Introduction to Computer Vision

Lecture 1. Course Introduction

31.10.22

Mikhail Belyaev

Skoltech
Skolkovo Institute of Science and Technology

Outline

- **Info about the course**
- Computer Vision problems & applications
- Convolutions
- *Medical Computer Vision*

Info about the course

Term 2, 3 credit course.

- Blended mode.
- Mon: 12:30-15:30, Fri:9:00-12:00
  - Lectures: the first ~90 minutes (with recording).
  - Practical exercises to work at home (provided after the lecture).
  - Online review of the exercises with QA after the next lecture: the remaining 60-90 minutes.
- 12 lectures + hands-ons
- Final project

Grading

- 3x20% - homeworks;
- 40% - final project.

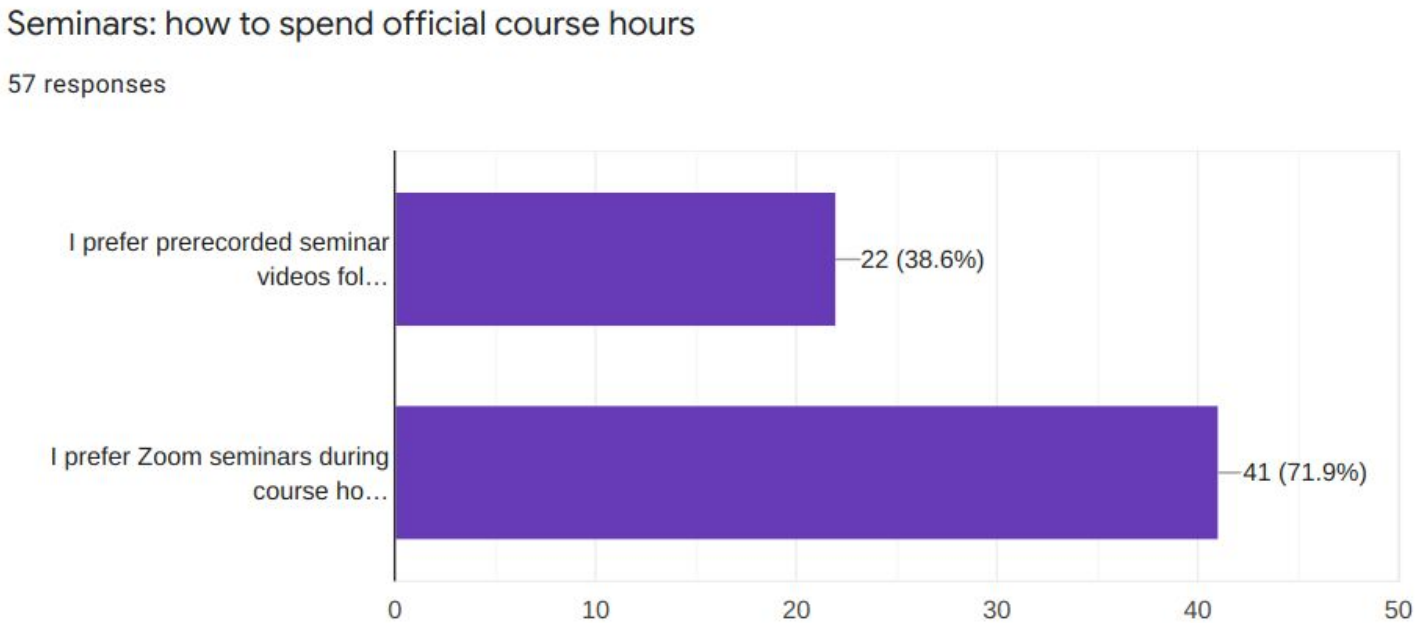
Info about the course

An old questonary about the format

Skoltech
Skolkovo Institute of Science and Technology

Info about the course

An old questonary about the format



Skoltech
Skolkovo Institute of Science and Technology

Info about the courses

Course outcomes

Knowledge

1. Statements of all major computer vision problems.
2. Mathematical details of the most important computer vision algorithms.

Skills

1. Select an appropriate method for solving particular computer vision problems.
2. Apply computer vision libraries.
3. Solve real-life computer vision problems.

Knowledge

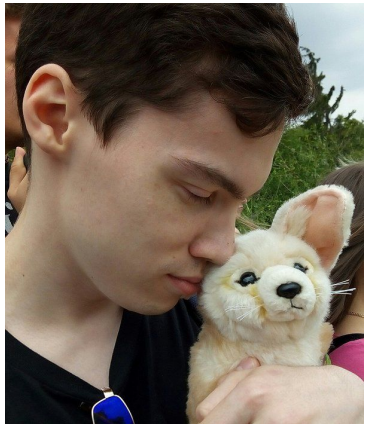
1. Implementing computer vision basic algorithms and compex pipelines

Course team

Mikhail Belyaev, Instructor

Natalia Soboleva, TA

Anton Vasilyuk, TA



Boris Shirokikh, TA

Course structure

Part 1. Classical CV algorithms. 9 lectures

1. Template matching. Edge detection.
2. Binary images. Morphological operations, contours in OpenCV.
3. Color spaces. Segmentation
4. Hough transform. Projective transform. RANSAC.
5. Object detection & classification: Haar cascade, Histogram of Oriented Gradients.
6. Keypoint detection. Blob detection. Scale invariance.
7. Image descriptors.
8. Optical flow
9. Object tracking.

Part 2. Convolutional neural networks intro. 2 lectures.

1. CNN basics. CNN for classification, detection & segmentation.

Course Evaluation - 2020

Standard Course Evaluation - AY2020/21		
1. Course objectives were clear to me	Strongly Agree	49 (71.01%)
	Agree	18 (26.09%)
	Disagree	2 (2.90%)
2. Key concepts and theories were well explained by the Course Instructor(s)	Strongly Agree	46 (65.71%)
	Agree	24 (34.29%)
3. Course content was difficult enough to be challenging	Strongly Agree	37 (53.62%)
	Agree	31 (44.93%)
	Disagree	1 (1.45%)

Course Evaluation - 2020

Standard Course Evaluation - AY2020/21		
1. Course objectives were clear to me	Strongly Agree	49 (71.01%)
	Agree	18 (26.09%)
	Disagree	2 (2.90%)
2. Key concepts and theories were well explained by the Course Instructor(s)	Strongly Agree	46 (65.71%)
	Agree	24 (34.29%)
3. Course content was difficult enough to be challenging	Strongly Agree	37 (53.62%)
	Agree	31 (44.93%)
	Disagree	1 (1.45%)

Everything was cool but why do we need the course?

Course Evaluation - 2020

Standard Course Evaluation - AY2020/21		
1. Course objectives were clear to me	Strongly Agree	49 (71.01%)
	Agree	18 (26.09%)
	Disagree	2 (2.90%)
2. Key co		.71%)
		.29%)
3. Course		.62%)
		.93%)
		5%)

DISCLAIMER

This course is not about Deep learning!

Everything was cool but why do we need the course?

Course Evaluation - 2020

Why do we need classical computer vision methods in 2021?

1. Many ideas in modern deep learning methods are based on classical computer vision methods.

2. Deep learning methods are data hungry. Using classical CV method you can create a prototype faster. Also, some unsupervised approaches can be used as to generate annotation candidates

Plagiarism policy

Rules are simple:

1. If you use external code, please add the link
2. Homeworks are individual assignments. If one homework is submitted twice, both student got ½ of the score.
3. Final projects are team-based. However, do not submit a random project from internet! It will hurt your overall score!

Homework 1

Do you know this game? If yes, what is the most boring part?

Homework 1

And counting scores at the end of the game is really boring!

The goal of HW1 is to create an automated scoring system.

Homework 1

The goal of HW1 is to create an automated scoring system for Ticket to ride!

Challenges

1. Detect all town markers and trains

2. Split cells vs trains

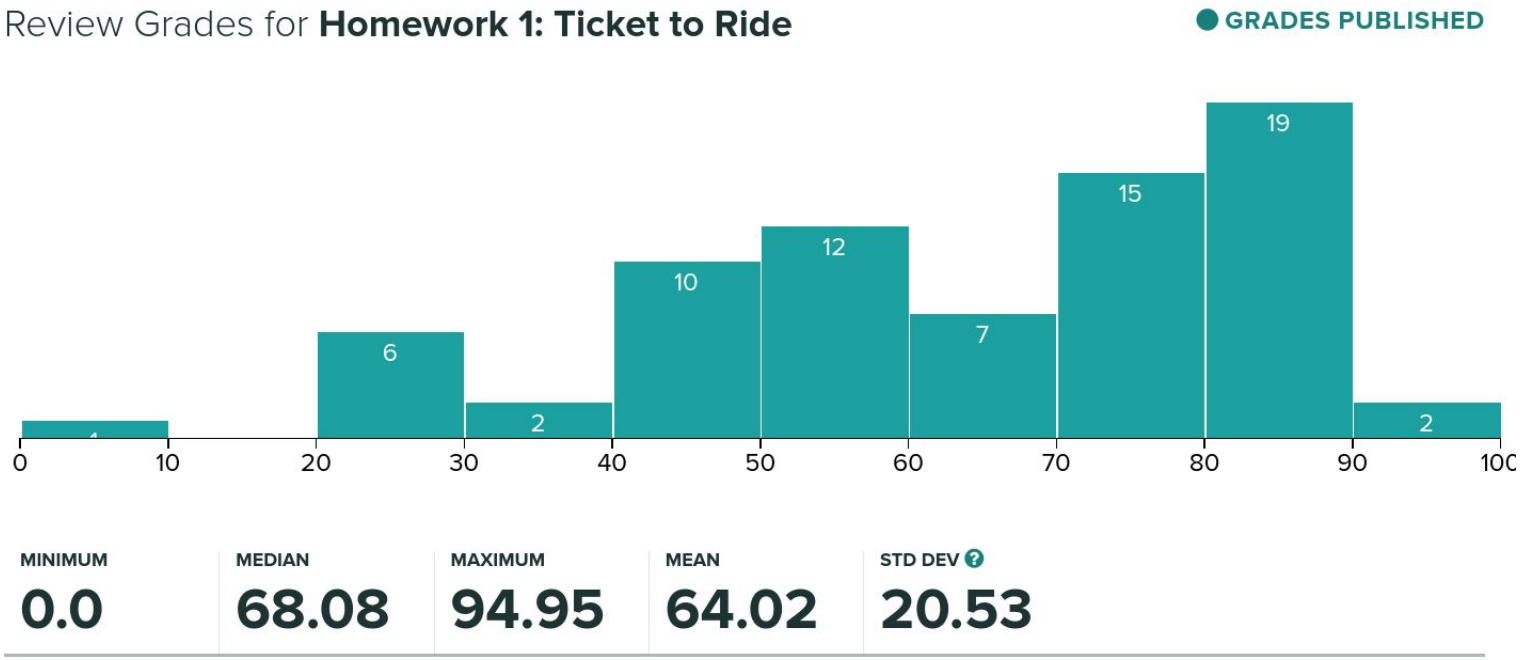
3. Estimate the colors (e.g. take into account different light conditions)

Homework 1 - grading

We will try to use gradescope as usual.

- 2.5 points for city detection
- 5.0 points per image for detecting trains.
- 2.5 points for the final game score.

10 points for each of the 10 test images will result in 100 points (max).



Outline

- Info about the course
- **Computer Vision problems & applications**
- Convolutions
- *Medical Computer Vision*

Computer Vision or Image processing?

What is the difference between Image Processing and Computer Vision? There are various definitions, but we'll stick with the following.

Computer Vision

- Image or Video -> Interpretation (understanding)

Image Processing:

- Image / Images -> A new Image / New Images

CV problems: Image Classification

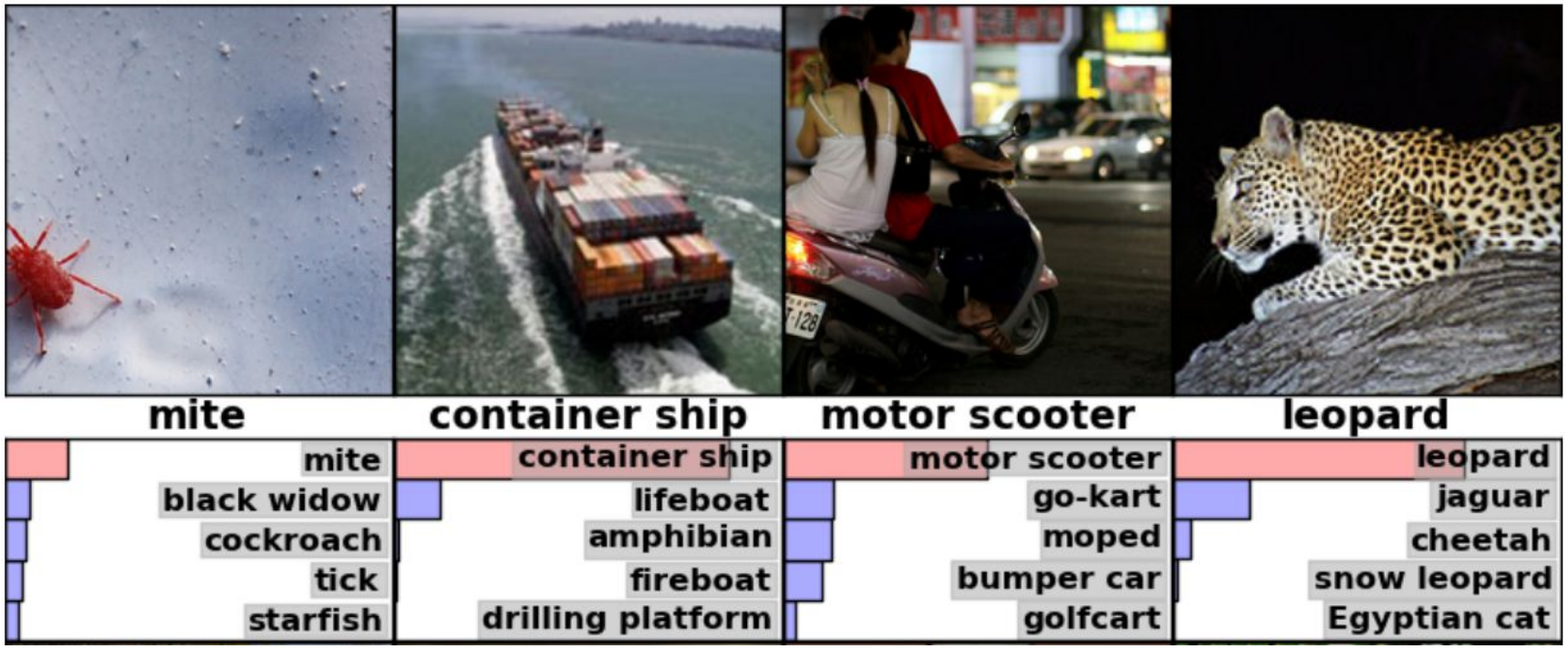


Image source: Krizhevsky et al. *ImageNet Classification with Deep Convolutional Neural Networks*

