Summer 8-11-2020

Deep Architectures for Visual Recognition and Description

Anuja Perunninakulath Parameshwaran

Recommended Citation

Perunninakulath Parameshwaran, Anuja, "Deep Architectures for Visual Recognition and Description." Dissertation, Georgia State University, 2020.
doi: https://doi.org/10.57709/18676819

DEEP ARCHITECTURES FOR VISUAL RECOGNITION AND DESCRIPTION

by

ANUJA P PARAMESHWARAN

Under the Direction of Michael Weeks, PhD

ABSTRACT

In recent times, digital media contents are inherently of multimedia type, consisting of the form text, audio, image and video. Several of the outstanding computer Vision (CV) problems are being successfully solved with the help of modern Machine Learning (ML) techniques. Plenty of research work has already been carried out in the field of Automatic Image Annotation (AIA), Image Captioning and Video Tagging. Video Captioning, *i.e.*, automatic description generation from digital video, however, is a different and complex problem altogether. This study compares various existing video captioning approaches available today and attempts their classification and analysis based on different parameters, *viz*., type of captioning methods (generation/retrieval), type of learning models employed, the desired output description length generated, etc. This dissertation also attempts to critically analyze the existing benchmark datasets used in various video captioning models and the evaluation metrics for assessing the final quality of the resultant video descriptions generated. A detailed study of important existing models, highlighting their comparative advantages as well as disadvantages are also included.

In this study a novel approach for video captioning on the Microsoft Video Description (MSVD) dataset and Microsoft Video-to-Text (MSR-VTT) dataset is proposed using supervised learning techniques to train a deep combinational framework, for achieving better quality video captioning *via* predicting semantic tags. We develop simple shallow CNN (2D and 3D) as feature extractors, Deep Neural Networks (DNNs and Bidirectional LSTMs (BiLSTMs) as tag prediction models and Recurrent Neural Networks (RNNs) (LSTM) model as the language model. The aim of the work was to provide an alternative narrative to generating captions from videos via semantic tag predictions and deploy simpler shallower deep model architectures with lower memory requirements as solution so that it is not very memory extensive and the developed models prove to be stable and viable options when the scale of the data is increased.

This study also successfully employed deep architectures like the Convolutional Neural Network (CNN) for speeding up automation process of hand gesture recognition and classification of the sign languages of the Indian classical dance form, '*Bharatnatyam*'. This hand gesture classification is primarily aimed at 1) building a novel dataset of 2D single hand gestures belonging to 27 classes that were collected from (i) Google search engine (Google images), (ii) YouTube videos (dynamic and with background considered) and (iii) professional artists under staged environment constraints (plain backgrounds). 2) exploring the effectiveness of CNNs for identifying and classifying the single hand gestures by optimizing the hyperparameters, and 3) evaluating the impacts of transfer learning and double transfer learning, which is a novel concept explored for achieving higher classification accuracy.

INDEX WORDS: Automatic Image Annotation (AIA), Computer Vision (CV), Image Captioning, Machine Learning and Video Tagging

DEEP ARCHITECTURES FOR VISUAL RECOGNITION AND DESCRIPTION

by

ANUJA P PARAMESHWARAN

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2020

DEEP ARCHITECTURES FOR VISUAL RECOGNITION AND DESCRIPTION

by

ANUJA P PARAMESHWARAN

Committee Chair: Michael Weeks

Committee: Rajshekhar Sunderraman

Juan Banda

George Pullman

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

August 2020

DEDICATION

This dissertation is dedicated to the two most important people in my life - my daughter Avani and husband Nikshep.

ACKNOWLEDGEMENTS

This dissertation work would not have been possible without the support of many people. First, I want to express my gratitude to my advisor Dr Michael Weeks for his constant support and guiding me to be a competent researcher. I must thank him for his patience in responding to over thousands of emails and giving important feedback in research meetings which helped distill many ideas pitched by me to discover fundamental concepts to place them in wider context. I am also grateful to him for reading through countless drafts and shaping my research work by providing insightful comments, teaching me the art of communicating ideas in an effective manner and for teaching me how to think and take an idea all the way to completion.

I would also like to thank Dr. Rajshekhar Sunderraman, my dissertation committee member, for his constant encouragement and support to push myself and do better. I also present my words of gratitude to Dr. Juan Banda and Dr. George Pullman, my other committee members, for their timely suggestions, help and guidance as members of the progress assessment committee.

Research work is indeed a joint effort of so many people. My studies were facilitated by the constant support and encouragement of my husband, Mr. Nikshep Patil and 4-year old daughter Avani, who were often deprived of my attention and care as I was busy with my studies. I express my sincere gratitude to both for their understanding and help during these challenging days. I am also deeply indebted to my beloved parents, Dr. P.S. Parameswaran and Ms. Padmavathy V K, brother Dr. Arun P Parameswaran, sister-in-law Ms. Dhatri, as well as my parents-in-law, Mr. Vijayendra Patil & Mrs. Mitravinda Patil as well as other relatives and friends for their invaluable support, encouragement, prayers and sacrifices, which made this work successful. Hereby, I express my sincere gratitude to them all for their help, concern and support during these studies.

I am also deeply indebted to my fellow colleagues, students and staff members of the Department of computer science, GSU for their active support and help. Special thanks are due to my colleague and close collaborator Ms. Heta Desai, for her wholehearted support and discussions that stimulated my research interest.

Lastly, I wish to thank all my friends, extended family members, and colleagues for their love and support during this PhD journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- 1D – One Dimensional
- 2D – Two Dimensional
- 3D – Three Dimensional
- AI – Artificial Intelligence
- AIA – Automatic Image Annotation
- aLSTM – Attention based long Short-Term Memory
- AMT – Amazon Mechanical Turk
- ANN – Artificial Neural Network
- BiLSTM – Bidirectional Long Short-Term Memory
- BLEU – BiLingual Evaluation Understudy
- BoW – Bag of Words
- CIDEr – Consensus-based Image Description Evaluation
- CNN – Convolutional Neural Network
- CRF – Conditional Random Fields
- CS – Computer Science
- CV – Computer Vision
- DAG – Directed Acyclic Graph
- DL – Deep Learning
- DNN -Deep Neural Network
- DVS – Descriptive Video Services
- EM – Expectation Maximization
- FCN – Fully Convolutional Network

- GRU – Gated Recurrent Unit
- GSU – Georgia State University
- HLF – High Level Features
- HMM – Hidden Markov Models
- HMVC – Hierarchical and Multimodal Video Captioning
- HOG – Histogram of Oriented Gradient
- HOHA – HOllywood Human Actions dataset
- IP – Image Processing
- LSTM – Long Short-Term Memory
- LSTM-E – Long Short-Term Memory with Visual Semantic Embedding
- MBH – Motion Boundary Histograms
- METEOR – Metric for Evaluation of Translation with Explicit Ordering
- MIMLL – Multi Instance Multi Label Learning
- ML – Machine Learning
- MSVD – Microsoft Video Description dataset
- MSR-VTT - Microsoft Research-Video to Text
- MT – Machine Translation
- MVAD – Montreal Video Annotation dataset
- NLG – Natural Language Generation
- NLI – Natural Language Inference
- NLP – Natural Language Processing
- NMT – Neural Machine Translation
- NN – Neural Network

- POS- Parts of Speech
- PReLU – Parametric Rectified Linear Units
- ReLU – Rectified Linear Units
- RNN – Recurrent Neural Network
- ROI – Region of Interest
- ROUGE – Recall Oriented Understudy for Gisting Evaluation
- SIFT – Scale Invariant Feature Transform
- SPICE – Semantic Propositional Image Captioning Evaluation
- SR – Semantic Representation
- SURF – Speeded Up Robust Features
- SVM – Support Vector Machine
- SVO – Subject, Verb, Object
- Tanh – Hyperbolic Tangent Function
- VEML – Video Event Markup Language
- WMD – Word Movers Distance

1 INTRODUCTION

This chapter presents a brief introduction into different video captioning techniques, along with the motivation and challenges associated with this difficult task. Further, it also gives an insight into the complex interactions between the Computer Vision (CV) and Natural Language Processing (NLP) modules and a quick appraisal of the content organization of this dissertation into different chapters. This chapter also briefly presents the motivation and challenges with respect to the single hand gesture classification task of the South Indian dance form - *Bharatanatyam.*

1.1 Overview

Recent reports indicate that videos, with its usage reported to be well over 65 percent of search results, dominate among the different forms of multimedia digital content on the internet [1]. Captioning a video is very useful as it helps in reaching out content to a larger audience, especially to those viewers who are non-native speakers of the language or for those people who are deaf or hard of hearing. Further, captions often provide a much better experience to an ordinary view of the videos. Recent empirical studies prove that captioning of videos improves attention span, comprehensive power and memory retention of the targeted audience [2]. However, despite sharing similar set of methods rooted in Artificial Intelligence (AI) and Machine Learning (ML), very little work could be reviewed that showed an appreciable interaction between the researchers in the fields of Natural Language Processing (NLP) and Computer Vision (CV). However, the scenario is fast changing in recent years with increased interest in image/video captioning and tagging. This resulted in the requirement of improved linguistic as well as visual information skills and enhanced cooperation between the experts in NLP as well as CV. Recent advances in deep learning architectures with respect to various domains (like speech, image, etc.), have also contributed significantly in enhancing this interdisciplinary cooperation. This has led to effective exploitation of the different multimodal cues, abundant in any image or video data packets, for robust feature representations. The ability to generate sentences or descriptions in natural language for a realistic video is the crucial prerequisite for achieving improved machine intelligence with wide ranging applications in the field of video retrieval, blind navigation etc. [3]

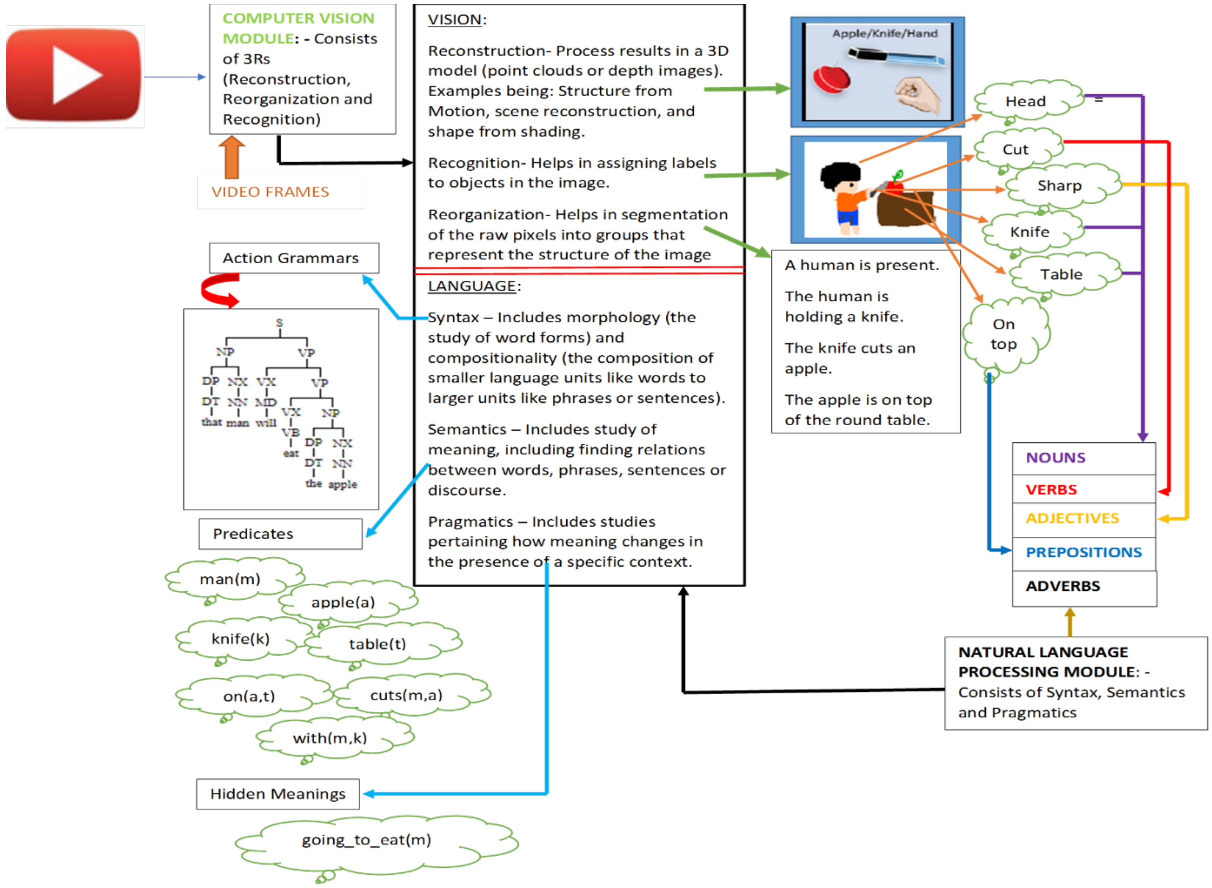
In this new language-vision community, the task of captioning videos irrespective of their domain or duration has emerged as a key but complex vision-language task. Unlike in the case of ordinary image captioning tasks, this task requires in depth analysis of actions in the temporal direction in addition to the routine analysis of simple objects and actors. In short, the video captioning task involves the following:

- Selecting an input video of any domain of interest (example- a cooking demonstration video),
- Analyzing the visual content of its frames and, ultimately
- Describing the contents by generating a sentence/sentences that verbalizes the salient aspects and events in the given video.

In an ideal case, the most informative part in the video should be encapsulated in the generated output text. The irony of the task in hand is, it requires one to describe the objects, their attributes and other features of the scene. For instance, the scene setting could be indoor or outdoor, that are readily visible to the naked eye but also requires one to anticipate certain future events and describe them in the generated sentence Those tasks which cannot be seen immediately but can be inferred. For example, the task also considers verbalizing people, objects and future events that are not seen in the frames of the video but would follow consequentially. Thus, in short, one needs to provide background/contextual information regarding the video that may not be explicit in it [3].

Visual recognition and description, though easy for humans to perform, are still very difficult and daunting tasks for computers to perform. In fact, a plethora of challenges from the vision as well as language generation perspectives will crop up during video captioning. One has to break down the problem into simpler terms. For example, a video clip might consist of many frames which are arranged sequentially. Each of the frame can be treated as a still 2D image which are in turn made up of millions of pixels. A computer must transform these low-level intensity values of the pixels into a high-level semantic concept like a cat or any other object, so that successful recognition is possible.

The recognition and classification of the object is in turn, dependent on various external factors such as the lighting, brightness, direction of the angle pose of the object, etc. In fact, certain characteristics of an object could be similar to the characteristics of a variety of different objects resulting is what is known as background clutter. The description/caption associated with a video is generally represented as a vector. It will be represented to the computer as a sequence of integers indicating the index of each word in the vocabulary. Thus, the whole task of detecting the objects of interest in a sequence of frames and annotating them with words that appropriately describe them is a complex task. It often requires a tedious pattern recognition process of identifying salient subsets of a grid, each with a few million brightness values and annotating them with the sequences of the corresponding integers. Moreover, at times, it might become necessary to detect and describe complex high-level concepts which cannot be directly seen in the visual scene but needs to be inferred. A typical example is a man being mobbed by a crowd on the street. In order to achieve such a description of this visual scene, the system would have to recognize that there are multiple people in the scene, analyze the poses of the detected multiple participants along with their facial expressions spatial and temporal arrangements, etc.



COMPUTER VISION MODULE: - Consists of 3Rs (Reconstruction, Reorganization and Recognition)

VISION:

Reconstruction- Process results in a 3D model (point clouds or depth images). Examples being: Structure from Motion, scene reconstruction, and shape from shading.

Recognition- Helps in assigning labels to objects in the image.

Reorganization- Helps in segmentation of the raw pixels into groups that represent the structure of the image

LANGUAGE:

Syntax – Includes morphology (the study of word forms) and compositionality (the composition of smaller language units like words to larger units like phrases or sentences).

Semantics – Includes study of meaning, including finding relations between words, phrases, sentences or discourse.

Pragmatics – Includes studies pertaining how meaning changes in the presence of a specific context.

A human is present.

The human is holding a knife.

The knife cuts an apple.

The apple is on top of the round table.

NATURAL LANGUAGE PROCESSING MODULE: - Consists of Syntax, Semantics and Pragmatics

Figure 1.1- The relationship between computer vision and natural language processing modules for various tasks.

Human perception is widely dominated by the visual modality for acquiring information by dedicating about 30 percent of the human brain for visual processing alone [1]. Computer Vision can be summarized by concept of 3Rs, i.e. Reconstruction, Recognition and Reorganization, and is generally viewed as a fact-finding technique from the available visual data cluster (i.e. images or video frames). The resulting output of this rather complex task will provide valuable information for other related tasks. For example, for the task of face recognition and detection, the output of the reconstruction tasks involve essentially 3D faces which can provide important and crucial information to successfully aid the recognition task. *Vice versa*, the outputs of the recognition task can be taken in as prior knowledge to aid the reconstruction task in creating an object specific 3D model. The reorganization task deals with lower level features that are interpreted as parts of the texture, color, etc., which build up to a higher-level vision. But reorganization task does not really refer to any specific object in the scene that can be translated into words. The 3Rs of CV are connected to language with the help of semantic information to make sense of the scene depicted by properly interpreting the objects, actions, events and relations depicted in the scene. The connection and interaction between the visual elements of the CV module and the language elements of the NLP module are illustrated in Figure 1.1. Hitherto, most of the video captioning techniques reported in literature often treat them as simple translation problems. Whenever translation from a source language (say English) to a target language (say French) happens, the exact meaning is sometimes lost. Thus, during translation of the low-level pixels on contours of an image/video frame to high level description in word labels (as in classification) or sentences (as in captioning), a wide gap in meaning might occur. This needs to be addressed as a semantic gap problem. The bridge between the visual data and language is closed upon by building words and phrases from the visual data to language data and is generally termed as 'bridging the semantic gap' [4]. The language and reasoning module generally consist of 4 types of semantics, namely:

(1) Lexical semantics - deals with various parts of speech tags like nouns, adjectives, verbs etc.

(2) Compositional semantics - dealing with parsing and grammars, for instance, building a syntax tree.

(3) Formal semantics - deals with generating the predicates.

(4) Distributional semantics which deals with latent variable (as seen in word2vec, embeddings and deep learning) [5].

From a CV point of view one can argue that good understanding of the video is one of the essential requirements before a good description of it is generated. However, a good understanding of the video alone will not suffice for guaranteeing the production of a good description *i.e.*, a good understanding of the video is a necessary but not sufficient condition for generating a complete description. For example, if a simple case of image captioning- using various state-of-the-art detectors to the input image to localize the objects is considered [6], [7], determination of its attributes [8], [9] , [10], computation of the scene properties [11], [12] and in the final stage, recognition of human-object interactions alone will not produce a good image description [3], [13], [14]. Instead a long, unstructured list of labels, invariably detector outputs is obtained, which would be insufficient for an appropriate image description. A formal linguistic model of syntax will be required to ensure that the output text generated is grammatically correct i.e., an introduction of a language model (LM) is required. It helps in creating a comprehensive but concise description that focuses only on objects important to the scene. This helps the output formally correct and thereby, grammatically well-formed sentences that match the final image description are obtained. This is where the NLP point of view gets introduced into the captioning problem (i.e., how to serialize the high-level concepts discovered in the video clip into fluent text). The NLP community views generating natural language sentences as a Natural Language Generation (NLG) problem (NLG is a subfield of NLP). The NLG problem can simply be defined as a task that deals with mapping a non-linguistic, internal computer representation of information into a linguistic representation (natural language). The result would be mostly in the form of a readily readable text either in English or any other human language (sentence/sentences describing the video). From a technical viewpoint, all NLG systems perform the following three tasks:

(1) Content determination and text planning

(2) Sentence planning

(3) Realization (building natural language generation systems).

As the title suggests, the major tasks under step (1) above are determining and marking out the important information that needs to be communicated to the user and text planning *i.e.,* how to structure the information in hand. Step (2), that is the sentence planning is very important to make the final text output read more fluently and make it appear to be closer to text written by human (makes it easily readable) rather than machine (does not follow rules of language). This phase also decides how the information gathered by the first step will be split among individual sentences and various paragraphs and what type of cohesive structures like pronouns, proverbs, etc., should be added to the text to make it flow fluently but at the same time without changing the actual information content therein. Lastly, in step (3), comes the important realization step. The main task of a realizer is to generate individual sentences in a grammatically correct form, ensuring correct usage of the rules of English language (numerous linguistic formalisms and theories can also be incorporated here) [15], [16].

Thus, in short, natural language description from videos not only requires a good understanding of the input video but also requires a sophisticated natural language generation system, making it an interesting problem to be tackled jointly by both the CV and NLP communities. The research problem of video caption generation is primarily inspired by the recent advances in machine translation. At the core of the video captioning problem is the ability to correctly and precisely identify, recognize and classify the set activities in the video (here the activity recognition becomes a sub-problem in video captioning). The reason why recognizing activities in a video is a daunting and challenging task is because of the primary nature, complexity and vast diversity of these in an input video. Consider an example where any of the following four scenarios occur [17], [18]:

(1) The set of activities in a video could be concurrent in nature (concurrent or overlapping activities) i.e. several activities could be occurring at the same timeline

(2) They could be interleaved i.e. activity A is happening at a certain time line, activity A is paused for a certain time while activity B is happening for another time period, then activity B is paused and activity A resumes for the remainder duration of the video

(3) There could also be a situation where direct interpretation of a frame or set of frames (indicating a activity) is difficult and/or ambiguous as the said frames could be dependent on the situation at hand only. For instance, an activity like open refrigerator can be interpreted in different ways depending upon the situation. It could be linked to several activities like cooking or cleaning thus making the interpretation little tricky

(4) If an activity has the involvement of multiple residents/actors i.e. all the activities being performed by the actors in parallel need to be recognized including those activities that the actors perform together.

At present, the world seems to be moving towards empowering machine or in other words edging towards achieving machine intelligence so that in the near future, an almost human like experience while interacting with machines (experience closer to that with that of human) can be achieved. The ability to generate natural sentences describing realistic videos is crucial in achieving machine intelligence. Thus, automatic video description is one such task which has found its rendering in various applications like human-robot interaction, automatic video subtitling and video surveillance.

Motivation. The main motivation behind the video description task is three-fold:

(1) Reaching out content to larger audience/non-native speakers

(2) It enables people who are deaf or hard of hearing to have an experience by generating verbal descriptions of surroundings. Additionally, it can become a helping tool for the visually impaired by generating verbal descriptions of surroundings through speech synthesis, for instance- automatically generating and reading out film descriptions. A good application would be to read out what the sign language interpretation would mean in various video clips (example-news clips)

(3) It improves the overall experience of the audience while watching videos. In short, captioned videos helps improve the attention span, comprehensive power and memory retention of the targeted audience, thus providing a better experience.

The video description task is undertaken here in an effort to:

a) Study the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) architectures in depth

b) Provide an alternate narrative to the traditional encoder-decoder pipeline or encoder-decoder with attention mechanism pipeline

c) To develop novel architectural framework with the use of simpler, lesser memory consuming, efficient shallow neural networks.

This framework should be able to address the video description tasks that can scale up performance to provide stable scores in terms of METEOR (Metric for Evaluation of Translation with Explicit Ordering) and CIDEr (Consensus based Image Description evaluation) evaluation metrics for bigger datasets. Furthermore, an effort is made to achieve higher hand gesture recognition accuracy through developing various models that identify the various gestures in the famous South Indian classical dance form of *Bharatanatyam* and classify them against their true labels. While visual recognition, encompassing vast areas of image/video recognition, detection, annotation/labelling, etc., continue to pause serious challenges to vision experts, the study with respect to hand gesture identification constitutes a small but significant step in facile implementation of the e-learning techniques. This is helpful for visual recognition of the hand gestures used in the classical dance form mentioned above.

The strong motivation behind this work being the huge shortcomings of the previous literature studies in the domain (hand gesture classification) and inadequate focus on deep neural architectures. Previous literature studies in hand gesture identification and classification made use of traditional image processing techniques to extract hand crafted features, thus restricting these classification techniques to work on smaller datasets. Most of these works focused on a very small subset of hand gestures, which were collected under controlled environment setting.

Research Objective. There were three objectives to this dissertation study:

(1) To provide a detailed appraisal of existing video description models, highlighting both their advantages as well as disadvantages and summarizing the benchmark datasets and evaluation metrics of these literature works.

(2) To demonstrate the effectiveness of CNNs in classifying single hand gestures in Bharatanatyam dance form by:

   a. Developing own novel dataset covering 27 out of the 28 single hand gestures.

   b. Developing novel CNN architecture models for classifying the hand gestures.

c. Studying the effects of hyper parameter optimization through manual variation and GridSearch algorithm.

d. Studying the effects of transfer learning in depth and what is the tradeoff between the scale of the dataset and the domain similarity of pre-training and training dataset.

e. To introduce a novel double transfer learning technique that illustrates the aforementioned tradeoff and greatly improves the classification performance of the CNN model.

(3) To develop a novel architectural framework for video description tasks that illustrates an alternate narrative and makes use of simpler, shallower neural networks to achieve the same thereby generating stable models whose performance is not jeopardized with the scale of the dataset.

1.2 Organization

The remainder of the dissertation is organized in six chapters. **Chapter 2** presents a background study related to deep architectures used for vision and language related tasks. This chapter describes the various deep learning architectures, which includes CNNs (AlexNet, GoogleNet, ResNet architectures) and RNNs (vanilla RNN, LSTMs and GRUs), it also mentions the conversely different methods of hand-crafted feature selection in videos, other than using a neural network. **Chapter 3** consists of a comprehensive review of description methods for video, the various datasets used in recent and older times to caption videos and the various evaluation metrics used to evaluate the quality of the captions generated by every captioning model. **Chapter 4** discusses our methodology, i.e. the deep learning approaches and architectures used for video description tasks panning two very different datasets MSVD and MSR-VTT. **Chapter 5** consists of detailed study on exploring convolutional neural networks and subsequent effects of single and double transfer learning for classification of *Bharatanatyam* single hand gestures. **Chapter 6** presents the directions for future work and finally, **Chapter 7** concludes the dissertation and enlists and highlights the contribution made in this dissertation.

2 BACKGROUND STUDY

This chapter provides the necessary technical background about Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), two powerful, yet distinct architectures, used extensively in this study. CNN and RNN architectures are data driven approaches. However, the CNN is relatively more efficient with spatial image/video data (1D or 2D or 3D), absence of temporal information (true for 3D data like videos), while the RNN architectures are networks with some inbuilt memory and work wonderfully for NLP applications. Besides, they are also very useful to decipher the underlying temporal information in video data.

In machine learning, normally one collects data and also trains a model with it (*i.e.,* make the model fit on the training dataset). Here, the trained model is used to predict responses for a second portion of the dataset, also called the validation set (mostly a part of the training dataset which is set aside).The validated model is then used for making predictions on new unseen data, also known as the test data, which provides an unbiased evaluation of the final model fit on the training dataset. Deep learning is a machine learning technique that makes use of neural networks having multiple hidden layers and shared parameters (weights). In this technique, feature engineering is automatically done using different algorithms which helps in extracting useful patterns from data. This makes the further classification tasks much easier for these developed models. The idea of deep learning is based on hierarchical feature learning, *i.e.,* extracting multiple layers of non-linear features and subsequently passing them to a classifier that combines all the relevant features for making good predictions. This essentially stacks up deep hierarchies of non-linear features by learning complex layers from the many layers in a deep neural network. It is difficult to learn complex features from only a few layers. An example to illustrate this concept is shown in Figure 2.1. Through appropriate use of a convolutional neural network (ConvNet/CNN) good features in images can be determined by passing the image through a set of convolutional layers. This in turn forms a hierarchy of nonlinear features that grow in complexity as the image flows from one convolutional layer to the other (for instance, blobs, edges → noses, eyes, cheeks → faces). The final layer of the ConvNet makes use of all these features for classification or regression [19]. The idea of simulating the neocortex's large array of neurons in an artificial neural network (ANN) is not new. However, in the past it has often yielded rather disappointing results than breakthroughs. But in recent times due to a large availability of labeled data (especially for tasks like the driverless car development) and substantial computing power in the form of high-performance GPUs, training of such huge neural networks are being achieved within a of few hours instead of few weeks. Thus, off late, deep learning is getting a lot of attention as a successful tool for achieving results which were not possible hitherto [20].

Visual recognition using deep learning architectures depend upon proper identification of the objects and participants in a video through classification of the extracted features into a fixed number of hard coded visual categories. It is especially true for those works using a 2D CNN for modeling the visual recognition task as the softmax classifier layer (ImageNet dataset). The softmax classifier is normally, a generalized binary logistic regression classifier of objects, spans multiple classes, i.e. has about 1000 fixed, manually picked classes in it. But for real examples in practice, the actual number of classes extracted from videos/images will be several times more, rendering the number of classes in ImageNet insufficient for modeling real time videos/images [21].

2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs, or ConvNets) [22] are neural network architectures, used extensively for understanding images as they perform multiple related tasks such as image classification, segmentation, object recognition in images, etc. They are specifically designed for handling data with some spatial topology (e.g. images, videos, sound spectrograms in speech processing, character sequences in text, or 3D voxel data) where they have achieved state-of-the art results, thus contributing to an upsurge in its usage in large-scale video classification and captioning tasks [21].

Convolution is the main operation performed by CNNs, which is a mathematical operation that describes a rule on how to mix or convolve two signals (for instance, mother signal like 256 X 256 image with a small kernel 3X3) as illustrated by Figure 2.2. The convolutional layer addresses the issue of overfitting by reducing the number of parameters used through a parameter sharing scheme which ensure that all neighboring neurons in one activation map use the same weights. This leads to a large reduction in the number of parameters in each of the subsequent convolutional layers [21].

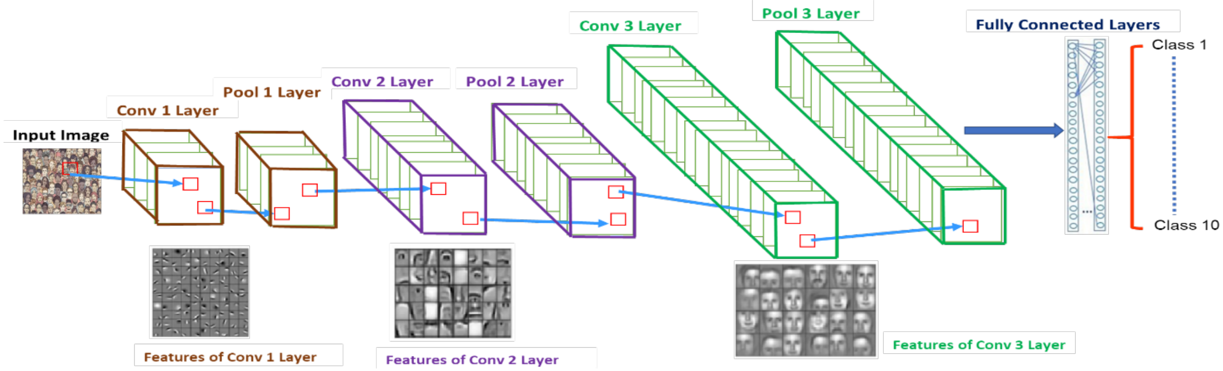
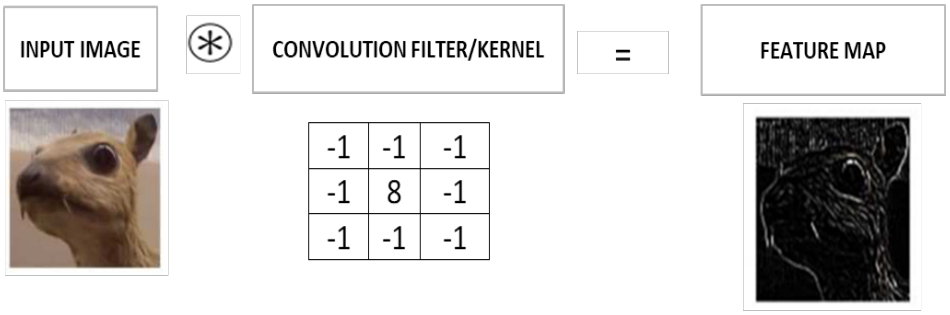


Figure 2.1- The hierarchical features obtained from a deep learning architecture of three convolution layers. The working is: each feature is a filter, which filters the input image for that feature (say a nose). If the feature is found, the responsible unit or units generate large activations, which can be picked up by the later classifier stages as a good indicator that the class is present. Figure modified from [19]

Pooling operations, generally Max-pool or Mean-pool are also important in a CNN. These operations often are referred to as fixed subsampling transformations as it is focused on reducing fixed patches of inputs to a single output value. Normal convolutional layers in the CNN systematically apply the filters to the input image to create a feature map output that summarizes the presence of various features within the input image. A major limitation here is that the feature maps generated record the precise position of the features in the input. As a result, even a small variance in position due to various operations like re-cropping, rotation, shifting, and other minor changes to the input image, would result in a completely different feature map. A simple solution to this problem is achieved through a signal processing concept known as ‘down sampling’. Here, a lower resolution version of an input signal is also created that still maintains the large or important structural elements. The pooling operation works as a robust down sampling solution. Thus, pooling operations provide basic invariance to rotations and translations making detection of an object - even if slightly translated to a corner of the image rather than the preferred center - possible, as the pooling operation funnels the information into the right place for the convolutional filters to detect that object. Moreover, pooling leads to slimming down of the information that needs to be saved, thus forming networks which fit into the GPU memory. The disadvantage however would be, if the pooling area is too large, then a lot of potentially important information will be thrown away or discarded, thereby directly impacting upon the predictive performance of the network [19].



