Image Classification

Nikolai Ilinykh

LT2318 H20 Artificial Intelligence: Cognitive Systems

November 16th, 2020

Outline

- What is “image classification” and why do we need it?
- What is the data that we use to “classify images”? What does it mean to “classify images”? What is included in the image classification pipeline?
- How do we get the data for our classifiers?
  - What does it mean to “extract image features”?
  - Classic features: SIFT, HOG, HSV
  - More advanced features: CNN
- When do we use pre-trained image features?
- Image Classification Tutorial

Image classification: a very important task

Discrete categories:
{ dog, cat, plane, car, … }

cat

Image classification: semantic gap

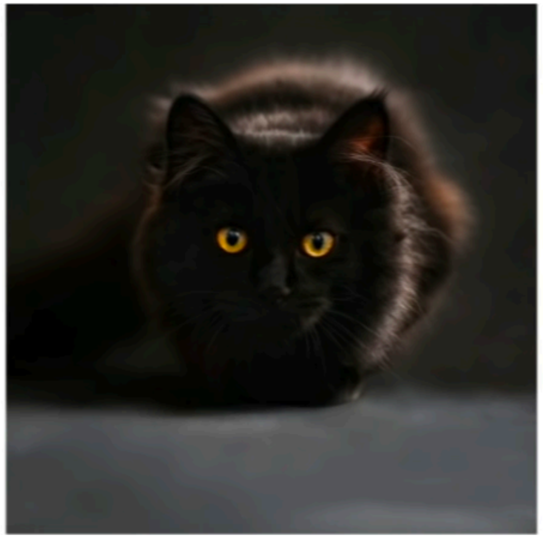
0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

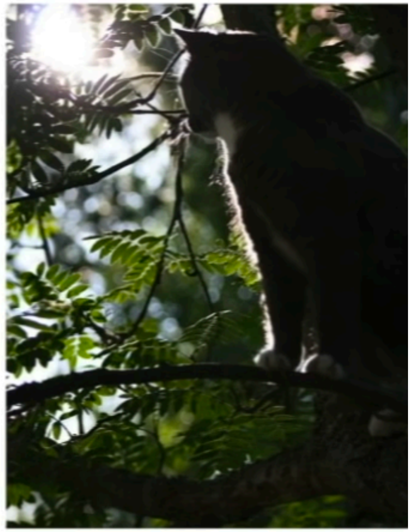
An image is a large grid of numbers between [0, 255]

An image can be of size 800 x 600 pixels, each pixel is represented via three numbers, which provide values of **RGB** (red, green, blue) channels

Challenges

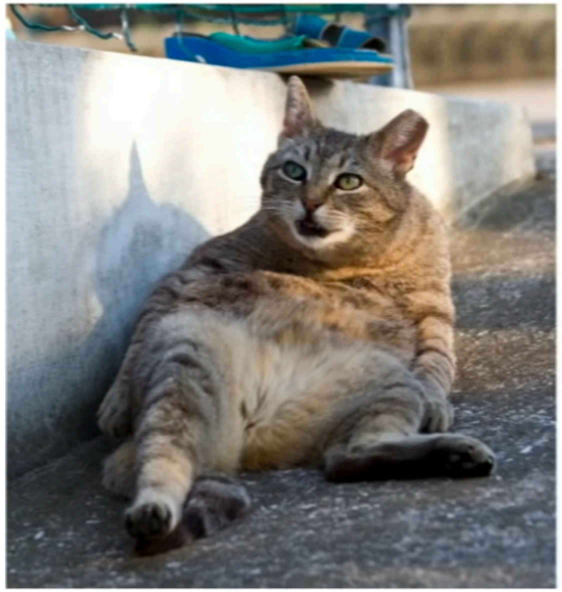
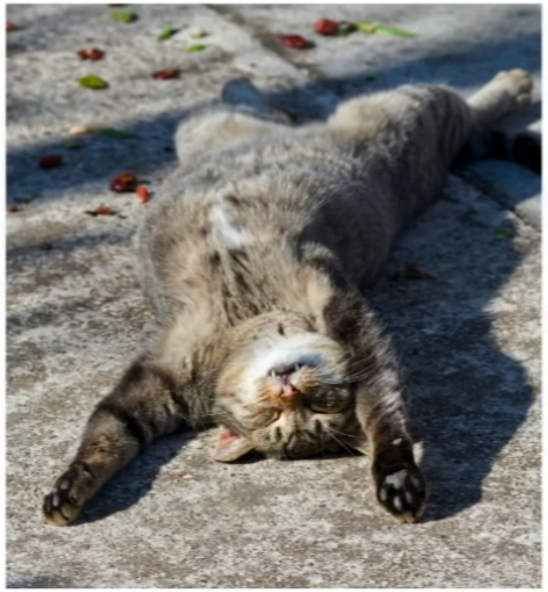
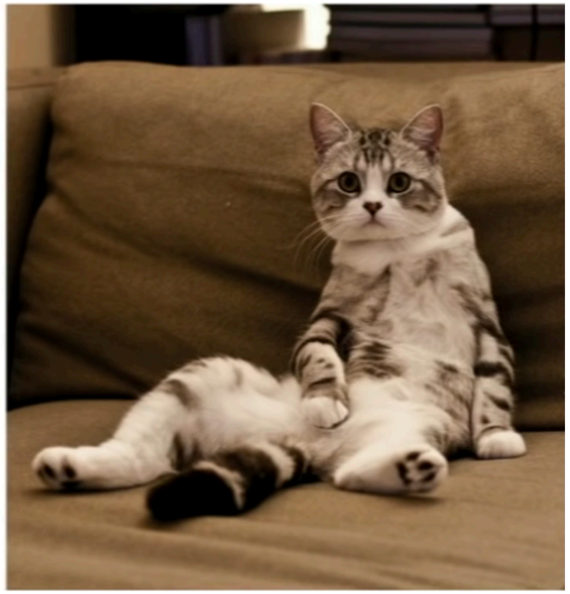
- Viewpoint
- Illumination



Challenges

- Deformation

Challenges

- Occlusion



Challenges

- Clutter

Challenges

- Intraclass variation

What is “image classification”?

