IMAGE CAPTIONING

Neural Natural Language Processing

HAI LE, LINA BASHAEVA, PRATEEK RAJPUT, EVGENIY GARSIYA

Image Caption Overview

- Automatically generate captions from input images
- Commonly used **Encoder - Decoder** frameworks
- Encoders are typically CNN (vectorial representation of images)
- Decoders are typically RNN (decode those representations -> natural language)

Motivation

- Heart of **computer vision** and **NLP**
- Many medical applications such as medical diagnosis, virtual assistants for visually impaired individuals/disabled individuals, etc...
- Other applications inclues image indexing, improvement of search engines, recommendation system for editing software

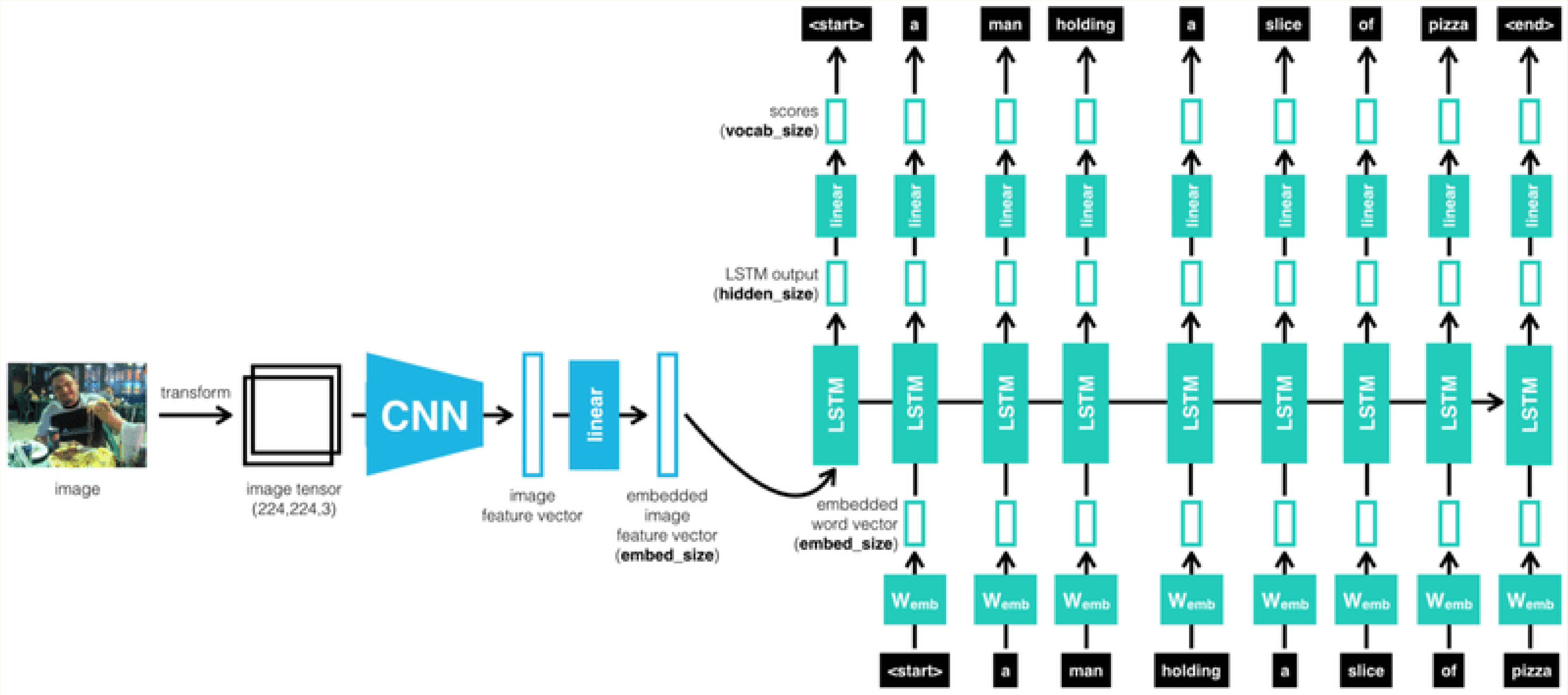
Goals/Objectives

- With the typical Encoder-Decoder framework, use different backbones and compare/contrast their performance.
- Which performs the best?
- Seq2seq vs. Transformer

ARE YOU READY?

Methodology description

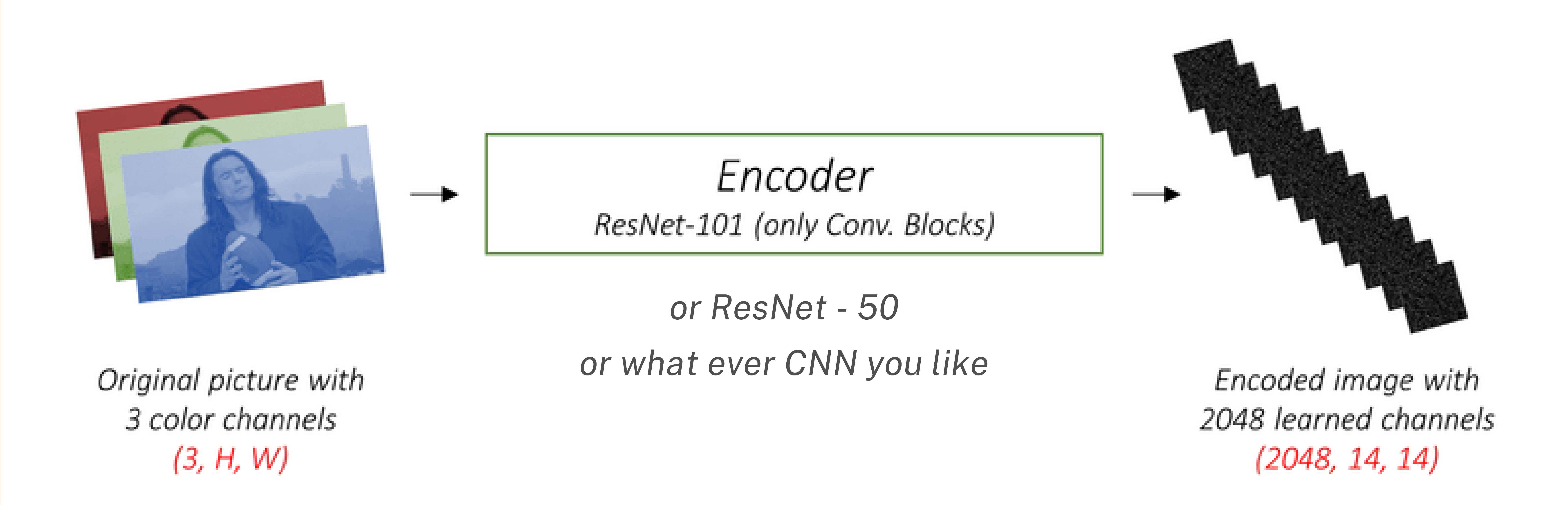
Image captioning pipeline



Blocks:

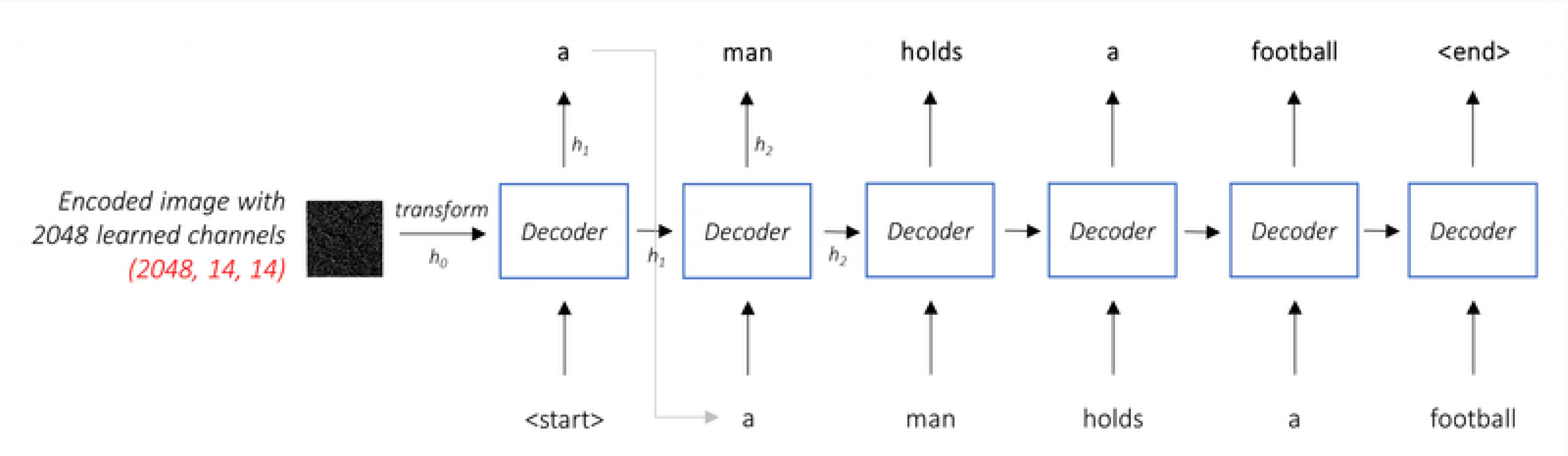
- *Transformations*
- *CNN*
- *Attention Net*
- *RNN*
- *Beam searching*

CNN



Encodes the input image with 3 color channels into a smaller image with "learned" channels.

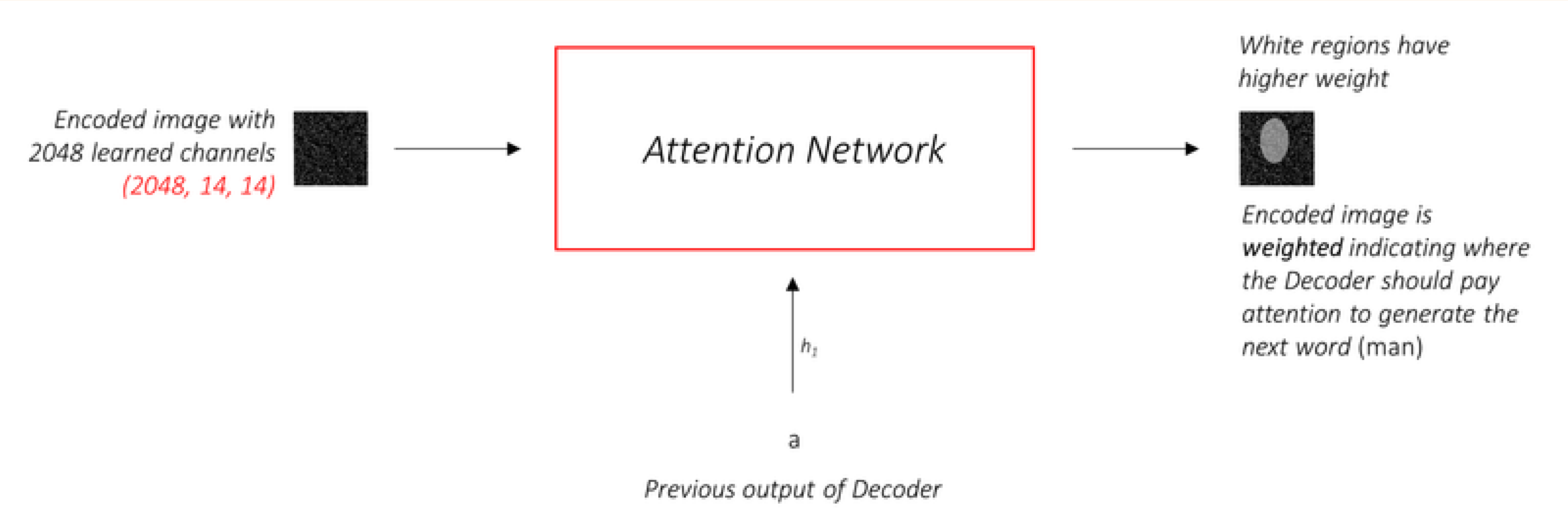
RNN
(without attention)



(LSTM as an example)

Decoder looks at the encoded image and generate a caption word by word.

