KONKAN GYANPEETH COLLEGE OF ENGINEERING, KARJAT

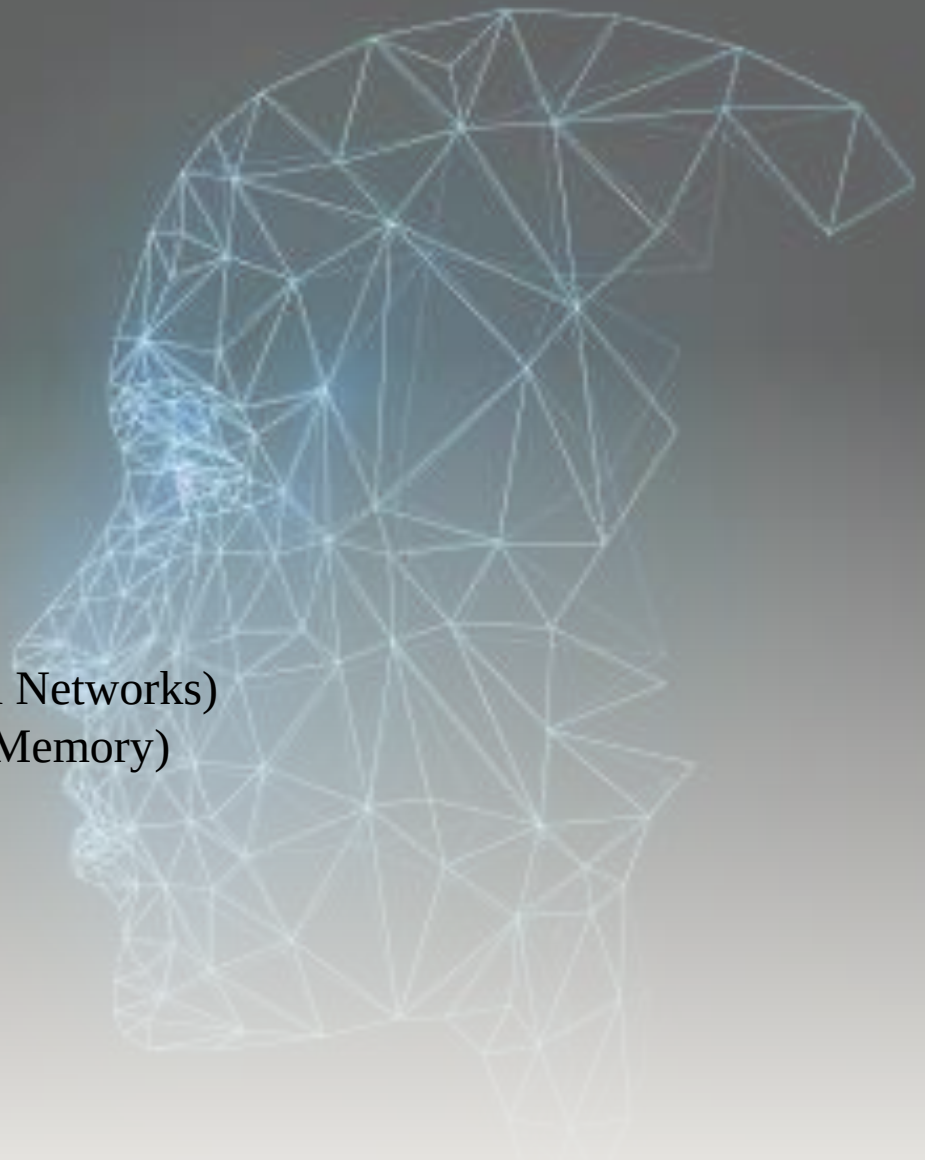
Affiliated to University of Mumbai, Approved by AICTE, New Delhi.

Automatic Image Captioning Using Deep Learning.

