Image Captioning using Attention

Final Project Discussion

Rupesh Yadav, 21i190004
Aakash Roy, 21i190007
Shoaib Ahmed, 21i190012
Srija Mukherjee, 21i190013

30th April, 2023

Problem Statement

The project aims to generate image captions using deep learning techniques like CNN and LSTM. We will also try to combine the attention mechanism to enable the algorithm to recognize image context by focusing on some specific parts and generate the appropriate sequence of words for the captions.

Related work

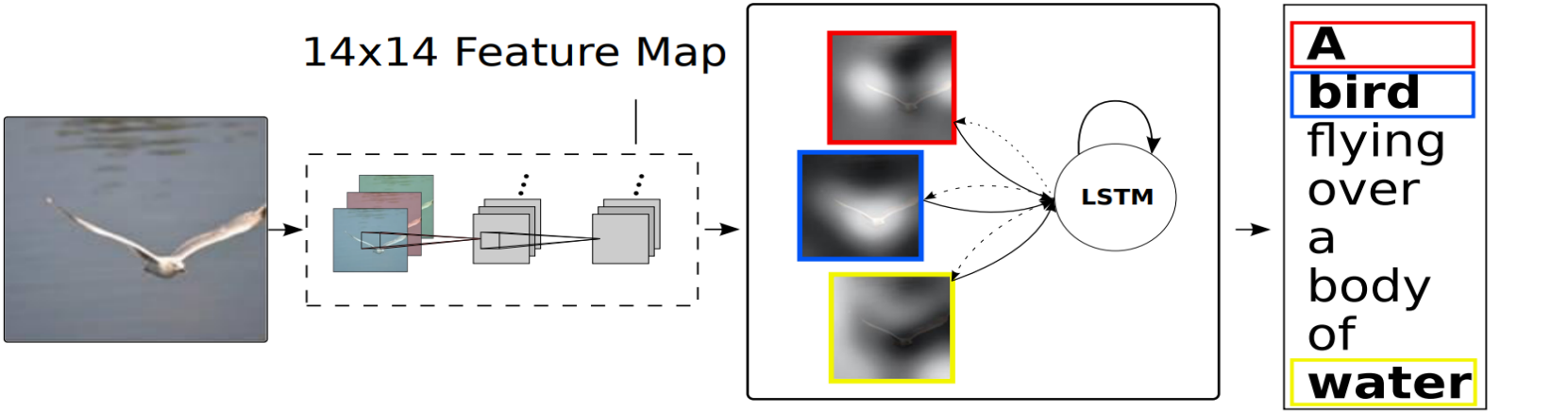
Prior to the use of NNs, two main approaches were dominant:

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

Moving to Neural Network based works

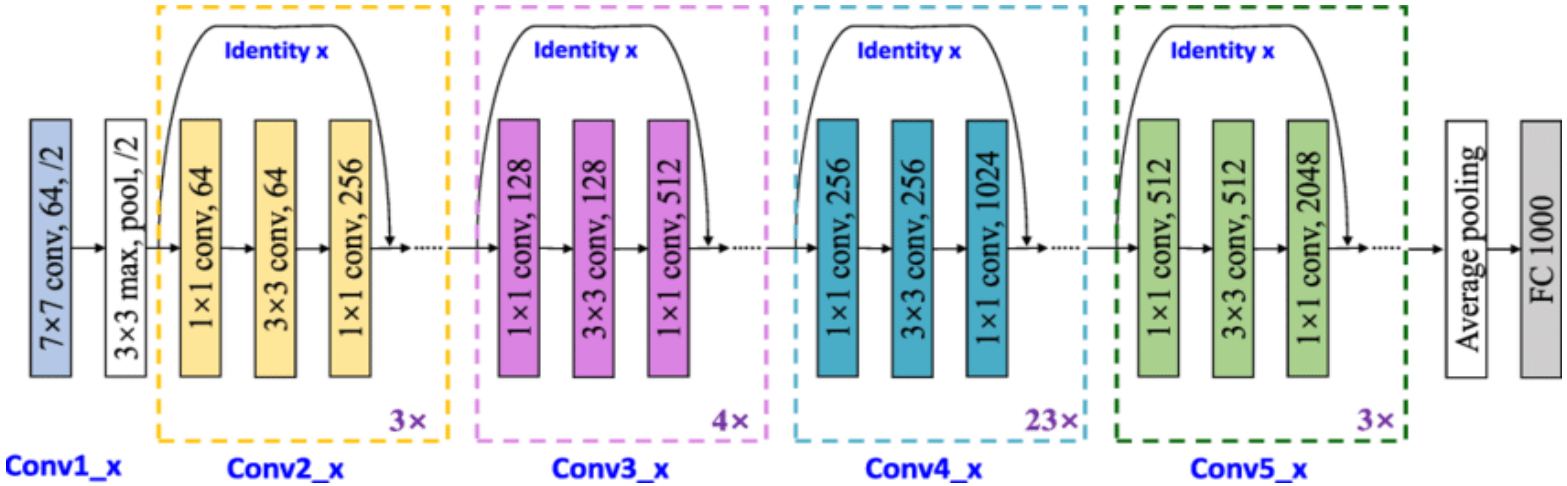
- Many methods for this task are based on RNNs and inspired by the successful use of sequence to sequence training with NNs for machine translation. [1]
- One major reason image caption generation is well suited to the encoder-decoder framework of machine translation is because it is analogous to “translating”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

