CS772 Project Presentation

Image Captioning

Rupesh Yadav | 21i190004
Shoaib Ahmad | 21i190012
Aakash Roy | 21i190007
Srija Mukherjee | 21i190013

Outline

- Problem considered
- Motivation
- Applications
- Previous Works
- “Show, attend and tell”
- Experiments
- Results
- Conclusions
- Future Work
- References

Motivation

- An immense amount of success has been seen in the field of language generation models, machine translation and object detection.
- Image captioning is one of the amalgamations of these fields where many different techniques have been used simultaneously to get a good model.
- “Attention” has helped in the task but has drawbacks which are addressed by using two different of mechanisms of attention : “hard” and “soft”

Previous Work

1. Prior to the use of NNs, two main approaches were dominant:
   a. Generating caption templates filled in based on object detections and attribute discovery
   b. Retrieving similar captioned images and modifying retrieved captions to fit the query

<u>Moving to Neural Network based works</u>

- ❖ Many methods for this task are based on RNNs and inspired by the successful use of sequence to sequence training with NNs for machine translation. [1]
- ❖ One major reason image caption generation is well suited to the encoder-decoder framework of machine translation is because it is analogous to “translating” an image to a sentence.

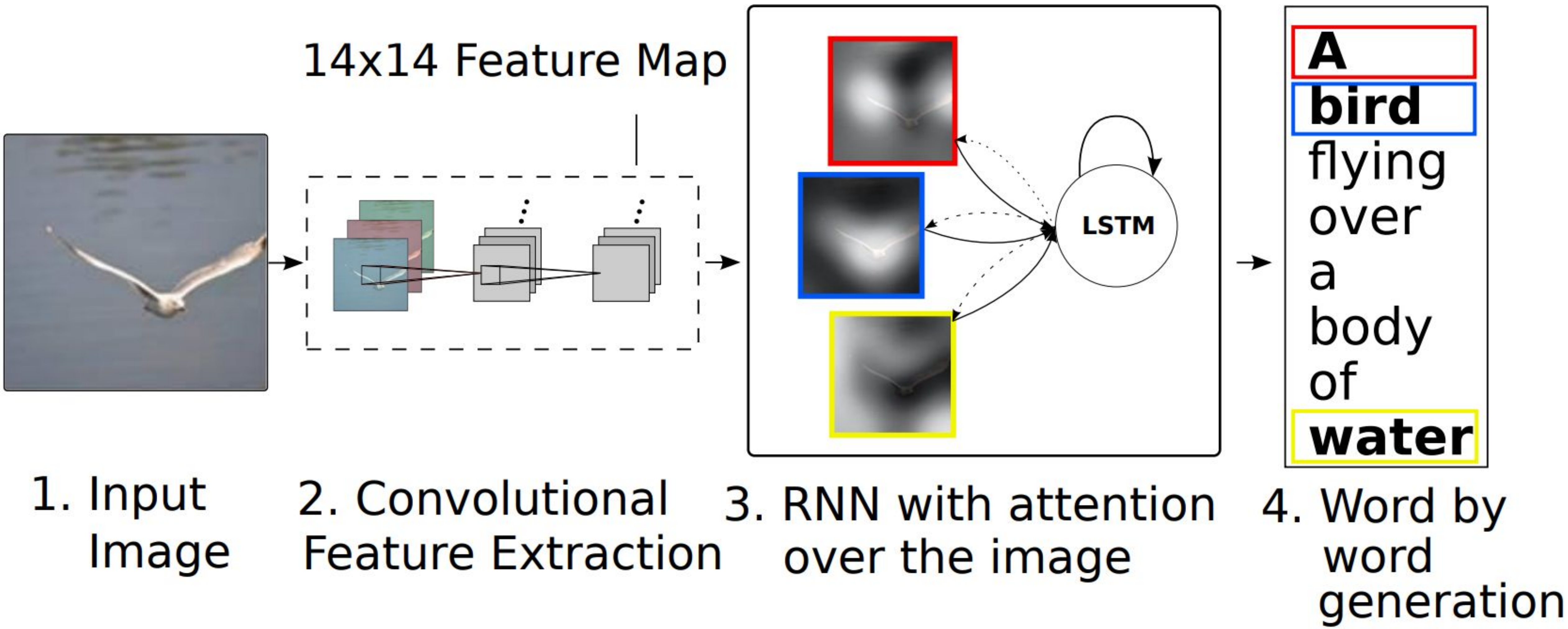
[1] Cho et al., 2014; Bahdanau et al., 2014; Sutskever et al., 2014

Previous Work

- ❖ Kiros et al. (2014a) were the first to use neural network for this task.
- ❖ Mao et al. (2014) replaced a feed-forward neural language model with a recurrent one
  - ➢ Above two models see the image at each time step of the output word sequence
- ❖ Vinyals et al. (2014) and Donahue et al. (2014) used LSTM RNNs for their models
  - ➢ show the image to the RNN at the beginning

- The discussed attention framework goes beyond "objectness" and learns to attend to abstract concepts
- In particular however, this idea work directly extends the work of Bahdanau et al. (2014); Mnih et al. (2014); Ba et al. (2014).

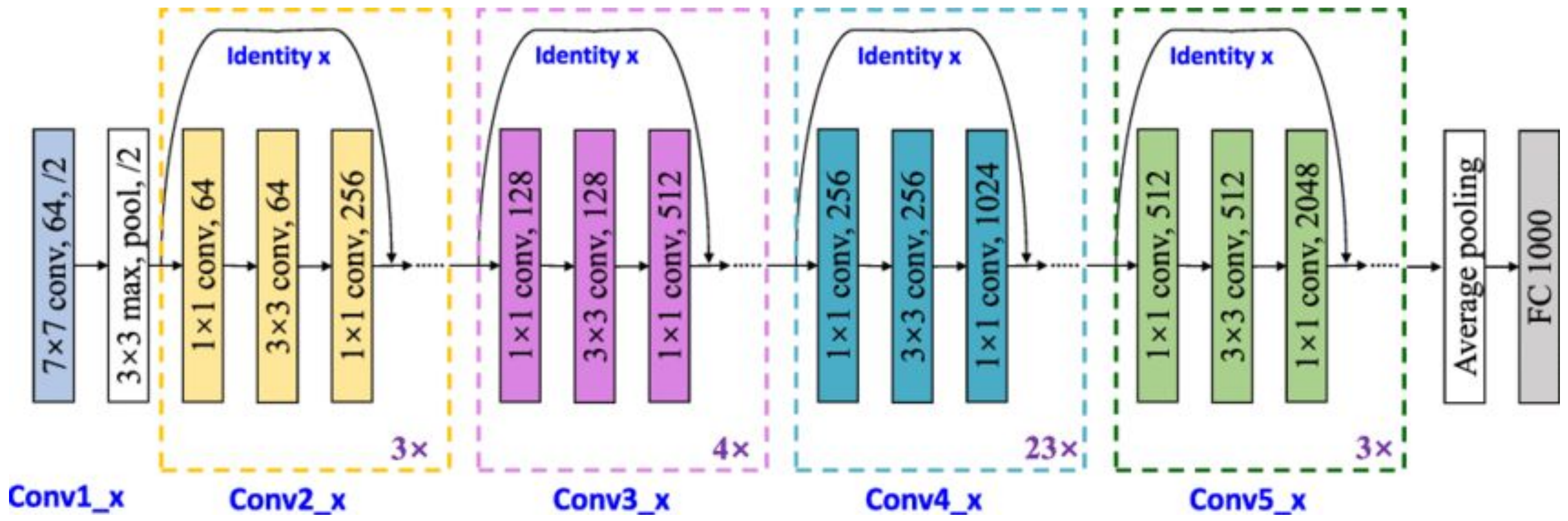
Model



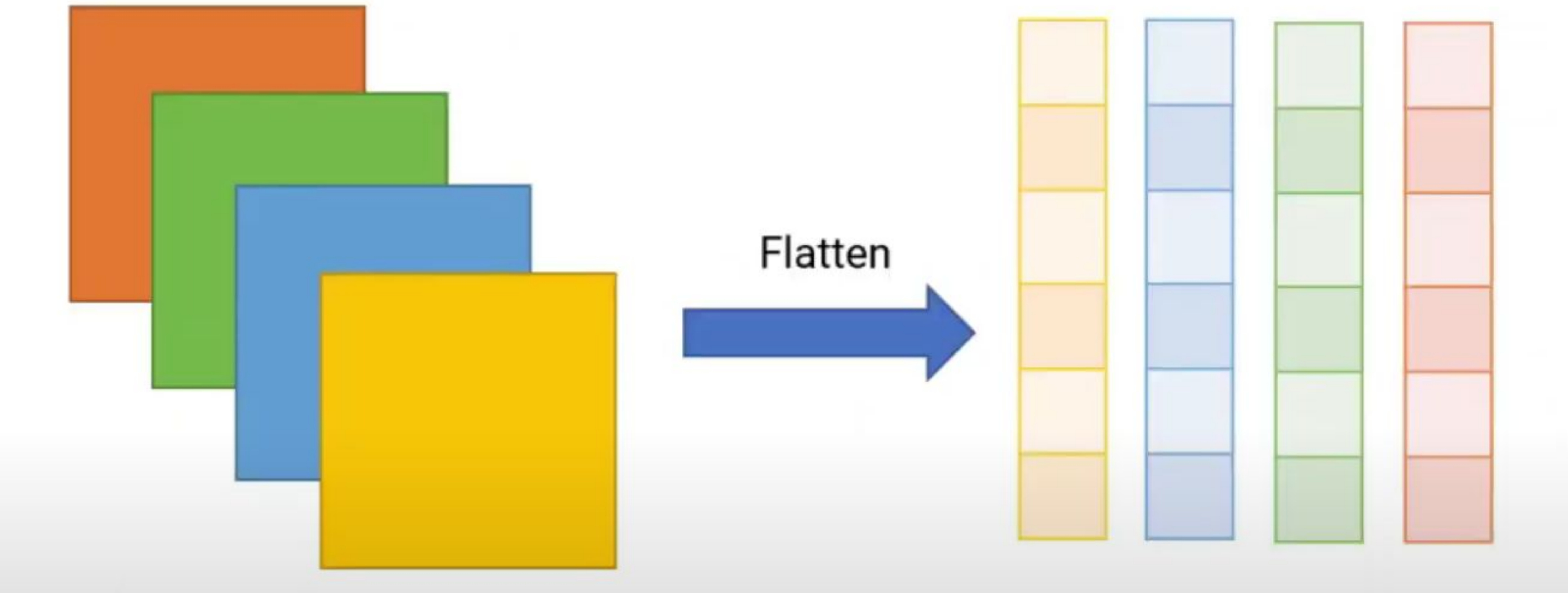
14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Convolutional Feature Extraction

