# Video Description using Deep Learning

**Summer Undergraduate Research Award**


**Suyash Agrawal**
2015CS10262
Computer Science
CGPA: 9.92
Mob: 9717060183
cs1150262@iitd.ac.in

**Madhur Singhal**
2015CS10235
Computer Science
CGPA: 8.66
Mob: 9540972599
cs1150235@iitd.ac.in

**Supervisor:-**
**Subhashis Banerjee**
Professor
Department of CSE
suban@cse.iitd.ac.in
IIT Delhi

**Prof. S. Arun Kumar**
Head of Department
Department of CSE
sak@cse.iitd.ernet.in

# 1 Introduction

***Video Description*** is the process of discovering knowledge, structures, patterns and events of interest in the video data and describing them in natural language. Video Description is an incredibly hard problem in computer vision and currently the only source of video description is manual labour.
Video Description has wide variety of applications. It can help visually impaired people "see" the world by describing the scene around them. It has also use in automated surveillance by analyzing the videos in real time and reporting criminal activities. It can also be used to efficiently index large video databases based upon their content for ease of accessibility.


Figure 1: Sample Video Description

Most of the prior work in this field is focused on generating natural language descriptions from images.
Our aim is to utilize these new techniques in the field of image captioning and generate natural language descriptions from video data. Basically this project will involve first encoding video data using CNNs and then using RNNs (specifically LSTMs) to generate descriptions of these videos. Further, we will also explore using optical flow based models to improve description of videos.

# 2 Objectives

Our main objective is to create a model capable of understanding videos and generating natural language descriptions of them. We will also create an end-user application for demonstration purposes. This task can be further divided into following sub-tasks:

1. To organize all available sources of data and split them into training, validation and testing sets.
2. To set up CNNs with pretrained weights for image captioning.
3. To construct Long Short Term Memory (LSTM) Encoder Decoders.
4. To train the network using the test set data and use validation set data for setting hyper-parameters.

5. To benchmark results and improve using techniques like attention modelling.

# 3 Basic Concepts

## 3.1 Deep Learning

Deep Learning is a branch of machine learning in which multiple parameter based models are used in series. In a deep network, there are many layers between the input and output, allowing the algorithm to be executed in multiple processing steps, composed of **multiple linear and non-linear transformations**. At each layer, the signal is transformed by a processing unit, like an artificial neuron, whose parameters are **'learned'** through training. Deep Learning has been shown to excel in tasks where the goal is to find **intuitive** patterns in the data.[8] In particular, in the field of Computer Vision, deep networks are increasingly used to extract **feature descriptions and inter-relationships between features** from images.[10]

