Image Caption Generator

- Yashaswi Shah

Why Image Captioning?

Aid to the blind - We can create a product, that describes real world scenario, which can help them to travel on roads.

Google Image Search - It can make Google Image Search as good as google search

CCTV Cameras : We can generate captions for CCTV videos and through these captions, we can raise alarms for malicious activities.

What do you see in this image?

Key Challenges

1. Accurate and Context relevant Images
2. Dataset
3. Evaluation Metrics

Literature Review

1. Image Caption Generator :
   https://www.ijitee.org/wp-content/uploads/papers/v10i3/C83830110321.pdf

   This paper discusses the traditional Image Captioning Model with CNN(pretrained) and LSTM.

2. Deep Learning based Automatic Image Caption Generation :
   https://ieeexplore.ieee.org/document/8978293
   This paper discusses the Image Captioning Model with attention mechanism vs Image Captioning without attention

3. Transformative Fusion: Vision Transformers and GPT-2 Unleashing New Frontiers in Image Captioning within Image Processing :
   https://www.researchgate.net/publication/376589260_Transformative_Fusion_Vision_Transformers_and_GPT-2_Unleashing_New_Frontiers_in_Image_Captioning_within_Image_Processing

Dataset

Flickr 8k Dataset : https://forms.illinois.edu/sec/1713398

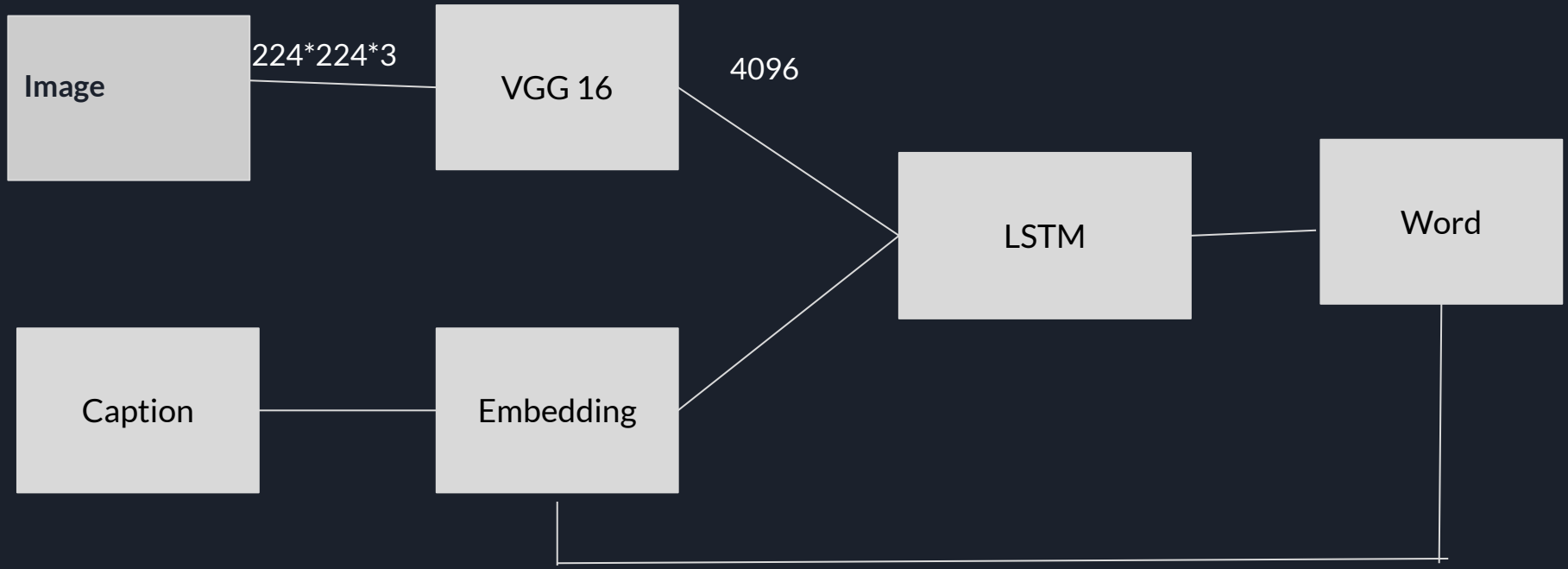
8091 Images with 5 caption for each images

	filename	index	caption
0	1000268201_693b08cb0e.jpg	0	a child in a pink dress is climbing up a set of stairs in an entry way .
1	1000268201_693b08cb0e.jpg	1	a girl going into a wooden building .
2	1000268201_693b08cb0e.jpg	2	a little girl climbing into a wooden playhouse .
3	1000268201_693b08cb0e.jpg	3	a little girl climbing the stairs to her playhouse .
4	1000268201_693b08cb0e.jpg	4	a little girl in a pink dress going into a wooden cabin .

Data Cleaning and Preprocessing

1. Merging file path to image name
2. Selecting 0th index caption
3. NLP tasks:
    a. Lower case
    ~~b. Stop words~~
    c. Punctuation
    d. <start> and <end> to every caption

Approach 1 : CNN + LSTM



Training and Test Data Set

80% Train data (6472 records)

20% Test data(1619 records)

Tokenization : Converted character captions to vectorized form for input to LSTM.

