Vision–Language Models

Jordan Boyd-Graber

University of Maryland

Introduction / Foundations

Slides from Mohit Iyyer, Vicente Ordonez, Fei-Fei Li, Justin Johnson, and Jacob Andreas

What can you do with an Image and Text?

Captioning

A silver car is in front of a white van in a mall parking lot.

What can you do with an Image and Text?

Question Answering

- What kind of car is in the foreground?
- What is written on the truck?
- What time of day was the picture taken?

What can you do with an Image and Text?

Question Answering (Requiring External Knowledge)

- Who designed the car in the foreground?
- What does the “E” stand for in the name on the truck?
- What is the name of the mall where the picture was taken?

What can you do with an Image and Text?

Question Answering (Requiring External Knowledge)

- Who designed the car in the foreground? Giorgetto Giugiaro
- What does the "E" stand for in the name on the truck?
- What is the name of the mall where the picture was taken?

What can you do with an Image and Text?

Question Answering (Requiring External Knowledge)

- Who designed the car in the foreground? Giorgetto Giugiaro
- What does the “E” stand for in the name on the truck? Emmett
- What is the name of the mall where the picture was taken?

What can you do with an Image and Text?

Question Answering (Requiring External Knowledge)

- Who designed the car in the foreground? Giorgetto Giugiaro
- What does the “E” stand for in the name on the truck? Emmett
- What is the name of the mall where the picture was taken? Twin Pines Mall

What can you do with an Image and Text?

Image Editing

Add a van full of Lybians to the image

Basics of Image Representation

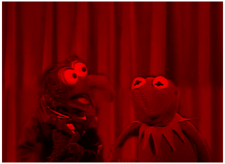
0.8	0.85	0.9	0.95	0.9	0.85	0.8	0.75	0.7	0.65
0.85	0.8	0.75	0.7	0.65	0.6	0.55	0.5	0.45	0.4
0.9	0.75	0.6	0.5	0.4	0.35	0.4	0.45	0.5	0.55
0.95	0.85	0.7	0.55	0.4	0.3	0.35	0.4	0.45	0.5
0.9	0.75	0.6	0.45	0.3	0.25	0.3	0.35	0.4	0.45
0.85	0.7	0.55	0.4	0.25	0.2	0.25	0.3	0.35	0.4
0.8	0.65	0.5	0.35	0.2	0.15	0.2	0.25	0.3	0.35
0.75	0.6	0.45	0.3	0.15	0.1	0.15	0.2	0.25	0.3
0.7	0.55	0.4	0.25	0.1	0.05	0.1	0.15	0.2	0.25
0.65	0.5	0.35	0.2	0.05	0.0	0.05	0.1	0.15	0.2

Black and white images are just a w by h matrix.

Basics of Image Representation

Color images are a 3 by w by h tensor.

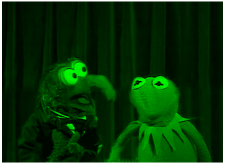
Basics of Image Representation



0.95	0.90	0.85	0.80	0.75	0.75	0.80	0.85	0.90	0.95
0.90	0.85	0.80	0.75	0.70	0.70	0.75	0.80	0.85	0.90
0.85	0.80	0.75	0.70	0.65	0.65	0.70	0.75	0.80	0.85
0.80	0.75	0.70	0.65	0.60	0.60	0.65	0.70	0.75	0.80
0.75	0.70	0.65	0.60	0.55	0.55	0.60	0.65	0.70	0.75
0.40	0.35	0.30	0.25	0.20	0.20	0.25	0.30	0.35	0.40
0.35	0.30	0.25	0.20	0.15	0.15	0.20	0.25	0.30	0.35
0.30	0.25	0.20	0.15	0.10	0.10	0.15	0.20	0.25	0.30
0.25	0.20	0.15	0.10	0.05	0.05	0.10	0.15	0.20	0.25
0.20	0.15	0.10	0.05	0.00	0.00	0.05	0.10	0.15	0.20

Color images are a 3 by w by h tensor.