Workflow


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

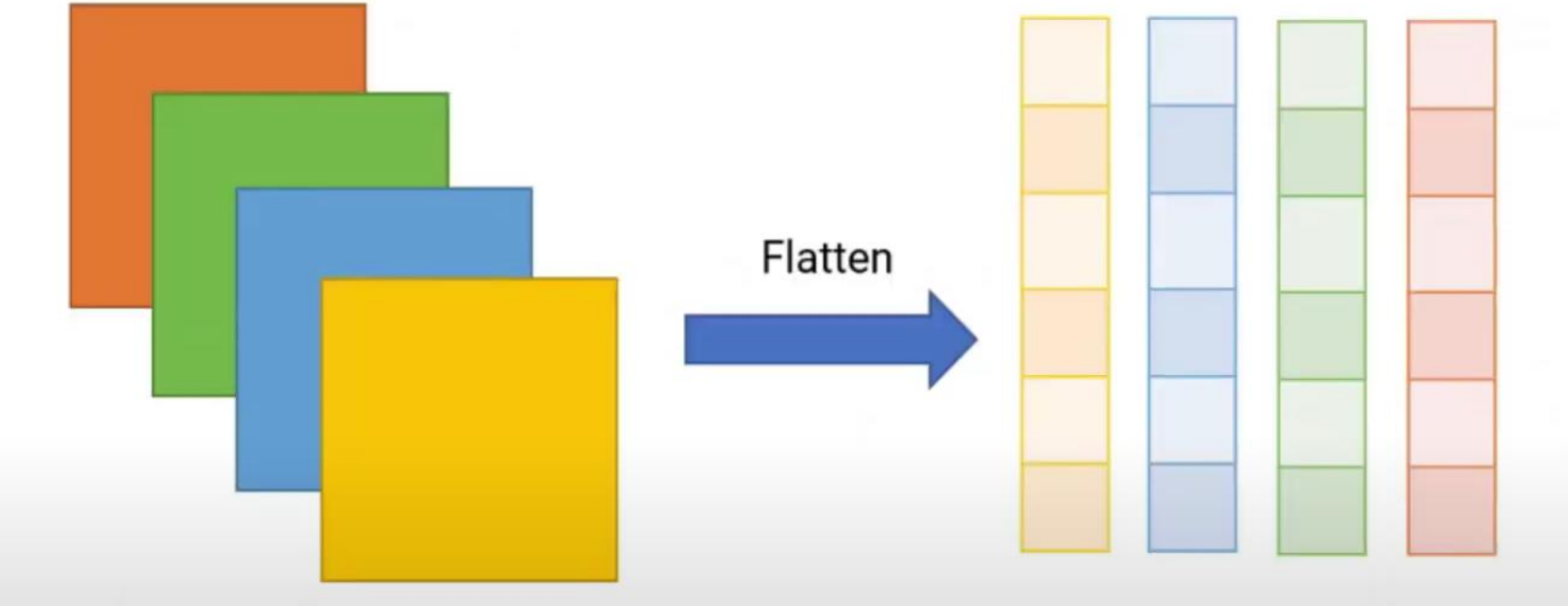
Architecture

Convolutional Feature Extraction



Architecture

Convolutional Feature Extraction

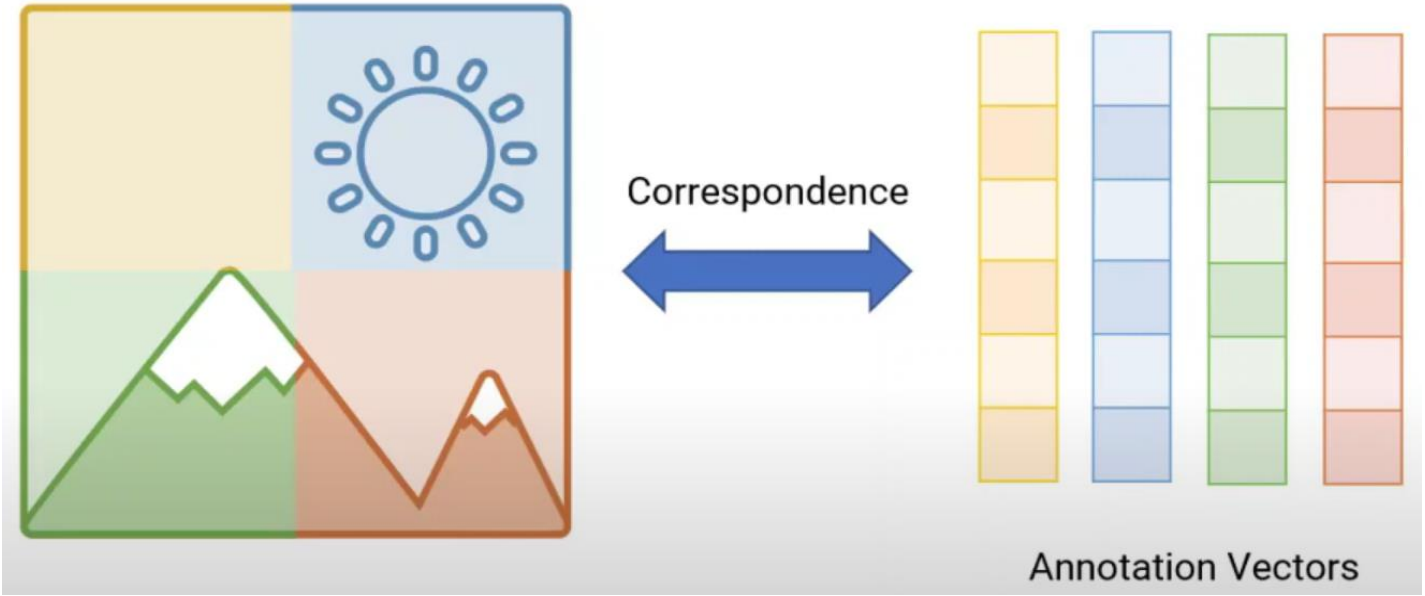


