A PROJECT REPORT ON

Generation of Image Captioning using CNN cascaded with RNN approach.

SUBMITTED BY

Pritam Kumar (403024)
Shaunak Joshi (403035)
Shubham Patil (403055)
Shubham Sureka (403078)

UNDER THE GUIDANCE OF

Prof. M.A.Bhalekar

Department Of Computer Engineering
MAEERs MAHARASHTRA INSTITUTE OF TECHNOLOGY
Kothrud, Pune 411 038
2016-2017

MAHARASHTRA ACADEMY OF ENGINEERING AND EDUCATIONAL RESEARCH'S

MAHARASHTRA INSTITUTE OF TECHNOLOGY PUNE

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that

Pritam Kumar (403024)

Shaunak Joshi (403035)

Shubham Patil (403055)

Shubham Sureka (403078)

of B. E. Computer successfully completed Interim Project Report in

Generation of Image Captioning using CNN cascaded with RNN approach.

to my satisfaction and submitted the same during the academic year 2017-2018 towards the partial fulfillment of degree of Bachelor of Engineering in Computer Engineering of Savitribai Phule Pune University under the Department of Computer Engineering , Maharashtra Institute of Technology, Pune.

Prof. M.A.Bhalekar
(Project Guide)

Prof.Dr. V.Y.Kulkarni
(Head of Computer Engineering Department)

ACKNOWLEDGEMENT

It gives us great pleasure and satisfaction in presenting the project on **"Generation of Image Captioning using CNN cascaded with RNN approach"**. We would like to express our deep sense of gratitude towards our project guide Prof. M.A.Bhalekar who contributed her valuable guidance and helped us to solve every problem that arised.

We would also like to thank Prof. (Dr.) V.Y. Kulkarni, HOD (Department of Computer Engineering) for the co-operation and motivation.

Our speacial thanks to the Project Co-ordinators Prof. M. V. Pathak, Prof. S. S. Bobde, Prof. L. B. Bhagwat and Prof. R. A. Patankar for their valuable support. We would also like to express our sincere appreciation to staff of Department of Computer Engineering, Maharashtra Institute of Technology Pune, for their extended help and suggestions at every stage.

Pritam Kumar (403024)
Shaunak Joshi (403035)
Shubham Patil (403055)
Shubham Sureka (403078)

Abstract

We present an Image Caption Generating System that recognizes and labels the image given to it. The user simply has to provide an image to the system and the system classifies and labels the input image successfully. The salient feature of the system is that it can identify objects overlooked by a human observer which makes it useful for wide number of applications. With some exceptions it can also identify objects in a distorted image.

Keywords: Image Captioning,Object Classification,CNN,RNN,LSTM.

Contents

List of Figures

List of Tables

Chapter 1

Problem statement

An Image Captioning system which uses Convolutional Neural Network(CNN) for extracting high-level features from Image and Long Short-Term Memory(LSTM) to generate the captions in natural language describing the input image.

Chapter 2

Problem definition

The main objective of this problem is to caption real world images using a Convolutional Neural Network and Long Short Term Memory. In todays world the amount of pictures generated are huge.To be able to classify these images and search for a particular image is a huge task.This task can be made easy if we can caption these images using Machine Learning.

Thus we aim to make the computer systems intelligent by providing them the capability to identify the objects in the image and define the relation between them. This is achieved by training the CNN on a pre-labelled database which teaches the system to get to know the real world. After learning about the things, LSTM is used to teach the human language of conversation to the system so that it can frame the sentences on its own to describe what it understands as a result of training in the image that is screenshot of the real world.

Chapter 3

Scope of the problem

The scope of this problem ranges from a mobile user to a machine learning scientist. Image captioning is helpful to identify objects overlooked by a human observer. It can be useful in cases where images are extremely cluttered to recognize an object.

This system can be deployed also as a mobile application where user takes an image and the system generates a description of the image. This will be helpful as a smart assistant for children and specially abled people. This system can be deployed for robots for understanding of its surroundings.It can be used even in an application to identify famous objects for a tourist who goes without a tour guide. The user can capture the monument or artifact in his phone camera and the application display what it is in form of text on the phone.

Chapter 4

Literature Survey

We have carried out literature survey to study analysis of different architecture proposed till now for generation of captions for a given image . We have searched research papers in the domain of Machine Learning and neural networks. We have created a summary as follows :

4.1 YOLO 9000:Better,Faster,Stronger[1]

The YOLO model is a simple to construct method that throws away complex pipelines used by state of the art object classifiers and simultaneously calculates the class confidences simultaneously in all the grids in which the image is divided.It outperforms object detectors and classifiers when tested on videos in terms of speed by a great degree.

It is a lot faster in processing objects in a real time video. It predicts a bounding box for each of the object present in that image and the all the bounding boxes and their corresponding class probabilities are predicted by a single CNN model simultaneously. It has even proved better than R-CNN and DPM in cases where it is trained on natural images and tested on classification task for any artwork. The object detection problem is proposed as a regression problem so the need of complex pipelines is removed. It makes less number of incorrect background classification errors compared to Fast R-CNN. Batch normalization proved as a very good substitute for dropout increasing the mean Average Precision by 2%. As compared to state of the art detection systems YOLO makes localization errors mainly because CNNs have a tendency to fail in localization. The cross-entropy function used for training and detection in small as well as large bounding boxes have a great difference of errors depending upon the size of boxes.

4.2 Relaxed Multiple-Instance SVM with Application to Object Discovery[2]

The paper proposes special formulation for MIL and applied it for robust weakly-supervised object discovery. MIL optimization problem is made into a convex problem and solved it efficiently using SGD. Besides of object discovery, RMI-SVM can also be used to solve other recognition tasks, such as visual tracking, image classification, and learning part-based object detection model.

Here, a technique has been proposed to solve the classical MIL problem, named relaxed multiple instance SVM (RMI-SVM). The positiveness of instance is treated as a continuous variable, Noisy-OR model is used to enforce the MIL constraints, and the bag label and instance label is jointly optimized in a unified framework. The optimization problem has been efficiently solved using stochastic gradient descent.The extensive experiments carried out in this paper demonstrate that RMI-SVM consistently achieves superior performance on various benchmarks for MIL.

