International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 9 Issue 12 ǁ 2021 ǁ PP. 42-52

Image Captioning

Ms. Vaishali, Rishabh Tiwari, Nimish Jain, Himanshu Tyagi, Tanishaq Jain
Assistant Professor, Department of CSE
Dr. Akhilesh Das Gupta Institute of Technology and Management, New Delhi, India
Students, 7th semester Department of CSE
Dr. Akhilesh Das Gupta Institute of Technology and Management, New Delhi, India
Corresponding Author: Ms. Vaishali

Abstract. *With the development of deep learning, the combination of computer vision and natural language process has aroused great attention in the past few years. Image captioning is a representative of this filed, which makes the computer learn to use one or more sentences to understand the visual content of an image. The meaningful description generation process of high level image semantics requires not only the recognition of the object and the scene, but the ability of analyzing the state, the attributes and the relationship among these objects. Though image captioning is a complicated and difficult task, a lot of researchers have achieved significant improvements. In this paper, we mainly describe three image captioning methods using the deep neural networks: CNN-LSTM based, CNN-CNN based and Reinforcement-based framework. Then we introduce the representative work of these three top methods respectively, describe the evaluation metrics and summarize the benefits and major challenges.*

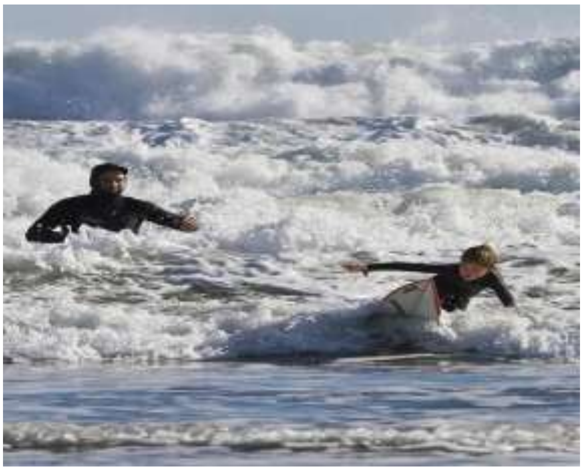
Date of Submission: 15-12-2021 Date of acceptance: 30-12-2021

I. INTRODUCTION

In the last few years, PC visionin picture preparing zone has gained huge ground, almost like picture arrangement [1] and visual perception [2]. Profiting by the advances of picture characterization and item identification, it gets conceivable to consequently produce a minimum of one sentences to grasp the visual substance of an image , which is that the issue referred to as Picture Inscribing. Creating total and normal picture portrayals consequently has enormous expected impacts, for instance , titles appended to news pictures, depictions related with clinical pictures, text- based picture recovery, data need to for dazzle clients, human-robot collaboration. These applications in picture inscribing have significant hypothetical and functional examination esteem. during this way, picture subtitling may be a more confounded however significant assignment within the time of artificialbrainpower.

Given another picture, an image inscribing calculation should yield an outline about this picture at a semantic level. as an example , in Fig. 1, the knowledge picture comprises of people , sheets and therefore the waves. within the base, there's a sentence depicting the substance of the picture—the articles arising within the picture,theactivity and therefore the scene are totally portrayed during thissentence.

For the image inscribing task, people can without much of a stretch comprehend the image substance and express it as characteristic language sentences as per explicitnecessities;nonetheless, for PCs, it requires the incorporated utilization of picture preparing, PC vision, normal language handling and other significant territories of exploration results. The test of picture subtitling isto plan a model which will completely utilize picture data to make more human-like rich picture portrayals. the many depiction age cycle of elevated level picture semantics requires not just the comprehension of things or scene acknowledgment within the picture, yet additionally the capacity to dissect their states, comprehend the connection among them and make a semantically and linguistically right sentence. it's as of now indistinct how the mind comprehends an image and puts together the visual data into a subtitle. Picture subtitling includes a profound comprehension of the planet and which things are remarkable pieces of everything.



A couple of individuals riding waves on top of boards.
Figure 1 An example of image captioning

Despite such challenges, the matter has achieved significant improvements over the past few years. Image captioning algorithms are typically divided into three categories. the primary category, as shown in Fig2. (a), tackles this problem using the retrieval-based methods, which first retrieves the closest matchingimages,thentransfertheirdescriptionsbecausethecaptionsofthequery images [3]. These methods can produce grammatically correct sentences but cannot adjust the captions consistent with the new image. The second category in Fig2. (b), typically uses template-based methods to get descriptions with predefined syntactic rules and slit sentences into several parts [4]. These methods first cash in of several classifiers to acknowledge the objects, also as their attributes and relationships in a picture , then use a rigid sentence template to make an entire sentence. Though it can generate a replacement sentence, these methods either cannot express the visual context correctly or generate flexible and meaningfulsentences.