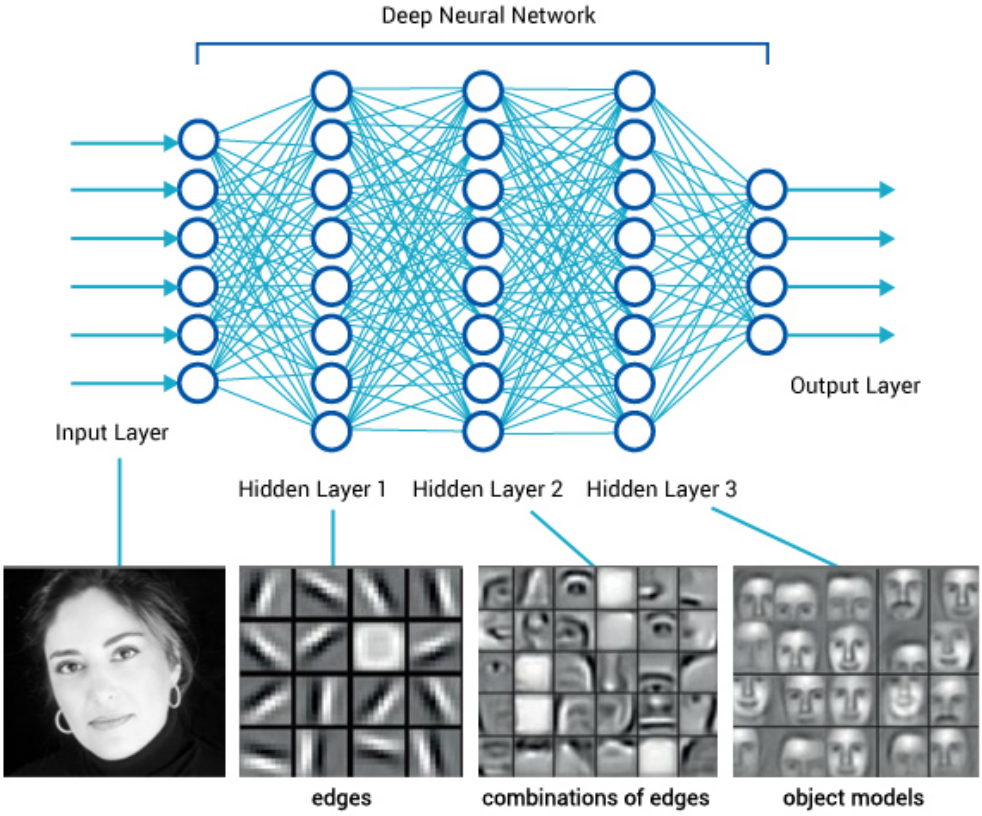


Figure 2: Illustration of Deep Learning as applied to Vision

### 3.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN, or ConvNet) are a type of feed-forward artificial neural network in which the connectivity pattern between the neurons is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to stimuli in a restricted region of space known as the **receptive field**. The receptive fields of different neurons partially overlap such that they tile the visual field. The response of an individual neuron to stimuli within its receptive field can be approximated mathematically by a **convolution operation**. A Convolutional Neural Network consists of the following layers.[3]

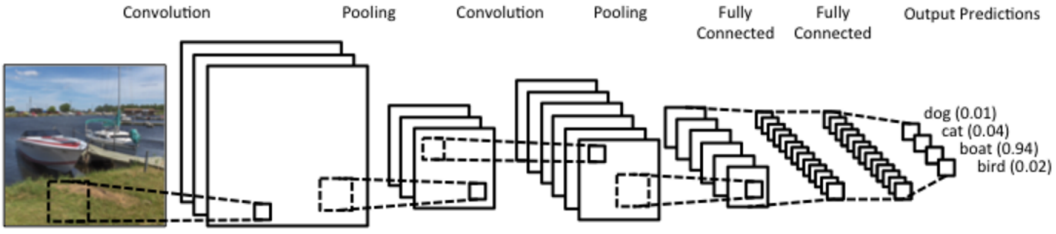


Figure 3: A Typical Convolutional Neural Network

#### 3.2.1 Convolutional Layer

The convolution layer is the core building block of a CNN. The layer's parameters consist of a set of **learnable filters** (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2-dimensional activation map of that filter.[10] As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input.

#### 3.2.2 Max Pooling Layer

It is common to periodically insert a Pooling layer in-between successive Conv layers in a ConvNet architecture. Its function is to **progressively reduce the spatial size** of the representation to reduce the amount of parameters and computation in the network, and hence to also control over-fitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the max operation.[10]

### 3.2.3 Fully-Connected Layer

Finally, after several convolutional and max pooling layers, the high-level reasoning in the neural network is done via fully connected layers. Neurons in a fully connected layer have **full connections to all activations** in the previous layer, as seen in regular Neural Networks. Their activations can hence be computed with a matrix multiplication followed by a bias offset.[11] Thus output of the fully connected layer is a vector with elements representing the 'probability' (not in a strictly statistical sense) of the image containing specific objects or actions.

## 3.3 Long Short Term Memory Networks

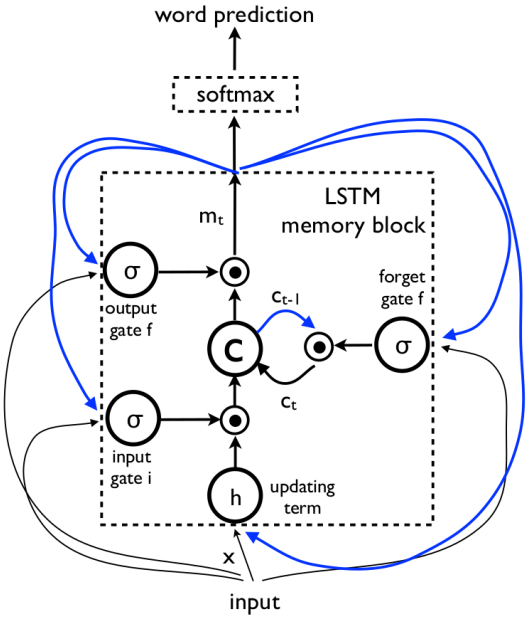


Figure 4: A single LSTM unit

Long Short Term Memory Networks are a type of Recurrent Neural Networks. These networks are based upon recursion, so that variable length inputs can be handled easily and sequential information can be processed with better results. LSTM's are specifically used for making RNNs learn long term patterns since traditional RNNs tend to **favour short term temporal dynamics**. It can be difficult to train traditional RNNs to learn long-term dynamics, likely due in part to the **vanishing and exploding gradients problem** that can result from propagating the gradients down through the many layers of the recurrent network, each corresponding to a particular time step[7]. LSTMs provide a solution by incorporating memory units that explicitly allow the network to learn when to "forget" previous hidden states and when to update hidden states given new information[4].