CV problems: Object detection

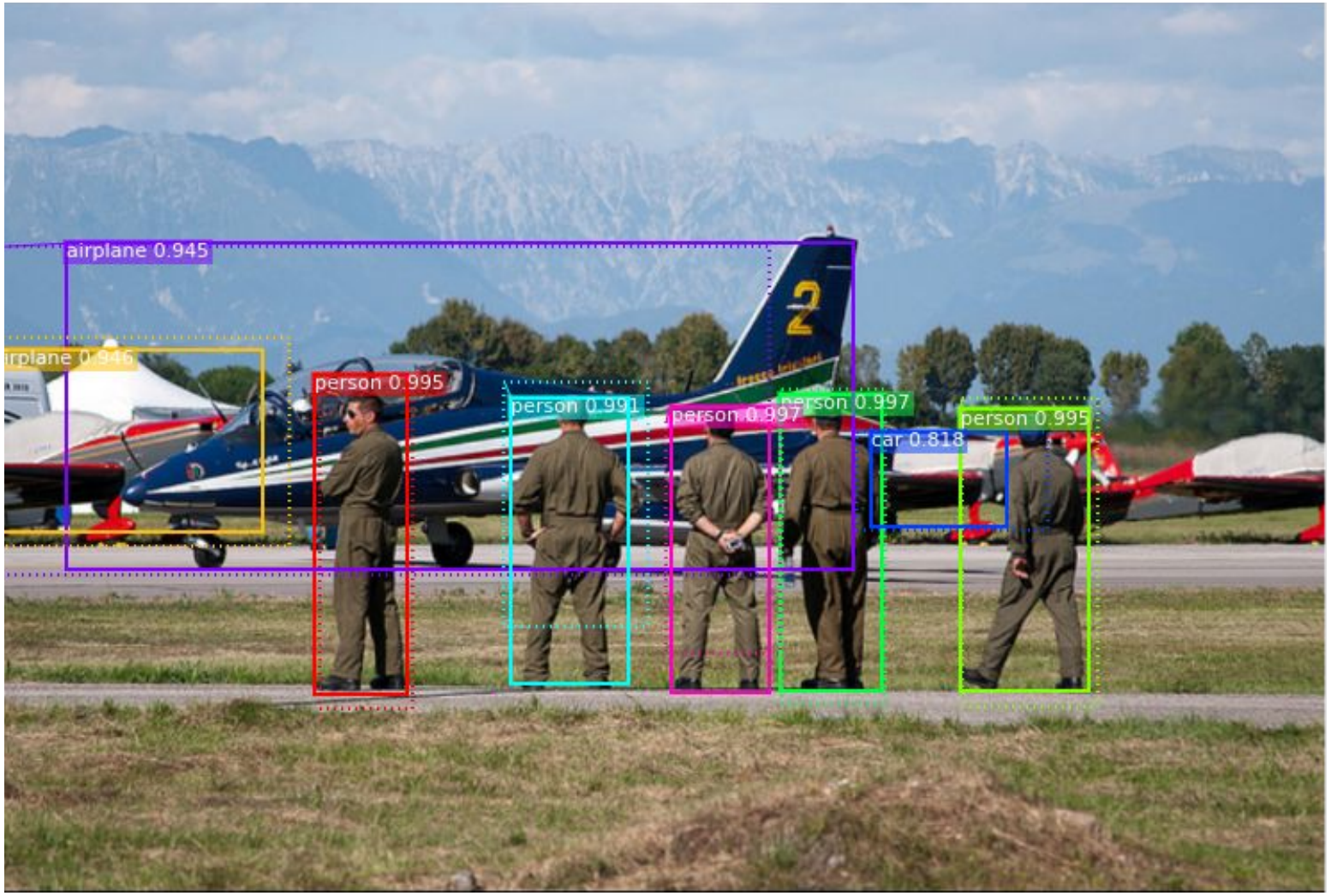
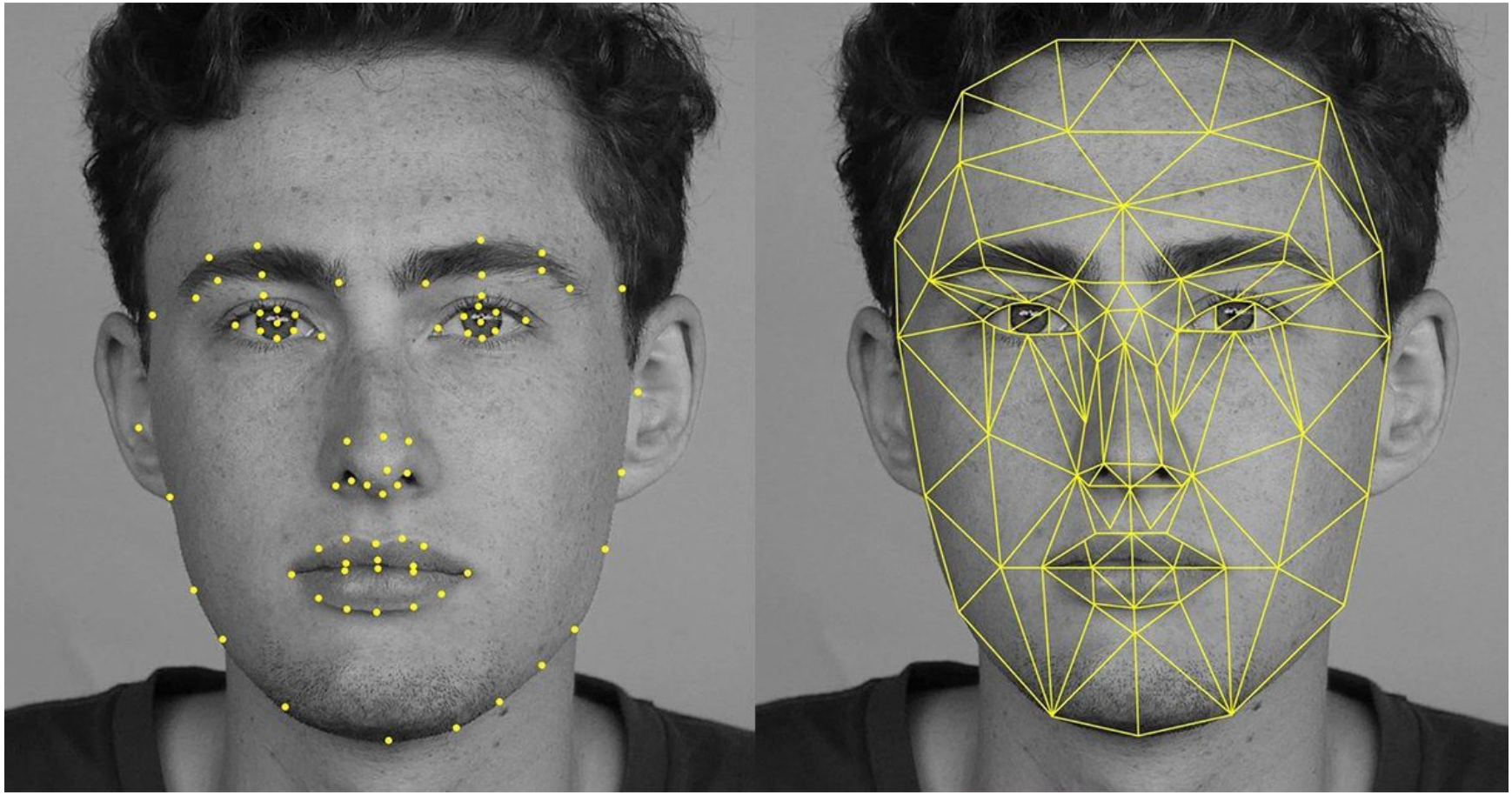


