Deep Learning Winter School for Computer Vision

Srijan Das

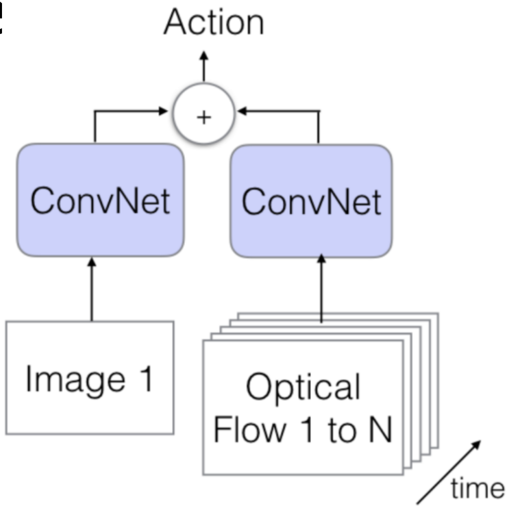
PhD Scholar

INRIA Sophia Antipolis

Recap…

Popular Action Recognition Frameworks

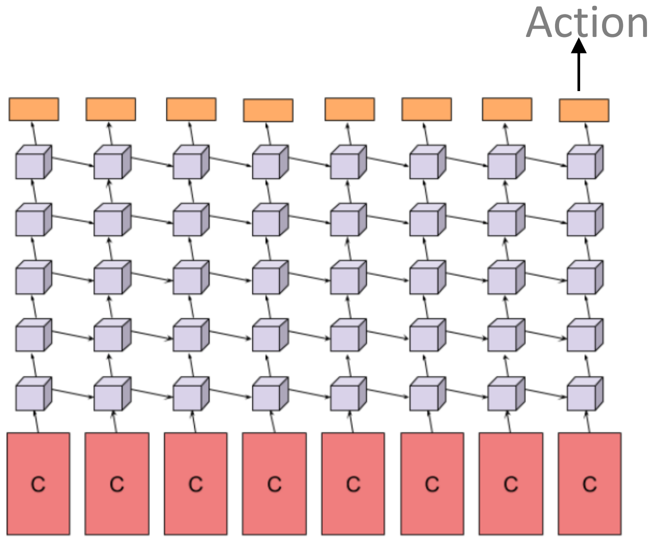
- Input: a fixed number of frames, Output: a class label



Two-stream CNNs

- 1 frame **RGB** + 10 frames of **optical flow**

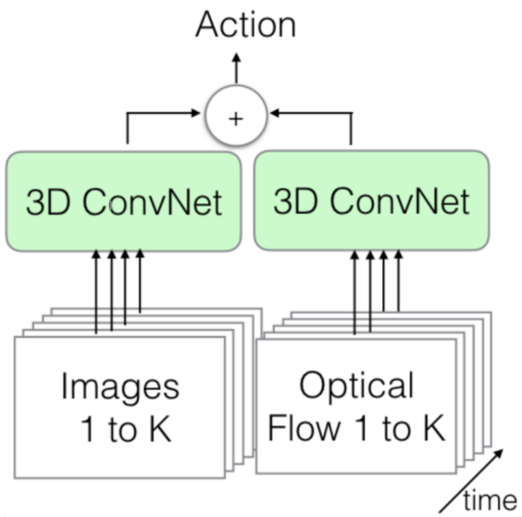
[Carreira and Zisserman, 2017]



Sequential models RNNs

- model 'sequences' of per-frame CNN representations (**RGB/3D Poses**)

[J. Ng et al., 2015]



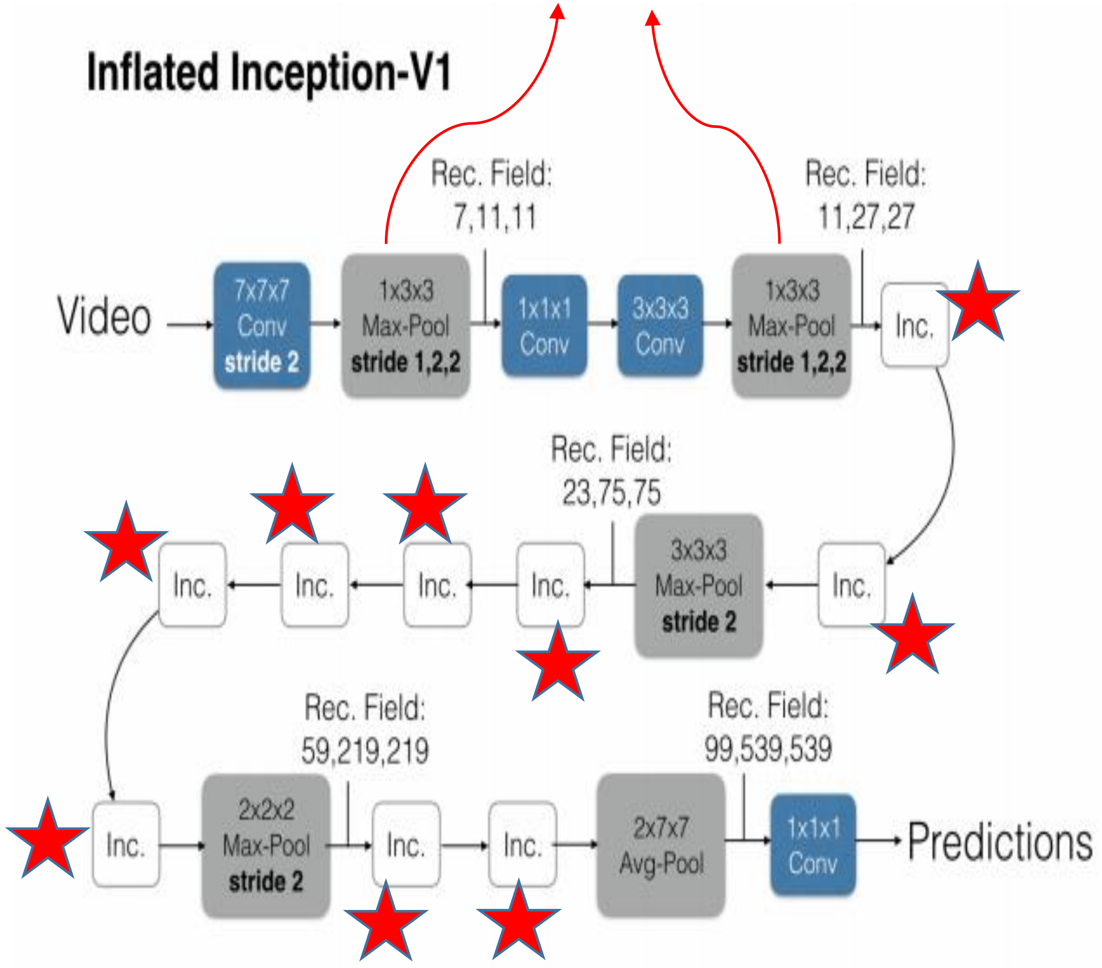
3-D XYT CNNs

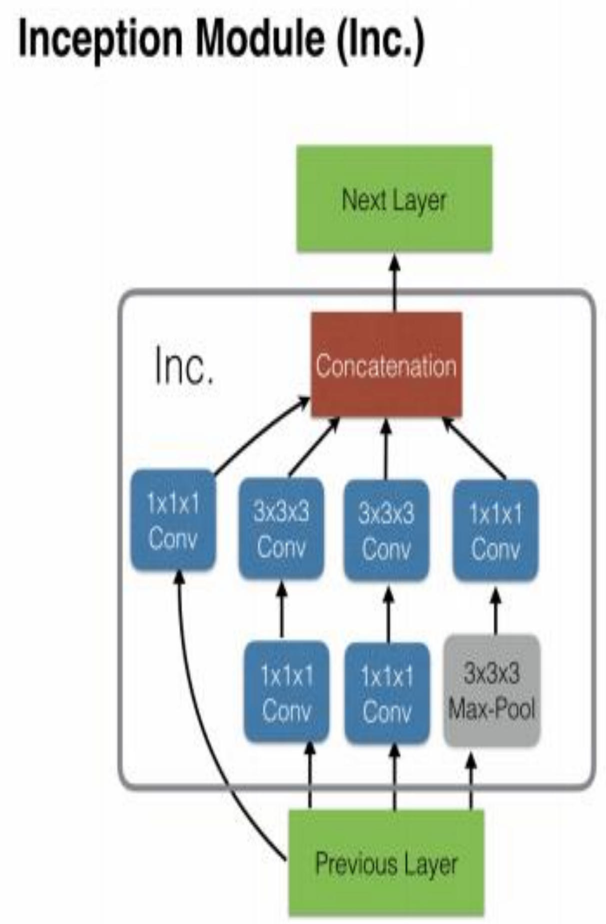
- 15~99 frames (**RGB** + **Flow**)
- Facebook C3D, Google I3D

Recap...

- Inflation
- Bottleneck
- Concept of inception

Handling space-time together with asymmetric operations





Recap…

Limitations of I3D

- Rigid spatio-temporal kernels limiting them to capture subtle motion
- No specific operations to help disambiguate similarity in actions.
- 3D (XYT) CNNs are not view-adaptive.