14*14*2048 feature maps

Annotation vectors

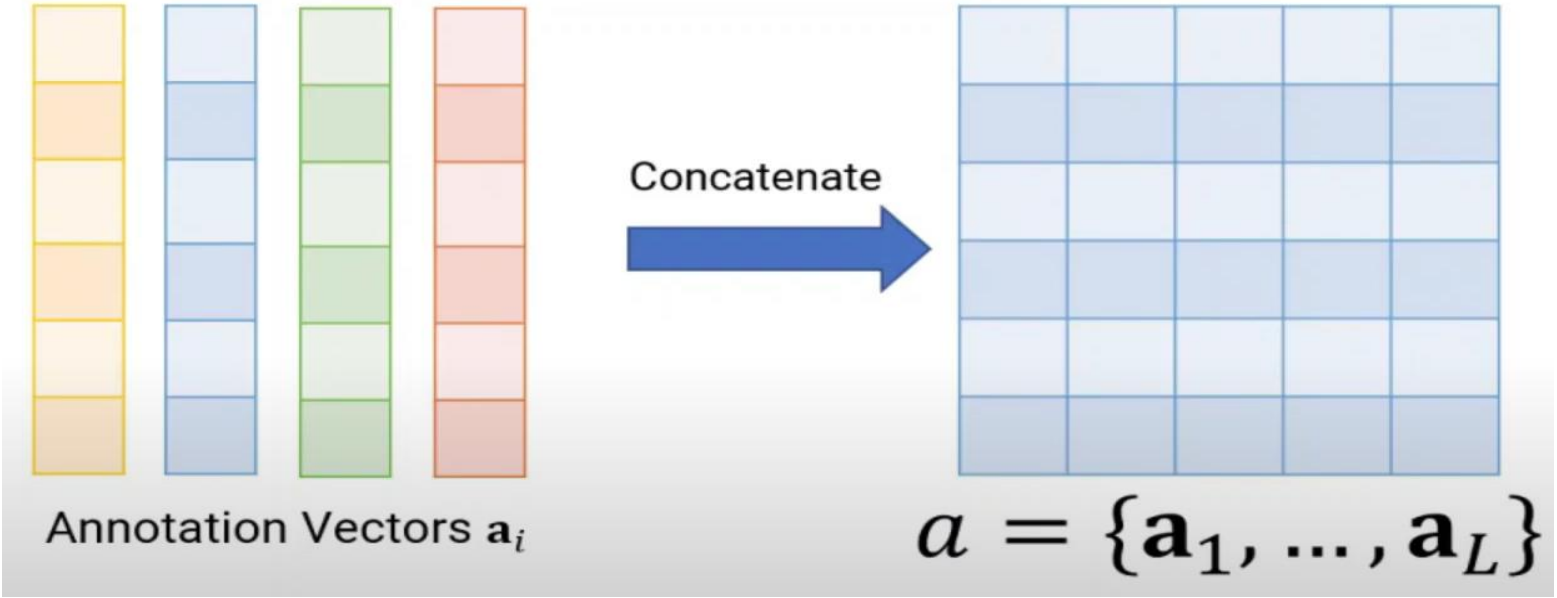
Architecture

Convolutional Feature Extraction

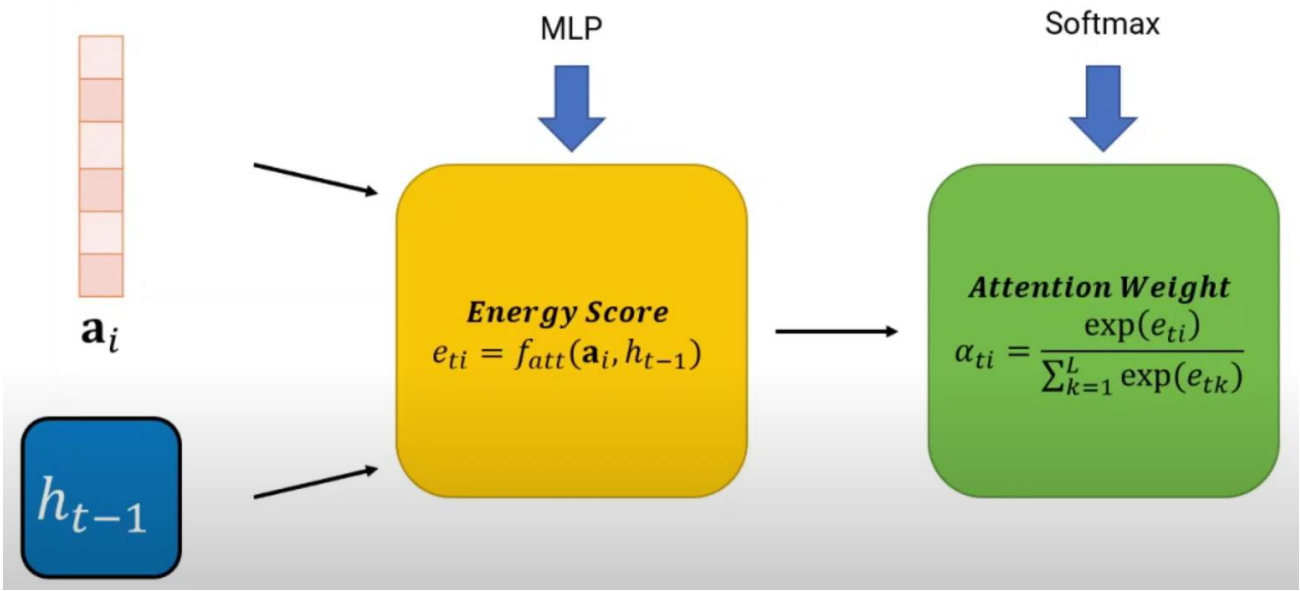


Architecture

Convolutional Feature Extraction



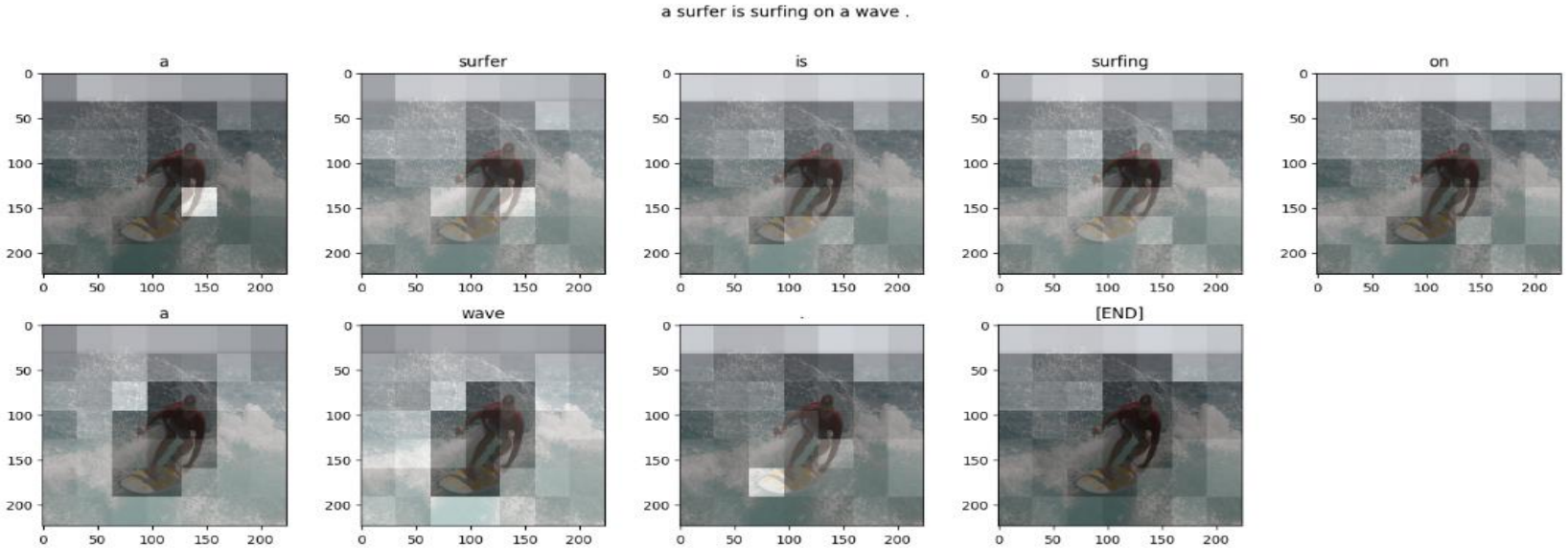
