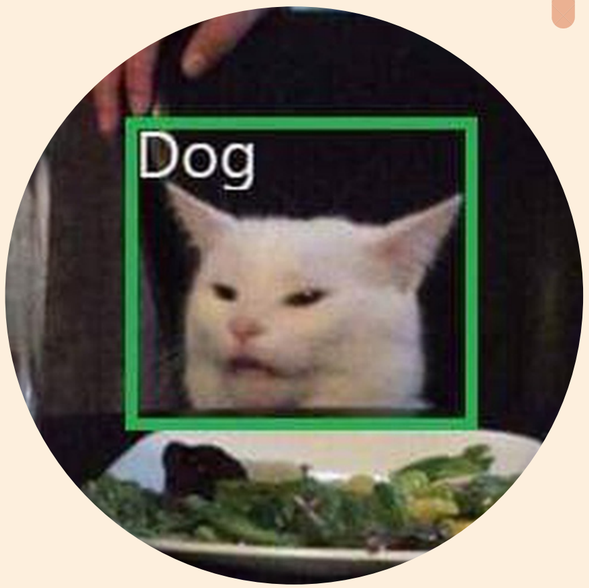
IMAGE CAPTIONING

Neural Natural Language Processing



FAKHRIDDIN TOJIBOEV, OLYA GORBUNOVA, FARID DAVLETSHIN, DMITRY GILYOV, HAI LE, LINA BASHAEVA, ALBERT SAYAPIN, 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 includes image indexing, improvement of search engines, recommendation system for editing software

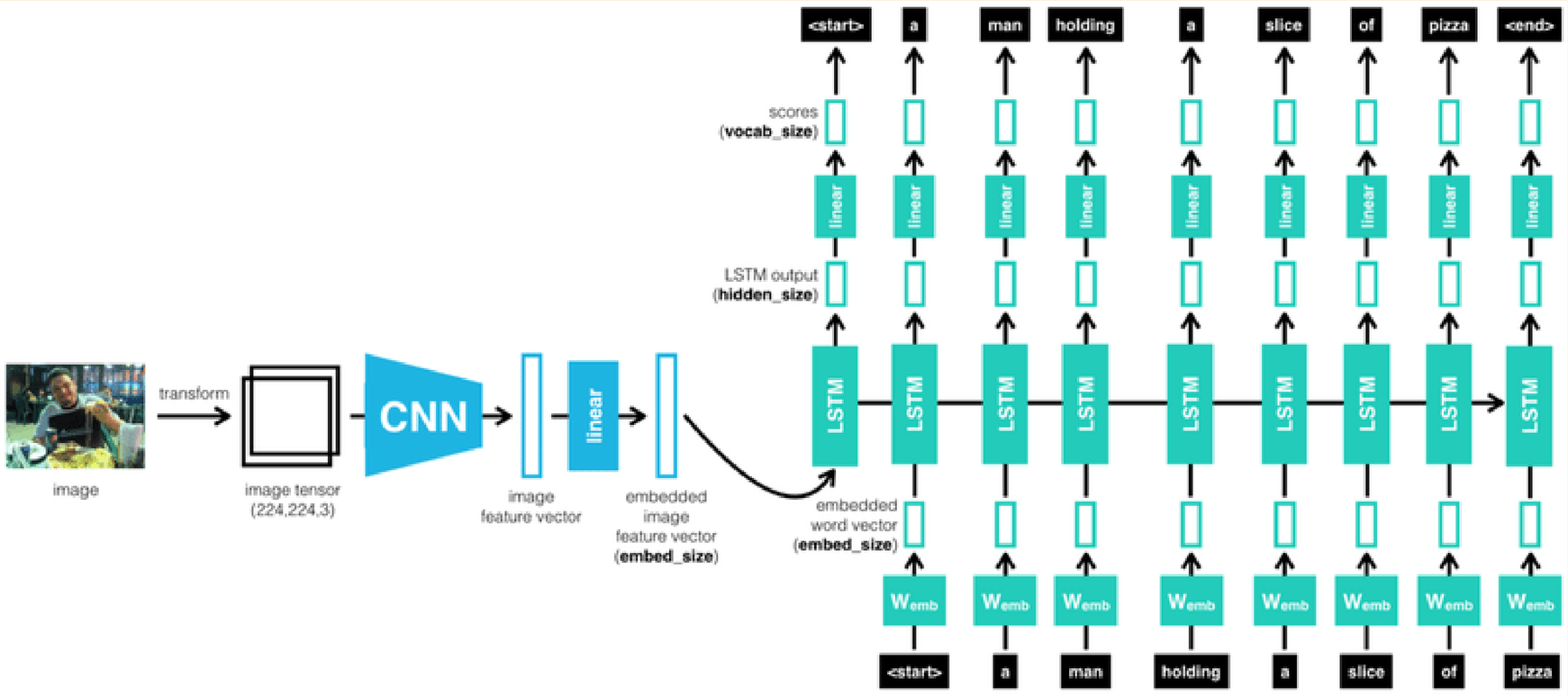
Goals/Objectives

- 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

Original picture with
3 color channels
(3, H, W)

