Imperial College London

Motion II

Dr Wenjia Bai

Department of Computing & Brain Sciences

Motion

- Optic flow methods estimate motion for each pixel in the image.
- In this lecture, we will talk about
  - How to estimate motion for objects of interest?
  - How to recognise actions in videos?

Optic flow methods aim to estimate a dense (pixel-wise) motion field.

L Sevilla-Lara, CVPR 2016. https://www.youtube.com/watch?v=QwmBSTWgr_s

Object tracking aims to estimate the motion for one or multiple objects in a video.

B Benfold. CVPR, 2011. https://www.youtube.com/watch?v=InqV34BcheM

Object tracking methods

- Lucas-Kanade tracker
- Correlation filter
- Tracking-by-detection

Lucas-Kanade tracker

- The Lucas-Kanade method is a general framework for both optic flow estimation and object tracking.
- Basic assumptions
  - Constant brightness
  - Small motion

Lucas-Kanade tracker

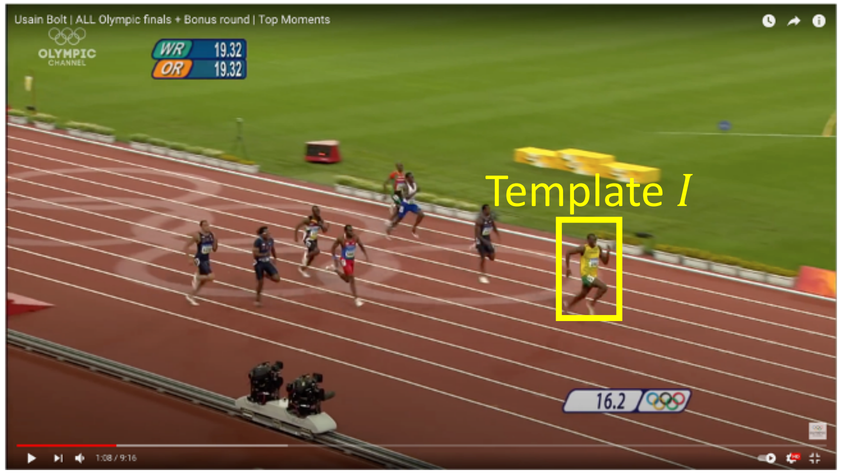
- Lucas-Kanade aims to estimate the motion from the template image I to the image J in the next time frame.
- With the brightness constancy constraint, we formulate the following optimisation problem,

$$\min_{u,v} E(u,v) = \sum_x \sum_y [I(x,y) - J(x+u, y+v)]^2$$

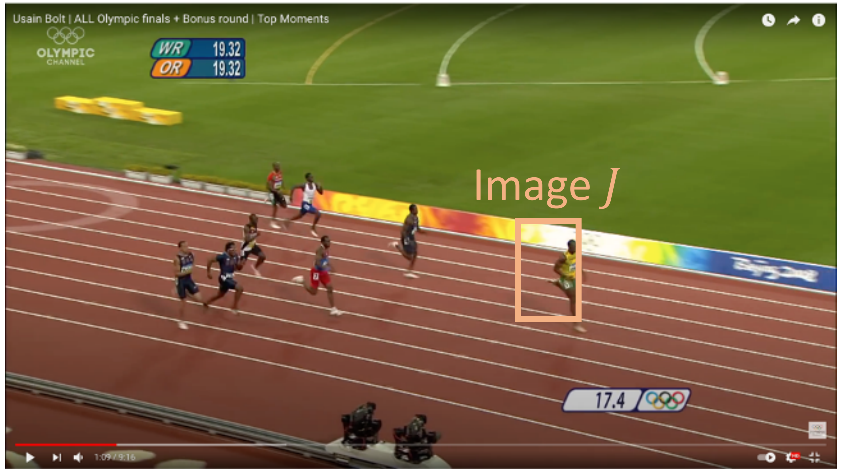
x, y: pixels in the template image

u, v: motion from template I to image J. Point (x, y) on I corresponds to Point $(x+u, y+v)$ on J.

Jean-Yves Bouguet. Pyramidal implementation of the Lucas Kanade Feature Tracker, Description of the algorithm, 1999.
Baker & Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. IJCV, 2004.



time t



time $t+1$

time $t+2$

Optimisation

- Cost function

$$\min_{u,v} E(u,v) = \sum_x \sum_y [I(x,y) - J(x+u, y+v)]^2$$

- Differentiate E with respect to u, v and let the derivatives be 0,

$$\frac{\partial E}{\partial u} = -2 \sum_x \sum_y [I(x,y) - J(x+u, y+v)] \frac{\partial J}{\partial x} = 0$$

$$\frac{\partial E}{\partial v} = -2 \sum_x \sum_y [I(x,y) - J(x+u, y+v)] \frac{\partial J}{\partial y} = 0$$

Optimisation

- With small motion assumption and Taylor expansion for J, we have

$$\frac{\partial E}{\partial u} = -2\sum_{x}\sum_{y}\left[I(x,y) - J(x,y) - \frac{\partial J}{\partial x}u - \frac{\partial J}{\partial y}v\right]\frac{\partial J}{\partial x} = 0$$

$$\frac{\partial E}{\partial v} = -2\sum_{x}\sum_{y}\left[I(x,y) - J(x,y) - \frac{\partial J}{\partial x}u - \frac{\partial J}{\partial y}v\right]\frac{\partial J}{\partial y} = 0$$

- With small motion assumption, we approximate $\frac{\partial J}{\partial x}$ by $\frac{\partial I}{\partial x}$, $\frac{\partial J}{\partial y}$ by $\frac{\partial I}{\partial y}$.
- We also have $\frac{\partial I}{\partial t} = J(x,y) - I(x,y)$.