Architecture



Architecture

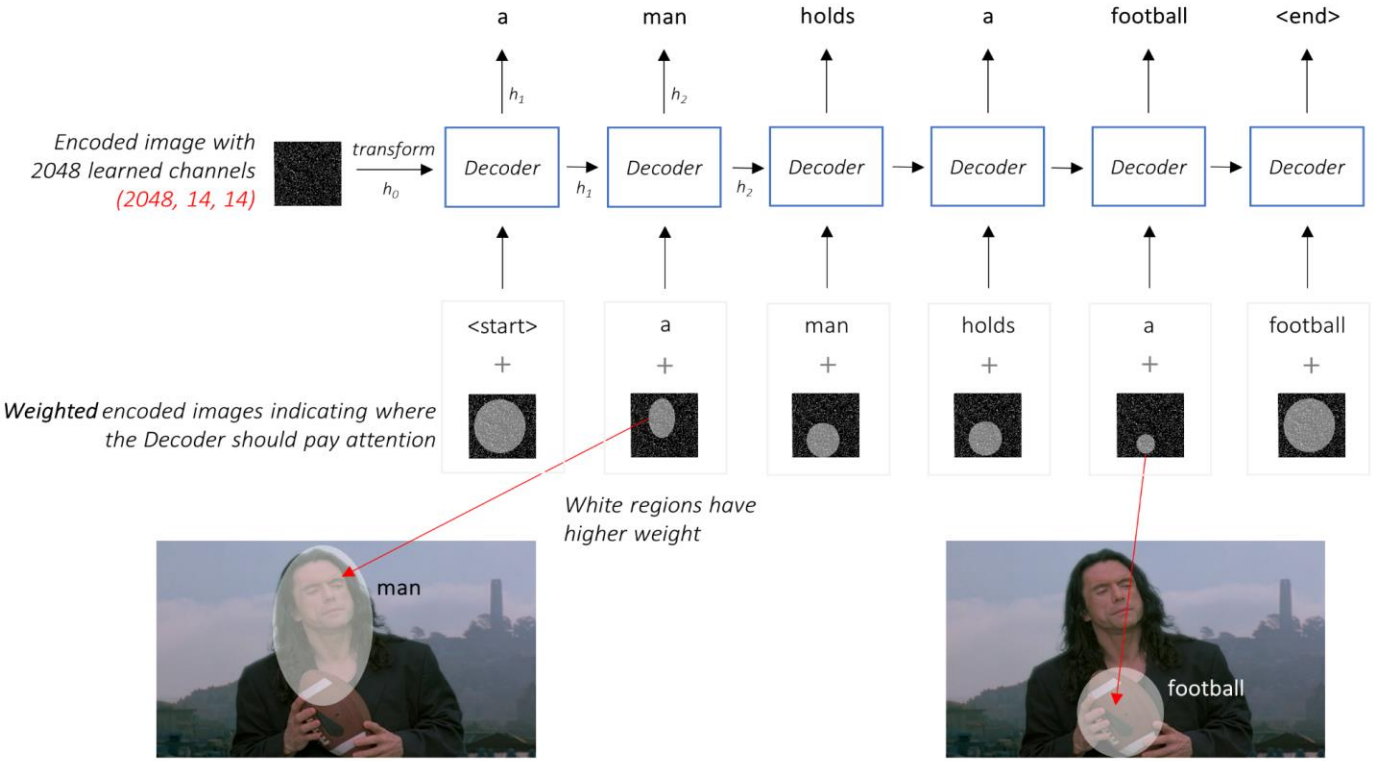
Attention Mechanism

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



Architecture

Idea of using Attention



Architecture

Attention Network

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

