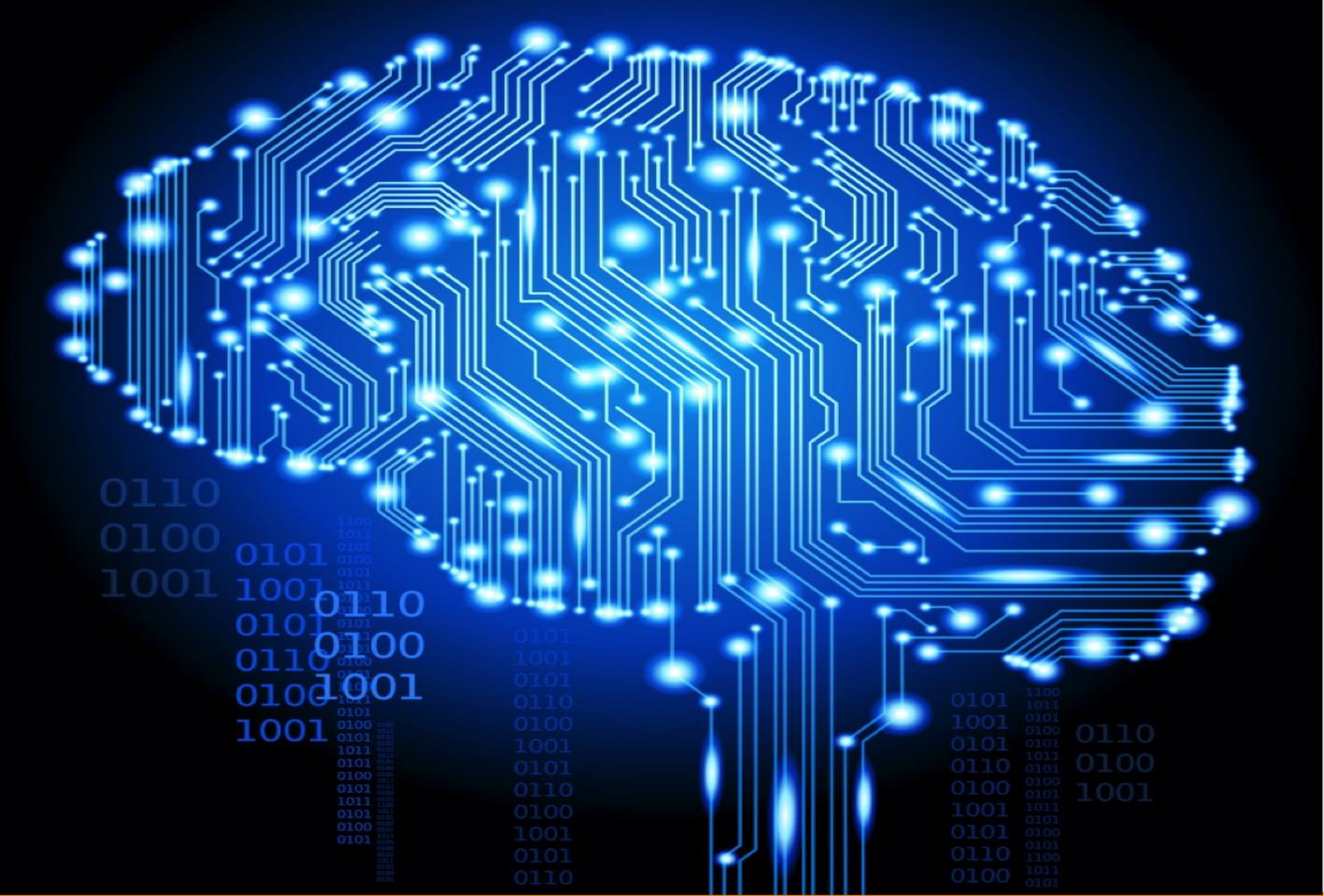


Lecture 4: Convolutional Neural Networks

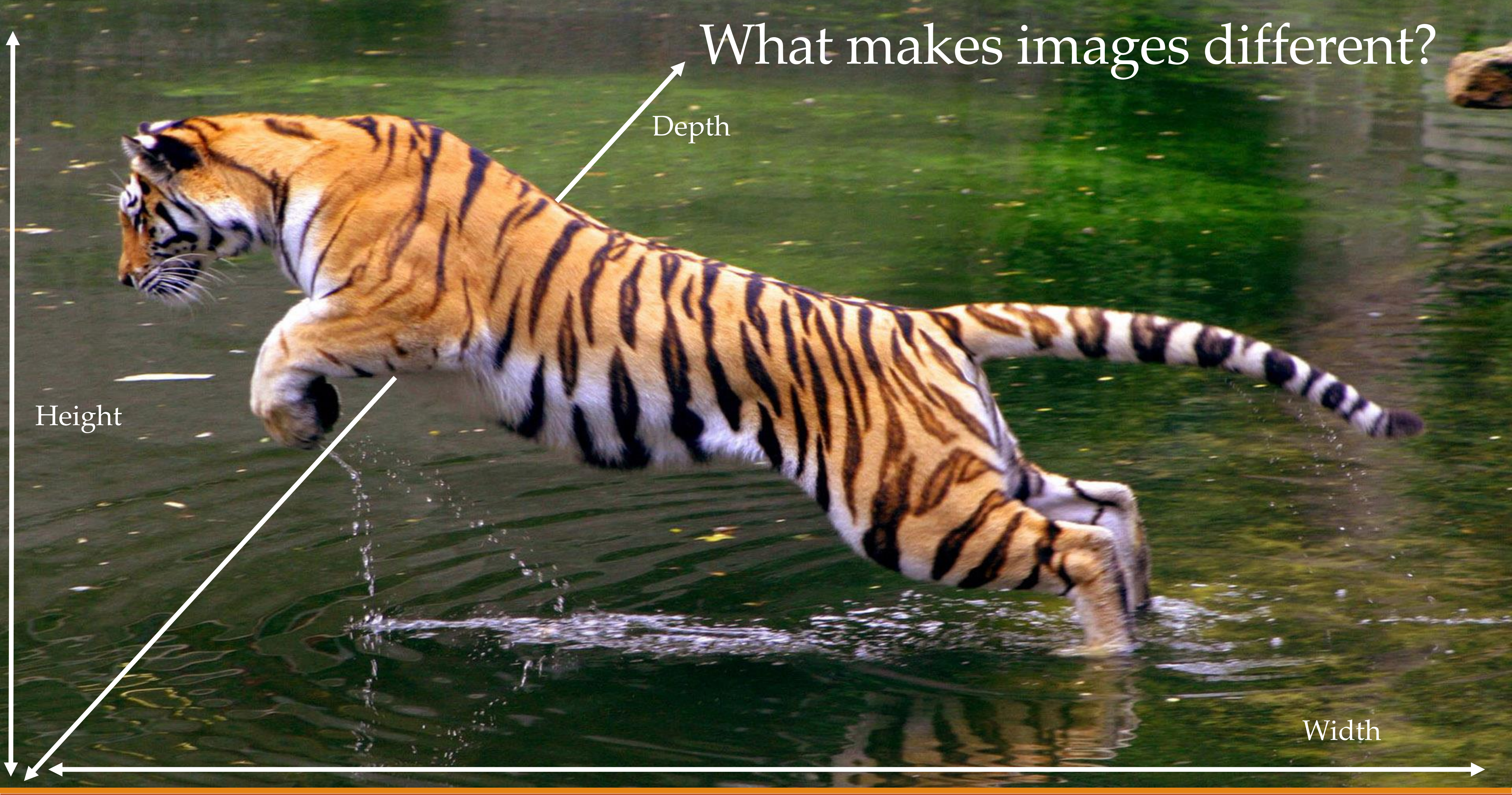
Deep Learning @ UvA

Lecture overview