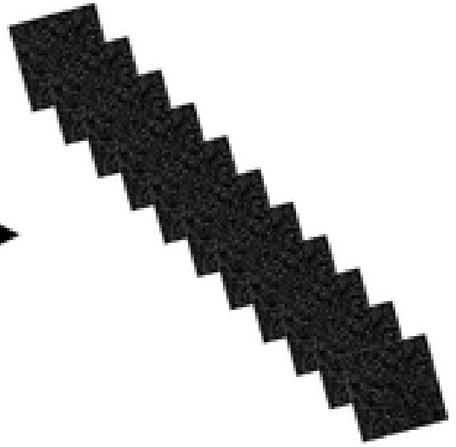
→

Encoder
ResNet-101 (only Conv. Blocks)

or ResNet - 50

or what ever CNN you like

→

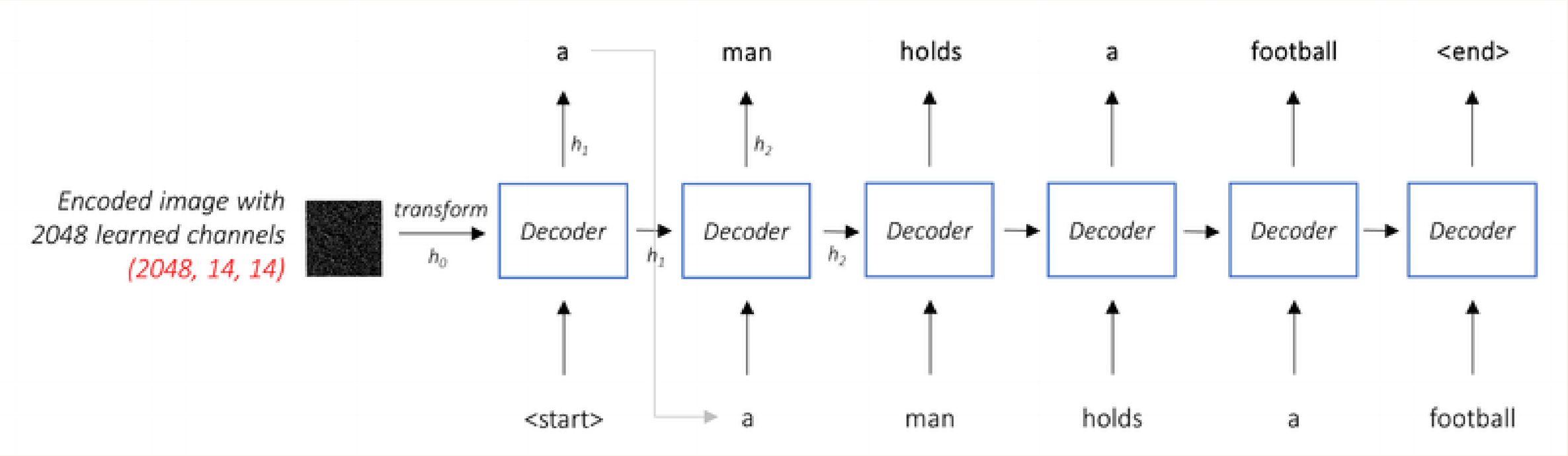


Encoded image with
2048 learned channels
(2048, 14, 14)

