**Journal of Management and Service Science, 2024,**
Vol. 04, Iss. 02, S. No. 063, pp. 1-9
ISSN (Online): 2583-1798

# Streamlining Information: Creating YouTube Video Summarizer Using Machine Learning

**Abhash Srivastava[1], Bramah Hazela[2], Shikha Singh[3], Vineet Singh[4]**

[1,2,3,4]Amity School of Engineering and Technology, Amity University, Uttar Pradesh, Lucknow, India
[1]abhash.srivastava@s.amity.edu, [2]bhazela@lko.amity.edu, [3]ssingh8@lko.amity.edu, [4]vsingh@lko.amity.edu

**How to cite this paper:** A. Srivastava, B. Hazela, S. Singh, V. Singh, "Streamlining Information: Creating YouTube Video Summarizer using Machine Learning," *Journal of Management and Service Science (JMSS),* Vol. 04, Iss. 02, S. No. 063, pp. 1-9, 2024.

https://doi.org/10.54060/a2zjournals.jmss.63

**Received:** 04/09/2023
**Accepted:** 20/05/2024
**Online First:** 10/06/2024
**Published:** 25/11/2024



## Abstract

*The aim of this study is to develop a user interface facilitating the retrieval of YouTube video summaries through the integration of Natural Language Processing (NLP) and Machine Learning techniques. With the continuous influx of videos uploaded to YouTube on a daily basis, locating relevant content has become increasingly challenging. Often, significant time and effort are expended in searching for desired content, with outcomes often proving futile due to the inability to extract meaningful information. Our project addresses this issue by providing a solution that efficiently summarizes videos, presenting users with concise yet comprehensive insights. Utilizing an abstractive summarization model, the system extracts transcripts from YouTube videos and generates condensed summaries, effectively reducing the time required for content consumption while preserving crucial information. While the implementation phase is still in progress, this paper presents the conceptual framework and initial findings of our research endeavor.*

## Keywords

*Natural Language Processing, Machine Learning, Abstractive summarization complexity*

## 1. Introduction

In the contemporary era of digitalization, technology plays a pivotal role in shaping societal advancement. The proliferation

of internet users, accessing online platforms round the clock, has resulted in a surge of information consumption. However, amidst this wealth of data, retrieving relevant information has become increasingly challenging and time-consuming. YouTube, as a premier platform for content dissemination, offers content creators unparalleled reach, leading to an exponential rise in the volume of available content. With approximately 3.7 million videos uploaded daily, navigating through this vast repository to find pertinent content has become akin to searching for a needle in a haystack.

Moreover, this abundance of content has given rise to the proliferation of clickbait videos, further complicating the search process and often leaving users with minimal or irrelevant information. To address this challenge, summarization techniques offer a promising solution. Summarization enables users to distil key concepts from text or video content efficiently, facilitating the identification of essential information while filtering out extraneous material. Additionally, by providing language conversion options, users with diverse linguistic preferences can access summarized content more effectively.

A brief overview of the video's content provided by the summary empowers users to make informed decisions about investing their time and attention. This paper focuses on video summarization utilizing the abstractive method, which generates coherent summaries by analyzing the text, as opposed to extractive summarization methods. Through this exploration, we aim to enhance the accessibility and utility of online content, ultimately improving the user experience on platforms like YouTube.

## 2. Literature Survey

In [1], the author proposed a model to generate summaries from YouTube videos, particularly useful when video transcripts are unavailable. This model employs an abstractive method for text summarization, allowing users to glean necessary information without watching lengthy videos, thereby optimizing time management.

In one research, [2] authors present a project utilizing an embedding layer to convert words into vector representations for generalization in prediction or summary generation. The TFIDF technique aids in identifying important terms within the corpus. The research in [3] employs an Automatic NLP-based LSA summarization algorithm on video subtitles, utilizing Latent Semantic Analysis (LSA) to extract features of sentences. This method prioritizes top-ranked subtitles for summarization.

In one research, [4] authors survey abstractive transcript summarization methods for YouTube videos, highlighting differences between extractive and abstractive techniques. The paper evaluates various summarization methods and models, weighing their pros and cons. Using the Hugging Face Transformer, [5] conducts abstractive summarization on YouTube transcripts, facilitated by RESTAPI. This approach provides condensed summaries based on video transcripts.