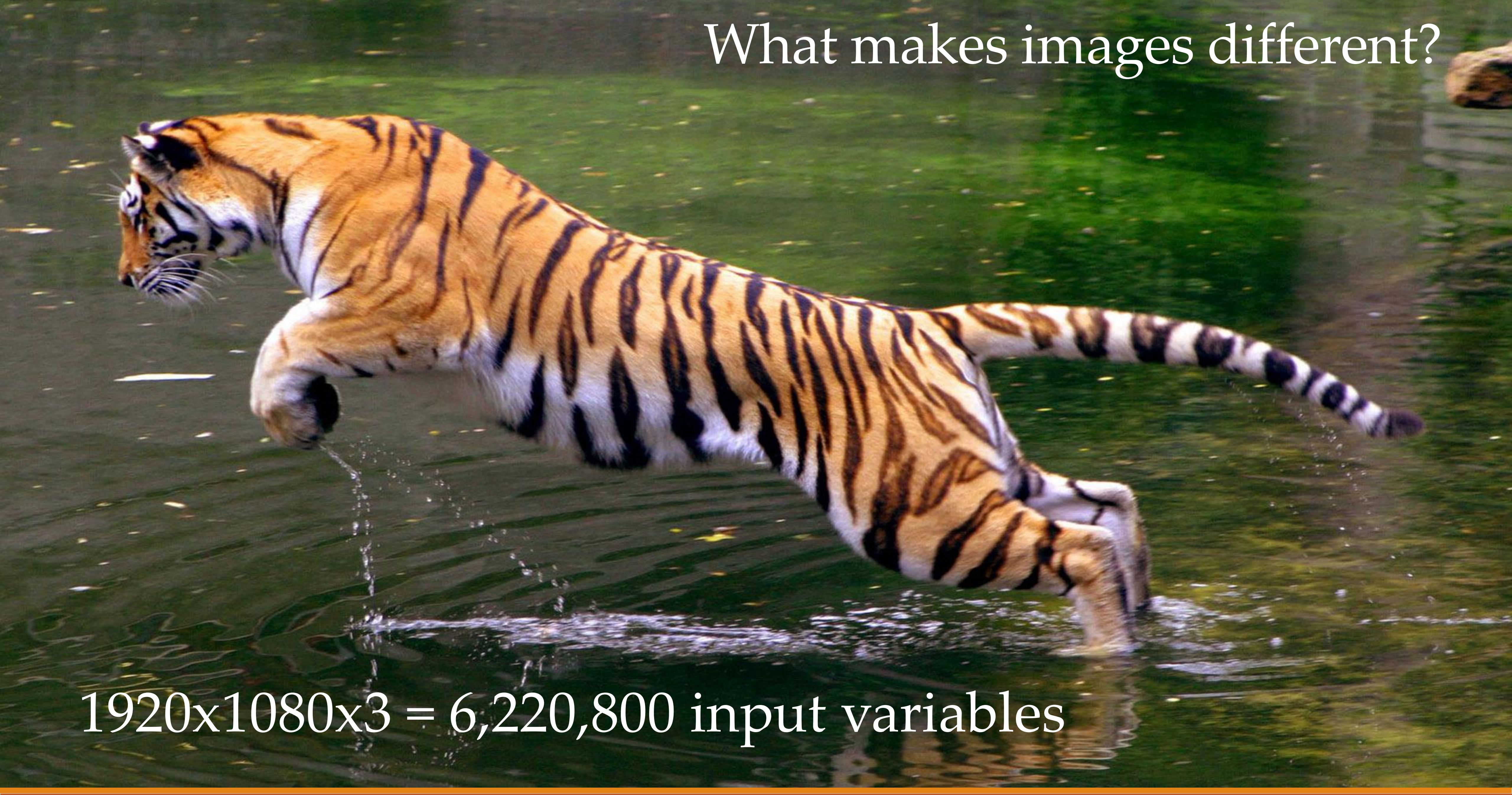
- Inductive bias: what makes images special?
- Convolution, pooling, dropout
- Study I: AlexNet
- Visualizations
- Transfer learning

What makes images different?

What makes images different?



What makes images different?



1920x1080x3 = 6,220,800 input variables

What makes images different?

What makes images different?