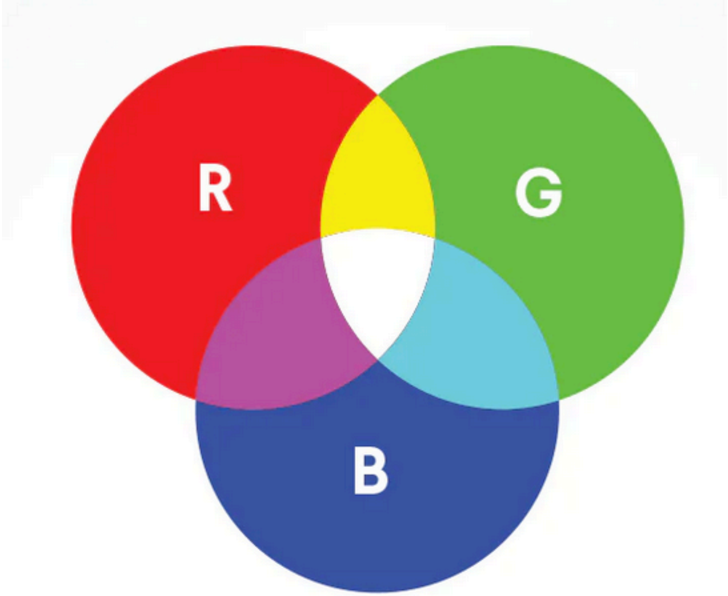
- Simply saying, **image classification** is the problem of grouping images into semantic classes based on some image features.
  - Examples of semantic classes: cats, dogs, humans, etc.
  - “grouping images” means finding specific patterns in many images that would allow the model to say that a particular image shows cats / dogs / humans, etc.
- Where is it used?
  - Robotics, Computer Vision, NLP, information retrieval, etc.
- More Challenges
  - Image quality
  - How to define a semantic label?
  - Bias in data / annotations
  - Interesting note: precision / recall trade-off

Representing images with “features”

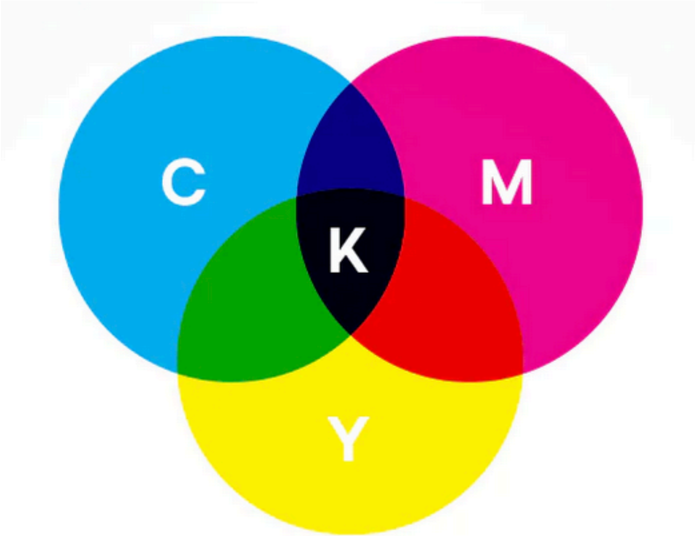
- CV vs NLP features
- Visual features
    - Color, size, center, orientation, etc.
    - Invariant to transformations
- Lexical features / semantic classes
    - Labels, context
- Learned vs. pre-engineered features
    - We need to choose how to represent an image