Image source: https://github.com/matterport/Mask_RCNN

CV problems: Keypoints detection



CV problems: Segmentation

Image source: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html

CV problems: Semantic segmentation

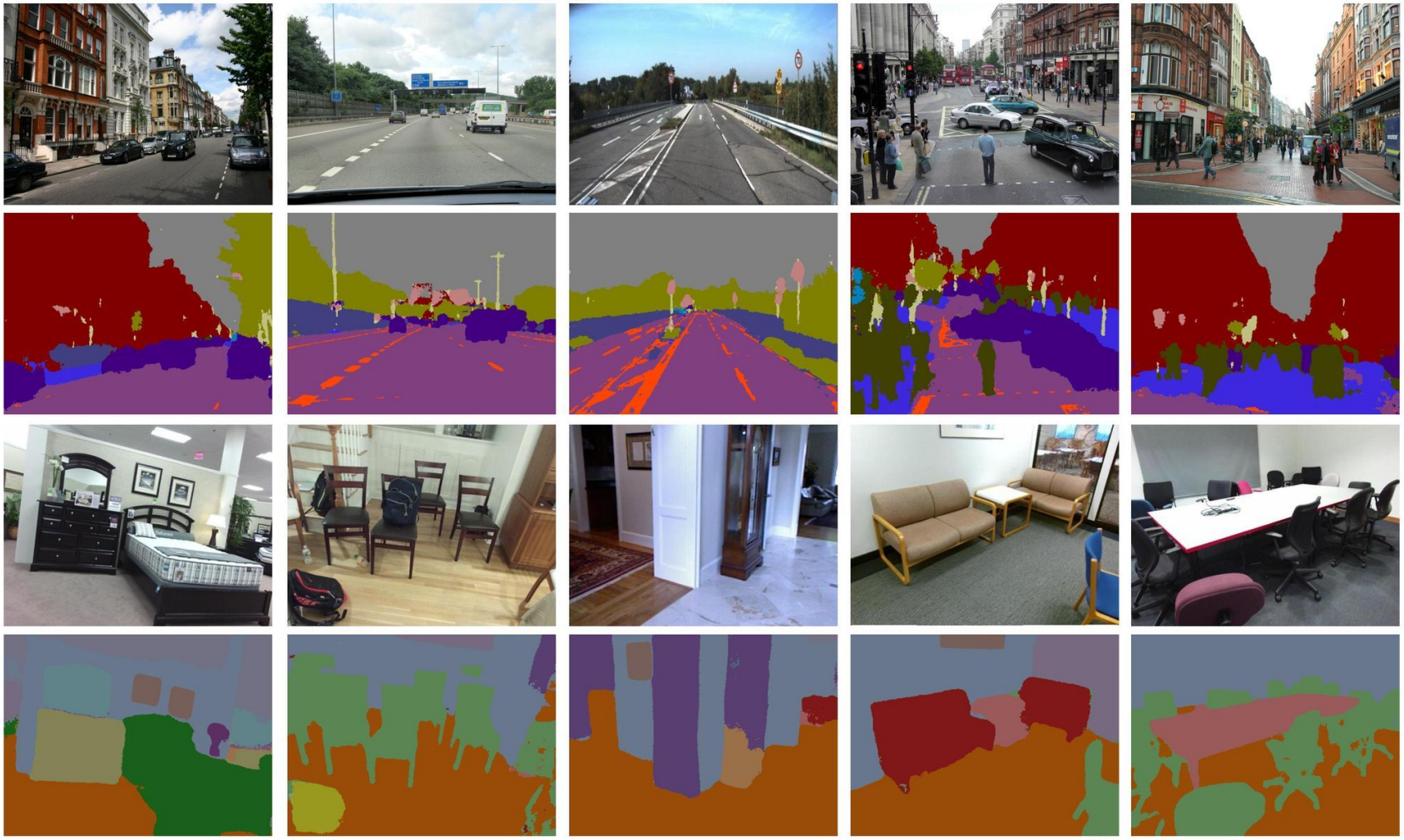


Image source: https://mi.eng.cam.ac.uk/projects/segnet/

CV problems: Image Captioning

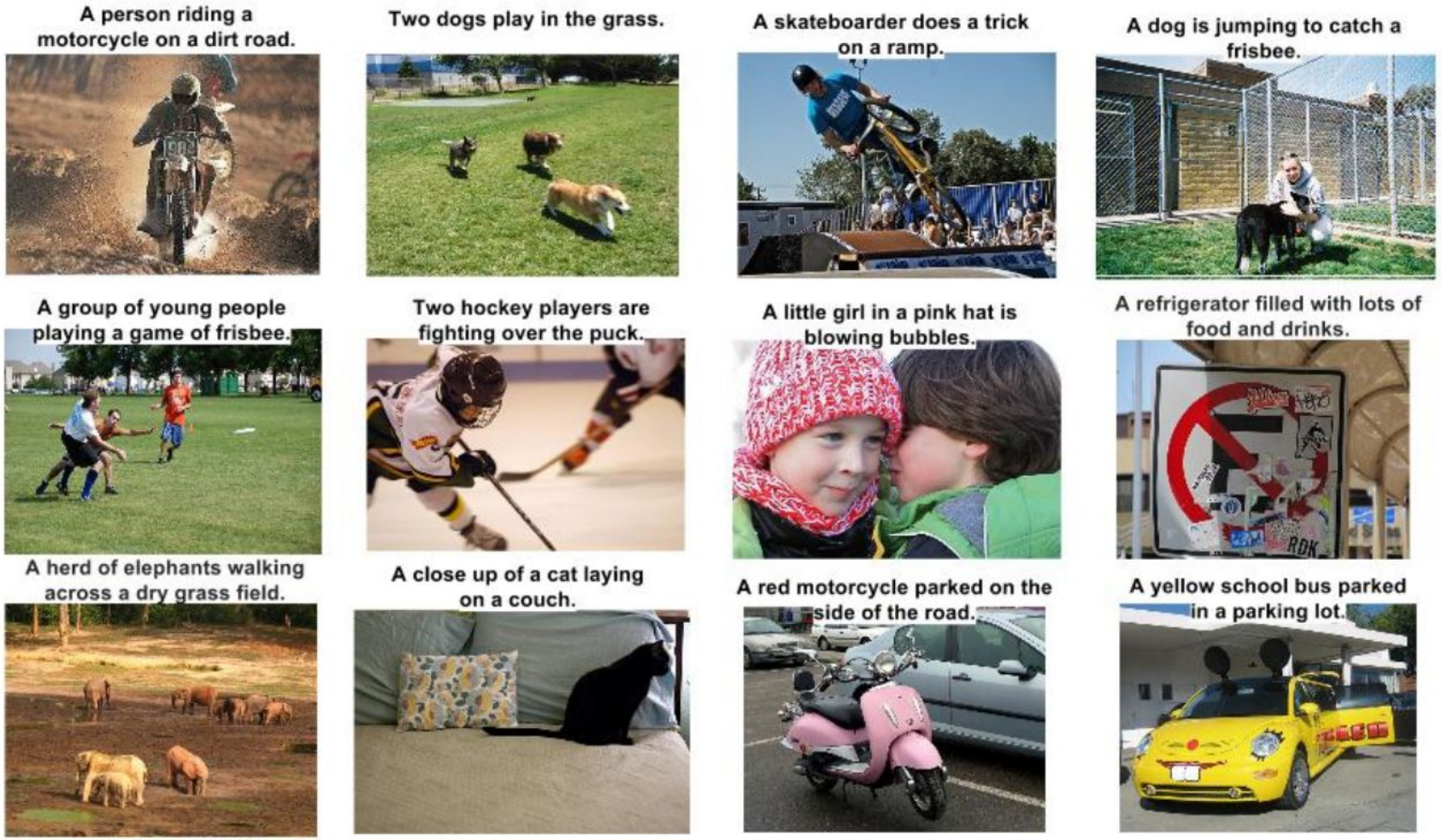


Image source: Vinyals et al. *Show and Tell: A Neural Image Caption Generator*

CV problems: Depth Estimation

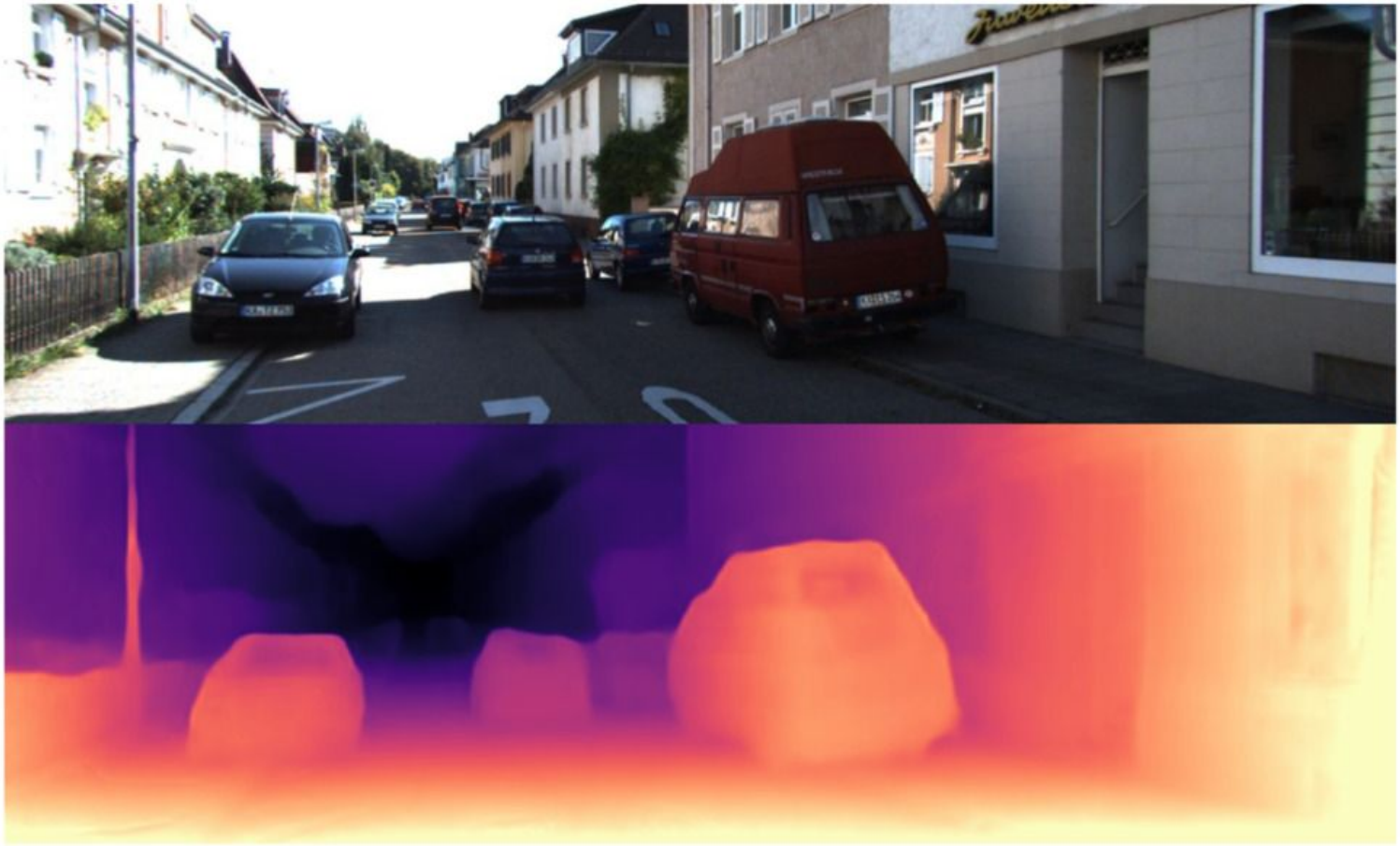
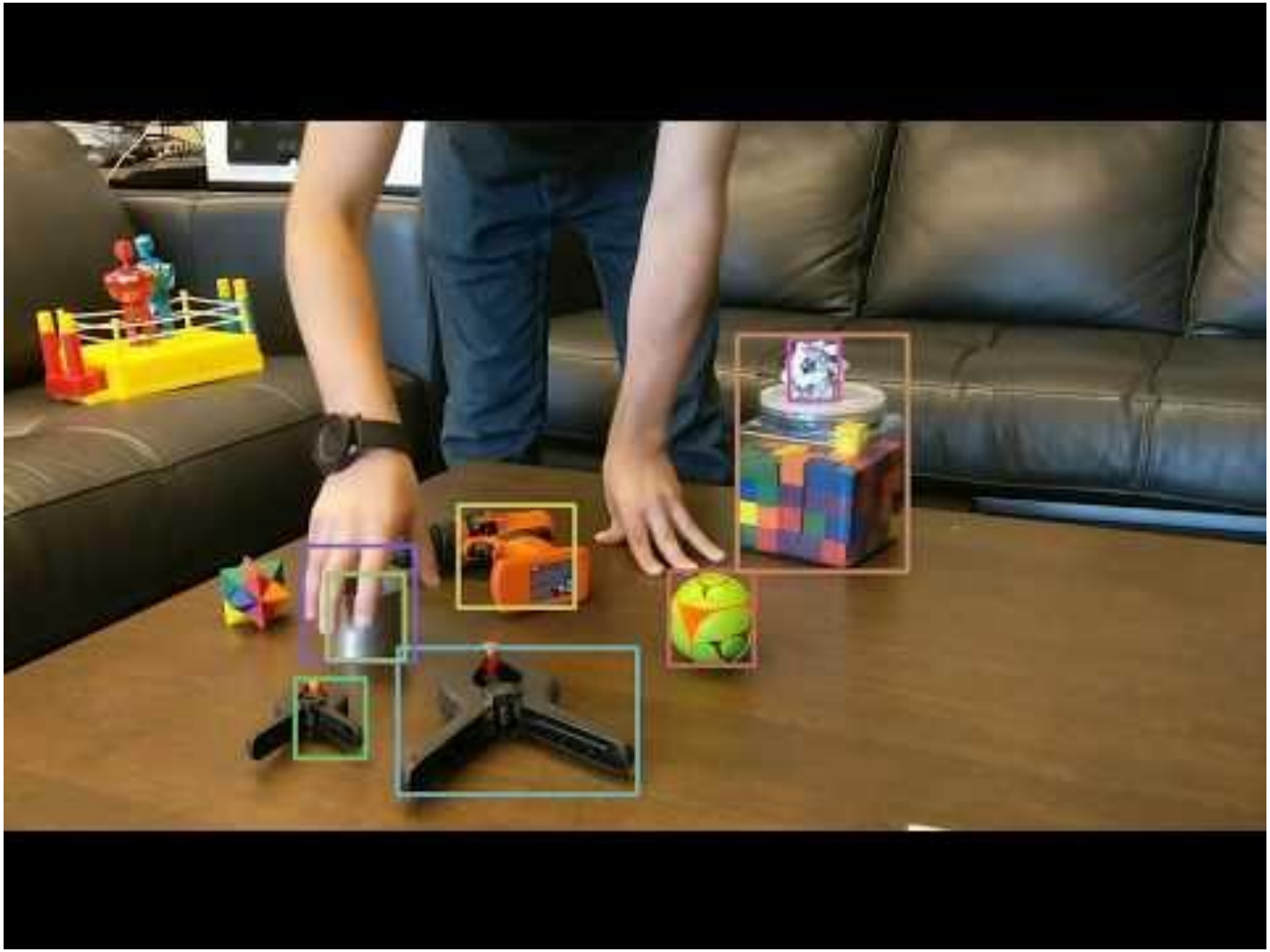


Image source: https://github.com/nianticlabs/monodepth2

CV problems: Object tracking



Video source: https://arxiv.org/abs/1705.06368

Image Processing problems: Super-resolution

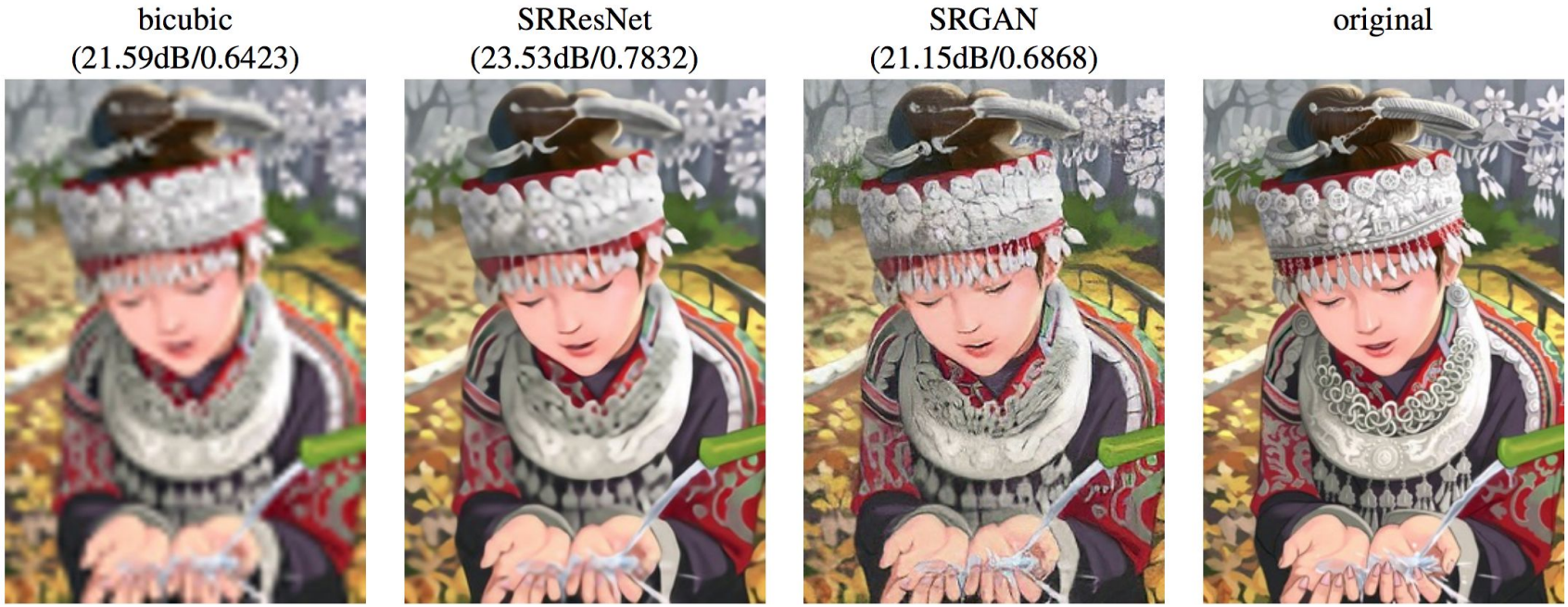


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

Image source: Ledig et al. *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*