What makes images different?

Image has shifted a bit to the up and the left!

Input dimensions are correlated

Traditional task: Predict my salary!

Shift 1 dimension makes no sense

Level of education	Age	Years of experience	Previous job	Nationality
“Higher”	28	6	Researcher	Spain

Level of education	Age	Years of experience	Previous job	Nationality
Spain	“Higher”	28	6	Researcher

Shifting images by several dimensions (pixels) barely makes a difference

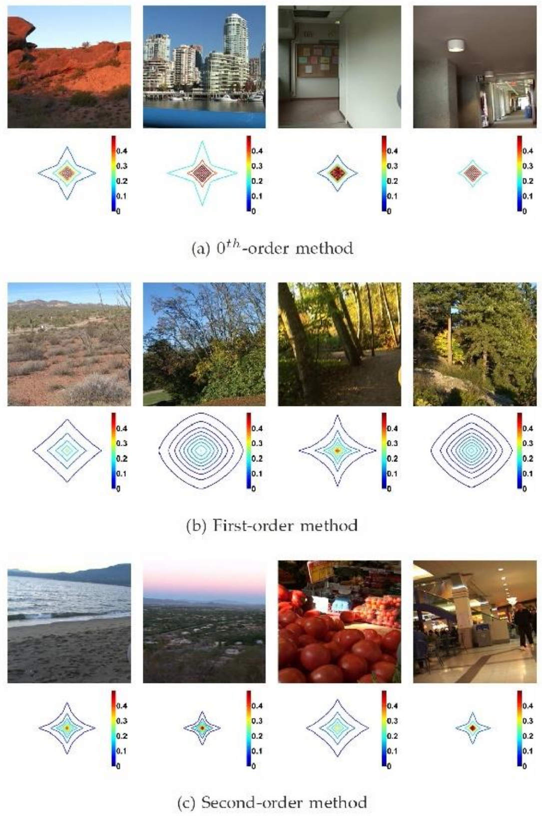
First 5x5 values

```
array([[51, 49, 51, 56, 55],
       [53, 53, 57, 61, 62],
       [67, 68, 71, 74, 75],
       [76, 77, 79, 82, 80],
       [71, 73, 76, 75, 75]], dtype=uint8)
```

First 5x5 values

```
array([[58, 57, 57, 59, 59],
       [58, 57, 57, 58, 59],
       [59, 58, 58, 58, 58],
       [61, 61, 60, 60, 59],
       [64, 63, 62, 61, 60]], dtype=uint8)
```

What makes images different?



- An image has spatial structure
- Huge dimensionality
  - 256x256 RGB image ~200M dimensions
  - 1-layered NN with 1,000 neurons → 200M parameters
- Images are stationary signals → they share features
  - Cropping/shifting/occluding dimensions → still an image
  - Possibly with same semantics
  - Basic natural image statistics are the same