Visual features: color

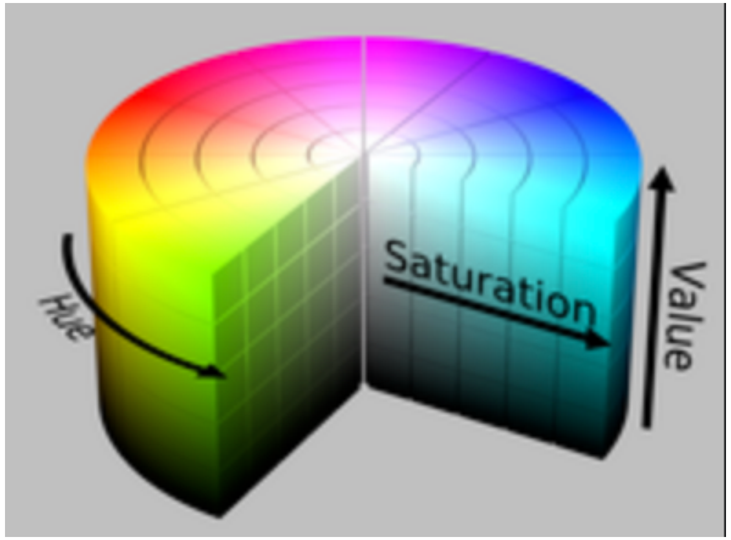
RGB



CMYK



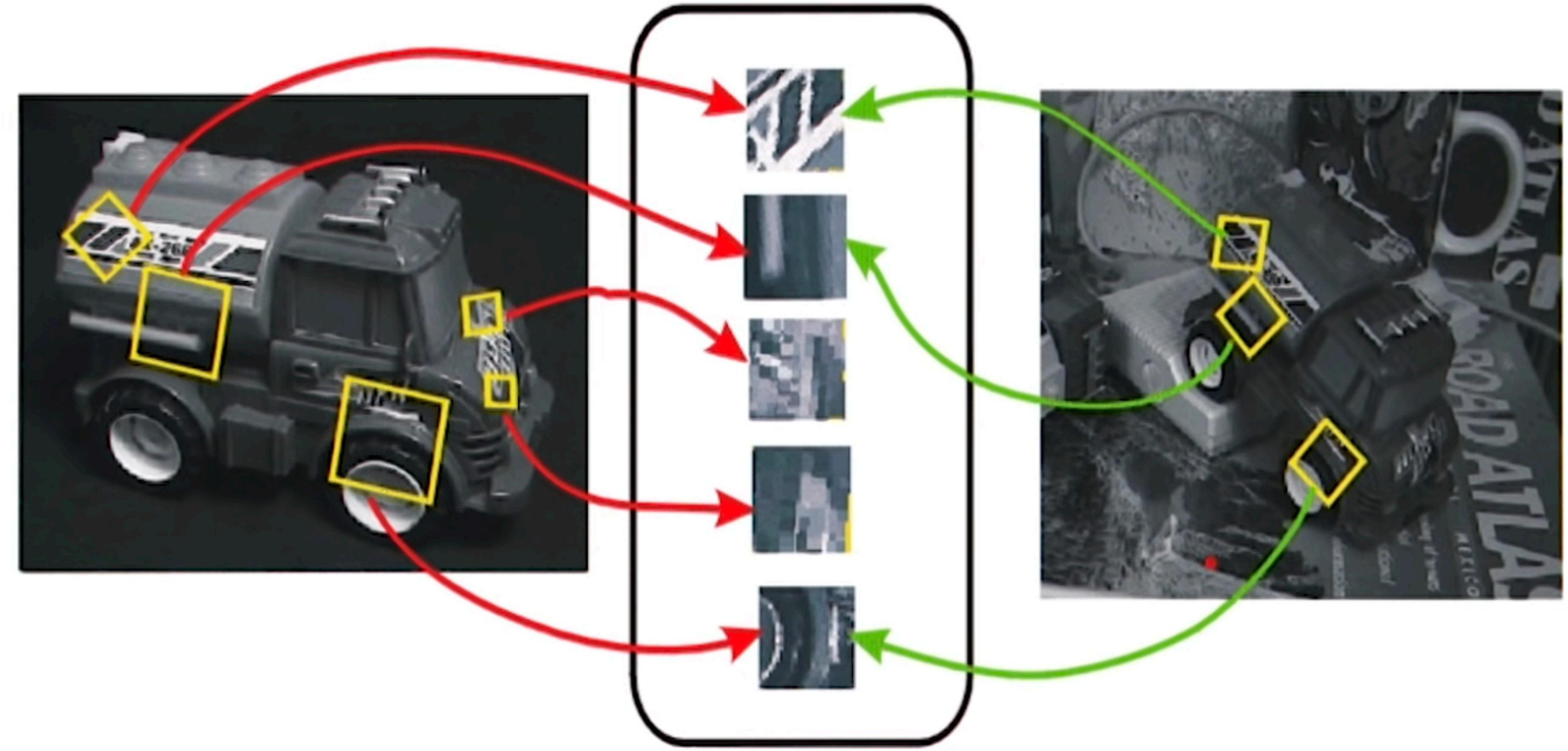
HUE



Visual features: SIFT, HOG, SURF

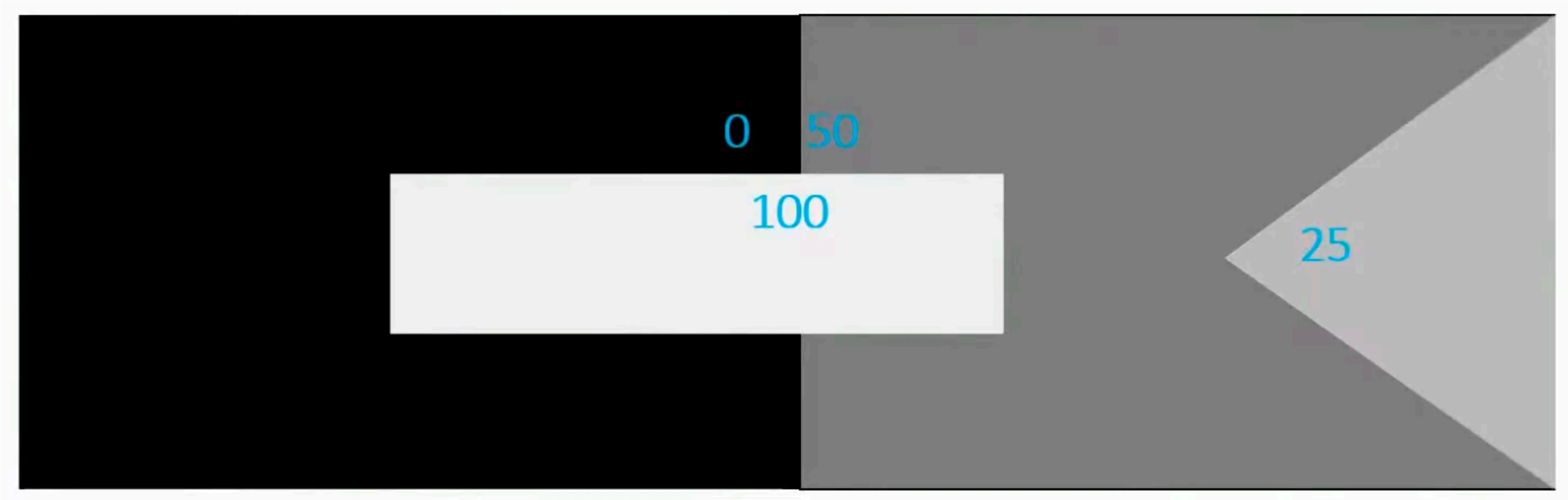
- **SIFT**: Scale-Invariant Image Transform
  - Commonly used in CV
  - Extract **invariant (not changeable)** image features
  - Applied to grayscale images
  - Mathematically complicated, computationally heavy
  - Based on histogram of gradients, e.g. computing the gradients of each pixel in the image takes a lot of time
  - Quite slow compared to SURF
  - Does not work well with lighting changes and blur

Visual features: SIFT



Visual features: HOG

- **HOG:** compute centered horizontal and vertical gradients
  - Tries to extract contrasts in various image parts
  - Computes gradients magnitudes and their directions

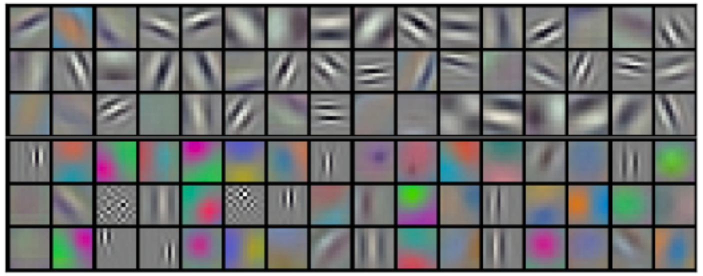


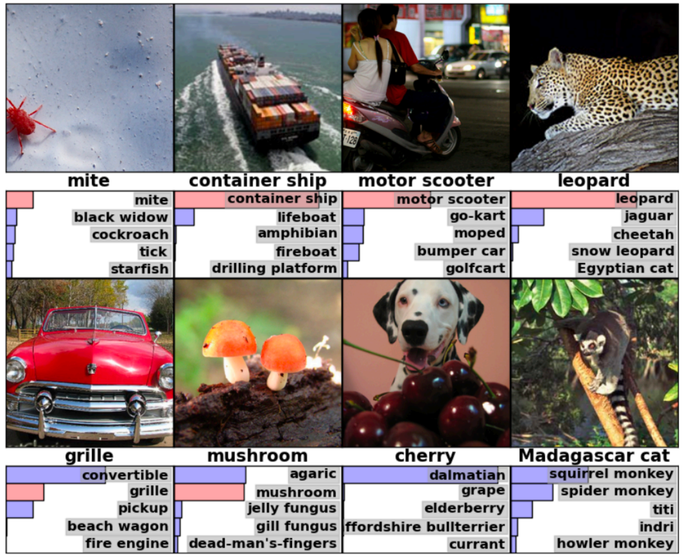
Visual features: CNN

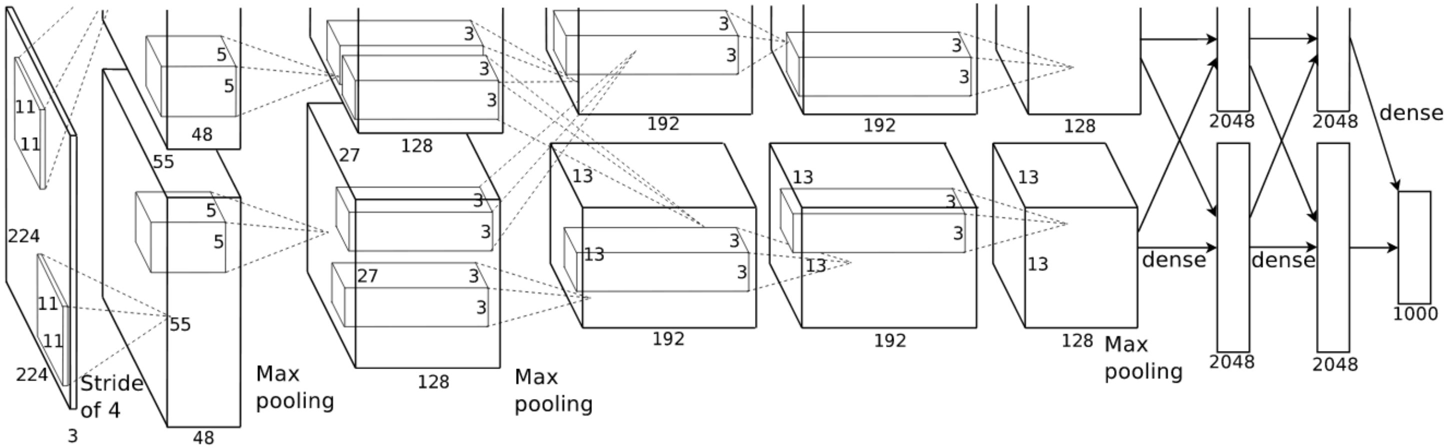
- **Convolutional Neural Networks** as feature extractors
  - SIFT and HOG features are low-level features which do not make use of hierarchical layer-wise representation learning
  - CNN is a hierarchical deep learning model which is able to model data at more and more abstract representations
  - CNN features are highly adaptive, they are trained end-to-end
  - CNN can learn features similar to SIFT and HOG from training examples alone, which is quite cool. Therefore, **using CNNs minimizes feature engineering**

CNNs

- Used for object detection, image classification, image captioning, etc.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. **ImageNet classification with deep convolutional neural networks.** In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (*NIPS'12*). Curran Associates Inc., Red Hook, NY, USA, 1097–1105.