In one research, authors [6] emphasize the importance of video summarization and skimming, especially in video management systems. The article discusses abstraction work for typical videos, including rhythmic analysis and cinematic conventions. In one research, [7] author explores extracting key ideas from videos for translation into English, creating subsystems to comprehend foreign videos.

In one research, [8] authors provide a technical background on document summarization, highlighting challenges in current techniques such as clustering-based methods and MMR strategies. Focusing on extractive summarization, [9] suggests combining multiple strategies to enhance summarization quality compared to relying on a single approach.

In [10], the author proposes a transcript summarizer utilizing NLP methods to extract and summarize content from video files, employing extractive text summarization techniques for various video sizes. The goal of the paper described in the provided text is to design a Chrome extension for YouTube Transcript Summarizer, offering features such as noise reduction, task management, and additional capabilities like email sharing, transcript translation, and speech synthesis.

## 3. Proposed System and Methodology

The proposed work entails the development of the YouTube Transcript Summarizer model. This model operates by initiating an HTTP request to the backend server using the YouTube transcript, subsequently executing transcript summarization, and delivering the summarized transcript as an HTTP response. The summarized transcript is then displayed on the Chrome Ex-tension, facilitating user access and comprehension (refer to Fig. 1 for an overview of the process).

### 3.1. Proposed System

The processes involve the following steps (as in Figure 2):

1. Open a YouTube video and activate the Chrome Extension by selecting the "summarize" option.
2. Initiate an HTTP request to the backend server, requesting the transcript for the provided YouTube ID.
3. Conduct transcript summarization utilizing predetermined algorithms and techniques.
4. Transmit the summarized transcript back to the Chrome Extension as an HTTP response.
5. Display the summarized transcript within the extension interface for user accessibility and review.

By following this systematic approach, users can efficiently obtain condensed summaries of YouTube video content, en-hancing their browsing experience and optimizing information retrieval.

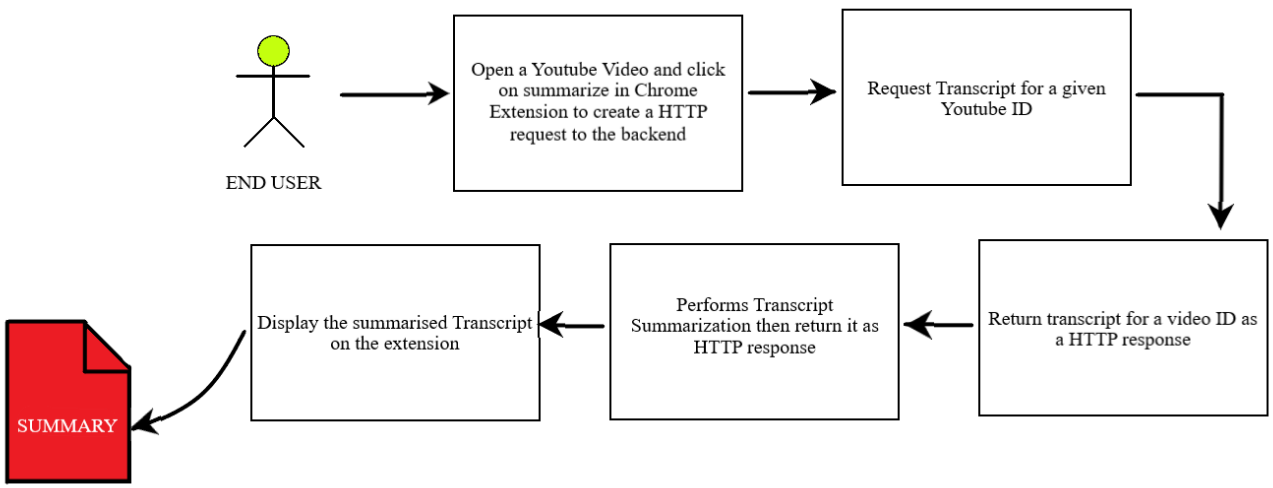


**Figure 1.** Project Stages

### 3.2. Methodology

Our transcript summarization method is divided into following parts:

**A. Text Cleaning**

We will leverage the capabilities of the Spacy library for our project. Spacy is specifically designed for tasks involving in-formation extraction and natural language understanding. With Spacy, text can be segmented into individual words, punctuation marks, and assigned word roots. Furthermore, Spacy offers functionalities such as serialization and text classifi-cation, adding versatility to our project.