4.3 Translating Videos to Natural Language Using Deep Recurrent Neural Networks[3]

In this paper a model is proposed for video description which uses neural networks for the entire pipeline from pixels to sentences and can potentially allow for the training and tuning of the entire network. In an extensive experimental evaluation,we showed that the approach generates better sentences than related approaches.

The mean pooling which captures the features of the FC layer of the CNN has helped increasing the performance metric significantly as compared to the state-of-the-art method that gives features from FC layer directly to the LSTM. The model trained on Flickr30k when tested on random frames from the video scored on subjects and verbs with accuracy of 75.16% and 11.65% respectively and 9.01% on objects.

4.4 Long-term Recurrent Convolutional Networks for Visual Recognition and Description[4]

This approach uses frames of the video as inputs to a corresponding CNN with LSTM units corresponding to each which produces a sentence summary based on a strong visual time series model.It outperforms a single CNN-RNN cascade model for an image and even the multi-model neural networks model. It outperforms KNN algorithm in mapping visual features against probable sentences.

The results consistently demonstrate that by learning sequential dynamics with a deep sequence model, we can improve upon previous methods which learn a deep hierarchy of parameters only in the visual domain,and on methods which take a fixed visual representation of the input and only learn the dynamics of the output sequence.

4.5 Show and Tell: A Neural Image Caption Generator[5]

In this paper a generative model is presented based on a deep recurrent architecture that combines Vision Deep CNN and Language Generating RNN to generate natural sentences describing an image. Their model is trained to maximize the likelihood of the target description sentence given the training image. This model is popularly known as Google NIC(Neural Image Caption) Generator.

They propose a neural and probabilistic framework to generate descriptions from images. This model make use of a recurrent neural network which encodes the variable length input into a fixed dimensional vector, and uses this representation to decode it to the desired output sentence. It is a single joint model that takes an image I as input, and is trained to maximize the likelihood p(S|I) of producing a target sequence of words S ={S1,S2,...} where each word St comes from a given dictionary, that describes the image adequately. The model is not tested and ready to handle the unsupervised data, both from images alone and text alone. Hence, it is not know about how to use unsupervised data to improve image description approaches.

4.6 Show, Attend and Tell: Neural Image Caption Generation with Visual Attention[6]

In this paper they introduce an attention based model that automatically learns to describe the content of images. They describe the methods to train this model in a deterministic manner using standard backpropagation techniques and stochastically by maximizing a variational lower bound.The model is able to automatically learn to fix it's gaze on salient objects while generating the corresponding words in the output sequence and this is represented through visualization. By visualizing the attention component learned by the model, they were able to add an extra layer of interpretability to the output of the model.

They introduce two attention-based image caption generators under a common framework. (i) a "soft" deterministic attention mechanism trainable by standard backpropagation methods and (ii) a "hard" stochastic attention mechanism trainable by maximizing an approximate variational lower bound.

4.7 Deep Visual-Semantic Alignments for Generating Image Descriptions[7]

In this paper, a model is presented that generates natural language descriptions of images and their regions. Their alignment model is based on a novel combination of Convolutional Neural Networks(CNNs) over image regions, bidirectional Recurrent Neural Networks(BRNNs) over sentences, and a structured objective that aligns the two modalities through a multimodal embedding. They also describe a Multimodal Recurrent Neural Network architecture that uses the inferred alignments to learn to generate novel descriptions of image regions. They detect objects in every image with a Region Convolutional Neural Network (RCNN). The CNN is pre-trained on ImageNet and fnetuned on the 200 classes. The BRNN takes a sequence of N words and transforms each one into an h-dimensional vector.

The model can only generate a description of one input array of pixels at a fixed resolution. The RNN receives the image information only through additive bias interactions, which are known to be less expressive than more complicated multiplicative interactions. Their approach consists of two separate models. Going directly from an image-sentence dataset to region-level annotations as part of a single model trained end-to-end remains an open problem.

4.8 Caffe: Convolutional Architecture for Fast Feature Embedding[8]

Caffe provides a clean and modifiable framework and have an orderly and extensible toolkit for state-of-the-art deep learning algorithms, with a collection of reference models. The framework is a BSD-licensed C++ library with Python and MATLAB bindings for training and deploying general purpose convolutional neural networks and other deep models efficiently on commodity architectures.

Fast CUDA code and GPU computation achieves processing speeds of more than 40 million images per day on a single K40 or Titan GPU. Caffe provides a complete toolkit for training, testing, finetuning, and deploying models, with well-documented examples for all of these tasks. Blobs (4-dimensional arrays) conceal the computational and mental overhead of mixed CPU/GPU operation by synchronizing from the CPU host to the GPU device as needed.

4.9 Microsoft COCO: Common Objects in Context[9]

This paper focuses on advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. A new large-scale dataset is introduced that addresses three core research problems in scene understanding they are detecting non-iconic views (or non-canonical perspectives) of objects, contextual reasoning between objects, the precise 2D localization of objects.

Annotation pipeline is split into 3 primary tasks they are category labeling: labeling the categories present in the image, instance spotting: locating and marking all instances of the labeled categories, instance segmentation: segmenting each object instance.The system currently only label ”things”, but labeling ”stuff” may also provide significant contextual information that may be useful for detection.

4.10 Deep Neural Network for Object Detection[10]

This paper explains a binary mask is created around the object.That is the pixel containing the value 1 inside the mask contains the object else it is 0.The binary mask is applied in multi-scale fashion, i.e in both vertical and horizontal fashion to identify the borders of the image. After that refinement process is applied and repeated until the classification is as precise as possible.So the basic approach is use the full image as an input and perform localization through regression using DNN.For precision of localization,DNN localizer is used on small set of large windows.

The comparison done with other techniques was somewhat biased, as this technique was trained on the larger VOC2012 training set while other techniques were published before that training set existed.Some mis-detections did occured due to similar looking objects or imprecise localization which was due to the ambiguous definition of object extend by the training data - in some images only the head of the bird is visible while in others the full body.

4.11 Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation[11]

In this paper a approach is introduced that uses Bounding boxes annotations than per pixel masks.To begin with an unsupervised region proposal method is used to generate candidate segmentation masks. The convolutional network is trained under the supervision of these approximate masks.Although the masks are coarse at the beginning, they are gradually improved and then provide useful information for network training.

So basically a rough mask is considered as a rectangle mask instead of going pixel by pixel using unsupervised methods (e.g Selective Search).A segmentation mask is generated using Grabcut method and it is trained against candidate segments to learn semantic features to pick better candidates.This process is iterated.After few iterations these method has been found as effective as pixel based mask method.

