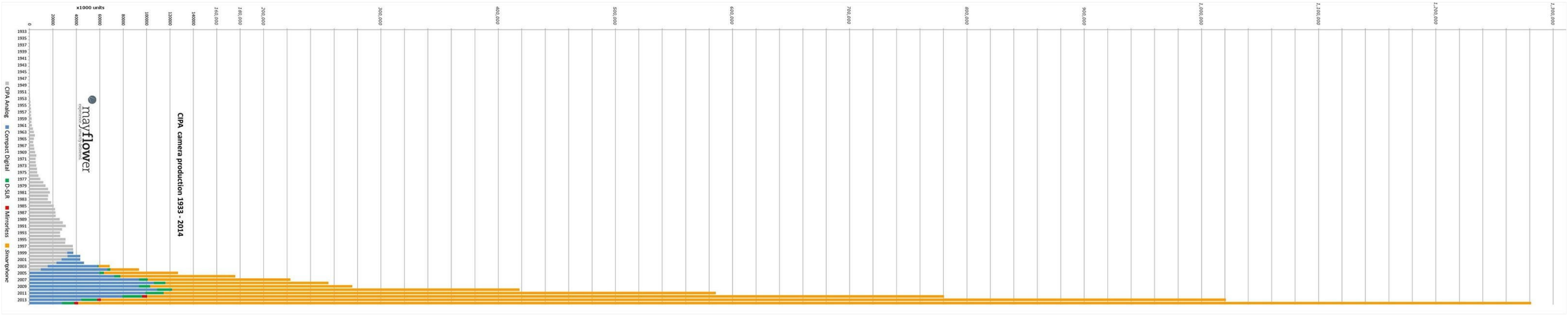


https://petapixel.com/2015/04/09/this-is-what-the-history-of-camera-sales-looks-like-with-smartphones-included/

History of Camera Sales

- Camera Production

12,000 1,300,000

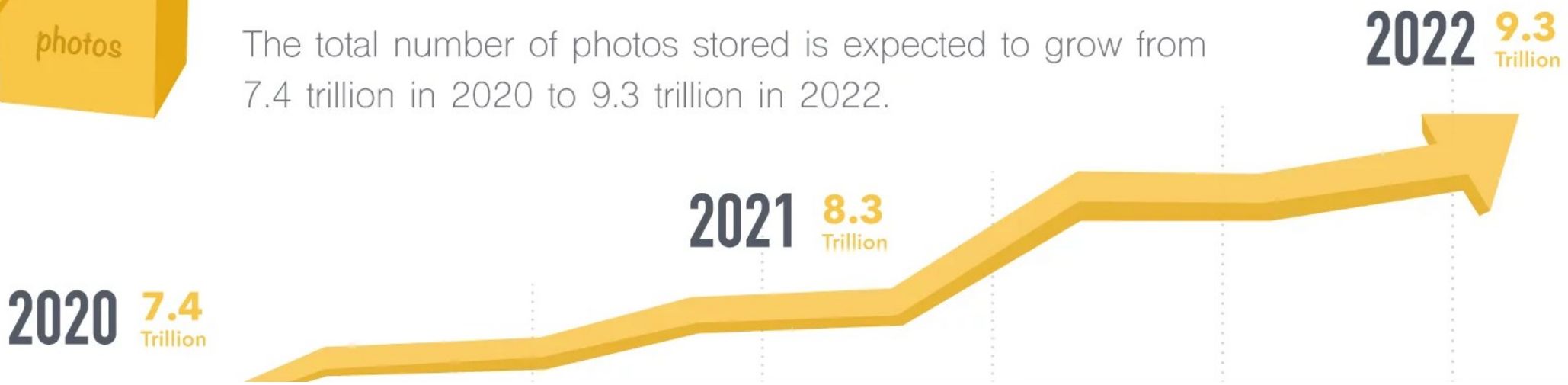


https://petapixel.com/2015/04/09/this-is-what-the-history-of-camera-sales-looks-like-with-smartphones-included/

How Many Photos Will be Taken in 2020?

7.4 Trillion Photos Stored

The total number of photos stored is expected to grow from 7.4 trillion in 2020 to 9.3 trillion in 2022.