ResNet101



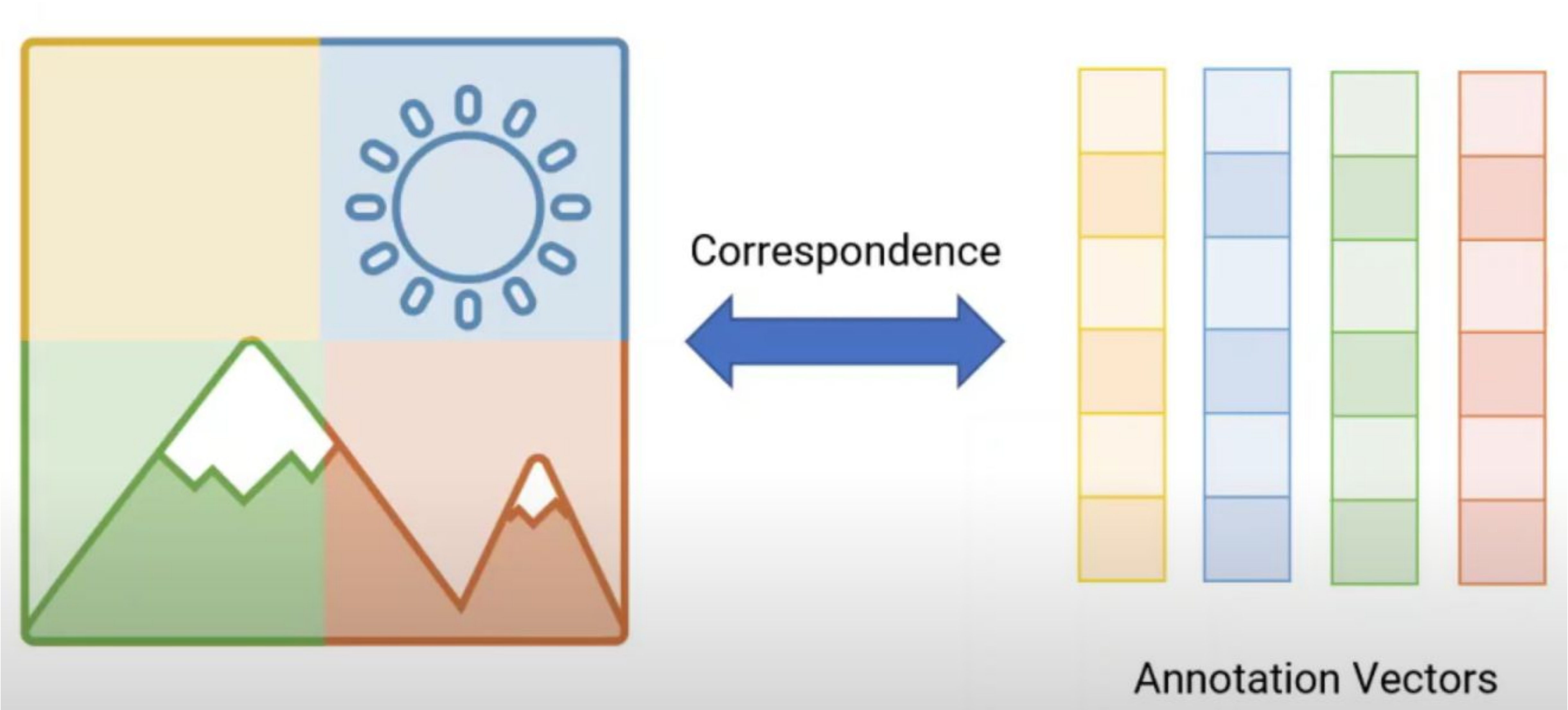
Convolutional Feature Extraction



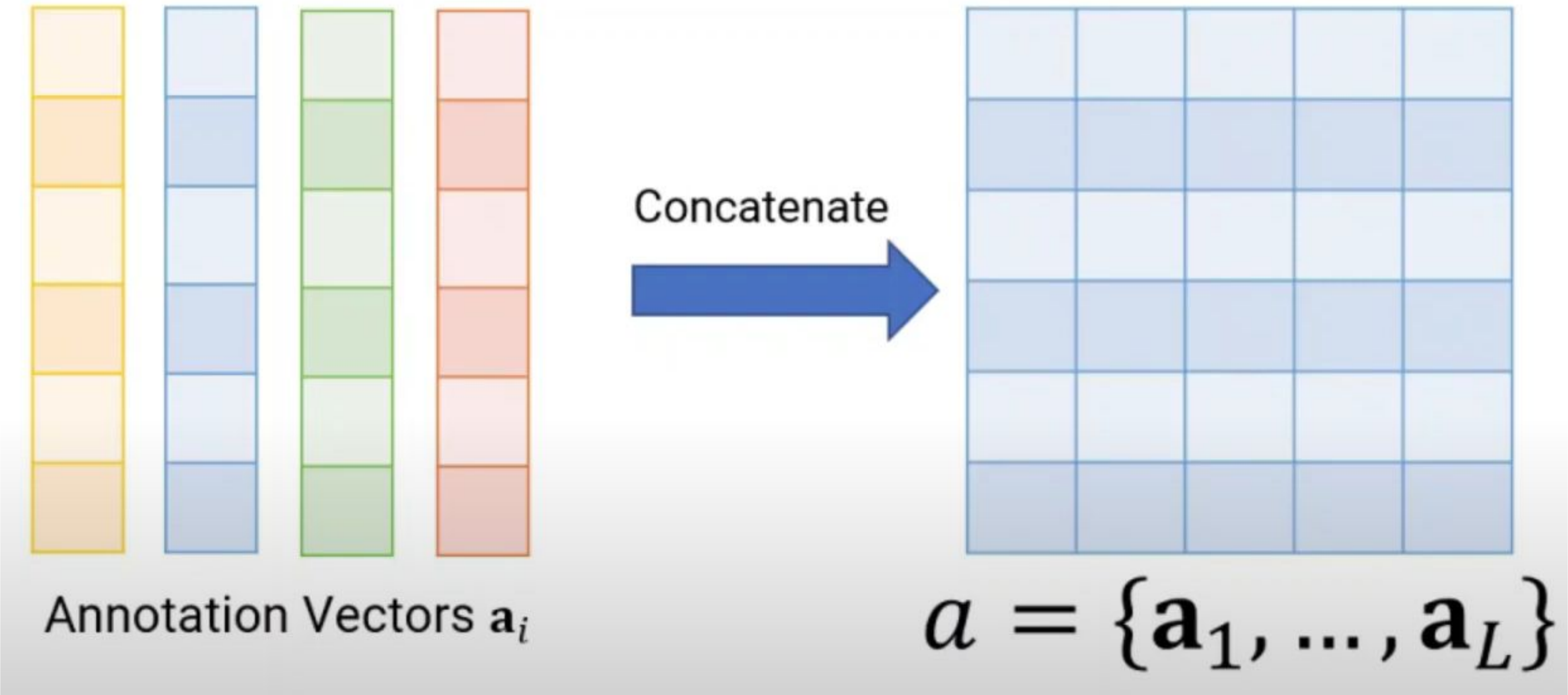
14*14*2048 feature maps

Annotation vectors

Convolutional Feature Extraction

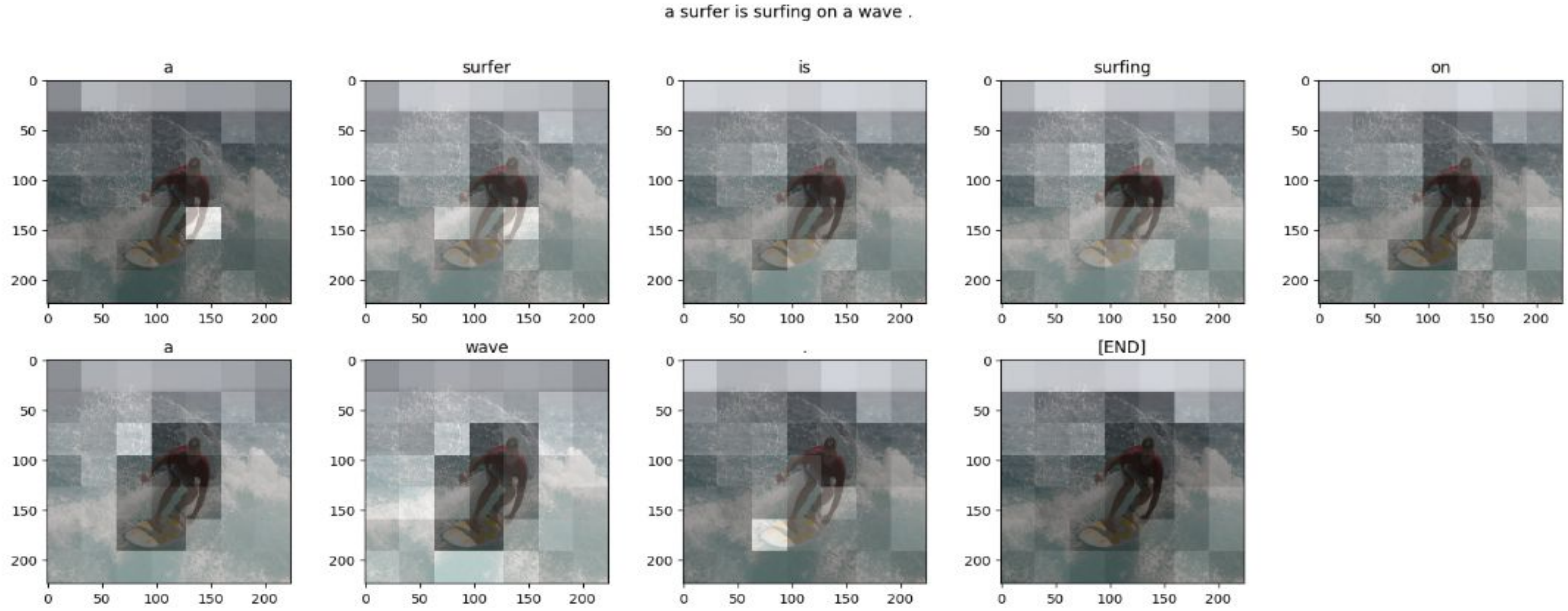


Convolutional Feature Extraction

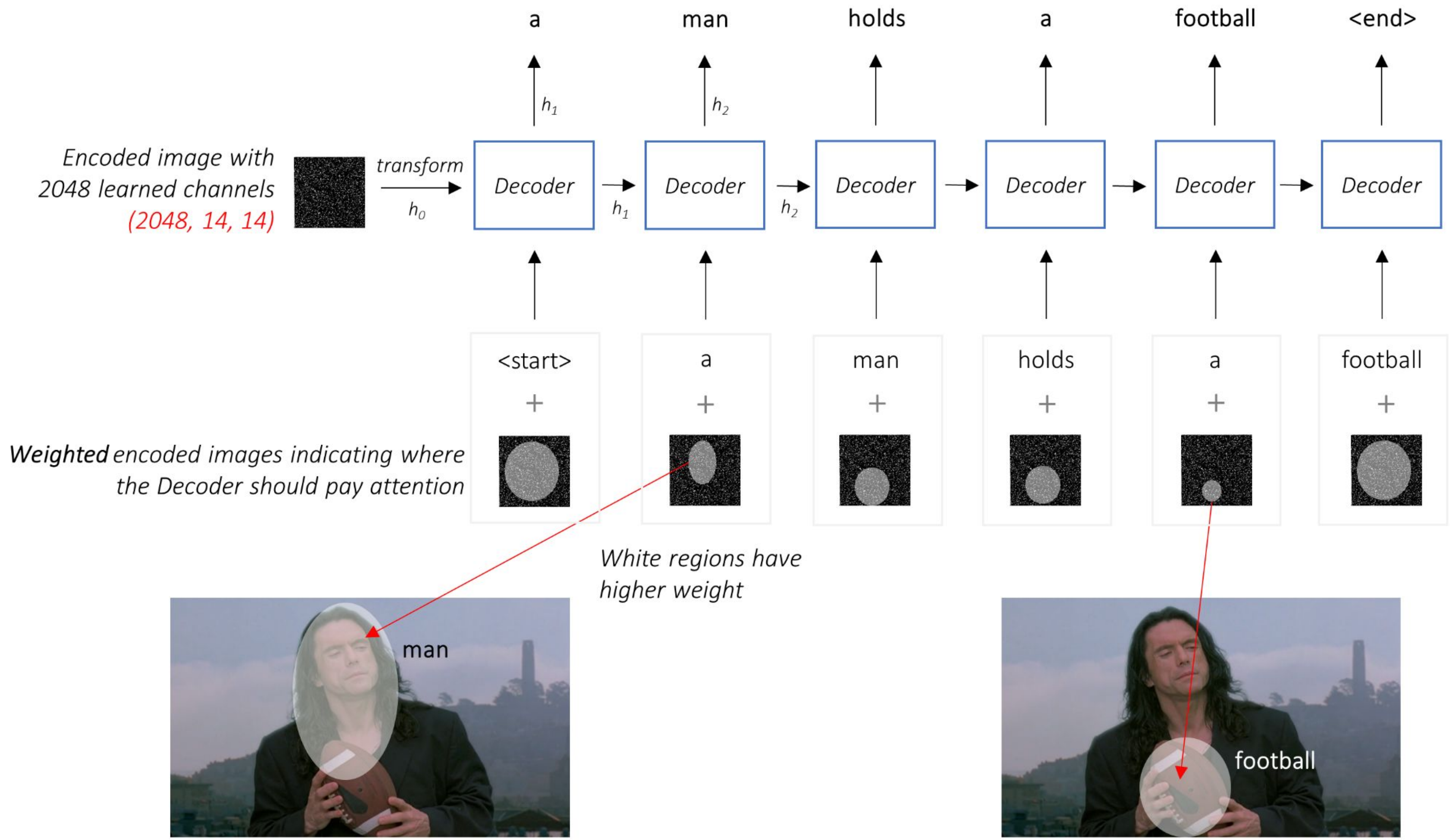


Attention Mechanism

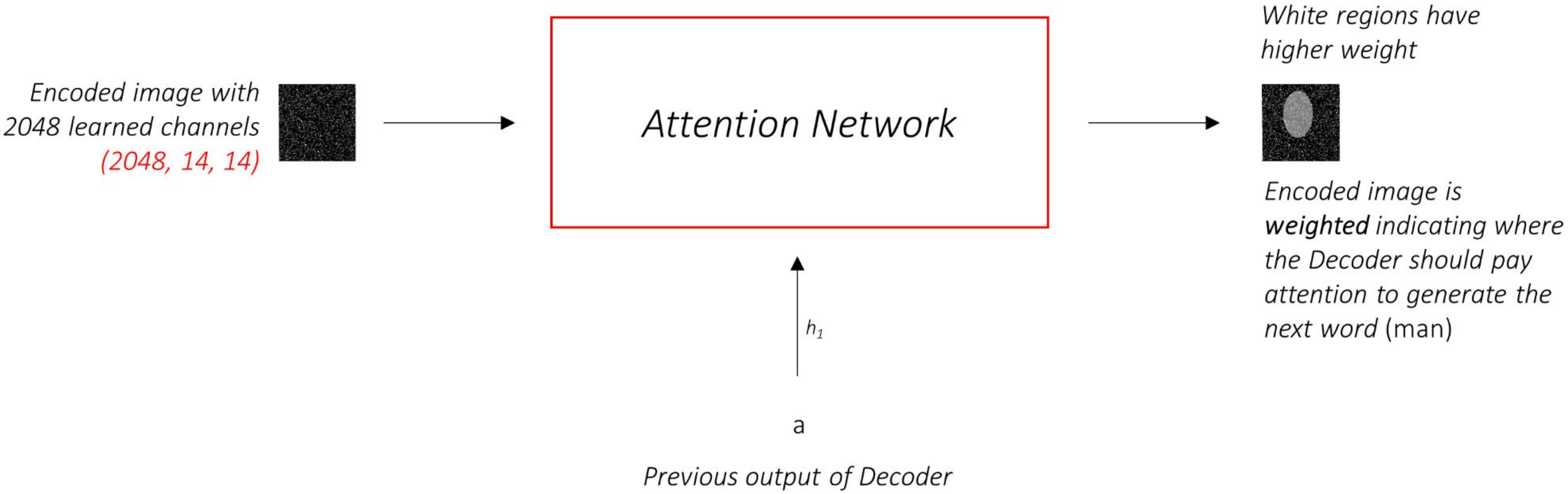
The action of selectively concentrate on few things, while ignoring others in deep neural networks.



