Computer Vision

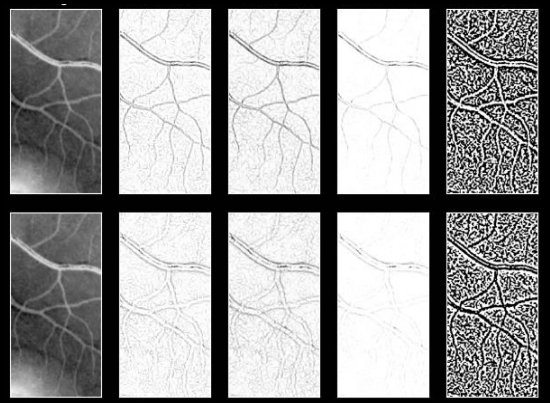
Hackathon Crash Course

Sam Sweere
Werner van der Veen

Purpose: gain high-level understanding from digital images or videos

Approaches:

- Classical
  - Fast & lightweight, but limited
  - Deterministic
  - Easily parametrizable
  - Suitable for edge computing, baselines
    - Examples: edge detection, denoising, inpainting, thresholding → *features*

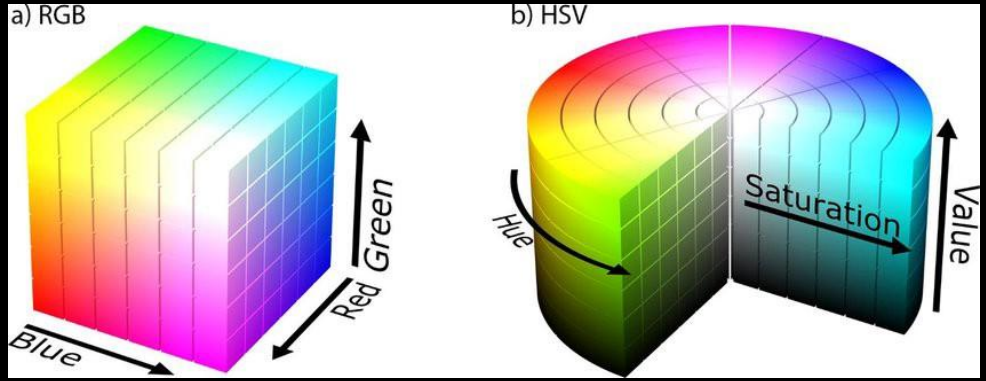


- Deep learning
  - Heavy but powerful
  - Compute/data hungry
  - Can also be interpretable
    - Examples: classification. detection, segmentation, generation → *meaning*

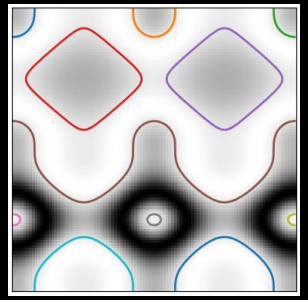
Classical Computer Vision methods
— Before we start: Handling image data —

- Images are rasters of pixel values (W × H × C)
  - `np.ndarray`
  - (R, G, B) vs. (H, S, V)
  - `opencv, PIL, imageio`
  - Often `np.bool`, `np.uint8`, `np.uint16` or np.float16
  - Easily displayed using `plt.imshow()`
  - `scikit-image`

```
import imageio

im = imageio.imread("image.png")

print(im.shape)

>>> (100, 200, 3)
```

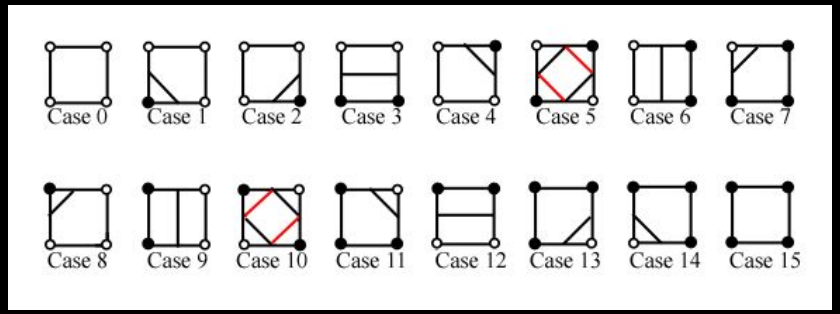


Classical Computer Vision methods
— Edges: finding iso-valued contours —



Marching Squares:

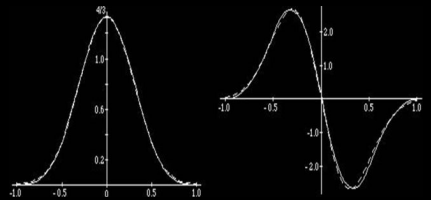
`skimage.measure.find_contours()`



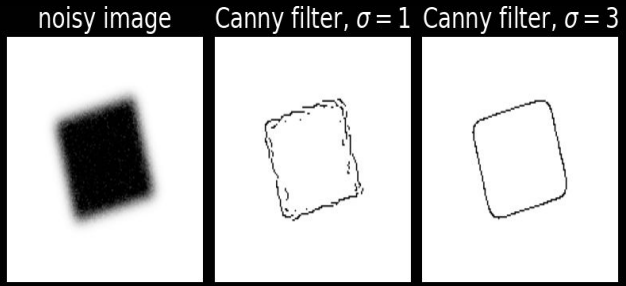
Algorithm:

In every grid of 2x2 pixels, draw an isoline according to whether its 4 pixels are lower or higher than the isovalue.

Classical Computer Vision methods
— Edges: edge detection —



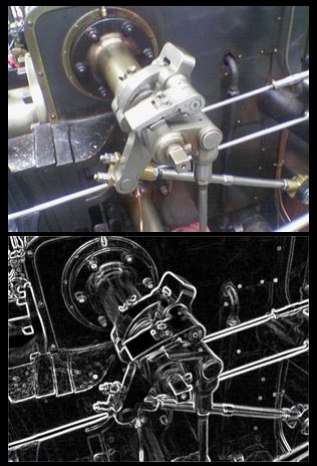
Canny filter



Multi-stage:

1. derivative of Gaussian (parameter σ is width)
2. edges thinned to 1-pixel by keeping only gradient maxima
3. remove weak, unconnected edges

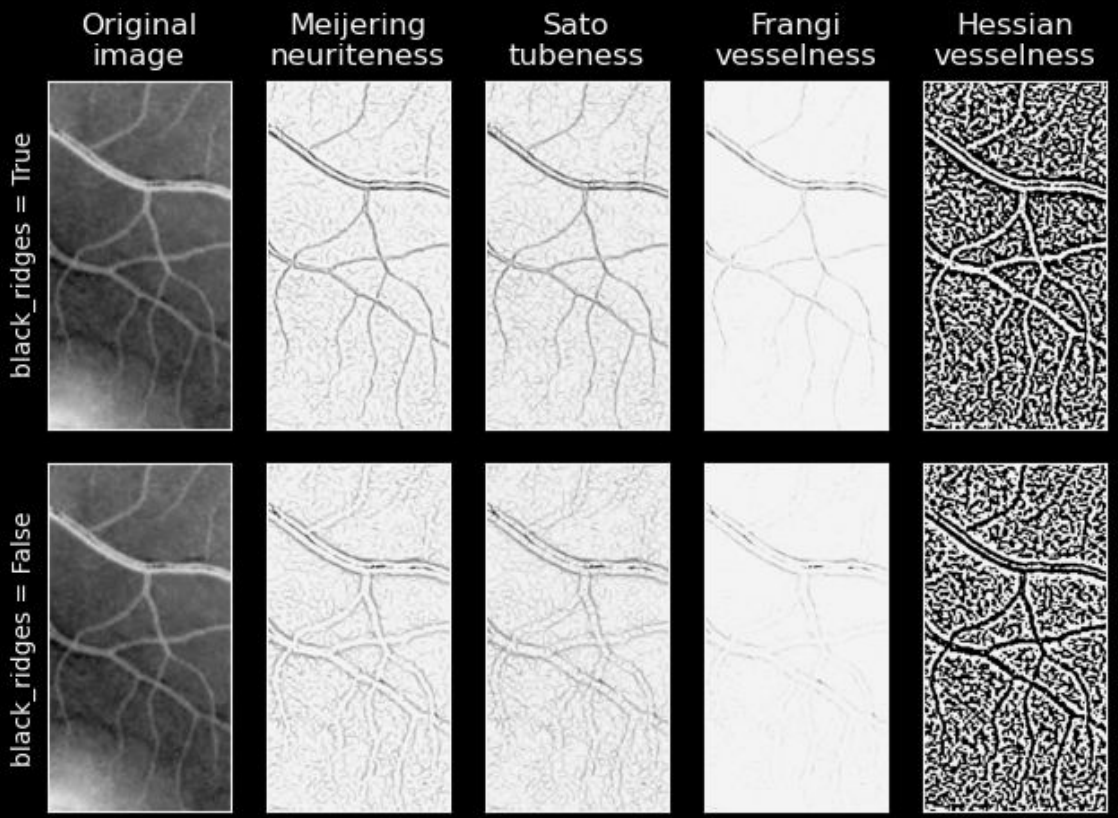
Sobel filter



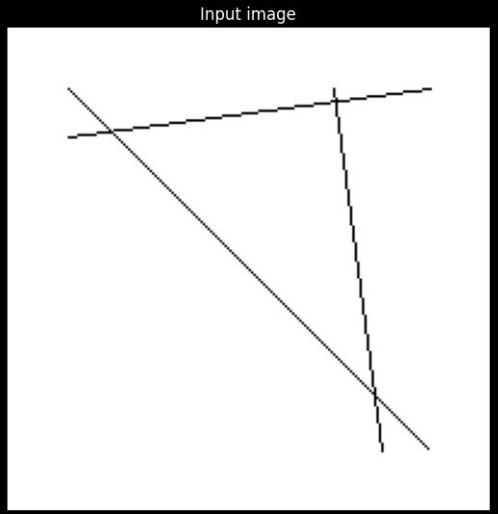
$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