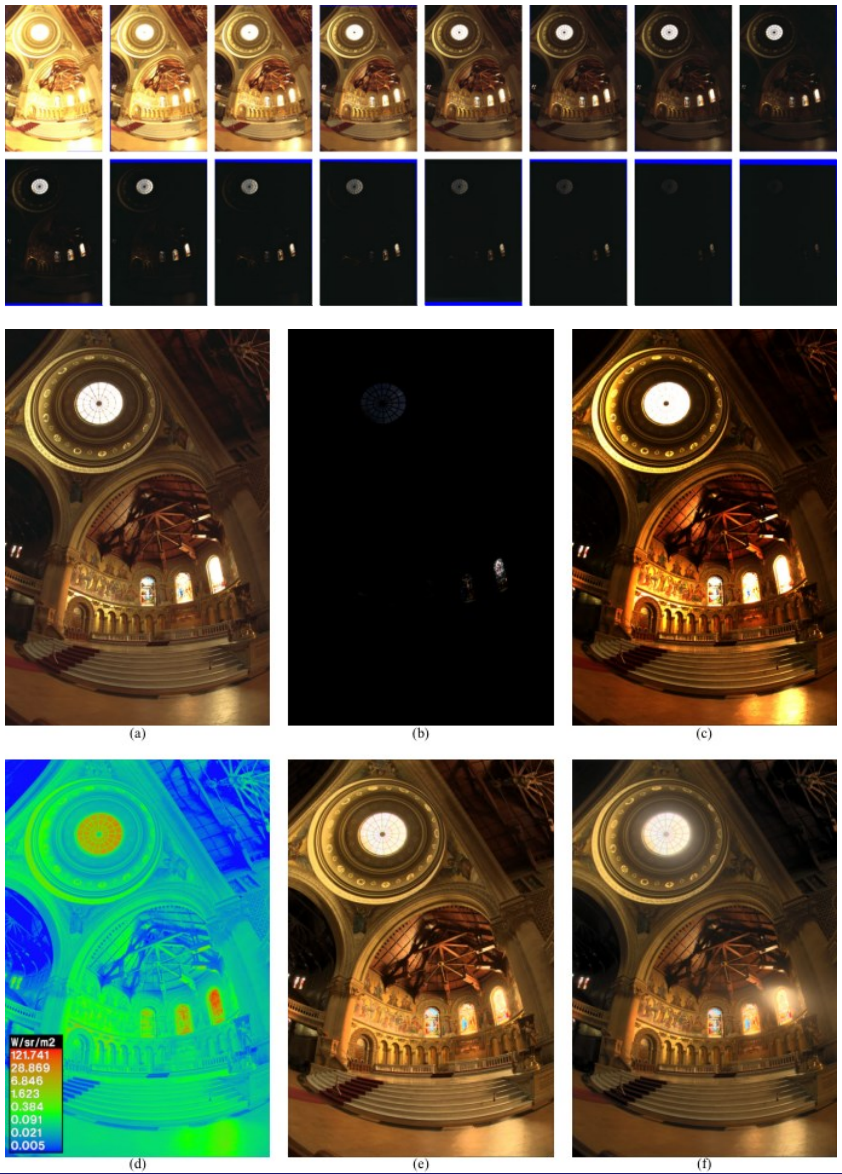


YEAR	2019	2020	2021	2022
PHOTOS STORED	6.4583 trillion	7.3754 trillion	8.2956 trillion	9.2944 trillion
% CHANGE	–	⇧ 14.2%	⇧ 12.5%	⇧ 12.0%

https://blog.mylio.com/how-many-photos-will-be-taken-in-2020/

Applications in Image Processing & Vision

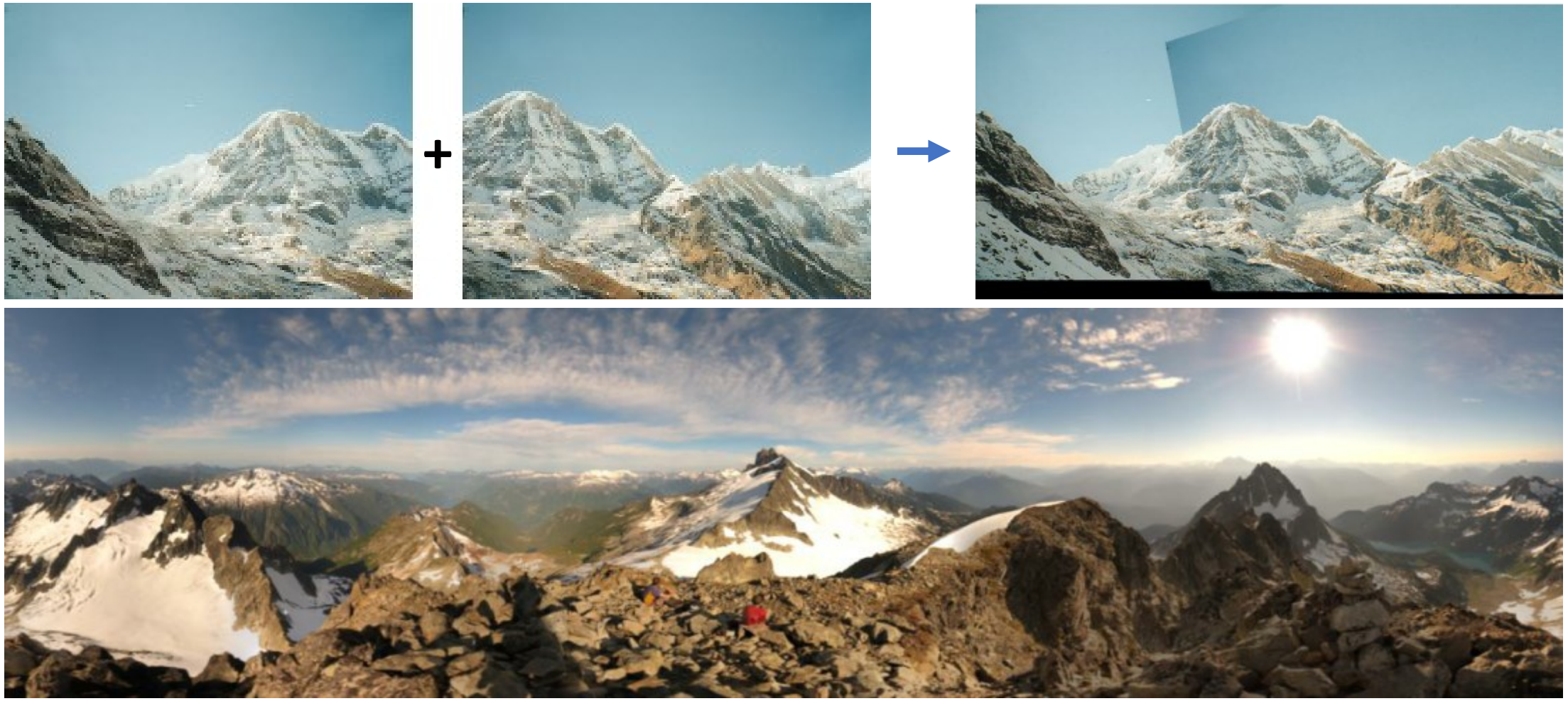
- High Dynamic Range (HDR)



