Summarizing videos with attention based network

A PROJECT REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
Artificial Intelligence

Submitted by

PRATHMESH SHINDE
(23/AFI/17)

Under the supervision of
Prof. Manoj Kumar

COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

MAY, 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to Prof. Manoj Kumar for his continuous guidance and mentorship that he provided me during the project. He showed me the path to achieve my targets by explaining all the tasks to be done and explained to me the importance of this project as well as its industrial relevance. He was always ready to help me and clear my doubts regarding any hurdles in this project. Without his constant support and motivation, this project would not have been successful.

Place: Delhi Prathmesh Shinde

Date:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, **Prathmesh Shinde**, Roll No – **23/AFI/17** student of M.Tech Artificial Intelligence, hereby declare that the project Dissertation titled **"Summarizing videos with attention based network"** which is submitted by me to the Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi Prathmesh Shinde

Date:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled **"Summarizing videos with attention based network"** which is submitted by **Prathmesh Shinde**, Roll No's – **23/AFI/17**, **Artificial Intelligence**, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Prof. Manoj Kumar

Date:
SUPERVISOR

Abstract

In this project, I explored a new and more efficient way to summarize videos by focusing on the most important moments—what we call keyshots. Instead of relying on the usual complex models like bi-directional LSTMs with attention, which are not only difficult to implement but also require a lot of computational resources, I took a different route. I designed a simpler model based on a soft self-attention mechanism that's much easier to work with and faster to train.

What makes this approach stand out is that it processes the entire video sequence in just one forward and one backward pass during training. That means it's not only lightweight but also well-suited for real-world applications where speed and efficiency matter. The self-attention mechanism allows the model to understand the importance of each frame in the context of the whole video—without needing any complex recurrence.

I tested this method on two popular video summarization datasets, TvSum and SumMe, and was excited to see that it outperformed many of the existing state-of-the-art techniques. This showed me that a simpler, more streamlined approach can still deliver powerful results. It was a rewarding experience to challenge the norm and come up with a solution that's both practical and effective.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Introduction to video summarization

The proliferation of personal videos, educational lectures, video diaries, social media messages, and other video content has led to video becoming the dominant medium for information exchange. According to the Cisco Visual Networking Index: Forecast and Methodology, 2016–2021, video was projected to constitute approximately 80

Video summarization refers to the process of condensing a video sequence into a more compact form while preserving its essential informational content. This can be achieved either through the extraction of representative still frames, commonly referred to as keyframes, or by generating a shorter video sequence composed of selected segments, known as keyshots or dynamic summaries. This process is conceptually analogous to lossy video compression, where individual frames serve as the basic units of reduction. In the present work, I concentrate specifically on keyshot-based video summarization. Video summarization presents an inherently complex challenge, even for human observers. To determine the most significant segments within a video, one must view the entire content and subsequently select portions based on the desired summary length. Ideally, keyshots should comprise segments that are both highly representative of the source video and mutually diverse in content. Several existing methods approach this problem by formulating it as a clustering task, using cost functions that emphasize representativeness and diversity. However, accurately defining the degree to which selected keyshots represent the original content and differ from one another is particularly challenging, as it must align with the viewer's perception of informational relevance. Conventional techniques attempt to approximate this by analyzing motion features, computing distances between color histograms,

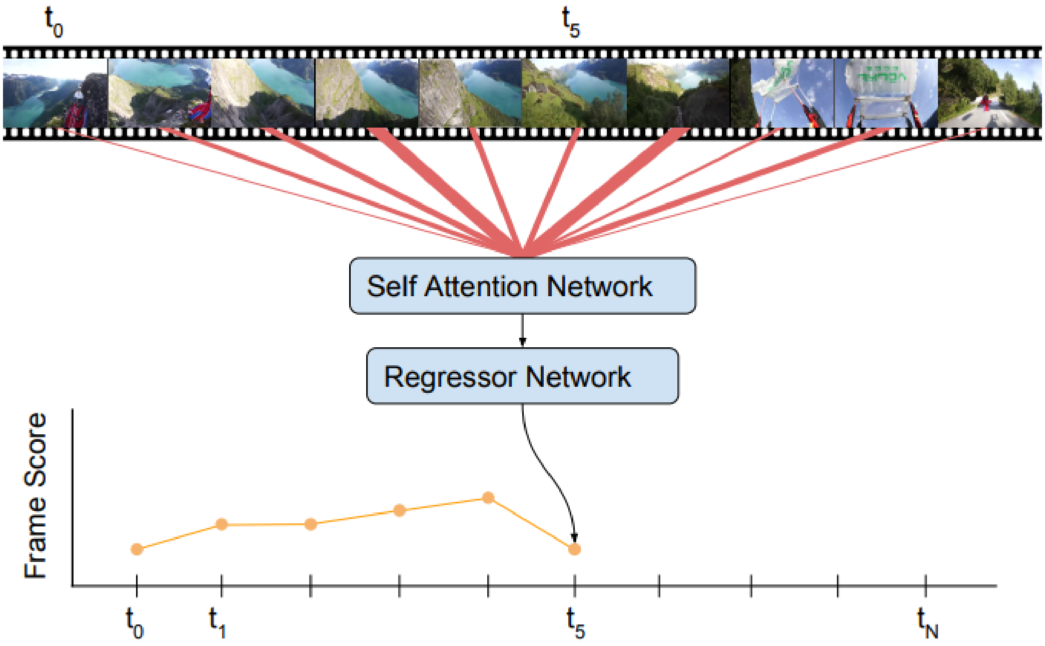


Figure 1.1: Each output from self-attention is influenced by a unique weight on every input feature. The weights allow us to calculate a weighted average of the input data which we then send to a neural network to measure how important each frame is.

measuring image entropy, or utilizing features derived from 2D/3D convolutional neural networks (CNNs) to assess semantic similarity. Despite their sophistication, these approaches fall short of fully capturing the contextual information embedded within video sequences.

To achieve automated summarization that approaches the quality of human-generated summaries, it is imperative that machines learn from human behavior. This can be effectively accomplished through behavioral cloning or supervised learning methodologies, allowing models to mimic human decision-making in the summarization process. Early approaches to video summarization predominantly relied on unsupervised methods, utilizing low-level spatio-temporal features in conjunction with dimensionality reduction and clustering techniques. The effectiveness of these methods hinges on the ability to define appropriate distance or cost functions that measure the similarity between candidate keyshots or frames and the original video content. However, as previously discussed, accurately capturing such relationships is a highly non-trivial task. Moreover, these methods inherently introduce bias based on the nature of the features employed—whether semantic, structural, or pixel-level—thereby limiting their generalizability and performance.