4.12 Explain Images with Multimodal Recurrent Neural Networks[12]

The architecture contains a language model part, an image part and a multimodal part. The language model part learns the dense feature embedding for each word in the dictionary and stores the semantic temporal context in recurrent layers. The image part contains a deep Convolutional Neural Network which extracts image features.The multimodal part connects the language model and the deep CNN together by a one-layer representation.

m-RNN model is learned using a perplexity based cost function.The errors are back-propagated to the three parts of the m-RNN model to update the model parameters simultaneously.

4.13 From Captions to Visual Concepts and Back[13]

In this paper,the system trains on images and corresponding captions, and learns to extract nouns, verbs, and adjectives from regions in the image. These detected words then guide a language model to generate text that reads well and includes the detected words. After this the deep multimodal similarity model is used to re-rank candidate captions.

Due to direct use of captions the caption detector trained from images with captions containing that specific word will be biased towards detecting an object corresponding to that word that are salient in the image.Training a language model (LM) on image captions captures commonsense knowledge about a scene. A multimodal joint representation for learning a caption detector is used which increases the efficiency of mapping between image features and text.

4.14 Multimodal semi-supervised learning for image classification[14]

The goal of this paper is to learn a classifier for images alone, but they use the keywords associated with labeled and unlabeled images to improve the classifier using semi-supervised learning. They first learn a strong Multiple Kernel Learning (MKL) classifier using both the image content and keywords, and use it to score unlabeled images. They then learn classifiers on visual features only, either support vector machines (SVM) or least squares regression (LSR), from the MKL output values on both the labeled and unlabeled images.

They also considered learning the textual-visual classifier and the visual-only classifier jointly, rather than sequentially, but it is unclear how to make the combined classifier benefit from the visual classifier.

4.15 Learning CNN-LSTM Architectures for Image Caption Generation[15]

This paper introduces a method which uses a Deep Convolutional neural network to create a semantic representation of an image, which is then decoded using a LSTM (Long-Short-Term Memory) network. All LSTMs share the same parameters. The vectorized image representation is fed into the network, followed by the special start of sentence token. The hidden state produced is then used by the LSTM to predict/generate the caption for the given image. Analogous to recent successful approaches in statistical machine translation. But using an encoder recurrent neural network, these model learn an expressive representation of the original sentence.Conducted extensive hyperparameter tuning on dropout to tackle overfitting.

The CNN-LSTM model learns to identify pictures in increasing detail and correct its earlier mistakes. However it is not always correct for unseen randomized validation images i.e the output was categorized into three sets which are namely Correct, Partially Correct and Wrong.

4.16 Support Vector Machine classification for Object-Based Image Analysis[16]

The SVM classification methodology was found very promising for Object-Based Image Analysis. Its overall accuracy was better than Nearest neighbor in every aspect when their confusion matrix was compared. The results were limited to test images up to 1000X1000 pixel size and very large remote sensing datasets is the barrier to overcome.

The main goal was to compare computation efficiency of SVM(Support Vector Machine) with Nearest neighbor Object-based Classifier results. A SVM approach for multi-class classification was followed , based on primitive image objects provided by a multi-resolution segmentation algorithm. Then a feature selection step, in order to provide the features for classification which involved spectral, texture and shape information. After that a module that integrated a SVM classifier and the segmentation algorithm was developed in C++. The SVM module is capable to use 4 types of kernels for training and classification: linear, polynomial, radial basis function and sigmoid.

It has great potential for remote sensing data. It surpassed Nearest Neighbor, Maximum Likelihood and Decision Tree Classifiers in robustness and accuracy.

4.17 Automatic Image Captioning[17]

The proposed new methods (Corr, Cos, SvdCorr, SvdCos) consistently outperform the state of the art EM (45% relative improvement) in captioning accuracy. The improved "adaptive" blob-tokens generation consistently leads to performance gains. The methods are less biased to the training set and more generalized in terms of retrieval precision and recall.

This paper includes proposition of four methods : Corr, Cos, SvdCorr, SvdCos to estimate a translation table, whose element can be viewed as the probability used to caption the term which is used as blob-token. The blob-tokens are generated using the K-means algorithm on feature vectors of all image regions in the image collection , with the number of blob-tokens, B, set at 500 (not optimal). The correlation-based translation table T_{corr} is defined by normalizing each column of $\mathrm{T}_{corr,0}$ such that each column sum up to 1. T_{corr} measures the association between a term and a blob-token by the co-occurrence counts or how similar the overall co-occurrence pattern of a term and a blob-token is.

4.18 Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models[18]

This paper describes a new approach to the problem of image caption generation, casted into the framework of encoder-decoder models. For the encoder, they use a joint image-sentence embedding where sentences are encoded using long short-term memory (LSTM) recurrent neural networks. Image features from a deep convolutional network are projected into the embedding space of the LSTM hidden states. A pairwise ranking loss is minimized in order to learn to rank images and their descriptions. For decoding, they introduce a new neural language model called the structure-content neural language model (SC-NLM). The SC-NLM differs from existing models in that it disentangles the structure of a sentence to its content, conditioned on distributed representations produced by the encoder.The SC-NLM model uses a sequence of word-specific structure variables, the structure variables help guide the model during the generation phrase and can be thought of as a soft template to help avoid the model from generating grammatical nonsense.

The result of this technique when compared with nearest neighbour was arguably good.And comparing with best results of Treetalk it could closely describe with the original captions. The model cannot align parts of captions to images and use these alignments to determine where to attend next.

4.19 Learning like a Child: Fast Novel Visual Concept Learning from Sentence Descriptions of Images[19]

It is known that children can form quickly and rough hypotheses about the meaning of new words in a sentence based on their knowledge of previous learned words.Similar approach has been used in this paper.They start with a model that has already been trained with a large amount of visual concepts.Thus they propose a method that allows the model to enlarge its word dictionary to describe the novel concepts using a few examples and without extensive retraining. In particular, we do not need to retrain models from scratch on all of the data. But fine-tuning the whole model using only the new data causes severe overfitting on the new concepts and decrease the performance of the model for the originally trained ones. To solve those problems they had to first fix the originally learned weights and then fix the baseline probability.