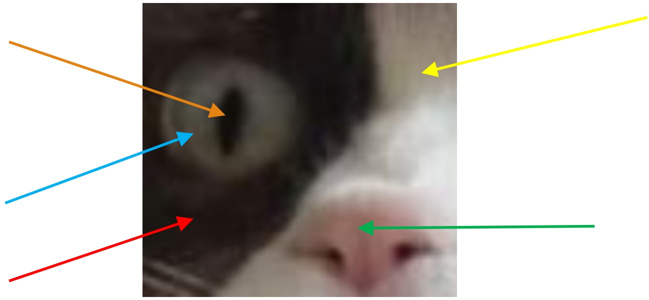
Convolutional Neural Networks

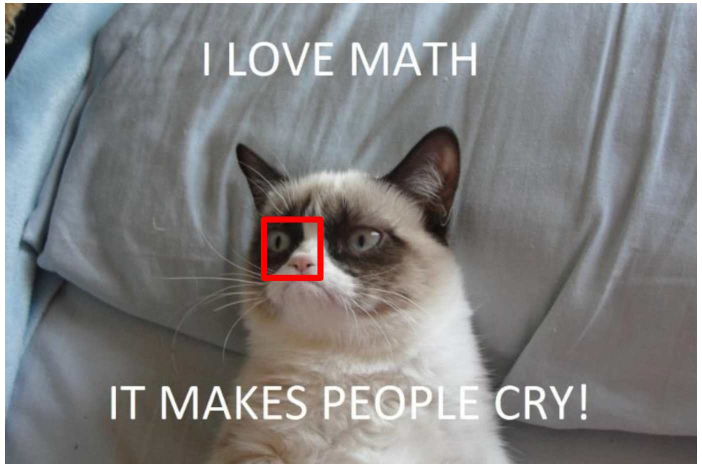
- Adding inductive bias to neural networks to deal with spatial signals
- Use convolutional filters to encode spatial structure
- Use local connectivity, parameter sharing, translation equivariance, to account for the huge input dimensionalities
- Use spatial pooling to remain robust to local variations

Why spatial?

- Images are 2-D
  - 3-D if you also count the extra channels
  - RGB, hyperspectral, etc.

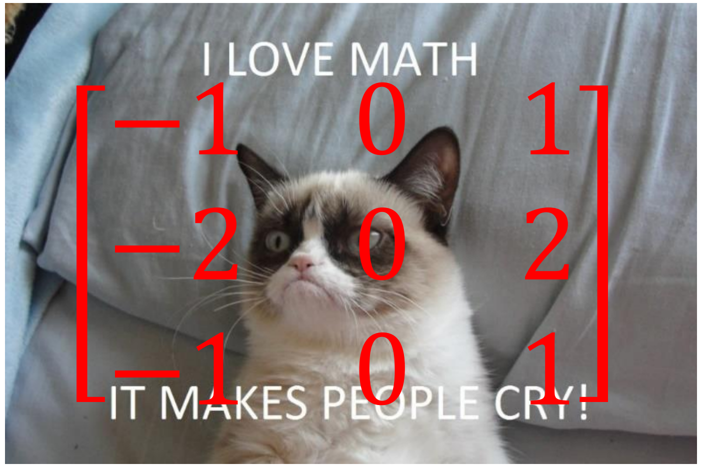
- What does a 2-D input really mean?
  - Neighboring variables are locally correlated