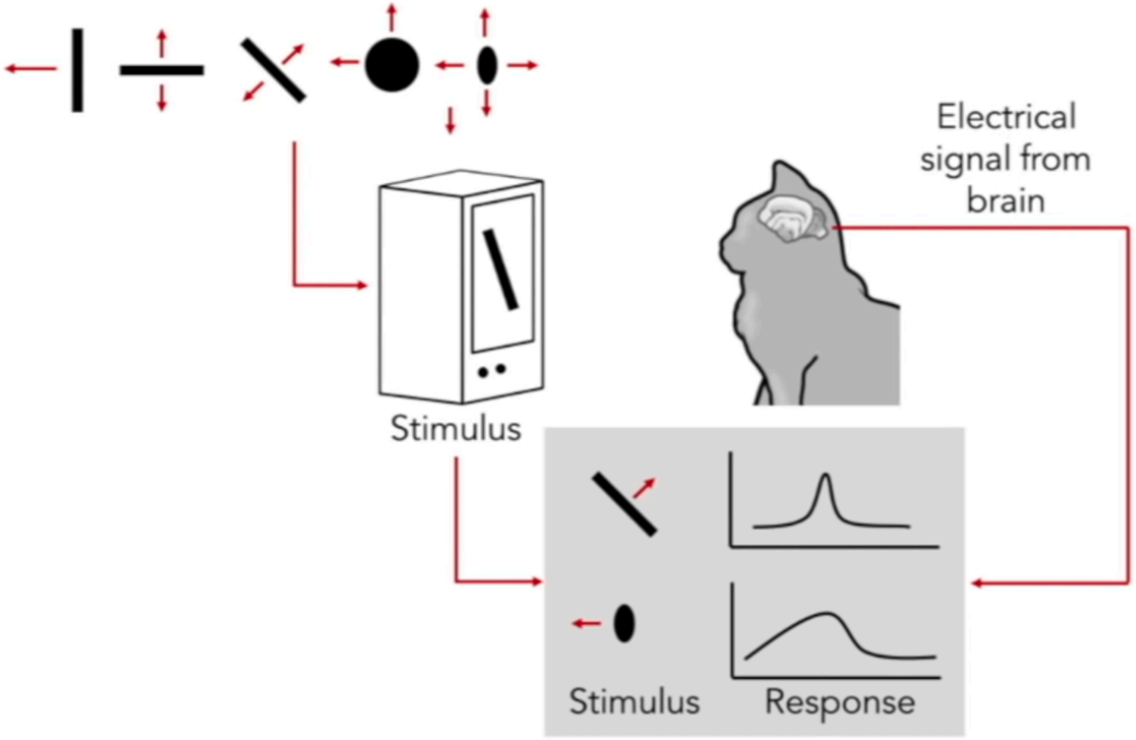


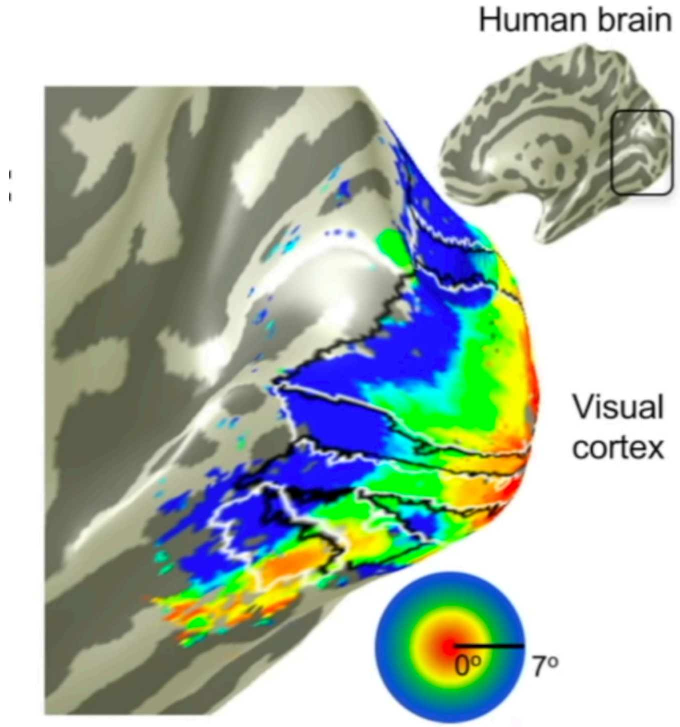




CNNs: inspiration

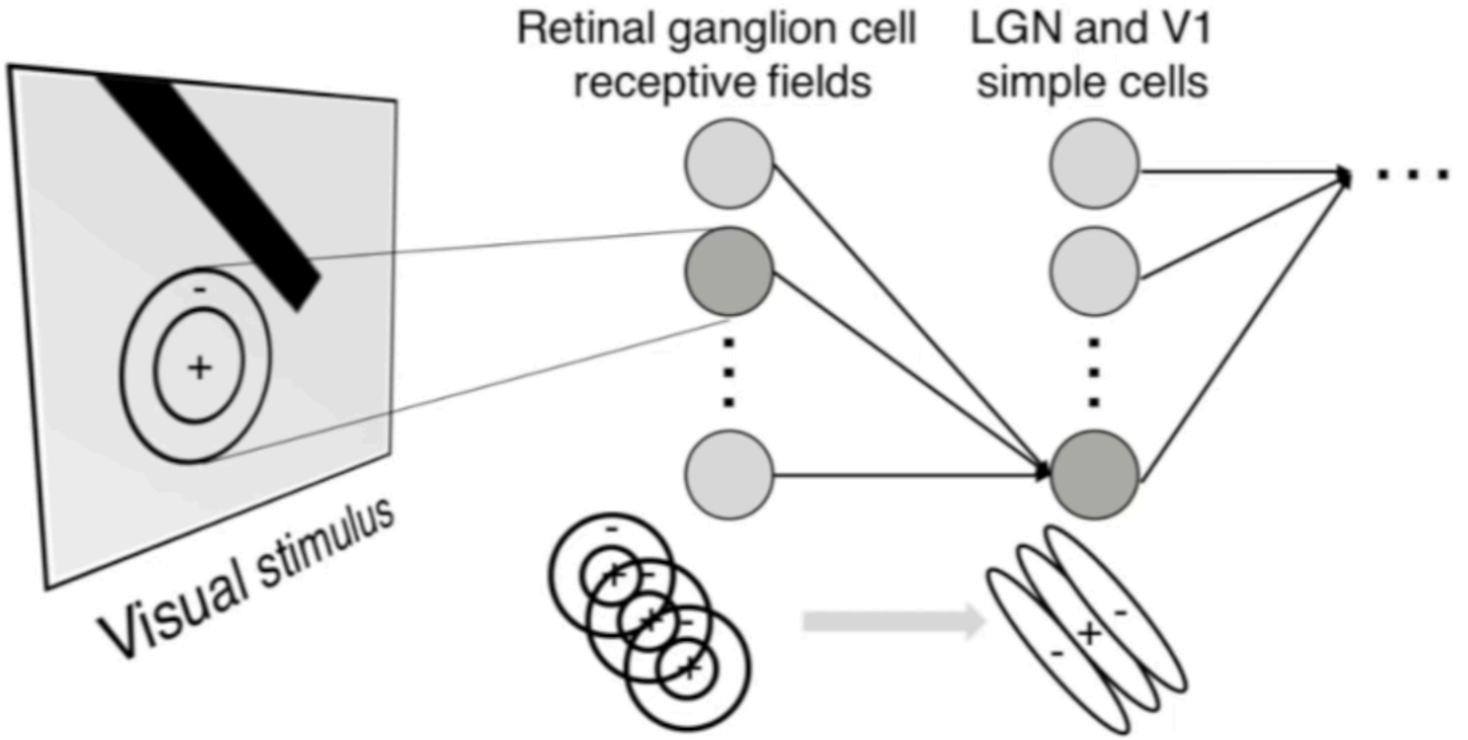
- Hubel and Wiesel (1959, 1962, 1968): cat’s visual cortex maps information in a **structured** and **hierarchical** way





CNNs: inspiration

- Hubel and Wiesel (1959, 1962, 1968): the information is processed **hierarchically, the complexity increases with the extent of how deep the processing is**



Simple cells:
Response to light orientation

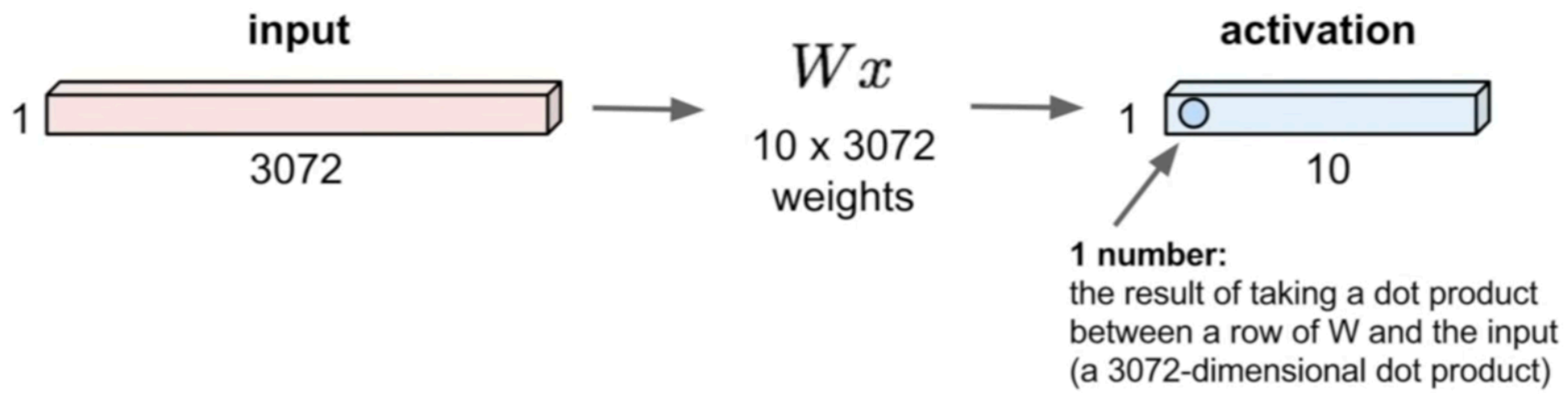
Complex cells:
Response to light orientation and movement

Hypercomplex cells:
response to movement with an end point

CNN vs FC

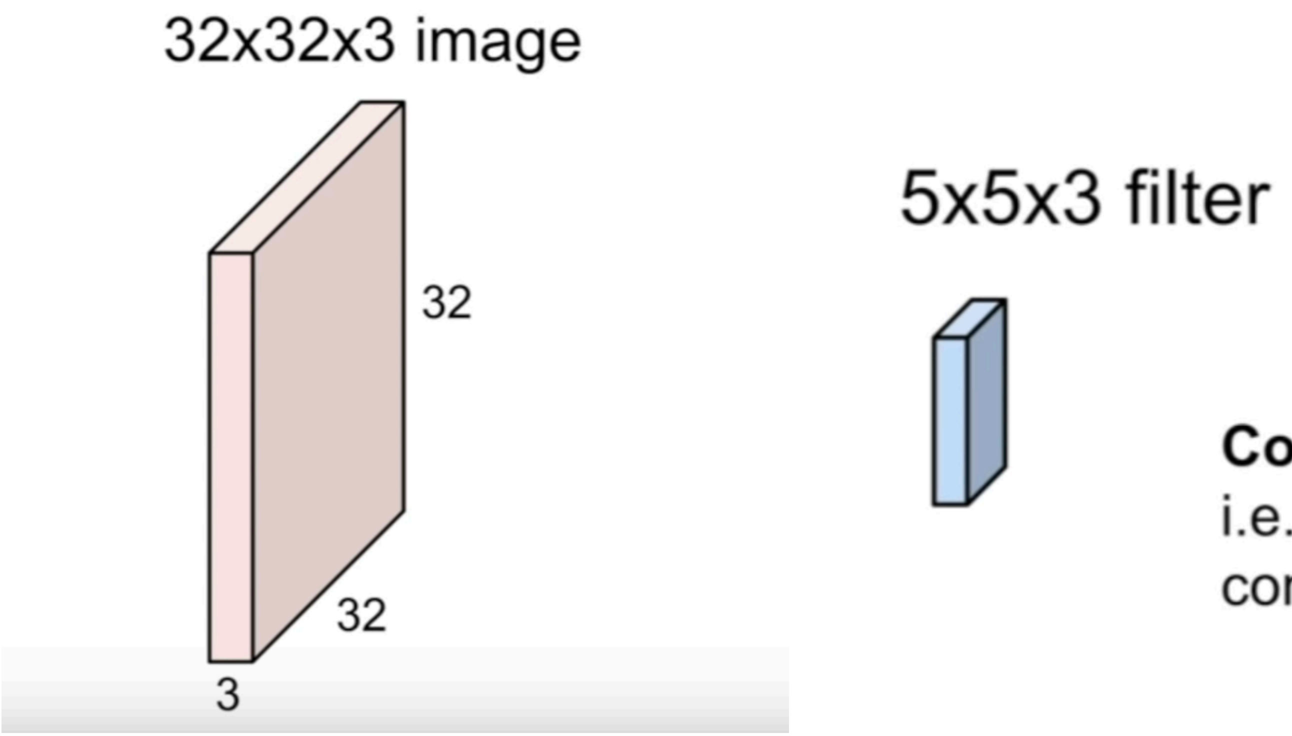
Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1