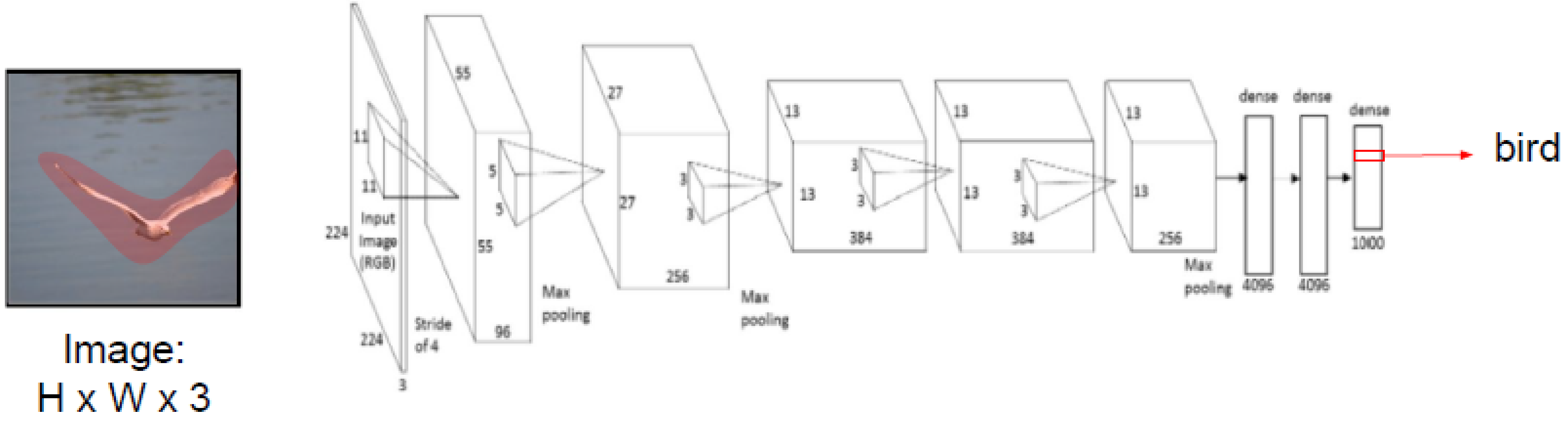
Outline: Attention Mechanism

- Introduction to Attention Mechanism

- Hard Vs Soft Attention

- Soft Attention Mechanism based Framework
  - Spatial Transformer Network
  - Self-Attention
  - Visual Attention for Action Recognition

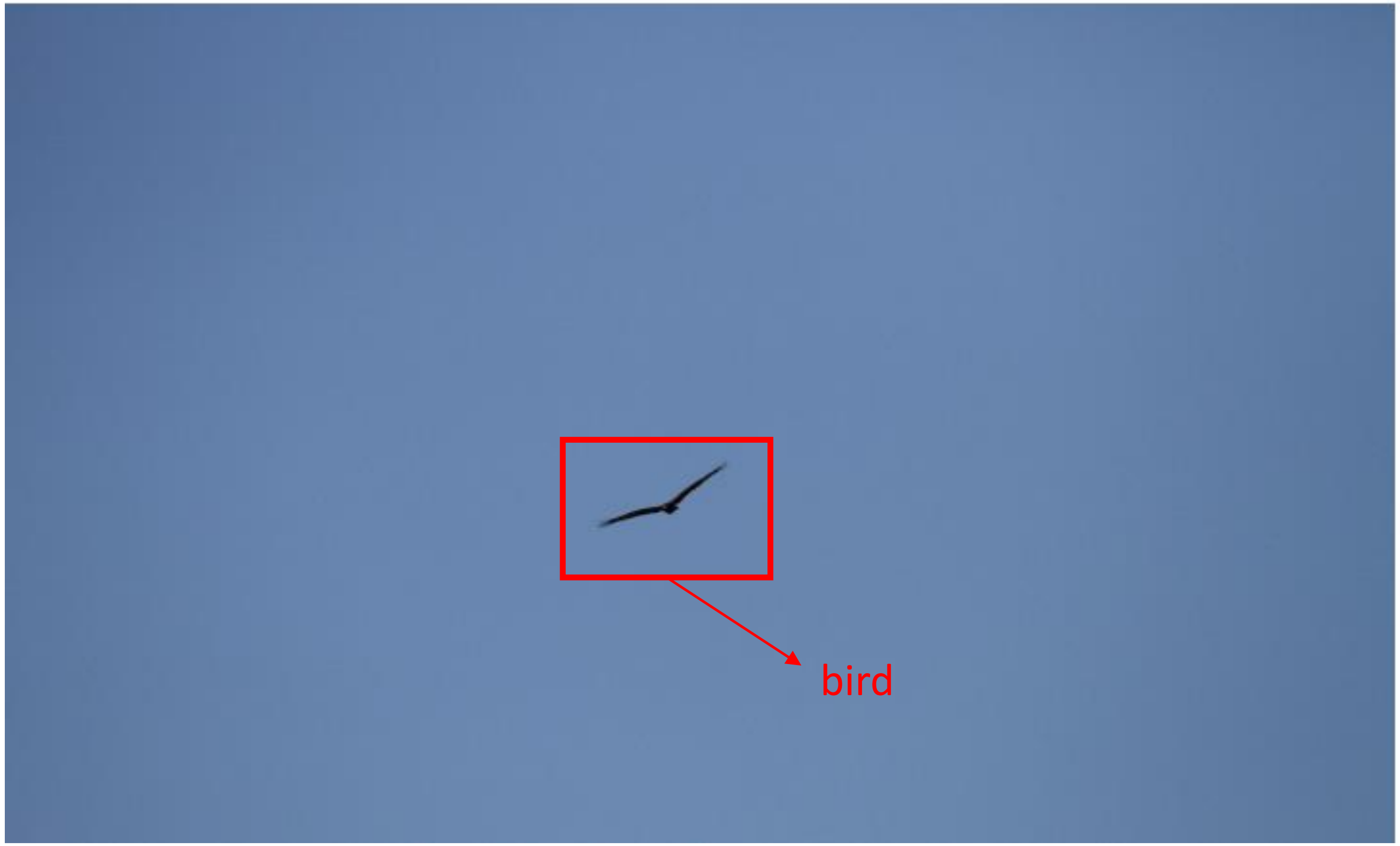
Introduction to Attention Mechanism



The whole input volume is used to predict the output...

...despite the fact that not all pixels are equally important

Introduction to Attention Mechanism



Do you need the whole image to classify that the object in this image is a bird?

Focus in the Spatial space is required!

Do we need them all?

- The girl is drinking water from a bottle
- Do you really need the whole video to infer that?

Do we need them all?

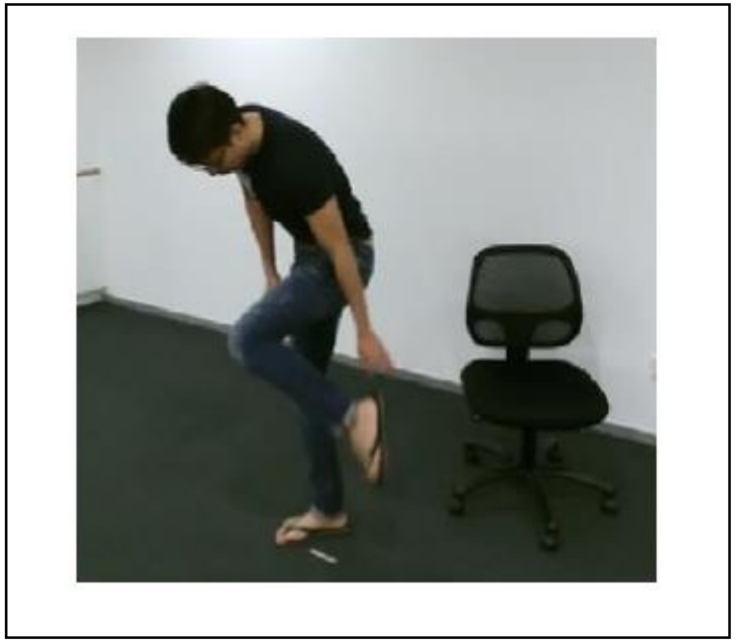
- Isn't this enough for an inference?

Focus in the Spatial space is required!

Do we need them all?

- Can you recognize this **action**?

  Wearing or taking off shoes



Do we need them all?

- Now probably you can answer!!!

- For videos, focus along the temporal space is also important.

Attention Mechanism

Idea: Focus in different parts of the input as you make/refine predictions in time