Optimisation

- We can rewrite the equations as,

$$\frac{\partial E}{\partial u} = -2 \sum_x \sum_y \left[-\frac{\partial I}{\partial t} - \frac{\partial I}{\partial x} u - \frac{\partial I}{\partial y} v \right] \frac{\partial I}{\partial x} = 0$$

$$\frac{\partial E}{\partial v} = -2 \sum_x \sum_y \left[-\frac{\partial I}{\partial t} - \frac{\partial I}{\partial x} u - \frac{\partial I}{\partial y} v \right] \frac{\partial I}{\partial y} = 0$$

- Or simply as,

$$\frac{\partial E}{\partial u} = -2 \sum_x \sum_y [-I_t - I_x u - I_y v] I_x = 0$$

$$\frac{\partial E}{\partial v} = -2 \sum_x \sum_y [-I_t - I_x u - I_y v] I_y = 0$$

Optimisation

- Rewrite

$$\frac{\partial E}{\partial u} = -2 \sum_x \sum_y [-I_t - I_x u - I_y v] I_x = 0$$

$$\frac{\partial E}{\partial v} = -2 \sum_x \sum_y [-I_t - I_x u - I_y v] I_y = 0$$

in matrix form, we have

$$-\sum_x \sum_y \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix} - \sum_x \sum_y \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0$$

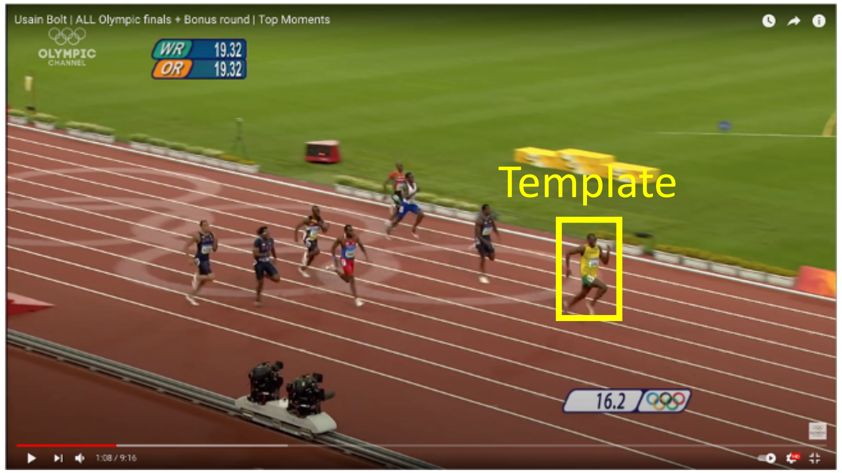
Optimisation

- Therefore the motion (u, v) can be solved,

$$\begin{bmatrix} u \\ v \end{bmatrix} = -\left(\sum_x \sum_y \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}\right)^{-1} \sum_x \sum_y \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix}$$

- We get the same solution as the Lucas-Kanade optic flow method, if you check the slides from last lecture.
  - Lucas-Kanade optic flow calculates matrix by summing over a small neighbourhood.
  - Lucas-Kanade tracker calculates matrix by summing over pixels within the template image.

Baker & Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. IJCV, 2004.



time t

time $t+1$

time $t+2$

Lucas-Kanade tracker

- In the previous derivation, we assume the motion is simply translation,

$$\min_{u,v} E(u,v) = \sum_x \sum_y [I(x,y) - J(x+u, y+v)]^2$$

- For general cases, we use a parametric model for the motion,

$$\min_{u,v} E(u,v) = \sum_x \sum_y [I(x,y) - J(W(x,y;p))]^2$$

  - For example,

$$W(x,y;p) = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

  - The problem can be solved similarly.

Object tracking

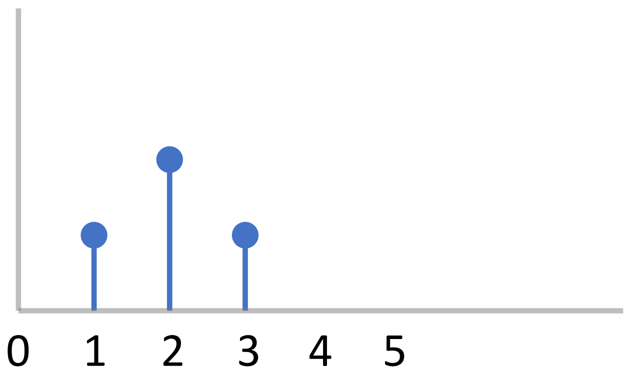
- The Lucas-Kanade method is a classical method.
  - However, the brightness constancy assumption may not always hold.
  - Lucas-Kanade only uses pixel intensities. It does not learn discriminative features for the template.
- Some recent object tracking methods may perform better.
  - Correlation filter method
  - Tracking by detection

Correlation filter

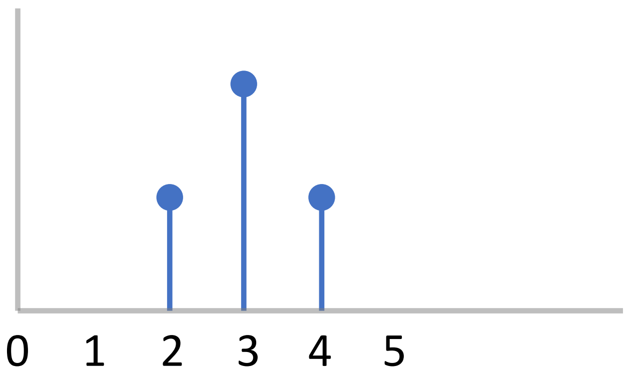
- We aim to maximise the correlation between template features and image features in the next time frame.

$$(f \star h)[n] = \sum_{m=-\infty}^{\infty} f[m]h[n+m]$$

- Correlation can be more robust to illumination changes than sum of squared difference.