This technique performs comparably with the model retrained from scratch on all of the data if the number of novel concept images is large, and performs better when there are only a few training images of novel concepts available.This can save huge computing power that we would have needed if we had trained the dataset from scratch.

4.20 Minds Eye: A Recurrent Visual Representation for Image Caption Generation[20]

The proposal of the bi-directional mapping between images and their sentence-based descriptions is introduced. Critical to the approach is a recurrent neural network (RNN) that attempts to dynamically build a visual representation of the scene as a caption is being generated or read. The representation automatically learns to remember long-term visual concepts. The model is capable of both generating novel captions given an image and reconstructing visual features given an image description. For learning, the Backpropagation Through Time (BPTT) Algorithm is used. The RNN model is bi-directional. Thus it can generate image features from sentences and sentences from image features. The model is capable of learning long-term interactions.

The first bi-directional model is capable of generating both novel image descriptions and visual features. The model is capable of learning long-term interactions. But the use of LSTM models that can have a bidirectional model as well as ability to learn long-term concepts is not entertained.

Chapter 5

Literature Survey Analysis

5.1 Analysis of Methods Used to Extract Features from image

5.1.1 YOLOv2 (Grid Approach) (Object localization with regression)

It predicts a bounding box for each of the object present in that image and all the bounding boxes and their corresponding class probabilities are predicted by a single CNN model simultaneously. As compared to state of the art detection systems YOLO makes localization errors (mainly because CNNs have a tendency to fail in localization). The cross-entropy function used for training and detection in small as well as large bounding boxes have a great difference of errors depending upon the size of boxes. It deals with processing of objects in real time video, which is currently in our future work.

5.1.2 RCNN (Region Convolutional Neural Network

It proposes a bunch of boxes in the image and see if any of them actually correspond to an object. R-CNN create these bounding boxes using a process called Selective Search. At a high level, Selective Search looks at the image through windows of different sizes and for each size tries to group together adjacent pixels by texture, color, or intensity to identify objects. Then it wraps the region to a standard square size and sends it to the classifier. It is really quite slow because it requires a forward pass of the CNN for every region proposal for every single image (thats around 2000 forward passes per image). It also has to train three models separately - the CNN to generate image features, the classifier that predicts the class, and the regression model to tighten the bounding boxes. This makes the pipeline extremely hard to train.

5.1.3 CNN with mean pooling

Like every other technique this technique has pooling layers present after each activation of the inputs obtained from a convolution layer but the average or mean pooling technique lags to extract distinct features or in other words the foreground instances or objects that we are interested in.Hence at every pooling instance the neural networks fails to obtain an optimum convergence and thus resulting in loss of classification accuracy and failure to predict the labels of the object present correctly.

5.1.4 LRCNN (Video) (Long Term Recurrent CNN)

This approach uses frames of the video as input to corresponding CNN with LSTM units corresponding to each which produce a sentence summary based on a strong visual time series model. It gives a worse mAP value in case of certain word classes. It is inferior compared to VGGNet model.

5.1.5 DCNN (Deep CNN

This approach takes lots of time for training the model and also it requires huge amount of training data to achieve the required accuracy, hence, it becomes less efficient when compared to other existing models and architectures of CNN.

5.1.6 DPMv5 (Deformable Part Models)

The DPM models parts with additional learned filters in positions anchored with respect to the whole object bounding box, allowing parts to be displaced from this anchor with learned deformation costs. The strong DPM has adapted this method for the strongly supervised setting in which part locations are annotated at training time. A limitation of these methods is their use of weak features (usually HOG).

The Selected Preferred Approach:

5.1.7 Fast RCNN (Region Convolutional Neural Networks)

It runs the CNN just once per image and uses a technique known as RoIPool (Region of Interest Pooling). At its core, RoIPool shares the forward pass of a CNN for an image across its subregions. Then, the features in each region are pooled (max pooling). So all it take is one pass of the original image as opposed to 2000. Fast R-CNN jointly trains the CNN, classifier, and bounding box regressor in a single model. It uses a single network to compute all three.

5.2 Analysis of Methods Used for Caption Generation

5.2.1 RNN (Recurrent Neural Network)

The basic idea is that RNN networks have loops. These loops allow the network to use information from previous passes, which acts as a memory. The length of this memory depends on a number of factors. **Vanishing gradient point problem** still exists in this system but in reverse.This problem causes the RNN to have trouble in remembering values of past inputs after more than 10 timesteps approx.

5.2.2 Deep RNN

It uses one or two hidden layers. The main advantage is that they can be used for difficult to complex problems. However, they need long training time sometimes. **Vanishing gradient point problem** is one of foremost reasons due to which deep neural networks like RNN become very hard to train,in the sense that the training takes an infeasible or a very large amount of time resulting in a very slight change in the parameters of the neural network.Due to this the neurons become dead and have extremely low learning process affecting the accuracy badly and rendering the RNN useless.

5.2.3 BRNN (Bidirectional RNN)

The RNN model is bi-directional. Thus it can generate image features from sentences and sentences from image features.

5.2.4 m-RNN (Multimodal RNN)

This architecture uses the inferred alignments to learn to generate novel description of image regions.

The Selected Preferred Approach:

5.2.5 Long Short Term Memory

It succeeded in overcoming the vanishing gradient problem, by introducing a novel architecture consisting of units called Constant Error Carousels. LSTM were thus able to learn very deep RNNs and successfully remembered important events over long (thousands of steps) durations of time.

5.3 Analysis of Methods used for Classification

5.3.1 RMI - SVM (Relaxed Multiple Instance - SVM)

Multiple instance learning consists of a bag (in this case a video frame) and an instance (a box or a cube in a frame) in the bag.The relation between the bag label and instance label is called MIL constraints(in the proposed method RMI-SVM).Object of interest in the labelled bags are considered as positive instance else they are negative instances. In RMI - SVM , The MIL constraints are prepared methodically using the NOR model.The NOR model bridges the gap between bag and the instances.For eg if all the instances are found negative we can conclude the bag is negative using the NOR model.The instances are detected using simple linear SVM. Finally the optimization is done using Stochastic Gradient Descent.

5.3.2 MKL (Multiple Kernel Learning)

MKL is used to combine heterogeneous information and is used as a similarity measure. MKL offers an alternative to feature vector and are compute efficient and fast since they can be mapped to dot products.Thus any learning algorithm that works with dot products can be also implemented using MKL method. The MKL workflow looks like as follows: Extracting features from available sources, MKL constructs kernel matrices for (Different features, Different kernel types, Different kernel parameters), Finding optimal combination of kernel and kernel classifier.