-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 2.2- Illustration of simple convolution operation where an image is convolved with an edge detector convolution kernel. Figure modified from [19]

In general, ConvNets consist of multiple convolution layers, followed by a pooling layer, a nonlinearity layer and lastly by one or more fully connected (fc) layer(s). The last, fully connected layer computes the logits of different classes just before being fed into a softmax classifier. Figure 2.3 is a simple illustration on how various layers can be interleaved to form an effective ConvNet. Ya LeCunn designed the first ConvNet in combination with backpropagation named LeNet for classification of handwritten digits (MNIST dataset). This neural network by LeCunn was inspired by the idea of neocognitron proposed by Fukushima in the year 1980, thus making the work in [23] a predecessor to today's CNN model. LeNet had multiple layers which could be trained in an end-to-end manner using the back-propagation algorithm. Due to the lack of huge labeled data and high computational power, LeNet failed to perform well for complex vision tasks. At present, several variations of ConvNets are available like: AlexNet, VGGNet, GoogLeNet, ResNet etc, many of which are capable of observing a particular trend and further deepening the neural network architecture by the addition of more convolutional layers. This increased depth enables the network to approximate the target function in a better way by generating appropriate feature representations with higher discriminative power. Many techniques like Maxout [24], [25] and Batch Normalization [26] are introduced to ease the training of such deep networks.

AlexNet [27] had five convolutional layers followed by three fully connected layers making it an 8-layer deep ConvNet. The architecture also introduced two novel components, originally absent in the basic LeNet architecture designed by LeCunn. These novel components were non-linear units called rectified linear units (ReLUs), which helped in speeding up the training process and a dropout for effectively relieving overfitting. The VGGNet [28] architecture had two versions: VGG16 and VGG19 models which were of 16 and 19 layers deep respectively. In general, VGGNet has largely enhanced discriminative power over AlexNet, thus improving the performance of CNNs for visual recognition tasks.

GoogLeNet [29] architecture was a 22-modular layered deep network with 56 convolutional layers and was inspired by the Hebbian principle with multi scale processing.- It is used to achieve greater optimization control for classification and detection tasks through NN architecture, which focuses on carefully designing a subsequent layer based on the learnings of the previous layer. An inception module was introduced into the convolutional layers, made up of 1x1, 3x3 and 5x5 filters. This increased both the depth and width of the neural network while at the same time maintaining an affordable computational cost by drastically reducing the number of parameters as compared to other architectures [30]. The inception module allows multiple convolutions and pooling, while simultaneously filtering the input and concatenates the results, taking advantage of multi-level feature extraction from each input. Additional improved extensions to the work in [29] include BN-InceptionV2, Inception-V3 [31] and Inception-V4 [32].

Prior to GoogLeNet winning the IMAGENET challenge in 2015, the work in [33] was released with an aim at achieving two things: (1) Improving model fitting by releasing an improved variant of ReLU called parametric ReLU (PReLU) and (2) Ability to train deeper architectures in a better manner by creating an initialization method specifically aimed at rectified nonlinearities.

ResNet [34] is one of the latest and advanced architectures that is 152 layers deep. ResNet made use of residual blocks which enabled layers to fit a residual mapping and used short connections to perform identity mapping. The authors of [34] also investigated architectures with 1000 layers on the CIFAR-10 dataset [35]. The work in [36] improved the ResNet model with stochastic depth by training a shorter neural network during the training phase and using a deep neural network for the testing phase, thereby achieving state-of- the-art results on the CIFAR dataset [30].

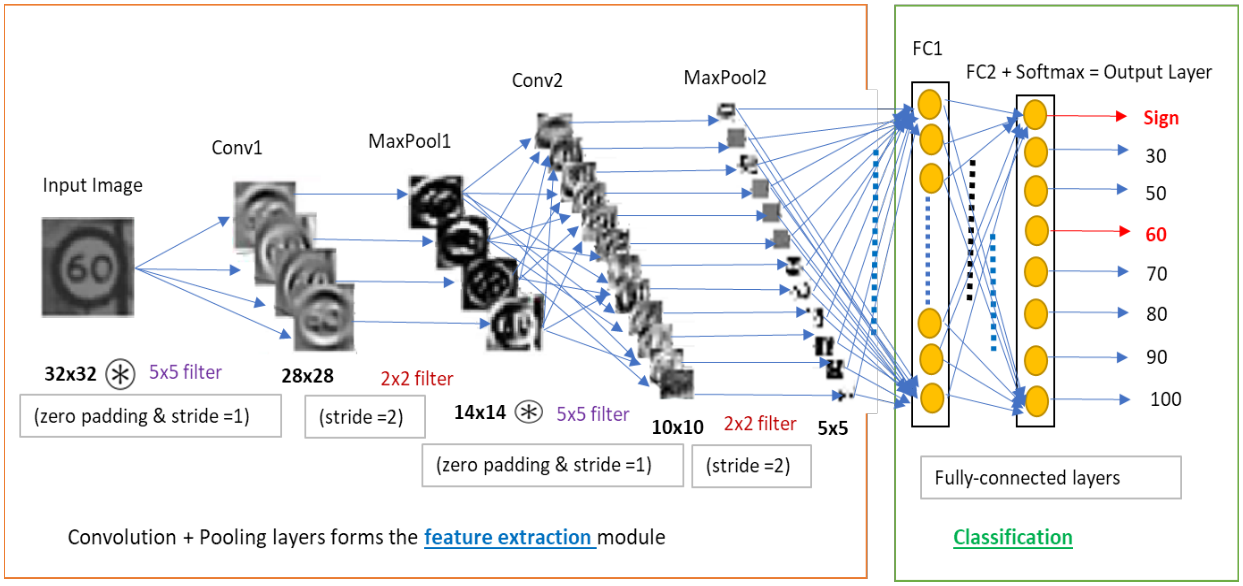


Figure 2.3- Illustration of using both the convolution and pooling layers of a CNN effectively on a traffic sign image. The image is filtered by four 5x5 convolutional kernels to create 4 feature maps, which are subsampled by max pooling. The next convolution layer applies twelve 5x5 convolutional kernels to these subsampled images and again the feature maps are pooled. The final layer is a fully connected layer where all generated features are combined and used in the classifier (essentially logistic regression). Modified from [19]

CNNs can also be applied to video clips for such tasks like action recognition, video classification, captioning, etc. A simple method involves treating the video clips, frame by frame and then applying CNN on the individual frames. This makes it possible for action recognition to take place at the frame level of the video clip (Ref. [37] uses this approach for analyzing the development of embryos). The disadvantage of getting down to the fine grain level of analyzing every frame for detection of an action/set of actions is that, it does not incorporate or encode the temporal information. Such techniques capture the spatial information extracted from the individual frames but fail to capture the motion information from contiguous frames. To capture and encode the motion information of the video clip in a CNN, [38] proposed 3D convolutional neural networks. The convolutional layers in the neural network now perform 3D convolutions capable of capturing features in both the spatial and temporal dimension as opposed to 2D convolutions which can capture features in the spatial dimension in 2D ConvNets. Multiple contiguous frames of the video clips are first stacked one on top of the other to form a cubic structure and then 3D convolution is performed with 3D kernels on the stacked cube for capturing the motion information into the feature set. Though the extension of conventional CNN models by stacking frames makes sense, the performance of these models is far from satisfactory, when compared to that of hand-crafted features [39]. Partly, this could be due to the complex nature of the spatial-temporal patterns in the videos, making it difficult to be captured by deep models with insufficient training data. In addition, the training of CNNs with 3D volumes as input is generally time-consuming [40]. The work in [41] was carried out with the intention to explore better ways to extend the basic CNN architecture to learn spatio-temporal clues in the video clips by comparing several similar architectures on a large video dataset. As reported in [41], the results with respect to the performance of CNN with a single frame or stacked frame as inputs are similar.

2.2 Recurrent Neural Networks (RNNs)

The primary motive behind the rise of recurrent neural networks (which possess some kind of memory, often referred to hidden state) is the strong desire to use previous information/context on the present/current moment as well for attaining maximum impact upon decision making. The necessity to be able to summarize the past and input it into the current state of the model is what made RNNs very important in a lot of sequence processing tasks of the vision and language processing domain. For instance, if one were to use a variant of RNN for language modeling, the neural network needs to be able to predict the next word in the sentence based on the words that came before. Unlike CNN architectures that are feedforward in nature, RNNs thus due to their dependence on past computations, include cyclic connections in the architecture. The emergence of various recurrent structures is connected to the dire need to explore the temporal information's contained in sequential data. The cyclic connection in the RNN architecture enables a memory of previous inputs to persist in its internal state [42]. Just as a language is considered as a sequence of words, a video clip is modeled as a sequence of frames. In language modeling, sentences are made up of sequence of words, with every word encoded as one-hot vector- i.e. a vector of all zeros except for a single one at the index of the word in a fixed vocabulary.

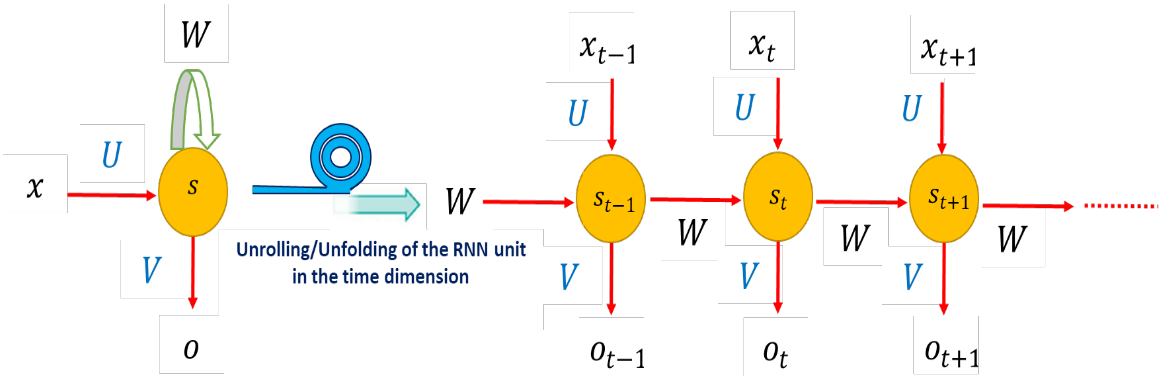


Figure 2.4- Unrolling of a simple recurrent neural network (RNN). Modified from [43]

In Figure 2.4, the term unrolling is used to state the spreading out of the cyclic loop to a full network that is sufficient for processing a full sequence (a complete sentence in the case of language modeling). For example, if one were to model a sentence that has 5 words then the network would be unrolled into a 5-layer neural network. The current hidden state/memory given as U and W are the weight parameters of the neural network and can be calculated by Equation 2.1:

$$s_t = f(Ux_t + Ws_{t-1}) \tag{2.1}$$

Where, x_t represents the input at a given time step *t*, s_t is the hidden state/memory at time step *t*, s_{t-1} is the hidden state at the previous time step, the function *f* usually represents a nonlinearity function, like *tanh* and *ReLU* and o_t is the output at time step *t* [43].

2.2.1 Vanilla/Simple RNNs

Vanilla RNNs are often difficult to train and the performance results achieved after such extensive training process is generally less impressive, making them less popular than LSTMs (another variant of RNN) for tasks like video classification and captioning (LSTMs have a different way of computing the hidden state). The hidden state of a simple RNN can be calculated by Equation 2.2:

$$s_t = \sigma(W_i x_t + U_i s_{t-1} + b_i) \tag{2.2}$$

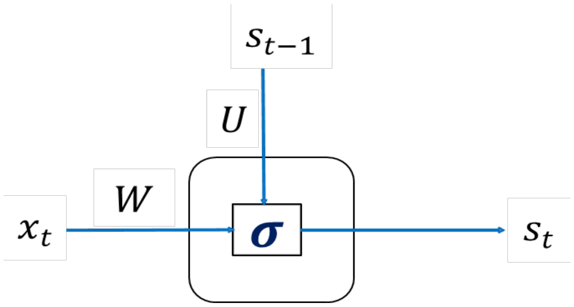


Figure 2.5- A single vanilla RNN cell unit. Here for all recurrent neural networks s_t is equivalent to h_t. Modified from [175]

The disadvantage of the vanilla RNN lies in the fact that they are not very good at capturing long term dependencies like the LSTMs. The vanishing and exploding gradients problem is a constant concern that persists with respect to training the simple RNN architecture as they use gradient based methods like back-propagation. Though not a fundamental problem of neural networks, this problem is caused due to the extra reliance on gradient based learning methods caused by certain activation functions. The vanishing gradient problem refers to the exponential shrinking of gradients magnitude as they are back propagated through time. That is, if a change in the parameter value causes a very little change or no change in the neural networks output, then the gradients of the network's output with respect to the parameters in the early layers become extremely small. Whereas, the exploding gradient problem refers to the explosion of long-term components due to the large increase in the norm of the gradient during training sequences with long-term dependencies [40]. The LSTMs deal with handling the disadvantages of the vanilla RNNs [40]. Vanilla RNNs can enforce a hard constraint over the norm of the gradient thereby reigning in on the exploding gradient problem [44], [45], [46].

2.2.2 Long Short-Term Memory (LSTM)

LSTMs are a special variant of RNNs which were introduced with the idea of handling and learning long term dependencies [47]. The authors [47] in their work introduced the LSTM model which was later refined, popularized and used by several other researchers. The overall structure of LSTM is similar to the vanilla RNN except for a few additional components like non-linear multiplicative gates and a memory cell introduced into its structure. The key feature of an LSTM is its cell state, which runs straight with only some minor linear interactions in the way. The gates (three gates that play an important role to control and protect the cell state) play an important role as regulatory bodies which help add or remove information to the cell state. For instance, if LSTM is used for a language modeling task then the following steps demonstrates the manipulation of the cell state given W_i, U_i, b_i, W_f, U_f, b_f, W_o, U_o, b_o, W_c, U_c and b_c are total number of parameters (weights and bias) for the 3 gates and cell state, i_t, f_t and $o_t$are the input, forget and output gating

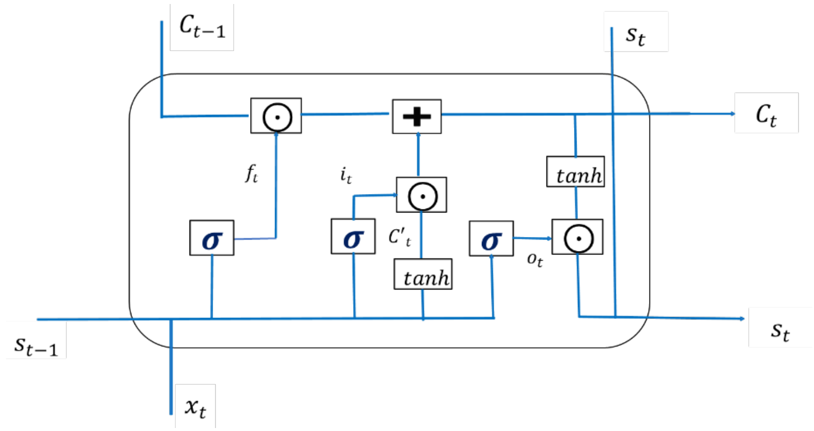


Figure 2.6- Architecture of a single cell unit of the LSTM network. Modified from [175]

networks for the time step *t* and the operator ⊙ denotes element-wise multiplication (Hadamard operator). The process can be explained from Equations 2.3 to e 2.8:

- The LSTM is going to decide what information is irrelevant and can be discarded from the cell state with the help of a sigmoid layer called the forget gate.

$$f_t = \sigma\left(W_f x_t + U_f s_{t-1} + b_f\right) \qquad 2.3$$

- LSTM then decides what new information needs to be stored in the cell state. The input gate layer and a *tanh* layer combined creates an update to the state.

$$i_t = \sigma(W_i x_t + U_i s_{t-1} + b_i) \qquad 2.4$$

$$C'_t = tanh(W_C x_t + U_C s_{t-1} + b_C) \qquad 2.5$$

- Update the old cell state to the new cell state by using the information from the forget and input gates.

$$C_t = f_t \odot C_{t-1} + i_t \odot C'_t \qquad 2.6$$

- The output gate will help output a filtered version of the cell state by first running it through a sigmoid layer which decides what parts of the cell state needs to be outputted. Meanwhile, the cell state is passed through a *tanh* layer which is in turn multiplied by the output of the sigmoid layer to give the final hidden state.

$$o_t = \sigma(W_o x_t + U_o s_{t-1} + b_o) \qquad 2.7$$

$$s_t = o_t \odot \tanh(C_t) \qquad 2.8$$

2.2.3 Gated Recurrent Unit (GRU)

According to [47], there are three variants to GRUs. While LSTMs and GRUs empower successful learning in RNNs, they also have a disadvantage in the form that they result in an increase in parameterization through their gate networks. This is because the gates have their own set of weights that are updated in the learning phase. LSTMs have three distinct gate networks whereas GRUs reduce the gate networks to two i.e. an update gate z_t and a reset gate r_t. Equations 2.9 and 2.10 represent the GRU RNN model:

$$s_t = (1 - z_t) \odot s_{t-1} + z_t \odot s_t' \qquad 2.9$$

$$s_t' = tanh(W_s x_t + U_s(r_t \odot s_{t-1}) + b_s) \qquad 2.10$$

Where the 2 gates z_t and r_t are represented by the following Equations 2.11 and 2.12.

$$z_t = \sigma(W_z x_t + U_z s_{t-1} + b_z) \qquad 2.11$$

$$r_t = \sigma(W_r x_t + U_r s_{t-1} + b_r) \qquad 2.12$$

The 3 variants to the GRUs are called variant1/GRU1, variant2/GRU2 and variant3/GRU3 which were primarily introduce with an aim to reduce the number of parameters use in the GRU model. In GRU1, each gate is computed by using only the previous hidden state and bias. This the gating equations for GRU1 can be represented by Equations 2.13 and 2.14 as:

$$z_t = \sigma(U_z s_{t-1} + b_z) \qquad 2.13$$

$$r_t = \sigma(U_r s_{t-1} + b_r) \qquad 2.14$$

GRU2 removed the bias and computed each gate by making use of only the previous hidden state. The gating equations for GRU2 can thus be represented by Equations 2.15 and 2.16:

$$z_t = \sigma(U_z s_{t-1}) \qquad 2.15$$

$$r_t = \sigma(U_r s_{t-1}) \tag{2.16}$$

GRU3 computed each gate by using only the bias. The gating equations for GRU3 can thus be written by Equations 2.17 and 2.18:

$$z_t = \sigma(b_z) \tag{2.17}$$

$$r_t = \sigma(b_r) \tag{2.18}$$

The total number of parameters reduced by GRU3 was greater than number of parameters reduced by GRU2 which in turn was greater than the number of parameters reduced by GRU1 (i.e. *parameters reduced(GRU3) > parameters reduced(GRU2) > parameters reduced(GRU1)*). Figure 2.7 illustrates a single Gated Recurrent cell unit showing all the gates involved.

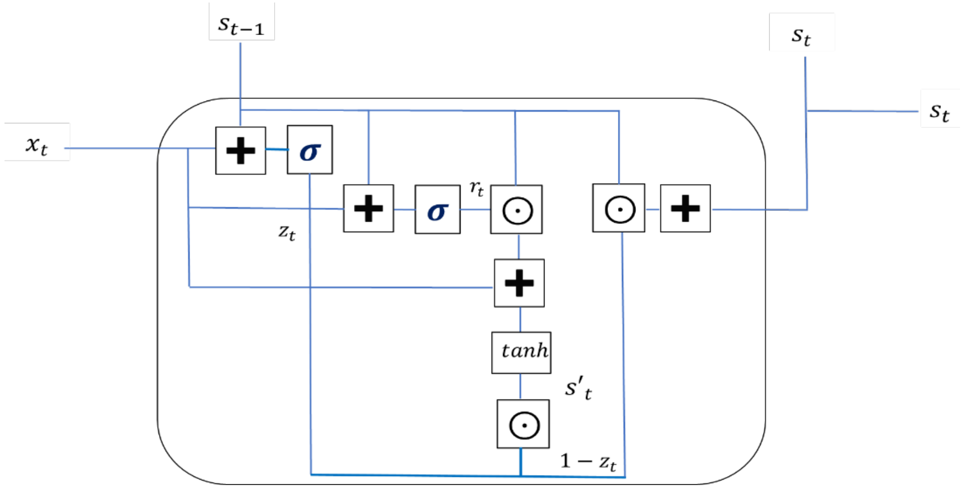


Figure 2.7- A Gated Recurrent Unit. In many literatures h and h̃ are used, where h is same as s and h̃ notation is the same as s' as stated in Equations 2.9 and 2.10 respectively. Modified from [176].

2.2.4 Representing Videos and Descriptions

For a given finite set of video-description pairs (considering supervised learning), a video caption generation model (deep learning models or non-deep learning models) consists of the following basic blocks:

(1) A computer vision technique that deals with the visual content i.e. the object and its properties, scene, participants etc. in every frame of the input video. This can be used to classify the scene, detect the objects, their attributes, the actors/residents present, relationships between the detected contents, recognizing actions etc.

(2) The caption generation module- a NLG technique that deals with the linguistic content associated with the video which can be used to turn the detector outputs into words or phrases. This in turn is combined to produce a natural language description that is syntactically and grammatically correct. While, a video caption retrieval model depends on matching videos and sentences. This kind of matching is important for determining the performance of any retrieval-based video captioning approach. Here, in any given sentence/set of sentences, it is preferred to rank the videos based on how well their frames depict the input sentence/set of sentences. Conversely, given any video, it is preferred to rank the set of sentences retrieved based on how well they describe the video. For both captioning methods to work, it is essential to know the proper representation for videos and their associated sentence/sentences.

Computer vision techniques help in extracting the spatio-temporal key points from each video. An object in a video clip can be defined with the help of spatial interest points and associated descriptors. However, there are certain methods that are used to compute the feature vector from a visual frame, for instance-polygon shape descriptors make use of feature descriptors rather than interest points [48]. A descriptor, which could be as simple as raw pixel values or as complicated as histogram of gradient orientations, is generally a vector describing a patch of an image or single video frame around an interest point/key points/feature. Unlike in the case of images, where only the spatial features need to be extracted so as to detect objects, in videos, it is imperative to extract spatio-temporal interest points to detect objects and their associated motion/movements across the frames. This can be achieved by using different descriptors and/or trajectories.

The Harris3D detector, one of the first spatio-temporal interest point detector introduced by Laptev [50] is an extension of the 2D Harris video detector [51]. The detector computed a spatio-temporal second moment matrix at every video point with the help of a separable Gaussian smoothing function and space time gradients, with the local maxima of the second moment matrix as main pint of interest [52]. Though conceptually very simple, a major drawback of the 3D Harris detector was its poor ability to produce adequate number of interest points. An alternative to the Harris detector was proposed by Dollar [53] which was named the periodic or cuboid detector. This detector used a 2D Gaussian filter for spatially smoothing the video frame and then temporally filtered the smoothed frame with a quadrature pair of one dimensional Gabor filters [54].Various other spatio-temporal detection approaches, based on the determinant of the 3D spatio-temporal Hessian matrix [55] to measure saliency (interestingness of locations in a video frame) by computing over several spatial and temporal scales were also developed subsequently. Interest points were then extracted by selecting the extrema by applying a non-maximum suppression algorithm. Several other feature descriptors and detectors were also developed for usage in video classification, captioning and video content retrieval [56] for successfully recognizing human action in videos. Typical Feature descriptors include several higher order derivatives (local jets): gradient information, optical flow and brightness information [53], [57], [58]; spatio-temporal extensions of image descriptors such as 3D-SIFT [59], HOG3D [60]; extended SURF [61] and local trinary patterns [62], [63]. In addition, many feature detectors had shape-based features: e.g., HOG [63], SIFT [64] and motion dependent features: e.g., optical flow, MBH [65] with high order encodings (Bag of Words, Fischer vectors) as well as trained classifiers: e.g. SVM, decision forests in its repertoire. This was highly useful for predicting actions using appropriate verb [56]. In recent times many deep learning models were proposed for action recognition tasks due to their ability to learn a hierarchy of features by building high-level features from low-level ones. Many CNN models have been proposed for solving action recognition efficiently after AlexNet was found highly successful for image classification tasks. [56], [66], [67], [68]. Architectures like Convolutional RBMs [69], 3D CNNs [56], RNN [68], [70], CNNs [41] and Two-Stream CNNs [71] have also been effective in feature detection and extraction for efficient action recognition. In videos, since motion information is a salient feature, many architectures focus on generating action region proposals for extraction of visual features instead of using simple object detectors. While deep learning architectures like CNNs and RNNs have improved the detection rate of multiple events from video data, task of differentiating human actions from its background is still considered very tedious [71], [72], [56].

Yet another way of tackling the action recognition problem is by treating it as a temporal action localization problem [73-83]. Here, action classifiers are applied densely in a sliding window manner or by using deep action proposals (DAPs) to the frames [84]. For temporal action proposal generation, the number of candidate temporal windows generated can be reduced with the help of dictionary learning [85] or with the help of a recurrent neural architecture [86], By and large, action classifiers work efficiently on a smaller number of temporal windows, discriminating each window into one of the actions of interest [87]. In addition, several additional problems like detecting actions from associated temporal / spatial frames have been addressed successfully [88], [89], [90], [91], [92], [72]. These algorithms primarily focus on spatio-temporal localization of actions. Though these Spatio-temporal algorithms provide more detailed localization information, they also are time consuming with higher computation costs, which make their usage difficult for routine applications requiring faster and efficient processing [87]. The work in [56] explains one of the common techniques used to generate action region proposals illustrated in Figure 2.8, thereby detecting the activity in the video successfully.

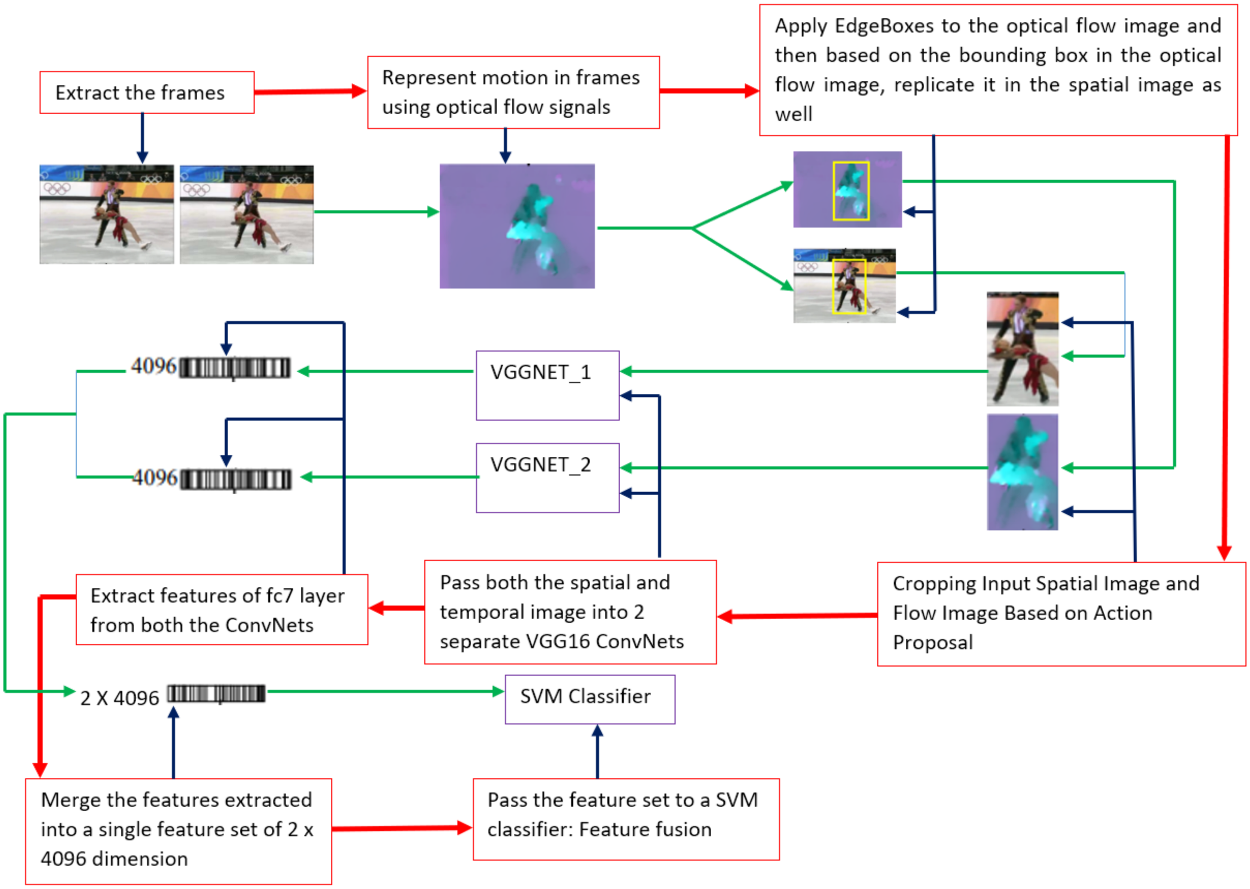

Extract the frames
Represent motion in frames using optical flow signals
Apply EdgeBoxes to the optical flow image and then based on the bounding box in the optical flow image, replicate it in the spatial image as well
4096
VGGNET_1
4096
VGGNET_2
Extract features of fc7 layer from both the ConvNets
Pass both the spatial and temporal image into 2 separate VGG16 ConvNets
Cropping Input Spatial Image and Flow Image Based on Action Proposal
2 X 4096
SVM Classifier
Merge the features extracted into a single feature set of 2 x 4096 dimension
Pass the feature set to a SVM classifier: Feature fusion

Figure 2.8- A common technique for generating action region proposals discussed in [56] for detecting the main activity in the video clip.

3 LITERATURE REVIEW OF VIDEO DESCRIPTION MODELS

In this section the literature work related to the field of video captioning task published during the years 2000 – 2017 is reviewed. There is an attempt to diversify and classify the existing models based on certain parameters like their primary narrative of approaching the problem, the type of captions generated, the learning method employed, etc. Additionally, this section also provides a detailed appraisal of the existing models, highlighting both their advantages as well as disadvantages. Finally, an honest attempt has been made to herein summarize the benchmark datasets used in various video captioning models and the evaluation metrics employed to assess the quality of the resultant video descriptions generated.