E.g.: Image Captioning

A bird flying over a body of water

Attention Mechanism

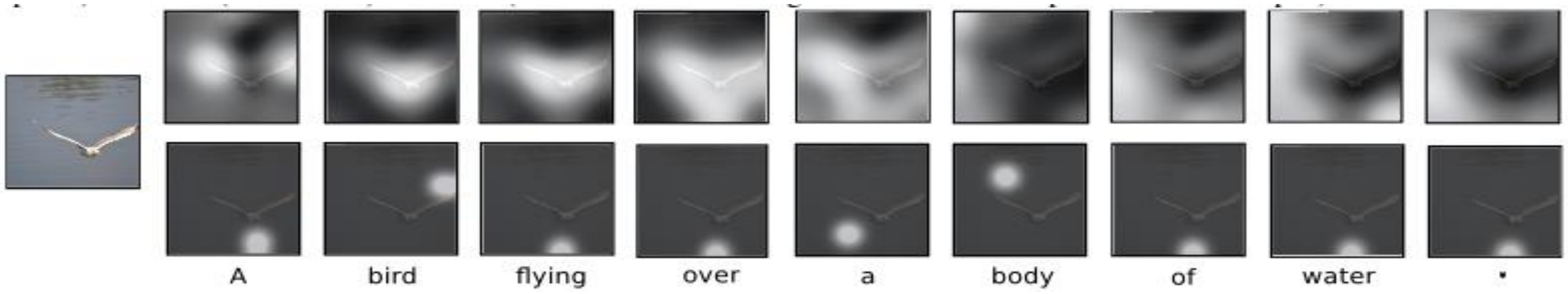
Attention mechanism

Hard attention

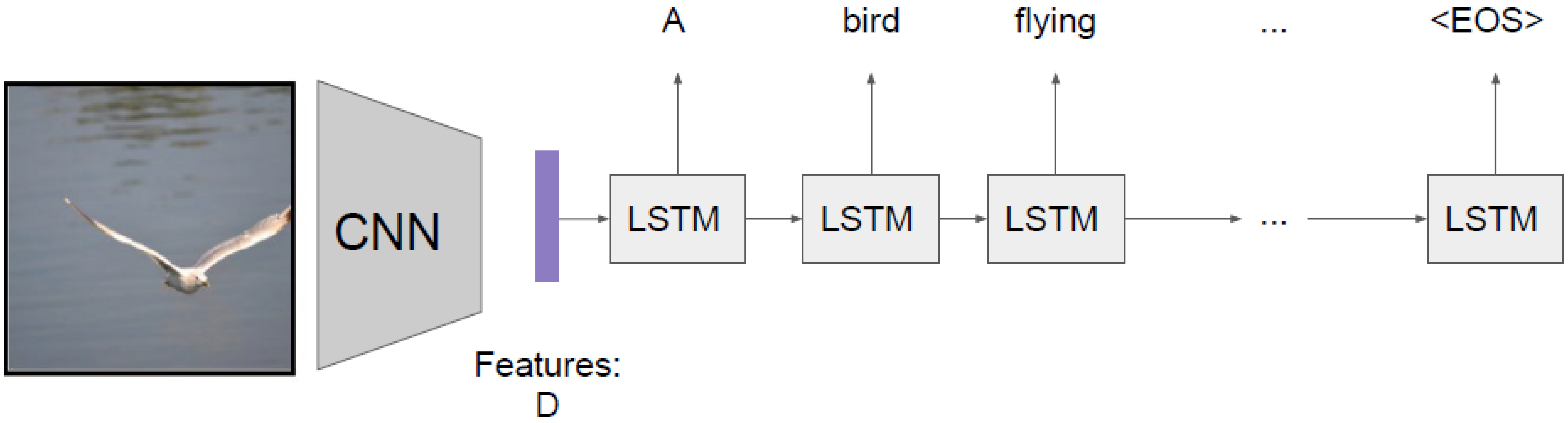
- Hard decisions while choosing parts of the input data.
- Cannot be learned easily through gradient decent (no global optimization).

Soft attention

- Weighs the RoI dynamically, taking the entire input into account.
- Can be trained end-to-end (global optimization).

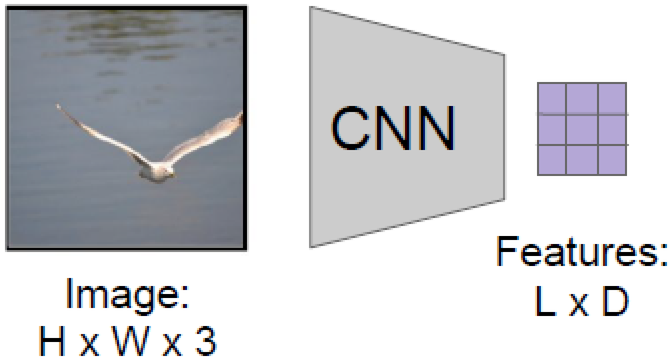


Solving image captioning

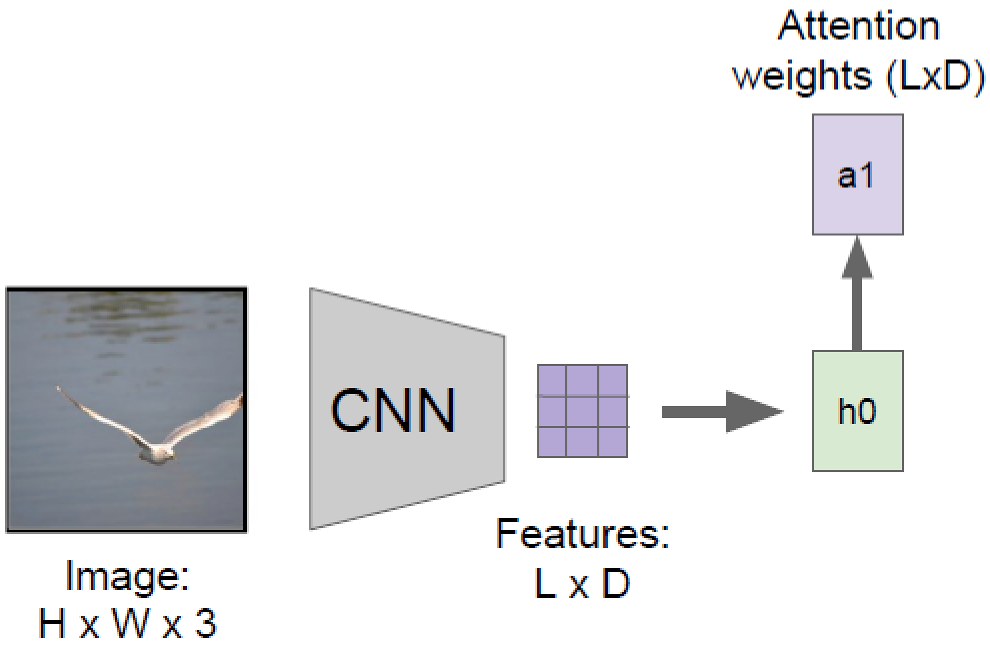


The LSTM decoder “sees” the input only at the beginning !