CNN: we want to preserve spatial structure

Convolution Layer



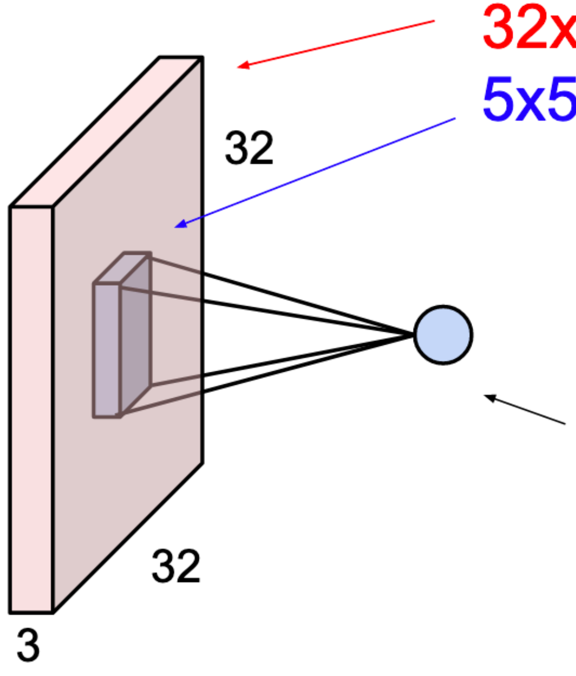
Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