Vocabulary size : 4474

Caption with highest number of words : 30

Feature Extraction :

VGG 16 : Pretrained network model on Imagenet dataset

The VGG16 architecture is used as the encoding layer to extract important features from the image

Resized image : 224*224*3 as input to VGG

Output : 1 dimensional 4096 output vector as the last prediction layer of VGG 16 is removed from the model, as we need features.

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000

LSTM

Two Inputs :

1. Tokenized Captions : 30 words max
2. Feature Vector : 4096 dimension

Output : predicted t+1 word

- LSTM starts with <start> token and predicts next word with input as image vector and <start> and generates predicted word.
- Then, LSTM takes <start> and predicted word and generates next word. This continues until it generates <end>
- It uses greedy search to predict next word.

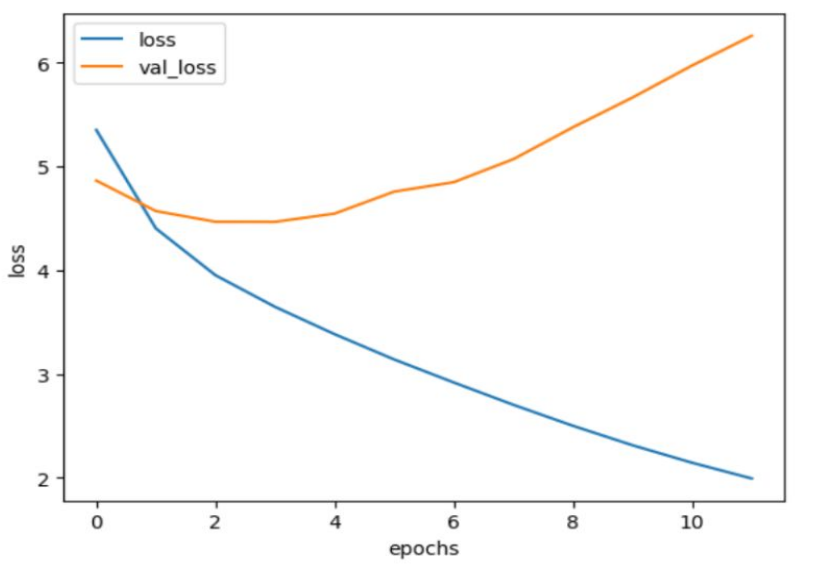
4474

Model: "model_1"

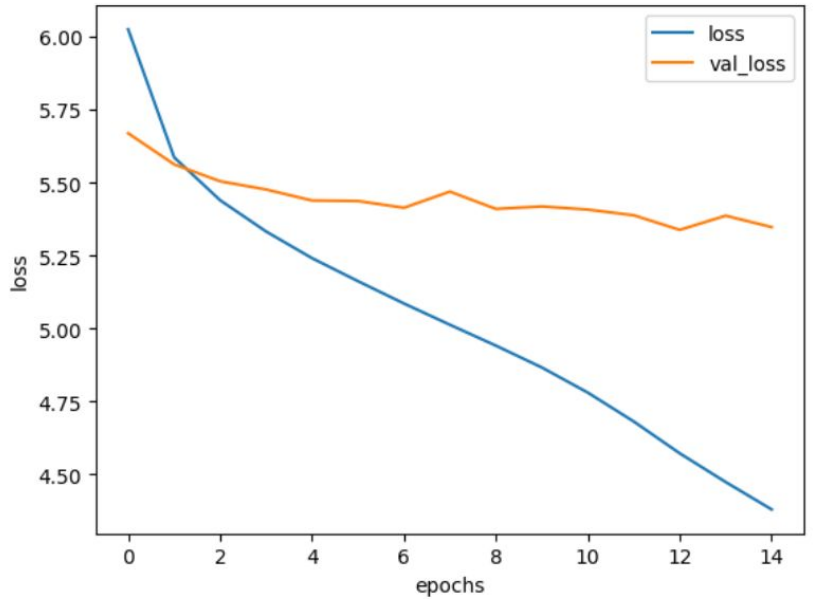
Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	[(None, 30)]	0	[]
embedding (Embedding)	(None, 30, 64)	286336	['input_3[0][0]']
input_2 (InputLayer)	[(None, 4096)]	0	[]
CaptionFeature (LSTM)	(None, 256)	328704	['embedding[0][0]']
ImageFeature (Dense)	(None, 256)	1048832	['input_2[0][0]']
add (Add)	(None, 256)	0	['CaptionFeature[0][0]', 'ImageFeature[0][0]']
dense (Dense)	(None, 256)	65792	['add[0][0]']
dense_1 (Dense)	(None, 4474)	1149818	['dense[0][0]']

Training

```
For epoch 12
```



Adam optimizer



SGD

BLEU Score

Predicted: man is skiing down mountain covered snow
Original: man snowboarding down mountain covered with snow
BLEU Score: 0.7142857142857143

