# Deep Tracking & Flow

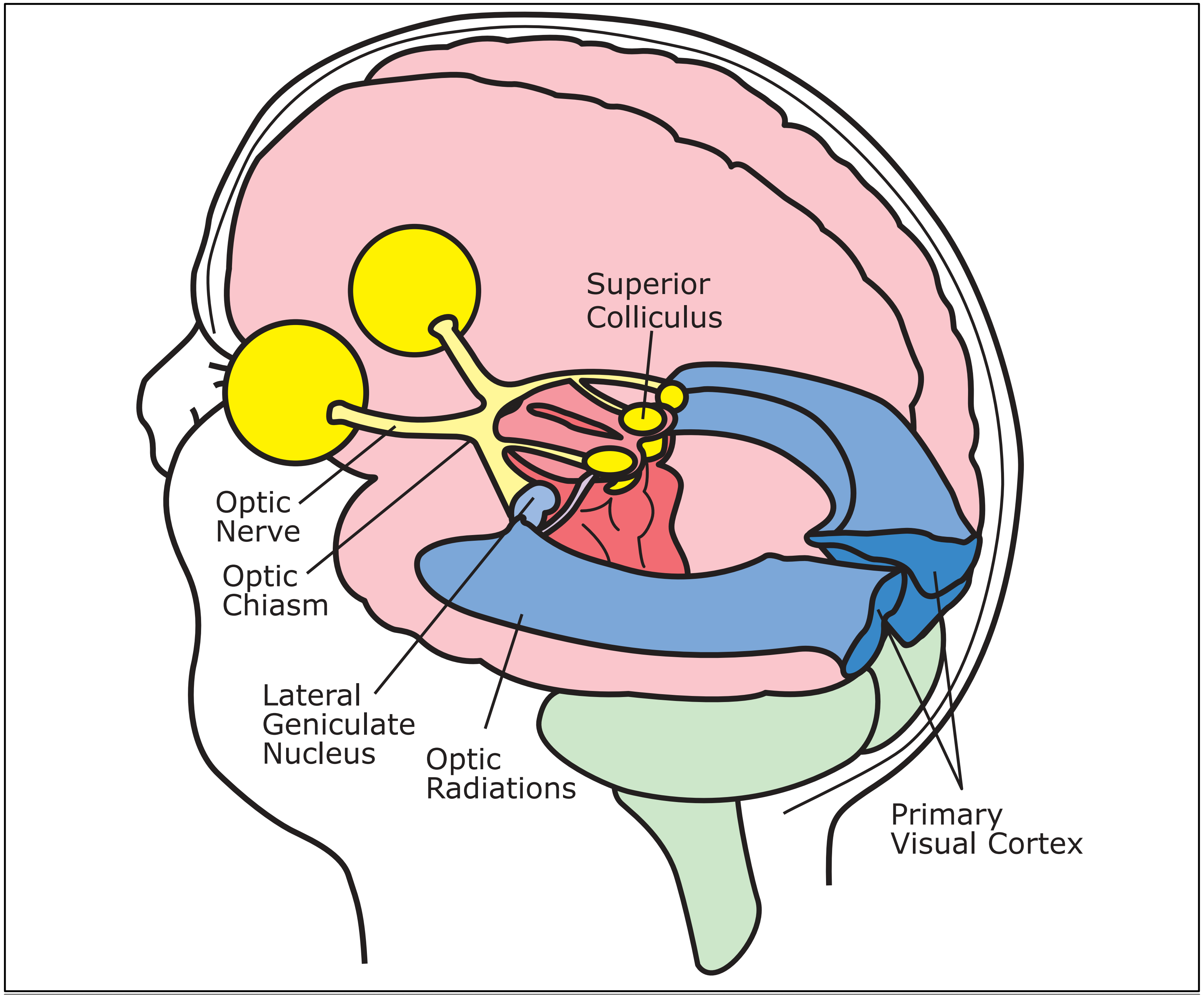
Instructor - Simon Lucey

**16-623 - Designing Computer Vision Apps**

# Today

- Deep Features
- Deep Tracking
- Deep Flow

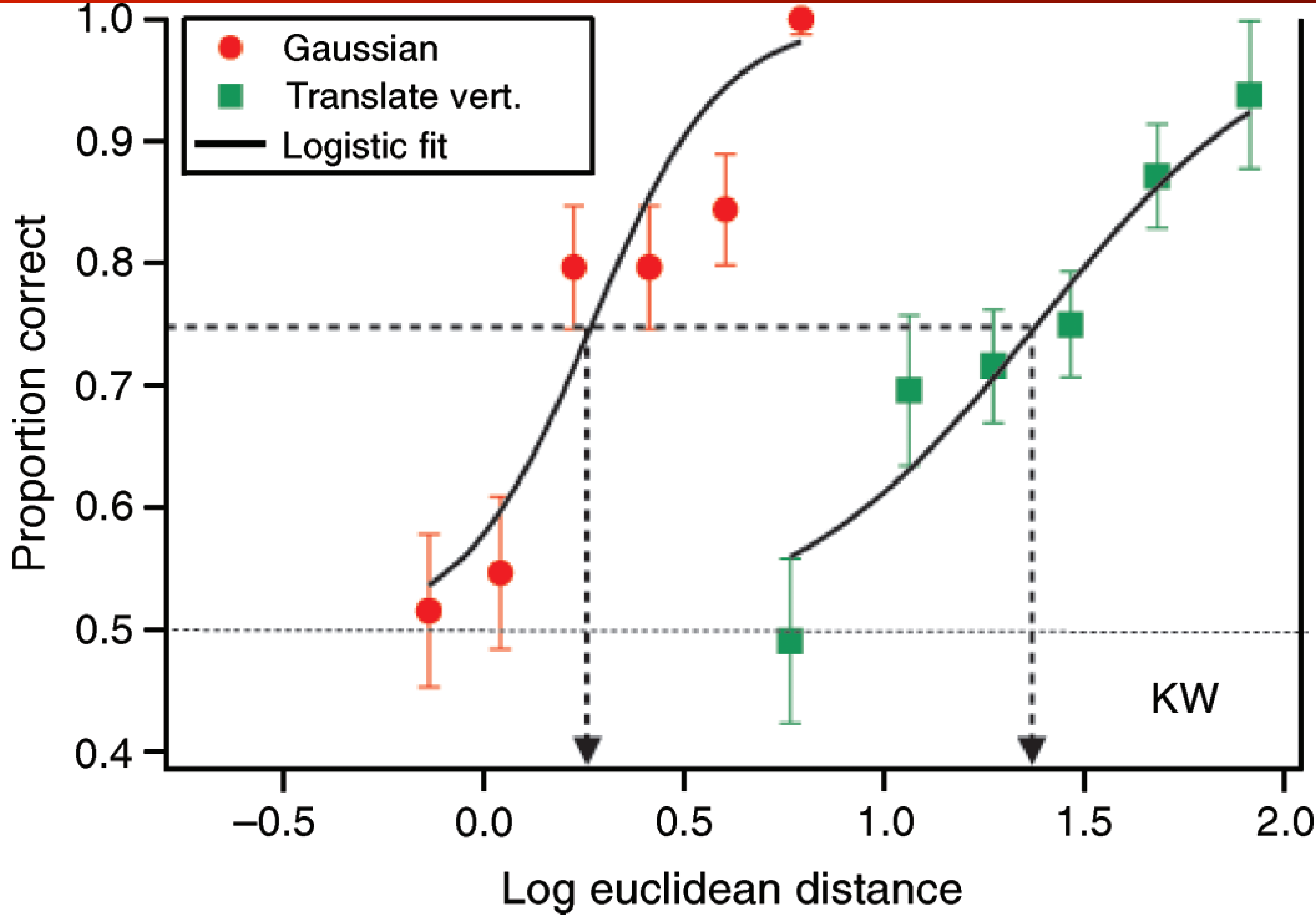
# Primary Visual Cortex



# Spatial Sensitivity

Kingdom, Field, Olmos, 2007

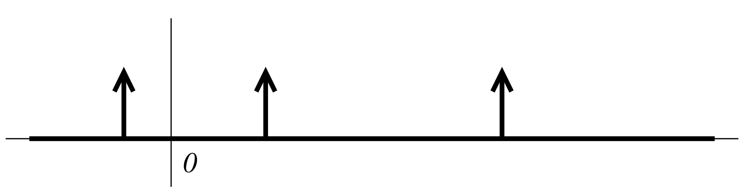
# Spatial Sensitivity



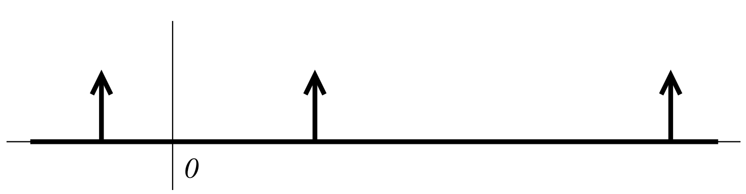

Kingdom, Field, Olmos, 2007

# Handling Geometric Distortion

- As pointed out in seminal work by Berg and Malik (CVPR'01) the effectiveness of SSD will degrade with significant viewpoint change.
- Two options to match local image patches:-




"1D Patch"

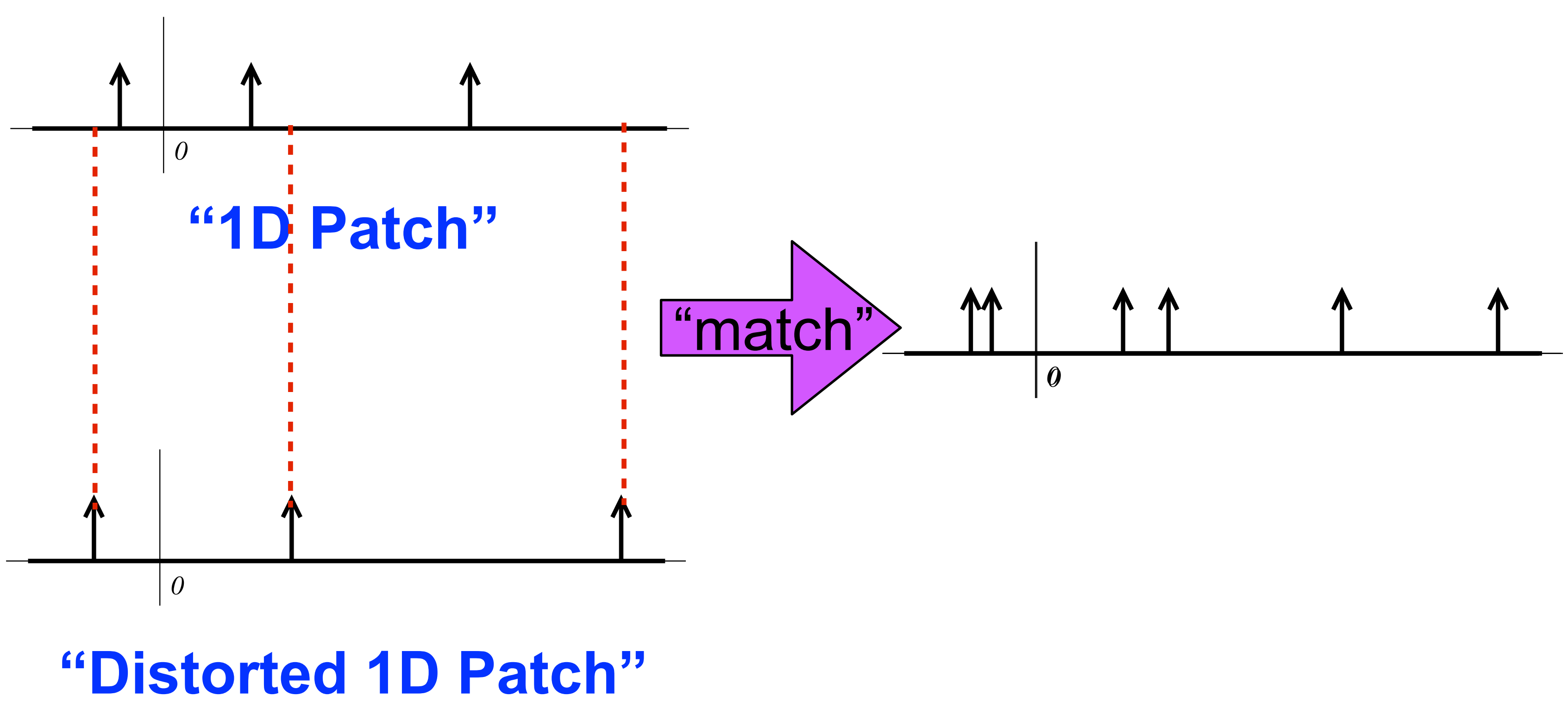



"Distorted 1D Patch"

Source: A. C. Berg

# Handling Geometric Distortion

- As pointed out in seminal work by Berg and Malik (CVPR'01) the effectiveness of SSD will degrade with significant viewpoint change.
- Two options to match local image patches:-



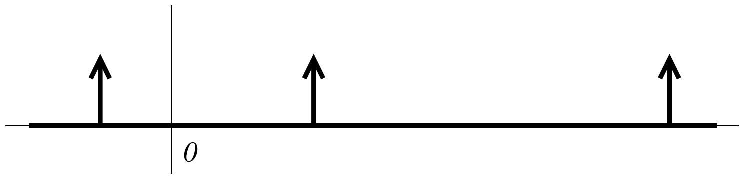

Source: A. C. Berg

# Handling Geometric Distortion

- As pointed out in seminal work by Berg and Malik (CVPR'01) the effectiveness of SSD will degrade with significant viewpoint change.
- Two options to match local image patches:-

1. simultaneously estimate the distortion and position of matching patch


"1D Patch"

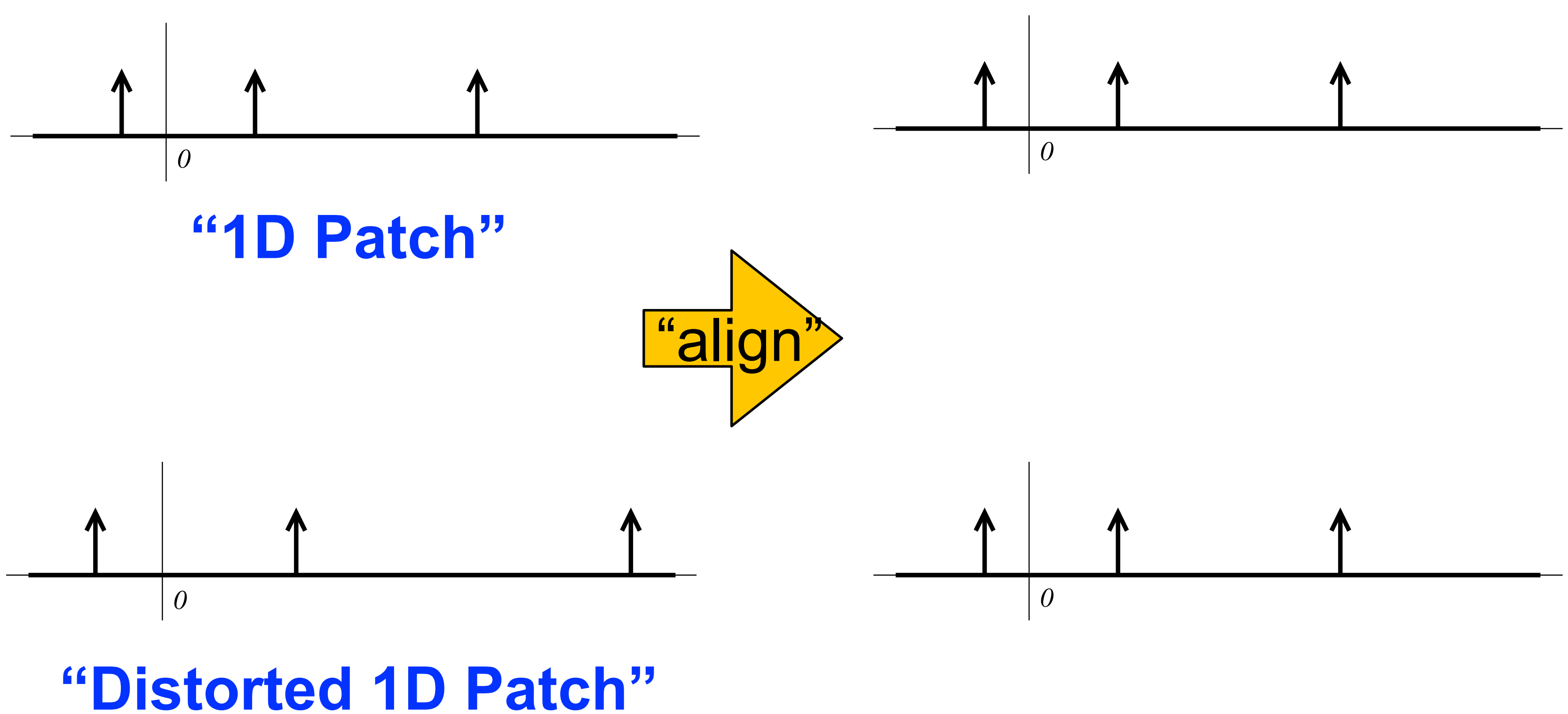



"Distorted 1D Patch"

# Handling Geometric Distortion

- As pointed out in seminal work by Berg and Malik (CVPR'01) the effectiveness of SSD will degrade with significant viewpoint change.
- Two options to match local image patches:-

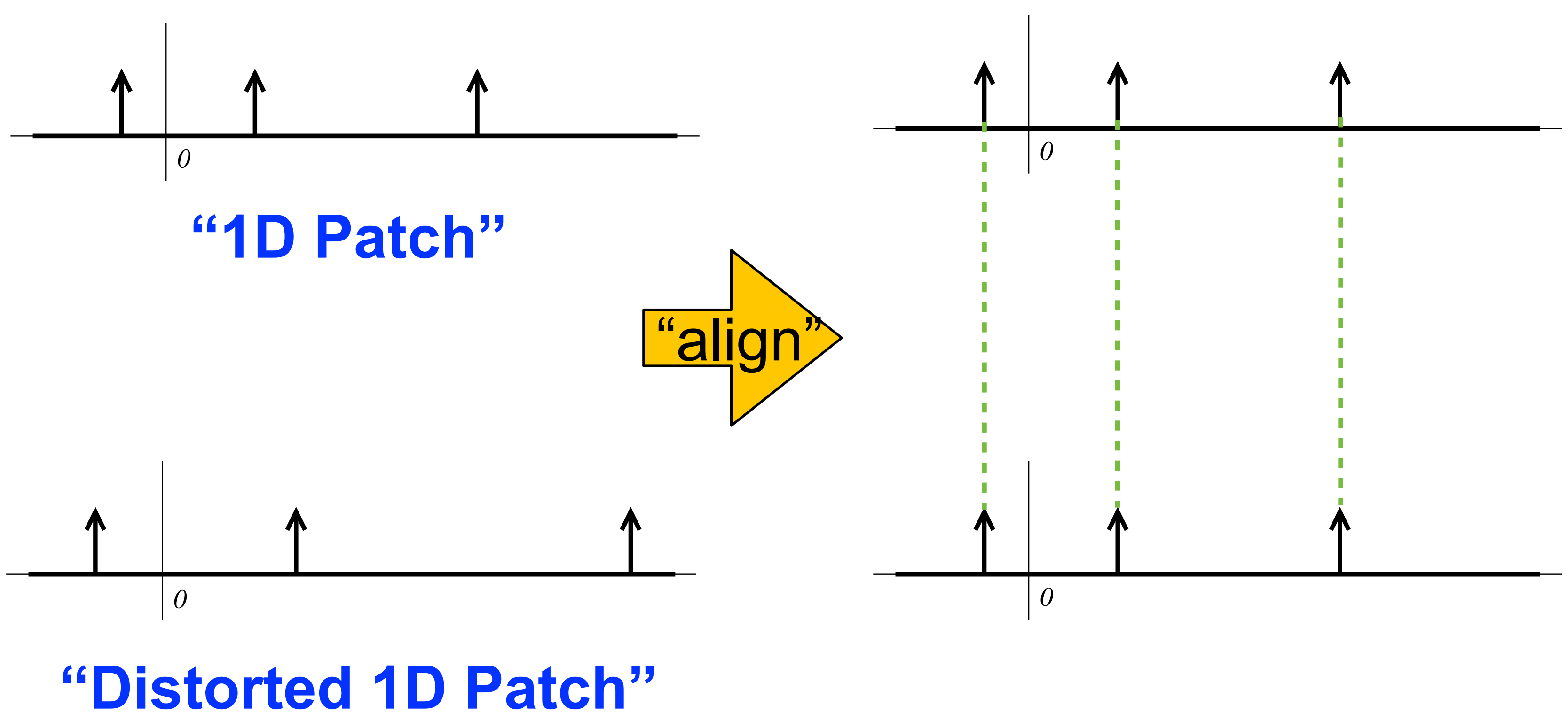
1. simultaneously estimate the distortion and position of matching patch



# Handling Geometric Distortion

- As pointed out in seminal work by Berg and Malik (CVPR'01) the effectiveness of SSD will degrade with significant viewpoint change.
- Two options to match local image patches:-

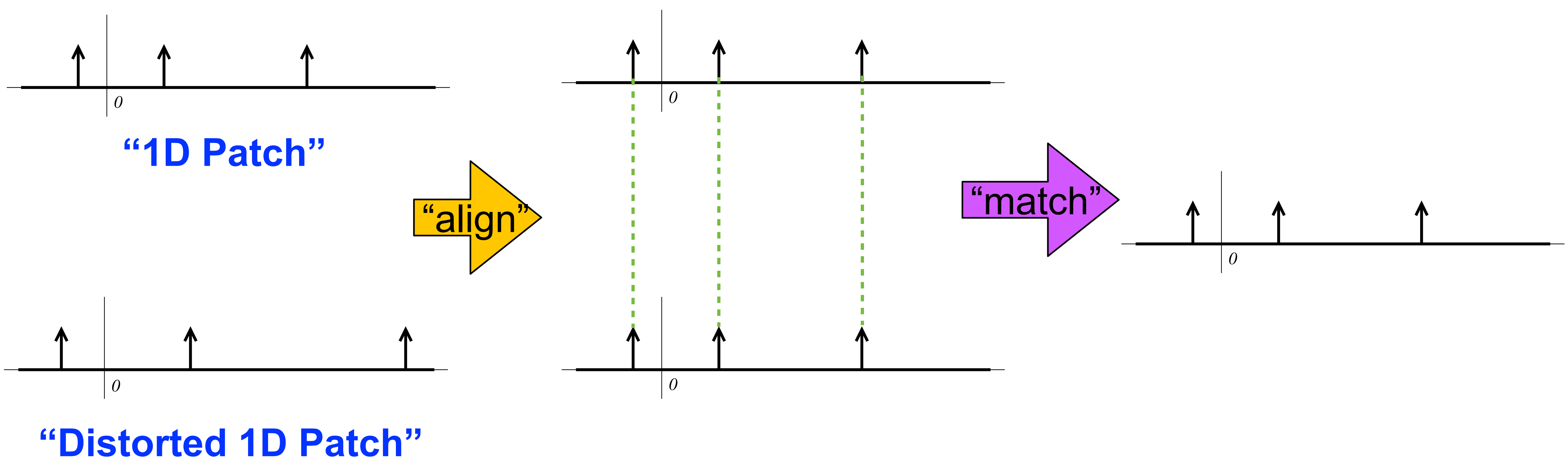
1. simultaneously estimate the distortion and position of matching patch



# Handling Geometric Distortion

- As pointed out in seminal work by Berg and Malik (CVPR'01) the effectiveness of SSD will degrade with significant viewpoint change.
- Two options to match local image patches:-

1. simultaneously estimate the distortion and position of matching patch

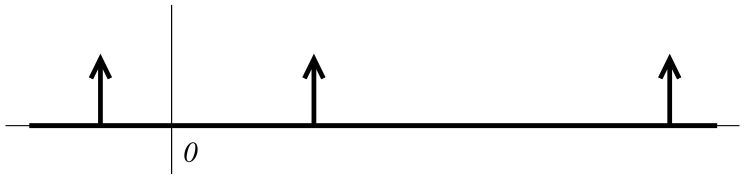


# Handling Geometric Distortion

- As pointed out in seminal work by Berg and Malik (CVPR'01) the effectiveness of SSD will degrade with significant viewpoint and/or illumination change.
- Two options to match patches:-

1. simultaneously estimate the distortion and position of matching patch.
2. to "blur" the template window performing matching coarse-to-fine.


