A REGION BASED SEMANTIC COMPOSITION FRAMEWORK TO VISUAL IMAGE AND VIDEO EVENT SPECIFICATION

Thesis

Submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

DINESH NAIK

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025

JANUARY, 2023

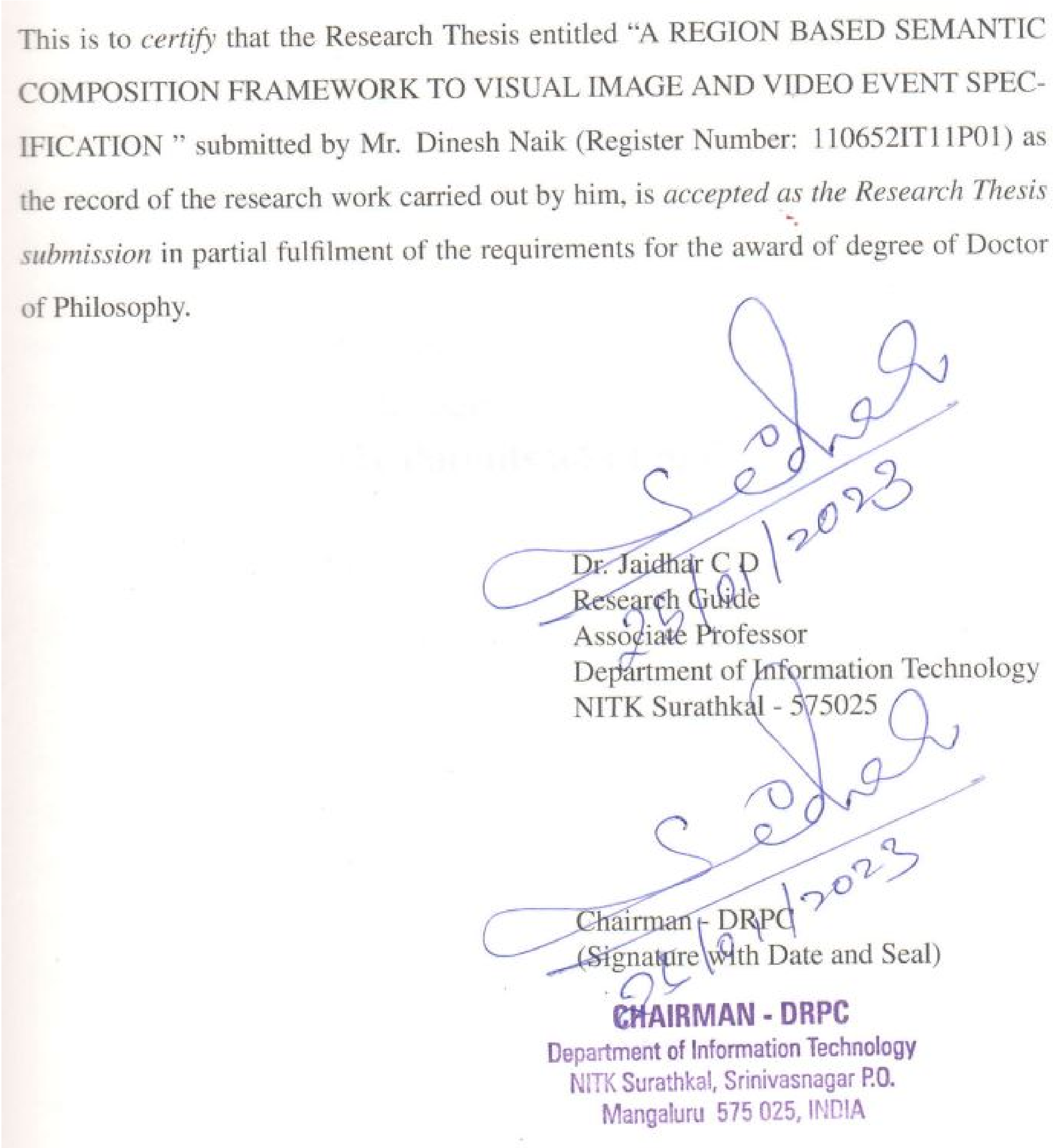
Declaration

I hereby *declare* that the Research Thesis entitled "A REGION BASED SEMANTIC COMPOSITION FRAMEWORK TO VISUAL IMAGE AND VIDEO EVENT SPECIFICATION" which is being submitted to the National Institute of Technology Karnataka, Surathkal in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy in Information Technology is a *bonafide report of the research work carried out by me*. The material contained in this thesis has not been submitted to any University or Institution for the award of any degree.

Place : NITK Surathkal
Date : 25-01-23

DINESH NAIK 25/01/23
Register No.: 110652IT11P01
Department of Information Technology

Certificate



This is to *certify* that the Research Thesis entitled "A REGION BASED SEMANTIC COMPOSITION FRAMEWORK TO VISUAL IMAGE AND VIDEO EVENT SPECIFICATION " submitted by Mr. Dinesh Naik (Register Number: 110652IT11P01) as the record of the research work carried out by him, is *accepted as the Research Thesis submission* in partial fulfilment of the requirements for the award of degree of Doctor of Philosophy.

25/01/2023

Dr. Jaidhar C D
Research Guide
Associate Professor
Department of Information Technology
NITK Surathkal - 575025

25/01/2023

Chairman - DRPC
(Signature with Date and Seal)

CHAIRMAN - DRPC
Department of Information Technology
NITK Surathkal, Srinivasnagar P.O.
Mangaluru 575 025, INDIA

This thesis is
dedicated to
My Parents and family

Acknowledgements

All prayer is directed first and primarily to Lord Shiva, the omnipotent, for your grace is sufficient. You stood by my side when no one else could stand me when the stress of my Ph.D. took its toll. I want to express my heartfelt gratitude to you, my Lord, for none of this would have been possible without your blessings.

I would like to thank my supervisor, Dr. Jaidhar C D, for his unmatched advice and support during my research journey. I shall be eternally thankful to you for motivating me, assisting me, and providing crucial insights during my Ph.D. path. I owe and am beholden to my advisor for his counsel and generosity. As an academician and researcher, Jaidhar Sir's foresight, curiosity, passion, and vision have proven inspirational. Over the course of our numerous meetings, his extensive expertise and remarkable ability to assess ambitious initiatives and point to the most appropriate foothold have aided in my development and growth as a researcher. I am indebted to him for allowing me the flexibility to follow my ideas and for regularly providing sound advise.

Additionally, I applaud Dr. Prakash S. Raghavendra, for introducing me to this developing field of study and motivating me to pursue doctoral studies.

I am grateful to my parents, Sri Krishna Naik and Smt Gopi, as well as my mother-in-law, Smt Susheela M. They have continually encouraged and supported me in both my personal and professional lives, and they have prayed for this without expecting anything in return. Again, my parents, you are my heroes in this life, and no words can express how deeply I am indebted to you for all your sacrifices and devotion. I hope life is good to me so that I can repay you even a bit. Many thanks!

I consider myself extremely fortunate to have learned from my wife, Rashmi M, and I appreciate her unwavering support, patience, and motivation. Special appreciation to her, who is the source of my strength and the most priceless gift the omnipotent has bestowed upon me. Without her caring and guiding, I would not be where I am now.

My brother and sisters, you occupy a particular place in my heart, and I appreciate

your never-ending willingness to provide a hand when assistance is needed. As Lord Shiva has given us with an incredible family, if we continue to remain together, we can move mountains.

I'd want to express my gratitude to my children, Twisha D N and Thejas D N, for their love and support. Without their sacrifices and, support none of this would have been possible.

I'd also like to express my gratitude to Profs. Ananthanarayana V S and Murulidhar N N for agreeing to serve on my RPAC committee and providing me with insightful feedback on this thesis, as well as for the talks we've had during our sessions.

My doctoral endeavour would not have been possible without the outstanding facilities provided by the National Institute of Technology Karnataka and the Department of Information Technology. All of this would not have been feasible without the Institute's and Department's assistance.

Finally, but certainly not least, I've been really lucky to learn from a variety of other incredibly remarkable individuals. I'd want to express my gratitude to all members of the Department of Information Technology's teaching and non-teaching personnel, as well as to my devoted graduate and postgraduate students. I've had the distinct pleasure of working alongside them and gaining knowledge from them.

(Mr. Dinesh Naik)

Abstract

A long-standing goal of artificial intelligence in Computer Vision has been to develop models capable of perceiving and comprehending the complex visual environment around us and communicating with us in natural language about it. Significant progress has been achieved toward this goal over the last few years as a result of parallel advancements in computing systems, data collection, and algorithms. Visual recognition has advanced at a breakneck pace, with computers now capable of classifying images, recognising them, and describing them in even longer words. They exceed humans in various categories, even surpassing them in some instances. Despite tremendous progress, the majority of improvements in visual recognition continue to occur when an image is labelled with one or a few different labels and swiftly explained in natural language. The majority of people find it straightforward to watch a brief video and describe what occurred (in words). Machines have a difficult time extracting meaning from video frames and generating a sentence description. Computer vision research has long been focused on comprehending visual media, such as images and videos. Additionally, a new issue within the scope of this study area, dynamic image and video transcription, has sparked the interest of a large number of people. This research presents models and methods for associating visual data with semantic labels and visual data with natural language utterances, thereby simplifying translation between domain constituents.

Semantic segmentation is a fundamental component of object recognition models, as it aims to classify things on a pixel-by-pixel basis. The primary goal of this research is to classify an individual object within an image pixel by pixel. The provided image is evaluated to ascertain the pixel-level properties that are present. Second, we suggested an encoder-decoder architecture with a hybrid loss function that employs a layered LSTM as the encoder and an LSTM model combined with an attention mechanism as the decoder. Thirdly, we propose a unique framework for video captioning that combines a bidirectional multi-layer LSTM encoder and a unidirectional decoder with a temporal attention technique to produce superior global representations for videos. Finally, we propose an efficient method for captioning videos using CNN in conjunction with a short-connected LSTM-based encoder-decoder model and a phrase context vector.

Keywords: Computer Vision; Object Detection; Semantic Segmentation; Object Recognition; Image/Video Captioning.

Contents

1 Introduction 1

1.1 Language and Vision 2

1.2 Early Progress in Video Captioning 2

1.3 Deep Neural Networks 3

1.3.1 Convolutional Neural Networks 3

1.3.2 Recurrent Neural Network 4

1.3.3 Long Short-Term Memory 5

1.4 Semantic Segmentation 6

1.4.1 Challenges 7

1.4.2 Contributions 7

1.5 Visual Captioning 7

1.5.1 Image Captioning 8

1.5.2 Video Captioning 9

1.5.3 Attention Guided Captioning 10

1.5.4 Challenges 11

1.5.5 Contributions 11

1.6 Word Embedding 12

1.7 Motivation 13

1.8 Overall Contributions 13

1.9 Organization of Thesis 14

2 Literature Survey 15

2.1 Semantic Segmentation 15

2.1.1 Traditional Methods 15

2.1.2 Region-Based Models 16

2.1.3 Fully Convolutional Network Based Models 16

2.1.4 Refinement Network 16

2.1.5 Weakly Supervised and Semisupervised Approaches 17

2.2 Visual Captioning 18

2.2.1 Deep Neural Networks 19
2.2.2 Recurrent Neural Network 20
2.2.3 Long Short-Term Memory 20
2.2.4 Attention Mechanism 21
2.2.5 Word Embedding 23
2.2.5.1 Count Based Techniques 24
2.2.5.2 Compositional Methods 24
2.2.5.3 Unsupervised Methods 24
2.2.5.4 Supervised Methods 24
2.2.5.5 Transformer Based Methods 25
2.2.5.6 Multimodal Approaches 25
2.3 Research Gaps Identified 27
2.4 Problem Statement 28
2.5 Research Objectives 28

3 Visual Image Data Analysis using Semantic Segmentation 29

3.1 Image Segmentation using Encoder-Decoder Architecture and RCA 29
3.1.1 Methodology 30
3.1.2 Experiments, Results and Discussion 33
3.1.2.1 Dataset 33
3.1.2.2 Experiments 33
3.1.2.3 Performance Analysis 35
3.1.3 Summary 35
3.2 Weaklier-Supervised Semantic Segmentation with Pyramid Scene Parsing Network 36
3.2.1 Methodology 36
3.2.1.1 Class Activation Mapping Model for Visual Interpretation 38
3.2.1.2 Accurate Superpixelization of Images 40
3.2.1.3 Generation of Multi Heat Map Slices Fusion . . . 42
3.2.1.4 Saliency-Edge-Color-Texture 44

3.2.2 Experiments, Results and Discussion 48
3.2.2.1 Dataset 48
3.2.2.2 Experiments 48
3.2.2.3 Performance Analysis 49
3.2.3 Summary 52
3.3 Weakly Supervised Image Annotation and Segmentation 52
3.3.1 Methodology 53
3.3.1.1 Feature Extraction 53
3.3.1.2 Model Selection 54
3.3.1.3 Object-Attribute Association and Image Segmentation 57
3.3.1.4 Combined Hierarchical Model 58
3.3.2 Experiments, Results and Discussion 60
3.3.2.1 Dataset 60
3.3.2.2 Experiments 60
3.3.2.3 Performance Analysis 66
3.3.3 Summary 68

4 Visual Video Data Analysis using Captioning **69**
4.1 Semantic Context Driven Language Descriptions of Videos using Deep Neural Network 69
4.1.1 Methodology 69
4.1.1.1 Preprocessing 71
4.1.1.2 Visual Encoder 72
4.1.1.3 Decoder 73
4.1.2 Experiments, Results and Discussion 76
4.1.2.1 Evaluations Metrics 76
4.1.2.2 Dataset 80
4.1.2.3 Experiments 80
4.1.3 Summary 92

4.2 A Novel Multi-Layer Attention Framework for Visual Description Prediction Using Bidirectional LSTM 93
4.2.1 Methodology 94
4.2.1.1 Preprocessing Unit 94
4.2.1.2 Proposed Multi-layer Attention Model 94
4.2.2 Experiments and Results 104
4.2.2.1 Dataset 104
4.2.2.2 Experiments 104
4.2.3 Summary 111
4.3 Video Captioning using a Sentence Vector-enabled Convolutional Framework with a Short-Connected LSTM. 112
4.3.1 Methodology 113
4.3.1.1 Preprocessing 113
4.3.1.2 Feature Extraction 114
4.3.1.3 Terminology and Notation 116
4.3.1.4 Encoder-Decoder 116
4.3.1.5 Contextual Hybrid Embedding Network (CHEN) 117
4.3.1.6 The Proposed Multi-head Attention Mechanism . 118
4.3.1.7 Training-Phase 119
4.3.2 Experiments, Results and Discussion 119
4.3.2.1 Experiments on MSVD Dataset 119
4.3.2.2 Experiments on MSR-VTT 124
4.3.3 Summary 128

5 Conclusions and Future Work **129**

References **134**

List of Tables

2.1 Summary of key existing works of image segmentation. 17

2.2 Summary of key existing works of video captioning. 25

3.1 Performance of the proposed method for scene labelling with and without RCA. 35

3.2 Comparison of proposed method for weaklier semantic segmentation using SLIC, MSF, SECT, and PSPNet with the state-of-the-art method. 50

3.3 Comparison of the performance of SIFT and SURF with and without threading 63

3.4 Comparison of the performance of proposed approach with other methods. 67

4.1 Sample input and output of proposed framework. 82

4.2 Evaluation of 2-layered and 3-layered LSTM in proposed framework using BLEU metrics. 83

4.3 Evaluation of 2-layered and 3-layered LSTM in proposed framework using METEOR, ROUGE, CIDEr and SPICE. 85

4.4 Failure cases: sample input and output given by the framework . . . 87

4.5 Evaluation of Model_3 using different λ hyper parameter. 88

4.6 Comparison of proposed layered LSTM method with existing video captioning methods. 89

4.7 Comparison of proposed framework with the state-of-the-art methods w.r.t METEOR and CIDEr score. 90

4.8 Performance of proposed multi-layer attention models on MSVD and MSR-VTT datasets. 107

4.9 Performance comparison of METEOR score with state-of-art-methods 109

4.10 Performance comparison of BLUE-4 Score with state-of-the-art-methods. 110

4.11 Performance score with tuned parameters. 110

4.12 Performance score with different Feature Extractor 111

4.13 Performance comparison of proposed framework on MSVD. 121

4.14 Comparison of proposed approach with the state-of-the-art methods on MSVD. 122

4.15 Performance of proposed video captioning using a Sentence Vector-enabled Convolutional Framework with a Short-Connected LSTM system on MSR-VTT. 125

4.16 Comparison of proposed approach with the existing method on MSR-VTT. 127

List of Figures

1.1 CNN architectural preview. 4

1.2 LSTM architectural preview. 5

1.3 The overview video captioning framework. 8

1.4 Overall taxonomy of image/video captioning. 10

1.5 Overall taxonomy of word embedding representations. 12

1.6 Organization of the thesis in line with the research objectives and contributions. 14

3.1 Architecture of the proposed approach for scene labelling using UCM, PRC, and RCA. 31

3.2 Workflow in the proposed approach for weaklier-supervised semantic segmentation. 37

3.3 System flow diagram with input image through different steps of proposed weaklier-supervised semantic segmentation: a) Input image, b) SLIC, c) SLIC slices, d) and e) heatmaps, f) LAB color and texture, g) edge maps, h) final output. 38

3.4 Architecture of ResNet-50 CAM model. 39

3.5 SLIC output for an image with K=100 and varying values of m: (a) Original image (b) m=1 (c) m=4 (d) m=10 (e) m=18 41

3.6 MSF flow diagram with input image, where (a) shows input image, (b) shows 1×1, 2×2 and 3×3 slicing, (c) shows 1×1, 2×2 and 3×3 fusion heatmaps, (d) shows the final combined heatmaps from MSF. 43

3.7 LAB color conversion of the image. 44

3.8 LBP Texture mapping of the image. 45

3.9 Canny edge mapping of the image. 45

3.10 Binarized saliency map generation of the image. 45

3.11 PSPNet Model: (a) input image, (b) feature is extracted using ResNet with diluted network, (c) After pooling for each feature, dimension reduction is done and after upsampling them concatenation is performed, (d) output image (Zhao et al., 2017). 47

3.12 Pixel Accuracy for the given input-1 is 0.673. (a) input image (b)predicted output (c) ground truth 51

3.13 Pixel Accuracy for the given input-2 is 0.629. (a) input image (b)predicted output (c) ground truth 51

3.14 Architecture of the proposed WS-MRF-CRP model for weakly supervised image segmentation. 53

3.15 SURF feature extraction. 54

3.16 Finite partition using Dirichlet (Wang and Zhao, 2017) 56

3.17 Graphical model of Dirichlet process (Guimarães et al., 2012) . . . 56

3.18 Illustration of CRP (Blei and Frazier, 2010) 57

3.19 The representation graph of 1^{st} and 2^{nd} order clique (Kato and Pong, 2006; Venmathi et al., 2019) 58

3.20 Relation between the component representation and the representation of cluster assignment in the hierarchical models (Kato and Pong, 2006; Venmathi et al., 2019) 59

3.21 An image segmented into 100, 150, 200 segments by super-pixeling 60

3.22 The stages of SURF algorithm 61

3.23 Sample images of different stages of proposed MRF-CRP model for weakly supervised image segmentationSegmentation 62

3.24 Result of SIFT + Colour 62

3.25 Result of SURF + Colour 62

3.26 Objects and their attributes shown together in the same color 64

3.27 Comparison between IBP and CRP. 64

3.28 Semantic Segmentation of the image with and without the CRP. First column on the left is the actual picture from the dataset, second one is without CRP and then the last is our actual methodology. 65

4.1 Proposed context driven video captioning framework. 70

4.2 LSTM unit (Sha et al., 2016). 72

4.3 Proposed framework of stacked/layered LSTM unit. 74

4.4 Plot of training loss versus number of epochs.: (a) 2-layered LSTM (b) 3-layered LSTM 80

4.5 Test samples: (a) Sample-1 (b) Sample-2 (c) Sample-3 (d) Sample-4 (e) Sample-5. 81

4.6 Failure cases: (a) Sample-1 (b) Sample-2 (c) Sample-3 86

4.7 Single loss versus hybrid loss. 88

4.8 Proposed multi-layer attention framework 95

4.9 LSTM Architectural Preview. 97
4.10 Bidirectional LSTM architecture. 98
4.11 Base line model architecture 101
4.12 Stacked LSTM model architecture 103
4.13 MSVD training loss at each epoch 107
4.14 MSRVTT training loss at each epoch 108
4.15 The proposed video captioning framework. 114
4.16 The predicted and reference samples of proposed approach. 123
4.17 Line chart comparison of pre-trained CNNs with BERT Embedding 128

List of Abbreviations

Abbreviation	Meaning	
AI	Artificial Intelligence	1
BERT	Bidirectional Encoder Representations from Transformers	21
BiLSTM	Bidirectional LSTM	13
BLEU	Bilingual Evaluation Understudy Score	77
BN	Bayesian Non-parametric	13
BoW	Bag of Words	24
BSD500	Berkeley Segmentation Dataset	54
CAM	Class Activation Mapping	36
CHEN	Contextual Hybrid Embedding Network	113
CIDEr	Consensus-based Image Description Evaluation	77
CNN	Convolutional Neural Network	3
CRP	Chinese Restaurant Process	29
CV	Computer Vision	1
DP	Dirichlet Process	55
ELMo	Embeddings from Language Models	24
FCN	Fully Convolutional Network	16
GAN	Generative Adversarial Network	21
GAP	Global Average Pooling	38
GCE	Global Consistency Error	66
GloVe	Global Vectors for word representation	69
GRU	Gated Recurrent Unit	11
IBP	Indian Buffet Process	58
IoU	Intersection over Union	49
LBP	Local Binary Pattern	44
LSTM	Long Short-Term Memory	3
METEOR	Metric for Evaluation of Translation with Explicit ORdering	77
MRF	Markov Random Fields	29
MSF	Multi Heat Map Slices Fusion	29

MSR-VTT	Microsoft Research Video to Text	104
MSVD	Microsoft Video Description	80
NASNet	Neural Architecture Search Network	70
NLP	Natural Language Processing	1
NMT	Neural Machine Translation	22
PRC	Probability of Regions Consistency	29
PRI	Probabilistic Rand Index	63
PSPNet	Pyramid Scene Parsing network	29
RCA	Region Consistency Activation	17
RCN	Residual Connection Network	113
RNN	Recurrent Neural Network	3
ROUGE	Recall-Oriented Understudy for Gisting Evaluation	77
SECT	Saliency Edge Color Texture	29
SGD	Stochastic Gradient Descent	32
SIFT	Scale Invariant Feature Transform	63
SLIC	Simple Linear Iterative Clustering	29
SPICE	Semantic Propositional Image Caption Evaluation	77
STAT	Spatial-Temporal Attention mechanism	26
SURF	Speeded Up Robust Feature	53
TFIDF	Term Frequency Inverse Document Frequency technique	79
UCM	Ultrametric Contour Map	29
WS-MRF-CRP	Chinese Restaurant Process Stacked with Weakly Supervised Markov Random Field	29

Chapter 1

Introduction

Videos are an incredibly rich and sophisticated source of information, accounting for the lion's share of internet resources. The growth of multimedia on the Internet in recent years has been astounding. For instance, around 95 million images are submitted to Instagram daily (Ceci, 2021), while approximately 400 hours of video are uploaded to YouTube every minute (Ceci, 2021). The exponential growth of visual data generated by this phenomenon presents both a tremendous problem and an opportunity to develop more intelligent computer algorithms for analysing and summarising the data.

For the majority of humans, multimedia content is intuitive, and images and videos are frequently used to augment and enhance human connection and communication (Bin et al., 2019; Olivastri et al., 2019; Zhao et al., 2019). When presented with a video, humans can deduce a great deal from it and analyse and describe the material in varied degrees of detail (Amirian et al., 2020). However, deciphering content from photos and video frames is extremely difficult for computers. The goal of language and vision research is to construct intelligent systems capable of autonomously analysing and comprehending this complicated visual input, as well as interacting and communicating in natural language. These algorithms may aid in the indexing and search of this visual data. A generic Artificial Intelligence (AI) system (Winston, 1992) would require an algorithm capable of recognising and describing various objects and their relationships in an image or video. This is a very intriguing and critical subject in the field of Computer Vision (CV) and AI.

This thesis examines the problem of characterising the content of images and videos from a fundamental standpoint. Semantic segmentation continues to be a significant challenge in the context of image and video understanding, as does image and video captioning, which combines CV and another branch of AI called Natural Language Processing (NLP) to generate sentence descriptions of an image or video. Video retrieval based on content, video segmentation and segment indexing, textual summaries of video clips, and video description for the partially sighted are just a few of the numerous important applications that can be accomplished when images and videos can be automatically translated into natural language.

1.1 Language and Vision

Both NLP and CV have advanced significantly in recent years, owing to revolutionary improvements in machine learning and the availability of large datasets. The two domains are fast overlapping: language is increasingly concerned with ”grounding” meaning in perception, while vision makes use of linguistic ontologies and attempts to “tell a story” through images by connecting things, activities, people, and situations. Until recently, there was a modest but growing corpus of work at the confluence of NLP and cognitive science on topics such as relating words to images, describing images in natural language, and interpreting natural language commands in terms of robot perception and action. Recent years have seen a huge growth in image/video captioning and retrieval efforts, owing to the availability of the MSCOCO (Lin et al., 2014) and Flickr30k (Young et al., 2014) large image/video captioning (Xu et al., 2016) datasets. Following this, datasets for image question answering were created. By comparison, video description has progressed more slowly.

1.2 Early Progress in Video Captioning

Video description, in particular for large vocabulary and activities, presents unique issues, such as modelling dynamics and actor-action-object correlations from sparse training data and coping with polysemy and ambiguity. The results regarding activity description in the video are limited to a small number of actions and objects. The majority of work on large-vocabulary description has concentrated on nouns/adjectives; early work on videos included metadata tagging and clustering captions and videos for retrieval tasks. Earlier work on video description relied on hand-crafted templates, grammar, and rules and was restricted to relatively small domains. (Barbu et al., 2012) and, (Yu and Siskind, 2013) for example, generate sentential descriptions for short videos but recognize only a limited collection of (5–10) objects and activities and generate a relatively narrow range of descriptive phrases using a manually engineered language. Numerous prior approaches to sentence description generation divided the problem into two sections. The initial step is content generation, during which they determine the most important objects that require description. The second method is surface realisation, in which they generate a sentence based on the content recognised. (Guadarrama et al., 2013) and (Krishnamoorthy et al., 2013), for example, employ a two-stage pipeline that first identifies the semantic content (subject, verb, object) and then creates a sentence from a template. (Krishnamoorthy et al., 2013) trained individual classifiers to identify candidate objects, actions, and scenes for the first time. They then employ an n-gram language model to select the most appropriate subject-verb-object combination for describing a video. This information is then used to construct a sentence. (Krishnamoorthy et al., 2013) examined a small number of videos having a subset of twenty things. (Guadarrama et al., 2013) described the first "in-the-wild" videos with extensive vocabulary. They demonstrated the benefit of linguistic expertise, but only for "zero-shot activity recognition," in which the right verb to describe the activity was never seen during training.

1.3 Deep Neural Networks

A potential disadvantage of all early image and video description algorithms was that scaling them required pre-selection and development of classifiers for a wide variety of objects, activities, and scenes, as well as the development of methods for identifying salient objects worth describing. Advances in deep learning, particularly in deep Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), have aided in the resolution of these issues. The approaches presented in this thesis are based on both deep CNN and RNN. As a result, we provide a brief review of both CNN and RNN based on their use in this thesis. Our proposed video description models are constructed using deep neural networks, specifically CNN for visual data modelling and Long Short-Term Memory (LSTM) units for language modelling. While CNN excel in object recognition, LSTM has lately exhibited higher performance on tasks such as speech recognition, machine translation, and the more related task of generating phrase descriptions for images. This section will provide an overview of CNN and RNN, more specifically LSTM networks, with an emphasis on sequence modelling.

1.3.1 Convolutional Neural Networks

CNNs are specialised neural network architectures that are optimised for processing and handling input with a fixed spatial topology, such as images or fixed-length word sequences. A CNN's purpose is normally to develop a spatially invariant representation of the input, which is advantageous for classification tasks. This is accomplished through the use of neural net topologies comprised of convolutional and pooling layers, as detailed below. Layer of convolution: A convolutional layer's objective is to automatically learn the weights of convolutional filters that may be used to detect local features such as edges and curvatures. Typically, a CNN's input is a multi-dimensional array (or a tensor). It would be a $256 \times 256 \times 3$ tensor in the case of colour images.

A filter is a collection of weights or parameters; for example, it could be 5 × 5 × 3. The filter is convolved over the full input tensor (mathematically) to generate an activation map for the CNN to learn local features. Intuitively, the filter weights aid the model in “learning” or “looking for” local features that (would appear in the activation map and) could aid in subsequent tasks. Each convolutional layer learns a collection of filters, not just one. Back-propagation is used to determine the weights and characteristics of these filters. Pooling layer: Pooling layers are frequently employed in CNNs to reduce the size of the convolutional layer’s learnt representation by implementing a fixed down sampling operation. While spatial down sampling results in the loss of some local spatial information, it ultimately aids the network in learning a decent condensed representation of the input.

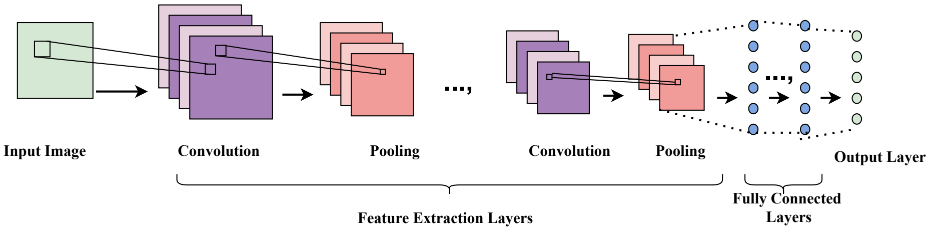


Figure 1.1: CNN architectural preview.

The majority of CNN architectures begin with numerous convolutional layers and end with a pooling layer. This pattern can be repeated indefinitely to generate stacks of convolutional and pooling layers, culminating in fully linked layers and a final classification target, as illustrated in Figure 1.1. The activations could serve as a condensed image representation just before the classification layer. Back-propagation is used to train the network’s parameters.

1.3.2 Recurrent Neural Network

A RNN is a feed-forward neural network that generalises feed-forward neural networks to sequences. Standard RNN learn to map a sequence of inputs $(x_1..., x_t)$ to a sequence of hidden states $(h_1, ..., h_t)$, and then from the hidden states to a sequence of outputs $(z_1, ..., z_t)$, based on the following recurrences:

$$h_t = f\left(W_{xh}x_t + W_{hh}h_{t-1}\right) \tag{1.1}$$

$$z_t = g\left(W_{zh}h_t\right) \tag{1.2}$$

where, f and g are non-linear element-wise functions such as the sigmoid or hyperbolic tangent, x_t is a fixed-length vector representation of the input, $h_t \in \mathbb{R}^N$ is the hidden state with N units, W_{ij} are the weights linking the layers of neurons, and z_t is the output vector.

1.3.3 Long Short-Term Memory

RNNs can be used to train to map sequences when the alignment of the inputs and outputs is known in advance, it was unclear whether they could be used to solve situations where the inputs (x_i) and outputs (z_i) have changing lengths. This challenge was handled by training an RNN to map input sequences to a fixed-length vector and then mapping the vector to an output sequence using another RNN. This is often known as the "encoder-decoder" framework in popular parlance. Another well-known issue with RNNs is that training them to acquire long-range dependencies can be tough. However, it is well established that LSTMs, which feature intentionally programmable memory units, may learn long-range temporal relationships.

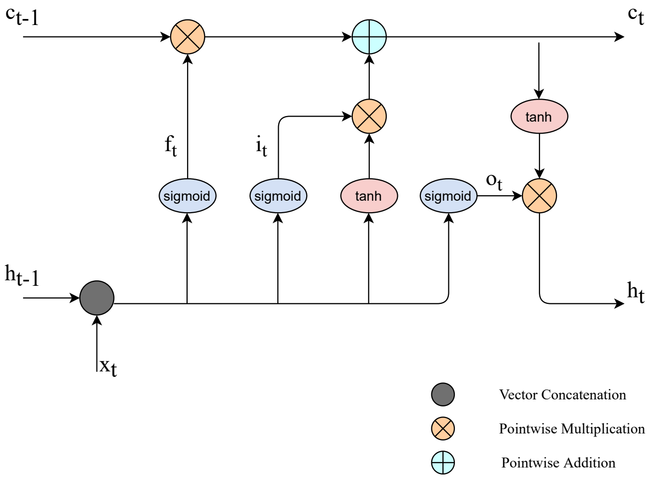


Figure 1.2: LSTM architectural preview.

The Figure 1.2, depicts the LSTM unit. The LSTM model is built around a memory cell c that encodes each input, creating a condensed representation of the sequence of observed inputs up to that point. The cell is modulated by sigmoidal gates with a range of [0, 1] that are applied multiplicatively. The gates control whether the LSTM retains (if the layer evaluates to 1) or discards the incoming value from the gate (if it evaluates to 0). The three gates – input gate (i) which determines whether the LSTM considers its current input (x_t), forget gate (f), which allows the LSTM to forget its previous memory (c_t), and output gate (o), which determines how much memory to transfer to the hidden state (h_t) – all enable the LSTM to learn complex long-term dependencies. The LSTM recurrences are therefore defined as follows:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1}), \tag{1.3}$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1}), \tag{1.4}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1}), \tag{1.5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \phi(W_{cx}x_t + W_{ch}h_{t-1}), \tag{1.6}$$

$$h_t = o_t \odot \phi(c_t), \tag{1.7}$$

- W_{*h} : Weights that relate each gate in the LSTM to previous hidden states.
- W_{*x} : Weighing units that connect the current input to each gate.
- σ : Sigmoid nonlinear activation functions.
- ϕ : Hyperbolic tangent nonlinear activation functions.
- $\odot$: Represents the operation of element-by-element multiplication.

Thus, the LSTM's gates enable it to represent a sequence through the acquisition of long-term dependencies. Thus, LSTMs are capable of “encoding” a sequence of inputs into a vector and “decoding” the vector to generate a sequence of outputs.

1.4 Semantic Segmentation

Semantic segmentation (Liu et al., 2019; Li et al., 2015) is the process of classifying images at the pixel level, such that each pixel in an image is assigned to a distinct class cluster. Since its introduction, deep learning semantic segmentation has been a critical field of image processing and CV research and application in a variety of domains. While image segmentation recognises the boundaries between objects in an image by categorising them using line and curve segments, instance segmentation classifies instances of all possible classes in an image so that each object is identified as a distinct entity. Nonetheless, semantic segmentation is distinct from conventional segmentation, which, on the one hand, only expresses the dividing of an image into clusters without making any attempt to comprehend or relate the partitioned clusters. On the other hand, semantic segmentation attempts to describe semantically meaningful objects in an image based on their well-defined relationship and comprehension.

1.4.1 Challenges

Semantic segmentation is a critical issue in the field of CV today. Semantic segmentation is a high-level activity that lays the way for total scene comprehension. The fact that an expanding number of applications rely on inferring knowledge from imagery emphasises the importance of scene understanding as a fundamental CV problem. Self-driving vehicles, human-computer interaction, and virtual reality are just a few of the applications. The visual system of a human being is remarkable in its ability to display a specific object in a scene readily. Additionally, it can provide a full description of a scene's contents based on a single glimpse at an image. For the last five decades, the fundamental goal of vision task research has been to replicate the ability of the human visual system. This has resulted in an exhaustive in-depth examination of challenges based on the following:

- Identifying the object - Object Detection
- Recognizing image contents with attributes - Object Annotation
- Using the attributes to describe the object - Attribute prediction and association

1.4.2 Contributions

- Design and implementation of an encoder-decoder neural network architecture that is paired with a novel technique for enhancing global label consistency, in order to produce an enhanced semantic picture segmentation model.
- Design and implementation of a hierarchical model that combines specialised local level methods with a global level procedure in order to accurately discover numerous main objects with uninitialized parameters in a poorly labelled image.
- Design and implementation of a task for classifying a single object from an image at the pixel level. The input image is processed to extract pixel-level information, and the object in the image is then demarcated and segmented for identification purposes.

1.5 Visual Captioning

When it comes to providing a decent caption for a video clip that is free of many semantic errors or misconceptions, humans are pretty capable. Humans can easily explain it and may be quite accurate in their portrayal. However, identifying the objects involved and predicting their associated relationships is always a arduous task for computers. Image captioning primarily started with traditional retrieval and template-based methods.

However, though they remained reliable and accurate at that time, those methods may not be too relevant nowadays. The reason is the advancement of deep learning and the tremendous advantages of NLP (Amirian et al., 2020). With the advent of CNNs and other their improved variants, it became significantly easier to analyse visual frames and anticipate the most appropriate sentences.

Visual captioning, alternatively referred to as video/image captioning, is the process of creating a description/caption for a video/image. The caption/sentence defines the items and actions in the image or video succinctly and precisely. Combining CV and NLP to generate video descriptions was previously considered a difficult task from a vision standpoint. The goal of establishing a correlation between video content comprehension and textual prediction has been the focus of considerable study for the past few years (Shorten et al., 2021; Aneja et al., 2018; Kiros et al., 2014; Krishna et al., 2017; Amirian et al., 2020). The overall video captioning framework explained in Figure 1.3. Establishing a connection between visual stuff and text prediction is a relatively simple task for humans. However, it has been viewed as a particularly difficult problem for machines and a vital component of machine intelligence. It has a wide range of applications, including video comprehension, video retrieval, and video subtitling.

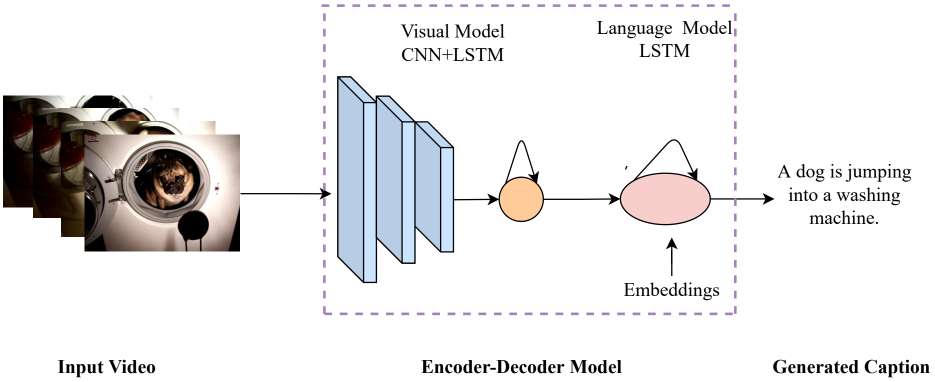


Figure 1.3: The overview video captioning framework.

1.5.1 Image Captioning

Image captioning is a term that refers to the process of automatically generating the textual description of an image (Karpathy and Fei-Fei, 2015). It is frequently used interchangeably with image annotation. It entails the use of both CV and NLP techniques to convert an image representation to a textual composition. Prior to the introduction of deep learning, jobs such as captioning were nearly impossible; however, with breakthroughs in complex algorithms, multimodal approaches, efficient hardware, and a massive volume of datasets, such tasks are becoming doable.

Image captioning can be used to solve a variety of real-world challenges, including assisting the blind, autonomous vehicles, academic bots, and military objectives. To a considerable extent, the majority of success in image captioning has come from the supervised domain, in which massive volumes of data consisting of images and around two to five label captions defining the activities of the images are provided. Thus, the model is tasked with learning the images' feature demonstration and mapping it to a language model, with the eventual objective of generating a textual representation of the image's description. While it may appear uncomplicated and straightforward to humans to characterise an image using words, it is far from simple to recreate in an artificial system and involves extensive techniques to extract features from the images as well as transfer the features to the relevant language model.

In general, CNNs are used to extract features, whereas RNNs are used to translate training annotations to visual features. Apart from identifying and extracting the most important and nuanced elements in an image, it is critical to understand the interactions and semantic relationships between such items, as well as how to depict them properly using appropriate tenses and sentence structures. Additionally, because the training labels are texts and the image attributes are different, language model techniques are necessary to examine the form, meaning, and context of a series of words. This process becomes considerably more complicated when keywords are necessary to emphasise the event or setting being described.

1.5.2 Video Captioning

Most individuals find it easy to describe a video in normal language, while machines find it difficult. Methodologically, defining the models or algorithms is difficult since it is hard to ascertain the contributions of the visual aspects and the accepted language model to the final description. Image captioning can be applied to the video keyframes and a tiny sample of the frames in-between the key frames to create video captioning (Gao et al., 2017; Krishna et al., 2017).

There are two phases to automatically creating natural language sentences summarising video information. The first stage is to comprehend the objects, actions, and their associations in terms of video clip attributes. The video clip is delivered as a succession of frames, which are interpreted as images. So, in each clip, we have a succession of frames that are input images. The retrieved data from the video clip is then placed in a conventional feature vector. The second step receives this vector. The second level is caption generation, which involves describing what has been retrieved in a comprehensible natural language sentence, thereby mapping the items identified in the first stage. The overall taxonomy for deep learning-based image/video captioning methods (Hossain et al., 2019) is shown in Figure 1.4.

