OPEN SOURCE TECHNOLOGIES

PROJECT REPORT

ON

IMAGE CAPTIONING BOT 

YATHARTH BABBAR (2017UCO1578)

ABHISHEK SINGH (2017UCO1582)

NAVJOT SINGH (2017UCO1600)

**SYNOPSIS**

**TITLE**

Image Captioning Bot using Deep Learning concepts and python libraries.

**PROBLEM STATEMENT**

Image captioning requires to recognize the important objects, their attributes and their relationships in an image. It also needs to generate syntactically and semantically correct sentences. So using Machine Learning and Deep Learning concepts, enable the machine to learn and generate captions or descriptions for given images. Deep learning based techniques should be used because these techniques are capable of handling the complexities and challenges of image captioning. Machine should be able to detect and recognize different objects in image. It also needs to understand scene type or location, object properties and their interactions. Generating well-formed sentences requires both syntactic and semantic understanding of the language and using all these details our machine should be able to describe the image for our user and further use this concept of describing images to solve real world problems.

**AIM/OBJECTIVE**

Our aim is to make a machine self sufficient to handle complexities and be able to detect object properties, their interactions, understand location or scene type and then caption an image using Deep Learning concepts such as Multi-layered Perceptrons, Convolution Neural Networks, Text Processing, Natural Language Processing, required datasets of images and then validate our model using popular evaluation metrics used in deep learning based model.

**INTRODUCTION**

**ABSTRACT**

Every day, we come across and see a large number of images from various sources such as the books, social media, internet, news, diagrams and advertisements etc. Some of these sources contain images that viewers would have to interpret themselves. Most images we view do not have a description, but we can largely understand them without their detailed captions. However, machine needs to interpret some form of image captions if we as humans need automatic image captions from it.

Therefore Image Captioning is an interesting artificial intelligence problem where a descriptive sentence is generated for a given image and is important for many reasons. Image captioning has its own convolution and challenges. It requires to recognize the important objects, their properties and their relationships in an image. It needs to understand how objects are related to each other inside image. It also needs to generate syntactically and semantically correct sentences.

It involves the dual techniques from computer vision to understand the content of the image and a language model from the field of natural language processing to turn the understanding of the image into words in the right order.

**APPLICATIONS AND ADVANTAGES**

Image Captioning has numerous applications in the field of medical, biomedicine industry, commerce, military, education, digital libraries, web searching. Social media platforms such as Facebook, Instagram and Twitter can directly generate descriptions from user images and collect lots of data about user. The descriptions of the image which user post can include where we are (e.g., office, beach, cafe), what we wear and importantly what we are doing there. Some of these applications are discussed below

• **Content-Based Image Retrieval (CBIR)**

Image indexing is very important for Content-Based Image Retrieval and one of its main application also. It is a technology which allows to organize images based on their visual appearance. They are based on the application of computer vision techniques to the image retrieval problem in large databases.

Content-Based Image Retrieval (CBIR) consists of retrieving the most visually similar images to a given query image from a database of images. It is also known as Query By Image Content (QBIC).

• **Self driving cars**

Automatic driving is one of the biggest challenges and many firms, organisations such as Tesla, BMW have invested heavily in this sector and various researches are going on globally for self-driving cars. Image captioning can help describe the automatic vehicle the scene around itself, location, objects nearby. Therefore if we can properly caption the scene around the car, it can give a boost to the self driving system.

• **Visual aid devices for blind people**

A product can be created for the blind people which will guide them travelling on the roads without the support of anyone else. This device can provide aid to blind people and make them independent. Image captioning can do this by first converting the scene around into text and then the text to voice. Both are now famous applications of Deep Learning and many companies are working on this. Ex – HORUS , eye-wear device build by NVIDIA has proven to be real life-changer for blind people and has optimistic response from people. It can be used to read a book, can recognize a friend, help navigating streets and tells about obstacles.

• **Image search**

Image captioning can be used to first convert image to text and then searching those keywords. It can make image search as efficient as web search which uses keywords. Ex- Google image search.

• **Automatic surveillance – CCTV cameras**

Nowadays CCTV cameras are installed everywhere and if image captioning is used on the images captured by these surveillance cameras, description can be generated, that what is happening around the area. Malicious and other hostile activities could be stopped, though the system has to be extreme efficient and accurate.

