

Convolutional Neural Networks (CNNs) dengan Keras

Alfan Farizki Wicaksono
Fakultas Ilmu Komputer
Universitas Indonesia

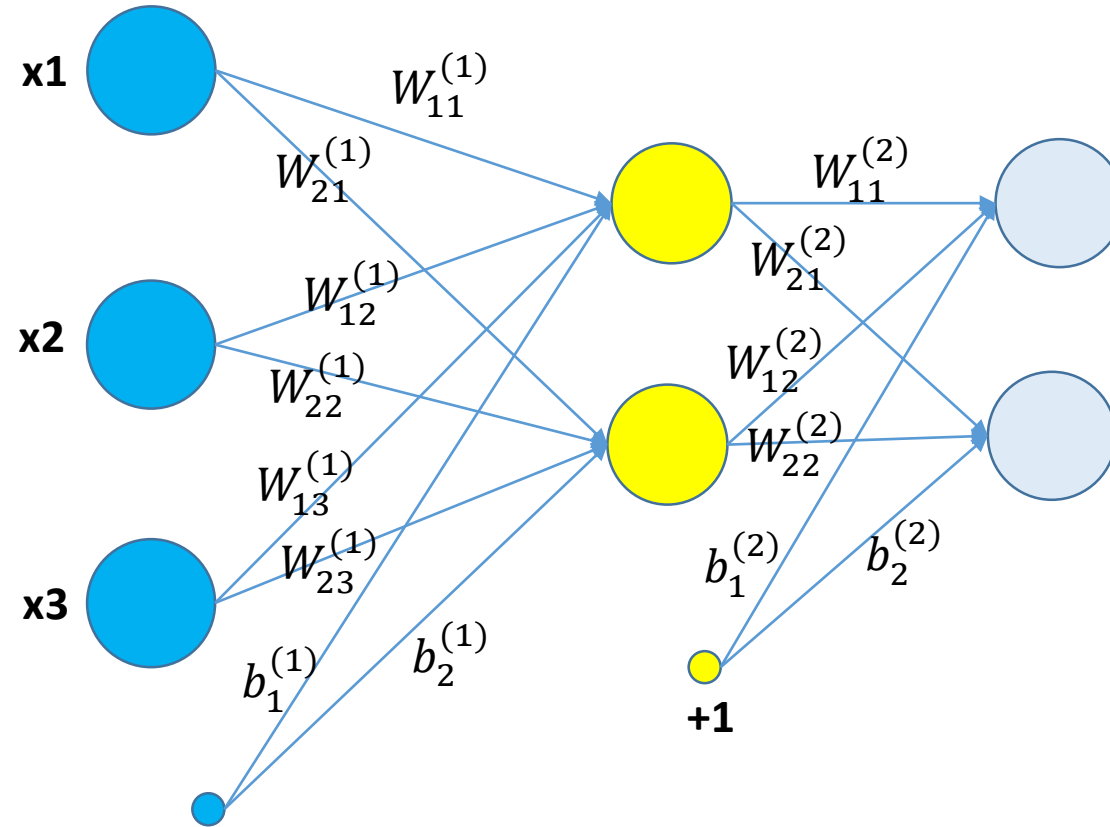
References

- <http://ufldl.stanford.edu/tutorial/>
- <http://cs231n.github.io/convolutional-networks/>
- <https://docs.gimp.org/en/plugin-convmatrix.html>

Fully-Connected Networks

- ANN biasa yang sudah dipelajari sebelumnya merupakan “fully connected networks”.
- Semua hidden unit terhubung dengan semua input unit.
 - Keras menyebutnya dengan istilah **Dense Layer**

Fully-Connected Networks



Fully-Connected Networks

- Seandainya input adalah sebuah image berukuran **28x28 pixel** (seperti kumpulan image pada dataset MNIST), input layer akan berukuran **784 unit**.
 - Masih feasible untuk melakukan komputasi berikutnya (learning)
- Bagaimana dengan image yang lebih besar?
 - 98x98 pixel: **9604 unit**
 - 128x128 pixel: **16384 unit !**
 - ...

Locally-Connected Networks

- **Solution:** to restrict the connections between the hidden units and the input units.
- **For image:** Each hidden unit will connect to only a small contiguous region of pixels in the input.
- **For other modalities:** there is often also a natural way to select “contiguous groups” of input units to connect to a single hidden unit as well.