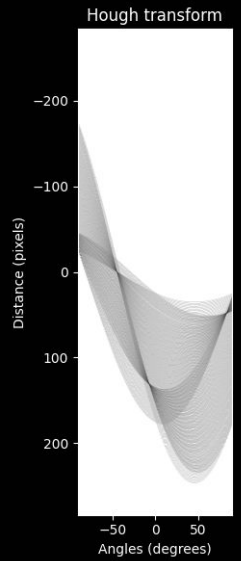
$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2} \qquad \mathbf{\Theta} = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x)$$

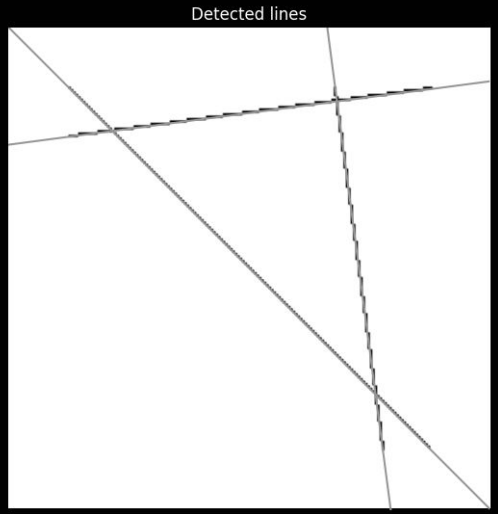
Classical Computer Vision methods
— Edges: ridge filters —



Classical Computer Vision methods
— Straight line detection: Hough transform —







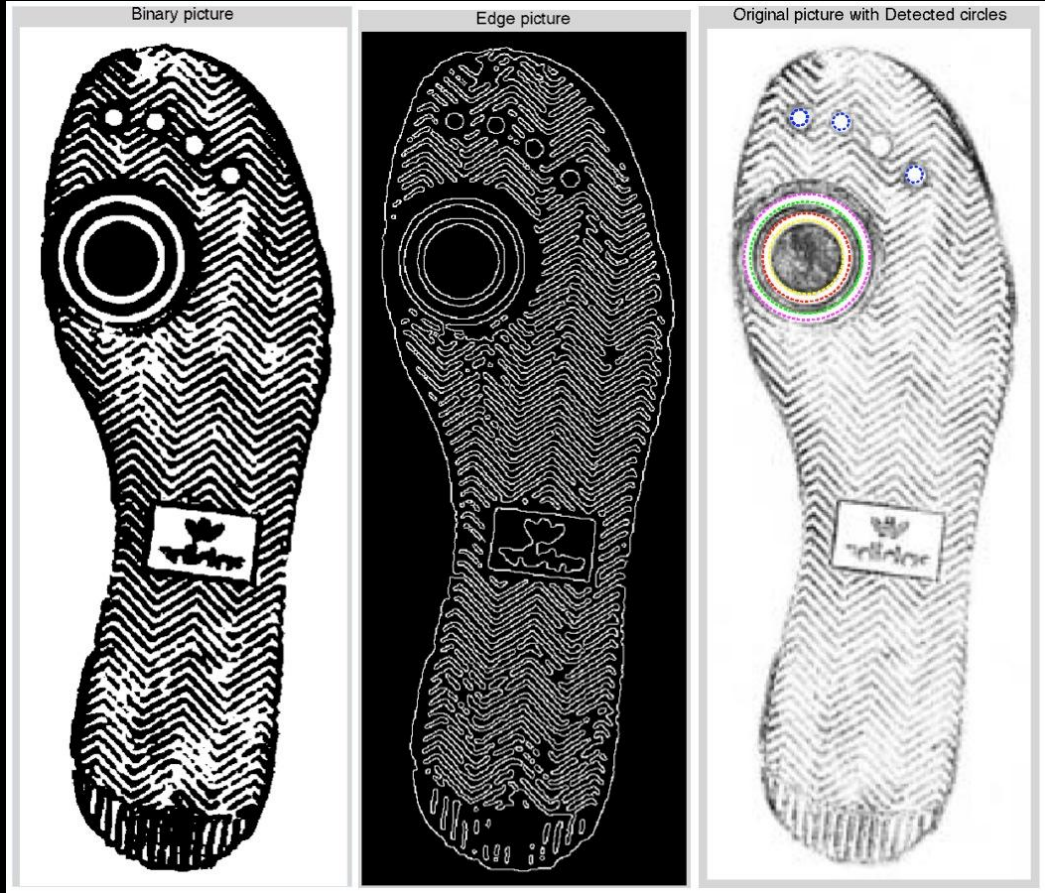
Algorithm:

1. For each point, draw lines with angle θ through it and find length *r* of its vector orthogonal to the origin.
2. Draw all (θ, *r*) pairs of all points in heatmap
3. Local heatmap maxima indicate lines in image

Classical Computer Vision methods
— Circle detection: Hough transform —

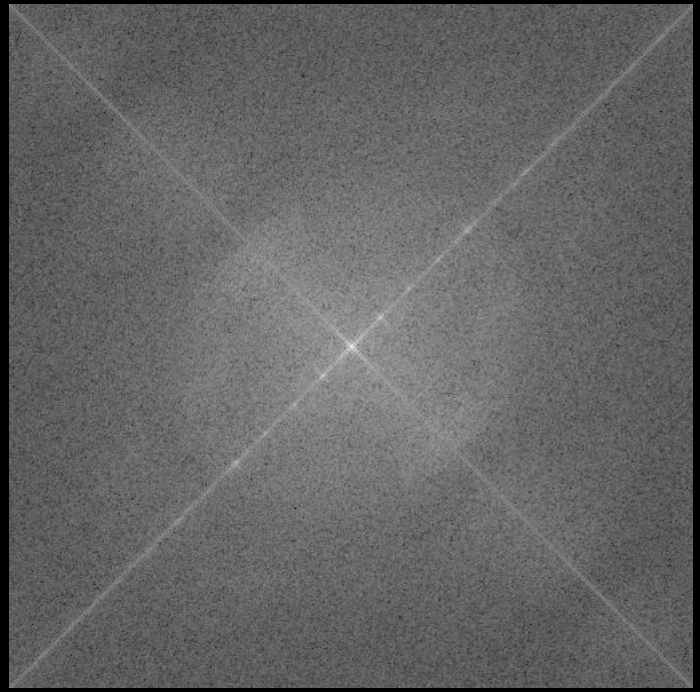
Algorithm:

1. For each pixel (x, y) and every radius $r \in R$, count the number of edge pixels on that circle.
2. Draw all (x, y, r) pairs of all points in a 3D heatmap.
3. Local heatmap maxima indicate circles in the original image



Classical Computer Vision methods
— Corner detection: Fourier transform —

- Decompose image signal into spectral components
- Analysis of
  - speed of change (distance from center)
  - direction (angle from center)
- Can be combined with e.g. edge detection



Classical Computer Vision methods
— Corner detection: Moravec transform —

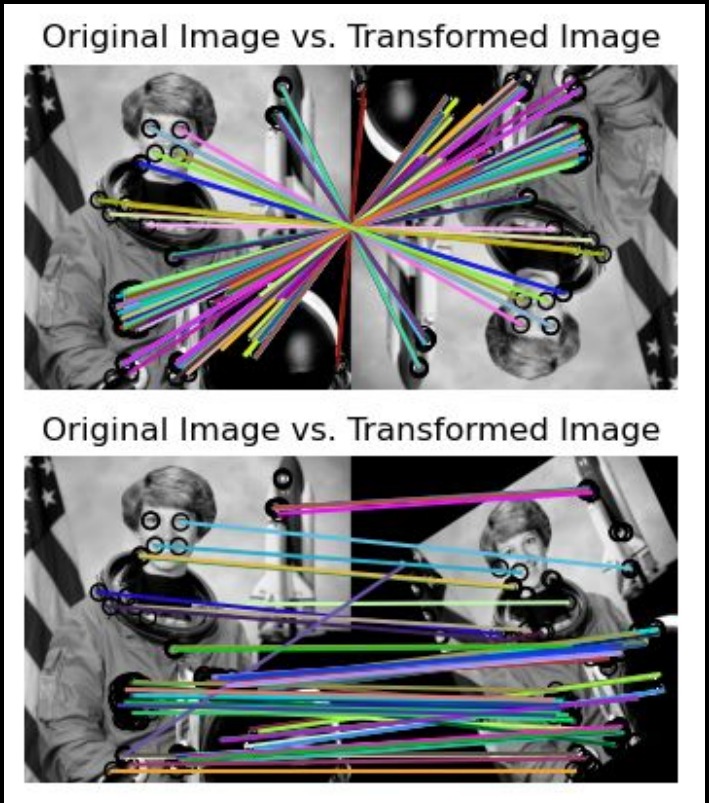
Algorithm:

1. For all local patches, the center is a corner if the patch is not similar to *any* nearby patches.
2. Corner strength is the dissimilarity.