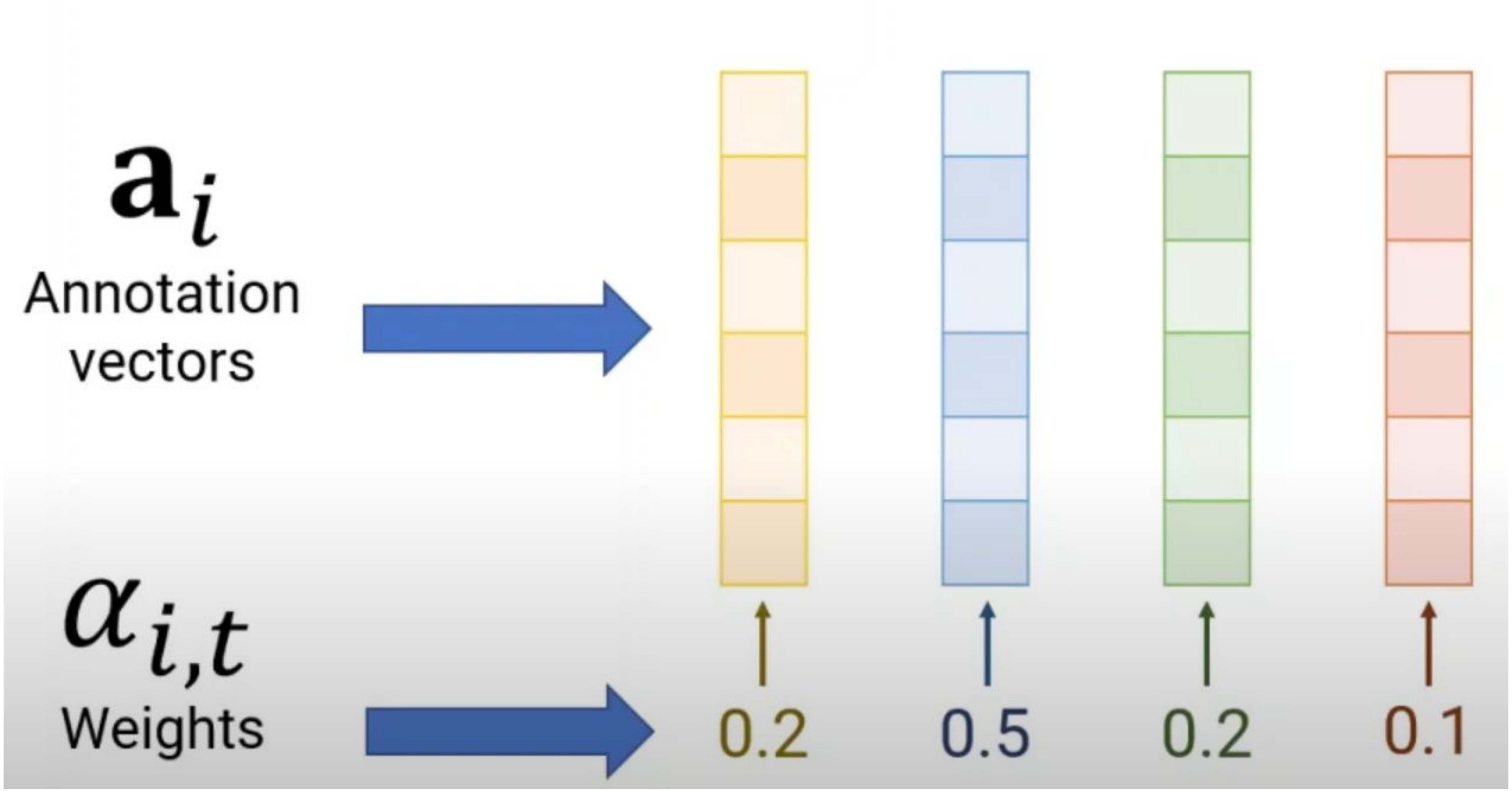
Idea of using attention



Attention network

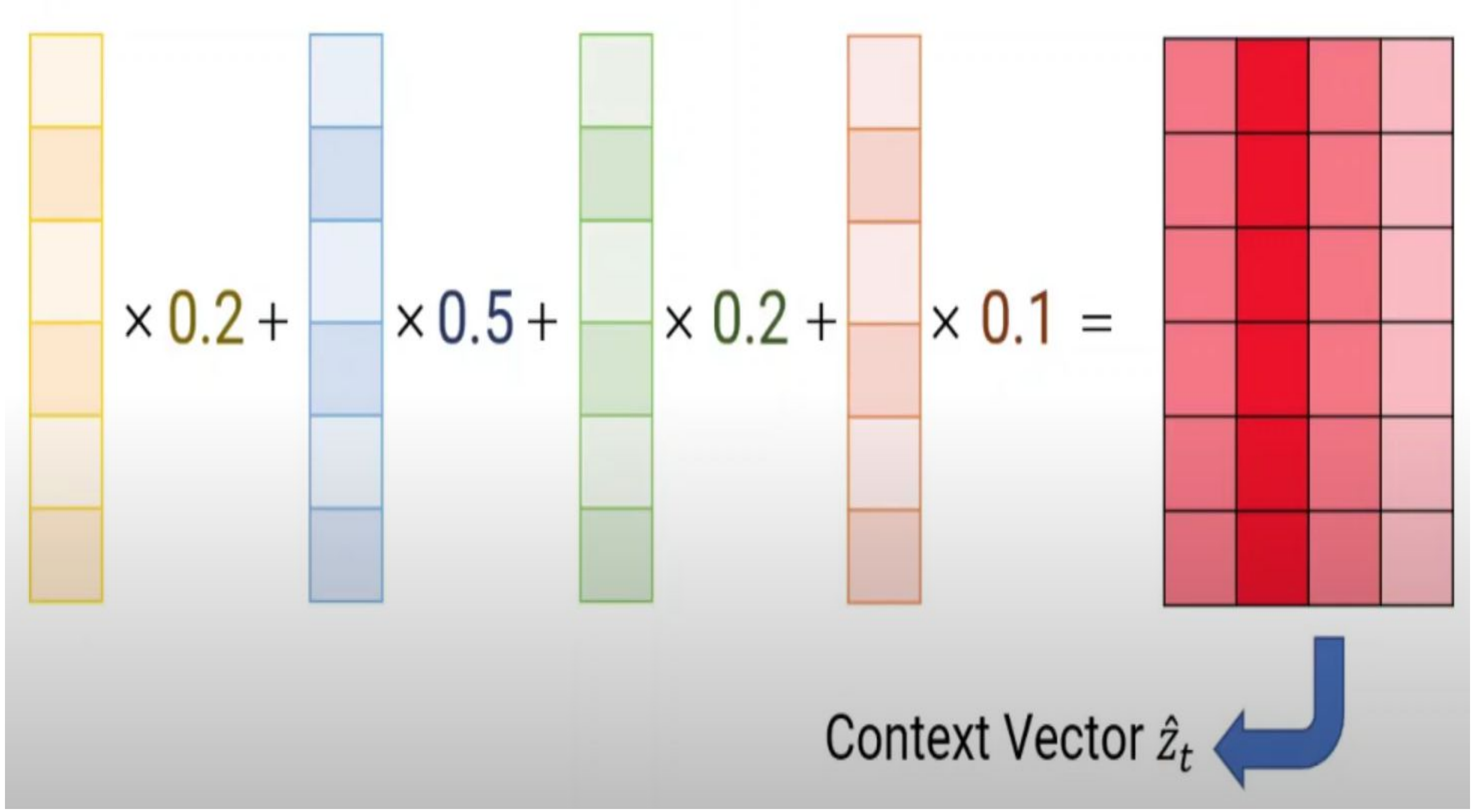


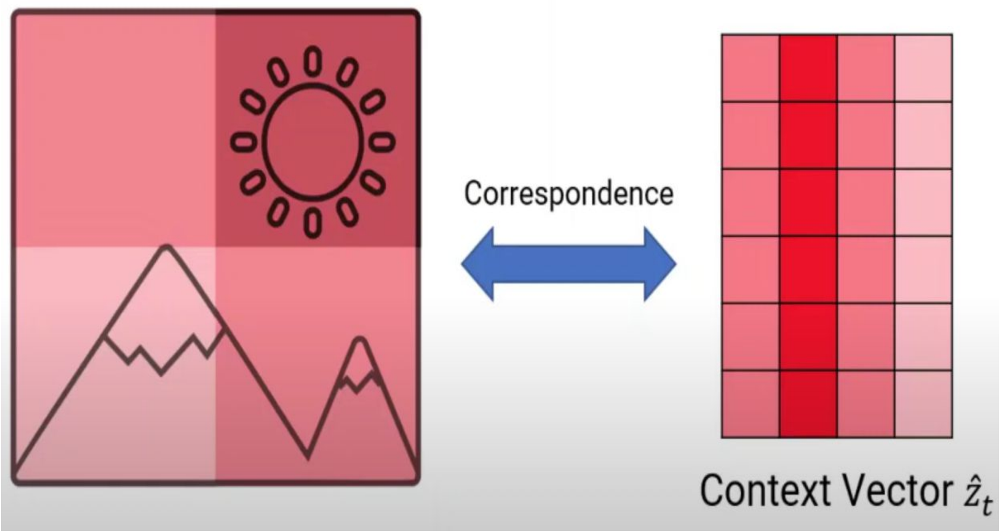
Attention : Soft and Hard



Soft Attention

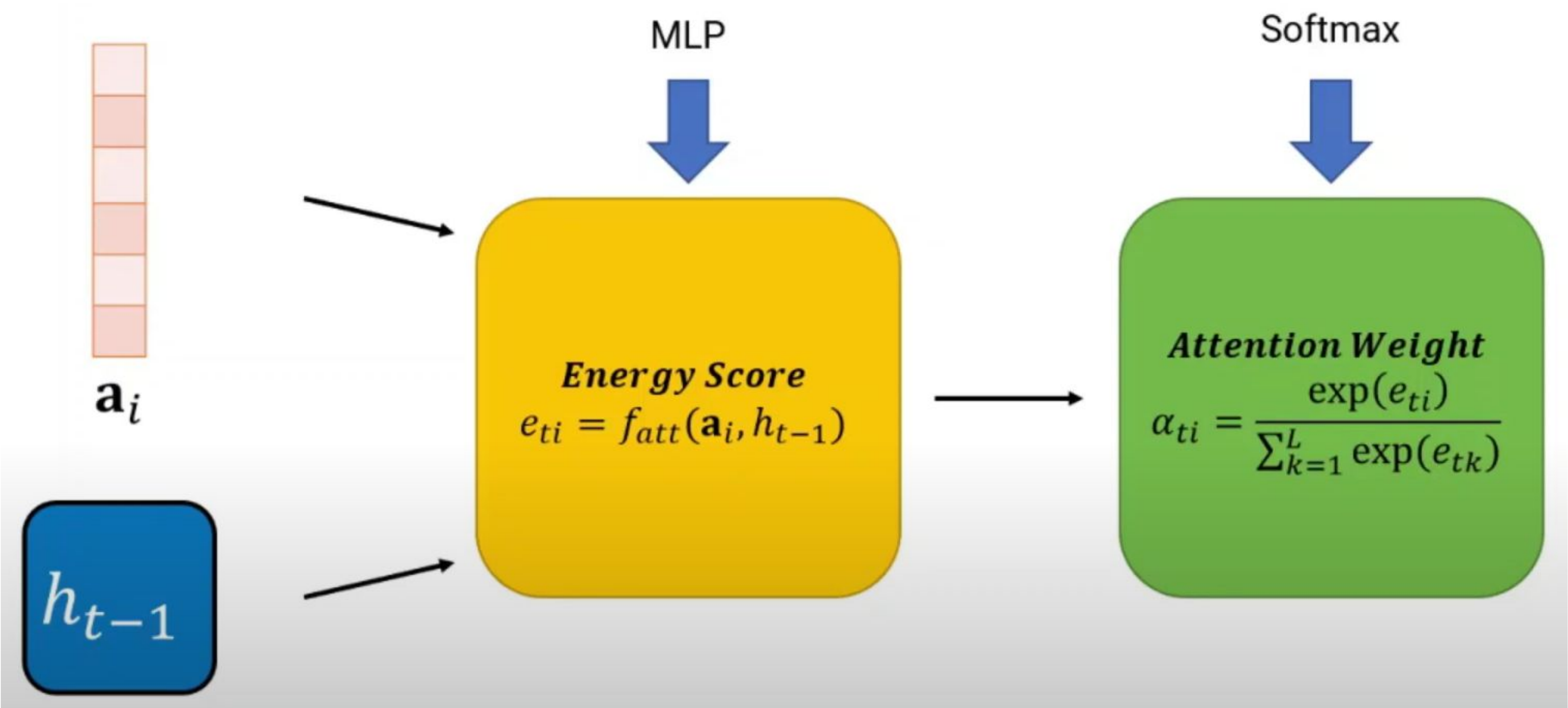
$$\hat{z}_t = \sum_{i=1}^{L} \alpha_{i,t} a_i$$