Image Processing problems: Inpainting

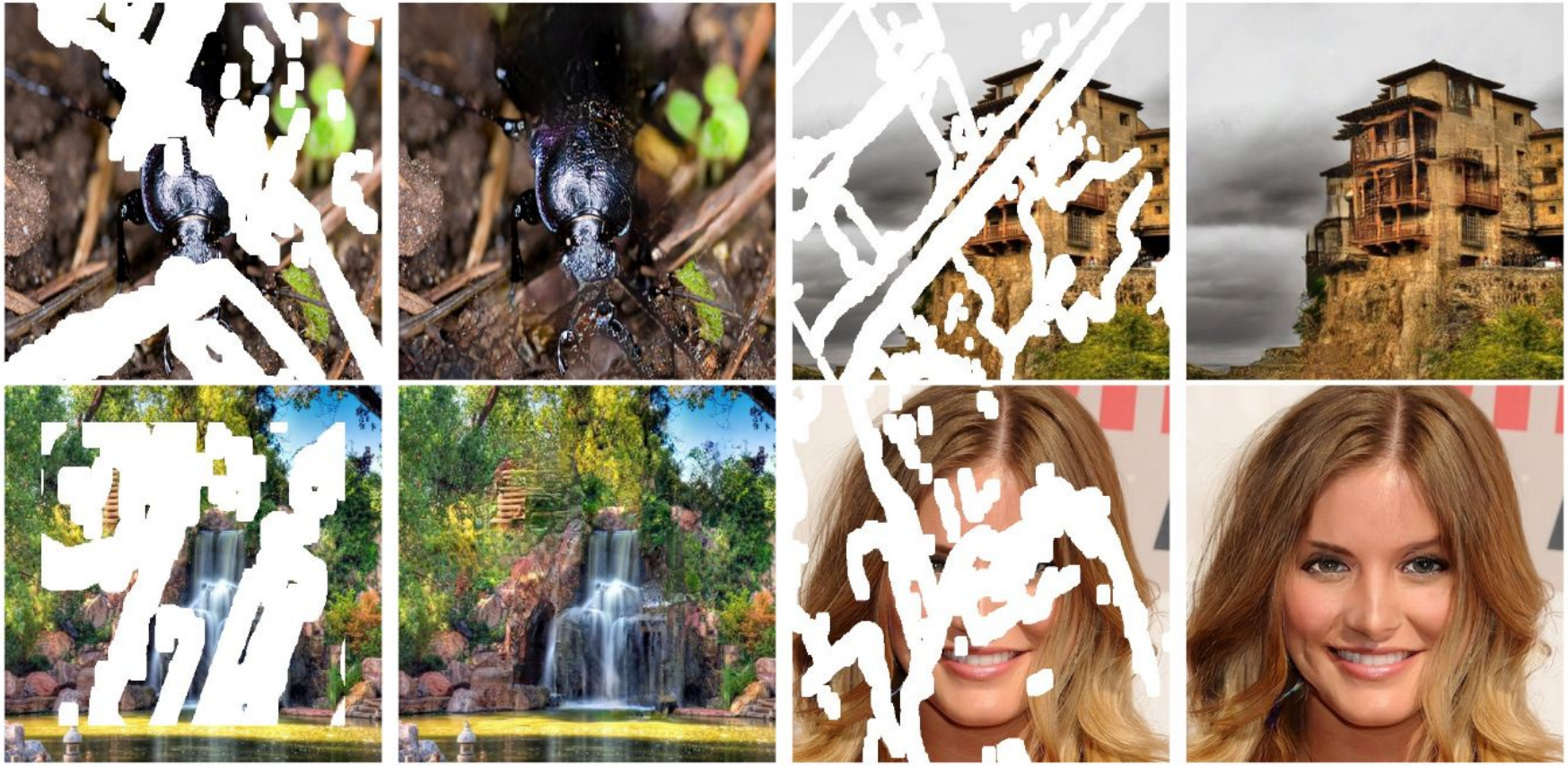


Image source: Liu et al. *Image Inpainting for Irregular Holes Using Partial Convolutions*

Image Processing problems: Image generation

Image sources: Gravity, Jurassic Park, Avatar

Image Processing problems: Style Transfer

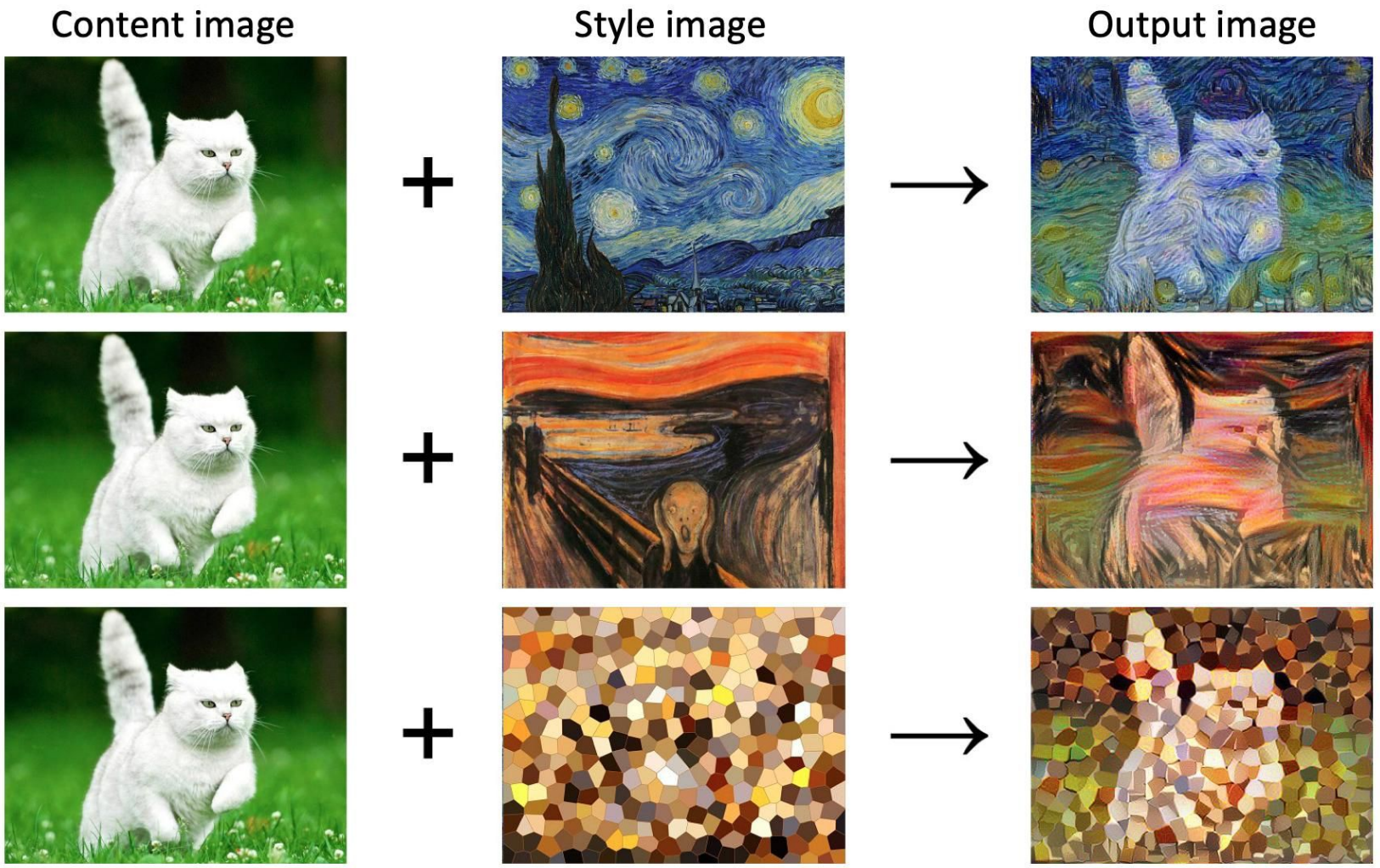


Image sources: https://blog.godatadriven.com/images/how-to-style-transfer

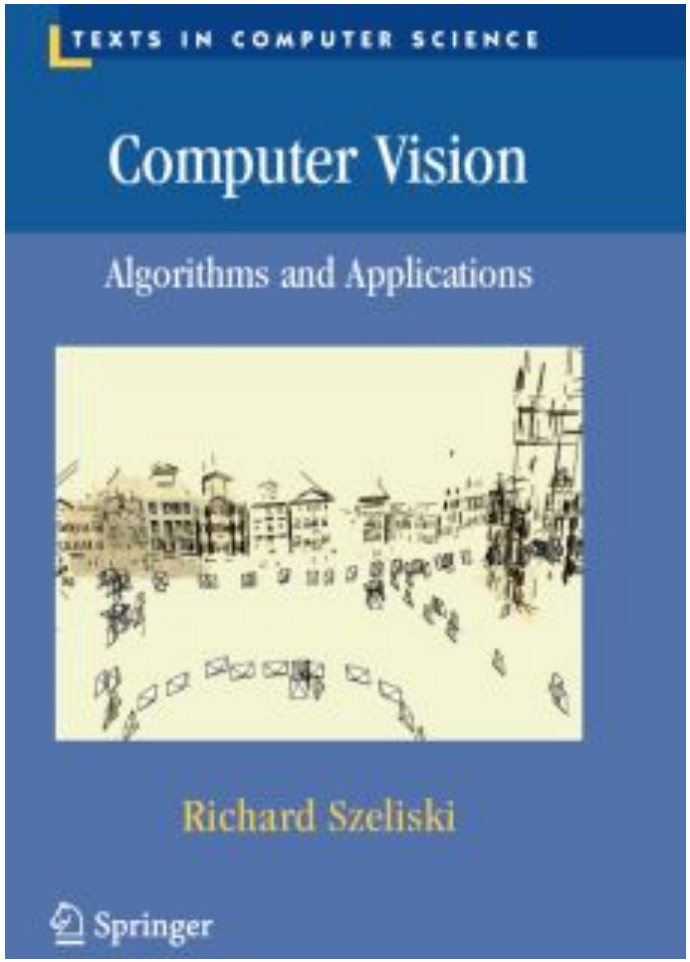
Image Processing problems: Face generation

Image sources: https://arxiv.org/pdf/1812.04948.pdf

Outline

- Info about the course
- Computer Vision problems & applications
- **Convolutions**
- *Medical Computer Vision*

Reading

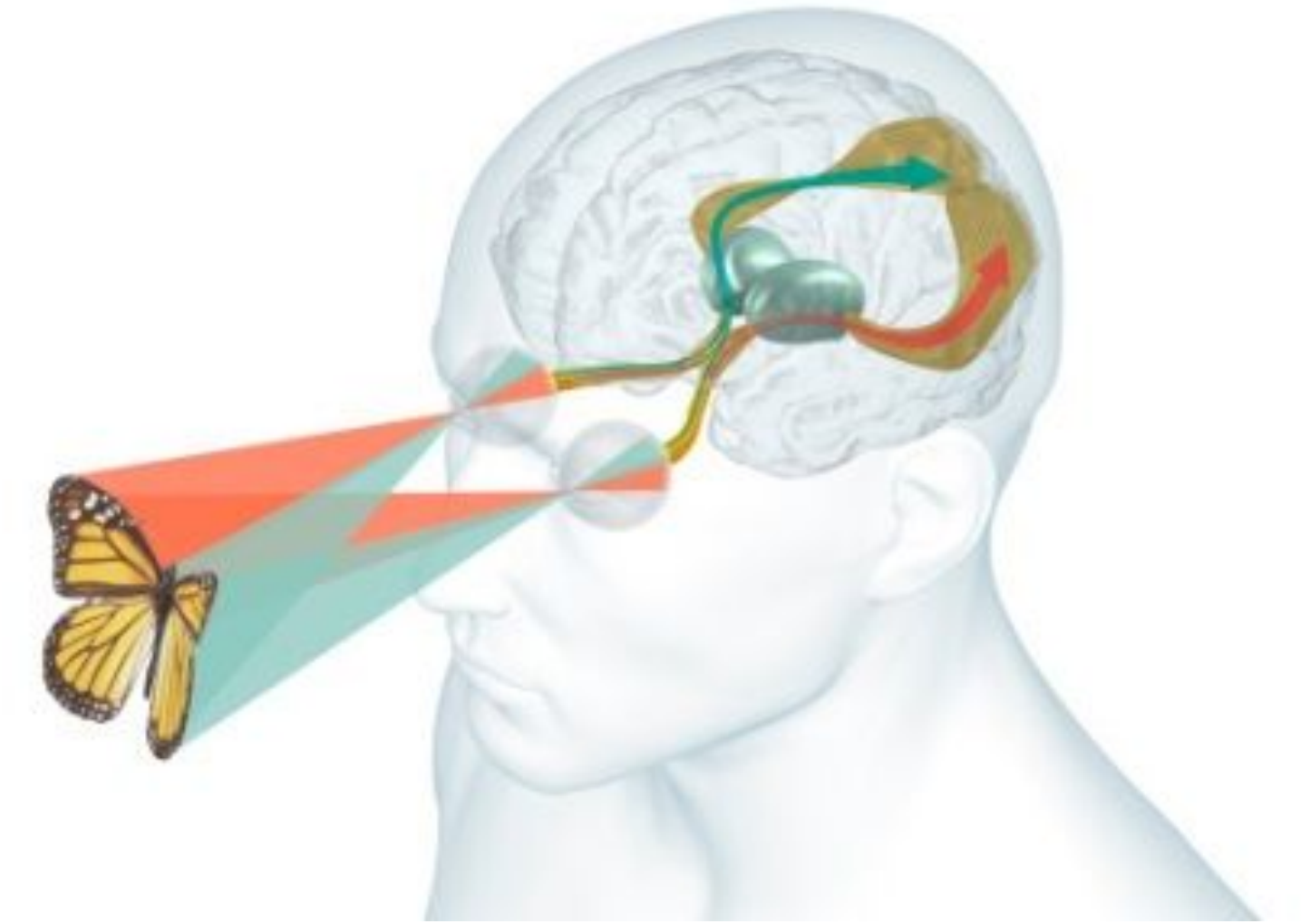


Good reading for this lecture.