Predicted: group of people standing in front of the background
Original: child shows other kids his still packaged toy
BLEU Score: 0

BLEU Score

Average Unigram Bleu score for 1618 test images for 1 gram : 0.16

Unigram Bleu Example :

Original : Man snowboarding down mountain covered with snow

Predicted : Man is skiing down mountain covered snow

Number of common words in predicted and original caption : 5/7(total words in original caption) = 0.7142

Approach 2 : Vision Encoder(Vit) Decoder(GPT2) Model

Image Captioning model uses encoder model to encode the image, after which an autoregressive language model i.e. the decoder model generates the caption

Hugging Face : https://huggingface.co/docs/transformers/model_doc/vision-encoder-decoder

Vision Transformer : Vision transformers are a type of transformers that perform visual-related tasks that include images. It treats image processing similarly to how words are processed in sentences, by dividing the image into patches and processing these patches sequentially. They are a transformer that also use attention mechanisms to find the relationships between input images. In this use case, they will connect our image with tokens or texts. This vector encodes visual information that the language model will use to generate relevant captions

GPT2 : GPT2 primarily designed for text generation tasks.

Vision Encoder(Vit) Decoder(GPT2) Model

Feature Extractor : Vision Transformer

feature_extractor = AutoFeatureExtractor.from_pretrained("google/vit-base-patch16-224-in21k")

Converts images into high-dimensional feature vectors that encapsulate the visual content of the image.

Tokenizer: GPT2

tokenizer = AutoTokenizer.from_pretrained(“gpt2”)

Processes the text, converting words into tokens that can be used by the decoder model (GPT-2) for generating captions.

The feature_extractor and tokenizer can be directly fed to the model

Beam Search for next word

K = 4

Beam Search is used to improve the quality of generated sequence.

the beam search expands all possible next steps and keeps the k most likely, where k is a user-specified parameter and controls the number of beams or parallel searches through the sequence of probabilities.

At each step, for each sequence in the beam, generate all possible next tokens and calculate their scores

set number of beams for beam search to 4

num_beams = 4

model.config.num_beams = num_beams

Fine Tuning

```
model = VisionEncoderDecoderModel.
from_encoder_decoder_pretrained(
   encoder_checkpoint,
   decoder_checkpoint
)
```

Epoch	Training Loss	Validation Loss
1	0.332300	0.258002
2	0.233000	0.249114
3	0.210300	0.250011
4	0.179400	0.255218
5	0.162000	0.262827

The output model consists of:

1. Feature Extractor
2. Tokenizer

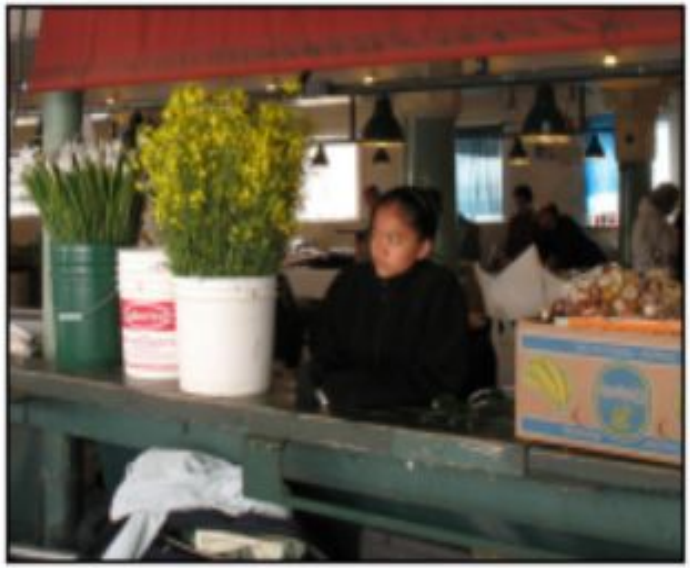
Which will be used to test images

Used 2seq Trainer from transformer library efficient training

BLEU Score

Average unigram for 1618 test images : 0.3

Predicted: a man doing a skateboard trick on a ramp.
Original: a man doing a skateboard trick on a skateboard ramp
BLEU Score: 0.90



Predicted: a man in a red jacket is reading a book in a bookstore.
Original: an oriental florist arranging flowers
BLEU Score: 0.00

Summary

CNN + LSTM	VIT + GPT2
CNN uses convolution filter to process data for feature extraction.	It divides images into patches and uses self attention for feature extraction
Implemented greedy search to predict next word in LSTM architecture	Implemented greedy search to predict next word
The overall unigram bleu score iis 0.16 for test dataset.	The overall bleu score is 0.3 for test dataset.

CNN + LSTM Vs VIT + GPT2

CNN + LSTM

Predicted: boy riding his bike is riding his bike on his bike
Original: boy on bike rides around the neighborhood
BLEU Score: 0.2727272727272727

VIT + GPT2

Predicted: a boy riding a bicycle on a paved road.
Original: a boy on a bike rides around the neighborhood .
BLEU Score: 0.50

Conclusion

Overall, compared to tradition CNN + LSTM model, Transformers have performed better in generating caption for images.