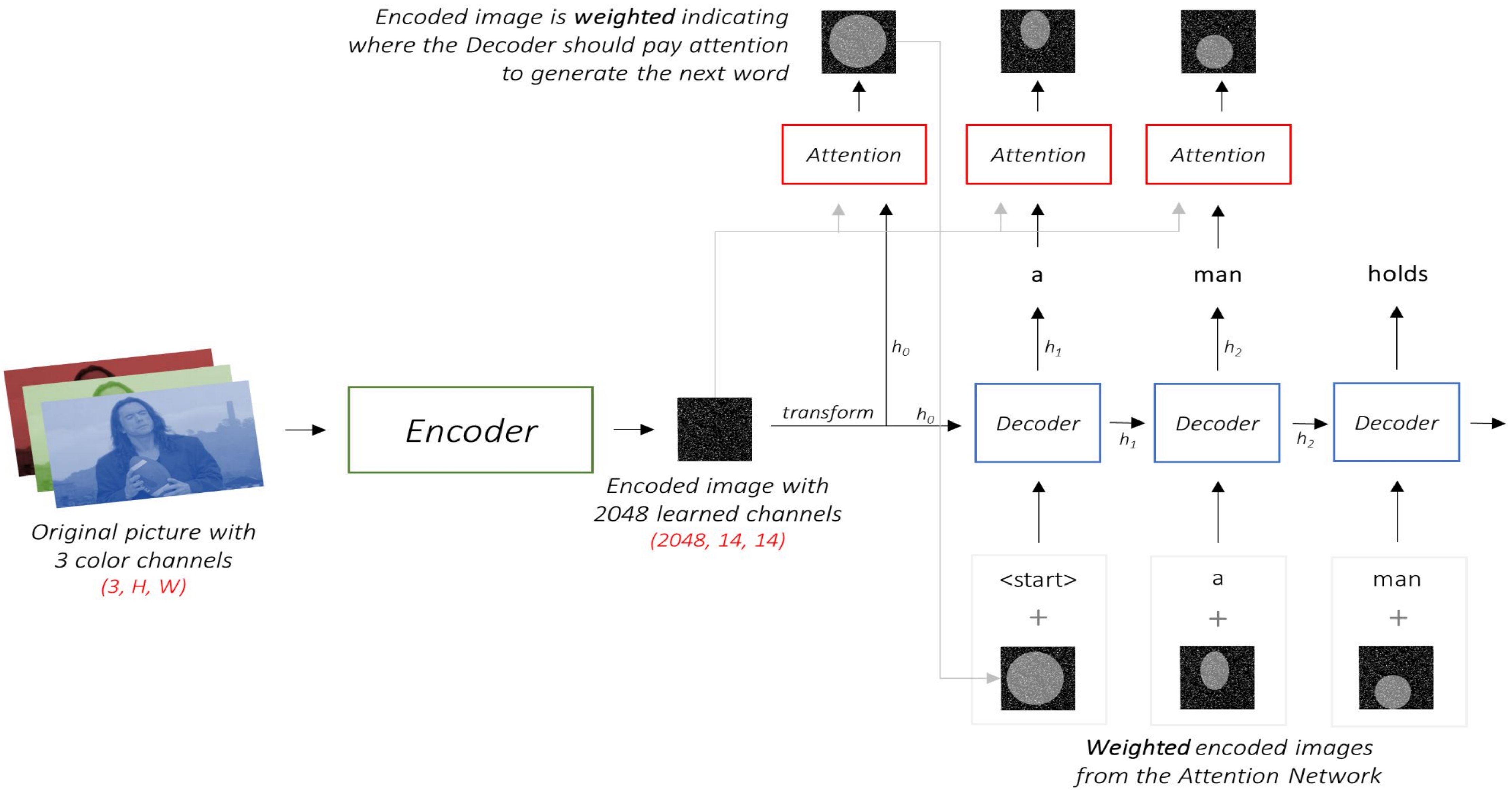


Context Vector

$$\hat{z}_t = \phi(\{a_i\}, \{\alpha_i\})$$



“Decoder with attention”



Doubly Stochastic ‘soft’ Attention

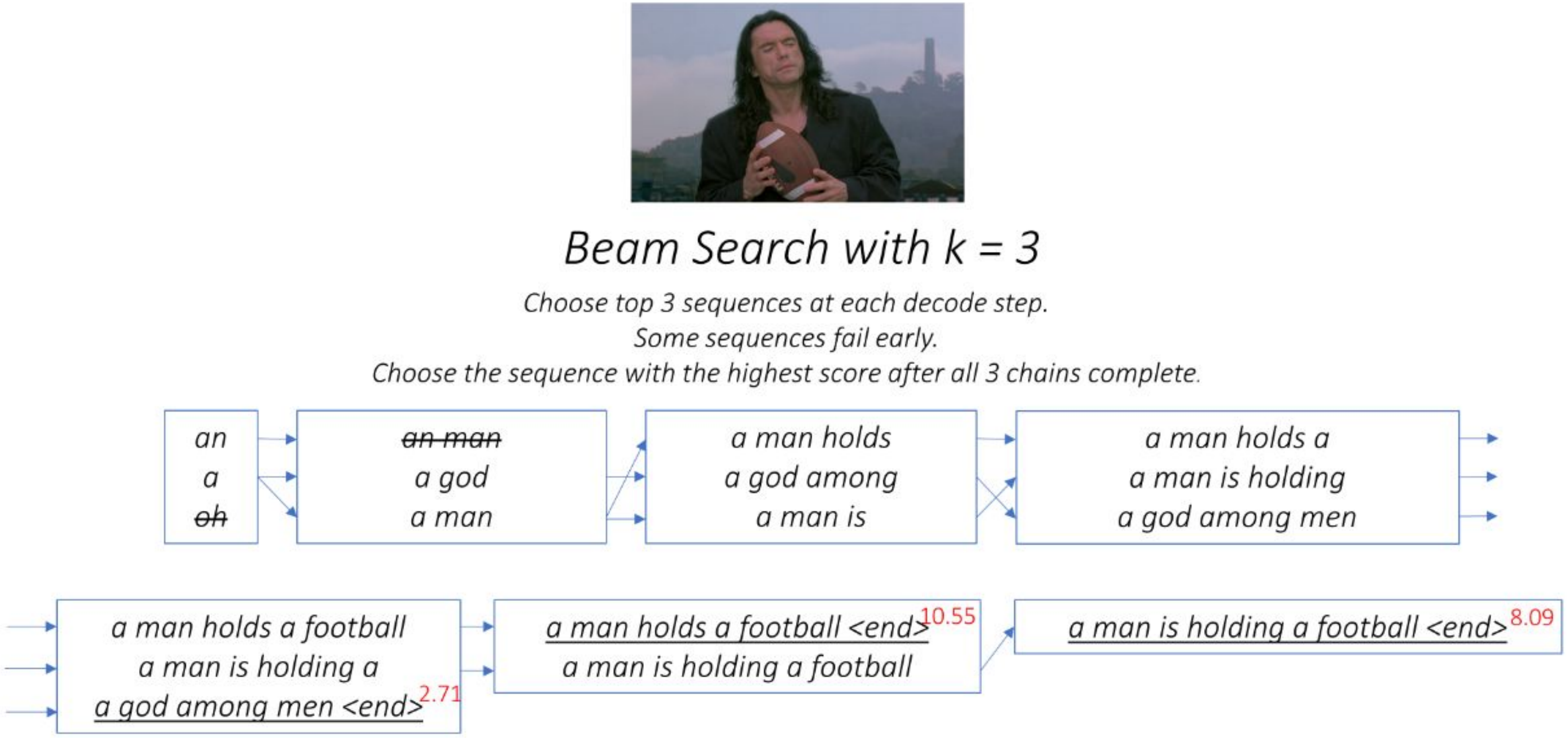
- Objective: Introduce doubly stochastic regularization in the deterministic model
- Allow sum of output from softmax to be approximately equal to 1
- Interpretation: Model pays equal attention to every part of the image
- Quantitatively, this should increase BLEU score
- Qualitatively, We expect more descriptive captions
- Minimize this penalized negative log-likelihood for end-to-end training

Loss Function:

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_i^L (1 - \sum_t^C \alpha_{ti})^2$$

Beam Search

- Beam Search selects the best possible sequence of words in language generation.
- It considers a basket of candidate sequences instead of greedily selecting the highest-scoring word at each step.
- At each decoding step, it generates k possible sequences and chooses the top k based on their scores.
- It continues this process until k sequences terminate and selects the one with the best overall score.
- Beam Search ensures a more comprehensive exploration of possible sequences and avoids sub-optimal outputs due to early incorrect choices.



Data preprocessing

- We normalized the images by the mean and standard deviation of the ImageNet image's RGB channels
  - mean = [0.485, 0.456, 0.406]
  - std = [0.229, 0.224, 0.225]

- We resized all MSCOCO images to 256x256 for uniformity.

- We created a *word_map* for the corpus, including the <start>,<end>, <pad> and <unk> tokens.
  - Words appearing <5 times are grouped under <unk> token
  - Word_map : 9849 words

Experiment setup

- Used pytorch to perform training on GPU
- Used pre-trained ResNet101 as encoder to generate feature representation of images
- Later on we also trained convolution block 3 and 4.
- The Attention network is simple – it's composed of only linear layers and a couple of activations.
- Used LSTMcell of pytorch as decoder model

Experiment setup

- Epochs = 30
- Embedding_dim = 512
- Attention_dim = 512
- Decoder_dim = 512
- encoder_lr = 1e-4
- decoder_lr = 4e-4

- Initially trained for 20 epochs without fine tuning the encoder

- Used code from COCO.api to compute BLEU, CIDEr and ROUGE_L scores
- Used nltk to compute METEOR score

Results

Training time per epoch: 1 hour (without training encoder)
2 hour (with training encoder)

Scores ->	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**	**CIDEr**	**ROUGE_L**
Paper's	70.7	49.2	34.4	24.3	23.9		
Our's (beam size 1)	71.29	53.93	39.65	29.05	23.77	95.01	52.51
Our's (beam size 3)	72.97	56.16	42.48	32.18	24.47	100.10	53.93
Our's (beam size 5)	72.81	55.99	42.43	32.27	24.51	100.16	54.02