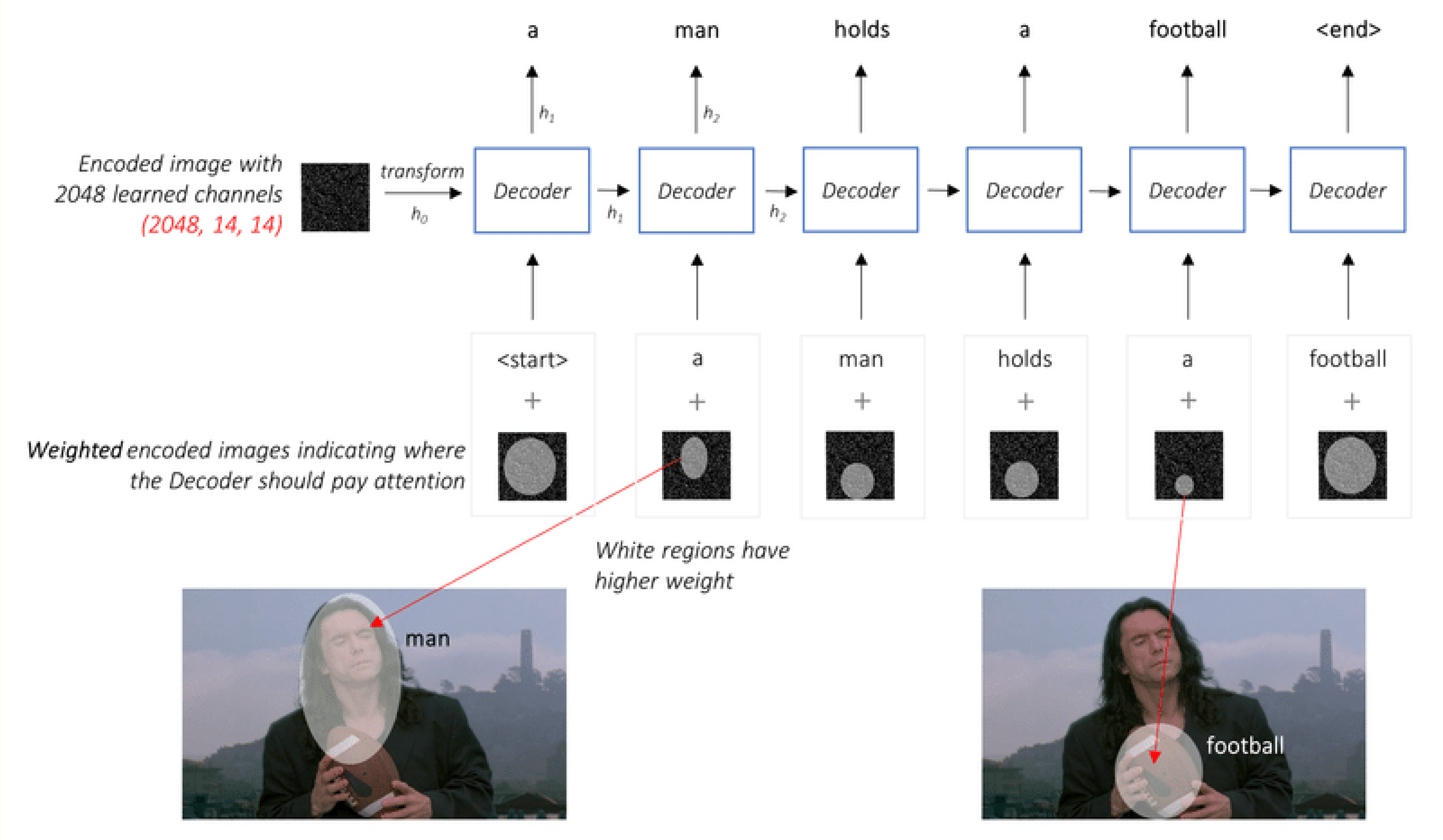
Attention net



Soft Attention: the weights of the pixels add up to 1. If there are P pixels in our encoded image, then at each timestep t:

$$\sum_{p}^{P} \alpha_{p,t} = 1$$

RNN
(with attention)

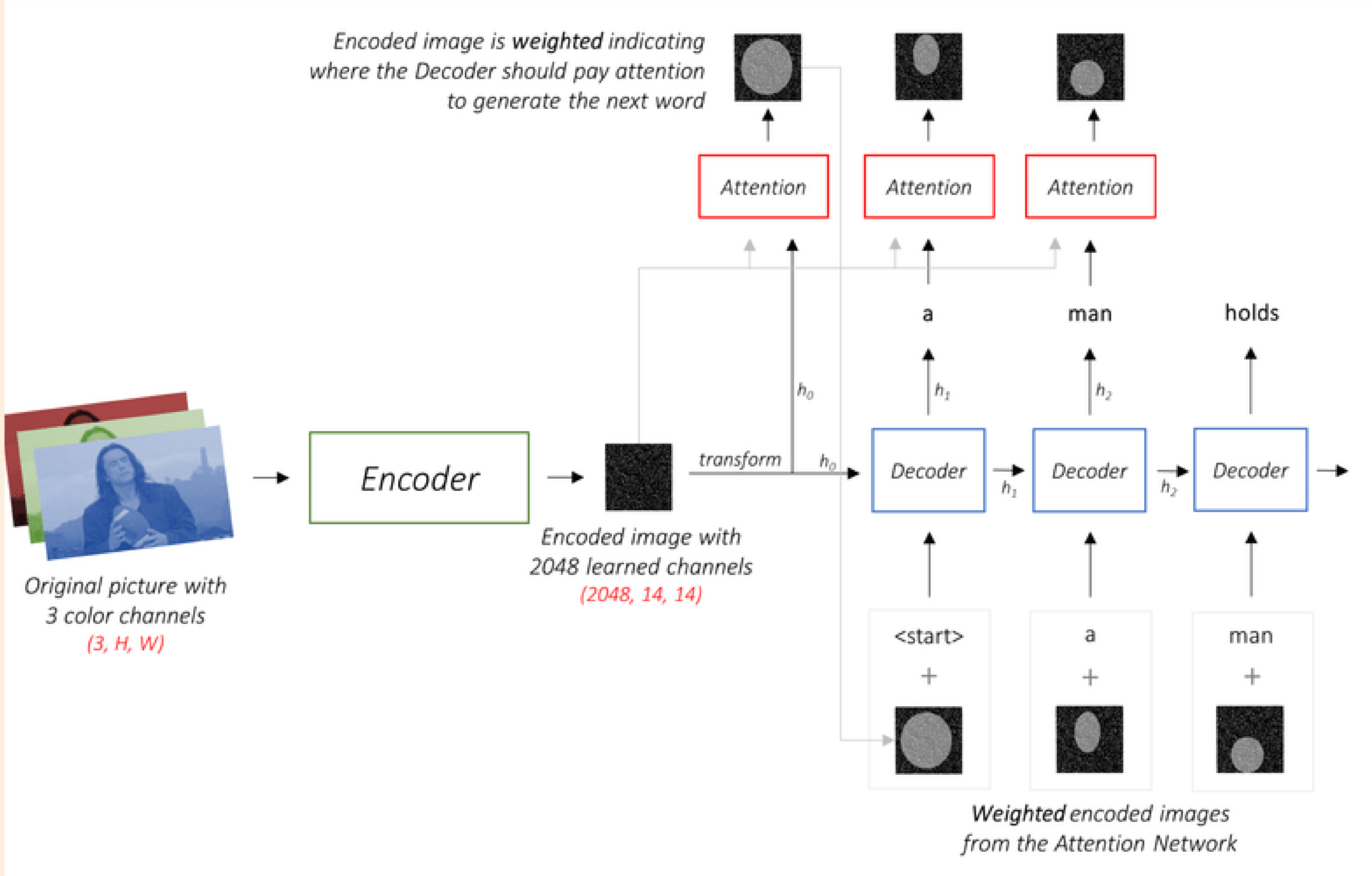


(LSTM as an example)