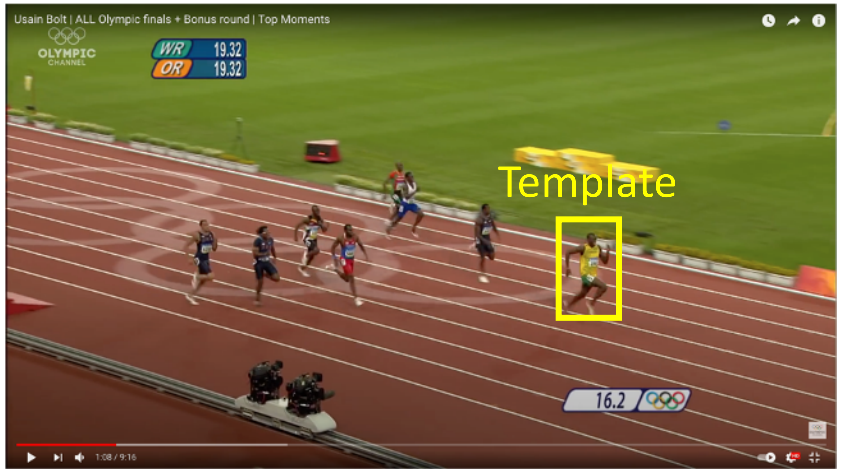
Template, t



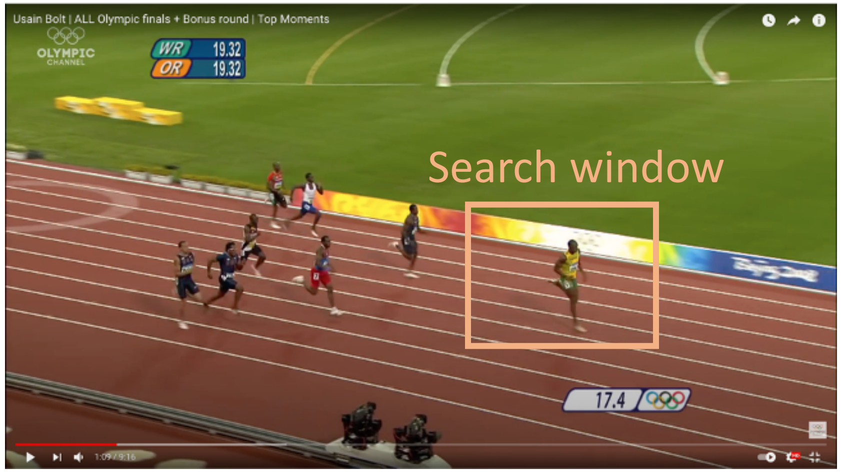
Image, $t+1$

Correlation filter

- For 2D images, we can find where the correlation achieves the maximum between the template features and features in a search window.
- The features can be learnt from CNNs.