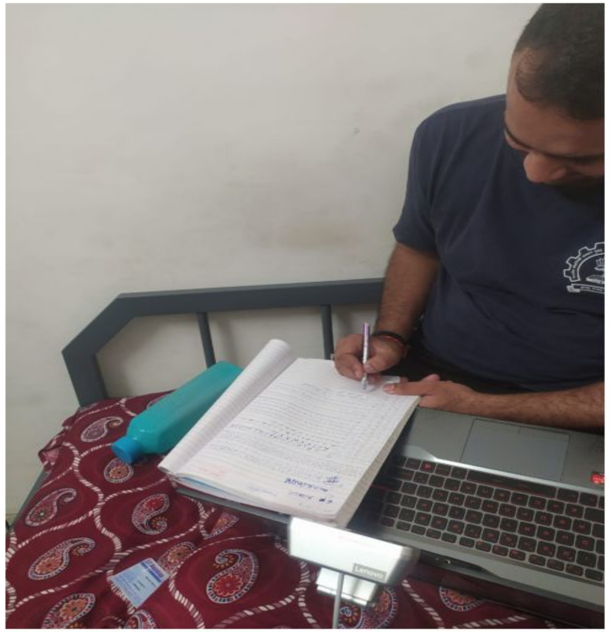
Modification

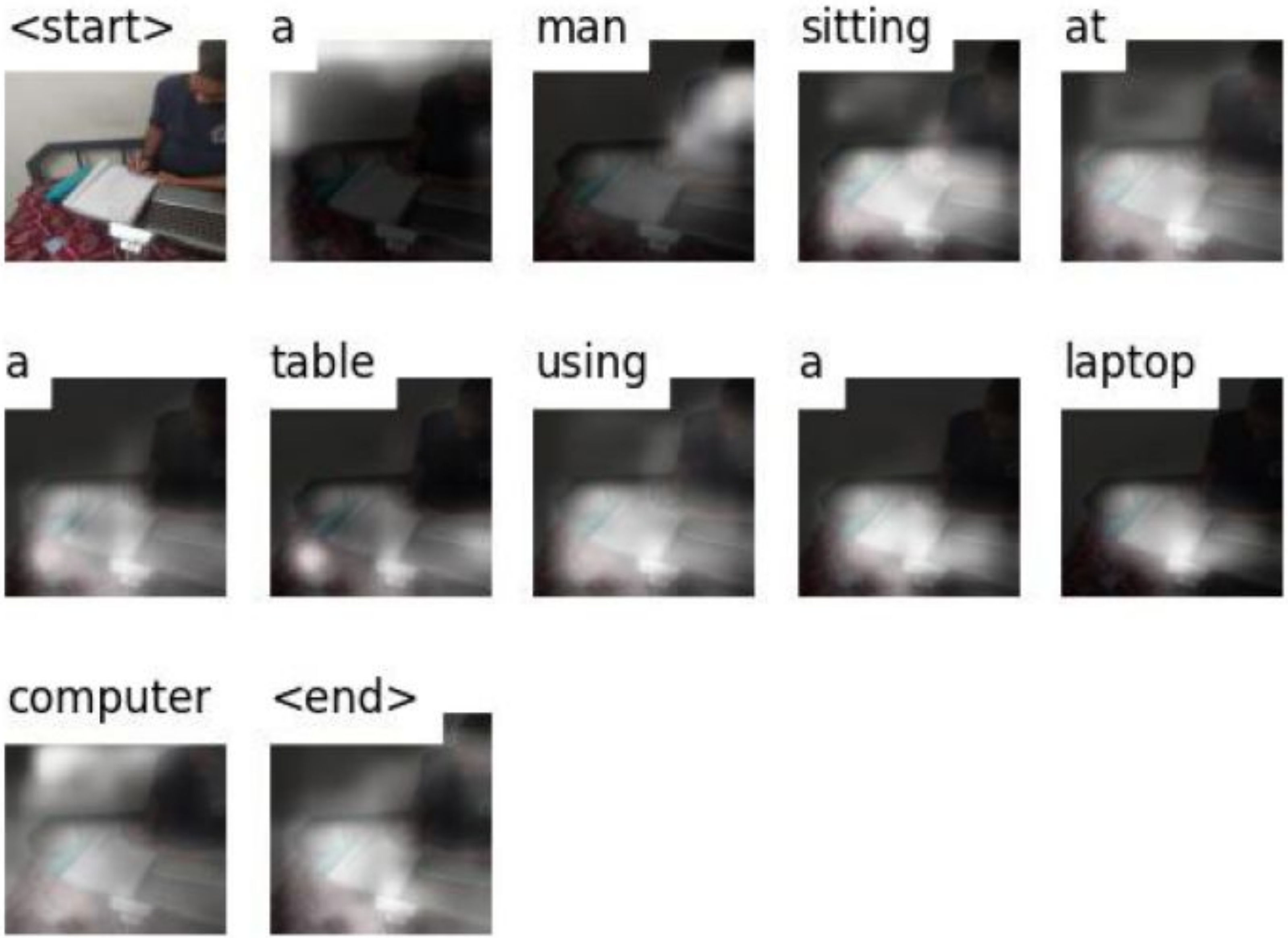
Instead of using just a single RNN for decoding purpose, we used multi layered RNN for the decoding purpose

Specifically, we used 2 layered RNN.

Scores ->	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**	**CIDEr**	**ROUGE_L**
Paper's	70.7	49.2	34.4	24.3	23.9		
Our's (beam size 1)	71.49	54.35	40.11	29.47	24.24	96.76	53.51
Our's (beam size 3)	72.97	56.16	42.48	32.18	24.47	100.10	53.93
Our's (beam size 5)	72.81	55.99	42.43	32.27	24.51	100.16	54.02

Visualizing attention





Analysis

- Attention mechanism selectively focus on different parts of the image when generating captions.
- The use of attention also allowed the model to be more robust to changes in image composition and viewpoint, since it could adaptively attend to different regions of the image.
- Using beam search algorithm during caption generation, it is able to generate diverse captions for the same image.

Limitation: Although the model attends to different part of the image, still most of the time it captures only single activity in the image.

References

- Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." *International conference on machine learning*. PMLR, 2015.
- https://github.com/kelvinxu/arctic-captions
- https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning
- https://github.com/yunjey/show-attend-and-tell
- https://cs.stanford.edu/people/karpathy/deepimagesent/
- https://cocodataset.org/#home