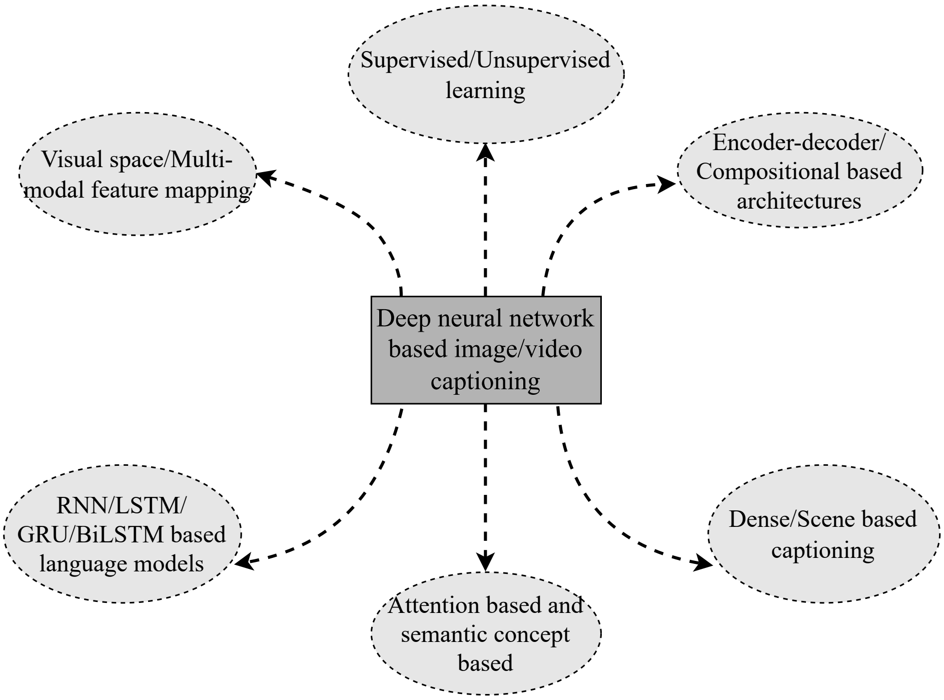

Supervised/Unsupervised learning
Visual space/Multi-modal feature mapping
Encoder-decoder/ Compositional based architectures
Deep neural network based image/video captioning
RNN/LSTM/ GRU/BiLSTM based language models
Dense/Scene based captioning
Attention based and semantic concept based

Figure 1.4: Overall taxonomy of image/video captioning.

1.5.3 Attention Guided Captioning

Attention has become increasingly important, and as a result, benchmarks in a variety of activities have improved, including machine translation, language modelling, and other NLP tasks, as well as CV tasks (Xu et al., 2020). Indeed, attention has been shown to connect the meaning of features, which aids in comprehending how such a feature relates to another.

When this is incorporated into a neural network, it encourages the model to focus on salient and significant features and ignore other noisy parts of the data space distribution. To approximate the concept of attention in image annotation, a model is trained to focus its computation on the identified salient regions while simultaneously generating captions utilising both soft and hard attention. The deterministic soft attention model is trained using normal backpropagation by weighting the annotated vector of picture features, but the stochastic hard attention model is trained by optimising a variational lower bound, which is set to 1 when the feature is salient.

Adaptive attention employed a hierarchical structure to merge high-level semantic information with visual information from a picture to build intuitive representations . The top-down and bottom-up methodologies are coupled through the use of semantic attention, which defines attribute detectors that enable it to switch between ideas dynamically. This enables the detectors to select appropriate candidates for attention calculation based on the inputs given.

1.5.4 Challenges

The fundamental components of video description or captioning are object detection, activity recognition, and sentence creation. The early research on natural language descriptions of visual data was mostly concerned with static images. These depended on a variety of algorithms and techniques for object recognition in images, as well as simple template-based approaches for sentence generation. The difficult issue of object recognition is related with identifying the objects, and describing the relationships between them remains an open problem. Numerous models, on the other hand, have attempted to bridge the divide between identifying the characteristics of images and relating them to natural descriptions. However, achieving desirable results is an open task. The exponential expansion and advancements in research facilitated a rapid change in the deep CNN domain, and other optimal versions significantly aided in image captioning. However, video captioning is typically a different problem than image captioning due to the complexity of identifying the numerous sets of objects and their relationships. Despite additional research challenges in visual captioning, a few attempts are largely helped by modern technology such as LSTM (Hochreiter and Schmidhuber, 1997) for expanded word prediction, RNNs , and Gated Recurrent Unit (GRU). Due to the shortcomings of current visual attention models, non-semantic and non-contextual subtitles mislead visual comprehension. The video captioning model necessitates a semantic correlation between seen contents and generated words, which is not taken into account in existing methods. Finally, exploiting the advantages of various sentence representation embeddings with cutting-edge deep learning techniques is an unresolved topic in the domain of picture and video captioning.

1.5.5 Contributions

- To that extent, the proposed framework's primary contributions to the connection of video and text are an encoder-decoder framework using a 2D-CNN model and layered LSTM as the encoder and an LSTM model integrated with an attention mechanism as the decoder with a hybrid loss function.

- A unique multi-layer BiLSTM encoder and a multi-layer unidirectional decoder are used to connect the video and text. Two levels of temporal soft attention are used in both the encoder and decoder units. This emphasis on a comprehensive global view of video segments imparts extra representational characteristics.
- A visual captioning system that combines a CNN and a short-connected LSTM-based encoder-decoder model with sentence context vectors, word embeddings, and multi-headed attention.

1.6 Word Embedding

The concept of word embeddings (Mikolov et al., 2013; Pennington et al., 2014; Joulin et al., 2017; Peters et al., 2018), is a highly effective tool in deep learning science for NLP applications. That is, a vocabulary's set of word vectors is capable of capturing the meaning of words and their relationship to their context. Word embeddings are vector representations that embed words in a feature space. As points in an n-dimensional space, word embeddings exhibit a number of fascinating features, including the following:

- An identifier (vector) for each word.
- Vectors are normally only a few hundred dimensions in length.
- In n-dimensional feature space, words with comparable meanings have similar vector values and are consequently nearby.
- Semantic regularities are analogous to geometrical qualities.

The overall taxonomy of the word embeddings given in Figure 1.5.

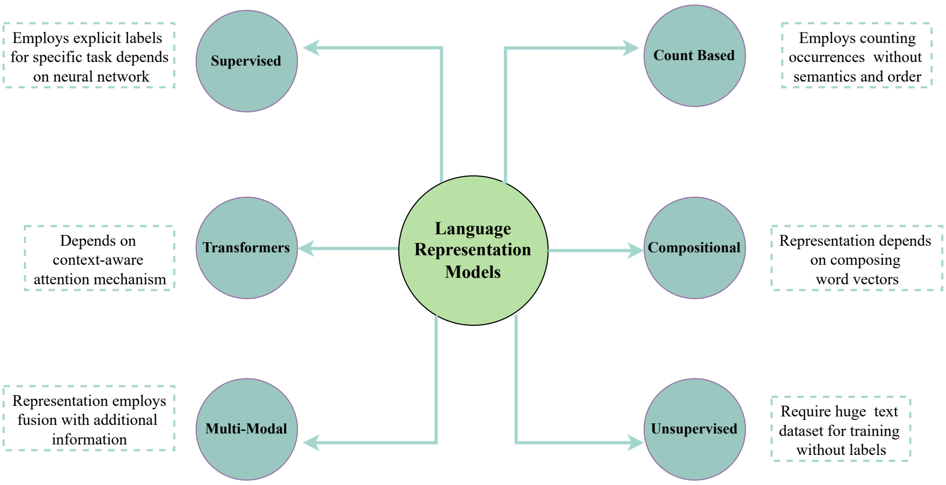

Employs explicit labels for specific task depends on neural network
Supervised
Count Based
Employs counting occurrences without semantics and order
Depends on context-aware attention mechanism
Transformers
Language Representation Models
Compositional
Representation depends on composing word vectors
Representation employs fusion with additional information
Multi-Modal
Unsupervised
Require huge text dataset for training without labels

Figure 1.5: Overall taxonomy of word embedding representations.

1.7 Motivation

- The human visual framework is so complex that it can create a rich depiction of scenes after a single look and quickly outline every object present.
- Image understanding and video captioning are the most widely used vision-based innovations.
- Accuracy plays an important role in correctness of the model.
- Emergence of advance machine learning and robotics.
- Describing the images and videos in natural language descriptions.
- Video captioning has many applications such as video indexing, human-robot interaction, assisting the visually disabled, automatic video subtitling, procedure generation for instructional videos, video surveillance and understanding sign language.

1.8 Overall Contributions

Inspired by deep image captioning models and the extensive use of that technology in CV, we addressed the gaps in object semantic recognition, their associations in terms of textual contents together with their application in real-life usage to describe the contents of images and videos. The overall contributions of this research work are as follows:

1. Design and implementation of an Encoder-Decoder Neural Network Architecture is combined with a novel strategy to improve global label consistency, to come with an enhanced image segmentation model.
2. Design and implementation of a Bayesian Non-parametric (BN) approach to solve the complex visual tasks using the non-parametric property with Chinese Restaurant Process Stacked with Weakly Supervised Markov Random Field (WS-MRF-CRP), which uses Markov Random Field (MRF) for low-level and Chinese Restaurant Process (CRP) for high-level processing.
3. Design and implementation of model that uses superpixelization clustering. Then, a Multi Heat Map Slices Fusion model produces an object seed heat map. A Saliency-Edge-Colour-Texture(SECT) model generates pixel-level annotations using the PSPNet model to create the object's final semantic segmentation.
4. Design and implementation of an encoder-decoder framework with a 2D-CNN model and layered LSTM as the encoder and an LSTM model integrated with an attention mechanism working as a decoder with a hybrid loss function.
5. Design and implementation of a novel video captioning framework that combines a multilayer Bidirectional LSTM (BiLSTM) encoder and unidirectional decoder with a framework of temporal attention to build superior global representations for videos.

6. Design and implementation of an approach for efficiently captioning videos that combines a CNN and a short-connected LSTM-based encoder-decoder model with a sentence context vector.

1.9 Organization of Thesis

This section outlines the organisation of the thesis. Chapter 2 presents a brief assessment of the literature, research gaps, problem statements, and research objectives. The proposed methodologies for achieving the research objectives are discussed in Chapters 3 and 4, followed by concluding remarks and future directions are described in Chapter 5. The overall organizaition of the report is depicted in Figure 1.6.

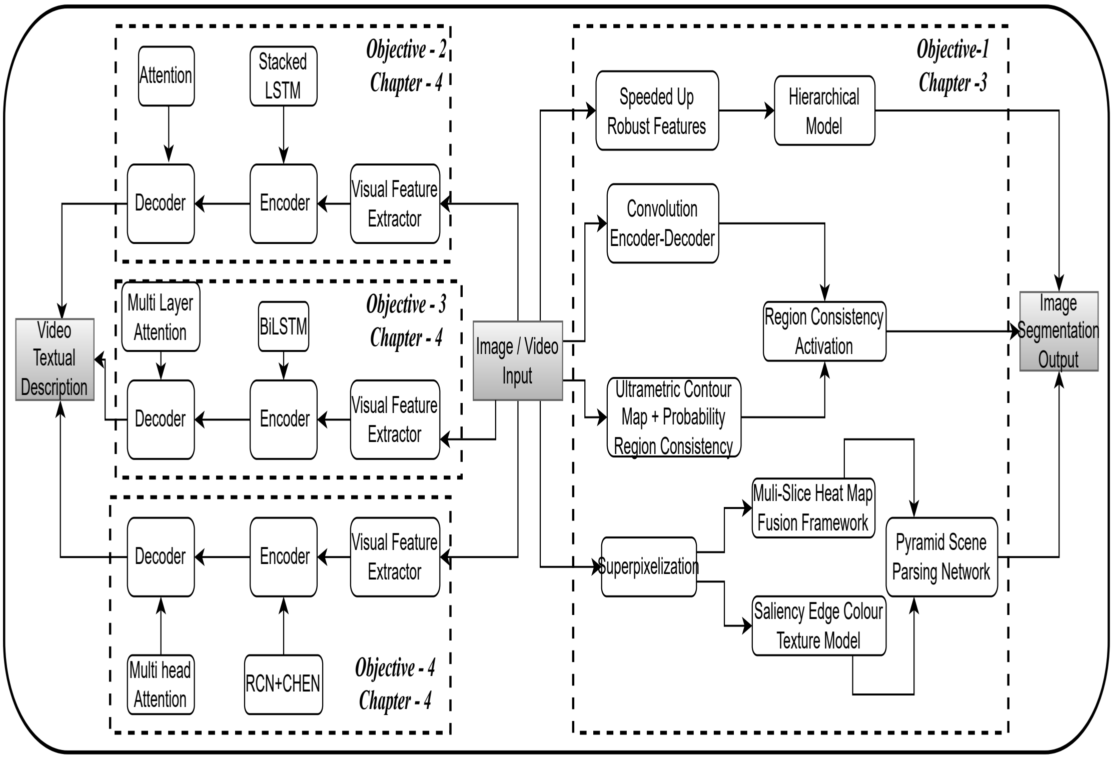


Figure 1.6: Organization of the thesis in line with the research objectives and contributions.

Chapter 2

Literature Survey

In this chapter, we examine relevant publications that assist position this dissertation in the context of the sub-field of semantic segmentation and image/video captioning's entire development. Initially, a summary of pertinent research publications, the majority of which were either recent or contemporary to the methodologies discussed in this dissertation. First, we will provide a quick overview of approaches to semantic image segmentation that have served as the foundation for some of the efforts in this dissertation (Chapters 3). Then, we examine innovations in image and video captioning, including various categories of enhancements that expand upon the models presented in Chapter 4 of this thesis. In addition, we will consider works that utilise external knowledge to enhance image and video captioning. Finally, we examine strategies that have been developed to address some of the difficulties associated with understanding the image/video captioning techniques currently in use.

2.1 Semantic Segmentation

The human visual framework is so exquisite that it can instantly construct a rich portrayal of scenes and quickly outline each object present. A primary objective of CV research has been to replicate the ability:

- Perceiving the scene's objects
- Depicting them according to their characteristics
- Identifying the said objects

Semantic segmentation (Liu et al., 2019; Kirillov et al., 2019; Ban et al., 2018; Kang and Nguyen, 2019), which attempts to classify objects at the pixel level, is a critical component of object recognition models. In this, the goal is to classify a single object in an image using only the image as input. Classification is performed at the pixel level. The input image is processed to determine the pixel-level properties contained within, and then the object in the image is delimited and segmented for identification purposes.

2.1.1 Traditional Methods

Prior to the advent of artificial neural networks, the majority of techniques to segmentation and semantic segmentation were based on unsupervised thresholding and clustering algorithms (Sezgin and Sankur, 2004). In the majority of circumstances, traditional semantic segmentation approaches (Ban et al., 2018; Kang and Nguyen, 2019) require less time to compute the model. Additionally, the majority of methodologies (Zhao and Wang, 2018; Cao et al., 2018) require less data than what is required in the present era of artificial neural networks and deep learning. There are even more advanced thresholding techniques that involve additional classes and are frequently classified as histogram shape-based, entropy-based, object attribute-based, and spatial-based (Sezgin and Sankur, 2004).

2.1.2 Region-Based Models

Regions are first taken from an image and defined using their constituent features in the region-based semantic segmentation design (Girshick, 2015; Ren et al., 2015; Li et al., 2017; Chen et al., 2017). Then, using a trained region classifier, the pixels per region with the highest incidence are labelled. The region-based techniques (Girshick et al., 2015) employ the divide and conquer strategy, in which numerous properties are recorded at multiple scales and then integrated to produce a whole.

2.1.3 Fully Convolutional Network Based Models

Fully Convolutional Network (FCN) models (Badrinarayanan et al., 2017; Shelhamer et al., 2016; Long et al., 2015) lack dense layers, as do other conventional CNN models; instead, they are constructed of 1×1 convolutions that provide the function of dense or fully connected layers. Additionally, a FCN accepts images of specific size as input and delivers outputs with the same spatial dimensions. This model is based on the encoder-decoder model in that it classifies pixels in an image into predetermined classes by extracting features using a convolution network in the encoder and lowering the dimensionality of the feature maps before they are upsampled by the decoder.

2.1.4 Refinement Network

Due to the resolution loss introduced by the encoder models (Chen et al., 2014, 2017; Wu et al., 2019) in the conventional FCN-based model, the decoder is able to produce fine-grained segmentation, particularly at the boundaries and edges. Though this problem has been addressed by the incorporation of skip connections, the addition of global information, and other ways, it is far from solved, and some algorithms (Chen et al., 2014; Li et al., 2019; Xiang et al., 2019) have incorporated several features or, in some situations, certain postprocessing functions to identify alternate solutions.

2.1.5 Weakly Supervised and Semisupervised Approaches

While the majority of models (Pinheiro and Collobert, 2015; Khoreva et al., 2017; Papandreou et al., 2015) require a significant number of images and their annotated labels, the process of manually annotating labels is quite daunting and time intensive, and hence semantic segmentation models have been attempted using weakly supervised approaches (Li et al., 2019; Dai et al., 2015). Given weakly annotated image data, the model was trained to provide a larger weight to pixels containing a class label. Trained on a subset of the ImageNet dataset (Deng et al., 2009), the networks focus on recognising essential pixels associated with previously identified single-class objects and inferring their class. Table 2.1 reports the summary of key existing works on image segmentation.

Table 2.1: Summary of key existing works of image segmentation.

Authors	**Methodologies**	**Limitations**
(Badrinarayanan et al., 2017)	Authors presented a novel and practical deep fully convolutional neural network architecture for semantic pixel-wise segmentation termed SegNet. This core trainable segmentation engine consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer	End-to-end learning of deep segmentation architectures is a harder challenge
(Li et al., 2015)	In this paper, authors investigate the learned features from ConvNets-VGG16 and propose a novel algorithm called Region Consistency Activation (RCA) for scene labelling.	Failed to achieve more global label consistency of a scene labelling system
(Ren et al., 2015)	This paper uses Multi instance linear framework, which is able to learn an object localization model from large-scale data without using bounding box annotations using pre-trained convolutional network.	Problem arises for inaccurate box prediction, where only part of the true object is captured or too much background is covered.

Table 2.1 Summary of key existing works on image segmentation (Contd.)

Authors	**Methodologies**	**Limitations**
(Xiao and Zhong, 2019)	Proposed method combines dilated convolution with multi-scale feature fusion to form a convolutional neural network for image semantic segmentation tasks.	Fails to achieve target detection during semantic segmentation to distinguish different individuals.
(Feng and Wang, 2019)	Used weakly supervised approach for semantic segmentation, and adopt the mini-batch k-means approach to cluster the massive training samples with image level labels and use the clustering centers as the initial value of the base matrix.	It performed well with uncorrelated clusters, but it cannot distinguish correlated clusters.
(Zhou et al., 2018)	In this paper, Weakly supervised segmentation is carried out with image-level labels, instead of expensive pixel-level masks. Used a Peak Response Maps(PRMs) to enable a classification network for instance mask extraction with help of convolutional network.	It is for weakly-supervised systems, so the PRMs can be misled by noisy co-occurrence patterns and sometimes it struggles to tell the difference between object parts and multiple objects.

2.2 Visual Captioning

Visual captioning, also known as video/image captioning, is the process of providing a written description/caption for a video/image (Gao et al., 2017; Amirian et al., 2020). The caption/sentence briefly describes the objects and actions depicted in the image or video. Combining CV and NLP to generate video descriptions was traditionally thought to be a vision-intensive operation. For the last few years, much research has been devoted to establishing a correlation between video content understanding and textual prediction.

Two major causes have fueled growth in image captioning research: 1) advancements in deep neural networks and 2) the availability of enormous corpora of images with text descriptions. Deep neural network models (Simonyan and Zisserman, 2015;

Cao et al., 2019) have grown in popularity because to their performance and ability to do end-to-end training in CV and NLP. The most of the attempts are also popular due to the combination of the CNN and RNN. (Shorten et al., 2021; Aneja et al., 2018; Kiros et al., 2014; Krishna et al., 2017; Amirian et al., 2020) . These captioning models begin by encoding an image with a CNN into a fixed-length feature vector, and then generate a description by either conditioning text generation on image features or by embedding image features and previously generated words into a multimodal space before predicting the next word. RNN has been a popular choice for caption generating in particular (Sutskever et al., 2014; Schuster and Paliwal, 1997). Numerous researchers generated the description using a simple RNN. Others employed LSTM, which did better on the task than ordinary RNN (Bin et al., 2019; Li et al., 2019; Yang et al., 2018). However, several works have favoured log bilinear and maximum entropy language models as well. For the visual representation, the majority of models use an intermediate representation from a convolutional neural network (such as the activations of the fully connected layer just before classification), because these features, while trained for object recognition, generalise well to other tasks. Nevertheless, a few models express images as a confidence vector across a finite number of visual concepts. This style is especially advantageous when the domain is clearly defined. Almost always, the visual pipeline's parameters are initialised using weights learned from the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al., 2015). A very interesting variation on the visual representation in image captioning is the addition of "attention," in which the model does not learn a single fixed representation of the image, instead learns a spatially weighted representation of the image that focuses on different locations in the image as it generates each word of the description.

2.2.1 Deep Neural Networks

CNN (Alzubaidi et al., 2021) is a type of neural network that has gained popularity in recent years. It is a type of neural network that is specifically designed for image/video captioning. However, video captioning is more complex than image captioning and cannot be performed only through the usage of CNN. The CNN has proved enormously successful in practical applications and has been incorporated into a variety of architectures for image identification, object detection, and object segmentation (Alzubaidi et al., 2021). By integrating numerous blocks and employing small filter sizes, CNN learns a detailed representation of the input, outperforming all other standard approaches in image-related tasks.

2.2.2 Recurrent Neural Network

RNN is a neural network that can process sequential data. While CNNs are usually used to process data with a fixed grid of values, such as images, RNNs are mostly used to analyse data with temporal dependencies (data input changes over time). RNNs can theoretically tolerate arbitrary input and output sizes, allowing for the development of highly adaptable neural network models. RNN recently has been applied to a number of tasks, including speech recognition, language modelling, image captioning, and video description. The article by (Venugopalan et al., 2014) discusses how two stacked RNNs are used to decode video frame information into a mapping space that is then utilised for captioning. The authors (Karpathy and Fei-Fei, 2015) have also employed bidirectional RNNs in video captioning, where they are used to learn the relationship between video frames and English sentences and to add word embedding.

2.2.3 Long Short-Term Memory

The vanishing gradients and exploding gradients are a common issue while training RNN models. Because derivatives propagate through several time steps, repeated multiples of values less than one gravitate to zero, whereas repeated multiples of values greater than one explode. RNN requires repeated multiplication of derivatives over time steps, making them prone to vanishing and exploding gradients. LSTM has been presented to address the problem of vanishing and exploding gradients.

Image and video captioning are the most widely used vision-based innovations, addressing a variety of real-world issues such as retrieving relevant videos, enabling visually impaired persons to comprehend real-world events, and visualising and analysing recordings(Bin et al., 2019; Olivastri et al., 2019; Zhao et al., 2019) . The older approaches, such as template-based and retrieval-based, appear to be out of date, notwithstanding their use to some extent. The recent interest and advancements in this field include neural image caption generation using deep learning (You et al., 2016; Lin and Zhang, 2021).

Most of the initial work focused on image captioning, then eventually broadened to incorporate video captioning. (Vinyals et al., 2014) built a deep-generative model for autonomously creating captions for images by combining CNN and RNN. (You et al., 2016) incorporate top-down and bottom-up methods for extracting additional characteristics of the image and coupling them with an RNN capable of selectively attending to semantic features identified in the image.

(Lin and Zhang, 2021) presented a sequence-to-sequence framework of CNN and RNN with an added attention module, as well as an additional memory unit for the encoder and decoder to supplement the memory of the RNN. However, in this approach, only the RNN component is used for training. The author (Zhang et al., 2021) suggested a graph-based partitioning and summarization technique, as well as two distinct types of GCN-LSTM interaction modules. However, the research coupled with graph represenations to explore the word relationships.

(Gao et al., 2020) developed a novel layered technique with adaptive attention that explicitly addresses the spatial or temporal attention necessary for picking certain regions or frames for word prediction. Additionally, the strategy emphasised low-level visual cues and high-level verbal contexts. However, the strategy may perform even better when numerous hLSTMat-based models are used in an ensemble.

(Liu et al., 2021) devised a new sibling convolutional encoder that encodes videos concurrently using a dual branch architecture. This combined semantic extraction model is coupled with "soft attention" and fed into an RNN-based decoder unit. However, the RNN decoder has a limited memory capacity; other deep networks such as LSTM and GRU have been shown to be more successful. (Xu et al., 2018) constructed an attention network to investigate attention fusion in both a hierarchical and end-to-end fashion. It also contains a sizable number of encoder and fusion attention modules.

For future time step decoding, (Bin et al., 2019) presented the BiLSTM, which processes videos in either forward or reverse orientations. It also develops a conceptual framework for soft attention that focuses on items with specified probabilities. (Song et al., 2019) introduced MS-RNNs, an end-to-end architecture that embraces stochastic variables to cater for data uncertainty. (Yang et al., 2018) annotate the videos using Generative Adversarial Network (GAN). The caption generator provides captions for recordings, however the discriminator determines if words are accurate or made up. (Zheng et al., 2020) introduced a interface that identifies actions by associating to both the subject and video dynamics. Their behaviour is then translated using both the subject's embedding and the video's temporal properties.

2.2.4 Attention Mechanism

The attention mechanism (Bahdanau et al., 2014; Shen et al., 2018) is one of the most significant accomplishments in the last decade of deep learning research. It has produced a slew of recent innovations in NLP, including Google's Bidirectional Encoder

Representations from Transformers (BERT) (Devlin et al., 2018) and the transformer architecture. A neural network is said to be an attempt to simplify the actions of the human brain. Attention mechanism is another attempt to replicate the same behaviour of selecting focusing on a few important things while ignoring others in deep neural networks.

Machine translation, a fundamental problem in NLP, tries to automate the translation of human languages. Neural machine translation has established itself as a highly successful paradigm for learning features directly from data, resulting in amazing advances in machine translation. However, the widely used encoder-decoder model architecture in neural machine translation has encountered difficulties with effectiveness and training. Almost all models of neural machine translation at the state-of-the-art are based on attention mechanisms. Attention models (Vaswani et al., 2017a; Xu et al., 2015) assist in relating input sequence units regardless of their spatial and temporal separation, hence increasing the parallelizability of sequence data processing. As a result, adding attention significantly enhances the Neural Machine Translation (NMT) model (Xu et al., 2020) .

By explicitly attending to all signals, attention mechanisms enable the model to better reflect long-range contextual dependency regardless of their position in the input or output sequence. The attention-enabled NMT model does not need encoding the entire source sentence into a single fixed-length feature vector. Rather than that, it converts the source text to a series of feature vectors. At each decoding step, the attention model searches for a collection of positions containing the most important information, and then the decoder constructs a translation word using the context vector, which is a weighted sum of feature vectors from selected positions.

Additionally, the attention mechanism can be classified into soft and hard attention based on the range of selection (Shen et al., 2018). In soft attention (also known as global attention), an overall position distribution is calculated. The generated probability represents the relative relevance of each position and is utilised as a weight in the context vector generation process. Due to the fact that soft attention is fully differentiable, it is simple to train using a gradient-based technique. Unlike soft attention, the hard attention mechanism compelled the NMT model to focus exclusively on a portion of the input sequence that was deemed most relevant and ignore the remainder. Due to the random nature of selection in hard attention, it cannot be easily taught by backpropagation. Local attention is a hybrid method that incorporates both hard and soft attention (Shen et al., 2018). It takes a subset of the input sequence at random and computes a distribution over the subsequence.

(Zhao et al., 2019), the soft attention module selectively selects the most pertinent frames and the proper location of objects within each frame. Additionally, the authors proposed an attention module that would focus on the most comparable phrases in order to exploit more precise language descriptions.

More precisely, in soft attention, a softmax layer can calculate the probability distribution at each decoding step using the energy function of the sequence of feature vectors. Following the probability distribution calculation, the context vector is a weighted point sum of feature vectors, with each feature vector's weight equal to its likelihood. The energy function in additive attention is a single-layer neural network that utilises the current and previous positions' feature vectors. The energy function in multiplicative attention is the Mahalanobis distance or the simple dot product of distinct feature vector pairs.

Transformer is a model (Tan and Bansal, 2019; Li et al., 2019) that is entirely based on attention mechanisms and so requires substantially less training time than a recurrent model. Transformers encode symbol positions in a significantly different way than the recurrent model does. It augments the input embedding at the bottom of the encoder and decoder with position encoding. In a transformer, there are two types of attention mechanisms: single-head attention and multi-head attention. Multi-head attention (Voita et al., 2019) can be thought of as a collection of concurrent single-head attention models. The output of multi-head attention is simply the sum of the outputs of each single-head attention. Because the transformer does not directly accept embeddings as inputs, its attention is also referred to as key-value attention. Typically, transformers accept key-value pairs and queries as inputs. Queries originate from earlier levels in an encoder-decoder architecture, while key-value pairs originate from encoder outputs.

2.2.5 Word Embedding

NLP is a term that refers to computer systems that are programmed to comprehend human language. Human language, like English or Hindi, is composed of words and sentences, and NLP aims to extract information from these sentences. A word embedding is a type of learnt text representation in which words with the same meaning are represented equally. Individual words are represented as real-valued vectors in a predetermined vector space using word embeddings. Each word is associated with a vector, and the vector values are discovered in a manner similar to that of a neural network. As a result, the approach is frequently grouped with deep learning.

When working with word embeddings, one can either employ pre-trained word vectors or train new vectors based on specific requirements or objectives. By and large, one straightforward method to use word embeddings is to work with pre-trained ones. For this purpose, it is intuitive that word embeddings are not all the same, and that there are an infinite number of ways to generate them. Certain types perform better than others, frequently depending on how they are used in specific activities. There are numerous pre-trained embeddings available, including Word2vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), FastText (Joulin et al., 2017), Embeddings from Language Models (ELMo) (Peters et al., 2018), and BERT. Utilizing these pre-trained vectors is rather straightforward: all that is required is to download the vectors package, or in certain cases, to load a pre-trained version of the model that will provide the user with the requisite word embeddings. The latter situation is valid for embeddings that require the production of highly contextualised word embeddings (such as ELMo and all variations of BERT). Certain language models that include pre-trained embeddings can also be used to generate representations for sentences or longer sequences. This is especially true for models like as BERT and its expansions.

2.2.5.1 Count Based Techniques

The majority of these strategies are statistical in nature, and one of the most well-known is the well-known Bag of Words (BoW) (Harris, 1954).

2.2.5.2 Compositional Methods

The key tenet of this particular form of symbolic approach is the concept of compositionality (Mitchell and Lapata, 2008).

2.2.5.3 Unsupervised Methods

The characteristic of unsupervised approaches (Mikolov et al., 2013) for producing vectors representing long text sequences is their reliance on unstructured data, which frequently examines unlabeled plain text.

2.2.5.4 Supervised Methods

To tackle specific tasks, supervised methods (Lai et al., 2015) are used, and the training phase is defined by the use of a labelled dataset created by human annotators. Additionally, these approaches are characterised by the presence of textual representations in the last layers preceding the prediction.

2.2.5.5 Transformer Based Methods

The transformer structure makes use of many multi-headed attention layers (Tan and Bansal, 2019; Li et al., 2019), which enables the model to concentrate on certain areas of the analysed text. This is done so that the model may focus on the relevant elements of a sentence, while ignoring the less important ones, and capture long-range dependencies that earlier models were unable to grasp.

2.2.5.6 Multimodal Approaches

The emphasis is on a novel method of information fusion that goes beyond word embeddings and makes use of multi-modality (Bergsma and Goebel, 2011). The fusion approaches are defined by the use of simple operations to perform a combination of features, the multi-task methods are defined by the presence of multiple terms in the loss function and the use of a multi-task learning approach, and the final group is defined by the use of the attention mechanism. Table 2.2 provides summary of key existing works on video captioning.

Table 2.2: Summary of key existing works of video captioning.

Authors	**Methodologies**	**Limitations**
(Shekharet al. , 2020)	The video captioning model proposed by the authors use a fusion of different type of features such as spatial features from a 2D-CNN, spatio-temporal features from a 3D-CNN and semantic features.	The decrease in CIDEr score may be because of using only the domain-specific semantic features. The ability to predict cross-domain semantic words is reduced.
(Xu et al., 2020)	The authors proposed a temporal-spatial and channel attention mechanism that enables the model to take advantage of various video features and ensures the consistency of visual features between sentence descriptions to enhance the effect of the model.	Though the model utilizing spatial and temporal attentions, it lacks in performance.

Table 2.2 Summary of key existing works of video captioning (Contd..)

Authors	**Methodologies**	**Limitations**
(Bin et al., 2019)	A novel video captioning framework, which integrates BiLSTM and a soft attention mechanism to generate better global representations for videos as well as enhance the recognition of lasting motions in videos proposed	1. exploring object-level spatial–temporal dependency, which is analogous to human visual understanding and 2. reasoning relationships among different objects at the same time (spatial reasoning) or the same object at a different time (temporal reasoning).
(Venugopalan et al., 2015)	Authors proposed a novel end-to-end sequence-to-sequence model to generate captions for videos. They used RNNs, specifically LSTMs, which have demonstrated state-of-the-art performance in image caption generation.	No attention mechanism are involved.
(Yan et al., 2020)	Authors proposed the use of a novel Spatial-Temporal Attention mechanism (STAT) within an encoder-decoder neural network for video captioning. The proposed STAT successfully takes into account both the spatial and temporal structures in a video, so it makes the decoder to automatically select the significant regions in the most relevant temporal segments for word prediction.	Authors draw a conclusion that temporal and spatial cues are complementary. When both attention mechanisms are utilized simultaneously, it will achieve the best performance

Table 2.2 Summary of key existing works of video captioning (Contd..)

Authors	Methodologies	Limitations
(Nabati and Behrad, 2020b)	The proposed architecture comprises two LSTM layers and a word selection module. The first LSTM layer has the responsibility of encoding frame features extracted by a pre-trained deep CNN. In the second LSTM layer, a novel architecture is used for video captioning by leveraging several decoding LSTMs in a parallel and boosting architecture	In the proposed architecture, several LSTMs are successively added to the network during the training phase by using a particular type of Adaptive Boosting (AdaBoost) algorithm.

2.3 Research Gaps Identified

- The majority of the work is done on strongly supervised images rather than weakly supervised.
- The lack of an object+attribute model learned from weakly labeled data, i.e., images with object and attribute labels but neither their associations nor their locations in the form of bounding boxes or segmentation masks.
- Early work on natural language description of visual data primarily focused on static images, and there is less emphasis on understanding dynamic video data and producing its description.
- Recognizing the fine characteristics of visual contents and the interconnections between objects is a difficult endeavour. The most difficult aspect of this situation is the subtlety of the action units.
- Need to build semantic context oriented model using latest deep learning techniques to provide efficient visual descriptions.
- Learning mid level representations between the visual and natural language domains is a crucial problem in visual-to-text techniques.
- By leveraging transformative advancements in deep learning and integrating them with cutting-edge approaches in CV and NLP, we will be able to develop more efficient and scalable systems for natural language video description.
- Even if we were able to distinguish many semantic features that appear throughout the visual data, it is difficult to rank their importance in line with the image or video's theme in order to provide more pertinent written descriptions. In addition, it is required to consider linguistic complexity to be employed.

2.4 Problem Statement

This thesis attempts to solve the research based on an analysis of the gaps uncovered in a survey of earlier works in visual image and video analysis. The goal of this research is to suggest a method for a more accurate analysis of visual images and video to interpret its content in terms of natural language descriptions. In light of this, the research problem addressed in this work is stated as follows:

"To design and develop a comprehensive region-based semantic composition framework for visual image and video event specification."

2.5 Research Objectives

The following objectives were formulated in light of the identified research gaps and the stated problem.

- To design an effective technique for enhancing and analyzing the pixel/image-level label accuracy of visual data.
- To design and implement a semantic context-driven framework for captioning images and videos by utilizing a visual captioning dataset.
- To create a framework for effective visual description generation using conceptual attention-guided captioning.
- To develop and implement a comprehensive captioning model with merged sentence and image feature vectors in order to improve the prediction accuracy of natural language descriptions.

Chapter 3

Visual Image Data Analysis using Semantic Segmentation

Semantic segmentation is a critical component of object recognition models, as it attempts to classify objects at the pixel level. In this connection, the research work inititially was concentrated at recongnition of images at the pixel levels. Initially, this chapter discusses the tasks involved in classifying a single object from an image at the pixel level. The input image is processed to extract pixel-level information, and the object in the image is then demarcated and segmented for identification purposes. To begin, a Chinese Restaurant Process Stacked with Weakly Supervised Markov Random Field (WS-MRF-CRP) is created, which employs Markov Random Fields (MRF) (Wang and Zhao, 2017) for low-level operations and Chinese Restaurant Process (CRP) (Blei and Frazier, 2011) for high-level operations. The second model begins with superpixelization, which is accomplished by Simple Linear Iterative Clustering (SLIC). A Multi Heat Map Slices Fusion (MSF) model generates an object seed heat map and a Saliency Edge Color Texture (SECT) from which pixel-level annotations are generated. Finally, the Pyramid Scene Parsing network (PSPNet) model is used to construct the object's final semantic segmentation. Additionally, this chapter explores another model for developing an enhanced image segmentation model using an Encoder-Decoder neural network architecture in conjunction with a novel strategy for improving global label consistency.

3.1 Image Segmentation using Encoder-Decoder Architecture and RCA

In this research, an encoder-decoder neural network architecture is merged with a unique technique for enhancing global label consistency and an image segmentation model to create an encoder-decoder neural network architecture. The work investigates and applies label distribution predictions drawn from the SegNet network (Badrinarayanan et al., 2017) to image labelling. To increase global label consistency, an approach called RCA (Li et al., 2015) is used. The RCA algorithm is based on a revolutionary transformation of the Ultrametric Contour Map (UCM) (Arbelaez et al., 2011) to the Probability of Regions Consistency (PRC) (Li et al., 2015). Proposed method achieved superior performance when compared to state-of-the-art approaches. The following are the most significant contributions made by the proposed work.

- Proposed a novel SegNet encoder-decoder neural network architecture for image labelling in the first step.
- Then implemented an algorithm called RCA to improve the global label consistency.
- RCA is based on a novel transformation from UCM to PRC.
- The RCA which utilizes the PRC, yields an improvement in the performance of the scene labelling system for global label consistency is implemented.

3.1.1 Methodology

Scene labelling, also known as semantic segmentation, is a critical step toward high-level picture interpretation since it assigns a specific object category to each pixel in an image. Due to the intra-class variances and mutual occlusions of objects in the image, scene labelling becomes a very difficult operation. Two essential difficulties in a scene labelling system are accurately identifying each pixel in an image and ensuring global label consistency throughout the entire image.

The initial step is to explore the SegNet encoder-decoder neural network architecture for image labelling. The global label consistency was then enhanced with the RCA method. It is based on the UCM's revolutionary transformation into the PRC. The Figure 3.1 depicts a block diagram of the proposed architecture for scene labelling. The model is constructed using two distinct input image representations. The first one generates the pixel-by-pixel label distribution representation. This representation was generated using the SegNet architecture. The UCM of the image input is the second image representation. Both of these representations are coupled utilising the RCA approach to produce an improved label consistency in the final image labelling.

Scene labelling aims to turn a raw image X into a label space $L : L = f(X;\xi)$, where ξ represents the parameters of this transformation. The above process is decomposed into two steps. In the first step, X is transformed into a label probability distributions space $Y : Y = g(X;(W;b))$. Where, W and b denote the trainable parameters of ConvNets (Krizhevsky et al., 2012). X is represented by a series of pixels that need to be labeled by the CNN $: X = [x_1; x_2; ..; x_n]$. These pixel representations are bounding

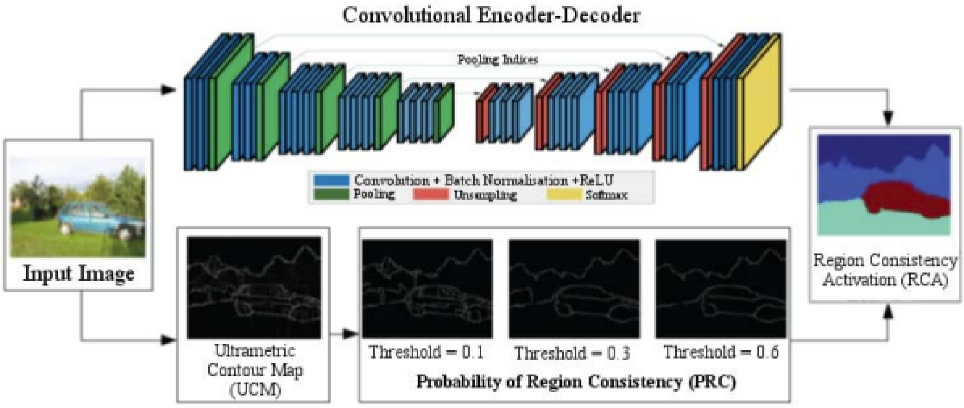

Convolutional Encoder-Decoder
Pooling Indices
Convolution + Batch Normalisation +ReLU
Pooling
Unsampling
Softmax
Input Image
Ultrametric Contour Map (UCM)
Threshold = 0.1
Threshold = 0.3
Threshold = 0.6
Probability of Region Consistency (PRC)
Region Consistency Activation (RCA)

Figure 3.1: Architecture of the proposed approach for scene labelling using UCM, PRC, and RCA.

boxes of different sizes centered on every pixel. For each representation x_i, we transform it into the label probability distribution space: $Y_i = g(x_i; (W; b))$. The final label probability distribution is obtained by combining all the label probability distributions together: $Y = [y_1; y_2; ..; y_n]$. The second step is to transform the label probability distributions into final labels: $L = h(Y; \psi)$. Where, ψ denotes the hyperparameters of the threshold.