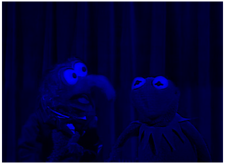
Basics of Image Representation



0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18	0.20
0.07	0.09	0.11	0.13	0.15	0.17	0.19	0.21	0.23	0.25
0.12	0.14	0.16	0.18	0.20	0.22	0.24	0.26	0.28	0.30
0.17	0.19	0.21	0.23	0.25	0.27	0.29	0.31	0.33	0.35
0.22	0.24	0.26	0.28	0.30	0.32	0.34	0.36	0.38	0.40
0.27	0.29	0.31	0.33	0.35	0.37	0.39	0.41	0.43	0.45
0.32	0.34	0.36	0.38	0.40	0.42	0.44	0.46	0.48	0.50
0.37	0.39	0.41	0.43	0.45	0.47	0.49	0.51	0.53	0.55
0.42	0.44	0.46	0.48	0.50	0.52	0.54	0.56	0.58	0.60
0.47	0.49	0.51	0.53	0.55	0.57	0.59	0.61	0.63	0.65

Color images are a 3 by w by h tensor.

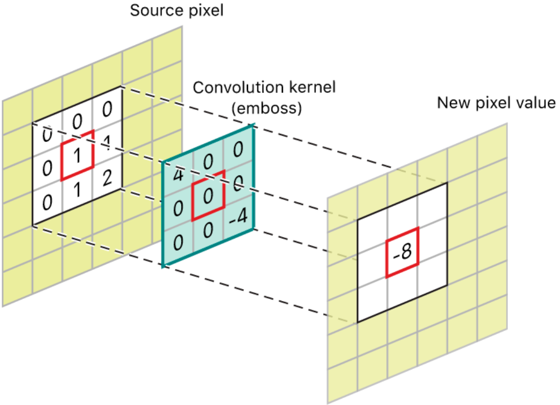
Basics of Image Representation



0.20	0.18	0.16	0.14	0.12	0.10	0.08	0.06	0.04	0.02
0.25	0.23	0.21	0.19	0.17	0.15	0.13	0.11	0.09	0.07
0.30	0.28	0.26	0.24	0.22	0.20	0.18	0.16	0.14	0.12
0.35	0.33	0.31	0.29	0.27	0.25	0.23	0.21	0.19	0.17
0.40	0.38	0.36	0.34	0.32	0.30	0.28	0.26	0.24	0.22
0.45	0.43	0.41	0.39	0.37	0.35	0.33	0.31	0.29	0.27
0.50	0.48	0.46	0.44	0.42	0.40	0.38	0.36	0.34	0.32
0.55	0.53	0.51	0.49	0.47	0.45	0.43	0.41	0.39	0.37
0.60	0.58	0.56	0.54	0.52	0.50	0.48	0.46	0.44	0.42
0.65	0.63	0.61	0.59	0.57	0.55	0.53	0.51	0.49	0.47

Color images are a 3 by w by h tensor.

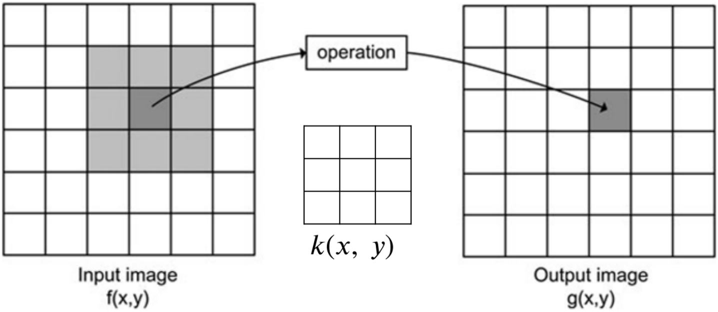
Turning Pixels into Meaning: Kernel



$$g(x,y) = \sum_u \sum_y k(u,v) f(x-u, y-v) \quad (1)$$

(Image from Apple)

Turning Pixels into Meaning: Kernel



$$g(x,y) = \sum_{u}\sum_{y} k(u,v) f(x-u, y-v) \tag{1}$$

(Image from Apple)

Kernel Examples

blur

$$\begin{bmatrix} 0.0625 & 0.1250 & 0.0625 \\ 0.1250 & 0.2500 & 0.1250 \\ 0.0625 & 0.1250 & 0.0625 \end{bmatrix}$$

De-emphasizes differences in adjacent pixel values.

Description and outputs from
`https://setosa.io/ev/image-kernels/`

Kernel Examples

sharpen

$$\begin{bmatrix} 0.0000 & -1.0000 & 0.0000 \\ -1.0000 & 5.0000 & -1.0000 \\ 0.0000 & -1.0000 & 0.0000 \end{bmatrix}$$

Emphasizes differences in adjacent pixel values. This makes the image look more vivid.