Acknowledgement

★ Kaggle
★ Google Colab

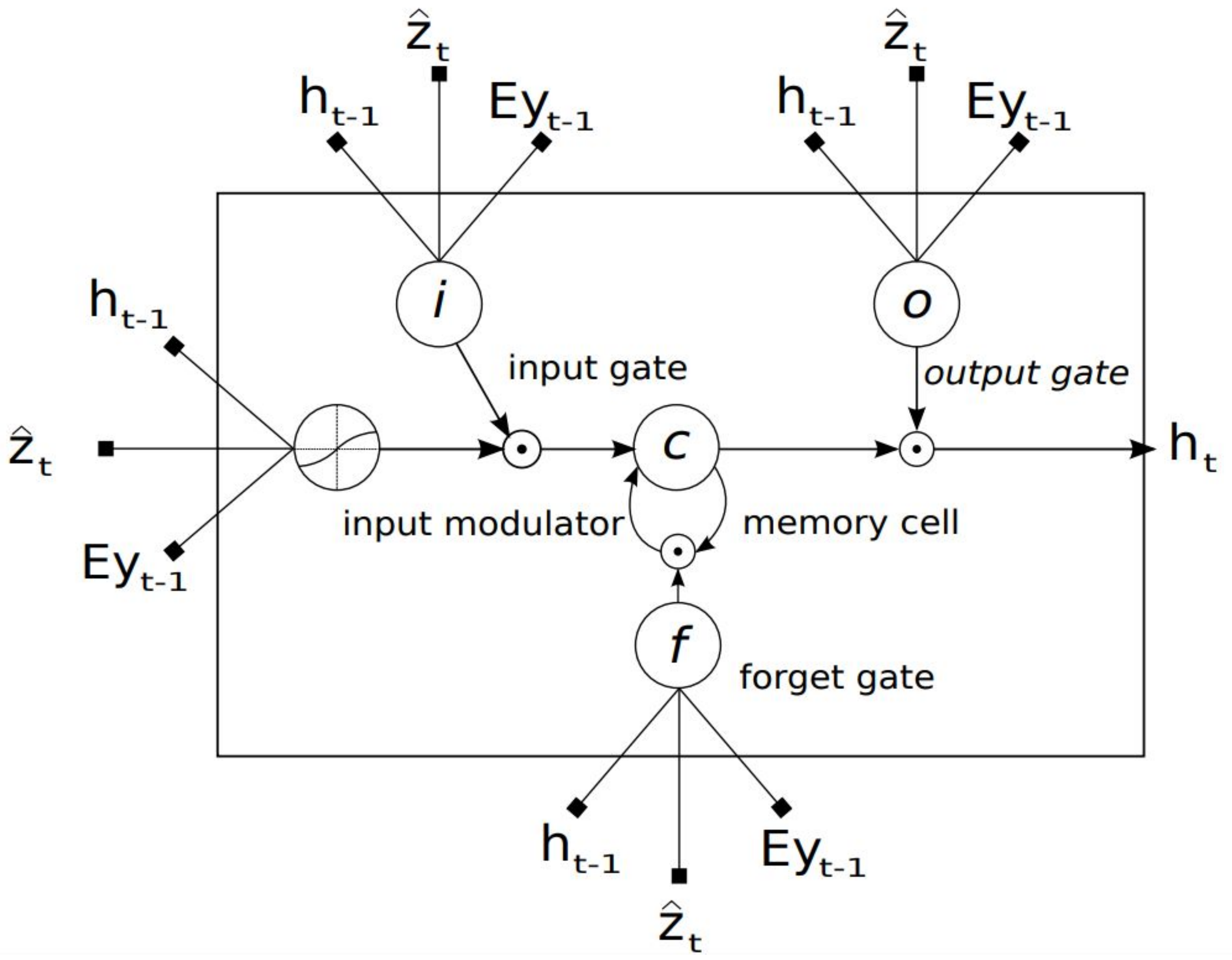
Thank You

LSTM architecture

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix}$$

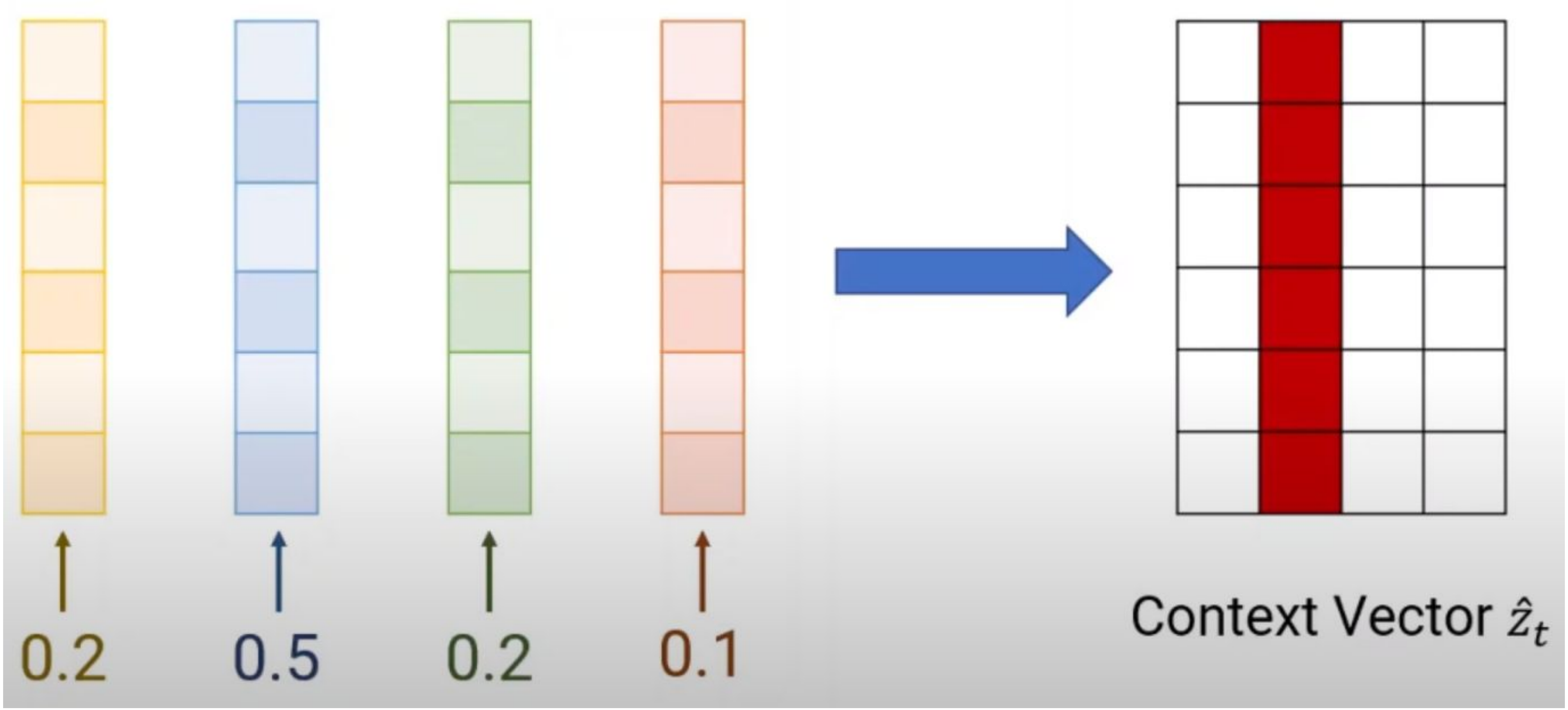
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).$$