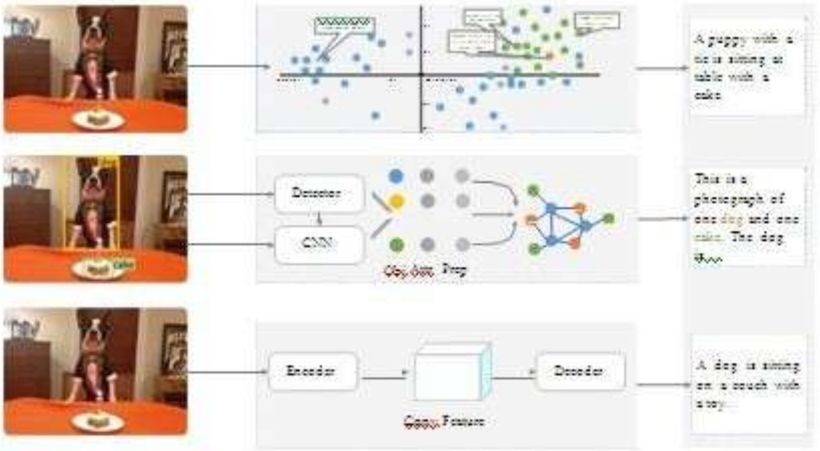

A puppy with a tie is sitting at table with a cake
Detector
CNN
Obj, Attr, Prep
This is a photograph of one dog and one cake. The dog
Encoder
Conv. Feature
Decoder
A dog is sitting on a couch with a toy.

Figure 2 Three catogories for image captioning

With the extensive application of deep learning, most up-to-date works fall under the third category called neural network-based methods in Fig2. (c). Inspired by machine learning’s encoder-decoder architecture [5], recent years most image captioning methods employ a Convolutional Neural Network (CNN) because the encoder and LSTM [6] to get captions [7], with the target to maximise the likelihood of a sentence given the visual features of a picture . Some methods are using CNN because the decoder and therefore the reinforcement learning because the decision-makingnetwork.

According to these different encoding and decoding methods, during this paper, we divide the image captioning methods with neural networks into three categories: CNN- LSTMbased, CNN-CNN based and reinforcement-based frameworkfor image captioning. within the next part, we'll mention their main ideas.

II. NEED FOR IMAGECAPTIONING

We must first understandhowimportantthisproblemistoworld scenarios. Let’ssee few applicationswhereananswertothepresentproblemare oftenvery useful.

- Self driving cars — Automatic driving is one among the most important challenges and if we can properly caption the scene round the car, it can provides a boost to the self driving system.
- Aid to the blind — we will create a product for the blind which can guide themtravelling on the roads without the support of anyone else. we will do that by first converting the scene into text then the text to voice. Both are now famous applications of DeepLearning.
- CCTV cameras are everywhere today, but along side viewing the planet , if we can also generate relevant captions, then we will raise alarms as soon as there's some malicious activity happening somewhere. this might probably help reduce some crime and/oraccidents.
- Automatic Captioning can help, make Google Image Search nearly as good as Google Search, as then every image might be first converted into a caption then search can be performed supported thecaption.

Image captioning is vital for several reasons. for instance , they will be used for automatic image indexing. Image indexing is vital for Content-Based Image Retrieval (CBIR) and therefore, it are often applied to several areas,including biomedicine,commerce,the military, education, digital libraries, and web searching. Social media platformslike Facebook and Twittercan directly generate descriptions from images.The descriptions can include where we are (e.g., beach, cafe), what we wear and importantly what we dothere.

III. FEATURES OF DEEP NEURALNETWORK

A neural network may be a network OR circuit of neurons, or during a modern sense, a man-madeneural network, composed of artificial neuronsor nodes[7]. Thus a neural network is either a biological neural network, made from real biological neurons, ora man-made neural network, for solving AI (AI) problems. The connections of the biological neuron are modeled as weights[5]. Apositiveweightreflectsanexcitatory connection, while negative values mean inhibitory connections. All inputs are modifiedby a weightand summed. This activity is referred to as a linear combination. Finally, an activationfunctioncontrols the amplitude of the output. for instance , a suitable range of output is typically between 0 and 1, or it mightbe −1 and 1[12].

These artificial networks could also be used for predictive modeling, adaptive control and applications where they will be trained via a dataset. Self-learning resulting from experience can occur within networks, which may derive conclusions from a posh and seemingly unrelated set of data.

A biological neural network consists of a groups of chemically connected or functionally associated neurons[12]. one neuron could also be connected to several other neurons and therefore thetotal numberof neurons and connections during a network could also beextensive. Connections, calledsynapses, are usually formed from axons to dendrites, though dendrodendritic synapsesand other connections are possible. aside from theelectrical signaling, there are other forms of signaling that arise from neurotransmitterdiffusion.

Artificial intelligence, cognitive modeling, and neural networks are information science paradigms inspired by the way biological neural systems process data. Artificialintelligence and cognitive modeling attempt to simulate some properties of biological neural networks[19].within theAIfield, artificialneural networks are applied successfully to speech recognition, image analysis and adaptive control, so as to construct software agents (in computer and video games) or autonomous robots[6]. Historically, digital computers evolved from the von Neumann model, and operate via the execution of explicit instructions via access to memory by variety of processors. On the opposite hand, the origins of neural networks are supported efforts to model information science in biological systems[9]. Unlike the von Neumann model, neural network computing doesn't separate memory andprocessing. Neural network theory has served both to raised identify how the neurons withinthe brain functionandtosupply theideaforeffortstomakeAI.