3.1 Introduction to Video Captioning

Over time researchers have successfully developed and refined methods to tackle the challenging task of video captioning. Recent methods focus more on modeling the vision aspects as well as language aspects jointly in a supervised setting. Earlier methods in literature [94-99][102] [103][105], however, were more focused on short, activity/ context specific datasets with smaller vocabularies, or limited objects and actions. Recent publications like [114-116], [119-120] have made the use of deep learning architectures very convenient. These methods directly model language, conditioned on the video content and churn out good performance results, primarily due to the availability of large video-sentence pair datasets. The existing video captioning literature can be broadly classified into non-deep learning models and deep learning models i.e. they can simply be divided into techniques which were developed prior to the application of deep learning architectures (like CNNs or different variants of RNNs) or those techniques that make good use of deep learning architectures, mostly in an end to end fashion to produce the desired translation text as an output. The non-deep learning models can further be classified into rule/template models, i.e. such methods primarily design simple heuristics for the identification of objects in the video frames along with a set of rules defined for producing verbs and prepositions. Finally, a sentence is generated by filling predefined templates with the recognized parts of speech (POS). Figure 3.1 depicts the timeline of literature works between the years 2000-2017. From the Figure 3.1, it is clear that not many models were developed during the early 2000's, perhaps due to the absence of strong language models. With the advent of RNNs we see a sudden surge in related literature, post 2012-2013.

Most of the early researchers used the rule/template-based models to align each part with the detected words from the visual content in the video (captured by object recognition), which were later gathered together to form a legitimized sentence in accordance with language constraints of the output text. A major disadvantage in designing such models (like in [99]) is rule engineering, which can very quickly become a tedious and insurmountable task, especially when scaled up. Moreover, rule-based approaches were ineffective while dealing with uncertainty and ambiguity of visual features extracted to facilitate high level semantic analysis, thus limiting flexibility and expandability of such approaches. Another sub-category of the non-deep learning model is the

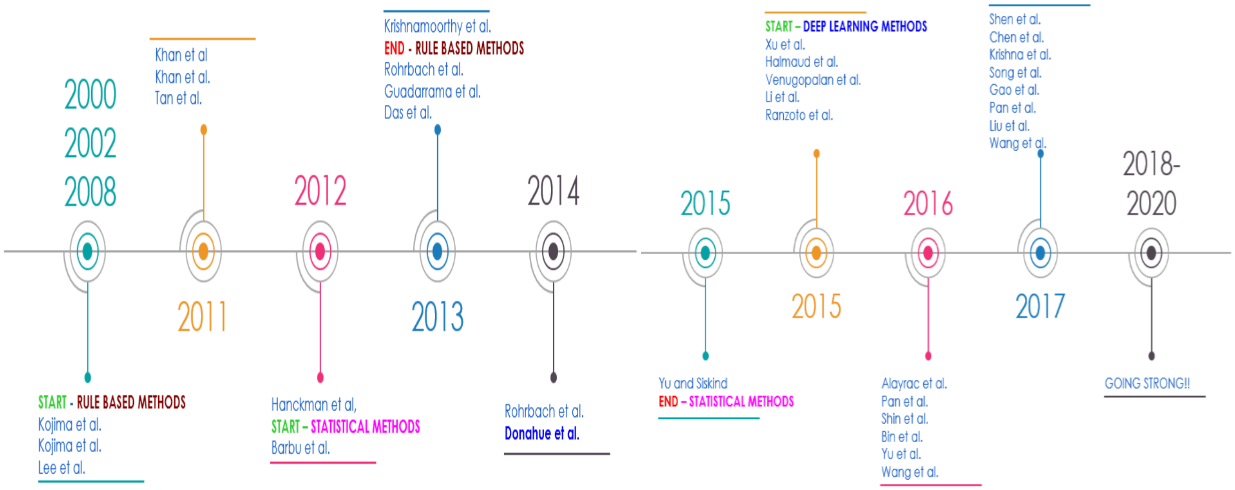


Figure 3.1- A Timeline depicting the literature works for video captioning between the years 2000-2017.

statistical machine translation model (they follow rules of machine translation which translate from one natural language to another, like French to English). The models under this category formulate the given problem into a data driven machine learning problem. They essentially eliminate the disadvantages of rule engineering faced by rule/template-based models, more flexible and easier to discover the underlying structures or events in a given video data. It uses different supervised learning techniques to train statistical models either fully or partly and then use the trained model to predict the output of test data. These methods extract a semantic representation (SR) from the visual content of the input video which is later translated to a natural language description.

Statistical models are of two kinds: generative models and discriminative models. Generative models like HMMs use an expectation maximization (EM) algorithm for model learning and Bayesian inference for decision making based on a given set of observed sequence observations. i.e., generative models focus on computing the joint probability distribution over inputs and outputs. The main disadvantage with respect to generative models is that they are not very effective in modeling complex features or are incapable to capture the relationships in an effective manner especially when the feature set is complex and of a higher dimensionality. Modeling such complex dependencies among inputs could lead to intractable models, while ignoring such complex dependencies altogether could lead to reduced performance of the model [93]. Discriminative models like support vector machines (SVMs), conditional random fields (CRFs) or even neural networks directly compute the posterior probability based on the set of observed sequence and use it for learning and classification purposes, *e.g.,* computation of conditional probability distribution, necessary for classification problems. Several statistical models use the Hidden Markov Model (HMM) or Conditional Random Fields (CRF) as modeling techniques for activity recognition in the video. HMMs are primarily used to model simpler activities while CRFs are used to model complex and unfamiliar activities. Statistical methods deal with slightly larger datasets than compared to those used by rule/template-based models, thus dealing with thousands of lexical entries and longer videos whose duration easily stretch to hours. As a result, the captioning task becomes more challenging due to the complexity and diverse nature of the videos in the dataset. These methods pave way to deal with larger datasets but have a disadvantage in terms of the final generation performance i.e. the caption generation performance of these methods is generally very low on large scale datasets like MSRVTT, ActivityNet Captions, MSVD etc.

Under deep learning models, all the sub-categories follow a data driven approach and can alternatively be called as sequence learning models. As the name suggests, these models make use of deep neural networks like a CNN or a RNN or both, to achieve their goal. Most of the methods in this category use RNNs and view the task as a machine translation problem i.e. translating visual sequences to natural language. This has been largely possible due to the recent advances made by the natural language processing community in the domain of neural machine translation (NMT). The sub-categories of the deep learning models are based on how these deep neural networks are trained: *i.e.*, whether they belong to supervised learning, weakly supervised learning or unsupervised feature learning techniques. The literature on video captioning can also be further sub classified based on the kind of captioning techniques used. For instance, some video captioning techniques belong to the caption generation based on visual input category. Such models mostly follow a general pipeline architecture by first predicting the most likely meaning of a given video clip by analyzing and going through its visual content. Subsequently, the focus is on generating a sentence/sentences reflecting the meaning. The general architecture for such a method usually flows in the following manner:

(1) Initially any computer vision technique is applied to the video clip to achieve the following tasks: classify the scene, detect objects, predict their attributes / relationships with one another and also recognize the main action happening in the scene.

(2) Next, the features extracted by the above computer vision techniques are then translated to natural language *i.e.,* the detector outputs are translated into words or phrases which in turn are combined to form a natural language sentence(s) using the natural language generation techniques.

(3) Further; the work will include caption retrieval in visual or multi modal space. Under this, the system focusses on retrieving candidate videos based on similarity with the query video clip. These systems basically exploit the similarity in visual space to transfer the captions to the query clip. The main disadvantage of the retrieval models is that they require huge amount of training data to provide relevant descriptions to the query clips. The main steps followed by such models can be described as follows:

a. Represent the query clip by specific important visual features

b. Retrieve the candidate set of clips from the training set based on the similarity measure in the feature space used

c. In the final stage, focus is to re-rank the description/set of descriptions of the candidate clips retrieved by making use of the visual and/or textual information contained in the retrieval set. Here, one can also alternatively combine fragments of the candidate descriptions according to a defined set of rules or schemes [3].

The retrieval techniques in the multimodal space casts the description retrieval problem as a retrieval problem from a multimodal space i.e. generally mapping the visual and associated textual data into a common embedding space. In general, the retrieval based captioning model require a lot more training data as compared to caption generation models. Video captioning models can also be classified based on the output description generated. While some captioning systems summarize the visual contents and describe the video using one single generic sentence, few others describe every frame of the video or the most important events of the video in multiple sentences or even in a paragraph.

3.2 Non-Deep Learning Models

3.2.1 Rule/Template based models

The rule-based models make use of descriptors to extract interest points/features from the input video. The features/concepts extracted are then classified. The illustration of the hand-crafted features being classified by a rule-based classifier is depicted in Figure 3.2. In the same figure, a brightly painted car is presented with its engine compartment being shown prominently, displaying the car either for auto enthusiasts / potential buyers. The feature extractor should pick up on the headlights, wheels, front grill, doors, and windows. The simple classifier should then put these features together to recognize a car in this image. Lastly, a mapping between the classified concepts to textual descriptions or content recounting is done with manually established ad hoc rules. Rule/template models generally come under the caption generation category of captioning models where language generation is treated as an engineering task.

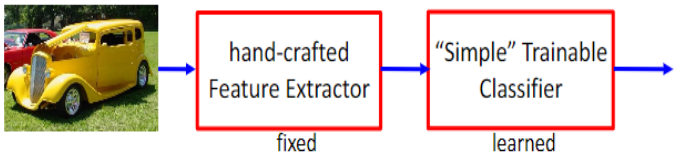


Figure 3.2- Illustration of rule-based methods which makes use of hand-crafted features using descriptors to extract the points of interest.

In the works presented in [94], the captioning system focused on getting compact descriptions as output given complex video contents as input. The captioning system consisted of two major components:

(i) A module that focused on learning both the audio and visual concepts which was applied to 10 second duration clips rather than entire video

(ii) A rule-based textual description generation.

The architecture focused on extracting three types of audio-visual features from the video clips. The visual features extracted from each of the detectors were concatenated together to form a feature set of visual features. The STIP (3D spatial-temporal interest points) feature was used for human action concept classification whereas the SIFT feature was used for scene classification, and mel-frequency cepstral coefficients (MFCC are coefficients that make up the MFC, i.e. the mel-frequency cepstrum, which is a representation of the short-term power spectrum of a sound and is typically used as a feature in audio processing) was used for audio concept classification. The bag-of-word (BOW) representation was applied on the extracted three sets of descriptors to convert them into three fixed dimensional feature vectors (human actions, scene classification, and audio concept) and an SVM classifier was utilized for concept learning for the three types of features. Hierarchical k-means was used to generate the audio and visual vocabulary sets, the size of which were based on empirical evidences from prior studies in [57], [95].

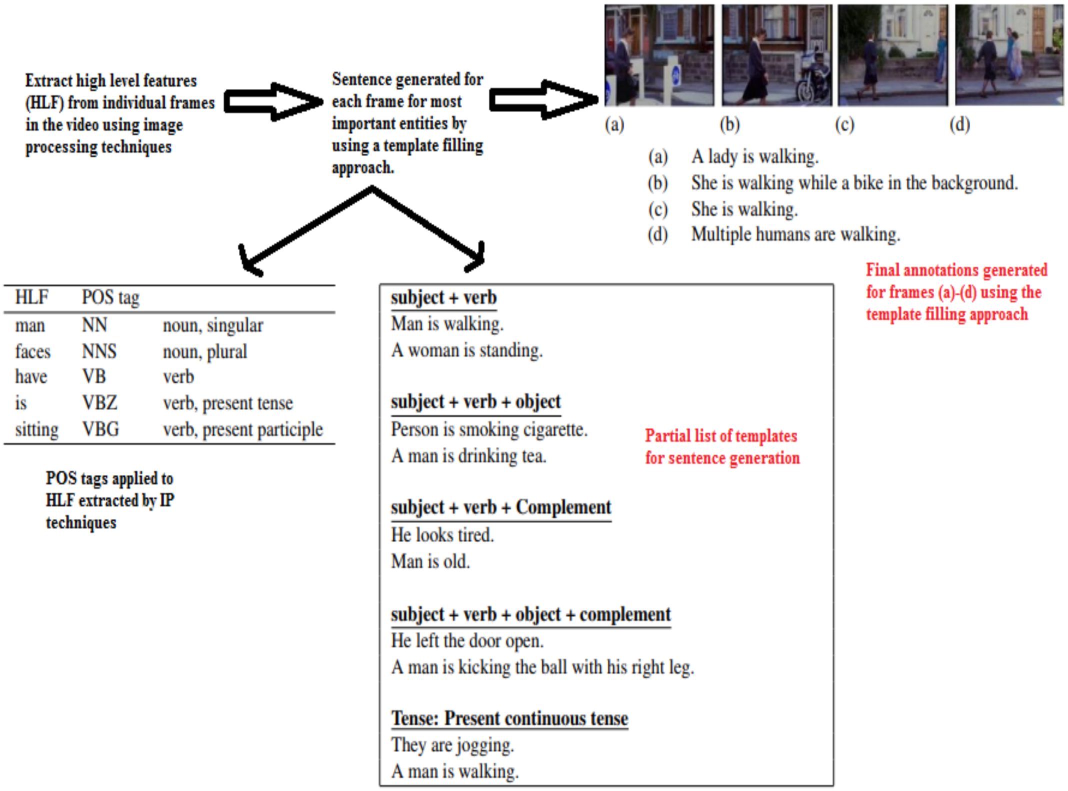
The classification results are later used to generate the textual descriptions, describing the contents of the video. Based on the concepts recognized (human action concepts, scene concepts and audio sound concepts), a set of predefined templates are used by concatenating the subject phrase with the action and scene phrases. The final description generated describing the video is concise and less verbose, as entire video level recounting is achieved by combining all the phrases from the 10 second clip level descriptions and by discarding redundant and duplicate phrases. Internet videos from NIST TRECVID 2010 multimedia event detection (MED) task [https://www.nist.gov/itl/iad/mig/multimedia-event-detection] were used for performing the experiments [94].

The work in [96], proposed a method for generating natural language descriptions of human behaviors appearing in real video sequences. This work assumed that for any activity in a video, humans are the active participants and the description generated were based on the position and orientation of the human head instead of the whole-body posture. The architecture extracted the head region from the human participant in each frame of the video and later estimated the 3-D pose and position of the head using a model-based approach. The head motion trajectory was divided into different segments, with each segment consisting of monotonous movement of the head. Various features such as degrees of changes of pose and position and the relative distances from other objects in the surroundings for each segment were evaluated. From the language modeling end, the most suitable verbs and other syntactic elements pertaining to language were selected. In the final step, the appropriate language description for describing and interpreting the human behavior in the scene is generated by machine translation technology [97]. The disadvantage of this method is that it fails to identify actions where there is sufficient hand or other body region movement. For instance, if one were to analyze the hand movements of a dance form, like *Bharatnatyam*, this model will not be efficient.

In [98], the authors attempt to bridge the semantic gap between visual content and textual descriptions by proposing a framework that describes human related activities from the video by using the concept hierarchies of actions. This work is an improvement on [96] which considered only pose and orientation of the head region for human activity detection. The work in [98], tried to highlight the relevance of interaction of humans with other objects in recognition of human activities by considering various sub-activities of the human body like the direction of the line of sight, hand positions, body posture with relation to other objects simultaneously. For each frame of the video sequence, the body and skin regions of the human participant were extracted. The positions of head and hands were calculated by perspective transformation and the orientation of the head is also considered. The next step involved connecting the position/ posture of the human obtained from the video to an action utilizing domain knowledge like allocation of equipment. From this, we can generate a conceptual description of the action by constructing concept hierarchies of actions for each body parts classified by combination of semantic primitives in a room observing each body part of the participant along with domain knowledge like allocation of equipment in a room, to the position/posture of the human obtained from the video images. Based on the correspondence of the action verb and visual feature extracted from the video, the most appropriate syntactic components like predicate, objects etc., are selected and used to fill into a case frame, which is generally used as a semantic representation of a sentence. The case frames generated for each body part is integrated into a whole expression that describes the main action in the video which is later output as the natural language description for the video based on certain syntactic rules and natural word dictionary.

In [99], a caption generation framework from visual content was proposed. The architecture as illustrated in Figure 3.3, made use of conventional image processing techniques to detect and extract high-level features (HLF) from every frame of the video content (HLFs such as humans objects, their moves and properties) [100]. With human considered to be the most important and interesting feature, the description generated focused primarily on humans as active participants and their activities. The natural language processing module dealt with merging these

HLFs into syntactically and semantically correct textual presentations in compliance with the lexicons extracted (HLPs in the form of entities and actions is used) and grammar of the language, by making use of a template-based approach (implemented using SimpleNLG [101]).


Extract high level features (HLF) from individual frames in the video using image processing techniques
Sentence generated for each frame for most important entities by using a template filling approach.
(a)
(b)
(c)
(d)
Final annotations generated for frames (a)-(d) using the template filling approach
POS tags applied to HLF extracted by IP techniques
Partial list of templates for sentence generation

(a) A lady is walking.
(b) She is walking while a bike in the background.
(c) She is walking.
(d) Multiple humans are walking.

HLF	POS tag	
man	NN	noun, singular
faces	NNS	noun, plural
have	VB	verb
is	VBZ	verb, present tense
sitting	VBG	verb, present participle

subject + verb
Man is walking.
A woman is standing.

subject + verb + object
Person is smoking cigarette.
A man is drinking tea.

subject + verb + Complement
He looks tired.
Man is old.

subject + verb + object + complement
He left the door open.
A man is kicking the ball with his right leg.

Tense: Present continuous tense
They are jogging.
A man is walking.

Figure 3.3- The architecture depicted by the rule-based video description approach in [99].

In [102], [103], a three-component framework, namely an image parsing, event inference and text generation, that focused on performing automatic semantic annotation of visual events (SAVE) was proposed. The first component, the image parser was utilized for scene content extraction. This component made use of bottom-up image analysis using a stochastic attribute image grammar in which a visual vocabulary from pixels, primitives, parts, objects and scenes, play a major role and also define their spatio-temporal or compositional relations with a bottom-up top-down strategy used for inference.

Table 3.1- Tabulation of all Video Captioning methods indicating the nature of the technique and linguistic output.

Ref No	Output	Caption Generation	Caption Retrieval
[94]	Dense/multiple captions	✓	
[96]	Dense caption	✓	
[98]	Dense caption	✓	
[99]	Dense caption	✓	
[102]	Dense caption	✓	
[104]	Single caption	✓	
[105]	Dense caption	✓	
[106]	Single caption	✓	
[107]	Single caption	✓	✓
[108]	Dense caption	✓	
[109]	Dense caption		✓
[112]	Single caption	✓	
[113]	Dense caption	✓	
[115]	Dense caption	✓	✓
[116]	Dense caption	✓	
[119]	Dense caption	✓	✓
[121]	Single caption	✓	
[122]	Single caption	✓	
[14]	Single caption	✓	
[123]	Single caption	✓	
[126]	Single Caption	✓	
[70]	Single caption	✓	
[114]	Single caption	✓	✓
[127]	Dense caption	✓	
[128]	Single caption	✓	
[129]	Single caption	✓	
[130]	Single caption	✓	
[131]	Single caption	✓	✓
[133]	Dense caption	✓	
[134]	Single caption	✓	
[135]	Single caption	✓	
[145]	Dense caption	✓	
[144]	Dense caption	✓	
[140]	Single caption	✓	

The second component, an event inference engine, adopted the Video Event Markup Language (VEML) for semantic representation, followed by a grammar-based approach used for event analysis and detection. The third and final component, the text generation engine, generates a text report using head-driven phrase structure grammar (HPSG). In [104], the model proposed was for generating a simple sentence that incorporated the subject, verb and object (SVO) for describing a short video clip. In the content planning stage of NLG, a combination of object and activity detectors as well as text mined knowledge were used to identify the most likely SVO triplet. In the surface realization stage of NLG, a simple template-based approach was employed to generate candidate sentences for a given SVO triplet, which were then ranked for plausibility and grammaticality by a statistical language model trained on web-scale data to arrive at the best overall description.

In [105], Hanckmann, Schutte, and Burghouts made use of an action classifier and description generator to generate descriptions for a video clip. Action classifier detects the actions in the video and enables them to be used as verbs by the description generator. The description generator finds the objects/people in the scene and make use of the verb provided by an action classifier to generate a sentence, based on appropriate verb, subject, direct / indirect objects.

3.2.2 Statistical models

The models discussed here come under two categories: the first category deals with training statistical models for various lexical entries which helps to eliminate the tedious efforts of rule engineering, where models of different Parts Of Speech (POS) may have a different mathematical expression or training strategy. Nouns, verbs and prepositions are then mosaicked together to yield grammatically and syntactically correct sentences. The second category does not explicitly train word models but instead constructs a structured model (Example- CRF) to formulate the relationship/ interaction among words in a sentence by treating them as latent variables. Finally sentence generation is achieved by inferring the latent labels given to the observed variables in the form of visual features. However, the drawback of such models is the lack of clarity regarding the presence of semantic meaning in the description based on visual concepts or due to the high correlations encoded in the structured model. Most of the captioning frameworks in this section made use of HMMs or CRFs to model their entities.

The work in [106], focused on generating sentential descriptions of a given video by describing the observed action (verb) with the participant object and their properties in the scene i.e., describing aspects like who did what to whom, where and how? (called the 5W's and 1H). The vocabulary set used to generate the sentential descriptions of a video however was very small and consisted of only 118 words out of which the distribution of the words was: 1 coordination, 48 verbs (primary action), 24 nouns, 20 adjectives, 8 prepositions, 4 lexical prepositional phrases, 4 determiners, 3 particles, 3 pronouns, 2 adverbs and 1 auxiliary. Further, the work in [106] used a detection-based tracking approach as illustrated in Figure 3.4, which considered every frame in the input video. For every frame, feature/object detectors were applied for each object class to return a set of candidate detections, which in turn were composed into tracks by the selection of a single candidate detection from each frame, maximizing the temporal coherency of the track. A Kanade-Lucas-Tomasi (KLT) feature tracker was used to augment the set of candidate detections by projecting each detection frames forward. An optimal set of detections that were coherent with the optical flow were selected based on a dynamic programming algorithm, which in turn yielded a set of object tracks for each input video. The tracks obtained were then smoothed over to get a time series of feature vectors for each video to describe the relative and absolute motion of event participants. A body posture codebook of persons detection is created, and the codebook indices of a person's detections were added to the feature vector. Finally, a Hidden Markov Model (HMM) based classifier was employed for designating each video with verb labels (basically verb label was recognition of the primary action in the video) with the roles, the participants/objects played.

The object tracks were processed to produce nouns from object classes, adjectives from object properties, prepositional phrases from spatial relations and adverbs and prepositional adjuncts from the track properties. Together with the verb (the main action recognized in the video) these were then woven into syntactically grammatically correct sentences. The dataset used here was the Year-one (Y1) dataset produced by DARPA which was specially designed for sentential description evaluation.

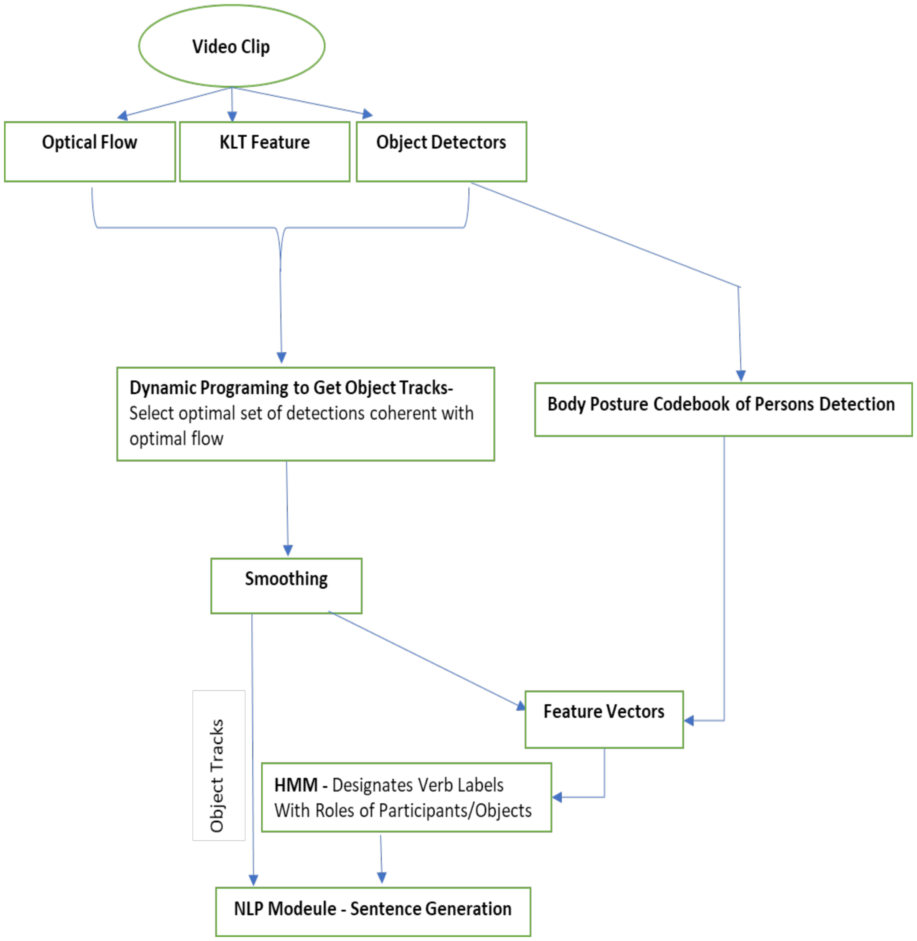

Video Clip
Optical Flow
KLT Feature
Object Detectors
Dynamic Programing to Get Object Tracks-
Select optimal set of detections coherent with optimal flow
Body Posture Codebook of Persons Detection
Smoothing
Object Tracks
Feature Vectors
HMM - Designates Verb Labels With Roles of Participants/Objects
NLP Modeule - Sentence Generation

Figure 3.4- The architecture in [106] makes use of advanced image processing techniques to extract points of interest/high level features from every frame of the video clip. Modified from [106].

In [107], Rohrbach, et al. translates visual content to natural language descriptions by following the same rules that apply for natural language translation from one language to another.

Rohrbach, et al. proposes a 2-step approach. First step is to learn an intermediate semantic representation (SR) of the form < *activity, tool, object, source, target* > using a probabilistic model. The SR is generated from the visual content of the video by assuming that the objects participating in an activity in the video have highly correlated relationships with each other. These relationships are modeled using a conditional random field (CRF) where visual entities are modeled as nodes by observing the video descriptors as unaries. Once the SR representation is obtained, in the next step, it is translated into natural language descriptions following the rules of a translation problem in NLG. For translating SR to natural language descriptions, the author makes use of statistical models instead of rule-based approaches thereby eliminating the need for predefining strict rules for language generation. Instead, the translation model can now directly learn from a parallel corpus of SRs and descriptions.

In [108], Guadarrama et.al focuses on dealing with datasets that have out of domain actions but have very short video clips, unlike the datasets dealt with previous methods which were of a specific domain only, for example, cooking. The authors proposed a novel language driven approach to describe YouTube videos by primarily tackling two issues: (1) There could be multiple ways to describe the same activity in a video and (2) A final description does not always have to be very specific to be deemed particularly useful, it could be generic in difficult cases. The work in [108] seemed path breaking as it did not require training the model using the videos of an exact activity to get the correct labeled text as output. If an accurate prediction could not be reached by the pre-trained model, then it would still return a less specific answer that is plausible from a pragmatic standpoint. Semantic hierarchies help choose an appropriate level of generalization based on the training data. Prior knowledge is learnt from web-scale natural language corpora which when used in conjunction with the semantic hierarchies helps penalize unlikely combinations of actors/actions/objects. As the language modeling module of the architecture, a web-scale language model is used to fill in the novel verbs (if the verbs do not appear in the training set also called zero shot verb recognition), which works intuitively. The best predicted subject/verb/object triple complete with surface realization is selected as a grammatical sentence by this end to end generation system.

The work in [109], proposes a hybrid system that combine top-down and bottom-up captioning approaches and consists of 3 components: a low level multimodal latent topic model for initial keyword annotation, a middle level of concept detectors and a high-level module to produce final lingual descriptions. The test video clip is processed in 3 ways:

(1) By the bottom-up strategy where low level video features predict keywords by making use of multimodal latent topic models to find a proposal distribution over some training vocabulary of text words [110], [111] and selecting the most probable keywords as potential subjects, objects as well as verbs through a natural language dependent grammar and parts of speech tagging

(2) By a top down strategy, where concepts are detected and stitched together as one moves from frame to frame of the video. Further, it is translated to lingual descriptions through a tripartite graph template

(3) By relating the predicted keywords with detected concepts to produce a set of ranked, well-formed natural language descriptions, high level semantic verification can be achieved.

In [111], the authors design a technique which makes use of a model that is trained with positive and negative sentential labels to generate textual description for a test video using trained models. The model first attempts to learn the correspondence between words and the associated video regions by training each of its word models with its corresponding region(s). The model exploits language semantics and thus can implicitly annotate the video by itself for training different types of words and extracting useful information from PLs as well as NLs.

In [113], the work mainly focused on producing a single sentence generation and producing coherent multi-sentence descriptions of a complex video at a variable level of detail. The framework proposed follows a two-step approach: the first step focuses on predicting a semantic representation (SR) from the video which then is translated to generate natural language descriptions in the last step. Since multiple sentences are generated describing different activities, the language model chooses from a probabilistic SR rather than a single MAP estimate thereby enforcing and improving intra-sentence consistency. Each video in the dataset is decomposed into a set of snippets, based on temporal segmentation which can be represented by video descriptors and a single sentence description. The SR which is a tuple of activity and participating objects/locations is built by modeling the relationships in a CRF. Moreover, the work in [113] focuses on recognition of objects in the scene by hand centric approaches and also on robust generation of sentences using a word lattice (Is a directed acyclic graph (DAG)). The SMT does not directly produce a cohesive description; instead it produces a list of sentences rather than coherent text. Post processing on this list of sentences obtained by using a set of domain independent rules to improve the cohesiveness (linguistic measure on how sentences relate to each other on a surface level) is enforced.

3.3 Deep Learning /Sequence Models

This section consists of the literature works which have used at least one deep neural network for modeling either the visual concepts or language concepts or combinations of both. Most of the literature works in this section make use of an encoder-decoder framework (Example- CNN encoder and RNN decoder or a RNN encoder and RNN decoder). Video data unlike image data is more complex as it has a temporal dimension along with the spatial dimension to model. The interaction of various actors and objects keep evolving over time making such vast information represented by a single temporally collapsed (or fixed) feature set prone to clutter, thus incoherently fusing distinct events and objects. A lot of models in this section exploits the temporal structure underlying the video which can often be classified into two categories: A local structure, that deals with activities that are relatively localized or evolving only over few consecutive frames (example-answering the telephone) and a global structure, that deals with a whole sequence in which various objects, actions, scenes, participants etc. appear in the video [14]. Figure 3.5 illustrates how the deep learning models generally work by automatically learning internal representations during feature extraction and eventually passing it to a trainable classifier.

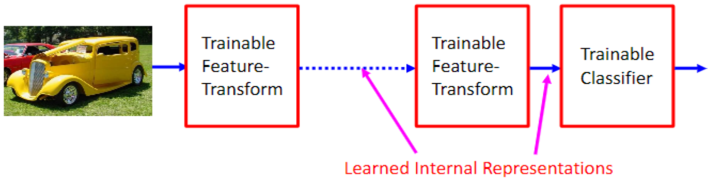

Trainable Feature-Transform
Trainable Feature-Transform
Trainable Classifier
Learned Internal Representations

Figure 3.5- Deep learning models use a layered, hierarchical structure to learn increasingly abstract feature representations from the training data and have earned a reputation for their ability to automatically learn feature representations from the input data.