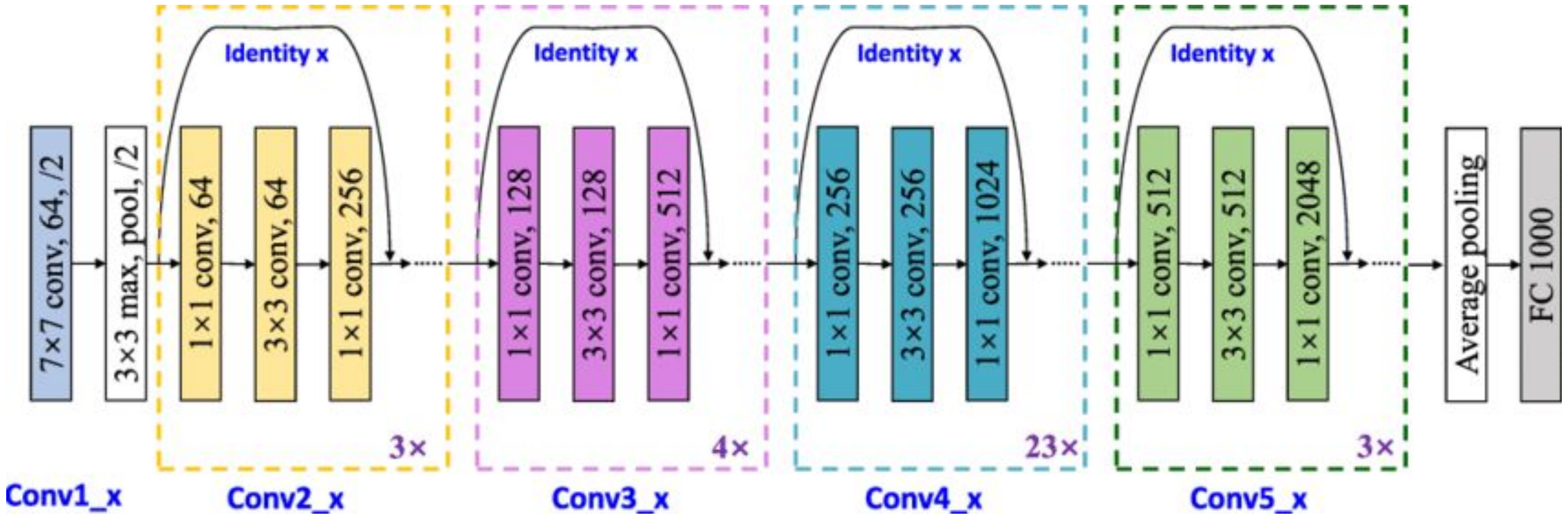


i_t = Input
f_t = Forget
C_t = memory
O_t = output
H_t = Hiddenstate

Hard Attention

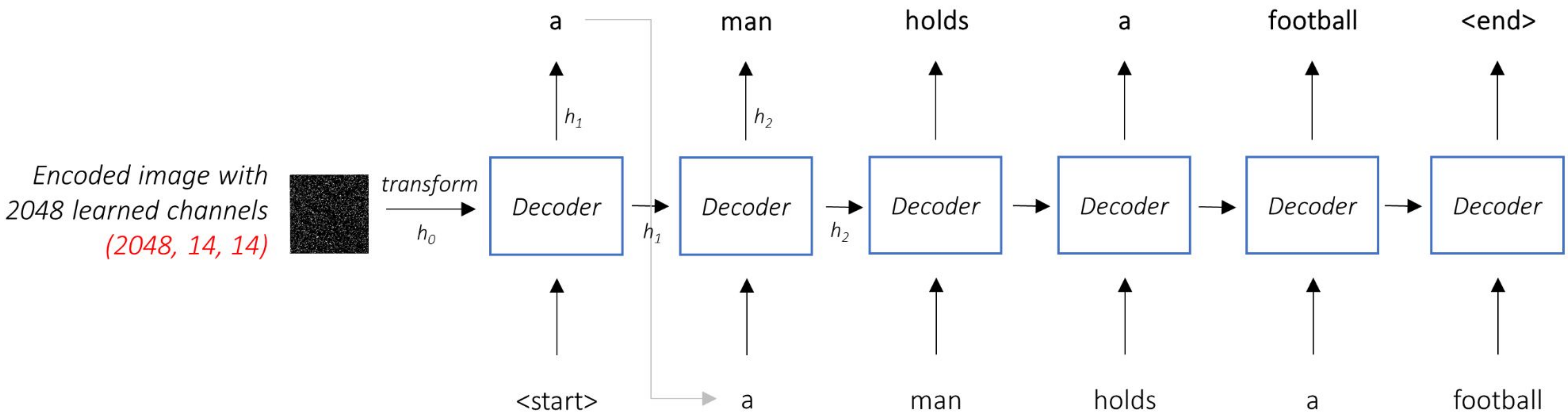


Convolutional Feature Extraction

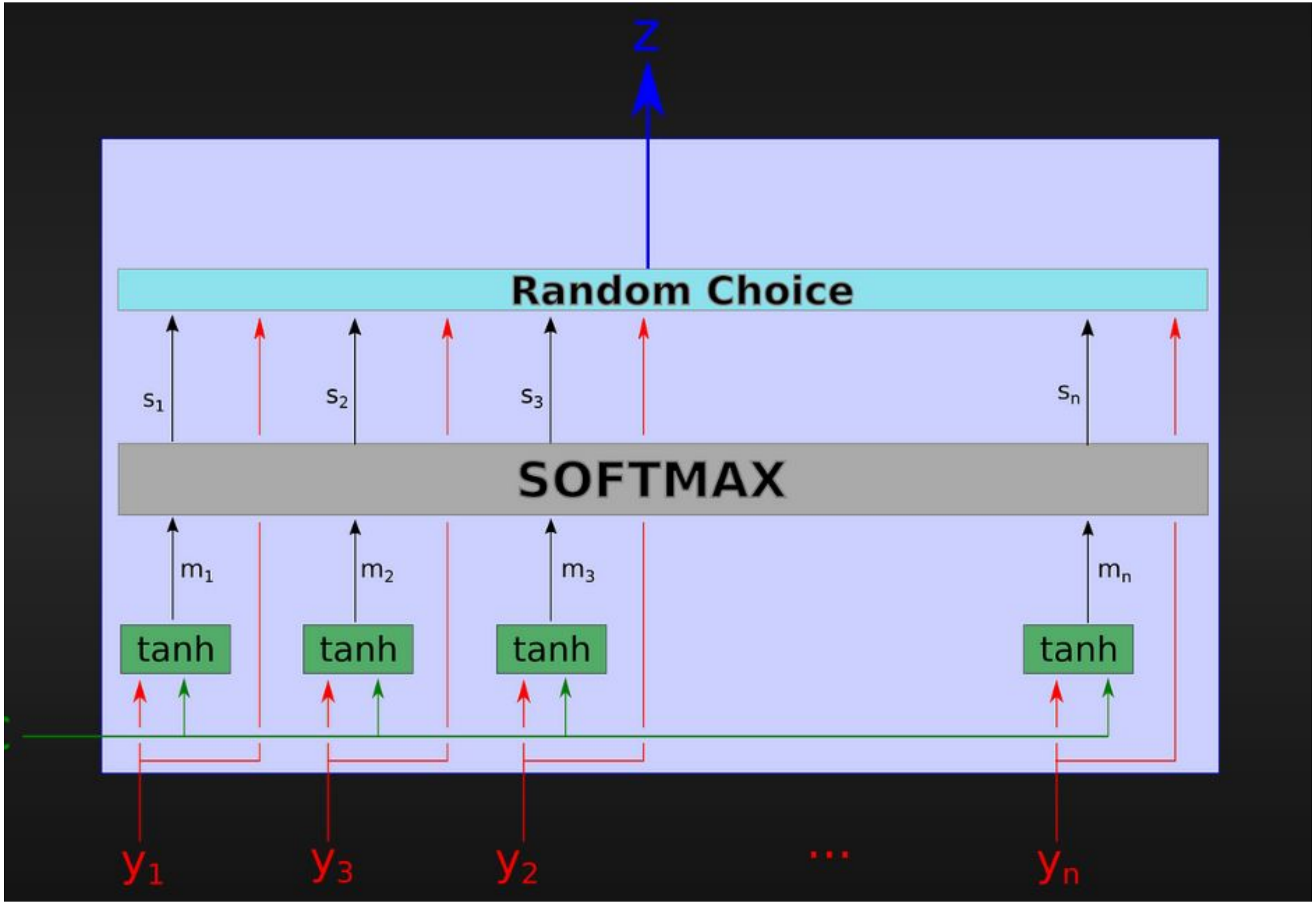


- is a deep convolutional neural network that uses residual connections.
- It consists of 101 layers, divided into four stages of residual blocks.
- Each residual block contains two or three convolutional layers and a shortcut connection.
- The shortcut connection allows the network to skip over certain layers, enabling it to learn higher-level features while avoiding the vanishing gradient problem.
- The architecture includes average pooling and fully connected layers at the end for classification.

Decoder without attention



Stochastic “Hard” Attention



Stochastic "Hard" Attention

- S_t: Location which model focuses to predict t-th word
- S_t,i = 1 in the one-hot vector if i-th location is used to predict t-th word. (i = 1,...L),

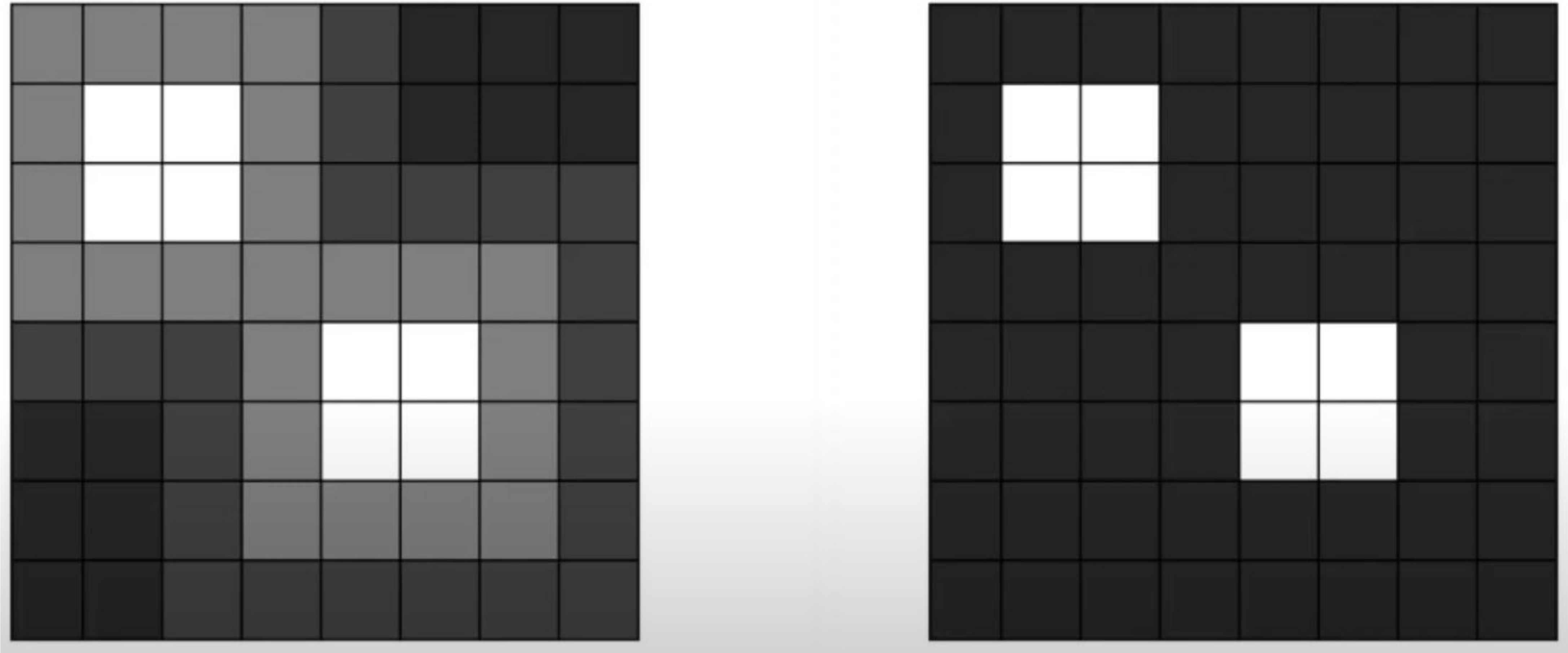
$$p(s_{t,i} = 1 \mid s_{j<t}, \mathbf{a}) = \alpha_{t,i}$$

$$\hat{\mathbf{z}}_t = \sum_i s_{t,i}\mathbf{a}_i.$$

Loss: $L_s = \sum_s p(s \mid \mathbf{a}) \log p(\mathbf{y} \mid s, \mathbf{a})$

- Hard attention focuses on any one of the L locations and is stochastic in nature
- Z_hat_t can be sampled from multinoulli(alpha_i)
- In making a hard choice, at every point, only one location is sampled