CNN: apply filter to the convolution layer

Convolution Layer

32x32x3 image

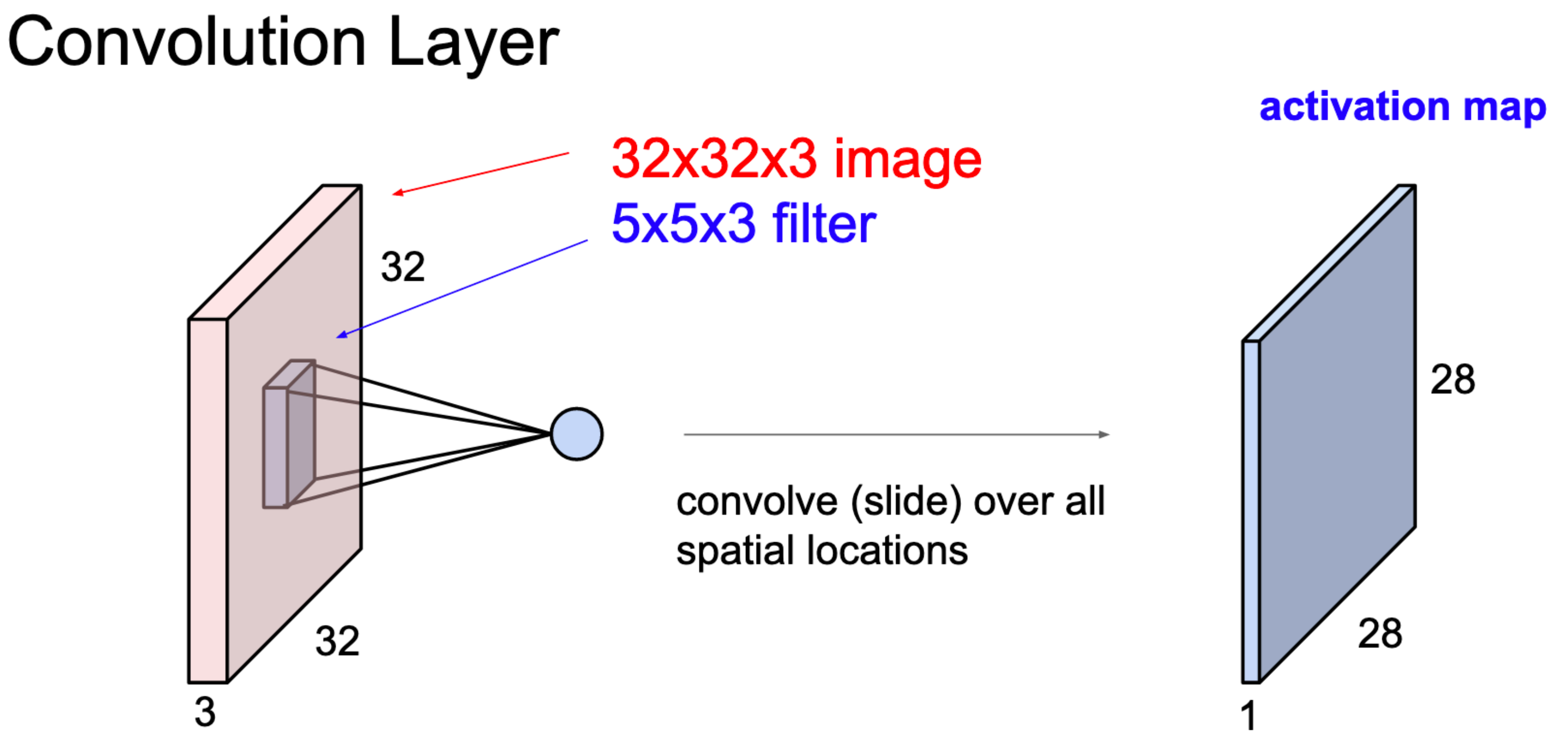
5x5x3 filter w



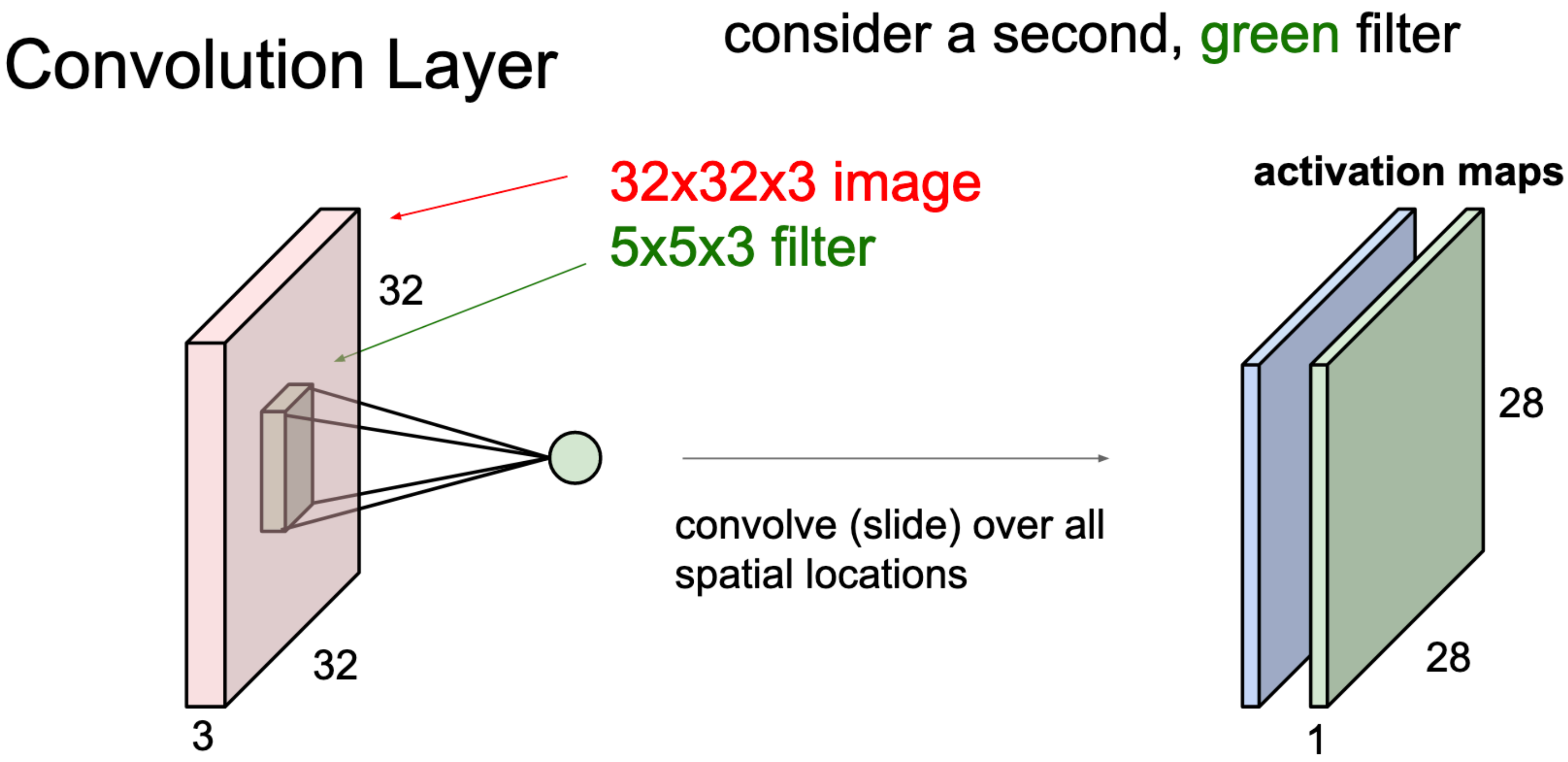
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