- Project Members:
  - Dhruv Solanki
  - Tejal Supe
  - Siddhi Undirkar

Under the Guidance of:
Prof. J.P.Patil

CONTENTS:



Abstract

- Image captioning means automatically generating a caption for an image. Automatically creating the description of an image using any natural language sentences is a very challenging task.
- It requires expertise of both image processing as well as natural language processing. Sharing photos through the internet (e.g. Instagram, Facebook, etc.), which becomes a common practice, leads to archives in the order of millions of images.
- Manual annotation is time consuming and extremely labor-intensive work. Thereafter, computer-assisted system are desired to lesson these difficulties by automation using machine learning.
- The annoted image can be used for retrieval purposes, in image processing algorithms, intelligent scanner, digital cameras, photocopiers, and printers. Our goal is to create system which will annotate images based on previously learned datasets.

Introduction

- Image annotation refers to the tagging of images with appropriate keywords. The process of annotating images manually is time-consuming and extremely labor-intensive work. Thereafter, computer-assisted systems are desired to lessen these difficulties by automation.
- The automatic image annotation is targeted to assign a few relevant text keywords to an input image that reflects its visual content. Machine learning techniques are used to develop the image annotation systems to map the low-level (visual) features to high-level concepts or semantics.
- We are going to use two neural network algorithm which is Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) based Long Short Term Memory (LSTM). In the project CNN will be used to train the images as well as to detect the objects in the image with the help of various pre trained models and RNN based LSTM will be used to generate captions from the generated object keyword

Literature survey

Sr. No.	Paper Title	Author's Name	Techniques Used
1	Image Captioning Using CNN and LSTM	Ali Ashraf Mohamad	Technique - CNN , LSTM Dataset – Flickr8k Accuracy – BLEU-0.46
2	Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering	Peter Anderson, Damien Teney, Mark Johnson , Stephen Gould, Lei Zhang, Xiaodong He, Chris Buehler	Technique - Faster R-CNN Dataset – MSCOCO Evaluation Matrix – BLEU-4 – 36.9 CIDEr – 117.9 SPICE – 21.5
3	Transfer Learning Using VGG16 with Deep CNN for classifying Image	Srikanth Tammina	Model : VGG16

Sr. No	Paper Title	Author's	Techniques used
4	Deep Learning based Automatic Image Caption Generation	Varsha Kesavan, Vaidehi Muley , Megha Kolhekar	Techniques – CNN(Encoder), RNN(Decoder) Dataset – MSCOCO Pre Trained Model - VGG16 model(Encoder) GRU network(Decoder)
5	Guiding the Long-Short Term Memory model for Image Caption Generation	Xu Jia, Efstratios Gavves, Basura Fernando, Tinne Tuytelaars.	Techniques- LSTM , gLSTM Dataset - *Flickr8K, Flickr30K and MS COCO* *Evaluation Matrix – BLEU, METEOR*
6	Image Captioning with Semantic Attention	Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, Jiebo Luo	Techniques – RNN Dataset – MSCOCO , *Flickr30K* *Evaluation Matrix – BLEU, METEOR,* ROUGE-L, CIDEr

Objectives

- The objective of our project is to extract the features of a given image and to automatically caption the objects present in the image.
- To build a system that will identify multiple objects from a given image.
- To create a system that will be able to caption the objects present in the image.
- To be able to train the model created to identify and caption the images.
- To be able to test and evaluate the performance of the system of both the training and testing phase.

Architecture:

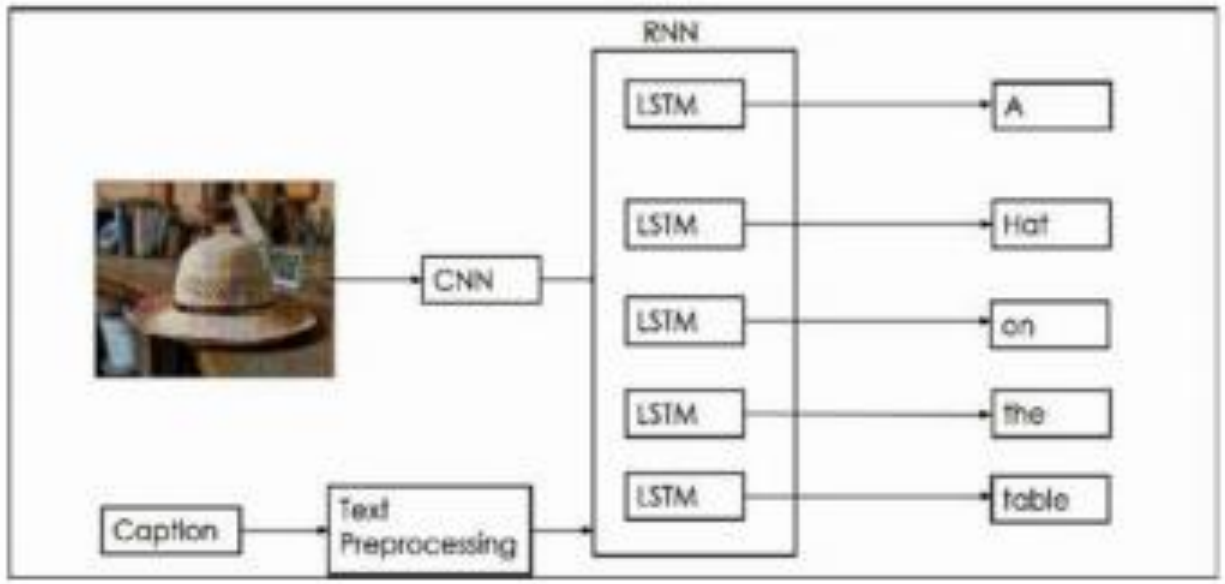
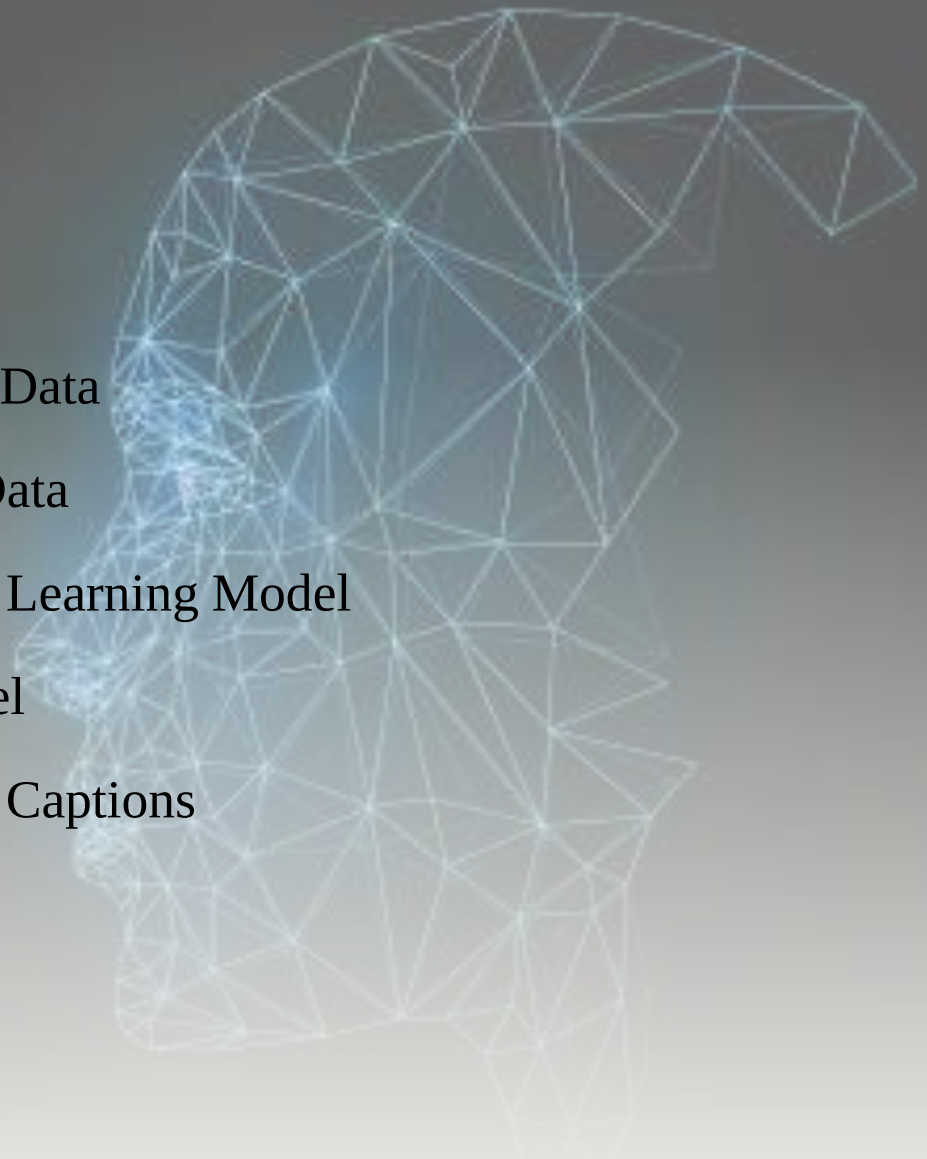