Convolutions

- Natural images have the property of being “stationary”, meaning that the statistics of one part of the image are the same as any other part.
- This suggests that the features that we learn at one part of the image can also be applied to other parts of the image, and we can use the same features at all locations.

Proses Ekstraksi Fitur dengan Convolutions

Kernel/Filter 3x3

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

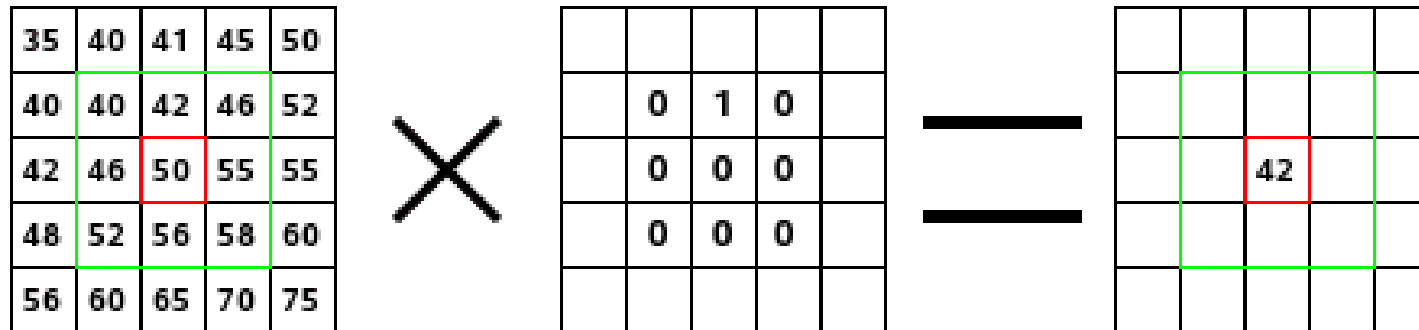
Image

4		

Convolved
Feature

Convolutions: salah satu teknik image processing

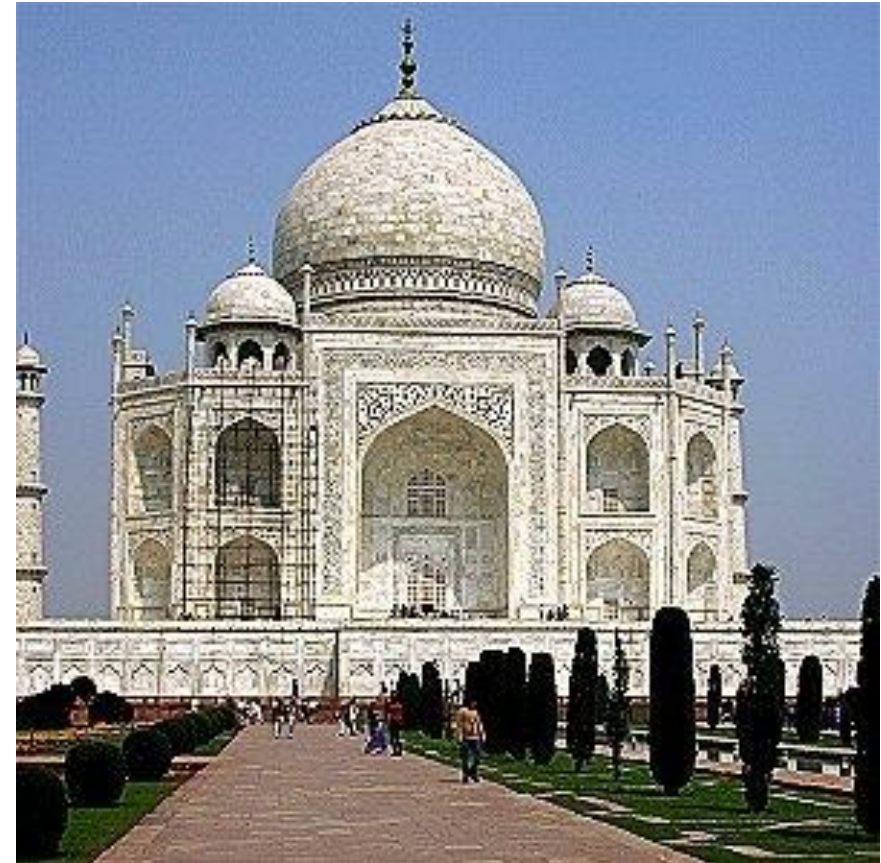
- Convolution is the treatment of a matrix by another one which is called “kernel”.