CNN: use filter to get an activation map

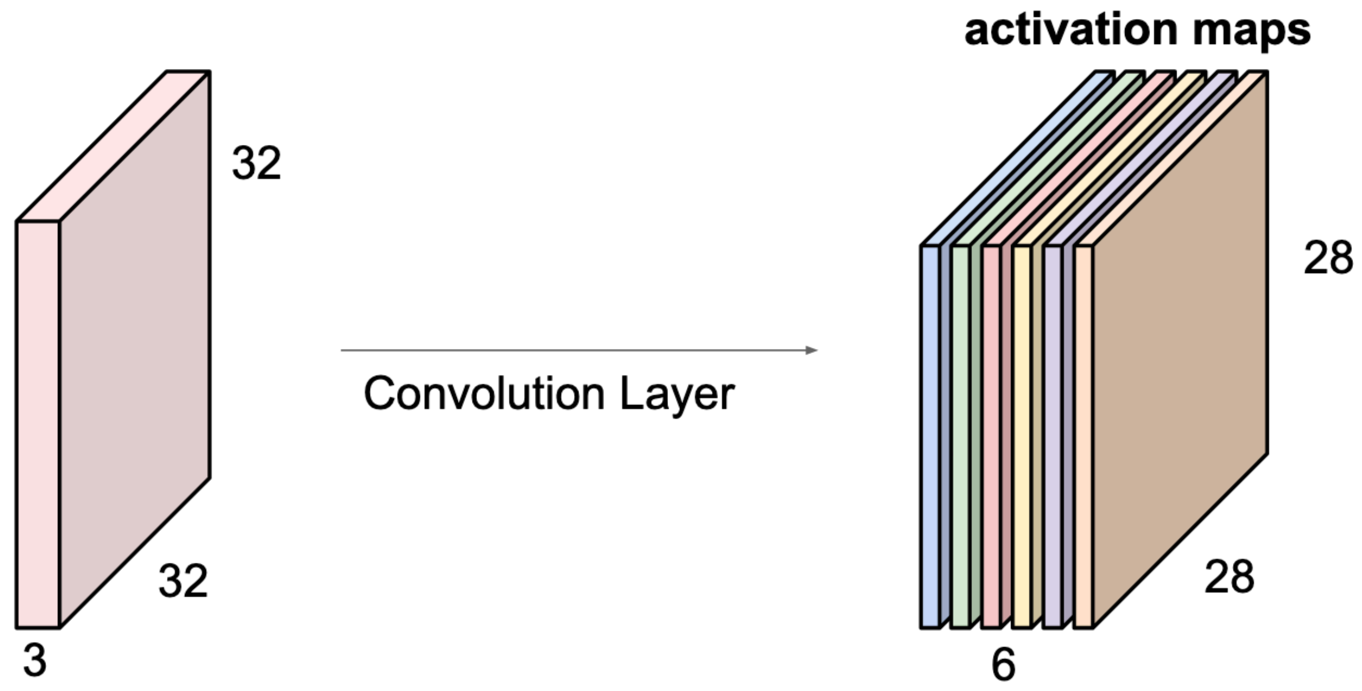


CNN: maps per filter



CNN: activation maps

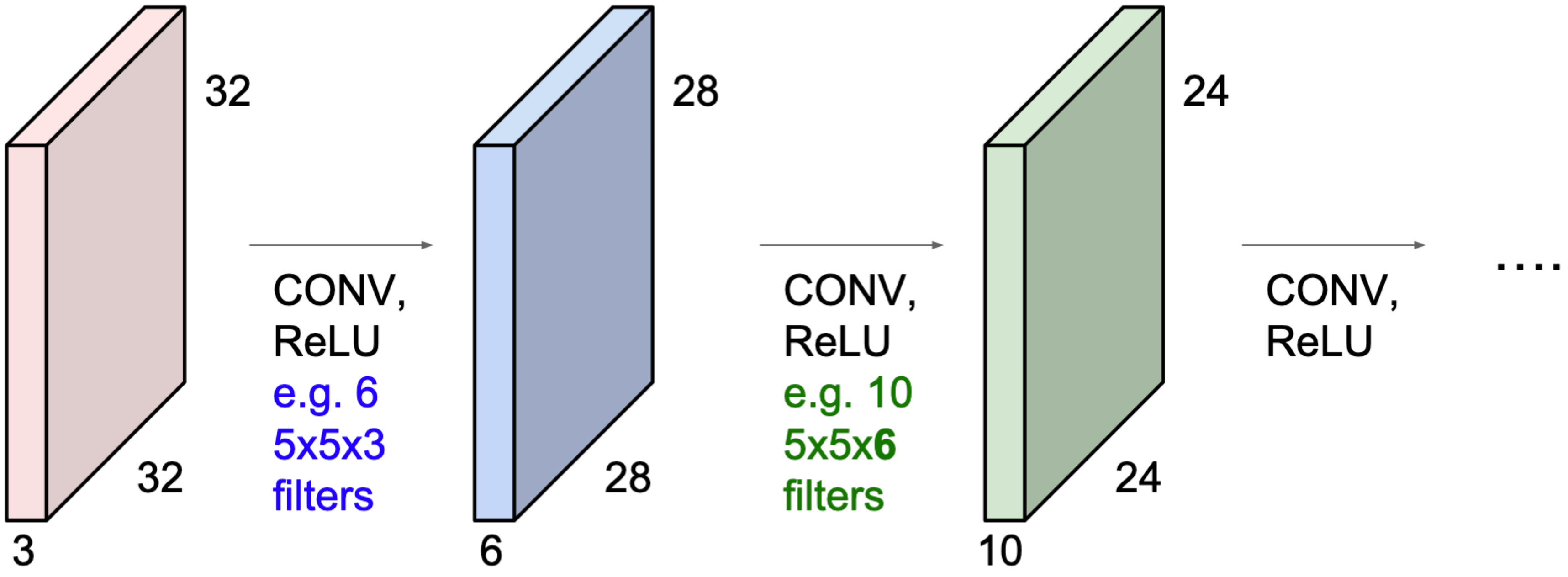
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