Attention : Soft and Hard



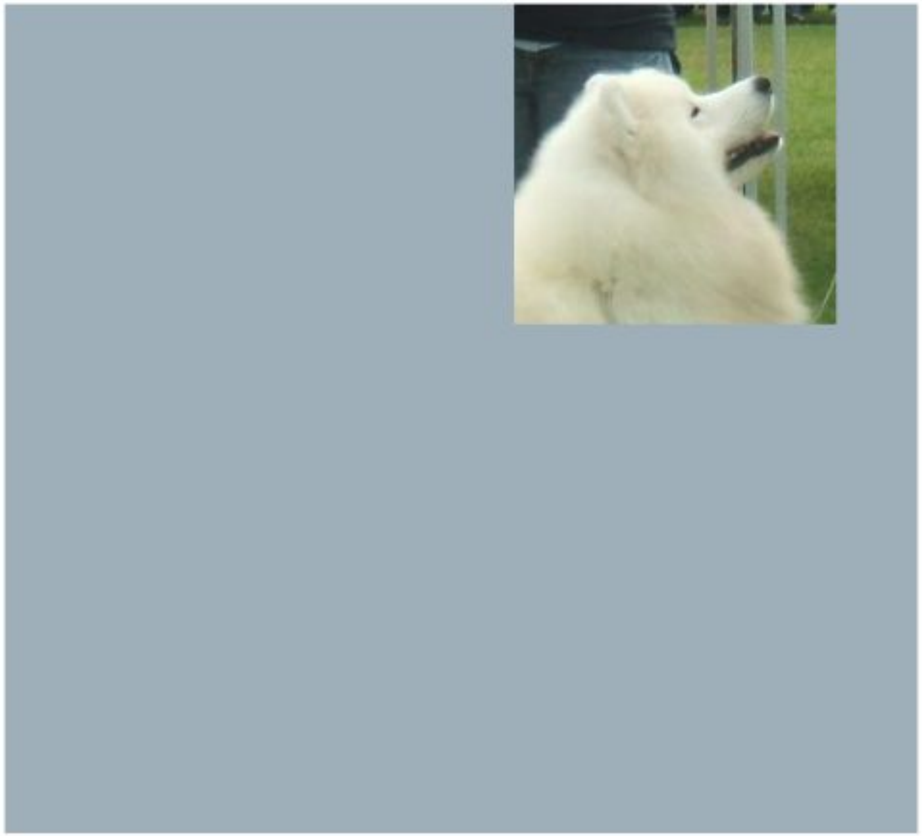
Soft

Hard

Attention : Soft and Hard

Soft Attention

Hard Attention



Deterministic "Soft" Attention

- Does not incorporate randomness as expectation of the context vector is taken
- Sampling the locations is not required, hence deterministic
- Standard Back propagation can be used for learning the weights as the model is differentiable under deterministic attention.

Context Vector:

$$\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^{L} \alpha_{t,i}\mathbf{a}_i$$

Deterministic Attention Model:

$$\phi(\{\mathbf{a}_i\}, \{\alpha_i\}) = \sum_i^L \alpha_i \mathbf{a}_i$$

Normalized Wtd Geometric Mean of softmax K-th word prediction

$$= \frac{\prod_i \exp(n_{t,k,i})^{p(s_{t,i}=1|a)}}{\sum_j \prod_i \exp(n_{t,j,i})^{p(s_{t,i}=1|a)}}$$

$$= \frac{\exp(\mathbb{E}_{p(s_t|a)}[n_{t,k}])}{\sum_j \exp(\mathbb{E}_{p(s_t|a)}[n_{t,j}])}$$

Beam Search

$$p(s_{t,i} = 1 \mid s_{j<t}, \mathbf{a}) = \alpha_{t,i}$$

$$\hat{\mathbf{z}}_t = \sum_i s_{t,i} \mathbf{a}_i.$$

Deterministic ‘Soft’ Attention

- Expected context vector z_t = $\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^{L} \alpha_{t,i}\mathbf{a}_i$
- The deterministic attention model $\phi\left(\{\mathbf{a}_i\},\{\alpha_i\}\right) = \sum_i^L \alpha_i \mathbf{a}_i$ *
- The whole model is smooth and differentiable, hence learning end-to-end is trivial by using backpropagation

* Introduced by Bahdanau et al. (2014)