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

RNN

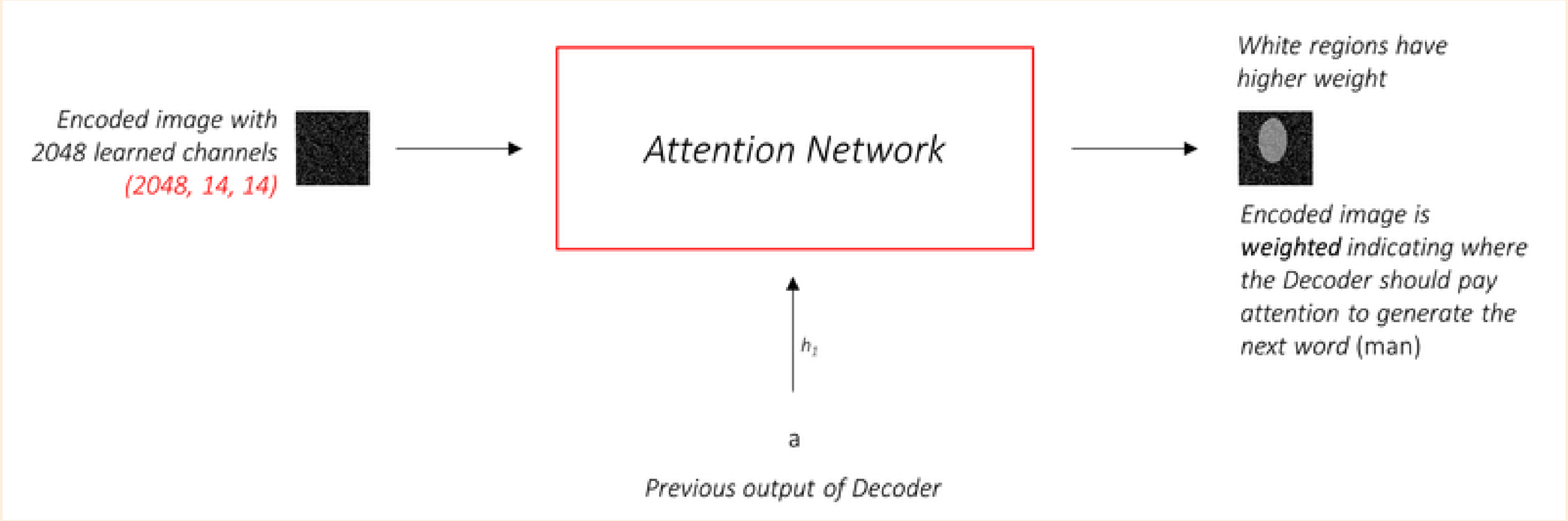
(without attention)



(LSTM as an example)