L. Meylan and S. Susstrunk, High dynamic range image rendering with a retinex-based adaptive filter, **IEEE TIP 2006**
P. E. Debevec and J. Malik, Recovering high dynamic range radiance maps from phtographs, **ACM SIGGRAPH 1997**

Applications in Image Processing & Vision

- Image Stitching



M. Brown and D. G. Lowe, Automatic panoramic image stitching using invariant features, **IJCV 2007**

Applications in Image Processing & Vision

- Scene Completion

Original Image | Input | Scene Matches | Output

J. Hays and A. A. Effros, Scene completion using millions of photographs, **ACM SIGGRAPH 2007**

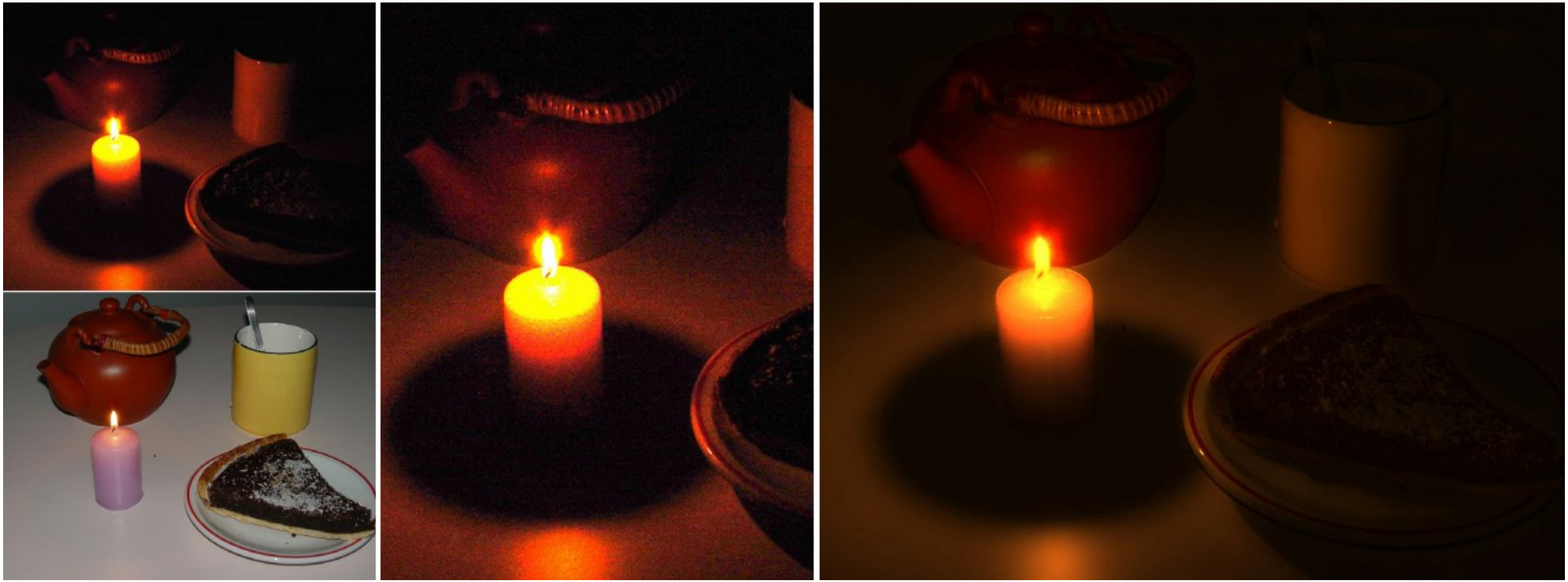
Applications in Image Processing & Vision