"1D Patch"

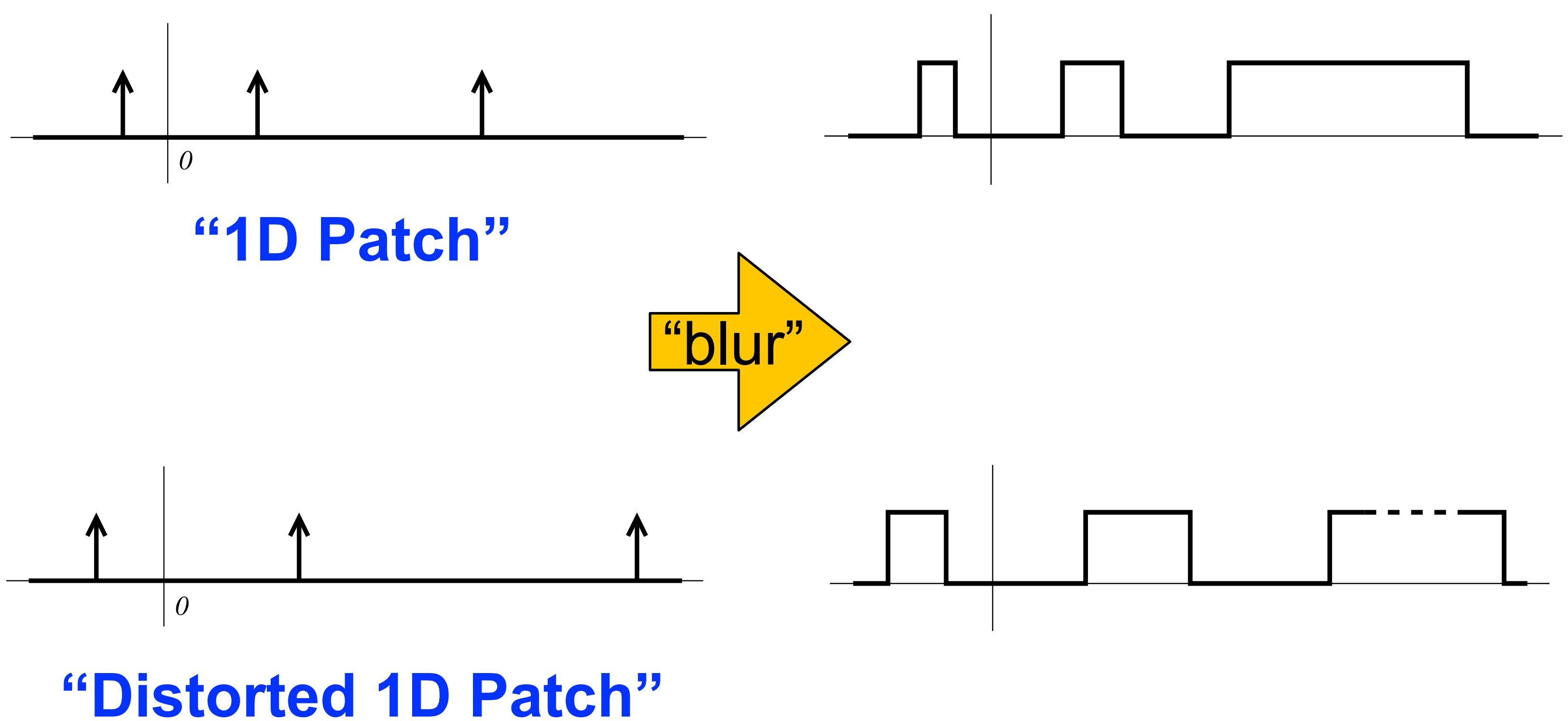



"Distorted 1D Patch"

# Handling Geometric Distortion

- As pointed out in seminal work by Berg and Malik (CVPR'01) the effectiveness of SSD will degrade with significant viewpoint and/or illumination change.
- Two options to match patches:-

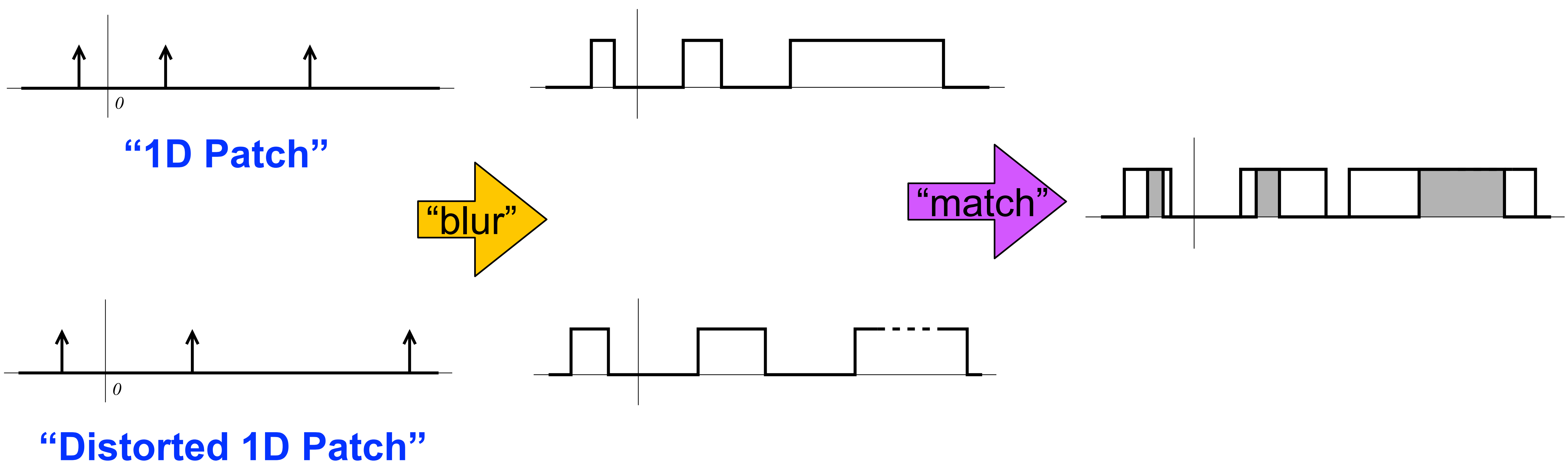
1. simultaneously estimate the distortion and position of matching patch.
2. to "blur" the template window performing matching coarse-to-fine.



# Handling Geometric Distortion

- As pointed out in seminal work by Berg and Malik (CVPR'01) the effectiveness of SSD will degrade with significant viewpoint and/or illumination change.
- Two options to match patches:-

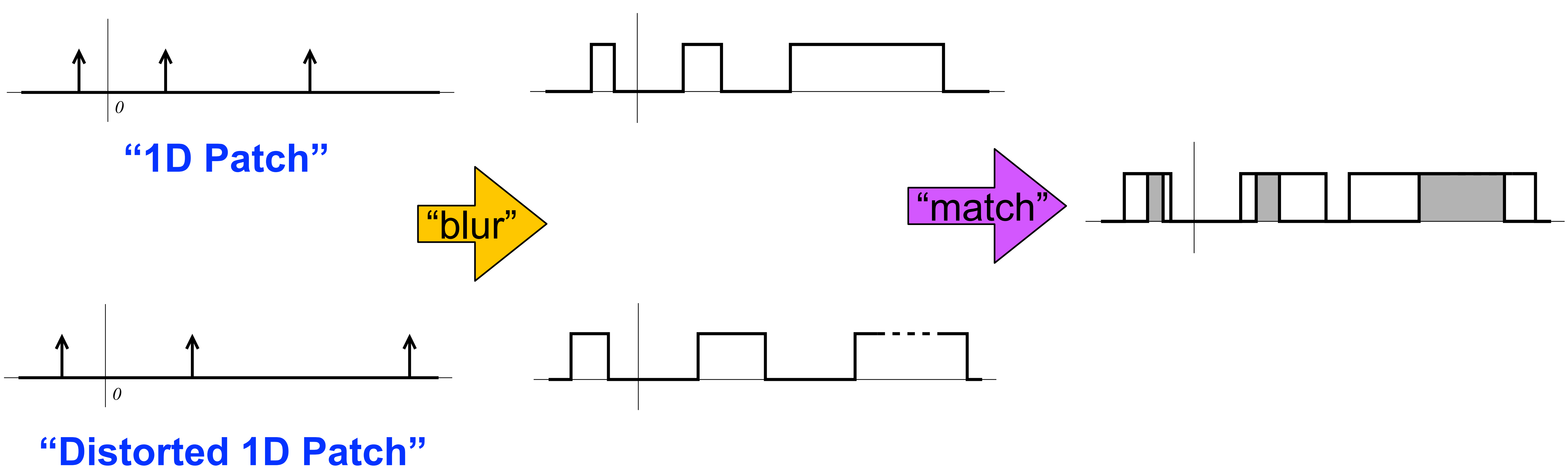
1. simultaneously estimate the distortion and position of matching patch.
2. to "blur" the template window performing matching coarse-to-fine.



# Handling Geometric Distortion

- As pointed out in seminal work by Berg and Malik (CVPR'01) the effectiveness of SSD will degrade with significant viewpoint and/or illumination change.
- Two options to match patches:-

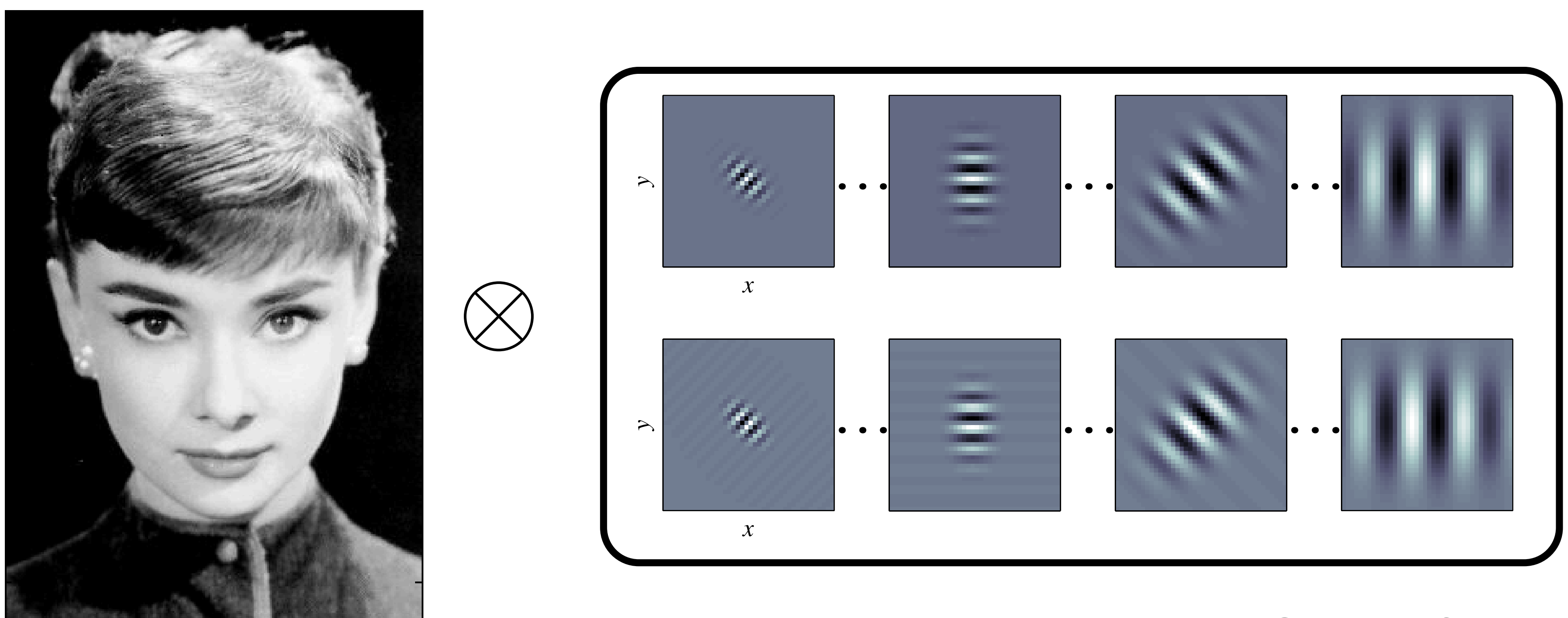
1. simultaneously estimate the distortion and position of matching patch.
2. to "blur" the template window performing matching coarse-to-fine.




*Option 2 is attractive, low computational cost!*

# Sparseness and Positiveness

- Blurring only works if the signals being matched are sparse and positive.
- Unfortunately natural images are neither.
- Combination of oriented filter banks and rectification can remedy this problem with little loss in performance.

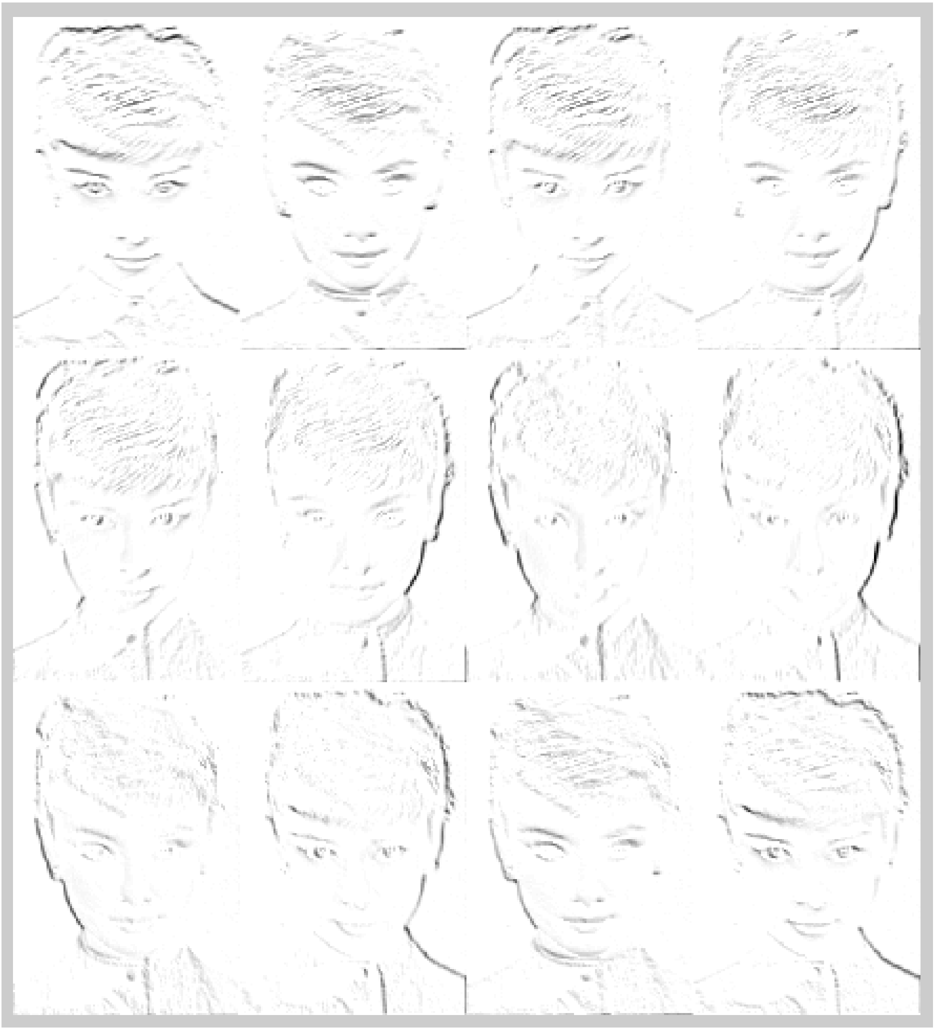



e.g., oriented gradients, Gabor filters

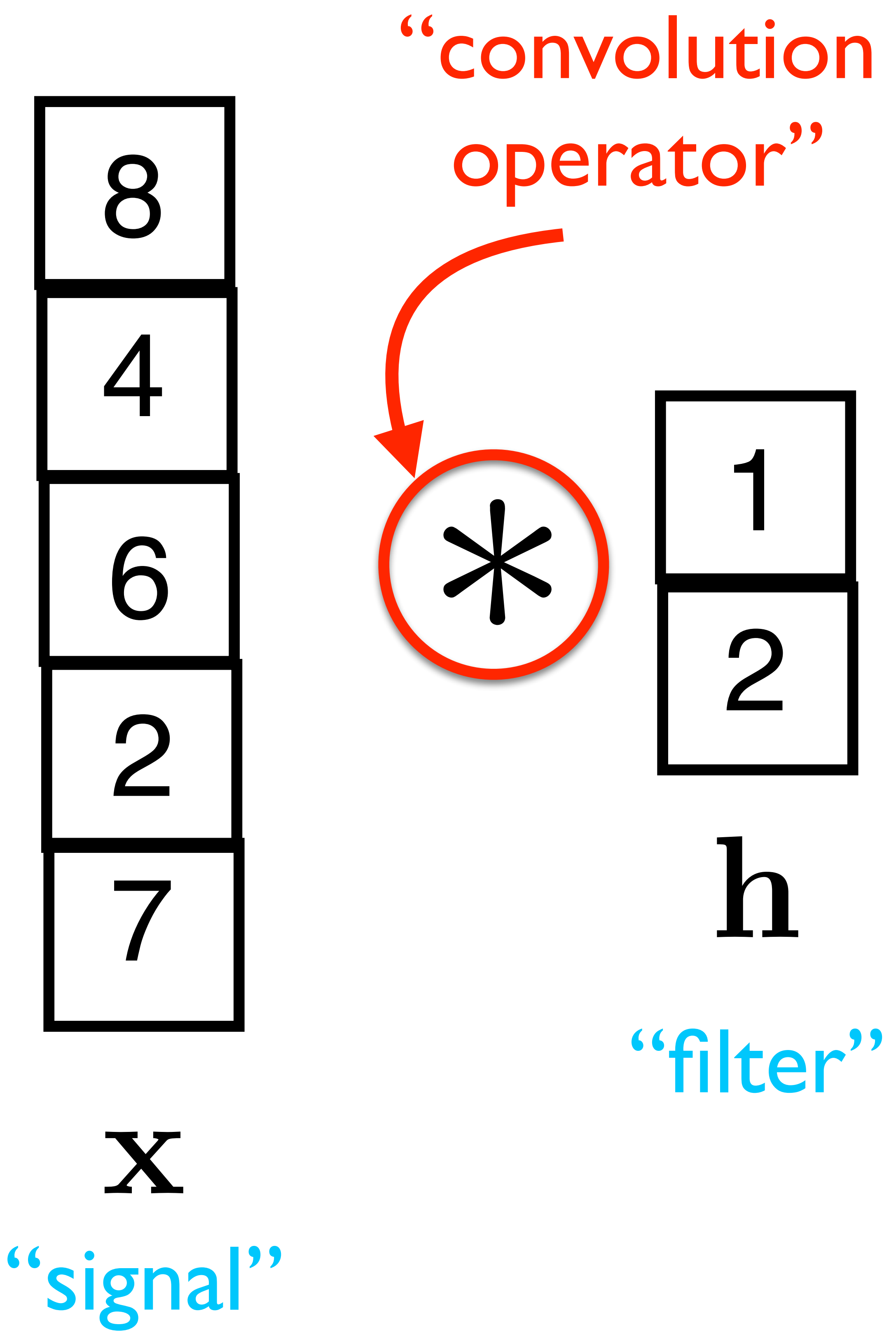

e.g., sigmoid, squared, relu, etc.

# Sparseness and Positiveness

- Blurring only works if the signals being matched are sparse and positive.
- Unfortunately natural images are neither.
- Combination of oriented filter banks and rectification can remedy this problem with little loss in performance.

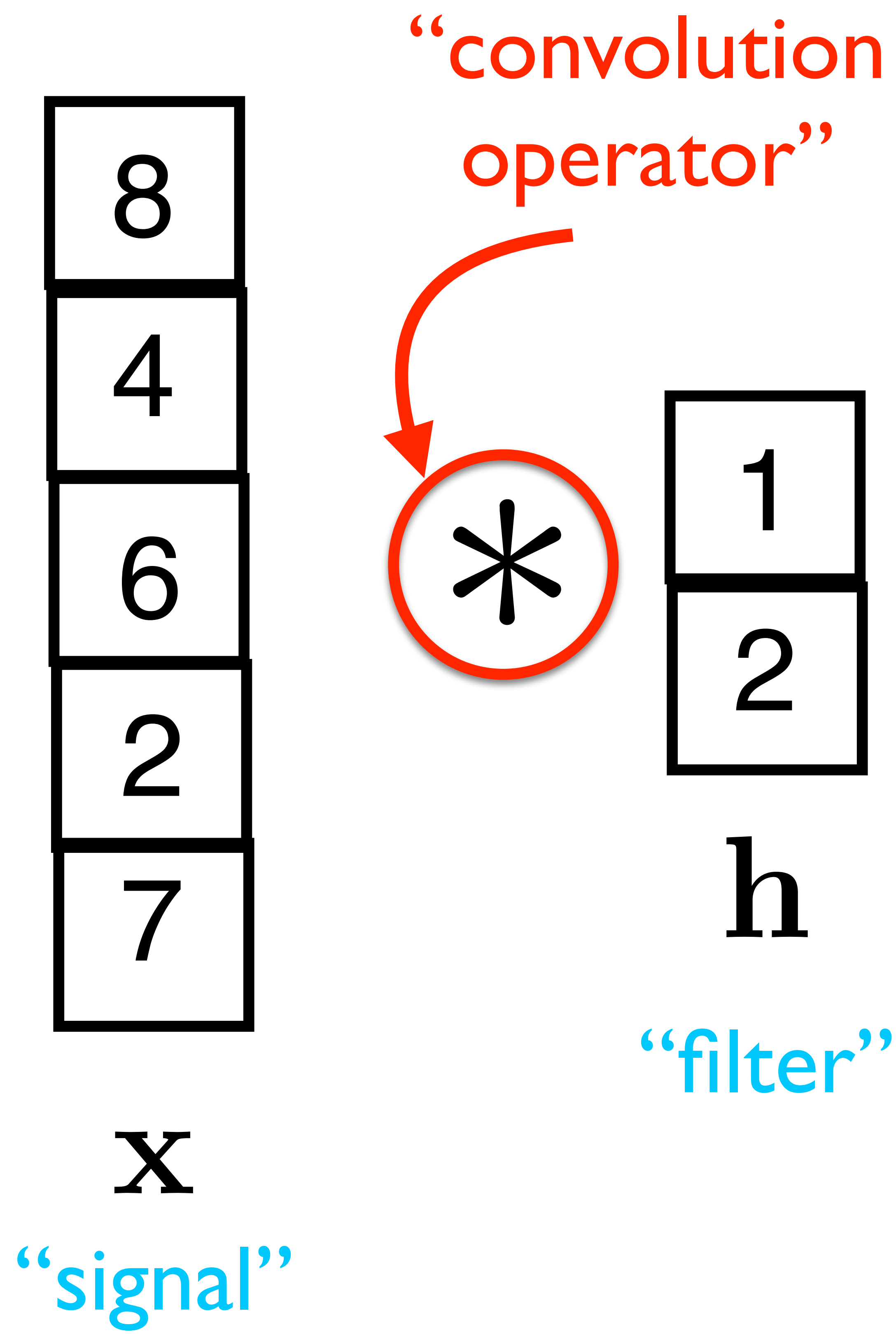


# Reminder: Convolution

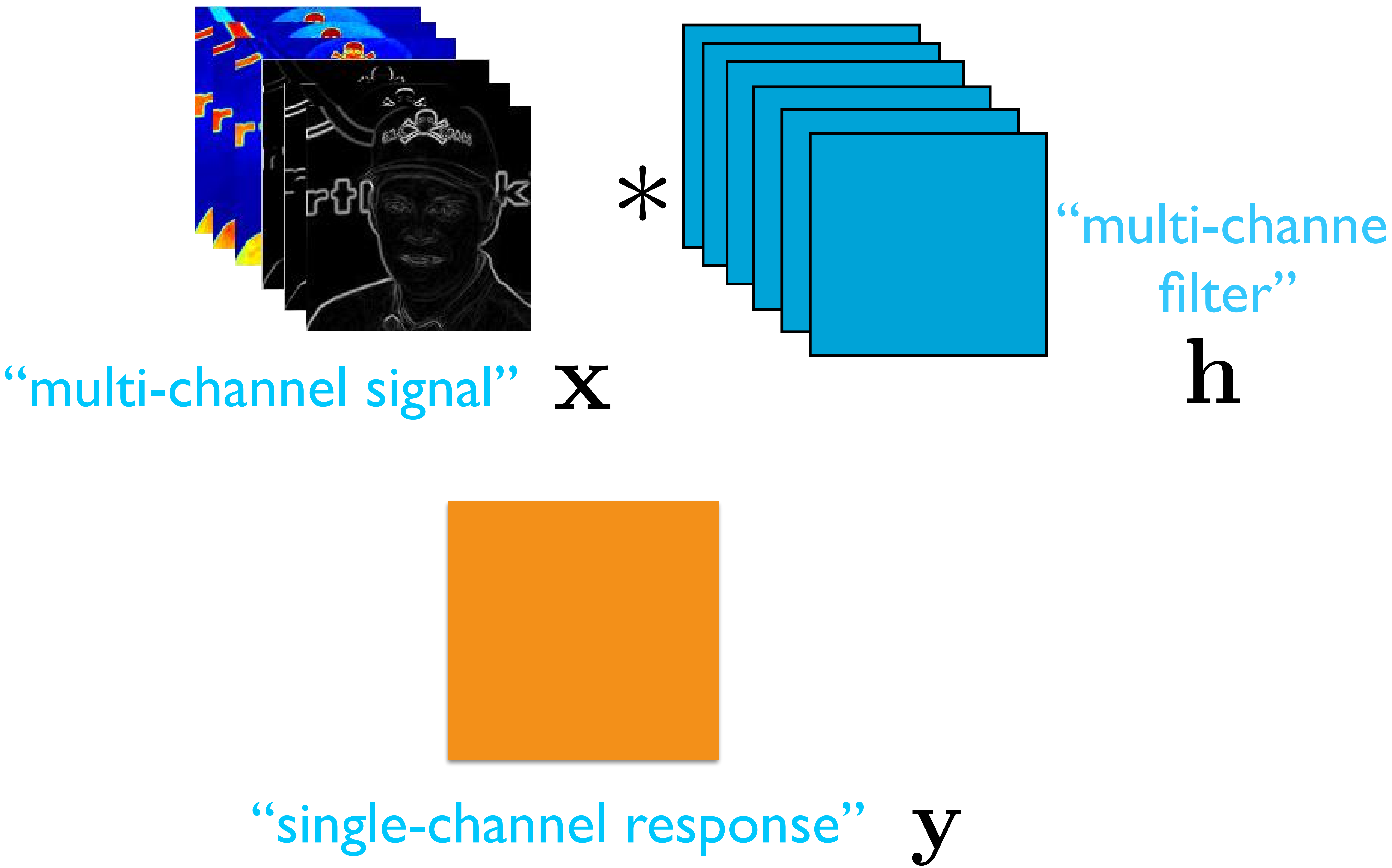


# Reminder: Convolution

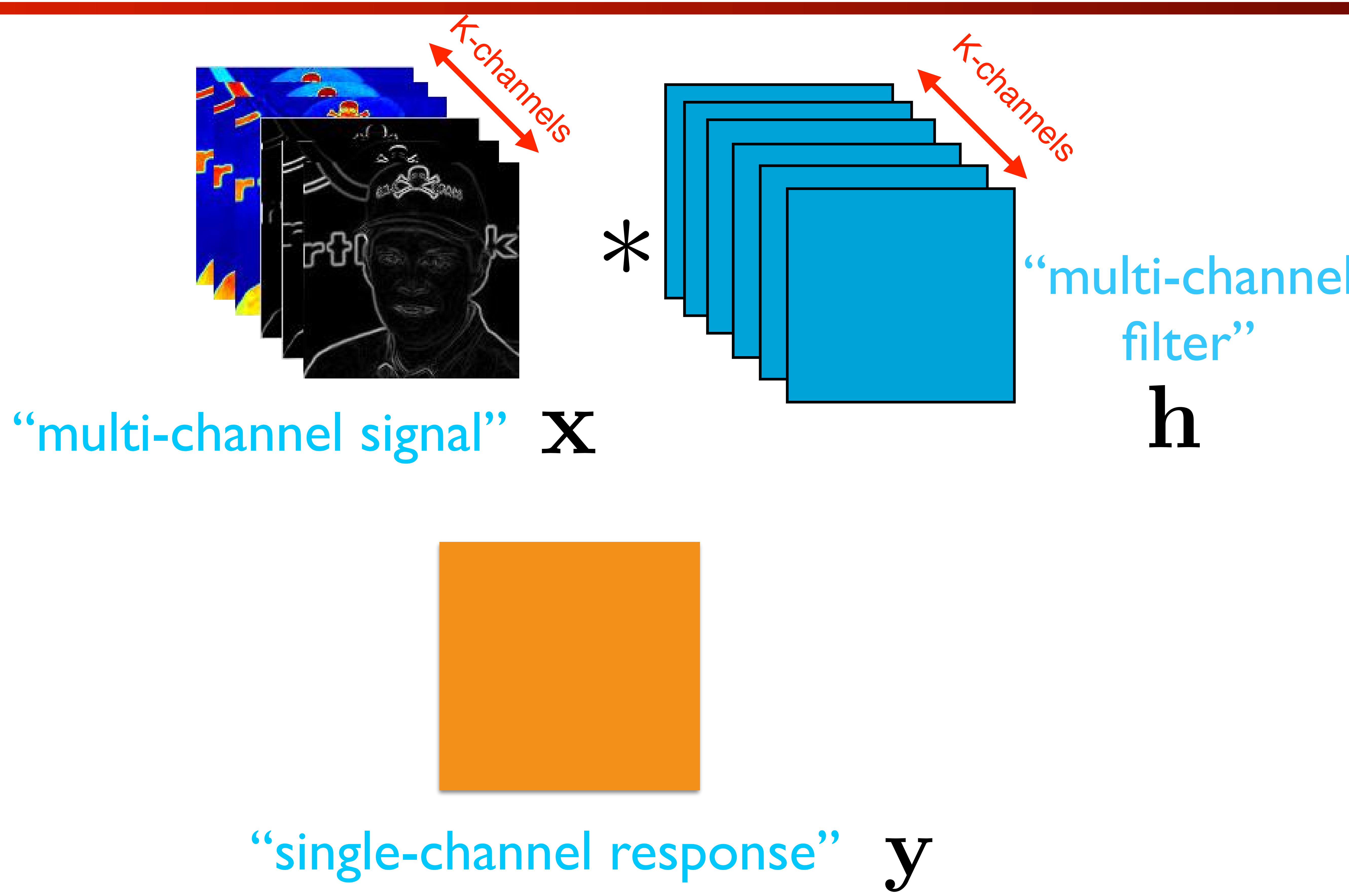
```
>> conv(x,h,'valid')
ans =

    20
    14
    14
    11
```