Attention Network

h_1

a

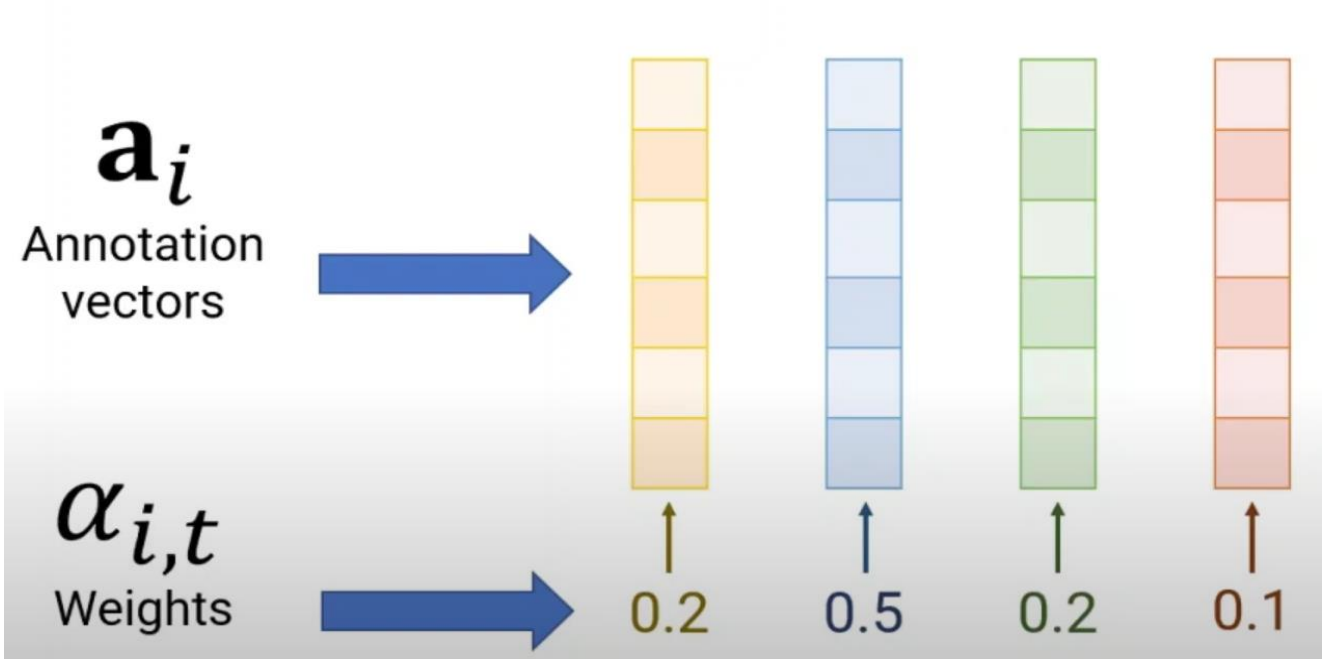
Previous output of Decoder

White regions have higher weight

*Encoded image is **weighted** indicating where the Decoder should pay attention to generate the next word* (man)