time t

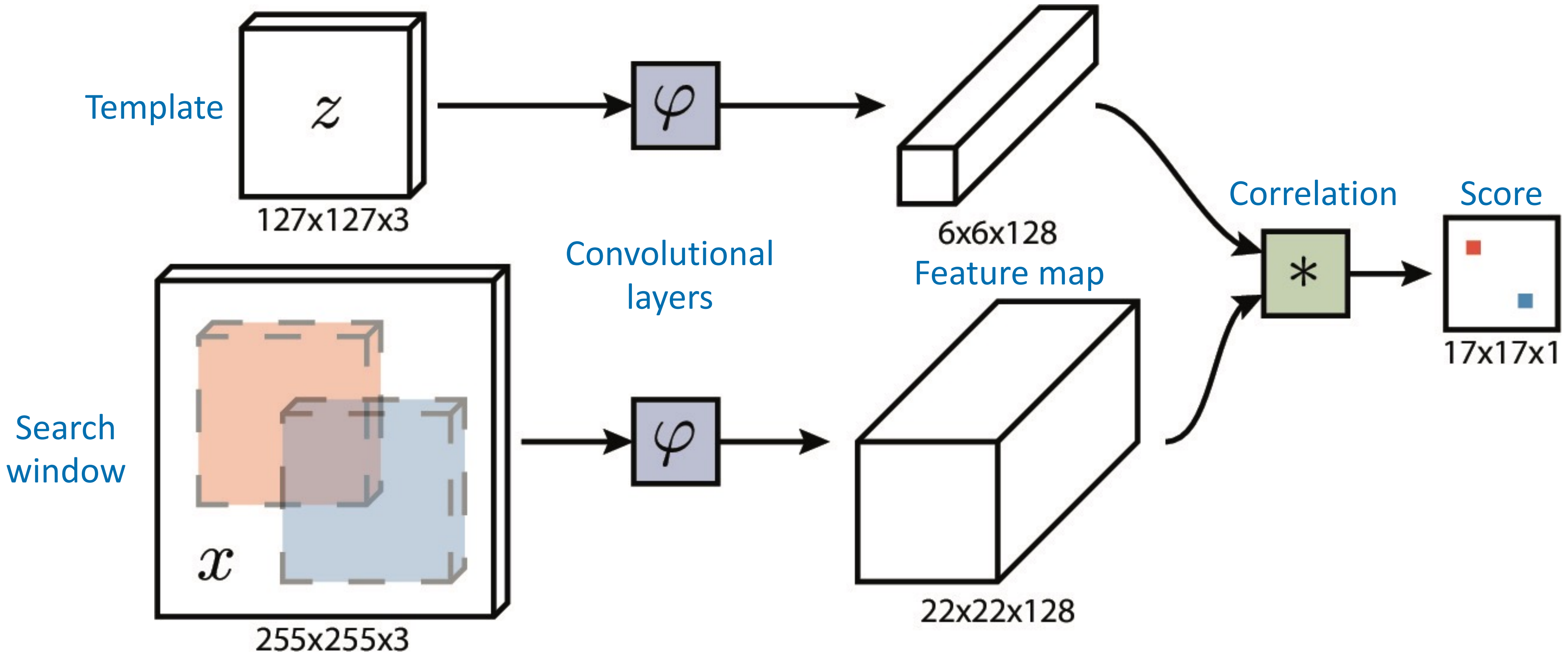


time $t+1$

time $t+2$

Correlation filter

- A recent approach uses a Siamese network to learn and compare features.



L. Bertinetto et al. Fully-Convolutional Siamese Networks for Object Tracking. ECCV Workshop, 2016.

AlexNet

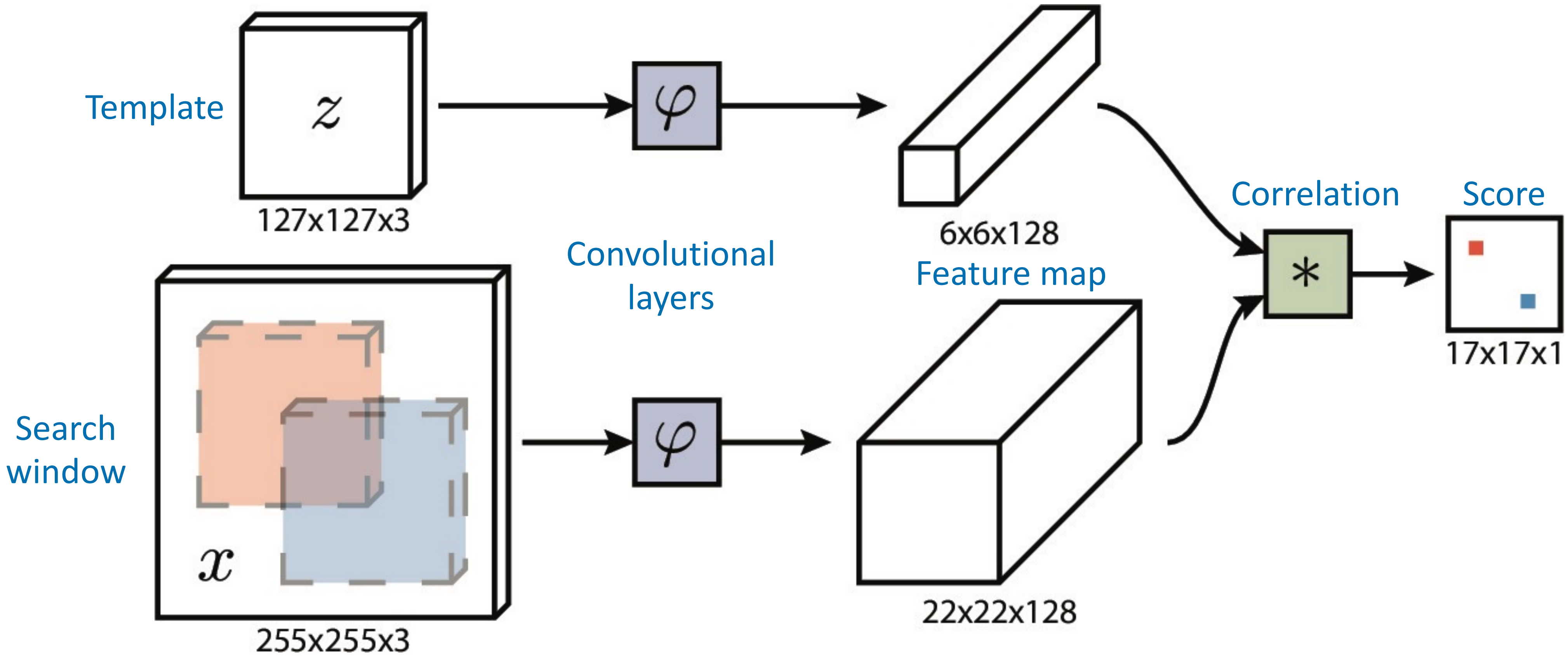
[224x224x3] Input
[55x55x96] Conv1, 11x11, 96, s=4
[55x55x96] Norm1
[27x27x96] Pool1
[27x27x256] Conv2, 5x5, 256
[27x27x256] Norm2
[13x13x256] Pool2
[13x13x384] Conv3, 3x3, 384
[13x13x384] Conv4, 3x3, 384
[13x13x256] Conv5, 3x3, 256 — convolutional feature map
[13x13x256] Norm3
[6x6x256] Pool3
[4096] FC1
[4096] FC2
[1000] FC3 (class score)

VGG-16

[224x224x3] Input
[224x224x64] 3x3 conv1, 64
[224x224x64] 3x3 conv1, 64
[112x112x64] Pool
[112x112x128] 3x3 conv2, 128
[112x112x128] 3x3 conv2, 128
[56x56x128] Pool
[56x56x256] 3x3 conv3, 256
[56x56x256] 3x3 conv3, 256
[56x56x256] 3x3 conv3, 256
[28x28x256] Pool
[28x28x512] 3x3 conv4, 512
[28x28x512] 3x3 conv4, 512
[28x28x512] 3x3 conv4, 512
[14x14x512] Pool
[14x14x512] 3x3 conv5, 512
[14x14x512] 3x3 conv5, 512
[14x14x512] 3x3 conv5, 512 — convolutional feature map
[7x7x512] Pool
[4096] FC, 4096
[4096] FC, 4096
[1000] FC, 1000

Correlation filter

- The network parameters are trained to predict a ground truth score map.