In [114], the authors proposed a framework that consists of compositional semantics language model, a deep video model and joint embedding model for performing three tasks, i.e. natural language generation, video retrieval and language retrieval. The compositional semantics language model essentially enforces semantic compatibility between essential and meaningful visual concepts in videos by capturing < *subject, verb, object* > triplet (SVO triplet). The language model is constructed as a dependency tree structure based on the initial word vector. A composition function is applied to two leaf nodes with corresponding weights and is used recursively until the root node is composed which is a representation of a SVO triplet in video-text space. The deep learning model extracts visual features from the video via a deep neural network, a temporal pyramid scheme is employed to capture motion information and finally a two-layer neural network is used to map the visual features to video-to-text space. The joint embedding model minimizes the distance between the outputs produced by the deep video model and compositional language model in a joint (video-text) space and the updates these two models jointly. In [115], the authors proposed a video captioning framework that dealt with a huge dataset of video-caption pairs, of no particular domain and described them using dense captions. The framework detects (in a single pass as and when they occur) as well as describes multiple events, by sampling the video features at different strides (1, 2, 4 and 8 computed in parallel with longer strides capturing longer events), occurring during the video. The framework consists of a proposal module (which is a variant of DAP for detecting long events by enabling action localization. DAP uses non-maximum suppression to eliminate overlapping outputs, they are kept separately here and are treated as individual events), that captures short/long events and a captioning module which makes use of the context (information of past and future events) to capture the dependencies between the various events in the video, thus jointly describing all the events in the video in a semantically and syntactically correct manner. For every video frame, the proposal module (layer of LSTM units), generates a set of proposals which in turn generates a score. Only those proposals are forwarded to the language model that have a higher score when compared against a set threshold value. The captioning model takes in the hidden representations of the proposal module as input and utilizes the context gathered from other captions while captioning the event.

In [116], the work introduced a lexical fully convolutional neural network (lexical FCN) architecture for dense captioning by enforcing weakly supervised multi-instance learning techniques to link video regions with lexical labels. The architecture consisted of a lexical FCN based visual model, a region-sequence generation and a language model. The lexical FCN outputs were used to generate multiple diverse region sequences which were informative using a novel sub modular maximization scheme introduced by Shen, et al. The difference with the approach in [116] lies in the training data as fine-grained data annotations are absent from the training corpus. Lexical FCN model maps frame regions and lexical labels. Lexical FCN builds a lexical vocabulary (consisting of 6690 words) from the video caption training set by extracting part-of-speech (POS) (could belong to any part of sentences including nouns, verbs, adjectives and pronouns) of each word from the entire training dataset while treating certain frequently used functional words as stop words which are removed from the lexical vocabulary [117]. Lastly, a pre-trained CNN is trained with the Multi Instance Multi Label Learning (MIMLL) loss firstly by converting them into FCN. Figure 3.6 illustrates the difference between Multi Instance Multi Label Learning with respect to Multi Instance Learning and Multi Label Learning. The use of a region proposal candidate generation algorithm for object detection that strongly relies on bounding box ground truth for any words or concepts is automatically ruled out (due to the lack on any in the training phase) and instead the work in [116] borrows the idea from YOLO [118] for generation of candidate regions i.e. a coarse region candidate from anchor points of the last FCN layer is generated, to ultimately produce dense captions by grounding the sentences to generated sequences of ROI (region of interest).

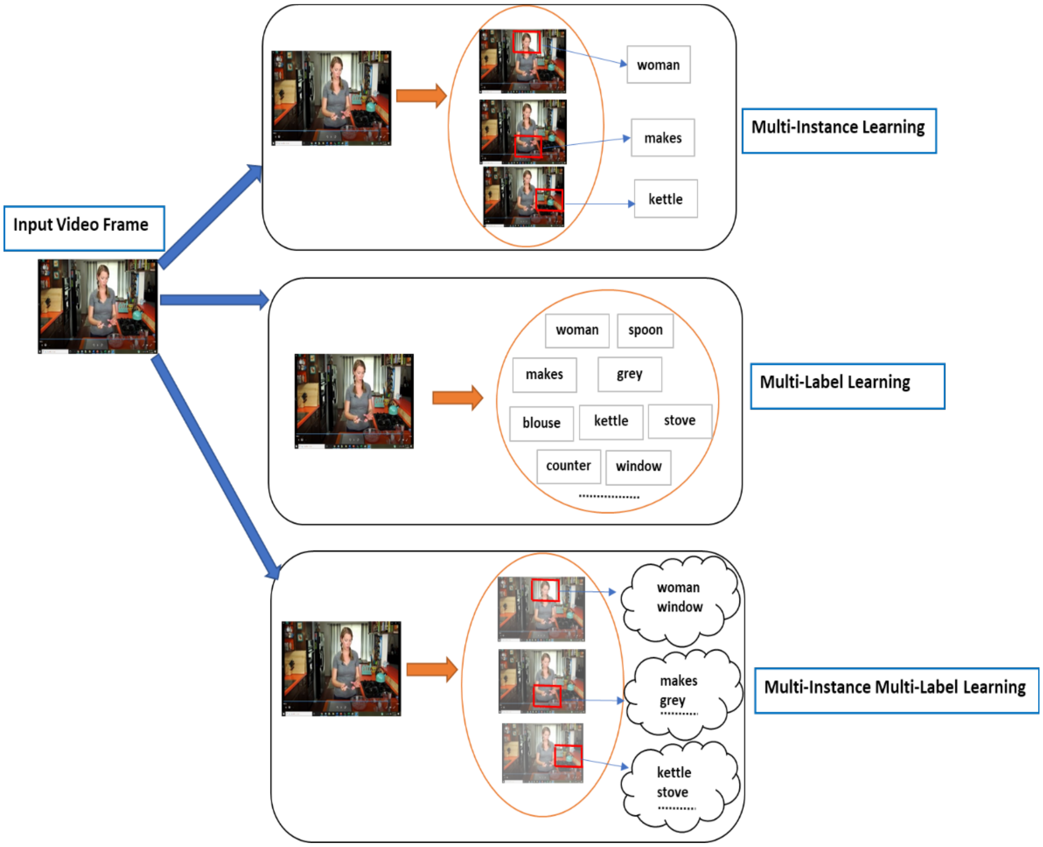


Figure 3.6- Illustration of the difference between Multi Instance Multi Label Learning with respect to Multi Instance Learning and Multi Label Learning in [116].

The work in [119] is very different from the works discussed so far as it attempts to automatically learn the main steps to complete the tasks (example changing a tire) from a set of narrated instruction videos in an unsupervised manner. Here, actual sequence and individual steps are unknown and are directly learnt from data rather than considering them to be fixed beforehand. This results in an advantage of the complimentary nature of visual signals from the video and the associated natural language captions (in this case narration) for resolving ambiguities in individual modalities. The proposed method also learns the variability in the ordering of steps to perform a task from the natural videos. In [120], the proposed method faces a new set of challenges arising from the variability in the overall structure of the sequence of steps in achieving a task. The narration of the task could have high variability with respect to the number, ordering etc., of the steps that constitute the script (order of sequence of steps). Besides, the visual appearance of each step depicted in the frames of the video could greatly vary, in other words, the people, objects, actions, etc., could be different, and the action performed by participants are captured from a different viewpoint. The problem of generating a script based on the video and transcribed audio data was modeled as two separate clustering problems (both the clustering methods focused on temporal clustering of transcribed text i.e., clustering on direct object relations that were extracted using a dependency parser, and video) which were to be performed sequentially and linked together by joint constraints. For example, two video segments with varying visual appearance but depicting the same step can be grouped together based on the similarity of the narration text used to describe them, and conversely, two video segments described with very different narrations can be grouped together under the same instruction step because of the similarity in visual appearance.

In [121], Pasunuru and Bansal proposed an architecture that shared knowledge with two related directed generation tasks:

(1) A temporally directed video prediction task using a unsupervised technique to learn richer context aware video encoder representations

(2) A logically directed language entailment generation task to learn better video entailed caption decoder representations.

In unsupervised video prediction module which is a bidirectional LSTM-RNN encoder and decoder model along with an attention model (The attention model is bidirectional LSTM-RNN encoder and a unidirectional LSTM-RNN decoder), the video representation is modeled by predicting the sequence of future video frames given the current frame sequence. In entailment generation module which also uses a bidirectional LSTM-RNN encoder and decoder with attention mechanism, a sentence (a hypothesis) is generated based on a premise (a sentence) based on logical deduction and implication of the premise. In [122], Pan et al. focused on a method that holistically exploited the relationship between the semantics of a sentence and the visual content from the frames of the video instead of generating individual words locally based on previous set of words and visual content (local generation of words approach produced contextually correct semantics, but the subject, verbs or objects described in the sentence may not be true. For instance, the sentence 'man is drinking water' is contextually correct, but man may not be the correct subject used as it is not present in the video). The architecture LSTM-E (long short term memory with visual-semantic embedding) aims at creating a visual embedding space for enforcing the relationship between the semantics of the entire sentence and the visual content, by formulating two loss functions namely, relevance and coherence loss functions, which measure the degree of relevance of the video content and sentence semantics and estimate the contextual relationships among the generated words in the sentence respectively.

In [14], Yao et al. refer to the video captioning as a video summarization task because they describe the sequence of multiple events occurring in the entire video with a single sentence by focusing on the most salient features of the video and describing them alone. The work also exploits the temporal structure underlying the video and addresses the shortcomings of [123] in two ways: (i) by employing a 3D ConvNet that incorporates spatio-temporal motion features that is pre-trained on an activity recognition video dataset and (ii) including attention mechanism explicitly so as to weigh the frame features, non-uniformly conditioned on previous word inputs rather than uniformly weighing features from all frames as in [124]. The proposed architecture made use of a 3D CNN-RNN encoder-decoder framework that incorporated attention mechanism to exploit the global temporal structure to generate effective video descriptions. Appearance features along with action features extracted from the individual frames were encoded to form the local temporal structure.

The work in [123] is an improvement over [124] and deals with open videos not restricted to any particular domain (like cooking etc.). This method uses a LSTM and encodes the temporal information from the subsequent frames of the video into a distributed vector representation generate sentential descriptions, thus not explicitly using any attention mechanism. A stacked LSTM structure is used to first encode the video frames individually by taking as input the output of a CNN and then generating a sentence word by word after all the frames were read in. This architecture models the temporal dimension of activities with the help of an optical flow [125], which is computed between pairs of consecutive frames. Flow CNN models have been shown to be beneficial for activity recognition [70], [71].

In [126], Rivera-Soto and Ord´onez designed a captioning framework by using a pre-trained 2D CNN for extracting visual features from the frames of the video and a stacked LSTM network as an encoding-decoding framework so that the framework could handle variable sized input and outputs. The vocabulary set consisted of tokenized words appearing in the dataset that were represented using an index that showcased the exact position of the words in the vocabulary. In [70], Donahue et al. were the first to propose deep learning models for video description tasks by proposing the LRCN model. The architecture was applied to videos of limited domain (cooking videos) and it employed a two-step approach for video captioning that made use of CRFs for obtaining semantic tuples of activity, object, tool and location, which were later translated to a sentence with the help of a LSTM. The video is observed as a whole sequence and not incrementally frame by frame at each time step. In [127], Shin et al. focused on developing a framework that could generate captions like a story by making use of the rich contents of the video frames by temporally segmenting the video with action localization. The multiple frames generate multiple captions which are then combined by natural language processing techniques like coreference resolution (helps connect independent captions generated from multiple frames which have no contextual relevance initially) and connective word generation (helps make the narrative generated more human like by finding appropriate transition or connective words like then) to form a multi sentence description like narrative for the video. In [128], Bin et al. employed a bidirectional LSTM for capturing the global temporal structure in the video. The joint visual model is designed by integrating the CNN and LSTM components where the former component extracts features from the frames that is integrated into the latter, thereby comprehensively exploring bidirectional global temporal information in video data. The visual representations extracted from the joint visual models are then fed into a LSTM which acts as a language model to enhance sentence generation. In [129], Gao et al. proposed an attention-based LSTM framework (aLSTM) with semantic consistency, to transfer videos to natural descriptions. The attention mechanism is a 2D CNN network that extracts the visual features from the video and inputs them to a LSTM decoder along with word embedding features of the previous time step to generate important words pertaining to the visual content and finally using a multimodal embedding to map the visual and sentence features into a joint space to guarantee the semantic consistency of the sentence description with respect to the visual video content.

In [130], the authors made use of an encoder-decoder framework for video captioning task by employing a 2D and/or 3D CNN encoder and a RNN decoder architecture called LSTMTSA (long short term memory-transferred semantic attributes). The visual module produces a visual representation by extracting visual features from 2D / 3D CNN and mean pooling them from sampled frames. The extracted feature set is fed into an LSTM layer only at the initial time. Image and video MIL models are used to mine semantic attributes from images and videos respectively which are dynamically fused using a transfer unit and incorporated into the LSTM for boosting video captioning. In [131], Li et al. proposed a caption retrieval, summarization-based framework. The framework first selects a sequence of representative frames by uniform sampling and then translates the representative frames to a sentence sequence. A summarization process inspired by LexRank [132] is proposed to generate the final description for the video sequence by constructing an adjacency graph on sentence sequences to re-rank the generated candidate sentences. In [133], Yu et al. stacked a paragraph generator over a sentence generator. Sentence generator is built upon a RNN for language modeling, a multimodal layer for integrating information from different source and an attention model for selectively focusing on the input video features. The paragraph generator is simply another RNN which models inter sentence dependency. Both made use of GRU variant of RNN. The paragraph generator receives the compact sentence representation encoded by the sentence generator, combines it with paragraph history and outputs a new initial state for the sentence generator.

In [134], Liu et al. proposed a hierarchical and multimodal video caption (HMVC) framework inspired by the work in [133]. The framework jointly learns the dynamics within both visual and textual modalities for the video captioning task. Unlike the work in [133], the framework in [134] transfers the latent intermediate knowledge from an external data source to enhance the video caption quality by leveraging the large scale image knowledge on a trained image caption model to transfer frame-level images into textual descriptions. The HMVC model makes use of three layers of LSTM for converting the video frames to textual descriptions by utilizing the visual features outputted by a CNN. In [135], Wang et al. proposed an architecture that has three main components: a CNN based video encoder, an LSTM based text decoder and multimodal memory. Pre-trained 2D and 3D CNNs are used to extract appearance features and motion information from videos respectively. The authors chose to attach a shared multimodal memory between the LSTM-based language model and CNN-based visual model for long range visual-textual information interaction, based on the work that Neural Turing Machine (NTM) [136] could capture very long range temporal dependency with external memory. Most of the literature works have introduced improvements such as attention mechanisms [14], [129], [137], hierarchical recurrent neural network [133], [138], [139], [140], features extracted using 3D convolutional neural networks [116], joint embedding space [141], language fusion [142], multi- task learning [121] to improve sequence to sequence modeling for video captioning tasks. The major problem however was a dependency on the maximum likelihood algorithm for training such models, which often gave rise to inconsistency issues, popularly known as the exposure bias (model is only exposed to the training data distribution, instead of its own predictions) [143]. This, in turn hindered the desired optimum performance of the captioning frameworks. Most of the popular text generation sequence models like feed forward neural networks and recurrent neural networks suffer from exposure bias. Exposure bias is a discrepancy that occurs when text generation models are trained to predict the next word given the previous ground truth words as input and at test time used to generate an entire sequence by predicting one word at a time, and by feeding the generated word back as input at the next time step. The discrepancy arises, as the model is trained on a different distribution of inputs, namely, words drawn from the data distribution, as opposed to words drawn from the model distribution. As a result, the errors made along the way tend to add up quickly and accumulate [143]. Also, many captioning frameworks tend to ignore topic information and try to maximize the overall likelihood for videos in all topics, which generally tend to tend to seek the most common mode in training sentences [144]. Such models thus are prone to generate plain descriptions without much details regarding the actual topic/content of video and fail in distinguishing confusing concepts within a topic.

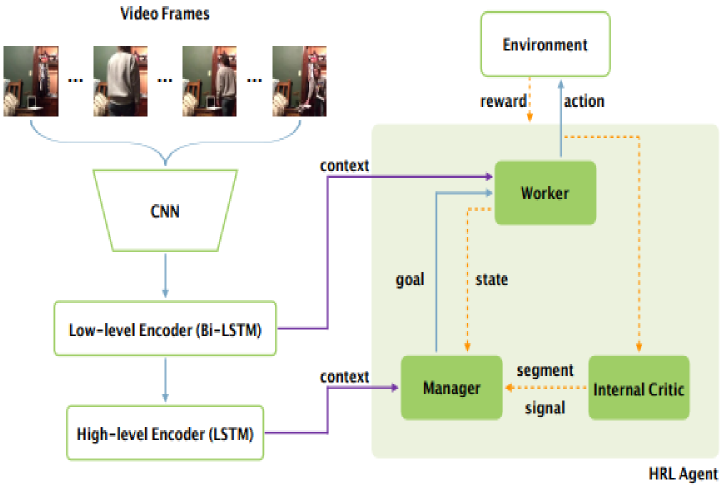


Figure 3.7- Overview of the HRL framework in [145] for video captioning. [145]

In [145], Wang et al. built an encoder-decoder HRL framework based on its effectiveness on Atari games [146], [147] and resolved the inconsistency issue of exposure bias by using a reinforcement learning algorithm. In the encoding stage, video features were extracted from the frames using a pre-trained CNN which is then passed through a low-level Bi-LSTM network as well as a high-level LSTM encoder successively. In the decoding stage, an HRL agent plays the role of a decoder and outputs a language description of a certain length using the words from the vocabulary set formed. The HRL agent has a low-level worker, and a high-level management. The former operates at a lower temporal resolution and emits a goal when it needs to signal the worker to accomplish a particular task. The low-level worker generates the corresponding words in the following few time steps, thus fulfilling the particular task and internal critic, determines whether the worker has accomplished the goal satisfactorily or not and sends a corresponding binary signal to the manager. This whole pipeline of events terminates once the end of the sentence token has been reached. Figure 3.7 illustrates the pipeline for the methodology in [145].

In contrast, in [144], a more sensitive approach with increased topic information, which in turn helps in narrowing down the general sentence distribution, enable the framework to focus on the discriminative visual contents of the video. This is particularly useful as modern captioning framework deals with open domain videos rather than videos belonging to only a domain such as cooking etc. Improved caption optimization techniques render it topic guided, helping in developing the multitask learning architecture M & M TGM, thereby helping in jointly training the caption generation system with associated topic prediction and sentence generation loss in an end to end manner. A multimodal topic mining approach is proposed to unravel latent video topics from the input video-description pairs. A supervised learning strategy is used to help the model predict latent topics with multimodal video features. To effectively and efficiently exploit topics, a topic aware language decoder model is used which implicitly functions as an ensemble of topic specific decoders for each topic, a more efficient computing method, as it requires only very less training data. In [140], the authors designed a 2D CNN encoder and hierarchical LSTM decoder with adjusted temporal attention mechanism (hLSTMat) for video captioning framework. The temporal attention mechanism focuses on using the visual words and neglecting non- visual words for improved performance of the captioning framework. The framework proposed automatically decides when and where video visual information can be incorporated into captioning and when and how a language model can be adopted effectively to generate the next word in the caption. A novel adjusted temporal attention mechanism is proposed to decide which information is important within the visual signal and when to make use of the visual information and which times to rely on the language model. The hierarchical LSTMs (two layers) are typically incorporated to obtain low level visual information and high-level language context information to generate effective captions describing the video frames.

3.4 Datasets Used for Video Captioning

Most of the datasets available today pertain to human action recognition tasks. The datasets differ with respect to whether: (1) The video clips in the dataset are short/long clips, (2) The video clips are manually trimmed or untrimmed videos. While temporal clutter causes a drop in recognition performance, untrimmed videos also contain additional information about the context of actions, (3) Addition of video clips that do not contain the target action class in the training set but an introduction of background videos that share similar scenes and objects which can be classified as positive videos, thus downplaying the role of appearance and static information since the background videos are distinguishable from action videos primarily based on the motion and (4) Introduction of videos that have multiple actions in each of the video clips of the dataset. For example, in Thumos 15, the testing video clips can have zero, one or multiple instances of an action (or different actions) that can occur anywhere and at any time in the given video clip. Recently, researchers have been collecting untrimmed video clips, illustrated in 3.13, that are obtained from multiple sources like movies [148], [149], YouTube [150], and wearable cameras [151], [152] to include in their dataset [153].

The KTH dataset [58] is primarily an action database, used mainly for action recognition tasks. Currently the database consists of 600 video files (92 training, 192 validation, 216 testing black and white video clips) which are classified into 6 classes of human actions i.e. walking, jogging, running, boxing, hand waving and hand clapping, which are performed several times by 25 different participants in 4 different scene settings (outdoors s1, outdoors with scale variation s2, outdoors with different clothes s3 and indoors s4). Figure 3.8 shows the various actions in the KTH action recognition dataset.

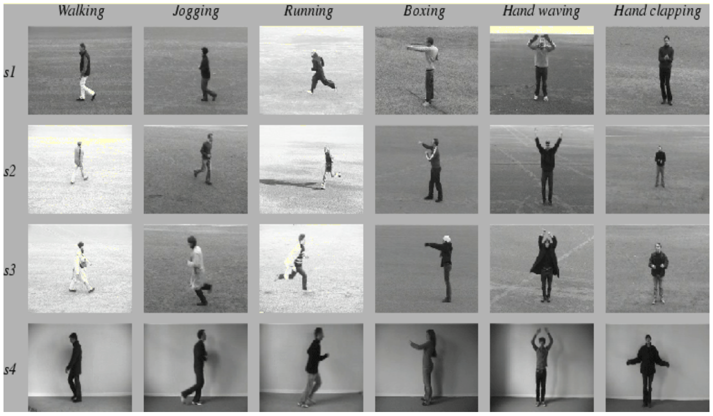


Figure 3.8- Sample actions from the KTH dataset. [58]

The Weizmann dataset [154] is yet another simple action recognition dataset, action classes of which are illustrated in Figure 3.9. The dataset consists of 10 action classes (walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack, skip) and has a total of 90 low-resolution video sequences which show 9 different participants, each performing 10 natural actions.

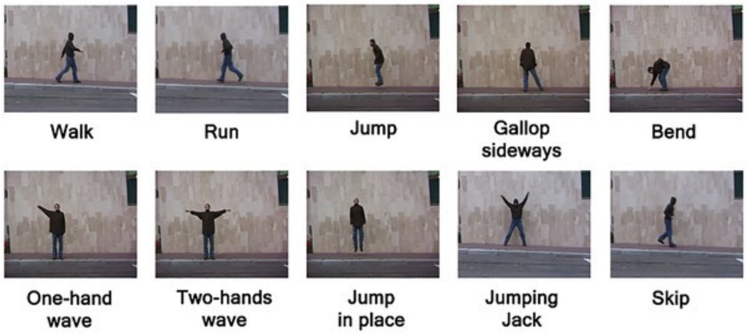


Figure 3.9- Sample actions from of the Weizmann dataset. [154]

UCF101 dataset [155] is an extension of UCF50 [156][158] which extends the action classes from 50 to 101 different action classes and has realistic video as opposed to other action recognition datasets, with actions performed by actors. Figure 3.10 illustrates sample classes of the UVF 1010 dataset. The 13320 realistic action videos are collected from YouTube providing the dataset with a larger diversity in terms of action, variations in camera motion, object appearance or pose, object scale, viewpoint, cluttered background, illumination conditions, etc. The action classes are further grouped into 25 categories based on sharing certain features like similar background, similar viewpoint, etc., with each category consisting of 4-7 videos of a particular action. The 101 action verbs can be classified under 5 broad labels: (1) Human-Object Interaction 2) Body-Motion Only 3) Human-Human Interaction 4) Playing Musical Instruments 5) Sports.

Table 3.2- Some popular benchmark datasets for video description and classification

Dataset	**Domain**	**Caption_Source**	**Classes**	**Number_of_Videos**
UCF101	Sports	-	101	13k
Sports 1M	Sports	-	487	1.1M
Thumos 15	Sports	-	101	21K
HMDB 51	Movie	-	51	7K
Hollywood 2	Movie	Script + DVS	12	4K
MPII cooking	cooking	Self (university students)	78	44
MPII MD	Movie	Script + DVS	Captions	68K
MVAD	Movie	DVS	Captions	49K
MSR-VTT	Open	AMT Workers	Captions	10K
MSVD	Human activities	AMT Workers	Captions	2K
YouCook	cooking	AMT Workers	Captions	88
Charades	Human activities	AMT Workers	157	10K
TACoS multi-level	cooking	AMT Workers	Captions	127
ActivityNet Captions	Open	AMT Workers	Captions	20K

Datasets like KTH and Weizmann were ill suited for evaluating sentential descriptions (for such datasets sentential descriptions would contain no other information other than the verb) for activity recognition tasks as they depict actions with only a single human participant whereas in reality the corpora should consist of clips showcasing complex actions with multiple participants.

While the KTH and Weizmann datasets made use of actors for performing a small set of scripted actions (those mentioned as the action classes) under controlled environment, datasets like CMU [157] and MSR Action [158] decided to change the setting a little by maintaining the scripted actions but against challenging dynamic backgrounds [153]. Yet some other datasets consist of a lot of action classes which are often irrelevant to the choice of verb. However, many of the modern datasets are collected from realistic sources (*e.g.,* YouTube), and untrimmed video clips

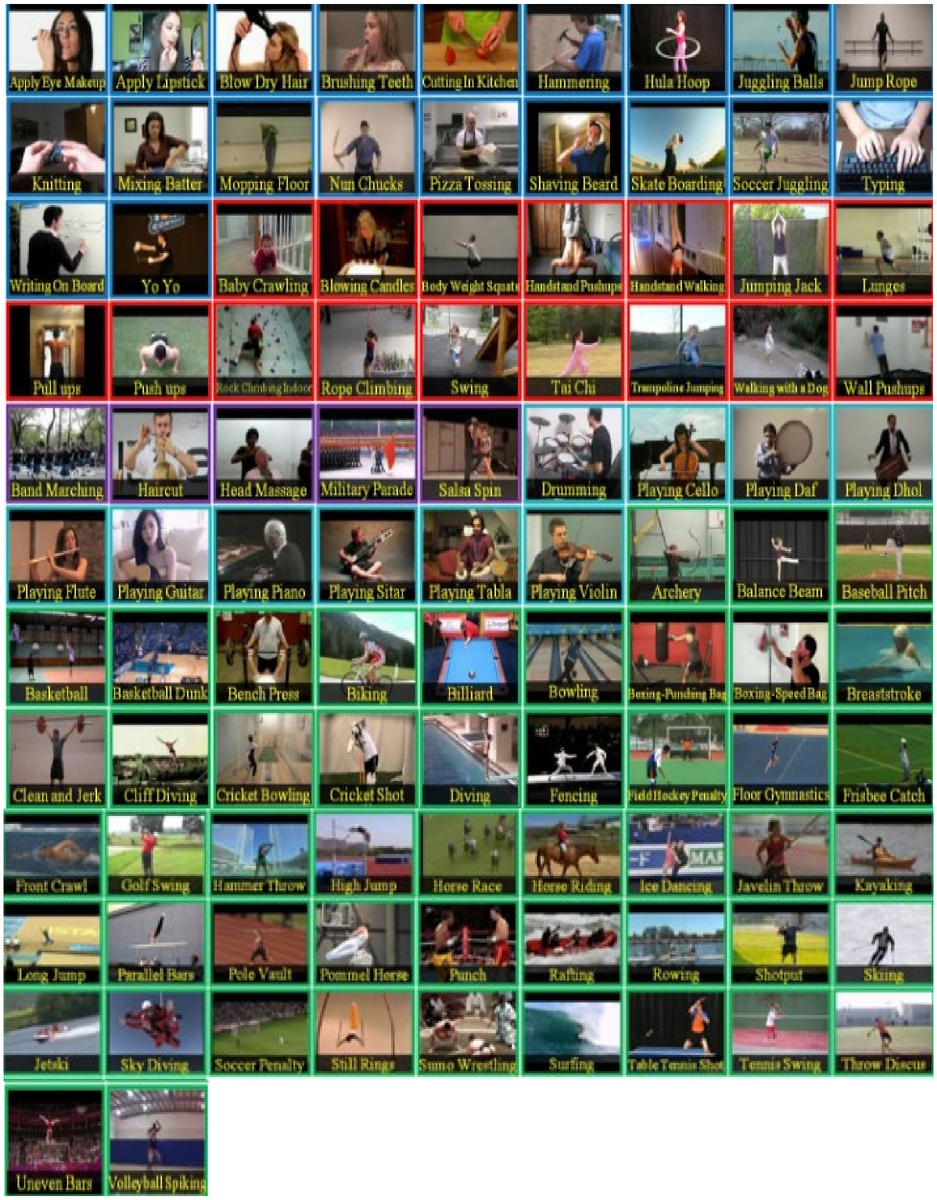


Figure 3.10- Sample videos from the UCF 101 dataset depicting all the 101 action classes. [155]

with large number of action classes and temporal clutter. A few other datasets are made with respect to certain chosen domains only, e.g., cooking, sports, Olympic sports, sports 1M, etc. Olympic Sports in [159] consists of 800 video clips and 16 action classes. It was first introduced in 2010 and is different from the previous datasets, with all the videos being downloaded from the Internet.

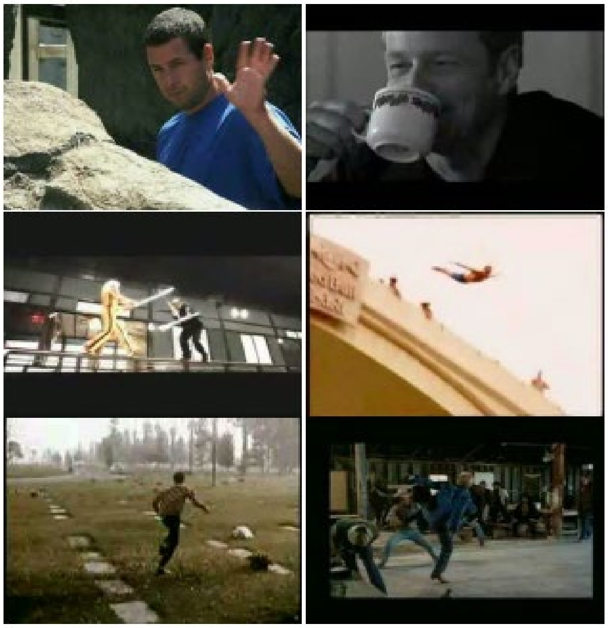


Figure 3.11- Sample video frames from HMDB51 (from top left to lower right, actions are: hand-waving, drinking, sword fighting, diving, running and kicking) dataset. The key challenges are faced are: large variations in camera viewpoint and motion, the cluttered background, and changes in the position, scale, and appearances of the actors. [148]

Sports 1M [41] consists of 1,133,158 video URLs, annotated automatically with 487 Sports labels or action classes, using the YouTube Topics API. Each action class consists of about 1000-3000 videos. These are real videos (not staged by actors) but are restricted predominantly to sports domain only. Thumos 15 dataset [160], an extension of the Thumos 14 dataset [161], is also an action recognition dataset with videos downloaded from YouTube, manually annotated for both action label as well as its temporal span. The Thumos 15 dataset included a new test set constituting

5613 positive and background untrimmed videos which was absent in the Thumos 14 dataset. Figure 3.12 illustrates the difference between the untrimmed videos of Thumos 15 dataset and trimmed videos of the UCF 101 dataset. The action classes in both these datasets are the same as in the UCF101 dataset.

HMDB51 [148] is a larger video dataset, used for human motion recognition task. This dataset consists of 51 distinct action classes from diverse sources like digitized movies, public databases (*e.g.*, Prelinger archive), videos from the internet, and YouTube, and Google. These videos were used to collect various realistic clips for the HMDB51 dataset. A typical sample frame is illustrated in Figure 3.11. The action categories were grouped under five types (different grouping from the UCF101 and Thumos 15):

(1) General facial actions: smile, laugh, chew, talk

(2) Facial actions with object manipulation: smoke, eat, drink

(3) General body movements: cartwheel, clap hands, climb, climb stairs, dive, fall on the floor, backhand flip, handstand, jump, pull up, push up, run, sit down, sit up, somersault, stand up, turn, walk, wave

(4) Body movements with object interaction: brush hair, catch, draw sword, dribble, golf, hit something, kick ball, pick, pour, push something, ride bike, ride horse, shoot ball, shoot bow, shoot gun, swing baseball bat, sword exercise, throw

(5) Body movements for human interaction: fencing, hug, kick someone, kiss, punch, shake hands, sword fight [148].