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

Attention net

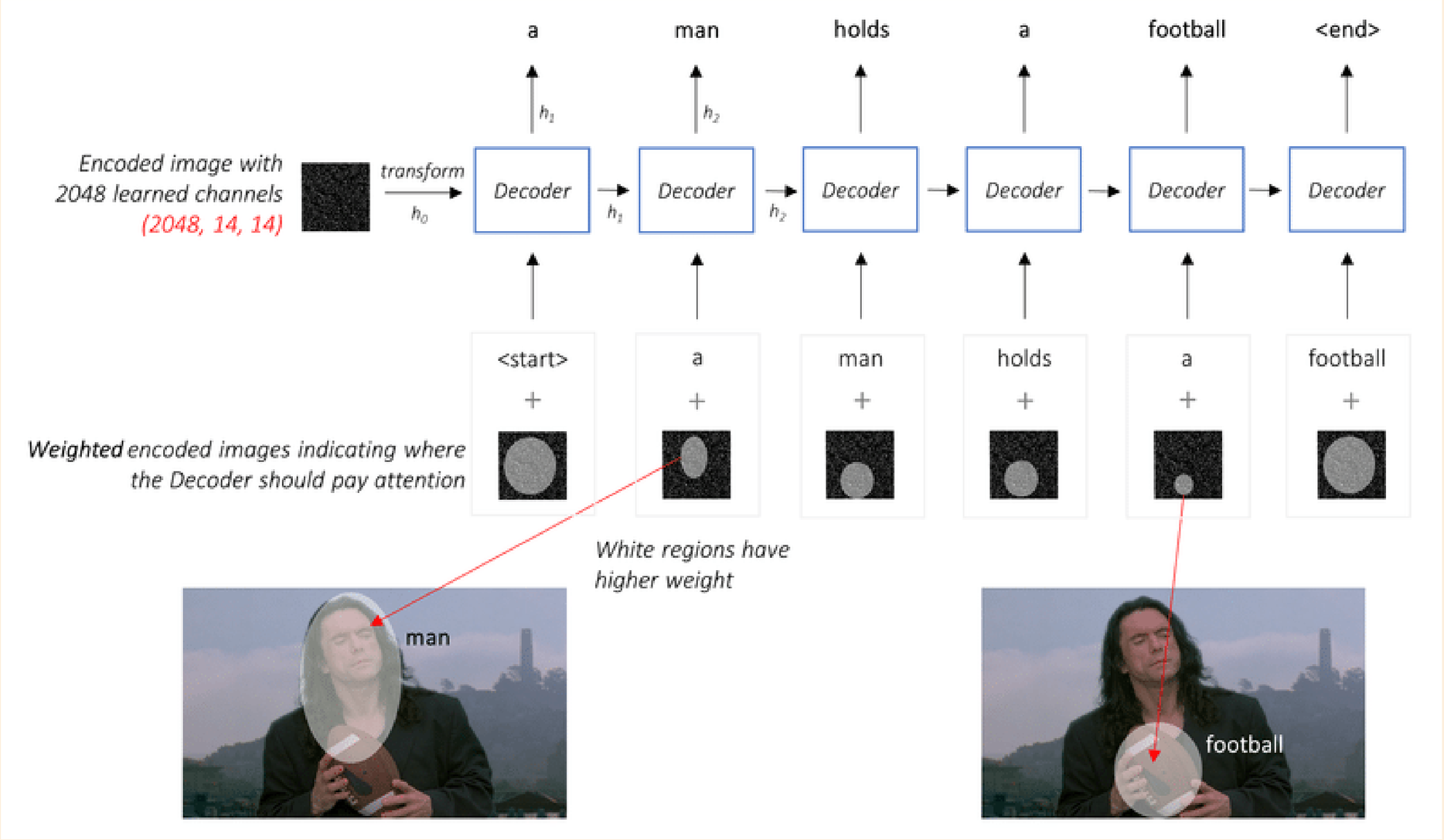


Soft Attention: the weights of the pixels add up to 1. If there are P pixels in our encoded image, then at each timestep t:

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

RNN

(with attention)



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

(LSTM as an example)

All together

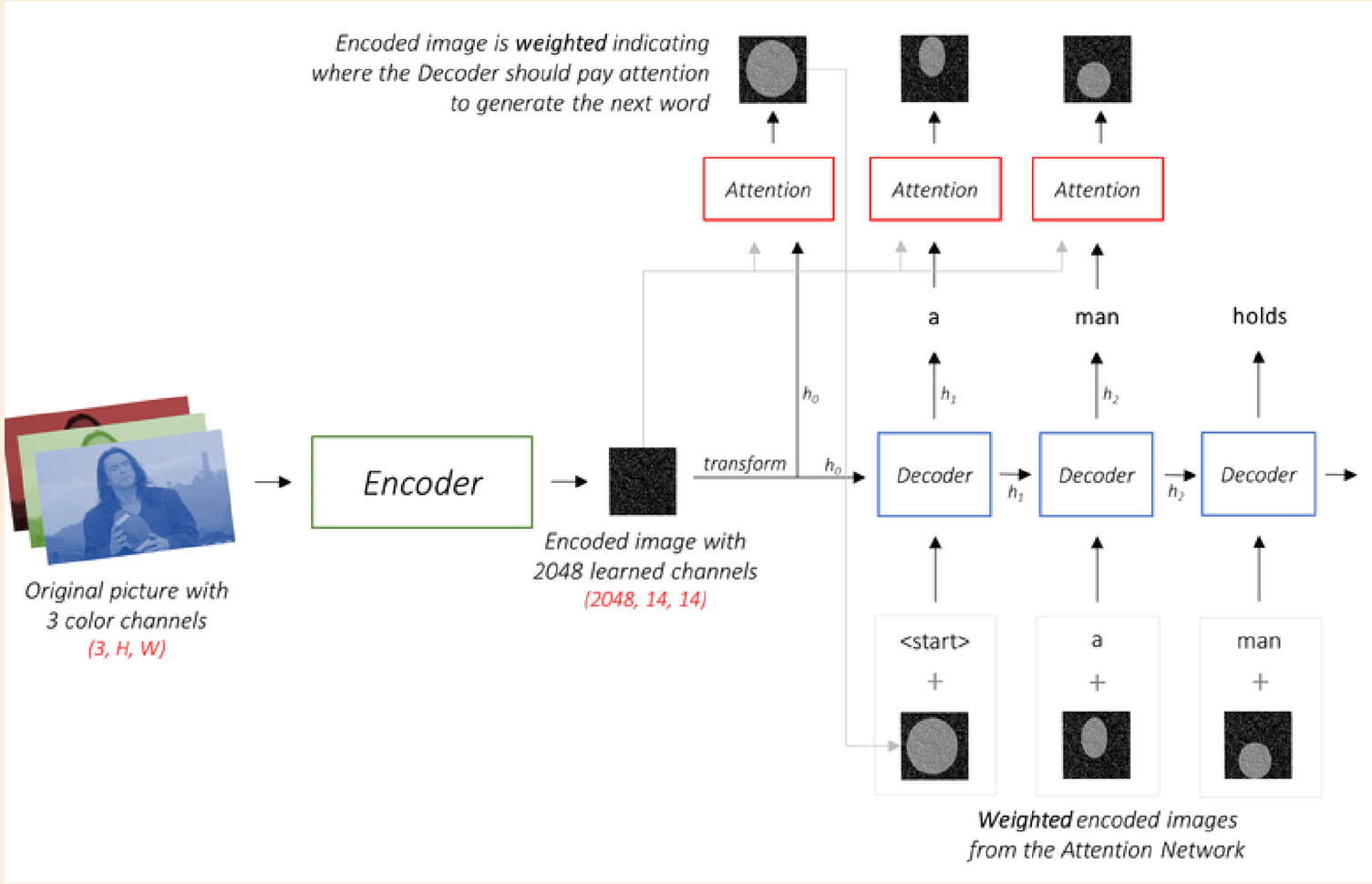
1. Image transformations and normalisation to suit CNN requirements