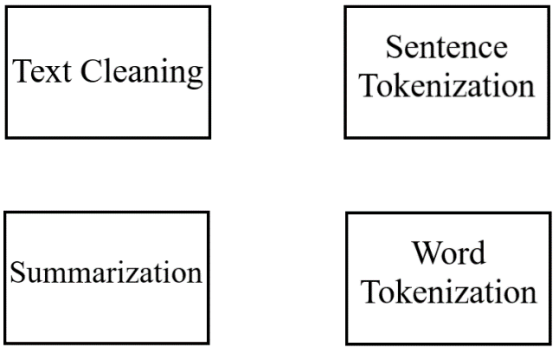


**Figure 2.** Processes of Summarization

Utilizing Spacy, any text can be transformed into a processed Doc object, enabling the inference of various properties. This allows for efficient analysis and manipulation of text data, facilitating tasks such as summarization and information extraction. The robust features and ease of use offered by Spacy make it an ideal choice for implementing our YouTube Transcript Summarizer model.

### B. Sentence Tokenization

The 'sent tokenize' function utilizes an instance of the Punkt Sentence Tokenizer from the NLTK tokenize module. This tokenizer has been pre-trained and possesses knowledge about where to identify a sentence's beginning and end based on characters and punctuation cues.

Word tokenization using NLTK offers several advantages, including various techniques such as White Space Tokenization, Dictionary-Based Tokenization, Rule-Based Tokenization, Regular Expression Tokenization, Penn Treebank Tokenization, Spacy Tokenization, Moses Tokenization, and Subword Tokenization. These techniques allow for flexible and accurate segmentation of text into individual words or tokens.

Text normalization, which encompasses various word tokenization methods, plays a crucial role in enhancing the accuracy of language understanding algorithms. Techniques such as stemming and lemmatization are employed to normalize the text by reducing words to their base or root forms, thereby standardizing the vocabulary and improving algorithm performance. Overall, these normalization procedures contribute to a more effective and robust language processing pipeline.

### C. Word Tokenization

Tokenization involves breaking a sequence of strings into various components, including words, phrases, symbols, and other tokens. In our implementation, a wrapper function named WordTokenize() invokes the tokenize() method on an instance of the Treebank class within the Word Tokenizer Table. This function facilitates the segmentation of a large text sample into individual words, a process commonly referred to as word tokenization.

Word tokenization is a crucial step in natural language processing tasks, as it enables the recording and analysis of each word for further processing. For instance, in sentiment analysis, word tokenization allows for the classification and counting of words to determine sentiment polarity or sentiment intensity. By breaking down the text into words, we can extract meaningful insights and perform various analyses to derive valuable information from the text data.

### D. Summarization

In our approach, we aim to calculate the frequency of each word in our text data and store this frequency alongside the text data in a dictionary. Subsequently, we tokenize the text data to segment it into individual words.

Once tokenized, we will proceed to determine the sentences to include in our final summary data. This selection will be based on the presence of high-frequency words within each sentence, with the frequency of each word calculated earlier. Sentences containing a higher number of high-frequency words will be prioritized for inclusion in the summary, thus ensuring that the most relevant and informative sentences are retained.

By implementing this approach, we can effectively summarize the text data while prioritizing sentences that contain key information, as indicated by the frequency of their constituent words. This method facilitates the extraction of salient content from the text, enhancing the overall quality and relevance of the generated summary.