In contrast, supervised learning-based models offer a more robust alternative by learning a mapping that generates summaries resembling those produced by human annotators. This approach bypasses the need to manually define heuristic distance functions and instead leverages annotated data to guide the learning process. Currently, two publicly available datasets—TvSum and SumMe—provide such annotated video summaries, each labeled by approximately 15 to 20 users. Notably, these annotations exhibit significant variability, with inter-annotator consistency reflected in a pairwise F-score of approximately 0.34. This low agreement highlights the inherently subjective nature of video summarization.

Given this subjectivity, designing a universal metric for clustering video frames into keyshots in a way that aligns with human judgment is particularly challenging. On this basis, I adopt a supervised learning approach in our work, as it offers a more viable path toward generating high-quality, human-like video summaries. Contemporary state-of-the-art approaches to video summarization predominantly utilize recurrent encoder-decoder architectures, typically employing bidirectional Long Short-Term Memory (LSTM) networks [14] or Gated Recurrent Units (GRUs) [6], often enhanced with soft attention mechanisms [4]. While these models have demonstrated considerable success in various sequence modeling tasks—such as machine translation and image/video captioning—they are computationally intensive, particularly in their bidirectional configurations.

There was a recent innovation by Vaswani et al. [34], switching sequence-to-sequence modeling to an attention-only approach, with no reliance on recurrent structures. On the basis of this idea, I propose VASNet, a model using only attention and designed for creating video summaries using keyshot data. I evaluate the results of VASNet on TvSum and SumMe datasets.

Because of its approach, VASNet can train and infer data without depending on a sequential order. Instead, it uses basic matrix and vector methods which makes it possible to process sequences with any length using just a forward and backward pass. This framework computes the weighted average of input features with weights generated using a self-attention method to estimate a frame's importance. An illustration of how the model works is shown in Fig. 1.

I believe that VASNet's flexible design makes it suitable for other tasks that involve sequences changing into sequences.

The key contributions of this work are as follows:

1. I introduce a novel approach to sequence-to-sequence transformation for video summarization, based solely on a soft self-attention mechanism. In

contrast to existing methods, our model eliminates the need for complex LSTM/GRU-based encoder-decoder architectures.

2. I empirically demonstrate that recurrent neural networks can be effectively replaced by a simpler attention-based mechanism for the task of video summarization, without compromising performance.

1.2 Related Work

People working on video summarization have quickly chosen to use new deep learning techniques, paying special attention to encoder-decoder structures coupled with attention mechanisms. This chapter highlights several important methods related to what I am doing.

Team Zhang et al. were the first to introduce LSTM networks for video summarization. Their method effectively summarizes videos by paying attention to all kinds of time differences in each frame. In addition, they use the determinantal point process which naturally promotes selecting diverse subsets, to make the summarization results better.

With this in mind, Ji et al. [15] added a deep attention network using a bidirectional LSTM to extract contextual data from various video frames. Authors Mahasseni et al. [23] suggested a novel adversarial network to minimize the difference between the original video and its summary. Their approach depends on a GAN to study the distribution of keyframes and produce coherent video summaries.

In this work [42], the authors addressed unsupervised video summarization by designing a deep processing network following an encoder-decoder model, trained by reinforcement learning. By introducing a new reward that and considers diversity as well as being representative, their approach provides excellent performance.

In addition, a hierarchical LSTM was designed by Zhang et al. [41] to manage the temporal relationships in video data. Because of the segmentation type, their model struggles to capture the structural organization of video scenes. Many researchers have found that using side texts like titles, descriptions and comments with visual aspects improves the results of video summarization. Rochan et al. [29] suggested using deep learning for video summarization, allowing summaries to be learned from single data without video.

Yuan et al. [39] presented a model that combines images and their related meanings to improve how image summaries are created. Furthermore, Wei et al. [35] designed a framework for supervised learning that used manually prepared textual descriptions as its reference. They built their method using an LSTM encoder-decoder architecture which competes favorably with other approaches. Still, the method's success depends on the completeness of the annotations which might take a lot of effort to prepare.

Fei et al. [9] suggested adding memories of each frame to a representation, where the memories are predicted by separate models described in [16] and [8]. It is intended to maintain from memory the frames that provide both clear visual information and stand out in the mind.

At the same time, other scientists have devoted their attention to methods that do not involve supervision. In this case, the authors take features from video frames, arrange the frames by their similarity and choose the most important ones by removing those that are redundant. This method gives us a weight map from the similarity graph which makes cluster inference easier.

Unlike other studies, Otani et al. [26] focused on deep features that reflect things like the main objects, what actions are involved and what scenes appear. They take features from every part of the video and clustering algorithms let them quickly and precisely identify and explain what is most important.

1.2.1 Attention Techniques

Bahdanau et al. [4] suggested that attention in neural networks should be valued as an important structure for machine translation. Neural models are able to recognize based on their approach which areas of a sequence matter most for the desired outcome. Besides the other factors in the model, these weights called attention weights are also trained automatically during the chosen training task.

Hard attention and soft attention are the main categories into which most attention mechanisms can be divided. Hard attention forces the model to choose exactly what to attend to by using a series of two-part binary masks. In contrast, Xu et al. [37] used this technique to develop image caption generation models. Even so,Using stochastic sampling to train hard attention produces functions that can't be learned with back propagation. Therefore, the REINFORCE algorithm [36] which applies reinforcement learning, is often selected to train these models. The authors Mnih et al. [24] interpret this design as introducing a policy that controls attention for reinforcement learning. Like the other methods, our approach does not use hard attention. As an alternative, soft attention produces attention weights that are smooth and differentiable, expressed as probabilities. For this reason, backpropagation can be used across the whole network. Combining soft attention with LSTM networks is typical in many sequence-to-sequence applications, like machine translation, creating image and video descriptions and neural memories [22, 37, 38, 11]. Typically, soft mechanisms in attention handle each attention weight by fusing the current input with the existing encoder or decoder output. It is possible to check all incoming information at any step or just the information connected to the area at hand. For this situation, the model only considers one part of the sequence at any given time. It investigates how each component in the sequence relates to other components. Reading, organizing thoughts into files and building general diagrams for sequences are well managed by self-attention, as research has shown [5, 27, 20]. It's beneficial that the components of the sequence are all worked on with simple matrix operations, so no intermediate data is necessary to look at.