Convolutions: salah satu teknik image processing

- Here's what happened!
- **Sharpen**

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0



Convolutions: salah satu teknik image processing

- Here's what happened!
- **Blur**

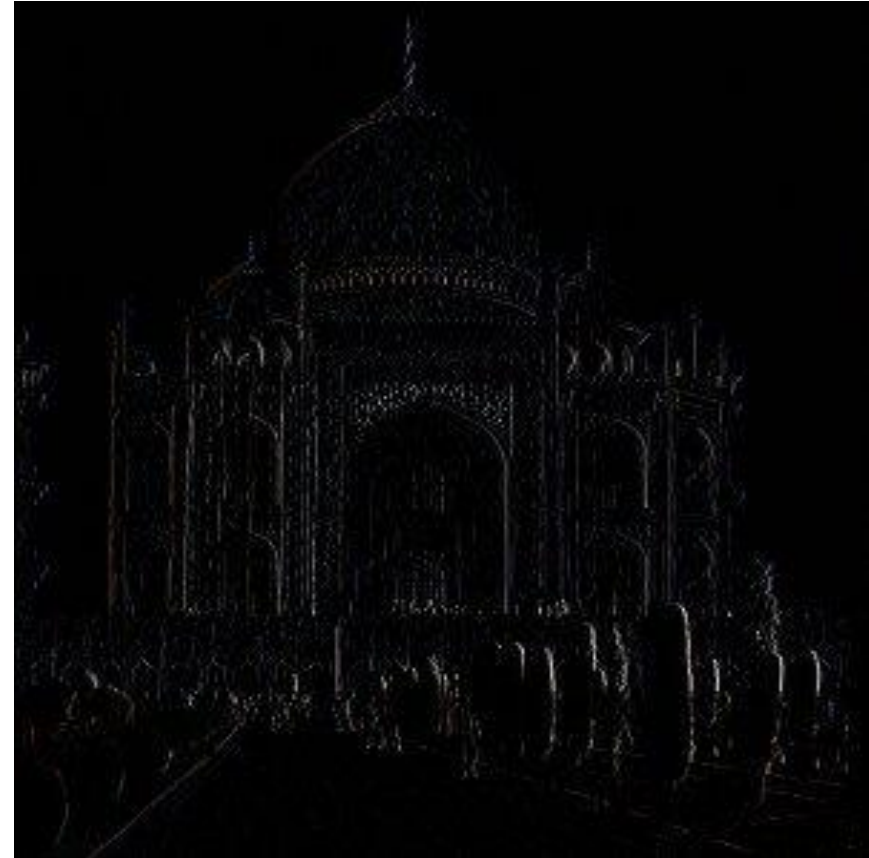
0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0