# Multi-Channel Convolution



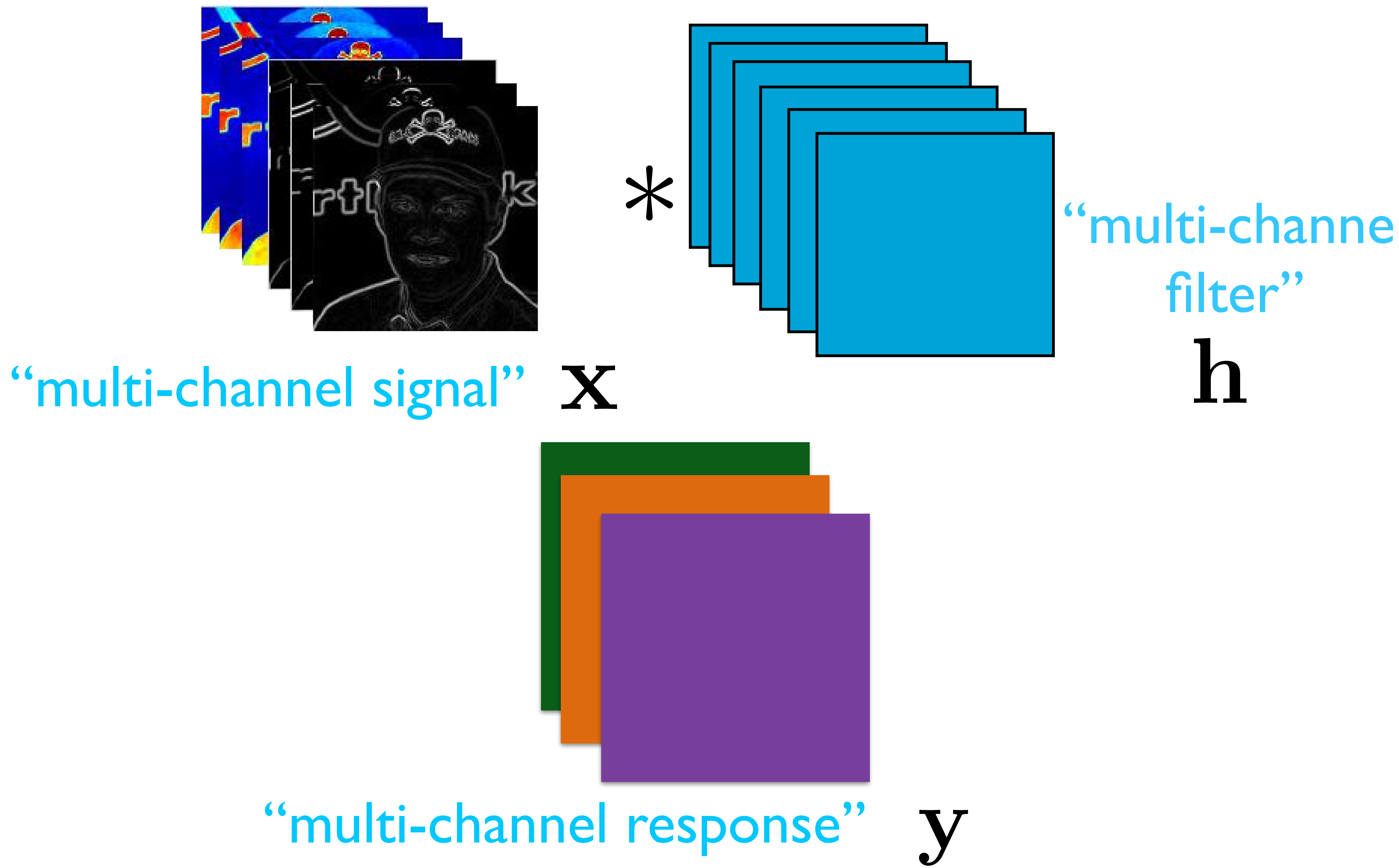
# Multi-Channel Convolution



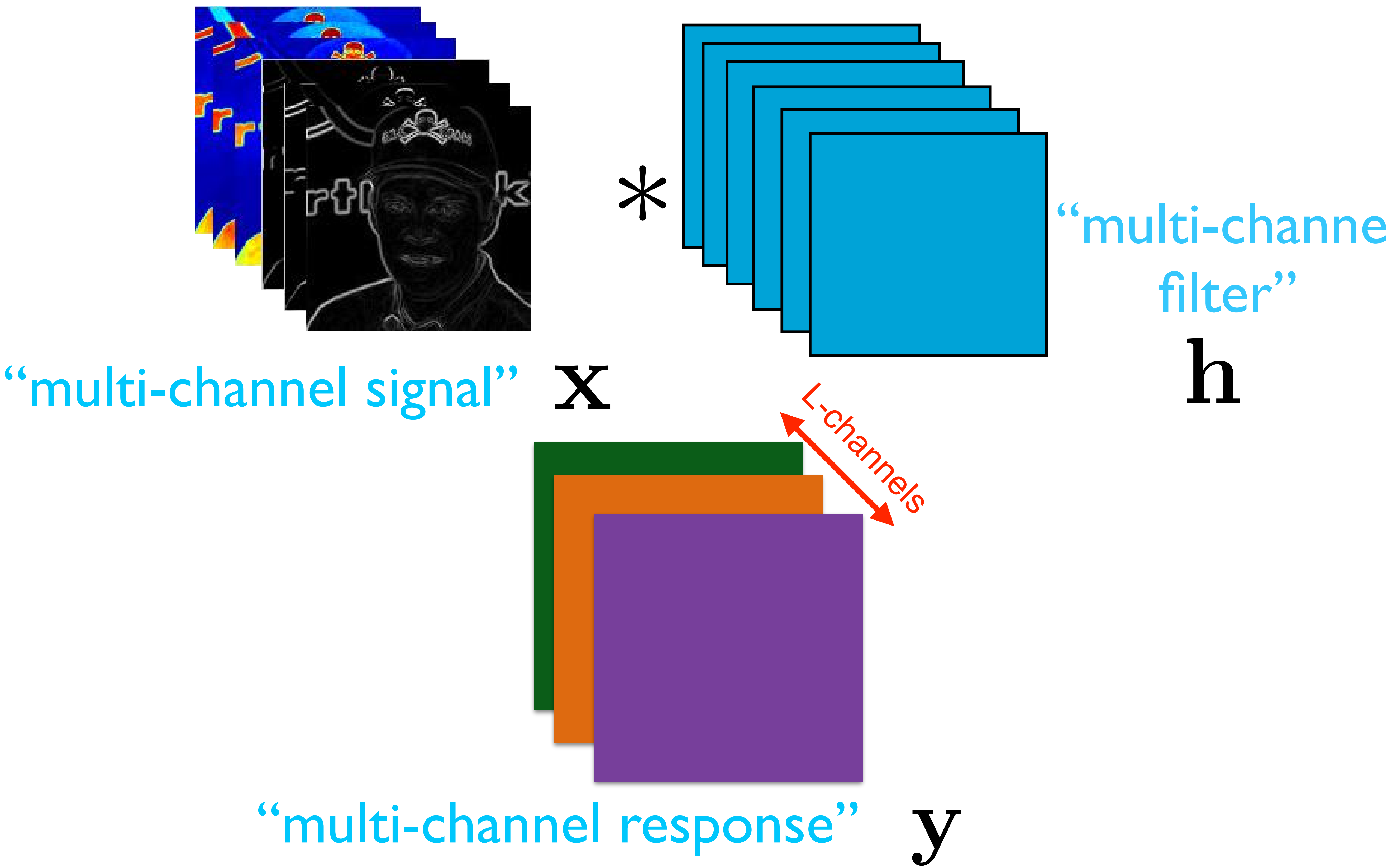
# Multi-Channel Convolution

$$\mathbf{y} = \sum_{k=1}^{K} \mathbf{x}^{(k)} * \mathbf{h}^{(k)}$$

# Multi-Channel Convolution



# Multi-Channel Convolution



# Multi-Channel Convolution

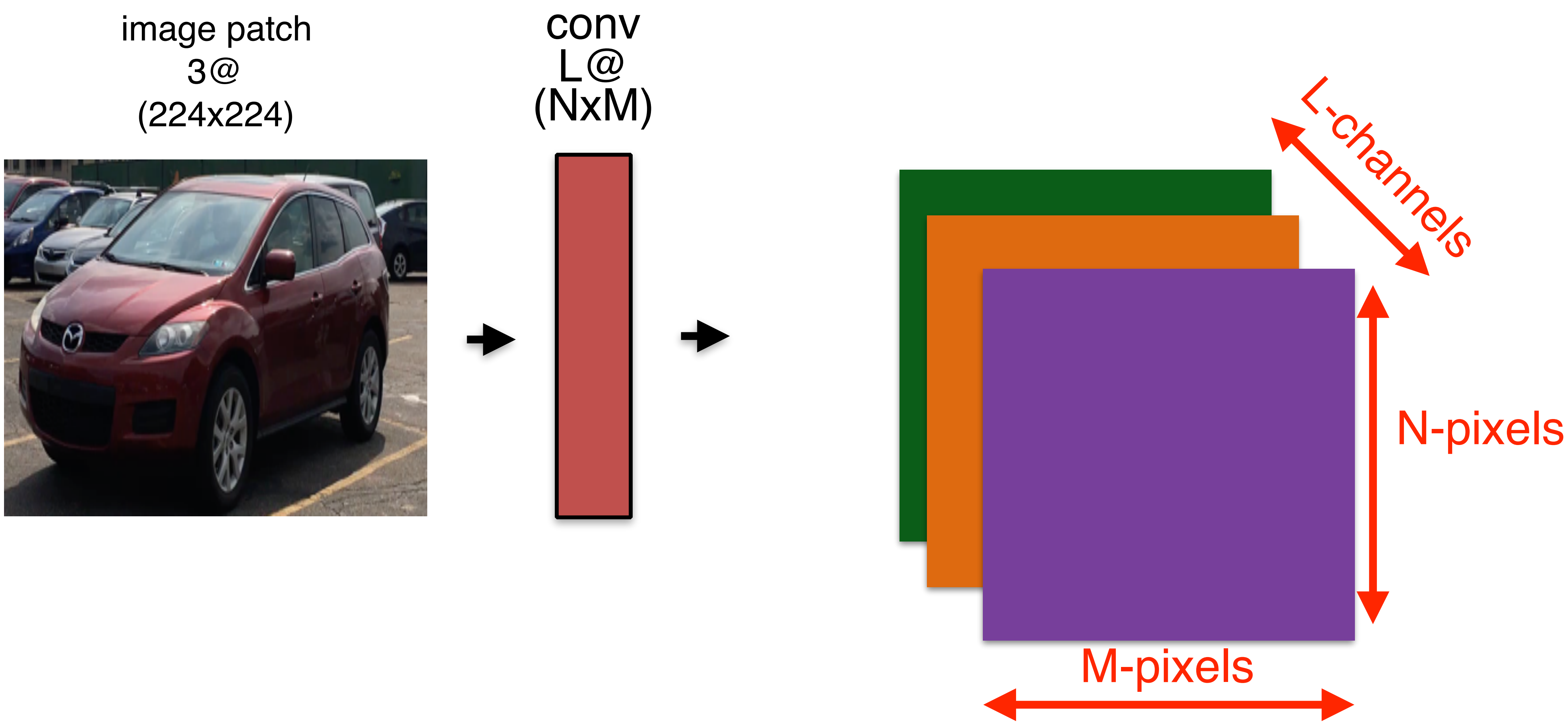
$$\mathbf{y}^{(l)} = \sum_{k=1}^{K} \mathbf{x}^{(k)} * \mathbf{h}^{(k,l)} \text{ for } l = 1 : L$$

# CNNs for Object Detection

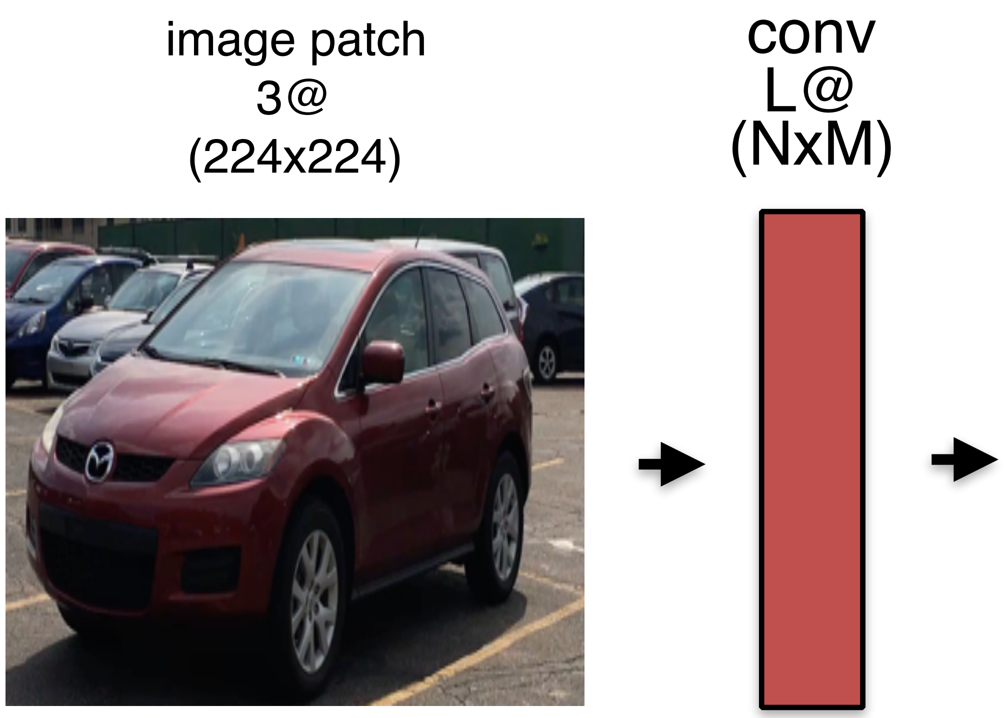
# CNNs for Object Detection

image patch
3@
(224x224)

# CNNs for Object Detection



# CNNs for Object Detection



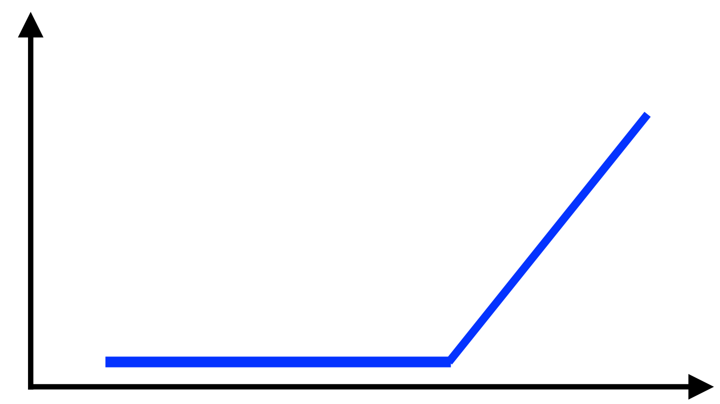

$$\eta \left\{ \sum_{k=1}^{K} \mathbf{x}^{(k)} * \mathbf{h}^{(k,l)} \right\}$$

$$\eta\{\} \rightarrow \text{non-linear function (relu, max pooling)}$$

# ReLU - Sparse and Positive

- Rectified Linear Unit

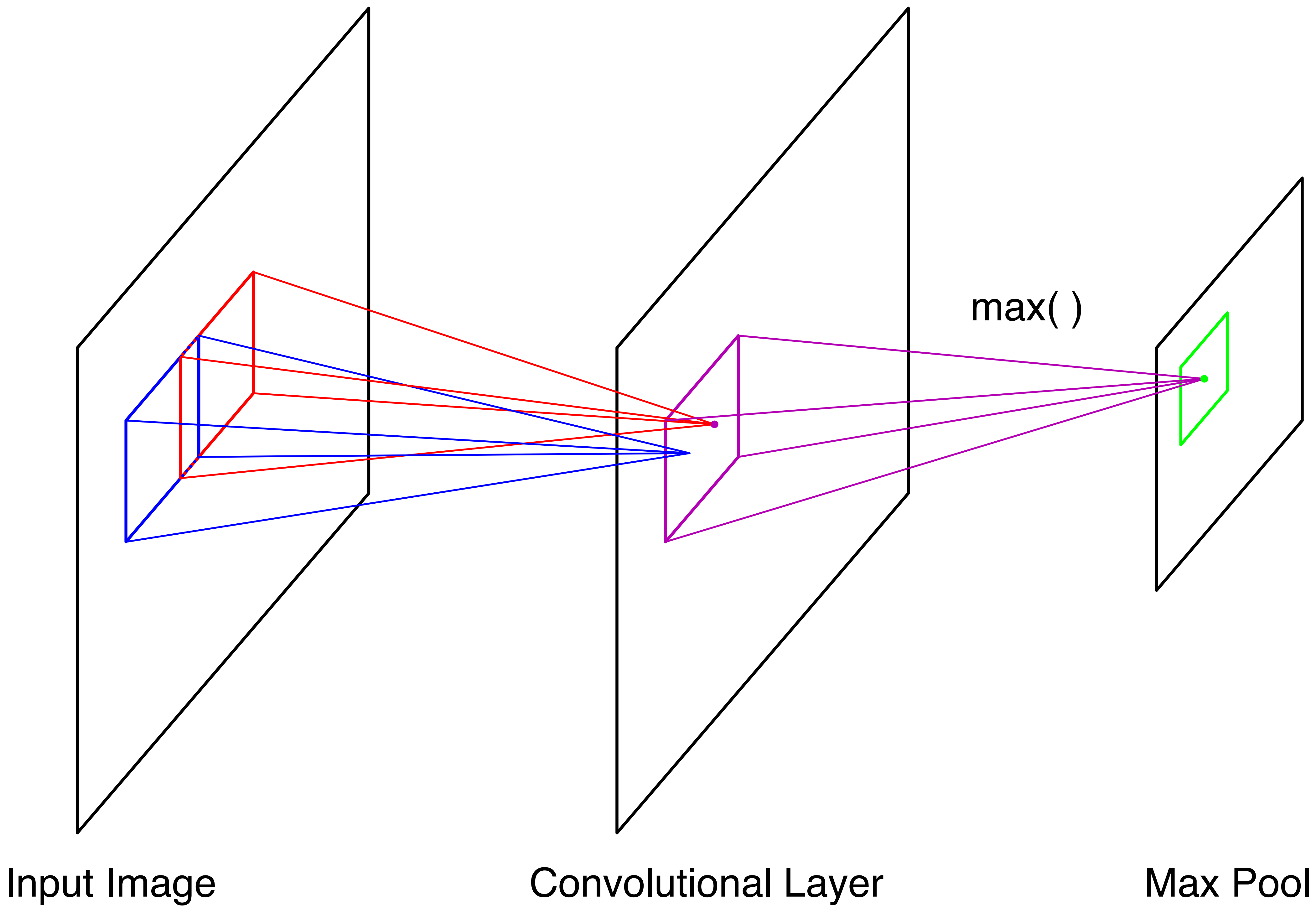
$$\mathrm{relu}\{x\} = \max(0, x)$$



- Connection to LASSO and sparsity??

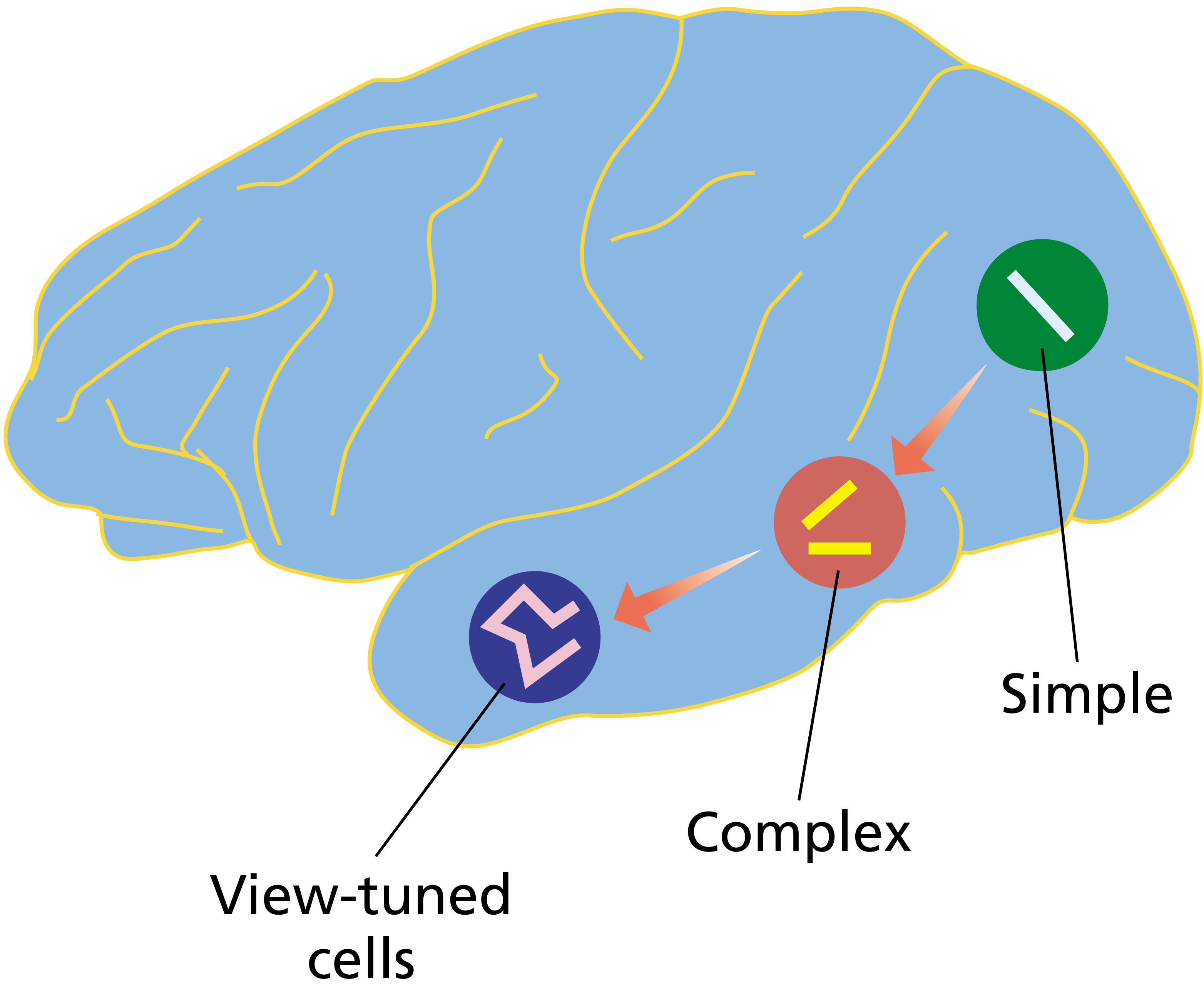
$$||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \frac{\lambda}{2}||\mathbf{x}||_1$$