Below we list the main equations governing the behaviour of the LSTM networks.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$
$$h_t = o_t \circ \sigma_h(c_t) \tag{5}$$

The symbol meanings are:
$\mathbf{x_t}$: Input vector
$\mathbf{h_t}$: Output vector
$\mathbf{c_t}$: Cell state vector
$\mathbf{W}$, $\mathbf{U}$ and $\mathbf{b}$: Parameter matrices and vector
$\mathbf{f_t}$: Forget gate vector. Weight of remembering old information.
$\mathbf{i_t}$: Input gate vector. Weight of acquiring new information.
$\mathbf{o_t}$: Output gate vector. Output candidate
$\sigma_\mathbf{g}$, $\sigma_\mathbf{c}$ and $\sigma_\mathbf{h}$: Activation functions

### 3.4 Training Deep Neural Networks

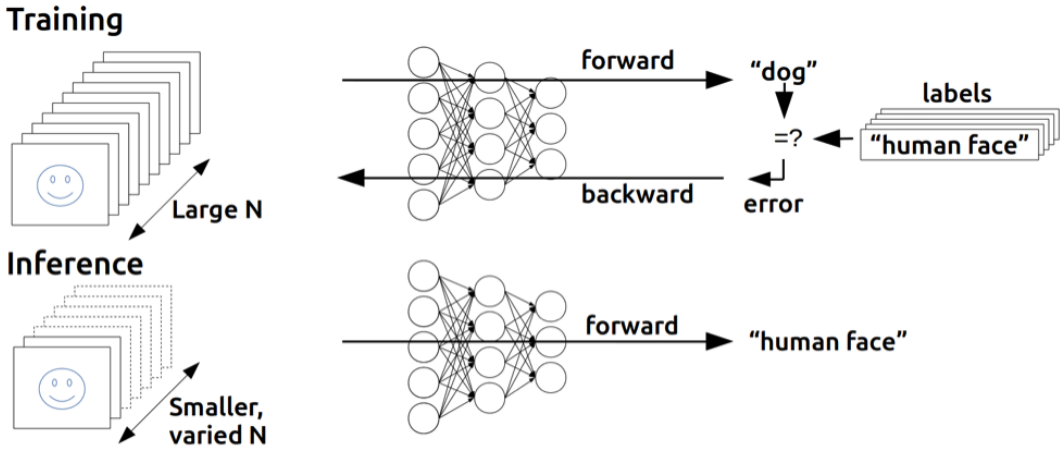


Figure 5: Training and Inference Processes

A Deep Neural Network is at it's core a parameter based function. All of these parameters are **trained** automatically from inputs and expected output tuples (training data). The training process revolves around minimizing a particular cost function using methods like **Stochastic gradient descent**. The input is given to the network in a feed forward

fashion and the parameters are modified from the last layer to the first **(Backpropagation)**. Neural Networks, by design, require huge amounts of training data and take a large time to get trained. For some perspective, most current state of the art image classifiers have $> 100$ million parameters and are trained on more than 1.2 million images.

### 3.5 Finetuning

Fine-tuning a network is a procedure based on the concept of **transfer learning**. We start training a CNN to learn features for a broad domain with a classification function targeted at minimizing error in that domain. Then, we replace the classification function and **optimize the network** again to minimize error in another, more specific domain. Under this setting, we are transferring the features and the parameters of the network from the broad domain to the special one.[9] In our project we will need to use the pre-trained image classification models to actually decode individual frames of the video, thus we are planning to **fine-tune those models with respect to the output of our LSTMs**.

## 4 Approach to the project

First, we shall be implementing image captioning in order to be able to encode individual video frames in fixed-length vector representation. Then, we will be using our constructed CNN to encode video frames sampled at a fixed interval. We will then use RNNs to translate this representation of video into natural language domain. This translation will be achieved by using a set of encoder and decoder RNNs. Since we are working with variable length input and output, we will specifically be using LSTMs for encoding and decoding purposes as they have been proven to be excellent in machine translation and generalize very well on long data input.

1. Image Captioning
   (a) Construct a VGG/Inception V3 with pre-trained weights for object classification.
   (b) Make an LSTM network[3] with CNN encoded image as input that will be used to translate the image representation into natural language text.
   (c) Train the model on datasets like MS COCO.
2. Video Representation
   (a) We will first sample video at a fixed rate and convert each individual frame to a fixed length vector representation using the CNN trained in image captioning part.
   (b) We will then try out different approaches of video representation like:
      - Mean Pooling over all video frame encodings obtained to get a fixed vector representation of video[1].
      - Using RNNs to encode this variable length video representation[2].

- Using 3D CNNs to directly encode videos and use simple RNNs for description generation[5].