Description and outputs from
`https://setosa.io/ev/image-kernels/`

Kernel Examples

emboss

$$\begin{bmatrix} -2.0000 & -1.0000 & 0.0000 \\ -1.0000 & 1.0000 & 1.0000 \\ 0.0000 & 1.0000 & 2.0000 \end{bmatrix}$$

Gives the illusion of depth by emphasizing the differences of pixels in a given direction (from the top left to the bottom right).

Description and outputs from
`https://setosa.io/ev/image-kernels/`

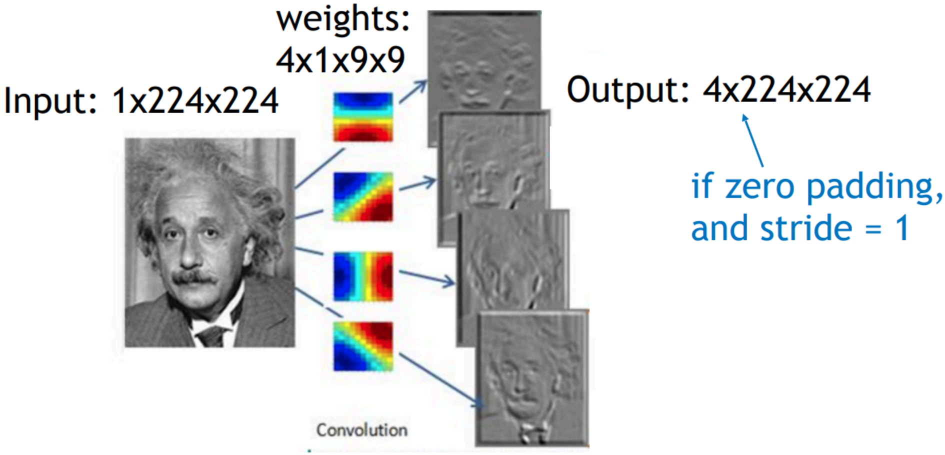
Kernel Examples

outline

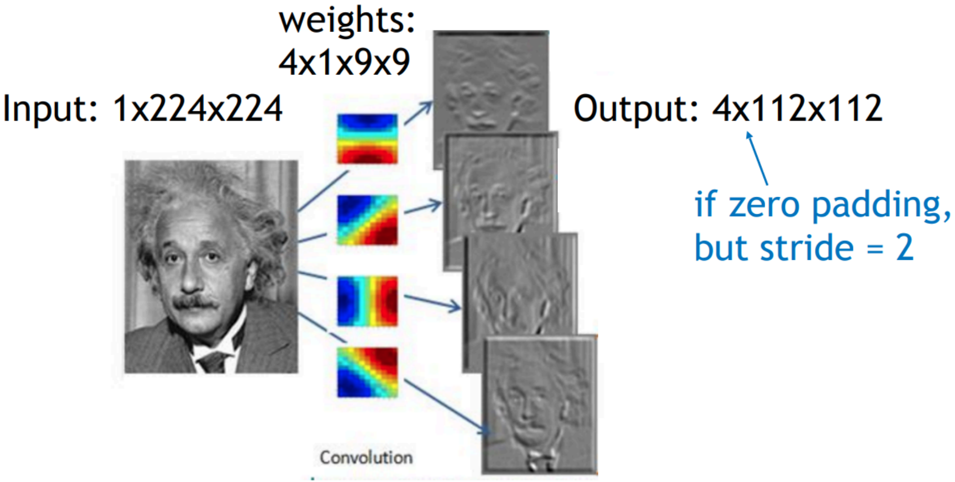
$$\begin{bmatrix} -1.0000 & -1.0000 & -1.0000 \\ -1.0000 & 8.0000 & -1.0000 \\ -1.0000 & -1.0000 & -1.0000 \end{bmatrix}$$

Highlights large differences. A pixel with similar neighbors appears black; one with different neighbors is white.