- Scene-Relighting

P. Debevec, Virtual cinematography: Relighting through computation, **IEEE Computer 2006**

Applications in Image Processing & Vision

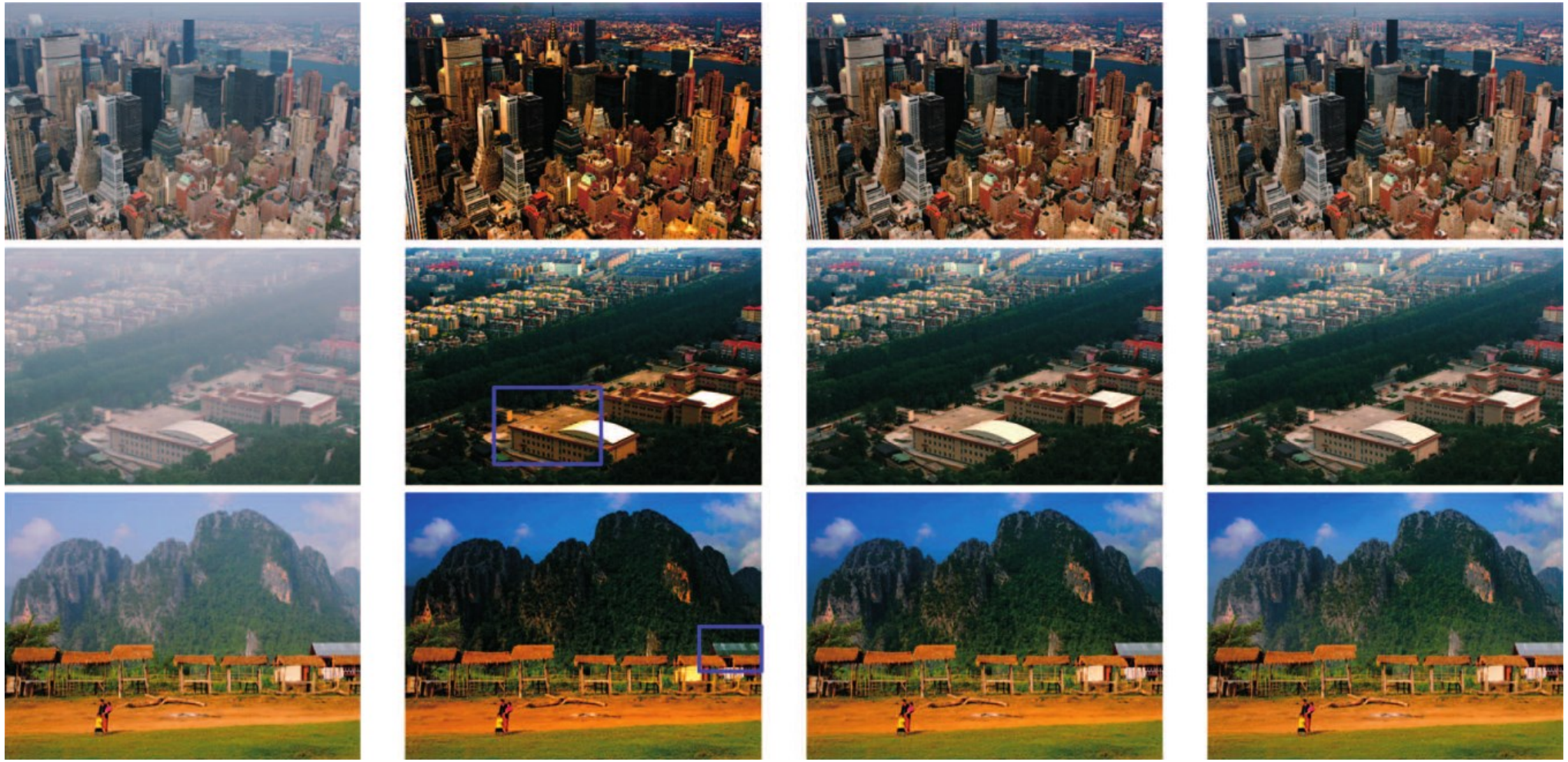
- Flash/No Flash Photography



E. Eisemann and F. Durand, Flash photography enhancement via intrinsic relighting, **ACM TOG 2004**

Applications in Image Processing & Vision

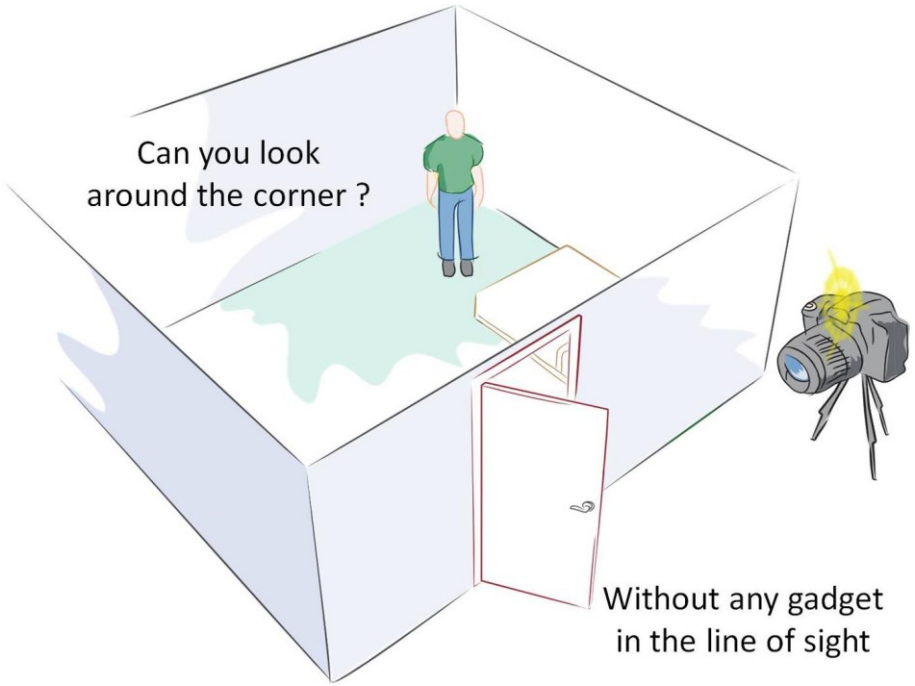