Convolutions: salah satu teknik image processing

- Here's what happened!
- **Edge Enhance**

	0	0	0	
	-1	1	0	
	0	0	0	



Convolutions: salah satu teknik image processing

- Here's what happened!
- **Edge Detect**

	0	1	0	
	1	-4	1	
	0	1	0	



Convolutions: salah satu teknik image processing

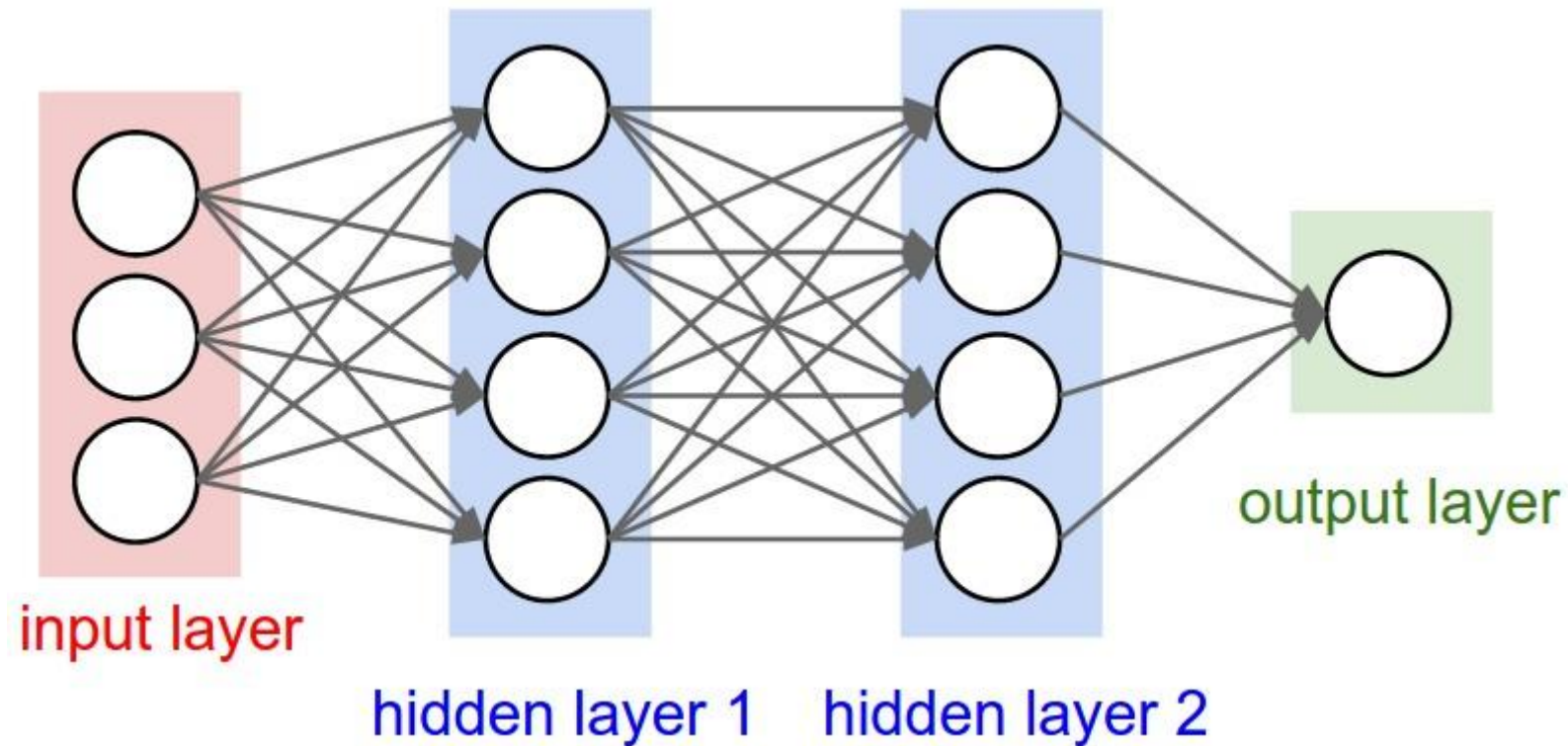
- Here's what happened!
- **Edge Detect**