IV. WHAT IS IMAGECAPTIONING

Image captioningmay be aprocessof generating image descriptions foran in depthunderstanding the various elements of the image. the weather include the objects/person present withinthe image, the background or the setting of the environment during which the image is predicated[5], and the relationship of the objects and every one the entities of the image with among themselves and therefore the environmental setup during which they exist[14]. Language or any sort of communication are often used to describe the many amount of data present around us within the world. Similarly, the language are often wontto provideusable and important information from the scenes depicted in the images[3]. This results in a far better understanding of the scene by generating captions out of imagesand usingthe captions to thoroughly understandthe knowledge from the pictures . Several factors are required to urge an in-depth understanding of a picture like the spatial and semantic information about the varied entities present within the image, the backdrop during which the image is predicated and therefore the relationships between all the weather of the image. For generation of captions from images, the 2 major tasks that must be performed on the pictures , are as follows:

1. Gaining information about the planet.
2. Generating sentences to explain the Visionworld.

So different methods of Computer Vision and tongue Processing (NLP) are incorporated for extracting information from the pictures and representing them within the sort of meaningful sentences[20].

Generation of captions or description from images has beenagood area of research. The workinimagecaptiongeneration are often tracedback to the year 2010 where Ali Farhadi provided an introspective about how the captions are often generated and the way the pictures are often described with the assistance of sentences[8]. Many other methods followed, but the foremost recent work by P. Anderson and team achieved state-of-the-art performance onimagecaptioning tasks[11]. A deep analysis, comparison and disadvantages of varied works are discussed within the paper and a callfor using alternative methods has been made to enhance the performance on image captioningproblems.

V. ARCHITECTURE

Since the input consists of two parts, an image vector and a partial caption, we cannot use the Sequential API provided by the Keras library. For this reason, we use the Functional API which allows us to create Merge Models[16].

First, let’s look at the brief architecture which contains the high level sub-modules:

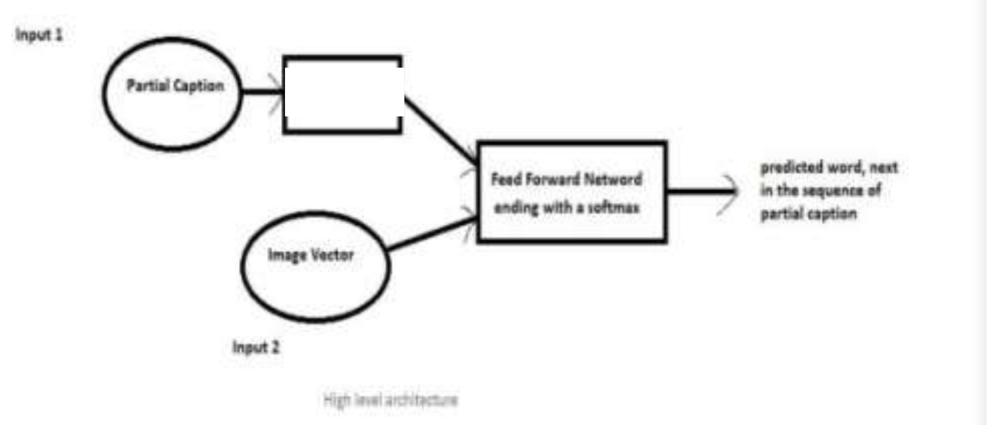
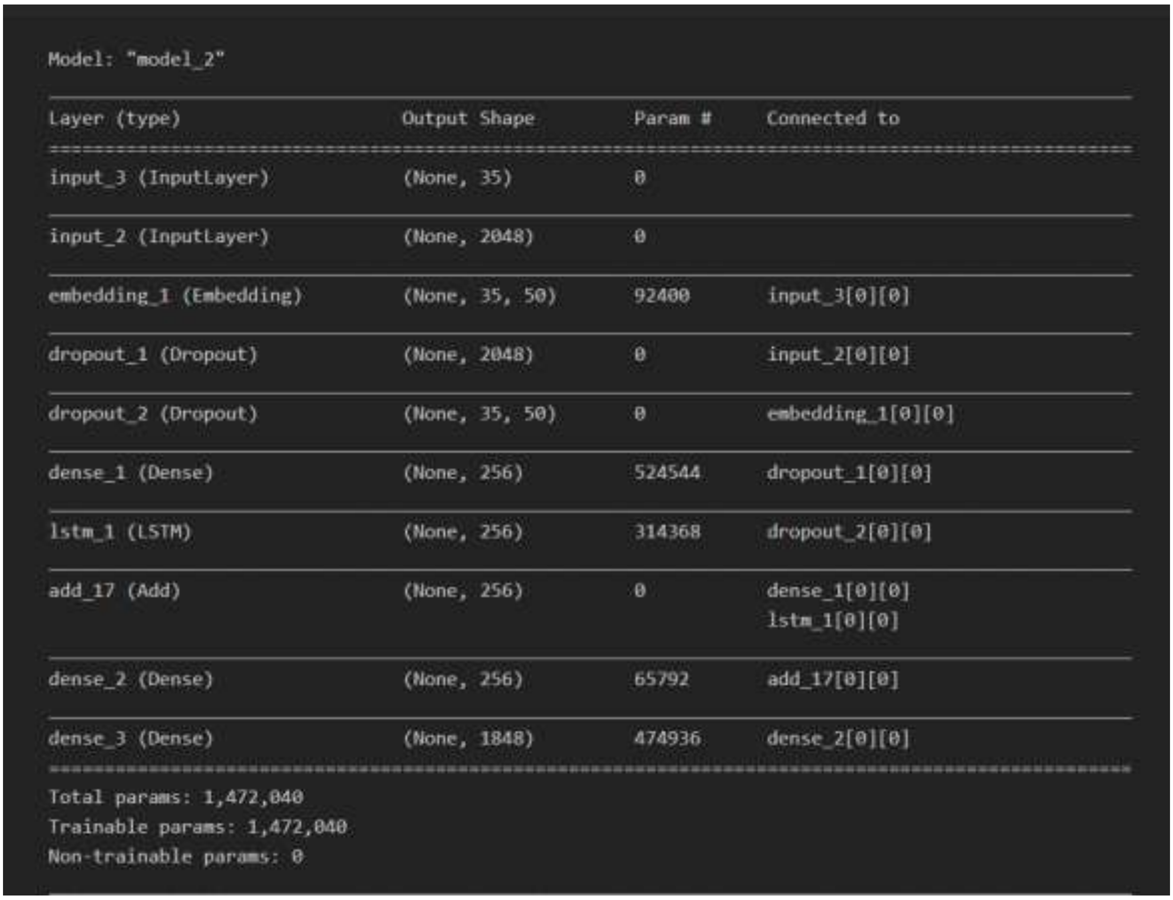