- De-haze



K. He et al., Single image haze removal using dark channel prior, **IEEE TPAMI 2010**

Applications in Image Processing & Vision

- Look Around Corners

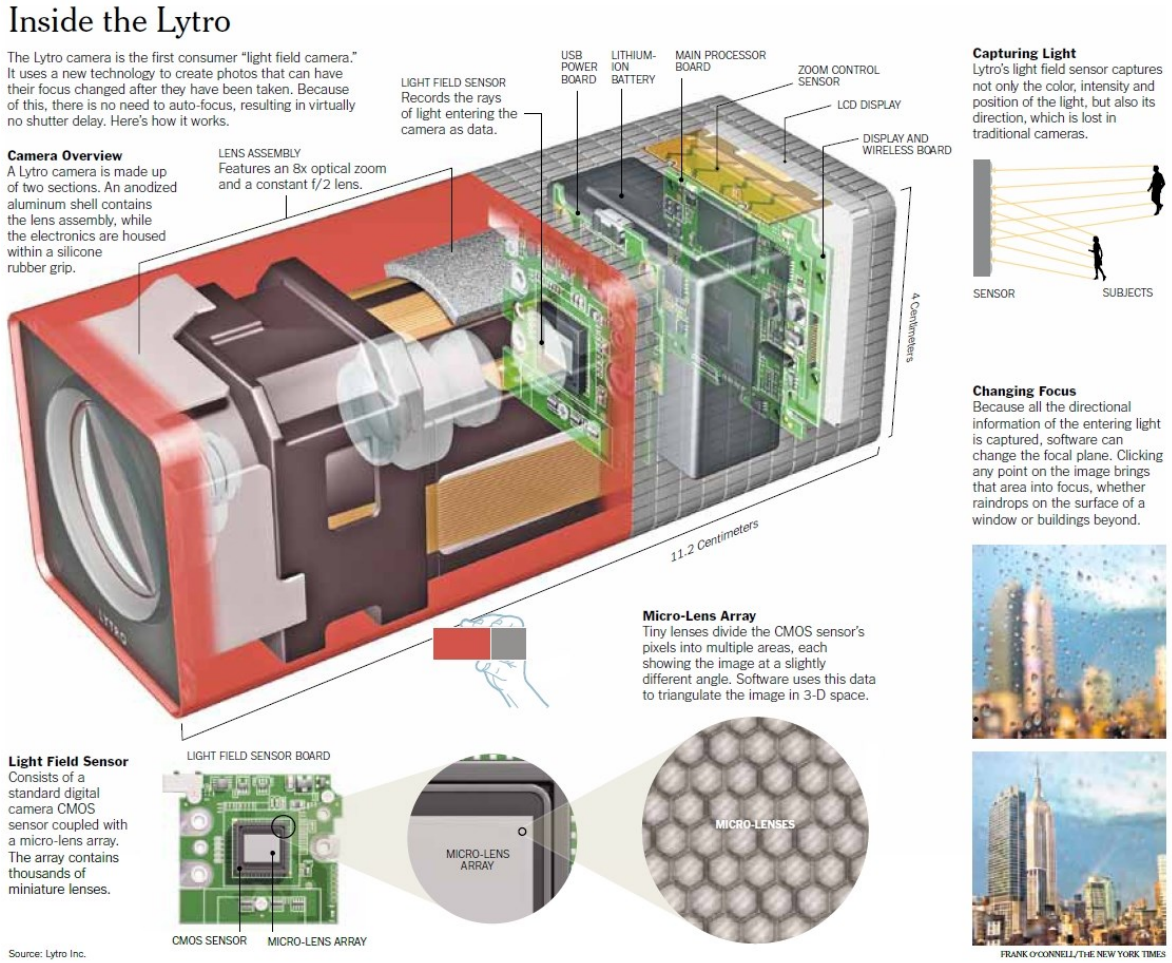


https://www.youtube.com/watch?v=JWDocXPy-iQ

A. Velten et al., Recovering three dimensional shape around a corner using ultra-fast Time-of-Flight imaging, **Nature Communications, 2012**