An image X is represented by a series of pixels x_i in the framework. Each of the pixels is a candidate for training or testing with ConvNets. The i^{th} layer of the ConvNets is denoted as H_i. H_o is the input of the feature learning system. If the i^{th} layer is a convolution layer, it has trainable parameters W_i and b_i, where W_i is the shared weight vector of convolution kernels and b_i is the bias vector. The output of the convolution layer can be described as:

$$H_i = f(W_i.H_{i-1} + b_i) \tag{3.1}$$

For a pooling layer H_i, the pooling process can be written as:

$$H_i = pool(H_{i-1}) \tag{3.2}$$

We assume Y denotes the output of ConvNets, which is the probability distribution of the different classes:

$$Y(i) = P(L = l^i | H_o; (W; b)) \tag{3.3}$$

The target of training is to minimize the negative log-likelihood of the ConvNets. In order to avoid overfitting, L2 regularization is used. Thus, the final loss function:

$$E(W, b) = -\sum_{i=1}^{|y|} logY_i + \frac{\lambda}{2} W^T W \tag{3.4}$$

The hyperparameter, λ called weight decay, is used to control the regularization effect. In the training process, we minimize $E(W; b)$ by updating the trainable parameters of ConvNets (Krizhevsky et al., 2012) using Stochastic Gradient Descent (SGD) (LeCun et al., 1998). Instead of a single column of ConvNet, heterogeneous multi-column ConvNets are employed. ConvNets of different structures are trained separately. We fused all the predictions from ConvNets with a sliding window fusion process to achieve higher accuracy in the test process.

The SegNet architecture is a convolutional encoder-decoder network. This network is topologically identical to the 13 convolutional layers in the VGG16 (Simonyan and Zisserman, 2015) network. The role of the decoder network is to map the low-resolution encoder feature maps to full input resolution feature maps for pixel-wise classification. There is a decoder layer for every encoder layer. The final output will be the label distributions of every pixel used for the final RCA algorithm and the UCM representation.

UCM's contour detector combines multiple local cues into a globalization framework based on spectral clustering. A threshold is applied to the UCM, and unless their probability is higher than that threshold, all the boundaries are ignored. This produces a segmentation map of the image. We assume that regions have a more substantial label consistency when they are in an over-segmented map compared with the under-segmented ones. Therefore, the relationship between the PRC and the threshold applied to UCM is described using the following equation:

Probability of Region Consistency

- Threshold is applied to the UCM and ignore all the boundaries unless their probability is higher than that threshold, which produces a segmentation map of the image.
- We apply different threshold levels to UCM.
- The relationship between the PRC and the threshold applied to UCM is described using the Equation (3.5).

$$PRC = \neg\alpha log(threshold). \tag{3.5}$$

The hyperparameter α describes the intensity of the inverse relationship between threshold and PRC.

Region Consistency Activation

Two parameters, PRC and label probability distributions P of a region produced from UCM are required to activate label probability distributions of all the pixels in the image. Two ingredients are considered to activate label probability distributions of all the pixels in the image - (1) PRC and, (2) Label probability distributions P of a region which is produced from UCM. The Equation (3.6) shows the computation of P for region R.

$$P = \frac{\sum_{i \in R} Y(i)}{sum(\sum_{i \in R} Y(i))} \tag{3.6}$$

The label distributions of a region R is the average of all the label distributions of the pixels in that region. The label distributions of each pixel is activated using the the Equation (3.7).

$$Y(i, j) = Y(i, j) + PRC \times P(j) \tag{3.7}$$

Where, i and j are the indices of pixel and labels. The RCA algorithm iteratively applies the Equation (3.7) to all the regions which are generated from UCM. The Algorithm for RCA is given below.

3.1.2 Experiments, Results and Discussion

3.1.2.1 Dataset

The proposed model was experimented using the images from CamVid dataset (Brostow et al., 2008), (Brostow et al., 2009) to measure the effectiveness in labelling scheme for this research work. From this dataset we used 600 RGB images of road traffic. The data was split into the training and testing parts using a 70 - 30% split of the original data.

3.1.2.2 Experiments

The SegNet architecture was built in Python using the Keras library that works on top of the Theano deep learning framework. This architecture was trained using the dataset

Algorithm 1: Algorithm for Region Consistency Activation (RCA)

```
Input: X: Label probability distributions;
UCM: Ultrametric contour map;
T: Thresholds vector;
$\alpha$: Hyperparameter defined in Equation (3.5);
Output: Y : Label probability distributions after RCA;
The number of classes: $cla \leftarrow size(X; 3)$;
The number of thresholds :$scales \leftarrow size(T; 2)$ $Y \leftarrow X$
; for $i = 1 \leftarrow scales$ do
    Create a segmentation from UCM:
    $S \leftarrow UCM > T(i)$
    Calculate the number of regions: $N \leftarrow count(S)$
    Calculate PRC of scale i, Equation (3.5):
    $PRC = -\alpha log(T(i))$
end
for $j = 1 \leftarrow N$ do
    Find pixel locations within region S(j):
    $L \leftarrow S(j)$
    Initialize label probability distributions of S(j):
    $P = zeros(cla)$
end
for $k = 1 \leftarrow cla$ do
    for $c = 1 \leftarrow count$ do
        $P(k) \leftarrow P(k) + X(L(1; c); L(2; c); k)$
    end
end
Equation (3.6): $P \leftarrow P = sum(P)$ for $k = 1 \leftarrow cla$ do
    for $c = 1 \leftarrow count$ do
        Equation (3.7), consistency activation
        M denotes Y (L(1; c);L(2; c)k) :
        M ← M + PRC × P(k)
    end
end
```

mentioned before to output the label distributions of all pixels in the image. This label distribution of the image is used in the RCA algorithm that is being implemented in the paper. The images are also hierarchically segmented using the UCM method. This segmentation was implemented using the skimage library in Python. This library has very convenient support for generating RAGs and performing hierarchical merging of regions based on these graphs. The images were segmented using the following threshold values: threshold = [0:05; 0:2; 0; 5; 0; 7; 0:9]. The experiment was repeated for all the above thresholds, and results were noted. The above two image representations were further combined using the RCA algorithm.

3.1.2.3 Performance Analysis

Table 3.1: Performance of the proposed method for scene labelling with and without RCA.

Thresholds	**Without RCA**	**With RCA**
0.05	80.3	82.6
0.2	80.3	81.4
0.5	80.3	83.9
0.7	80.3	81.3
0.9	80.3	82.5

Table. 3.1 clearly shows an improvement in labelling accuracy when the RCA algorithm is used, with the biggest gain occurring between 0.4 and 0.6. Additionally, we observed accuracy gains ranging from 1% to 3% when utilising the RCA algorithm versus not using it while using the standard dataset. This demonstrates the critical nature of global pixel label consistency.

Time Complexity Analysis

The proposed model consists of three components: SegNet model, UCM, and RCA. So, the overall complexity of this work is the combination of the complexities of these three components. But, a major contributor here is the deep learning model. The SegNet is used as a feature extractor. Hence, the estimated overall complexity is approximately Complexity of(CNN model) + Complexity of UCM)+ Complexity (RCA). The time complexity of SegNet is roughly equal to the number of parameters (30m) based on (Yi et al., 2019). Therefore, the time complexity is $O(N)$, where N is the number of parameters in SegNet.

3.1.3 Summary

This research work is to investigate the learnt features from ConvNets and to construct a scene classification technique called RCA. Experiments demonstrate that the RCA algorithm effectively enhances a scene labelling system's global label consistency.

3.2 Weaklier-Supervised Semantic Segmentation with Pyramid Scene Parsing Network

Semantic image segmentation is the essential task of CV. It requires dividing visual input into different meaningful interpretable categories. In this work image attribution and segmentation approach is proposed. It can identify complex objects present in an image. The proposed model starts with superpixelization using SLIC (Achanta et al., 2012). A MSF model (Li et al., 2019) produces an object seed heat map, and a SECT model (Li et al., 2019) generates pixel-level annotations. Lastly, the PSPNet model (Zhao et al., 2017) for developing the final semantic segmentation of the object. The proposed model was implemented, and compared with the earlier works, it excelled the performance score. The significant contributions of the proposed work are:

- Colloboartion of different compactness levels of superpixels in the SLIC model to determine the optimal superpixel size for improved accuracy of the model.
- Use of the PSPNet model for objet segmentation.
- Deploying Class Activation Mapping (CAM) model (Tagaris et al., 2019) to visualize the discriminative region used in the recognition process.
- Generating pseudolabel annotations using SECT methodology.

3.2.1 Methodology

The proposed model divides the input image into superpixels, or non-overlapping groups of pixels with similar features. The technique used to accomplish this is called SLIC. The super-pixelated image is then sent into the MSF model. When an image is supplied, a heat map corresponding to the image is generated using the CAM model, which expresses the pixel level reaction of each item class across superpixels. The generated heat map is ineffective in determining the object's outline or true structure. Thus, a SECT model is used for the pixel-level annotations. By merging the results of MSF and SECT, we effectively generate pseudo-supervised annotations for images. A LAB colour and texture diagram are created by merging the outputs of MSF and SLIC. The image's saliency-edge modification results in the suppression of noisy pixels. Simultaneously, color-texture edge-based inference is employed to mine sections of objects that are distant from seeds. The semantic segmentation network's supplied framework learns numerous object features from the images and accurately delineates them based on all the pixel-level weaklier supervised annotations collected. The suggested model accomplishes this by utilising a PSPNet. The use of PSPNet enables the aggregated representation of all distinct regional settings.

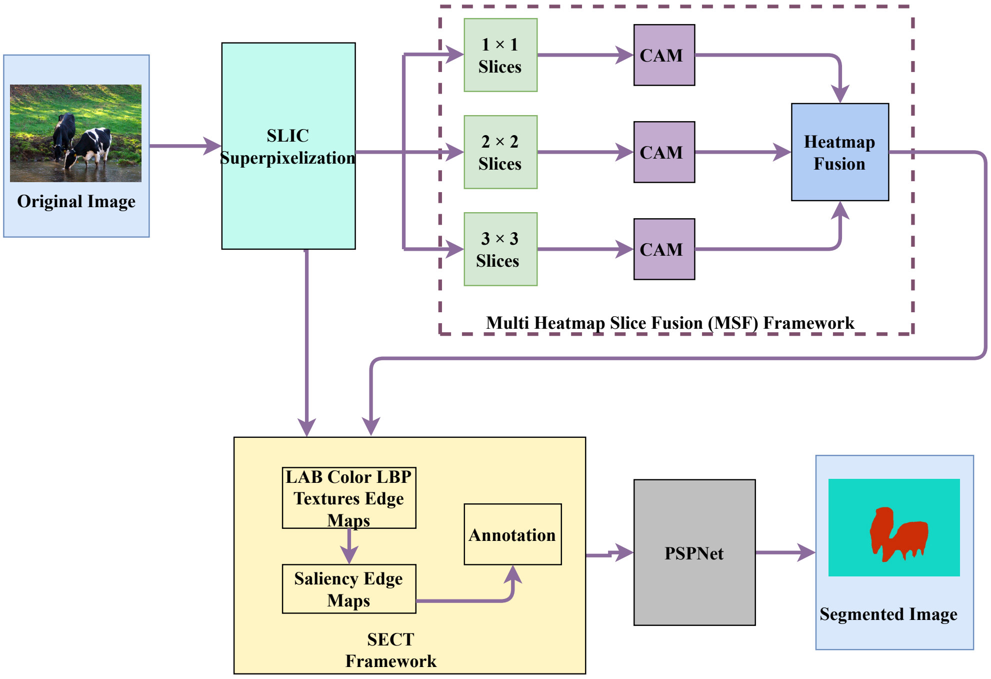


Figure 3.2: Workflow in the proposed approach for weaklier-supervised semantic segmentation.

Given a dataset with $N(N_s + N_m)$ samples and c object categories, here two parts have been created depending on how many object categories are present. Those images that contain a single category have been considered as N_s, while images that have multiple object types are considered as N_m. ResNetCAM-Keras (ResNet-50) model (Zhou et al., 2016), which is pre-trained on ImageNet (Russakovsky et al., 2015) provides the image as well as pixel-level priors to describe each dataset image, is used to generate the class activation map as feature expressions.

The proposed workflow for image segmentation is shown in Figure 3.2. First, the input images are over segmented into superpixels by using SLIC; next, a low-level analysis is done for low-level attributes of the objects in the image by combined heat maps. Pseudo annotations are generated with the MSF model combined later with the SECT modification methodology. Finally, accurate object-level annotation is obtained from pseudo annotation with iterative feature learning by employing the PSPNet model. Figure 3.3 depicts how an input image would appear at every stage, namely: a) Input image, b) SLIC, c) SLIC slices, d) and e) heatmaps, f) LAB color and texture, g) edge maps, and h) final output.

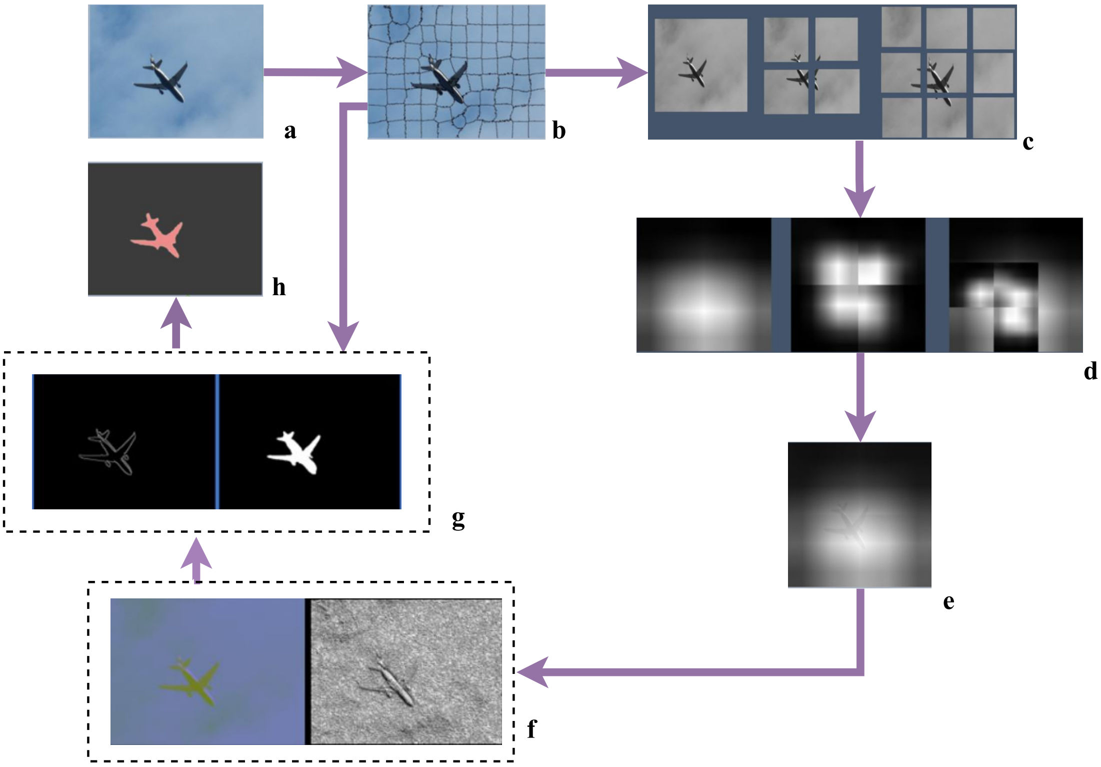


Figure 3.3: System flow diagram with input image through different steps of proposed weaklier-supervised semantic segmentation: a) Input image, b) SLIC, c) SLIC slices, d) and e) heatmaps, f) LAB color and texture, g) edge maps, h) final output.

3.2.1.1 Class Activation Mapping Model for Visual Interpretation

A typical choice for an object classification task is the CNN model, rehashed squares of convolution and max-pooling layers, trailed by at least two densely associated layers. The last layer is the dense layer with a softmax activation function and a node for every potential object class. The CNN models have the risk of overfitting to the training dataset due to many intermediate layers. So to avoid overfitting, dropout layers are used. In the last few years, experts have started employing Global Average Pooling (GAP) (Lin et al., 2013) to minimize overfitting. GAP layers help reduce the overall spatial dimensions of any 3D tensor, such as the layer that performs max pooling. GAP performs a very extreme level of dimensionality reduction compared to others.

ResNet-50 CAM-Keras (ResNet-50) is pre-trained on ImageNet, and it provides pixel level, image level prior description for every image. It is used to generate a class activation map as feature expressions. In this model, the GAP layer is included, then the softmax activation function is used as a densely connected layer, and it has 2048 activation maps of 7×7 dimensions each. Let the k^{th} activation map be represented as f_k where, $k \in \{1, \ldots, 2048\}$.

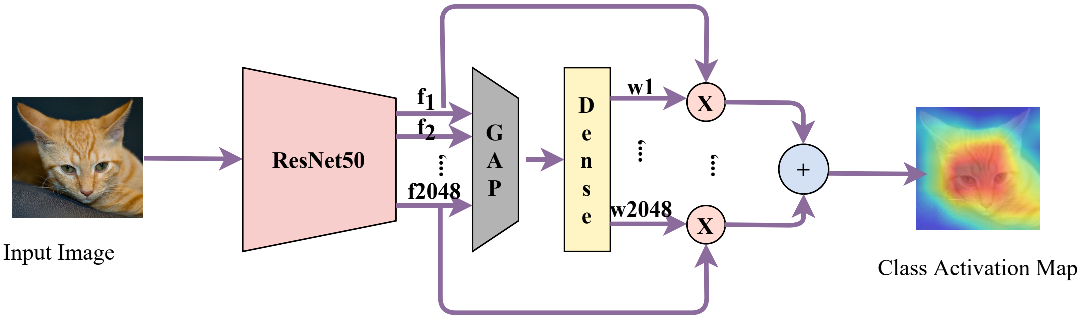


Figure 3.4: Architecture of ResNet-50 CAM model.

The average pooling, 2D GAP layer diminishes the former layer's size (1,1,2048) as it considers the average value of all the given feature maps. The inputs are merely flattened due to the next flatten layer, without changing any information contained in the previous GAP layer. The single node in the final dense layer is connected to each node present during the last flatten layer. The weight connecting the k^{th} node is considered W_k in the flatten layer concerning every output node and matching image category. The class activation map are obtained using Equation (3.8).

$$w_1 \times f_1 + w_2 \times f_2 + \ldots, + w_{2048} \times f_{2048} \tag{3.8}$$

Figure 3.4 shows the workflow in Resnet-50 CAM (Zhou et al., 2016). Many class enactment maps for pictures are obtained to investigate the segmentation capacity of ResNet-50. Bilinear upsampling is utilized to resize every activation map to 224×224. This brings about a class activation map having the size 224×224.

A weighted activation map is constructed for each image in a CAM. It aids in identifying the region that a CNN examines when classifying a picture. CAMs are trained unsupervised rather than supervised. This means that the objects do not need to be labeled manually, and the localization is a form of ”free” learning. The only change to the architecture is to remove the completely connected dense layers at the end to keep the spatial information confined in the output of the last convolution layer. Following that, a global average pooling layer is created. This layer is typically used for regularisation to prevent overfitting in the network. Finally, the output softmax layer is added, with as many neurons as classes to categorize. Implementing the class activation map technique depends on the addition of global average pooling layers following the final convolutional layer to spatially reduce the image dimensions and lower parameters, hence limiting over-fitting.

3.2.1.2 Accurate Superpixelization of Images

A superpixel may be defined as a group of pixels with common characteristics (such as pixel distance, pixel intensity, pixel color). SLIC is employed for superpixel generation. It accepts as its input a certain arbitrary number N of K superpixels, which are similar in size, almost equal. It generates superpixels based on specific similarity measures, such as pixel distance, pixel intensity, or pixel color. The reason for our choice was simple, though it was a deviation from the base paper (Li et al., 2019), which employs a contour-based detection algorithm for the same. As the original SLIC (Achanta et al., 2010), outputs compact superpixels that are regular and nearly uniform, with the minimum possible computational overhead, all while achieving excellent segmentation quality. This will be useful for more accurate results in the further steps. At the onset of the SLIC algorithm, at regular grid intervals S, K superpixel cluster centers are chosen, where $C_k = [l - k, a_k, b_k, x_k, y_k]$. The value of k ranges from $k = [1, K]$. The approximate size of any given superpixel would be N/K, and the area covered, or its spatial extent would be almost S^2. Based on the above, it is easy to say that any given

Algorithm 2: General structure of Super-pixelization algorithm

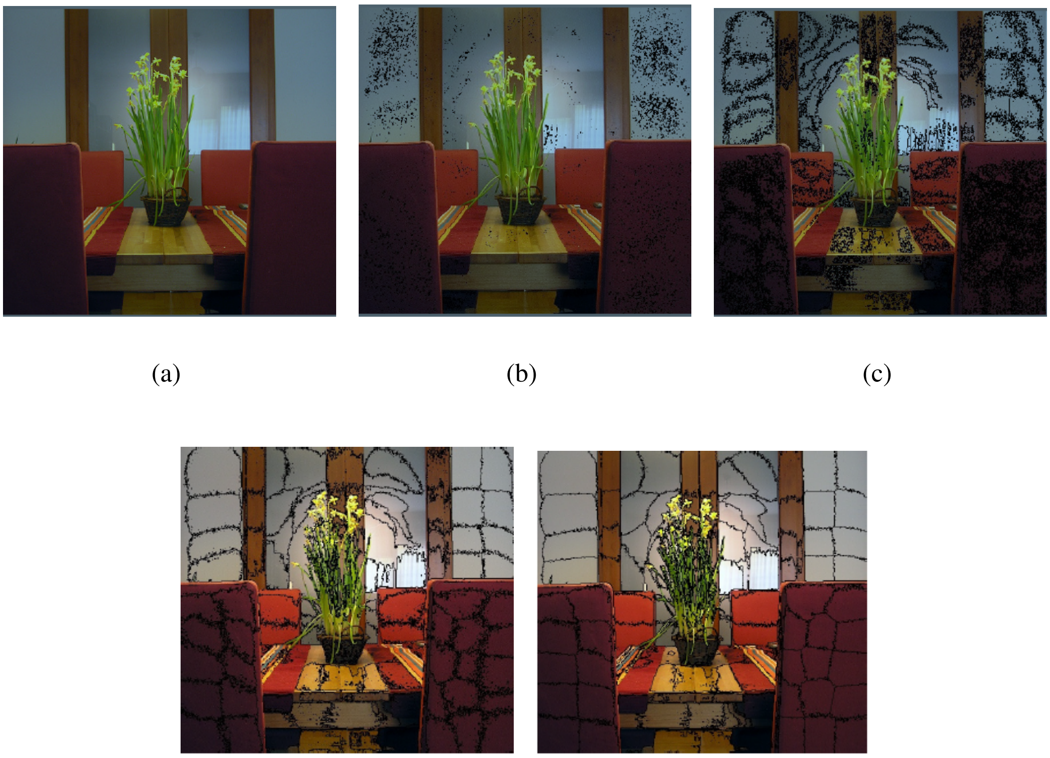
```
Input   : An Image with N pixels
Output  : Labeled pixels
Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular
  grid steps S.
Perturb cluster centers in an $n \times n$ neighborhood, to the lowest gradient
  position.
do
    for each cluster center $C_k$ do
        Assign the best matching pixels from a $2S \times 2S$ square neighborhood
          around the cluster center according to the distance measure.
    Compute new cluster centers and residual error E ( L1 distance between
      previous centers and recomputed centers)
while $E > threshold$;
Enforce connectivity.
```

pixel associated with given C_k lies in an area that is $2S \times 2S$ around C_k.

The distance measure used is defined by the Equation (3.9) as a normalized distance D_s to be used.

$$D_s = d_{lab} + (m/S) * d_{xy} \tag{3.9}$$

where, $d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$, $d_{xy} = \sqrt{((x_k - x_i)^2 + (y_k - y_i)^2)}$ and D_s is the sum of the xy plane distance (d_{xy}) and the lab distance (d_{lab}), normalized by the grid interval S. In order to control the compactness of the superpixels, m, a new variable, D_s is introduced. The greater is the chosen value of m, the more compact is the cluster. Here, we experimented with the different compactness values to check which size of superpixel cluster ensure the best and most accurate outputs for our object annotations. By varying the value of m, our experiments showed a significant variation in the experimental results. Figure 3.5 explains the SLIC output with $K = 100$ and different m values. The image (a) being the orininal image and (b), (c), (d), and (e) are SLIC output images with different m values.


(a) (b) (c)

(d) (e)

Figure 3.5: SLIC output for an image with K=100 and varying values of m: (a) Original image (b) m=1 (c) m=4 (d) m=10 (e) m=18

Once this is done, noisy pixels are removed as K evenly placed cluster centers are sampled such that the lowest gradient position can be found in a 3×3 region surrounding it. The centers are moved to new locations accordingly. The next step is image gradient computation and is computed using Equation (3.10).

$$G(x, y) = \|I(x+1, y)I(x-1, y)\|^2 + \|I(x, y+1)I(x, y-1)\|^2 \tag{3.10}$$

Where, the bracketed term is the L2 normalization, and $I(x, y)$ is the lab vector corresponding to the pixel at position (x, y). This accounts for both the color as well as the intensity of the pixels. After this, every cluster center C_k is tied to the closest image pixels, where an overlap is found between the search area and every pixel of the image. Once the algorithm has iterated through each image pixel and completed the association, the average lab XY vector of all the pixels belonging to the cluster is assigned as the new cluster center C_k. This association and recomputation are repeated until convergence is arrived upon. The remainder stray pixels are associated with joining disjoint segments with the largest neighboring cluster in the last step.

Once an accurate superpixelization of the images is completed, object labelling becomes a multi-label classification problem where the superpixels formed would have multiple attribute labels and an object label.

3.2.1.3 Generation of Multi Heat Map Slices Fusion

The image classification approach MSF provides fine coarse object details. An image from the PASCAL VOC2012 dataset (Everingham et al., 2012) containing k object categories is considered, and each class's pixel-level response is expressed in k corresponding heat maps generated as part of CAM model. The response center $\vec{V_{cent_c}}$ is selected, based on the discriminated category c. This is the result of the classification stage, and the dimensions are sorted in descending order of response values. Top m features dimensions are selected out of 1524, and corresponding m heat maps are generated. The value of threshold less than 1 is considered to choose the combinations of features to express object category c; other features are considered noise and removed.

A weighted average calculation is performed for m obtained heat maps h_i. To generate final heat map H_{ori_c} for category c we followed Equation (3.11).

$$H_{ori_c}(x, y) = \sum_{i=1}^{m} v_i \times h_i(x, y) \tag{3.11}$$

(x, y) is the representation of pixel in heat map and the response value v_i of h_i is set as weight.

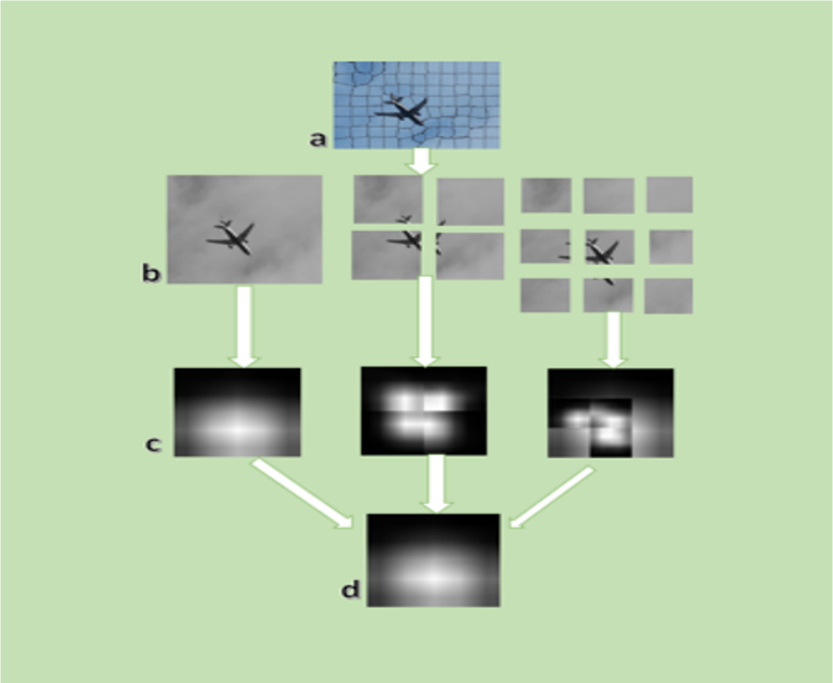


Figure 3.6: MSF flow diagram with input image, where (a) shows input image, (b) shows 1×1, 2×2 and 3×3 slicing, (c) shows 1×1, 2×2 and 3×3 fusion heatmaps, (d) shows the final combined heatmaps from MSF.

The MSF method is proposed to be the CAM model. The objects in an image with high and low scales will be descrimated in the process of gerearating heatmaps. In MSF, the original image is divided into 13 (i.e., 2×2 and 3×3) slices based on their original position and generated heat maps for each slice. Then we generate two heat maps H_{22_c} and H_{33_c} with the same size of H_{ori_c}. The weights for each slice and H_{ori_c} are assigned before the slicing process. This weights are utilized to compute the confident score of the response vector n and category c. The genearted heat maps for the highest intensity of pixels with the component H_{map_c} of category c as seen in Equation (3.12).

$$H_{map_c}(x, y) = max(H_{ori_c}, H_{22_c}, H_{33_c}) \tag{3.12}$$

The normalized heatmaps obtained with a maximum of 1. The method of utilizing more slices achieve better response value and inaccuarte regions are eliminated. A flow diagram of the entire MSF process with an input image is shown in Figure 3.6.

3.2.1.4 Saliency-Edge-Color-Texture

The fusion of multiple heat map slices provides heat for each category c as H_{map_c}. It has highlighted all the key components in addition to focusing on the fuselage. Even though it highlights all main aspects, it does not provide useful information on the overall structure and outline of object regions in the image. To overcome this problem, we have used the modification to add SECT methodology. The saliency map is more of changing an image's representation into image segmentation for more authentic learning.

In the proposed work, we have used the binarized saliency map to get a better result. The use of an Edge map helps in determining the outline of the object region. The model has used the Canny edge detection method for determining the edge of object regions. Color helps determine the region covered by the object and provides useful information about the object region's overall structure. We have converted RGB images to the LAB coloring scheme to get better results about structure. Lastly, the texture provides information in the spatial distribution and arrangement of colors or intensities in an image. For getting texture similarity, we have encoded the Local Binary Pattern (LBP) texture.

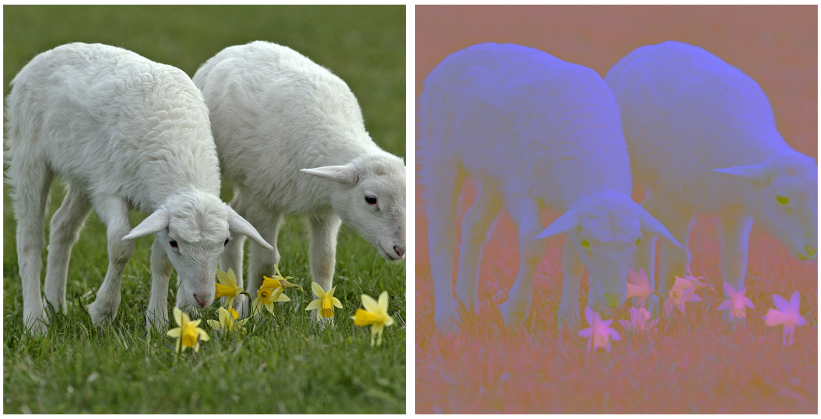


Figure 3.7: LAB color conversion of the image.

Figures 3.7, 3.8, 3.9, and 3.10 shows the LAB color conversion (Mokrzycki and Tatol, 2009), LBP texture mapping (Mokrzycki and Tatol, 2009), Canny Edge Mapping (Eshaghzadeh and Salehyan, 2016), and Binarized Saliency Map generation (Es-

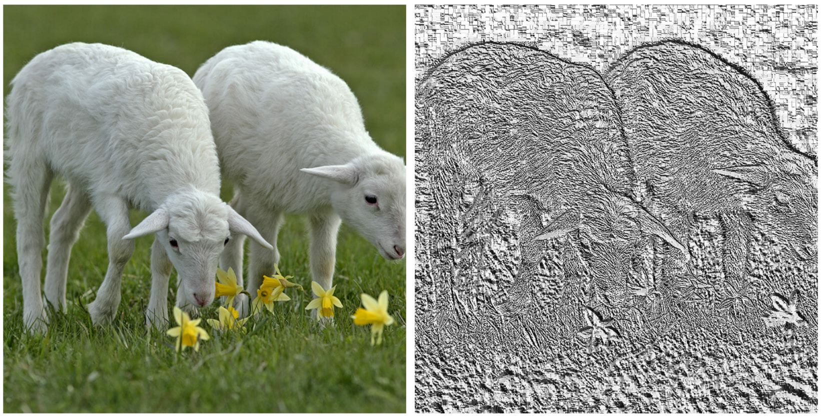


Figure 3.8: LBP Texture mapping of the image.

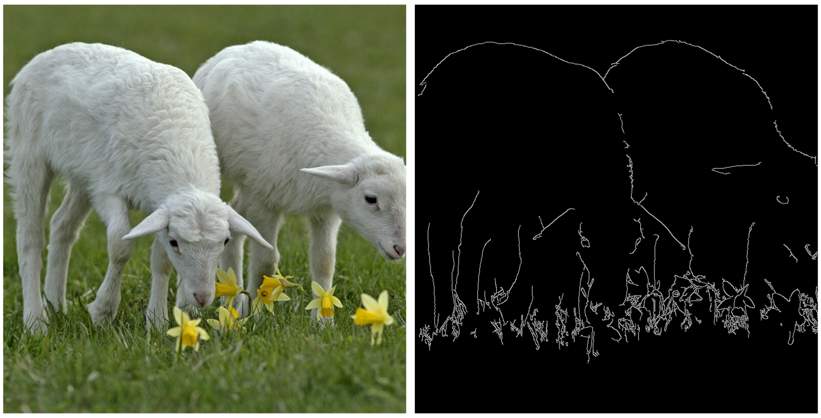


Figure 3.9: Canny edge mapping of the image.

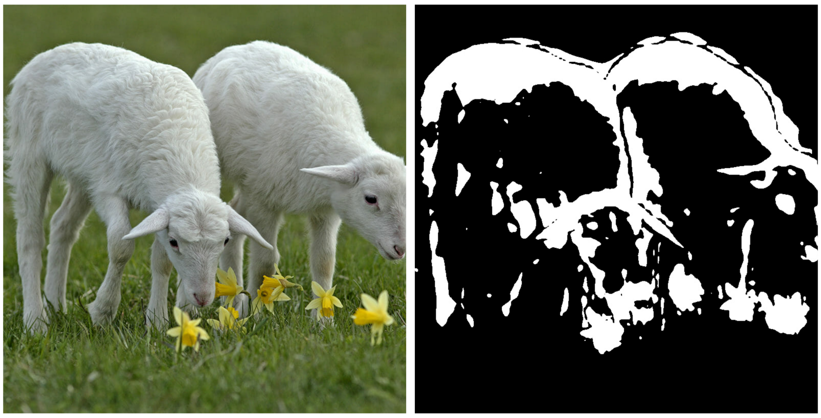


Figure 3.10: Binarized saliency map generation of the image.

haghzadeh and Salehyan, 2016), respectively, of the input image. The SECT modification provides pixel-level similarity analysis. It is based on low-level imaging attributes. The results were later combined with a heat map generated in MSF. The combination yields a pseudo supervised annotation. This way, we get the advantage of implementing a network learning method with different patterns via different strategies.

Other than components of SECT method, superpixalized image and heat map generated from MSF are key attributes in generating pseudo annotations. The superpixels have been generated using SLIC method. It divided image i into s superpixel regions sp_i(i= 1, 2, 3, ..., s). Then for these superpixel regions, we define average heat map as $H_{map_c}(sp_i)$, average saliency map as $S_{map_c}(sp_i)$ and average color-texture as $F_{map_c}(sp_i)$. Also $E_{rel}(sp_i,\ sp_j)$ for each pair of superpixels sp_i and sp_j on edge map is defined, which states corresponding edges appearing on the line connecting two superpixels.

Based on above definitions, similarity measures have been defined based on color-texture-edge correlation and saliency-edge correlation, as in Equations (3.13), and (3.14):

$$Sim_s(i,j) = \frac{(exp(-w_e.E_{rel}(sp_i, sp_j)))}{(1 + w_s.\|S_{map_c}(sp_i) - S_{map_c}(sp_j)\|_2^2)} \tag{3.13}$$

$$Sim_{ct}(i,j) = \frac{(exp(-w_e.E_{rel}(sp_i, sp_j)))}{exp(w_{ct}.\|F_{map_c}(sp_i) - F_{map_c}(sp_j)\|_2^2)} \tag{3.14}$$

$Sim_s(i,j)$, and $Sim_{ct}(i,j)$ gives the similarity measure between superpixel sp_i and sp_j. w_e, w_s, and w_{ct} are the weights of different imaging attributes, which are set as 3.5, 10, and 10. The value of the weights are obtained from different experiments conducted on the similarity measures till we got the satisfied result.

Using the above similarity measures, we have generated a weighted contribution of the global heat map, which provides an updated result of the superpixels response value. It is defined as Equation (3.15).

$$map_c{}^{new}(sp_i) = \sum_{j=1}^{s} Sim_*(i,j).H_{map_c}(sp_j) \tag{3.15}$$

After completion of update for all superpixels, value of $H_{map_c}(sp_i)$ is replaced by $H^{new}_{map_c}(sp_i)$. We have used two measurements using the Color-Texture-Edge method first, while the Saliency-Edge method performs the next operation to obtain the final heat map. The first operation mines the object region more effectively while later achieves noise suppression. This way, we achieve a more accurate object outline and higher precision for pseudo annotation image.

The confident score S_{fc} calculated at image classification stage is used in generating pseudo annotation along with three thresholds, which are category confident score th_s, foreground partition threshold th_{fg} and background partition threshold th_{bg}. We also calculate $h(x, y)$, the maximum response value among the k heat maps generated for an image containing k categories, which are weighed by the confident score S_{fc}. The pseudo-label annotation L_{map} is defined using Equation (3.16):

$$L_{map}(x,y) = \begin{cases} c(x,y) & h(x,y) > th_{fg} \wedge S_{fc} \geq th_s \\ 0 & h(x,y) < th_{bg} \\ 255 & else \end{cases} \tag{3.16}$$

Here $c(x, y)$ is defined as the corresponding category, 0 is used for background value, and 255 is used for the ignored region. We achieve different values of th_s, th_{fg} and th_{bg} based on different experiments. Figure 3.11 shows the overall architecture of the proposed PSPNet and their different modules with input and output image.

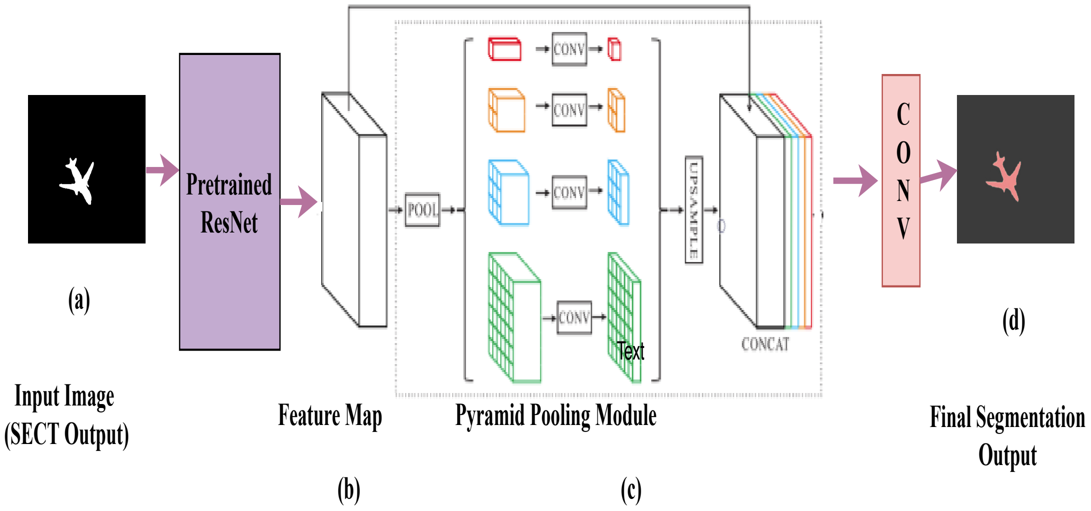


Figure 3.11: PSPNet Model: (a) input image, (b) feature is extracted using ResNet with diluted network, (c) After pooling for each feature, dimension reduction is done and after upsampling them concatenation is performed, (d) output image (Zhao et al., 2017).