# Max Pooling - Down Sampling

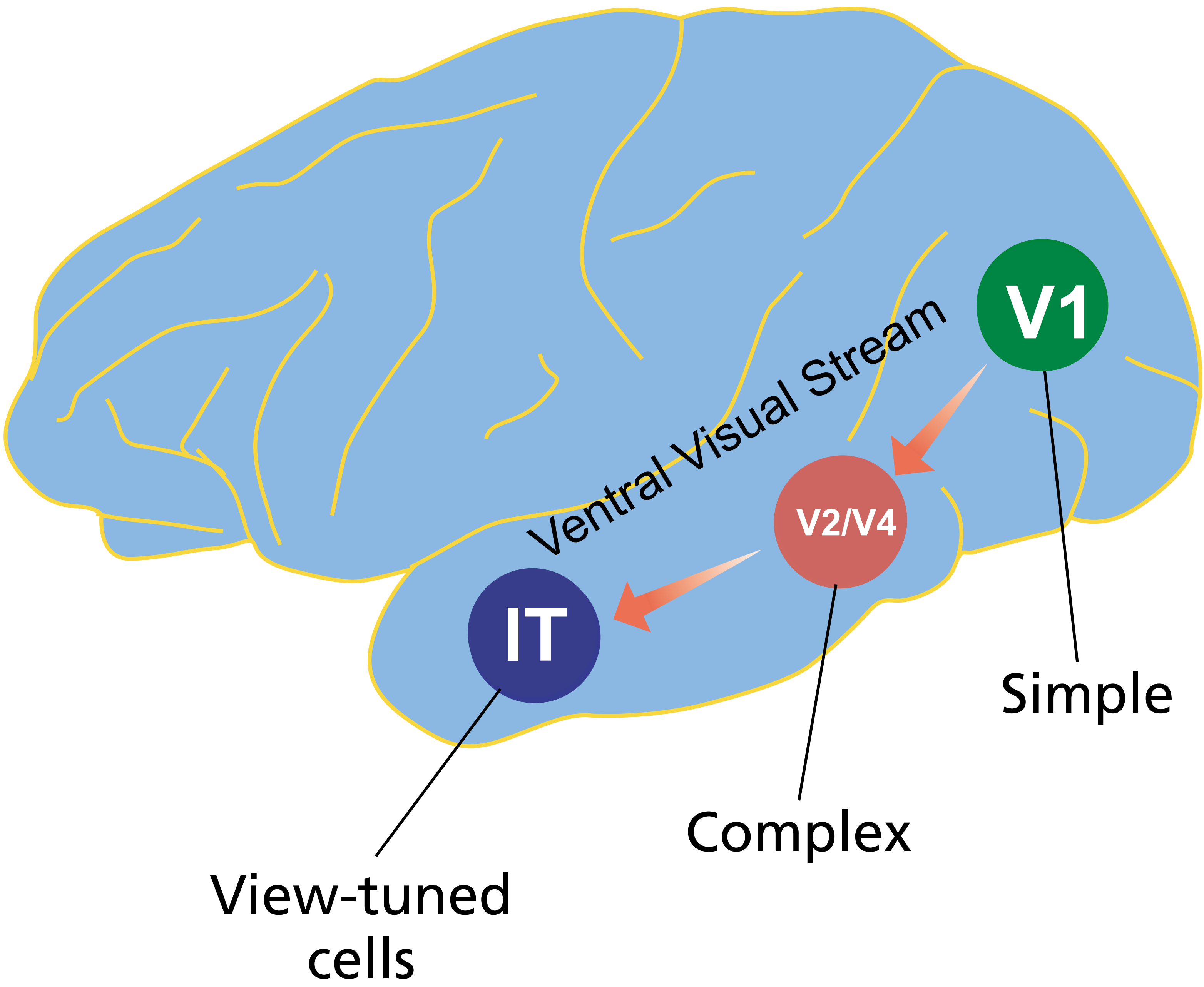



LeCun 1980

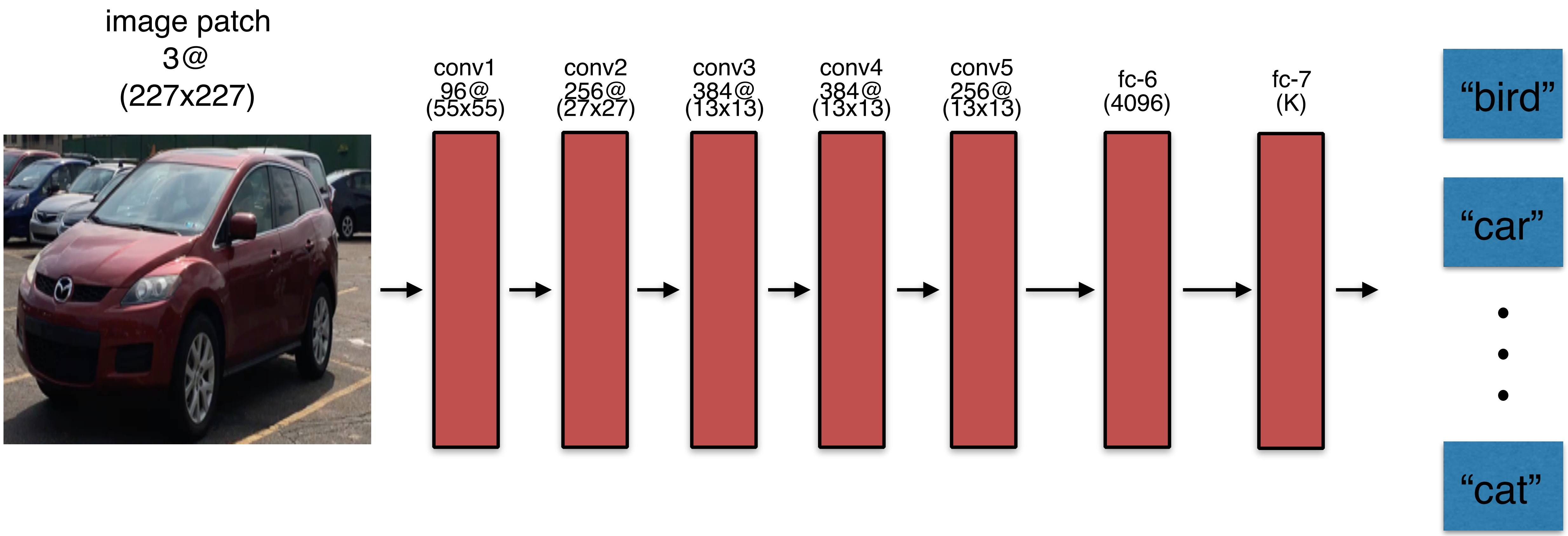
# Hierarchical Learning



# Hierarchical Learning

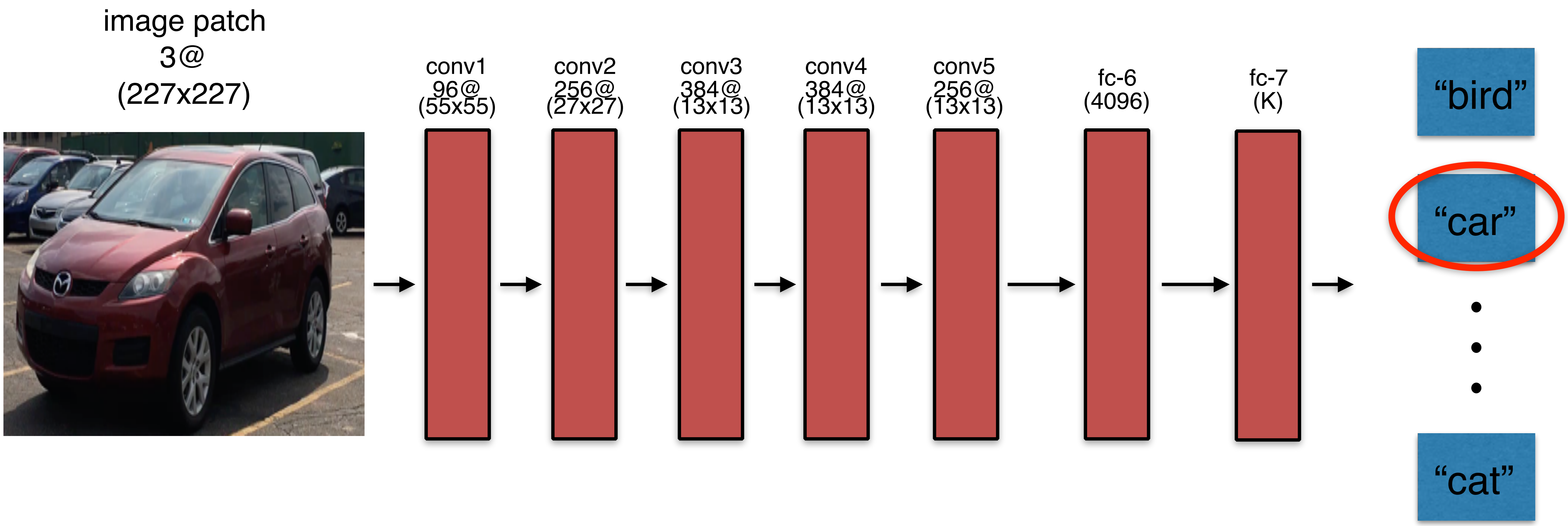


# Current State of the Art



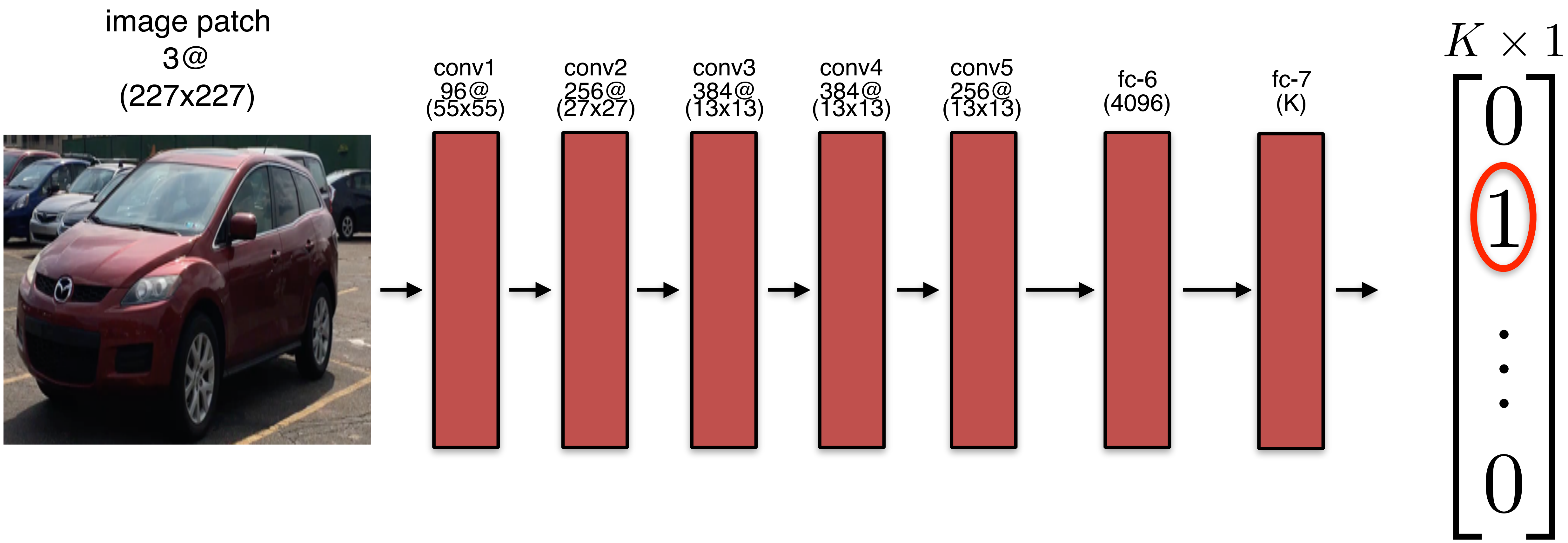

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

# Current State of the Art



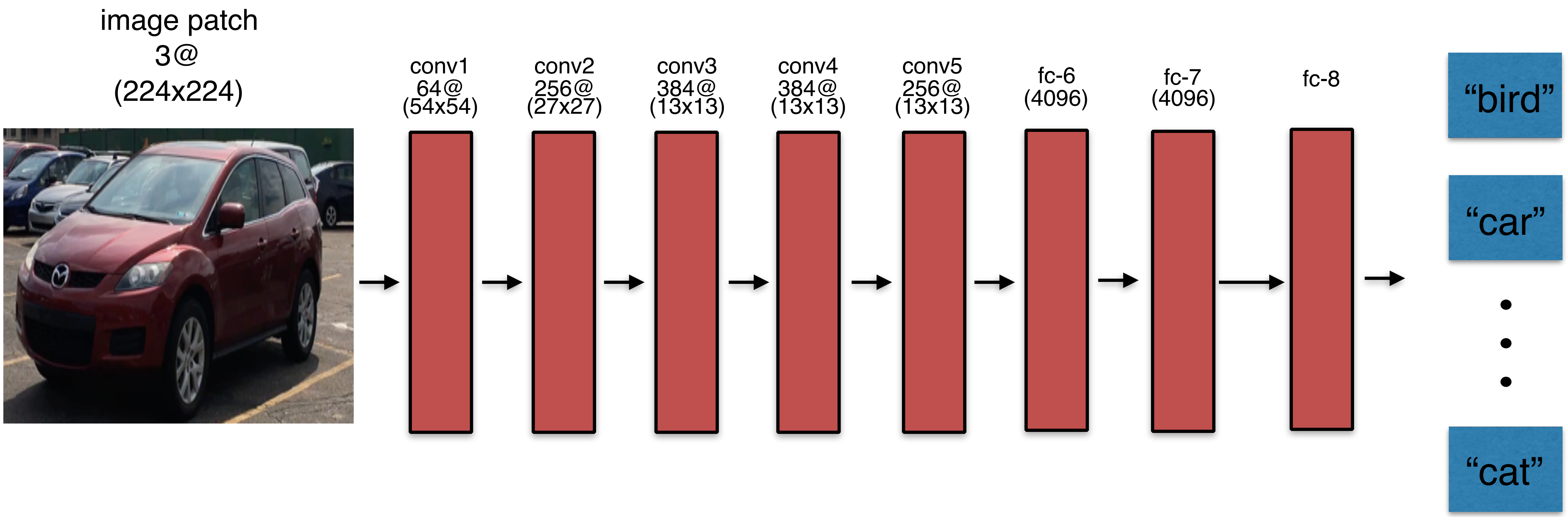

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

# Current State of the Art




A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

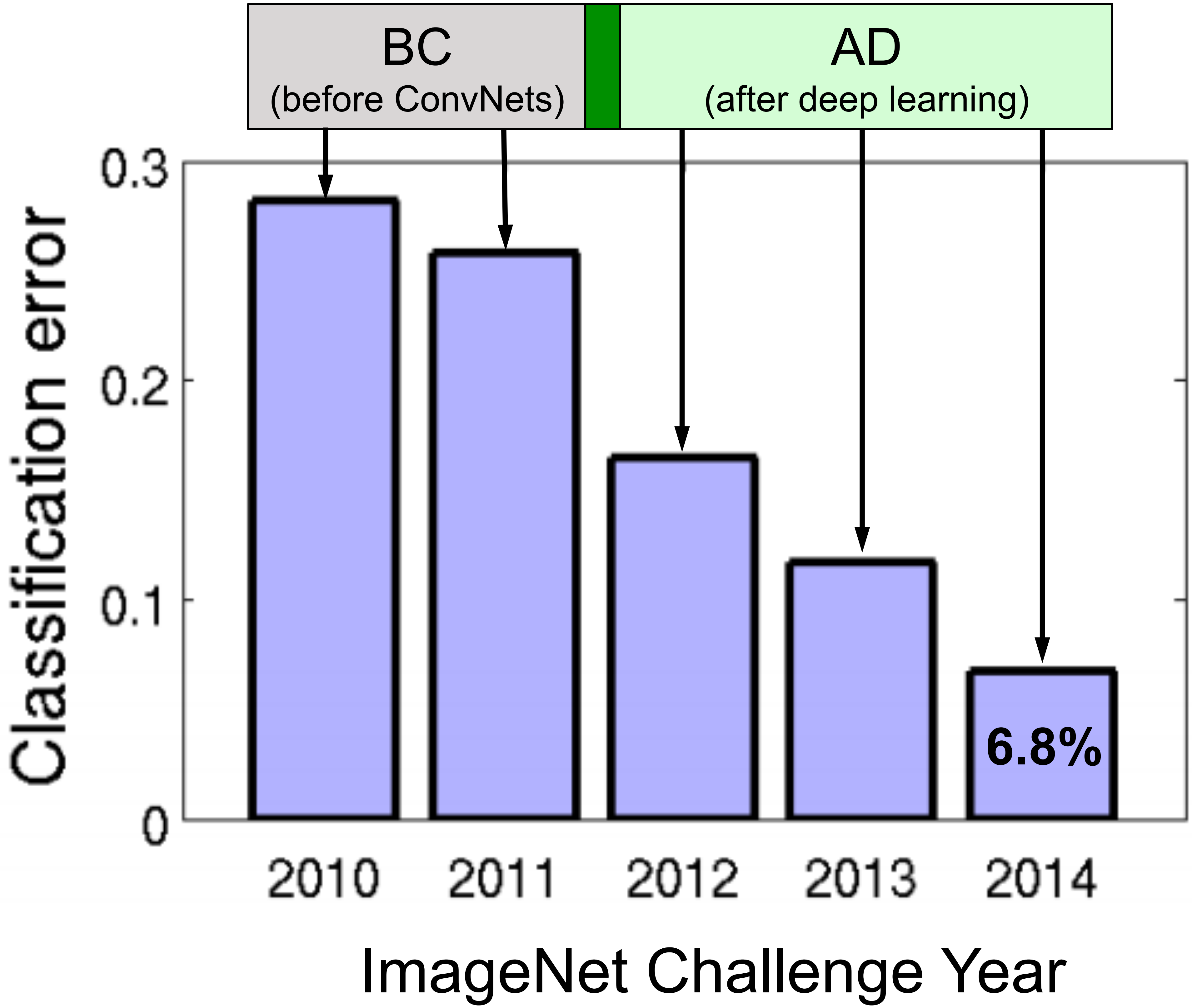
# Current State of the Art - Pose Selection



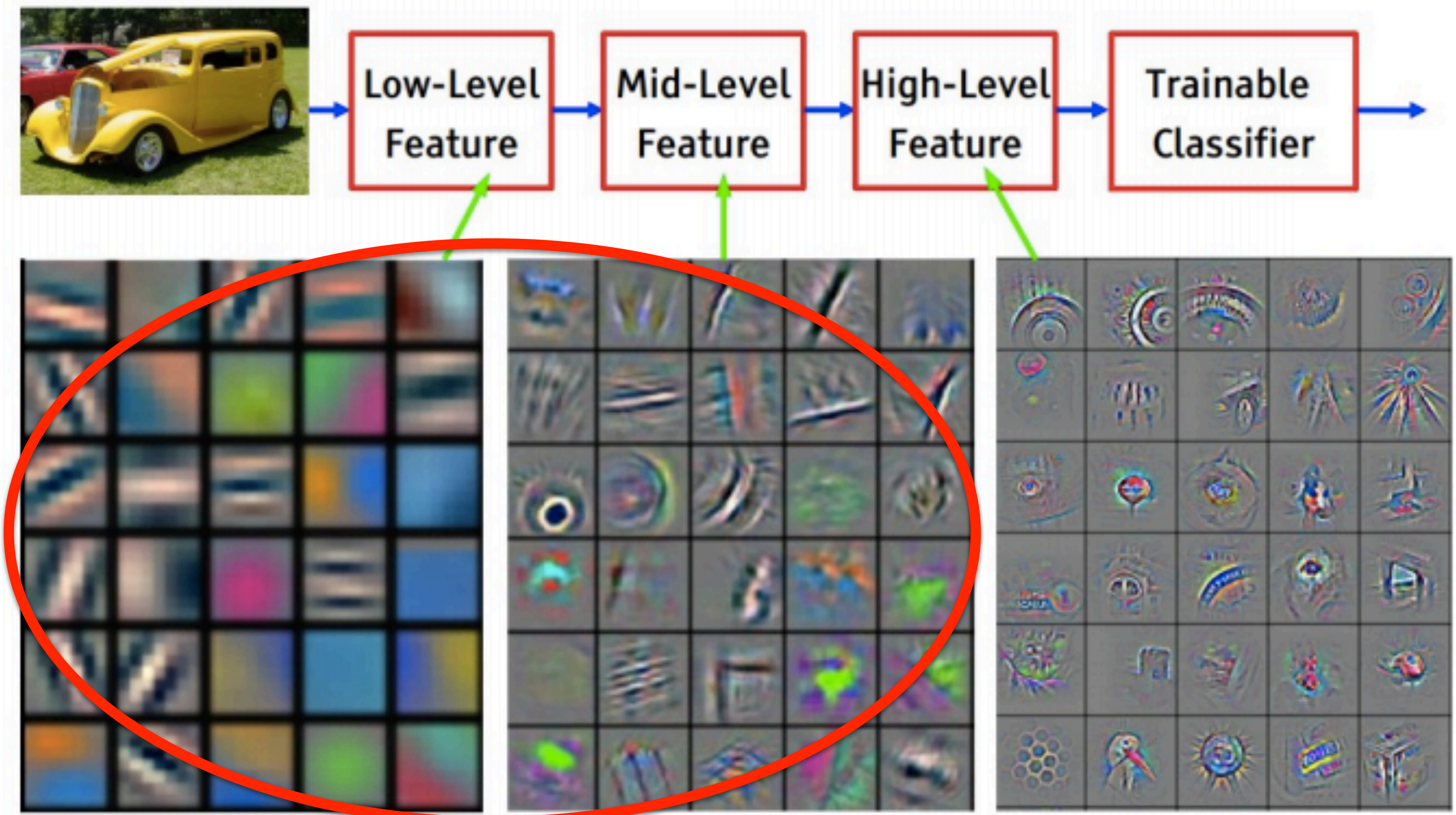

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

K. Chatfield, V. Lempitsky, A. Vedaldi and A. Zisserman. “Return of the Devil in the Details: Delving Deep into Convolutional Networks.” In BMVC, 2014.

# Impact on Object Recognition

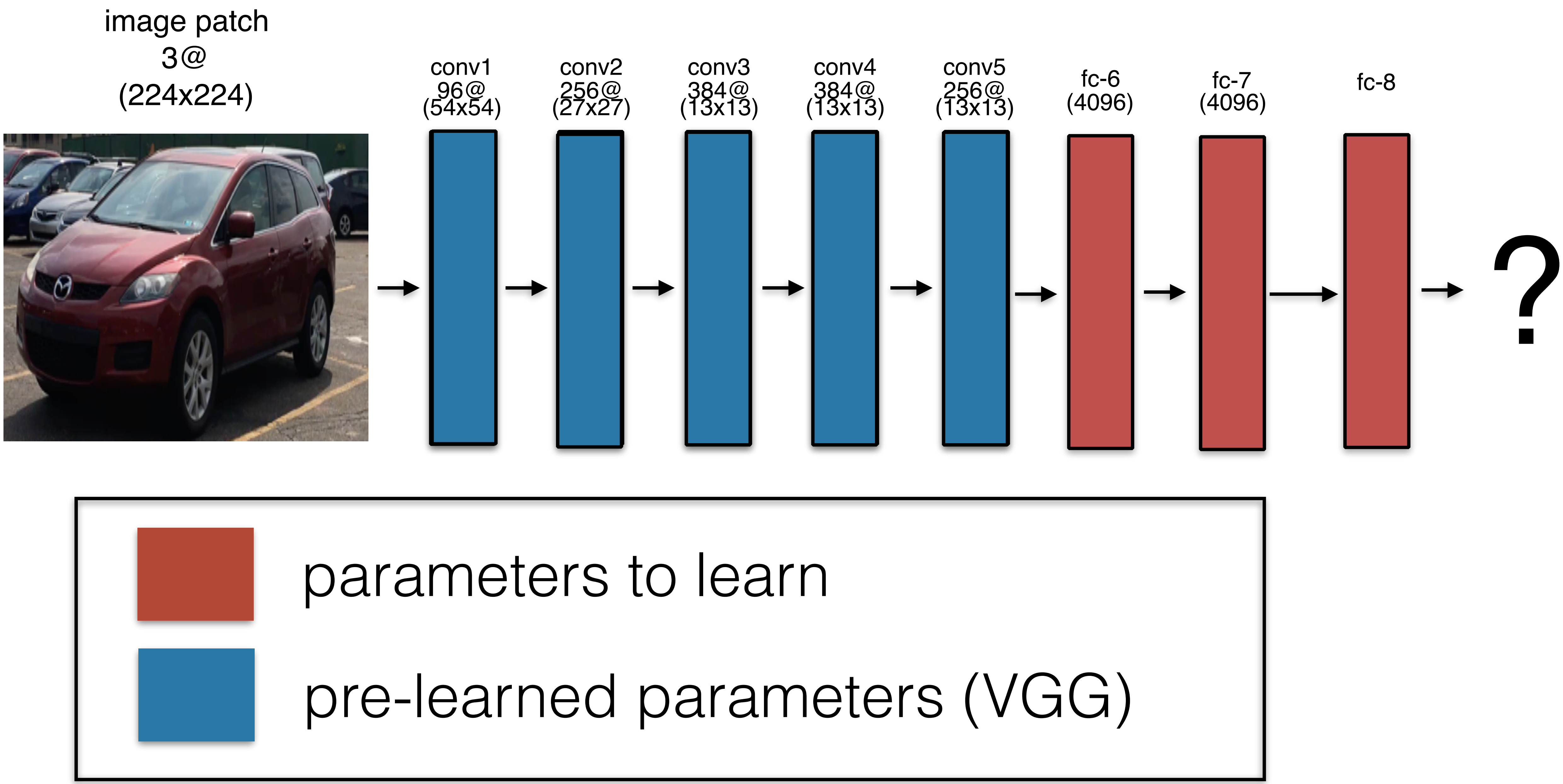


# Visualizing CNNs




Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

# CNNs as Feature Extraction





A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

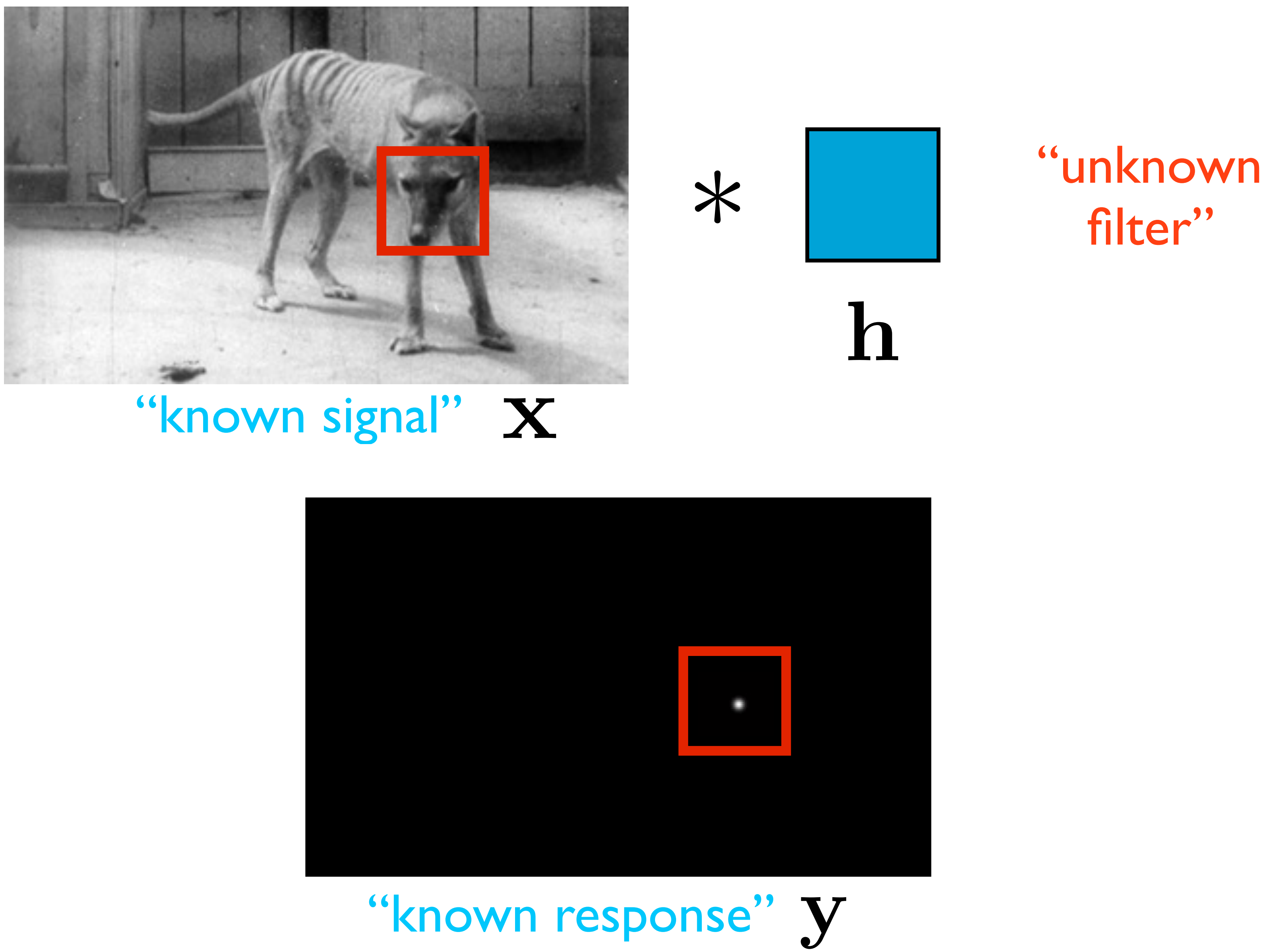
K. Chatfield, V. Lempitsky, A. Vedaldi and A. Zisserman. "Return of the Devil in the Details: Delving Deep into Convolutional Networks." In BMVC, 2014.