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:



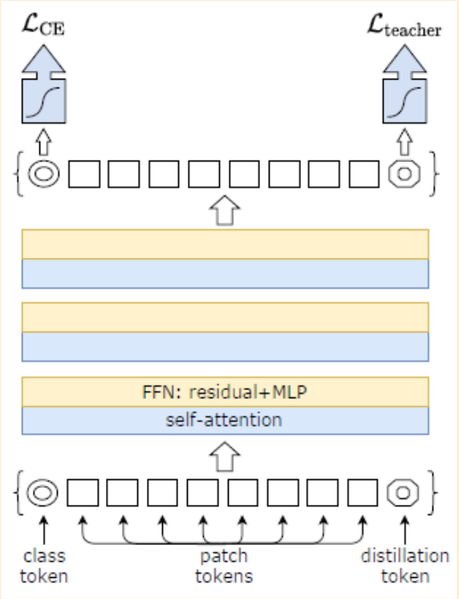
- *the encoded image and the previous hidden state is used to generate weights for each pixel in the Attention network.*
- *the previously generated word and the weighted average of the encoding are fed to the LSTM Decoder to generate the next word*

Implemented pipeline

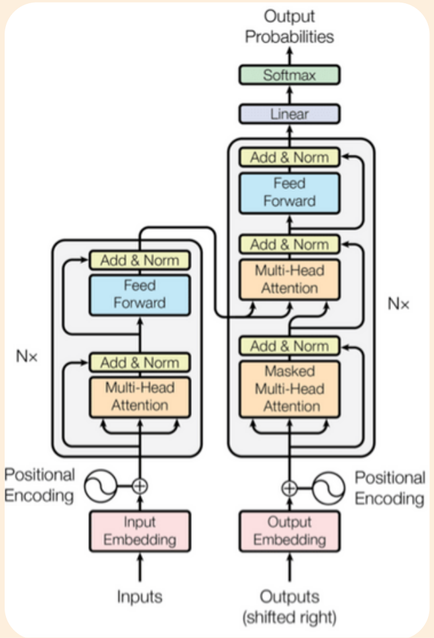
Frameworks/Backbones

01 VGG +LSTM	01 VGG + TRANSFORMER
02 DENSENET161 + LSTM	02 DENSENET161 + TRANSFORMER
03 DEIT + LSTM	03 DEIT + TRANSFORMER