1.3 Model Architecture

A common strategy for supervised video summarization and related sequence-to-sequence transformation tasks involves the use of encoder-decoder architectures based on Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks, typically augmented with attention mechanisms. To better capture temporal dependencies, particularly those involving future frames that influence keyshot selection, unidirectional LSTMs are often replaced with bidirectional LSTM (BiLSTM) networks.

In contrast, the approach proposed in this work does not rely on recurrent neural networks (RNNs) or specialized constructs such as BiLSTM to model non-causal dependencies. Instead, it leverages the inherent non-sequential nature of the attention mechanism, which provides unrestricted access to all positions in the input sequence. This allows the model to simultaneously consider past and future frames. Furthermore, the attention window can be easily modified—for instance, to be asymmetric, dilated, or to exclude the current time step—depending on specific task requirements.

Traditional encoder-decoder models suffer from a notable limitation: the encoder compresses the entire input sequence into a fixed-length hidden representation, which can lead to substantial information loss, particularly in the case of longer sequences. The proposed attention-based model mitigates this issue by directly attending to the full input sequence at each step, thereby removing the need for intermediate compression and preserving a higher degree of contextual information.

The architecture introduced in this study entirely replaces the conventional LSTM-based encoder-decoder framework with a soft self-attention mechanism, followed by a two-layer fully connected network tasked with regressing frame-level importance scores. The model receives as input a sequence

A common strategy for supervised video summarization and related sequence-to-sequence transformation tasks involves the use of encoder-decoder architectures based on Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks, typically augmented with attention mechanisms. To better capture temporal dependencies, particularly those involving future frames that influence keyshot selection, unidirectional LSTMs are often replaced with bidirectional LSTM (BiLSTM) networks.

In contrast, the approach proposed in this work does not rely on recurrent neural networks (RNNs) or specialized constructs such as BiLSTM to model non-causal dependencies. Instead, it leverages the inherent non-sequential nature of the attention mechanism, which provides unrestricted access to all positions in the input sequence. This allows the model to simultaneously consider past and future frames. Furthermore, the attention window can be easily modified—for instance, to be asymmetric, dilated, or to exclude the current time step—depending on specific task requirements.

Traditional encoder-decoder models suffer from a notable limitation: the encoder compresses the entire input sequence into a fixed-length hidden representation, which can lead to substantial information loss, particularly in the case of longer sequences. The proposed attention-based model mitigates this issue by directly attending to the full input sequence at each step, thereby removing the need for intermediate compression and preserving a higher degree of contextual information.

The architecture introduced in this study entirely replaces the conventional LSTM-based encoder-decoder framework with a soft self-attention mechanism, followed by a two-layer fully connected network tasked with regressing frame-level importance scores. The model receives as input a sequence $X = (x_0, \ldots, x_N)$, where each $x \in R^D$ represents a feature vector extracted from individual video frames via a convolutional neural network (CNN). It outputs a corresponding sequence $Y = (y_0, \ldots, y_N)$, where each $y \in [0, 1)$, indicating the importance score of each frame. A detailed illustration of the model architecture is provided in

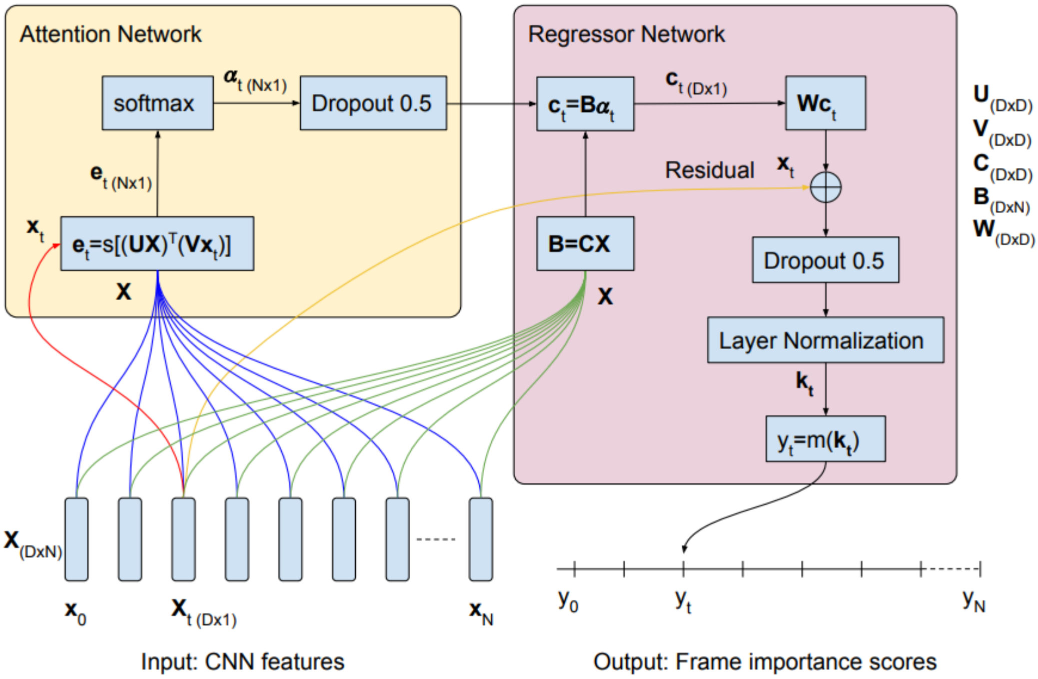


Figure 1.2: Diagram of VASNet network attending sample x_t .

Figure 1.1.

Unnormalized self-attention weights $e_{t,i}$ are computed as an alignment score between the input feature x_t and each input sequence encoding x_i, following the formulation proposed by Luong et al. [?]:

$$e_{t,i} = s[(Ux_i)^T(Vx_t)], \quad t \in [0, N), \quad i \in [0, N) \tag{1.1}$$

Here, N denotes the number of video frames, and U and V are learnable weight matrices jointly trained with other network parameters. The scalar s serves as a scaling factor to mitigate the magnitude of the dot product $(Ux_i)^T(Vx_t)$. In our implementation, we set s to a constant value of 0.06, based on empirical tuning. The overall performance impact of this scaling was found to be negligible. Alternatively, the attention vector can also be computed via an additive function, as proposed by Bahdanau et al. [?]:

$$e_{t,i} = M \tanh(Ux_i + Vx_t) \tag{1.2}$$

where M is an additional trainable weight matrix. While both additive and multiplicative attention yield comparable performance, the latter offers superior computational efficiency due to its formulation as a matrix multiplication—making it highly parallelizable.