	-2	-1	0	
	-1	1	1	
	0	1	2	



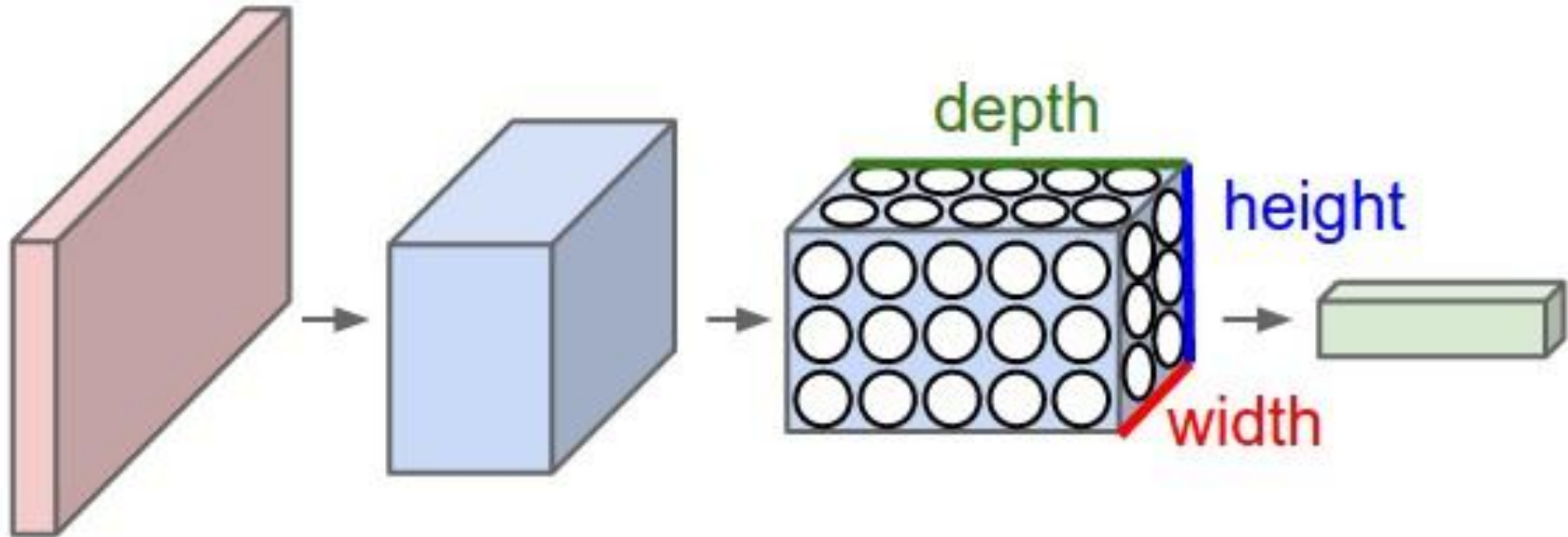
Convolutional Neural Networks

- Regular Feed-Forward ANNs



Convolutional Neural Networks

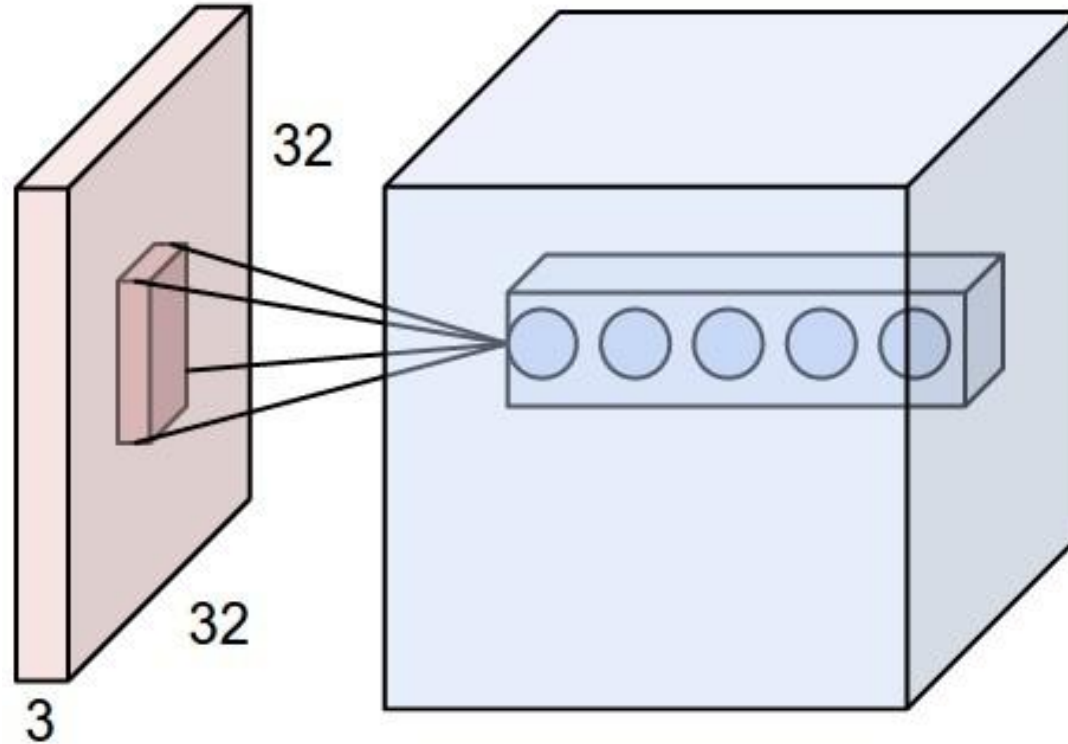
- CNNs



Convolutional Neural Networks

- CNNs

Dengan 5 Kernel/Filter



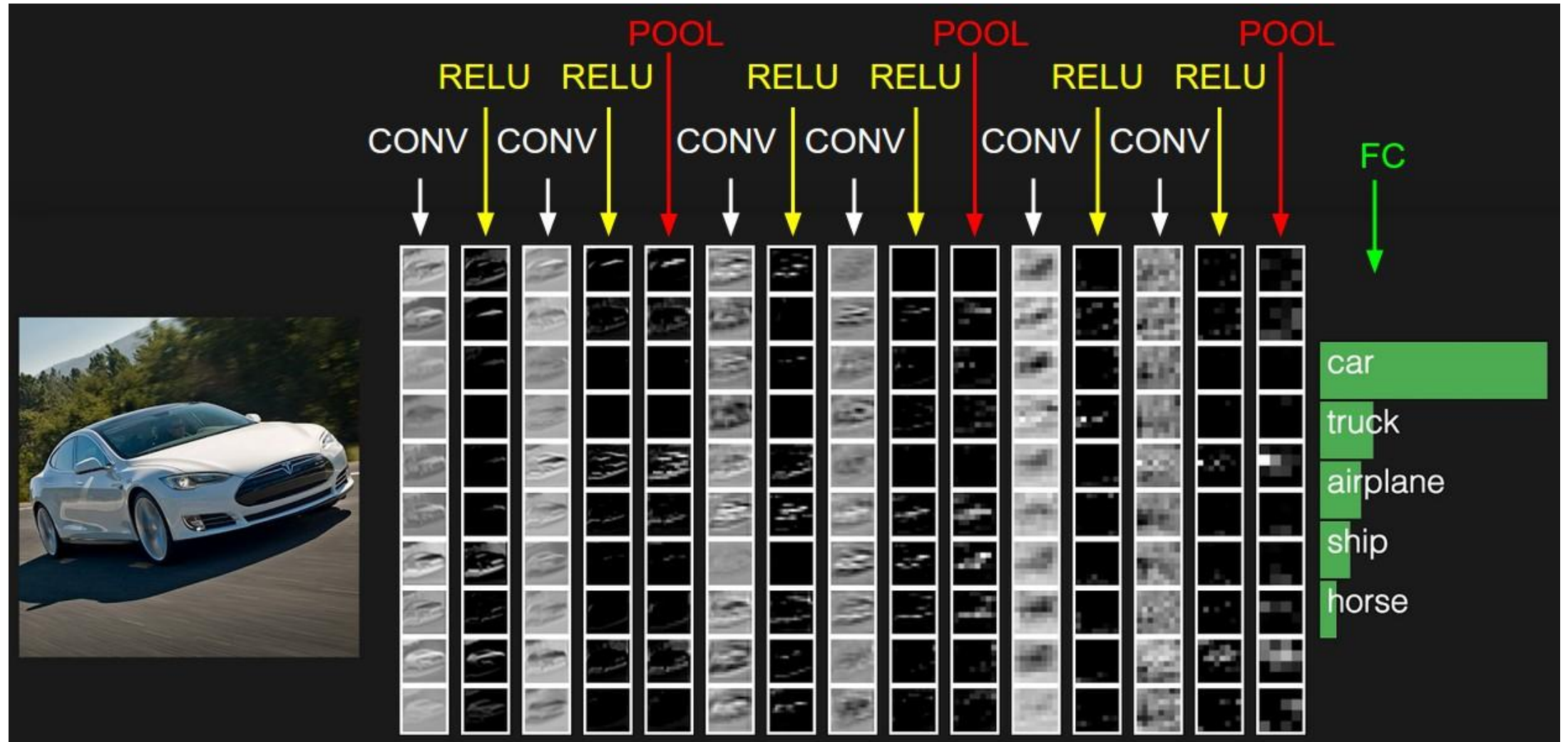
Convolutional Neural Networks

- Silakan baca:
- <http://cs231n.github.io/convolutional-networks/>

Convolutional Neural Networks

- Sebuah arsitektur Deep Learning yang mempunyai “ingredients”
 - Convolutional Layer
 - Pooling Layer