The book is available online

What is an *image*?

How do humans perceive images?



How do humans perceive images?

What are the colors of this dress?

How do humans perceive images?

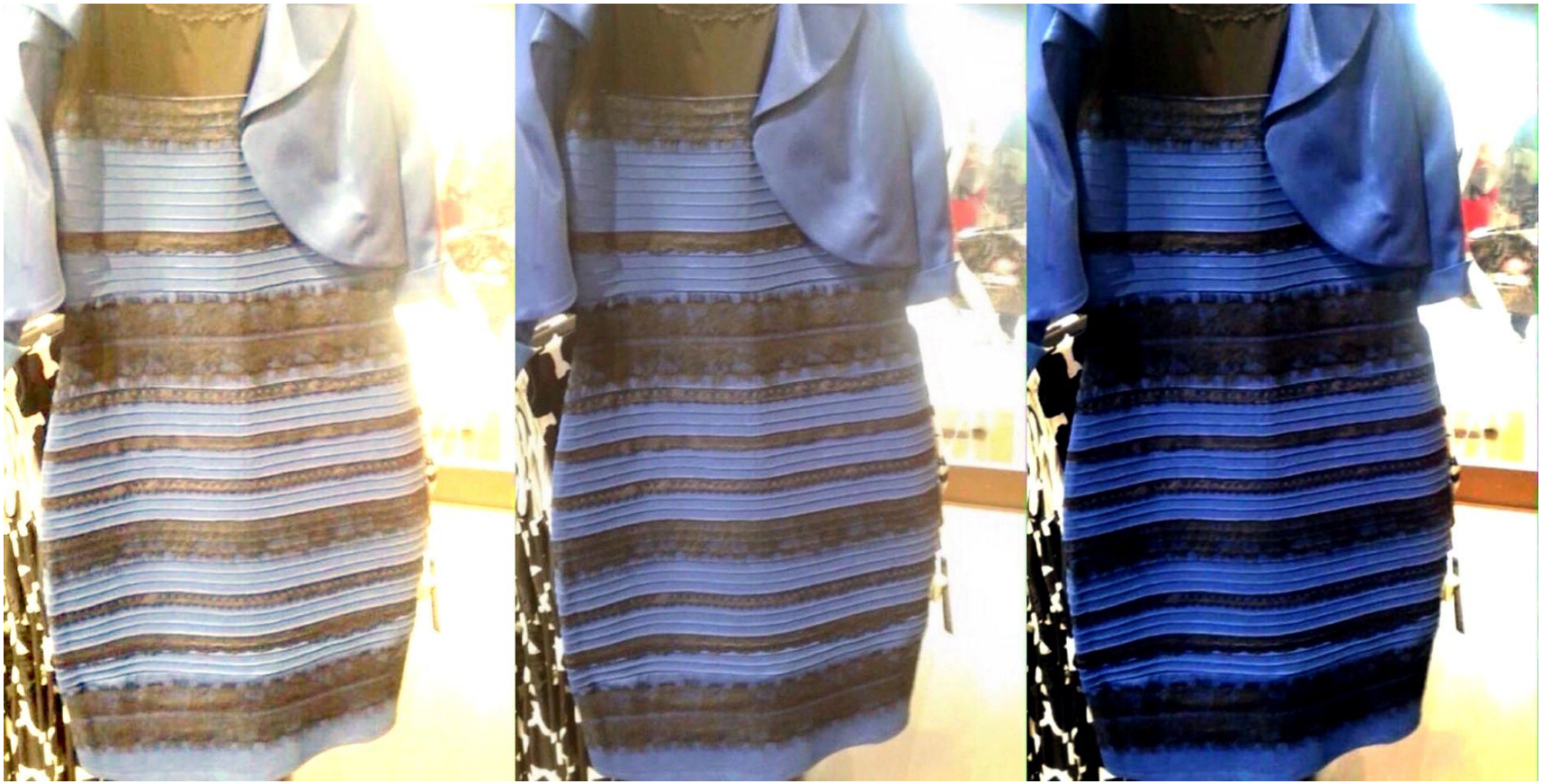
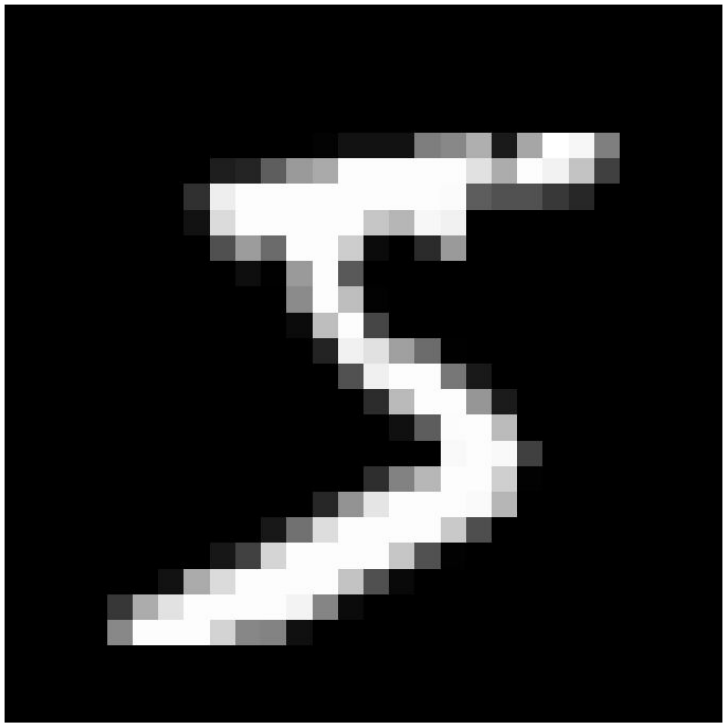


Image perception is subjective: different people will see this dress (left) either as blue and black or as white and brown.

How do humans perceive images?

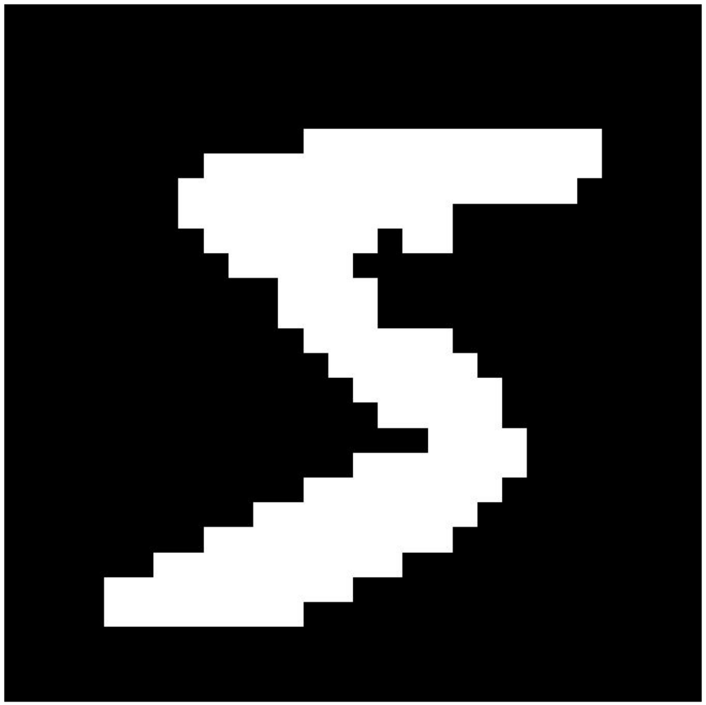
Image perception is subjective: do you see a young woman or an old one?

How do machines store images?



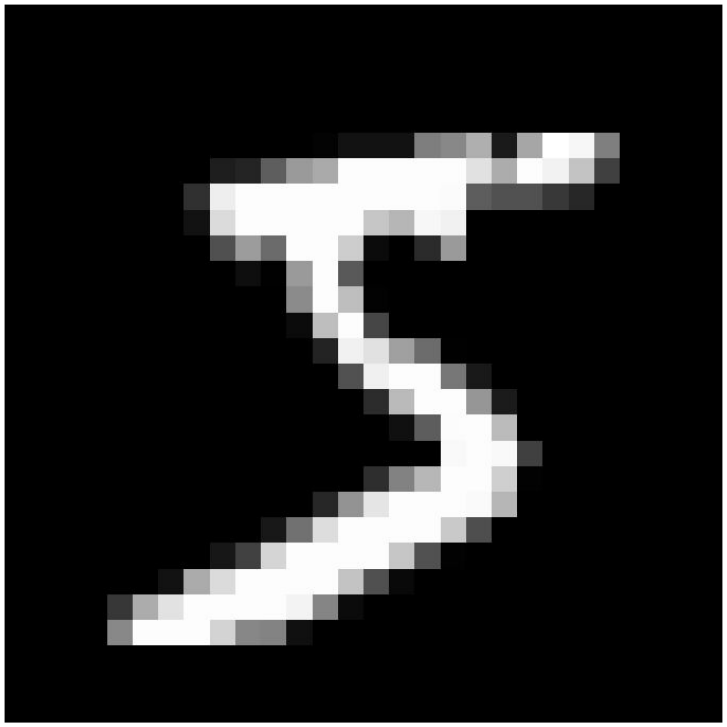
```
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   3  18  18  18 126 136 175  26 166 255 247 127   0   0   0   0
  0   0   0   0   0   0   0   0  30  36  94 154 170 253 253 253 253 253 225 172 253 242 195  64   0   0   0   0
  0   0   0   0   0   0   0  49 238 253 253 253 253 253 253 253 253 251  93  82  82  56  39   0   0   0   0   0
  0   0   0   0   0   0   0  18 219 253 253 253 253 253 198 182 247 241   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0  80 156 107 253 253 205  11   0  43 154   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0  14   1 154 253  90   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0 139 253 190   2   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0  11 190 253  70   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0  35 241 225 160 108   1   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0  81 240 253 253 119  25   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0  45 186 253 253 150  27   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  16  93 252 253 187   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 249 253 249  64   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0  46 130 183 253 253 207   2   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0  39 148 229 253 253 253 250 182   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0  24 114 221 253 253 253 253 201  78   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0  23  66 213 253 253 253 253 198  81   2   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0  18 171 219 253 253 253 253 195  80   9   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0  55 172 226 253 253 253 253 244 133  11   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0 136 253 253 253 212 135 132  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

How do machines store images: Binary



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

How do machines store images: Grayscale



```
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   3  18  18  18 126 136 175  26 166 255 247 127   0   0   0   0
  0   0   0   0   0   0   0   0  30  36  94 154 170 253 253 253 253 253 225 172 253 242 195  64   0   0   0   0
  0   0   0   0   0   0   0  49 238 253 253 253 253 253 253 253 253 251  93  82  82  56  39   0   0   0   0   0
  0   0   0   0   0   0   0  18 219 253 253 253 253 253 198 182 247 241   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0  80 156 107 253 253 205  11   0  43 154   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0  14   1 154 253  90   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0 139 253 190   2   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0  11 190 253  70   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0  35 241 225 160 108   1   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0  81 240 253 253 119  25   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0  45 186 253 253 150  27   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  16  93 252 253 187   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 249 253 249  64   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0  46 130 183 253 253 207   2   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0  39 148 229 253 253 253 250 182   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0  24 114 221 253 253 253 253 201  78   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0  23  66 213 253 253 253 253 198  81   2   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0  18 171 219 253 253 253 253 195  80   9   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0  55 172 226 253 253 253 253 244 133  11   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0 136 253 253 253 212 135 132  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

Integer values from 0 to 255 (1 byte per pixel)

How do machines store images: RGB

Red

Green

Blue

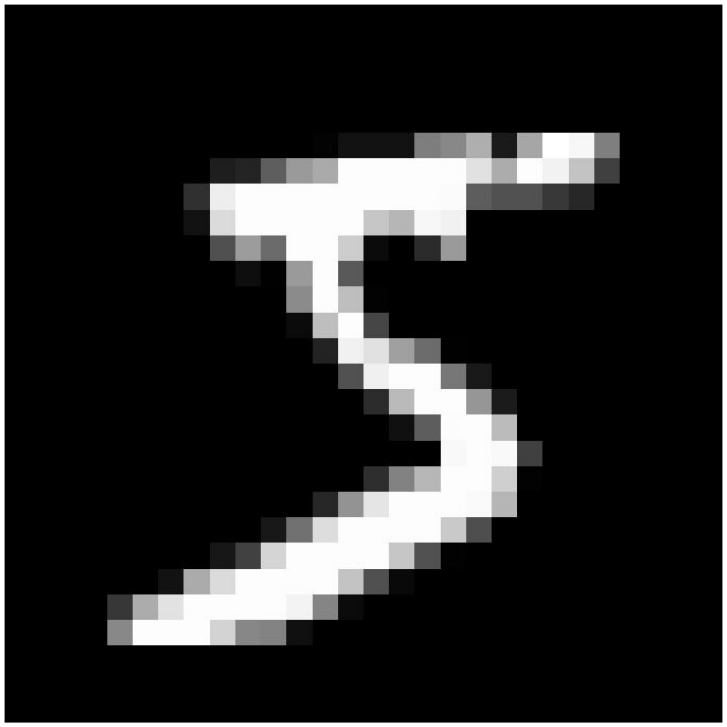
How do machines store images: RGB

Red

Green

Blue

How do machines store images?