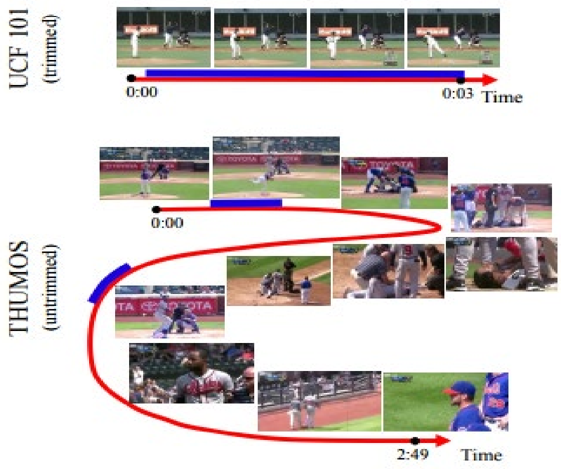


Figure 3.12- Distinction between trimmed UCF101 dataset and untrimmed Thumos 2015 dataset. [153]

HOHA (Hollywood Human Actions Dataset) [57] consists of a total of 8 action classes, formed from 32 hollywood movie clips. The prominent action labels used include: AnswerPhone, GetOutCar, HandShake, HugPerson, Kiss, SitDown, SitUp and StandUp. An extended form of HOHA is Hollywood-2 (human action and scenes dataset) [149], and contains two separate classification classes:

(1) Twelve classes of human actions

(2) Ten classes of scenes.

There are over 3669 video sequences that run for approximately 20.1 hours in total. MPII-MD [162] is MPII movie description dataset featuring movie snippets aligned to scripts and DVS (Descriptive video service). DVS is a linguistic description that allows visually impaired people to follow a movie. The Montreal Video Annotation Dataset (M-VAD) [163] is another dataset which is similar to the MPII-MD dataset but contains AD data with automatic alignment and is a collection of around 49,000 short video clips from 92 different movies. For every movie clip, an appropriate single description is created with the help of DVS. The movie snippets of high visual and textual diversity are included in the dataset, generating only a single reference sentence for each movie clip, making the video captioning task on the M-VAD dataset a very challenging task. Figure 3.13 lists various popular video captioning datasets along with certain reference sentences against the sample video clips (ground truth). Figure 3.14 illustrates a sample video clip from the M-VAD dataset showing the AD and DVS descriptions associated with the clip.

Several datasets like the CMU-MMAC dataset, YouCook dataset etc., were created exclusively for domain cooking. Most of these datasets are demonstration videos of fine grained to coarse grained video clips, *e.g.*, video demonstrating a particular cooking task. The CMU-MMAC (CMU Multi-Modal ACtivity database) dataset [164] contains multimodal cooking activities of five recipes: brownie, eggs, pizza, salad, and sandwich. The modalities include RGB videos from static and wearable cameras, multi-channel audios, motion capture, inertial measurement units (IMU), etc. Though the number of subjects involved has not been asserted, the inference is between thirty-nine and forty-five subjects. Each subject performed all the recipes.

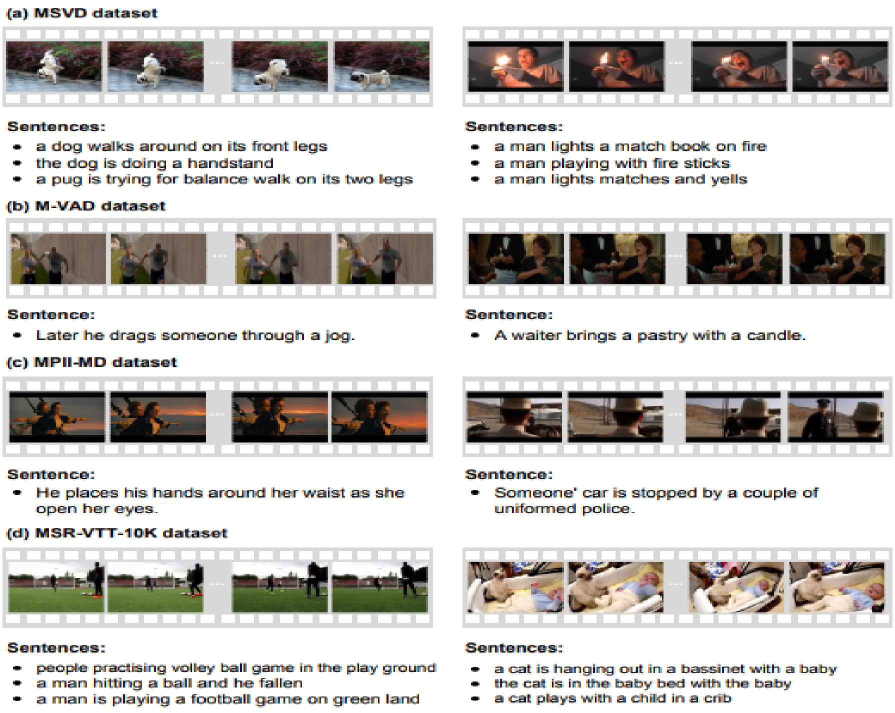
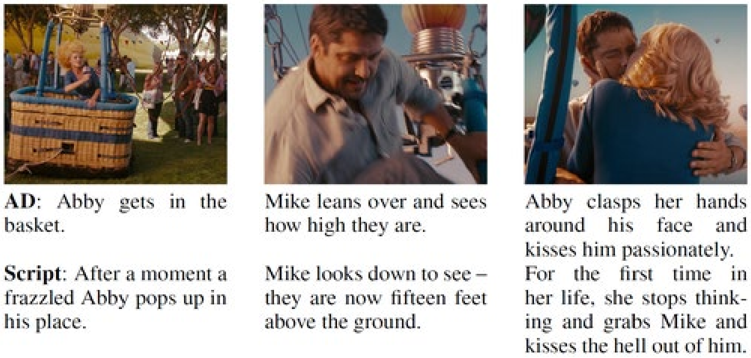

(a) MSVD dataset
Sentences:
a dog walks around on its front legs
the dog is doing a handstand
a pug is trying for balance walk on its two legs
Sentences:
a man lights a match book on fire
a man playing with fire sticks
a man lights matches and yells
(b) M-VAD dataset
Sentence:
Later he drags someone through a jog.
Sentence:
A waiter brings a pastry with a candle.
(c) MPII-MD dataset
Sentence:
He places his hands around her waist as she open her eyes.
Sentence:
Someone' car is stopped by a couple of uniformed police.
(d) MSR-VTT-10K dataset
Sentences:
people practising volley ball game in the play ground
a man hitting a ball and he fallen
a man is playing a football game on green land
Sentences:
a cat is hanging out in a bassinet with a baby
the cat is in the baby bed with the baby
a cat plays with a child in a crib

Figure 3.13- Examples of the captions associated with the MSVD, M-VAD, MPII- MD and MSR-VTT-10K datasets respectively. [40]

MPII sequentially created three datasets related to the cooking domain: (i) MPII Cooking dataset [164], (ii) MPII Cooking Composite dataset [165] and (iii) MPII Cooking 2 dataset [166]. While (i) focuses on fine grained activity and (ii) on composite basic level activities, (iii) was a unified and upgraded version of [164], [165], [167]. YouCook dataset [109] consists of 88 in-house cooking videos crawled from YouTube, classified under six different cooking styles (e.g., baking, grilling, etc.). All the videos are taken in a third-view viewpoint and different kitchen environments. Each video is annotated with multiple human descriptions by AMT. Each annotator in AMT is instructed to describe the video in at least three sentences totaling a minimum of 15 words, resulting in 2,668 sentences for all the videos [40]. The TACoS Multi-Level dataset (TACoS-ML [113]) is built following the MPII Cooking 2 dataset [166] and consist of 185 long videos along with associated textual descriptions collected by AMT workers.



AD: Abby gets in the basket.

Script: After a moment a frazzled Abby pops up in his place.

Mike leans over and sees how high they are.

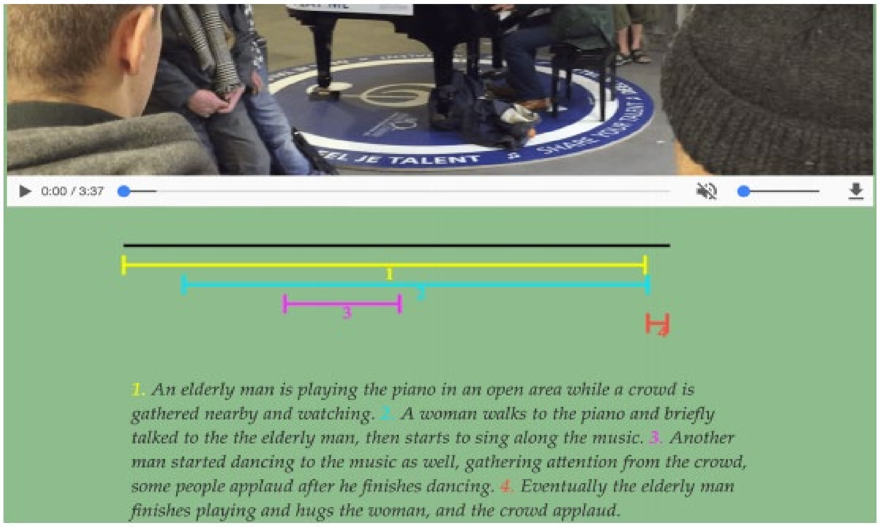
Mike looks down to see – they are now fifteen feet above the ground.

Abby clasps her hands around his face and kisses him passionately.

For the first time in her life, she stops thinking and grabs Mike and kisses the hell out of him.

Figure 3.14- Sample video clip from the Montreal Video Annotaion Dataset (M-VAD) dataset showing the AD and DVS description associated with the clip. [163]

Microsoft Research Video Description Corpus (MSVD) [168] contains 1,970 YouTube snippets collected on Amazon Mechanical Turk (AMT) by requesting workers to pick short clips depicting a single activity. These video clips were then labeled with single sentence descriptions by annotators. The original corpus has multi-lingual descriptions but only the English descriptions were used for video captioning task. On an average, around 40 English descriptions per video were available. The standard split of MSVD employed was 1,200 videos for training, 100 for validation and 670 for testing, as suggested in [40], [108]. MCG-WEBV dataset [169] is yet another large set of YouTube videos which has 234, 414 web videos with annotations on several topic-level events like a conflict in Gaza. MSR Video to Text (MSR-VTT-10K) [170] is a large-scale benchmark for video captioning task. It consists of a total of 10K Web video clips of 41.2 hours duration, obtained from a commercial video search engine, e.g., music, people, gaming, sports, and TV shows and covers the most comprehensive 20 categories. Each video clip is annotated with about 20 natural sentences by AMT workers. Lastly, the ActivityNet captions dataset [115] is an open domain dataset consisting of dense captioned 20K videos amounting to 849 video hours, with a total of 100K descriptions, each marked with a unique start and stop time. Figure 3.15 illustrates a sample video clip from the dataset describing the clip in detail through dense captions. The collection of videos in this dataset relies on the efficiency of AMT workers, whose annotation task focuses on describing the video with a paragraph, with at least a sentence describing every major event occurring in the video.



1. An elderly man is playing the piano in an open area while a crowd is gathered nearby and watching. 2. A woman walks to the piano and briefly talked to the the elderly man, then starts to sing along the music. 3. Another man started dancing to the music as well, gathering attention from the crowd, some people applaud after he finishes dancing. 4. Eventually the elderly man finishes playing and hugs the woman, and the crowd applaud.

Figure 3.15- Sample video clip from the ActivityNet captions dataset, describing the video with a paragraph, with at least a sentence describing every major event occurring in the video. [115]

Table 3.3- Comparison of datasets used by video description techniques

Ref No	Dataset Used	Additional comments
[94]	Dataset of 565 training and 140 test videos	One of the older papers which did not use any of the modern-day activity recognition or video captioning dataset.
[96]	No dataset	Not a data driven approach
[98]	The authors created their own dataset based on the surveillance of human activities in a machine room of their laboratory.	30+ verbs for in-door human activities and 9 object models. The allocations of a door, a table and other equipment in the lab were given in advance
[99]	Dataset was manually created from a subset of rushes videos used for 2007 and 2008 TREC video summarization task.	The dataset had 1 TRECVID annotation and annotations produced by 20 different human subjects
[102]	Dataset of 90 different scenes	Mostly focused on urban traffic and maritime scenes
[104]	English portion of YouTube data collected by chen & Dolan (2011)	Each of the videos were short and had multiple descriptions
[105]	DARPA dataset	-
[106]	Y1 dataset	-
[107]	TACoS Corpus	Made use of human activity videos in the kitchen scenarios from the dataset.
[108]	Large YouTube corpus	-
[109]	TRECVID MED12 and YouCook	Tested on 2 different datasets
[112]	Dataset of 94 video clips	-
[113]	TACoS dataset	-
[115]	ActivityNet captions dataset	-
[116]	MSR-VTT dataset	
[119]	Dataset had narrated instruction videos for 5 tasks	The 5 tasks were: making coffee, changing car tire, performing cardiopulmonary resuscitation (CPR), jumping a car and repotting a plant
[121]	YouTube2Text or MSVD dataset, MSR-VTT dataset and M-VAD dataset	This paper makes use of unsupervised video representation learning Task by using the UCF-101 action videos dataset.
[122]	YouTube2Text or MSVD	-
[14]	YouTube2Text or MSVD and DVS	-
[123]	MSVD, MPII-MD and M-VAD	-
[126]	MSVD	-
[70]	TACoS multilevel	-

[114]	YouTube2Text	-
[127]	Montreal, MPII and MS video	-
[128]	MSVD	-
[129]	MSVD and MSR-VTT	-
[130]	MSVD, M-VAD and MPII-MD	-
[131]	MSR-VTT	-
[133]	YouTube clips and TACoS multilevel	-
[134]	MSVD, MPII-MD and MSR-VTT	-
[135]	MSVD and MSR-VTT	-
[145]	MSR-VTT and Charades	The charades captions dataset is rarely used for various captioning frameworks.
[144]	MSR-VTT and YouTube2Text	-
[140]	MSVD and MSR-VTT	-

3.5 Evaluation Metrices

Evaluation measures like BLEU@N (BiLingual Evaluation Understudy) [171], METEOR (Metric for Evaluation of Translation with Explicit Ordering) [173], CIDEr (Consensus based Image Description evaluation) [172] and ROUGE (Recall Oriented Understudy for Gisting Evaluation) [177] are popular machine translation metrics (*e.g.,* to evaluate quality of text generated during translation, say, French to English), which are commonly adopted for quantitative evaluation of different video captioning tasks. Latest literature works in video captioning however also use SPICE (Semantic Propositional Image Captioning Evaluation) [178] and WMD (Word Movers Distance) [179] evaluation metrics to evaluate the quality of captions generated. Overall, evaluation metrics can be classified as either human/subject metrics or automatic metrics which evaluate the machine translated captions as illustrated by Figure 3.16. Human evaluation is preferable but has some limitations as there could be various ways of interpreting and describing a video with a caption (with reference of object of importance or foreground background subjects or by highlighting the main activity verb of the video), leaving the performance unsatisfactory.

Automatic evaluation metrics save time of the human experts and evaluate the machine generated caption based on some criteria (example-precision).

BLEU@N [171] was one of the oldest evaluation measures developed for evaluating the quality (quality is indicated by a number by checking the correlation between the machine generated text and that of a human) of the generated text by outputting a number between 0 and 1, with 1 representing high similarity between candidate and reference texts. BLEU@N measures a fraction of N-gram (1-gram, 2-gram, 3-gram up to 4-gram) that are common between the generated text or hypothesis and a set of reference texts. In language to language translation context, it is well known that BLEU@N is not a perfect evaluation measure as it is unable to perform translation of languages lacking word boundaries. Moreover, an increased BLEU@N score does not indicate an improved translation quality. BLEU@N score is achieved by evaluating the overlap of a candidate text with a set of reference texts, but in reality, the sentences can be framed in different ways leading to little or no overlap making BLEU@N an ineffective metric. According to [12], the N-gram matching followed in BLEU@N metric also leads to ineffective results for highest N (*i.e.,* 4) matching at sentence level as it rarely occurs, rendering BLEU@N to indicate poor performance when comparing two sentences. Though a lot of critics argue on the effectiveness of using BLEU@N score as an indicator of improved translation quality, many researchers are using it as only one (not sole) of the evaluation measures in their work. Thus, two other evaluation metrics: METEOR (Metric for Evaluation of Translation with Explicit ORdering) and CIDEr (Consensus-based Image Description Evaluation), were proposed to be used along with BLEU@N by many NLP-CV communities. The METEOR metric has shown better correlation with human subjects unlike BLEU@N which correlates weakly with human judgment, thus making it a better metric for measuring the quality of the description generated. However, [174] found that unigram-

BLEU performed still better than METEOR for image caption generation task. METEOR [173] was initially proposed to evaluate the description produced for image captioning tasks. METEOR returns a score to highlight the quality of generated description by computing the F-measure based on N-grams matches of the candidate and reference sentences by sophistically resolving word level correspondences using exact matches, stemming and semantic similarity (i.e., exact word matches will include similar words based on WordNet synonyms and stemmed token). In the F-measure calculation of METEOR, parameters are set in a way that favors recall over precision in its computation. The METEOR score is also calculated at the sentence level like the BLEU@N metric.

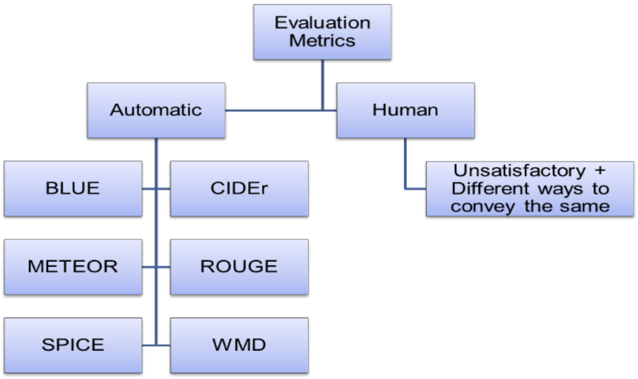


Figure 3.16- Classification of evaluation metrics

CIDEr [172] was proposed mainly for evaluation of image captioning tasks. This metric inherently captures the notions of grammaticality, saliency, importance and accuracy (precision and recall) by calculating sentence similarity. To intuitively encode the measure of consensus of how often the N-grams in a candidate description is present in the reference sentences, Term Frequency Inverse Document Frequency (TF-IDF) weighting for each n-gram [37] is performed. Finally, a CIDErN score that accounts for both precision and recall is calculated for N-grams of length N (1 to 4) using a similarity measure (average cosine similarity) between candidate and reference sentences.

Table 3.4- Tasks the automatic evaluation metrics were originally proposed for or borrowed from.

Evaluation Metrics	**Tasks Originally Proposed for**	**Underlying Idea**
BLEU [170]	Machine Translation	N-gram Precision
CIDEr [171]	Image Captioning	tf-idf weighted n-gram similarity
METEOR [172]	Machine Translation	N-gram (synonym matching)
ROUGE [176]	Document Summarization	N-gram Precision
SPICE [177]	Image Captioning	Scene-graph synonym matching
WMD [178]	Document Similarity	Earth Mover Distance on word2vec

SPICE and WMD are lesser used evaluation metrics for video captioning. SPICE [178] calculates and measures the cosine similarity by generating scene graph tuples from the reference and generated captions. The semantic graphs encode objects, their attributes and relationships through a dependency parse tree and the SPICE score is finally computed with respect to the F1-score between the graph tuples of machine generated descriptions and the ground truth/reference captions. The WMD [179] metric is slightly advanced with respect to its ability to address and deal with different words having the same semantic meaning and also be able to associate different meaning to context even when they have same attributes, objects and their relations. WMD measures the dissimilarity between two text documents in case of a NLP-NLP machine translation task by making use of word embeddings which are semantically meaningful vector representations of words learnt from text corpora.

Table 3.5 – The BLEU, METEOR, CIDEr and ROUGE automatic evaluation metrics are very popular and frequently used in video/image captioning tasks.

Automatic Evaluation Metric	**Mathematical Equation That Represents It**
BLEU	$\log Bleu = \min\left(1 - \frac{l_r}{l_c}, 0\right) + \sum_{n=1}^{N} w_n log p_n$ $\frac{l_r}{l_c}$ = ration of length of reference (GT) and generated captions, w_n =positive weights p_n=geometric average modified N-gram precision
METEOR	$F_{mean} = \frac{10. Precision. Recall}{Recall + 9. Precision}$ Where, $penalty,\ p = 0.5(\frac{N_c}{N_u})^3$,N_c is total number of chunks of generated caption and N_u is number of unigrams matched. METEOR score s = $F_{mean}(1 - p)$
CIDEr	$CIDEr_n(c_i, S_i) = \frac{1}{m}\sum_j \frac{g^n(c_i). g^n(S_{ij})}{\|\|g^n(c_i\|\|\|g^n(S_{ij})\|\|}$ $CIDEr(c_i, S_i) = \sum_{n-1}^{N} w_n CIDEr_n(c_i, S_i)$ Where, c_i represents the generated sentence and S_i represents the reference, GT. The numerator of the $CIDEr_n$ score represents the TF-IDF vector (n-gram) and denominator represents the cosine similarity score between the generated caption and reference captions.
ROUGE	$ROUGE_n = \frac{\sum_{S\in\{Reference\ Summaries\}} \sum_{gram_n \in S} count_{match}(gram_n)}{\sum_{S\in\{Reference\ Summaries\}} \sum_{gram_n \in S} count(gram_n)}$

4 VIDEO SEQUENCES TO TEXT TRANSLATION THROUGH SEMANTIC CONCEPT GENERATION

Brief Abstract. Recent times have witnessed rapid growth in the fields of Automatic Image Annotation (AIA), Image Captioning, Activity Recognition and Video Tagging using Advanced Machine Learning / Deep Learning techniques / architectures. In contrast, progress in the case of more complex and challenging Video Captioning is rather slow as it inherits the complexities of all the aforementioned tasks. Essentially, Video Captioning involves description of a video clip, capturing its overall visual semantics with natural language. In the case of open or domain specific datasets, Video Captioning through development of a neural network architecture or utilizing combination models like the common encoder decoder architecture of Convolutional Neural Networks and Recurrent Neural Networks (CNN-RNN) is very challenging. These encoder-decoder models which are the current state-of-the-art architectures for video description perform the video to natural language translation via a black box model. This paper provides an alternative narrative on how a high quality visual semantic tag extractor can aid in generating good captions by utilizing refined techniques from the object detection and activity recognition tasks. At the same time, sincere attempts are being made to offer a deeper insight into the different strengths and weaknesses of popular deep visual captioning models. The proposed captioning model is memory efficient and has two major components in its architecture: 1) semantic tag prediction models – Deep Neural Network (DNN) and Bidirectional Long Short-Term Memory (BiLSTM), which have a significant contribution in captioning and 2) caption generation language model, which is a two layer stacked model of Long Short-Term Memory (LSTM). Comparative studies of our models against standard Microsoft Research Video Description (MSVD) and Microsoft Research-Video

to Text (MSR-VTT) datasets for their sentence generation ability in terms of METEOR and CIDEr were found highly satisfactory.

4.1 Introduction

Recent reports indicate that videos, with its usage exceeding sixty-five percent of search results, dominate among different kinds of multimedia digital content on the internet [1]. Captioning a video is very useful as it helps in reaching out and explaining the content to a larger audience, especially to those viewers who are non-native speakers of the language in the video or for those persons who are deaf or hard of hearing. Even otherwise, caption is helpful to users as it provides much better experience while watching videos. Empirical studies indicate that video-captioning helps in improving attention span, comprehending/understanding and memory retention of the targeted audience [2]. In recent times, the advances in deep learning architectures with respect to various domains (like speech, image, etc.), have contributed significantly in enhancing active cooperation between the groups of language and vision communities. This has led to effective exploitation of the different multimodal cues abundant in any image or video data packets for robust feature representations. The ability to generate sentences or descriptions in natural language for a given video is the crucial step towards achieving machine intelligence with wide ranging applications in the field of video retrieval, blind navigation, etc. [3]

Visual recognition and description, though easy for humans to perform, is still a difficult and daunting task for computers. In fact, a plethora of challenges from both the vision as well as language generation perspectives crop up during video captioning. The modern video captioning models have taken a lot of inspiration from existing deep image captioning and action recognition models. Just as the earlier works of image captioning models focused on translation of an image to natural language by focusing on constructing various linguistic templates or syntactic trees, earlier video captioning models too heavily restricted the captions generated from a given visual concept due to lack of advanced language models, feature extractors and attention mechanisms. Early video captioning models relied on hand crafted video features whereas the language model that followed was based on a template or a shallow statistical machine translation approach that was used to produce a caption. But modern video captioning techniques focus on developing joint embedding or encoder-decoder models which can be trained in an end-to-end fashion to effectively and fluidly translate the automatically extracted video features to natural language descriptions.

Video captioning generate suitable descriptions to describe the visual content of the video i.e. a task of intelligent understanding of the sequential visual information and translating this understanding to natural language. However, this task of generating descriptions from dynamic video clips involve multiple challenges. Figure 4.1 describes a general hierarchical flow on how to categorize the deep video captioning models. The first hurdle is with respect to the availability of suitable datasets itself. Compared to image classification and recognition tasks, video annotations and captioning are more complex, tedious, ambiguous and expensive. Hence, supervised learning techniques have video description tasks with limited text descriptions and accompanying datasets. The second challenge is how to develop an appropriate captioning model to identify and recognize the major events in a clip. This should encapsulate the inter/ intra dependencies of multiple activities occurring in a clip (for medium to large video datasets with longer average running times per video), which may or may not be interleaving with each other. It should also lead to developing a model capable of inferring the dependencies of various activities that may not be visually explicit as some activities may be present only in a subtle or hidden manner and need to be inferred. The challenge here is how to develop a model perspective that can capture both the spatial and temporal information of the video in an efficient way and incorporating it into the model.

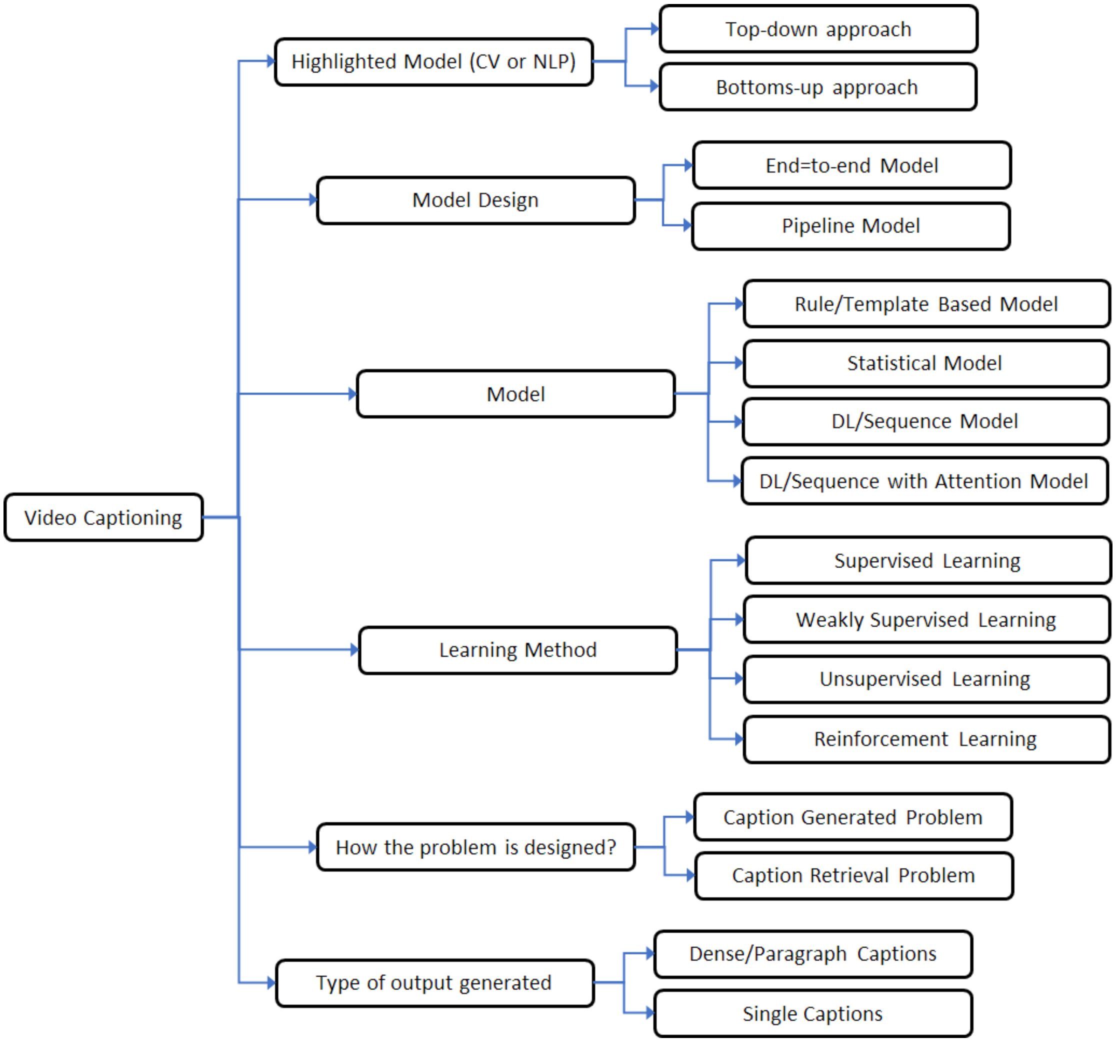


Figure 4.1 - A hierarchy of classification of video captioning models

The workflow of our proposed system partly based on [188] is depicted in Figure 4.2. We have attempted to develop a video captioning generation model by first developing a tag prediction model that extracts the semantic tags from key frames of the video clip. The captioning language model is conditioned on the features extracted from the frames as well as the tags generated from the tag prediction model. The main achievement of this work is in the usage of simple models like Deep Neural Networks (DNN), LSTM and BiLSTM as potential alternative for tag prediction, that are equally capable of addressing the underlying issue of exploiting the mutual relationships between various video representations and attributes for improved video captioning. This work showcases promising results and could lead to a way for further useful exploitation of simple models for greater effectiveness and efficiency.

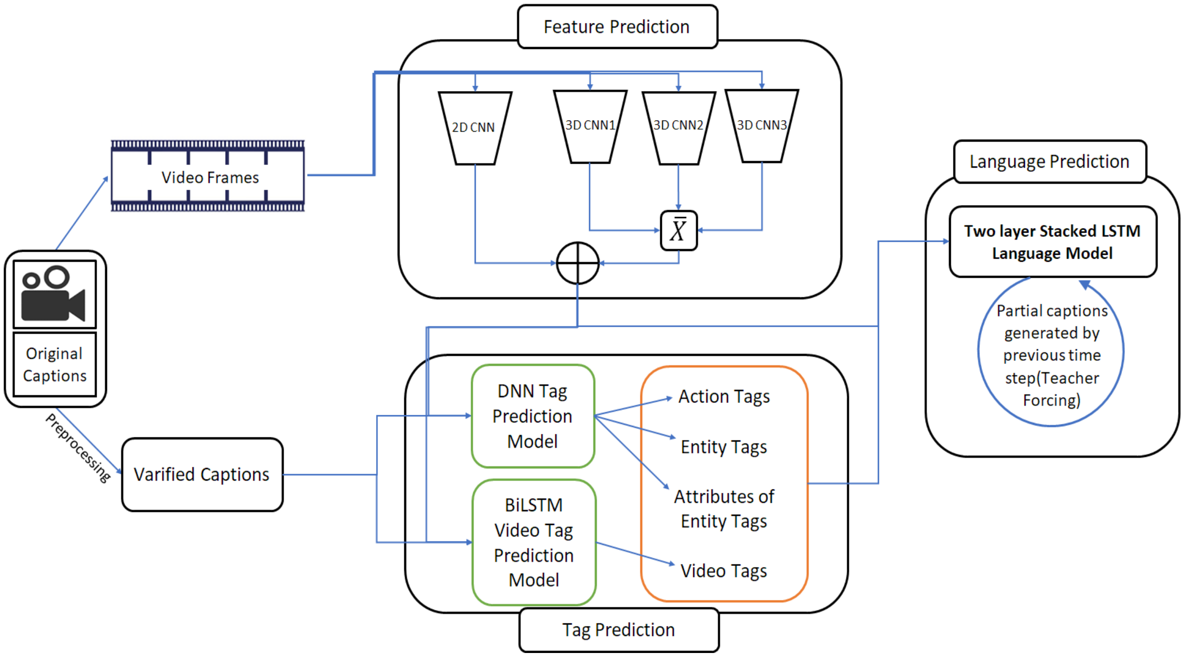


Figure 4.2 - Pipeline of our video captioning model.