Architecture

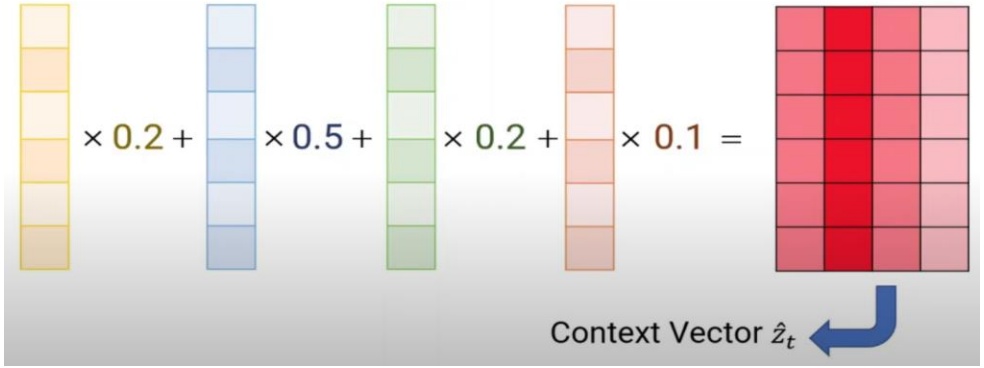
Soft Attention

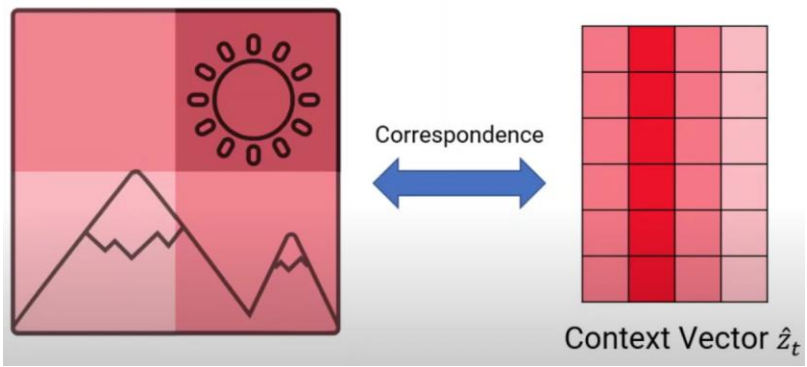


Architecture

Soft Attention

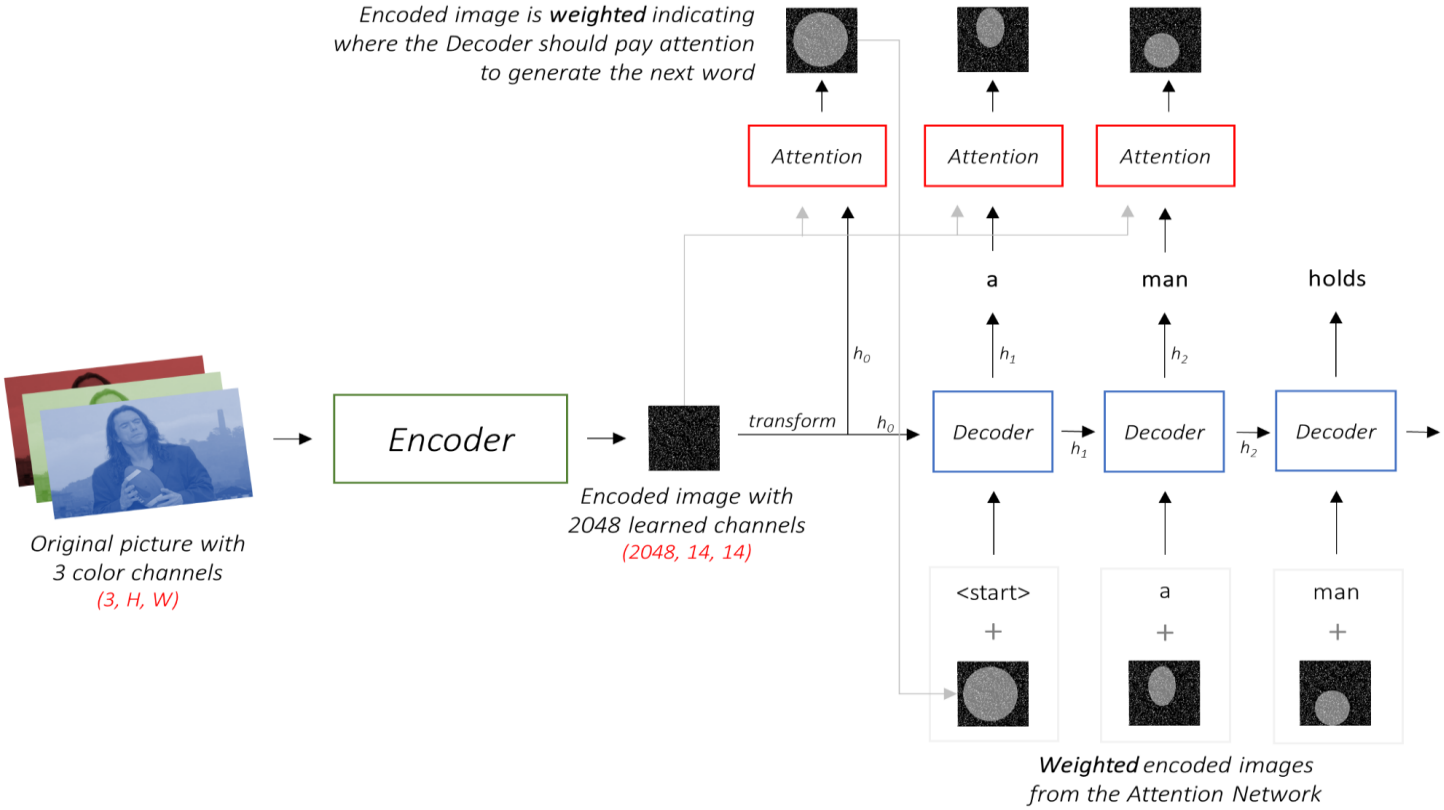
$$\hat{z}_t = \sum_{i=1}^{L} \alpha_{i,t} a_i$$





Architecture

Decoder with “Attention”



Architecture

Doubly Stochastic "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