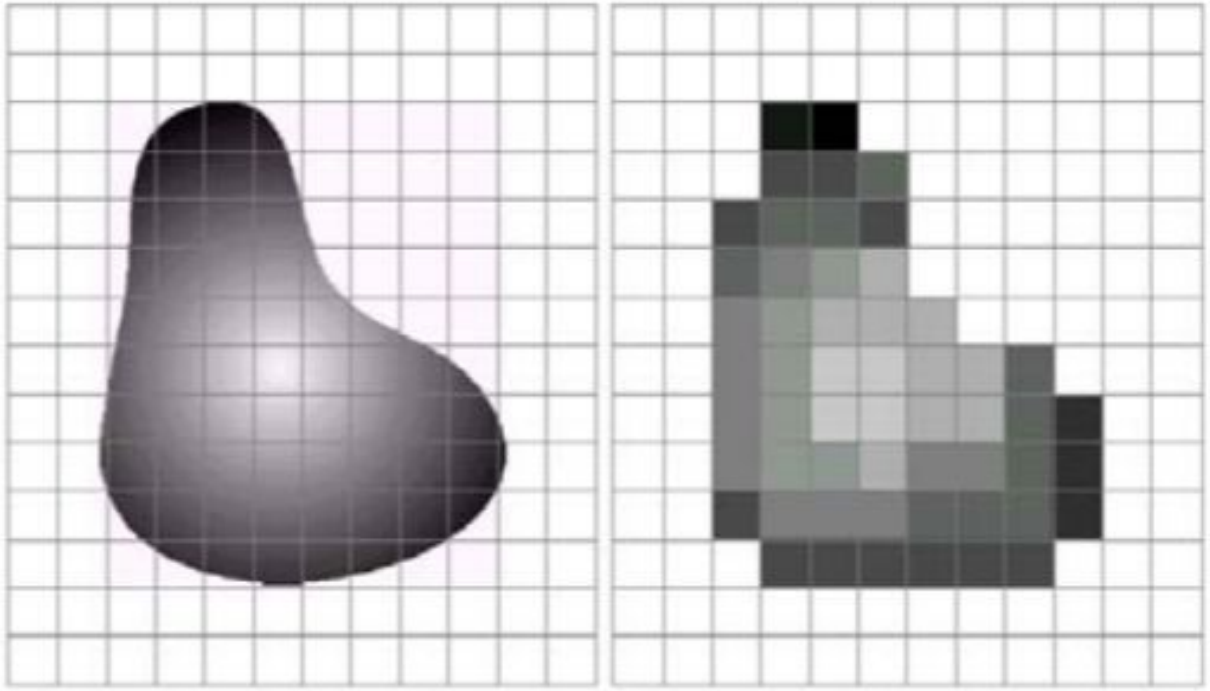


```
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   3  18  18  18 126 136 175  26 166 255 247 127   0   0   0   0
  0   0   0   0   0   0   0   0  30  36  94 154 170 253 253 253 253 253 225 172 253 242 195  64   0   0   0   0
  0   0   0   0   0   0   0  49 238 253 253 253 253 253 253 253 253 251  93  82  82  56  39   0   0   0   0   0
  0   0   0   0   0   0   0  18 219 253 253 253 253 253 198 182 247 241   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0  80 156 107 253 253 205  11   0  43 154   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0  14   1 154 253  90   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0 139 253 190   2   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0  11 190 253  70   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0  35 241 225 160 108   1   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0  81 240 253 253 119  25   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0  45 186 253 253 150  27   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  16  93 252 253 187   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 249 253 249  64   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0  46 130 183 253 253 207   2   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0  39 148 229 253 253 253 250 182   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0  24 114 221 253 253 253 253 201  78   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0  23  66 213 253 253 253 253 198  81   2   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0  18 171 219 253 253 253 253 195  80   9   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0  55 172 226 253 253 253 253 244 133  11   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0 136 253 253 253 212 135 132  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

This a matrix. But from mathematical points of view we have another option

How do machines store images?

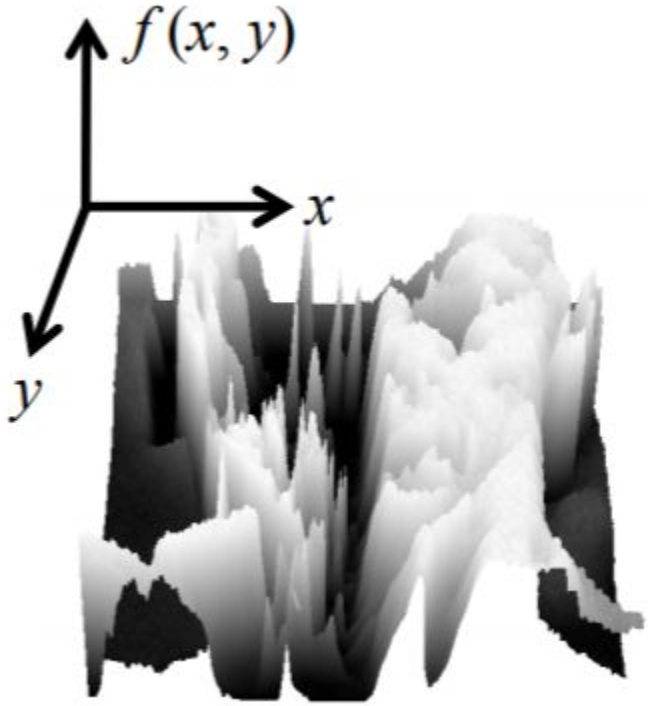
We can consider image as a discrete representation of a 2D function.



Source: D. Hoiem

How do machines store images?

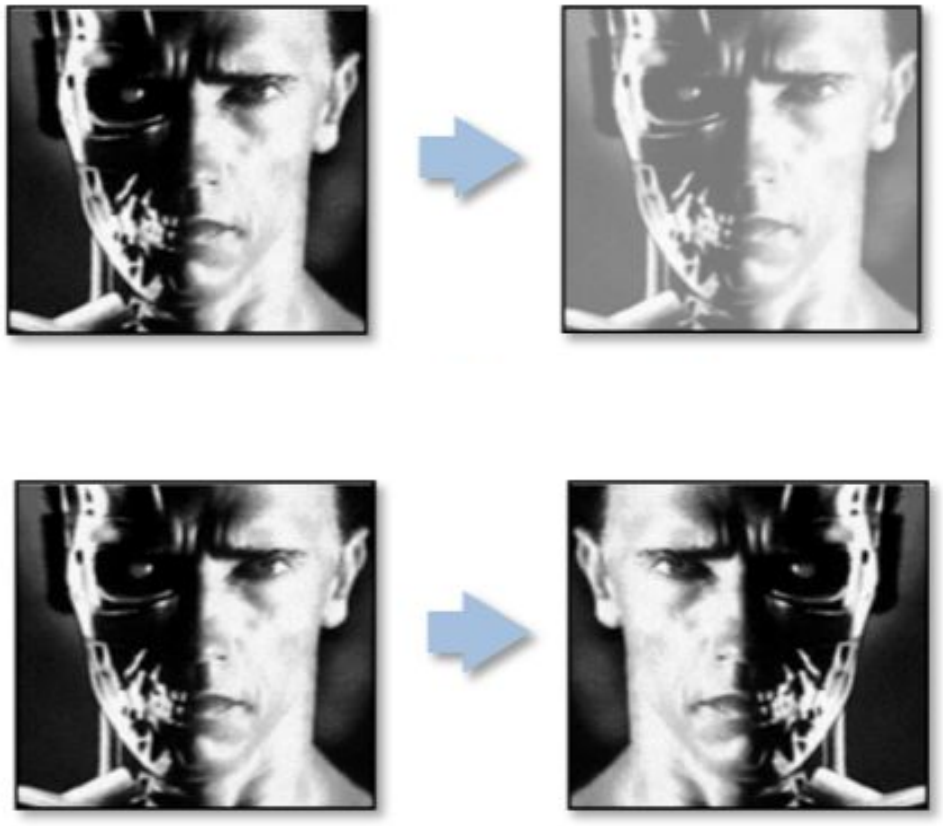
We can consider image as a discrete representation of a 2D function.



Source: N. Snavely

How do machines store images?

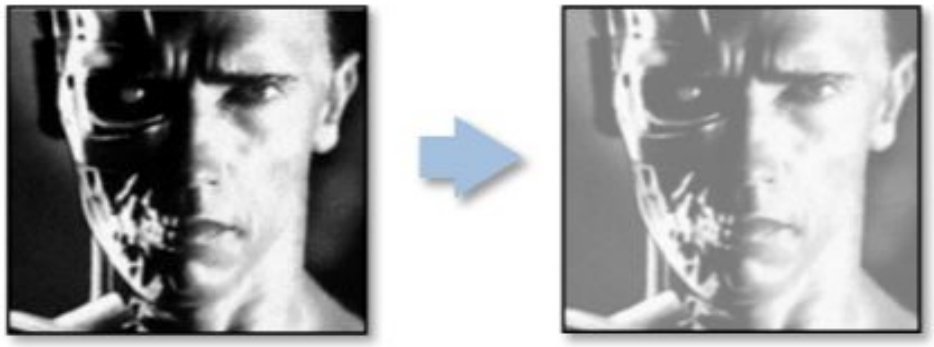
So some standard operations can be applied



Source: N. Snavely

How do machines store images?

So some standard operations can be applied

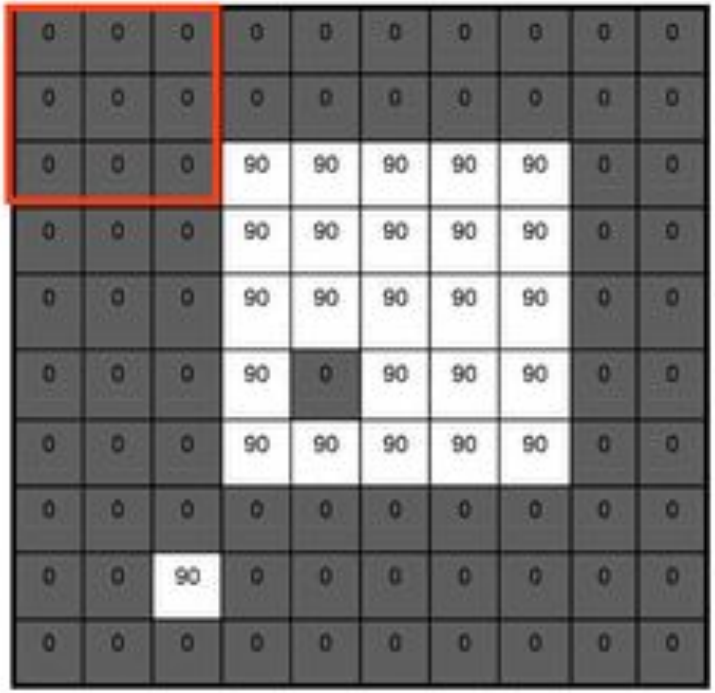


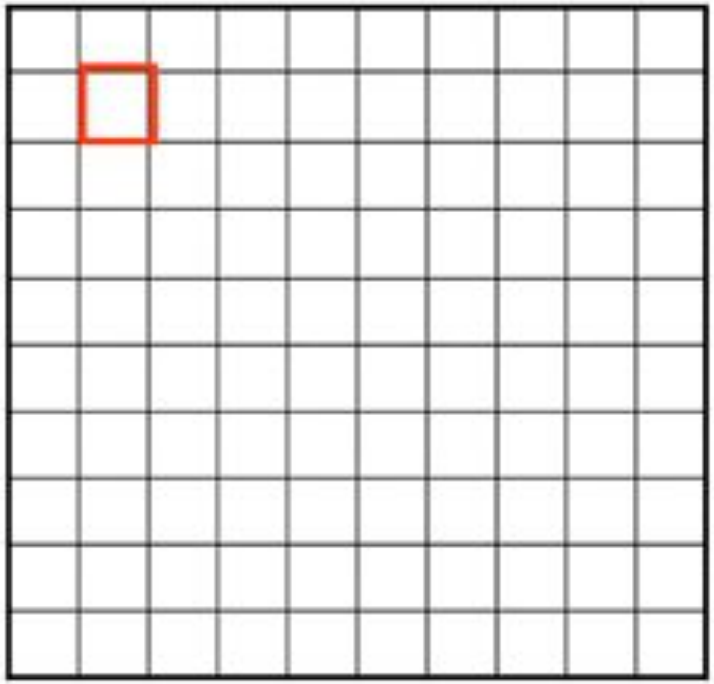