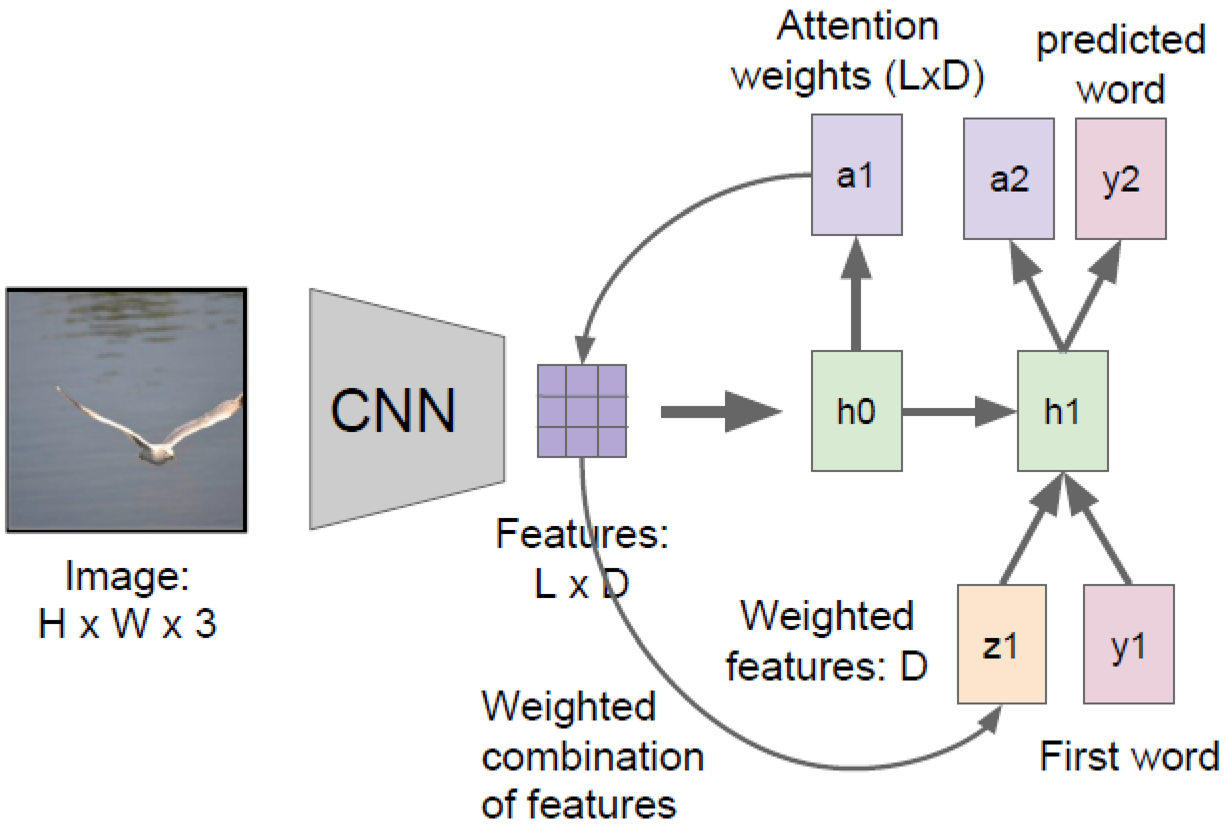
Solving image captioning with attention



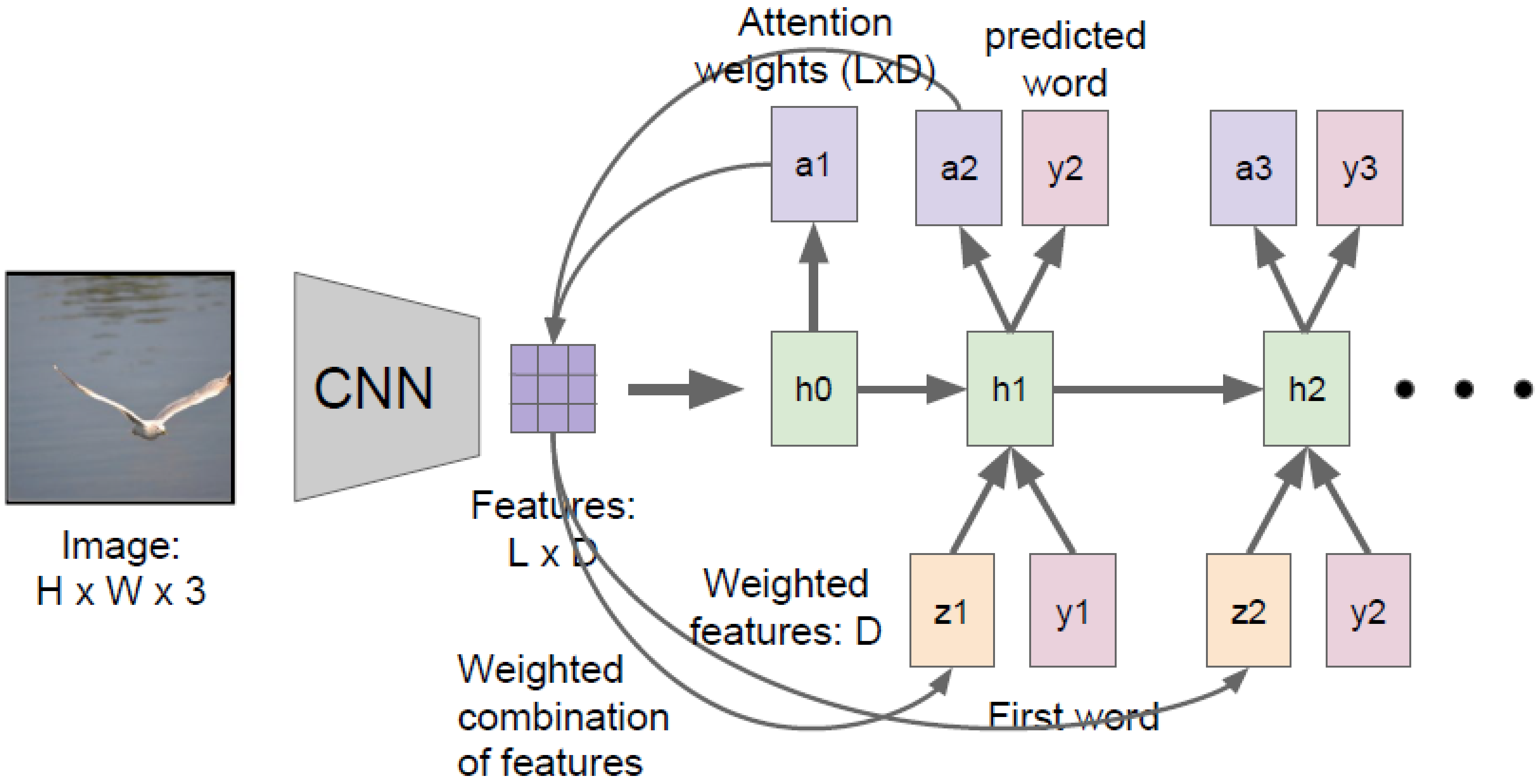
Solving image captioning with attention



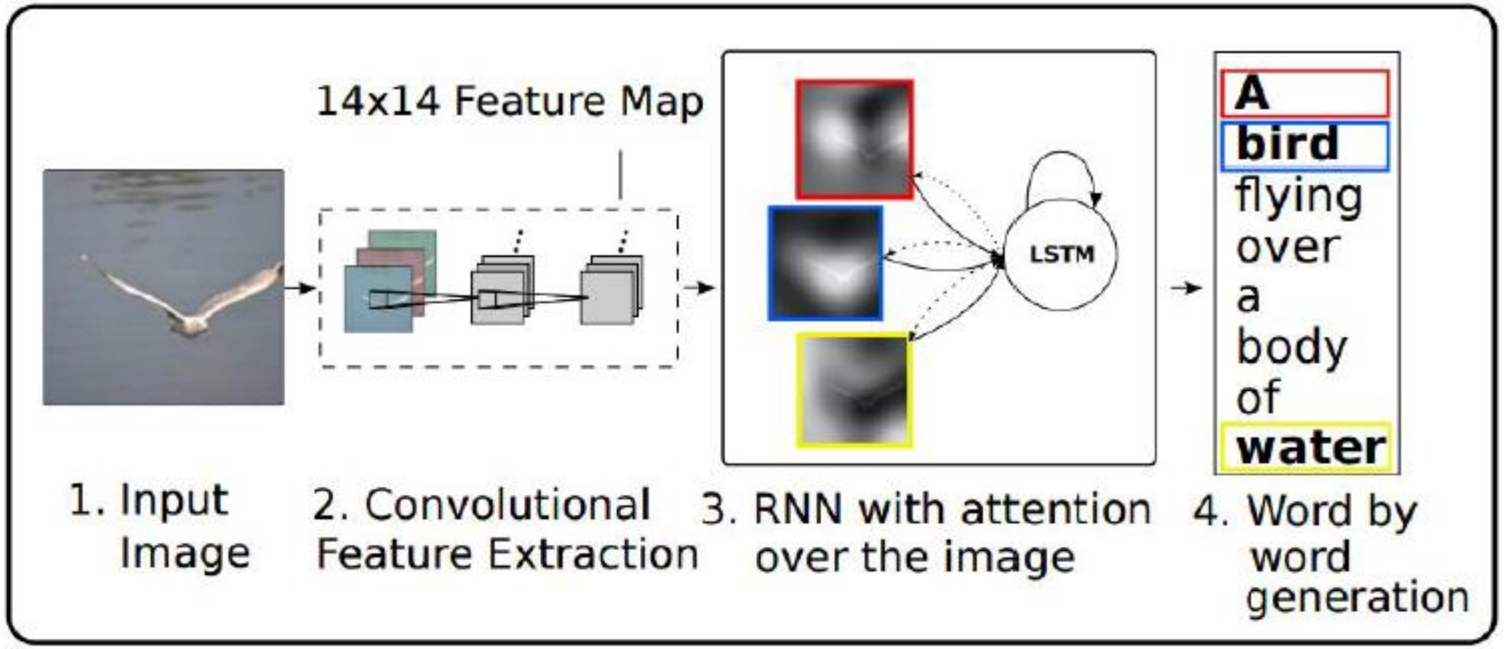
Solving image captioning with attention



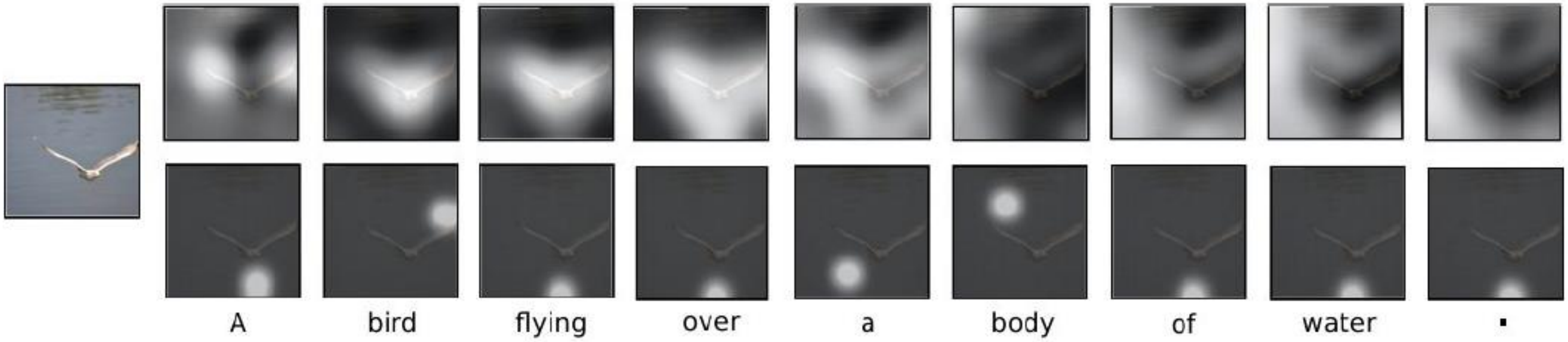
Solving image captioning with attention



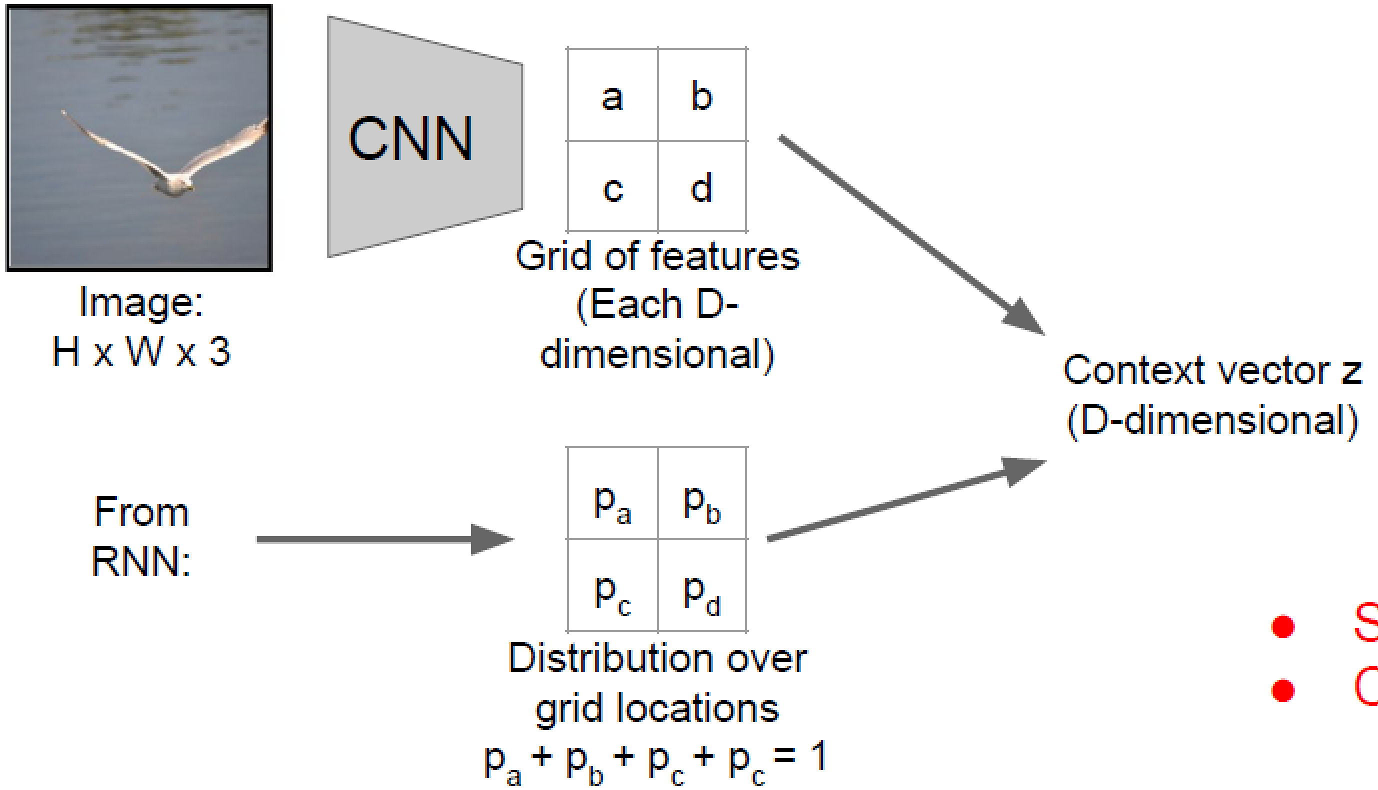
Solving image captioning with attention



Hard vs Soft attention!



Soft Attention



Soft attention:
Summarize ALL locations
$\mathbf{z} = p_a a + p_b b + p_c c + p_d d$

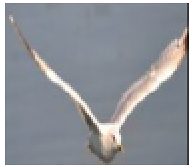
Differentiable function
Train with gradient descent

- Still uses the whole input !
- Constrained to fix grid

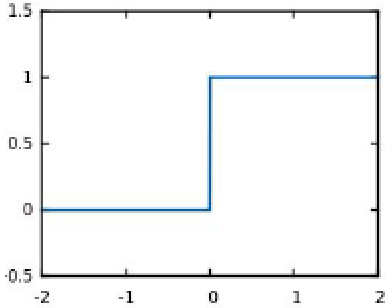
Hard Attention

Input image:
H x W x 3

Box Coordinates:
(xc, yc, w, h)



Cropped and
rescaled image:
X x Y x 3



Gradient is 0 almost everywhere
Gradient is undefined at x = 0

Hard attention:
Sample a subset
of the input

Not a differentiable function !

Can't train with backprop :(

need [reinforcement learning]

Few Popular Attention Mechanisms

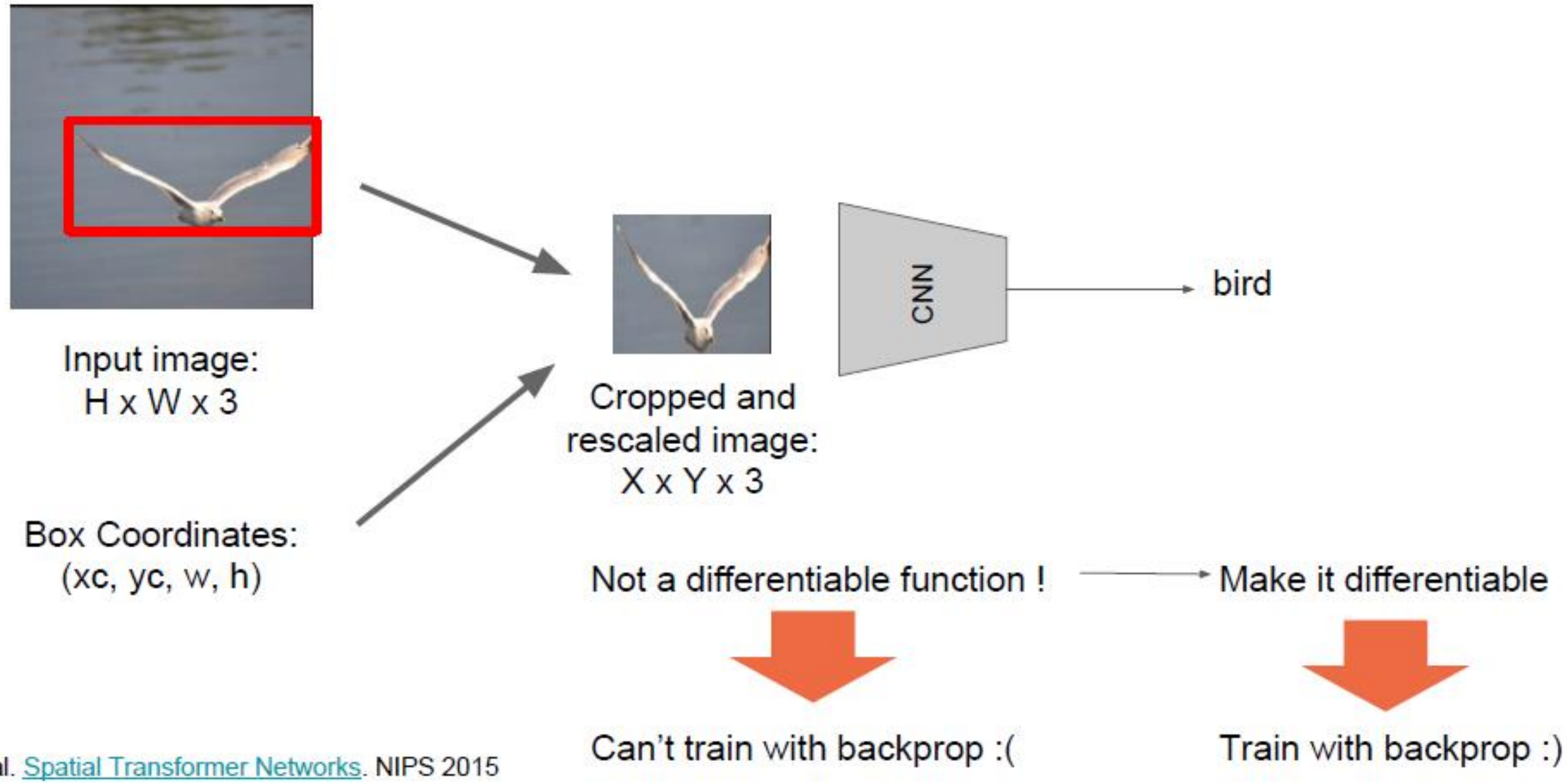
- Spatial Transformer Network (STN)

- Self-Attention

- Visual Attention for Action Recognition

Spatial Transformer Network (STN)

Motivation



Jaderberg et al. Spatial Transformer Networks. NIPS 2015

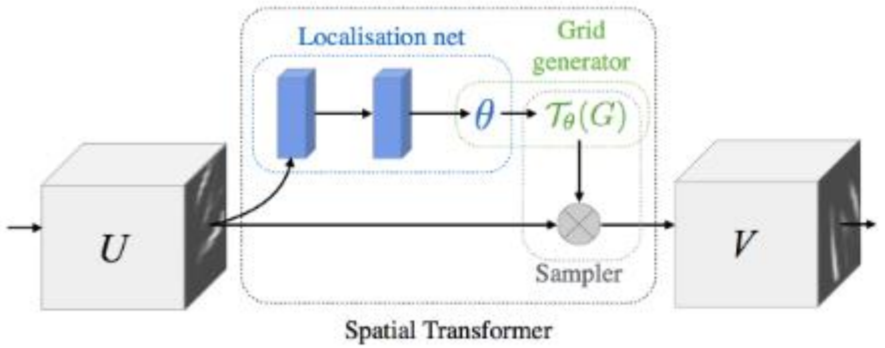