Input 1
Partial Caption
Feed Forward Netword ending with a softmax
predicted word, next in the sequence of partial caption
Image Vector
Input 2
High level architecture

Figure 2.7



```
Model: "model_2"
_____________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
=====================================================================================================
input_3 (InputLayer)            (None, 35)           0
_____________________________________________________________________________________________________
input_2 (InputLayer)            (None, 2048)         0
_____________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 35, 50)       92400       input_3[0][0]
_____________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 2048)         0           input_2[0][0]
_____________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 35, 50)       0           embedding_1[0][0]
_____________________________________________________________________________________________________
dense_1 (Dense)                 (None, 256)          524544      dropout_1[0][0]
_____________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 256)          314368      dropout_2[0][0]
_____________________________________________________________________________________________________
add_17 (Add)                    (None, 256)          0           dense_1[0][0]
                                                                 lstm_1[0][0]
_____________________________________________________________________________________________________
dense_2 (Dense)                 (None, 256)          65792       add_17[0][0]
_____________________________________________________________________________________________________
dense_3 (Dense)                 (None, 1848)         474936      dense_2[0][0]
=====================================================================================================
Total params: 1,472,040
Trainable params: 1,472,040
Non-trainable params: 0
_____________________________________________________________________________________________________
```

The **LSTM (Long Short Term Memory)** layer is nothing but a specialized Recurrent Neural Network to process the sequence input (partial captions in our case)[16]. Recall that we had created an embedding matrix from a pre-trained Glove model which we need to include in the model before starting thetraining:

Notice that since we are using a pre-trained embedding layer, we need to **freeze** it (trainable = False), before training the model, so that it does not get updated during the backpropagation.

Finally we compile the model using the adam optimizer Finally the weights of the model will be updated through backpropagation algorithm and the model will learn to output a word, given an image feature vector and a partial caption. So in summary, wehave:

Input_1 -> Partial Caption Input_2 -> Image feature vector

Output -> An appropriate word, next in the sequence of partial caption provided in the input_1 (or in probability terms we say **conditioned** on image vector and the partial caption)

Hyper parameters during training:

The model was then trained for 30 epochs with the initial learning rate of 0.001 and 3 pictures per batch (batch size). However after 20 epochs, the learning rate was reduced to 0.0001 and the model was trained on 6 pictures per batch.

This generally makes sense because during the later stages of training, since the model is moving towards convergence, we must lower the learning rate so that we take smaller steps towards the minima. Also increasing the batch size over time helps your gradient updates to be more powerful.

Time Taken: I used the GPU+ Gradient Notebook on www.paperspace.com and hence it took me approximately an hour to train the model. However if you train it on a PC without GPU, it could take anywhere from 8 to 16 hours depending on the configuration of yoursystem.

VI. SUMMARY OF IMAGE CAPTIONING

This research paper consists of-

1) The proposed methodology for the implementation of theproject.
2) This project has used Resnet50 instead of InceptionV3 and VGG16 while using various images.
3) This project has used LSTM rather than traditional methods of using RNN.

4) This project improves the accuracy of image captioning to 81% (approx).

VII.METHODOLOGY

Data Preprocessing — Images

Images are nothing but input (X) to our model. As you may already know that any input to a model must be given in the form of a vector.

We need to convert every image into a fixed sized vector which can then be fed as input to the neural network. For this purpose, we opt for **transfer learning** by using the InceptionV3 m odel[23].