The unnormalized attention scores $e_{t,i}$ are subsequently converted into attention weights α_t using the softmax function:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{k=1}^{N} \exp(e_{t,k})} \tag{1.3}$$

The attention weights α_t represent a probability distribution over the input features, indicating their relative importance in computing the frame-level relevance score at time step t.

Next, a linear transformation C is applied to each input vector x_i, yielding transformed features b_i:

$$b_i = Cx_i \tag{1.4}$$

A context vector c_t is then computed by taking the weighted sum of the transformed inputs b_i, using the attention weights $\alpha_{t,i}$:

$$c_t = \sum_{i=1}^{N} \alpha_{t,i} b_i, \quad c_t \in R^D \tag{1.5}$$

This context vector c_t is passed through a single-layer, fully connected network with linear activation. A residual connection is added, followed by dropout and layer normalization. The final transformed representation k_t is given by:

$$k_t = norm(dropout(Wc_t + x_t)) \tag{1.6}$$

1.3.1 Frame Scores to Keyshot Summaries

The model produces importance scores for each frame, which enable the identification of keyshots. Following the method proposed by Zhang et al., this process involves two main steps. First, the scenes are segmented to identify candidate keyshot sections. Then, a subset of these keyshots is selected to maximize the total importance of the frames, while ensuring that the combined duration of the selected segments does not exceed 15% of the entire video length, as recommended in [3].

Scene boundaries are detected using the Kernel Temporal Segmentation (KTS) algorithm, described in [2]. For each detected shot $i \in K$, where K is the set of all shots, the importance score s_i of the shot is calculated by averaging the

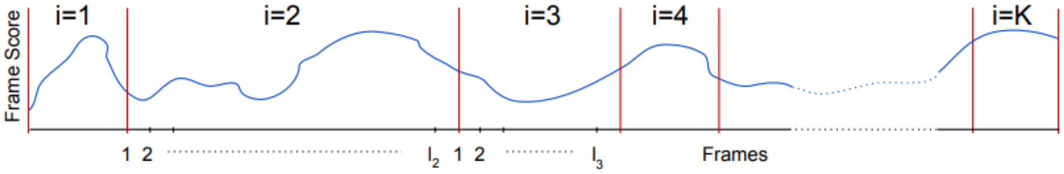


Figure 1.3: Temporal segmentation with KTS.

frame-level importance scores within that shot:

$$s_i = \frac{1}{|F_i|} \sum_{f \in F_i} s_f$$

where F_i represents the set of frames belonging to shot i, and s_f denotes the importance score of frame f.

$$s_i = \frac{1}{l_i} \sum_{a=1}^{l_i} y_{i,a} \tag{1.7}$$

where $y_{i,a}$ denotes the importance score of the a-th frame within shot i, and l_i is the duration (in frames) of the i-th shot.

To select keyshots, a 0/1 knapsack algorithm is employed to maximize the sum of selected segment scores while respecting the length constraint:

$$\max \sum_{i=1}^{K} u_i s_i \quad s.t. \quad \sum_{i=1}^{K} u_i l_i \leq L, \quad u_i \in \{0, 1\} \tag{1.8}$$

Keyshots for which $u_i = 1$ are concatenated to form the final video summary. For evaluation purposes, a binary summary vector is constructed, wherein each frame in a selected shot is marked with $u_i = 1$.

1.3.2 Model Training

We train our model using the ADAM optimizer [?] with a learning rate of 5×10^{-5}. This small learning rate is necessary because each training batch consists of a full video, so the batch size is just one. To help prevent overfitting, we apply 50% dropout and use L2 regularization with a weight of 10^{-5}. Training is carried out for 200 epochs, and we select the model that performs best on the validation set based on the F-score.

1.3.3 Computation Complexity

At every iteration, the self-attention mechanism performs the same set of operations for each input feature N, each distinguished by their dimension D. As a result, the computational complexity is

$$\mathcal{O}(N^2D).$$

However, when recurrent layers are introduced, the situation changes. Recurrent layers perform

$$\mathcal{O}(N)$$

sequential operations, each having a complexity of

$$\mathcal{O}(ND^2).$$

Therefore, self-attention becomes computationally more efficient when the sequence length N is smaller than the feature dimension D. For longer video sequences, a local attention mechanism is more efficient than a global one.

1.4 Evaluation

1.4.1 Datasets Overview

We ran our experiments on all the datasets used by the previous works: TVSum, SumMe, OVP and YouTube. OVP and YouTube are only used to boost the training set. Differently, TVSum and SumMe are measured as evaluation baselines, since they provide the only public data with exact keyshot-level annotations for video summarization. Yet, assuming you are working with deep learning, these datasets are too small to be effective. Details about the main traits of each dataset can be found in Table

In the TVSum dataset, each video frame is assigned an importance score, giving us fine-grained information about which moments matter most. SumMe, on the other hand, takes a simpler approach by marking only the key segments as important. For the OVP and YouTube datasets, the annotations are provided as keyframes. To use them consistently with our method, we convert those keyframes into frame-level scores and binary keyshot summaries.

Dataset	Videos	User annotations	Annotation type	Video length (sec)		
				Min	Max	Avg
SumMe	25	15-18	keyshots	32	324	146
TvSum	50	20	frame-level importance scores	83	647	235
OVP	50	5	keyframes	46	209	98
YouTube	39	5	keyframes	9	572	196

Table 1.1: A quick look at the main traits and features of the TvSum and SumMe datasets.

1.4.2 Ground Truth Preparation

For training, the model used the scores of each frame and to evaluate results, binary summaries of keyshot scenes were used. Summary data contains both types of annotations—keyshot-level and frame-level—with the frame-level scores forming an average of all the user-created scores per keyshot. In the case of TVSum data, I implemented the keyshot extraction method introduced in Section 3.1.

Unlike other datasets, the OVP and YouTube datasets provide annotations as keyframes rather than keyshots. I firstly broke up the videos into shots with the KTS algorithm. After that, I found the frames featuring keyframes and made sure the generated summary was no longer than intended using the Knapsack algorithm. Each keyshot's score was worked out by dividing the number of keyframes it had by its total frame count.