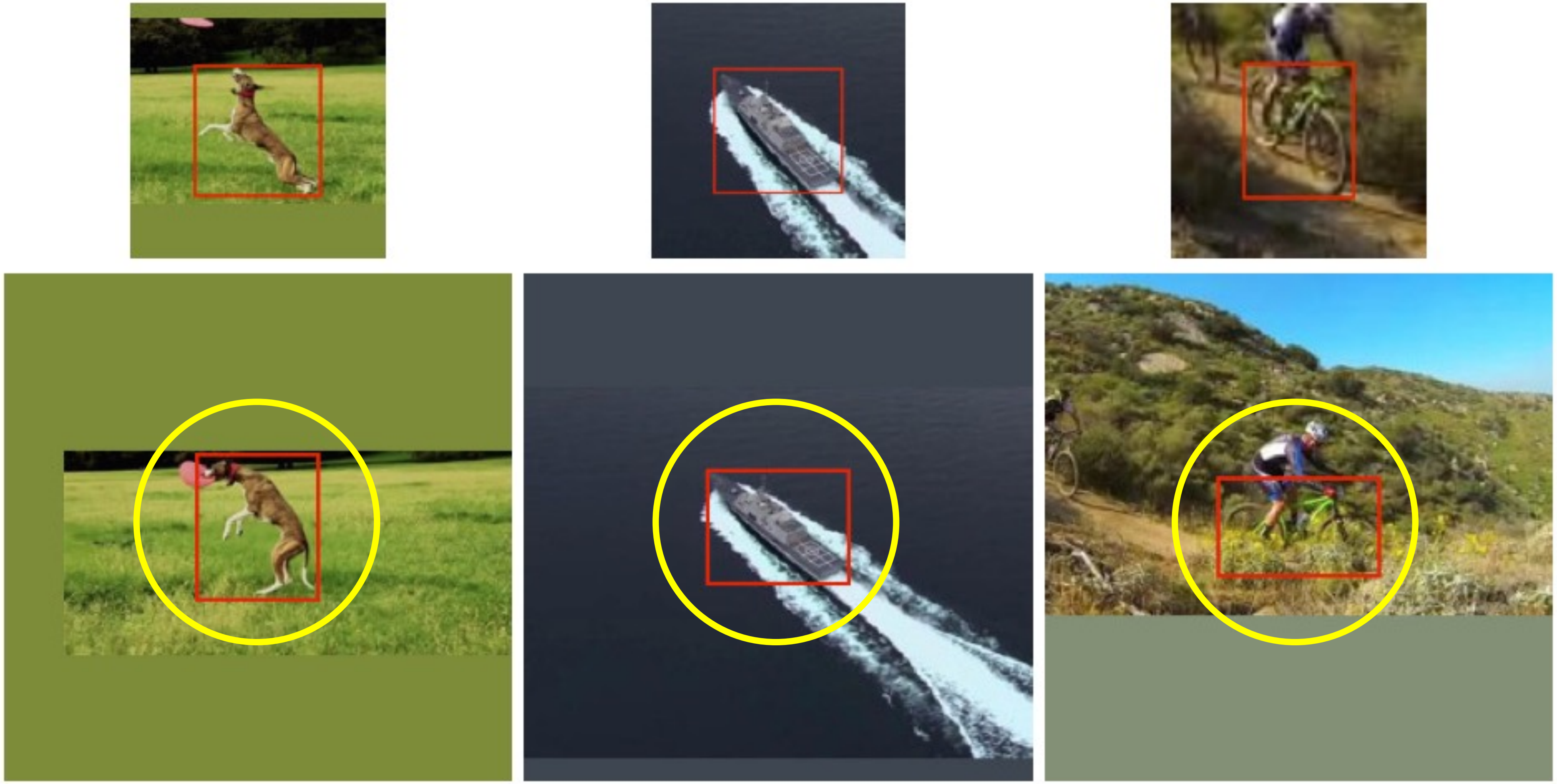


L. Bertinetto et al. Fully-Convolutional Siamese Networks for Object Tracking. ECCV Workshop, 2016.

Training data

Template

Search
window



The ground truth score is +1 within the yellow circle and -1 outside.

Example tracking results.

https://www.youtube.com/watch?v=zS6kiI8DgDY

Tracking by detection

- Isn’t object tracking a bit like object detection?
  - Yes, we can perform tracking by detection.
  - We want to learn what the object of interest looks like.

- Apart from their similarity, there are also some differences between object tracking and object detection.
  - Tracking concerns about videos.
  - In tracking, we have an initial bounding box. We are only interested in how this object moves. We do not need to detect every objects.