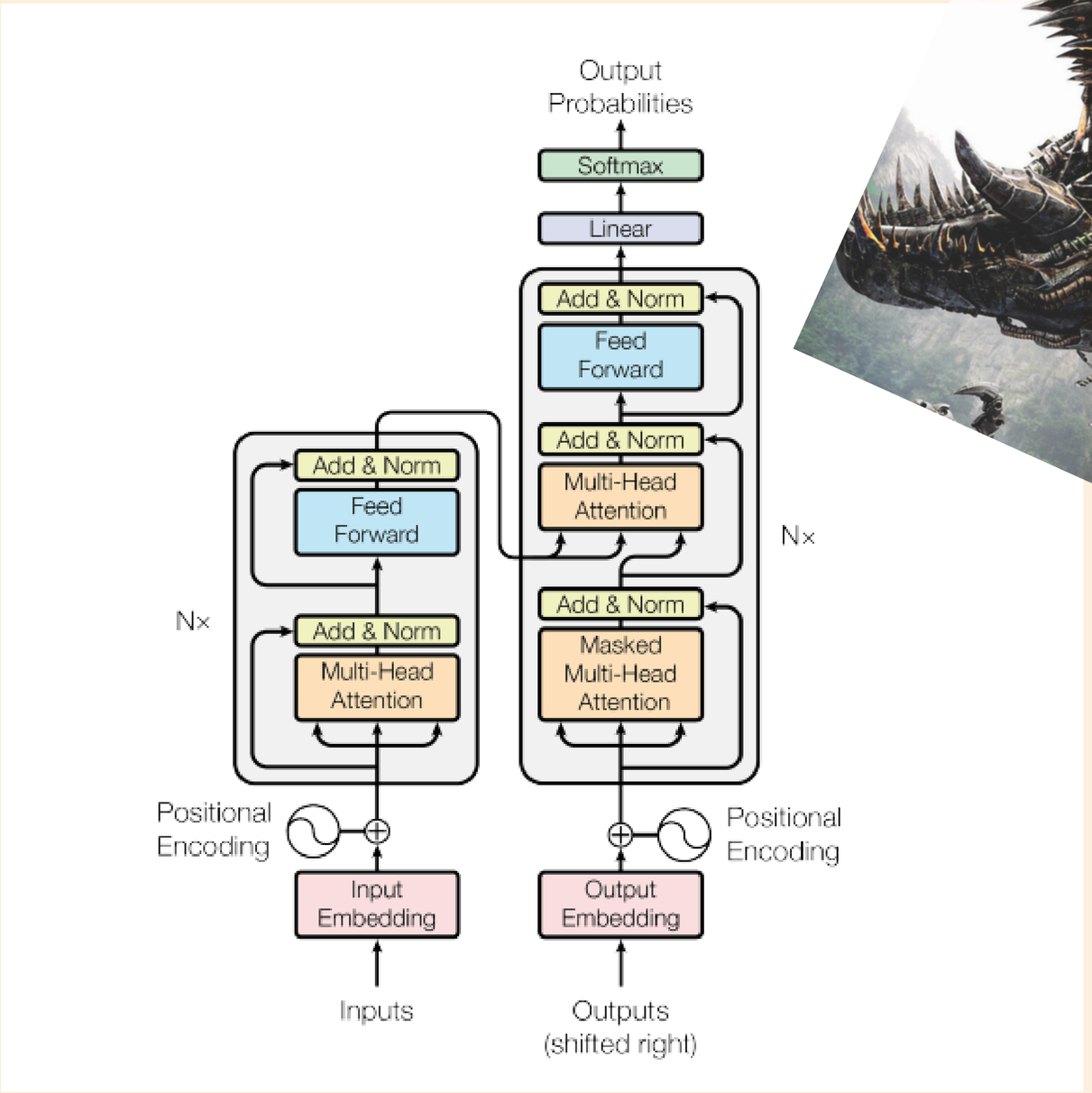
DEIT FRAMEWORK



TRANSFORMER FRAMEWORK

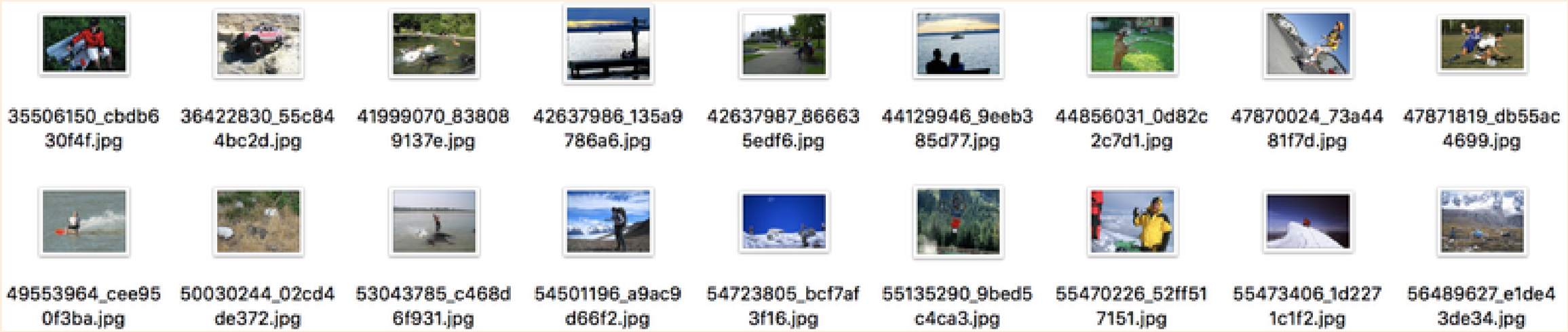


Transformers

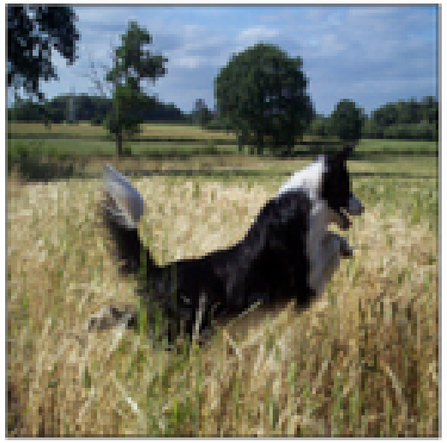


Flickr8k Data

- *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 .