Classical Computer Vision methods
— Matching: ORB —

Oriented FAST and rotated BRIEF (ORB)

- Comparatively rotation & scale invariant
- FAST: efficient corner detection method
- BRIEF: “fingerprints” keypoints for matching
- Fast enough for real-time applications

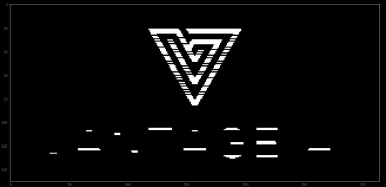


Classical Computer Vision methods
— Morphological processing —

Original

Opening (E+D)

(rectangle 1x4)



Dilation

(radius 2)

Closing (D+E)

(radius 6)

Erosion

(rectangle 1x5)

Area closing

(threshold 500)

Skeleton

Area opening

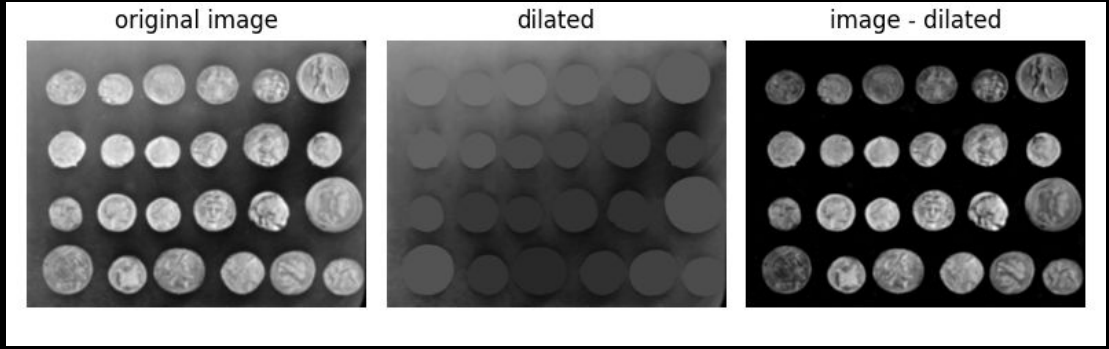
(threshold 150)

Works well with `skimage.measure.regionprops`, to give you each component's:

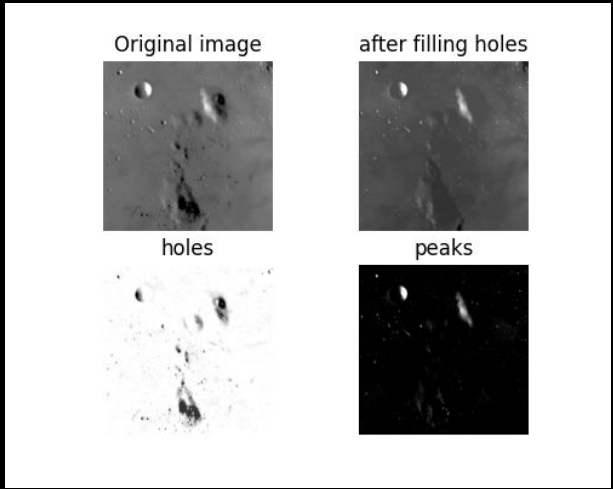
- area
- bounding box
- centroid
- eccentricity
- orientation
- convex hull image
- *and much more*

Classical Computer Vision methods
— Stacking morphological processing: simple and effective —

Reconstruction by dilation



Reconstruction by erosion and dilation

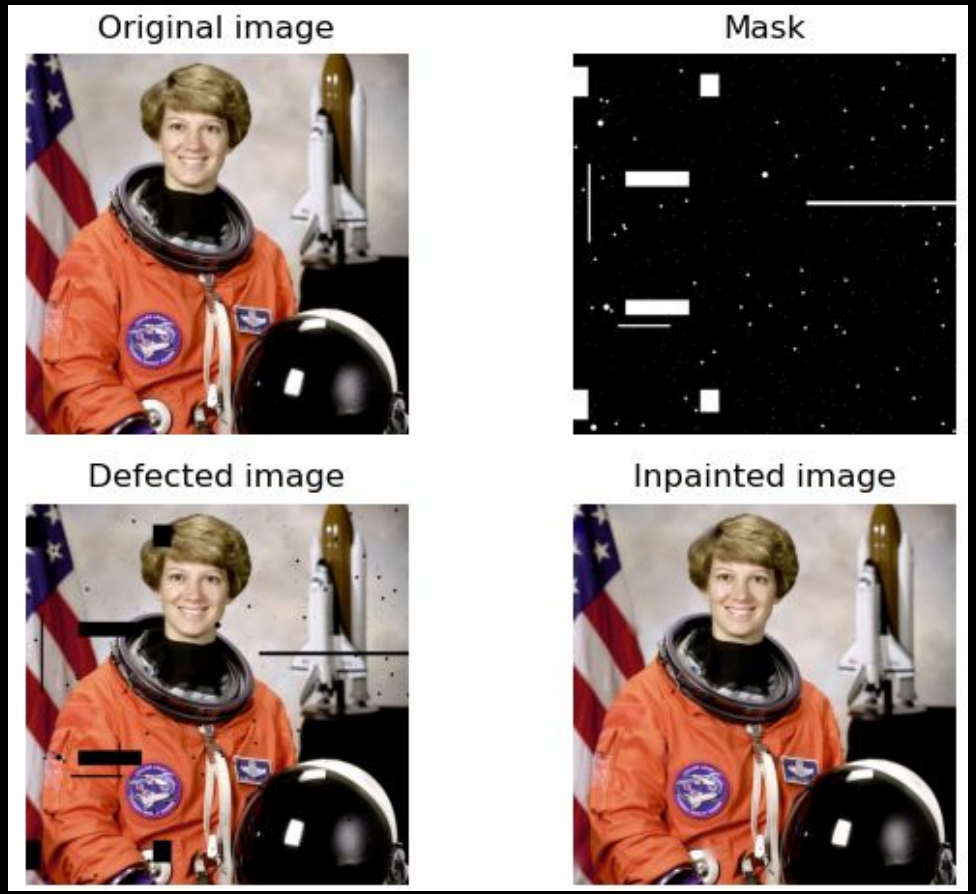


Method: use “seed” to specify values that spread, and “mask” to clip the values.

Classical Computer Vision methods
— Inpainting: biharmonic equation —

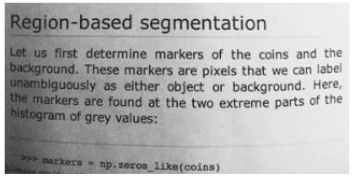
Biharmonic equation:

- Uses 4th-order partial differential equation to interpolate between surrounding patterns

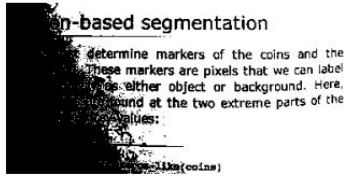


Classical Computer Vision methods
— Thresholding —

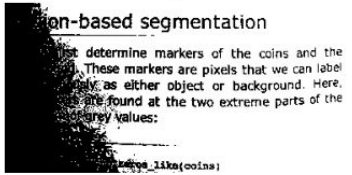
Original



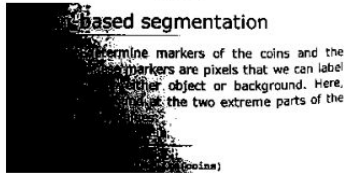
Isodata



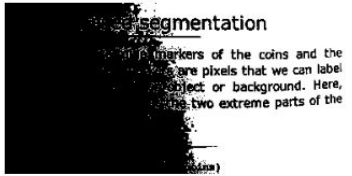
Li



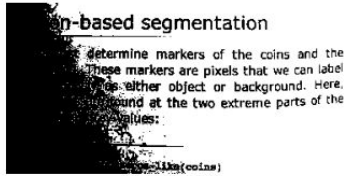
Mean



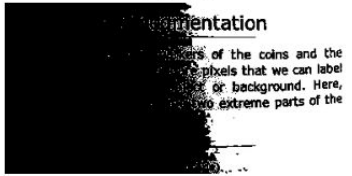
Minimum



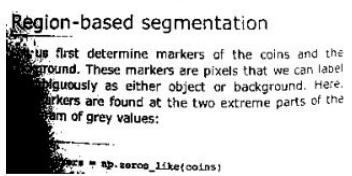
Otsu



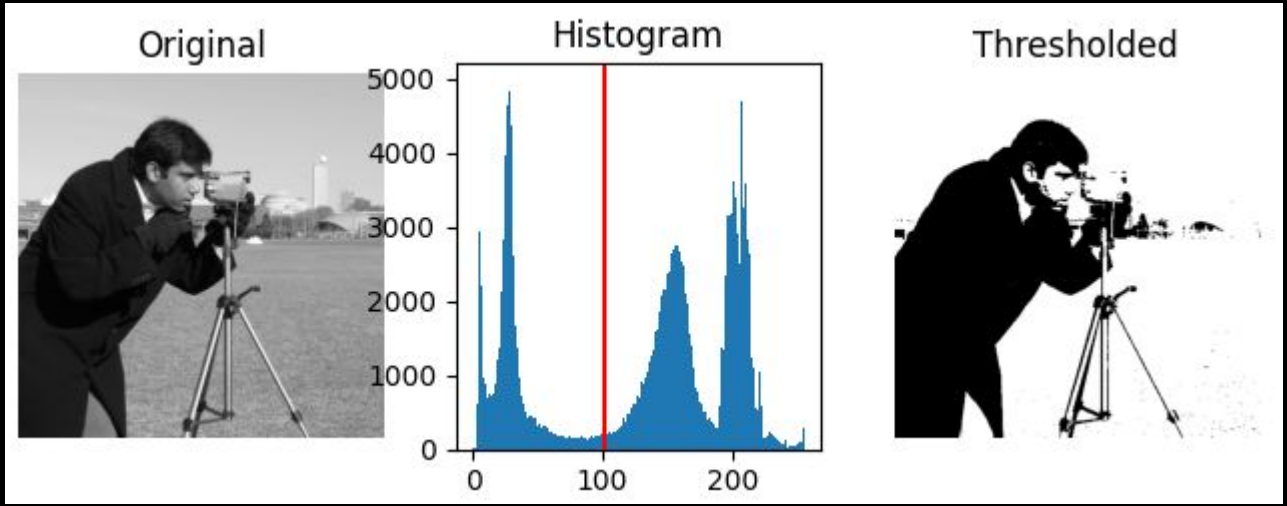
Triangle



Yen



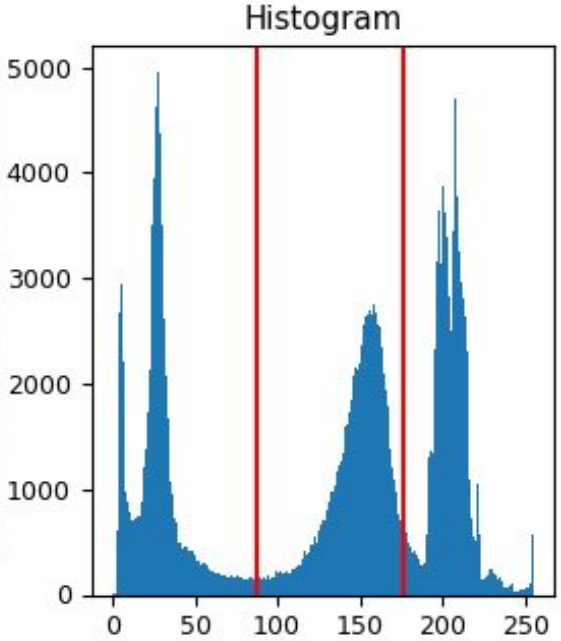
Classical Computer Vision methods
— Thresholding: Otsu—