5.3.3 SVM (Support Vector Machines)

We will be using a large dataset and training time will be high, and SVM doesnt perform well with a large dataset. If the dataset has more noise it is not efficient in finding the target classes. It doesnt provide probability estimates and we have to use an expensive cross validation methods.

5.3.4 LSR (Least Squares Regression)

a. Suppose we have a regression model which classifies a relationship among variables.Our aim is to classify the variables as accurately as possible.So we take measure of the distance of individual variables from the regression model.We take the sum of distances and square them up,let the output we get be represented by Distance D.The more we minimize the distance D the more precise model we get.This method is Least Square Regression.

5.3.5 Cos, Corr, SVDcos, SVDcorr (Single Value Decomposition Correlation)

Singular value decomposition gives us the way to decompose a matrix of any size say NxN into three matrix with having some constraint.The matrix decomposes into UxE(read sigma)xV* where U and V* are orthogonal matrix and E(sigma) is a diagonal matrix having values only at diagonal.It is used to get rid of redundant data or to reduce the dimensions of a matrix.

The Selected Preferred Approach:

5.3.6 Softmax Classifier

Softmax classifier outputs probabilities rather than margins. Probabilities are much easier for us as humans to interpret, so that is a particularly nice quality of Softmax classifiers.

5.4 Performance Metrics Used

- BLEU (1,2,3,4)
- METEOR
- mAP (mean Average Precision)
- Recall@k and median rank (R@1 R@5 R@10 Med r)
- Average Precision
- CIDEr
- ROUGE-L
- CorLoc
- CIDEr-D
- PPL (Perplexity)

5.5 Datasets

- MSCOCO
- Flickr8k and Flickr30k
- PASCAL VOC 2012, Pascal-1k, Pascal 2007, Pascal06-all and Pascal07-all
- Picasso
- People Art
- CIFAR-10
- Pascal-Context
- IAPR TC-12
- Corel Image
- Proposed NVC
- TREC9
- Yummly

5.6 Libraries

- Keras
- Tensorflow
- Theano

5.7 Architectures of Convolutional Networks

- AlexNet
- ImageNet
- ResNet (ResNet50)
- VGGNet (VGG16 and VGG19)
- GoogLeNet/Inception (Inception V3)
- Xception
- SqueezeNet

Chapter 6

System architecture

6.1 CNN

This module is designed to extract the high level features of the input image from computed image matrix. It uses different functions like convolution2D, ReLULayer, MaxPool, generateFClayer.

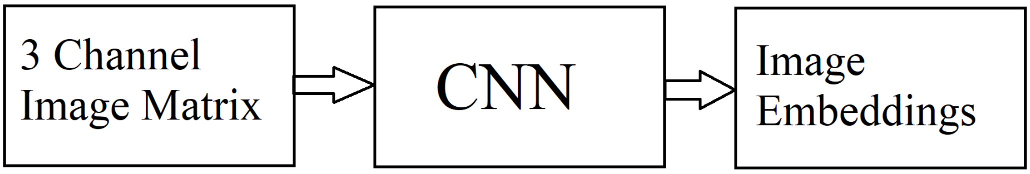

Figure 6.1: CNN System Architecture

6.2 Low Level Feature Extractor

This module applies convolution filter after ReLU activation is done successfully. After then, it implements MaxPool function to obtain Max Pooled Image matrix with all its parameters.

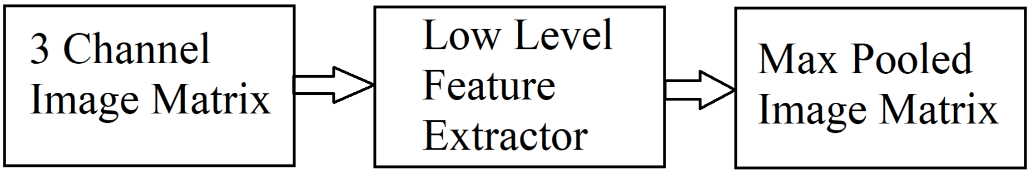

Figure 6.2: Low Level Feature Extractor

6.3 High Level Feature Extractor

In this module, the Max Pooled Image Matrix is processed to generate Fully Connected layer which is used to extract high level features embeddings. It involves the implementation of functions like generateFlattenlayer, dropOut, generateFCLayer.

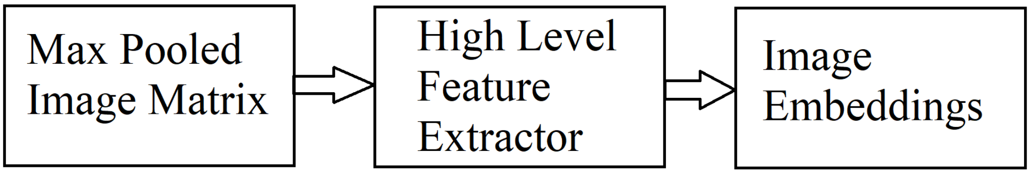

Figure 6.3: High Level Feature Extractor

6.4 LSTM

The module is designed to take high level feature image embeddings as input to generate the captions describing the input image with the help of functions like getFeatureEmbeddings, generateCaptions.

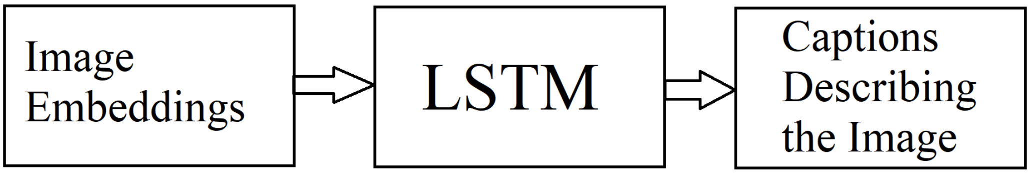

Figure 6.4: LSTM System Architecture

6.5 LSTM Memory Cell

This module consists of very integral part of LSTM i.e LSTM Memory Cell which generate the predicted next word vector and also passes an input value to next hidden cell. In this module, there are three inputs to be given to each memory cell which consists of high level feature image embeddings, word vector embeddings from previous layers and tanh function output from previous hidden cell. It makes use of different functions like tanh, sigmoid, inputGating, cellStateCompute and outputGating.