**RELATED WORK**

Some of the major project work that has been successfully executed with the help of image captioning and deep learning are :-

• **Google Photos** : It uses Content-Based Image Retrieval to store organise user photos based on visual appearance and objects present in images. User photos are organised and searchable by the places and things in them – no tagging is required. Ex- Just search "dog" to find all the photos of your pet dog and search “mountain” to found all your mountain vacation pictures.

• **SkinVision** : Uses image captioning and deep learning concepts to analyze user photo and give the user an instant risk assessment and advice on what user should do next, so that he/she can see a doctor in time, if needed. It makes it possible to detect skin cancer at an early stage when it’s most treatable and has less expensive treatment options**.** The latest research proves that it can detect 95% of skin cancer. The sensitivity of general practitioners ranges from 61% and 66%, while the sensitivity of dermatologists is between 75% and 92%.

• **DeepMind** : It is an organisation which researches and builds safe AI systems that learn how to solve problems and advance scientific discovery for all. Similarly they build an AI system

which can play video games in a fashion similar to that of humans.

• **Descartes Labs** : relies on 4 petabytes of satellite imaging data and a machine learning algorithm to figure out how healthy the corn crop is from space. It uses images from satellite to predict the crop yield in United States of America.

• **Fed Ex and other courier services** : are using hand written digit recognition system from many times now to detect pin code correctly.

• **Tesla/Google Self Drive Cars** : All the self drive cars are using image/video processing with neural network to attain their goal.

Image captioning can help describe the automatic vehicle the scene around itself, location, objects nearby. Therefore if we can properly caption the scene around the car, it can give a boost to the self driving system.

**PREREQUISITES**

• Neural Networks ( Multilayer Perceptron, Convolutional Neural Networks, Recurrent Neural Networks)

A **Neural Network** is a network or circuit of neurons. An artificial neural network is composed of artificial neurons or nodes. Thus, a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred to as a linear combination. Finally, an activation function controls the amplitude of the output.

A **Multilayer Perceptron (MLP)** is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to refer to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation). Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.

An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

A **Convolutional Neural Network (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. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNNs have the ability to learn these filters/characteristics.

A **Recurrent Neural Network (RNN)** is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behaviour. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

The term “recurrent neural network” is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other

is infinite impulse. Both classes of networks exhibit temporal dynamic behavior. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feedforward neural network, while an infinite impulse recurrent network is a directed cyclic graph that can not be unrolled.

Both finite impulse and infinite impulse recurrent networks can have additional stored states, and the storage can be under direct control by the neural network. The storage can also be replaced by another network or graph, if that incorporates time delays or has feedback loops. Such controlled states are referred to as gated state or gated memory, and are part of long short-term memory networks (LSTMs) and gated recurrent units. This is also called Feedback Neural Network.

• **Language Model ( Natural Language Processing )**

Natural language processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. Natural language processing are used in speech recognition, natural language understanding, and natural language generation.

• **Word Embeddings**

A word embedding is a learned representation for text where words that have the same meaning have a similar representation. It is this approach to representing words and documents that may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems. Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values are learned in a way that resembles a neural network, and hence the technique is often lumped into the field of deep learning.

• **Transfer Learning**

Transfer learning (TL) is a research problem in machine learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. It helps in reusing or transferring information from previously learned tasks for the learning of new tasks, and hence it has the potential to significantly improve the sample efficiency of a reinforcement learning agent.

• Python syntax and data structures

• Python Libraries such as keras, tensorflow, numpy, nltk, pandas, etc

**Keras** is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System).

**TensorFlow** is free and opensource software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.

**NumPy** is a Python library which adds support for large, multi dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. NumPy was created by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is an open-source software and has many contributors.

**Natural Language Toolkit**(**NLTK)** is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit, plus a cookbook. NLTK is intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems. There are 32 universities in the US and 25 countries using NLTK in their courses. NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

**Pandas** is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

**PROPOSED WORK**

**Dataset Used**

Several images datasets are available such as Pascal VOC, Flickr8K, Flickr30K and MSCOCO. Flickr 8K Image Captioning dataset has been used in the proposed work. Flickr 8K dataset is provided by University of Illinois. This dataset contains 8000 images with 5 captions for each image and has memory size of 2.21GB.

This dataset has been splitted into three disjoint sets. Training dataset contains 6000 images whereas Development and Test dataset contains 1000 images each.