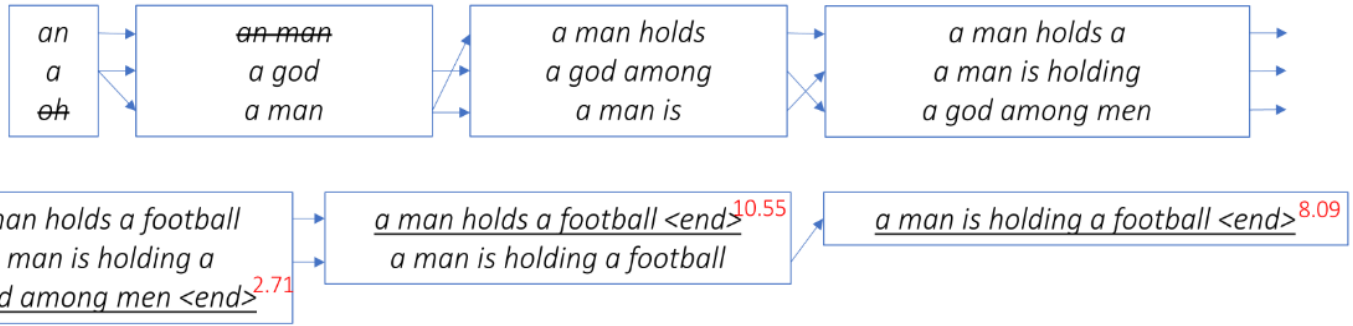
Loss Function: $L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_{i}^{L}(1 - \sum_{t}^{C} \alpha_{ti})^2$

Architecture

Beam Search

Beam Search with k = 3

Choose top 3 sequences at each decode step.
Some sequences fail early.
Choose the sequence with the highest score after all 3 chains complete.