# Today

- Deep Features
- Deep Tracking
- Deep Flow

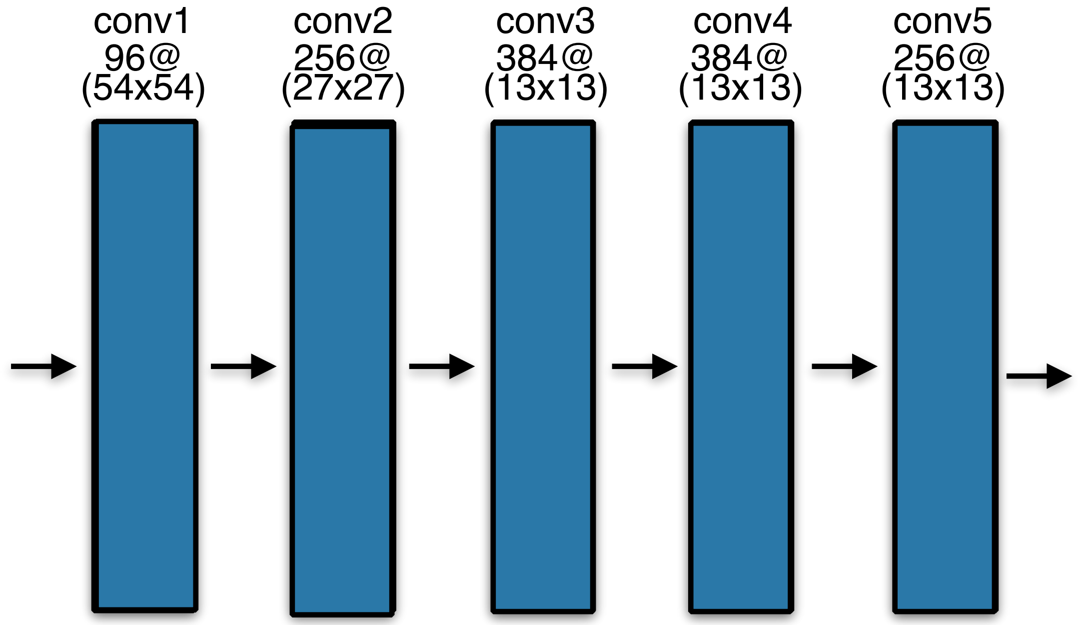
# Drawback to Conventional Methods

- Most methods for object tracking employ “online” learning.
- Online methods are expensive, have to make simplifying assumptions (e.g. circulant Toeplitz) to make things efficient.

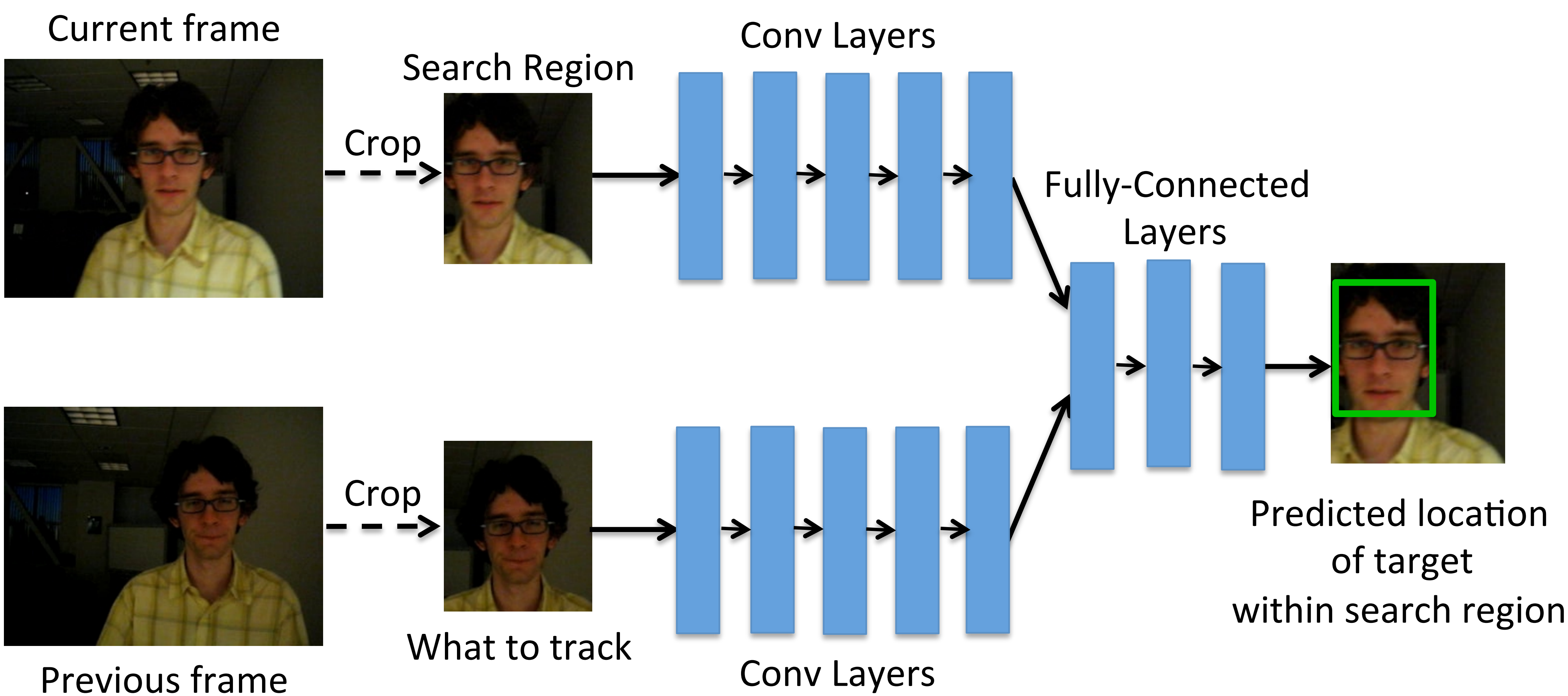


# Deep Tracking Methods

- Recently, there have been works that have tried to explore the employment of tracking using deep learning features.
- As efficiency is key, strategy is to learn from a large ensemble of labeled offline videos.
- Of particular interest are two papers,
  1. D. Held, S. Thrun, and S. Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.
  2. L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr “Fully-Convolutional Siamese Networks for Object Tracking”, ArXiv 2016.

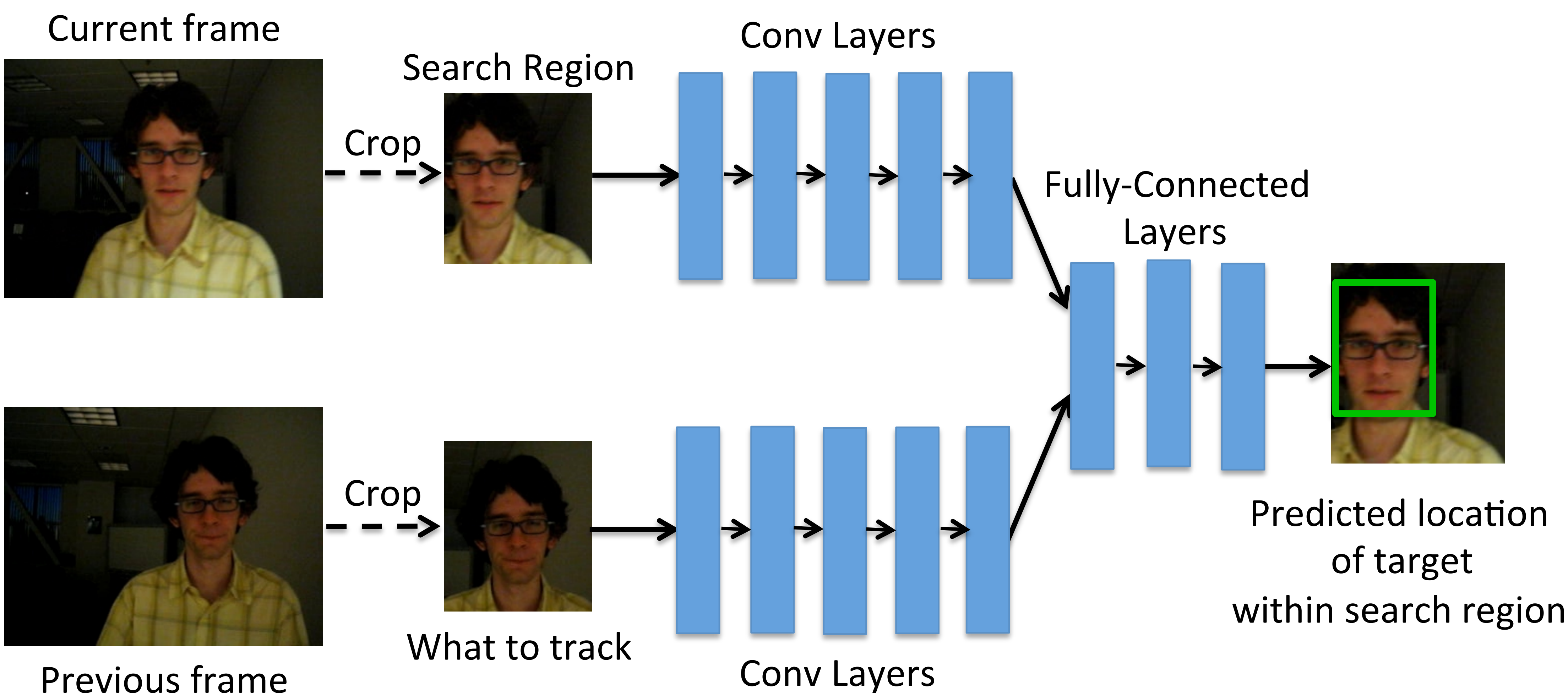


# Deep Regression Networks




D. Held, S. Thrun, and S. Savarese "Learning to Track at 100 FPS with Deep Regression Networks", ECCV 2016.

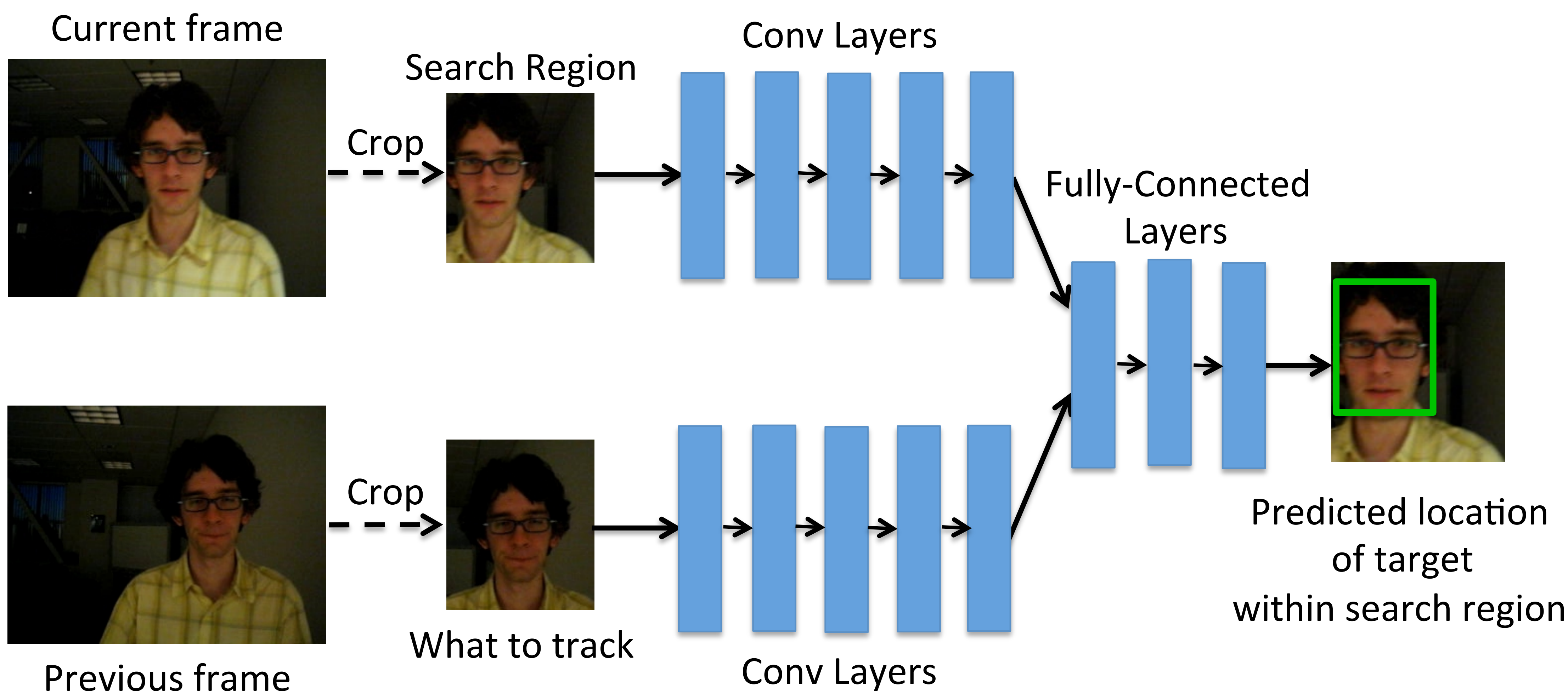
# Deep Regression Networks




**What does this method remind you of?**

D. Held, S. Thrun, and S. Savarese "Learning to Track at 100 FPS with Deep Regression Networks", ECCV 2016.

# Deep Regression Networks



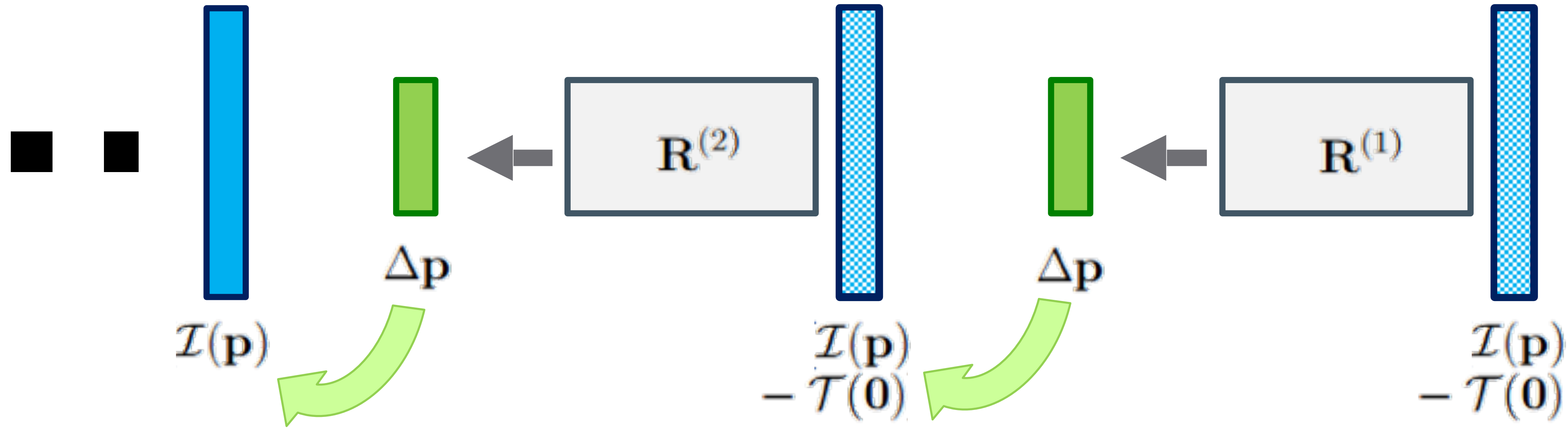

**What does this method remind you of?**

**Why is it fast?**

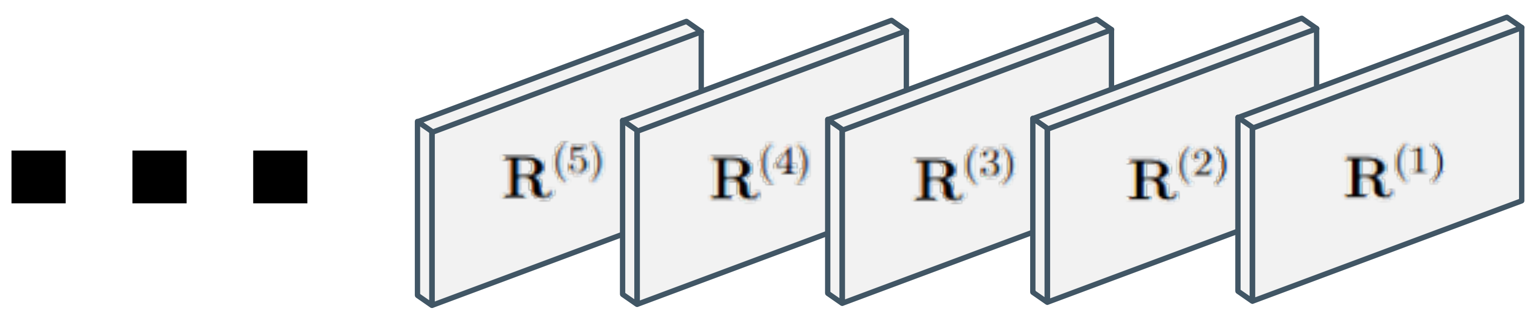
D. Held, S. Thrun, and S. Savarese "Learning to Track at 100 FPS with Deep Regression Networks", ECCV 2016.

# Supervised Descent Method (SDM)

- SDM assumes a linear relationship between appearance and geometry: $\Delta \mathbf{p} = \mathbf{R}[\mathcal{I}(\mathbf{p}) - \mathcal{T}(\mathbf{0})]$
- Iteratively updates until convergence



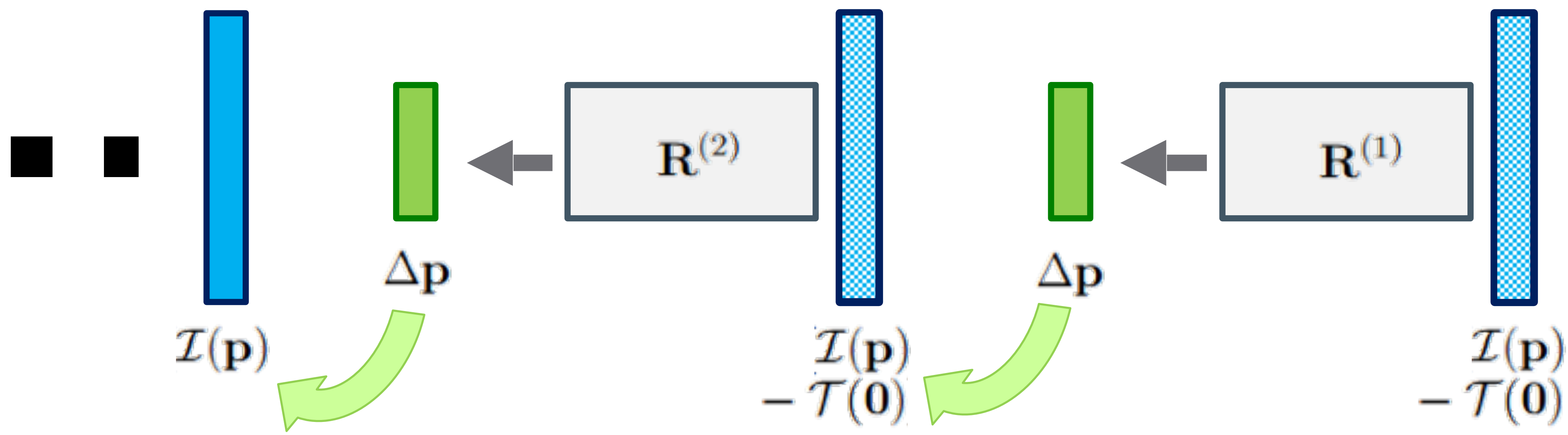

- However, $\mathbf{R}^{(k)}$ is iteration specific.



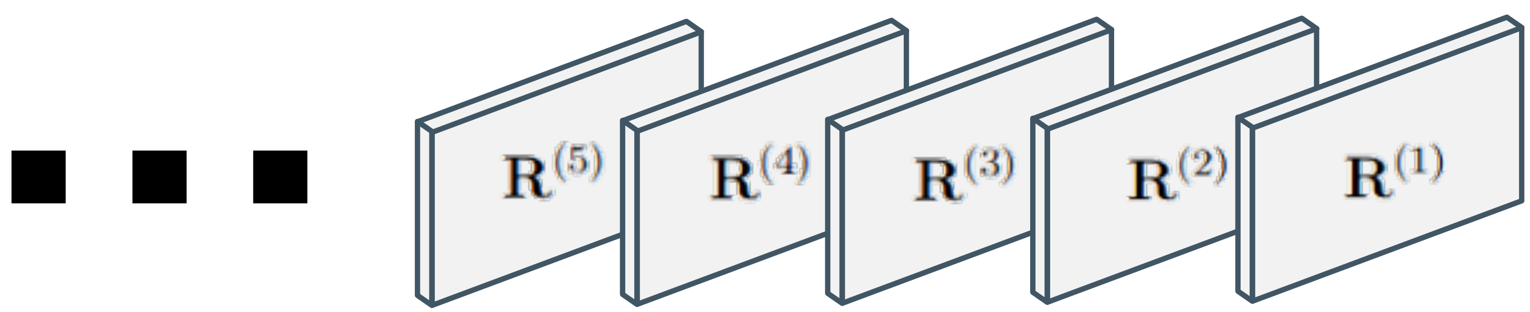

Xiong & De la Torre 2013

# Supervised Descent Method (SDM)

- SDM assumes a linear relationship between appearance and geometry: $$\Delta \mathbf{p} = \mathbf{R}[\mathcal{I}(\mathbf{p}) - \mathcal{T}(\mathbf{0})]$$
- Iteratively updates until convergence




- However, $\mathbf{R}^{(k)}$ is iteration specific.