**Data Cleaning**

We performed some basic cleaning on text(captions) like lower casting all words, removing special tokens and numbers. A vocabulary of all unique words present across the 40000 captions is created. The vocabulary is filtered to contain words that occur at least 10 times. Reducing the vocabulary size results in less overfitting and less computation.

Python code for data cleaning :

**def** clean\_text(sentence):

sentence = sentence.lower()

sentence = re.sub("[^a-z]+"," ",sentence) *#Substitute anything not a char by space*

sentence = sentence.split()

sentence = [s **for** s **in** sentence **if** len(s)>1] *#Deleting sentences of length 1*

sentence = " ".join(sentence)

**return** sentence

**for** key,caption\_list **in** descriptions.items():

**for** i **in** range(len(caption\_list)):

caption\_list[i] = clean\_text(caption\_list[i]) descriptions["1000268201\_693b08cb0e"]

**OUTPUT**

['child in pink dress is climbing up set of stairs in an entry way',

'girl going into wooden building',

'little girl climbing into wooden playhouse',

'little girl climbing the stairs to her playhouse', 'little girl in pink dress going into wooden cabin']

**Data Preprocessing**

In the proposed work, every image is converted into a fixed sized vector which can fed as input to the neural network. For this purpose, we opt for transfer learning by using ResNet-50 model (pre-trained model). This model was trained on Imagenet dataset to perform image classification. We removed the last

softmax layer from the model and extracted the feature vector for every image.

Every unique word in the vocabulary is represented by an integer between 1 and 1845. Two dictionaries namely “idx\_to\_word” ( returns the word at a particular index) and “word\_to\_idx” (returns the index of a particular word) have been created.

Python code for data preprocessing :

word\_to\_idx = {}

idx\_to\_word = {}

**for** i,word **in** enumerate(total\_words):

word\_to\_idx[word] = i+1

idx\_to\_word[i+1] = word

idx\_to\_word[1846] = 'startseq'

word\_to\_idx['startseq'] = 1846

idx\_to\_word[1847] = 'endseq'

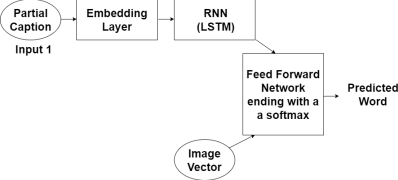
word\_to\_idx['endseq'] = 1847

vocab\_size = len(word\_to\_idx) + 1

**Basic Idea**

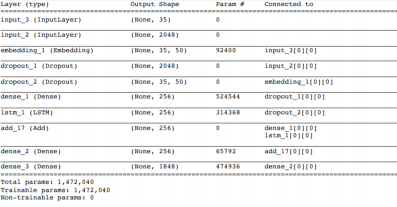
We give an image as input to the model and expect a caption (or sentence) as output. But, the model which we have trained cannot generate entire sentence at once. We also need to provide a partial caption (read using Recurrent Neural Networks) as input to the model along with the image. A single word in the vocabulary is given as output which is appended to the partial caption and fed to the model again. Like this, we generate the entire sentence or caption which describes the input image.

**FRAMEWORK/MODEL**

• **High Level Architecture of the model **

LSTM (Long Short Term Memory) is a specialised Recurrent Neural Network used to process the partial captions. The weights of the model will be updated using back propagation algorithm and the model will learn to output a word, given an image feature vector and a partial caption.

• **Model Summary**

****

**PREDICTIONS**

The image captioning model was implemented and we were able to generate some captions. Since no model in the world is perfect, our model also makes mistakes like colors getting mixed with background and incorrect grammar. To get good results, images used for testing must be semantically related to those used for training the model. Some of the captions generated by the model are shown below :





**Some Important Functions Used**

model = ResNet50(weights="imagenet",input\_shape=(224,224,3)) model.summary()

model\_new = Model(model.input,model.layers[-2].output)

**def** preprocess\_img(img):

img = image.load\_img(img,target\_size=(224,224)) img = image.img\_to\_array(img)

img = np.expand\_dims(img,axis=0)

*# Normalisation*

img = preprocess\_input(img)

**return** img

**def** encode\_image(img):

img = preprocess\_img(img)

feature\_vector = model\_new.predict(img)

feature\_vector = feature\_vector.reshape((-1,)) *#print(feature\_vector.shape)*

**return** feature\_vector

**def**

data\_generator(train\_descriptions,encoding\_train,word\_to\_idx,m ax\_len,batch\_size):