The pseudo-label annotations L_{map} with image i are processed through PSPNet for semantic segmentation part. PSPNet is ‘Pyramid Scene Parsing Network,’ ranked first place in ImageNet Scene Parsing Challenge 2016. It provides a superior framework for pixel-level prediction tasks and effectively produces good quality results on the scene parsing task. The PSPNet has trained on PASCAL VOC 2012 dataset based on the images and pseudo-label annotations. This trained model has used later to find the object region's semantic segmentation present in the image.

In this work, ResNet50 (He et al., 2016) is used as a CNN model. It takes an image as input and extracts its features, and Global pooling has been highlighted to generate a single feature. The feature map generated is passed through a pyramid parsing module with four-level different pyramid scales to get sub-region average pooling, each with size 1×1, 2×2, 3×3, and 6×6, respectively. After each pyramid level, the 1×1 convolution layer reduces the dimension of the different levels with representation $N \times N$ to $N \times 1$ in global feature weight. Then features are fused under different pyramid scales. This fusion provides us with dimensionally reduced feature map regions, which are processed for upsampling. After which, they are concatenated for context aggregation. Finally, a convolution layer will output the final segmentation.

3.2.2 Experiments, Results and Discussion

3.2.2.1 Dataset

For the implementation of weaklier supervised semantic segmentation model, we consider PASCAL VOC2012 (Everingham et al., 2015) and ImageNet, which is available on PASCAL VOC2012 official site with various object and attributes. ResNetCAM-Keras (ResNet-50) is pre-trained on ImageNet, gives global and local image level description and generates class activation maps as feature descriptions. The split up of 1464, 1449, and 570 images are utilized for training, validation, and testing, respectively.

3.2.2.2 Experiments

Every original test image in the dataset goes through various steps to produce image maps to generate pseudo-label annotations. The image maps generated from the original image are heat maps through MSF and CAM, LAB color, LBP Texture, Saliency and Edge maps, and the image's superpixel version. The selected image from the dataset is passed through SLIC to generate superpixels in the image. The same image was sliced into 2×2 and 3×3 slices. Each slice and the original image go through the CAM model to generate heat maps for the given image. This heatmap is fused with the full operation to generate the MSF result. The same image is used to generate LBP texture, LAB color scheme, Canny edge map, and Saliency map. The LAB, LBP, and Edge map are combined with MSF results to produce a slight annotation combined with a saliency-edge map and superpixalized image to produce the pseudo annotation.

The Color-Texture-Edge and Saliency-Edge maps are combined separately using the similarity measures discussed above. These processes are done sequentially and using the three threshold values, th_s=0.15, th_{fg}=0.5 and th_{bg}=0.05. These values are selected based on experiments conducted while generating annotations. The pseudo annotation along with the original image processed through PSPNet.

PSPNet model with CNN as the backbone is trained with pseudo annotation and original image PASCAL VOC 2012 train dataset with 1449 images. The dataset is split bach-wise, each with size 2. The images are resized into 448×448 pixels. The dataset is then set to train on the CPU device with learning rate 0.01, the momentum of 0.9, maximum epochs of 10, and used standard per-pixel Softmax Cross-Entropy Loss to train PSPNet and then finally assigns weights to the layers and stored checkpoints. The trained PSPNet model provides a segmented image while clearly showing the object region and the outline.

3.2.2.3 Performance Analysis

The Intersection over Union (IoU) and Pixel Accuracy is the evaluation measures used to measure and analyze the performance of proposed models with the existing model. IoU defined in Equation (3.17) is an essential performance metrics to quantify the overlap percentage between the target mask and the prediction output. This metric measures overlapping factors between pixels.

$$\text{IoU} = \frac{\text{target } \cap \text{ prediction}}{\text{target } \cup \text{ prediction}} \tag{3.17}$$

We calculate the IoU score for each class and then averaged over all classes to provide a mean IoU [mIOU] score of our PSPNet model. The proposed work performs image level segmentation and compared with existing work (Li et al., 2019). .

Table 3.2 represents the IoU score for each class and mean IoU score for the PASCAL VOC2012 test dataset calculated for existing and proposed model i.e., Baseline and PSPNet model, respectively. The results from PSPNet have mIoU accuracy of

Table 3.2: Comparison of proposed method for weaklier semantic segmentation using SLIC, MSF, SECT, and PSPNet with the state-of-the-art method.

Class Labels	IoU Baseline (Li et al., 2019)	IoU Proposed
(Background)	82.7	85.6
(Aeroplane)	70.1	72.1
(Bicycle)	28.2	27.4
(Bird)	48.5	51.6
(Boat)	37.0	41.3
(Bottle)	51.1	55.2
(Bus)	71.1	75.8
(Car)	69.8	71.9
(Cat)	56.4	17.2
(Chair)	10.1	42.7
(Cow)	46.8	34.4
(Dining Table)	36.9	52.8
(Dog)	39.0	40.3
(Horse)	47.5	61.7
(Motorbike)	73.8	45.7
(Person)	49.3	74.9
(Potted Plant)	36.7	43.6
(Sheep)	67.0	54.1
(Sofa)	30.7	31.8
(Train)	47.1	67.4
(TV)	36.0	53.5
(mIoU)	49.3	52.4

52.4% on PASCAL VOC 2012 test dataset, which is better than the Baseline with mIoU accuracy of 49.3%. The proposed approach achieved state-of-the-art performance.

Pixel Accuracy is a part of performance metrics to report the percent of pixels in the correctly classified image. Here the pixel accuracy is commonly noted for image class separately for the dataset. The Pixel Accuracy calculated using Equation (3.18).

$$\text{Pixel Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.18}$$

The True Positive (TP) in the Equation (3.18) represents a pixel that belongs to the given class. The True Negative (TN) represents a pixel that does not belong to the given class, a False Positive (FP) indicates a predicted output had no associated target mask and a False Negative (FN) indicates a target mask had no associated predicted output.

The trained PSPNet model provides us with a segmented image while clearly showing the object region and the outline. Some results obtained for PASCAL VOC2012 test dataset from PSPNet with Pixel Accuracy are shown in Figure 3.12 and Figure 3.13. Each figure comprises the input image, predicted output, and ground truth, respectively.

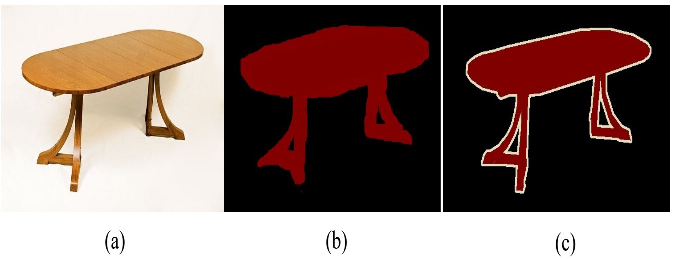

(a) (b) (c)

Figure 3.12: Pixel Accuracy for the given input-1 is 0.673. (a) input image (b)predicted output (c) ground truth

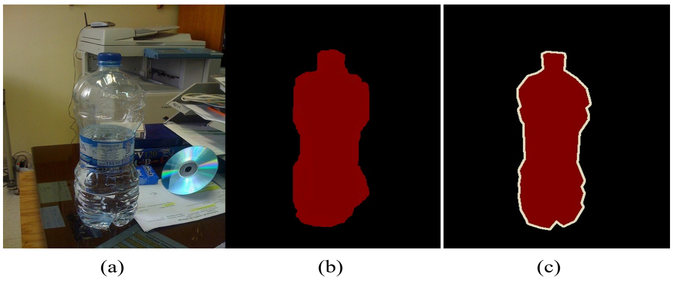

(a) (b) (c)

Figure 3.13: Pixel Accuracy for the given input-2 is 0.629. (a) input image (b)predicted output (c) ground truth

Time Complexity Analysis

The complexity of training a neural network model for each epoch is defined by $T_E = N_b T_{fb}$ (Justus et al., 2018), where N_b, and T_{fb} are the number of batches and approximate time required to backward and forward passes. T_{fb} is proportional to the number of parameters in the deep learning model. The proposed approach has three components. Hence, the complexity is defined by SLIC, three parallel CAM models, and training a PSPNet model (an extension of CAM). For CAM, we have used ResNet50. Therefore, the overall complexity is $O(complexity\, of\, (SLIC) + 3 * O(complexity\, of\, (CAM)) + e * Nb * O(complexity of\, PSPNet)$. This is approximately equal to $O(N) + 3 * O(P) + e * T_b * N_b * O(P)$, where N, P, e, Nb, and T_b represent the number of pixels in the image, parameters in ResNet50, epochs, batches, and time estimation for forward and backward passes, respectively.

3.2.3 Summary

This work's main objective was to learn the appearance of an object present in it with the proposed image attribution and segmentation model. First, superpixelization of the image is done through SLIC. A MSF model produced an object seed heat map, and a SECT model does the pixel level annotation. Object features are learned, and semantic segmentation is carried through the iterative PSPNet model. We also deployed a CAM model to visualize the images. The proposed model outperformed for class level semantic segmentation against some of the existing models. The proposed approach achieved state-of-the-art performance on the benchmark dataset. Further, this model can be improved to yield an object-level semantic segmentation and developed to identify sharp edges more clearly in the segmented image.

3.3 Weakly Supervised Image Annotation and Segmentation

The many components of image processing, such as object identification, object categorization, image segmentation, and attribute learning, are inextricably linked. This research work suggested a BN technique (Hanea et al., 2015) for solving complicated visual tasks by utilising the non-parametric property to constrain the model. We build a MRF-CRP that employs MRF (Kato and Pong, 2006; Venmathi et al., 2019) at the low level and CRP (Blei and Frazier, 2010) at the high level. The suggested approach automatically discovers and incorporates associations between distinct object and attribute classes. The input image is clustered into individual components using the MRF, and then the components are merged, and the image-attribute association is generated using the CRP. Experiments on the Berkeley Segmentation dataset (Arbelaez et al., 2011) demonstrated that the proposed model outperforms other weakly supervised models already available. The following are the most significant contributions made by the proposed work.

- Instead of building the model for a specific type of dataset, the objective of this work is to increase the robustness of the proposed model for multiple datasets.
- In this proposed approach, we tried to identify the association between objects and attributes rather than just detection of objects and attributes.
- Accurately detecting the multiple main objects with non-initialized parameters present in a weakly-labeled image.

- A hierarchical model that stacks a specific local-level method with a global-level method.
- Training the model for object and attribute annotation from weak image-level so that the model learns to describe the object-attribute association.

3.3.1 Methodology

The architecture of the proposed approach for image segmentation is shown in Figure 3.14. It consists of the three stages between the weakly annotated image and the final segmented image with the object-attribute association and are listed below.

- The first stage consists of Speeded Up Robust Feature (SURF) (Bay et al., 2006) + Color Algorithm for feature extraction.
- The second stage consists of model selection in order to constraint the regulation of the model.
- The third stage is the final stage which results in object-attribute association using a hierarchically stacked model.

A detailed explanation of above mentioned three stages are given in the subsequent sections.

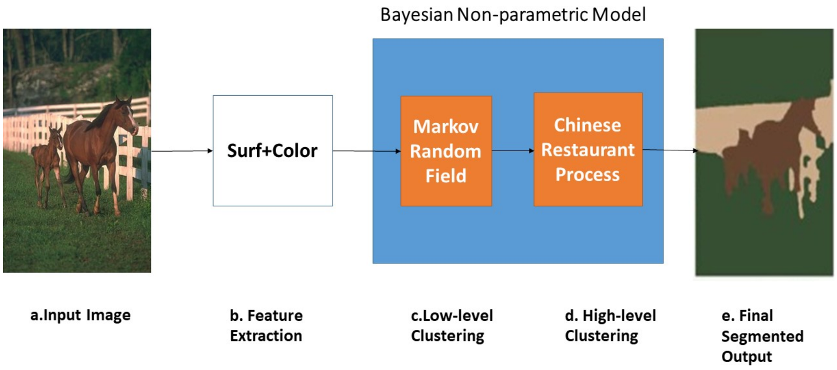


Figure 3.14: Architecture of the proposed WS-MRF-CRP model for weakly supervised image segmentation.

3.3.1.1 Feature Extraction

All the 500 images one by one are decomposed into different super-pixels that are the segmented patches of the image, which generally contain the objects of interest that makes the further process easy. This is done using the algorithm known as Hierarchical Segmentation. Each segment can be represented using two normalized features of the histogram:- SURF (Bay et al., 2006) and color. Each image (i) from the Berkeley Segmentation Dataset (BSD500) from the training dataset is firstly decomposed into N(i) super-pixels using the algorithm of hierarchical segmentation. Further, the major problem of joining the object with its attribute is narrowed down to the simple problem of multi-label classification for each super-pixel. The SURF feature extraction process is shown in Figure 3.15.

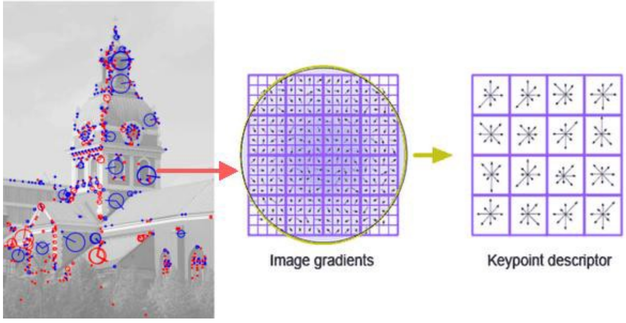


Figure 3.15: SURF feature extraction.

The Algorithm (Ordóñez et al., 2018) for the keypoint detection is shown in Algorithm 3:

3.3.1.2 Model Selection

In the traditional clustering methods, a fixed number of clusters present in the data. It addresses the first objective of building a model with growing parameters. The reason for the need for a non-parametric model is to increase the robustness of the model so that we don't have to pre-define the number of clusters present in the image. The meaning of non-parametric is a model that has a growing/infinite number of parameters. The Bayesian is a stochastic probability distribution model, helps with the probability of the parameters given the input data.

$$P(parameters|data) \propto P(data|parameters)P(parameters) \tag{3.19}$$

The reasons considered while selecting the model are to reduce the expensive computations and prevent underfitting and overfitting. The computation cost of Bayesian Non-parametric is less as compared to others. Also, a well-specified don't have an issue of overfitting and, being a non-parametric model having a growing amount of data

```
Algorithm 3: Algorithm for the keypoint detection
Input: Set of selected bands for images.
Output: A set of keypoints K for each selected band.
Parameters: Numbers of sublevels $N_{sub}$
  Calculate octaves $N_{oct}$ according to image size
  Upsample images to get images whose size is divisible by number of octaves $N_{oct}$.
  for <band b in images> do
      Upsample band using bilinear interpolation.
        Build pyramidal space space.
        Smooth using Gaussian filter.
        Compute contrast factor $k$ from gradient histogram of smoothed band.
        for $o \leftarrow 1, N_{sub}$ do
            Subsample last sublevel image by factor of 2.
              for $s \leftarrow 1, N_{sub}$ do
                  Use Gaussian filter for smoothing.
                    Compute conductivity $g$ .
                    Discretized nonlinear diffusion equation.

            end
      end
      Locate keypoints in scale space.
        Compute determinant of Hessian matrix.
        Detect keypoints by searching for points that are maxima of their
        $neighbourhood \leftarrow K_1^b, K_2^b$.
        Refine position and scale of each keypoint.
end
```

prevents overfitting.

In Bayesian Non-parametric, there are various classes, but the one used extensively is the Dirichlet Process (DP) (Wang and Zhao, 2017). The DP gives the probability distributions over probability measures. G is the distribution over probability measure provided by Equation (3.20).

$$G \sim DP(\alpha, G_0) \tag{3.20}$$

For the partitions over the space $(A1, ...An)$ shown in Figure 3.16 is given by Equation (3.21).

$$(G(A1), ..., G(AK)) \sim Dirichlet(\alpha G_0(A1), ..., \alpha G_0(An)) \tag{3.21}$$

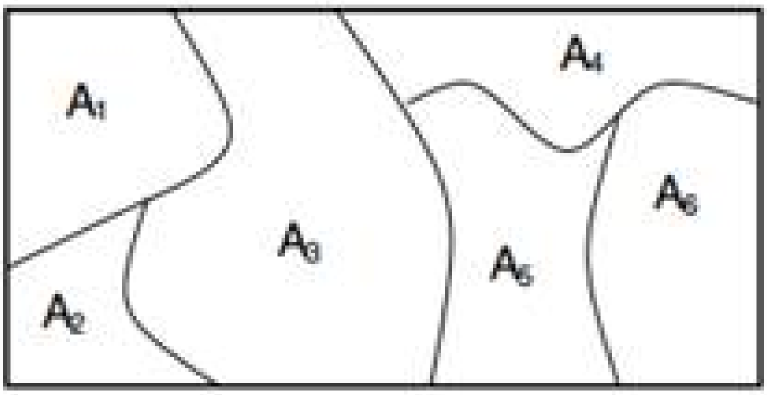


Figure 3.16: Finite partition using Dirichlet (Wang and Zhao, 2017)

The DP has following two parameters:

1. G_0- which is the base distribution of DP
2. α- which is the inverse variance of DP which is given by

$$V[G(A)] = G_0(A)(1 - G_0(A))/(\alpha + 1) \tag{3.22}$$

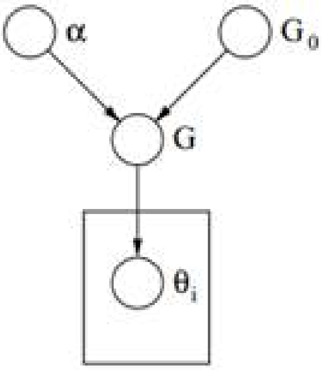


Figure 3.17: Graphical model of Dirichlet process (Guimarães et al., 2012)

The Figure 3.17 represents graphical view of DP. Here, G_0 is the base distribution, α is a scalar hyperparameter, G a random distribution over parameter space, and θ_i is a parameter vector that is drawn from the G distribution, and it is an element of space (Guimarães et al., 2012).

Thus, the DP is a class of Bayesian Non-Parametric is the most suitable for our application considering the fact that the number of parameters is not fixed and always growing. Also, the various ways of representing BN, including CRP, gives the user an easy way to implement the model.

3.3.1.3 Object-Attribute Association and Image Segmentation

Chinese Restaurant Problem

The CRP is one of the representation forms of the DP(Wang and Zhao, 2017). The various representation of the DP examines the problem from different points of view resulting in different formulation but are mathematically equivalent. CRP makes use of a Chinese restaurant analogy, which tells the probability which table will be occupied by a new customer. The probability of the table chosen by i^{th} customer:

$$p(y_i = k|y_{1:(i-1)}, \alpha) \propto \begin{cases} n_k, & \text{for } k \leq K \\ \alpha, & \text{for } k = K + 1 \end{cases} \tag{3.23}$$

where n_k : number of customers seated at table k

K : number of tables occupied by the i-1 customers

α :Dispersion value of DP

Whenever a new customer comes, he/she can sit at a new table or a previously occupied table. As the number of seated customer increases the probability of the new customer seating on an occupied table increase. The model used the distance between the observed data items to cluster data in a non-parametric manner. A new customer chooses an occupied table according to the familiarity. Combining this with MRP address the 2^{nd} objective of building a hierarchical model. The CRP acts as a higher level method for clustering in the hierarchical model. Figure 3.18 gives an illustration of CRP. The approach operates with, customer selects either another customer or no customer. In our approach, the clusters are equivalent to the tables and customer is equivalent to the integers. This helps in clustering at the global level where clusters act like tables and help in global level clustering.

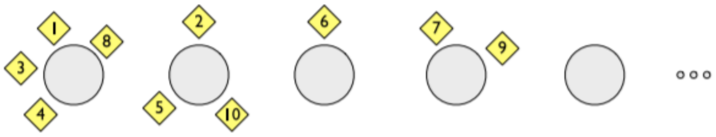


Figure 3.18: Illustration of CRP (Blei and Frazier, 2010)

Markov Random Field

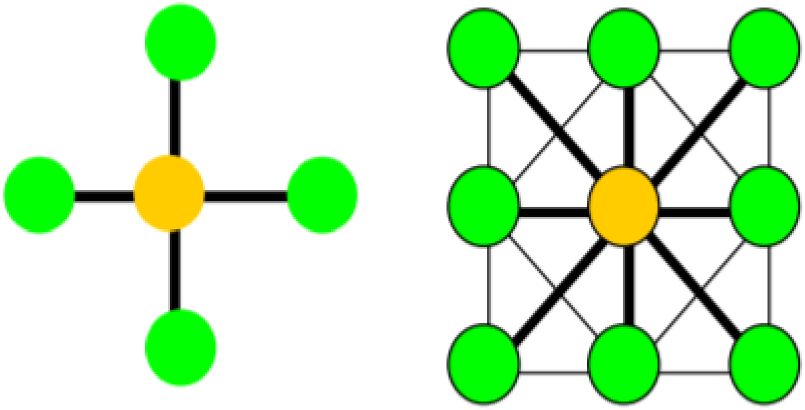
MRF gives the conditional probability in neighbourhood of each variable

$$p(y_i|y_{-i}) = p(y_i|y_{\partial i}) \tag{3.24}$$

y_{-i} gives the vertices configuration without i and ∂ gives the set of neighbouring vertices of i. The probability of the configuration is obtained by

$$p(y_{1:N}) = \frac{1}{Z} exp\Big(- \sum_{\mu \in U} V_\mu(y_{1:N}) \Big) \tag{3.25}$$

where Z is normalisation constant, V is clique potential function, and U is the cliques set. However in most application we only consider first and second order clique shown in Figure 3.19. The first-order checks whether the data item matches with the corresponding neighbours and second-order smooth prior of the model.



(a) (b)

Figure 3.19: The representation graph of 1^{st} and 2^{nd} order clique (Kato and Pong, 2006; Venmathi et al., 2019)

This property of MRF helps in the local level clustering that is the first level of clustering and it's output is used as an input for global clustering at global level using CRP. It address the 2^{nd} objective of building a hierarchical model.

3.3.1.4 Combined Hierarchical Model

The proposed approach is a hierarchical model which uses MRF for local clustering and CRP for global clustering with infinite number of tables(clusters). In this case, to have an infinite number of clusters depending on the endless number of latent features, we choose CRP over Indian Buffet Process (IBP) (Griffiths and Ghahramani, 2011) due to its efficient working in finding the infinite clusters analogous to the limitless tables in the Chinese Restaurant. The proposed approach uses objects and attributes as latent factors and helps capture their correlations within it and across the super-pixels. The CRP instead of the IBP so as to have better accuracy in the detection of the infinite latent features and their association with the objects. The data's input image is initially clustered into the lower level, then the higher level CRP can be used to merge all various components into the larger clusters.

This clustering of different objects and attribute cluster at the global level addresses the 3^{rd} objective of the object-attribute association. The building of a hierarchical model helps build a model that can find the association between objects and attributes using the global clustering using CRP. The relation in the hierarchical model is shown in the Figure 3.20. The components that are formed previously are mapped to the customers like the final mapping of the clusters to the tables. The model has the CRP at the higher level and then the MRF at the lower level. The output of the MRF algorithm is basically a clustering at the local level. To form a reasonably big cluster, an extra layer of global CRP is used to the group of the already clustered at MRF to achieve even better results. The main aim is to associate each image/super-pixel with the corresponding latent factor vector which will then basically correspond to the other objects, the attributes/unannotated attribute present in that image using MRF-CRP.

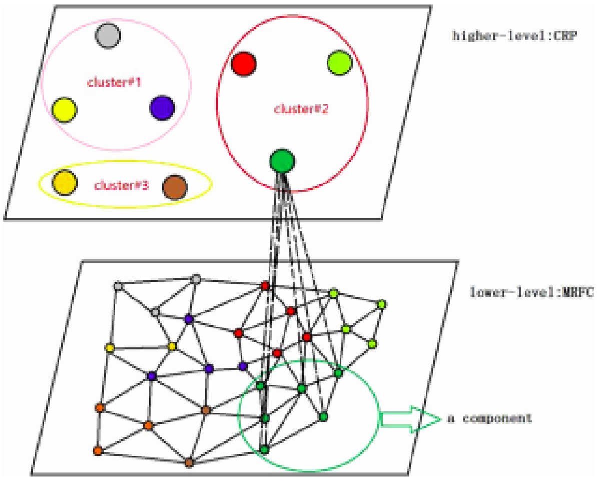


Figure 3.20: Relation between the component representation and the representation of cluster assignment in the hierarchical models (Kato and Pong, 2006; Venmathi et al., 2019)

3.3.2 Experiments, Results and Discussion

3.3.2.1 Dataset

2D Images of the BSD500 (Arbelaez et al., 2011) are used to evaluate the performance of the proposed approach. The dataset has 500 natural images and ground-truth annotations made by people. The data is clearly split into different sets called train, validation, and test. For training and validation, the original 300 images are used, and for testing, 200 new images and human annotations are added. On average, five different people ”segmented” each image.

3.3.2.2 Experiments

The proposed approach describes briefly the different objective from the original image to the final segmented image with object-attribute association i.e., to extract the features, associate the objects with various different attributes and then finally the clustering of the objects with the similar attributes in the same color so as to semantically segment it.

SURF is used to extract the features from all the 2D images. It does its operations by using the box filters. It is used for feature extraction for which, it uses the Hessian matrix approximation. The BN is used to identify the relation and is a probabilistic model, which is eventually represented using CRP. We can use this in the MRF stacked with CRP to form the clusters of the objects with similar attributes. Once the clusters are formed, they merge together to form even bigger clusters which can directly be used for the semantic segmentation.

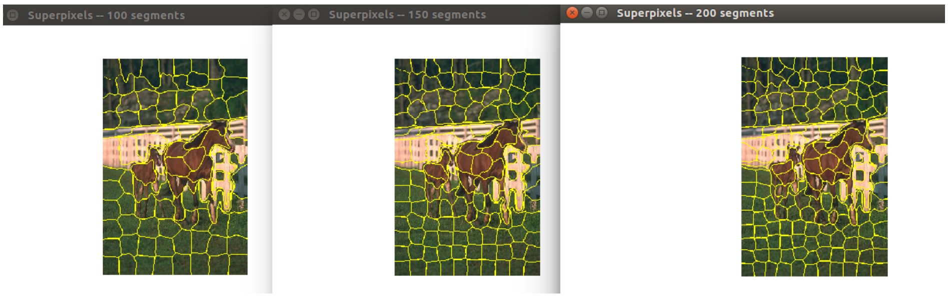

Superpixels -- 100 segments
Superpixels -- 150 segments
Superpixels -- 200 segments

Figure 3.21: An image segmented into 100, 150, 200 segments by super-pixeling

Figure 3.21 shows the different super-pixels of an image from the BSD500 dataset. This is done using the Hierarchical Segmentation algorithm. Initially, every image is decomposed into its various different super-pixels that are the segmented patches of the image which generally contain the objects of interest and makes further process easy.

The major problem of joining the object with their attribute is basically narrowed down to the simple problem of multi-label classification for each of the super-pixel. On the left column, the image has 100 super-pixels, 150 in the middle, and 200 super-pixels in the last column of Figure 3.21. We can see that other than multiple segments of the horse, but a segment only consists of either the horse, grass, or the fence, and does not combine two or more objects.

In Figure 3.22, SURF has been applied to the image. The second image shows the feature points detected by the approach, and the third image shows the vector points of all the features and their magnitude. The larger the radius of the circle, the greater is the magnitude of the vector point. The vector point to the direction of the feature change to map out the exact number of good features that can be used to predict the different points present in the image later on to be segregated into different segments based on their attributes.

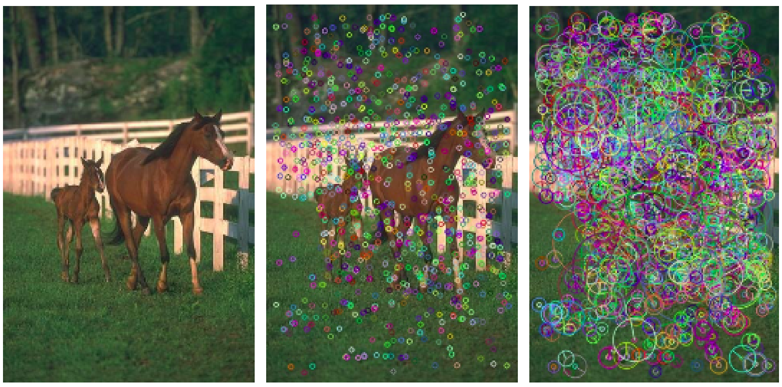


Figure 3.22: The stages of SURF algorithm

CRP is a type of BN method without the exact number of latent features. The latent features are binary. An object either does or does not possess a feature. It works well for an infinite number of clusters. MRF model provides a simple and effective way to model the spatial dependencies in image pixels. It is used to model the connection between two neighbor pixels.

Figure 3.23 shows us the intermediate stages of the proposed approach. The image of the left is the original image from the BSD500 dataset. For the given image initially SURF is applied to extract out all the features. The Hierarchical algorithm, including CRP and MRF, is used stacked to each other from the object-attribute association. It does this by segmenting the image into various different clusters. As seen in the second

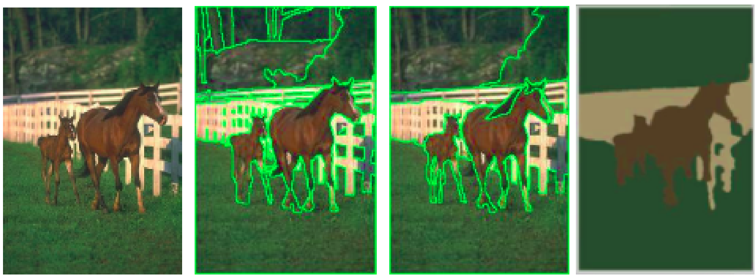


Figure 3.23: Sample images of different stages of proposed MRF-CRP model for weakly supervised image segmentationSegmentation

image of the Figure 3.23, it has 61 segments in the picture. These clusters are formed depending on the super-pixel-values. The clusters merge into each other to form bigger clusters, and eventually, when no such merging is possible, each cluster is colored differently to show the semantic segmentation. The 61 clusters in the second image merge together to form only 13 segments in the third image of Figure 3.23 and then finally coloring different clusters to get the final output which is semantically segmented as shown in the fourth image of Figure 3.23.

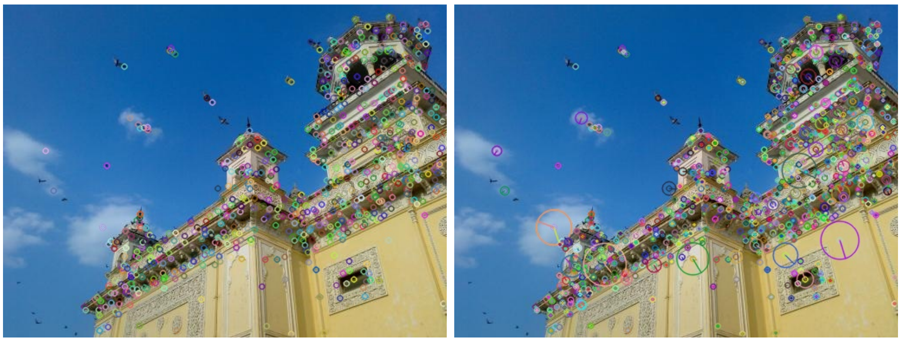


Figure 3.24: Result of SIFT + Colour

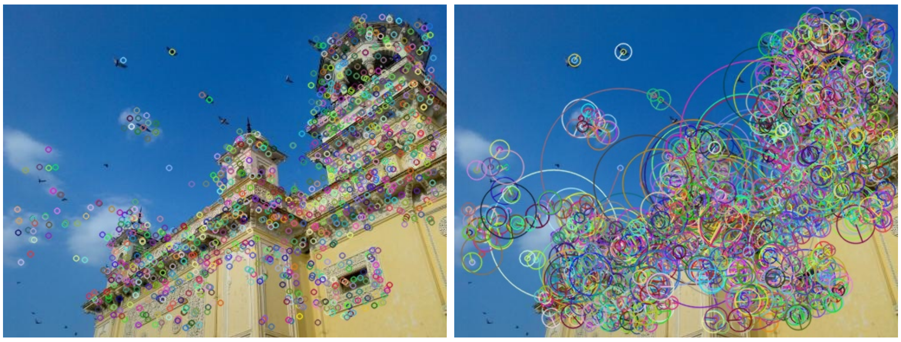


Figure 3.25: Result of SURF + Colour

As observed through the above images, SURF+Colour does a much accurate job than SIFT (Lowe, 2004) + Colour at finding the key points and their vectors. SURF +

Colour has been implemented using the OpenCV framework in Python. Since both of these algorithms are comparatively slow, we used Threading to reduce the time taken to detect features in the image. This has resulted in significant improvement in their performance.

Time was measured for each of the feature detection algorithms by applying threading (2 threads in separate processes) and without threading on a video of 2 mins 48 secs. The result obtained is shown in Table 3.3.

Table 3.3: Comparison of the performance of SIFT and SURF with and without threading

SIFT + Color without Threading	**SIFT + Color with Threading**	**SURF + Color without Threading**	**SURF + Color with Threading**
12 mins 26 secs	9 mins 56 secs	8 mins 17 secs	6 mins 13 secs
84 features per frame	84 features per frame	154 features per frame	154 features per frame

The final result of the proposed approach is shown in Figure 3.26. The original image is given as an input on the left side, and the final output image on the right side of Figure 3.26 consists of the object-attribute association. The intermediate stages of the processing and the proposed approach is shown in Figure 3.23. SURF was used over Scale Invariant Feature Transform (SIFT) for the purpose of feature extraction from the image because the performance and the time taken by SURF was better than that of SIFT. CRP was used over the MRF in the hierarchy as it was found to have better performance, as shown in the Table 3.4. The clusters formed in the intermediate stages merge together to form bigger clusters depending on their association as given by the Bayesian Probabilistic model, which tells about the association between the objects and the attributes by clustering using CRP.

As can be seen from the Figure 3.27 the difference in segmentation when using IBP and CRP. The segmentation performed in CRP is better due to the better clustering as compared to the IBP. As proved later by the performance analysis Table 3.4 the Probabilistic Rand Index (PRI) of MRF-CRP is 0.79 while it is 0.75 for MRF-IBP. PRI is a measure of the similarity among the clusterings of the data. It basically calculates the number of correct decisions made by the proposed approach, since the score of CRP is better than that of IBP, CRP with MRF is better than IBP with MRF.

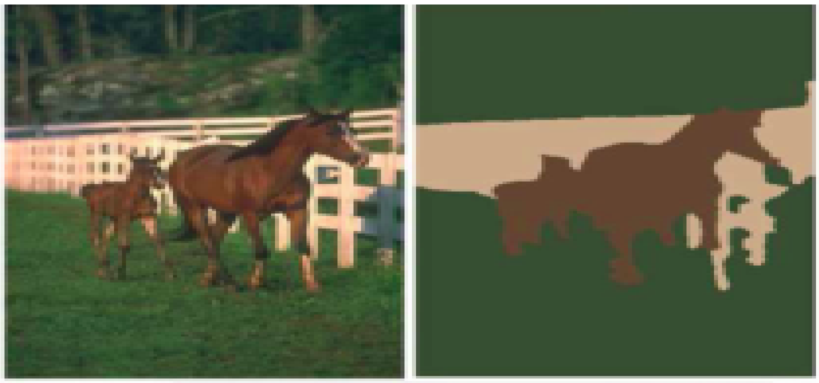


Figure 3.26: Objects and their attributes shown together in the same color

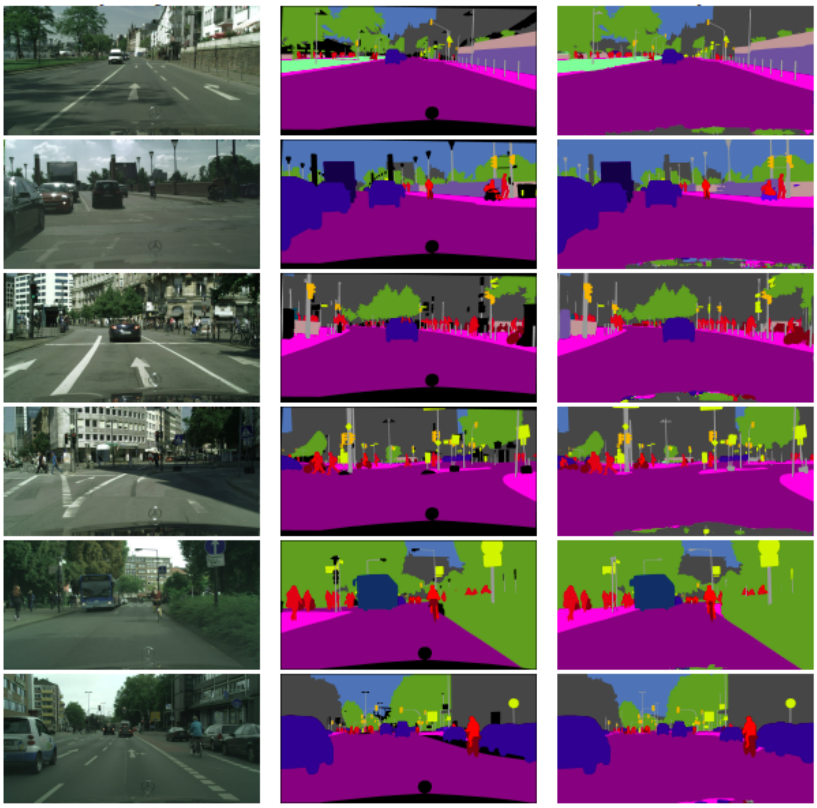


Figure 3.27: Comparison between IBP and CRP.

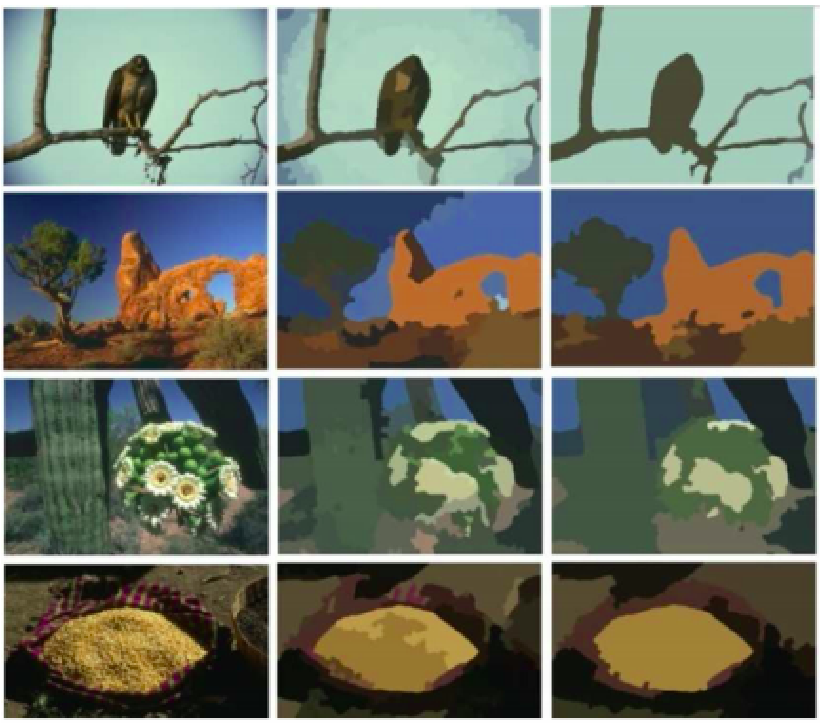


Figure 3.28: Semantic Segmentation of the image with and without the CRP. First column on the left is the actual picture from the dataset, second one is without CRP and then the last is our actual methodology.

As seen in the Figure 3.28, MRF stacked with CRP works better for the clustering of the images and the segmentation performed is much better. Without the CRP the clustering is not good and hence the segmentation is not very good which is better when CRP is used.

The 2^{nd} image from the Figure 3.28 also shows the output of the image with only using the MRF algorithm. From the results shown in Figure 3.28, it can be noticed and seen that our proposed approach without using the CRP algorithm, basically over-segmented the images into a large number of components to be assured of the accuracy of the local level clustering while the overall model i.e. shown in the third image. MRF-CRP has segmented the images into proper and very meaningful regions. It can be seen as being blurry and having some segments which are not very well defined. That's why the CRP algorithm is used to enhance the segments and hence improve the segmentation.

3.3.2.3 Performance Analysis

The performance of the proposed approach is compared to the other existing methods used for the semantic segmentation on the parameters like PRI, F1-measure and the global consistency error. The algorithms to which proposed approach is compared are MRF without CRP, DD-CRP (Blei and Frazier, 2010) and MRF with IBP as shown in Table 3.4. The comparison clearly indicates that the proposed approach works the best in all the 3 parameters. The dataset used for performance analysis in the Table 3.4 is the BSD500.

Table 3.4 compares parameters PRI, F1-measure and Global Consistency Error (GCE) respectively. The PRI calculation is as shown in Equation (3.26). Where, (3.26), TP is the true positive, TN is true negative, FP is false positive and FN is false negative. PRI calculates the number of correct decisions made by the algorithm, since the score of MRF-CRP (0.79) is better than that of any other methodology i.e. MRF(0.72), DD-CRP (0.69), MRF-IBP (0.75), proposed approach is the most efficient. The higher value of the PRI is indicative of the fact that MRF-CRP is most efficient.

$$PRI = \frac{TP + TN}{TP + TN + FN + FP} \tag{3.26}$$

F1-measure is the measurement of the accuracy of the test images. It involves both the calculation of Recall as well as the Precision. Having higher F-measure is evidence of having an efficient algorithm. The F-measure of the proposed approach i.e. MRF-CRP is 0.71 is the higher as compared to all the other algorithms which have scores 0.66, 0.57, 0.67. The expression to calculate the F1-measure is as follows:

$$F1 - measure = \frac{(1+\beta^2) * true\ positive}{((1+\beta^2) * true\ positive) + (\beta^2 * false\ negative) + (false\ positive)} \tag{3.27}$$

The GCE makes an assumption that one of the segmentation is a refinement for the another one. It measures the extent to which one of the segmentation is seen as a betterment of the other. Segmentation is basically the distribution of different pixels in different sets.