CNN: a stack of CONV, FC, POOL + activations

Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



CNN: conclusion

- Smaller filters, deeper architectures
- Tend to remove POOL and FC, keep CONV only

- **Example CNN network structure, VGG16**

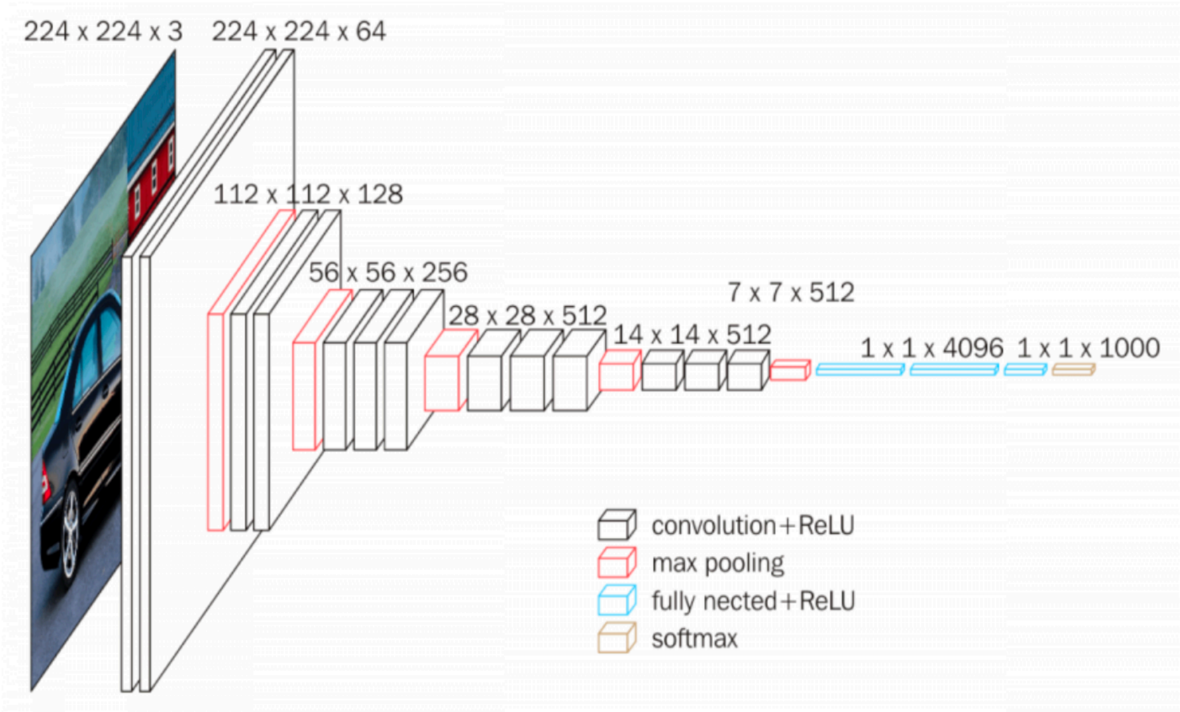


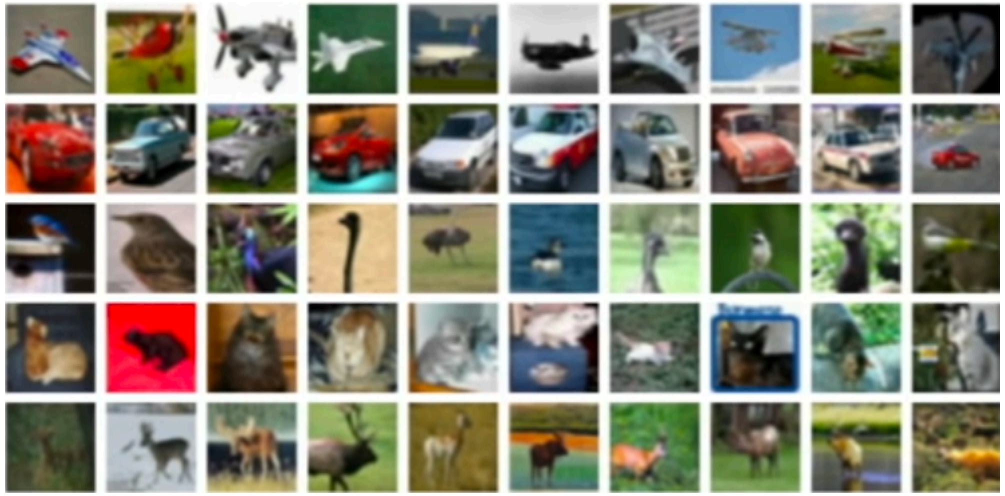
Image classification: pre-trained CNNs

- **Transfer Learning:** use CNNs trained on various datasets for different tasks
- Most models are trained on ImageNet (http://image-net.org/)
- There are many published models:
  - PyTorch pre-trained models: https://pytorch.org/docs/stable/torchvision/models.html#classification
- In our tutorial, we are going to work with pre-trained CNNs. In particular, we are going to use them to train our own classifier.

Image Classification Pipeline

- Collect a **dataset of images and labels**
- Use machine learning to train a classifier
  - we train, we validate, we test
- Evaluate the classifier on new images

- Example training set:



References

- Stanford Course: **CS231n: Convolutional Neural Networks for Visual Recognition**.
- Tutorials from previous iterations of the **LT2318 H20** course.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (*NIPS'12*). Curran Associates Inc., Red Hook, NY, USA, 1097–1105.
- S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, 2015, pp. 730-734, doi: 10.1109/ACPR.2015.7486599.

Useful Links

- PyTorch Image Classification Tutorial:
  - https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html
- TensorFlow Image Classification Tutorial:
  - https://www.tensorflow.org/tutorials/images/classification
- More recent work on CNNs:
  - https://github.com/matterport/Mask_RCNN
  - https://github.com/facebookresearch/detectron2
- Accuracy scores for published CNNs:
  - https://keras.io/api/applications/