**What is a potential issue here?**

Xiong & De la Torre 2013

# Deep Regression Networks

Previous video frame centered on object

Current video frame, shifted, with ground-truth bounding box

D. Held, S. Thrun, and S. Savarese "Learning to Track at 100 FPS with Deep Regression Networks", ECCV 2016.

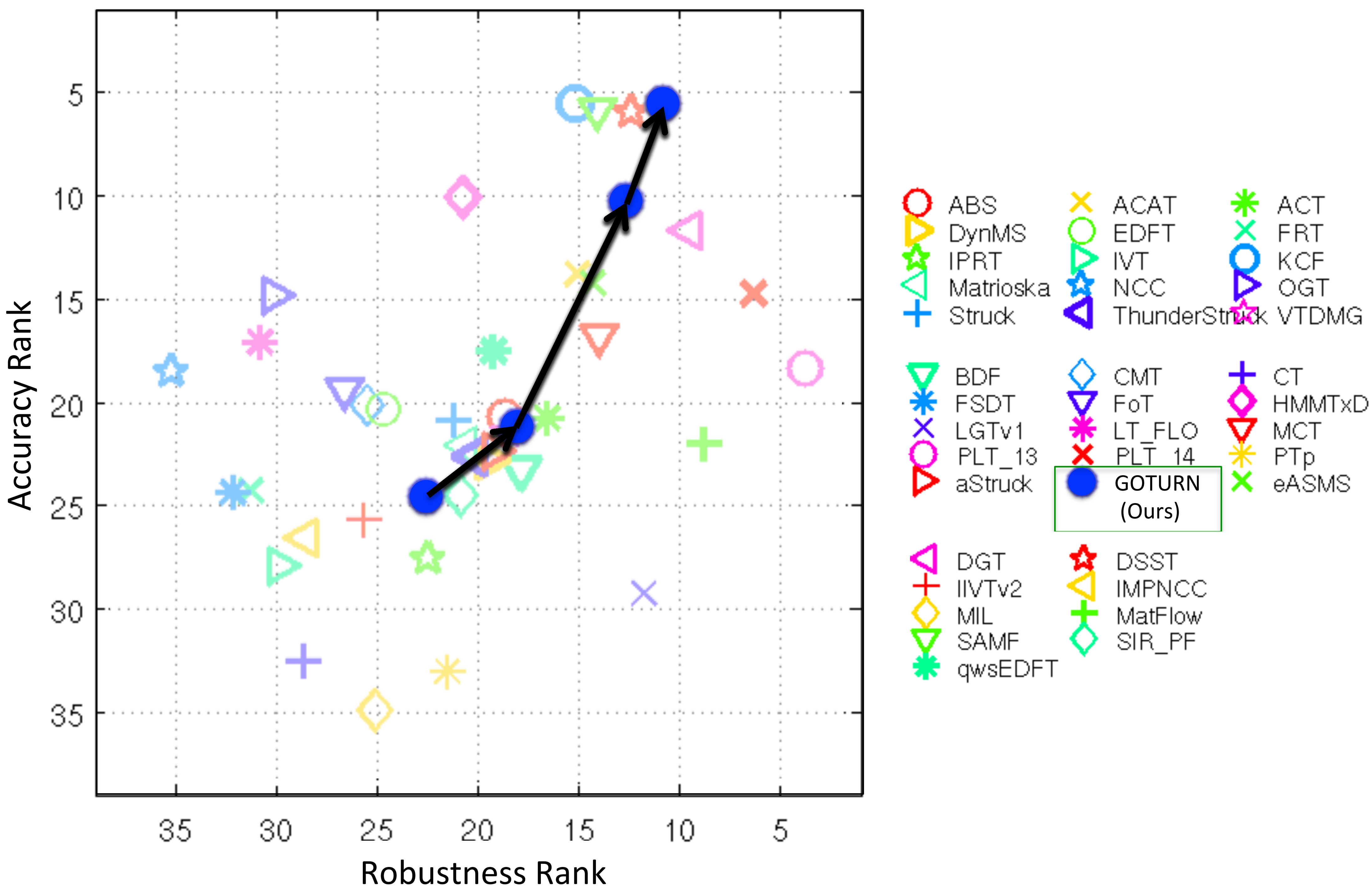
# Deep Regression Networks

Image
centered on
object

Shifted image
with ground-truth
bounding box

D. Held, S. Thrun, and S. Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.

# Deep Regression Networks




D. Held, S. Thrun, and S. Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.

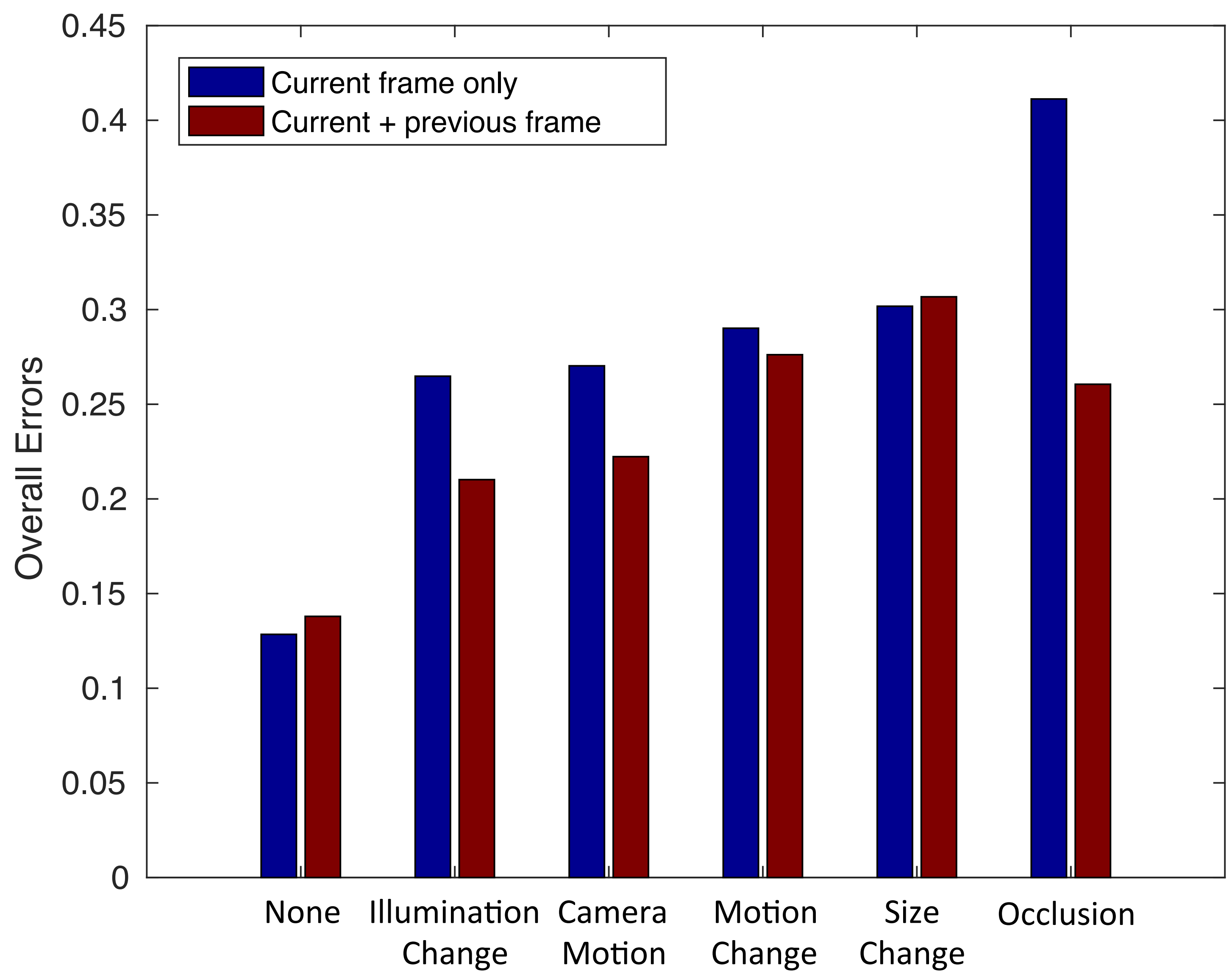
# How does it work?

- Two hypotheses,
  1. The network compares the previous frame to the current frame to find the target object in the current frame.
  2. The network acts as a local generic "object detector" and simply locates the nearest "object."

D. Held, S. Thrun, and S. Savarese "Learning to Track at 100 FPS with Deep Regression Networks", ECCV 2016.

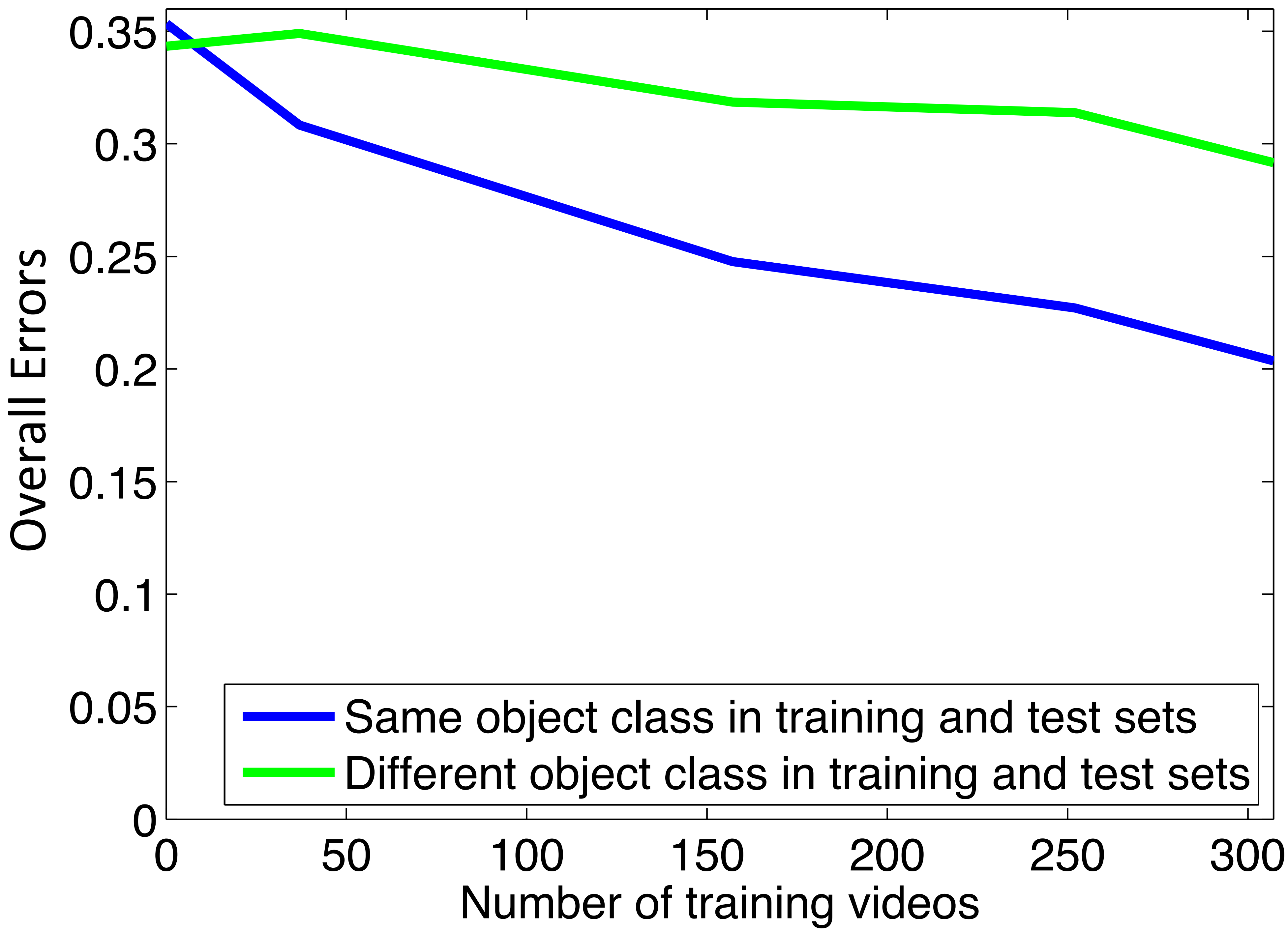
# How does it work?

- Two hypotheses,
  1. The network compares the previous frame to the current frame to find the target object in the current frame.
  2. The network acts as a local generic "object detector" and simply locates the nearest "object."



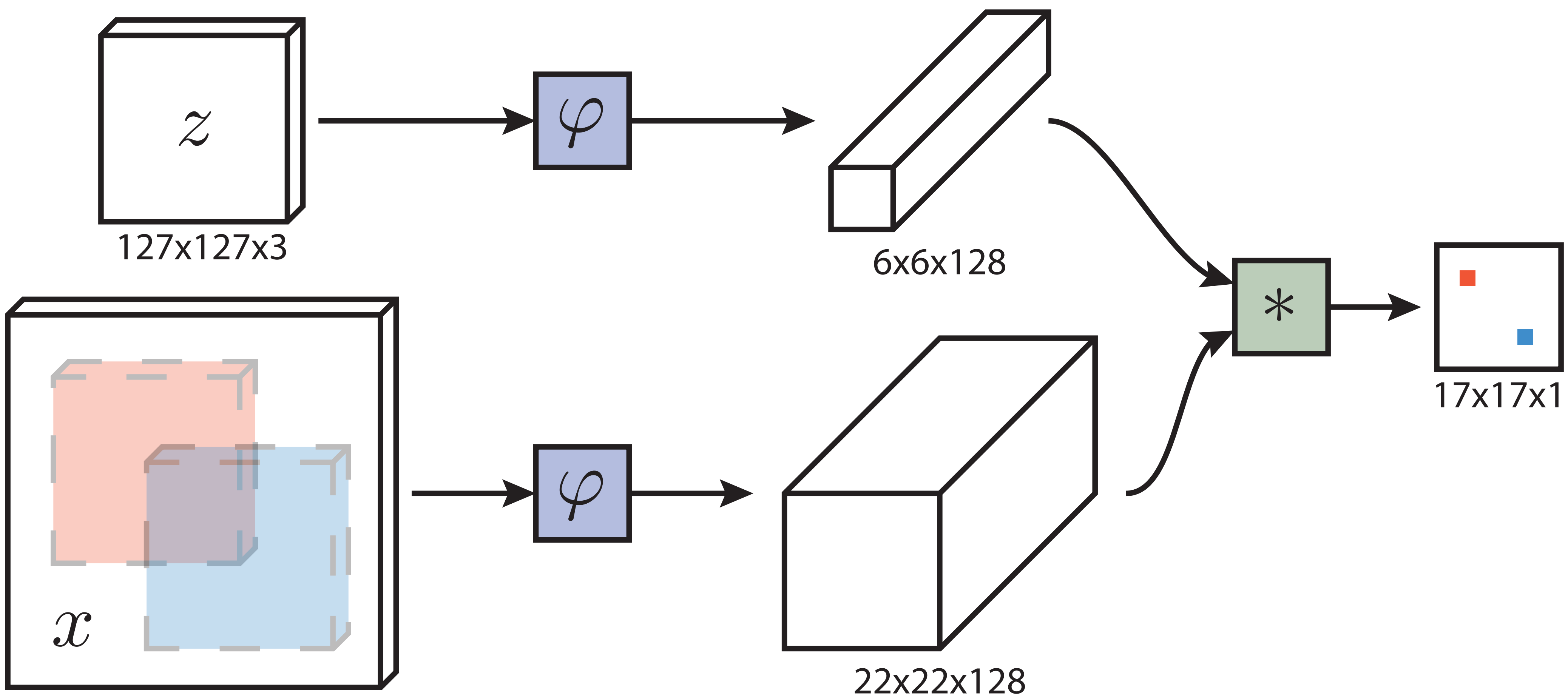

D. Held, S. Thrun, and S. Savarese "Learning to Track at 100 FPS with Deep Regression Networks", ECCV 2016.

# Generality vs. Specificity




D. Held, S. Thrun, and S. Savarese "Learning to Track at 100 FPS with Deep Regression Networks", ECCV 2016.

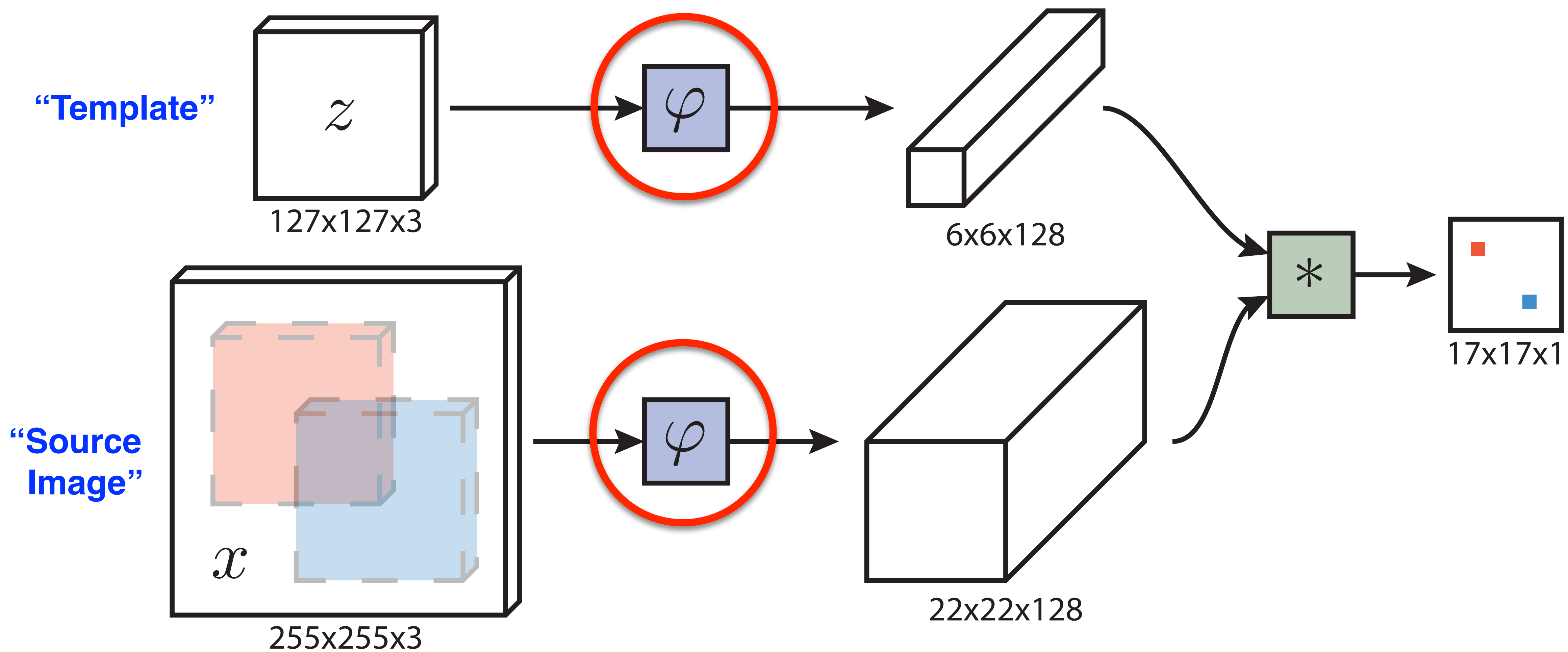
# Fully Convolutional Siamese Networks




L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr "Fully-Convolutional Siamese Networks for Object Tracking", ArXiv 2016.

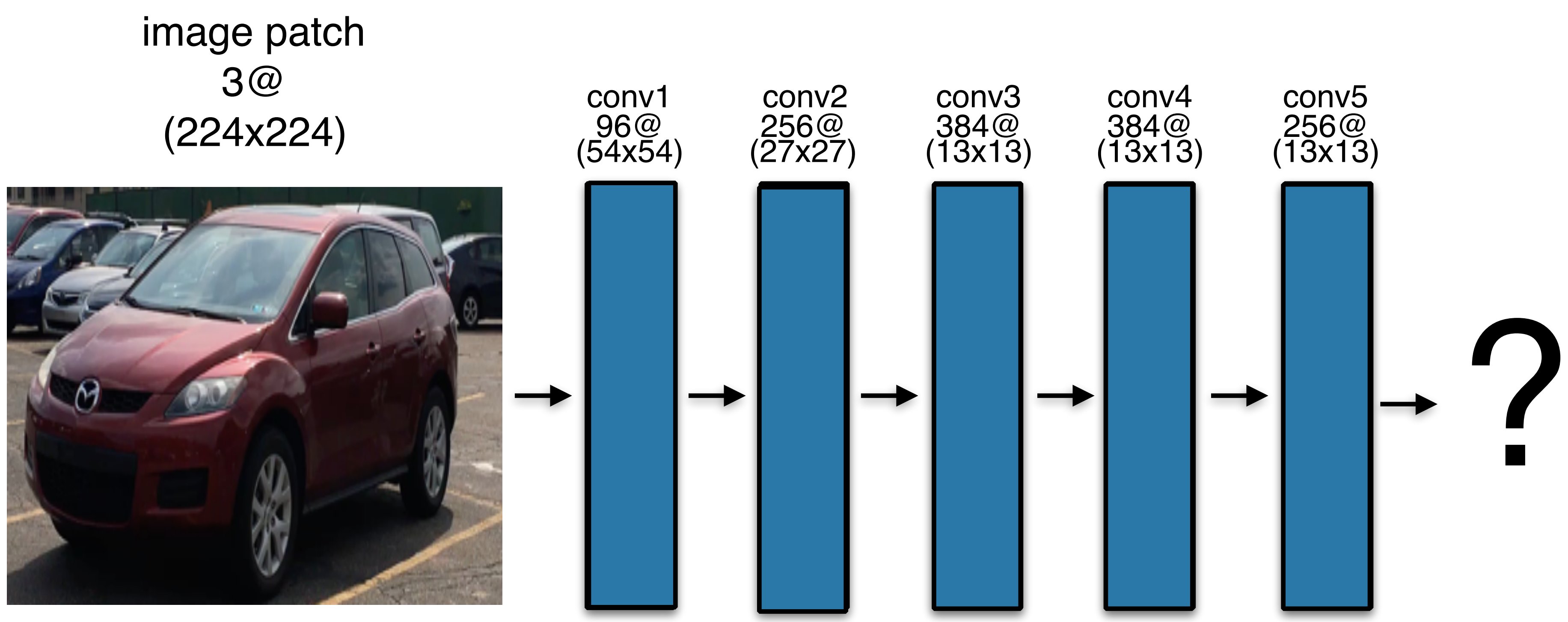
# Fully Convolutional Siamese Networks

**"Siamese" as they apply an identical transformation to both inputs**



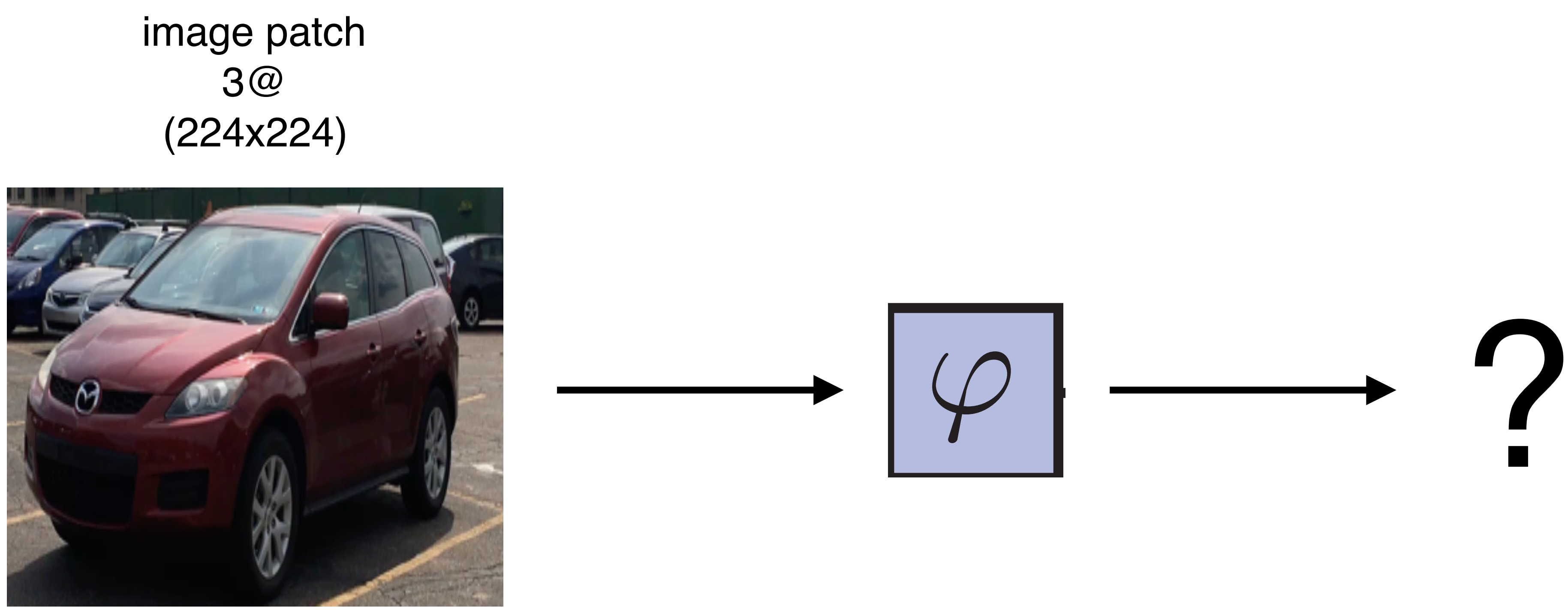

L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr "Fully-Convolutional Siamese Networks for Object Tracking", ArXiv 2016.

# Fully Convolutional




L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr "Fully-Convolutional Siamese Networks for Object Tracking", ArXiv 2016.

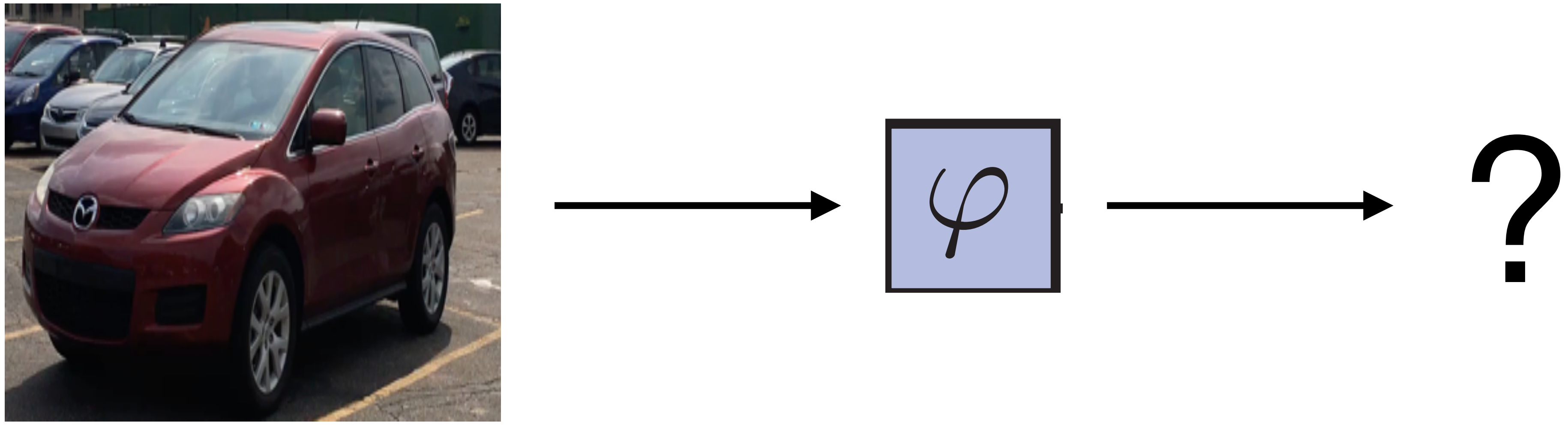
# Fully Convolutional




L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr "Fully-Convolutional Siamese Networks for Object Tracking", ArXiv 2016.