Decoder looks at different parts of the image at different points in the sequence

All together

1. Image transformations and normalisation to suit CNN requirements



2. Captions tokenisation and building vocabulary

3. Image encoding with CNN

4. Transform the encoding to create the initial hidden state for the RNN Decoder.

5. At each decode step:

- *the encoded image and the previous hidden state is used to generate weights for each pixel in the Attention network.*
- *the previously generated word and the weighted average of the encoding are fed to the LSTM Decoder to generate the next word*

Implemented pipeline

Frameworks/Backbones

01 INCEPTIONV3 + GRU

02 DENSENET121 + GRU

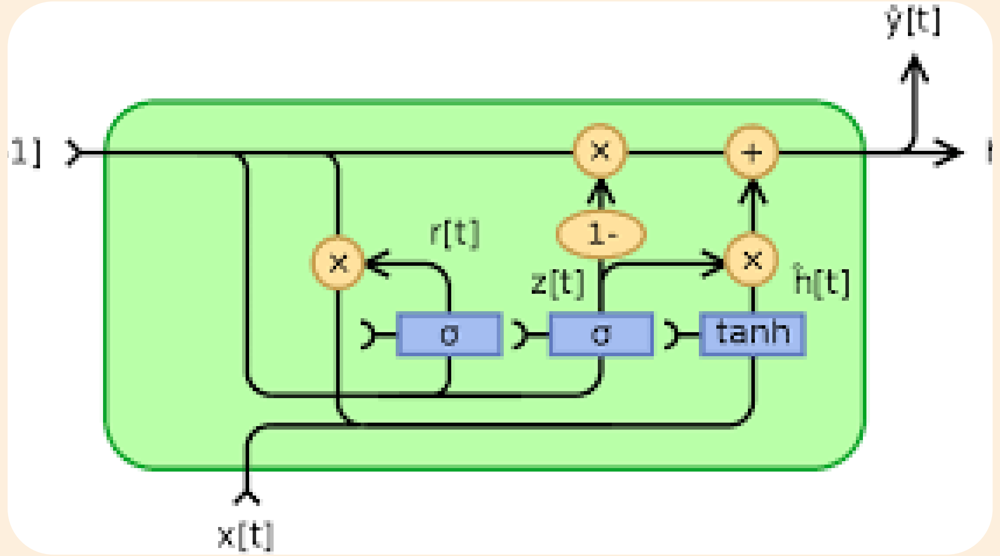
03 RESNET101 + GRU

01 INCEPTIONV3 + TRANSFORMER

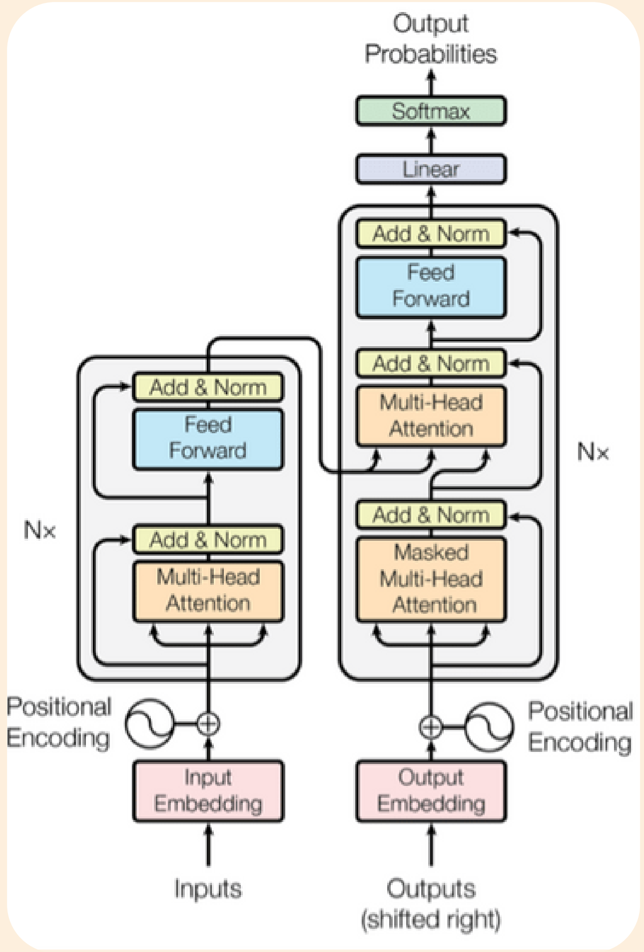
02 DENSENET121 + TRANSFORMER

03 RESNET101 + TRANSFORMER

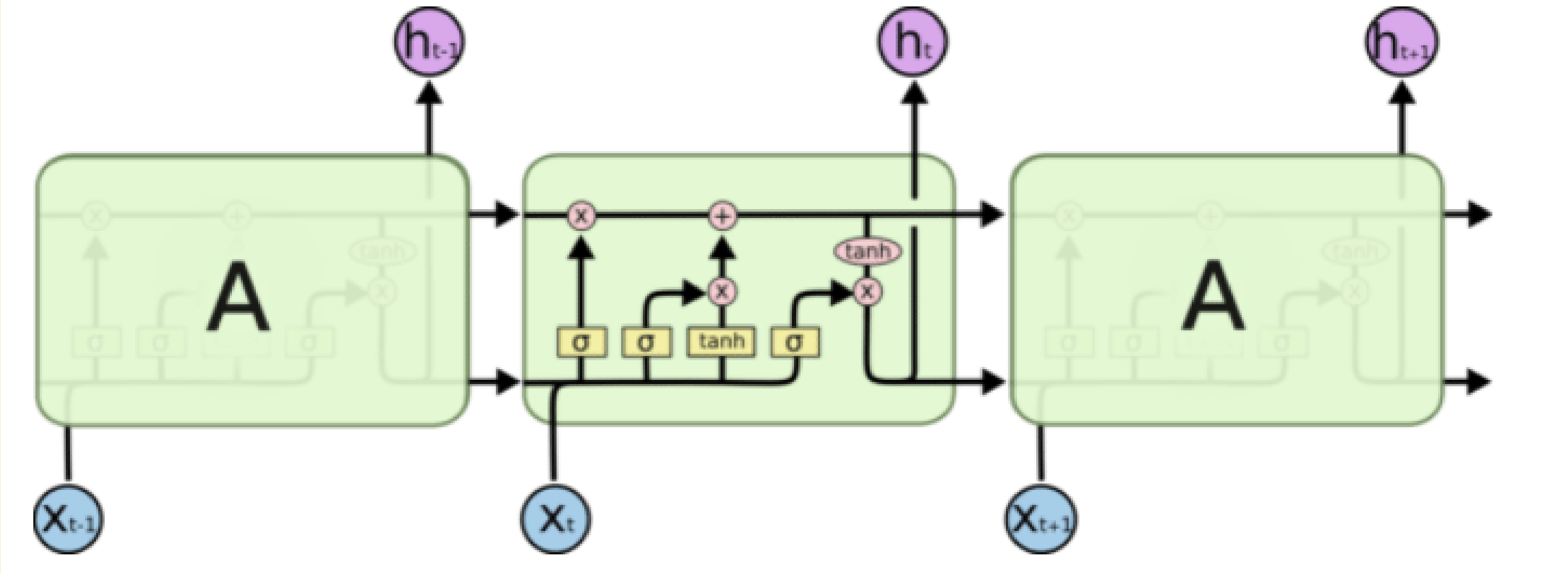
GRU FRAME WORK

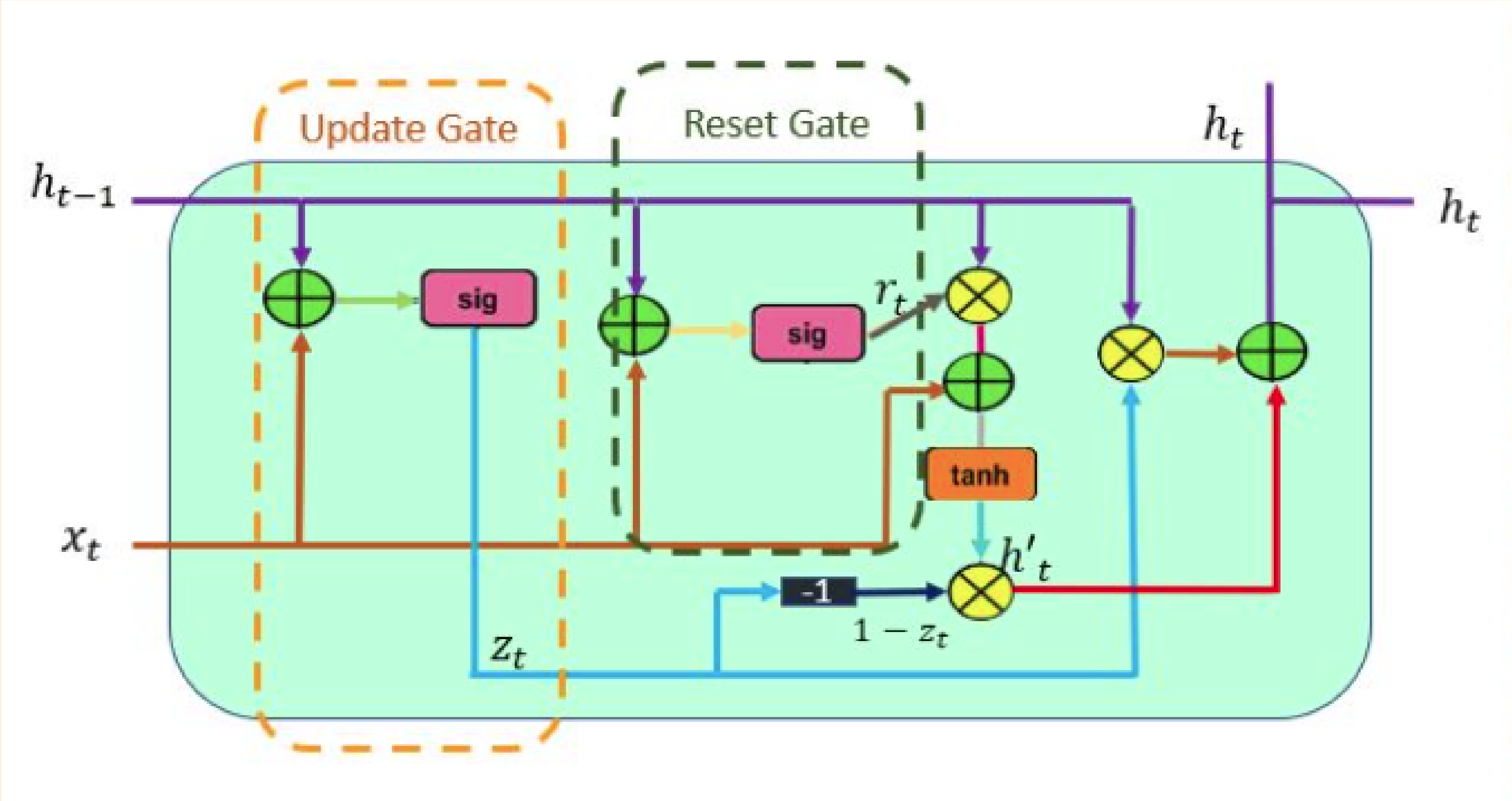


TRANSFORMER FRAME WORK



GRU network





LSTM

- *Input gate*
- *Update gate*
- *Output gate*

GRU

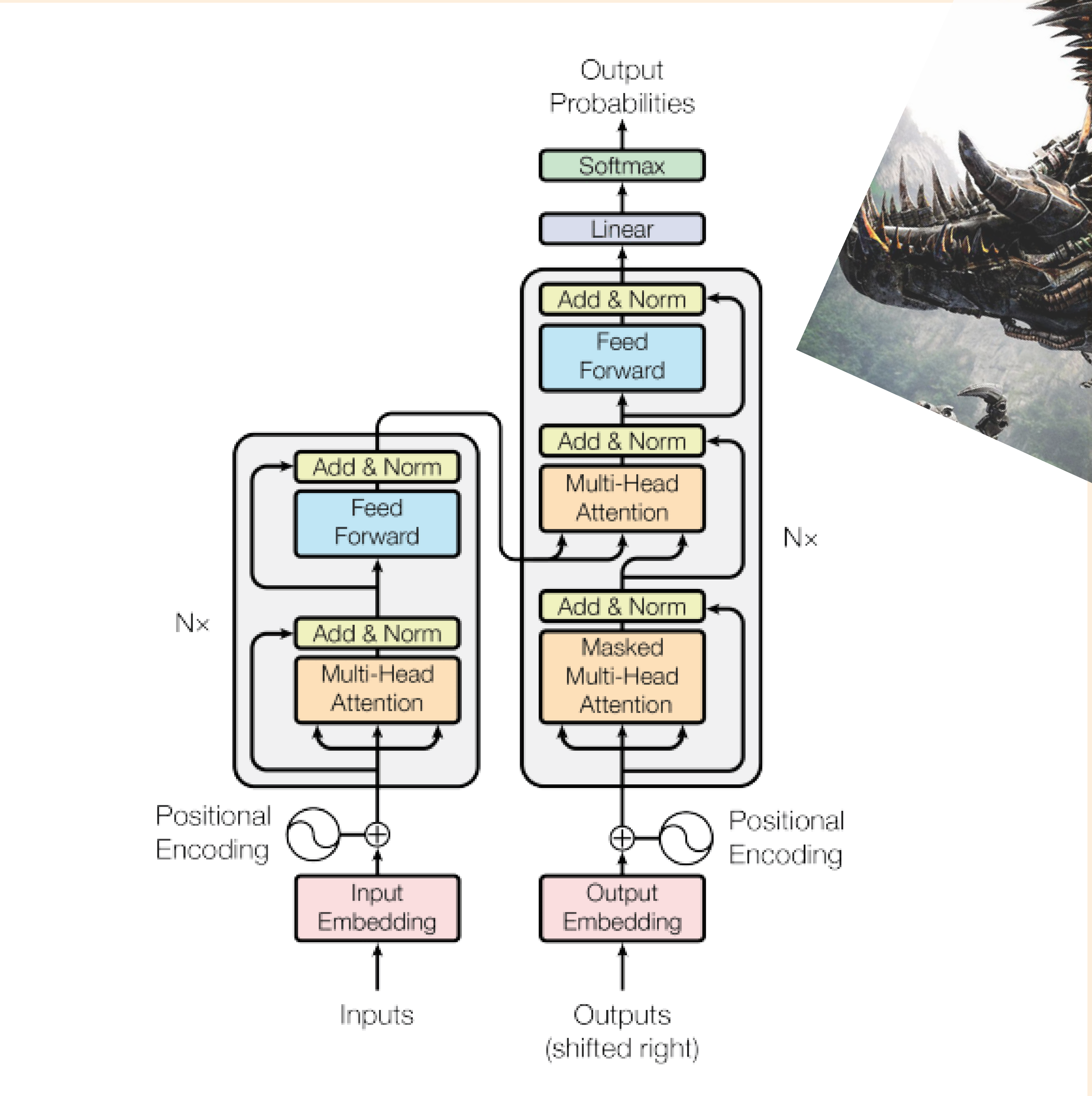
- *Update gate*
- *Reset gate*

GRU vs LSTM

- *The GRU has two gates, LSTM has three gates*