3. Natural Language Conversion

   (a) Based on how we choose to encode our video, we will have to select appropriate architecture of LSTM to be used to decode the representation

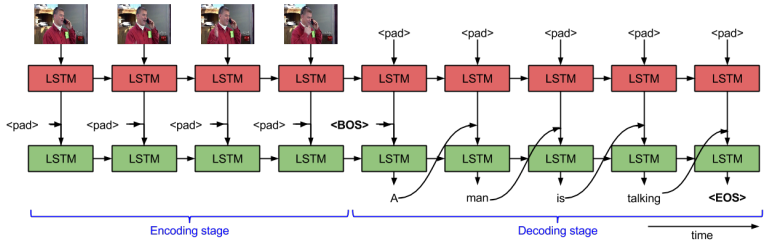


Figure 6: Video description model with 2 LSTM levels

   (b) One popular choice[2](Figure 6) that we will try out first will be to use a two level LSTM model that will do a sequence to sequence mapping from variable length video representation to variable length natural language sentence.

   (c) Then we will train our model on the training data we have obtained and plot the learning curves.

   (d) We will also have to check for over-fitting and under-fitting during our training process and finetune our hyper parameters according to it.

4. Further Possibilities

   (a) We will look for techniques of data augmentation and transfer learning[1] to compensate for the limited amount of training data for video description.

   (b) We will also look into some recent techniques of using optical flow for attention modelling[2] which has shown in some cases to improve the results of action recognition.

   (c) We will also consider using more efficient vocabulary representation as compared to one hot encoding because vocabulary in this case would be very large.

   (d) Try to make forward propagation fast and more memory efficient.

   (e) We will also try to develop an end user application that will speak out description of a video that a person records.

   (f) A future extension of this project is to design a system which can answer questions based upon a video, similar to [6].

# 5 Data sets

Though video description datasets are scarce, we have found some viable options that can be utilized. These datasets are not very large (unlike image captioning datasets like MSCOCO), but we are confident that indicative results can be obtained from them. Some of these datasets are:

1. **Microsoft Research - Video to Text (MSR-VTT)**: The dataset contains 41.2 hours and 200K clip-sentence pairs in total, covering the most comprehensive categories and diverse visual content, and representing the largest dataset in terms of sentence and vocabulary.
2. **MPII Movie Description Dataset (MPII-MD)**: MPII-MD contains around 68,000 video clips extracted from 94 Hollywood movies. Each clip is accompanied with a single sentence description which is sourced from movie scripts and audio description (AD) data.
3. **Montreal Video Annotation Dataset (M-VAD)**: The M-VAD movie description corpus is another recent collection of about 49,000 short video clips from 92 movies. It is similar to MPII-MD, but only contains AD data and only provides automatic alignment.

We are also considering the possibility of using annotated movies for visually impaired people as datasets to train for video description.

# 6 Uses and applications

- Assisting visually impaired people to get description of their surroundings, thus enabling them to "see".
- Very useful for automated surveillance and theft detection by being able to analyze large amounts of data which is unfeasible to be done by humans.
- Allowing content based video retrieval by describing the contents of video in textual format which is indexable by web crawlers.
- This can also be used to detect catastrophic events through security cameras like fire breakout, murder etc.
- This project can also be applied in helping robotic vision as this project basically allows one to understand what is happening in the video and thus robots will be able to get a "true" sense of their surroundings.

# 7 Budget and duration

## 7.1 Budget

Rs. 10,000 will be required for purchasing GPU instance in AWS servers as the project will require an on-demand access to very powerful GPU during training process.

## 7.2 Duration

We will try to complete this project by the end of the summer break i.e. the end of July, 2017.

# References


[1] Subhashini Venugopalan *Natural Language Video Description using Deep Recurrent Neural Networks*, 2015.

[2] Venugopalan, Subhashini and Rohrbach, Marcus and Donahue, Jeff and Mooney, Raymond and Darrell, Trevor and Saenko, Kate *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015

[3] Oriol Vinyals and Alexander Toshev and Samy Bengio and Dumitru Erhan *Show and Tell: A Neural Image Caption Generator* 2014.

[4] Wojciech Zaremba, Ilya Sutskever *Learning to Execute* ICLR 2015

[5] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, Aaron Courville *Describing Videos by Exploiting Temporal Structure* ICCV 2015

[6] Aishwarya Agrawal, Jiasen Lu , Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, Devi Parikh *VQA: Visual Question Answering* 2016

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

[8] Bengio, Yoshua; LeCun, Yann; Hinton, Geoffrey *Deep Learning* 2015

[9] Angie K. Reyes, Juan C. Caicedo and Jorge E. Camargo *Fine-tuning Deep Convolutional Networks for Plant Recognition* LifeCLEF 2015

[10] Andrej Karpathy *CS231n Convolutional Neural Networks for Visual Recognition* http://cs231n.github.io/convolutional-networks/

[11] Santanu Chaudhury, Anupama Mallik, Hiranmay Ghosh *Multimedia Ontology: Representation and Applications*