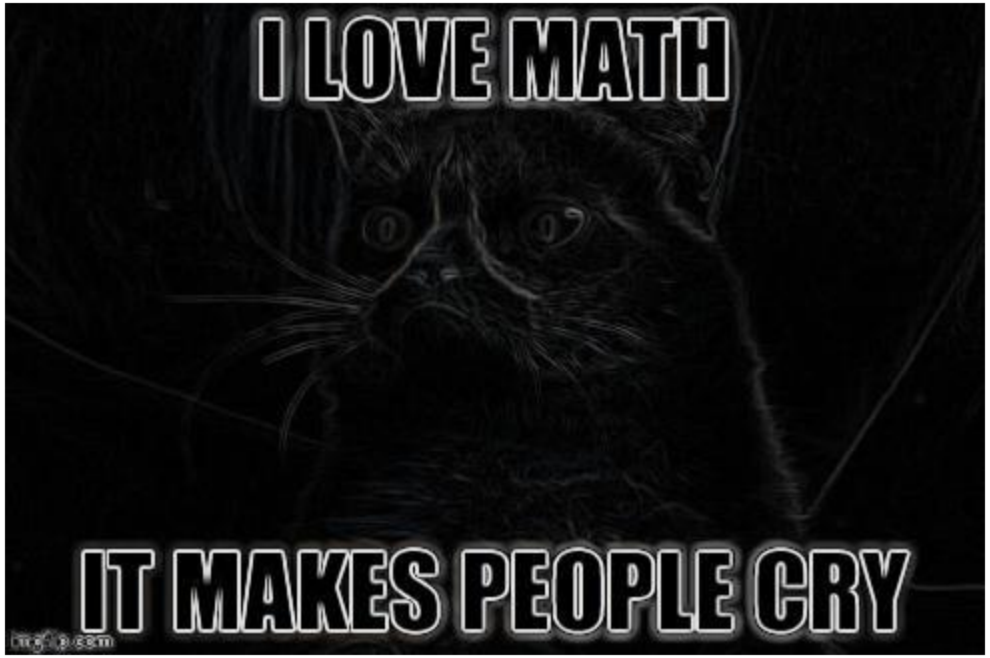


Example filter when K=1

e.g. Sobel 2-D filter



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



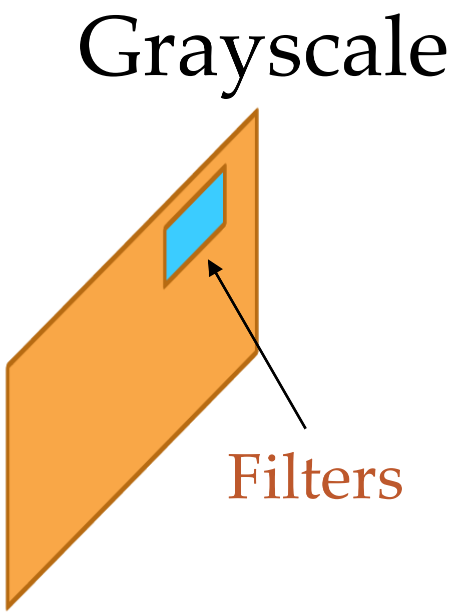
Learnable filters

- Image processing and computer vision has many handcrafted filters
  - Canny, Sobel, Gaussian blur, morphological filters, Gabor filters, etc
- Are they optimal for recognition?
- Can we learn optimal filters from our data instead?
- Are they going resemble the handcrafted filters?

$$vs. \quad \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

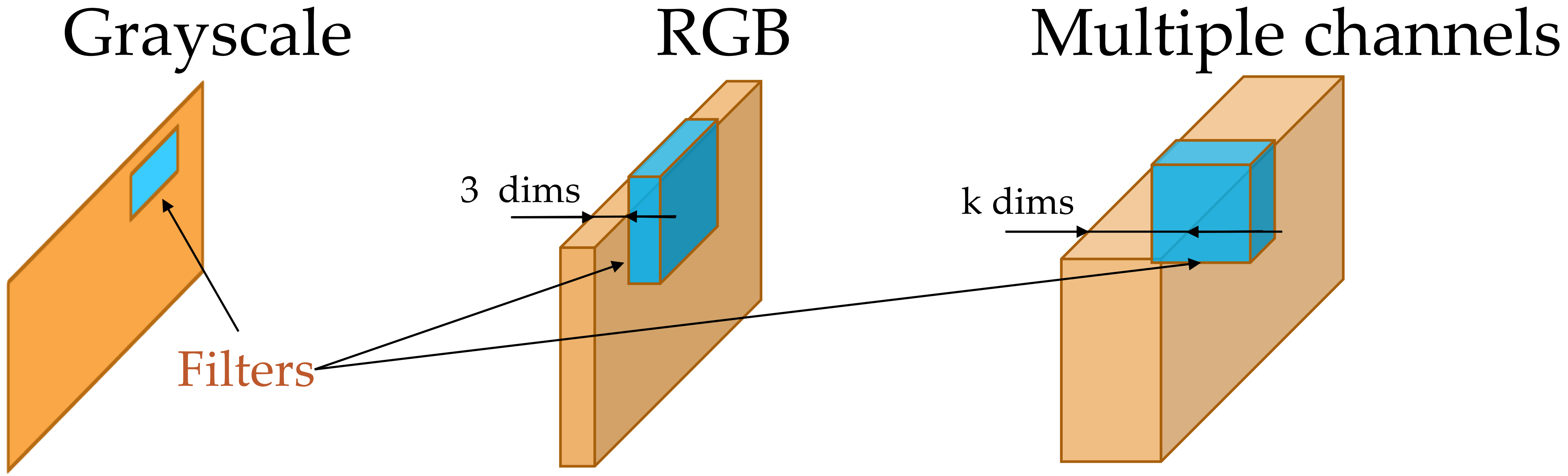
2-D Filters (Parameters)