- *The structure of GRU is simpler. It has one gate less than LSTM, which reduces matrix multiplication, and GRU can save a lot of time without sacrificing performance.*
- *GRU does not possess any internal memory, they don't have an output gate that is present in LSTM*
- *In LSTM the input gate and target gate are coupled by an update gate and in GRU reset gate is applied directly to the previous hidden state. In LSTM the responsibility of reset gate is taken by the two gates i.e., input and target.*

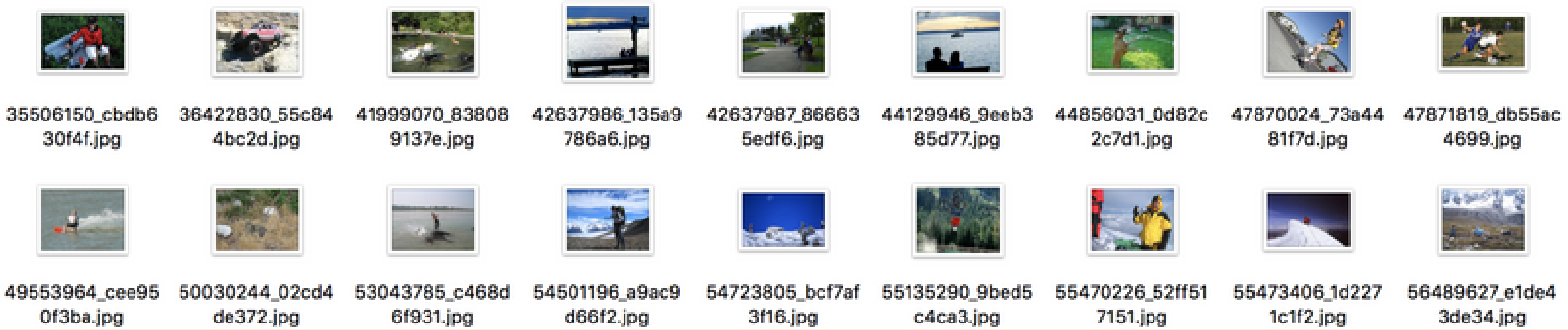
GRU uses less training parameter and therefore uses less memory and executes faster than LSTM whereas LSTM is more accurate on a larger dataset.