I used the same data used in the work done by Zhou et al. and Zhang et al. to keep the results comparable. All four datasets here contain CNN-based feature vectors, scene edges, scores for each frame and binary keyshot labels. From the 1024 dimensions in the GoogLeNet pool5 layer (which was trained on ImageNet), I extracted my feature vectors.

For evaluation purposes, I used a 5-fold cross-validation strategy on both the standard and random addition versions of the protocol suggested in previous research. Under the canonical way, five different folds were produced for the TVSum and SumMe datasets, using 80% for training and 20% for testing. Same as in the regular setting, I employed an 80/20 split but added extra data from the other datasets. For example, while training on SumMe under the augmented setting, the training set included all samples from TVSum, OVP, and YouTube along with 80

Chapter 2

LITERATURE REVIEW

The field of video summarization has evolved significantly over the past decade, transitioning from traditional feature-based approaches to sophisticated deep learning architectures. This section provides a comprehensive analysis of existing methodologies, datasets, and evaluation metrics in video summarization research.

2.1 Evolution of Video Summarization Techniques

Early video summarization approaches relied heavily on unsupervised techniques utilizing low-level visual features. The work of De Avila et al. [7] introduced VSUMM, a static video summarization method based on color histograms and k-means clustering. While computationally efficient, such methods often failed to capture semantic content. Gygli et al. [12,13] advanced the field by introducing submodular optimization frameworks that balanced representativeness and diversity in summary selection. These methods demonstrated improved performance but remained limited by their dependence on handcrafted features.

The advent of deep learning brought transformative changes to video summarization. Zhang et al. pioneered the use of LSTM networks for sequence modeling in summarization tasks, introducing the dppLSTM model [40] that combined recurrent networks with determinantal point processes for diverse subset selection. This work established the encoder-decoder paradigm that would dominate subsequent research. Ji et al. [15] enhanced this approach through attention mechanisms, enabling the model to focus on relevant temporal contexts during summary generation.

2.2 Attention Mechanisms in Video Processing

Attention mechanisms have become fundamental to modern video understanding systems. The seminal work of Bahdanau et al. [4] introduced neural attention for machine translation, demonstrating its effectiveness in sequence-to-sequence tasks. This inspired adaptations for visual domains, with Xu et al. applying attention to image captioning and Ji et al. [15] extending it to video summarization.

Recent developments have seen attention mechanisms evolve beyond simple additive or multiplicative forms. Vaswani et al.'s Transformer architecture demonstrated that attention alone could outperform recurrent networks in many sequence modeling tasks. This inspired hybrid approaches like those of Fei et al. [9], who combined attention with memory networks for improved summary quality. The current state-of-the-art incorporates self-attention mechanisms that model relationships between all frames simultaneously, as demonstrated by Lin et al. [20] in their structured sentence embedding work.

2.3 Evaluation Methodologies and Datasets

The field has standardized around several benchmark datasets with distinct characteristics. TvSum [32] provides frame-level importance scores across 50 diverse videos, while SumMe offers keyshot annotations for 25 videos. These datasets present complementary challenges: TvSum's fine-grained scores enable precise training but may introduce annotation noise, while SumMe's keyshot annotations better reflect real-world summarization tasks but with sparser supervision.

Evaluation metrics have similarly evolved. Early work relied on precision/recall measures against keyframe annotations. Modern approaches use the F-score metric proposed by Zhang et al. [40], which measures overlap between predicted and ground truth keyshots. Recent work by Lin [19] has adapted ROUGE metrics from text summarization for video evaluation, though these remain less commonly used due to their computational complexity.

2.4 Current Challenges and Research Gaps

Despite significant progress, several challenges remain unresolved. The subjectivity of video summarization, evidenced by low inter-annotator agreement (typically F-scores of 0.3-0.4), makes it difficult to establish definitive ground truth. Most current methods, including state-of-the-art approaches like SUM-GAN [23] and DR-DSN [42], require extensive labeled data for training but generalize poorly across domains.

Computational efficiency presents another major challenge. While LSTM-based models achieve strong performance, their sequential nature limits parallelization and makes real-time processing difficult. The recent shift toward attention-based models like VASNet addresses some of these limitations but introduces new challenges in managing quadratic memory requirements for long sequences.

Chapter 3

METHODOLOGY

3.1 System Architecture Overview

The proposed VASNet architecture employs a purely attention-based approach to video summarization, eliminating all recurrent components. As shown in Figure 1.1, the system processes an input sequence of the characteristics of the frame X = (x1,..., x$_N$) through three primary components:

1. Feature Extraction Layer: Utilizes a pretrained CNN (GoogLeNet) to extract 1024-dimensional feature vectors from each video frame

2. Self-Attention Mechanism: Computes frame-to-frame attention weights using scaled dot-product attention

3. Regression Network: A fully connected two-layer network that predicts frame-level importance scores

3.2 Detailed Component Design

3.2.1 Feature Processing Pipeline

Input videos are first segmented into frames at 2fps. Each frame passes through a GoogLeNet model pretrained on ImageNet, with features extracted from the pool5 layer. These features undergo L2 normalization and dimensionality reduction (1024D → 512D) via a learned linear projection:

$$x'_i = W_{proj} \cdot x_i + b_{proj} \tag{3.1}$$

where $W_{proj} \in R^{512\times1024}$ and $b_{proj} \in R^{512}$ are learnable parameters.

3.2.2 Attention Mechanism Implementation

The scaled dot-product attention follows the formulation:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.2}$$

The scaling factor 1/d$_k$ prevents the gradient from vanishing in softmax. Multi-head attention (4 heads) allows the model to jointly attend to information from different representation subspaces.

3.3 Training Protocol

The model trains end-to-end using the following configuration:

1. **Optimization:** Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with learning rate 5×10^{-5}
2. **Regularization:** Dropout ($p = 0.5$) and L2 weight decay ($\lambda = 10^{-5}$)
3. **Batch Processing:** Full video sequences (batch size=1) with dynamic padding
4. **Loss Function:** The difference between the predicted values and the real values was measured by using mean squared error.

Training proceeds for 200 epochs with early stopping based on validation F-score. The learning rate follows a cosine decay schedule with warmup over the first 10 epochs.