Figure 1. Architecture

Phases:

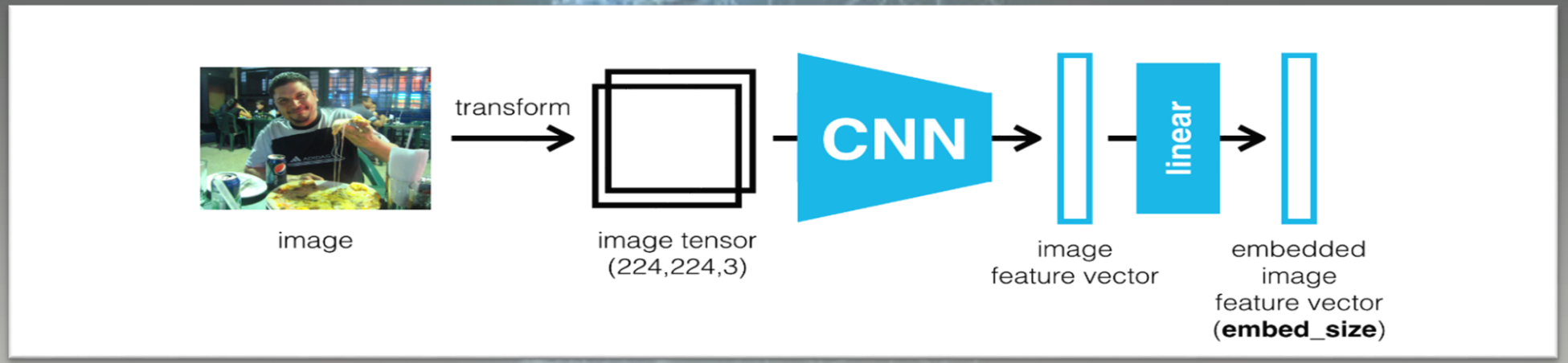


- Phase 1: Prepare Photo Data
- Phase 2: Prepare Text Data
- Phase 3: Develop Deep Learning Model
- Phase 4: Evaluate Model
- Phase 5: Generate New Captions

CNN(Convolution Neural Networks):

- The convolutional neural network, or CNN for short, is a specialized type of neural network model designed for working with two-dimensional image data, although they can be used with one-dimensional and three-dimensional data.
- Convolutional neural networks do not learn a single filter; they, in fact, learn multiple features in parallel for a given input.
- CNN is often called as encoder because it encodes the content of the image into a smaller feature vector.