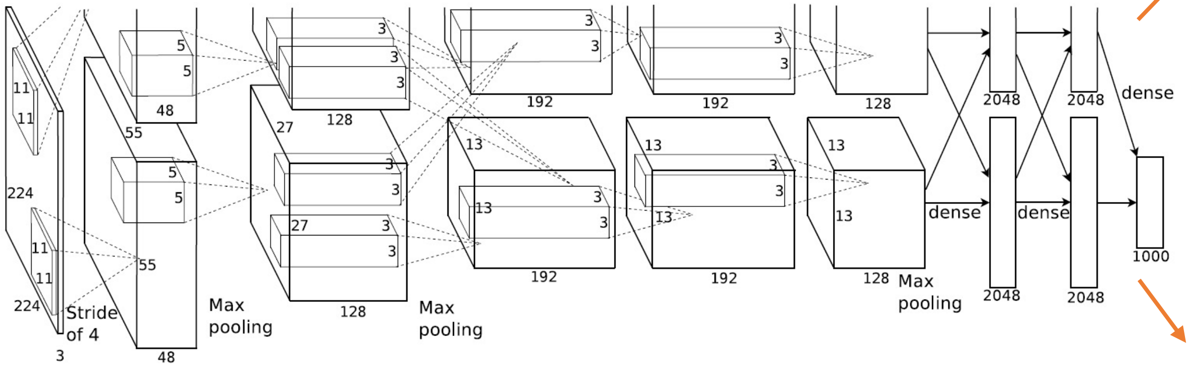
H. Nam et al. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. CVPR, 2016.
M. Danelljan et al. ATOM: Accurate Tracking by Overlap Maximization. CVPR, 2019.

Tracking by detection

- Since we are interested in the object (initial bounding box) defined in the first time frame, we can learn features specific for this object.
- We can perform online learning.
  - Online: using information from this video, while it is streamed.
  - Offline: using information from an existing training set.

Object detection

- At each sliding window, we perform two tasks:
  - Task 1: classification
  - Task 2: localisation

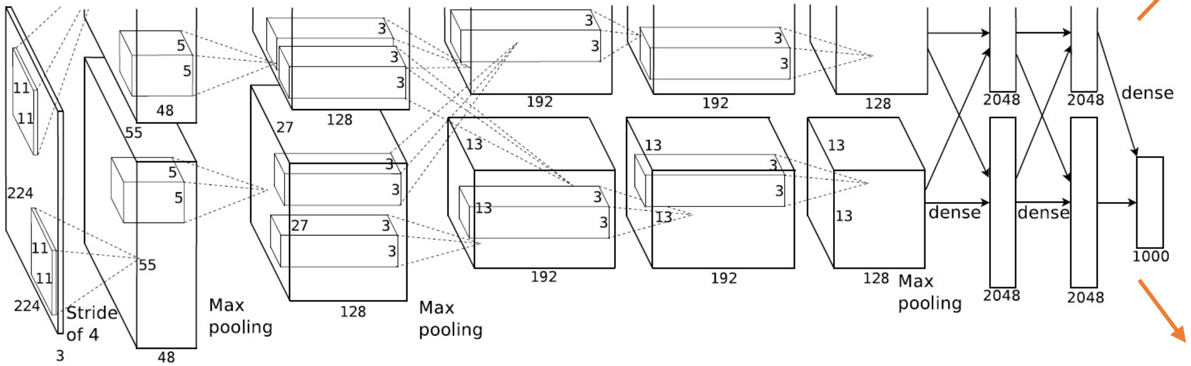


Fully connected to K classes → Class score → $L_{classification}$ ← Ground truth class label

Fully connected to 4 variables → Bounding box (X, Y, W, H) → $L_{localisation}$ ← Ground truth bounding box

Tracking by detection

- Using the same framework as object detection
  - However, we only perform binary classification, instead of multi-class classification.
  - The localisation task enables us to adjust the size of the object during tracking.

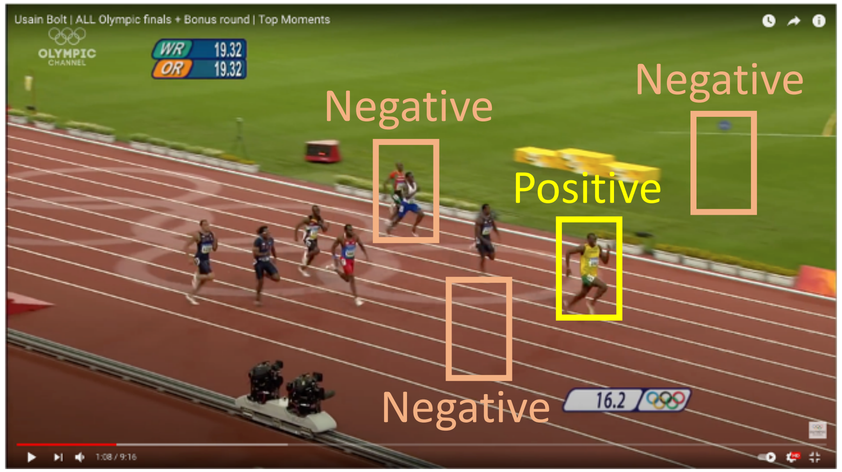


Fully connected to **2** classes → Class score → $L_{classification}$ ← Ground truth class label

Fully connected to 4 variables → Bounding box (X, Y, W, H) → $L_{localisation}$ ← Ground truth bounding box

Training data

- The positive and negative samples for binary classification are extracted from the first time frame. More samples can be generated by
  - Data augmentation (translation, rotation, etc).
  - Using more time frames while the video is streamed.