Applications in Image Processing & Vision

- Light-Field Cameras for Plenoptic Imaging

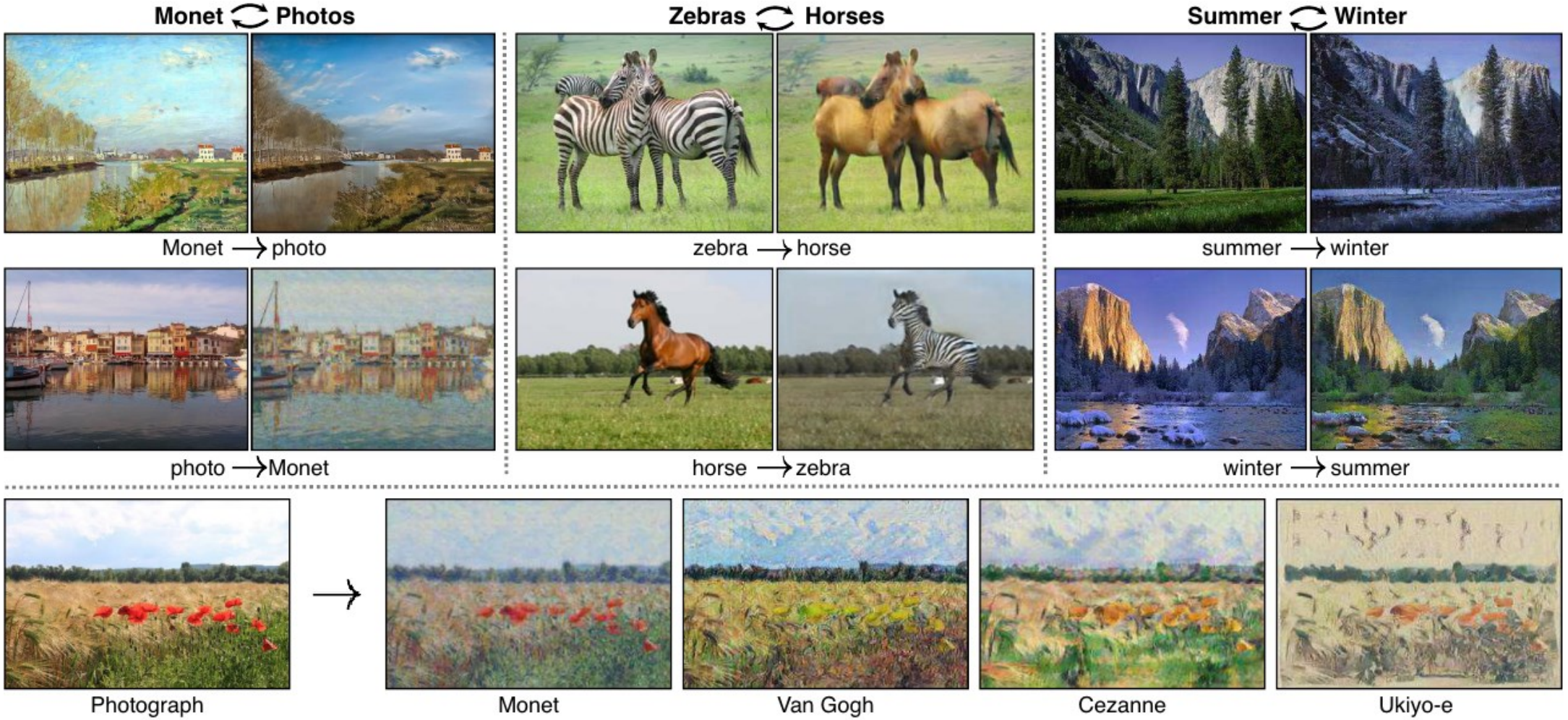


R. Ng et al., Light field photography with a hand-held Plenoptic camera, **Doctoral dissertation, 2005**

https://www.youtube.com/watch?v=ciqhoFr9940

Applications in Image Processing & Vision **with AI**

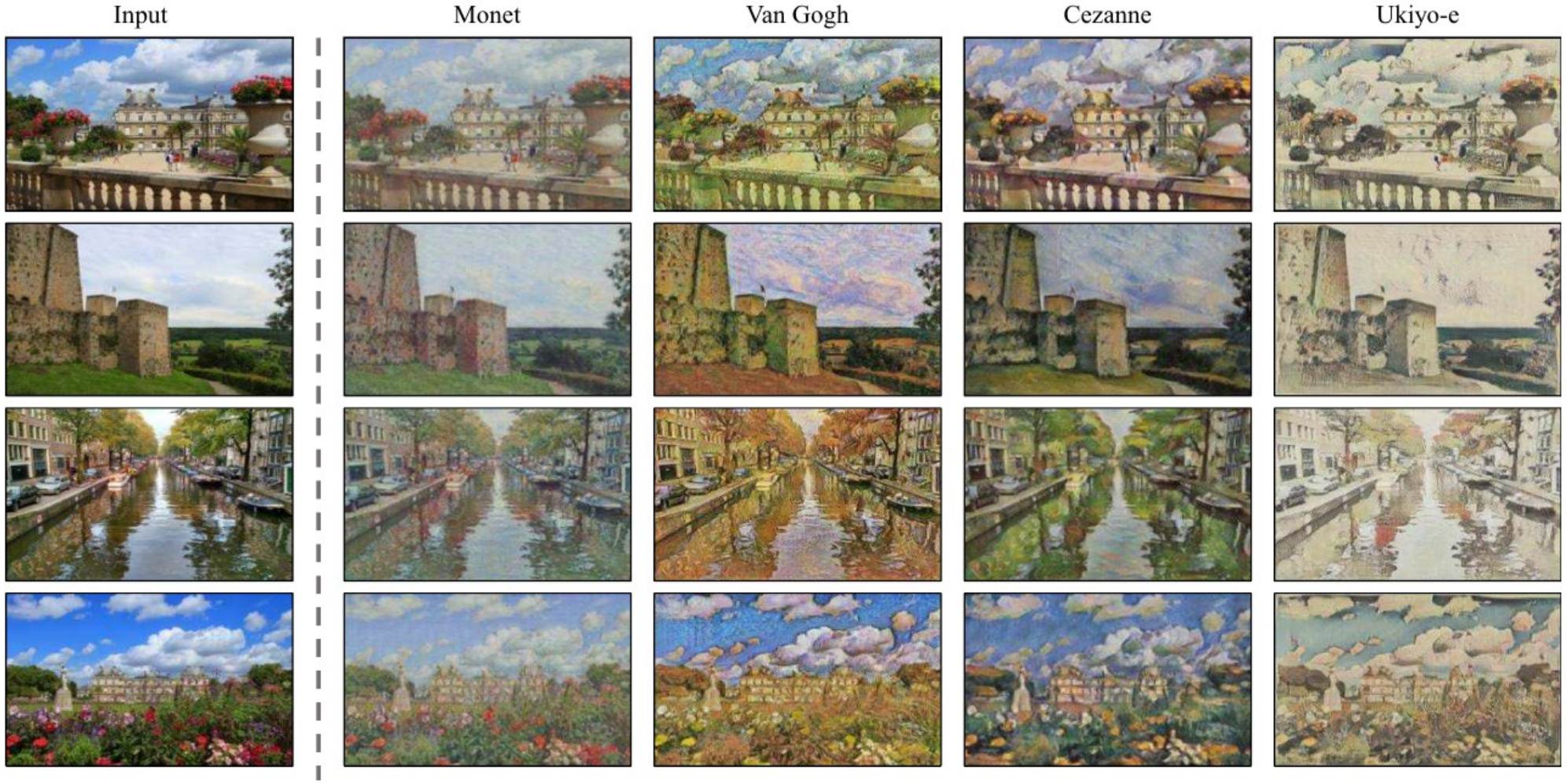
- Style Transfer



J. Y. Zhu et al., Unpaired image-to-image translation using cycle-consistent adversarial network, **ICCV 2017**

Applications in Image Processing & Vision **with AI**

- Style Transfer



J. Y. Zhu et al., Unpaired image-to-image translation using cycle-consistent adversarial network, **ICCV 2017**