Dataset (s)

Here we're using the **Microsoft COCO: Common Objects in Context**

- 330k+ images with 2.5M+ object instances labeled in 80 categories
- Widely used benchmark for training and evaluating deep learning models
- Useful for image captioning, retrieval, and scene understanding
- **Training Data Size (No. of Images) : 5,66,480**
- **Validation Data Size (No. of Images) : 25000**
- **Testing Data Size (No. of Images) : 25000**

- Used Andrej Karpathy's dataset containing image captions for COCO dataset
- We used 5 captions corresponding to each image

Technique

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

Technique

Experimental 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**

Technique

Experimental 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

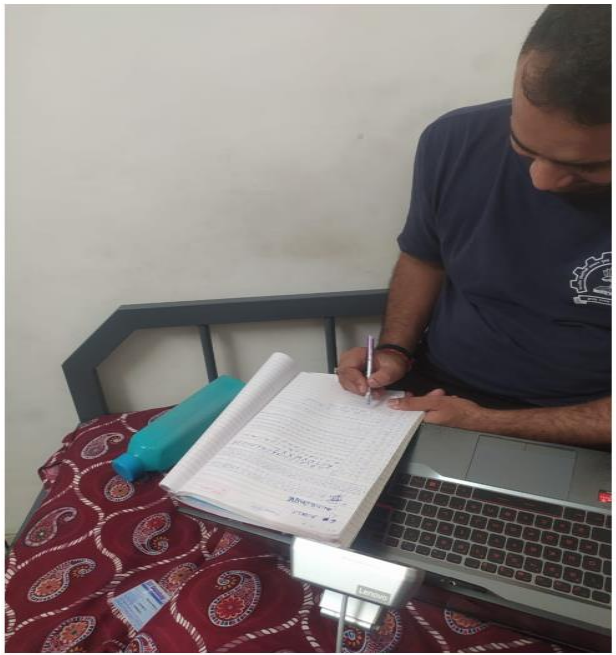
Results

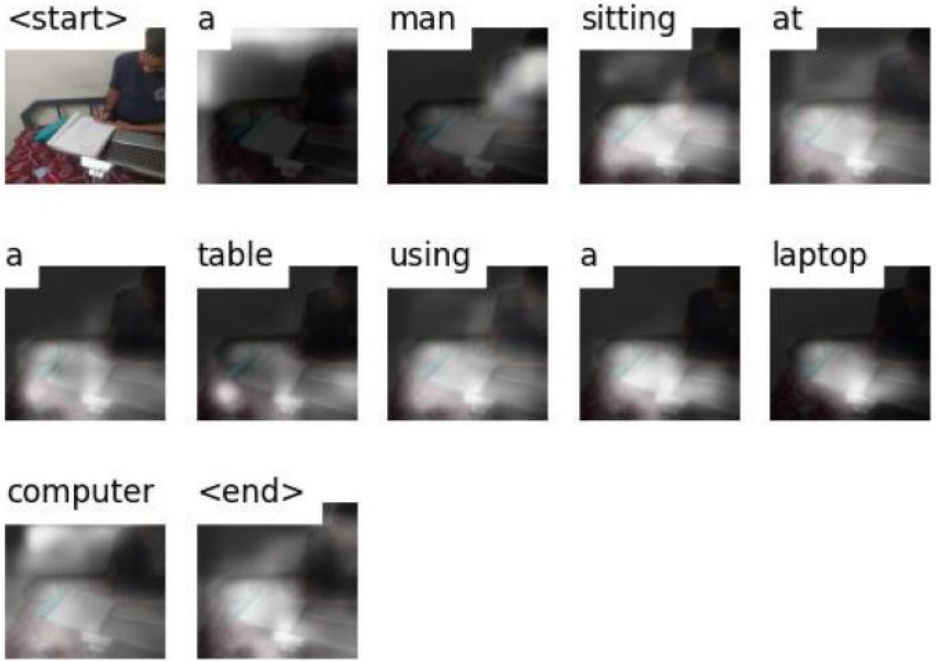
- **Instead of using just a single RNN for decoding purpose, we changed the model to multi layered RNN for the decoding purpose**
- **Specifically, we used 2 layered RNN**

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)	71.29|**72.05**	53.93|**55.26**	39.65|**41.11**	29.05|**30.39**	23.77|**24.24**	95.01|**97.88**	52.51|**53.34**
Our's (beam size 3)	72.97|**73.11**	56.16|**56.52**	42.48|**43.05**	32.18|**32.87**	24.47|**24.83**	100.10|**101.35**	53.93|**54.27**
Our's (beam size 5)	72.81|**72.49**	55.99|**55.80**	42.43|**42.46**	32.27|**32.34**	24.51|**24.67**	100.16|**100.68**	54.02|**53.89**

Results

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

Demo

Thank You