# Fully Convolutional

image patch
3@
(224x224)




$$\varphi\{\mathbf{e}_i * \mathbf{x}\} = \mathbf{e}_i * \varphi\{\mathbf{x}\}$$

$$\mathbf{e}_i = [0, \ldots, \underset{i\text{-th index}}{1}, \ldots, 0]^T$$

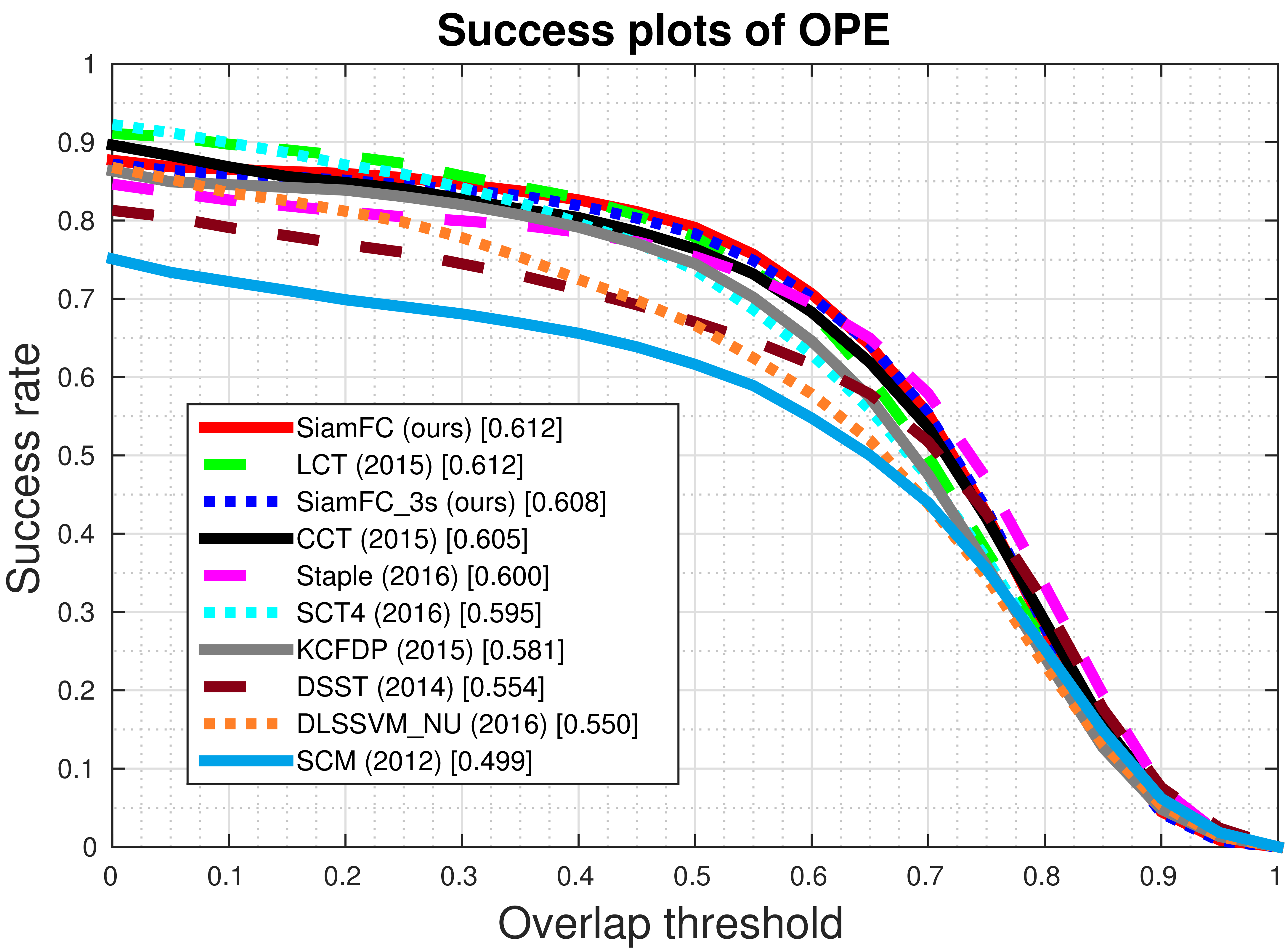
L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr "Fully-Convolutional Siamese Networks for Object Tracking", ArXiv 2016.

# Fully Convolutional Siamese Networks

- Example training sequences.

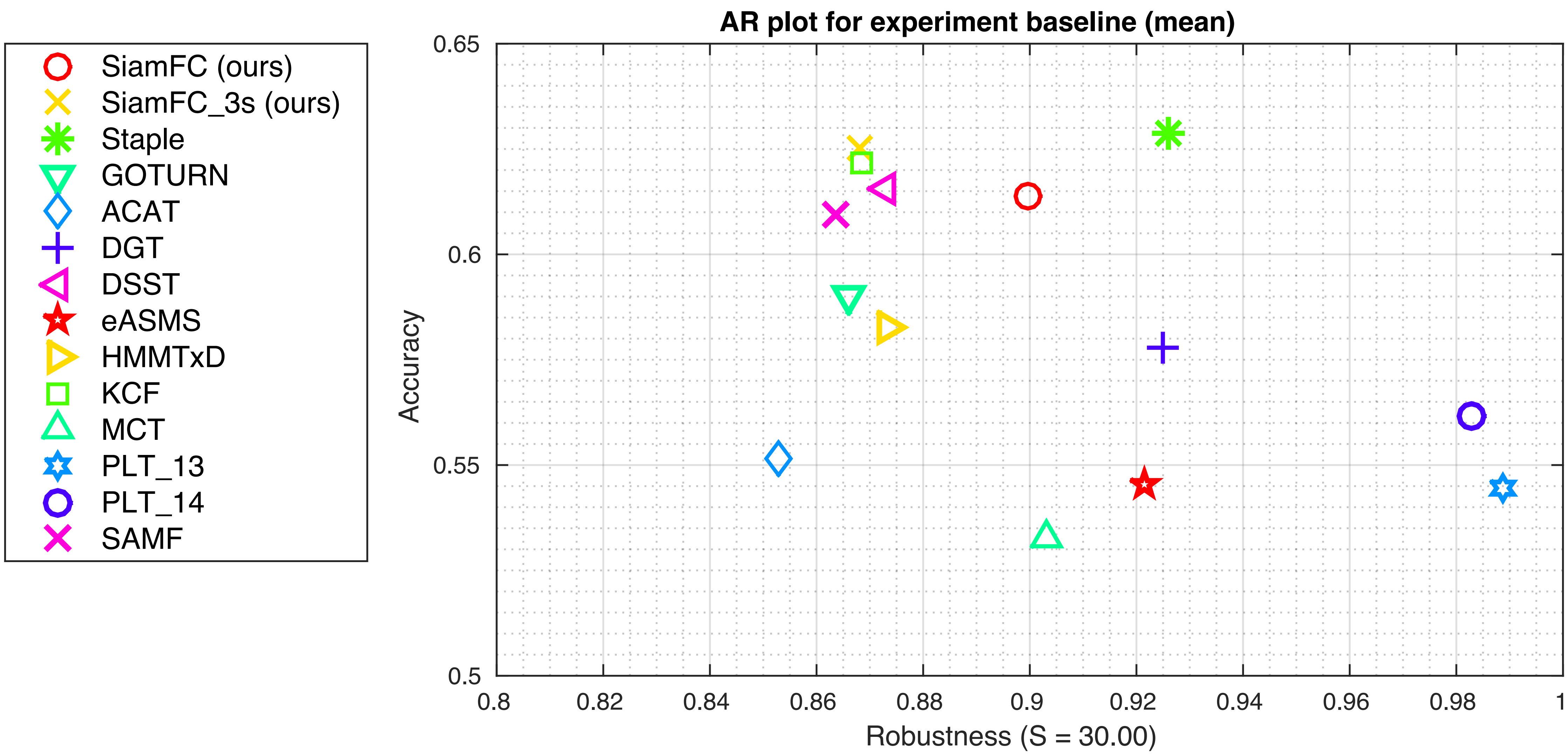
L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr "Fully-Convolutional Siamese Networks for Object Tracking", ArXiv 2016.

# Fully Convolutional Siamese Networks



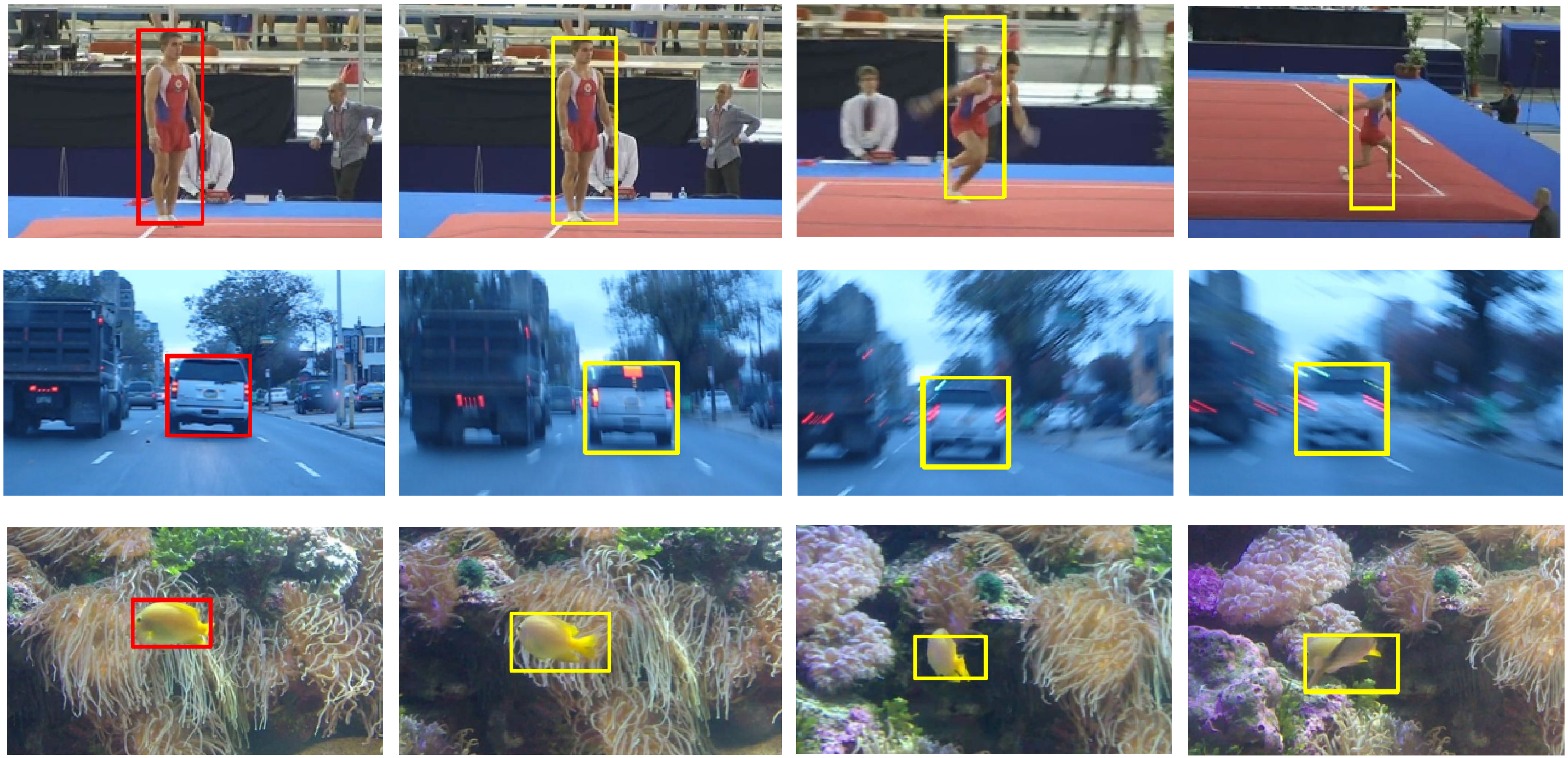

L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr “Fully-Convolutional Siamese Networks for Object Tracking”, ArXiv 2016.

# Fully Convolutional Siamese Networks




L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr "Fully-Convolutional Siamese Networks for Object Tracking", ArXiv 2016.

# Fully Convolutional Siamese Networks



Frame 1 (init.) Frame 50 Frame 100 Frame 200

L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr "Fully-Convolutional Siamese Networks for Object Tracking", ArXiv 2016.

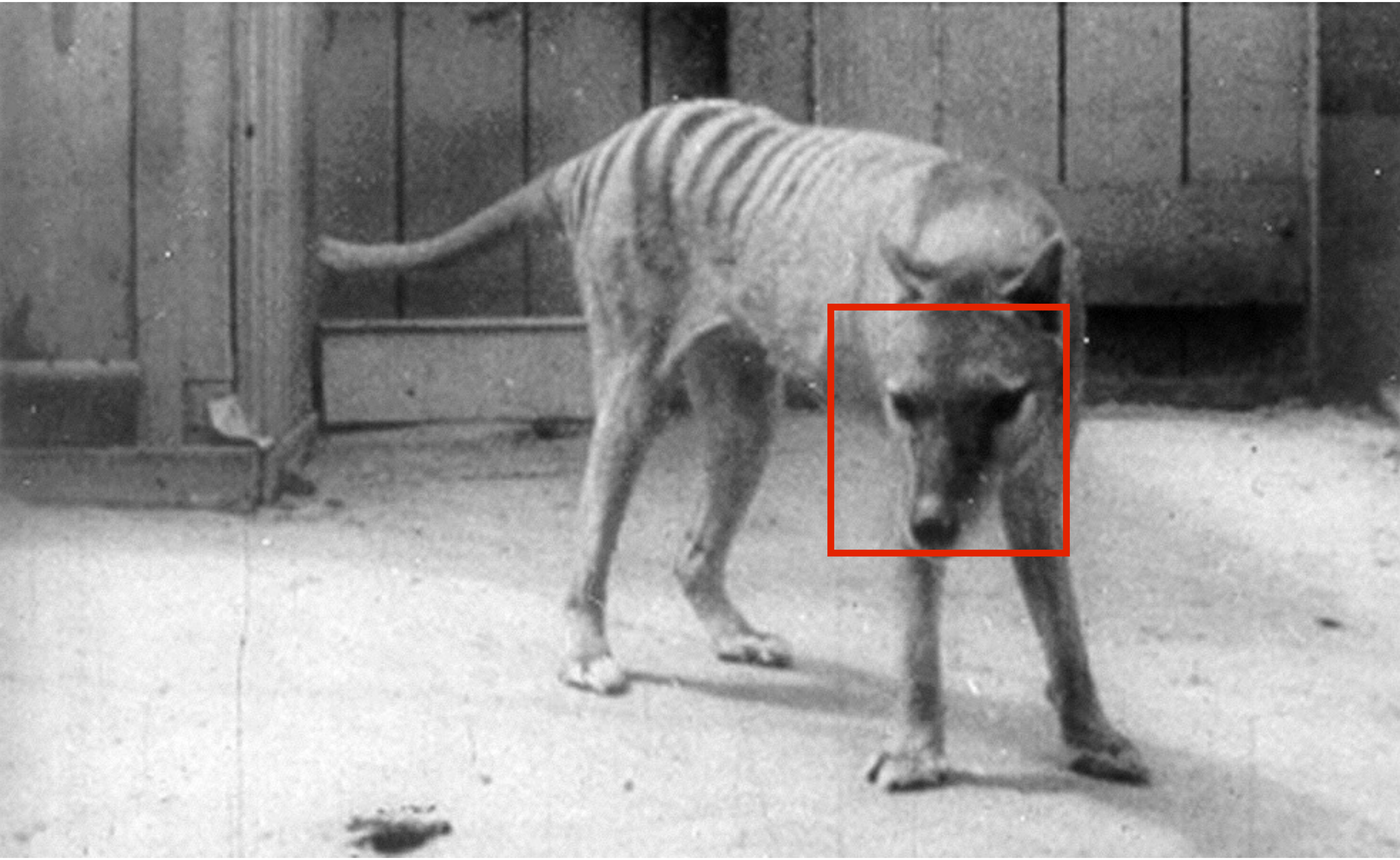
# Fully Convolutional Siamese Networks

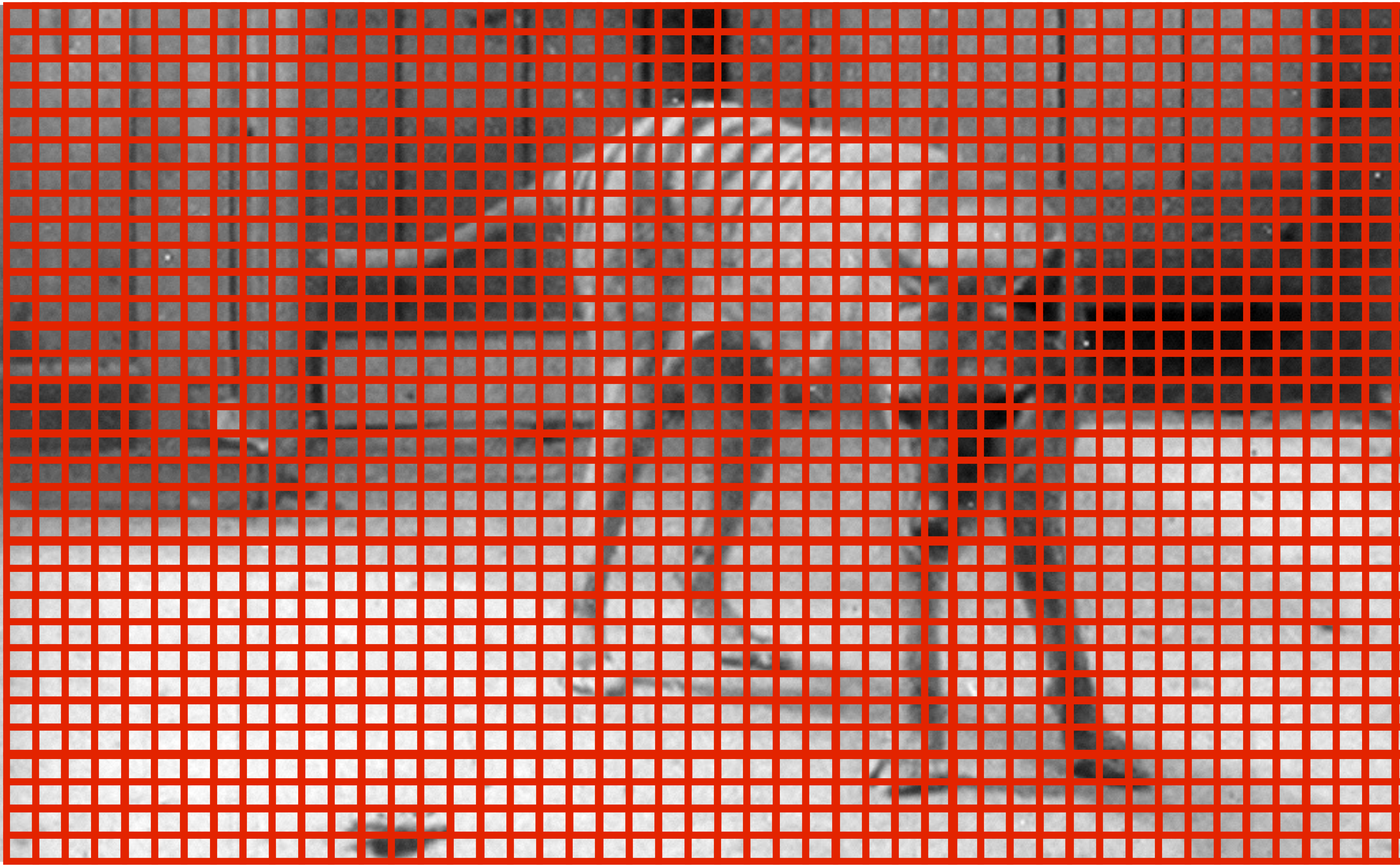

Frame 1 (init.) Frame 50 Frame 100 Frame 200

L. Bertinetto J. Valmadre J. F. Henriques, A. Vedaldi, P. H. S. Torr "Fully-Convolutional Siamese Networks for Object Tracking", ArXiv 2016.

# Today

- Deep Features
- Deep Tracking
- Deep Flow





# Flow = Parts Based Registration

$$\min_{\mathbf{x}} \sum_{i=1}^{N} D_i(\mathbf{x}_i) + \lambda \; R(\mathbf{x})$$

# Flow = Parts Based Registration

$\boxed{\text{Does the image at } \mathbf{x}_i \text{ look like the } i^{\mathrm{th}} \text{ part?}}$

$$\min_{\mathbf{x}} \sum_{i=1}^{N} D_i(\mathbf{x}_i) + \lambda \, R(\mathbf{x})$$

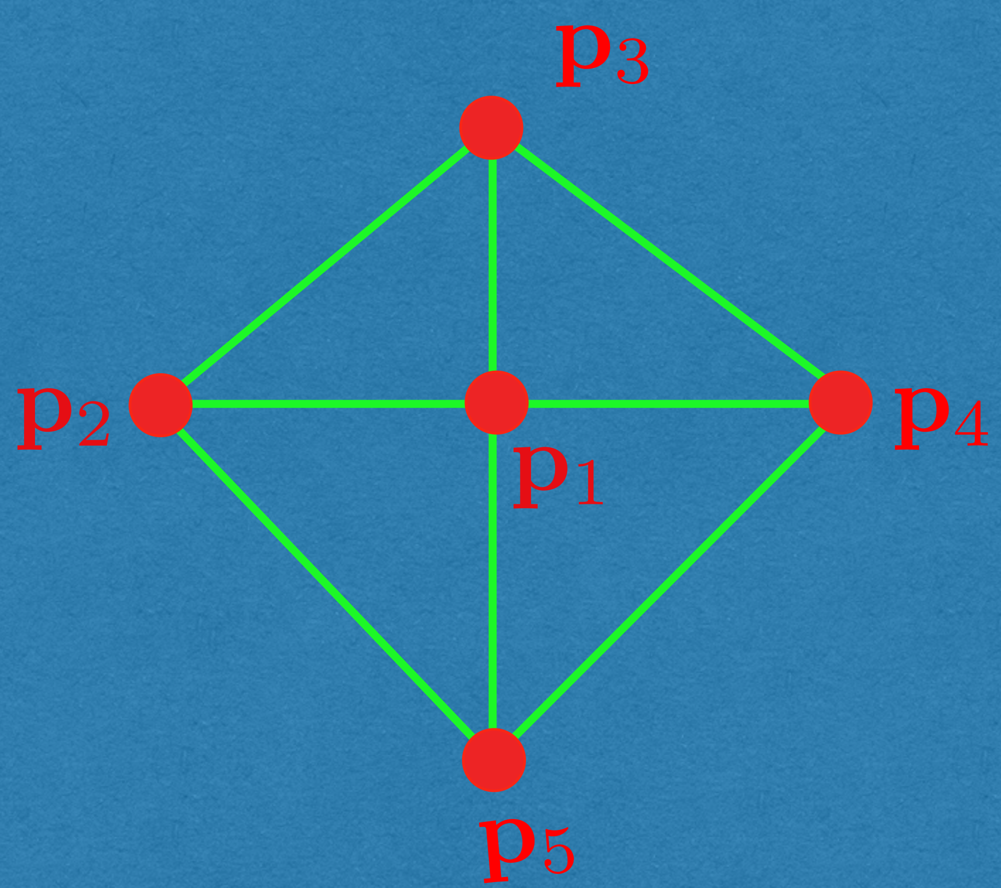
# Flow = Parts Based Registration

Does the image at $\mathbf{x}_i$ look like the $i^{\text{th}}$ part?

$$\min_{\mathbf{x}} \sum_{i=1}^{N} D_i(\mathbf{x}_i) + \lambda \; R(\mathbf{x})$$

Do the joint locations of the parts match the object?