This model was trained on Imagenet dataset to perform image classification on 1000 different classes of images. However, our purpose here is not to classify the image but just get fixed-length in formative vector for each image. This process is called **automatic feature engineering[12].**

Now, we pass every image to this model to get the corresponding 2048 length feature vector.

We save all the train features in a Python dictionary and save it on the disk using Pickle file, n amely "**encoded_train_images.pkl**" whose keys are image names and values are correspo nding 2048 length feature vector. This process might take an hour or two if you do not have a high end PC/laptop[17].

Similarly we encode all the test images and save them in the file "**encoded_test_images.pkl**".

Data Preprocessing — Captions

We must note that captions are something that we want to predict. So during the training period, captions will be the target variables (Y) that the model is learning to predict. But the prediction of the entire caption, given the image does not happen at once. We will predict the caption **word by word**. Thus, we need to encode each word into a fixed sized vector[21]

, but for now we will create two Python Dictionaries namely "wordtoix" (pronounced — word t o index) and "ixtoword" (pronounced — index to word)[14].

Stating simply, we will represent every unique word in the vocabulary by an integer (index). As seen above, we have 1652 unique words in the corpus and thus each word will be represented by an integer index between 1 to 1652[19].

These two Python dictionaries can be used asfollows:

wordtoix['abc'] -> returns index of the word 'abc' ixtoword[k] -> returns the word whose index is 'k'

There is one more parameter that we need to calculate, i.e., the maximum length of a caption . So the maximum length of any caption is 34.

Data Preparation using Generator Function

This is one of the most important steps in this case study. Here we will understand how to prepare the data in a manner which will be convenient to be given as input to the deep learning model.

Consider we have 3 images and their 3 corresponding captions as follows:

(Train image 1) Caption -> The black cat sat on grass

(Train image 2) Caption -> The white cat is walking on road (Train image 2) Caption -> The white cat is walking on road

(Test image) Caption -> The black cat is walking on grass

Now, let's say we use the **first two images** and their captions to **train** the model and the **third image** to **test** our model.

Now the questions that will be answered are: how do we frame this as a supervised learning problem?, what does the data matrix look like? how many data points do we have?, etc.

First we need to convert both the images to their corresponding 2048 length feature vector as discussed above. Let "**Image_1**" and "**Image_2**" be the feature vectors of the first two images respectively

Secondly, let's build the vocabulary for the first two (train) captions by adding the two tokens "startseq" and "endseq" in both of them: (Assume we have already performed the basic cleaning steps)

Caption_1 -> "startseq the black cat sat on grass endseq" Caption_2 -> "startseq the white cat is walking on road endseq"

vocab = {black, cat, endseq, grass, is, on, road, sat, startseq, the, walking, white} Let's give an index to each word in the vocabulary:

black -1, cat -2, endseq -3, grass -4, is -5, on -6, road -7, sat -8, startseq -9, the -10, walking -
11, white -12

Now let's try to frame it as a **supervised learning problem** where we have a set of data
points D = {Xi, Yi}, where Xi is the feature vector of data point 'i' and Yi is the corresponding target variable.

Let's take the first image vector **Image_1** and its corresponding caption "**startseq the black cat sat on grass endseq**". Recall that, Image vector is the input and the caption is what we need to predict. But the way we predict the caption is as follows:

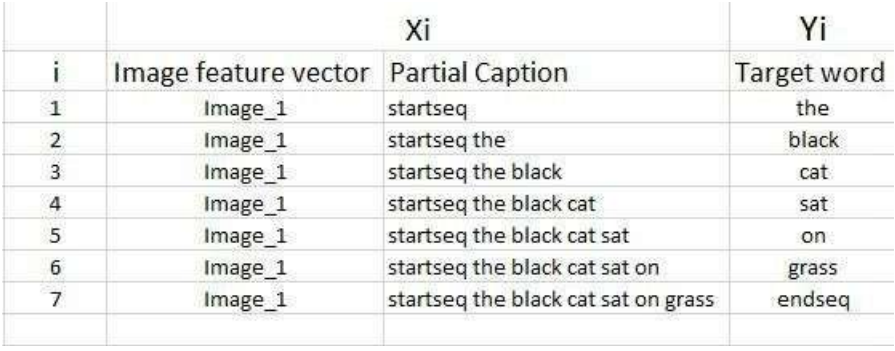
For the first time, we provide the image vector and the first word as input and try to predict the second word, i.e.:

Input = Image_1 + 'startseq'; Output = 'the'

Then we provide image vector and the first two words as input and try to predict the third word, i.e.:

Input = Image_1 + 'startseq the'; Output = 'cat' And so on…

Thus, we can summarize the data matrix for one image and its corresponding caption as follows:

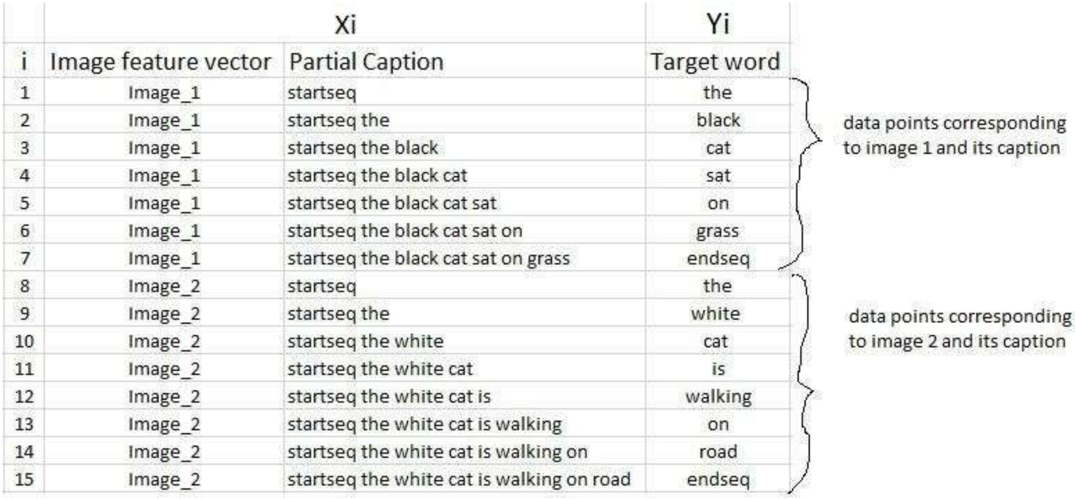


	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	startseq	the
2	Image_1	startseq the	black
3	Image_1	startseq the black	cat
4	Image_1	startseq the black cat	sat
5	Image_1	startseq the black cat sat	on
6	Image_1	startseq the black cat sat on	grass
7	Image_1	startseq the black cat sat on grass	endseq

Table 2.1 Data points corresponding to one image and its caption

It must be noted that, one image+caption is **not a single data point** but are multiple data points depending on the length of the caption.

Similarly if we consider both the images and their captions, our data matrix will then look as
follows:



	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	startseq	the
2	Image_1	startseq the	black
3	Image_1	startseq the black	cat
4	Image_1	startseq the black cat	sat
5	Image_1	startseq the black cat sat	on
6	Image_1	startseq the black cat sat on	grass
7	Image_1	startseq the black cat sat on grass	endseq
8	Image_2	startseq	the
9	Image_2	startseq the	white
10	Image_2	startseq the white	cat
11	Image_2	startseq the white cat	is
12	Image_2	startseq the white cat is	walking
13	Image_2	startseq the white cat is walking	on
14	Image_2	startseq the white cat is walking on	road
15	Image_2	startseq the white cat is walking on road	endseq

data points corresponding to image 1 and its caption

data points corresponding to image 2 and its caption

Table 2.2

Data Matrix for both the images and captions

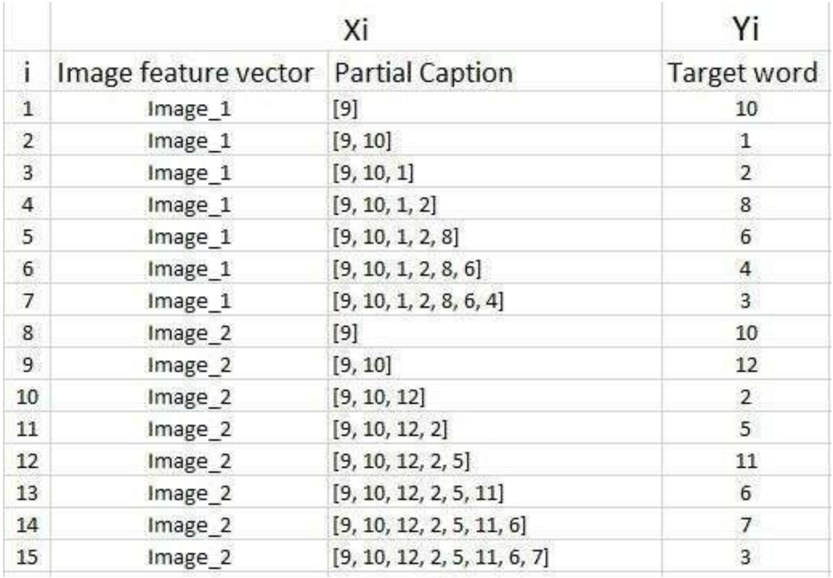
We must now understand that in every data point, it's not just the image which goes as input to the system, but also, a partial caption which helps to **predict the next word in the sequence.**

Since we are processing **sequences**, we will employ a **Recurrent Neural Network** to read these partial captions (more on this later).

However, we have already discussed that we are not going to pass the actual English text of the caption, rather we are going to pass the sequence of indices where each index represents a unique word.