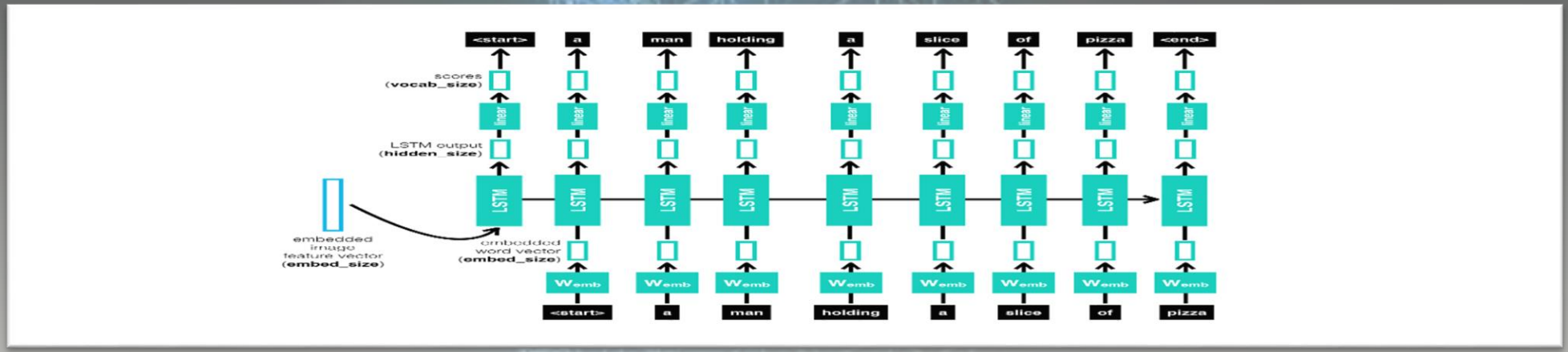
CNN Model:


transform
CNN
linear
image
image tensor
(224,224,3)
image
feature vector
embedded
image
feature vector
(embed_size)

LSTM(Long Short Term Memory):

- Recurrent neural networks (RNN) are the state of the art algorithm that remembers its input, due to an internal memory, which makes it perfectly suited for machine learning problems that involve sequential data.
- RNN remembers things for just small durations of time, i.e. if we need the information after a small time it may be reproducible, but once a lot of words are fed in, this information gets lost somewhere.
- This issue can be resolved by applying a slightly tweaked version of RNNs – the Long Short-Term Memory Networks. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.
- Lstm is called as decode because it decodes the vector and try to turn into a caption.

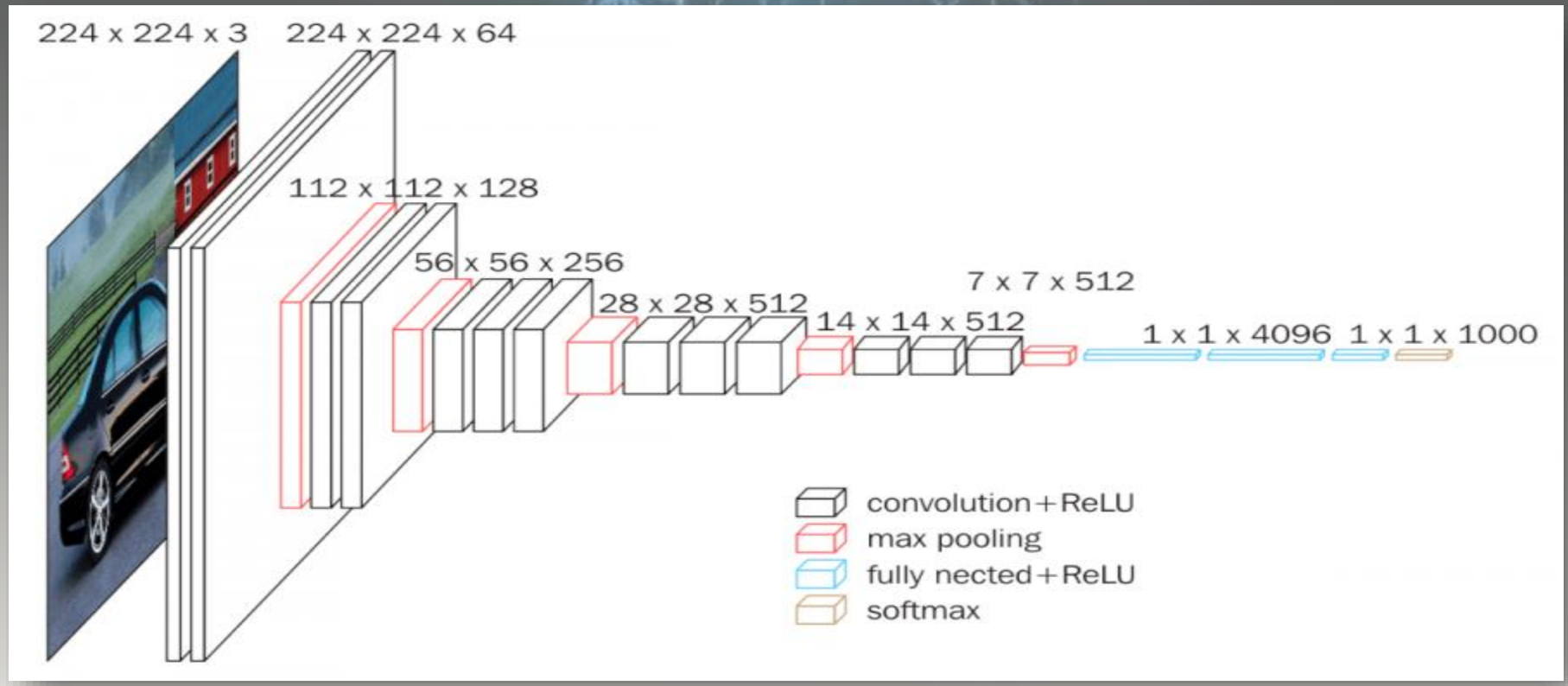
LSTM Model:


<start>
a
man
holding
a
slice
of
pizza
<end>
scores
(vocab_size)
Linear
LSTM output
(hidden_size)
LSTM
embedded
image
feature vector
(embed_size)
embedded
word vector
(embed_size)
Wemb
<start>
a
man
holding
a
slice
of
pizza

VGG16

- There are various pre-trained Convolutional Neural Network Trained model such as VGG19, VGG16, Inceptionv3, ResNet, Mobile Net etc. but in our project we are going to use VGG16 pre-trained CNN model.VGG16 has 13 convolutional layers, some of which are followed by maximum pooling layers and then 4 fully connected layers and finally a 1000-way softmax classifier.
- At last there are 3 fully connected layer followed by last softmaxlayer.In our project we are going to cut-off the last layer because vgg16 is use to classify the image but we don’t want to classify the image but just get fixed-length information vector of image.

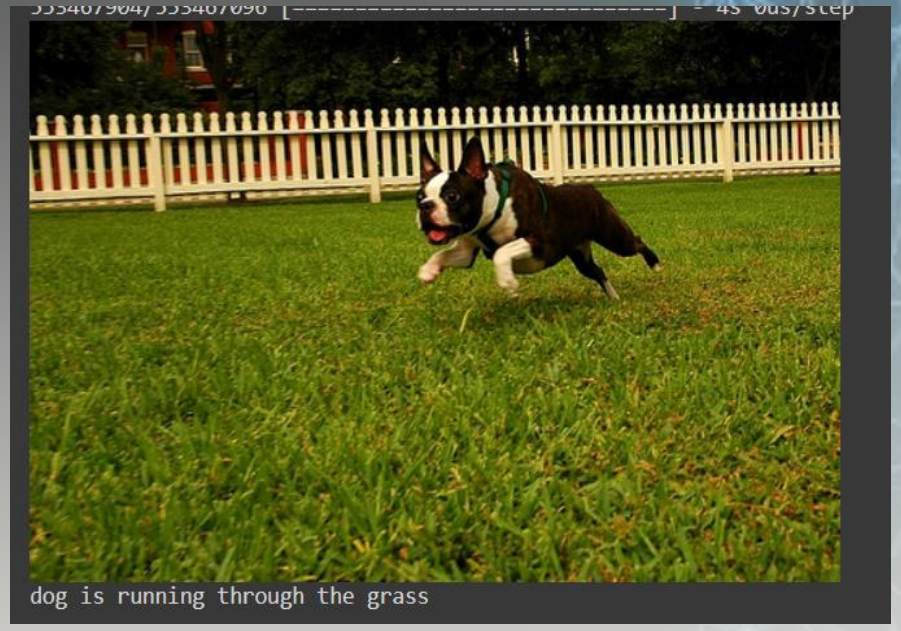
VGG16 Model


224 x 224 x 3
224 x 224 x 64
112 x 112 x 128
56 x 56 x 256
28 x 28 x 512
14 x 14 x 512
7 x 7 x 512
1 x 1 x 4096
1 x 1 x 1000
convolution + ReLU
max pooling
fully nected + ReLU
softmax