 - Fully-Connected Layer (regular ANN) di bagian akhirnya

Convolutional Neural Networks

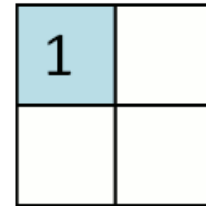
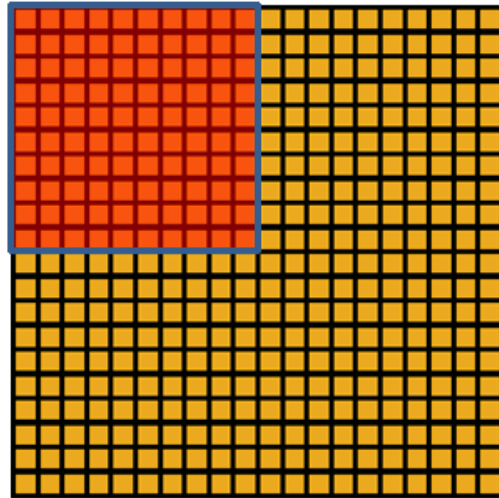


Pooling

- Setelah mendapatkan features yang diekstraksi dari **Convolutional Layer**, features yang dihasilkan terkadang masih berukuran sangat besar sehingga memperlambat proses training.
- Remember **stationary property**!
 - features that are useful in one region are also likely to be useful for other regions
- Thus, to describe a large image, one natural approach is to aggregate statistics of these features at various locations.