Spatial Transformer Network (STN)

Properties of STNs -

- **modular**: STNs can be inserted anywhere into existing architectures with relatively small tweaking.
- **differentiable**: STNs can be trained with backprop allowing for end-to-end training of the models they are injected in.
- **dynamic:** STNs perform active spatial transformation on a feature map for each input sample as compared to the pooling layer which acted identically for all input samples.

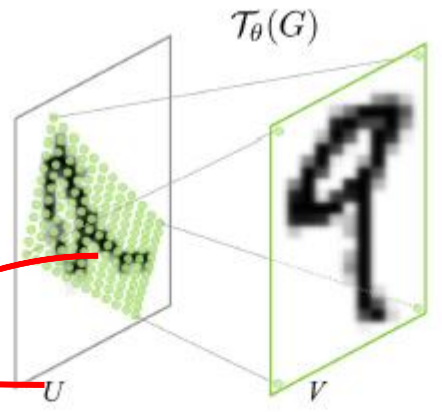
Components of STN -

1. Localisation Network
2. Grid Generator
3. Sampler



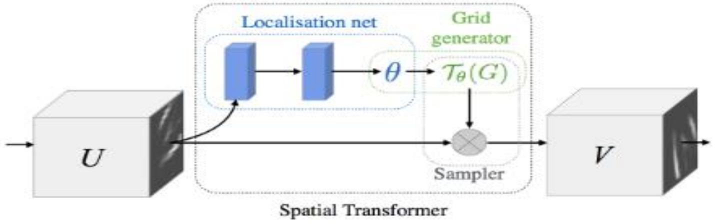
Spatial Transformer Network (STN)

Concretely, the grid generator first creates a normalized meshgrid of the same size as the input image U of shape (H, W), that is, a set of indices (x^t, y^t) that cover the whole input feature map (the subscript t here stands for target coordinates in the output feature map). Then, since we're applying an affine transformation to this grid and would like to use translations, we proceed by adding a row of ones to our coordinate vector to obtain its homogeneous equivalent. This is the little trick we also talked about last week. Finally, we reshape our 6 parameter θ to a 2x3 matrix and perform the following multiplication which results in our desired parametrised sampling grid.