“Optimal” threshold:

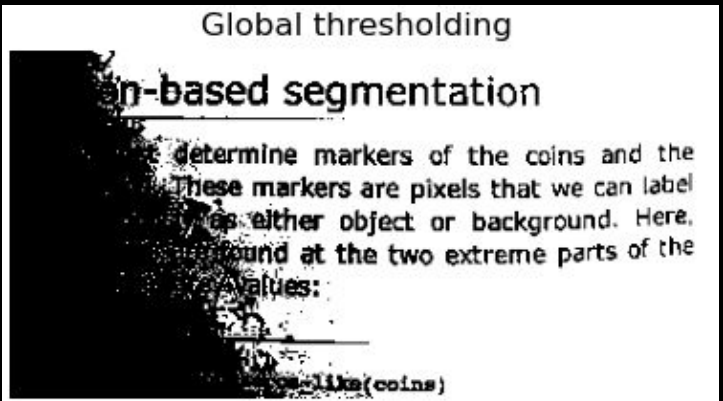
- Maximum variance between two classes
- (and therefore minimal intra-class variance)

Classical Computer Vision methods
— Thresholding: multi-Otsu—



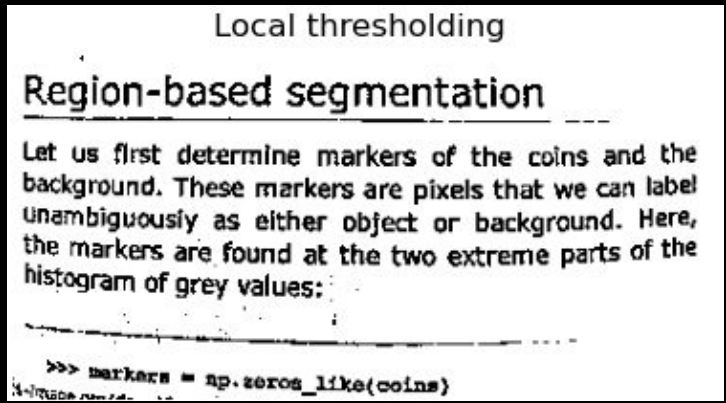
Classical Computer Vision methods
— Thresholding: local Otsu—

Global thresholding

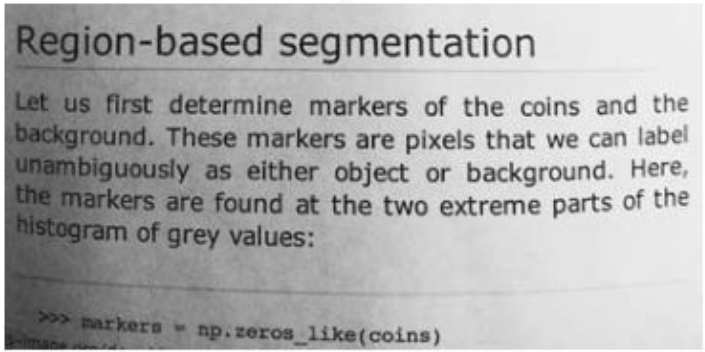


- Much slower, but more accurate
- Local neighborhood

Local thresholding

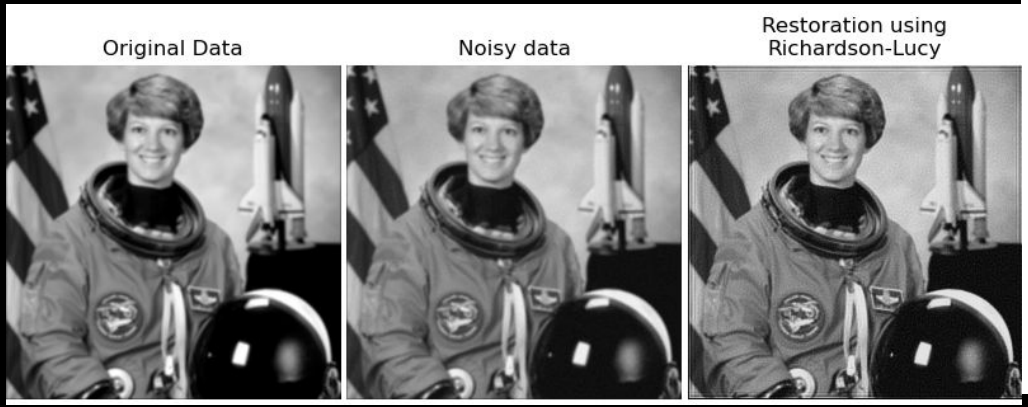


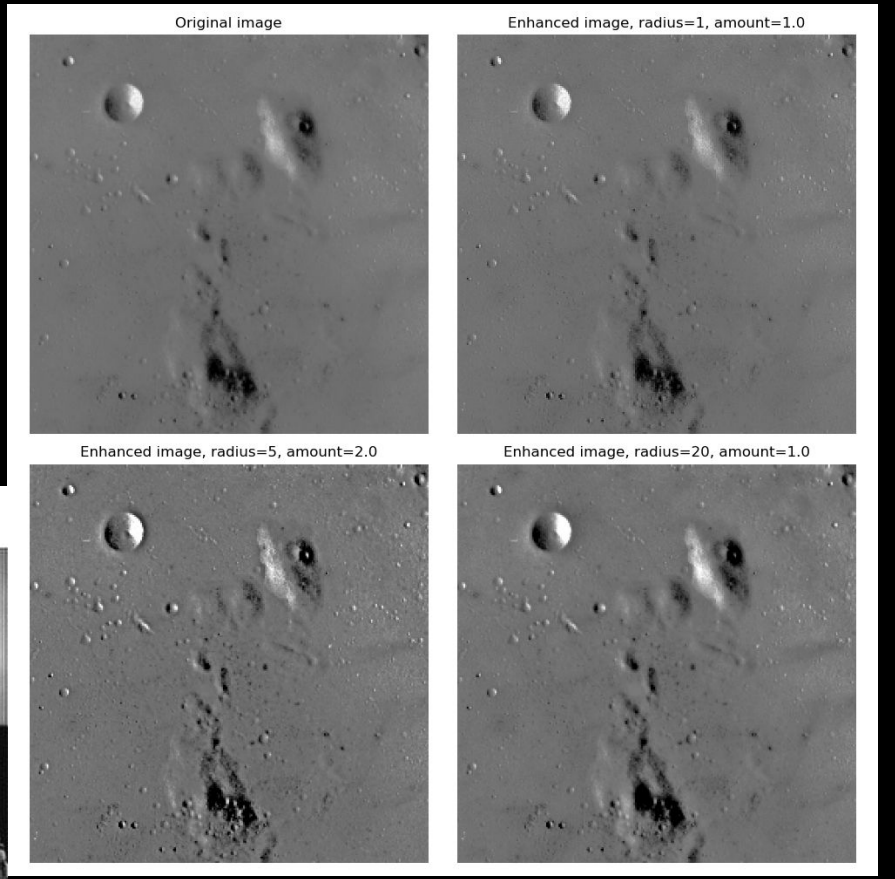
Original



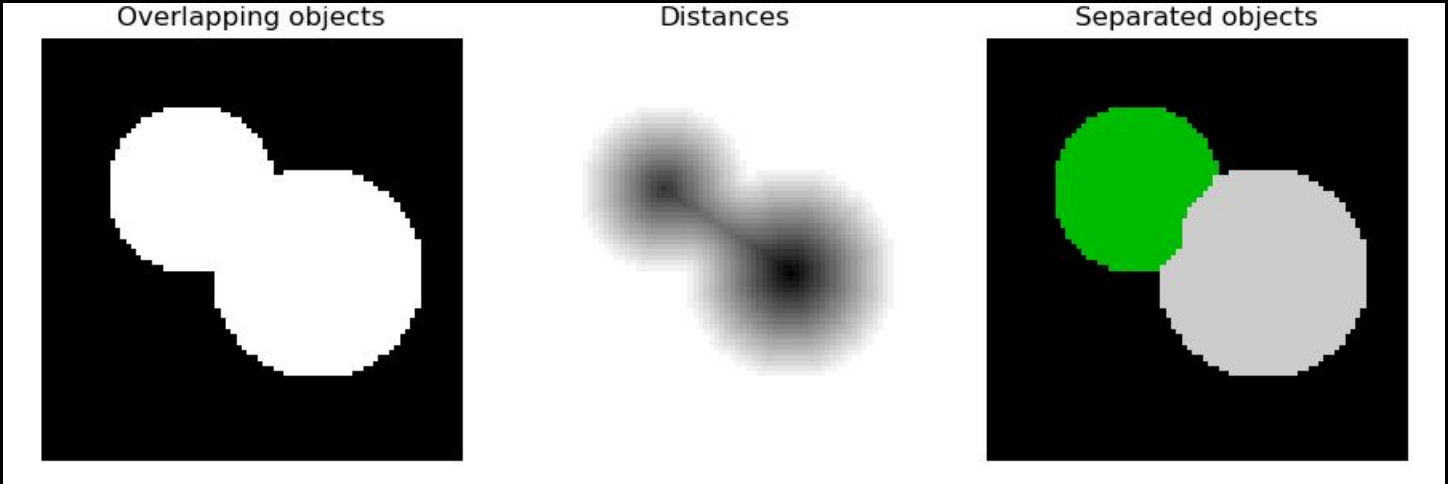
Classical Computer Vision methods
— Sharpening—