Transformers

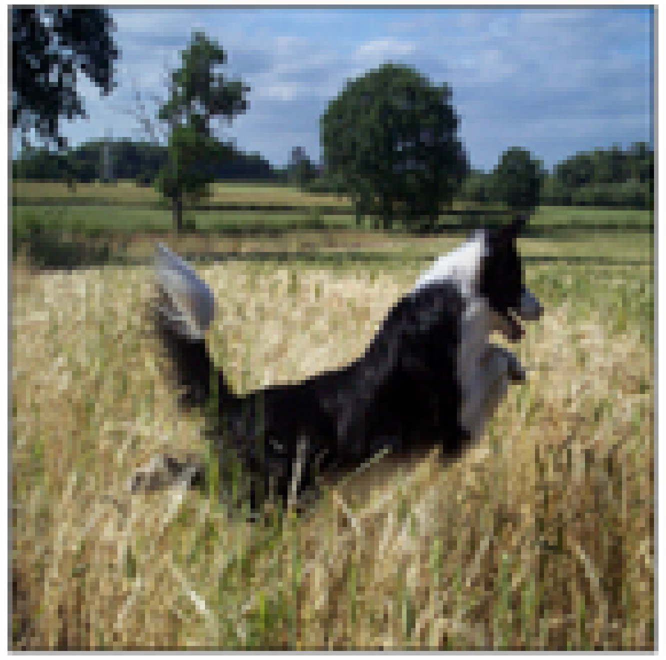


Flickr8k Data

- *A benchmark collection for sentence-based image description and search,*
- ***8,000*** *images*
- *each paired with five different captions*
- *clear descriptions of the salient entities and events.*
- *images were chosen from six different Flickr groups*
- *images tend don't contain any well-known people or locations, but were manually selected to depict a variety of scenes and situations*



Flickr8k Data examples



the dog is running through a field .

a white and black dog leaps through long grass in a field .

a black and white dog is running through the grass .

a black and white dog bounds through tall wheat grass .

a black and white dog bounds through a field .

the girl is holding a green ball .

a young girl wearing white looks at the camera as she plays .

a smiling young girl in braids is playing ball .

a little girl in white is looking back at the camera while carrying a water grenade .

a girl in a white dress .

Images preprocessing

1. Resize transform to suit CNN requirements:

- *InceptionV3 - Resize(299)*
- *DenseNet121 - Resize(224)*
- *ResNet101 - Resize(224)*

2. Normalisation:

- *InceptionV3 - scaled between -1 and 1*
- *DenseNet121 - scaled between 0 and 1 and each channel is normalized with respect to the ImageNet dataset*
- *ResNet101 - converted from RGB to BGR, then each color channel is zero-centered with respect to the ImageNet dataset, without scaling.*

Text preprocessing

- *Tokenization*
- *We tried different embedding sizes: 50, 100, 200, 300*
- *At first we set it equal to 50 due to our limited computational efficiency*
- *But the less is embedding size, the less information we can save for our text, that's why we tried to increase it*
- *The best performance was shown with it equal to 200 and then the performance stopped to increase with the growth of this parameter*

Captions preprocessing

a man holds a football

<start> a man holds a football <end>

<start> a man holds a football <end> <pad> <pad>

9876 1 5 120 1 5406 9877 9878 9878 9878

Results and discussions

Obtained results

	AVG BLEU
Inceptionv3 + GRU	0.24
DenseNet121 + GRU	0.15
ResNet101 + GRU	0.28
Inceptionv3 + Transformer	---
DenseNet16 + Transformer	0.38
ResNet101 + Transformer	---

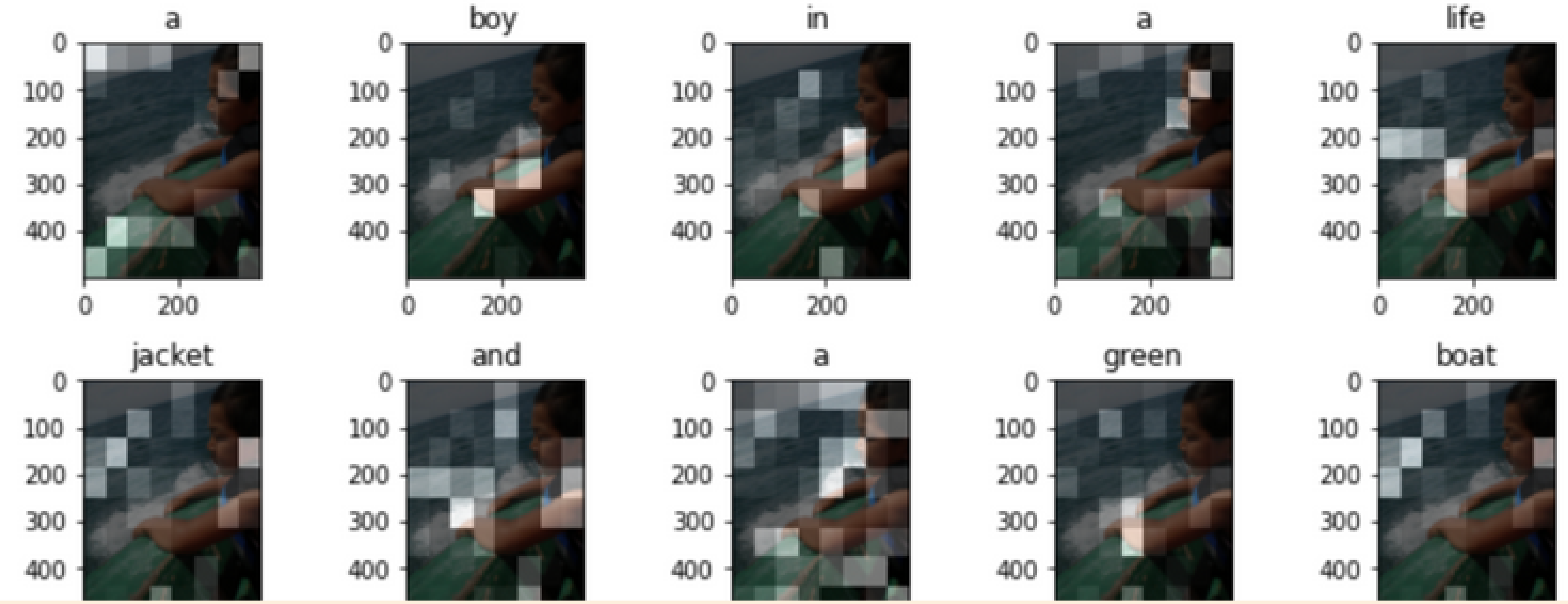
Examples

DenseNet121 + Transformer

generated caption: a man and woman are posing for a picture .
GT: ['A man and woman , on a park trail , pose in front of a lake and distant mountain .
bleu4: 33.12