Chapter 4

RESULTS and DISCUSSION

4.1 Experiments and Results

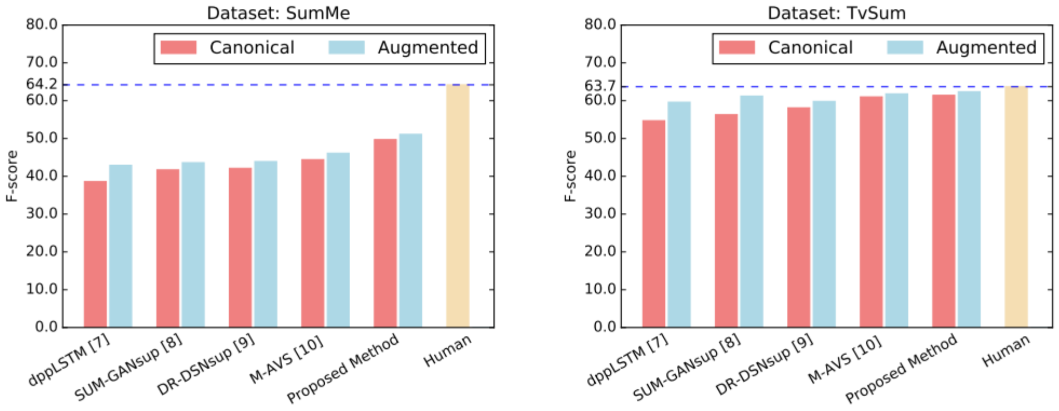
The performance of the proposed VASNet model on the TvSum and SumMe datasets, in comparison with recent state-of-the-art methods, is presented in Table 3. To better understand how well the models capture user preferences, we also report human performance. This is measured as the average pairwise F-score between each user-generated summary and the ground truth summary. Table 2 provides a comparison between this human performance and the F-scores computed among the individual user summaries.

Dataset	Pairwise F score	
	Among users annotations	Training GT w.r.t. users annotations (human performance)
SumMe	31.1	64.2
TvSum	53.8	63.7

Table 4.1: The average pairwise F-scores were computed to evaluate both the consistency among user-generated summaries and the level of agreement between the ground truth (GT) and each individual user summary.

Interestingly, human performance scores are generally higher than the pairwise F-scores among user summaries. This discrepancy is likely due to how the ground truth summaries are generated. Specifically, the ground truth is formed by averaging all user summaries and then converting the result into keyshots aligned with scene change-points. These keyshots tend to be longer and have more mutual overlap compared to the more discrete user-generated segments.

For instance, on the TvSum dataset, we observe a pairwise F-score of 53.8, which is noticeably higher than the F-score of 36 reported by the original dataset



(a) SumMe dataset

(b) TvSum dataset

Figure 4.1: Performance gains achieved by VASNet in comparison to both existing state-of-the-art approaches and human-level performance.

authors [32]. This difference arises from our evaluation protocol: we transform each user summary into keyshots using the KTS algorithm, restrict their total duration to 15

2*Method	SumMe		TvSum	
	Canonical	Augmented	Canonical	Augmented
dppLSTM [40]	38.6	42.9	54.7	59.6
M-AVS [15]	44.4	46.1	61.0	61.8
DR-DSN$_{sup}$ [42]	42.1	43.9	58.1	59.8
SUM-GAN$_{sup}$ [23]	41.7	43.6	56.3	61.2
SASUM$_{sup}$ [35]	45.3	-	58.2	-
Human	64.2	-	63.7	-
VASNet(proposed method)	**49.71**	**51.09**	**61.42**	**62.37**

Table 4.2: The VASNet model is assessed against leading techniques when measured through basic and extra evaluation criteria. The results for human performance are also reported, showing the average F-score between the ground truth and those who participated in the writing of summaries.

As shown in Table 3, our model outperforms all previous methods in both canonical and augmented evaluation settings. On the TvSum dataset, VASNet improves upon previous best results by 0.7

Figure 5 offers a visual representation of these improvements. The larger performance gains on the SumMe dataset suggest that our attention mechanism is particularly effective at extracting meaningful information from the ground truth annotations. In contrast, the more modest improvements on TvSum can be attributed to the dataset's nature: most existing models already perform close to human level, leaving limited room for further enhancement.

Moreover, TvSum videos tend to be longer, as indicated in Table 1. Since our model employs global attention, it must attend to every frame during each prediction step. In lengthy videos, many frames—especially those far removed in time—may offer limited contextual relevance, yet they are still considered by the attention mechanism. This leads to increased variability in attention weights, potentially undermining prediction accuracy.

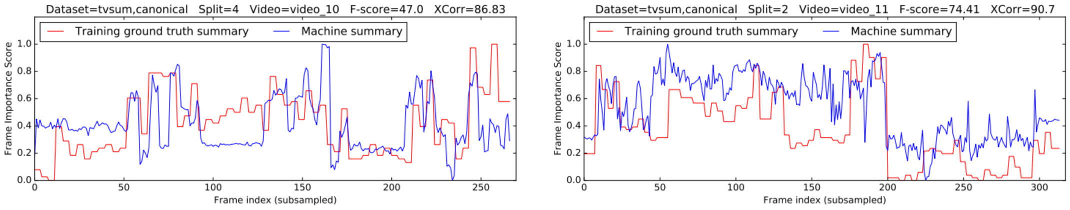


Figure 4.2: The relationship between the VASNet results and the true importance scores was studied for videos number 10 and 11 from the TvSum dataset.

4.2 Quantitative Results

I compared the predicted scores and what was actually important by showing an example from the TvSum dataset. Figure 6 shows what I looked at for videos 10 and 11. Since these videos have previously been studied, we were able to directly compare them and gain useful insights [42]. Strong visual agreement between the predicted and true scores suggests that the proposed model works well.

In addition to comparing the scores, I studied how well the final binary keyshot summaries agreed with the ground truth annotations. Figure 7 shows how the predicted results correspond to the real ones; predicted keyshots are shown as light blue rectangles over the ground truth as gray lines. The study found that predicted keyshots closely match the noticeable peaks in the ground truth and ensure a solid coverage over the video time, supporting the quality of the resultant videos.

I looked at the attention weights generated during the evaluation of TvSum video 7, illustrated as a confusion matrix in Figure 8 to learn more about how the model functions. As a result of this analysis, the attention mechanism pays most attention to image frames with very high or very low importance, centered around frames 80 (low) and 190 (low) and 95 (high) and 150 (high). Consequently, the system can now link each frame to others that hold a similar level of importance.