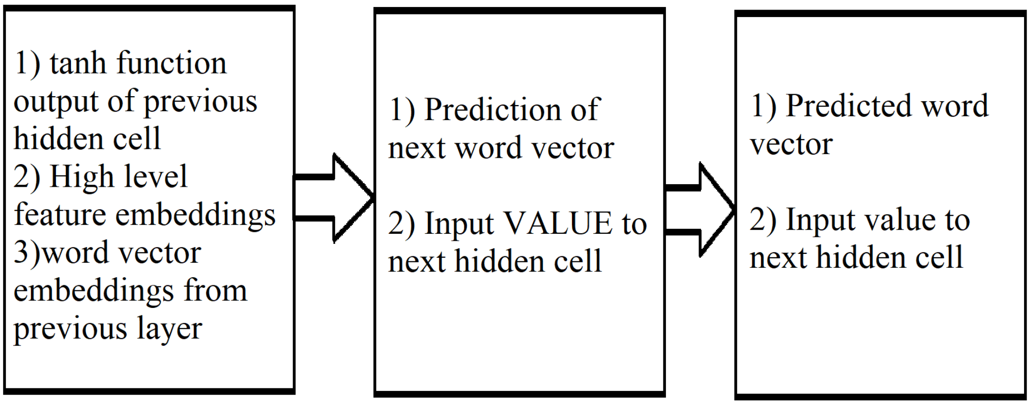
1) tanh function output of previous hidden cell
2) High level feature embeddings
3)word vector embeddings from previous layer
1) Prediction of next word vector
2) Input VALUE to next hidden cell
1) Predicted word vector
2) Input value to next hidden cell

Figure 6.5: LSTM Memory Cell

Chapter 7

Hardware and Software requirements

7.1 Hardware Requirements:

- Memory: 8GB or more
- Chipset: i5 or above(preferably 64 bit)
- GPU: NVIDIA GTX Series

7.2 Software Requirements:

- **Platform:** Python Virtual Machine on python 2.7/3
- **Operating System:** Ubuntu 14.04 LTS or above
- **Programming Language:** Python
- **Dataset:** PASCAL VOC 2012 or CIFAR-10
- **NLP libraries and frameworks:** NLTK, NEON framework
- **Deep Learning libraries:** Keras,Tensorflow
- **Dependency packages:** Scipy,Numpy

Chapter 8

Feasibility study

8.1 Technical feasibility

8.1.1 Software feasibility

The software components used in this project are open source libraries and platforms. Python is the primary programming language used to implement the project. In order to build neural network models keras open source library has been used which provides a simpler way to code the model along with open source library for machine learning which is Tensorflow as a backend for keras implementation. Also neon framework has been used which is an open source deep learning framework extremely robust and fully optimized for implementation on hardware.

8.1.2 Hardware feasibility

The hardware components are required mainly to increase the speed of training the neural network and test time implementation. The training takes roughly 7 to 8 hours to train a model with a dataset of 60000 images on an NVIDIA GPU by using a batchwise training approach optimally utilizing all the cores which doesnt exceed the desired limit of optimal time required. Param Shavak has also been used to train the same model and the same dataset which gives an extremely high performance of training and opens a window for datasets having large number of training examples or even those datasets having larger number of features.

8.2 Economical Feasibility

The software and the hardware part of the project implementation doesnt require any cost so far. Due to use of free and open source libraries for deep learning, optimization and preprocessing for the software part and due to the use of Param Shavak which doesnt incur any cost for training the model.

8.3 Schedule Feasibility

The solution would be built in an estimated time of 7-8 months. The model building,training and implementation would be divided into sequential phases as each component in the project is functionally dependent on the previous one. Time constraints have been imposed for every phase and especially in the phase of increasing accuracy and fine tuning the parameters of the model and making it more robust.

8.4 Operational Feasibility

The system would be used for commonplace purpose like an application for children to learn scene description and even recognizing objects in the image shown to them.Such an application would prove to be beneficial for pre-primary kids.

Chapter 9

Design

9.1 UML modeling

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system.UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by Object Management Group and UML 1.0 specification draft was proposed to the OMG in January 1997.