- Difference between original and blurred, scaled back, and added to original





Classical Computer Vision methods
— Segmentation: Watershed algorithm—



1. Calculate distance from background
2. Flood local maxima in distance map
3. Edges between basins are object boundaries

Deep Learning Computer Vision methods

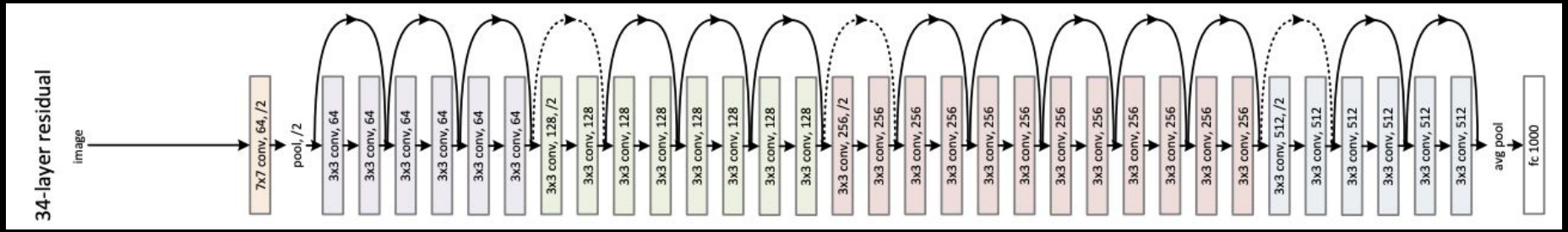
- Classification
- Object detection
- Segmentation
- Generation

Deep Learning Computer Vision methods
— Classification —

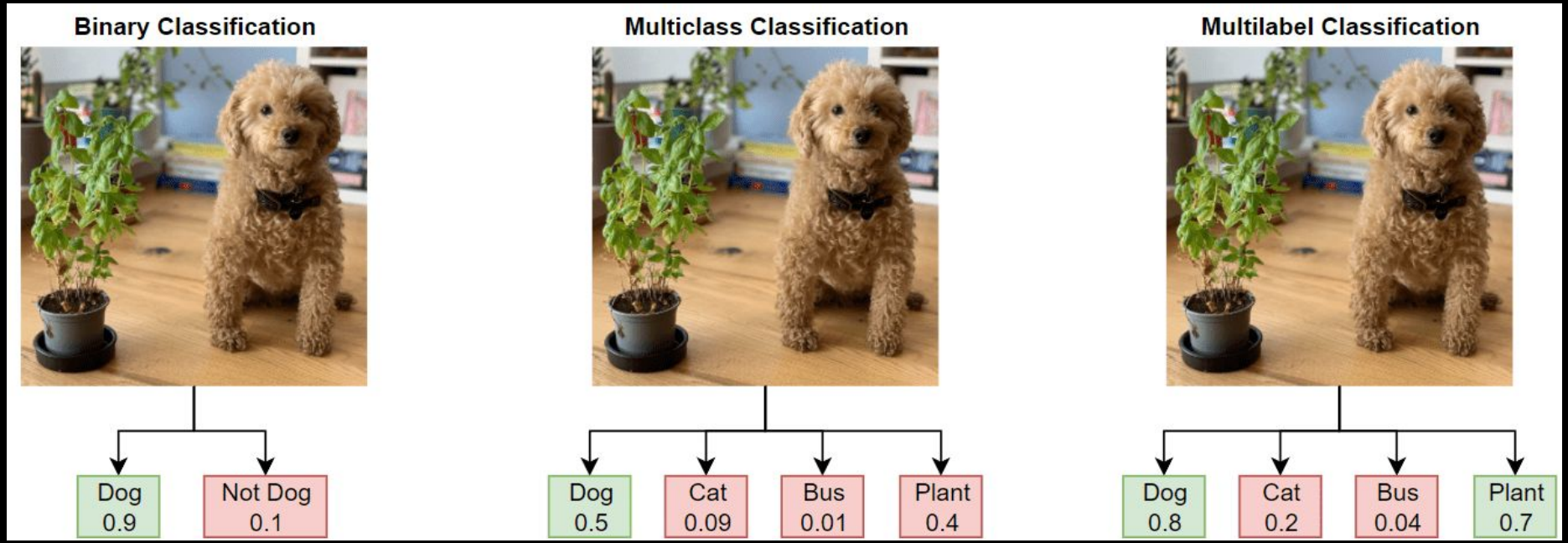
Pretty much always CNNs

Tried and true: ResNet pretrained on ImageNet

- Transfer learning (unfreeze only last *N* layers)
- Rescale input to [0, 1]
- Data augmentation & feature engineering
- Saliency maps
    - (Gradient weighted) class activation mapping