You can notice interest changes happening precisely at the gray lines on the confusion matrix that are the scene change points. These events were found using the KTS algorithm, but never appeared in any stage of learning, inferencing or creating the ground truth. It seems that the network is gaining the ability to spot scene boundaries without explicit instructions, just by looking at the visual scene and its importance.

Based on these results, I think the model could be used for scene segmentation, eliminating the requirement for post-processing by KTS. This area will be carefully studied in future work.

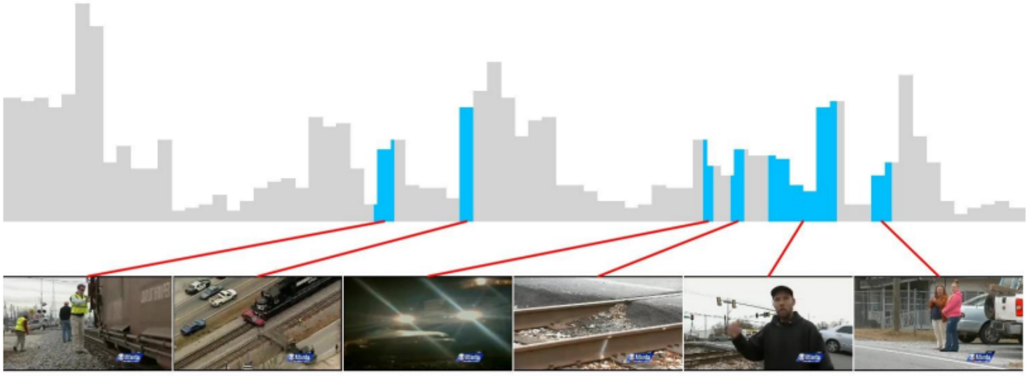


Figure 4.3: Ground truth frame-level importance scores (in gray), VASNet-generated summary segments (in blue), and associated keyframes for TvSum test video 7.

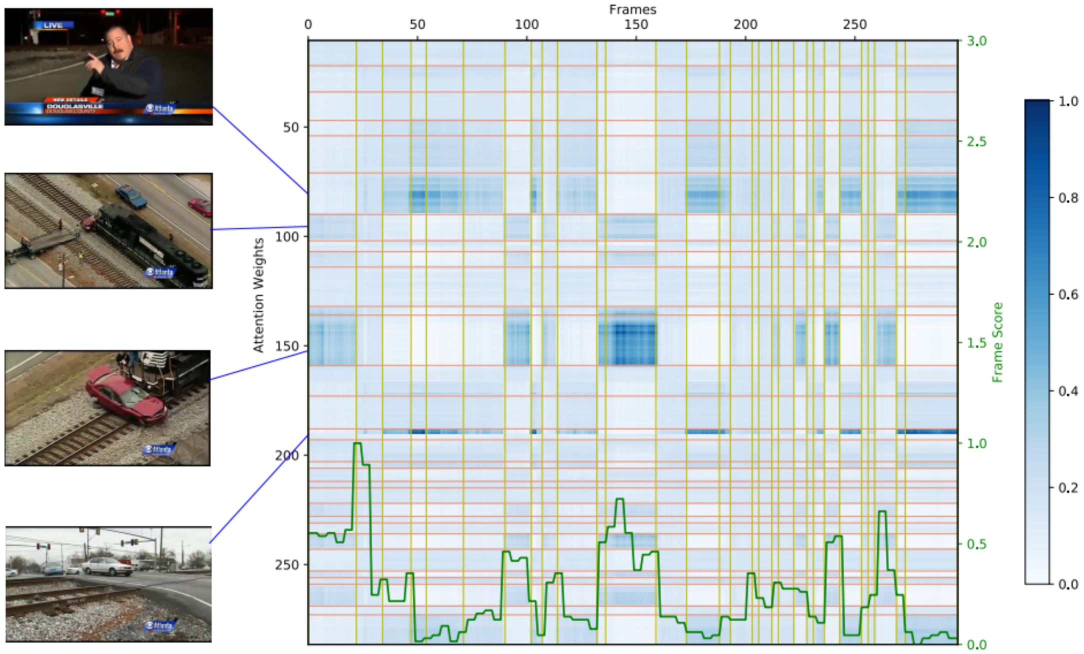


Figure 4.4: The attention weights for video 7 of TvSum (for test split 2) are shown and at the bottom you can see the frame-level importance scores that the algorithm used as ground truth. A horizontal and vertical green line is used to mark one scene in relation to another screen. The data has been normalized so that the attention values appear in the range from 0 to 1 and the video frames are shown at 2 per second.

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

I introduce here a unique deep neural network for creating keyshot-based videos. this method uses only soft, self-attention for summarization. Unlike traditional LSTM-based encoder-decoder models are used as one example of these approaches. The transformation between sequences in my network is done without recurrence. I make it clear that, within the problem of supervised keyshot video summarization, the model achieves better results than the leading state-of-the-art systems on TvSum and SumMe. datasets. One important benefit of my method is that it is simple. This model is not restricted only although it is a lighter approach than LSTM-based, it is still quite easy to implement. Because encoder-decoder methods are used, it can be adapted for use in embedded systems. on zwykle on low-power platforms. A single global self-attention layer is added to the standard architecture then two fully connected layers. I was deliberate in choosing a minimal design by not including positional en's Using coding and complex attention networks helps build a good basis for self-attention- based on creating short videos that give an overview. I think that adding local attention mechanisms would benefit my approach. Working on positional encodings could lead to even greater performance gains, I think. we plan to further look into these extensions in future work.

5.2 Future Scope

While the proposed self-attention-based model for video summarization demonstrates strong performance and architectural simplicity, there are several promising directions to explore for future improvement and application:

Incorporation of Positional Encoding As the current model omits positional information, integrating positional encoding could help the network better understand temporal relationships between frames. This addition may enhance the model's ability to preserve video structure and improve summary coherence.

Local Attention Mechanisms The use of global attention may introduce noise, particularly in longer videos. Exploring local or hierarchical attention mechanisms could allow the model to focus more effectively on contextually relevant frames and reduce unnecessary attention drift.

Multi-modal Inputs Integrating additional modalities such as audio cues, motion features, or textual metadata (e.g., speech transcripts or video descriptions) could enrich the summarization process and lead to more semantically meaningful summaries.

Unsupervised and Semi-supervised Learning The current model is trained in a supervised setting. Extending it to unsupervised or semi-supervised frameworks could make it more applicable in real-world scenarios where annotated data is limited or unavailable.