Pooling

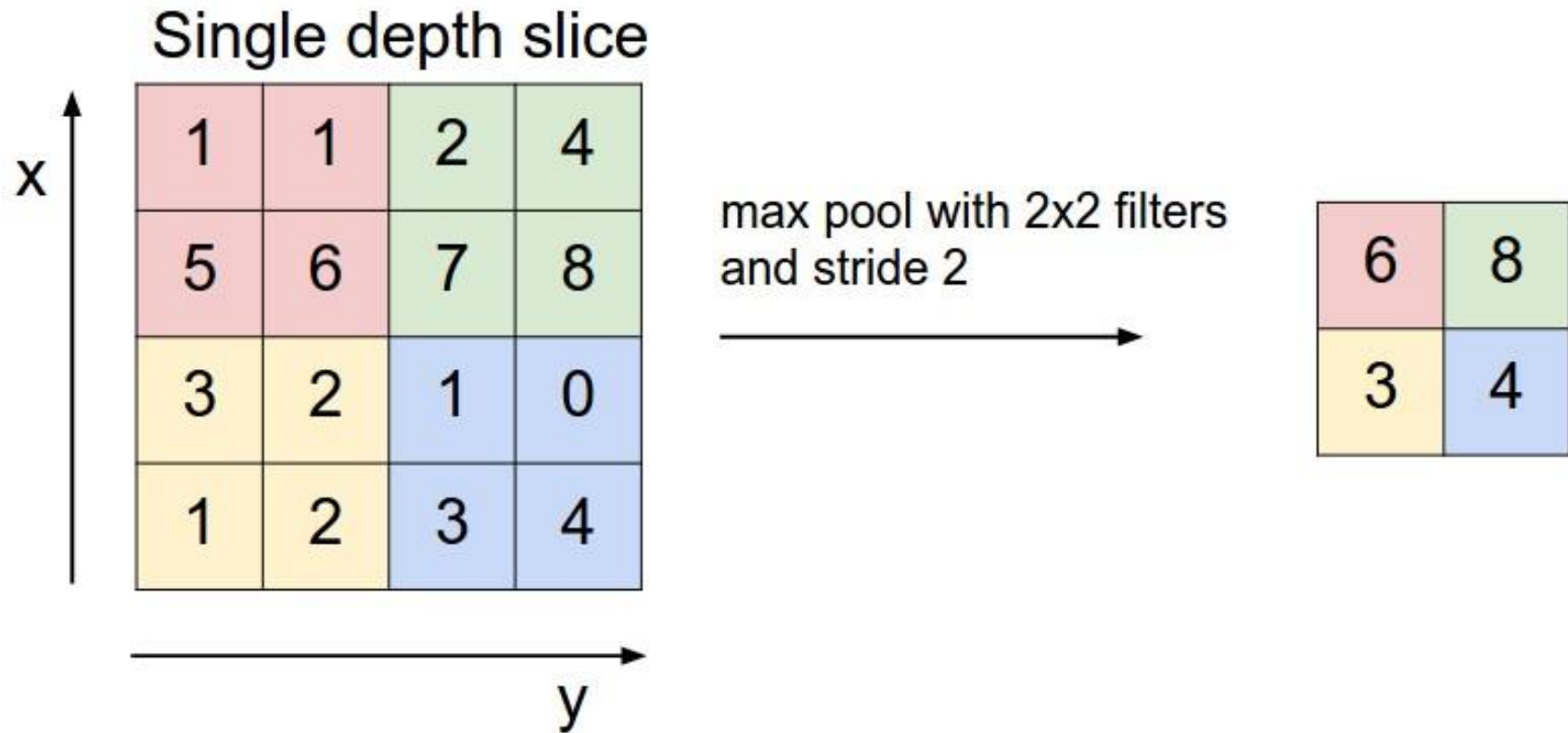
- For example, one could compute the mean (or max) value of a particular feature over a region of the image.