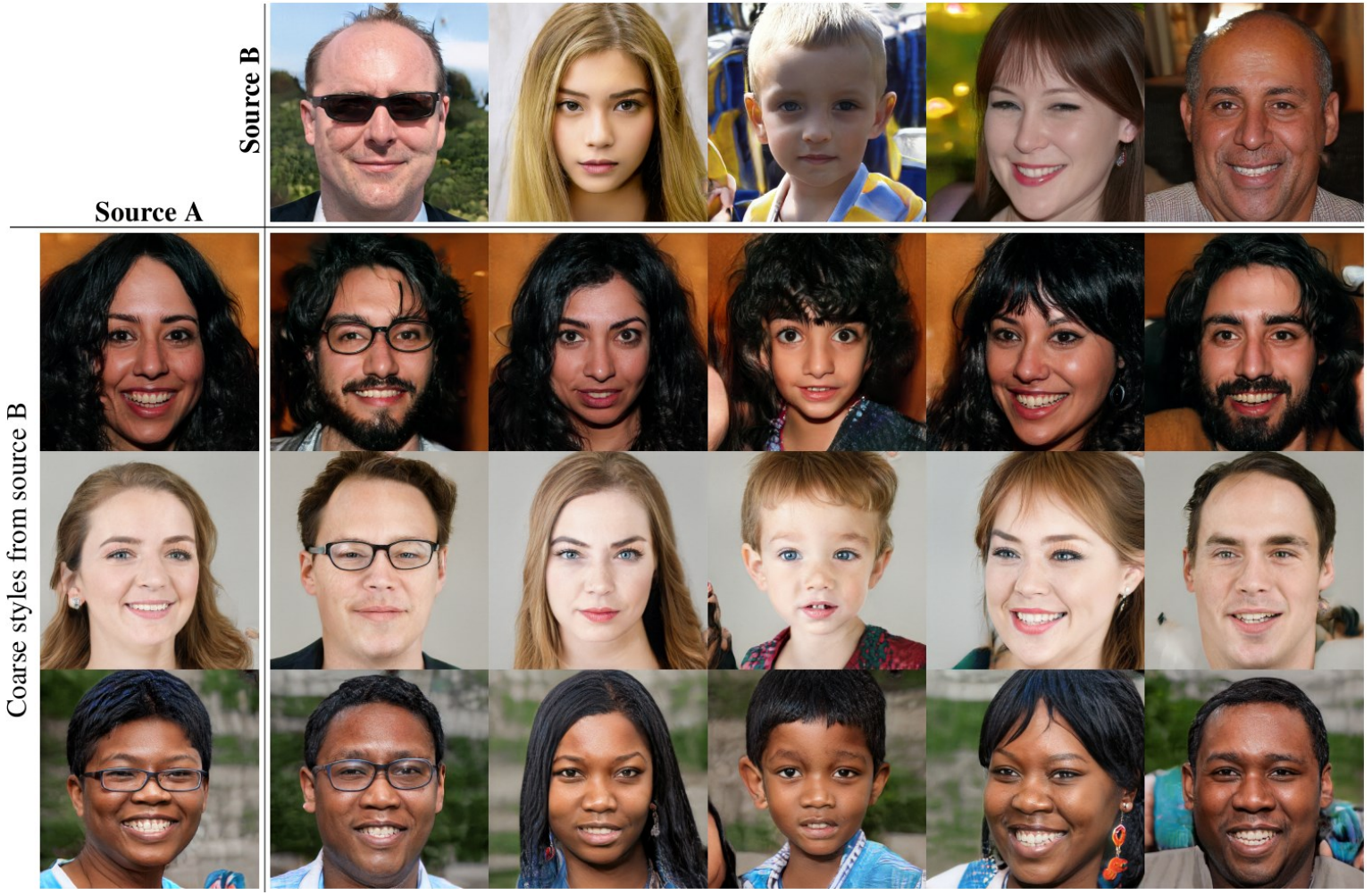
Applications in Image Processing & Vision **with AI**

- High-Quality Image Generation

T. Kerras et al., Progressive growing of GANs for improved quality, stability, and variation, **ICLR 2018**

Applications in Image Processing & Vision **with AI**

- StyleGAN



T. Kerras et al., A style-based generator architecture for generative adversarial networks, **CVPR 2019**

Applications in Image Processing & Vision **with AI**

- StyleGAN

T. Kerras et al., A style-based generator architecture for generative adversarial networks, **CVPR 2019**

Applications in Image Processing & Vision **with AI**

- Which Face is Real?