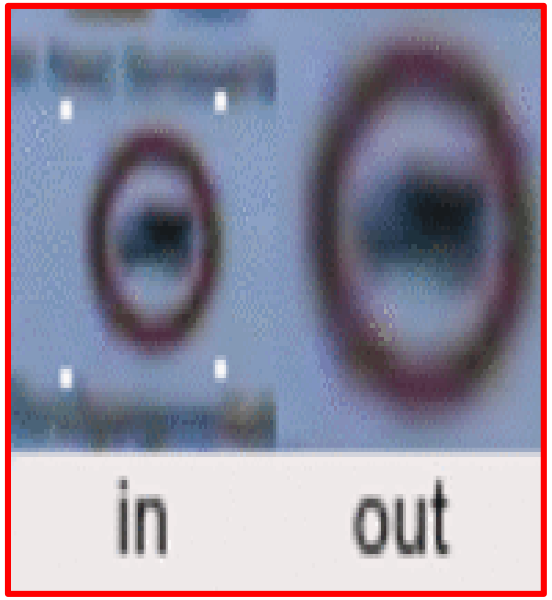


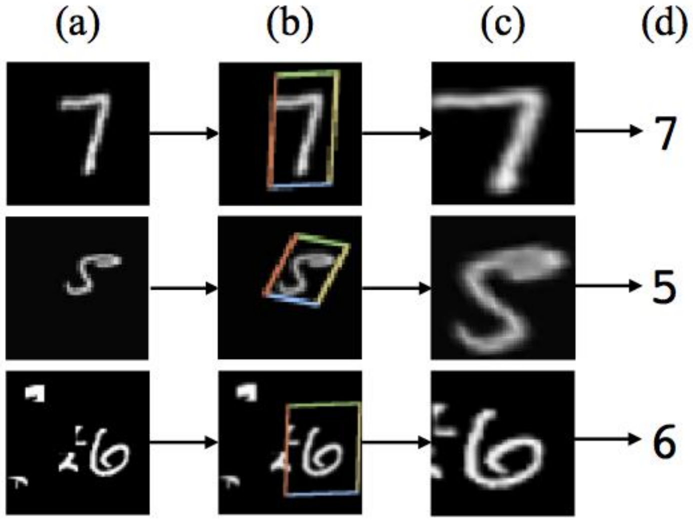
$$\begin{bmatrix} x^s \\ y^s \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{bmatrix} x^t \\ y^t \\ 1 \end{bmatrix}$$

The column vector $\begin{bmatrix} x^s \\ y^s \end{bmatrix}$ consists in a set of indices that tell us where we should sample our input to obtain the desired transformed output.



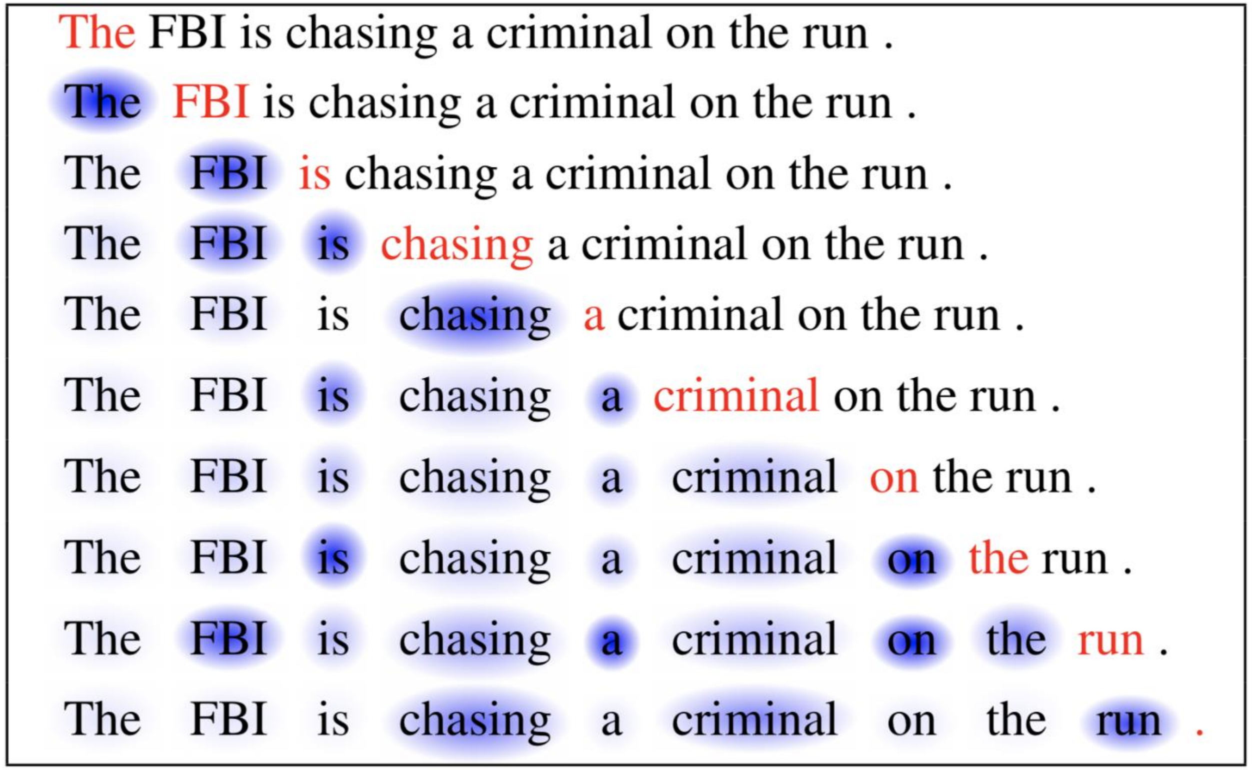
Results from STN





Self-Attention

Self-attention, also known as **intra-attention**, is an attention mechanism relating different positions of a single sequence in order to compute a representation of the same sequence. It has been shown to be very useful in machine reading, abstractive summarization, or image description generation.