9.1.1 Goals of UML:

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

9.1.2 UML diagrams for the project:

1. **Use Case diagram**

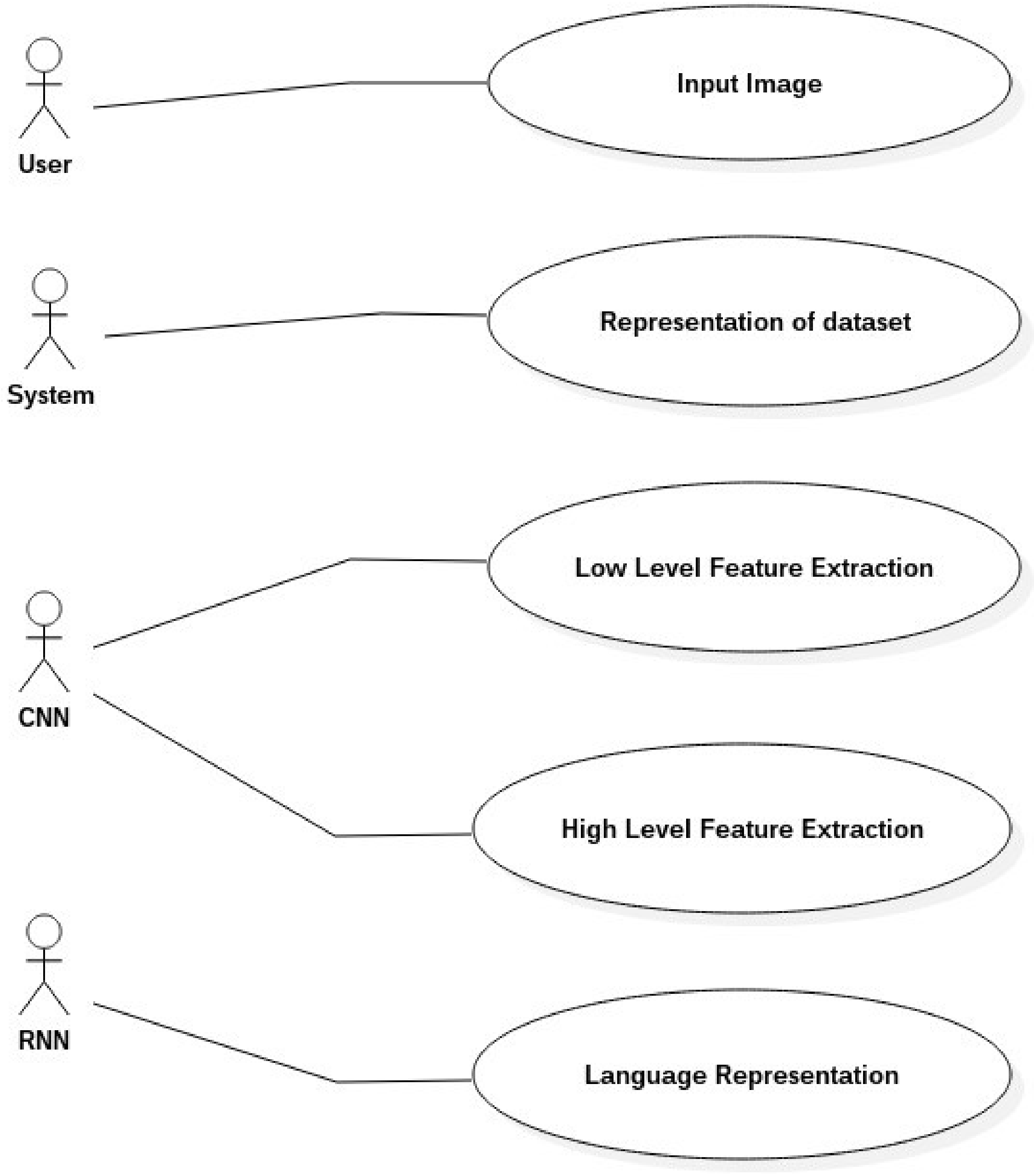
User
Input Image
System
Representation of dataset
Low Level Feature Extraction
CNN
High Level Feature Extraction
RNN
Language Representation

Figure 9.1: Use Case diagram

A use case diagram at its simplest is a representation of a users interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that "Use case diagrams are the blueprints for your system".

2. **Class diagram**

   A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the systems classes, their attributes, operations (or methods), and the relationships among objects.