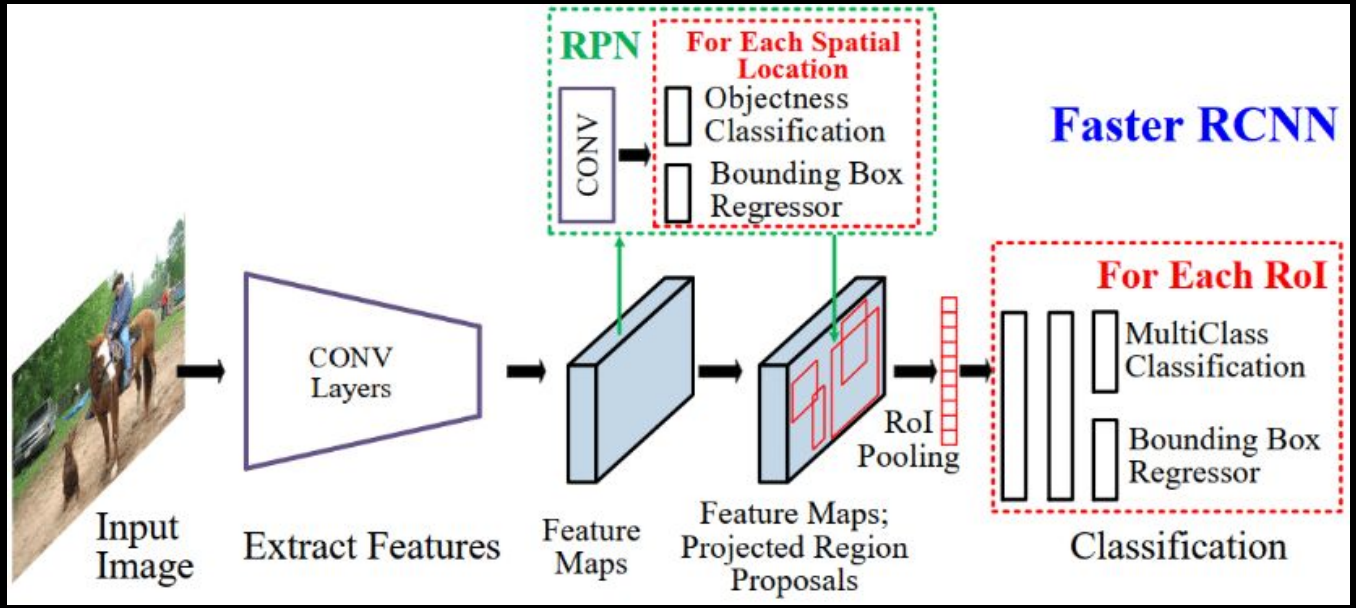


Deep Learning Computer Vision methods
— Classification —

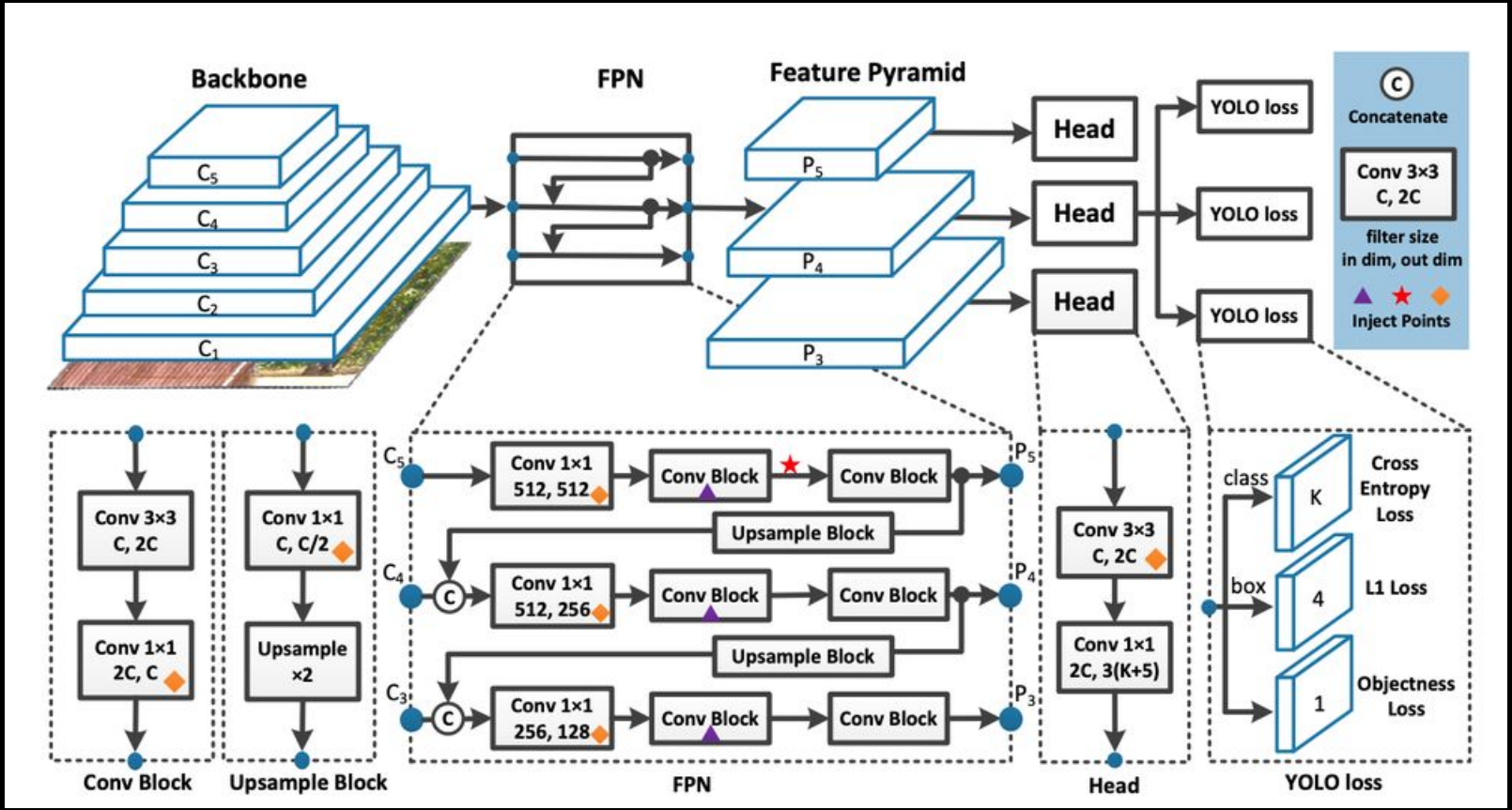


Deep Learning Computer Vision methods
— Object detection: two-shot (Faster RCNN) —

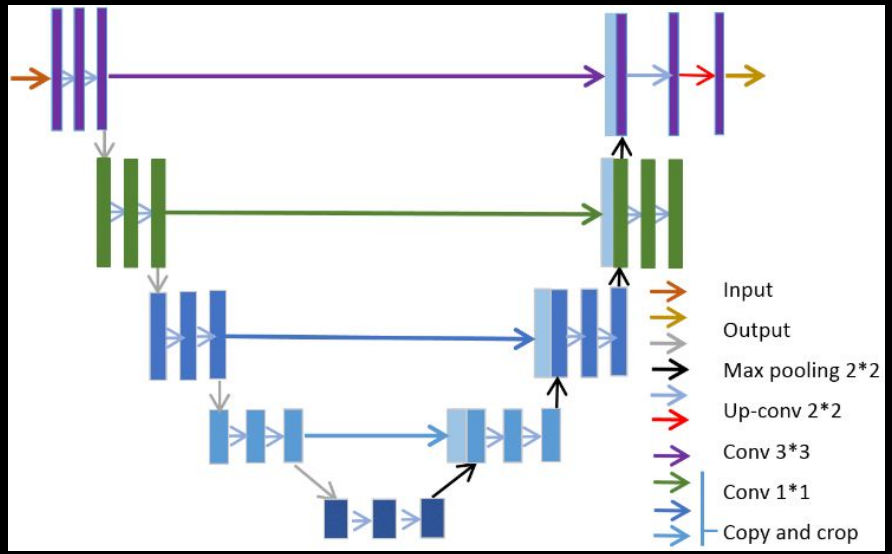
1. Feature extraction
2. Anchor box proposals
3. Prediction heads
4. Regression heads



Deep Learning Computer Vision methods
— Object detection: one-shot (YOLOv7) —



Deep Learning Computer Vision methods
— Segmentation: U-Net (2014) —



Semantic segmentation:

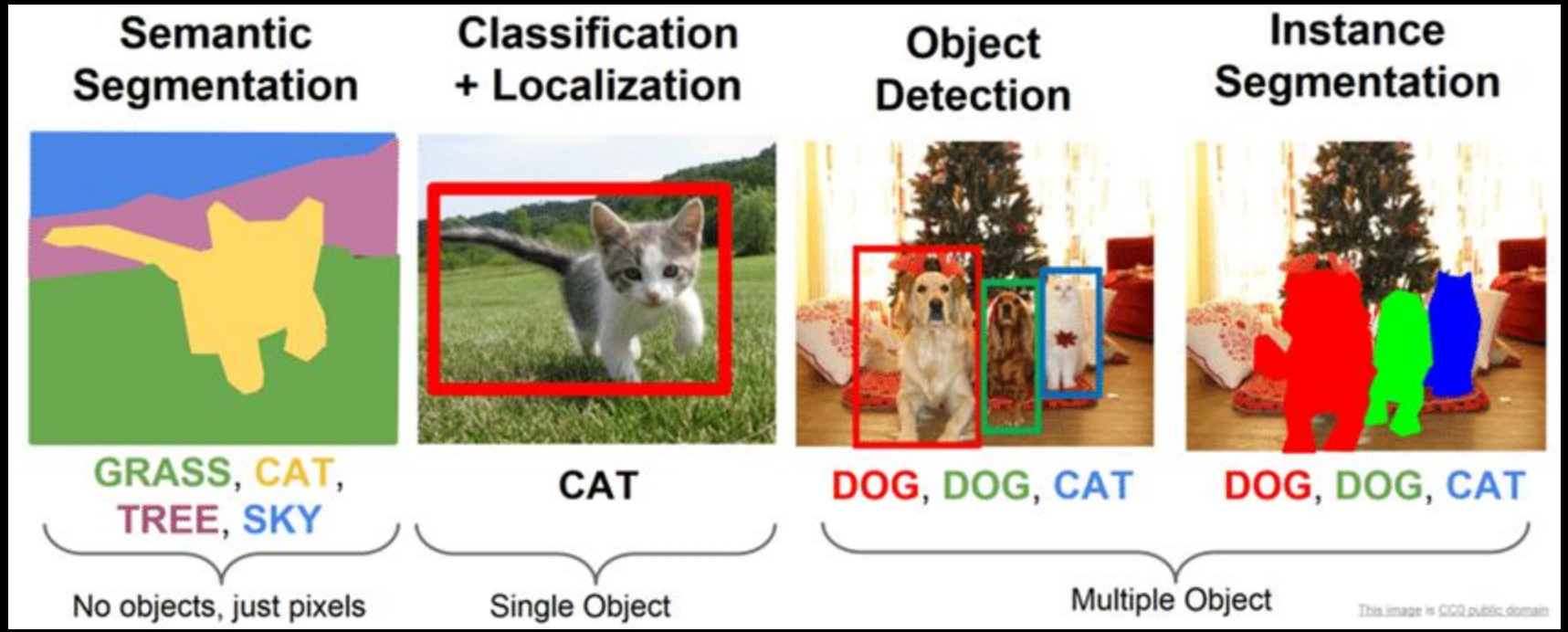
Image-wide pixel-to-pixel

Mostly used in biomedical imaging

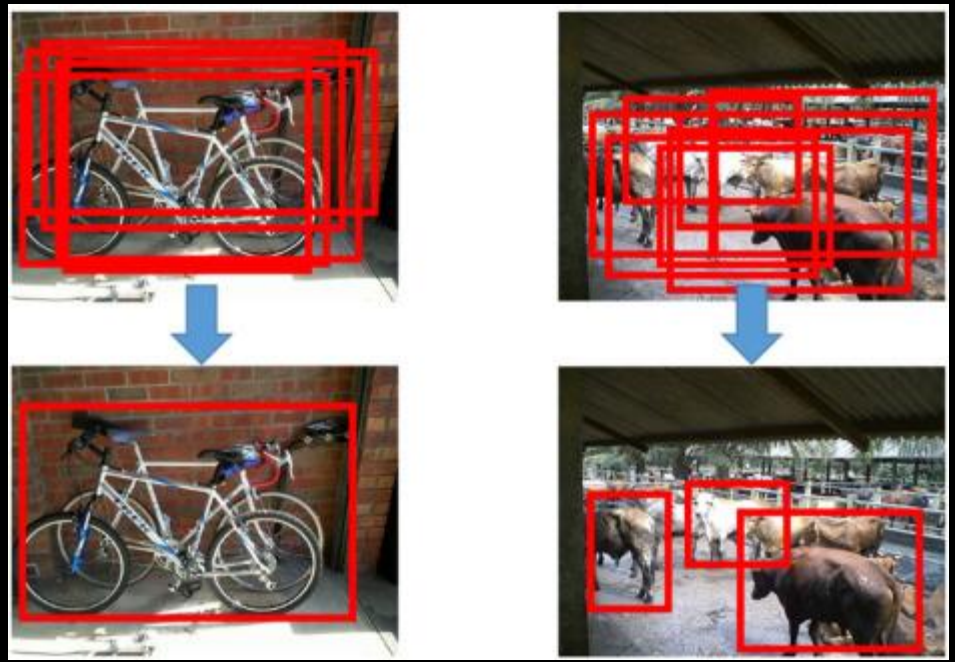
Metrics: Jaccard index, F_1-score

Usually pretrained on MS-COCO

Deep Learning Computer Vision methods
— Segmentation types —



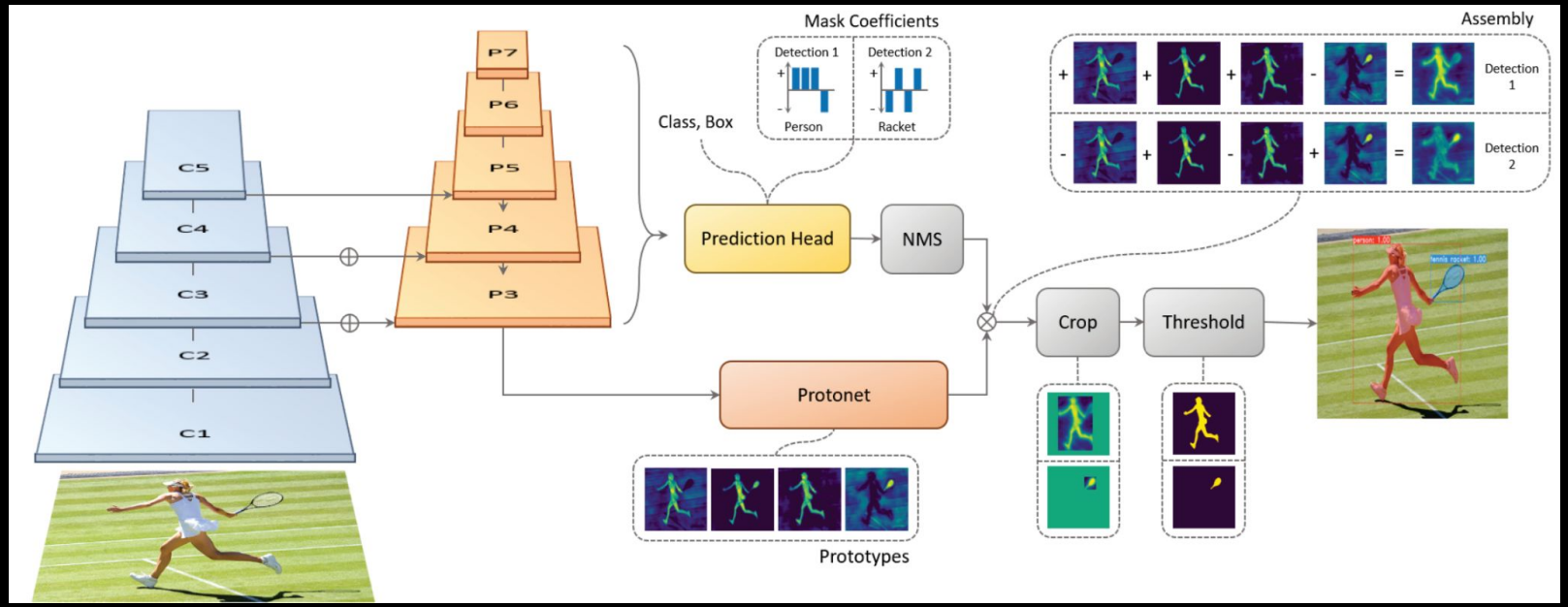
Deep Learning Computer Vision methods
— Segmentation Intermezzo: Non-Max Suppression —



Algorithm:

1. Select the box *s* with highest confidence, remove it from *P*, and add it to *K*.
2. If IoU(*s*, *p*) > θ for any $p \in P$, then remove *p* from *P*.
3. Repeat steps 1 & 2 until *P* is empty.
4. The list *K* contains the final boxes.

Deep Learning Computer Vision methods
— YOLACT —



Instance segmentation:

Image to (mask, class) pairs

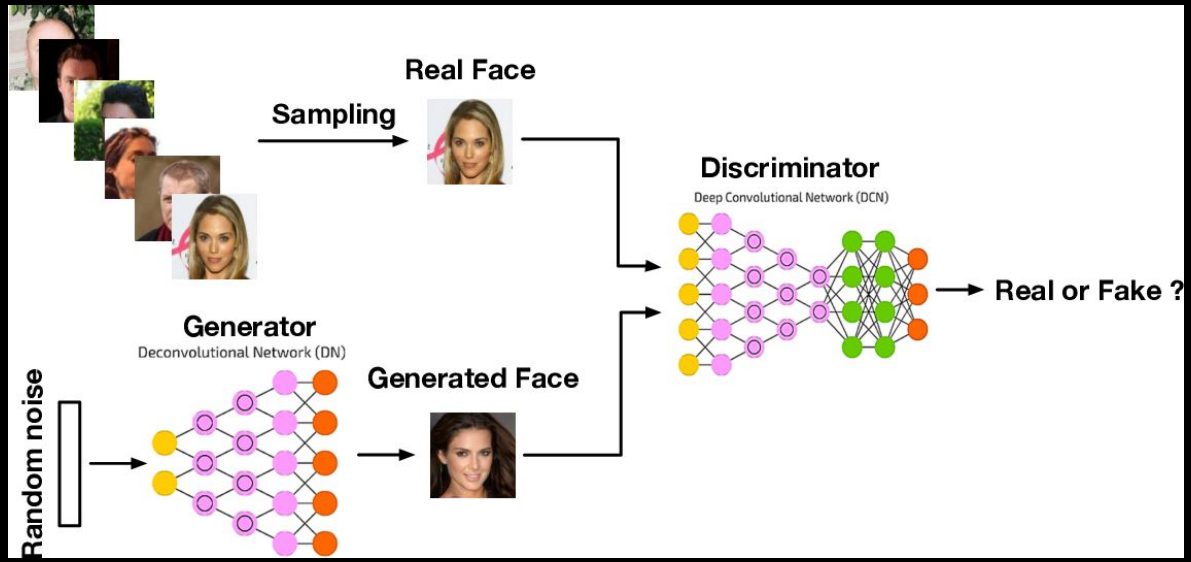
Metrics: mAP, $F_{\square}$-score

Insightful

Algorithm:

1. Propotype generation branch (semantic segmentation, but supervised by heads)
2. Mask coefficient branch (object detection–like, then non-max suppression)
3. Mask assembly (Linear combination of prototypes with coefficients from masks)
4. Crop, sigmoid, threshold to produce result.

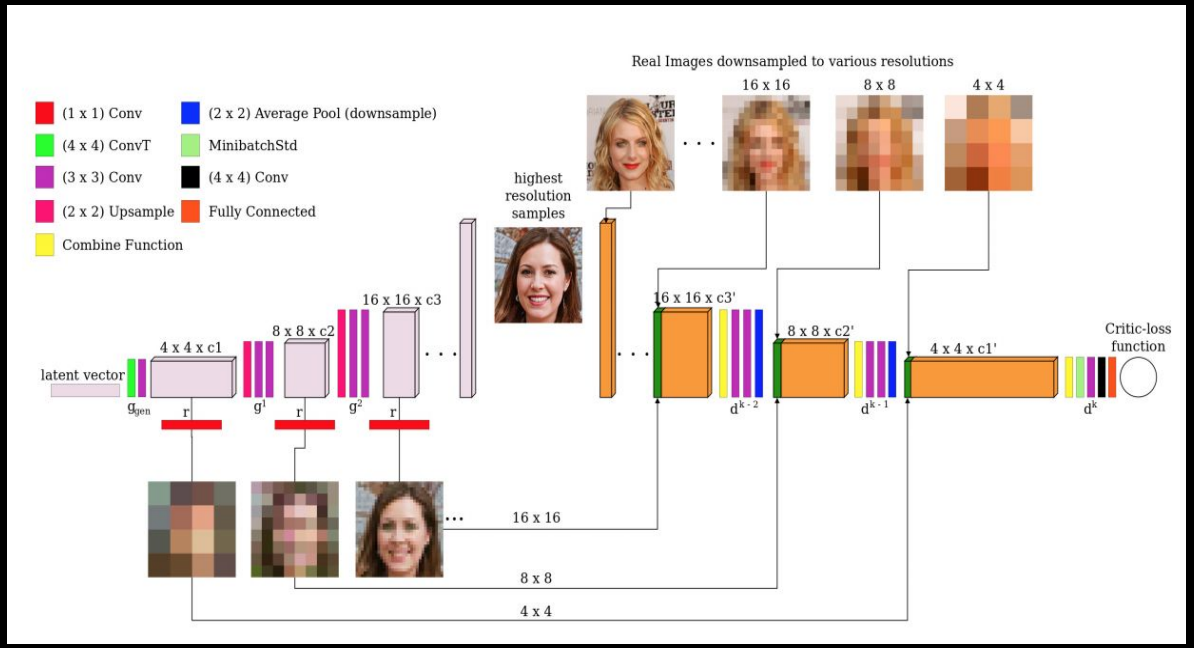
Deep Learning Computer Vision methods
— Generation: Adversarial networks—



- Pitting two networks against each other
- Generator metric: *Fréchet inception distance*
  - Compares distributions of Inception V3 feature maps of real and generated images
- Common challenge: overpowered discriminators

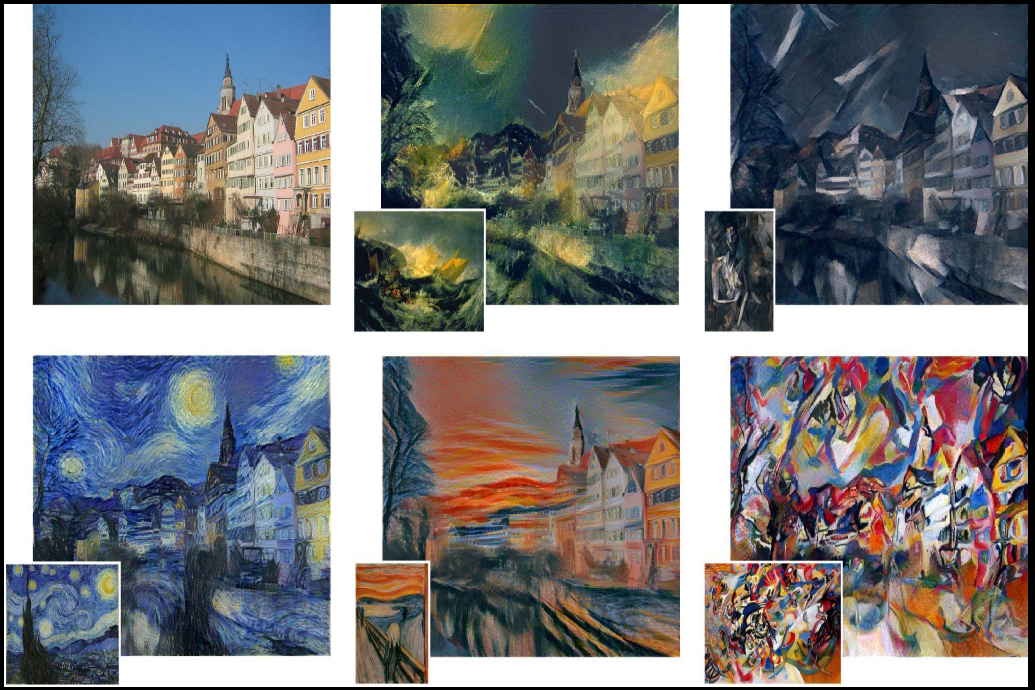
Deep Learning Computer Vision methods
— Generation: StyleGAN2 —