X1,X2, y = [],[],[]

n =0

**while True**:

**for** key,desc\_list **in** train\_descriptions.items(): n += 1

photo = encoding\_train[key]

**for** desc **in** desc\_list:

seq = [word\_to\_idx[word] **for** word **in** desc.split() **if** word **in** word\_to\_idx]

**for** i **in** range(1,len(seq)):

xi = seq[0:i]

yi = seq[i]

*#0 denote padding word*

xi =

pad\_sequences([xi],maxlen=max\_len,value=0,padding='post')[0]

yi =

to\_categorcial([yi],num\_classes=vocab\_size)[0]

X1.append(photo)

X2.append(xi)

y.append(yi)

**if** n==batch\_size:

**yield**

[[np.array(X1),np.array(X2)],np.array(y)]

X1,X2,y = [],[],[]

n = 0

**def** predict\_caption(photo):

in\_text = "startseq"

**for** i **in** range(max\_len):

sequence = [word\_to\_idx[w] **for** w **in** in\_text.split() **if** w **in** word\_to\_idx]

sequence =

pad\_sequences([sequence],maxlen=max\_len,padding='post')

ypred = model.predict([photo,sequence]) ypred = ypred.argmax() *#WOrd with max prob always - Greedy Sampling*

word = idx\_to\_word[ypred]

in\_text += (' ' + word)

**if** word == "endseq":

**break**

final\_caption = in\_text.split()[1:-1]

final\_caption = ' '.join(final\_caption)

**return** final\_caption

**FUTURE SCOPE**

The proposed work is just a first-cut solution and a lot of modifications can be made to improve the solution like :

• Using a larger dataset such as Flickr 30K dataset which has 30000 images , MS COCO datasets as datasets differ in types of images, number of images used and number of captions used to describe each image. Therefore different dataset can generate different results thus improvisation can be done.

• Doing more hyper parameter tuning.

• Using various evaluation metrics for deep-learning such as BLEU(Bilingual evaluation understudy) or ROUGE(Recall Oriented Understudy for Gisting Evaluation) can be used to evaluate and measure the performance of the model trained.

• Generation based methods can generate novel captions for every image. However, these methods fail to detect prominent objects and properties and their relationships to some extent in generating accurate and multiple captions. In addition to this, the accuracy of the generated captions largely depends on syntactically correct and diverse captions which in turn rely on powerful and sophisticated language generation model.

• Employing ensembles to achieve better performance. • Changing the model architecture.

• Working on open domain dataset will also be an interesting avenue for research in this area.

• External knowledge can be added in order to generate attractive image captions. Supervised learning needs a large amount of labelled data for training. Therefore, unsupervised learning and reinforcement learning will be more popular in future in image captioning.

**CONCLUSION**

In the proposed project we have learned about that how machine learning and deep learning-based concepts can be used to enable the machine to learn and be self-sufficient to understand the objects in image, how they are inter-related with each other, understand the scene or mood of the image and be able to generate caption for the image and describe it. We learned about the architecture used in the process of making the machine learn about image captioning. We learned about different neural networks and how they function such as

Multilayer Perceptron, Convolutional Neural Networks and Recurrent Neural Networks have been used in this project. We also learned about the different datasets and different evaluation metrics. We see how image caption has numerous applications and how some of the biggest companies around the world are using this technology to build systems such as HORDUS, visual aid device, Google Image search, self-driving cars and SkinVision etc. Although we saw that how deep learning-based image captioning methods have achieved a remarkable progress in recent years, but a robust image captioning method that is able to generate high quality captions for nearly all images is yet to be achieved. With the advent of novel deep learning network architectures, automatic image captioning will remain an active research area in future.

**REFERENCES**

• https://cs.stanford.edu/people/karpathy/cvpr2015.pdf • https://arxiv.org/abs/1411.4555

• https://machinelearningmastery.com/develop-a deeplearning-caption-generation-model-in-python/

• https://arxiv.org/pdf/1810.04020.pdf

• https://towardsdatascience.com/image-captioning-with keras-teaching-computers-to-describe-pictures c88a46a311b8

• https://towardsdatascience.com/how-to-easily deploymachine-learning-models-using-flask-b95af8fe34d4

**GITHUB LINK**

https://github.com/abhishek99singh/Image\_Caption\_Bot