Since we have already created an index for each word, let's now replace the words with their indices and understand how the data matrix will look like:

Data matrix after replacing the words by their indices



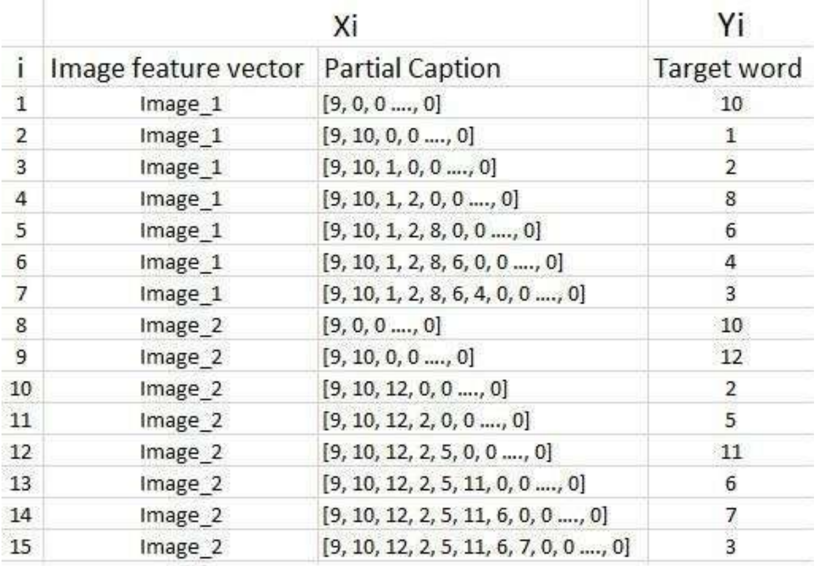
	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	[9]	10
2	Image_1	[9, 10]	1
3	Image_1	[9, 10, 1]	2
4	Image_1	[9, 10, 1, 2]	8
5	Image_1	[9, 10, 1, 2, 8]	6
6	Image_1	[9, 10, 1, 2, 8, 6]	4
7	Image_1	[9, 10, 1, 2, 8, 6, 4]	3
8	Image_2	[9]	10
9	Image_2	[9, 10]	12
10	Image_2	[9, 10, 12]	2
11	Image_2	[9, 10, 12, 2]	5
12	Image_2	[9, 10, 12, 2, 5]	11
13	Image_2	[9, 10, 12, 2, 5, 11]	6
14	Image_2	[9, 10, 12, 2, 5, 11, 6]	7
15	Image_2	[9, 10, 12, 2, 5, 11, 6, 7]	3

Table 2.3

Since we would be doing **batch processing** (explained later), we need to make sure that each sequence is of **equal length**. Hence we need to **append 0's** (zero padding) at the end of each sequence. But **how many** zeros should we append in each sequence?

Well, this is the reason we had calculated the maximum length of a caption, which is 34 (if you remember). So we will append those many number of zeros which will lead to every sequence having a length of 34.

The data matrix will then look as follows:



	Xi		Yi
i	Image feature vector	Partial Caption	Target word
1	Image_1	[9, 0, 0 …., 0]	10
2	Image_1	[9, 10, 0, 0 …., 0]	1
3	Image_1	[9, 10, 1, 0, 0 …., 0]	2
4	Image_1	[9, 10, 1, 2, 0, 0 …., 0]	8
5	Image_1	[9, 10, 1, 2, 8, 0, 0 …., 0]	6
6	Image_1	[9, 10, 1, 2, 8, 6, 0, 0 …., 0]	4
7	Image_1	[9, 10, 1, 2, 8, 6, 4, 0, 0 …., 0]	3
8	Image_2	[9, 0, 0 …., 0]	10
9	Image_2	[9, 10, 0, 0 …., 0]	12
10	Image_2	[9, 10, 12, 0, 0 …., 0]	2
11	Image_2	[9, 10, 12, 2, 0, 0 …., 0]	5
12	Image_2	[9, 10, 12, 2, 5, 0, 0 …., 0]	11
13	Image_2	[9, 10, 12, 2, 5, 11, 0, 0 …., 0]	6
14	Image_2	[9, 10, 12, 2, 5, 11, 6, 0, 0 …., 0]	7
15	Image_2	[9, 10, 12, 2, 5, 11, 6, 7, 0, 0 …., 0]	3

Table 2.4

Word Embedding

As already stated above, we will map the every word (index) to a 200-long vector and for this purpose, we will use a pre-trained GLOVE Model:

Now, for all the 1652 unique words in our vocabulary, we create an embedding matrix which will be loaded into the model before training.

TECHNOLOGY USED

Neural Network

The inventor of the first neurocomputer, Dr. Robert Hecht-Nielsen, defines a neural network as –"...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs."The idea of ANNs is based on the belief that working of human brain by making the right connections, can be imitated using silicon and wires as living **neurons** and**dendrites[1]**.

The human brain is composed of 86 billion nerve cells called **neurons.** They are connected to other thousand cells by **Axons.** Stimuli from external environment or inputs from sensory organs are accepted by dendrites[8]. These inputs create electric impulses, which quickly travel through the neural network. A neuron can then send the message to other neuron to handle the issue or does not send it forward.

CNN

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other[1]. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn thesefilters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex[11]. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visualarea.