https://www.whichfaceisreal.com/index.php

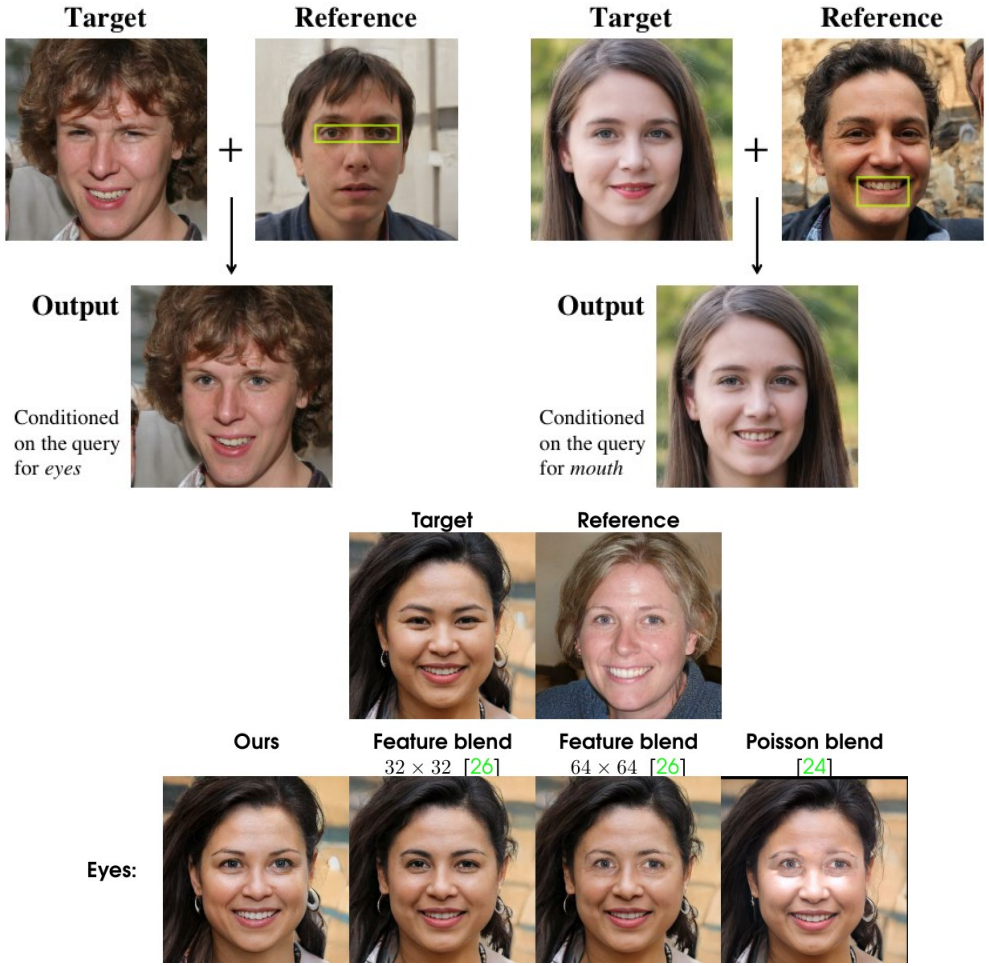
Applications in Image Processing & Vision **with AI**

- Which Face is Real?

https://www.whichfaceisreal.com/index.php

Applications in Image Processing & Vision **with AI**

- Editing in Style



Editing in Style: Uncovering the Local Semantics of GANs

Edo Collins[1] Raja Bala[2] Bob Price[2] Sabine Süsstrunk[1]

[1] School of Computer and Communication Sciences, EPFL, Switzerland

[2] Interactive and Analytics Lab, Palo Alto Research Center, Palo Alto, CA

E. Collins et al., Editing in Style: Uncovering the local semantics of GANs, **CVPR 2020**

Applications in Image Processing & Vision **with AI**

- 3D Reconstruction

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Ben Mildenhall* UC Berkeley
Pratul P. Srinivasan* UC Berkeley
Matthew Tancik* UC Berkeley
Jonathan T. Barron Google Research
Ravi Ramamoorthi UC San Diego
Ren Ng UC Berkeley

* Denotes Equal Contribution



B. Mildenhall et al., NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, **ECCV 2020**

Applications in Image Processing & Vision **with AI**

- **Dynamic** 3D Reconstruction

C. Gao et al., Dynamic View Synthesis from Dynamic Monocular Video, **ICCV 2021**

Applications in Image Processing & Vision **with AI**

- Lip Sync Synthesis in Video (Visual-Audio)

A Lip Sync Expert Is All You Need for Speech to Lip Generation In The Wild

"Accurately Lip-syncing Videos In The Wild"

K R Prajwal[*+], **Rudrabha Mukhopadhyay**[*+], Vinay Namboodiri[#], C.V. Jawahar[+]

[+]IIIT Hyderabad, [#]IIT Kanpur

ACM Multimedia, 2020 (Oral)

**** Both authors have contributed equally to this work.***

K. R. Prajwai et al., A Lip Sync Expert is All You Need for Speech to Lip Generation in the Wild, **ACM Multimedia 2020**

Wish You All The Best!

Hak Gu Kim hakgukim@cau.ac.kr
https://www.irislab.cau.ac.kr