First, we introduce the local refinement error $E(S_1, S_2, p_i)$, which assesses the degree to which two segmentations S_1 and S_2 agree at pixel p. Let $R(S, p)$ represent the group of pixels in segmentation S that belong to the same segment as pixel p. Where $|.|$ represents cardinality and $. \setminus .$ represents set difference. This quantity is defined in two directions per pixel, hence it is not symmetric. $E(S_1, S_2, p_i)$ is 0 when S_1 is a complete refinement of S_2 and 1 otherwise.

$$E(S_1, S_2, p_i) = \frac{(|R(S_1, p_i) \setminus R(S_2, p_i)|)}{|R(S_1, p_i)|} \tag{3.28}$$

$$GCE(S1, S2) = \frac{1}{n} min \left\{ \sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i) \right\} \tag{3.29}$$

Segmentation here measures the error the two segmentations $S1$, $S2$. It takes $S1$, $S2$ as the input parameter and outputs anything in the range [0,1]. In order to avoid the GCE, all local refinements must point in the same direction, namely from one segmentation to the subsequent.

Table 3.4: Comparison of the performance of proposed approach with other methods.

Algorithms	**Probabilistic Rand Index**	**F1-measure**	**Global Consistency Error**
MRF-CRP (ours)	0.79	0.71	0.23
MRF	0.72	0.66	0.19
DD-CRP	0.69	0.57	0.23
MRF-IBP	0.75	0.67	0.18

The experimental results obtained for the proposed approach are compared with other methods shown in Table 3.4. The GCE metric can be used to evaluate the consistency of a pair of segmentations. The measure is designed to be tolerant to refinement, that is, if subsets of regions in one segment consistently merge into some region in the other segmentation the consistency error should be low. To better understand how the GCE error metrics work, it is interesting to consider what the metrics report on two extreme cases: A completely under-segmented image, where every pixel has the same label (i.e. the segmentation contains only one region spanning the whole image), and a completely over-segmented image in which every pixel has a different label. From the definitions of the GCE ,we can see that both measures evaluate to 0 on both of these extreme situations regardless of what segmentation they are being compared to. The reason for this can be found in the tolerance of these measures to refinement. Any segmentation is a refinement of the completely under-segmented image, while the completely over-segmented image is a refinement of any other segmentation.

Time Complexity Analysis

The proposed approach has three modules, namely: SURF, MRF, and CRP. The overall time complexity is the contribution of these three modules. Hence the overall complexity is defined as $O(Complexity\ of\ (SURF)) + O(Complexity\ of\ (MRF))$ $+O(Complexity\ of\ (CRP))$. The time complexity of SURF is $O(mn + k)$ (Drews et al., 2011), where m, n, and k denote image width, height, and the number of key points, respectively. The estimated complexity of MRF is $O(L^V E)$ (Schwarz et al., 2012). Where L, V, and E denote the number of labels, number of vertices, and number of edges, respectively. The time complexity of CRP is $O(L)$, where L is the number of labels (clusters). Hence, the estimated time complexity is approximately $O(mn + k) + O(L^V E) + O(L)$.

3.3.3 Summary

In this work, the proposed approach has been segmented in three different levels. It uses the selective advantages of the MRF on low level and better performance of CRP on high-level. To increase the robustness of the proposed approach, BN is used. The three different segments are detection of the objects in the image, attribute prediction and association, and semantic segmentation. The first and the most important task is to detect the different objects. We have compared the SIFT feature detection algorithm with the SURF feature detection algorithm and found that performance of SURF better as shown in Table 3.3. The second objective is achieved by using the BN. The third objective of semantic segmentation and object-attribute association is achieved by making clusters using MRF stacked CRP. For the local level, MRF starts clustering by forming set of components. After low-level, these components are merged into larger clusters using the high level CRP. The model can be used for 3D images with edge detection technique for improving the efficiency will be perceived in future.

Chapter 4

Visual Video Data Analysis using Captioning

4.1 Semantic Context Driven Language Descriptions of Videos using Deep Neural Network

The tremendous addition of data to the Internet in the form of text, photos, and videos complicated CV jobs in the vast data domain. Recent advances in video data exploration and visual information captioning have proven a difficult task in CV. The ability to create visual captions is due to the integration of visual information with natural language descriptions. This work offers an encoder-decoder architecture that utilises a 2D-CNN with layered LSTM as the encoder and an LSTM combined with an attention mechanism as the decoder with a hybrid loss function. Spatial features are captured using visual feature vectors taken from video frames using a 2D-CNN. To collect temporal information, the visual feature vectors are input into the layered LSTM. The attention mechanism enables the decoder to detect and focus on relevant objects and to correlate the visual context and language content in order to generate semantically accurate captions. The decoder uses the visual features and Global Vectors for word representation (GloVe) word embeddings (Pennington et al., 2014) to build natural semantic descriptions for the videos. The core contributions of the proposed framework are as follows:

- Using GloVe word embeddings, specifically used 100-dimensional GloVe depending on the size of the vocabulary in the dataset.
- A layered LSTM encoder trials with visual feature extractor networks for extracting the temporal features to understand the activities in videos.
- A hybrid loss function was used to bridge the gap between semantic context of video and word prediction.
- Testing the efficiency of proposed framework with eight well known performance evaluation metrics.

4.1.1 Methodology

The proposed framework consists of encoder-decoder for generating an appropriate caption for a video as shown in Figure 4.1. The pre-processing stage focuses on preparing the image frames derived from the input video to match the dimension requirements of pre-trained CNN. The visual encoder combines CNN-based visual features and the stacked LSTM. The decoder part is defined as a combination of attention and a single LSTM layer. To select the significant features, the Soft Attention has been used. The

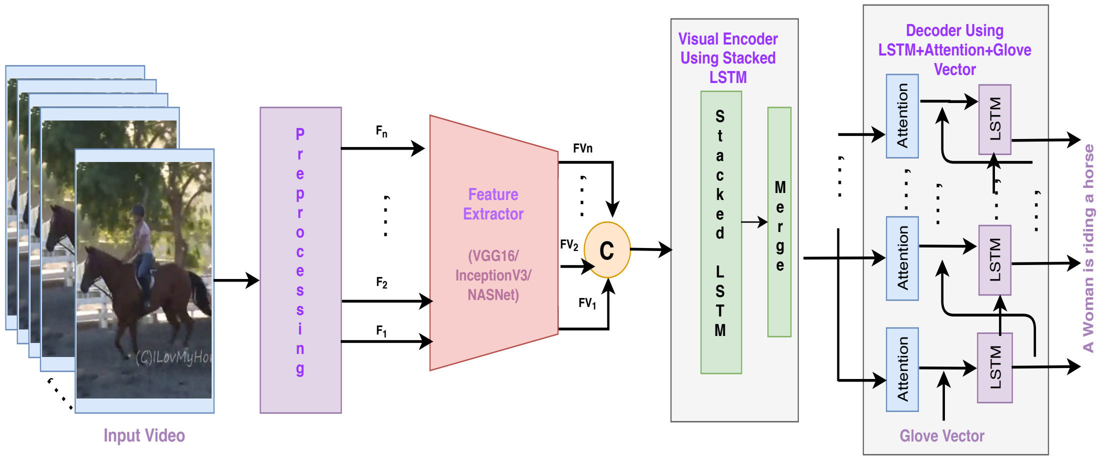


Figure 4.1: Proposed context driven video captioning framework.

advantage of Neural Architecture Search Network (NASNet) and stacked LSTM is that a varying number of convolutional cells and the number of filters in the convolutional cells yields better accuracy than the traditional methods. Another view is that nonlinearity, and careful selection of connections among neurons together add to better results. Although two stages search the feature space created by two types of cells, stacked LSTM predicts the best captions.

The deeper layers in stacked LSTM are understood to combine the learned representation from previous layers to create new representations at high levels of abstraction. This adding depth is a type of representation optimizations.

In the proposed framework, the visual language model called encoder-decoder was used with NASNet (Zoph et al., 2018), InceptionV3 (Szegedy et al., 2016), VGG16 (Simonyan and Zisserman, 2015). The decoder consists of the attention mechanism to address long sequences in machine translation—this action of selectively concentrating on a relevant word to be predicted while ignoring others in succession. Each sub-components of the proposed framework are described in detail in following subsections.

4.1.1.1 Preprocessing

In the preprocessing stage, extraction of frames are done. The extracted frames are resized to meet the input dimensions of deep learning models, namely VGG16, NASNet, and InceptionV3. Primarily, feature vectors are extracted, which are a high-level representation of videos, using three distinct models with varying dimensions.

NASNet Large

The NASNet is a convolutional network originally used for image captioning. It takes 331×331 image size as input and resulting feature vector dimension of 4032 per frame. The NASNet architecture is defined as the blocks or cells, and these blocks are defined as the feature map with normal and reduction dimension. The blocks are called as Normal Blocks and Reduction Blocks. The Normal Block usually measures the feature map from the respective layer, and the Reduction Cell/Block reduces the feature map by a factor of 2. The controller decoder finds these Normal and Reductions Blocks information.

InceptionV3

Google developed this deep learning model for image captioning. The input image size should be 299×299. It results in a vector of dimension 2048 per frame. This model consisting of an "inception cell" working in parallel and then ultimately give the concatenated results. The kernel size in this model uses 1×1 convolutions to reduce the input channel depth. Each cell consists of different kernels with 1×1, 3×3, 5×5 dimensions, which learn to extract features from the input. Max pooling and padding is used to retain the dimensions for concatenation.

VGG16

Oxford developed VGG16 deep neural network. It takes an input image of size 224 $\times$ 224 pixels. The output feature vector is of size 4096. This deep neural network's advantage is using a small receptive field with a kernel size 3×3 dimension. The smallest possible size kernel captures the abstract information within frames through traversal all along the image grid's directions. The potential smaller values as the kernel with 11 dimensions act as a linear transformation of the input. The process is followed by a ReLU unit.

Given a video as a sequence of frames V = $\{F_1, F_2, ..., F_n\}$, where the video V has n frames and F_i represents i^{th} frame of the video. The Feature Extractor generates set of feature vectors FV = $\{FV_1, FV_2, ..., FV_n\}$.

4.1.1.2 Visual Encoder

The visual encoder is a stacked/layered approach. Visual features are further processed using stacked LSTM to capture temporal information. LSTM units' output is merged and then send to the decoder.

Single LSTM Unit

The LSTM network introduced in (Hochreiter and Schmidhuber, 1997). Architecture of single LSTM unit based on (Sha et al., 2016) is given in Figure 4.2, and relation is defined in Equation (4.1).

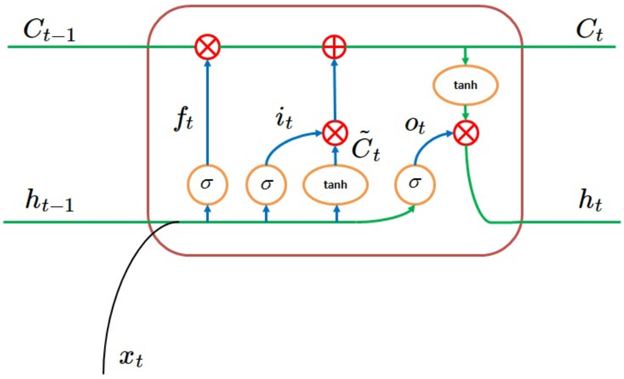


Figure 4.2: LSTM unit (Sha et al., 2016).

$$\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)\\
\widetilde{C}_t &= \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g)\\
C_t &= f_t * C_{t-1} + i_t * \widetilde{C}_t\\
h_t &= o_t * tanh(C_t)
\end{aligned} \tag{4.1}$$

Where, i_t, f_t, and o_t denotes input, forget, and output gates respectively. x_t, C_t, and h_t represent the current input, cell state, and hidden states, respectively. C_{t-1} and h_{t-1} are the input from preceding timestep. The symbol * represents the element wise multiplication. W_{xi}, W_{hi}, W_{xf}, W_{hf}, W_{xo}, W_{ho}, W_{xg}, W_{hg}, b_i, b_f, b_o, and b_g are the parameters.

Stacked LSTM With Dropout

The use of the stacked LSTM visual encoder is to encode the spatial CNN feature vectors and to exploit temporal information of the frames in the videos. To improve deep learning model performance and avoid overfitting, the dropout layer is induced on the feature vectors to randomly switch off few cells during the training. The implementation is also tried with multiple layers of LSTM and finally, the output of the layers are merged, and the result is given to the next layers. The output of layer i is defined as in Equation (4.2).

$$o_t^{(i)}, h_t^{(i)} = LSTM^{(i)}(x_t, h_{t-1}^{(i)}) \tag{4.2}$$

The output of each previous LSTM layers are concatenated to obtain the output vector $o_t^{(f)}$ of the encoder as shown in Equation (4.3).

$$o_t^{(f)} = \sum_{t=0}^{n} o_t^{(1)} + o_{n-t}^{(2)} \tag{4.3}$$

The proposed stacked LSTM unit is depicted in Figure 4.3. The network length is the measure of the time span of a training set. The h_t, c_t and x_t denotes the output of last moment, current cell state and, current input respectively. The experimental result showed that 2-layered LSTM with combinations of NASNet, attention, and embedding is better than 3-layered LSTM. Though the 3-layered LSTM seems better in abstract representation, overall better performance has resulted in 2-layered stacked LSTM because of different combinations.

4.1.1.3 Decoder

The decoder takes the feature vector from the encoder and utilized give best match to the original input using attention and GloVe vectors.

Attention Mechanism

The output context vector from the encoder is fed to the decoder and generates a sequence of words describing the video. Training the model and giving the input sequence with a very long text sequence is not good. This sort of single, less contextual information from the encoder does not give the decoder excellent semantic and specific

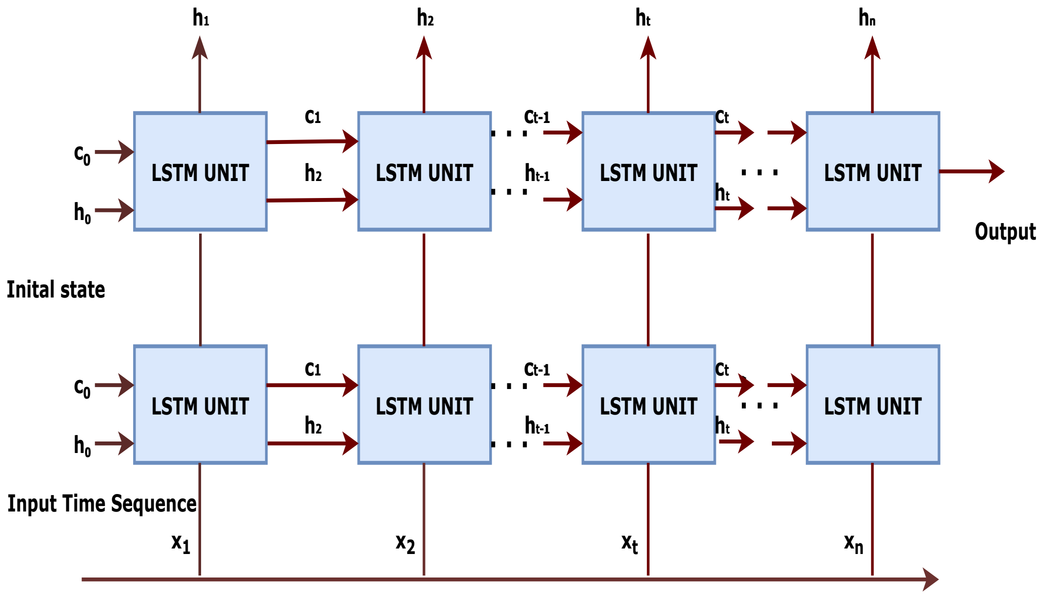


Figure 4.3: Proposed framework of stacked/layered LSTM unit.

information. The attention approach gives more contextual meaning to the used decoder. The decoder learned how much semantic “attention” it should give to each input word at every decoding step.

The encoder output ($o_t^{(f)}$) fed to the decoder which is more contextually meaningful. The decoder's last hidden state and encoder hidden states are combined to calculate attention weights. A feed-forward neural network learns these weights.

The value for context vector c_i for the output word y_i is determined using Equation (4.4).

$$c_i = \sum_{j=1}^{n} \alpha_{ij} o_j^{(f)} \tag{4.4}$$

The value for weights α_{ij} is computed by using a standard softmax function given by the Equation (4.5).

$$\alpha_{ij} = exp(e_{ij}) / \sum_{k=1}^{n} exp(e_{ik}) \tag{4.5}$$

e_{ij} is the calculated output score for the input at j and output at i using Equation (4.6).

$$e_{ij} = a(s_{i-1}, o_j^{(f)}) \tag{4.6}$$

Attention Based LSTM

The decoder is an attention-based LSTM network. Attention mechanism combined with LSTM to focus on input sequence when predicting specific output sequence with more contextual understanding. Hence, the proposed decoder attention helps in selecting salient features for producing output sequence using a layer of LSTM.

In the proposed framework, every word in the caption encoded using GloVe. The vector representation model GloVe is used as an unsupervised learning technique to enable word representations of the given input word sequence. This model's training stage gives a cumulative global word-word co-occurrence representation from an input word corpus. The resulting vector depicts more informative and exciting linear structures of the word vector space. These embeddings are passed to the last layer to generate the sequence.

The previous hidden states h_{t-1}, the previous predicted word w_{t-1}, and the present context vector are combined to form the LSTM's (Venugopalan et al., 2014) hidden state. During each time step context vector is adjusted so that decoder selectively attends the input sequence. Hence, the output of the decoder is given by Equation (4.7).

$$o_t, h_t = LSTM([w_{t-1} + Attention[h_{t-1}; o_{t-1}^{(f)}]], h_{t-1}) \tag{4.7}$$

Loss Functions

The approach used here is an attention-based LSTM. The main idea of using two loss functions is to ensure the contextual relationship between words generated and the semantic relations between the video features and the descriptions to be developed for the video's respective scene. The process maintains a simultaneous check between video translation and semantic efficiency.

Loss 1 : Translation From Videos To Words

Cross entropy loss is used for calculating the cost of translation and is given in Equation (4.8).

$$Loss1 = -\frac{1}{N}\sum_{n}^{N} y \ln a + (1-y)\ln(1-a) \tag{4.8}$$

Where, N = represents number of training examples, y indicates actual values, and a denotes predicted values.

Loss 2 : To Bridge The Semantic Gap

Mean squared error loss helps bridge the semantic gap by estimating how far off the average predicted value is from the ground truth value. Thus, minimal value sees the close relationship between an estimated and actual value and ensures higher semantic similarity. The relation is defined in Equation (4.9).

$$Loss2 = -\frac{1}{N}\sum_{n}^{N}\sum_{k}^{c}(y_k^n - a_k^n)^2 \tag{4.9}$$

Where, N represents number of training examples, c indicates dimension of output vector, y denotes actual values, and a represents predicted values.

Combined Loss

The combined loss measure is given in Equation (4.10).

$$NewLoss = \lambda Loss1 + (1 - \lambda)Loss2 \tag{4.10}$$

Where, λ is a hyperparameter between 0 and 1.

4.1.2 Experiments, Results and Discussion

4.1.2.1 Evaluations Metrics

The video and image captioning method is a result of collaboration between deep neural network and advancement in NLP techniques. The NLP-related benchmark measures that are often used to evaluate automatically created captions and reference captions explained in the subsequent subsections. All of these indicators were applied to the aforementioned standard datasets.

The performance of the proposed framework was evaluated on different metrics. The BLUE (Papineni et al., 2002) algorithm evaluates the quality of the text, considered to be the matching between machines output and that of reference. The score is always between 0 and 1. This score indicates how similar the machines predicted output to that of reference with values closer to one representing more similar texts.

The Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Banerjee and Lavie, 2005) is evaluating the machine translation output based on the harmonic mean of unigram precision and recall. The Semantic Propositional Image Caption Evaluation (SPICE) (Anderson et al., 2016), is to alleviate the limitations of existing n-gram based metrics. This method uses the semantic propositional context component of caption evaluation. The Consensus-based Image Description Evaluation (CIDEr) (Vedantam et al., 2014) metric measures the similarity of generated text against human-generated sentence. This measure uses grammaticality, saliency, and accuracy inherently captured. The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004) is similar to BLUE compares predicted text with reference sentence.

BLEU

The Bilingual Evaluation Understudy Score (BLEU), is a score that is calculated by matching a predicted machine translation against human generated reference. Although BLEU was intended for machine translation, it may be used to analyse text output for a variety of NLP jobs. An ideal match will have score of 1.0, but a perfect mismatch is valued 0.0. Basicaly comparison based on lexical count of words in the candidate translation (predicted words) with reference text, where if n=1 then it corresponds to a token and a n=2 corresponds to a word pair. Order of the words is not considered while comparison. The BLEU score is skeptical of syntactical accuracy. It is just concerned with the total count of terms that match the reference or actual caption. Mathematically, the above metric is defined in Equation (4.11).

$$\text{BLEU} = \underbrace{\min\left(1, \exp\left(1 - \frac{\text{reference-length}}{\text{output-length}}\right)\right)}_{\text{brevity penalty}} \underbrace{\left(\prod_{i=1}^{4} \text{Precision}_{\ i}\right)^{1/4}}_{\text{n-gram overlap}} \tag{4.11}$$

Where, $Precision$ is defined in Equation (4.12): The value, C_W represents the total number of correct words in generated sentences. The value, T_W signifies the total number of words in the generated sentence.

$$Precision = \frac{C_W}{T_W} \tag{4.12}$$

METEOR

METEOR was presented as a solution to BLEU's difficulties. METEOR substituted semantic matching for the exact lexical matching required by BLEU. METEOR makes use of the English lexical database WordNet to account for a range of match types, including exact word, stemmed word, synonymy, as well as paraphrase matching. METEOR scores for predicted and reference sentences are determined by calculating using unigram Precision P and unigram Recall R, as described in Equations (4.13) and (4.14), respectively.

$$P = \frac{UG_{PR}}{UG_P} \tag{4.13}$$

where UG_{PR}, the count of unigrams in the predicted and reference sentences and UG_P is the count of unigrams in the predicted sentence. Further, the Recall is defined in the Equation (4.14).

$$R = \frac{UG_{PR}}{UG_R} \tag{4.14}$$

where, UG_R denotes the total number of unigrams in the reference. The mean harmonic index (F) is derived using precision and recall are defined in Equation (4.15).

$$F_{\text{measure}} = \frac{10PR}{R+9P} \tag{4.15}$$

For longer matches, containing non-adjacent mappings between the predicted and reference sentences, a penalty is designed and implemented. Penalty P_n relation is defined as depicted in Equation (4.16):

$$P_n = 0.5 * \frac{C}{UM} \tag{4.16}$$

Where, C denotes the count of chunks as well as UM stands for the number of matched unigrams. Consequently, the METEOR index for a certain alignment can be calculated using the following Equation (4.17):

$$\text{METEOR} = F_{\text{mean}}\ (1 - P_n) \tag{4.17}$$

The higher the METEOR score, the more closely it is associated with human judgement.

ROUGE

The ROUGE metric was developed to evaluate text summaries. It uses n–grams to determine the recall score of the generated phrases that correspond to the reference sentences.

Recall and Precision is calculated for the longest common subsequence in Equations (4.18) and (4.19) respectively.

$$R_{LCS} = \frac{LCS\left(R_s, P_s\right)}{L_r} \tag{4.18}$$

$$P_{LCS} = \frac{LCS\left(R_s, P_s\right)}{L_p} \tag{4.19}$$

The F-measure score in Equation (4.15) can be calculated using LCS for predicted summary P_s of length L_p and reference summary R_s of length L_r as depicted in Equation (4.20).

$$\text{ROUGE}_{LCS(R_s,P_s)} = F_{LCS} = \frac{\left(1+\beta^2\right) R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}} \tag{4.20}$$

Here, the value $LCS\left(R_s, P_s\right)$ denotes the longest common subsequence of R_s and P_s. β is the ratio of LCS-Precision to LCS-Recall.

CIDEr

CIDEr is an analysis method for human consensus-based image description. This metric's primary purpose is to compare a predicted caption to a single or set of human-annotated reference captions for the image. It stems and converts all candidate and reference sentence words to their root forms. Each phrase is regarded by CIDEr as a collection of n–grams containing between one and four words. It calculates the number of n-grams that exist in both the predicted and reference phrases to encode the consensus between them. N-grams, which are extremely prevalent in reference phrases, have been given a reduced weight using the Term Frequency Inverse Document Frequency technique (TFIDF). The $CIDEr_n$ score is computed as in Equation (4.21):

$$\mathrm{CIDEr}_n\left(c_i, S_i\right) = \frac{1}{m} \sum_j \frac{g^n\left(c_i\right) \cdot g^n\left(s_{ij}\right)}{\left\|g^n\left(c_i\right)\right\| \cdot \left\|g^n\left(s_{ij}\right)\right\|} \tag{4.21}$$

In this, $g^n\left(c_i\right)$ is a vector representing all n–grams with length n and $\left\|g^n\left(c_i\right)\right\|$ represents magnitude. Same is true for $g^n\left(s_{ij}\right)$.

4.1.2.2 Dataset

Microsoft Video Description (MSVD) (Chen and Dolan, 2011) dataset, which is a benchmark dataset for video captioning. The dataset consists of a total of 1,970 short video clips from YouTube and 41 descriptions in English for each video. It also contains 80,000 clip-description pairs in different languages. For the proposed framework, English captions and dataset split in (Venugopalan et al., 2014) used.

4.1.2.3 Experiments

Training Parameters

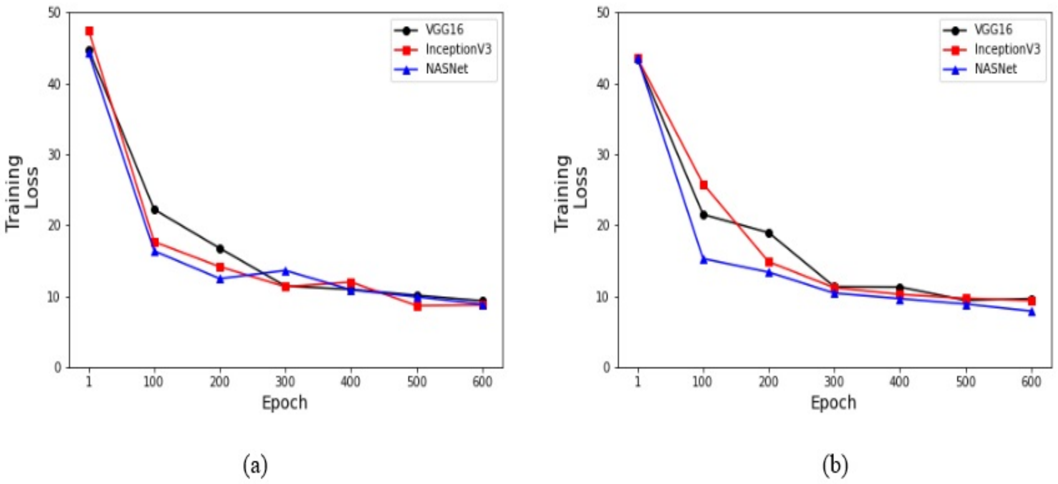


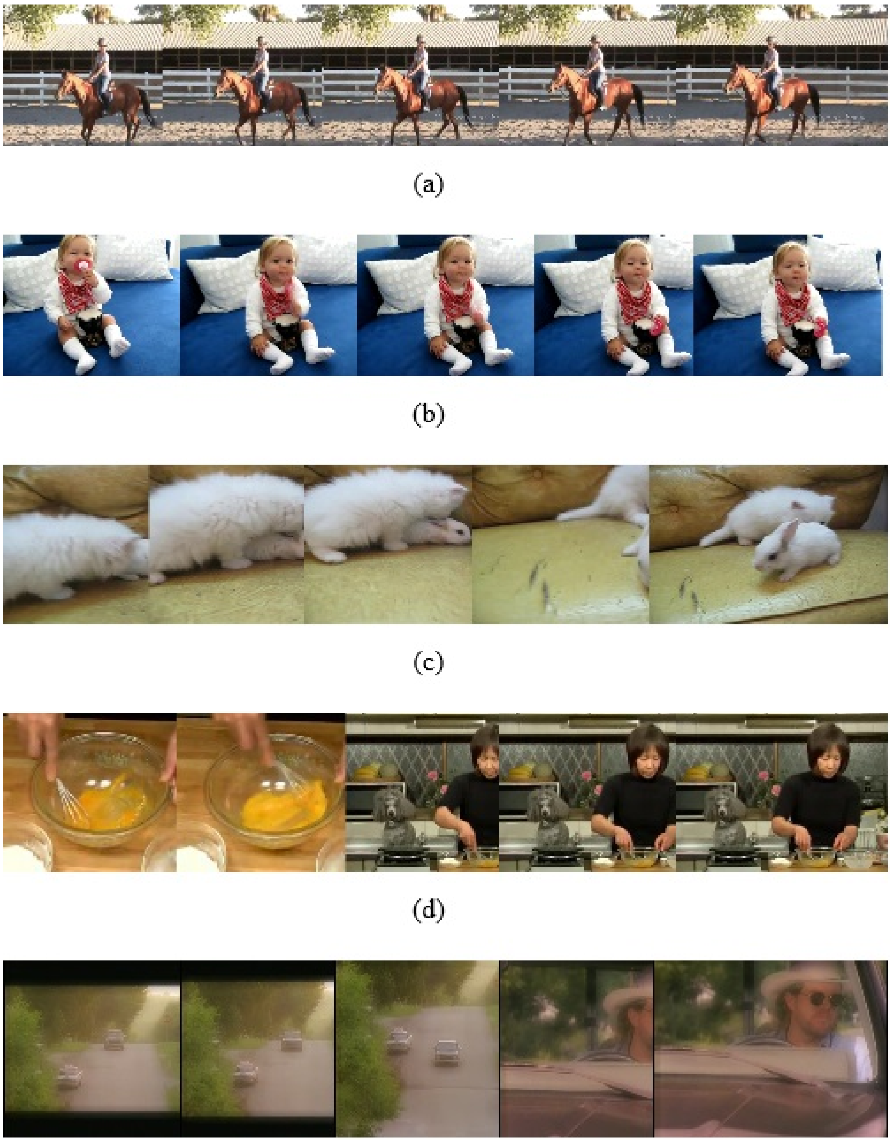
Figure 4.4: Plot of training loss versus number of epochs.: (a) 2-layered LSTM (b) 3-layered LSTM

Using the dataset split up mentioned earlier, proposed framework was trained for 600 epochs. The proposed framework performed well with the following training parameters: batch size=128, learning rate=0.001, and optimizer=Adam. Figure 4.4 is a plot of training loss against the number of epochs trained for 2-layered and 3-layered, LSTM respectively. One can observe that the loss decreases drastically, up to 100 epochs, and then decrease gradual. The proposed framework got stabilized between

400 and 600 epochs.

Sample Results With Built Models

In this work, three models proposed based on the different pre-trained models for feature extraction in the visual encoder part. Proposed Model_1 utilizes VGG16 based visual Feature Extractor, Model_2 uses InceptionV3 to extract features, and Model_3 uses NASNet as Feature Extractor. All these models used GloVe embedding and attention in the decoding part.



(a)

(b)

(c)

(d)

(e)

Figure 4.5: Test samples: (a) Sample-1 (b) Sample-2 (c) Sample-3 (d) Sample-4 (e) Sample-5.

Figure 4.5 shows some frames of five different test samples. The corresponding ground-truth and generated captions by three models are given in Table 4.1. The best

Table 4.1: Sample input and output of proposed framework.

Figure No.	**Ground-truth**	**Model_1**	**Model_2**	**Model_3**
Figure 4.5 (a)	{A girl is riding a horse and jumping barriers}, { A girl is riding a horse}, {A woman is riding a horse in an outdoor arena and she makes a jump}, { A woman is riding a horse}, {A person riding a horse is jumping hurdles}.	{a man is riding a horse}.	{a man is riding a horse }.	{a girl is riding a horse}.
Figure 4.5 (b)	{A baby is playing}, {A baby is playing with a pacifier}, {A baby plays on a bed}, {a baby is sucking on a soother and watching the camera}, {The toddler put the pacifier in and out of his mouth}.	{a man is slicing a potato}.	{a baby is playing with toys}.	{3a baby is playing with a camera}.
Figure 4.5 (c)	{A cat is playing with a bunny}. ; {A cat is playing with a rabbit}, {A kitten is playing with a rabbit},{a white cat playing with a white bunny}, {A kitten and a rabbit are playing}.	{ a kitten is trying to climb a tree}.	{a cat is playing}.	{ a white kitten is playing}.
Figure 4.5 (d)	{A woman is cutting something},{A woman cuts up some worms},{someone show how to prepare the japanese food},{a person coking},{ a women is making dish}.	{A woman is preparing a dish}	{The woman is mixing ingredients in a bowl}	{A woman is mixing some eggs}.
Figure 4.5 (e)	{A person is driving a car},{ the man is drive the car on the road and seeing the place}, {Someone is driving a car}, {A car is driving down a road}, {A car is moving}.	{a man is running in the water}	{a man is driving a car}.	{a man is driving a car }.

result with Proposed model_3 is due to the approach used here is an attention-based LSTM along with NASNet feature extractor. Further, the idea of using two loss functions is to ensure the contextual relationship between words generated and the semantic relations between the video features and the descriptions to be developed for the video's respective scene. The process maintains a simultaneous check between video translation and semantic efficiency.

Deeper networks, such as VGG16, InceptionV3, exhibit a slower decline in efficacy. This could be because those networks have a more complex structure, which gives them more room to learn attributes of the images that are unaffected by noise. The blurring, noise, or fogy produces a tiny shift in the filter responses in the primary convolutional layer. However, the penultimate convolutional layer exhibits significant variations in the filter responses. This modifies the first layer reaction, resulting in more or less significant alterations at the higher layer. The deviations in the results of this model would have been avoided by making some layers trainable and non-trainable. The other options that use SGD optimizer to make the model converge play with the hyperparameters, like setting the learning rate very low. The reason is in VGG16, the parameter space is huge, and to deal with this issue, it doesn't have any sophisticated techniques like BatchNorm used in later models.

Obtained experimental results shows that Model_3 performed well in identifying the objects in the images and the semantic consistency than the other two models. For test samples in Figures 4.5 (a) and (c), Model_3 gave captions close to the ground-truth captions than other two models. In test sample Figure 4.5 (b), Model_1 identified a non-existing object. For test samples in Figures 4.5 (d) and (e) all the models gave almost similar captions to the ground-truth captions.

Comparison of Models Performance

Table 4.2: Evaluation of 2-layered and 3-layered LSTM in proposed framework using BLEU metrics.

Models	**MSVD**							
	2 Layer Stacked LSTM				**3 Layer Stacked LSTM**			
	BLEU1	**BLEU2**	**BLEU3**	**BLEU4**	**BLEU1**	**BLEU2**	**BLEU3**	**BLEU4**
VGG16 + Stacked LSTM + GloVe (Model_1)	69.1	50.1	38.2	27.0	68.1	48.8	37.0	25.58
InceptionV3 + Stacked LSTM + GloVe (Model_2)	74.3	60.1	49.7	40.2	73.6	59.8	49.5	38.5
NASNet + Stacked LSTM + GloVe (Model_3)	**78.4**	64.8	54.2	43.7	78.2	**65.3**	**55.1**	**44**

Table 4.2 shows the BLEU (Papineni et al., 2002) performance metrics evaluated for the proposed framework with 2-layered and 3-layered LSTM on the MSVD dataset. The model with NASNet extracted features, GloVe, and 2-layered LSTM almost performed equally compared to the 3-layered NASNet model. But, this NASNet model with 2-layered and 3-layered almost outperformed other proposed frameworks with VGG16 and InceptionV3 as NASNet identifies videos' objects more accurately with the help of more abstract representations from layered LSTMs.

In stacked LSTM, a level of abstractions of temporal input observations is also added. The GloVe represents words in n-dimensional space with unique meaning in each dimension. It captures a correlation between other words, which helps the stacked LSTM map from videos to descriptions correctly. The overall observations with this level of experiment and values conclude that Model_3 with a 2-layered approach is optimal and less costly, considering the BLEU metrics of all three models.

The suggested framework outperforms other current methodologies in terms of overall model performance. NASNet's LSTM and GloVe embedding are unusual in their two-layered structure. There are fewer floating-point operations and parameters in NASNets than in competing designs. To create a cell with the optimum performance, NASNet uses a controller RNN to identify the best combination of operations from a set of operations, rather than creating the block by hand. The input values to the network are fed through many levels of LSTM and propagate over time within a single LSTM cell with two layers of LSTM. Consequently, the parameters are well spread throughout several layers of the system. As a result, each time step has a complete set of inputs. While Word2Vec relies solely on local statistics (such as the context in which words are used), GloVe takes into account global data (such as the co-occurrence of terms) in order to produce word vectors.

To further validate the model's performance, we added an additional LSTM layer than the suggested framework, demonstrating the critical role of LSTM and its properties in producing superior outcomes to the two-layered strategy. As a result, proposed Model_3 outperformed the other two methods in the combinations indicated.

Table 4.3 provides performance achieved by the proposed three models with 2-layered and 3-layered stacked LSTM with four standard metrics mentioned and are

Table 4.3: Evaluation of 2-layered and 3-layered LSTM in proposed framework using METEOR, ROUGE, CIDEr and SPICE.

Models	**MSVD**							
	2 Layer Stacked LSTM				**3 Layer Stacked LSTM**			
	METEOR	**ROUGE**	**CIDEr**	**SPICE**	**METEOR**	**ROUGE**	**CIDEr**	**SPICE**
VGG16 + Stacked LSTM + GloVe (Model_1)	24.7	60.7	32.4	3	24.1	60.9	29.6	3
InceptionV3 + Stacked LSTM + GloVe (Model_2)	**33.3**	66.6	58.4	4.8	31.1	67.0	64.4	4.9
NASNet + Stacked LSTM + GloVe (Model_3)	32.3	**68.8**	70.7	**5.1**	31.8	**67.5**	**71.4**	4.9

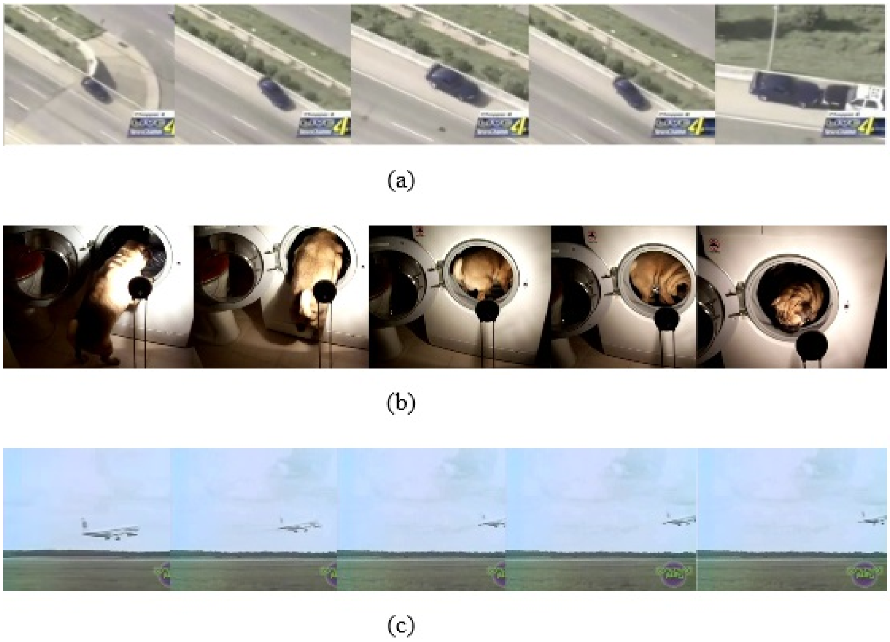
compared. We observe 2-layered Model_3 result compared to the 3-layered counterpart outperforms the latter with efficient utilization of NASNet cells and connections. Though the results seem to be significantly closer considering all three models with different LSTM levels and different performance metrics, 2-layered proposed framework slightly have an edge on their 3-layered counterparts. In general, Model_3 consider being more efficient in utilizing inherent features of that models to give the best result.

While the additional strength garnered by the more deeper architecture in LSTMs is not fully understood theoretically, it has been observed empirically that deep RNNs may perform better than shallower ones on certain tasks and datasets. Generally, two layers of LSTMs have been demonstrated to be sufficient for detecting more complicated features (Bin et al., 2019; Salman et al., 2018). Additional layers make training more difficult due to increased layering results in information saturation, and increased complexity and also may lead to poor performance. As a result of our trials, it is clear that two-layered LSTM performed admirably across all measurement parameters.

Inception has inception layers and fewer parameters than VGG16, which is merely a simple array of convolutional max-pooling layers with dropouts added at the outset for speed optimization. Also, these dropouts effectively handle the model's overfitting issues by dynamically flipping connections with the activation layer. For regularisation, there is additionally an auxiliary classifier. A complicated collection of filters within a 'cell' can considerably improve outcomes in InceptionV3. The NASNet model outlines creating such a cell as an optimization process and then stacks numerous copies of the best cell to create a large network. NASNet has designed a new optimized architecture that employs a controller RNN module to choose the top-performing cells. As we see the unique combinations, all of these structural modules performed on the MSVD dataset more effectively.