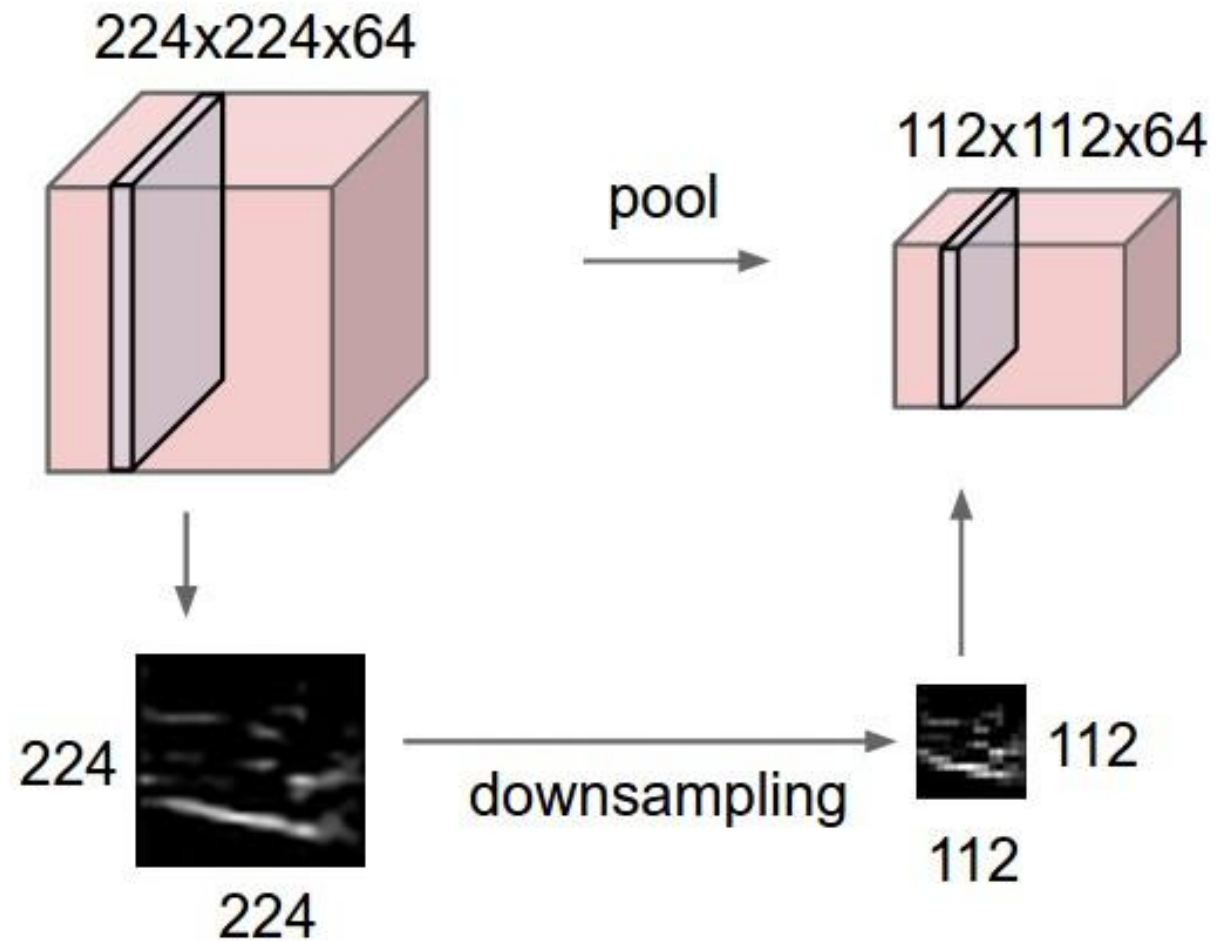
Convolved
feature

Pooled
feature

Max Pooling



Pooling



CNNs in Keras

```
model = Sequential()

# convolution layer, 64 filter berukuran 3x3
# input shape=ukuran image, 3x256x256
model.add(Convolution2D(64, 3, 3, border_mode='same',
                        input_shape=(3, 256, 256)))
# output layer berdimensi (num_sample, 64, 256, 256)

model.add(Activation('relu'))

# convolution layer, 32 filter berukuran 3x3
model.add(Convolution2D(32, 3, 3, border_mode='same'))
# output layer berdimensi berdimensi (num_sample, 32, 256, 256)
```