$$g(x,y) = f(x,y) + 20$$

$$g(x,y) = f(-x,y)$$

Source: N. Snavely

Convolutions: moving average example





Sources: S. Seitz, Image filtering tutorial

Convolutions: moving average

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Sources: S. Seitz, Image filtering tutorial

Convolutions: some math

Let us start with general 2 dimensional functions

$$(f * g)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') g(x - x', y - y') dx'\, dy'$$

Convolutions: some math

Let us start with general 2 dimensional functions

$$(f * g)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') g(x - x', y - y') dx'\, dy'$$

But images are discrete representations of 2D functions

$$(f * g)(x, y) = \sum_{x'=-\infty}^{\infty} \sum_{y'=-\infty}^{\infty} f(x', y') g(x - x', y - y')$$

Convolutions: some math

Let us start with general 2 dimensional functions

$$(f * g)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') g(x - x', y - y') dx'\, dy'$$

But images are discrete representations of 2D functions

$$(f * g)(x, y) = \sum_{x'=-\infty}^{\infty} \sum_{y'=-\infty}^{\infty} f(x', y') g(x - x', y - y')$$

And, finally, images have finite support

$$(f * g)(x, y) = \sum_{x'=0}^{X} \sum_{y'=0}^{Y} f(x', y') g(x - x', y - y')$$

Convolutions: some math

Question: what is g for moving average?

$$(f * g)(x, y) = \sum_{x'=0}^{X} \sum_{y'=0}^{Y} f(x', y') g(x - x', y - y')$$

Convolutions: some math

Question: what is g for moving average?

$$(f * g)(x, y) = \sum_{x'=0}^{X} \sum_{y'=0}^{Y} f(x', y') g(x - x', y - y')$$

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

This operation & the corresponding matrix are also called filters, kernels, convolutional matrices.

Convolutions: moving average

How does it modify images?

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Convolutions: moving average

How does it modify images?

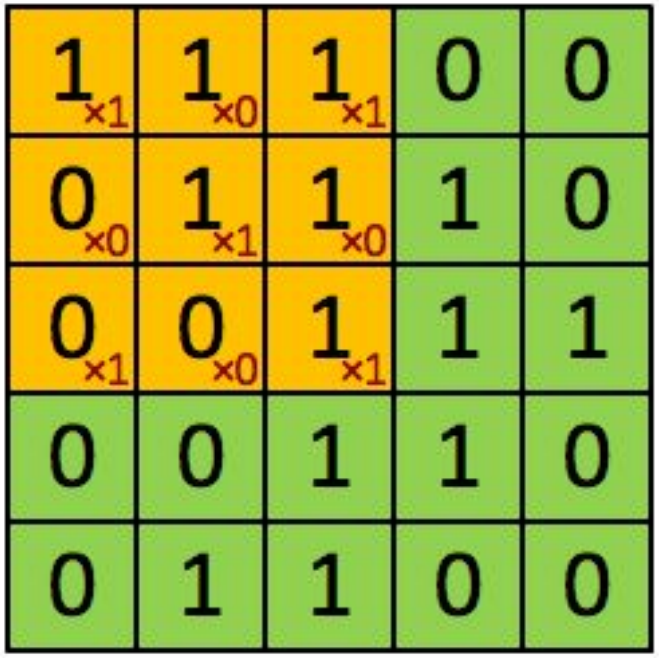
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Original image

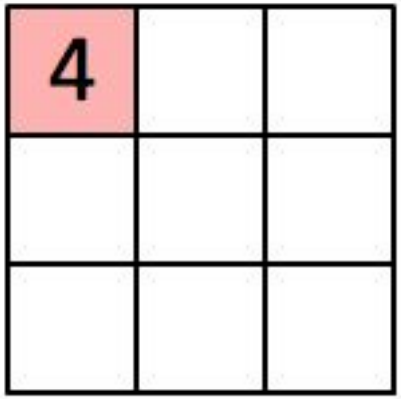
Smoothed image

Source: J. Niebles

Convolutions in 2D



Image



Convolved Feature

Source: [Stanford deep learning tutorial.]

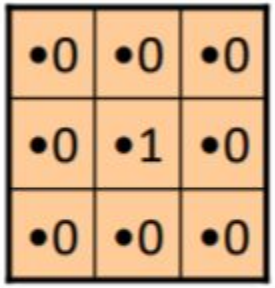
Convolutions in 2D: Practical exercise 1

Let's implement a Python function that convolves an image with a kernel.

$$(f * g)(x, y) = \sum_{x'=0}^{X} \sum_{y'=0}^{Y} f(x', y') g(x - x', y - y')$$

Convolutions: examples of filters

How does it modify images?

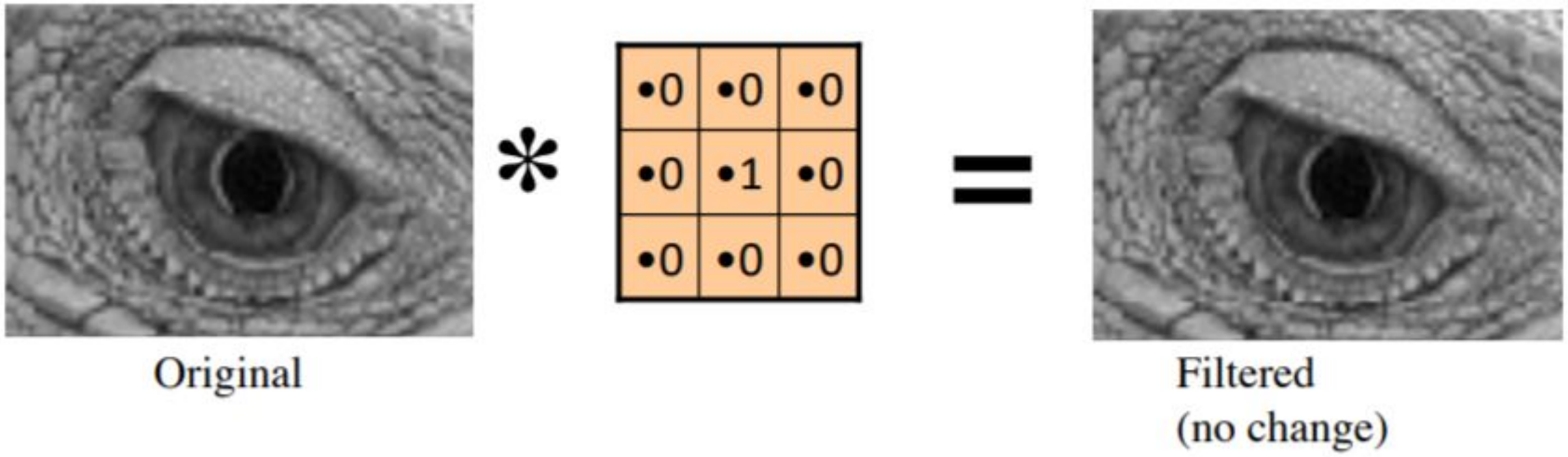


Source: J. Niebles

Convolutions: examples of filters