When other indicators such as METEOR, ROUGE, CIDEr, and SPICE are evaluated, the proposed Model_3 with two-layered LSTM outperforms the ROUGE and SPICE scores. This implies that the SPICE score always takes the textual dataset's semantics into account, as well as the association of an additional attention layer in our model. Whereas ROUGE is similar to BLUE and has a higher score, it makes logical that Model_3 constantly outperforms all other measures, as we demonstrated in other measurements. The Model_3 with three LSTM layers outperformed the CIDEr score because the more abstract level of information learned by the third layer automatically captures better grammaticality, saliency, and accuracy.

Failure cases



(a)

(b)

(c)

Figure 4.6: Failure cases: (a) Sample-1 (b) Sample-2 (c) Sample-3

Table 4.4 shows an inappropriate predicted output, which is not very close to the ground truth for all three models, which has considered the 2-layer LSTM stack. Though the failure cases ascertain, Model_3 is slightly better in giving results than the other two models. In this, Figure 4.6 (a) depicts a sample input image featuring ground truth captions. The output is slightly near the ground truth with the combination of attention and the stacked LSTM in the proposed Model_3, due to the more in-depth learning of parameters without convergence and the increased focus on the required captions appropriate for the image locations. The model with NASNet has fewer parameters than

Table 4.4: Failure cases: sample input and output given by the framework

Figure No.	**Ground-truth**	**Model_1 predicted caption**	**Model_2 predicted caption**	**Model_3 predicted caption**
Figure 4.6 (a)	{A car running from the police},{A guy is riding too fast in his bike.},{A man is driving backward and spins the car around.}	{a man is playing a guitar}	{a car is going up }.	{a car is chasing a car}.
Figure 4.6 (b)	{A dog climbed into a clothes washing machine.}, {A bull dog is jumping into a washing machine.}, {The puppy went into the dryer.}, {The dog crawled into the dryer.}	{a man is putting some vegetables in a pan}	{a man is beating a concrete into a water}	{a man is making a fancy dish}.
Figure 4.6 (c)	{ Airoplane in the Air}, {The plane took off from the runway.},{An airplane is taking off.},{the person going on the airplane}	{a man is riding a bike}.	{a woman is pushing a rock}.	{a woman is running in the air }

the other conventional networks, but it makes the best use of the features to accurately predict over half of the ground truth words, outperforming some of the existing approaches. Model_1 fared poorly, as there were no matches because VGG16 entirely misread the words due to insufficient learning. Model_2 predicted the terms as accurately as Model_3 but with a better score than Model_1. This is due to the inception modules, composed of smaller filters, technically known as pointwise convolutions, accompanied by convolutional layers with various filter sizes applied concurrently. This enables Inception networks to learn more complicated features and predict words with a high degree of accuracy compared to the ground truth. In Figures 4.6 (b) and 4.6 (c), all models failed to forecast accurately due to the complicated nature of the frames, which prevented them from learning all the features precisely due to an abundance of complex textures.

Single Loss vs Hybrid Loss

Figure 4.7 depicts the variation in BLUE4 and CIDEr metrics' performance while using single and hybrid loss during training. It is observed that the proposed NASNet Feature Extractor model performed well in the hybrid loss since single loss focus on only translation loss whereas hybrid loss, considers the semantic gap between video and captions. Hence, hybrid loss proved to work better for the proposed framework.

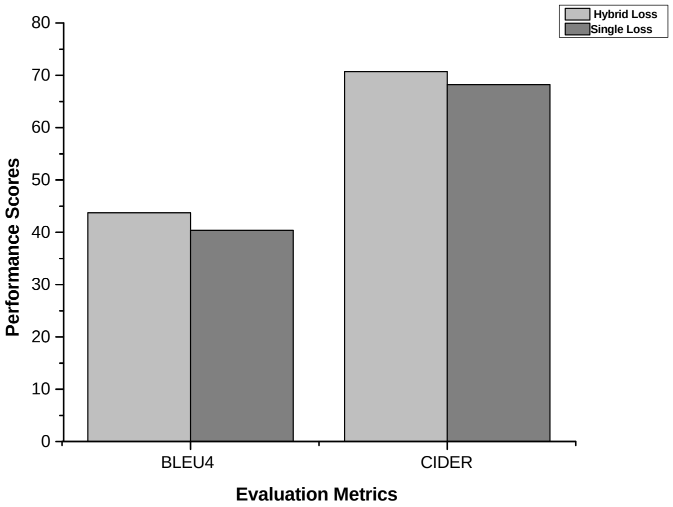


Figure 4.7: Single loss versus hybrid loss.

Table 4.5: Evaluation of Model_3 using different λ hyper parameter.

Lambda Values	Model_3							
	BLUE Score				Other Metrics			
	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE	CIDEr	SPICE
$\lambda = 0.1$	78.4	64.8	54.2	43.7	32.3	68.8	70.7	5.1
$\lambda = 0.3$	76.1	62.1	50.9	39.7	31.3	66.9	67.5	4.9
$\lambda = 0.7$	74.9	60.7	49.7	38.9	30.9	66.6	63.3	4.8
$\lambda = 0.8$	75.2	61.4	50.7	40.3	31.2	67.1	66.9	5.0
$\lambda = 0.9$	75.2	61.3	50.4	39.9	30.8	66.6	64.1	4.9

Table 4.5 shows the different performance scores experimented for hyperparameter λ with different values. We observe different performance values are corresponding to the different λ values. The best performance score has resulted with λ = 0.1 for the proposed 2-layered Model_3.

The values of λ, one of the tweaking factors relating with hybrid loss. Overfitting occurs in any model primarily as a result of the model learning even the slightest details contained in the data. Thus, after learning all conceivable patterns, the model performs admirably on the training set however fails to deliver satisfactory results during the testing phase. It crumbles when confronted with previously unknown data. To avoid overfitting, the model's complexity should be reduced. This applies a regularisation parameter λ. As a result, comparatively simple models are less prone to overfitting than complicated models. In this context, a simple model is one in which the dispersion of hyperparameters has a low entropy, and hence many possibilities are attempted. We discovered that the optimal value is 0.1.

Comparison with Existing Works

Table 4.6: Comparison of proposed layered LSTM method with existing video captioning methods.

Method	BLEU1	BLEU2	BLEU3	BLEU4
S-VC (Li et al., 2015)	-	-	-	35.1
SA (Yao et al., 2015)	-	-	-	40.3
MM-VDN (Xu et al., 2015)	-	-	-	37.6
LSTM-E (Pan et al., 2016)	74.9	60.9	50.6	40.2
HBNEVC (Baraldi et al., 2017)	-	-	-	42.5
LVMVP (Nian et al., 2017)	-	-	-	40.1
LSTM-GAN (Yang et al., 2018)	-	-	-	42.9
SE-GRU (Hao et al., 2020)	-	-	-	42.9
BPLSTM (Nabati and Behrad, 2020b)	78.4	64.8	53.8	42.9
UTS (Sah et al., 2020)	-	-	-	43.00
STAT_LOC_V (Yan et al., 2020)	-	-	-	43.2
STAT_LOC_L (Yan et al., 2020)	-	-	-	42.9
p-RNN(VGGNet) (Yu et al., 2016)	77.3	64.5	**54.6**	**44.3**
Model_3 (Proposed)	**78.4**	**64.8**	**54.2**	**43.7**

Table 4.6 compares the obtained experimental results of the proposed NASNet Feature Extractor, Model_3 with some of the existing state-of-the-art video captioning works on the MSVD dataset. The proposed NASNet model gave better results for BLEU1, BLEU2, BLEU3, and, BLEU4 metrics. The reason behind this is because BLEU score calculation searches for the same words in the text. The combination of NASNet with layered approach betters the representation and prediction.

Table 4.7: Comparison of proposed framework with the state-of-the-art methods w.r.t METEOR and CIDEr score.

Method	**METEOR**	**CIDEr**
S-VC (Li et al., 2015)	29.3	-
SA (Yao et al., 2015)	29.6	51.7
S2VT (Venugopalan et al., 2015)	29.2	-
S2VT[VGGNet+Optical flow] (Venugopalan et al., 2015)	29.8	-
MM-VDN (Xu et al., 2015)	29.0	-
MP-LSTM (Venugopalan et al., 2014)	29.1	-
LSTM-E[VGGNet] (Pan et al., 2016)	29.5	-
LSTM-E[C3D] (Pan et al., 2016)	29.9	-
LSTM-E[VGGNet+C3D] (Pan et al., 2016)	31.0	-
LSTM-GAN (Yang et al., 2018)	30.4	-
p-RNN[C3D] (Yu et al., 2016)	30.3	-
p-RNN[VGGNet] (Yu et al., 2016)	31.1	-
LVMVP (Nian et al., 2017)	29.9	51.1
BPLSTM (Nabati and Behrad, 2020b)	32.0	62.20
HRNE (Pan et al., 2016)	32.1	-
HBNEVC (Baraldi et al., 2017)	-	63.5
SE-GRU (Hao et al., 2020)	-	62.3
STAT (Tu et al., 2017)	-	67.5
MA-LSTM (Xu et al., 2017)	-	70.4
UTS (Sah et al., 2020)	33.20	71.10
STAT_LOC_V (Yan et al., 2020)	30.5	62.8
STAT_LOC_L(Yan et al., 2020)	31.0	62.5
Model_3 (Proposed)	**32.3**	**70.7**

Though the p-RNN (Yu et al., 2016) outperforms our Model_3 in BLEU3 and BLEU4 attributed to the fact that their RNN model is not compelled and video features are generally fed into the multilayer, our model outperforms all of the other approaches listed in Table 4.6 due to the inclusion of soft attention in the decoder and a contextual vector generated for the captions. The results in (Nabati and Behrad, 2020b) are identical in BLEU1 and BLEU2 because the measure is a just lexical matching of words between reference[input sentence] and candidate sentence[predicted sentence]. When it comes to BLEU3 and BLEU4 the proposed model is bettered due to the contextual understanding of preceding and succeeding words of the any target word. Moreover the proposed model is attached to an attention mechanism, which finds the lexical and semantics of the words surrounded whereas the method in (Nabati and Behrad, 2020b) is not with attention mechanism.

Table 4.7 shows the obtained experimental results of the proposed framework using NASNet with some of the existing static frame-level approaches on video captioning works on the MSVD dataset. The proposed NASNet framework gave better results for METEOR metrics because it first compares tokens, synonyms, and paraphrases. Some of the existing baseline papers having multiple different features on the same video dataset. We observe that 2-layered Model_3 shows better performance. Table 4.7 also compares the CIDEr metrics and the proposed framework gave better results over the existing works because CIDEr uses lengthier n-grams to capture the grammatical properties and higher semantics of the text.

Experimentation Environment Details

All studies are done on a machine configured with an Intel Core i7-10750H CPU running at 2.60GHz, 2592Mhz, six cores, twelve logical processors, sixteen gigabytes of RAM, and an NVIDIA GeForce GTX 1650 GPU. Keras with TensorFlow is used as the backend.

Advantages

Real-world applications like automatic video subtitling, surveillance footage, text-based video retrieval affordability for blind users, video comprehension, multimedia recommendation is made possible by video and image captioning advances. These include helping people with various degrees of vision disability, self-driving vehicles, sign interpretation, human-robot interaction, and intelligent video subtitling. Various 2D-CNN models are experimented with layered LSTM to obtain the suitable model to extract spatio-temporal features from the video in the proposed work. Also, the attention mechanism captures the contextual information to predict the best phrases for the videos. The experimental results have also proven the same and made the proposed approach practically applicable in various real-world scenarios mentioned above.

Limitations

While our approach is capable of producing a sentence for video and has demonstrated promising outcomes, it has significant drawbacks. The majority of our failures result in an inaccurate object name being used in phrases, for example, when small objects with similar shapes or appearances are confused. As a result, reliably finding correct objects in images, including those that are hazy or obscured, and anticipating associated captions would remain an open topic. Video and Sentential data goes unidirectionally down to the next level via the visual encoder. As a result, utilising the Bidirectional LSTM, erroneous information can still be eliminated. While we built a sentence vector with GloVe, the model can still incorporate the most recent embeddings such as BERT (Devlin et al., 2018).

Time Complexity Analysis

The proposed approach consists of a CNN feature extractor followed by a layered LSTM model. So, the total complexity of the system comprises the complexity of CNN model used and the complexity of training the layered LSTM model. This is approximately equal to $O(CNNmodel) + O(LSTMmodel)$. Thus the overall complexity can be defined as $O(N) + e * b * T_b * O(L)$. Where N, e, b, L, and T_b, denote the number of parameters in CNN feature extractor, epochs, batches, parameters in layered LSTM model, and time estimation of backward and forward passes, respectively.

4.1.3 Summary

The proposed framework fully explores the spatial and temporal information among the video frames' whole sequence. In this work, an efficient and new framework is proposed by integrating multiple LSTM, different Feature Extractors, Soft Attention, hybrid loss functions and GloVe embedding mechanism at the decoding stage. The visual encoder is a combination of CNN-based visual features and the layered LSTM. The decoder part is defined as a combination of attention and a single LSTM layer. To select the significant features, the Soft Attention has been used. This work induced the hybrid loss to focus on semantic consistency.

Based on the experiments, the framework achieved approximately 24.5 % more than S-VC, 9% more than LSTM-E, and 2% more than BPLSTM in BLUE score criteria. Further, the proposed model outperformed SA by 9% and 36% in terms of METEOR and CIDEr, respectively. Thus, the suggested model outperformed the majority of current studies in terms of a variety of evaluation metrics.

In the future, we improve proposed model to work with domain-specific datasets, such as movies and documentaries, and to extend the architecture to incorporate GAN. Additionally, we would like to experiment with techniques such as beam search, which is used to determine the optimal word combination for a caption.

4.2 A Novel Multi-Layer Attention Framework for Visual Description Prediction Using Bidirectional LSTM

The massive inflow of data to the internet in the form of text, images, and videos has recently exacerbated the difficulties of computer vision-based tasks in the world of big data. Integrating visual content with natural language to create visuals or video explanations has proven to be a difficult task for many years. However, recent experiments in image/video captioning that make use of LSTM have piqued researchers' interest in its possible usage in video captioning. The proposed work describes the development of a unique video captioning framework that combines a multilayer BiLSTM encoder and a unidirectional decoder with a temporal attention framework in order to create superior global representations for videos. The following are the most significant contributions made in this research work.

- The proposed framework makes use of a novel multi-layer BiLSTM encoder and a multi-layer unidirectional decoder.
- Both the encoder and decoder units employ two layers of temporal soft attention. This emphasis on the complete global view of video segments adds additional representational features.
- Additionally, to ascertain the superiority of the proposed framework, three variants of the models are examined.

- Additional trials on two benchmark video captioning datasets demonstrate proposal's superiority over other existing standard methodologies.

4.2.1 Methodology

4.2.1.1 Preprocessing Unit

Video Preprocessing

In order to lessen the computational cost, we take 30 frames from each video that are evenly spaced. In this step, the VGG-16 model is used to extract the features from video frames, which is then fed into the encoding unit to provide a global view of the videos.

Text Preprocessing

In order to clean up the corpus, deleting any unnecessary spacing and special symbols that were there. We eliminate sentences with fewer than three words and more than 30 words since more than 90 percent of the sentences have lengths larger than three words and fewer than 30 words, respectively. The $< BOSs >$ and $< EOS >$ tokens are added at the beginning and end of each phrase, respectively, to mark the beginning and end of sentences. When a batch of these sentences is formed, a token $< pad >$ is added to ensure that all of the sentences are of the same length, which increases the computational speed of the batch.

4.2.1.2 Proposed Multi-layer Attention Model

The proposed framework results are obtained by coupling a state-of-the-art notion called attention with a variation of the RNN, particularly the LSTM network, which is capable of learning long-term dependencies. We are attempting to determine the impact of normalisation and model size on BiLSTM in terms of efficiency, performance, and accuracy, as well as gain a better understanding of the reasons for these outcomes.

The proposed multi-layer attention framework for video description generation is depicted in Figure 4.8. Architectures for encoder and decoder are used in conjunction to create the framework under consideration. There are 1024 hidden units in the encoder, which is made up of a BiLSTM. The encoder generates a 2048-byte output since this concatenates the BiLSTM's forward as well as backward LSTM outputs in a single operation.

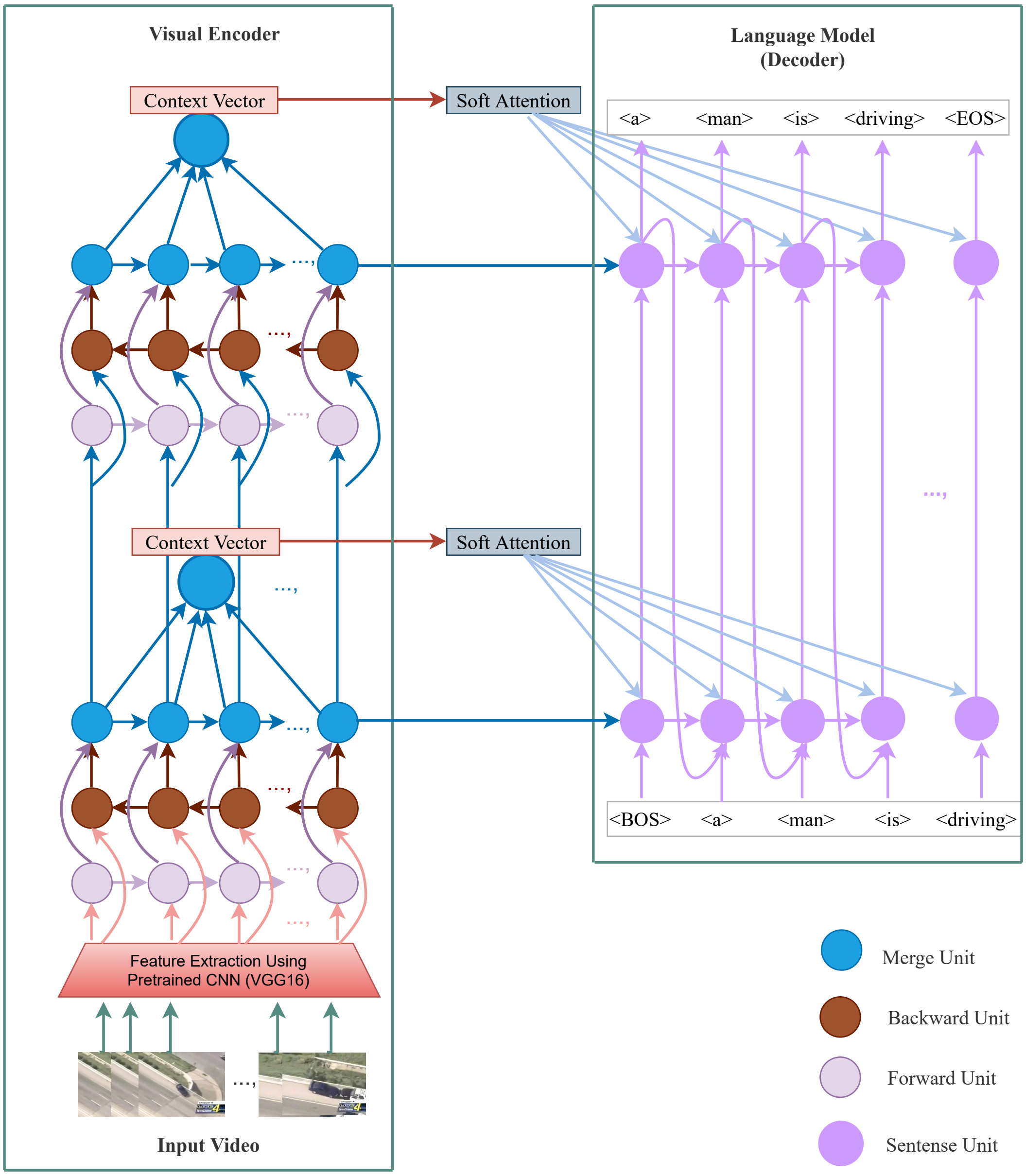


Figure 4.8: Proposed multi-layer attention framework

The decoder is constructed using a single direction layered LSTM unit. To achieve higher performance, this unit is merged with 2048 hidden units, a 1024-node attention layer, a 256-node embedding layer, and a fully - connected layers with nodes matching to the vocabulary of the corpus. Additionally, the decoder is composed of a fully connected layers with nodes representing the vocabulary of the corpus. The decoder LSTM has a concealed size that is twice as large as the encoder LSTMs, which is a significant advantage.

LSTM encoders are bidirectional, which means that their output is double the size of their hidden layer, as explained above. Utilizing a decoder of above mentioned size enables us to take advantage of the encoder's hidden states in the LSTM decoder, which is a considerable advantage. This permits the encoder's overall video content interpretation to be propagated to the decoder's global video representation.

The encoder BiLSTM is used to generate a global representation of the input video from each video that has been processed once it has been obtained after preprocessing. This data is kept in order to make it available to the decoder's attention layer at each and every time step of the usual decoding stage. When a end of the time step is reached, the attention unit delivers a context vector that contains the encoder output and the decoder's hidden state. When a sentence is input, the decoder goes over each word and produces the next word in the sentence. At the input stage, each word is fed into the decoding embedding unit, and the decoder embedding's output is fused with the context vector received from the attention layer. As input, this combined vector is passed to the LSTM decoder, which decodes it. The output of the LSTM is sent to a fully connected layers, which generates a vector with a length equal to the vocabulary size of the corpus and including information about the next term.

Sequential Model Using LSTM-Based Neural Network

To use a single input sequence, conventional RNN can theoretically take account of arbitrary long-term relationships in word sequence. When utilising LSTM units as RNNs, the vanishing gradient problem is partially solved due to fact that LSTM units ensure gradients to continue to flow unchanged or unmodified. Initially, a video clip $V = (F_1, ..., F_N)$, is used, with F_t denoting the video's t^{th} frame as the initial starting point for the captioning task. In this case, the primary goal is to encode video with words and express the result as a feature vector $V_{feature}$. The repeating nature of each frame must be considered in order to emphasise the time dependence of the frames' content. The RNN variation maps the input word sequence $X = (x_1, ..., x_t)$ to an output word sequence $Z = (z_1, ..., z_t)$, which can be expressed as

$$h_t = \phi(W_{hx}x_t + W_{hh}h_{t-1}), \tag{4.22}$$

$$z_t = \psi(W_o h_t), \tag{4.23}$$

Where

- $X = (x_1, ..., x_t)$:specifies input sequence at each step
- $Z = (z_1, ..., z_t)$: determines the order of the outputs at each step
- W_{h*} : Weights that are related to previously hidden states / the present input.
- W_o translates the hidden states between the hidden and output spaces.
- h_{t-1} and h_t : represent the RNN's hidden states at $t-1$ and t, respectively.
- ϕ and ψ represent nonlinear, two functions, respectively.

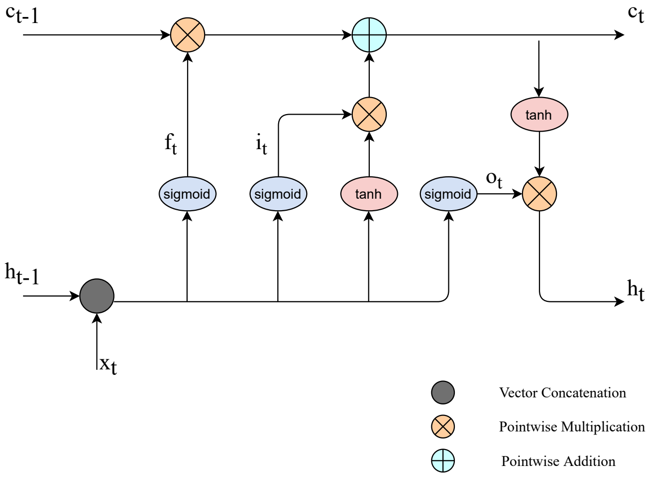


Figure 4.9: LSTM Architectural Preview.

While traditional RNNs suffer from the gradient vanishing or explosion problem, an upgraded RNN stores information in a memory cell and uses numerous control gates to have read-write operation from and to the memory unit or cell respectively, resulting in improved performance when leveraging extremely long temporal dependency relationships. Refer to Figure 4.9 for an illustration of the core LSTM architecture, and all of the gate information can be logically stated as follows:

$$i_t = \sigma(W_{ix} x_t + W_{ih} h_{t-1}), \tag{4.24}$$

$$f_t = \sigma(W_{fx} x_t + W_{fh} h_{t-1}), \tag{4.25}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1}), \tag{4.26}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \phi(W_{cx}x_t + W_{ch}h_{t-1}), \tag{4.27}$$

$$h_t = o_t \odot \phi(c_t), \tag{4.28}$$

- W_{*h} : Weights that relate each gate in the LSTM to previous hidden states.
- W_{*x} : Weighing units that connect the current input to each gate.
- σ : depicts the sigmoid nonlinear activation functions.
- ϕ : depicts the hyperbolic tangent nonlinear activation functions.
- $\odot$: represents the operation of element-by-element multiplication.

Bidirectional LSTM

CNNs have entirely independent inputs and outputs, but in some cases, the model may need to recollect prior meaningful words in order to choose the next relevent word. For example, you might be watching a video clip and pausing to estimate the finish; your guess will be based on already watched portion of clips and what interpretation has come to mind so far. RNN recalls the previous event and tries to predict the next word in this way. This way it tunes to solve the CNN problem by introducing a hidden units as a layer into the network.

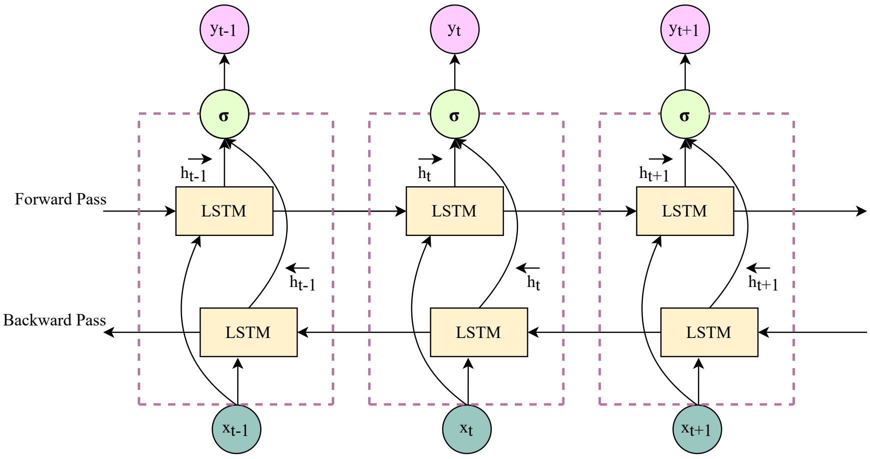


Figure 4.10: Bidirectional LSTM architecture.

An LSTM recalls every piece of information over the course of time, just as it remembers prior inputs. It is advantageous in the prediction of time series. Bidirectional

RNNs connect two RNNs together, allowing them to provide information on both the forward and backward sequences. LSTM provides stronger sequence processing capabilities and has the capacity to detect lengthy dependencies in sequences. LSTM-based networks are used to analyse the temporal feature for video and, the framework uses them to do so. It does this by mapping the video-level activity into the language model, which results in word-for-word video descriptions.

Whilst employing unidirectional LSTM, only past data may be used as inputs; thus, only previous data can be preserved. Alternatively, when employing BiLSTM, inputs can be processed in either direction: forward or backward. This strategy is far more effective at any moment in time, because it allows you to obtain knowledge from both the future and the past by combining two latent states.

In BiLSTM, contextual information is processed in both the forward and backward directions, allowing for the retention of information from both the past and the future. According to recent study, bidirectional RNNs produce better results when relying on sequential voice recognition processing and image captioning for a lengthy period of time. The Bidirectional LSTM architecture used in our proposed framework depicted in Figure 4.10.

Temporal Attention

The study in recent past on neural deep learning network has made extensive use of the attention mechanism, particularly in domains requiring vision, such as video/image captioning and machine translation. The basic notion is to increase the emphasis on a specific section of an image or video frame. A mechanism of attention that placing a greater emphasis on essential or crucial video frames with objectivity and their associations such as human actions, as opposed to its spatial mechanism, which promotes on the image's more semantically significant components.

Temporal attention can be viewed as a context set of visual elements with a window of visuals. Depending on the context, these visual features are referred to as regions or frames. At each discrete time step, the attention vector in Equation (4.29) can be created in conjunction with the dynamic weights for each visual element. The value in Equation (4.30) is appropriately accommodated by the dyanamic weights. For each visual element along the last concealed state shown in Equation (4.31), a relevancy score is produced.

$$sa_t = \sum_{i=1}^{m} \alpha_i^t vc_i, \tag{4.29}$$

$$\sum_{i=1}^{m} \alpha_i^t = 1, \tag{4.30}$$

$$\gamma_i^t = W_{rel} tanh(W_a vc_i + U_a h_{t-1} + b_a), \tag{4.31}$$

Equation (4.32) used to standardise the acquired relevance ratings.

$$\alpha_i^t = exp(\gamma_i^t) / \sum_{j=1}^{m} exp(\gamma_j^t), \tag{4.32}$$

- *FOV* : Field of View
- vc_i : Represents i^{th} element of context set.
- $VC = (vc_1, ..., vc_m)$: Denotes visual context set.
- α_i^t :Signifies dynamic activation weights for each element in context set.
- sa_t : Generated attention vector.
- γ_i^t : Context set relevance score.
- W_{rel} : relevance parameter for Context set .
- W_a :Context element parameter.
- U_a : Hidden state learning parameter.

The approach, soft attention attempts to replicate the attention allocation cycle for a given field of vision. By integrating forward and backward passes with temporal attention and applying temporal attention in the process, the current framework constructs sentences word by word in two phases. The approach produces a context set for each specific situation by utilising CNN highlights of edges and other latent information states in the merging layers. Moments after the production of a word, temporal attention directs the language model's focus to explicitly wordly locations that are more semantically significant. When the input word sequence is combined with the output, it is advantageous to consider the attention vector with the input, as illustrated in the base model, Figure 4.11.

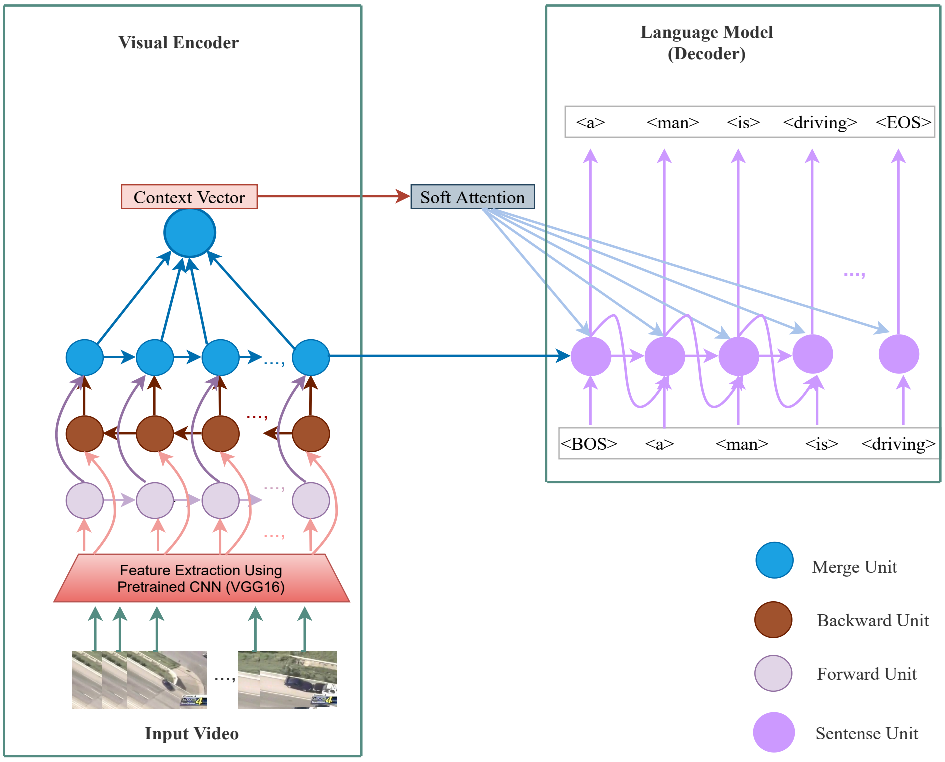


Figure 4.11: Base line model architecture

Batch Normalization's mathematical model

When training a deep neural network with multiple layers, the outcomes may differ due to factors such as the learning algorithm's design and the initial random weights. Due to the fact that the weights are updated after each mini-batch, the distribution of inputs to the network's deep layers differs with each mini-batch. They can make it more challenging for the model to acquire new skills by following a moving object. A deep neural network's "internal covariate shift" refers to a change in the proportion of inputs to layers due to the shift in the internal covariate dispersion of a network. Large neural network models that have a large number of inputs are standardised batch-by-batch using a technique known as batch normalization. This strategy significantly decreases the number of epochs necessary for training while simultaneously stabilising the model's learning process. During training, batch normalisation can be accomplished by computing the mean and standard deviation of each input parameter for each mini-batch. Finally, these results are employed to restore the network's representational capacity; a transformation logic is established. The mini batch mean is calculated using Equation (4.33),

$$\mathrm{E[x]}_B = \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{4.33}$$

where x_i is values of x over a minibatch, $B = x_{1....m}$.

The mini bacth variance can be defined with help of Equation (4.34),

$$\mathrm{Var[x]}_B^2 = \frac{1}{m}\sum_{i=1}^{m} \left(x_i - \mathrm{E[x]}_B\right)^2, \tag{4.34}$$

Now, for a layer with d-dimensions, x =$(x_1...x_d)$, each dimensions of its input can be normalized using, Equation (4.35),

$$\widehat{\mathrm{x}}_i^k = \frac{\mathrm{x}_i^k - \mathrm{E[x]}_B^k}{\sqrt{\mathrm{Var[x]}_B^{k^2} + \epsilon}}, \tag{4.35}$$

where $k \in [1, d]$ and $i \in [1, m]$. The ϵ is an arbitrary small constant for numerical staility. The final transformation is logically defined in Equation (4.36),

$$\mathrm{y}_i^k = \gamma^k . \widehat{\mathrm{x}}_i^k + \beta^k, \tag{4.36}$$

The γ and β are learnable parameter during the optimization process. By default, γ elements are set to 1 and β elements to 0.

Stacked LSTMs

Figure 4.12 depicts the organisation of layers in a hierarchical manner for stacked LSTM, which is comparable to that of a straightforward feed-forward network. The reason for this is to increase the complexity of the model. By stacking LSTMs, it is possible to define more complicated patterns in each layer of the model. However, it is not yet known how effective stacking LSTMs is theoretically, but it has been demonstrated empirically that deep RNNs perform better in some circumstances (Goldberg, 2016) when compared to shallower RNNs.

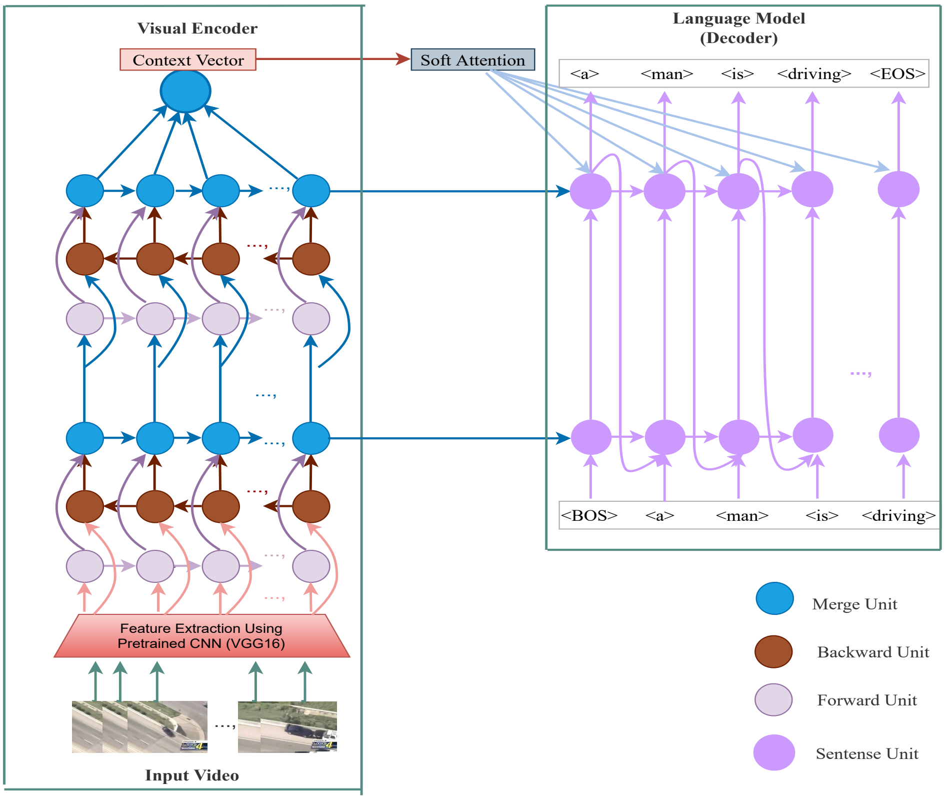


Figure 4.12: Stacked LSTM model architecture

Multi-Layer Attention

Following preprocessing in this proposed framework, each video is transmitted onto the encoder BiLSTM, which produces a global representation of the original video. During each time step, the encoder's output is kept in order to be fed into the decoder's attention (Bahdanau et al., 2014) layer. During each time step, the attention unit receives the encoder output from the decoder as well as the hidden state and returns a context vector. When a sentence is input, the decoder takes each word in turn and produces the following word in the sentence.

The framework proposes adding two layers of LSTM and two attention layers shown in Figure 4.8. The outcome of the framework is discussed in the result section. The intent behind the framework as based on earlier research by (Schuster and Paliwal, 1997; Karpathy and Fei-Fei, 2015; Ullah et al., 2017). The performance of bidirectional LSTMs is always significantly superior than the performance of unidirectional models, which has been found in a variety of fields such as image captioning, speech recognition, and action recognition. Adding another layer of attention allows for a more in-depth examination of the results of both layers of attention. The addition of two layers of attention to the framework can help it to optimise the results even more effectively.

4.2.2 Experiments and Results

4.2.2.1 Dataset

MSVD

This standard dataset called MSVD (Chen and Dolan, 2011) corpus, is comprised of 1970 videos and additional 80000 captioning sentences. Each video clip is between eight and twenty-five seconds long. Each sentence has roughly seven words, and each video contains approximately 43 sentences total for the duration of the video. The dataset contained 1970 videos, which was utilised in this research work. The dataset divided into two groups of 80 percent training and 20 percent test sets, respectively.

MSRVTT

The Microsoft Research Video to Text (MSR-VTT) (Xu et al., 2016) corpus has ten thousand video clips with twenty descriptive text for each clip. The dataset categorises videos broadly, including ”music,” ”TV shows,” and ”tourism.” Each clip lasts between 10 and 30 seconds. The dataset contains about 20,000 unique words, with an average of around ten words per description. The framework was built using a dataset split into 6513, 497, and 3000 rows for train, validation, and test, respectively.

4.2.2.2 Experiments

Inspire by the base line (Bin et al., 2019), in this work we present a total of four frameworks, one of which being the base model. The following paragraph summarises and lists all of the models.

- Base line model: This design is backed up by research outlined in (Bin et al., 2019). The encoder and decoder, respectively, are made up of one BiLSTM and one unidirectional LSTM, as indicated in the Figure 4.11.
- Base line model with Batch-Normalization: A batch normalisation layer at the encoder's output and another batch normalisation layer at the decoder's LSTM output were incorporated in this model.
- Stacked LSTM: Two BiLSTMs are layered together for the encoder in this architecture, while two unidirectional LSTMs are stacked together for the decoder as

depicted in Figure 4.12. The attention layer makes use of the encoder's second BiLSTM output at every time interval, as well as the hidden state of the decoder's second LSTM. However, the hidden states of both language model LSTMs were initialised with the hidden states of the visual model.

- Multi-layer attention model :The encoder and decoder are constructed by stacking two BiLSTMs for the visual model and two unidirectional LSTMs for the language model, respectively. The following step is to use the output of the first BiLSTM of the visual encoder at each time step, as well as the hidden unit of the first LSTM of the langauge model, for the first attention layer. The second attention layer is formed using the outcome of the second BiLSTM of the visual encoder at every sampling interval, as well as the hidden unit of the second LSTM of the word embedding. To preconfigure the hidden states of either the LSTMs in the langauge model, they must first be populated using the visual encoder's hidden units. This proposed architecture is shown in Figure 4.8.

The fundamental distinction between the Base line model and Stacked LSTM models is that the Stacked LSTM model's encoder and decoder are built of two layered LSTMs, as previously described and illustrated in Figure 4.12. Similarly, the second LSTM output is considered while building the soft attention mechanism in the encoder, and it is used to output one word at a time when developing the word output mechanism in the decoder. Due to the fact that the Stacked LSTM Model has two LSTM across both the visual encoding as well as language model stages, it is twice as large as the Baseline. It is imperative to highlight between the Stacked and Multi-Layer Attention Models because the Multi-layer attention model incorporates an additional attention layer interconnecting the encoder's first BiLSTM and the decoder's first LSTM, as depicted in Figure 4.8.