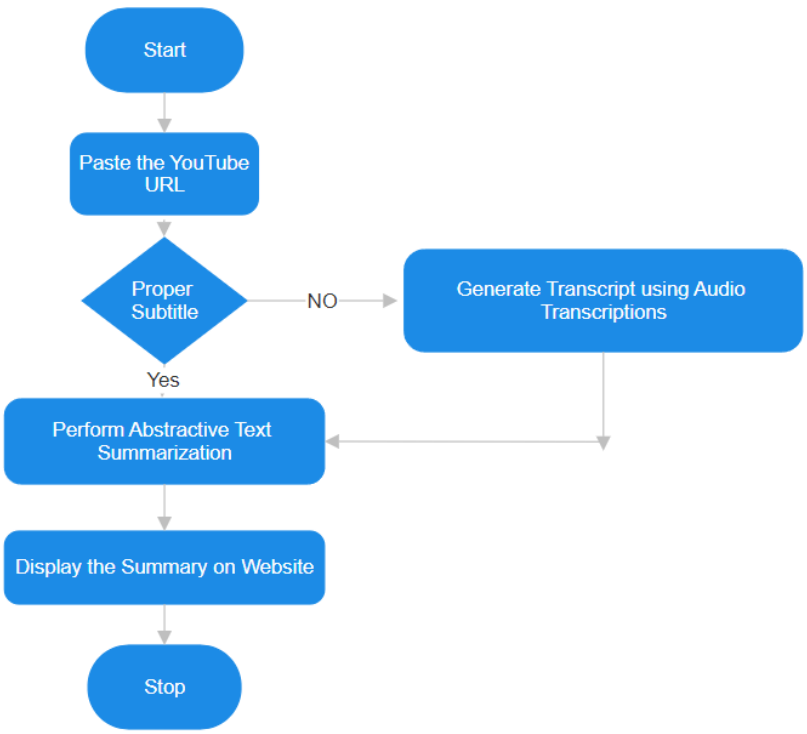


**Figure 3.** Flow-Chart of the System

**E. Checking Grammar**

Grammar and spelling checks are conducted using Language Tool, an open-source program also employed as OpenOf-fice's spellchecker. This package allows programmers to identify grammatical and spelling errors using either a Com-mand-line interface or a Python code snippet (CLI).

By leveraging Language Tool, developers can ensure the accuracy and correctness of written text, enhancing the quality and professionalism of their content. Whether through manual verification or automated checks within their code, Language Tool serves as a valuable tool for maintaining linguistic precision and coherence.

### 3.3. Flowchart

As illustrated in Fig. 3:

A. Users insert the YouTube URL into the summary extension.
B. If subtitles are available, a transcript is generated; otherwise, audio transcription is utilized to create the transcript.
C. The generated transcript undergoes abstractive-based summarization.

D. Ultimately, the extension displays the summary for user perusal.

### 3.4. Features

After the content has been summarized, the user is presented with four options via buttons:

A. Translation: Users have the choice to translate their text summary into another language, such as Hindi or Marathi.
B. Speak: Users can listen to the condensed transcript being read aloud by clicking the speak button.
C. Download: The option to download the condensed text into various file formats enables users to obtain their summarized content in a format of their preference.
D. Send to mail: If users wish to send the transcript file to their email, they can utilize this option for easy sharing and storage.

## 5. Performance measure

An evaluation process was introduced concurrently with the SumMe dataset for a videotape summary (Apostolidis et al. 2021). Another method utilizes the Mathews (Matthews 1975) correlation coefficient to evaluate implementation. A different approach was used in (Mahasseni et al. 2017; Yuan et al. 2019a) to estimate submission using a single ground-truth summary instead of considerable user summaries. The F-score evaluation of dynamic summary on dataset TVSum, SumMe, YouTube, are:

The following observations are made based on the F-score evaluation of the static and dynamic video summaries.

- The TVSum dataset is the maximum used dataset by many of the existing video summarization techniques during the last decade, where the Convolutional Neural Network Bi-Convolutional Long Short Term Memory Generative Adversarial Network method (Sreeja and Kovoor 2022) is outperformed on TVSum dataset with an F-score of 69.0% on the static summary.
- It is observed that the SumMe dataset is the second most highly used dataset by the exiting video summarization techniques during the last decade, where Deep Attentive Preserving (Ji et al. 2020) is outperformed with an F-score of 45.5% to generate the static summary on SumMe dataset.
- Multi Convolutional Neural Network outperformed on Open Video Project dataset in terms of F-score of 82.0% to generate the static summary over Convolutional Neural Network (Purwanto et al. 2018), and Property Constrained Dual Learning (Zhao et al. 2019).
- It is observed that the Multi-edge optimized LSTM RNN for video summarization (Chai et al. 2021) approach outperformed the recent video summarization techniques on the YouTube dataset in terms of F-score of 85.8% for generating the static summary.
- On a static summary using the VSUMM dataset, the Multi-edge optimized LSTM RNN for video summarization (Chai et al. 2021) approach delivered the best results compared to other video summarization approaches employed recently where the F-score value for the static summary is 92.4%.
- It is noticed that the F-score value for the Convolutional Neural Network Bi-Convolutional Long Short Term-Memory Generative Adversarial Network method (Sreeja and Kovoor 2022) for generating the dynamic summary on the Tvsum dataset is 72.0%, better than the other exiting techniques.
- The Dilated Temporal Relational-Generative Adversarial Network (Zhang et al. 2019a) method performed well on the dynamic summary using SumMe dataset with the F-score value of 51.4 %.
- It is observed that the unsupervised learning-based Cycle-SUM (Yuan et al. 2019a) method outperformed with the