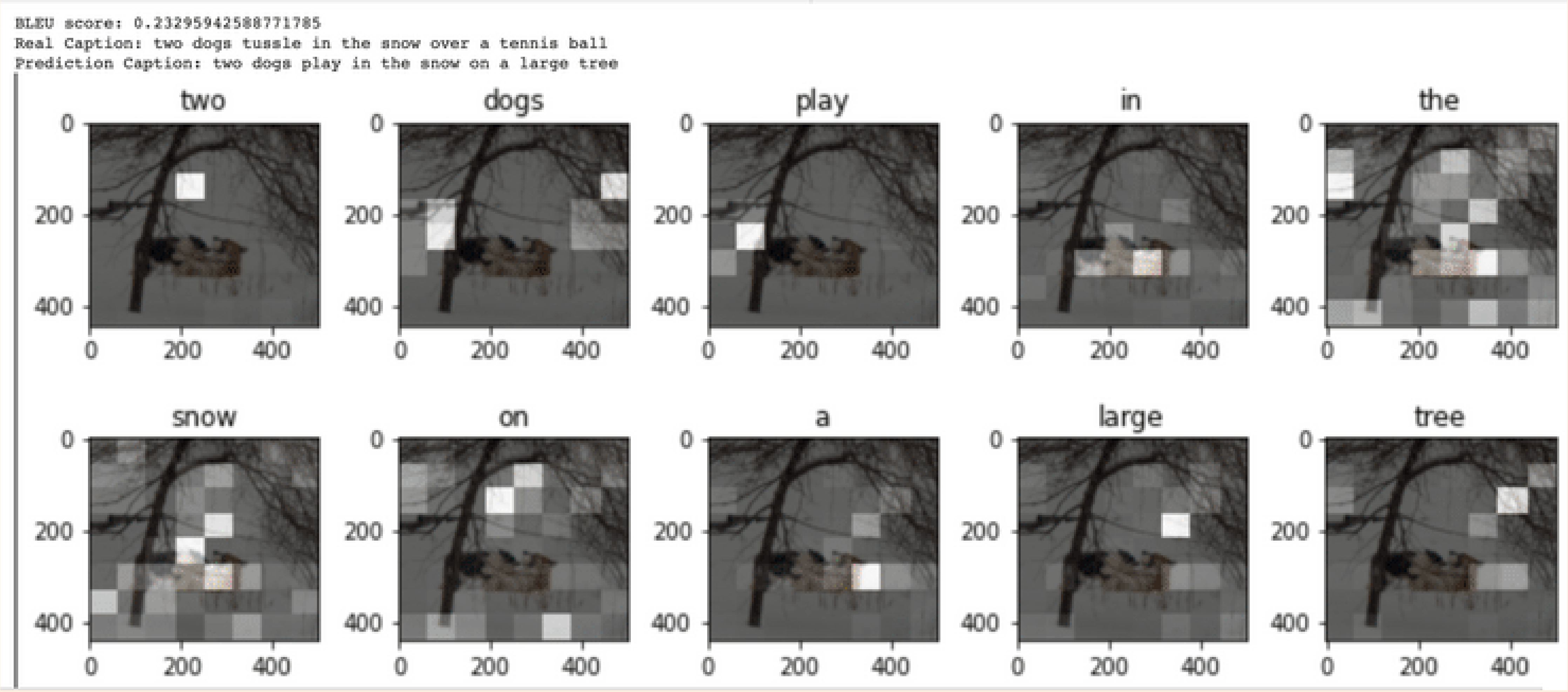
DenseNet121 + GRU

BLEU score: 0.337796316910539 64
Real Caption: a child holds on to the side of a small
Prediction Caption: a boy in a life jacket and a green boat

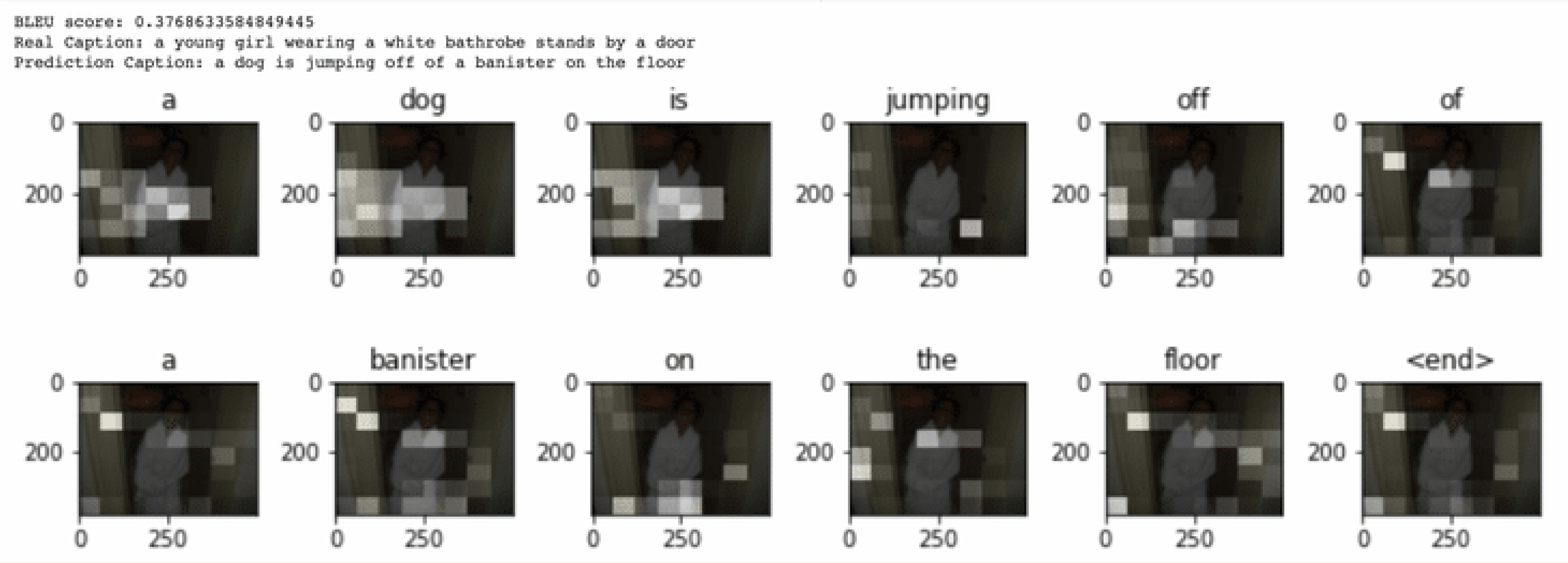


Examples

ResNet101 + GRU



Inceptionv3 + GRU



Conclusion

- ***The best perfomance were demonstrated on transformer based net: avg bleu = 0.38***

- ***The most qualified captions were predicted on transformer based net also***

- ***One of the possible variants to improve results - try to use LSTM net and much bigger dataset like COCO***

- ***Network Hyper parameters tuning (like embedding and hidden size) can also help with improving but it can increase calculation resources and training time***

Thank you for your attention!

HAVE A GREAT DAY AHEAD.