- Smooth latent space between any two faces
  - Through complex normalization layers
- Skip connections to improve upsampling



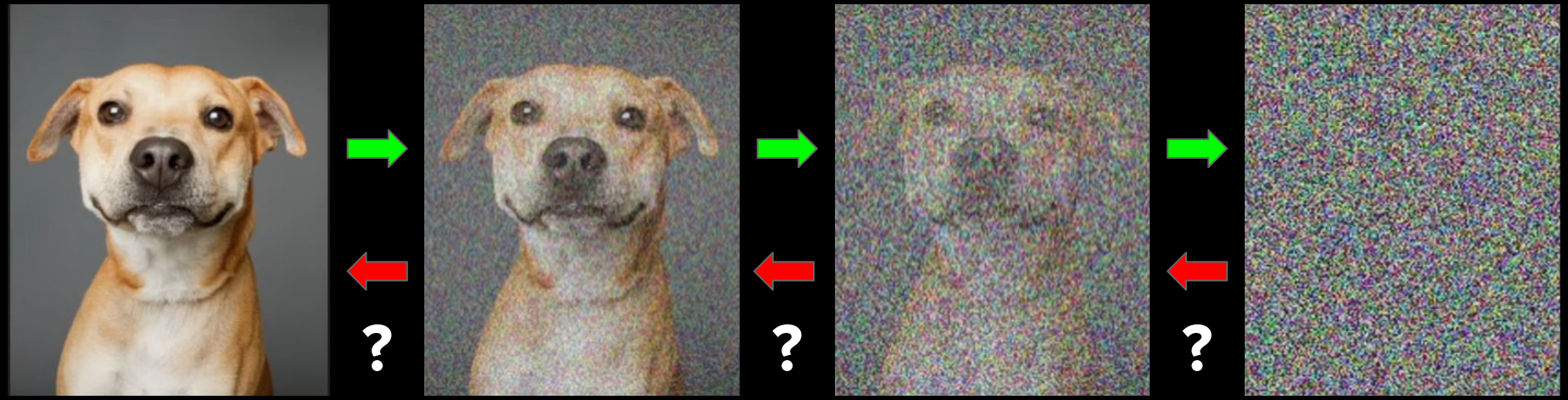
Deep Learning Computer Vision methods
— Generation: Neural style transfer —

Use the smooth latent space to combine low-level feature maps (“style”) from one image, and the high-level feature maps (pose, facial structure) of another.



Deep Learning Computer Vision methods
— Generation: Segmentation Masks (GauGAN) —

Deep Learning Computer Vision methods
— Generation: Diffusion Models —



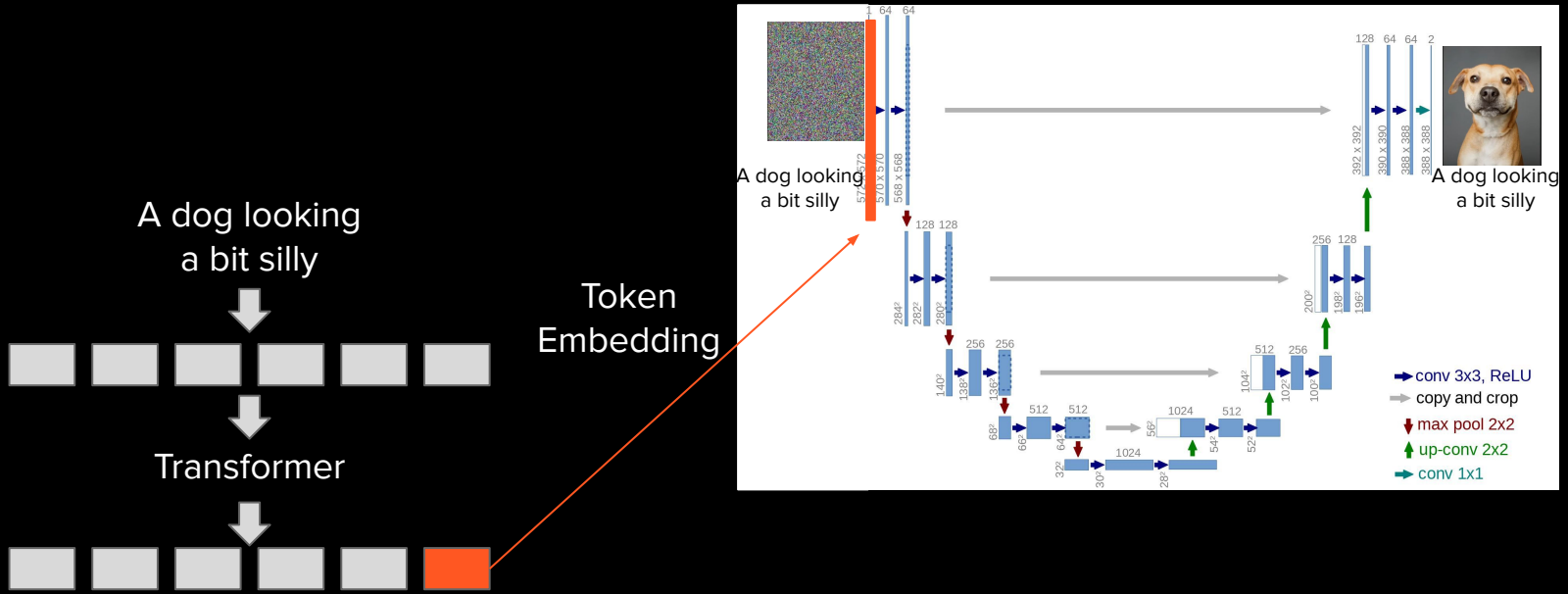
A dog looking a bit silly

A dog looking a bit silly

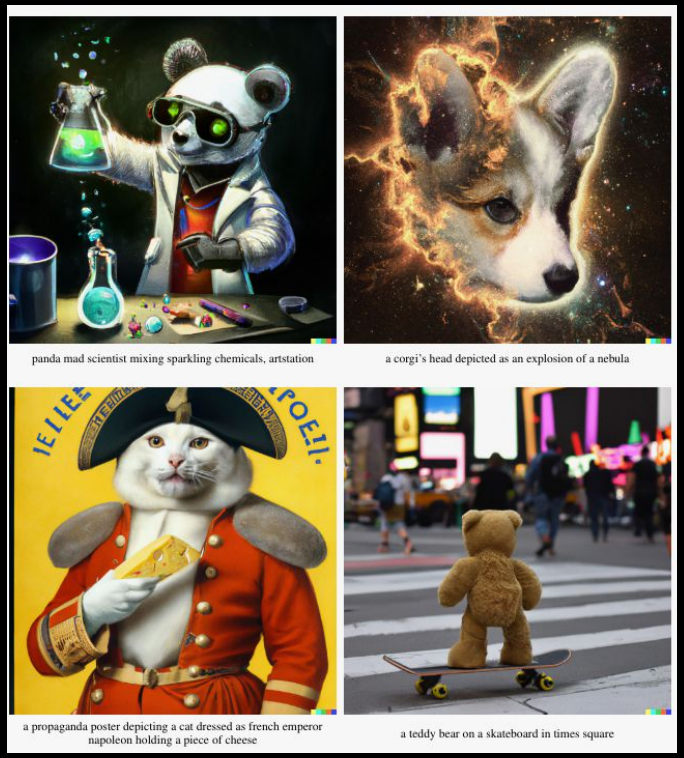
A dog looking a bit silly

A dog looking a bit silly

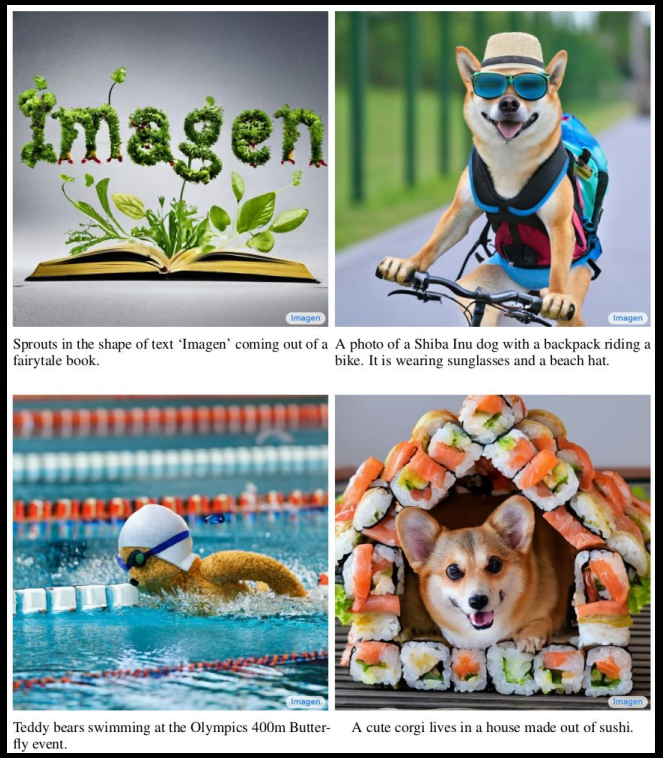
Deep Learning Computer Vision methods
— Generation: Diffusion Models —



Deep Learning Computer Vision methods
— Generation: Diffusion Models —



DALL-E 2



IMAGEN

Deep Learning Computer Vision methods
— Generation: Diffusion Models —

Diorama::2 wave carved from transparent glass, glowing inside, atmospheric, 3d render, hyperrealistic, redshift render, --q 2

Cat Knight

The last supper, cyberpunk

Jaba The Trump

MidJourney

Stable Diffusion

“Artificial intelligence consultants getting ready for a game hackathon”

Go to: https://github.com/Sam-Sweere-Vantage-AI/cv-hackathon