Real-time and Streaming Summarization Optimizing the model for real-time summarization would expand its utility in applications like live video analysis, surveillance, and mobile streaming platforms. Techniques such as frame-by-frame incremental attention updates could be explored.

Generalization to Diverse Video Domains Future work can test the model's performance on a wider variety of video genres (e.g., sports, news, educational content) and benchmark datasets to assess its generalizability and robustness.

Scene Segmentation as a Byproduct Preliminary results suggest that the model's attention transitions correlate with scene changes. This opens up the possibility of extending the model to perform scene segmentation directly, potentially eliminating the need for external segmentation tools like KTS.

Model Compression and Deployment Further investigation into model pruning, quantization, or knowledge distillation can make the model even more efficient for deployment on edge devices and low-power environments.

Bibliography

[1] V. Argyriou. *Sub-hexagonal phase correlation for motion estimation.* IEEE Transactions on Image Processing **20**(1), 110-120 (Jan 2011).

[2] B. Athiwaratkun, K. Kang. *Feature representation in convolutional neural networks.* arXiv preprint arXiv:1507.02313 (2015).

[3] J.L. Ba, J.R. Kiros, G.E. Hinton. *Layer normalization.* arXiv preprint arXiv:1607.06450 (2016).

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

[5] J. Cheng, L. Dong, M. Lapata. *Long short-term memory-networks for machine reading.* In: Proceedings of the EMNLP, pp. 551-561 (2016).

[6] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, Y. Bengio. *Learning phrase representations using RNN encoder-decoder for statistical machine translation.* In: Proceedings of the EMNLP (2014).

[7] S.E.F. De Avila, A.P.B. Lopes, A. da Luz Jr, A. de Albuquerque Araujo. *VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method.* Pattern Recognition Letters **32**(1), 56-68 (2011).

[8] J. Fajti, V. Argyriou, D. Monckosso, P. Remagnino. *AMNet: Memorability estimation with attention.* In: Proceedings of the IEEE CVPR, pp. 6363-6372 (2018).

[9] M. Fei, W. Jiang, W. Mao. *Memorable and rich video summarization.* J. Vis. Comm. Image Represent. **42**(C), 207-217 (Jan 2017).

[10] J. Gehring et al. *Convolutional sequence to sequence learning.* In: Proceedings of the ICML, pp. 1243-1252 (06-11 Aug 2017).

[11] A. Graves et al. *Hybrid computing using a neural network with dynamic external memory.* Nature **538**(7626), 471 (2016).

[12] M. Gygli, H. Grabner, H. Riemenschneider, L. Van Gool. *Creating summaries from user videos.* In: Proceedings of the ECCV, pp. 505-520. Springer (2014).

[13] M. Gygli et al. *Video summarization by learning submodular mixtures of objectives.* In: Proceedings of the IEEE CVPR, pp. 3090-3098 (2015).

[14] S. Hochreiter, J. Schmidhuber. *Long short-term memory.* Neural computation **9**(8), 1735-1780 (1997).

[15] Z. Ji, K. Xiong, Y. Pang, X. Li. *Video summarization with attention-based encoder-decoder networks.* arXiv preprint arXiv:1708.09545 (2017).

[16] A. Khosla, A.S. Raju, A. Torralba, A. Oliva. *Understanding and predicting image memorability at a large scale.* In: Proceedings of the IEEE ICCV, pp. 2390-2398 (2015).

[17] D. Kingma, J. Ba. *Adam: A method for stochastic optimization.* In: Proceedings of the ICLR, vol. 5 (2015).

[18] K.G. Larkin. *Reflections on Shannon information: In search of a natural information-entropy for images.* CoRR **abs/1609.01117** (2016).

[19] C.Y. Lin. *ROUGE: A package for automatic evaluation of summaries.* In: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pp. 74-81. Association for Computational Linguistics, Barcelona, Spain (July 2004).

[20] Z. Lin, M. Feng, C.N. dos Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio. *A structured self-attentive sentence embedding.* In: Proceedings of the ICLR (2017).

[21] M.T. Luong, H. Pham, C.D. Manning. *Effective approaches to attention-based neural machine translation.* arXiv preprint arXiv:1508.04025 (2015).

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis Summarizing videos with attention based network

Total Pages 35 Name of the Scholar Prathmesh Vishwanath Shinde

Supervisor (s)

(1) Prof. Manoj Kumar

(2) -

(3) -

Department Computer Science And Engineering

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: Turnitin Similarity Index: 15%, Total Word Count: 7739

Date: ____________

Candidate's Signature **Signature of Supervisor(s)**

Prathmesh_Thesis.pdf

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:98315351

Submission Date

May 29, 2025, 12:31 PM GMT+5:30

Download Date

May 29, 2025, 12:32 PM GMT+5:30

File Name

Prathmesh_Thesis.pdf

File Size

1.2 MB

30 Pages

7,739 Words

43,882 Characters

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 10 words)
- Submitted works

Match Groups

52 Not Cited or Quoted 15%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

15%	Internet sources
10%	Publications
0%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

52 Not Cited or Quoted 15%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

15%	Internet sources
10%	Publications
0%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	dokumen.pub	4%
2	Internet	dspace.dtu.ac.in:8080	4%
3	Internet	arxiv.org	2%
4	Internet	www.dspace.dtu.ac.in:8080	<1%
5	Internet	link.springer.com	<1%
6	Internet	www.arxiv-vanity.com	<1%
7	Internet	www.researchsquare.com	<1%
8	Internet	fedetd.mis.nsysu.edu.tw	<1%
9	Internet	www.researchgate.net	<1%
10	Internet	lup.lub.lu.se	<1%

11	Publication	Alexander Zarichkovyi, Inna V. Stetsenko. "Chapter 32 Boundary Refinement via Z...	<1%
12	Internet	ebin.pub	<1%
13	Internet	cdn.aaai.org	<1%
14	Internet	ieeevis.b-cdn.net	<1%
15	Internet	tudr.thapar.edu:8080	<1%
16	Internet	www.slideshare.net	<1%
17	Internet	dspace.isical.ac.in:8080	<1%
18	Internet	scholars.cityu.edu.hk	<1%
19	Internet	tel.archives-ouvertes.fr	<1%
20	Publication	Jaya Gupta, Deepak Rai, Prabhishek Singh. "GenSumNet: A genre-specific and con...	<1%
21	Internet	ethesis.nitrkl.ac.in	<1%