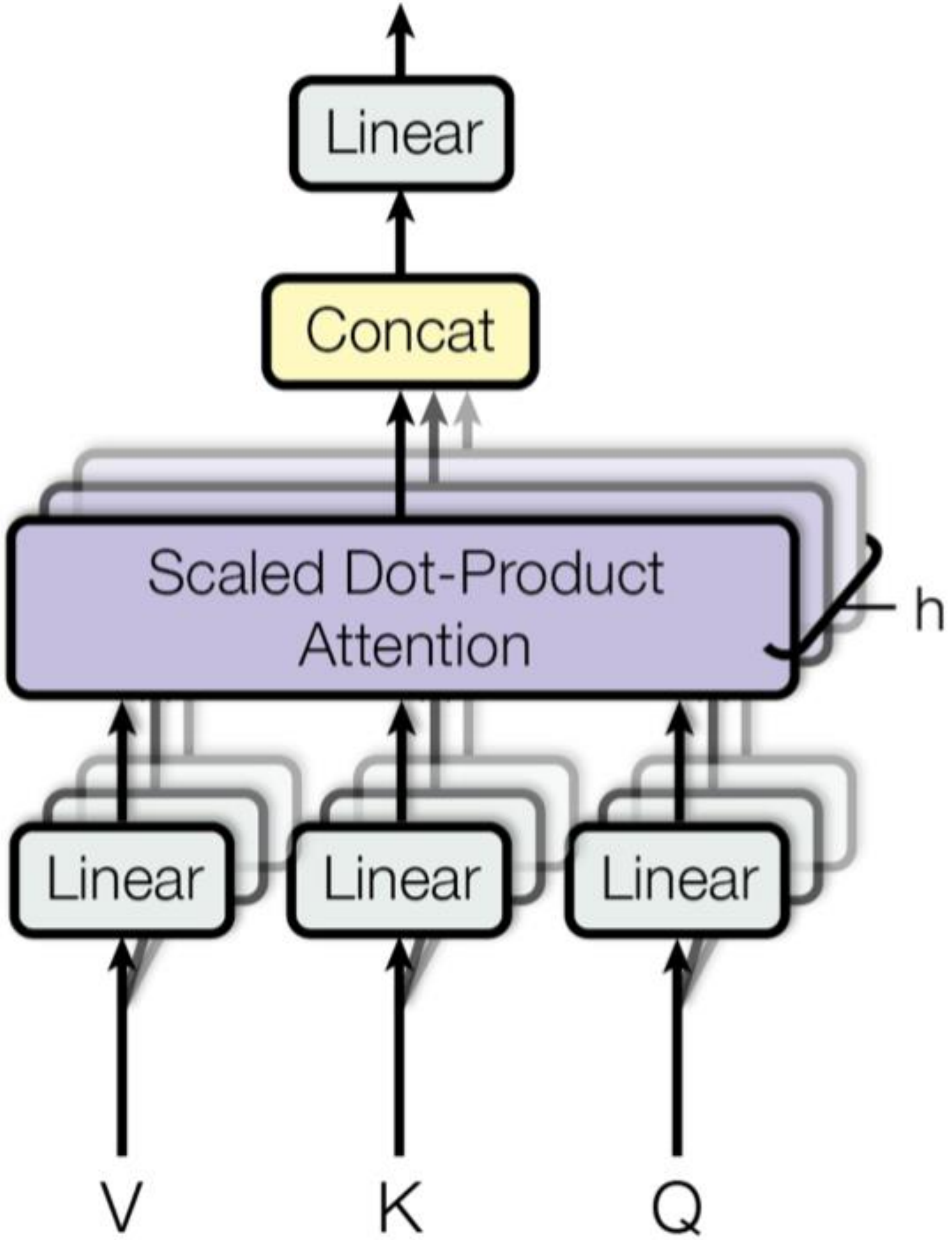


Self-Attention

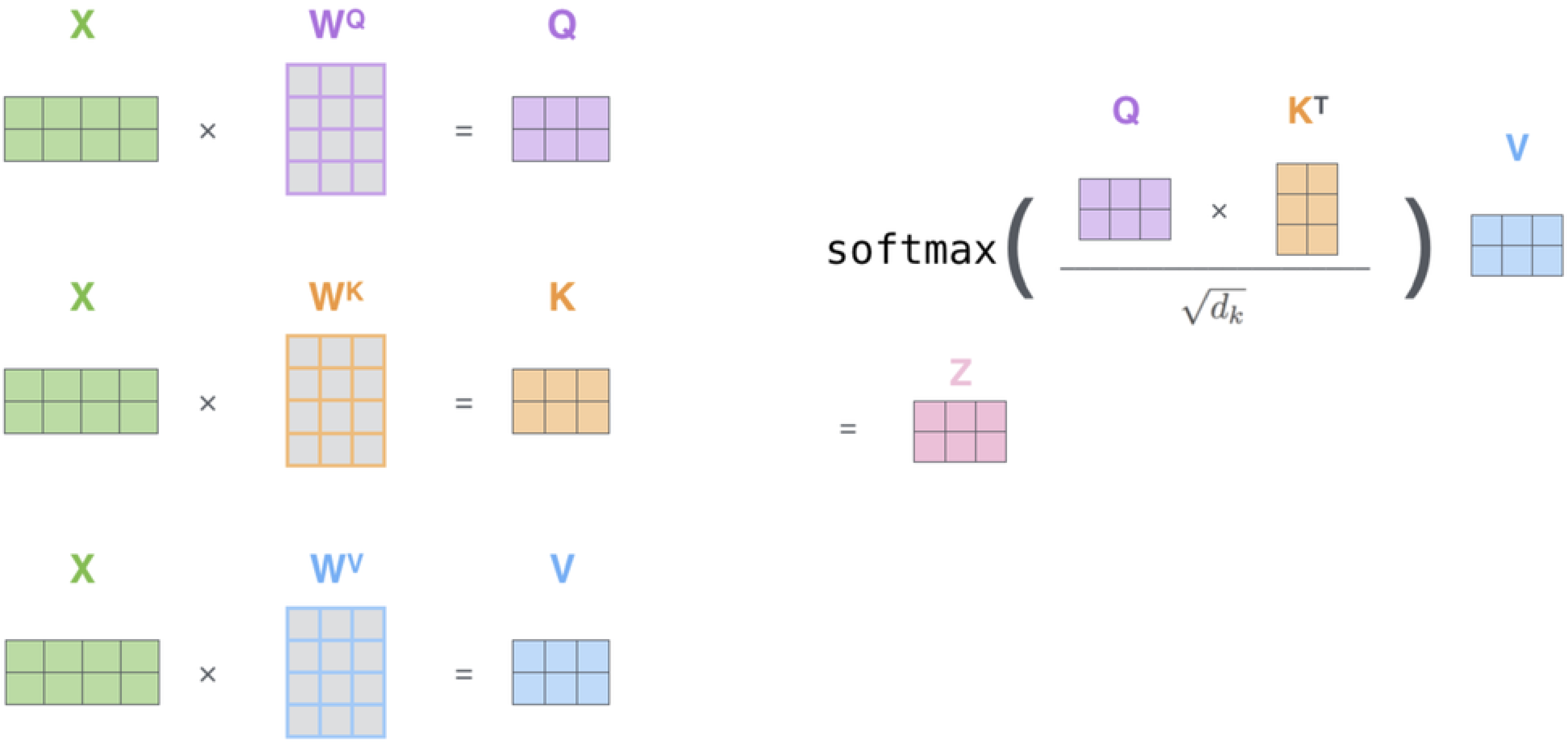
The major component in the transformer is the unit of *multi-head self-attention mechanism*. The transformer views the encoded representation of the input as a set of **key-value** pairs, $(\mathbf{K}, \mathbf{V})$, both of dimension n (input sequence length); in the context of NMT, both the keys and values are the encoder hidden states. In the decoder, the previous output is compressed into a **query** ($\mathbf{Q}$ of dimension m) and the next output is produced by mapping this query and the set of keys and values.

The transformer adopts the scaled dot-product attention: the output is a weighted sum of the values, where the weight assigned to each value is determined by the dot-product of the query with all the keys:

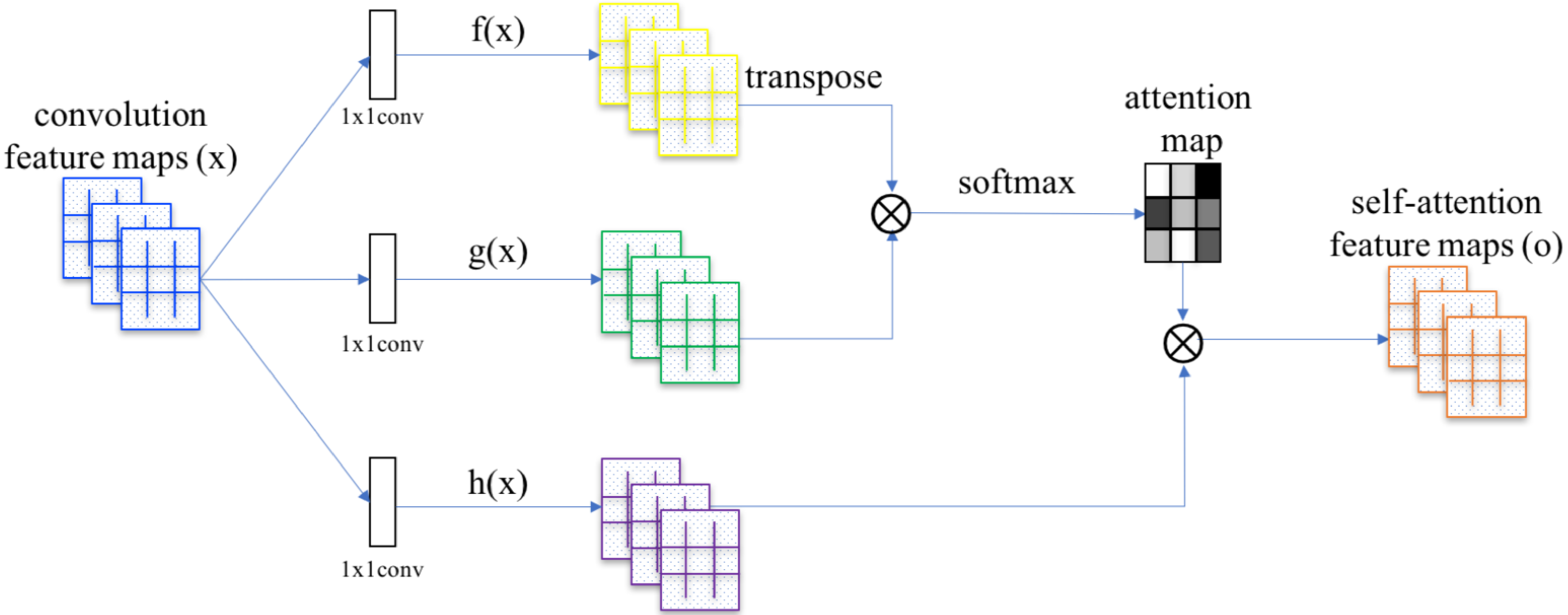
$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{n}})\mathbf{V}$$