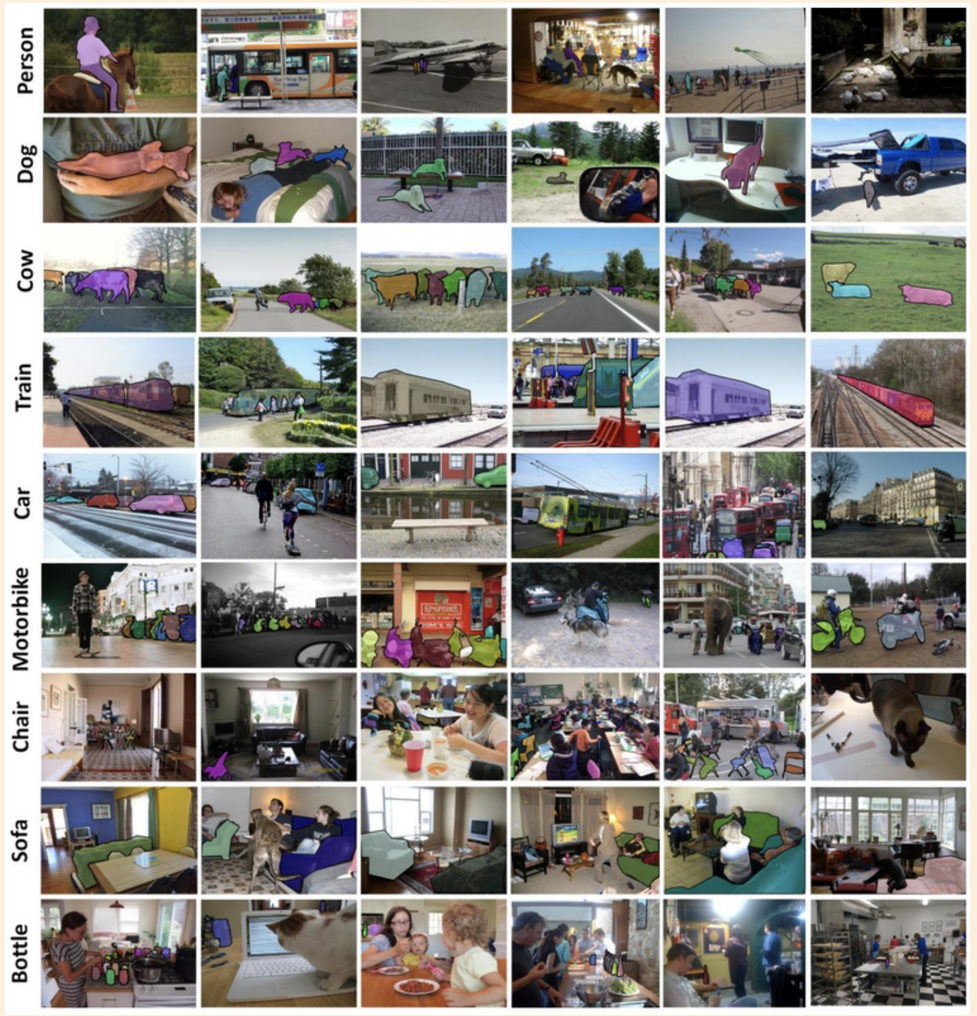
a smiling young girl in braids is playing ball .

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

a girl in a white dress .

COCO Data

- *Is a large-scale object detection, segmentation, key-point detection, and captioning dataset*
- ***328K** images*
- *The first version of MS COCO dataset was released in 2014. It contains 164K images split into training (83K), validation (41K) and test (41K) sets*
- *We used **COCO 2014 test set** for validation*



COCO Data examples

a laughing girl standing next to a cake and a wood carrousel horse on a kitchen counter.
a young lady laughing in a kitchen with a cake in front of her on the counter.
a girl standing in a kitchen holding a knife.
a young girl is smiling in the kitchen with a cake.
a woman having a laugh by a cake with a knife in hand.

a group of men cutting a giant sheet cake.
a group of servicemen cutting a cake as others watch
a couple of people that are cutting a cake
a group of formally dressed men cut a large decorated cake at a ceremony.
several people in navy uniforms cutting a cake

Images preprocessing

1. Resize transform to suit CNN requirements:

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

3. For train set:

- *Random crop*
- *Random horizontal flip*

Text preprocessing

- *Tokenization*
- *Used pre-trained GloVe*
- *We tried and used 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

Table 1. BLEU scores of different models on COCO val2014 dataset, %

Model	BLEU1	BLEU2	BLEU3	BLEU4
DenseNet161+LSTM(69ep)	27.48	9.72	3.93	2.14
VGG16+LSTM(3ep)	26.04	9.96	4.16	2.21
VGG16+Transformer	52.76	33.04	20.27	12.39
DenseNet161+Transformer	30.24	12.80	6.00	3.32
DeiT+LSTM	25.96	9.55	8.32	9.36
DeiT+Transformer	29.94	12.90	6.19	4.06

Table 2. BLEU scores of different models on Flickr8k test set, %

Model	BLEU1	BLEU2	BLEU3	BLEU4
DenseNet161+LSTM(69ep)	55.27	30.76	17.11	10.23
VGG16+LSTM(3ep)	55.41	34.34	21.13	13.29
VGG16+Transformer	52.76	33.04	20.27	12.39
DenseNet161+Transformer	65.98	44.79	30.04	19.75
DeiT+LSTM	53.48	31.06	17.61	10.61
DeiT+Transformer	62.57	44.09	35.11	29.80

Examples

DenseNet161 + Transformer

generated capture: a man riding a bike on a street .
GT: Several motorcycles riding down the road in formation.