LSTM

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

Convolution Layer

Convolutional layers in a convolutional neural network systematically apply learned filters to input images in order to create feature maps that summarize the presence of those features in the input.Convolutional layers prove very effective, and stacking convolutional layers in deep models allows layers close to the input to learn low-level features (e.g. lines) and layers deeper in the model to learn high-order or more abstract features, like shapes or specific objects[8].A limitation of the feature map output of convolutional layers is that they record the precise position of features in the input. This means that small movements in the position of the feature in the input image will result in a different feature map. This can happen with re-cropping, rotation, shifting, and other minor changes to the input image[22].A common approach to addressing this problem from signal processing is called down sampling. This is where a lower resolution version of an input signal is created that still contains the large or important structural elements, without the fine detail that may not be as useful to the task.

Pooling Layer

A pooling layer is a new layer added after the convolutional layer. Specifically, after a nonlinearity (e.g. ReLU) has been applied to the feature maps output by a convolutional layer; for example the layers in a model may look asfollows:

1. InputImage
2. ConvolutionalLayer
3. Nonlinearity
4. PoolingLayer

The addition of a pooling layer after the convolutional layer is a common pattern used for ordering layers within a convolutional neural network that may be repeated one or more times in a givenmodel[17].
The pooling layer operates upon each feature map separately to create a new set of the same number of pooled feature maps.

ResNet-50 Model

ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. This model was the winner of ImageNet challenge in 2015. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers successfully. Prior to ResNet training very deep neural networks was difficult due to the problem of vanishing gradients[7].

AlexNet, the winner of ImageNet 2012 and the model that apparently kick started the focus on deep learning had only 8 convolutional layers, the VGG network had 19 and Inception or GoogleNet had 22 layers and ResNet 152 had 152 layers. In this blog we will code a ResNet-50 that is a smaller version of ResNet 152 and frequently used as a starting point for transferlearning.

REFERENCES

[1]. Krizhevsky, Alex, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks." International Conference on Neural Information Processing Systems Curran Associates Inc. 1097-1105.(2012)

[2]. Girshick, Ross, et al. "Region-based Convolutional Networks for Accurate Object Detection and Segmentation." IEEE Transactions on Pattern Analysis & Machine Intelligence 38.1:142-158.(2015)

[3]. Devlin, Jacob, et al. "Language Models for Image Captioning: The Quirks and What Works." Computer Science(2015)

[4]. Fang, H., et al. "From captions to visual concepts and back." Computer Vision and Pattern Recognition IEEE, 1473 -1482. (2015)

[5]. Cho, Kyunghyun, et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical MachineTranslation."Computer Science (2014)

[6]. Hochreiter, Sepp, and J. Schmidhuber. "Long Short- TermMemory."Neural Computation 9.8: 1735-1780.(1997)

[7]. Karpathy, Andrej, and F. F. Li. "Deep visual-semantic alignments for generating image descriptions." Computer Vision and Pattern Recognition IEEE, 3128-3137.(2015)

[8]. Sermanet, Pierre, et al. "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks." Eprint Arxiv (2013)

[9]. Sundermeyer, M., et al. "Comparison of feedforward and recurrent neural network language models." IEEE International Conference on Acoustics, Speech and Signal Processing IEEE, 8430-8434.(2013)

[10]. Simonyan, Karen, and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." Computer Science (2014)

[11]. Szegedy, Christian, et al. "Going deeper with convolutions." IEEE Conference on Computer Vision and Pattern Recognition IEEE, 1-9.(2015)

[12]. He, Kaiming, et al. "Deep Residual Learning for Image Recognition." IEEE Conference on Computer Vision and Pattern Recognition IEEE Computer Society, 770-778.(2016)

[13]. Mao, Junhua, et al. "Explain Images with Multimodal Recurrent Neural Networks." Computer Science(2014)

[14]. Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." IEEE Conference on Computer Vision and Pattern Recognition IEEE Computer Society, 3156-3164.(2015)

[15]. Xu,Kelvin,etal."Show,AttendandTell:NeuralImageCaptionGenerationwithVisualAttention."Computer Science,2048-2057. (2015)

[16]. Papineni, K. "BLEU: a method for automatic evaluation of MT."(2001)

[17]. Satanjeev, Banerjee. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with HumanJudgments.

ACL-2005.228-231. (2005)

[18]. Flick, Carlos. "ROUGE: A Package for Automatic Evaluation of summaries." The Workshop on Text Summarization Branches Out2004:10.(2014)

[19]. Vedantam, Ramakrishna, C. L. Zitnick, and D. Parikh. "CIDEr: Consensus-based Image Description Evaluation." Computer Science ,4566-4575.(2014)

[20]. Anderson,Peter,etal."SPICE:SemanticPropositionalImageCaptionEvaluation."AdaptiveBehavior11.4382-398.(2016)

[21]. Ranzato,Marc'Aurelio, etal. "SequenceLevel TrainingwithRecurrentNeuralNetworks."ComputerScience(2015)

[22]. Kalchbrenner, Nal, E. Grefenstette, and P. Blunsom. "A Convolutional Neural Network for Modelling Sentences." Eprint Arxiv (2014)

[23]. Aneja, Jyoti, A. Deshpande, and A. Schwing. "Convolutional Image Captioning."(2017)

[24]. Gu, Jiuxiang, et al. "Stack-Captioning: Coarse-to-Fine Learning for Image Captioning."(2018)