- If images are 2-D, parameters should also be organized in 2-D
  - That way they can learn the local correlations between input variables
  - That way they can “exploit” the spatial nature of images



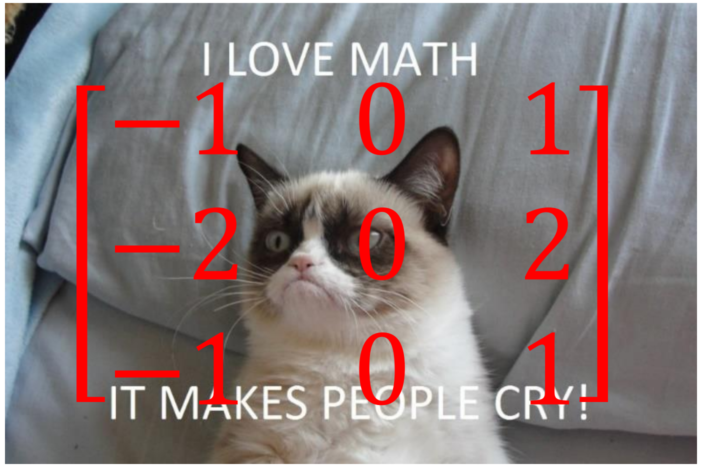
3-D Filters (Parameters)

- Similarly, if images have k channels, parameters should also have k channels

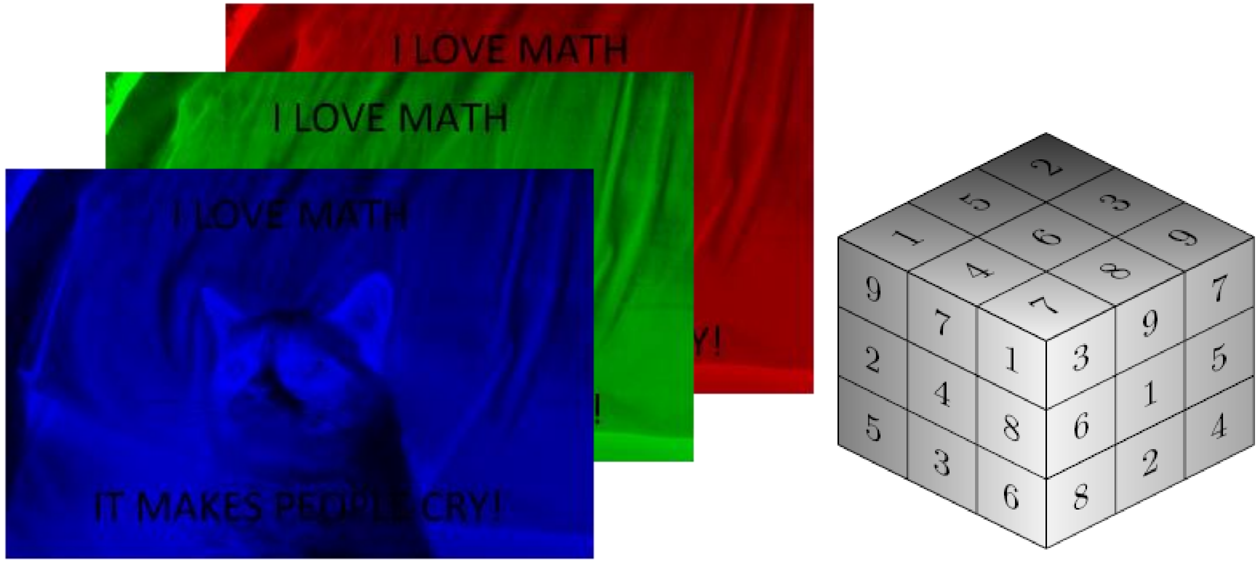


What does a 3-D filter look like?

2-D filter



3-D filter



Hypothesis

- Image statistics are not location dependent
  - Natural images are stationary
- The same filters should work on every corner of the image similarly
- Perhaps move and reuse the same (red, yellow, green) filter across the whole image?