Description and outputs from
`https://setosa.io/ev/image-kernels/`



You can also learn kernels as part of network



Don’t have to compute kernel on every pixel

Alexnet

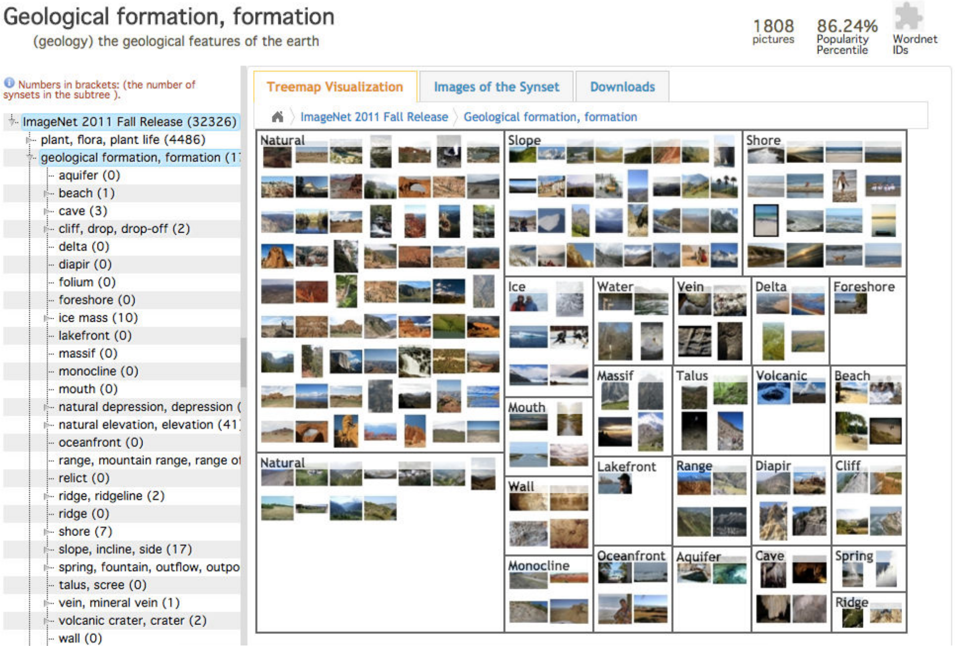
ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

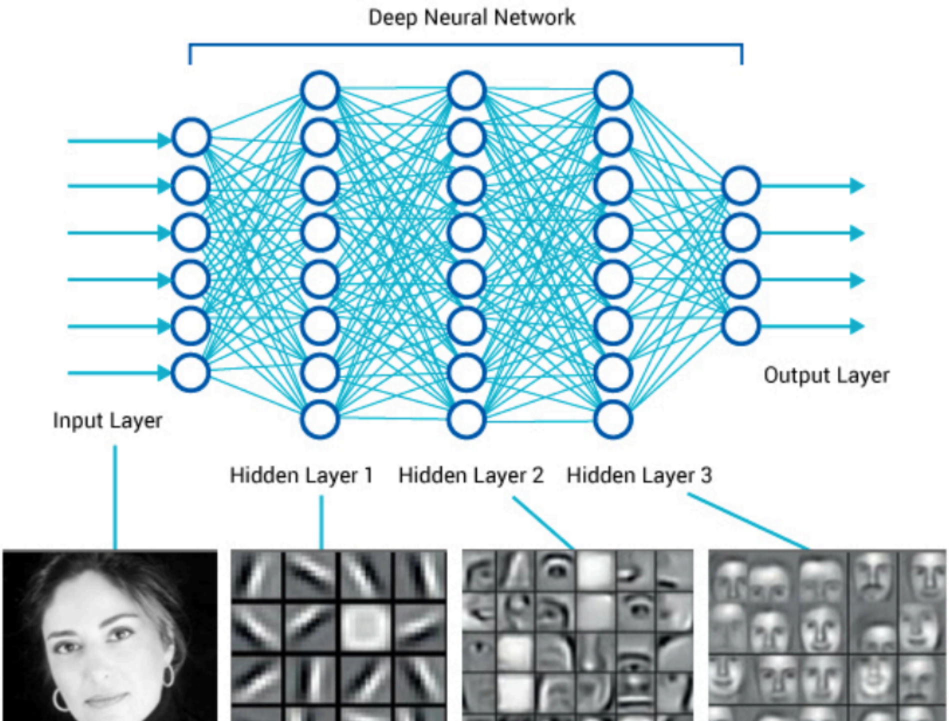
Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Started Deep Learning Revolution



Using ImageNet (Deng et al, 2009) image inventory


Deep Neural Network
Input Layer
Hidden Layer 1
Hidden Layer 2
Hidden Layer 3
Output Layer

What's the problem?

- Need to learn complicated matrices to capture features
- Not clear what you should pay attention to at any layer
- Despite nonlinearities, still need to learn matrices to create correct representations

What's the problem?

- Need to learn complicated matrices to capture features
- Not clear what you should pay attention to at any layer
- Despite nonlinearities, still need to learn matrices to create correct representations