Recent captioning techniques have focused more on modeling the vision and language aspects jointly in a supervised setting. In this section we have attempted to compare only those works which are relevant and comparable to our proposed work. The earlier methods (template based models) were focused more on specific activity and/or context based datasets with smaller vocabularies of limited objects and actions. In contrast, 'Deep learning models', alternately also called 'sequence learning models' follow a data driven approach, making use of deep neural networks such as CNN, RNN or combination of both, to achieve the desired goal. Sequence learning architectures are also known as encoder-decoder architectures, focusing directly on translating video content into natural language sentences. This is however completely inspired by the late revelation (emergence) and quick advancement of recurrent neural architectures, primarily the LSTMS as useful language models.

A cursory survey of the available literature on video captioning techniques reveals that the task has evolved rather gradually through three phases, always looking at and analyzing the given problems through different perspectives, as illustrated in Figure 3. In the initial period or first phase, the methods [96] [107] [181] were developed by the computer vision and natural language processing researchers working independently and not in tandem as an integrated unit. During this time, not much attention was paid for developing a good language model. Instead, more often, standard sentence templates highlighting manually extracted important features from the video clips and fitting them into a single predefined template, with stress on grammatical correctness were employed. This template language model initially split sentences into fragments (example – subject, verb and object (SVO)) based on the well laid rules of grammar and subsequently associated these fragments with the visually detected word/feature from the video. This led to a very limited scope of descriptions being generated and often failed to replicate its success to a decent size dataset within a broader domain. The second phase [106] [108] [109] was very brief and somewhat coincided with the third phase of the model development. This made use of some statistical methods to deal with decent size datasets. The third phase deployed deep learning architectures either for the feature extraction phase, or as a language model or for both. Deep learning models are currently considered to be the state of the art for video captioning and a lot of related vision / language tasks. The limitations of the prior phases were overcome using the deep learning models, especially with its large datasets, now being considered as its strength rather than weakness or handicap, rendering it highly suitable for such tasks.

The model architectures for video captioning tasks are very varied. They depend on a lot of pre-factors which are connected to the type of the dataset used, like, whether it is open domain, domain specific dataset, the length of the individual videos in the dataset, the total number of events in a single video, whether dense captions or single captions are expected, etc. The sub-categories of the deep learning models are selected based on how these deep neural networks are trained: that is, whether they are supervised learning, weakly supervised learning or unsupervised feature learning techniques. Overall, the literature on video captioning can also be further sub classified based on the type of captioning techniques they represent (single/dense or generating/retrieval techniques).

Initial deep learning models [114] [115] [121] [122] for the video captioning tasks were inspired by the progress in machine translation and the early success of image captioning models, thus formulating the problem as a natural extension to image captioning. Here a single semantic representation of a frame in the clip was chosen to be representative of the whole video. The major task involved was in extracting that single representation feature and passing it on to the language model for effective translation in the form of a natural sentence. This kind of a model was called the encoder-decoder model where the encoder encodes the video into a semantic representation and then decodes this representation into natural language descriptions (single/dense captions) [205] [123] [138] [133] [207] [14]. This method did achieve limited success with respect to outputting proper descriptions from unseen videos, but the scale was restricted to small datasets with short videos having only smaller average running times per video or showcasing a single major event. This method could not consider other dynamics like more interleaving/non-interleaving events occurring during the time frame of the video. Lately however the trend has shifted towards developing models that can successfully capture and exploit the underlying temporal structures of the video along with the spatial dynamics so as to aid in generating comprehensive descriptions for videos. Modern models also try to focus on more relevant areas of information over time (attention mechanisms) to help make better predictions [214].

4.2 The Video Captioning Task

Video captioning emphasizing on capturing higher level visual concepts is not explored is not explored adequately, as the quality of the captions produced is highly dependent on the performance of the semantic tag extractor models. Among others, [207] explores a caption retrieval method where classifiers are trained on a predefined set of tags extracted from the captions of the training dataset using certain heuristics from linguistics and NLP. In contrast, our caption generation model focuses on the ability to capture concepts from within frames of videos in addition to what [207] does. The video captioning task can generally be modeled as the probability *P(w|v),* where w is a caption which is a sequence of words represented as $w = w_1, ...w_t$ and v is a sequence of frames from the video clip represented as $v = f_1, ...f_k$. Our work is based on the concepts discussed in [213] and tackles the video captioning task by breaking it down as two sub tasks which occur sequentially: 1) video feature extraction and tag predictions based on training captions and extracted visual features 2) caption generation based on the tags predicted by the tag prediction model. That is, mathematically the probability *P(w|v)* can be broken down into an equation involving two different probabilities conditioned on different inputs as expressed in Equation 4.1:

$$P(w|v) = \sum_c P_\theta(w|c)\, P(c|v) \tag{4.1}$$

where, $P(c|v)$ is the tag prediction model which predicts the higher-level attributes for video v, i.e. visual concept c from the video features extracted and $P_\theta(w|c)$ is the language model conditioned on the concepts extracted. The tag prediction model is a conditional tag model given visual inputs.

4.2.1 Video Feature Extraction and Semantic Tag Prediction

Video clips consist of variable number of frames. The feature extraction module combines the features extracted from 1) a pretrained VGG16 neural network [28] on ImageNet [203] and 2) a stacked model with three parallel 3D CNN models with a single classifier. The visually semantic tags, extracted by the tag prediction model is defined as those words in the captions that can be categorized either as actions, entities/objects or attributes of these entities. These three language components of a caption are considered deliberately as they are among the most common, visually perceivable categories for describing any visual content in natural language. The visual concept/tag could be defined as 1-gram, 2-gram, or as a mixture of N-gram representation to capture important and complicated correlations among visual concepts. But in our work, we will be considering only the 1-gram representation of the visual concepts. Since ground truth values for these visual concepts are not a part of the dataset considered, we generated our version of the ground truth for the tags by utilizing the Stanford log-linear part of speech tagger [182] to extract the visual concepts from the captions available and categorized them into the aforementioned word categories. That is, words extracted from the captions tagged with "NN", "NNP", "NNPS", "NNS" and "PRP" would be the entity tags, whereas the words tagged with "VB", "VBD", "VBG", "VBN", "VBP" and "VBZ" would be action tags and lastly the words tagged with "JJ", "JJR" and "JJS" would be attribute of the entity tags. In our work we have implemented two different neural network architectures to extract visual concepts as illustrated in Figure 4.3. The first neural model is a simple deep neural network (DNN) made of various dense layers that classify the visual concepts into either objects, actions and attributes whereas a second neural network is a bidirectional LSTM for predicting the overall tag (video classification) by making use of the extracted frame level features of the video. Lastly, a binary cross entropy loss function is computed over sigmoid outputs to predict the visual tags in a scenario which is highly analogous to the multi-label learning setting.

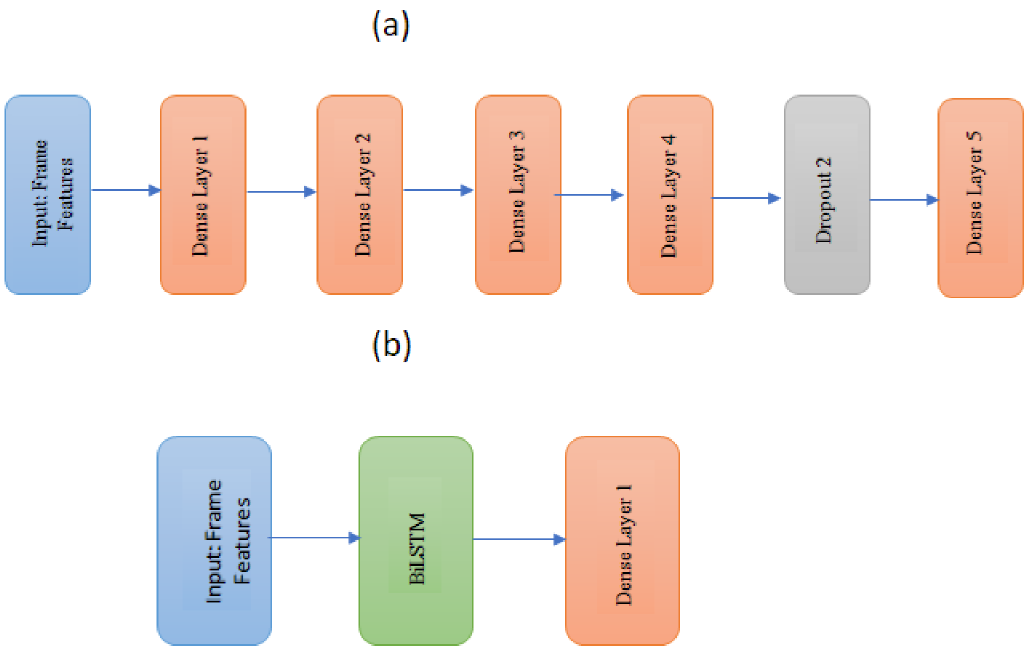


Figure 4.3 - (a) (above) DNN model for tag prediction. (b) (below) BiLSTM model that takes in every frame for tag detection.

4.2.2 The Caption Generation Language Model

The language model used is a simple two-layer stacked LSTM [47] model which is conditioned on the predefined set of visual concepts extracted from the tag prediction model (derived from the experience in [184][185][186]). The caption generation model takes in input from - the feature extractor, the two tag prediction models and the actual/ground truth output of the model at the previous time step (teaching forcing learning technique). The maximum number of words needed in the caption is already predefined (length of the caption is seventeen).

4.3 Experiments and Results

We make use of the YouTube2Text/Microsoft Video Description Corpus (MSVD) dataset [168], one of the earliest open domain datasets for our experiments which contains videos collected from YouTube by the Amazon Mechanical Turk (AMT) workers as well as the MSR-VTT dataset [170].

4.3.1 Datasets

MSVD. The MSVD dataset has a total of 1970 video clips covering a wide range of topics from sports to music and each clip has been captioned by the AMT workers in several languages besides English. If the English captions alone were to be considered, then there are forty parallel sentences per video, but only sixteen verified captions for each video clip, taking the total number of verified English captions for the whole dataset to 25,850 captions. There are about 85,550 unverified English captions associated with the whole dataset. A vocabulary of about 16k unique English words can be built using this dataset. We split the dataset, keeping 70% of them as training data while 30% as testing data. That is, the dataset was split into 1379 videos for training dataset (further split into 1229 training videos and 150 validation videos) and 591 test videos.

MSR-VTT. MSR-VTT is a widely used large-scale benchmark dataset for video captioning. It is a larger dataset than MSVD and of 10,000 video clips collected from YouTube covering a diverse set of 20 categories like sports (~ 784 videos), gaming (~332 videos), cooking (~ 232 videos), beauty and fashion (~341) etc. Each video has 20 human-annotated English captions. We again follow the 70-30% data split scheme. That is, 7000 video clips in training set (further split into 6500 training videos and 500 validation videos) and 3000 in testing.

4.3.2 Evaluation Metrics

In our experiments, we evaluate all the captioning models across two commonly used metrics for video captioning, namely METEOR [173], and CIDEr [172]. METEOR uses a uni-grams based weighted F-score and a penalty function to penalize incorrect word order, and it is claimed to have better correlation with human judgment. However, CIDEr is considered to be more robust to incorrect annotations as it adopts a voting-based approach. We evaluated their performances following the standard practice of using the Microsoft COCO Evaluation Server [180].

4.3.3 Implementation Details

Feature Extractor. The feature extraction module combines the features extracted from 1) a pretrained VGG16 neural network [28] on ImageNet [203] 2) a stacked model with three parallel 3D CNN models with a single classifier. The feature vector obtained from the average pooling layer of the flattened layer of the 2D CNN is 2048 while the feature vector obtained from the last global average pooling layer of the 3D stacked CNN architecture is 1024-dimension long vector for each frame. This experiment samples 40 key frames from the MSVD dataset while 80 key frames were sampled from the MSR-VTT dataset.

Language Decoder Network. The study makes use of a double stack LSTM network as the language decoder network. In the experiment, the LSTM model has hidden size of 512. We tune the hyper-parameters of our language model on the validation set. The study utilized the Adam optimizer with a fixed learning rate of 1×10^{-4} with no gradient clipping used. The models were trained using a batch size of 64 for 50 long epochs and early stopping was applied to get the best-performed model. TensorFlow framework was used for the development of the models. The training was conducted using two NVIDIA GTX-1080 Ti GPUs.

4.3.4 Experimental Results

The experiment was designed with an aim to keep the memory requirement on the lower side by making use of shallower models with comparatively less parameters to train. The second aim of the experiment was to achieve the goal of obtaining quality captions with simpler CNN and

RNN architectures, which were able to scale well and deliver stable results with bigger datasets as well (MSR-VTT).

Since the labels are unbalanced with a leaning bias towards the most common labels, both the tag prediction models use the micro average precision i.e. $\mu AP = \frac{\sum TP}{\sum TP + \sum FP}$, as means to evaluate their effectiveness, as illustrated in Table 4.1. The hyperparameters of the tag prediction models as well as the caption generation language model were optimized using the grid search algorithm.

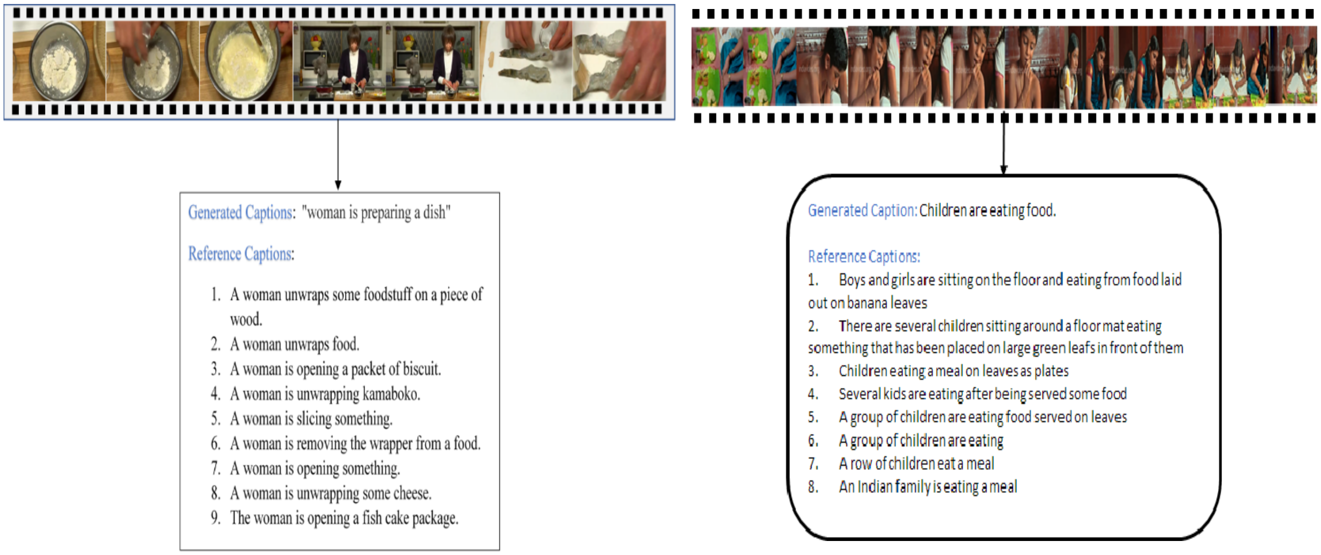
Table 4.1- The μAP scores for DNN and BiLSTM tag prediction models

MSVD Dataset		
Tag Prediction Model	**Tag Type**	**Micro Average Precision (μAP)**
DNN	Entity/Object	0.73
DNN	Action	0.57
DNN	Attribute	0.50
BiLSTM	Tag for entire video	0.73
MSR-VTT Dataset		
DNN	Entity/Object	0.60
DNN	Action	0.57
DNN	Attribute	0.53
BiLSTM	Tag for entire video	0.70

As for the captioning model, the METEOR [173] and CIDEr [172], evaluation metrics computed with Microsoft COCO Evaluation Server [180] were used to evaluate their performances. We used the ground truth tags with the captioning models, as well as the predicted tags generated by our two deep models and reported the captioning results using the actual tags and predicted tags separately. The gap in results clearly shows that the tag prediction model can be further improved upon.

Table 4.2 - METEOR and CIDEr results on captions generated

MSVD Dataset		
Caption Generation Model	**METEOR score**	**CIDEr score**
Two-layer stacked LSTM with ground truth tags	0.45	1.086
Two-layer stacked LSTM with tags predicted from DNN and BiLSTM models	0.316	0.69
MSR-VTT Dataset		
Two-layer stacked LSTM with ground truth tags	0.412	0.65
Two-layer stacked LSTM with tags predicted from DNN and BiLSTM models	0.29	0.59



Generated Captions: "woman is preparing a dish"

Reference Captions:

1. A woman unwraps some foodstuff on a piece of wood.
2. A woman unwraps food.
3. A woman is opening a packet of biscuit.
4. A woman is unwrapping kamaboko.
5. A woman is slicing something.
6. A woman is removing the wrapper from a food.
7. A woman is opening something.
8. A woman is unwrapping some cheese.
9. The woman is opening a fish cake package.

Generated Caption: Children are eating food.

Reference Captions:

1. Boys and girls are sitting on the floor and eating from food laid out on banana leaves
2. There are several children sitting around a floor mat eating something that has been placed on large green leafs in front of them
3. Children eating a meal on leaves as plates
4. Several kids are eating after being served some food
5. A group of children are eating food served on leaves
6. A group of children are eating
7. A row of children eat a meal
8. An Indian family is eating a meal

Figure 4.4 - Examples of quality captions generated by our model for MSVD dataset.

Table 4.3 compares the METEOR scores of our model against some of the existing work. Our model performs better than the plain encoder decoder model of [S2VT] and is comparable with few complex state-of-the-art models. If we were to consider the ground truth tags instead of the predicted tags, then our model outperforms the models considered from other works as illustrated in Table 4.2.

Table 4.3 - Comparison of the METEOR scores of our model with various works on MSVD dataset. All values reported as %.

MSVD Dataset		
Ref.No	**Models**	**METEOR score**
[205]	MA-LSTM	33.6
[124]	LSTM-YT	29.1
[123]	Sequence to sequence video to text (S2VT) (RGB-VGG)	29.8
[123]	Sequence to sequence video to text (S2VT) (RGB- VGG + Flow (AlexNet)	29.2
[122]	LSTM-E	31.0
[130]	LSTM-TSA	33.5
[206]	GRU-RCN	31.1
[138]	Hierarchical recurrent neural encoder (HRNE)	33.9
[133]	Hierarchical Recurrent Neural Networks (p-RNN) (Dense captions)	32.6
[207]	Long attention in LSTMs	33.4
[14]	3D CNN with temporal attention	29.6
Our Model		**31.6**

A major lacuna of the captioning model is that at times the caption is generated with the wrong gender. For instance, the model generated the caption: “a man is slicing a piece of food” instead of the referenced caption (ground truth): “a woman is slicing a vegetable”. Overall, the tag prediction models have strong detection results for various objects and actions and the caption model often produced good quality captions. Table 4.3 compares the METEOR score of our captioning model with other well-known works on MSVD dataset, while Table 4.4 compares the METEOR score of our captioning model with other known works on MSR-VTT dataset. Our results are better than the models in [122], [123], [14] and achieves comparable results with respect to models that implement attention mechanisms effectively like the models in [205], [130], [138],

[133] and [207]. This indicates the scope for further improvement in our tag prediction models, as it directly impacts the caption generation model. Figure 4.4 showcases some examples of quality captions generated by our framework on MSVD dataset, while Figure 4.5 showcases good quality captions generated by our framework on MSR-VTT dataset.

Table 4.4- Comparison of the METEOR scores of our model with various works on MSR-VTT dataset. All values reported as %.

MSR-VTT Dataset		
Ref.No	**Models**	**METEOR score**
[212]	V2T_Navigator	28.2
[141]	VideoLab	27.7
[207]	Multi-faceted attention	26.9
[208]	Alto	26.9
[209]	RUC-UVA	26.9
[210]	TDDF	27.8
[211]	PickNet	27.2
Our Model		**27.86**

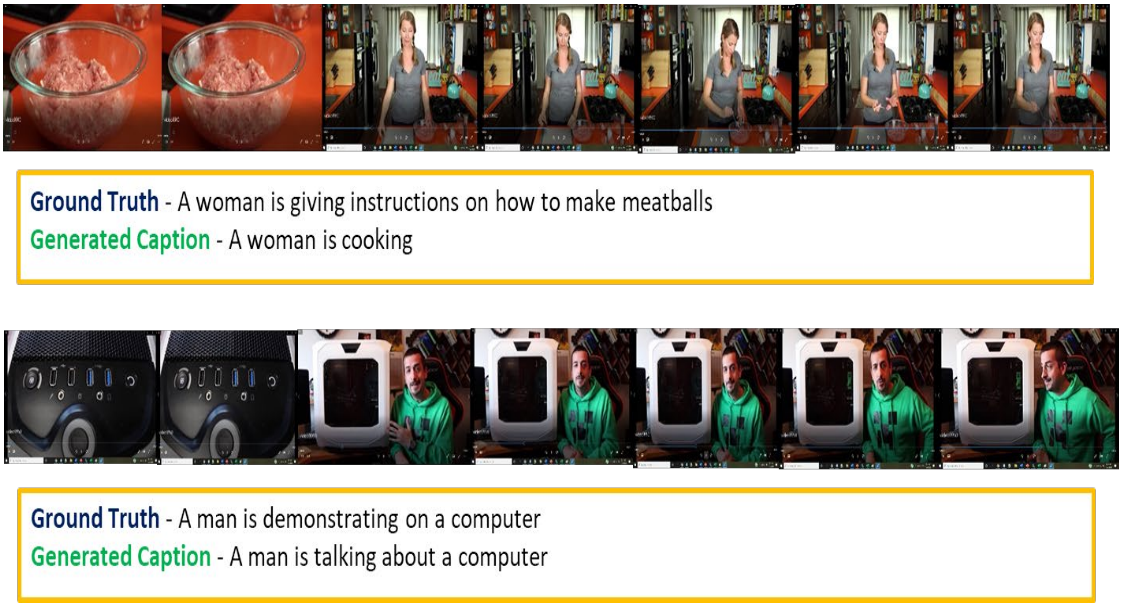

Ground Truth - A woman is giving instructions on how to make meatballs
Generated Caption - A woman is cooking
Ground Truth - A man is demonstrating on a computer
Generated Caption - A man is talking about a computer

Figure 4.5- Examples of quality captions generated by our model for MSR-VTT dataset

4.4 Summary

Video captioning is an open-ended complex research problem whose best performance is yet to catch up with the human-level captioning. This work attempts video captioning via two consecutive steps: 1) predicting semantic tags from the visual features and captions in the dataset, and 2) utilizing the extracted tags for generating single sentence captions for the videos. This framework enables us to provide an alternative path for improved captioning besides helping us gain valuable insight into the important factors responsible for the success/failure of the captioning models currently in vogue. The results (METEOR scores) indicate our model to be better, compared to the normal encoder-decoder architectures as well as the encoder-decoder architectures that exploit temporal information and models that employ joint embedding. Also, our results are closely comparable to the complex language models employing attention mechanisms. The micro average precision scores for the tag prediction models indicate scope for further improvement, which in turn, will further improve the quality of captions being generated.

5 UNRAVELING OF CONVOLUTIONAL NEURAL NETWORKS WITH BHARATANATYAM MUDRA CLASSIFICATION

Brief Abstract. Non-verbal forms of communication are universal, being free of any language barrier and widely used in all art forms. For example, in Bharatanatyam, an ancient Indian dance form, artists use different hand gestures, body postures and facial expressions to convey the story line. Bharatanatyam – a classical dance form which has origins from the southern states of India, is on the verge of being completely automated partly due to acute dearth of qualified and dedicated teachers/gurus. In an honest effort to speed up this automation process and at the same time preserve the cultural heritage, we have chosen to identify and classify the single hand gestures/*mudras*/*hastas* against their true labels by using variations of the convolutional neural networks (CNNs) that demonstrates the exceeding effectiveness of transfer learning irrespective of the domain difference between the pre-training and the training dataset and using CNN architectures like i) singular models, ii) ensemble models, and iii) few specialized models. This work is primarily aimed at 1) building a novel dataset of 2D single hand gestures belonging to 27 classes that were collected from Google search engine (Google images), YouTube videos (dynamic and with background considered) and professional artists under staged environment constraints (plain backgrounds). 2) exploring the effectiveness of Convolutional Neural Networks in identifying and classifying the single hand gestures by optimizing the hyperparameters 3) evaluating the impacts of transfer learning and double transfer learning, which is a novel concept explored in this paper for achieving higher classification accuracy. The cleansing of mislabeled data from the initial collected dataset was done through explored through two novel techniques: i) label transferring based on distance-based similarity metric using convolutional siamese neural network and ii) label assignment based on image class identification/classification using

autoencoders. Since the background in many frames are highly diverse, the acquired data is real and dynamic, compared to images from closed laboratory settings. This study achieved an accuracy of >95%, both in single and double transfer learning models, as well as their stacked ensemble model. The results emphasize the crucial role of domain similarity of the pre-training / training datasets for improved classification accuracy and, also indicate that doubly pre-trained CNN model yield the highest accuracy.

5.1 Introduction

Among other things, ancient India is well known for its distinct and rich forms of art and literature. Particularly to be mentioned are its different classical dance forms: *Bharatnatyam, Kathak, Kathakali, Kuchipudi, Manipuri, Mohiniattam* and *Odissi*. These dance forms were traditionally performed in places of worship like temples using well-choreographed *mudras* (hand gestures) to communicate the story line. These *mudras* as well as different body postures and facial expressions have been codified in the famous book '*Natya-shastra*' (Science of Dance), authored by the ancient sage Bharata Muni sometime during 200 B.C.E and 200 C.E. It may be noted that the very word *BhaRaTa* signifies the three essentials of this classical art form: '*Bha*' – from '*bhava*' or emotions (conveyed through *mudras*, /body gestures and facial expressions), '*Ra*' – from '*raaga*' – the melody (of the accompanying music), and '*Ta*' – from '*taala*' or rhythm. A depiction of a *Bharatanatyam* dancer in full costume and makeup is depicted in Figure 5.1b. The '*Natya-shastra*' is the holy bible for all forms of Indian theatre arts, especially '*natya* and *nritya*' (drama and dance) as illustrated in Figure 5.1a. Subsequently, many of these dance forms have been further improved and enriched by different artists from time to time. At present, despite its huge popularity, this dance form is experiencing a severe dearth of qualified teachers.

Bharatanatyam gives more importance to body postures and hand gestures - latter known as *hastas* or *mudras*. There is a total of 52 hand gestures, out of which, 28 are single hand gestures (*asamyukta hastas*), while the remaining 24 are double hand gestures (*samyukta hastas*). This study focuses on the more important single hand gestures with the aim to: 1) create a dataset of the 27 single hand gestures - as depicted in Figure 5.2 - from various sources (one is omitted as it requires more than one frame to depict the gesture which is out of the scope of this work). 2) classify the images into their respective classes using a deep learning architecture, primarily the CNN, to automatically learn features as opposed to the traditional image processing technique using image descriptors, 3) observe the effects of proper hyper-parameter selection by simple variation (trial and error method) as well as optimization methods (using GridSearch), 4) study the effect of transfer learning (single and double) when domains of the pre-training and training datasets are highly dissimilar and, 5) evaluate how the different singular and ensemble CNN architecture styles impact the accuracy of a given model in classifying the *mudras* correctly against their true labels.

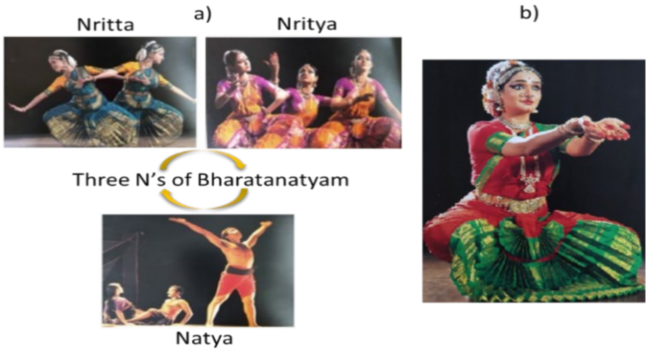


Figure 5.1 - a) The three N's of Bharatanatyam: Nritta – focuses of only the technical dance movements devoid of any facial expression i.e. pure dance or the various combinations of the foot movements in rhythmic patterns that do not associate themselves with or convey any meaning, Nritya– focuses on the expressions mainly facial expressions. The Nritya helps in portraying the different moods with highly stylized gestures, postures and body language, and Natya –which focuses on story telling. Pictures of dancers referenced from [215]. b) A depiction of a Bharatanatyam dancer in full costume and makeup as well as the musical instruments used in a Bharatanatyam recital, [215].

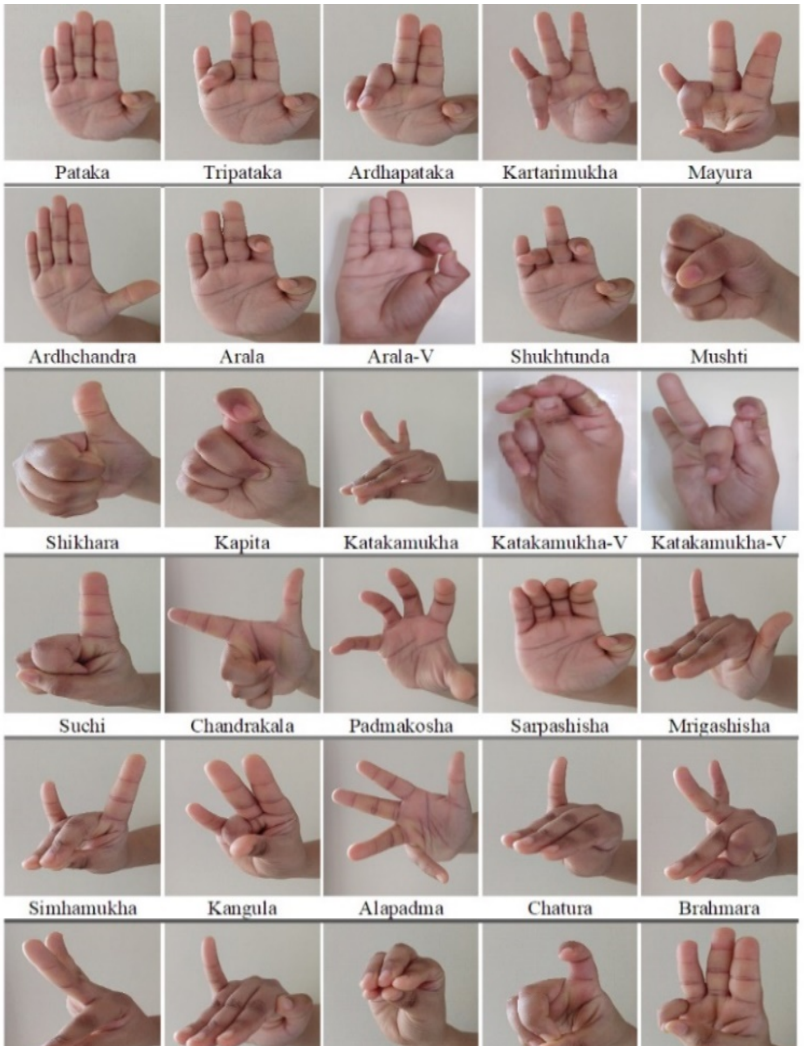


Figure 5.2 - All the twenty-seven single hand gestures/mudras/hastas used for the classification task. We have also considered a variation to arala hasta and two variations of the katakamukha hasta.