Scope

- The scope of this project is to classify the images from the different classes available which makes it easy to search an image based on a keyword.
- The system developed in this project is such that it will add a caption to multiple objects present in an image.
- The images after been identified and captioned will be saved into the separate database.

Output


dog is running through the grass


man in red shirt is sitting on bench

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 34)]         0
__________________________________________________________________________________________________
input_1 (InputLayer)            [(None, 4096)]       0
__________________________________________________________________________________________________
embedding (Embedding)           (None, 34, 256)      1940224     input_2[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 4096)         0           input_1[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 34, 256)      0           embedding[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 256)          1048832     dropout[0][0]
__________________________________________________________________________________________________
lstm (LSTM)                     (None, 256)          525312      dropout_1[0][0]
__________________________________________________________________________________________________
add (Add)                       (None, 256)          0           dense[0][0]
                                                                 lstm[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 256)          65792       add[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 7579)         1947803     dense_1[0][0]
==================================================================================================
Total params: 5,527,963
Trainable params: 5,527,963
Non-trainable params: 0
```

Model

Evaluation Metrics

```python
#the below function evaluates the skill of the model
def evaluate_model(model, descriptions, photos, tokenizer, max_length):
    actual, predicted = list(), list()
    for key, desc_list in descriptions.items():
        prediction = generate_desc(model, tokenizer, photos[key], max_length)
        actual_desc = [d.split() for d in desc_list]
        actual.append(actual_desc)
        predicted.append(prediction.split())

    print('BLEU-1: ', corpus_bleu(actual, predicted, weights=(1.0, 0, 0, 0)))
    print('BLEU-2: ', corpus_bleu(actual, predicted, weights=(0.5, 0.5, 0, 0)))
    print('BLEU-3: ', corpus_bleu(actual, predicted, weights=(0.3, 0.3, 0.3, 0)))
    print('BLEU-4: ', corpus_bleu(actual, predicted, weights=(0.25, 0.25, 0.25, 0.25)))

def max_length(descriptions):
    lines = to_lines(descriptions)
    return max(len(d.split()) for d in lines)
```

```
Dataset:  6000
Descriptions: train= 6000
Vocabulary Size:  7579
Description Length: , 34
Dataset:  1000
Descriptions: test= 1000
Photos: test= 1000
BLEU-1:  0.5179559306141585
BLEU-2:  0.27731376452861534
BLEU-3:  0.18970508252415758
BLEU-4:  0.08479132901530388
```

Conclusion

- Through this project, we learned about the deep learning techniques used for image captioning problem.
- We learned that the result of generated captions is influenced by the training dataset.
- The Flickr8k dataset contains many outdoor images of humans and dogs and our model gives better results on outdoor images with people and dogs and confuses many different things in other images to dogs and people.
- We have implemented pre-trained CNN and LSTM model and used BLUE evaluation metrics.
- The Bilingual Evaluation Understudy Score, or BLEU for short, is a metric for evaluating a generated sentence to a reference sentence. A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0.
- We have Achieved a Effective BLUE score of 0.5179 for our model.

References

- www.researchgate.net/publication/332109738_A_Survey_in_Deep_Learning_Model_for_Image_Annotation.
- www.researchgate.net/publication/328993377_Building_Detection_and_Segmentation_Using_a_CNN_with_Automatically_Generated_Training_Data
- Claudio Cusano, Gianluigi Ciocca. *Image* nnnoto/ion usinp*SVM*
- Mahdia Bakalem, Nadjia Benblidia. *A Comparative* /rape *Annotation*. 978-1-4799- 4796-6/13/ c 2013 IEEE. (2012).

THANK YOU!