Implementation Details

The work took into account both the dataset video-sentence as well as a sample. When an input sentence sequence is fed further into the decoder, a new word is produced alongside the original word. The proposed models were optimised using an Adadelta (Zeiler, 2012) and initial learning parameters. The additional hyperparameters are defined as follows: ρ with a value of 0.9 and ϵ with a value of 10^{-6}, respectively. In this instance, a mini-batch of size 64 was employed to train the model. The framework trained iteratively until the best results.

- **MSVD :** To train these models using the dataset, the framework was created using open source Google Colab. Each epoch took 570 seconds, 800 seconds, and 1040 seconds, respectively, for the Base Model, the Stacked LSTM model, and the Multi-Layer Attention model.

- **MSR-VTT :** To train these models using the dataset, the framework was created using open source Google Colab. Each epoch lasted 1156, 1520, and 1940 seconds, respectively, for the Base Model, Stacked Model, and Multi-Attention Model.

A word was constructed by initialising the global video interpretation with the $< BOS >$ token and then outputting the following word using the decoder's output. The framework includes the previous output word as an input word for that video's caption until the decoder outputs a $< EOS >$ token or the maximum count of words possible for that video's caption. Regardless of whether the model accurately captions a video, if the amount of words in the reference sentences exceeds the number of words in the caption, the model's BLUE (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) scores are dropped, respectively. To obtain the required results, the framework determines the maximum count of words to output as the average count of words in the reference collection for that individual video clip.

Results and Analysis

The metrics BLEU and METEOR are two of the most commonly used in video/image captioning. METEOR performance metric pulls from the source documents and evalautes the precise steming, paraphrasing, and synonym matching. It makes use of the WordNet database (Feinerer and Hornik, 2020) and determines similarity scores at sentence level, allowing it to capture all semantic components of a sentence.

When comparing a candidate sentence to numerous reference sentences, BLEU-n uses modified precision, the ratio of the number of candidate n-grams in the corpus to the total number of candidate n-grams.

The BLEU score assesses textual and lexical coherence but not semantic coherence. Because METEOR is more robust than BLEU in our research, we used METEOR as our prominent metric and BLEU as a supplementary metric to evaluate model variations.

Table 4.8 contains information about the performance of all the models on the MSVD and MSR-VTT datasets. When compared to the Base line model, the Base line model with (Batch-Normalisation) outperforms the Base line on the test set by a sig-

Table 4.8: Performance of proposed multi-layer attention models on MSVD and MSR-VTT datasets.

MODEL	MSVD					MSR VTT				
	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	BLEU1	BLEU2	BLEU3	BLEU4	METEOR
Base line model	66.01	49.42	38.69	27.19	48.14	57.75	37.49	29.50	16.05	36.25
Base line model with BN	62.07	40.28	27.07	16.61	39.30	63.09	38.84	26.99	14.02	35.82
Stacked LSTM	67.49	51.98	41.90	31.23	49.19	58.18	41.41	32.02	17.61	37.88
Multi-layer attention(Proposed)	70.50	56.62	49.60	33.07	51.77	60.33	43.72	34.12	19.61	39.47

nificant margin. Stacked LSTM model performance exceeds both the Base Model and the Base Model with (Batch-Normalisation). It regularly generates high-quality results. When both models are trained to the exact count of epochs, it outperforms the Base line model considerably. In comparison to the Base line model, the Stacked/layered LSTM model is twice as large and hence capable of incorporating more detailed semantics from the video clips than another two models.

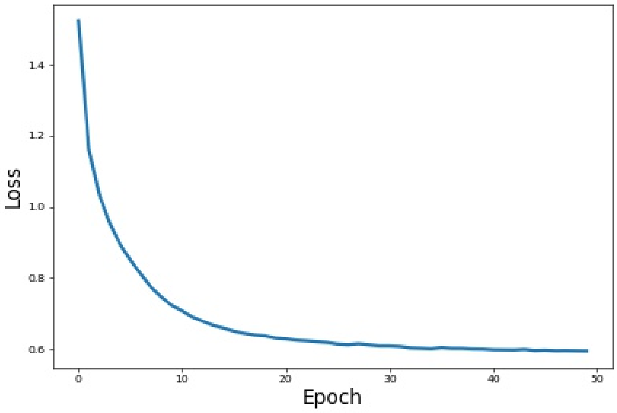


Figure 4.13: MSVD training loss at each epoch

The Figure 4.13 depicts the training loss curve for the developed framework using the MSVD dataset. This demonstrates that the model loss dropped steadily and stabilised around the 50-60 range epochs. The best findings have been selected and are given in Table 4.8. The proposed Multi-layer attention model outperforms the other three models in terms of overall performance. However, even though the proposed framework size is the same as that of the Stacked LSTM model and the training loss of both models is virtually the same, the proposed framework outperforms both models on the test datasets, demonstrating that the first attention layer has an impact. When comparing the Stacked LSTM model to the Multi-layer attention model, the latter model benefits from the additional layers of attention.

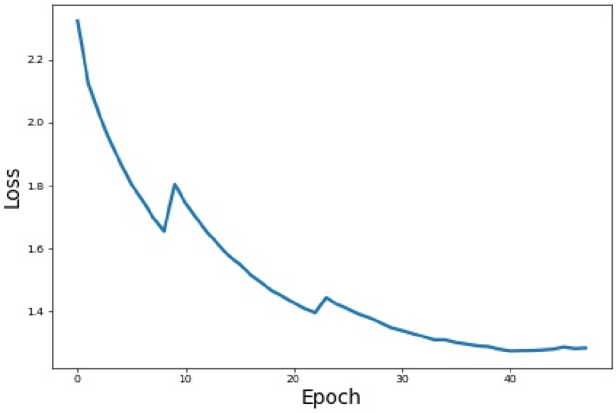


Figure 4.14: MSRVTT training loss at each epoch

The Figure 4.14 displays the MSR-VTT dataset's training loss curve. This illustrates that the model loss decreased gradually and stabilised around the 40-50 epoch range. The performance results of all the model measured on the MSR-VTT dataset is also summarised in Table 4.8. The proposed Multi-layer attention model outperforms all three of the other variations in terms of performance by a wide margin. For this reason, relevant words have been focused by soft attention at each layer, allowing them to be predicted.

All of the variants in the proposed framework were evaluated using the performance metric METEOR, which was then compared to some of the existing video captioning state-of-the-art works on both datasets, as shown in Table 4.9. As a result of semantic attention being paid at both the encoding and decoding stages in our implementations, all of the variations on both standard datasets that were used exceeded all of the other findings. The adoption of BiLSTM has also been shown to have a considerable impact on the performance of the models in question.

Table 4.10 displays the performance metric BLUE score, BLEU4, for all variations of our proposed framework when compared to some existing state-of-the-art video captioning works on both datasets. To the best of our understanding, the suggested framework surpassed practically most of the existing video captioning results that have been listed. In contrast to this, a satisfactory result on the bigger dataset MSR-VTT does not indicate that, even though sufficient semantics are involved in the proposed framework, further fine-tuning of parameters will result in a positive result.

Table 4.9: Performance comparison of METEOR score with state-of-art-methods

MODEL	METEOR	
	MSVD	MSRVTT
LSTM(Pan et al., 2016)	26.9	23.4
LSTM-E[VGG] (Pan et al., 2016)	29.5	-
LSTM-E[C3D] (Pan et al., 2016)	29.9	-
MM-VDN (Xu et al., 2015)	29.0	-
LK (Venugopalan et al., 2016)	30.3	-
S2VT-unidirectional (Venugopalan et al., 2015)	29.6	25.2
S2VT-bidirectional (Venugopalan et al., 2015)	29.7	25.6
S2VT-reinforced (Venugopalan et al., 2015)	29.9	25.9
S2VT-VGG (Venugopalan et al., 2015)	29.2	-
S2VT-VGG+Flow(Alexnet) (Venugopalan et al., 2015)	29.8	-
DVWA-uni (Bin et al., 2019)	29.6	25.7
DVWA-BiLSTM (Bin et al., 2019)	29.8	26.1
DVWA-ReBiLSTM (Bin et al., 2019)	30.3	26.2
DVWA-uni SA (Bin et al., 2019)	30.2	25.9
DVWA-BiLSTM SA (Bin et al., 2019)	30.5	26.2
DVWA-ReBiLSTM SA(shortcut) (Bin et al., 2019)	30.7	26.4
DVWA-ReBiLSTM SA(attention) (Bin et al., 2019)	30.9	26.6
Base line model	48.14	36.25
Base model with BN	39.30	35.82
Stacked LSTM	49.19	37.88
Multi-layer attention model (Proposed)	51.57	39.47

As a part of ablation study we performed several other experiments. The outcomes of suggested framework with dropout parameters (Srivastava et al., 2014) are presented in Table 4.11. The results indicate that the framework discovers more value when nodes are not dropped than when nodes are dropped. This could be because two-layer attention strategy is connected with layered bidirectional encoders and unidirectional decoders. Due to the framework's two-layered attention, it intelligently selects the best fragments or visual frames from which to learn and anticipate new values without interfering with the growth or reduction of network nodes.

In addition to VGG-16, the proposed system is evaluated using another feature vec-

Table 4.10: Performance comparison of BLUE-4 Score with state-of-the-art-methods.

MODEL	BLEU4	
	MSVD	MSRVTT
STAT (Yan et al., 2020)	52.0	39.3
SpatioTempo(Aafaq et al., 2019)	47.9	38.3
LSTM (Pan et al., 2016)	31.2	-
LSTM-E[ALEX](Pan et al., 2016)	38.9	-
LSTM-E[C3D](Pan et al., 2016)	41.7	-
FGM (Xu et al., 2015)	13.68	-
LSTM-YT (Venugopalan et al., 2014)	31.19	-
MP-LSTM (Venugopalan et al., 2014)	33.3	-
Base line model	27.19	16.05
Base model with BN	16.61	14.02
Stacked LSTM	31.23	17.61
Multi-layer attention model (Proposed)	33.07	19.61

Table 4.11: Performance score with tuned parameters.

MODEL	MSVD					MSR VTT				
	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	BLEU1	BLEU2	BLEU3	BLEU4	METEOR
Multi-layer attention (Proposed) without dropout	70.50	56.62	49.60	33.07	51.77	60.33	43.72	34.12	19.61	39.47
Multi-layer attention (Proposed) with dropout	67.79	52.29	45.36	30.36	50.59	58.02	41.30	31.82	16.99	38.35

tor named NASNet Feature Extractor (Zoph et al., 2018). Table 4.12 summarises the findings. As demonstrated in Table 4.12, this Feature Extractor explored with and without a drop layer. Our model fared better when no drop out is included in the BLUE performance metric, however METEOR performs better when a drop layer and NASNet feature extractor are included. This illustrates that, as previously demonstrated, the Multi-attention framework optimises to select the best segments or frames, resulting in a high METEOR score.

Time Complexity Analysis

The proposed approach consists of a CNN feature extractor followed by a BiLSTM based encoder-decoder model. So, the total complexity of the system comprises the complexity of CNN model used and the complexity of training the residual BiLSTM

Table 4.12: Performance score with different Feature Extractor

MODEL	MSVD				
	BLEU1	BLEU2	BLEU3	BLEU4	METEOR
Multi-layer attention model (Proposed) without dropout and NASNet Feature Extractor	60.10	41.27	34.36	19.81	39.40
Multi-layer attention model (Proposed) with dropout and NASNet Feature Extractor	58.29	38.87	31.65	17.16	42.37

model. This is approximately equal to $O(CNN\, model) + O(BiLSTM\, model)$. Thus, the overall complexity can be defined as $O(N) + e * b * T_b * O(B_L)$. Where N, e, b, B_L, and T_b, denote the number of parameters in CNN feature extractor, epochs, batches, parameters in layered BiLSTM model, and time estimation of backward and forward passes, respectively.

Limitations

The experimental studies revealed that the proposed model performed admirably on the training part of the standard dataset used. However, the outcome on the test portion is less than the result on the training portion. Though the framework earned a higher METEOR score than any previous study, it fell short of achieving a superior BLUE score. Whereas appropriate measures must be taken to increase the BLEU score. The proposed model takes into account the average length of phrases used during training. As a result, it may exclude some critical terms from the sentences. Additional study in this area may help improve the model's performance.

4.2.3 Summary

It is proposed in this research work to use a novel Multi-layer attention-based model for video captioning that is both efficient and effective, and it is then compared to other modifications of the base model. The framework makes use of two LSTM networks: one for the visual encoder and another for the language model.

The visual encoder is implemented using stacked BiLSTMs on resampled video data in order to maintain input at every time interval for attention. The encoder's hidden states are then used to create a global perspective of the video, which is subsumed into the language model. The decoder unit, which was utilised to convert the video captions into sentences word for word. The framework's implementation was carried out on the MSVD and MSR-VTT datasets, respectively. To the best of our knowledge, the proposed approach surpassed practically most of the existing state-of-the-art visual captioning results that have been published and listed. The best way to increase the effect in the future is to modify it even further in order to achieve the best outcome on a larger dataset.

4.3 Video Captioning using a Sentence Vector-enabled Convolutional Framework with a Short-Connected LSTM.

The primary goal of video/image captioning is to convey in plain natural language the dynamics of a video clip. For this aim, the most widely used design paradigm is the groundbreaking structurally enhanced encoder-decoder architecture. Recent advances emphasise the importance of utilising a variety of novel structural alterations to increase efficiency while also establishing their viability in real-world applications. Encoder-decoders are increasingly using well-known and well-researched technology developments such as deep CNN and Sentence Transformers. In this research work, a strategy for efficiently captioning videos by combining a CNN and a short-connected LSTM-based encoder-decoder model with a sentence context vector. This sentence context vector emphasises the relationship between the video and text spaces. The attention mechanism, inspired by the human visual system, is utilised to selectively focus on the context of the essential frames. Additionally, a contextual hybrid embedding block for connecting the two vector spaces formed during the encode and decode stages. The proposed architecture is designed using well-established CNN architectures and a variety of word embeddings. It is evaluated using two benchmark datasets for video captioning, MSVD and MSR-VTT, and conventional assessment metrics such as BLEU, METEOR, ROUGE, and CIDEr. The following are the most significant contributions made in this research method.

- The proposed model was motivated by efforts to extract features using a range of deep neural trained networks, including NASNet Large (Zoph et al., 2018), Inception-v4 (Szegedy et al., 2017), ResNet152 (He et al., 2016), and VGG-16 (Simonyan and Zisserman, 2015).
- The framework experimented with various combinations of recent word embeddings such as BERT (Devlin et al., 2018), ELMo (Peters et al., 2018), and GloVe (Pennington et al., 2014) and compared the results to proposed neural networks.

- Proposed a model adapted with a novel notion of Residual Connection Network (RCN), which overcomes the concerns of vanishing gradients and accuracy saturation.

- The proposed framework also experimented to use the novel Contextual Hybrid Embedding Network (CHEN) to augment the model with additional contextual data.

- The framework augmented with extra attention to the decoding stage by using multi-headed attention.

- Trials on two real-world video captioning datasets have been conducted to demonstrate the proposed model's superiority to existing state-of-the-art approaches.

- The model's outcomes were measured using a variety of conventional performance indicators, including BLUE (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE (Lin, 2004), and CIDEr (Vedantam et al., 2014).

4.3.1 Methodology

To begin, model extracts frames from the videos during the preprocessing stage. Adjustments are made to the retrieved frames to make them compatible with the input dimensions of pre-trained CNN models, notably VGG-16((Simonyan and Zisserman, 2015)), NASNet-Large((Zoph et al., 2018)), Inception-v4 ((Szegedy et al., 2017)), and ResNet-152 ((He et al., 2016)). The proposed framework's objective is to generate a summary of the videos in English-like sentences. To do this, this research work proposed a unique encoder-decoder system based on the encoder's CHEN and RCN. The proposed framework takes as input information extracted using a pretrained CNN model and encodes them in order to acquire the video's spatial and temporal features. Figure 4.15 illustrates the proposed approach.

4.3.1.1 Preprocessing

Successive frames in a video have a lot in common, therefore they have the same number of characteristics. We chose a set of essential frames from the video to decrease the number of features. The extraction of key frames is based on a template matching scheme. Equation (4.37) is used to compute the template match score.

$$T_{Score}(x,y) = \frac{\sum_{x',y'}(Temp(x',y') \cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'} Temp(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \tag{4.37}$$

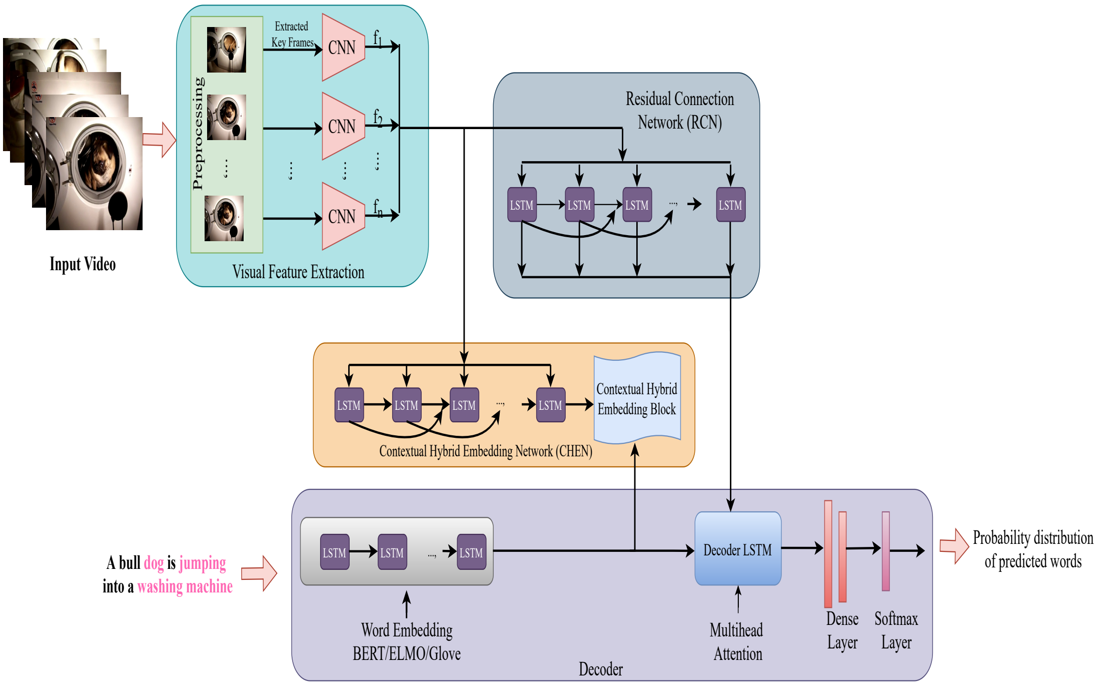

Input Video
Preprocessing
Extracted Key Frames
CNN
CNN
CNN
f1
f2
fn
Visual Feature Extraction
Residual Connection Network (RCN)
LSTM
Contextual Hybrid Embedding Block
Contextual Hybrid Embedding Network (CHEN)
A bull dog is jumping into a washing machine
Word Embedding BERT/ELMO/Glove
Decoder LSTM
Multihead Attention
Dense Layer
Softmax Layer
Decoder
Probability distribution of predicted words

Figure 4.15: The proposed video captioning framework.

Where I, $Temp$, and T_{Score} denote, respectively, the image, the template, and the template match score. (x, y) specifies the pixel coordinates. The parameter x' can be any value between 0 and the width of the template, while y' can be any value between 0 and the height of the template. In the proposed system , we compute a similarity between two frames, I and $Temp$. Where I denotes the current keyframe and $Temp$ denotes the next frame in the sequence. If the T_{Score} value is greater than the threshold, the frame $Temp$ is skipped due to its similarity to the keyframe I, and the next frame in the sequence is set as $Temp$. If the T_{Score} is less than the threshold, the key frame $Temp$ is chosen, and the process continues until the end of the frame sequence.

4.3.1.2 Feature Extraction

Starting with pre-trained neural networks, the proposed system extracts feature and generates matching feature vectors from videos. It is necessary to enlarge and scale the video clips to fit the input requirements of pre-trained deep learning models, which are used to represent the majority of videos. The pre-processed images are fed into a CNN model sequentially. With the proposed architecture, we attached and deployed the following networks: NASNet-Large, Inception-v4, ResNet-152, and VGG-16.

Algorithm 4: Algorithm for selecting key frames using template matching

```
Input: Input video V; Threshold T
Output: Set of key frames F extracted from video V
Initialize: $F \leftarrow \emptyset$
  Extract first frame $f$ from V
  $k_f \leftarrow f$
  $c_f \leftarrow f$
  while $c_f \neq$ last frame in video do
      if $k_f == c_f$ then
          $F = F \cup k_f$
      else
          $C \leftarrow$ Template Match between $k_f$ and $c_f$ using Equation (4.37)
          if $C \leq T$ then
              $k_f = c_f$
          else
              discard $c_f$
              $c_f \leftarrow$ next frame in the sequence
          end
      end
end
```

NASNet-Large

NASNet-Large accepts as input, a video frame with a resolution of (331 × 331) and generates a feature vector with a resolution of 4032 per frame. While the fundamental concept of NASNet-Large has been predefined, similar to (Zoph et al., 2018), the building blocks or cells have not yet been determined. Alternately, they are examined using a reinforcement learning search strategy in which the number of iterations and convolutional filters are treated as scalable free parameters. It associates the normal and reduction cell types to generate the feature vector.

VGG-16

VGG-16 was created in Oxford (Visual Geometry Group). It accepts as input images with a resolution of 224 × 224 pixels. The length of the resulting feature vector is 4096 bytes. The convolutional layers of VGG-16 use a 3 × 3 receptive field, the smallest size capable of capturing all features. Additionally, there are 1 × 1 convolution filters that linearly transform the input before passing it to a ReLU unit. VGG-16 consists of three interconnected layers, and ReLU activates all of the hidden layers.

Inception-v4

Inception-v4 is a CNN architecture that improves upon earlier incarnations of the Inception family by simplifying the architecture and employing more inception modules than Inception-v3. It consists of 22 layers (27, including the pooling layers). At the conclusion of the last inception module, it employs global average pooling. Obviously, it is a really sophisticated classifier. As with any extremely deep network, it is susceptible to the problem of vanishing gradients.

ResNet-152

ResNet-152 is capable of forming a 152-layer dense network by using residual representation functions rather than signal representation directly. ResNet-152 provides skip connections (or shortcut connections) to match the input of the previous layer to the input of the next layer without altering the input. By avoiding a connection, you can construct a larger network. ResNet-152 introduced the notion of residual learning, in which the subtraction of a feature's input is learned by shortcut connections. It has been shown that residual learning can improve model training performance, particularly when the model incorporates a multilayer perceptron with much more than 20 layers, and can also solve the problem of deep network accuracy degradation.

4.3.1.3 Terminology and Notation

The concept of representing video clips that require captioning, mathematically can be expressed as, V = $\{v_1, v_2, v_3...., v_n\}$, where n is frame count value. As explained earlier, the visual features extracted from video V with the help of pre-trained models can be visualized with the following details. The F = $\{f_1, f_2, f_3, ..., f_n\}$ $\epsilon R^{d_i * n}$, where d_i represents dimensional view of the feature vector for a single frame. The feature vector of the caption is visualized by the symbol $W \epsilon R^{d_w * c}$, where d_w represents the size of a word embedding and c represents the caption's word count. The overall view of model's captioning relationships can be viewed as $W' \epsilon R^{d_w * c}$.

4.3.1.4 Encoder-Decoder

The presented model's encoder is a two-layered LSTM made of short connections or residual layers that impose non-linearity on the data processing. This unit permits and absorbs temporal characteristics extracted at the frame level during the preparatory stage from the two-dimensional pre-trained CNN data. LSTMs attempt to limit saturation inaccuracy by converting a series of frames to a fixed-dimensional vector representation.

The primary focus of the decoding stage is to develop the ability to guess the caption based on the video data. Each word in the caption is created separately and predicted with a proper contextual understanding of the surrounding words of the generated word. Thus, given an input sequence and the encoder's output, the decoder anticipates the caption's next word. The model is binded with attention that operates with separate heads called multi-head that incorporates the encoder result and the decoder state to pay attention to some areas of the encoded output that impact the final decoding unit output. The loss function used to represent the translation of videos to words is shown in Equation (4.38).

$$loss_1 = -\sum_{t=1}^{N_w} \log P(w_t|E, w_1, w_2, ..., w_t - 1) \tag{4.38}$$

The value of, N_w depicts the caption word count. The Equation (4.39) signifies the likelihood of the predicted word w_t being formed given the previously generated words w_1, w_2, ..., w_{t-1} and the output from the encoding unit *E*.

$$P(w_t|E, w_1, w_2, ..., w_{t-1}) \tag{4.39}$$

4.3.1.5 Contextual Hybrid Embedding Network (CHEN)

This proposed novel unit converts the feature map F from the video created during the preliminary stage to a 768-Dimension feature map using an LSTM layer. The resulting feature map was then compared to the hidden state of an LSTM autoencoder trained on the caption set during the training phase. The autoencoder uses the regenerated contextual caption vector with a fixed dimension to anticipate the next word to the decoding stage. As the contextual caption vector summarises the captions we termed it as sentence vector.

Now we assume, if the language model's sentence vector is SV and the sentence vector from CHEN is SV' , then the loss is determined as given in Equation (4.40).

$$loss_2 = \sum_{k=0}^{d_w} e_k \tag{4.40}$$

Where, e_k defined in Equation (4.41).

$$e_k = \begin{cases} \frac{1}{2}(sv_k - sv'_k)^2 & for \quad |sv_k - sv'_k| \leq \delta \\ \delta|sv_k - sv'_k| - \frac{1}{2}\delta^2 & Otherwise \end{cases} \tag{4.41}$$

4.3.1.6 The Proposed Multi-head Attention Mechanism

The proposed model is motivated by the concept of multi-head attention, which cycles through an attention mechanism many times in parallel. Multiple attention heads, intuitively, enable distinct elements of the sequence to be attended to differently. The attention unit achieves training stability by enhancing consistent performance improvements above conventional attention.

The proposed framework's encoding stage always provides detailed information about the video. This contextual information is also contingent on the model's comprehension of the frames. However, for the decoding stage to predict the next caption term being given the current term, just a subset of contextual data attributes must be selected. Thus, scalar dot product attention (Vaswani et al., 2017b) enables the decoding stage to focus exclusively on a portion of the encoding unit data. Take the following into account: Equation (4.42), Q, The value field stores data about the encoder's output, whereas the K field stores data about the decoder's previous state. The encoder output is weighted for various regions, depending on the current state of the decoder, in order to build a context vector for the following word. This is one of the heads of the attention system's several heads. Multiple heads enable the decoder to simultaneously attend to data from the encoder at various points in heterogeneous representational spaces. Equation (4.42) provides the mathematical definition for single-head attention.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_n}})V \tag{4.42}$$

Equation (4.43) visualises the concept of envisaged multi-head attention.

$$(Q, K, V) = (R_d W^Q, R_e W^K, R_e W^V) \tag{4.43}$$

The representational values R_d signifies the decoder output states, R_e symbolizes the encoder output states, $\{W^Q, W^K, W^V\} \, \epsilon R^{d_i * d_n}$ are the defined weights for multiple attention heads.The key notion d_i represents the dimension of the attention input and d_n depict the count of units in an attention head.

4.3.1.7 Training-Phase

To initiate, the model is trained using the language model to generate the relevant sentence vector for all captions in the datasets. Additionally, as illustrated in Figure 4.15, the proposed architecture is trained using hybrid loss built by merging $loss_1$ and $loss_2$ bridge the semantic gap between video and words. The hybrid loss defined in Equation (4.44).

$$loss = \lambda loss_1 + (1 - \lambda) loss_2 \tag{4.44}$$

The variable λ symbolises a tuneable hyper-parameter with values ranging from 0 to 1.

4.3.2 Experiments, Results and Discussion

Numerous experiments with various parameter combinations are conducted, and performance is evaluated using various benchmark evaluation metrics. Additionally, the proposed system's performance is compared to existing works using MSVD and MSR-VTT datasets.

4.3.2.1 Experiments on MSVD Dataset

Data preprocessing

The videos in the corpus have an average duration of 10.2 seconds. As a result, we sample 28 frames for every clip equally. Each frame is given to the NASNet-Large CNN unit, which extracts a 4032-dimensional feature vector from it. Thus, for each video we obtain a 28*4032-D feature vector. In this step each video contains at least 28 frames, therefore padding is unnecessary here. Captions for the clips are compiled from a variety of sources and vary in length. As a result, we eliminate the punctuation from the captions, lower case them, and tokenize them. The vocabulary is refined to eliminate misspelt and uncommon terms. Captions are lengthened to a maximum of twenty characters. Any caption that exceeds the maximum length is clipped, whereas captions with fewer than 20 tokens are padded. 768-D, 300-D, or 1024-D vectors are created for each token utilising word embedding algorithms such as BERT, GloVe, and ELMo. We assessed the efficiency of mentioned word embeddings using proposed model in Results and Analysis section. The tokens *bos* and *eos* denote the start and end of the caption, respectively. While *pad* and *unk* serve as padding and unknown words, respectively.

The suggested approach is motivated by the need to validate video captioning using datasets. The approach was initially validated using the MSVD dataset. The video to be processed is represented by the 28*4032 feature vector. It is then transferred to the encoder stage, which is a two-layer LSTM with a residual layer or a few short connections. It is believed that the LSTM contains 512 units. Thus, we acquire a 28*512-dimensional vector from layer one and a 28*512-dimensional vector from layer two, resulting in a 28*1024-dimensional vector from the encoder. A single 768-unit LSTM layer and a pre-trained LSTM autoencoder comprise the Contextual hybrid embedding block. The unit count of the attention layers is set to 512. It is expected that the decoder stage LSTM contains a total of 1024 units. For training, Adam Optimizer is employed with an 10^{-4} learning rate and a batch size of 64. A beam search with a beamwidth of three is used for testing.

Results and Analysis on MSVD

Numerous investigations were conducted using the proposed technique on the MSVD dataset based on the train-validation-test split described in (Venugopalan et al., 2015).

Table 4.13 exhibits the effect of performance scores on the three embeddings with pre-trained Feature Descriptors of our novel proposed approach. When BERT alone considered, NASNet-Large exceeds all other models except for the METEOR score. The advantage of this model is that it employs two convolutional cells, which aids in obtaining the optimal output. It employs a controller to optimise performance by utilising fewer parameters but also floating-point operations.

Table 4.13: Performance comparison of proposed framework on MSVD.

Embedding	**CNN model**	**BLEU1**	**BLEU2**	**BLEU3**	**BLEU4**	**METEOR**	**ROUGH**	**CIEDEr**
ELMO	ResNet-152	79	66.9	57.2	46.6	32.7	69.2	71.6
	VGG-16	77.2	63.7	52.8	42	31.2	67.4	63.6
	Inception-v4	79.8	68.2	58.4	47.5	33.2	69.2	72.4
	NASNet-Large	81.3	69.7	59.4	48.5	33.5	70	72.5
GLoVe	ResNet-152	77.1	65.5	55.2	44.9	30.6	67.6	70.1
	VGG-16	76.1	63.7	51.3	41.9	30.5	64.7	65.4
	Inception-v4	77.3	66.1	56.9	46.2	31.8	67.8	70.1
	NASNet-Large	78.4	68.5	57.2	47	32.2	68.5	70.8
BERT	ResNet-152	79	67.1	57.5	48.4	34.4	69.6	72.5
	VGG-16	77.8	64.3	53.9	43.6	31.4	67.2	63.7
	Inception-v4	80.7	69.4	59.6	48.8	34.3	70	78.1
	NASNet-Large	**82**	**69.9**	**60.3**	**50.3**	33.9	**70.5**	**78.6**

Table 4.13 also relates the ELMo embedding performance of all Feature Descriptors. According to the statistics, the proposed model outperformed all other models. The suggested model takes advantage of ELMo's benefits in terms of syntax, semantics, and polysemy notions in order to get superior outcomes. The obtained results also provides an overview of proposed model with pre-trained all Feature Extractor with a GloVe vector embedding. All other models were outperformed by the suggested Feature Extractor model. The reason for this is that both the encoding and decoding levels are augmented by context-sensitive data. This reliably extracts the region and its associated captions from the embeddings. Additionally, the multihead attention considerably improves the performance.

When the proposed model with NASNet-large alone is seen across all the three embeddings, the BERT findings outperformed the results obtained with the other two embeddings. The BERT makes use of the target word's context by examining the surrounding terms. This novel combination of the suggested NASNet and BERT word embedding outperforms the others.

Table 4.13 also summarises the performance of three different embeddings while using Feature Descriptor, Inception-v4 as a reference. The study demonstrates that BERT with Inception-v4 outperforms other methods. Notable explanations for this could include the increased number of inception modules on the encoding unit as well as the bidirectionally trained embedding, that provide a more accurate sense of language context and flow than single direction language models.

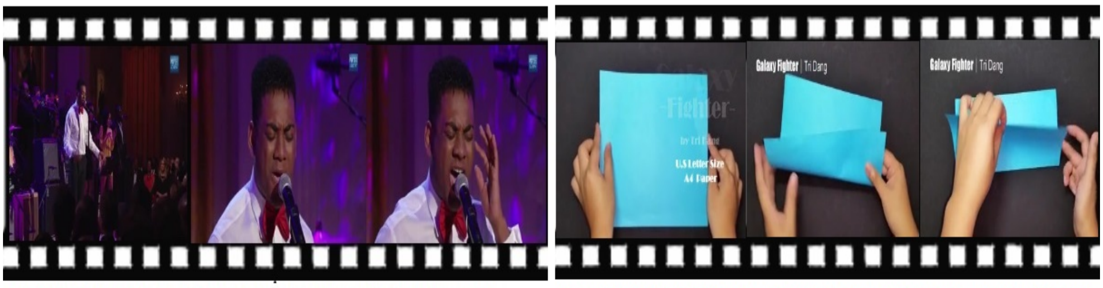
Additionally, the experimental findings obtained using the Feature Extractor VGG-16 and three-word embeddings followed a very similar pattern. While the results are marginally different for all three, BERT retains the performance advantage. This demonstrates how BERT works extremely consistently across nearly all Feature Descriptors.

The efficiency of Feature Descriptor ResNet-152 and its effectiveness on three embeddings is compared in Table 4.13. The score observation demonstrates that BERT surpasses the other two embeddings, despite the small variation in outcomes. It is also clear that the ResNet-152 performance score outperforms VGG-16 in practically every parameter. BERT's best feature is that it is extremely bidirectional and contextual. Transformers are used in the BERT, while LSTMs are used in the other two variants. The descriptor is a multi-layered method that, when used with the BERT, yields better results because of its more detailed discoveries.

Table 4.14: Comparison of proposed approach with the state-of-the-art methods on MSVD.

Methadology	B4	METEOR	ROUGE	CIDEr
STAT_LOC_V STAT_LOC_L (Yan et al., 2020)	42.9	31.0	-	62.5
UTS (Sah et al., 2020)	43.0	33.20	-	71.10
SE-GRU (Hao et al., 2020)	42.9	33.5	-	62.3
DM-TSC (Xu et al., 2020)	48.8	33.40	69.70	82.00
ELT-VC (Wei et al., 2020)	46.80	34.40	-	85.70
VC-BPLSTM (Nabati and Behrad, 2020b)	42.90	32.00	68.30	62.20
STD-SA (Aafaq et al., 2019)	47.90	35.0	71.50	78.10
BDE-DS(Chen et al., 2019)	44.10	32.12	-	70.10
NMB-SS(Lin and Zhang, 2021)	40.3	31.5	-	-
MARN (Pei et al., 2019)	48.6	35.1	71.9	92.2
GS-VD(Yadav and Naik, 2021)	41.2	33.4	-	-
Proposed Method	50.3	33.9	70.5	78.6

On the MSVD dataset, Table 4.14 compares the performance of a proposed model with other recent and state-of-the-art approaches. All other models were clearly outperformed by our suggested model with the Feature Descriptor NASNet-Large. We also experimented with the other two Feature Descriptors and compared them to the suggested model. The proposed model clearly outperformed in all performance criteria. BLEU4, METEOR, ROUGE, and CIDEr scores are shown in the Table 4.14. The results show that suggested model was more accurate than the other models at recognising the exact collection of words in the dataset, as measured by the n-gram hit ratio. Using the CIDEr metric, the suggested model came in fourth place. The model received significant ROUGE and METEOR ratings when compared to other models. Overall, the results demonstrate that a model with a short encoder-to-contextual vector relationship improves performance. In addition, the defined model combination learns a new set of information for the encoder's succeeding phases, as well as a collection of data from the embeddings.



Ground Truth: {a man is singing on stage, a man is singing, an is performing at stage, a man is singing before a crowd, musicians on stage performing }

Ours: a man is singing

Ground Truth: {a man making a paper airplane, a person folding paper, a person is folding a piece of paper, someone is folding origami, a man is doing origami }

Ours: a person is folding a piece of paper

Figure 4.16: The predicted and reference samples of proposed approach.

Qualitative Analysis

Figure 4.16 samples shows a practical comparison of our novel developed framework with ground truth captions. The video frames and original data captions from the MSVD dataset are also shown in the Figure 4.16. The new method yielded a sufficient number of video captions. In a few cases, however, the model incorrectly identifies the composite action and objects. The proposed model, on the other hand, performed a good job of identifying objects and activities. Further, model also makes advantage of contextual hybrid embedding, which aids in the development of relevant phrases.

4.3.2.2 Experiments on MSR-VTT

Let us suppose for convenience that the CNN feature extraction method and word embedding are NASNet-Large and BERT, respectively. The video to be studied is represented by the 80*4032 feature vector. It is then supplied to the encoding and a two-layer LSTM with either a short link or a residual layer. It is believed that the LSTM contains 512 units. Thus, we obtain an 80*512-D vector from layer one and another 80*512-D feature vector from layer two, yielding an 80*1024-D vector from the encoding stage. The contextual hybrid embedding block is composed of a single 768-unit LSTM layer and a semantic sentence vector. The attention layers' unit count is set to 512. It is expected that the Decoder LSTM contains a total of 1024 units. Adam Optimizer is used for training, with an 8e-4 learning rate and a batch size of 64. During testing, a beam search with a beamwidth of three is used.

Data preprocessing on MSR-VTT

The average duration of the videos in the corpus is 14.83 seconds. As a result, we sample 80 frames equally for each clip. Each frame is sent into a pre-trained CNN model, which generates an N-dimensional feature vector from it. N is 4032, 4096, 2048, and 2048 for NASNet-Large, VGG-16, Inception-v4, and ResNet-152, respectively.

Thus, for each video from NASNet-Large, we have an 80*4032-D vector. The videos' captions are derived from a number of sources and range in length. As a consequence, we remove all punctuation, lowercase, and tokenize the captions. The vocabulary has been whittled down to eliminate misspellings and rare terms. Captions are increased in length by a maximum of twenty characters. Captions that exceed the maximum length are clipped, and captions that include fewer than twenty tokens are padded. For each token, a 768-D, 300-D, or 1024-D vector is generated using methods such as BERT, GloVe, and ELMo. In the Results section, we assess the performance of our model to that of several word embedding. The bos and eos tokens denote the start and the end of the caption, respectively, while pad serves as a padding token and unk is used to forecast unknown phrases.

Results and Analysis on MSR-VTT

The comparison of proposed method's efficiency to that of state-of-the-art techniques on the MSR-VTT dataset utilising BLEU, METEOR, ROUGE, and CIDEr metrics id done. The experiment is repeated with the train-validation-test split as described in (Xu et al., 2016). The results are then assessed qualitatively and quantitatively. The qualitative outcome is depicted in Figure 4.16, which includes both the ground truth captions as well as the captions obtained with the proposed model.

In the quantitative analysis, results are contrasted with the BLEU, METEOR, ROUGE, and CIDEr scores to previously published studies. For feature extraction in this work, four pre-trained CNN models are used: NASNet-Large, VGG-16, ResNet-152, and Inception-v4. The proposed model with Inception-v4 as feature extractor surpasses NASNet-Large, VGG-16 and ResNet-152 respectively.

Table 4.15: Performance of proposed video captioning using a Sentence Vector-enabled Convolutional Framework with a Short-Connected LSTM system on MSR-VTT.

Embedding	CNN model	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGH	CIEDEr
ELMO	ResNet-152	71.7	54.3	42.5	32.7	23.6	54.2	32.5
	VGG-16	68.9	53.5	40.6	29.9	23.1	53.2	28.4
	Inception-v4	72.1	57.8	45	36.4	26.2	56.4	38.3
	NASNet Large	72.7	58.1	45.2	34.3	24.7	56.3	36.3
GLoVe	ResNet-152	70.1	54.3	42.5	32.7	23.6	54.2	32.5
	VGG-16	67.2	51.9	39.8	28.4	21.6	52.7	27.8
	Inception-v4	71.2	55.2	42.1	32.5	24.7	53.7	31.8
	NASNet Large	70.2	54.1	40.3	30.6	23.2	52.7	29.5
BERT	ResNet-152	72.6	57.1	43.7	32.9	24.6	55.4	35.5
	VGG-16	68.3	53.2	40.8	30.5	23.1	53.2	28.9
	Inception-v4	**73.6**	**59.4**	46.6	**39.2**	**27.8**	**59.3**	**40.5**
	NASNet Large	71.5	58.9	47.4	37	25.1	57.5	37.6

Three distinct word embedding approaches are used, including BERT, ELMo, and GloVe. We compared the results of each feature extraction technique to BERT, ELMo, and GloVe embedding. BERT embedding outdoes ELMo and GloVe in investigations. The same pattern is used to extract features from NASNet-Large, Inception-v4, ResNet-152, and VGG-16. Table 4.15 compares the performance of word embedding for NASNet-Large features. We can see that BERT embedding produced superior outcomes for all metrics.