In recent years, lots of research are being carried out in the field of sign language [189-197] and hand gesture [198-202] recognitions using traditional image processing and machine learning techniques [187]. Earlier, traditional image processing techniques like moments, shape descriptors, etc., were used for feature extraction in sign language/hand gesture recognition. The classifiers used were either rule-based classifier or a Support Vector Machine (SVM), lacking any automatic feature learning techniques. These studies were less efficient, requiring tremendous effort for feature engineering, thus limiting their use to only small or medium size datasets. In other words, these studies could not transcend into the domain of deep learning, nor could they adequately explore the effectiveness of convolutional neural network (CNN) in solving complex multi-class classification problems [187]. Studies so far in the field of recognition of *mudras* in Indian classical dance, esp., *Bharatanatyam*, were done without using deep learning architectures or competitive datasets. For example, in [198], the authors proposed using vertical-horizontal intersections and type of *mudras* as features for the identification of double hand *mudras* from images. A rule-based classifier was developed, having an overall accuracy of 95.25%. The work in [199] followed an extensive approach using a combination of HOG features with an SVM classifier for classification of the *mudras* followed by comparison of their framework using the SIFT, SURF, LBP and HAAR features with the same classifier. The outcome of the study highlighted that a combination of HOG features with SVM classifier produced results with an accuracy of 90%. The work in [200] had a different approach, using a fuzzy network instead of resorting to the usual image processing techniques. The authors proposed a fuzzy L-membership function and a three-stage system for the recognition of various hand gestures in *Bharatanatyam*. Here, the first stage of the framework focused on obtaining the edges or outline of the individual images using the sobel edge detection operator. The second stage of the framework determined the center of the outline and calculated eight spatial distances. The final stage consisted of finding the similarity of the unknown hand gesture (a form of unseen image) by matching the hand gesture with existing images and calculating the fuzzy L membership for each distance. The model claimed an accuracy of 85.1%. The work in [202], combined the work in [198], [199] and [200] by developing a framework to work on the edges/outlines of the *mudras*, extracting their hu-moments, eigen values and horizontal / vertical intersection features using an artificial neural network (ANN) as classifier. This work had mentioned for the first time, the advantages of using a shallow convolutional neural network (CNN) for automatic recognition of *mudra*s, though not in detail. The dataset used in [202] was also not as comprehensive as compared to this study's dataset.

All previous studies [198-202] have a few things in common: 1) they focused more on traditional image processing techniques and hand crafted features 2) did not cover all the twenty-eight single hand gestures (*mudras*),but focus only on a small subset 3) very small dataset (fear of overfitting a CNN model) collected under a controlled environment setting by utilizing actors in a staged background (no dynamic background) making it really difficult to explain the effectiveness of convolutional neural networks and rendering these methods ineffective when used in a real-scenario setting like when the shots of the performance are taken outdoors against backgrounds of nature etc., particularly true for [202], and finally, 4) inadequate focus on deep neural architectures and lower classification accuracies obtained, giving motivation to this work. The present study is an earnest attempt for implementing facile automation of identification and classification of different single hand gestures (*mudras*) from a large repertoire of datasets, thereby helping in preserving for posterity the traditions, art and culture with the help of modern image processing tools. The major challenge in achieving this goal was the close resemblances between certain hand gestures, which could lead to potential misclassification of the gestures, especially while depending on shape descriptors. The problem could be more severe, as most of the traditional image processing techniques do not have the power to automatically enforce feature learning, with feature extraction mostly being hand engineered.

5.2 Data Acquisition, Pre-processing, Cleansing and Augmentation

The classification pipeline of the selected single-hand *Bharatnatyam mudras* using CNNs involves the following 5 stages: 1) data acquisition 2) data preprocessing 3) data cleansing 4) data augmentation 5) classification, as illustrated by Figure 5.3 and Figure 5.4 (from our works in

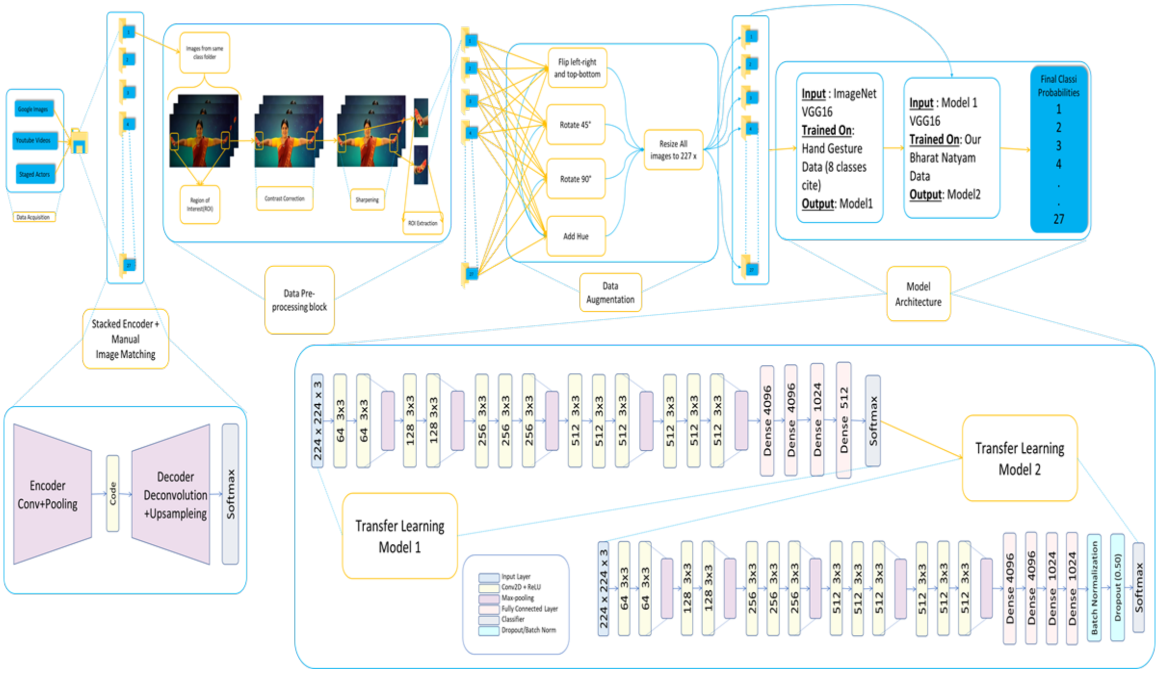


Figure 5.3 - Overall step-by procedures of data-transformation and final model architecture for single and double Transfer Learning [187]

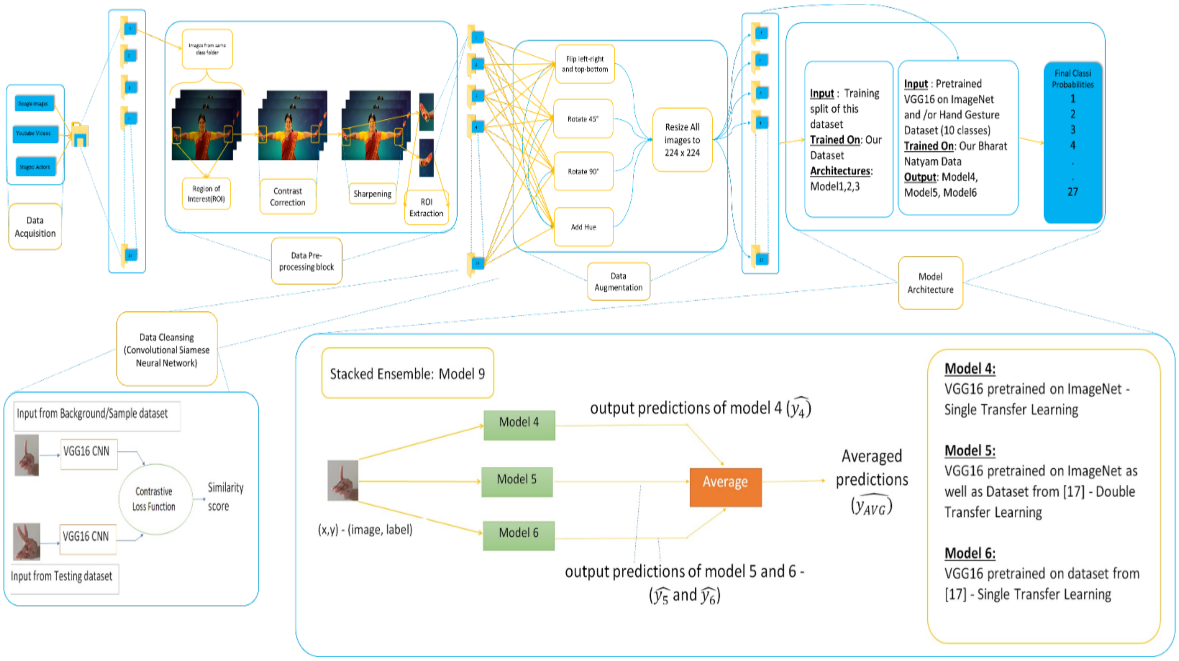

Google Images
Youtube Videos
Staged Actors
Data Acquisition
Images from same class folder
Region of Interest(ROI)
Contrast Correction
Sharpening
ROI Extraction
Data Pre-processing block
Flip left-right and top-bottom
Rotate 45°
Rotate 90°
Add Hue
Resize All images to 224 x 224
Data Augmentation
Input : Training split of this dataset
Trained On: Our Dataset
Architectures: Model1,2,3
Input : Pretrained VGG16 on ImageNet and /or Hand Gesture Dataset (10 classes)
Trained On: Our Bharat Natyam Data
Output: Model4, Model5, Model6
Final Classi Probabilities
1
2
3
4
.
.
27
Model Architecture
Data Cleansing (Convolutional Siamese Neural Network)
Input from Background/Sample dataset
VGG16 CNN
Contrastive Loss Function
Similarity score
VGG16 CNN
Input from Testing dataset
Stacked Ensemble: Model 9
output predictions of model 4 ($\widehat{y_4}$)
Model 4
Model 5
Model 6
Average
Averaged predictions ($\widehat{y_{AVG}}$)
(x,y) - (image, label)
output predictions of model 5 and 6 - ($\widehat{y_5}$ and $\widehat{y_6}$)
Model 4:
VGG16 pretrained on ImageNet - Single Transfer Learning
Model 5:
VGG16 pretrained on ImageNet as well as Dataset from [17] - Double Transfer Learning
Model 6:
VGG16 pretrained on dataset from [17] - Single Transfer Learning

Figure 5.4 - The complete classification pipeline showcasing an instance of the ensemble model, Model 9 [188]

[187][188] respectively). Since DL models require large amounts of data, it was necessary to collect varied data from as many sources as possible. The dataset collected was of medium size as

building a larger set would have been highly time-consuming and not needed for the task at hand [220]. The dataset is a novel dataset of single hand gestures belonging to the selected 27 categories (reason behind the 27 is because one hand gesture cannot be described by a single frame). The 2D images of every *mudra* were acquired by three different sources as mentioned earlier.

Data preprocessing is particularly useful while working with frames collected from YouTube video, where, most of the time, only part of the images was to be used. For example, the frame captures the whole dancer, but only the hand gesture region is required for proper classification. The data preprocessing module is also equipped with the traditional image processing techniques to segment bulky data and crop/save only the region of interest. In addition, it also improves the contrast of certain images using suitable smoothing functions and median filters. The acquired data required a good amount of cleansing mainly to 1) remove data not related to the dataset 2) properly classify mislabeled data or data with noisy label, so as to help in improving the quality of the training data. The above steps increase the classification accuracy of the model by improving the overall quality of training instances collected and at the same time minimizing the manual time required for weeding out the misfits. Thus, data cleansing mainly dealt with fixing mislabeled data. Google images of the mudras collected were often somewhat messy, necessitating some post processing cleansing. Part of it were also found to be either wrong or mislabeled/noisy data. Noisy labels cause major problems as correct set of training labels is a prerequisite for any supervised machine learning/deep learning techniques. We employed two different techniques for setting right the mislabeled data.

This study [187][188] focused on generating an automated solution using deep architectures to solve the problem of noisy labels by casting it as: i) an image classification problem by invoking convolutional autoencoders and ii) a one-shot recognition problem, thereby minimizing the requirements of manual supervision using convolutional siamese neural networks. Towards this end, we first attempted automation of the data cleansing process by treating it as an image classification problem, solving the same using a convolutional autoencoder architecture for image classification. We decided on using the convolutional autoencoder instead of a CNN primarily because of limited data. The training dataset in this case comprised of images collected only from actors and the YouTube video. The images collected from the google search engine are web labeled images rather than human labeled images. They are used because they are one of the fastest mediums to gather and construct big datasets but not necessarily accurate as the web search itself need not be accurate with possibility of noisy labels creeping in. Having human verifiers to verify the web labeled images is time consuming and expensive, which made us resort to a convolutional autoencoder to achieve the same.

The second technique used was convolutional siamese neural network for rectifying mislabeled data yielding better results as compared to [187]. The convolutional Siamese neural network drove its inspiration from the siamese neural network in [218] except for the replacement of the twin artificial neural networks (ANNs) which were replaced by twin convolutional neural network (CNN) architectures. The Convolutional siamese neural network was initially proposed by [217] for performing one-shot classification by developing first learning deep convolutional siamese neural networks for verification on the Omniglot character dataset [219]. The architecture for data cleansing process utilizes the applications of this network by choosing to cast it as a one-shot recognition problem instead of a standard classification problem for the mislabeled data cleaning process. While using a CNN for classification applications, the input (image/video) is generally fed to a convolutional layer and propagated forward into a series of layers. The output will be the corresponding class probabilities that better generalize the input. The training and testing datasets used are generally similar and not too different from each other. The effectiveness of machine learning (ML) techniques or deep learning architectures can be leveraged only with a huge amount of data available. The limitations of such methods become glaring with limited data or even unclassified data in the dataset, as retraining the entire network again could be an expensive task. One-shot learning is an effective and an inexpensive technique to achieve correct classification of wrongly classified data or even mislabeled (noisy label) data. The convolutional siamese neural network as in [217] for one shot learning classifying noisy labels was used in this study. This usage of a unique structure to automatically acquire features from the paired inputs, enabled successful generalization of the model even with limited examples. The work in [217] succeeded in performing one-shot classification by developing deep convolutional siamese neural networks for verification and subsequent comparison of their performances to an existing state-of-the-art classifier developed for an Omniglot data set. This study used almost the same optimizer and loss function as in [217], but with a few modifications to suit to the problem at hand: The major changes were: 1) using the rgb color channels instead of the singular channel, 2) skipping layer wise learning rates and momentum considerations, as hyper-parameter optimization was not the main focus for cleansing our image dataset in the present study 3) usage of the VGG16 pre-trained on ImageNet dataset as twin CNN architecture 4) usage of the dataset in [204] as well as our true positive images from the YouTube videos and staged actors, and 5) using lesser number of epochs (100 epochs) for network convergence. The one-shot classification in [217] used two images as training samples to the twin CNNs, to check whether the query images belonged to the same category/class. In other words, the convolutional siamese neural network learns the difference between the two images (to guess how similar they are) rather than simply focusing on classifying them and outputs a weighted similarity score. A convolutional siamese networks consists of two identical CNNs, each taking one of the two input images as illustrated in Figure 5.4. The last layer (encoded feature vector) of the twin networks are subsequently fed to a contrastive loss function (distance-based loss function for ensuring semantically similar examples are embedded close together) for necessary optimization and estimation of the similarities between the two images. Unlike the standard multi-class classification neural networks, normally using a cross entropy loss function, a different contrastive loss function was used here as the task at hand required an ability to differentiate between the paired input images and outputting a corresponding similarity score, which ultimately determines the accuracy of the label assigned to the particular query input image. The contrastive loss function is good for evaluating the response of the Convolutional Siamese Network in distinguishing a given pair of images. The training of such a network, usually called a verification model, identifies the input image pairs depending upon whether they belong to the same or perhaps different classes. Lastly, the verified model is used to evaluate new query images in a pairwise manner. The one-shot accuracy obtained after 100 epochs was 82.87%.

Data augmentation is essential to get bigger datasets, required for deep and wide neural networks (example-ResNet), capable of avoiding model overfitting and increasing the quantity of data collected. Before proceeding further, the following two questions need to be addressed: 1) why data augmentation, before training our deep model was important? and 2) whether the model could be trained directly using minimum data available? Training of a neural network is nothing but tuning its parameters or weights in such a way that it can map the input to the output optimally, minimizing the loss function. In the present study, the inputs and outputs are images and the class labels used for the classification. When the number of parameters is large (*e.g.*, in very deep models), the model needs to be trained with a proportional number of examples for enhanced performance. However, it is also possible to train a deep model with limited number of examples, in which case, one could hope to converge the model by employing transfer learning using a pre-trained deep model. By all means, augmentation helps in increasing the size of the relevant data present in the dataset. This step should thus be performed before training the deep model. We employed data augmentation as an essential step prior to training the deep model using the "offline data augmentation" method, to achieve dataset enlargement by a factor equal to the number of transformations performed. The flip, rotation, translation, gaussian noise and scale transformations were performed using TensorFlow data augmentation commands as well as the Keras Image generator to augment our existing datasets. After data augmentation the dataset has a decent total of: 18,992 images.

5.3 Image classification using deep learning framework: Convolutional Neural Network

Convolutional Neural Networks (CNNs) have been very useful for several tasks, primarily for image classification where it is already acclaimed as a state-of-the-art architecture. The remarkable efficiency of these networks is due to the fact that these networks do not require hand crafted features to perform classification. Instead they automatically learn features as it unravels. According to LeCun, CNNs can be used to recognize various objects by directly training the model on their images as they are robust architectures to scale, camera, viewpoint, noise, etc. Our study focused on different variations of the CNN model that were developed for classification of the single hand gestures of the *Bharatanatyam* classical dance form and concluded the following: 1) architecture depth - whether shallow or deep networks are better suited for our purpose 2) impact of hyperparameter optimization 3) impact of transfer learning with respect to the domain of the pre-training and the training dataset, and 4) ensemble *vs* singular architecture.

We have designed 1) singular CNN models closely emulating the VGG16 CNN model [28] and the ResNet model [216], 2) ensemble model using three different CNN models which were a combination of singularly pretrained and/or doubly pretrained CNN models 3) VGG16 like model that not only illustrated the importance of singularly pre-trained network while using ImageNet dataset [203] but also showcased the domain importance, highlighting the superior performance of the doubly pretrained networks and CNN models pre-trained on [204] instead of ImageNet [203] alone. The dataset [204] consists of a set of near infrared images acquired by the Leap Motion sensor which has 10 classes, collected using 10 different subjects (5 men and 5 women).

Table 5.1 gives a brief description of the models developed for the *mudra* classification task and their classification accuracy. Model 1 uses a VGG16 like architecture except for the presence of two additional dense layers and a dropout layer. Model 2 is similar to Model 1 but trained with more epochs. Model 3 uses the simple ResNet architecture following [216], not pre-trained on any dataset. Model 4 is VGG16 pre-trained on ImageNet and trained on our comprehensive dataset. Model 5 is VGG16 double pre-trained on both ImageNet [203] and later [204] before being trained on our dataset. Model 6 is VGG16, but pre-trained on [204] alone. Models 7, 8 and 9 are stacked ensemble models which takes the average output of all the three models used to create the ensemble. Model 7 stacks Models 1, 2 and 4. Model 8 stacks Models 1, 4 and 5. Model 9 stacks Models 4, 5 and 6. Hyperparameter optimization was achieved through the traditional GridSearch algorithm. Stochastic Gradient Descent (SGD) optimizer with initial learning rate of 0.0001 with step decay after every 5 epochs was used.

5.4 Results

The results of this study are summarized in Table 5.2. Table 5.2 compares the overall classification accuracy achieved while using Models 1-9 with published work [198-202] in this

Table 5.1 - Table showcasing type of models developed, the number of layers, epochs, validation loss/accuracy and training loss/accuracy

Model.No	Type_of_CNN	Layers	Epoch	Val_Loss	Val_Acc	Train_Loss	Train_Acc
Model 1	VGG16 + additional layers	20	100	0.874	84.61	0.0482	98.64
Model 2	VGG16 + additional layers	20	250	0.865	87.74	0.00039	99.89
Model 3	O_ResNet	54	250	0.5001	91.15	6.32e-04	99.97
Model 4	Transf_Img	20	70	0.213	94.56	0.0318	99.14
Model 5	Transf_Dob	20	20	0.084	98.25	0.0527	99.16
Model 6	Transf_Oth	20	70	0.0984	97.57	0.0492	99.16
Model 7	Ens_1	-	250	0.857	87.92	0.0661	99.83
Model 8	Ens_2	-	250	0.959	86.44	0.0076	99.78
Model 9	Ens_3	-	70	0.099	97.32	6.32e-04	99.97

domain (hand gestures). The ResNet model (Model 3) ensemble using the three different pretrained models (Model 9) as well as all the transfer learning models (Models 4-6) performed exceptionally well when compared to singular non-pretrained CNN networks. VGG16 model was chosen as the base as it consistently provided better results for all the hyperparameter combinations used. For Models 1 and 2, the variation in the kernels used in convolutional layers did not impact result significantly, though the number of filters in multiples of sixteen had to be specified in order to ensure better performance in GPU kernels.

Model 4 and Model 5 were recently published in [18]. The plot for models 2-6 is showcased in the plots below in Figure 5.5. Our studies clearly indicated that the classification accuracy of the double transfer learning model (Model 5) is the best, followed by the single transfer learning model (Model 6) which used the dataset in [204] as pre-training dataset. The stacked ensemble model using all the three pre-trained models (Model 9) had a greater classification accuracy as compared to the singular models or a combination of ensemble models. Surprisingly Model 8 had a lower classification accuracy than Model 7, perhaps due to close similarities in these two pre-training models used (The 3 models stacked up were Models 1, Model 4 and a minor variant of the latter, thereby, enforcing the belief that ensembles tend to yield better results, given significant diversity among the models used).

Table 5.2- Comparison of the overall classification accuracy achieved while using our models, Models 1-9 and also comparing them with existing works in the domain.

Ref.No	**Mapping of models**	**Classification accuracy**
[198]	-	95.25%
[199]	-	80% (SIFT, SURF, LBP, Haar features + SVM)
[199]	-	90% (HOG features + SVM)
[200]	-	85.1%
[201]	-	85.29%
[202]	-	94.71% (shallow CNN)
[202]	-	97.1%, 98% and 96.8% (Hu, eigen vector and intersection as features + ANN)
Our Models:		
Singular CNN models		
Model 1	~VGG16	84.61
Model 2	~VGG16	87.74
Model 3	ResNet	91.15
Transfer Learning (TL) Models		
Model 6	Transf_Oth – single TL using [204] as pretraining dataset	97.57
Model 4	Transf_Img – single TL using [203] as pretraining dataset	94.56% (Single transfer learning)
Model 5	Transf_Dob – double transfer learning using [203] as first pretraining dataset and [204] as second pretraining dataset	98.25% (Double transfer learning)
Ensemble Models		
Model 7	Ens_1 - One pretrained VGG16 in addition with model 1 and 2	87.92
Model 8	Ens_2 - Two pretrained VGG16 in addition with model 1	86.44
Model 9	Ens_3 - All three pretrained VGG16 models	97.32

thus, enforcing the belief that ensembles tend to yield better results, given significant diversity among the models used).

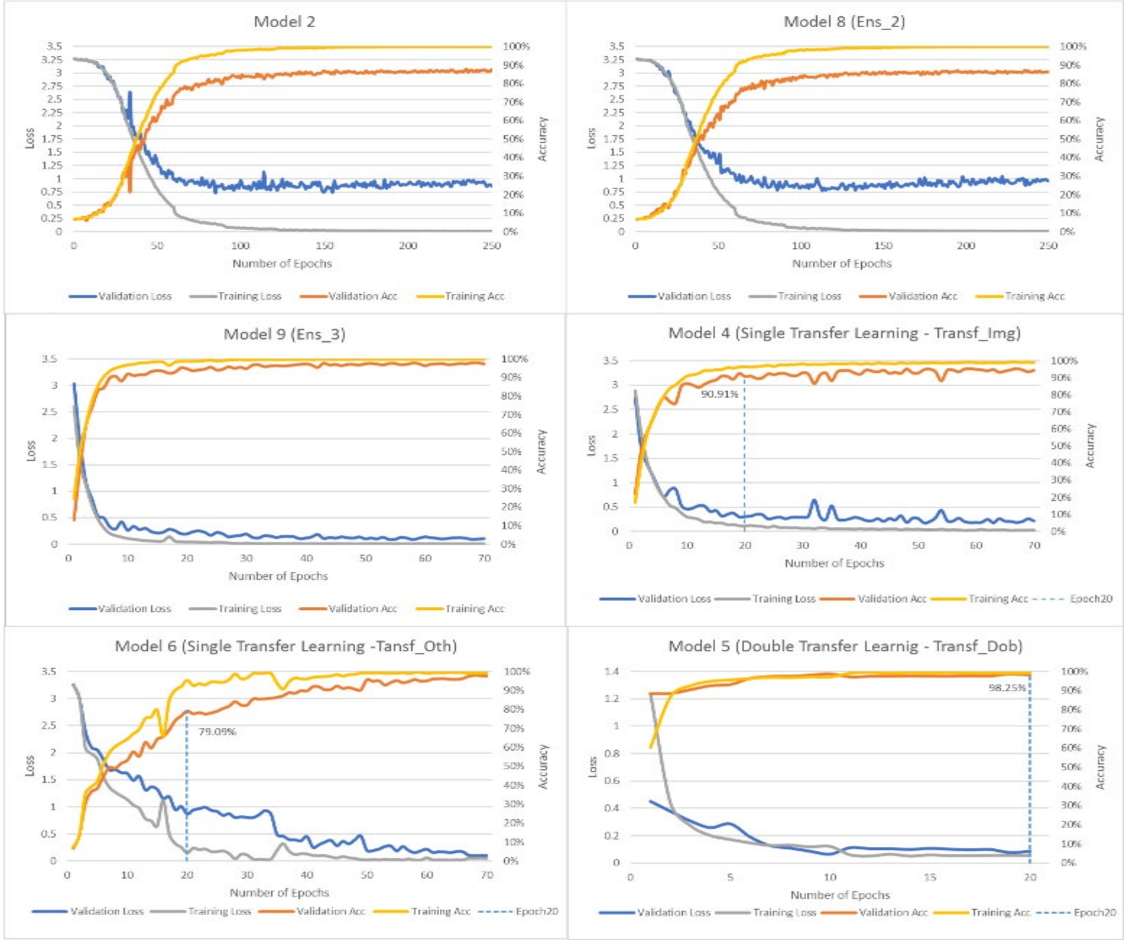


Figure 5.5 - The classification accuracy plots of Model 2, Model 8, Model 9, Model 5, Model 6 and Model 4 in left to right fashion (at every new row).

5.5 Summary

Convolutional neural networks (CNNs) have the inherent ability to automatically learn features without manual intervention and hence very popular among different deep learning architectures used in Computer Vision and Natural Language Processing applications. Along with data augmentation, transfer learning was also very effective, especially when the dataset size is limited. Stacked ensemble models performed better when there was a distinction between the models being used. The deeper the network, more complex features could be learned from the data. This concept was very important, especially as, the dynamic background included in the image dataset made it difficult to learn features of the objects using shallow CNNs alone. A higher classification accuracy was achieved when the domains of both pre-training and training datasets were similar. The domain and scale of the pre-training dataset played a crucial role in effective transfer learning. Using the above transfer learning techniques, our studies could achieve very high classification accuracy in most of the cases, with the highest value recorded being 98.25%. Further studies are required to use these techniques with necessary modifications, to explore the remaining mudras of this popular classical dance form. Indeed, visual recognition, encompassing vast areas of image/video recognition, detection, annotation/labelling, etc., continue to cause serious challenges to vision experts. In this regard, these studies constitute a small but significant step in facile implementation of e-learning techniques for visual recognition in general and the Indian classical dance, *Bharatanatyam*, in particular.

6 FUTURE WORK

This study explores different architectures and develop appropriate, memory efficient models to achieve deeper integration of the linguistic and visual semantics for describing a wide range of events in ordinary videos to natural language, irrespective of their visual domains. Current deep architectures, presently in use for video description tasks are of limited utility due to the following four main reasons: 1) They rely largely and solely on linguistic knowledge available in the paired image/video – sentence/sentences corpora 2) An architecture catering to a smaller dataset (like MSVD) fails to generalize and capture multiple event sequences for longer and diverse video datasets (like ActivityNet caption dataset) 3) As a consequence of 2), These methods fail to track and capture the dynamic interactions between the different subjects on the scene 4) Many of the works don't generate detailed and accurate linguistic descriptions of everyday scenes, particularly those involving larger datasets with intimate interactions, like the movie datasets. One of the promising research directions would be to integrate the results obtained with double pretraining in the visual recognition task into the visual description task and see if the tag prediction model can be further improved.

Additionally, a critical SWOT analysis of this work indicates that future research should concentrate on the following major areas further improving the findings of this study: (1) generation of diverse and accurate descriptions, (2) integration of prior linguistic knowledge, using double pre-trained 2D CNN and double pre-trained 3D CNN as feature extractors and incorporating soft attention methods, (3) improving focus interactions and (4) improvements in describing segmented and coherent event sequences useful in generating longer, multi-activity videos.

7 CONCLUSION

Translating the pixels from key frames of a video to natural language descriptions that describe its semantic content is gaining a lot of interest, especially during the past couple of years. The video captioning task enables several modern-day applications like video description services, human-robot interaction etc. This research dissertation focuses on surveying the different video captioning techniques, datasets and evaluation metrics currently in vogue with particular emphasis on two different tasks: 1) studying the suitability of CNN architecture to explore the classification of hand gestures used in the South Indian Classical Dance form: *Bharatnatyam* and 2) generating natural language descriptions that capture the semantics of activities depicted in diverse video corpora.

The work on *Bharatanatyam* single hand gesture recognition primarily explores, 1) different aspects in building a novel dataset of 2D single hand gestures belonging to 27 classes that were collected from Google search engine (Google images), YouTube videos (dynamic and with background considered) and professional artists under staged environment constraints (plain backgrounds). 2) the effectiveness of various Convolutional Neural Networks (singular, ensemble and pre-trained) in identifying and classifying the single hand gestures by optimizing the hyperparameters 3) evaluating the impacts of transfer learning and double transfer learning, which is a novel concept explored in this study for achieving higher classification accuracy.

The work in video captioning deals with generating natural language descriptions to describe the videos in MSVD and MSR-VTT datasets. This study approaches the task of video captioning by breaking down the problem into two portions: (1) predicting semantic tags using simpler, memory efficient neural architectures with lower parameters and (2) using the predicted semantic tags in the caption generated by the language decoder model. Two tag prediction models were introduced that were able to predict good quality tags, thus, the performance of the captioning model was also good. This study shows an interesting phenomenon, i.e., for the MSVD dataset comparable results were achieved while for a larger dataset like MSR-VTT the model performed better than most existing models. The uniqueness of this result is the stability the models, especially the tag prediction models bring to the video captioning task, while most architectures discussed in literature fail to scale up with the dataset. The result gotten so far showcases that this is a promising direction of work which can yield even better results than following a plain encoder-decoder construct.

REFERENCES

[1] Kakkar, Deepasha. 2018. “How Videos Generate Quick SEO Results.” Search Engine Watch. https://searchenginewatch.com/2018/02/19/how-videos-generate-quick-seo-results/.

[2] M. A. Gernsbacher, “Video captions benefit everyone,” Policy insights from the behavioral and brain sciences, vol. 2, no. 1, pp. 195–202, 2015.

[3] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank, “Automatic description generation from images: A survey of models, datasets, and evaluation measures,” Journal of Artificial Intelligence Research, vol. 55, pp. 409–442, 2016.