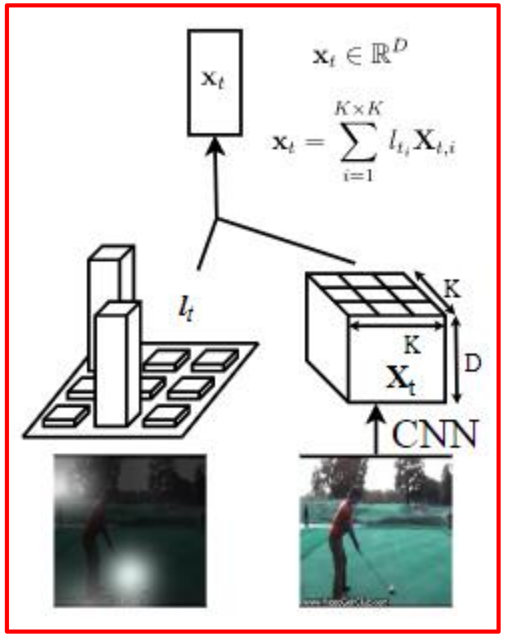
Self-Attention

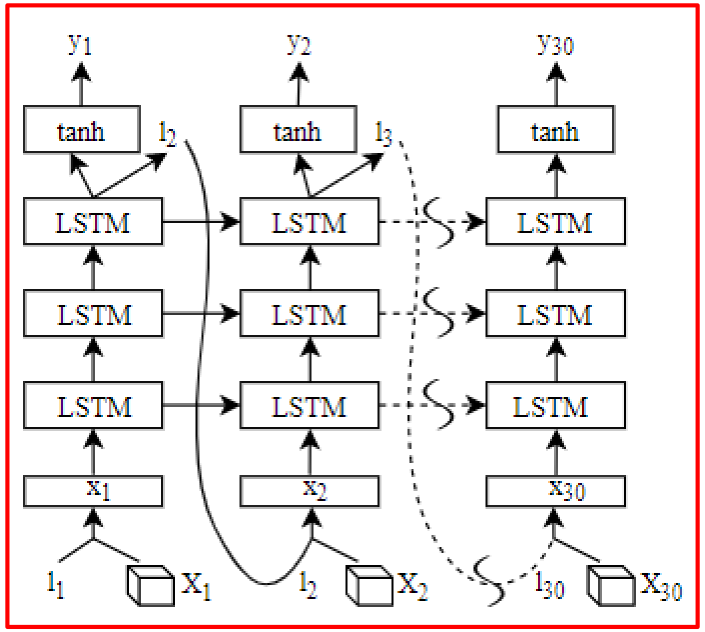


Self-Attention



Visual Attention for Action Recognition

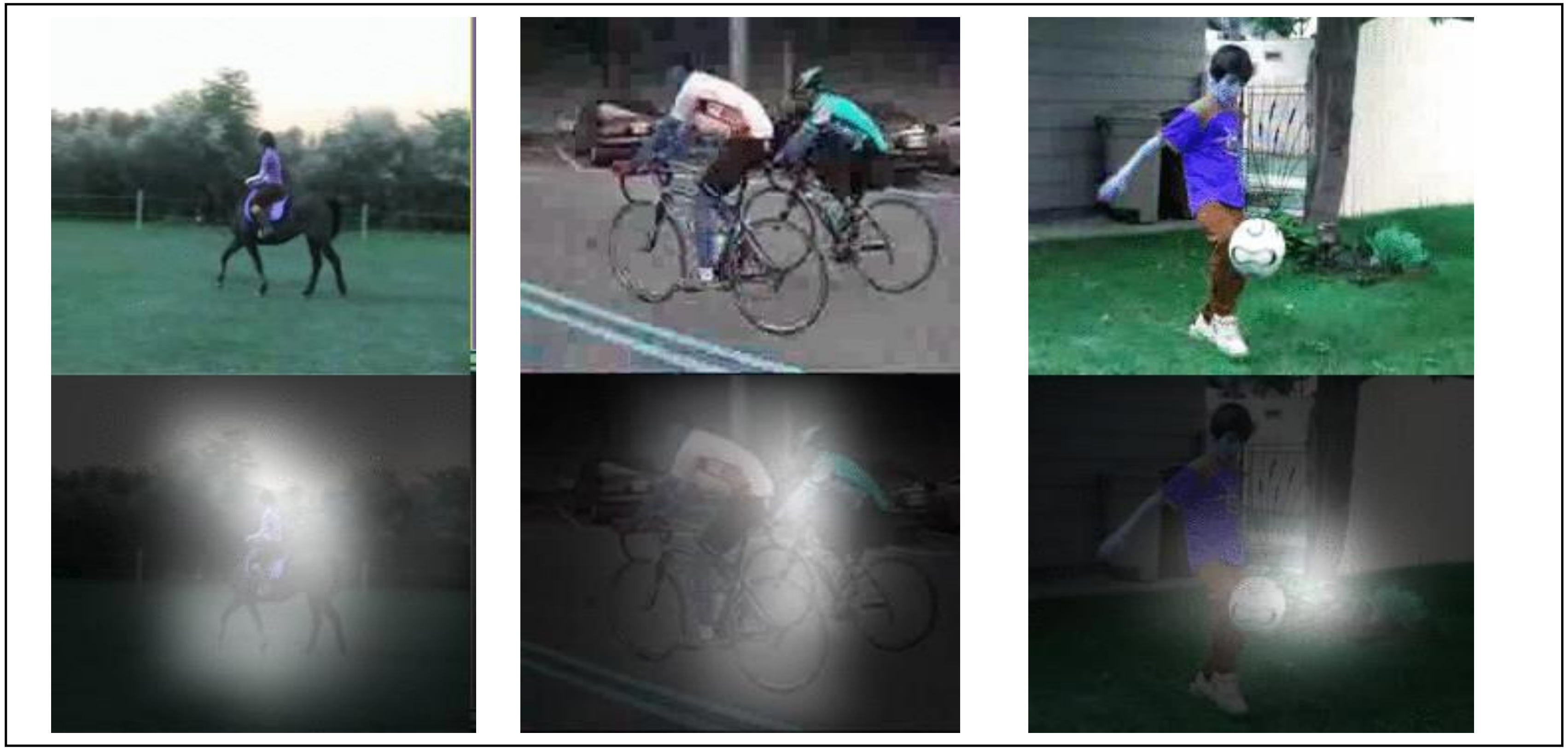




At every time step, predict the next important region in the feature map

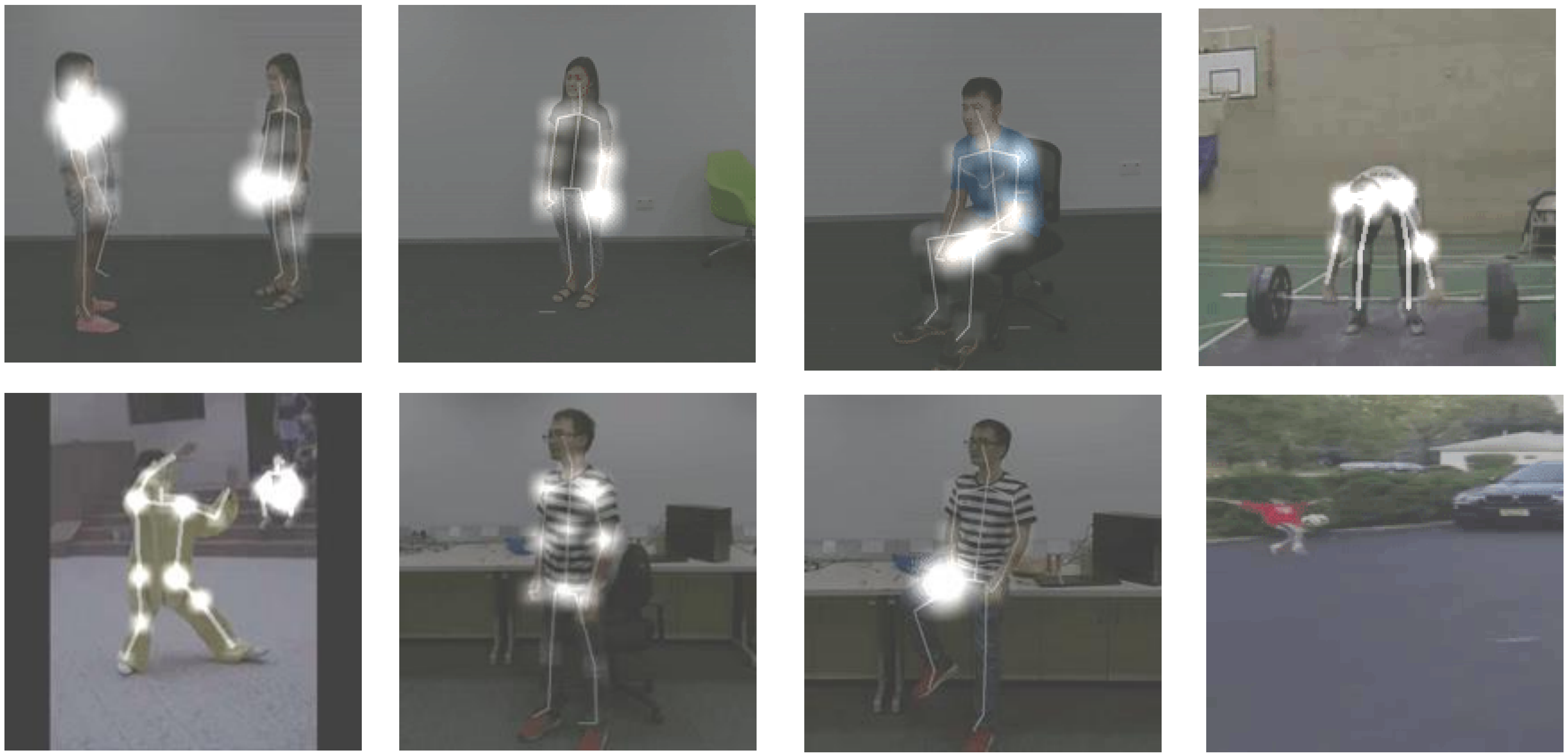
Example videos of soft-attention for Action Recognition in the state-of-the-art

Sharma et al., (ICLRW 2015)



Example videos of soft-attention for Action Recognition in the state-of-the-art

Xiaong et al., (AAAI 2018)



Conclusion

- Attention Mechanism is still an open research problem with issues like
  - How to incorporate attention in earlier layers?
  - How to adapt STN for videos?
- How to speculate attention for future events to occur????

Other important topics not covered!

- Cross-view Action Recognition
- Action Detection
- Weakly supervised Action Detection
- Domain adaptation for cross-data Action Recognition

References

- Deep Learning for Computer Vision, Summer Seminar UPC TelecomBCN, 4-6 July 2016 (Attention Models)
- Kevin's Blog : Deep Learning Paper Implementations: Spatial Transformer Networks - Part II
- Action Recognition using Visual Attention (S. Sharma)

Thanks ….
All the best for the final Presentation!

e-mail: srijan.das@inria.fr