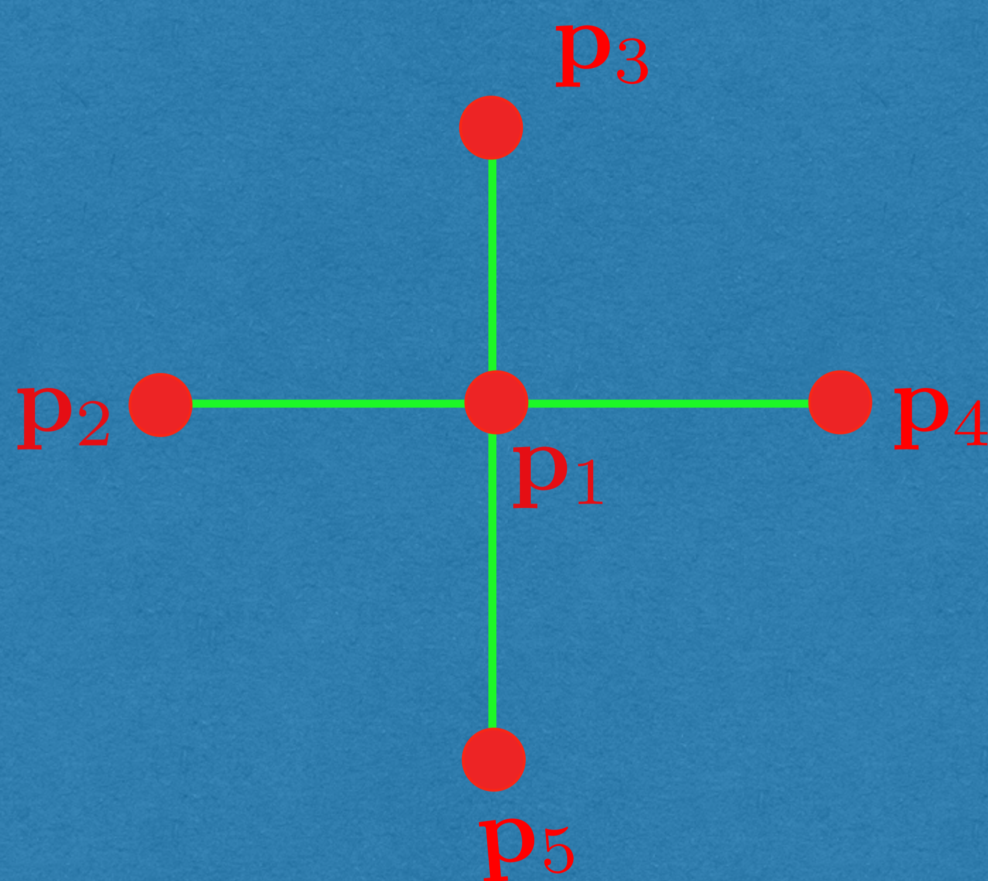
# Reminder - Exhaustive Search




$$\mathcal{O}(M^N)$$

*"We can do much better than this if the graph is sparse."*

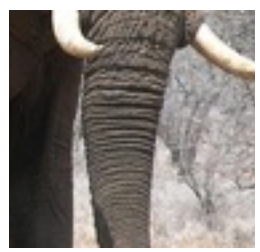
# Reminder - Exhaustive Search



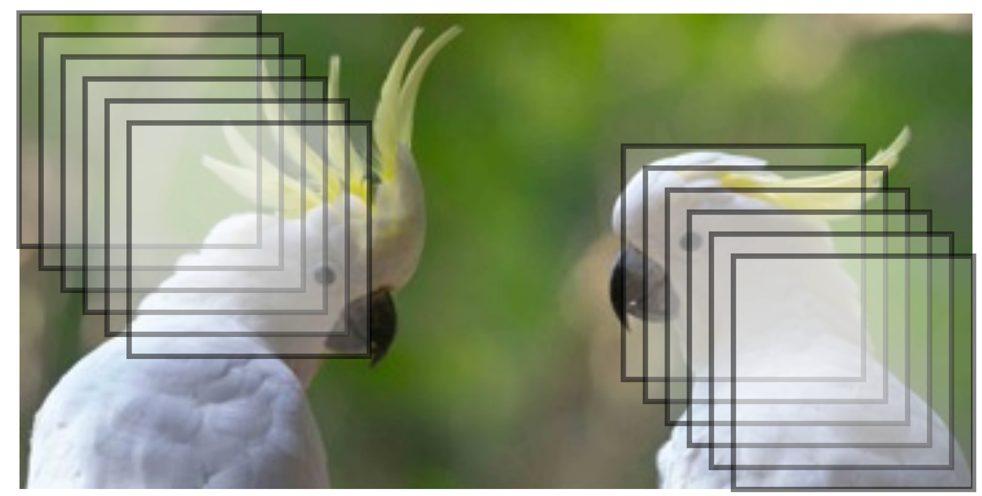

$$\mathcal{O}(NM^2)$$

*"We can do much better than this if the graph is sparse."*

# Learning $\{D_i(\mathbf{x}_i)\}_{i=1}^N$



patch at pixel *u*
(positive)

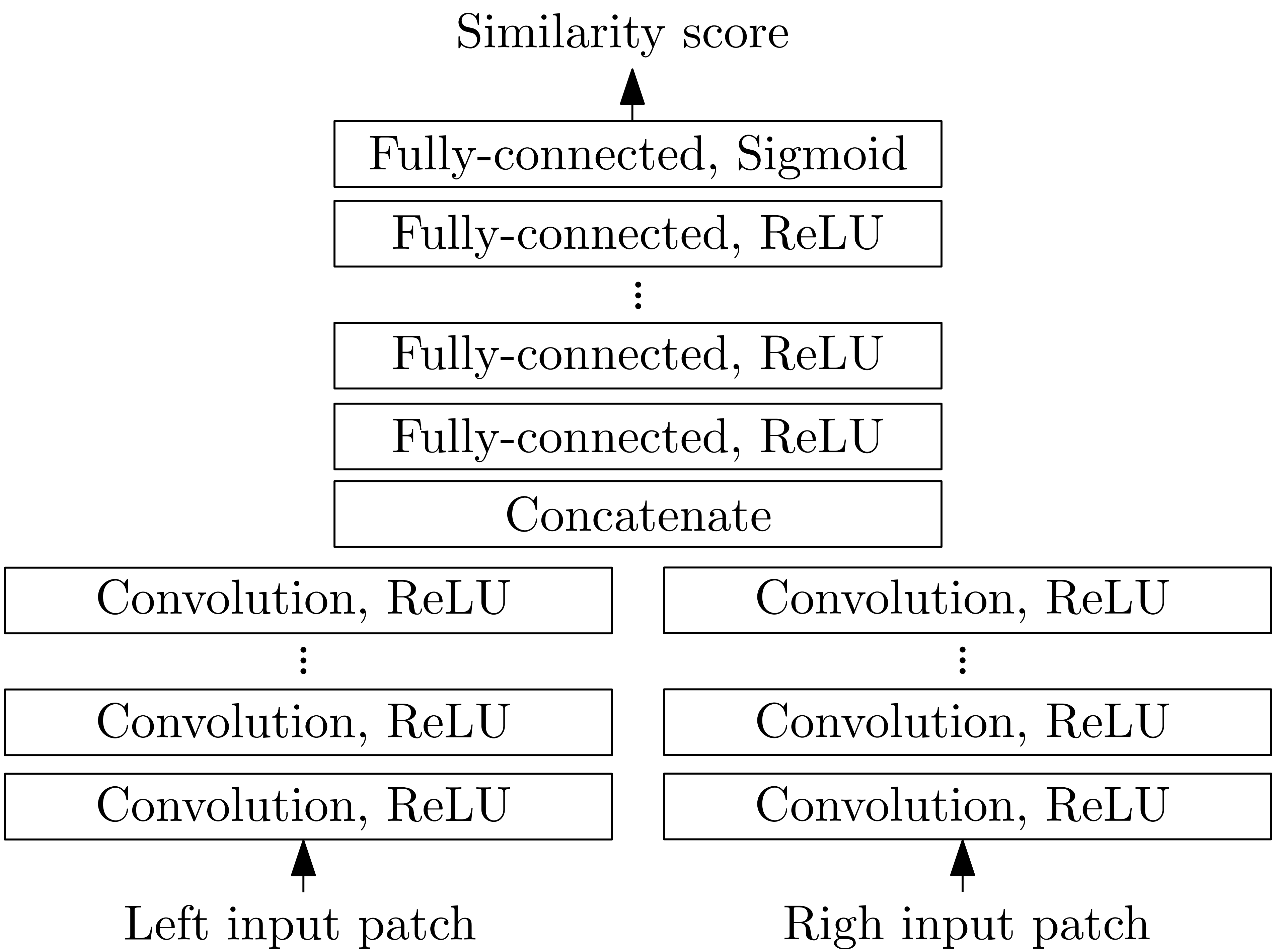


all other patches
(negatives)

for every pixel *u*

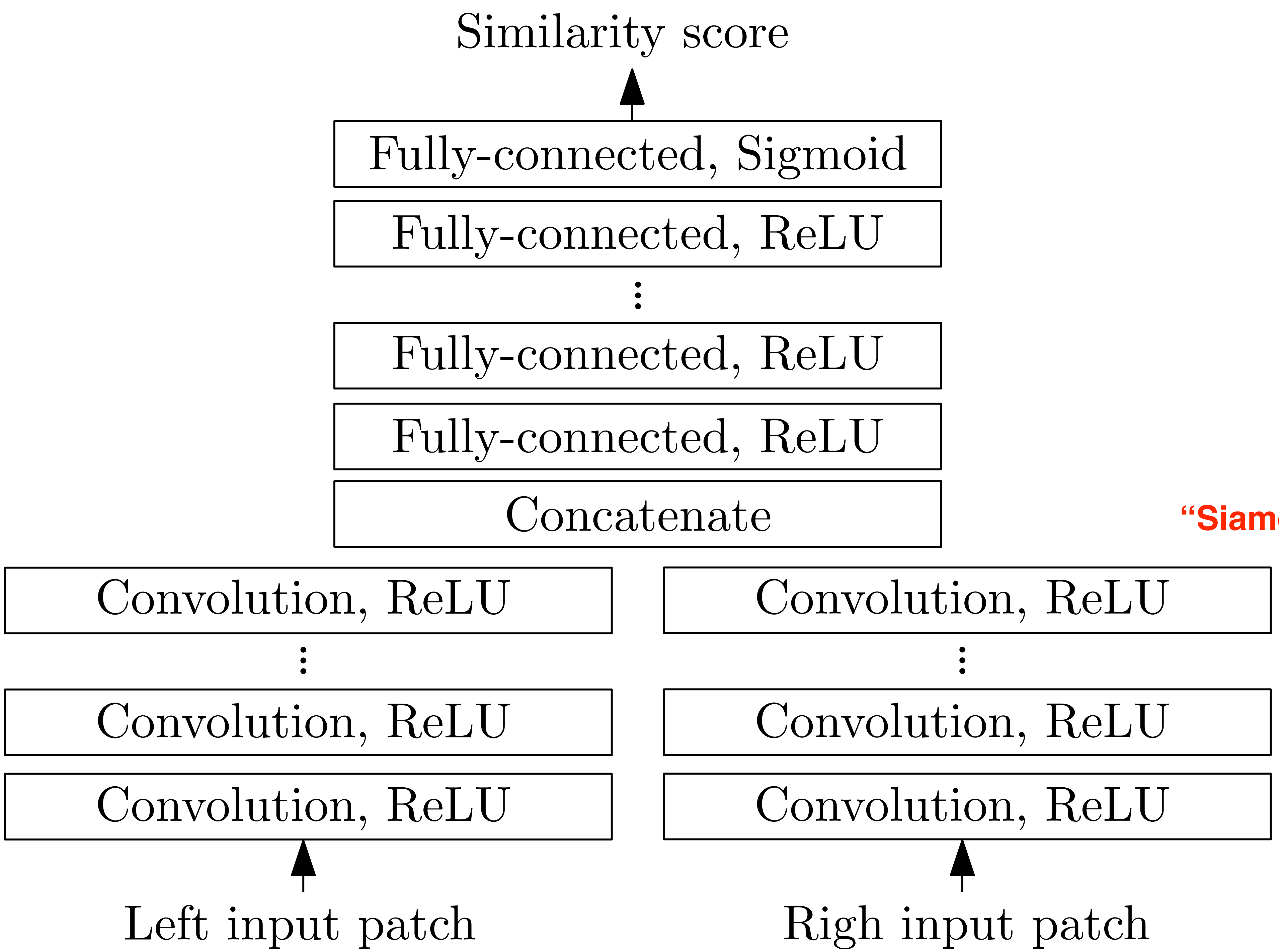
H. Bristow, J. Valmadre, and S. Lucey "Dense Semantic Correspondence where Every Pixel is a Classifier", ICCV 2015.

# Learning $\{D_i(\mathbf{x}_i)\}_{i=1}^{N}$




J. Zˇbontar & Y. LeCunn "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches", JMLR 2015.

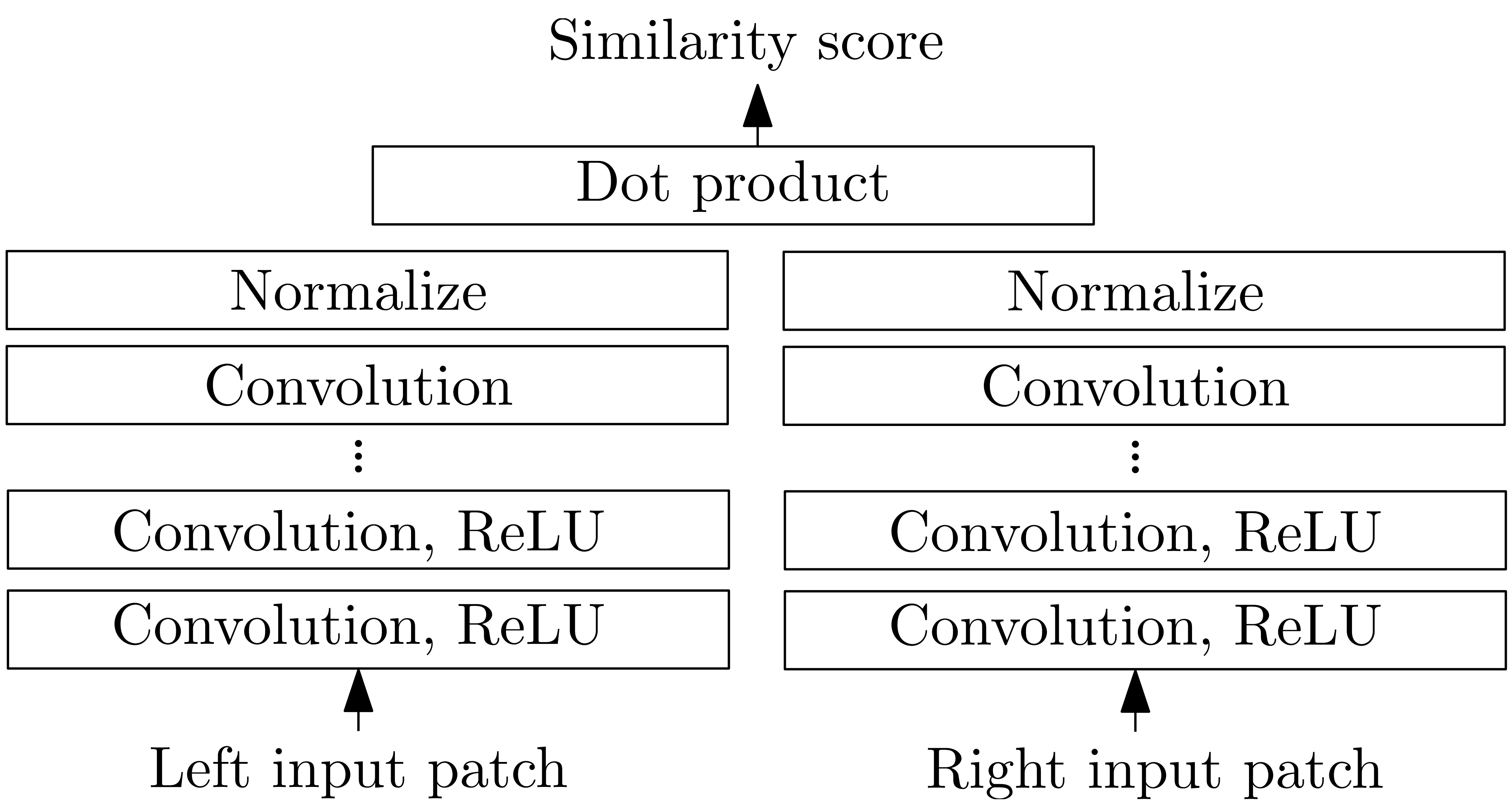
# Learning $\{D_i(\mathbf{x}_i)\}_{i=1}^N$




**"Siamese" network**

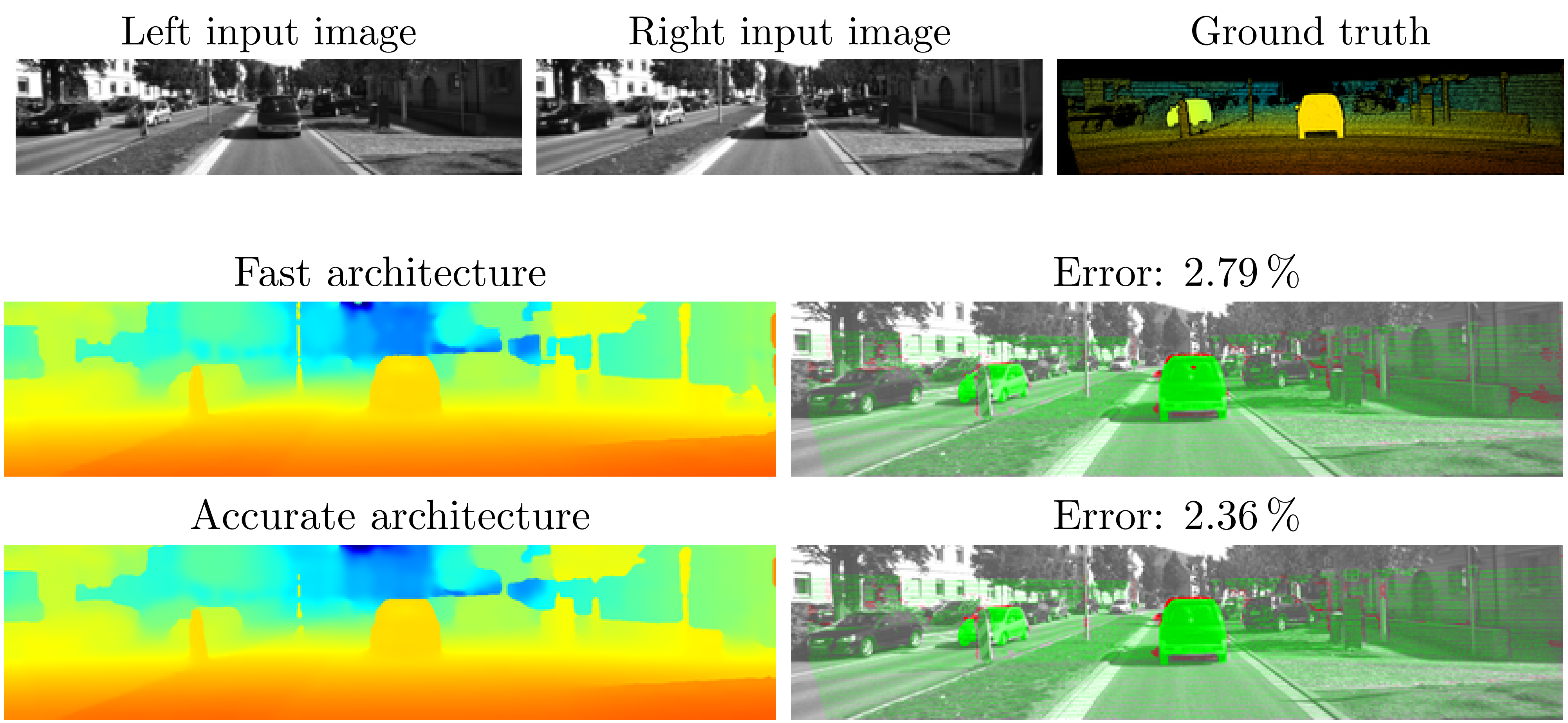
J. Zˇbontar & Y. LeCunn "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches", JMLR 2015.

# Fast Architecture




J. Zˇbontar & Y. LeCunn “Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches”, JMLR 2015.

# Results - KITTI 2015




J. Zˇbontar & Y. LeCunn “Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches”, JMLR 2015.

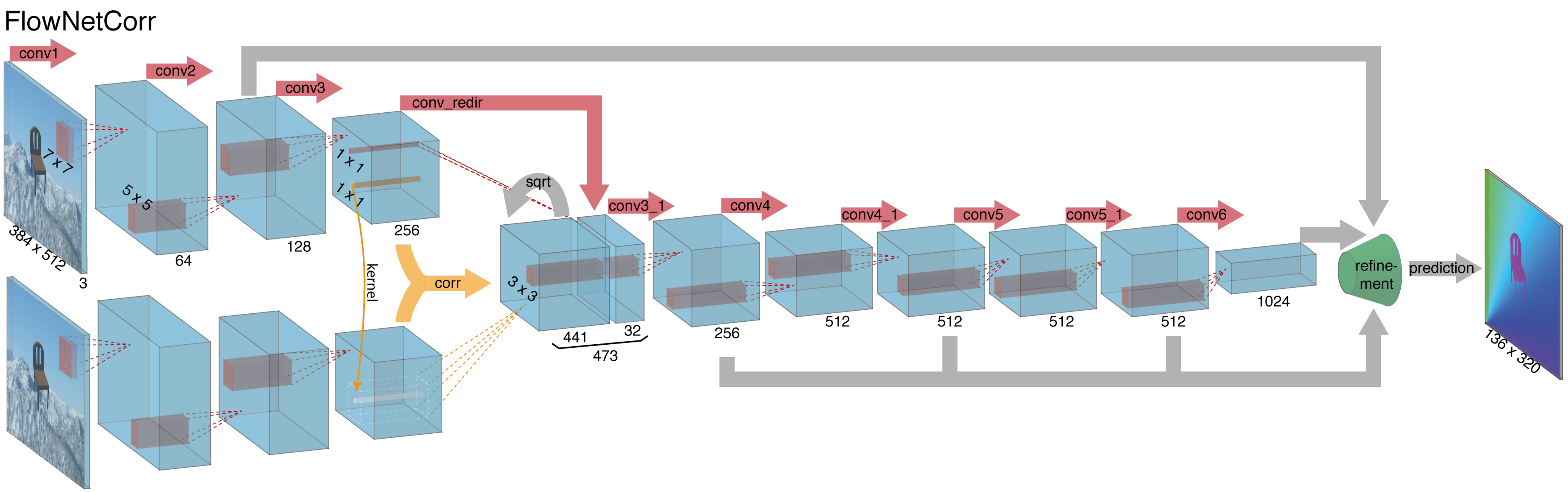
# More Data?

N. Mayer, D. Cremers, T. Brox, et al. “A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation”, CVPR 2016.

# More Data?

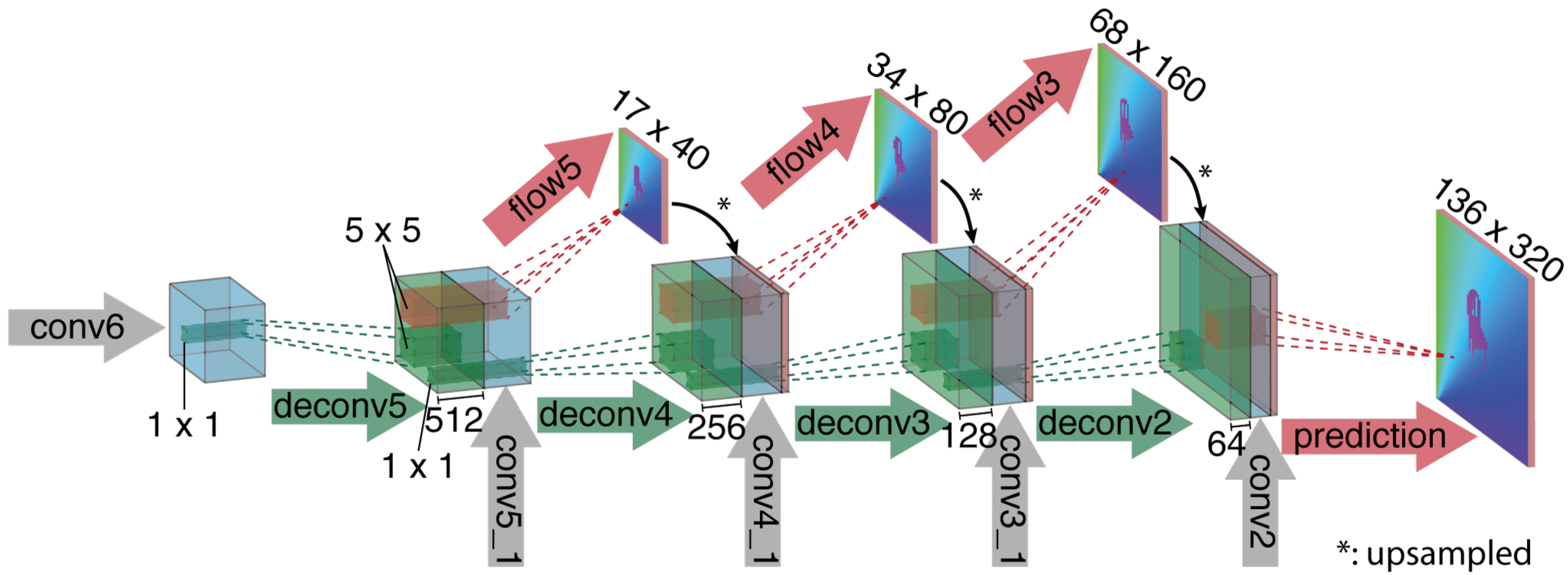
N. Mayer, D. Cremers, T. Brox, et al. “A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation”, CVPR 2016.

# FlowNet



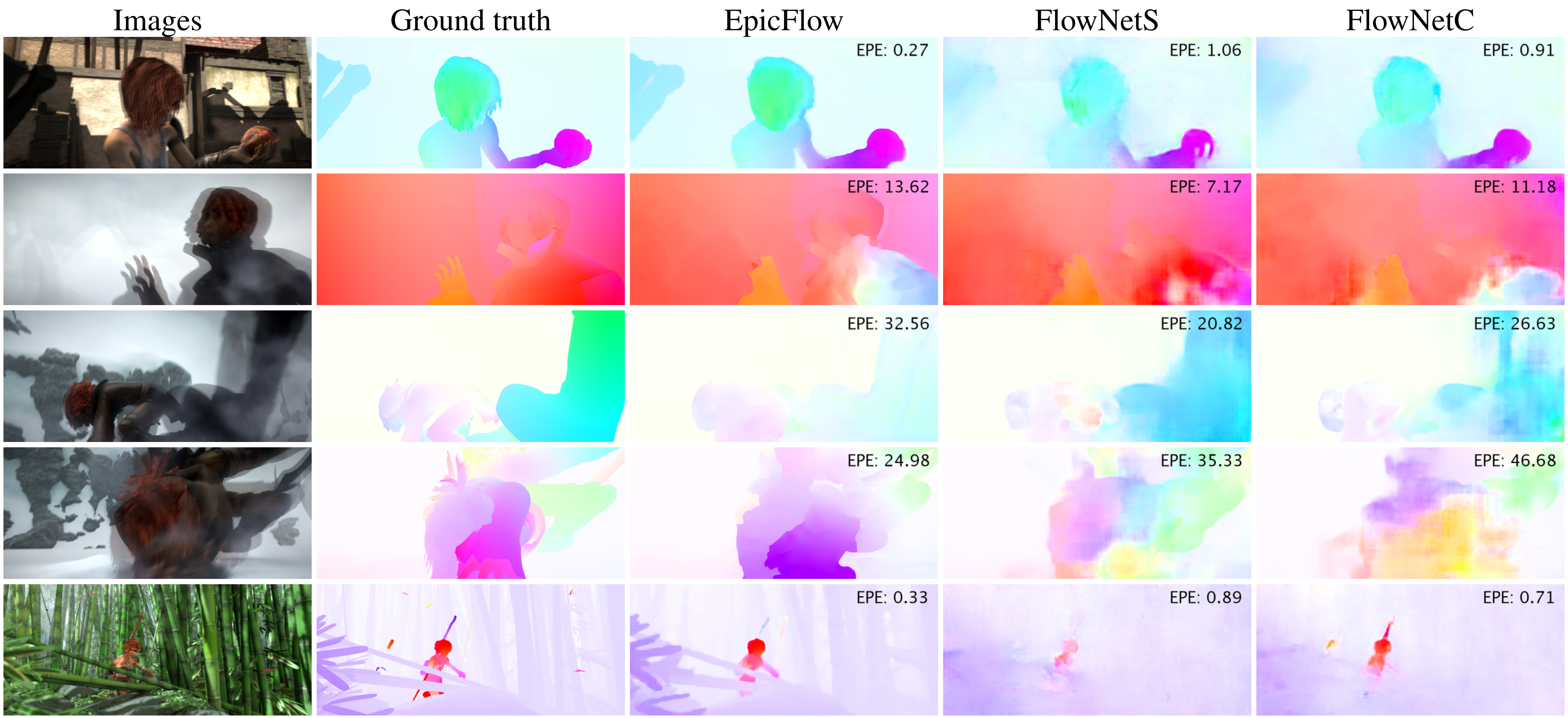

P. Fisher, D. Cremers, T. Brox, et al. “FlowNet: Learning Optical Flow with Convolutional Networks”, ICCV 2015.

# FlowNet - Refinement




P. Fisher, D. Cremers, T. Brox, et al. “FlowNet: Learning Optical Flow with Convolutional Networks”, ICCV 2015.

# FlowNet - Results




P. Fisher, D. Cremers, T. Brox, et al. "FlowNet: Learning Optical Flow with Convolutional Networks", ICCV 2015.