How does it modify images?

•0	•0	•0
•0	•1	•0
•0	•0	•0



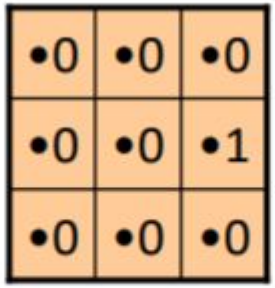
Original

Filtered
(no change)

Source: J. Niebles

Convolutions: examples of filters

How does it modify images?

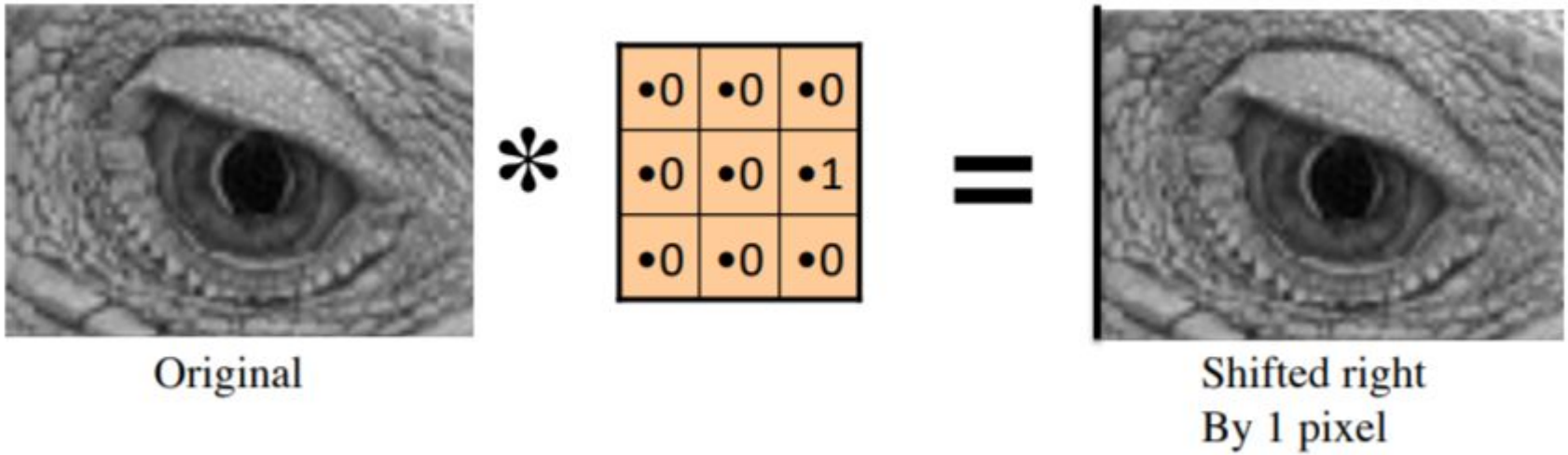


Source: J. Niebles

Convolutions: examples of filters

How does it modify images?

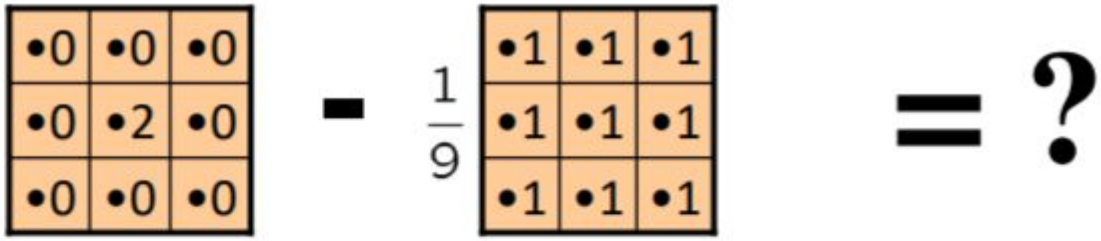
•0	•0	•0
•0	•0	•1
•0	•0	•0



Source: J. Niebles

Convolutions: examples of filters

How does it modify images?



Source: J. Niebles

Convolutions: examples of filters

Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = ?$$

(Note that filter sums to 1)

"details of the image"

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Source: J. Niebles

Convolutions: examples of filters

- Let's add it back:

Source: J. Niebles

Convolutions: examples of filters

Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = ?$$

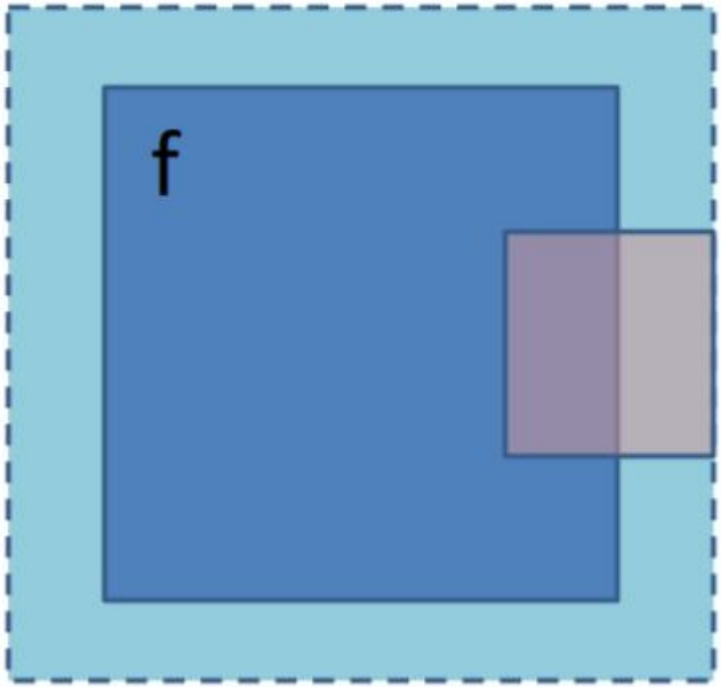
(Note that filter sums to 1)

Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$

Source: J. Niebles

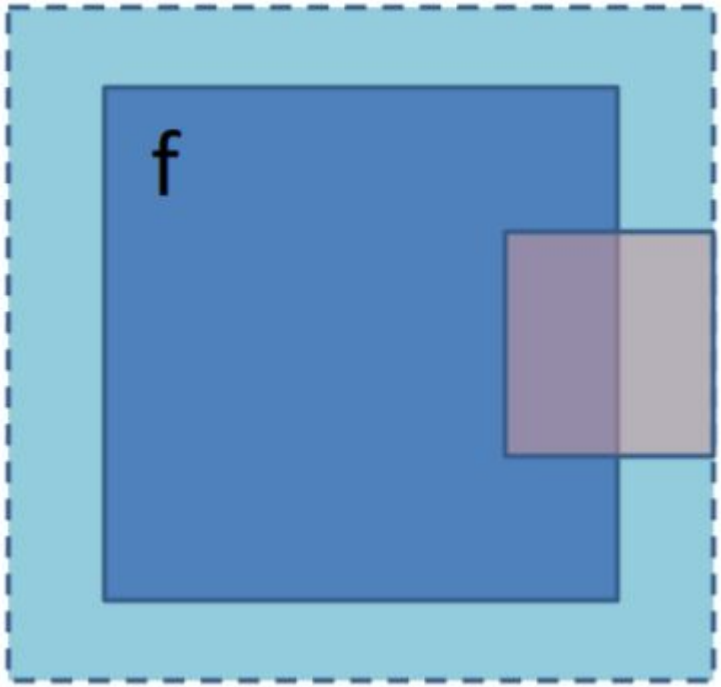
Convolutions: boundary conditions



How to process image boundaries?

Source: J. Niebles

Convolutions: boundary conditions



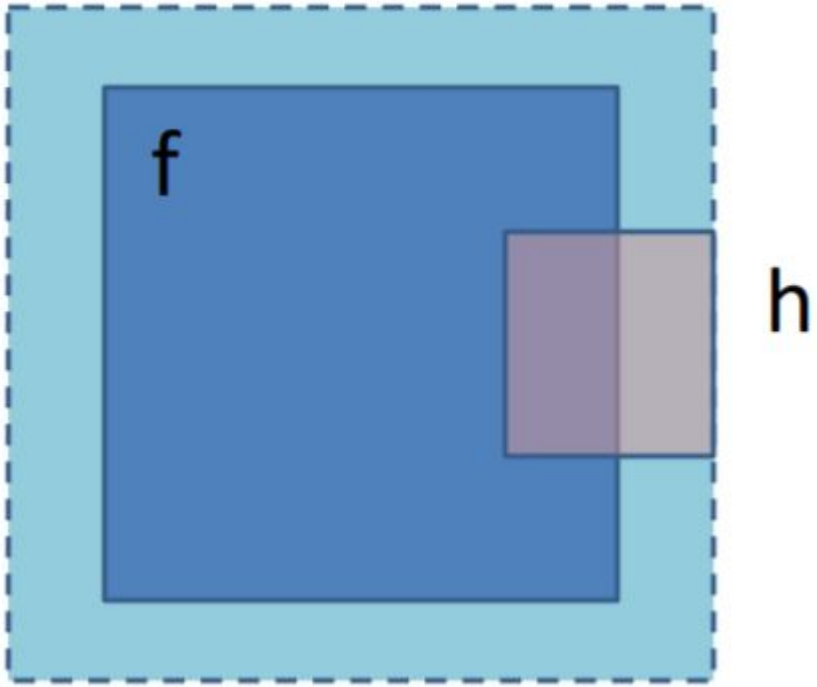
How to process image boundaries?

- **Reduce size** of the processed image
- **Add zeros** or constans
- Mirror the image

Source: J. Niebles

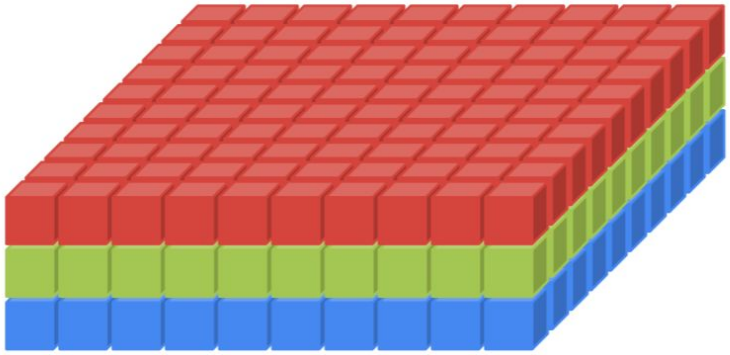
Convolutions: boundary conditions - Task 2

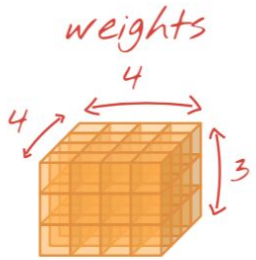
Let’s extend our convolution function with simple zero padding.



Source: J. Niebles

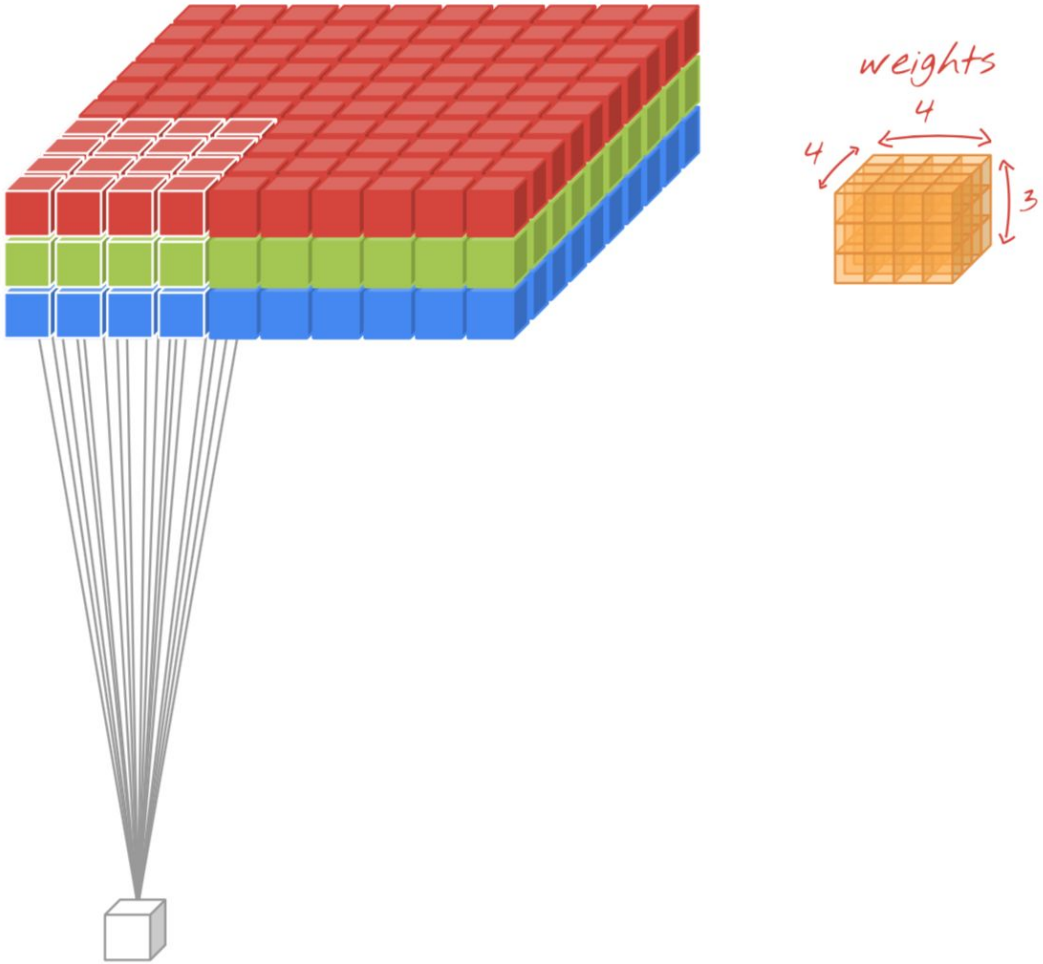
Convolution in 3D





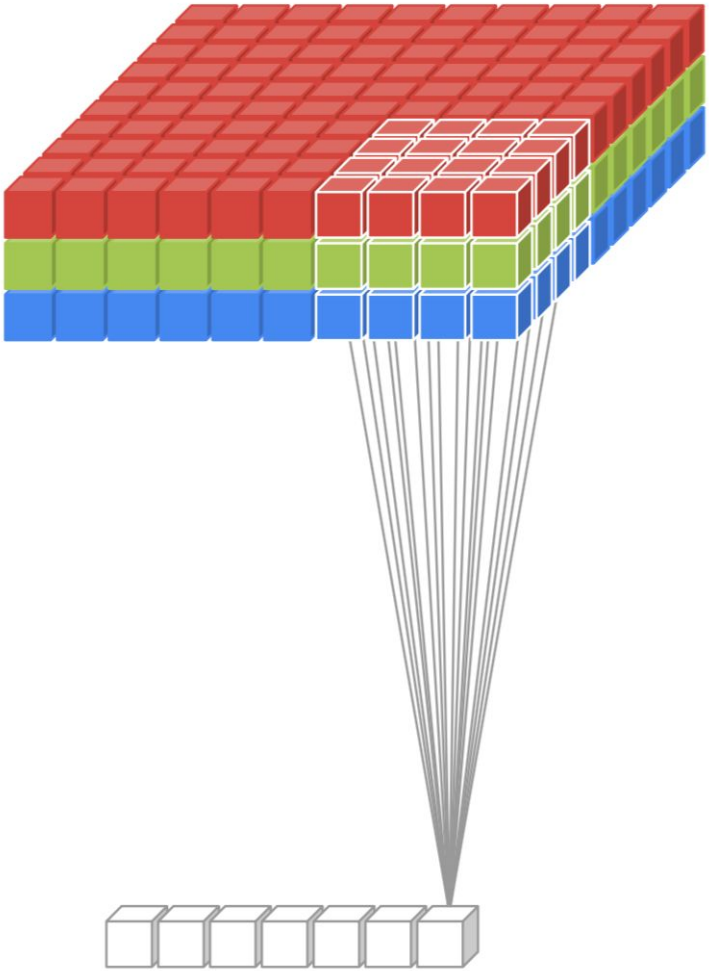
Source: https://twitter.com/martin_gorner

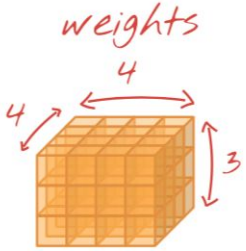
Convolution in 3D



Source: https://twitter.com/martin_gorner

Convolution in 3D



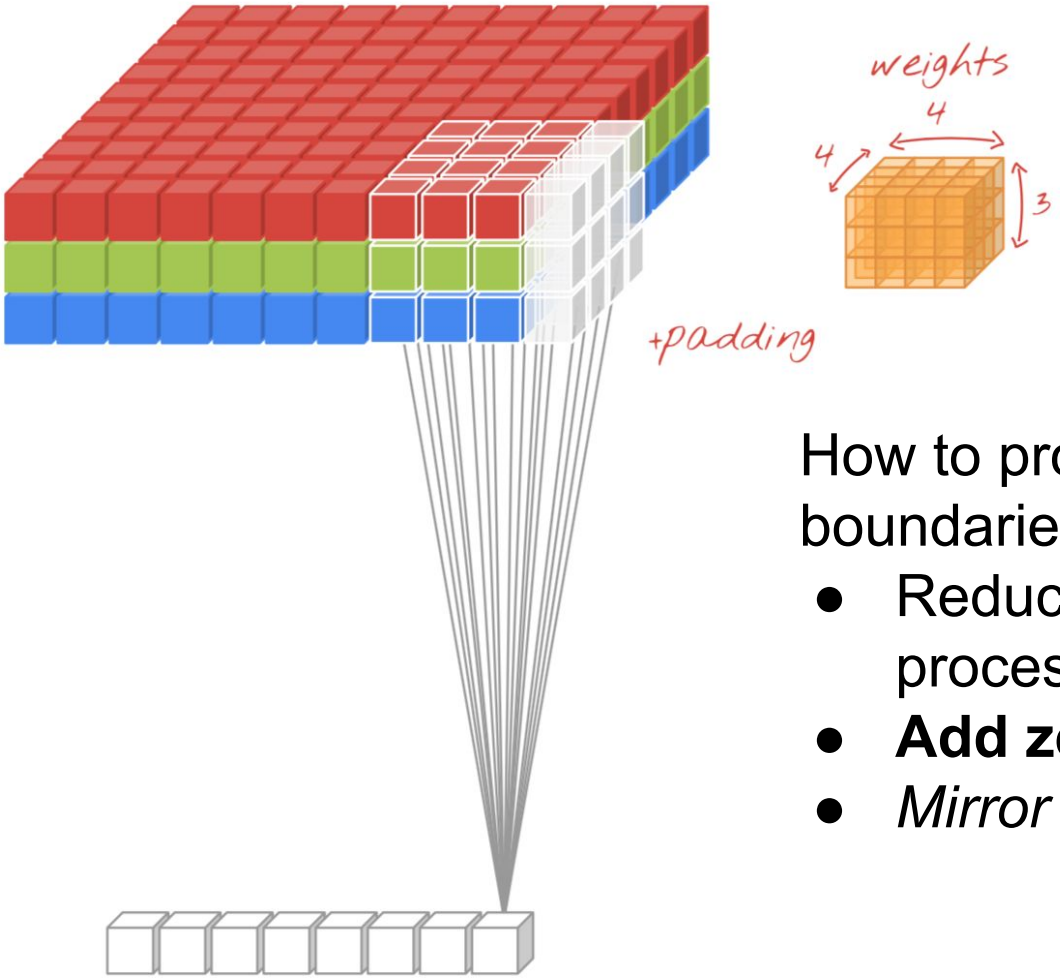


How to process image boundaries?

- **Reduce size** of the processed image
- Add zeros or constans
- *Mirror the image*

Source: https://twitter.com/martin_gorner

Convolution in 3D

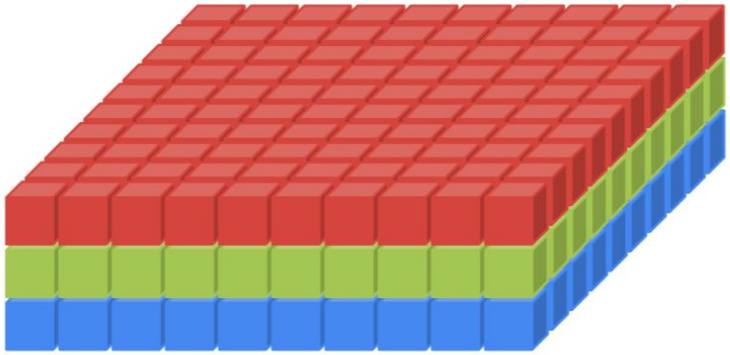


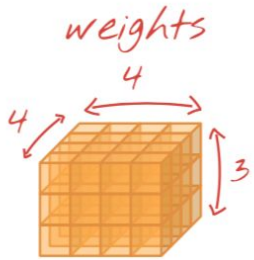
How to process image boundaries?

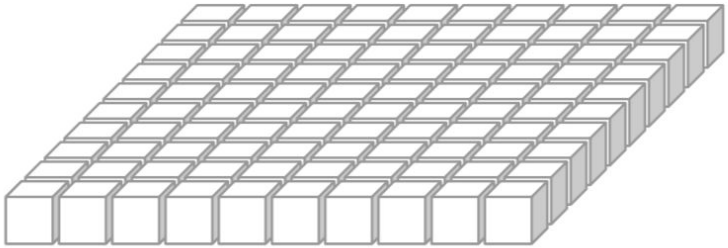
- Reduce size of the processed image
- **Add zeros** or constans
- *Mirror the image*

Source: https://twitter.com/martin_gorner

Convolution in 3D







Outline

- Info about the course
- Computer Vision problems & applications
- Convolutions
- ***Medical Computer Vision***