Figure 4. Good image caption

generated capture: a man is sitting on a bench with a small dog .
GT: A cute kitten is sitting in a dish on a table.

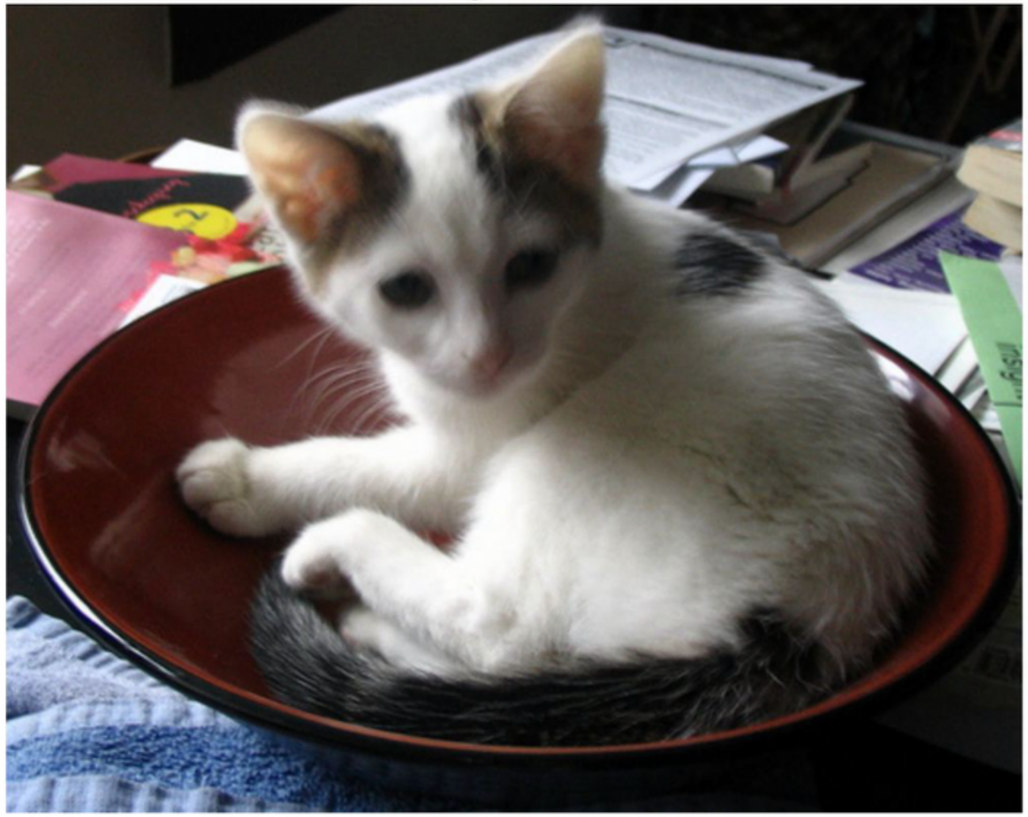


Figure 5. Bad image caption

Examples

DenseNet161 + LSTM

Gen.: a black and white dog is running through the grass .

GT: A man is sitting on a black and brown dog .

BLEU@4: 0.39

Figure 6. Good image caption

Gen.: a man in a red shirt is playing with a red ball .

GT: A bicyclist rides on ramps .

BLEU@4: 0.06

Figure 7. Bad image caption

Examples

VGG16 + LSTM

Gen.: a man is standing on a rock .

GT: A boy stands on a rock in a creek , holding a stick .

BLEU@4: 0.21

Figure 8. Good image caption

Gen.: a dog is running through the grass .

GT: A brown and black dog is jumping through a sprinkler .

BLEU@4: 0.09

Figure 9. Bad image caption

Examples

DeiT + LSTM

Gen.: a man in a blue wetsuit is surfing .

GT: a lone surfboarder jumping a wave on a white surfboard .

BLEU@4: 0.31

Gen.: a dog is digging a hole .

GT: A man skiing down a hill .

BLEU@4: 0.24

Examples

DeiT + Transformers

Gen.: a young boy wearing a striped shirt and a black shirt .

GT: A small young girl in a pink shirt drinking a large chocolate milkshake .

BLEU@4: 0.21

Gen.: a brown dog is standing in a grassy field .

GT: A brown dog is panting hard on grass during a sunny day .

BLEU@4: 0.24

Conclusion

- *The best performing model is Densenet paired with Transformer*
- *The introduction of a transformer based decoder eliminate the need for recursion.*
- *Multi-head attention and positional embeddings allows the ability to process sentences as a whole and learn relationships between.*
- *LSTM decoders provide a quicker computation time, albeit at the cost of performance.*

Thank you for your attention!

HAVE A GREAT DAY AHEAD.