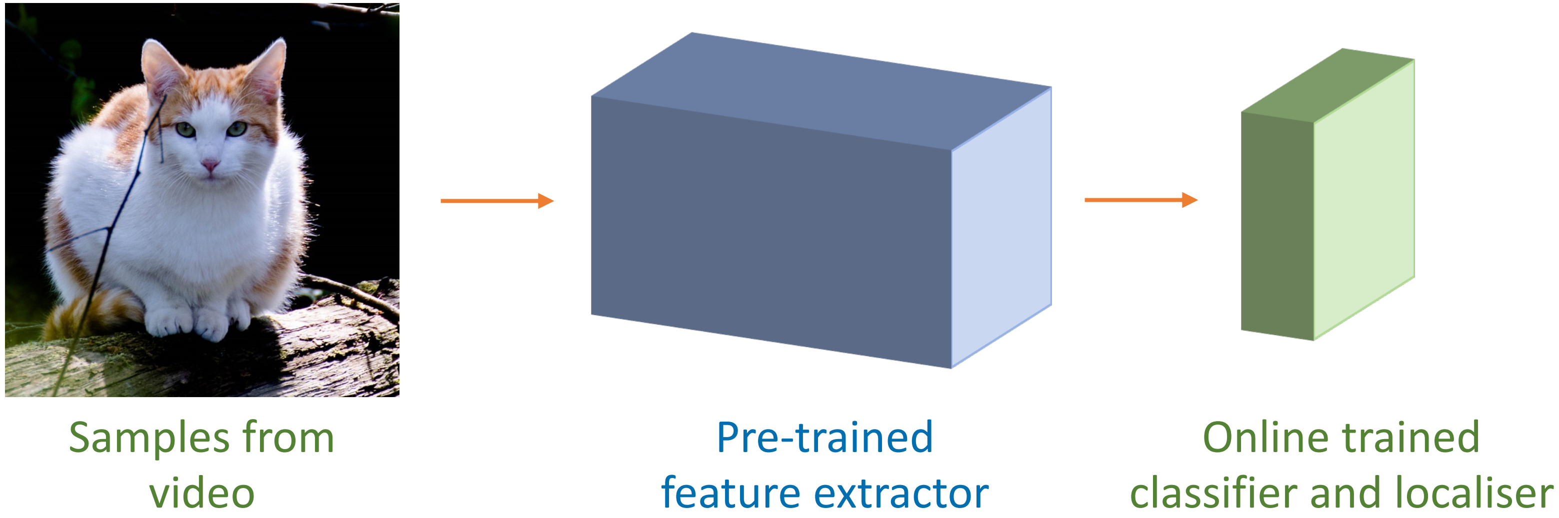
time t

time $t+1$

time $t+2$

Training data

- Most layers of the convolutional network are pre-trained offline using large datasets such as ImageNet.
- Only the last few layers are trained online, using dozens or hundreds of samples extracted from the video.



Tracking by detection

M. Danelljan et al. ATOM: Accurate Tracking by Overlap Maximization. CVPR, 2019.

Tracking by detection

TLD Vision https://www.youtube.com/watch?v=zgQS_EW-0zw

Action recognition

- Apart from object tracking, action recognition is another interesting application that involves videos and motion.

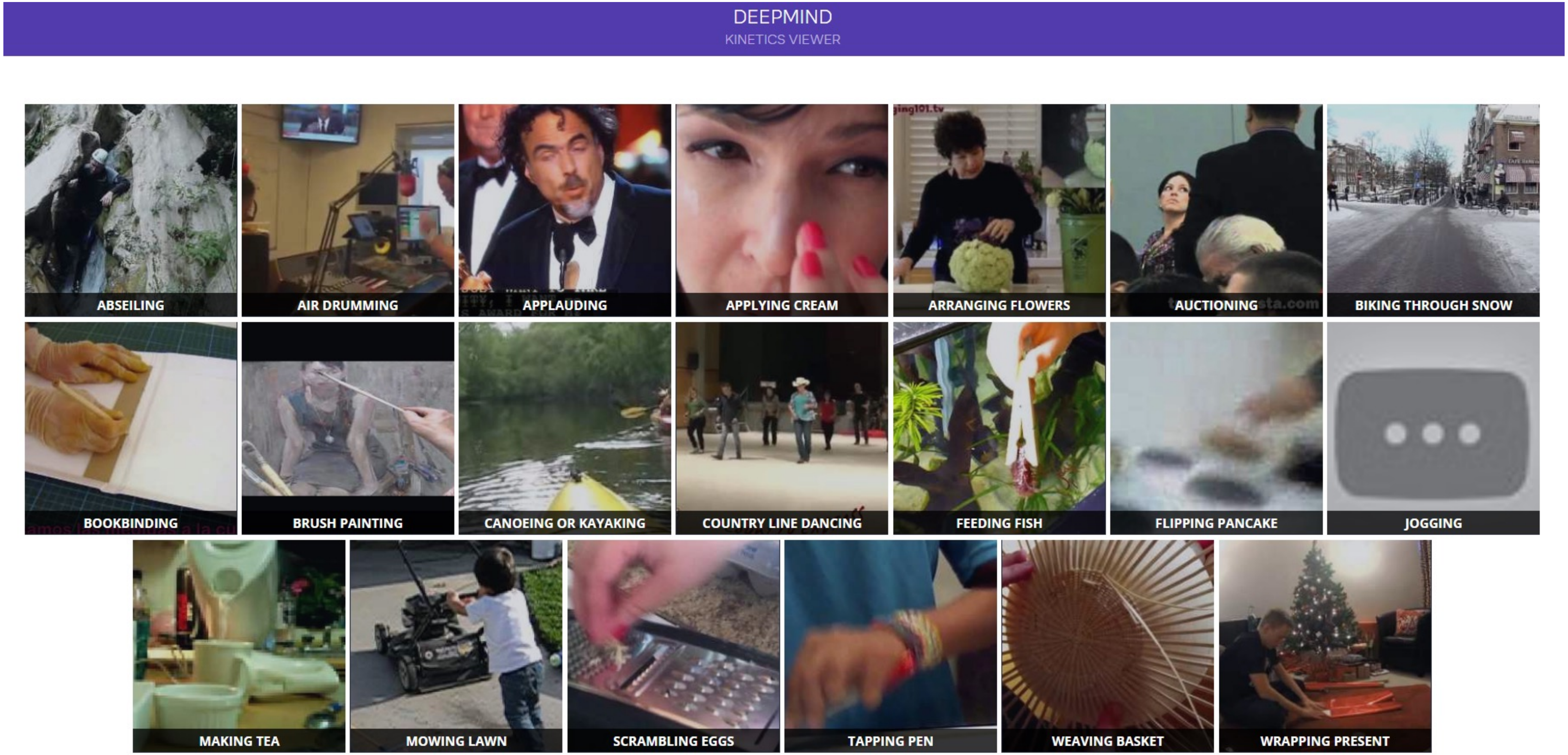
It is challenging to recognise actions from still images.