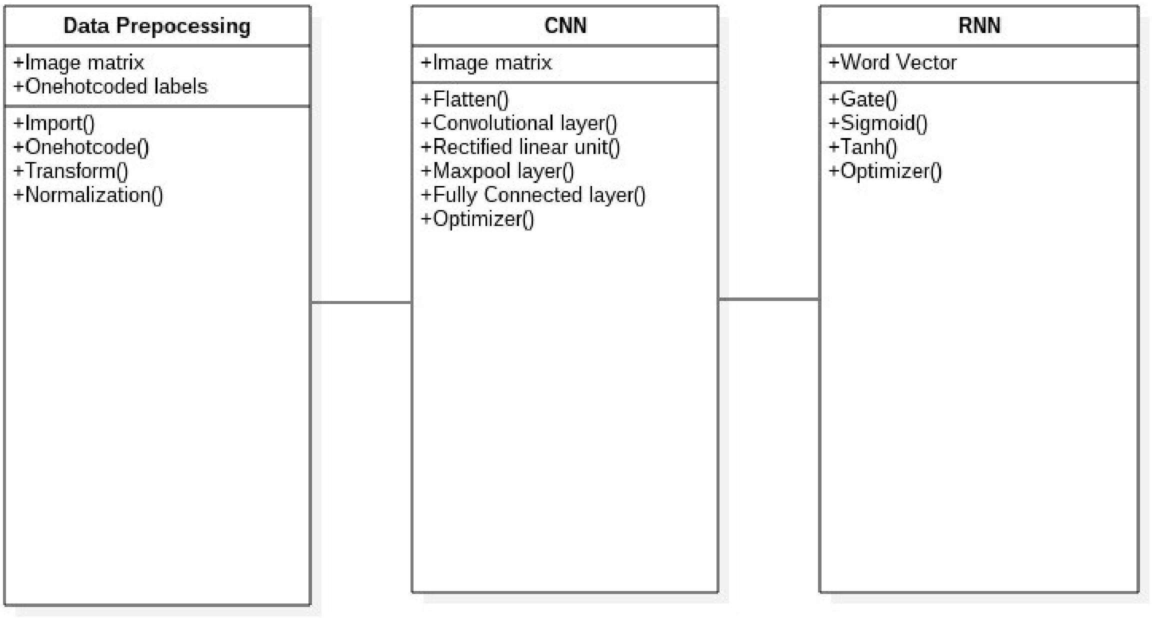

Figure 9.2: Class diagram

The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

3. **State diagram**

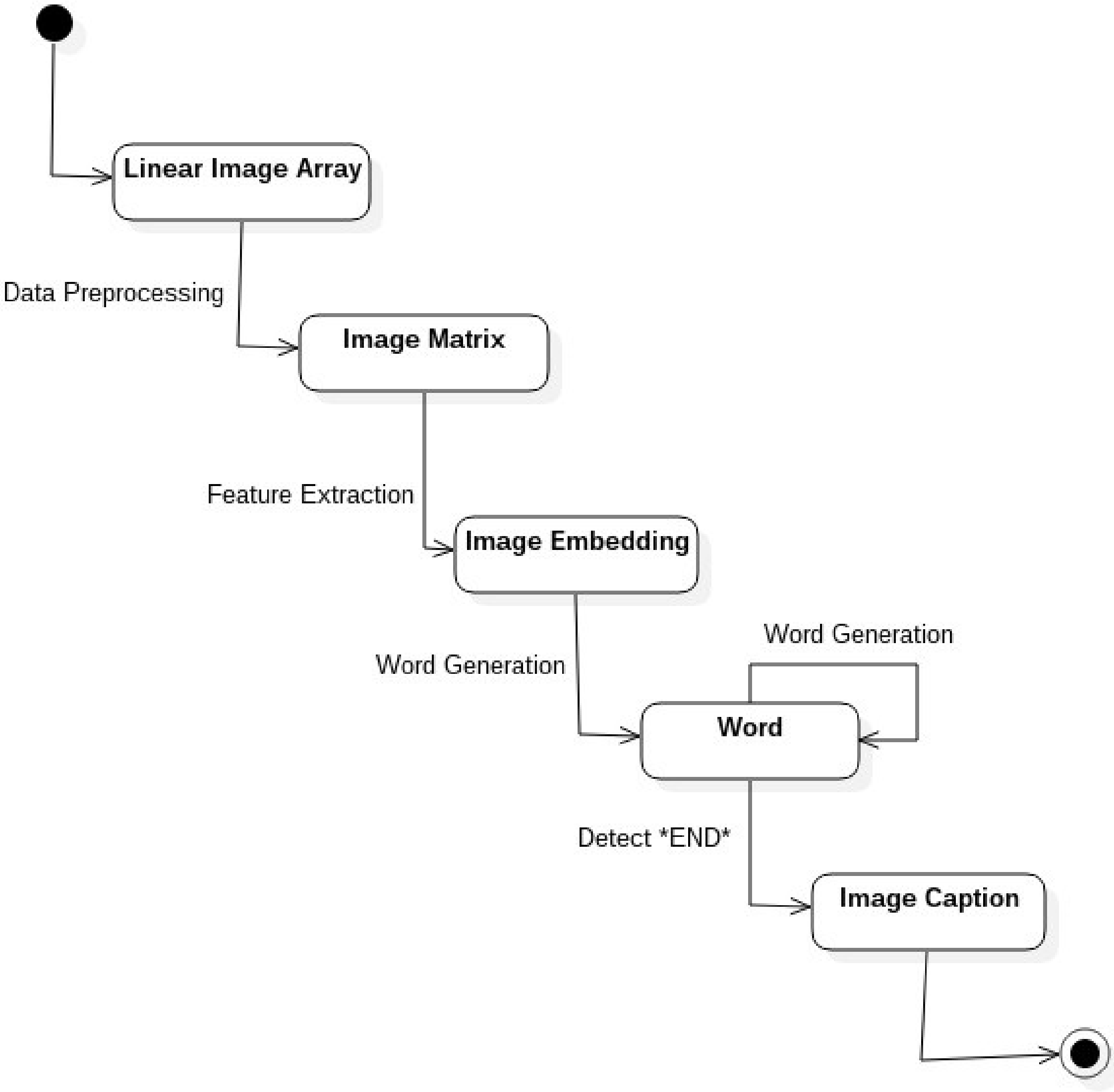

Figure 9.3: State diagram

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.State diagrams are used to give an abstract description of the behavior of a system. This behavior is analyzed and represented as a series of events that can occur in one or more possible states. Hereby each diagram usually represents objects of a single class and track the different states of its objects through the system.

Chapter 10

Time-line analysis

Table 10.1: Plan of project execution

Sr.no.	Time span(per-phase)	Phase
1.	July	Formation of group and discussion of domain
2.	August	Searching of topic
3.	September-October	Analysis and Review of Existing Approaches
4.	November	Functional Design and Architecture
5.	December-February	Coding and Unit Testing
6.	March	Integration Testing
7.	April	Report Writing

Chapter 11

Conclusion

This report covers an extensive list of the initial requirements and analysis of the project and provides a blueprint of the functionality and productivity delivered by the system. Thus, with the implementation of the said modules properly, the system is able to generate captions for a given input image at real time.

Chapter 12

Future scope

1. A Question Answering System on Videos for Public and Personal Security and Surveillance.
2. Assisting Visually Impaired People.
3. Environmental Analysis for Robotic System.

Bibliography

[1] YOLO 9000: Better, Faster, Stronger.
Author: Joseph Redmon, Santosh Divvala, Ross Girshick, ALi Farhadi.
Year: 2016

[2] Relaxed Multiple-Instance SVM with Application to Object Discovery
Author: Xinggang Wang, Zhuotun Zhu, Cong Yao, Xiang Bai.
Year: 2015.

[3] Translating Videos to Natural Language Using Deep Recurrent Neural Networks.
Author: Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, Kate Saenko.
Year: 2015.

[4] Long-term Recurrent Convolutional Networks for Visual Recognition and Description.
Author: Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, Trevor Darrell.
Year: 2016.

[5] Show and Tell: A Neural Image Caption Generator.
Author: Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan.
Year: 2015.

[6] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.
Author: Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio. Year: 2016.

[7] Deep Visual-Semantic Alignments for Generating Image Descriptions.
Author: Andrej Karpathy, Li Fei-Fei.
Year: 2015.

[8] Caffe: Convolutional Architecture for Fast Feature Embedding.
Author: Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell.
Year: 2014.

[9] Microsoft COCO: Common Objects in Context. Author: Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollar. Year: 2015.

[10] Deep Neural Network for Object Detection.
Author: Christian Szegedy, Alexander Toshev, Dumitru Erhan.
Year: 2013.

[11] Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation.
Author: Jifeng Dai, Kaiming He, Jian Sun.
Year: 2015.

[12] Explain Images with Multimodal Recurrent Neural Networks.
Author: Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Alan L. Yuille.
Year: 2014.

[13] From Captions to Visual Concepts and Back.
Author: Hao Fang, Li Deng, Margaret Mitchell, Saurabh Gupta, Piotr Dollr, John C. Platt, Forrest Iandola, Jianfeng Gao, C. Lawrence Zitnick, Rupesh K. Srivastava, Xiaodong He, Geoffrey Zweig.
Year: 2015.

[14] Multimodal semi-supervised learning for image classification.
Author: Matthieu Guillaumin, Jakob Verbeek and Cordelia Schmid.
Year: 2010.

[15] Learning CNN-LSTM Architectures for Image Caption Generation.
Author: Moses Soh.
Year: 2016.

[16] Support Vector Machine classification for Object-Based Image Analysis.
Author: A. Tzotsos, D. Argialas.
Year: 2008.

[17] Automatic Image Captioning.
Author: Jia Yu Pan, Hyung Jeong Yang, Pinar Duygulu, Christos Faloutsos.
Year: 2004.

[18] Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models.
Author: Ryan Kiros, Ruslan Salakhutdinov, Richard S. Zemel.
Year: 2014.

[19] Learning like a Child: Fast Novel Visual Concept Learning from Sentence Descriptions of Images.
Author: Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, Alan Yuille.
Year: 2015.

[20] Minds Eye: A Recurrent Visual Representation for Image Caption Generation.
Author: Xinlei Chen, C.Lawrence Zitnick.
Year: 2015.