Table 4.15 compares the performance of all pre-trained feature extractors in the unique proposed framework discussed in our study. When BERT embedding was utilised, it is clear that the Inception-v4 feature extractor topped all other models. The strength of Inception-v4 is that it extracts features by utilising more inception modules than most other models. The Inception-v4 model outperformed earlier versions due to its capacity to learn additional parameters from big datasets such as the MSR-VTT and its use of bidirectional embedding features.

When, Table 4.15 is considered for comparison, NASNet-Large's with various embedding strategies demonstrate that the BERT embedding composition is capable of producing practically any performance statistic. As previously noted, for the MSVD dataset, the NASNet equally performs better on larger datasets like MSR-VTT too.

Table 4.15 can also be viewed to compare an Inception-v4's performance to practical values derived from three embeddings. The model outperformed ELMo and GloVe embeddings significantly. The learning capability of the network, along with parallel max pooling and more inception modules on Inception-v4, improved the overall results. Additionally, a comparative analysis of a simple ResNet-152 model to three different embedding strategies is made. When BERT embedding was incorporated, the model outperformed the other embedding scores. The boost in virtually all scores is due to the addition of additional layers to learn complicated features, residual connections that provide more information to each level of the hidden layers, and a context-sensitive vector coupled to the encoding stage.

The observation is also noted in Table 4.15, a correlation between VGG-16 and three-word embedding models. The model's success with BERT demonstrates the critical nature of transformer-attached embeddings that fully comprehend the context of words within the reference phrase. As a result, the next stage predicts better sentences based on the first stage's input word. By using all the experimental analysis, BERT outperformed ELMo and GloVe with respect to all the feature extraction techniques and all the evaluation metrics. The proposed model compared to several existing works on the

MSR-VTT dataset, and the results are summarized in Table 4.16.

Table 4.16: Comparison of proposed approach with the existing method on MSR-VTT.

Models	**BLEU4**	**METEOR**	**ROUGE**	**CIDEr**
VD-MB (Xu et al., 2020)	37.90	28.40	59.3	40.5
EL-TI(Wei et al., 2020)	38.50	26.90	-	43.70
Ms-VC(Nabati and Behrad, 2020a)	39.50	27.50	-	42.80
STAT_LOC_V (Yan et al., 2020)	35.1	24.6	-	36.9
STAT_LOC_V (Yan et al., 2020)	35.2	25.1	-	35.8
STAT_LOC_MotFeat (Yan et al., 2020)	36.5	25.4	-	39.9
M3-V (Wang et al., 2018)	35	24.6	-	-
M3-C (Wang et al., 2018)	35.1	25.7	-	-
M3-VC (Wang et al., 2018)	38.13	26.58	-	-
LSTM-GAN (Yang et al., 2018)	36	26.1	-	-
VGG-16(RGB) (Nabati and Behrad, 2020b)	34.5	25.8	57.2	36.1
GoogLeNet(RGB) (Nabati and Behrad, 2020b)	35.9	26.1	58	37.5
ResNet-152 (RGB) (Nabati and Behrad, 2020b)	36.6	27	58.7	40.5
GRU-EVEhft – (IRV2) (Aafaq et al., 2019)	32.9	26.4	57.2	39.2
GRU-EVEhft – (CI) (Aafaq et al., 2019)	36.1	27.7	**59.9**	**45.2**
DS-SCA [2D+3D+SEM] (Shekharet al. , 2020)	33.4	24.5	56.8	28.7
DS-SCA [VLAD+2D+3D+SEM] (Shekharet al. , 2020)	38.7	27.5	60.7	41.6
BDE-DS (Chen et al., 2019)	38.70	26.70	-	41.0
Proposed Method	**39.2**	**27.8**	59.3	40.5

Figure 4.17 is depicted with all the results comparison of four models and thier performance metrics. The study shows that Inception-v4 outperforms all the other models due to the fact that more inception modules associated and acheiveing greater accuracy.

Time Complexity Analysis

The proposed approach consists of a CNN feature extractor followed by a residual LSTM model. So, the total complexity of the system comprises the complexity of CNN model used and the complexity of training the residual LSTM model. This is approximately equal to $O(CNN\ model) + O(residual\ LSTM\ model)$. Thus the overall complexity can be defined as $O(N) + e * b * T_b * O(R_L)$. Where N, e, b, R_L, and T_b, denote the number of parameters in CNN feature extractor, epochs, batches, param-

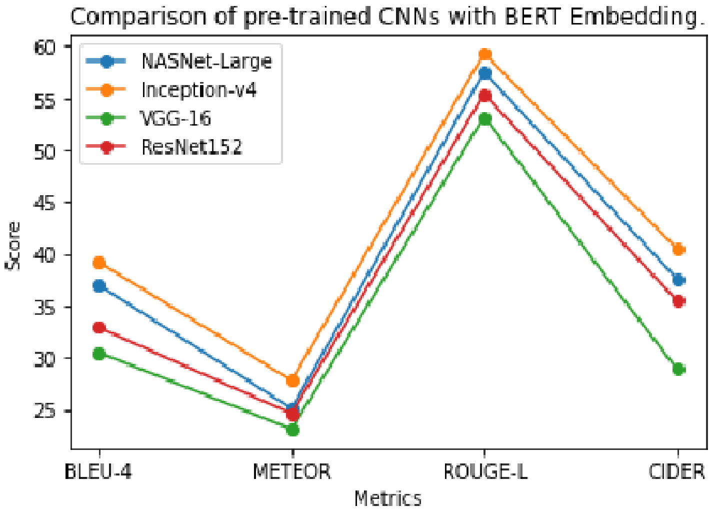


Figure 4.17: Line chart comparison of pre-trained CNNs with BERT Embedding

eters in residual LSTM model, and time estimation of backward and forward passes, respectively.

4.3.3 Summary

In this research work, we augumented a novel contextual hybrid embedding blocks and short-connections in the video captioning encoder. It is critical to realise that the contextual block, also called the semantic phrase vector, in the encoder is a representation of the video in the caption space. It assists the decoder in optimising the model's overall performance by supplying more information. We have validated our technique and analysis by conducting experiments on the MSVD and MSR-VTT datasets. The proposed framework compared to other standard existing methods. It is found that, our methodology produced significantly better outcomes.The suggested model is also being compared to other feature extraction techniques such as NASNet-Large, VGG-16, Inception-v4, and ResNet-152. Several word embedding approaches, such as BERT, ELMo, and GloVe, are used to assess the model's performance. On the basis of the experimental investigation, Inception-v4 feature extraction outperformed VGG-16, ResNet-152, and NASNet-Large in terms of performance. When it comes to word embedding strategies, BERT excels ELMo and GloVe by a wide margin.

Chapter 5

Conclusions and Future Work

Over the last few years, the discipline of CV has advanced at a breakneck pace. Rather than reducing chances for research in CV, progress on this central task has energised the field and unlocked a wide variety of difficult problems that had previously eluded our efforts but now appeared within the scope of research community. We proposed models and strategies in this research work that push the boundaries of visual identification by expanding the label space beyond a finite collection of categories to the space of natural language utterances.

Specifically, this thesis work focused on building approaches for pixel-level image understanding and on generating natural language descriptions of activities portrayed in distinct video datasets. Additionally, this thesis work introduces strategies for substantially extending study in semantic image segmentation and video captioning. Semantic image segmentation is a fundamental component of object recognition models, as it aims to classify things on a pixel-by-pixel basis. Chapter 3 examines the tasks involved in identifying an image at the pixel level as a different object. The proposed contributions are aimed at classifying distinct items inside an image at the pixel level. The input image is processed to extract pixel-level information, and the object in the image is then demarcated and segmented for identification purposes.

In this research work, inititally focused primarily on three approaches to study the object's semantic limit and accuracy. The initial objective was to analyse the learnt features from ConvNets and implement the RCA method for scene labelling. We observed that learnt features are even more effective and can improve the labelling accuracy when employing the RCA algorithm, with the maximum accuracy gain occurring between 0.4 and 0.6. Gains in precision range from 1 to 3 percent when the RCA algorithm is utilised as opposed to when it is not. This demonstrates clearly the significance of global pixel label consistency. In the second objective, the proposed strategy has been separated into three different stages. It utilises the selective benefits of MRF at low levels and the superior performance of CRP at high levels. BN is utilised to strengthen the robustness of the proposed method. Object detection in an image, attribute prediction and association, and semantic segmentation are the three distinct segments.

The first and most essential task is to detect the different objects. Comparing the SIFT feature detection algorithm to the SURF feature detection technique, we discovered that SURF performs better. Using the BN, the second objective is attained. The final objective of semantic segmentation and object-attribute association is attained by utilising MRF stacked CRP to create clusters. MRF begins clustering by constructing a set of local-level components. After the low level, the high-level CRP is used to integrate these components into larger clusters. On the basis of parameters such as PRI, F1-measure, and GCE, the performance of the proposed method is compared to that of other existing semantic segmentation methods. With a PRI score of 0.79, an F-measure of 0.71, and a GCE score of 0.23, the comparison suggests that the proposed method performs the best in all three parameters.

The final objective of Chapter 3 was to learn the appearance of an object using the image attribution and segmentation model. First, the image is superpixelized using SLIC to accomplish the task. A MSF model created the object seed heat map, while an SECT model annotates the pixel level. The iterative PSPNet model is used to learn object characteristics and perform semantic segmentation. Additionally, we implemented a CAM model to visualise the images. Class-level semantic segmentation was outperformed by our model relative to other current methods. The results from PSPNet show a mIoU accuracy of 52.4% on the PASCAL VOC 2012 test dataset, which is superior to the Baseline's 49.3% mIoU accuracy.

Techniques for automatically describing images/videos should be capable of detecting significant occurrences worth describing. They should be able to accurately describe a wide variety of visuals that contains a diversified array of events, objects, scenes, and other attributes. Chapter 4 explains the goal of video/image captioning, which is to portray the dynamics of a video clip in simple natural language by utilizing the most frequently used design paradigm and a pioneering structurally upgraded encoder-decoder architecture. We demonstrated our techniques' adaptability by assessing them against open-domain benchmark image and video datasets. Despite recent significant advancements in visual recognition, it is evident that many obstacles remain until machines can detect the visible environment and communicate with us using natural language.

The first proposed framework in Chapter 4 investigates spatial and temporal information throughout the entire sequence of video frames. At the decoding step, the system is implemented by merging multiple LSTM, different Feature Extractors, Soft Attention, hybrid loss functions, and a GloVe embedding mechanism. The visual encoder is an amalgamation of visual features derived from CNN and the layered LSTM. The decoder component is comprised of attention and a single LSTM layer. The framework performed around 24.5 percent better than S-VC, 9 percent better than LSTM-E, and 2 percent better than BPLSTM based on experimental results. In terms of METEOR and CIDEr, the proposed method outperformed SA by 9 percent and 36 percent, respectively.

The second research strategy employs an innovative Multi-Layer Attention-based strategy for video captioning. The framework makes use of two LSTM networks: one for the visual encoder and one for the language model. To preserve input at every time period for attention, the visual encoder is implemented utilising stacked BiLSTMs on preprocessed video data. The encoder's hidden states are then used to generate a global view of the video, which is incorporated into the language model. The decoder unit was used to transform the video captions into whole sentences. All of the framework's variations were tested using performance measures. As a result of semantic attention being paid at both the encoding and decoding stages in our implementations, all variations on both standard datasets surpassed all other findings. It has also been demonstrated that BiLSTM has a significant impact on the performance of the models in consideration.

In the third research study, the video captioning encoder complemented with a novel contextual hybrid embedding blocks and short connections. In the encoder, the contextual block, also known as the semantic phrase vector, represents the video in the caption space. It aids the decoder in optimising the overall performance of the model by providing more information. In comparison to other current standard methodologies, the proposed framework delivered much superior results. Other feature extraction techniques, such as NASNet-Large, VGG-16, Inception-v4, and ResNet-152, are compared to the proposed model. Several word embedding methods, including BERT, ELMo, and Glove, are utilised to evaluate the performance of the model. In terms of performance, Inception-v4 feature extraction outperformed VGG-16, ResNet-152, and NASNet-Large. BERT significantly outperforms ELMo and Glove in terms of word embedding methods.

Language and vision is a rapidly growing field of current research. This thesis discussed ways for overcoming some of the difficulties encountered in this fast growing domain. We hope that some of the concepts and insights discussed in this thesis will prove valuable in further endeavours. As the future work, the approaches can be explored as

Describing video events in context

Almost all effort in video description has focused on developing descriptions for autonomous events. Techniques for creating a brief description of an event were developed using short video captions. Even with longer video snippets, the emphasis was primarily on describing a significant occurrence. While this approach to captioning may be appropriate for some purposes, such as providing descriptions for the visually handicapped, detecting and articulating many events in videos may also enable more effective ways for video retrieval and may be video question answering. The difficulty here is in identifying several events and producing consistent descriptions within the context. The proposed video captioning models can be fine-tuned to give more efficient context based solutions to describe the longer videos. The approaches may also be useful for captioning the real time videos.

Generating textual summaries of long videos

From a description standpoint, an interesting option would be to provide textual summaries of lengthy videos rather than using short videos. The approaches proposed in the captioning models may be useful after little modifications.

Enhancing Movie Description

We may go beyond simple sentence descriptions of movies by using our successful methods for character identification that result in more specific descriptions, including character names and behaviours associated with characters.

Joint localization and description

One option to overcome our model's shortcomings is to employ a multi-task network in which the visual representation is trained using both temporal segmentation and description loss. Another approach to this challenge could be to present a single end-to-end model capable of concurrently identifying and describing regions in the image.

Paragraph descriptions for sequences in the video

Another approach for describing events in lengthy videos is to concentrate on the language model in order to provide coherent descriptions of many events.

Domain specific dataset applications

The proposed methods can also be applied to domain-specific datasets, such as movies and documentaries, and tested with techniques such as beam search, which determines the ideal word combination for a caption.

Specific network models

Various feature extraction strategies, such as I3D. (Two-Stream Inflated 3D ConvNets), GANs, and using a three-dimensional neural network in conjunction with two-dimensional CNNs, may have been analysed to compare their outcomes.

Edge based analysis on images

The approaches proposed might have also been implemented using 3D images and edge detection algorithms to enhance the perceived efficacy. In the future, the proposed segmentation techniques will be employed to improve the outcomes of object-level semantic segmentation and refined to find sharper edges in segmented images.

References

Aafaq, N., Akhtar, N., Liu, W., Gilani, S. Z., and Mian, A. (2019). “Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning”. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12487–12496.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2010). “Slic superpixels”. Technical report.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). “SLIC superpixels compared to state-of-the-art superpixel methods”. *IEEE transactions on pattern analysis and machine intelligence*, *34*(11), 2274–2282.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., and Farhan, L. (2021). “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions”. *Journal of Big Data*, *8*(1), 1–74.

Amirian, S., Rasheed, K., Taha, T. R., and Arabnia, H. R. (2020). “Automatic image and video caption generation with deep learning: A concise review and algorithmic overlap”. *IEEE Access*, *8*, 218386–218400.

Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2016). “Spice: Semantic propositional image caption evaluation”. In *European conference on computer vision*, 382–398. Springer.

Aneja, J., Deshpande, A., and Schwing, A. G. (2018). “Convolutional image captioning”. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5561–5570.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). “Contour Detection and Hierarchical Image Segmentation”. *IEEE Trans. Pattern Anal. Mach. Intell.*, *33*(5), 898–916.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481–2495.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). “Neural machine translation by jointly learning to align and translate”. *arXiv preprint arXiv:1409.0473*.

Ban, Z., Liu, J., and Cao, L. (2018). “Superpixel segmentation using Gaussian mixture model”. *IEEE Transactions on Image Processing*, *27*(8), 4105–4117.

Banerjee, S. and Lavie, A. (2005). “METEOR: An automatic metric for MT Evaluation with Improved Correlation with Human Judgments”. *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 65–72.

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

Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S., Fidler, S., Michaux, A., Mussman, S., Narayanaswamy, S., Salvi, D., et al. (2012). “Video in sentences out”. *arXiv preprint arXiv:1204.2742*.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). “SURF: Speeded Up Robust Features”. In Leonardis, A., Bischof, H., and Pinz, A. (Eds.), *Computer Vision – ECCV 2006*, 404–417. Springer Berlin Heidelberg.

Bergsma, S. and Goebel, R. (2011). “Using visual information to predict lexical preference”. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, 399–405.

Bin, Y., Yang, Y., Shen, F., Xie, N., Shen, H. T., and Li, X. (2019). “Describing Video With Attention-Based Bidirectional LSTM”. *IEEE Transactions on Cybernetics*, *49*(7), 2631–2641.

Blei, D. M. and Frazier, P. I. (2010). “Distance dependent Chinese restaurant processes”. In *ICML*.

Blei, D. M. and Frazier, P. I. (2011). “Distance dependent chinese restaurant processes.”. *Journal of Machine Learning Research*, *12*(8).

Brostow, G. J., Fauqueur, J., and Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, *30*(2), 88–97.

Brostow, G. J., Shotton, J., Fauqueur, J., and Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds, 44–57.

Cao, P., Yang, Z., Sun, L., Liang, Y., Yang, M. Q., and Guan, R. (2019). “Image captioning with bidirectional semantic attention-based guiding of long short-term memory”. *Neural Processing Letters*, *50*(1), 103–119.

Cao, X., Zhou, F., Xu, L., Meng, D., Xu, Z., and Paisley, J. (2018). “Hyperspectral image classification with Markov random fields and a convolutional neural network”. *IEEE Transactions on image processing*, *27*(5), 2354–2367.

Ceci, L. (2021), “Hours of video uploaded to YouTube every minute 2007-2020 ”. `https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/`. [Online; accessed 15-FEB-2022].

Chen, D. and Dolan, W. (2011). “Collecting Highly Parallel Data for Paraphrase Evaluation”. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 190–200., Portland, Oregon, USA. Association for Computational Linguistics.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2014). “Semantic image segmentation with deep convolutional nets and fully connected crfs”. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), 834–848.

Chen, T., Zhao, Q., and Song, J. (2019). “Boundary Detector Encoder and Decoder with Soft Attention for Video Captioning”. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, 105–115. Springer.

Chen, X., Kundu, K., Zhu, Y., Ma, H., Fidler, S., and Urtasun, R. (2017). “3d object proposals using stereo imagery for accurate object class detection”. *IEEE transactions on pattern analysis and machine intelligence*, *40*(5), 1259–1272.

Dai, J., He, K., and Sun, J. (2015). “Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation”. In *Proceedings of the IEEE international conference on computer vision*, 1635–1643.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). “Imagenet: A large-scale hierarchical image database”. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). “BERT: Pre-training of deep bidirectional transformers for language understanding”. *CoRR*, *abs/1810.04805*.

Drews, P., de Bem, R., and de Melo, A. (2011). Analyzing and exploring feature detectors in images. In *2011 9th IEEE International Conference on Industrial Informatics*, 305–310. IEEE.

Eshaghzadeh, A. and Salehyan, N. (2016). “Canny edge detection algorithm application for analysis of the potential field map”. *Geophysics*, *62*, 807–813.

Everingham, M., Eslami, S., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). “The pascal visual object classes challenge: A retrospective”. *International journal of computer vision*, *111*(1), 98–136.

Everingham, M., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. (2012). “The pascal visual object classes challenge 2012 (voc2012)”. In *Results*.

Feinerer, I. and Hornik, K. (2020). “wordnet: Wordnet interface”. R package version 0.1-15.

Feng, Y. and Wang, L. (2019). “A Weakly-Supervised Approach for Semantic Segmentation”. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2311–2314. IEEE.

Gao, L., Guo, Z., Zhang, H., Xu, X., and Shen, H. T. (2017). “Video Captioning With Attention-Based LSTM and Semantic Consistency”. *IEEE Transactions on Multimedia*, *19*(9), 2045–2055.

Gao, L., Li, X., Song, J., and Shen, H. T. (2020). “Hierarchical LSTMs with Adaptive Attention for Visual Captioning”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(5), 1112–1131.

Girshick, R. (2015). “Fast r-cnn”. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2015). “Region-based convolutional networks for accurate object detection and segmentation”. *IEEE transactions on pattern analysis and machine intelligence*, *38*(1), 142–158.

Goldberg, Y. (2016). “A primer on neural network models for natural language processing”. *Journal of Artificial Intelligence Research*, *57*, 345–420.

Griffiths, T. L. and Ghahramani, Z. (2011). “The Indian Buffet Process: An Introduction and Review”. *J. Mach. Learn. Res.*, *12*, 1185–1224.

Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., and Saenko, K. (2013). “Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition”. In *Proceedings of the IEEE international conference on computer vision*, 2712–2719.

Guimarães, S. J. F., Cousty, J., Kenmochi, Y., and Najman, L. (2012). “A Hierarchical Image Segmentation Algorithm Based on an Observation Scale”. In Gimel’farb, G., Hancock, E., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Windeatt, T., and Yamada, K. (Eds.), *Structural, Syntactic, and Statistical Pattern Recognition*, 116–125., Berlin, Heidelberg. Springer Berlin Heidelberg.

Hanea, A., Napoles, O. M., and Ababei, D. (2015). “Non-parametric Bayesian networks: Improving theory and reviewing applications”. *Reliability Engineering and System Safety*, *144*, 265–284.

Hao, X., Zhou, F., and Li, X. (2020). “Scene-Edge GRU for Video Caption”. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 1, 1290–1295.

Harris, Z. S. (1954). “Distributional structure”. *Word*, *10*(2-3), 146–162.

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

Hochreiter, S. and Schmidhuber, J. (1997). “Long Short-Term Memory”. *Neural Computation*, *9*(8), 1735–1780.

Hossain, M. Z., Sohel, F., Shiratuddin, M. F., and Laga, H. (2019). “A comprehensive survey of deep learning for image captioning”. *ACM Computing Surveys (CsUR)*, *51*(6), 1–36.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). “Bag of Tricks for Efficient Text Classification”. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 427–431. Association for Computational Linguistics.

Justus, D., Brennan, J., Bonner, S., and McGough, A. S. (2018). Predicting the computational cost of deep learning models. In *2018 IEEE international conference on big data (Big Data)*, 3873–3882. IEEE.

Kang, B. and Nguyen, T. Q. (2019). “Random forest with learned representations for semantic segmentation”. *IEEE Transactions on Image Processing*, *28*(7), 3542–3555.

Karpathy, A. and Fei-Fei, L. (2015). “Deep visual-semantic alignments for generating image descriptions”. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3128–3137.

Kato, Z. and Pong, T.-C. (2006). “A Markov random field image segmentation model for color textured images”. *Image and Vision Computing*, *24*(10), 1103–1114.

Khoreva, A., Benenson, R., Hosang, J., Hein, M., and Schiele, B. (2017). “Simple does it: Weakly supervised instance and semantic segmentation”. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 876–885.

Kirillov, A., He, K., Girshick, R., Rother, C., and Dollár, P. (2019). “Panoptic segmentation”. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9404–9413.

Kiros, R., Salakhutdinov, R., and Zemel, R. S. (2014). “Unifying visual-semantic embeddings with multimodal neural language models”. *arXiv preprint arXiv:1411.2539*.

Krishna, R., Hata, K., Ren, F., Fei-Fei, L., and Carlos Niebles, J. (2017). “Dense-captioning events in videos”. *Proceedings of the IEEE international conference on computer vision*, 706–715.

Krishnamoorthy, N., Malkarnenkar, G., Mooney, R., Saenko, K., and Guadarrama, S. (2013). “Generating natural-language video descriptions using text-mined knowledge”. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). “Imagenet classification with deep convolutional neural networks”. *Advances in neural information processing systems*, *25*.

Lai, S., Xu, L., Liu, K., and Zhao, J. (2015). “Recurrent convolutional neural networks for text classification”. In *Twenty-ninth AAAI conference on artificial intelligence*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). "Gradient-based learning applied to document recognition". *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, G., Ma, S., and Han, Y. (2015). "Summarization-Based Video Caption via Deep Neural Networks". In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, 1191–1194., New York, NY, USA. Association for Computing Machinery.

Li, H., Xiong, P., Fan, H., and Sun, J. (2019). "Dfanet: Deep feature aggregation for real-time semantic segmentation". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9522–9531.

Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., and Chang, K.-W. (2019). "Visualbert: A simple and performant baseline for vision and language". *arXiv preprint arXiv:1908.03557*.

Li, S., Tao, Z., Li, K., and Fu, Y. (2019). "Visual to Text: Survey of Image and Video Captioning". *IEEE Transactions on Emerging Topics in Computational Intelligence*, *3*(4), 297–312.

Li, X., Ma, H., and Luo, X. (2019). "Weaklier supervised semantic segmentation with only one image level annotation per category". *IEEE Transactions on Image Processing*, *29*, 128–141.

Li, X., Ma, H., Wang, X., and Zhang, X. (2017). "Traffic light recognition for complex scene with fusion detections". *IEEE Transactions on Intelligent Transportation Systems*, *19*(1), 199–208.

Li, Y., Sohel, F., Bennamoun, M., and Lei, H. (2015). "Outdoor scene labelling with learned features and region consistency activation". *2015 IEEE International Conference on Image Processing (ICIP)*, 1374–1378.

Lin, C.-Y. (2004). "ROUGE: A Package for Automatic Evaluation of Summaries". *Text Summarization Branches Out*, 74–81.

Lin, J.-C. and Zhang, C.-Y. (2021). "A New Memory Based on Sequence to Sequence Model for Video Captioning". In *2021 International Conference on Security, Pattern Analysis, and CyberneticsSPAC)*, 470–476.

Lin, M., Chen, Q., and Yan, S. (2013). "Network in network". *arXiv preprint arXiv:1312.4400*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). "Microsoft coco: Common objects in context". In *European conference on computer vision*, 740–755. Springer.

Liu, C., Chen, L.-C., Schroff, F., Adam, H., Hua, W., Yuille, A. L., and Fei-Fei, L. (2019). "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 82–92.

Liu, S., Ren, Z., and Yuan, J. (2021). "SibNet: Sibling Convolutional Encoder for Video Captioning". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(9), 3259–3272.

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Lowe, D. G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints". *60*, 91–110.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). "Distributed representations of words and phrases and their compositionality". *Advances in neural information processing systems*, *26*.

Mitchell, J. and Lapata, M. (2008). "Vector-based models of semantic composition". In *proceedings of ACL-08: HLT*, 236–244.

Mokrzycki, W. and Tatol, M. (2009). "Perceptual difference in l* a* b* color space as the base for object colour identfication". *Image Processing and Communication Challenges*, 403–412.

Nabati, M. and Behrad, A. (2020a). "Multi-sentence video captioning using content-oriented beam searching and multi-stage refining algorithm". *Information Processing and Management*, *57*(6), 102302.

Nabati, M. and Behrad, A. (2020b). "Video captioning using boosted and parallel Long Short-Term Memory networks". *Computer Vision and Image Understanding*, *190*, 102840.

Nian, F., Li, T., Wang, Y., Wu, X., Ni, B., and Xu, C. (2017). "Learning explicit video attributes from mid-level representation for video captioning". *Computer Vision and Image Understanding*, *163*, 126 – 138. Language in Vision.

Olivastri, S., Singh, G., and Cuzzolin, F. (2019). "End-to-End Video Captioning". *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 1474–1482.

Ordóñez, , Argüello, F., and Heras, D. B. (2018). "Alignment of Hyperspectral Images Using KAZE Features". *Remote Sensing*, *10*(5).

Pan, P., Xu, Z., Yang, Y., Wu, F., and Zhuang, Y. (2016). "Hierarchical Recurrent Neural Encoder for Video Representation with Application to Captioning". In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1029–1038.

Pan, Y., Mei, T., Yao, T., Li, H., and Rui, Y. (2016). "Jointly modeling embedding and translation to bridge video and language". In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4594–4602.

Papandreou, G., Chen, L.-C., Murphy, K. P., and Yuille, A. L. (2015). “Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation”. In *Proceedings of the IEEE international conference on computer vision*, 1742–1750.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). “Bleu: a Method for Automatic Evaluation of Machine Translation”. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318.

Pei, W., Zhang, J., Wang, X., Ke, L., Shen, X., and Tai, Y. (2019). “Memory-Attended Recurrent Network for Video Captioning”. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8339–8348.

Pennington, J., Socher, R., and Manning, C. D. (2014). “GloVe: Global Vectors for Word Representation”. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). “Deep contextualized word representations”. *CoRR*, *abs/1802.05365*.

Pinheiro, P. O. and Collobert, R. (2015). “From image-level to pixel-level labeling with convolutional networks”. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1713–1721.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). “Faster r-cnn: Towards real-time object detection with region proposal networks”. *Advances in neural information processing systems*, *28*.

Ren, W., Huang, K., Tao, D., and Tan, T. (2015). “Weakly supervised large scale object localization with multiple instance learning and bag splitting”. *IEEE transactions on pattern analysis and machine intelligence*, *38*(2), 405–416.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). “Imagenet large scale visual recognition challenge”. *International journal of computer vision*, *115*(3), 211–252.

Sah, S., Nguyen, T., and Ptucha, R. (2020). “Understanding temporal structure for video captioning”. *Pattern Analysis and Applications*, *23*(1), 147–159.

Salman, A. G., Heryadi, Y., Abdurahman, E., and Suparta, W. (2018). “Single layer and multi-layer long short-term memory (LSTM) model with intermediate variables for weather forecasting”. *Procedia Computer Science*, *135*, 89–98.

Schuster, M. and Paliwal, K. K. (1997). “Bidirectional recurrent neural networks”. *IEEE transactions on Signal Processing*, *45*(11), 2673–2681.

Schwarz, C. G., Fletcher, E., Singh, B., Liu, A., Smith, N., DeCarli, C., and Carmichael, O. (2012). Most edges in markov random fields for white matter hyperintensity segmentation are worthless. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2684–2687. IEEE.

Sezgin, M. and Sankur, B. (2004). “Survey over image thresholding techniques and quantitative performance evaluation”. *Journal of Electronic imaging*, *13*(1), 146–165.

Sha, L., Chang, B., Sui, Z., and Li, S. (2016). “Reading and Thinking: Re-read LSTM Unit for Textual Entailment Recognition”. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2870–2879., Osaka, Japan. The COLING 2016 Organizing Committee.

Shekhar, C. C. et al. (2020). “Domain-specific semantics guided approach to video captioning”. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1587–1596.

Shelhamer, E., Long, J., and Darrell, T. (2016). “Fully convolutional networks for semantic segmentation”. *IEEE transactions on pattern analysis and machine intelligence*, *39*(4), 640–651.

Shen, T., Zhou, T., Long, G., Jiang, J., Wang, S., and Zhang, C. (2018). “Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling”. *arXiv preprint arXiv:1801.10296*.

Shorten, C., Khoshgoftaar, T. M., and Furht, B. (2021). “Text data augmentation for deep learning”. *Journal of big Data*, *8*(1), 1–34.

Simonyan, K. and Zisserman, A. (2015). “Very deep convolutional networks for large-scale image recognition”. *The 3rd International Conference on Learning Representations (ICLR2015)*.

Song, J., Guo, Y., Gao, L., Li, X., Hanjalic, A., and Shen, H. T. (2019). “From Deterministic to Generative: Multimodal Stochastic RNNs for Video Captioning”. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(10), 3047–3058.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). “Dropout: a simple way to prevent neural networks from overfitting”. *The journal of machine learning research*, *15*(1), 1929–1958.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). “Sequence to sequence learning with neural networks”. In *Advances in neural information processing systems*, 3104–3112.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). “Inception-v4, inception-resnet and the impact of residual connections on learning”. *Thirty-first AAAI conference on artificial intelligence*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). “Rethinking the Inception Architecture for Computer Vision”. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826.

Tagaris, T., Sdraka, M., and Stafylopatis, A. (2019). “High-resolution class activation mapping”. In *2019 IEEE International Conference on Image Processing (ICIP)*, 4514–4518. IEEE.

Tan, H. and Bansal, M. (2019). “lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490.*

Tu, Y., Zhang, X., Liu, B., and Yan, C. (2017). “Video Description with Spatial-Temporal Attention”. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM ’17, 1014–1022., New York, NY, USA. Association for Computing Machinery.

Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., and Baik, S. W. (2017). “Action recognition in video sequences using deep bi-directional LSTM with CNN features”. *IEEE access*, *6*, 1155–1166.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017a). “Attention is all you need”. *Advances in neural information processing systems*, *30.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017b). “Attention Is All You Need”. *CoRR*, *abs/1706.03762.*

Vedantam, R., Zitnick, C. L., and Parikh, D. (2014). “CIDEr: Consensus-based image description evaluation”. *CoRR*, *abs/1411.5726.*

Venmathi, A., Ganesh, E., and Kumaratharan, N. (2019). “Image segmentation based on markov random field probabilistic approach”. In *2019 international conference on communication and signal processing (ICCSP)*, 0490–0495. IEEE.

Venugopalan, S., Hendricks, L. A., Mooney, R., and Saenko, K. (2016). “Improving lstm-based video description with linguistic knowledge mined from text”. *arXiv preprint arXiv:1604.01729.*

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. (2015). “Sequence to sequence – video to text”. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 4534–4542.

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

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2014). “Show and Tell: A Neural Image Caption Generator”. *CoRR*, *abs/1411.4555.*

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). “Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned”. *arXiv preprint arXiv:1905.09418.*

Wang, J., Wang, W., Huang, Y., Wang, L., and Tan, T. (2018). “M3: Multimodal memory modelling for video captioning”. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7512–7520.

Wang, X. and Zhao, J. (2017). “Hierarchical non-parametric Markov random field for image segmentation”. *IET Computer Vision*, *11*(8), 717–724.

Wei, R., Mi, L., Hu, Y., and Chen, Z. (2020). “Exploiting the local temporal information for video captioning”. *Journal of Visual Communication and Image Representation*, *67*, 102751.

Winston, P. H. (1992). “Artificial intelligence”. MIT: Addison-Wesley Longman Publishing Co., Inc.

Wu, H., Zhang, J., Huang, K., Liang, K., and Yu, Y. (2019). “Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation”. *arXiv preprint arXiv:1903.11816*.

Xiang, W., Mao, H., and Athitsos, V. (2019). “ThunderNet: A turbo unified network for real-time semantic segmentation”. In *2019 IEEE winter conference on applications of computer vision (WACV)*, 1789–1796. IEEE.

Xiao, D. and Zhong, P. (2019). “Image semantic segmentation using deep convolutional nets, fully connected conditional random fields, and dilated convolution”. In *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 1872–1877. IEEE.

Xu, H., Venugopalan, S., Ramanishka, V., Rohrbach, M., and Saenko, K. (2015). “A Multi-scale Multiple Instance Video Description Network”. *CoRR*, *abs/1505.05914*.

Xu, J., Mei, T., Yao, T., and Rui, Y. (2016). “Msr-vtt: A large video description dataset for bridging video and language”. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5288–5296.

Xu, J., Wei, H., Li, L., Fu, Q., and Guo, J. (2020). “Video description model based on temporal-spatial and channel multi-attention mechanisms”. *Applied Sciences*, *10*(12), 4312.

Xu, J., Yao, T., Zhang, Y., and Mei, T. (2017). “Learning Multimodal Attention LSTM Networks for Video Captioning”. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM ’17, 537–545., New York, NY, USA. Association for Computing Machinery.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). “Show, attend and tell: Neural image caption generation with visual attention”. In *International conference on machine learning*, 2048–2057. PMLR.

Xu, N., Liu, A., Nie, W., and Su, Y. (2018). “Attention-in-Attention Networks for Surveillance Video Understanding in Internet of Things”. *IEEE Internet of Things Journal*, *5*(5), 3419–3429.

Xu, R., Xiong, C., Chen, W., and Corso, J. (2015). "Jointly modeling deep video and compositional text to bridge vision and language in a unified framework". In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Yadav, N. and Naik, D. (2021). "Generating Short Video Description using Deep-LSTM and Attention Mechanism". In *2021 6th International Conference for Convergence in Technology (I2CT)*, 1–6. IEEE.

Yan, C., Tu, Y., Wang, X., Zhang, Y., Hao, X., Zhang, Y., and Dai, Q. (2020). "STAT: Spatial-Temporal Attention Mechanism for Video Captioning". *IEEE Transactions on Multimedia*, *22*(1), 229–241.

Yang, Y., Zhou, J., Ai, J., Bin, Y., Hanjalic, A., Shen, H. T., and Ji, Y. (2018). "Video Captioning by Adversarial LSTM". *IEEE Transactions on Image Processing*, *27*(11), 5600–5611.

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., and Courville, A. (2015). "Describing videos by exploiting temporal structure". In *2015 IEEE International Conference on Computer Vision (ICCV)*, 4507–4515.

Yi, Y., Zhang, Z., Zhang, W., Zhang, C., Li, W., and Zhao, T. (2019). Semantic segmentation of urban buildings from vhr remote sensing imagery using a deep convolutional neural network. *Remote sensing*, *11*(15), 1774.

You, Q., Jin, H., Wang, Z., Fang, C., and Luo, J. (2016). "Image Captioning with Semantic Attention". *CoRR*, *abs/1603.03925*.

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions". *TACL*, *2*, 67–78.

Yu, H. and Siskind, J. M. (2013). "Grounded language learning from video described with sentences". In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 53–63.

Yu, H., Wang, J., Huang, Z., Yang, Y., and Xu, W. (2016). "Video Paragraph Captioning Using Hierarchical Recurrent Neural Networks". In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4584–4593.

Zeiler, M. D. (2012). "ADADELTA: An Adaptive Learning Rate Method". *CoRR*, *abs/1212.5701*.

Zhang, Z., Xu, D., Ouyang, W., and Zhou, L. (2021). "Dense Video Captioning Using Graph-Based Sentence Summarization". *IEEE Transactions on Multimedia*, *23*, 1799–1810.

Zhao, B., Li, X., and Lu, X. (2019). "CAM-RNN: Co-Attention model based RNN for video captioning". *IEEE Transactions on Image Processing*, *28*(11), 5552–5565.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). “Pyramid scene parsing network”. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

Zhao, Z. and Wang, X. (2018). “Multi-segments Naïve Bayes classifier in likelihood space”. *IET Computer Vision*, *12*(6), 882–891.

Zheng, Q., Wang, C., and Tao, D. (2020). “Syntax-aware action targeting for video captioning”. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). “Learning deep features for discriminative localization”. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.

Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., and Jiao, J. (2018). “Weakly supervised instance segmentation using class peak response”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3791–3800.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). “Learning transferable architectures for scalable image recognition”. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8697–8710.

Publications

Journal Papers

1. **Dinesh Naik** and C. D. Jaidhar, (2022). “Semantic context driven language descriptions of videos using deep neural network”. *Journal of Big Data*, 9, 17 (Springer) (SCOPUS, SCIE Indexed)
DOI: https://doi.org/10.1186/s40537-022-00569-4

2. **Dinesh Naik** and C. D. Jaidhar, (2022), “A Novel Multi-Layer Attention Framework for Visual Description Prediction Using Bidirectional LSTM”. *Journal of Big Data*, 9, 104 (Springer) (SCOPUS, SCIE Indexed)
DOI: https://doi.org/10.1186/s40537-022-00664-6.

3. **Dinesh Naik** and Jaidhar C.D., “Video Captioning using a Sentence Vector-enabled Convolutional Framework with a Short-Connected LSTM”. *Multimedia Tools and Applications*-Under-Review. (SCOPUS, SCIE Indexed)

Conference Papers

1. **Dinesh Naik** and C. D. Jaidhar, (2016), “Image segmentation using encoder-decoder architecture and region consistency activation”. 11th *International Conference on Industrial and Information Systems (ICIIS)*, pp. 724-729, (SCOPUS Indexed)
DOI: 10.1109/ICIINFS.2016.8263033.

2. **Dinesh Naik** and C. D. Jaidhar, (2021), “Weaklier-Supervised Semantic Segmentation with Pyramid Scene Parsing Network”. 2nd *International Conference on Secure Cyber Computing and Communications (ICSCCC)*, pp. 288-295 (SCOPUS Indexed)
DOI: 10.1109/ICSCCC51823.2021.9478107.

3. **Dinesh Naik** and C. D. Jaidhar, (2021), “Weakly Supervised Image Annotation and Segmentation”. 12th *International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1-8 (SCOPUS Indexed)
DOI: 10.1109/ICCCNT51525.2021.9579713.

Curriculum Vitae

Dinesh Naik
Part-Time Ph.D Scholar
Department of Information Technology
National Institute of Technology Karnataka Surathkal
Srinivasanagar P.O, 575025
Karnataka State.

Permanent Address

Dinesh Naik
D.No. 16-66/3(10),
Udaynagar
NITK Surathkal Srinivasnagar Post-575025
Dakshina Kannada
Karnataka State
Email: din_nk@nitk.edu.in
Mobile: +919480401300.

Academic Records

1. M.Tech. in Computer Science and Engineering from NMAMIT NITTE, Karnataka State, 2006.
2. B.E. in Computer Science and Engineering from UBDTCE Davangere, Karnataka State, 2001.

Research Interests

Computer Vision
Digital Image Processing

Programming Languages

C, CPP, Python, MATLAB, Scripts.