Action recognition

- How do we curate a dataset for action recognition?
- Kinetics Human Action Video Dataset [1,2]
  - 600 human action classes
  - More than 600 examples per class
  - Each from a unique YouTube video

It is challenging to recognise actions from still images.

[1] J. Carreira and A. Zisserman. Quo Vadis, action recognition? A new model and the Kinetics dataset. CVPR 2017.
[2] https://deepmind.com/research/open-source/open-source-datasets/kinetics/



Kinectics dataset, https://deepmind.com/research/open-source/open-source-datasets/kinetics

How does image classification work?

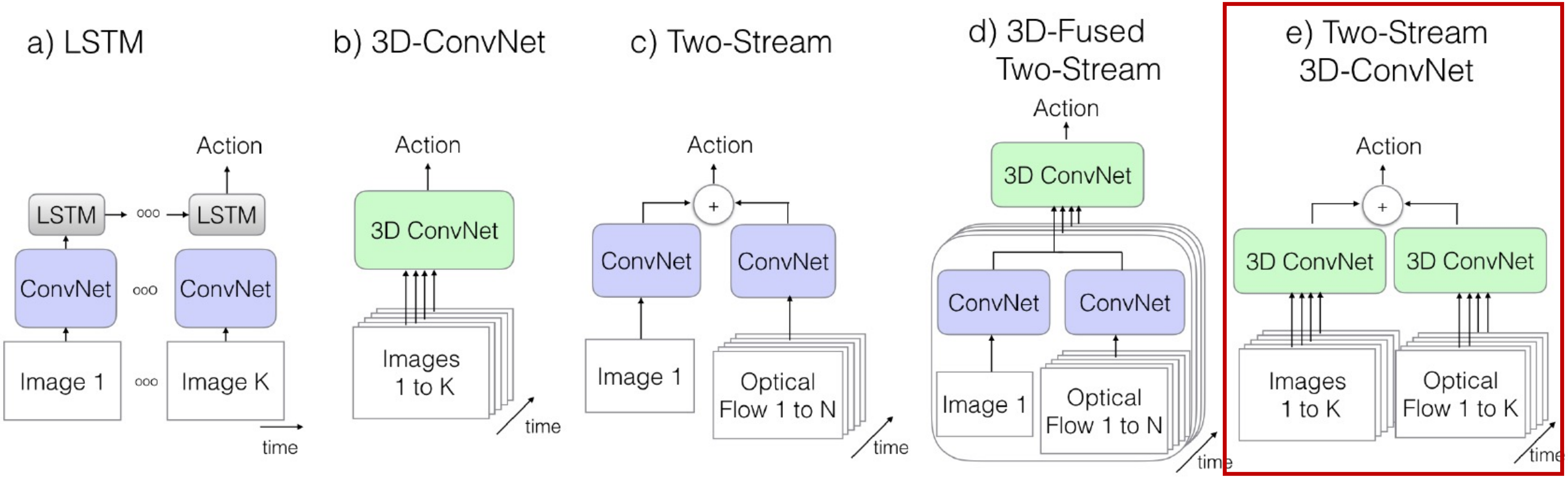
Input image $\xrightarrow{\text{Feature extraction}}$ Feature, representation $\xrightarrow{\text{Classifier}}$ Class score, output label

Transform input images into low-dimensional vectors so that they can be easily compared or matched.

Trained with a dataset consisting of images and labels (or even without labels)

How do we recognise actions?

- We use both image and motion information.
  - Image: static features extracted from a single image
  - Motion: dynamic features extracted from videos, e.g. optic flow fields
- We can combine both to train a classifier for action recognition.



Network architectures for action recognition, which use both image and motion information.

J. Carreira and A. Zisserman. Quo Vadis, action recognition? A new model and the Kinetics dataset. CVPR 2017.

Architecture	UCF-101			HMDB-51			miniKinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	81.0	–	–	36.0	–	–	69.9	–	–
(b) 3D-ConvNet	51.6	–	–	24.3	–	–	60.0	–	–
(c) Two-Stream	83.6	85.6	91.2	43.2	56.3	58.3	70.1	58.4	72.9
(d) 3D-Fused	83.2	85.8	89.3	49.2	55.5	56.8	71.4	61.0	74.0
(e) Two-Stream I3D	**84.5**	**90.6**	**93.4**	**49.8**	**61.9**	**66.4**	**74.1**	**69.6**	**78.7**

The action recognition accuracy is improved by using both image and motion information, compared to using image information alone.

J. Carreira and A. Zisserman. Quo Vadis, action recognition? A new model and the Kinetics dataset. CVPR 2017.

Summary

- Object tracking
  - Lucas-Kanade tracker
  - Correlation filter
  - Tracking-by-detection
- Action recognition