[4] R. Zhao and W. I. Grosky, “Bridging the semantic gap in image retrieval,” in Distributed multimedia databases: Techniques and applications. IGI Global, 2002, pp. 14–36.

[5] P. Wiriyathammabhum, D. Summers-Stay, C. Ferm¨uller, and Y. Aloimonos, “Computer vision and natural language processing: Recent approaches in multimedia and robotics,” ACM Computing Surveys (CSUR), vol. 49, no. 4, p. 71, 2017.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 9, pp. 1627–1645, 2010.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[8] C. H. Lampert, H. Nickisch, and S. Harmeling, “Learning to detect unseen object classes by between-class attribute transfer,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 951–958.

[9] T. L. Berg, A. C. Berg, and J. Shih, “Automatic attribute discovery and characterization from noisy web data,” in European Conference on Computer Vision. Springer, 2010, pp. 663–676.

[10] D. Parikh and K. Grauman, “Relative attributes,” in Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 503–510.

[11] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” International journal of computer vision, vol. 42, no. 3, pp. 145–175, 2001.

[12] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in Computer vision and pattern recognition, 2006 IEEE computer society conference on, vol. 2. IEEE, 2006, pp. 2169–2178.

[13] A. Prest, C. Schmid, and V. Ferrari, “Weakly supervised learning of interactions between humans and objects,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 3, pp. 601–614, 2012.

[14] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, “Describing videos by exploiting temporal structure,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 4507–4515.

[15] E. Reiter and R. Dale, Building natural language generation systems. Cambridge university press, 2000.

[16] Reiter, E. and Dale, R., 1997, “Building applied natural language generation systems,” Natural Language Engineering, vol. 3, no. 1, pp. 57–87, 1997.

[17] E. Kim, S. Helal, and D. Cook, “Human activity recognition and pattern discovery,” IEEE Pervasive Computing, vol. 9, no. 1, 2010.

[18] T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu, “epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition,” in Pervasive Computing and

Communications, 2009. PerCom 2009. IEEE International Conference on. IEEE, 2009, pp. 1–9.

[19] T. Dettmers, "Deep learning in a nutshell: Core concepts," URL: http://devblogs. nvidia. com/parallelforall/deep-learning-nutshell-coreconcepts/. [Zugriff: 02.01. 2016], 2015.

[20] R. D. Hof, "Deep learning–with massive amounts of computational power, machines can now recognize objects and translate speech in real time," Artificial intelligence is finally getting smart, 2015.

[21] A. Karpathy, "Connecting images and natural language," Ph.D. dissertation, Stanford University, 2016.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[23] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a backpropagation network," in Advances in neural information processing systems, 1990, pp. 396–404.

[24] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," arXiv preprint arXiv:1302.4389, 2013.

[25] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in International Conference on Machine Learning, 2013, pp. 1058–1066.

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[28] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich et al., “Going deeper with convolutions.” Cvpr, 2015.

[30] S. H. Hasanpour, M. Rouhani, M. Fayyaz, and M. Sabokrou, “Lets keep it simple, using simple architectures to outperform deeper and more complex architectures,” arXiv preprint arXiv:1608.06037, 2016.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.

[32] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning.” in AAAI, vol. 4, 2017, p. 12.

[33] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[34] He, K., Zhang, X., Ren, S. and Sun, J., 2016, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[35] He, K., Zhang, X., Ren, S. and Sun, J., 2016, October., “Identity mappings in deep residual networks,” in European Conference on Computer Vision. Springer, 2016, pp. 630–645.

[36] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, “Deep networks with stochastic depth,” in European Conference on Computer Vision. Springer, 2016, pp. 646–661.

[37] F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano, “Toward automatic phenotyping of developing embryos from videos,” IEEE Transactions on Image Processing, vol. 14, no. 9, pp. 1360–1371, 2005.

[38] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 1, pp. 221–231, 2013.

[39] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, 2013, pp. 3551–3558.

[40] Z. Wu, T. Yao, Y. Fu, and Y.-G. Jiang, “Deep learning for video classification and captioning,” arXiv preprint arXiv:1609.06782, 2016.

[41] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[42] K. Kawakami, “Supervised sequence labelling with recurrent neural networks,” Ph.D. dissertation, Ph. D. thesis, Technical University of Munich, 2008.

[43] D. Britz, “Recurrent neural networks tutorial, part 1–introduction to rnns,” 2015.

[44] T. Mikolov, “Statistical language models based on neural networks,” Presentation at Google, Mountain View, 2nd April, 2012.

[45] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in International Conference on Machine Learning, 2013, pp. 1310–1318.

[46] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An empirical exploration of recurrent network architectures,” in International Conference on Machine Learning, 2015, pp. 2342–2350.

[47] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[48] R. Dey and F. M. Salem, “Gate-variants of gated recurrent unit (gru) neural networks,” arXiv preprint arXiv, 2017.

[49] S. Krig, “Interest point detector and feature descriptor survey,” in Computer Vision Metrics. Springer, 2016, pp. 187–246.

[50] I. Laptev, “On space-time interest points,” International journal of computer vision, vol. 64, no. 2-3, pp. 107–123, 2005.

[51] C. Harris and M. Stephens, “A combined corner and edge detector.” In Alvey vision conference, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[52] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, “Evaluation of local spatio-temporal features for action recognition,” in BMVC 2009-British Machine Vision Conference. BMVA Press, 2009, pp. 124–1.

[53] P. Doll´ar, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on. IEEE, 2005, pp. 65–72.

[54] W. Kienzle, B. Sch¨olkopf, F. A. Wichmann, and M. O. Franz, “How to find interesting locations in video: a spatiotemporal interest point detector learned from human eye movements,” in Joint Pattern Recognition Symposium. Springer, 2007, pp. 405–414.

[55] G. Willems, T. Tuytelaars, and L. Van Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” in European conference on computer vision. Springer, 2008, pp. 650–663.

[56] F. Rezazadegan, S. Shirazi, N. S¨underhauf, M. Milford, and B. Upcroft, “Enhancing human action recognition with region proposals,” 2015.

[57] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008, pp. 1–8.

[58] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local svm approach,” in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 3. IEEE, 2004, pp. 32–36.

[59] P. Scovanner, S. Ali, and M. Shah, “A 3-d sift descriptor and its application to action application to action recognition,” in Proc. 15th Int. Conf. Multimedia, pp. 357–360.

[60] A. Klaser, M. Marszałek, and C. Schmid, “A spatio-temporal descriptor based on 3d-gradients,” in BMVC 2008-19th British Machine Vision Conference. British Machine Vision Association, 2008, pp. 275–1.

[61] L. Yeffet and L. Wolf, “Local trinary patterns for human action recognition,” in Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009, pp. 492–497.

[62] H. Wang, A. Kl¨aser, C. Schmid, and C.-L. Liu, “Dense trajectories and motion boundary descriptors for action recognition,” International journal of computer vision, vol. 103, no. 1, pp. 60–79, 2013.

[63] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1. IEEE, 2005, pp. 886–893.

[64] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.

[65] N. Dalal, B. Triggs, and C. Schmid, “Human detection using oriented histograms of flow and appearance,” in European conference on computer vision. Springer, 2006, pp. 428–441.

[66] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, “Sequential deep learning for human action recognition,” in International Workshop on Human Behavior Understanding. Springer, 2011, pp. 29–39.

[67] L. Wang, Y. Qiao, and X. Tang, “Action recognition with trajectory pooled deep-convolutional descriptors,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4305–4314.

[68] Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1110–1118.

[69] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, “Convolutional learning of spatio-temporal features,” in European conference on computer vision. Springer, 2010, pp. 140–153.

[70] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2625–2634.

[71] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in Advances in neural information processing systems, 2014, pp. 568–576.

[72] G. Yu and J. Yuan, “Fast action proposals for human action detection and search,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1302–1311.

[73] S. Karaman, L. Seidenari, and A. Del Bimbo, “Fast saliency based pooling of fisher encoded dense trajectories,” in ECCV THUMOS Workshop, vol. 1, no. 2, 2014, p. 5.

[74] Y. Ke, R. Sukthankar, and M. Hebert, “Event detection in crowded videos,” in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE, 2007, pp. 1–8.

[75] D. Oneata, J. Verbeek, and C. Schmid, “The lear submission at thumos 2014,” 2014.

[76] H. Pirsiavash and D. Ramanan, “Parsing videos of actions with segmental grammars,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 612–619.

[77] Y.-C. Su and K. Grauman, “Leaving some stones unturned: dynamic feature prioritization for activity detection in streaming video,” in European Conference on Computer Vision. Springer, 2016, pp. 783– 800.

[78] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia, “Temporal localization of fine-grained actions in videos by domain transfer from web images,” in Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015, pp. 371–380.

[79] L. Wang, Y. Qiao, and X. Tang, “Action recognition and detection by combining motion and appearance features,” THUMOS14 Action Recognition Challenge, vol. 1, no. 2, p. 2, 2014.

[80] J. Yuan, Y. Pei, B. Ni, P. Moulin, and A. Kassim, “Adsc submission at thumos challenge 2015,” in CVPR THUMOS Workshop, vol. 1, 2015, p. 2.

[81] S. Ma, L. Sigal, and S. Sclaroff, “Learning activity progression in lstms for activity detection and early detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1942–1950.

[82] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao, “A multistream bi-directional recurrent neural network for fine-grained action detection,” in Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on. IEEE, 2016, pp. 1961–1970.

[83] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei- Fei, “Every moment counts: Dense detailed labeling of actions in complex videos,” International Journal of Computer Vision, vol. 126, no. 2-4, pp. 375–389, 2018.

[84] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, “Automatic annotation of human actions in video,” in Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009, pp. 1491–1498.

[85] F. Caba Heilbron, J. Carlos Niebles, and B. Ghanem, “Fast temporal activity proposals for efficient detection of human actions in untrimmed videos,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1914–1923.

[86] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, “Daps: Deep action proposals for action understanding,” in European Conference on Computer Vision. Springer, 2016, pp. 768–784.

[87] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, “Sst: Single-stream temporal action proposals,” in Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, 2017, pp. 6373–6382.

[88] W. Chen, C. Xiong, R. Xu, and J. J. Corso, “Actionness ranking with lattice conditional ordinal random fields,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 748– 755.

[89] G. Gkioxari and J. Malik, “Finding action tubes,” in Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. IEEE, 2015, pp. 759–768.

[90] M. Jain, J. Van Gemert, H. J´egou, P. Bouthemy, and C. G. Snoek, “Action localization with tubelets from motion,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 740–747.

[91] J. C. van Gemert, M. Jain, E. Gati, C. G. Snoek et al., “Apt: Action localization proposals from dense trajectories.” in BMVC, vol. 2, 2015, p. 4.

[92] A. Yao, J. Gall, and L. Van Gool, “A hough transform-based voting framework for action recognition,” in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010, pp. 2061–2068.

[93] C. Sutton, A. McCallum et al., “An introduction to conditional random fields,” Foundations and Trends® in Machine Learning, vol. 4, no. 4, pp. 267–373, 2012.

[94] C. C. Tan, Y.-G. Jiang, and C.-W. Ngo, “Towards textually describing complex video contents with audio-visual concept classifiers,” in Proceedings of the 19th ACM international conference on Multimedia. ACM, 2011, pp. 655–658.

[95] Y.-G. Jiang, X. Zeng, G. Ye, D. Ellis, S.-F. Chang, S. Bhattacharya, and M. Shah, “Columbia-ucf trecvid2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching.” in TRECVID, vol. 2, 2010, pp. 3–2.

[96] A. Kojima, M. Izumi, T. Tamura, and K. Fukunaga, “Generating natural language description of human behavior from video images,” in Pattern Recognition, 2000. Proceedings. 15th International Conference on, vol. 4. IEEE, 2000, pp. 728–731.

[97] W. Hu, T. Tan, L. Wang, and S. Maybank, “A survey on visual surveillance of object motion and behaviors,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 34, no. 3, pp. 334–352, 2004.

[98] A. Kojima, T. Tamura, and K. Fukunaga, “Natural language description of human activities from video images based on concept hierarchy of actions,” International Journal of Computer Vision, vol. 50, no. 2, pp. 171–184, 2002.

[99] M. U. G. Khan, L. Zhang, and Y. Gotoh, “Human focused video description,” in Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE, 2011, pp. 1480–1487.

[100] A. F. Smeaton, P. Over, and W. Kraaij, “High-level feature detection from video in trecvid: a 5-year retrospective of achievements,” in Multimedia content analysis. Springer, 2009, pp. 1–24.

[101] A. Gatt and E. Reiter, “Simplenlg: A realisation engine for practical applications,” in Proceedings of the 12th European Workshop on Natural Language Generation. Association for Computational Linguistics, 2009, pp. 90–93.

[102] M. W. Lee, A. Hakeem, N. Haering, and S.-C. Zhu, “Save: A framework for semantic annotation of visual events,” in Computer Vision and Pattern Recognition Workshops, 2008. CVPRW’08. IEEE Computer Society Conference on. IEEE, 2008, pp. 1–8.

[103] M. U. G. Khan, L. Zhang, and Y. Gotoh, "Towards coherent natural language description of video streams," in Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE, 2011, pp. 664–671.

[104] N. Krishnamoorthy, G. Malkarnenkar, R. J. Mooney, K. Saenko, and S. Guadarrama, "Generating natural-language video descriptions using text-mined knowledge." in AAAI, vol. 1, 2013, p. 2.

[105] P. Hanckmann, K. Schutte, and G. J. Burghouts, "Automated textual descriptions for a wide range of video events with 48 human actions," in European Conference on Computer Vision. Springer, 2012, pp. 372–380.

[106] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi et al., "Video in sentences out," arXiv preprint arXiv:1204.2742, 2012.

[107] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, "Translating video content to natural language descriptions," in Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, 2013, pp. 433–440.

[108] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko, "Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zeroshot recognition," in Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, 2013, pp. 2712–2719.

[109] P. Das, C. Xu, R. F. Doell, and J. J. Corso, "A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching," in Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013, pp. 2634–2641.

[110] D. M. Blei and M. I. Jordan, “Modeling annotated data,” in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. ACM, 2003, pp. 127–134.

[111] P. Das, R. K. Srihari, and J. J. Corso, “Translating related words to videos and back through latent topics,” in Proceedings of the sixth ACM international conference on Web search and data mining. ACM, 2013, pp. 485–494.

[112] H. Yu and J. M. Siskind, “Learning to describe video with weak supervision by exploiting negative sentential information.” in AAAI. Citeseer, 2015, pp. 3855–3863.

[113] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele, “Coherent multi-sentence video description with variable level of detail,” in German conference on pattern recognition. Springer, 2014, pp. 184–195.

[114] R. Xu, C. Xiong, W. Chen, and J. J. Corso, “Jointly modeling deep video and compositional text to bridge vision and language in a unified framework.” in AAAI, vol. 5, 2015, p. 6.

[115] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, “Dense captioning events in videos,” in Proceedings of the IEEE International Conference on Computer Vision, vol. 1, no. 2, 2017, p. 6.

[116] Z. Shen, J. Li, Z. Su, M. Li, Y. Chen, Y.-G. Jiang, and X. Xue, “Weakly supervised dense video captioning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2017, p. 10.

[117] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, “Featurerich part-of-speech tagging with a cyclic dependency network,” in Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics, 2003, pp. 173–180.

[118] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779– 788.

[119] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien, “Unsupervised learning from narrated instruction videos,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4575–4583.

[120] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy, “What’s cookin’? interpreting cooking videos using text, speech and vision,” arXiv preprint arXiv:1503.01558, 2015.

[121] R. Pasunuru and M. Bansal, “Multi-task video captioning with video and entailment generation,” arXiv preprint arXiv:1704.07489, 2017.

[122] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, “Jointly modeling embedding and translation to bridge video and language,” 2016.

[123] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, “Sequence to sequence-video to text,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 4534–4542.

[124] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, “Translating videos to natural language using deep recurrent neural networks,” arXiv preprint arXiv:1412.4729, 2014.

[125] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, “High accuracy optical flow estimation based on a theory for warping,” in European conference on computer vision. Springer, 2004, pp. 25–36.

[126] R. A. Rivera-Soto and J. Ord´onez, "Sequence to sequence models for generating video captions."

[127] A. Shin, K. Ohnishi, and T. Harada, "Beyond caption to narrative: Video captioning with multiple sentences," in Image Processing (ICIP), 2016 IEEE International Conference on. IEEE, 2016, pp. 3364–3368.

[128] Y. Bin, Y. Yang, F. Shen, X. Xu, and H. T. Shen, "Bidirectional longshort term memory for video description," in Proceedings of the 2016 ACM on Multimedia Conference. ACM, 2016, pp. 436–440.

[129] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based lstm and semantic consistency," IEEE Transactions on Multimedia, vol. 19, no. 9, pp. 2045–2055, 2017.

[130] Y. Pan, T. Yao, H. Li, and T. Mei, "Video captioning with transferred semantic attributes," in CVPR, 2017.

[131] G. Li, S. Ma, and Y. Han, "Summarization-based video caption via deep neural networks," in Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015, pp. 1191–1194.

[132] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," journal of artificial intelligence research, vol. 22, pp. 457–479, 2004.

[133] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4584–4593.

[134] AA. Liu, N. Xu, Y. Wong, J. Li, Y.-T. Su, and M. Kankanhalli, “Hierarchical & multimodal video captioning: Discovering and transferring multimodal knowledge for vision to language,” Computer Vision and Image Understanding, vol. 163, pp. 113–125, 2017.

[135] J. Wang, W. Wang, Y. Huang, L. Wang, and T. Tan, “Multimodal memory modelling for video captioning,” arXiv preprint arXiv:1611.05592, 2016.

[136] A. Graves, G. Wayne, and I. Danihelka, “Neural turing machines,” arXiv preprint arXiv:1410.5401, 2014.

[137] Y. Y. H. K. J. Choi and G. Kim, “Video captioning and retrieval models with semantic attention.”

[138] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, “Hierarchical recurrent neural encoder for video representation with application to captioning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1029–1038.

[139] L. Baraldi, C. Grana, and R. Cucchiara, “Hierarchical boundary-aware neural encoder for video captioning,” in Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, 2017, pp. 3185–3194.

[140] J. Song, Z. Guo, L. Gao, W. Liu, D. Zhang, and H. T. Shen, “Hierarchical lstm with adjusted temporal attention for video captioning,” arXiv preprint arXiv:1706.01231, 2017.

[141] V. Ramanishka, A. Das, D. H. Park, S. Venugopalan, L. A. Hendricks, M. Rohrbach, and K. Saenko, “Multimodal video description,” in Proceedings of the 2016 ACM on Multimedia Conference. ACM, 2016, pp. 1092–1096.

[142] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng, “Semantic compositional networks for visual captioning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2017.

[143] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, “Sequence level training with recurrent neural networks,” arXiv preprint arXiv:1511.06732, 2015.

[144] S. Chen, J. Chen, Q. Jin, and A. Hauptmann, “Video captioning with guidance of multimodal latent topics,” in Proceedings of the 2017 ACM on Multimedia Conference. ACM, 2017, pp. 1838–1846.

[145] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang, “Video captioning via hierarchical reinforcement learning,” arXiv preprint arXiv:1711.11135, 2017.

[146] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, “Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation,” in Advances in neural information processing systems, 2016, pp. 3675–3683.

[147] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, “Feudal networks for hierarchical reinforcement learning,” arXiv preprint arXiv:1703.01161, 2017.

[148] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: a large video database for human motion recognition,” in Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 2556–2563.

[149] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 2929–2936.

[150] J. Liu, J. Luo, and M. Shah, “Recognizing realistic actions from videos in the wild,” in Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on. IEEE, 2009, pp. 1996–2003.

[151] H. Pirsiavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 2847–2854.

[152] M. S. Ryoo and L. Matthies, “First-person activity recognition: What are they doing to me?” in Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013, pp. 2730–2737.

[153] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, “The thumos challenge on action recognition for videos in the wild,” Computer Vision and Image Understanding, vol. 155, pp. 1–23, 2017.

[154] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 2. IEEE, 2005, pp. 1395– 1402.

[155] K. Soomro, A. R. Zamir, and M. Shah, “Ucf101: A dataset of 101 human actions classes from videos in the wild,” arXiv preprint arXiv:1212.0402, 2012.

[156] K. K. Reddy and M. Shah, “Recognizing 50 human action categories of web videos,” Machine Vision and Applications, vol. 24, no. 5, pp. 971–981, 2013.

[157] Y. Ke, R. Sukthankar, and M. Hebert, “Efficient visual event detection using volumetric features,” in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 1. IEEE, 2005, pp. 166–173.

[158] J. Yuan, Z. Liu, and Y. Wu, “Discriminative subvolume search for efficient action detection,” in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 2442–2449.

[159] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, “Modeling temporal structure of decomposable motion segments for activity classification,” in European conference on computer vision. Springer, 2010, pp. 392– 405.

[160] A. Gorban, H. Idrees, Y. Jiang, A. R. Zamir, I. Laptev, M. Shah, and R. Sukthankar, “Thumos challenge: Action recognition with a large number of classes,” in CVPR workshop, vol. 5, no. 6, 2015.

[161] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, “Thumos challenge: Action recognition with a large number of classes,” 2014.

[162] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, “A dataset for movie description,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3202–3212.

[163] A. Torabi, C. Pal, H. Larochelle, and A. Courville, “Using descriptive video services to create a large data source for video annotation research,” arXiv preprint arXiv:1503.01070, 2015.

[164] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, “A database for fine grained activity detection of cooking activities,” in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 1194–1201.

[165] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele, “Script data for attribute-based recognition of composite activities,” in European Conference on Computer Vision. Springer, 2012, pp. 144–157.

[166] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele, “Recognizing fine-grained and composite activities using hand-centric features and script data,” International Journal of Computer Vision, vol. 119, no. 3, pp. 346–373, 2016.

[167] Y. Huang and Y. Sun, “Datasets on object manipulation and interaction: a survey,” arXiv preprint arXiv:1607.00442, 2016.

[168] D. L. Chen and W. B. Dolan, “Collecting highly parallel data for paraphrase evaluation,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011, pp. 190–200.

[169] J. Cao, Y.-D. Zhang, Y.-C. Song, Z.-N. Chen, X. Zhang, and J.-T. Li, “Mcg-webv: A benchmark dataset for web video analysis,” Beijing: Institute of Computing Technology, vol. 10, pp. 324–334, 2009.

[170] J. Xu, T. Mei, T. Yao, and Y. Rui, “Msr-vtt: A large video description dataset for bridging video and language,” in Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on. IEEE, 2016, pp. 5288–5296.

[171] Papineni, K., Roukos, S., Ward, T. and Zhu, W.J., 2002, July. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics (pp. 311-318). Association for Computational Linguistics.

[172] Vedantam, R., Lawrence Zitnick, C. and Parikh, D., 2015. Cider: Consensus-based image description evaluation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4566-4575).

[173] Banerjee, S. and Lavie, A., 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization (pp. 65-72).

[174] Kuznetsova, P., Ordonez, V., Berg, T.L. and Choi, Y., 2014. Treetalk: Composition and compression of trees for image descriptions. *Transactions of the Association for Computational Linguistics*, *2*, pp.351-362.

[175] Olah, C., 2015. Understanding lstm networks.

[176] Britz, D., 2015. Recurrent neural network tutorial, part 4-Implementing a GRU/LSTM RNN with Python and Theano. *URL http://www. wildml. com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano*.

[177] Lin, C.Y., 2004, July. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74-81).

[178] Anderson, P., Fernando, B., Johnson, M. and Gould, S., 2016, October. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision* (pp. 382-398). Springer, Cham.

[179] Kusner, M., Sun, Y., Kolkin, N. and Weinberger, K., 2015, June. From word embeddings to document distances. In *International conference on machine learning* (pp. 957-966).

[180] Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P. and Zitnick, C.L., 2015. Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325.

[181] Thomason, J., Venugopalan, S., Guadarrama, S., Saenko, K. and Mooney, R., 2014, August. Integrating language and vision to generate natural language descriptions of videos in the wild. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (pp. 1218-1227).

[182] Toutanova, K., Klein, D., Manning, C., Morgan, W., Rafferty, A., Galley, M. and Bauer, J., 2000. Stanford log-linear part-of-speech tagger. The Stanford Natural Language Processing Group, Stanford University Std.

[183] Rohrbach, A., Rohrbach, M. and Schiele, B., 2015, October. The long-short story of movie description. In German conference on pattern recognition (pp. 209-221). Springer, Cham.

[184] Desai, H.P*., Parameshwaran, A.P*., Sunderraman, R. and Weeks, M., 2020. Comparative Study Using Neural Networks for 16S Ribosomal Gene Classification. Journal of Computational Biology, 27(2), pp.248-258.

[185] Desai, H.P*., Parameshwaran, A.P*., Sunderraman, R. and Weeks, M., "16S Ribosomal Gene Classification Using Recurrent Neural Network Models." 15th International Symposium on Bioinformatics Research and Applications (ISBRA) 2019.

[186] Desai, H.P., Parameshwaran, A.P., Sunderraman, R. and Weeks, M., December 2020, "Deep Ensemble models for 16S Ribosomal Gene Classification", 16th International Symposium on Bioinformatics Research and Applications Conference 2020 (ISBRA)

[187] Parameshwaran, A.P., Desai, H.P., Sunderraman, R. and Weeks, M., 2019. Transfer Learning for Classifying Single Hand Gestures on Comprehensive Bharatanatyam Mudra Dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0).

[188] Parameshwaran, A.P., Desai, H.P., Weeks, M. and Sunderraman, R., 2020, January. Unravelling of Convolutional Neural Networks through Bharatanatyam Mudra Classification with Limited Data. In 2020 10th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0342-0347). IEEE. "

[189] Solís, F., Martínez, D. and Espinoza, O., 2016. Automatic mexican sign language recognition using normalized moments and artificial neural networks. Engineering, 8(10), pp.733-740.

[190] Zadghorban, M. and Nahvi, M., 2018. An algorithm on sign words extraction and recognition of continuous Persian sign language based on motion and shape features of hands. Pattern Analysis and Applications, pp.1-13.

[191] Fagiani, M., Principi, E., Squartini, S. and Piazza, F., 2015. Signer independent isolated Italian sign recognition based on hidden Markov models. Pattern Analysis and Applications, 18(2), pp.385-402.

[192] Fernando, M. and Wijjayanayake, J., 2015. Novel approach to use HU moments with image processing techniques for real time sign language communication. Int. J. Image Process, 9, pp.335-345.

[193] Dixit, K. and Jalal, A.S., 2013, February. Automatic Indian sign language recognition system. In 2013 3rd IEEE International Advance Computing Conference (IACC) (pp. 883-887). IEEE.

[194] Premaratne, P., Yang, S., Zou, Z. and Vial, P., 2013, July. Australian sign language recognition using moment invariants. In International Conference on Intelligent Computing (pp. 509-514). Springer, Berlin, Heidelberg.

[195] Pradhan, A., Kumar, S., Dhakal, D. and Pradhan, B., 2016. Implementation of PCA for recognition of hand gesture representing alphabets. International Journal, 6(3).

[196] Adithya, V., Vinod, P.R. and Gopalakrishnan, U., 2013, April. Artificial neural network-based method for Indian sign language recognition. In 2013 IEEE Conference on Information & Communication Technologies (pp. 1080-1085). IEEE.

[197] Singha, J. and Das, K., 2013. Indian sign language recognition using eigen value weighted Euclidean distance-based classification technique. *arXiv preprint arXiv:1303.0634*.

[198] Anami, B.S. and Bhandage, V.A., 2018. A vertical-horizontal-intersections feature based method for identification of bharatanatyam double hand mudra images. Multimedia Tools and Applications, 77(23), pp.31021-31040.FirstName Alpher,, FirstName Fotheringham-Smythe, and FirstName Gamow. Can a machine frobnicate? Journal of Foo, 14(1):234–778, 2004.

[199] Kumar, K.V.V. and Kishore, P.V.V., 2017. Indian Classical Dance Mudra Classification Using HOG Features and SVM Classifier. International Journal of Electrical & Computer Engineering (2088-8708), 7(5). Authors. The frobnicatable foo filter, 2014. Face and Gesture 2014 submission ID 324. Supplied as additional material efg324.pdf.

[200] Saha, S., Ghosh, L., Konar, A. and Janarthanan, R., 2013, September. Fuzzy L membership function-based hand gesture recognition for Bharatanatyam dance. In 2013 5th International Conference and Computational Intelligence and Communication Networks (pp. 331-335). IEEE.

[201] Hariharan, D., Acharya, T. and Mitra, S., 2011, June. Recognizing hand gestures of a dancer. In International conference on Pattern Recognition and Machine Intelligence (pp. 186-192). Springer, Berlin, Heidelberg.

[202] Anami, B.S. and Bhandage, V.A., 2018. A Comparative Study of Suitability of Certain Features in Classification of Bharatanatyam Mudra Images Using Artificial Neural Network. Neural Processing Letters, pp.1-29.

[203] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. CVPR, 2009, pp. 248–255.

[204] T. Mantecón, C.R. del Blanco, F. Jaureguizar, N. García, “Hand Gesture Recognition using Infrared Imagery Provided by Leap Motion Controller”, Int. Conf. on Advanced Concepts for Intelligent Vision Systems, ACIVS 2016, Lecce, Italy, pp. 47-57, 24-27 Oct. 2016. (doi: 10.1007/978-3-319-48680-2_5)

[205] Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. 2017. Learning multimodal attention LSTM networks for video captioning. In Proceedings of the 25th ACM international conference on Multimedia. 537–545.

[206] Ballas, N., Yao, L., Pal, C. and Courville, A., 2015. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*.

[207] Xiang Long, Chuang Gan, and Gerard de Melo. 2018. Video captioning with multifaceted attention. Transactions of the Association for Computational Linguistics 6 (2018), 173–184.

[208] Shetty, R. and Laaksonen, J., 2016, October. Frame-and segment-level features and candidate pool evaluation for video caption generation. In *Proceedings of the 24th ACM international conference on Multimedia* (pp. 1073-1076).

[209] Dong, J., Li, X., Lan, W., Huo, Y. and Snoek, C.G., 2016, October. Early embedding and late reranking for video captioning. In *Proceedings of the 24th ACM international conference on Multimedia* (pp. 1082-1086).

[210] Zhang, X., Gao, K., Zhang, Y., Zhang, D., Li, J. and Tian, Q., 2017. Task-driven dynamic fusion: Reducing ambiguity in video description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3713-3721).

[211] Chen, Y., Wang, S., Zhang, W. and Huang, Q., 2018. Less is more: Picking informative frames for video captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 358-373).

[212] Jin, Q., Chen, J., Chen, S., Xiong, Y. and Hauptmann, A., 2016, October. Describing videos using multi-modal fusion. In *Proceedings of the 24th ACM international conference on Multimedia* (pp. 1087-1091).

[213] Yao, L., Ballas, N., Cho, K., Smith, J.R. and Bengio, Y., 2015. Oracle performance for visual captioning. arXiv preprint arXiv:1511.04590.

[214] Bilkhu, M., Wang, S. and Dobhal, T., 2019. Attention is all you need for Videos: Self-attention based Video Summarization using Universal Transformers. *arXiv preprint arXiv:1906.02792*.

[215] Pratibha Prahlad, "Bharatanatyam (Dances of India)," Wisdom Tree Publisher, April 30, 2009.

[216] K.He, X.Zhang, S.Ren, and J.Sun, "Deep residual learning for image recognition," In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778), 2016.

[217] G.Koch, R.Zemel, and R.Salakhutdinov, "Siamese neural networks for one-shot image recognition," In ICML deep learning workshop (Vol. 2), 2015.

[218] J.Bromley, I.Guyon, Y.LeCun, E.Säckinger, and R.Shah, "Signature verification using a" siamese" time delay neural network," In Advances in neural information processing systems (pp. 737-744), 1994.

[219] B.Lake, R.Salakhutdinov, J.Gross, J.and Tenenbaum, " One shot learning of simple visual concepts," In Proceedings of the annual meeting of the cognitive science society (Vol. 33, No. 33), 2011.

[220] H.W.Ng, and S.Winkler, "A data-driven approach to cleaning large face datasets," In 2014 IEEE International Conference on Image Processing (ICIP) (pp. 343-347). IEEE, October 2014.