F-score value of 77.3 % to generate the dynamic summary.

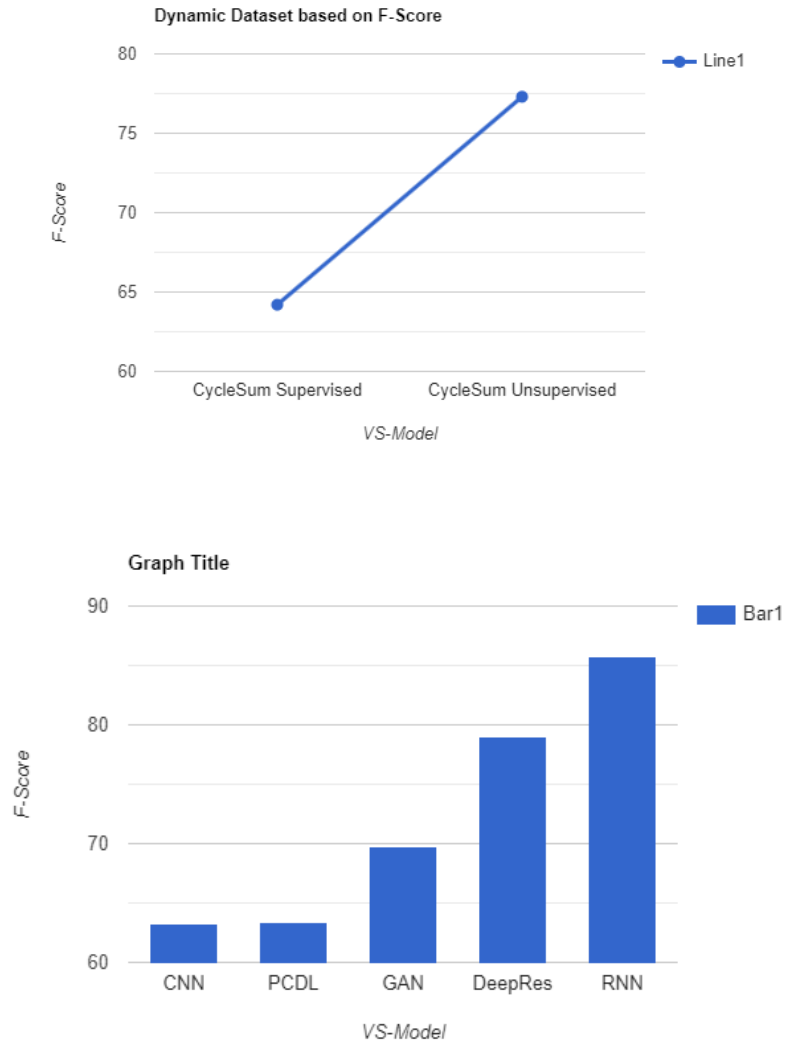


**Figure 4.** Dynamic and Static Summarization based on F-Score

## 6. Conclusion and Future Scope

In this project, the transcript summary is generated using the Abstractive summarization method. Unlike traditional techniques that rely on extracting sentences directly from the original content, abstractive summarization employs paraphrasing to condense the text. This approach eliminates the need for a deep understanding of technical concepts, making it more accessible and efficient. Existing video summarization systems often require substantial technical expertise. However, summarizing videos based on subtitles offers a quicker alternative, as processing text is inherently faster and simpler than training machine learning models on various videos.

This project holds potential benefits for hearing-impaired individuals who struggle to comprehend videos without subtitles or transcripts. By providing a generated summary, this tool enhances accessibility and facilitates understanding of video

content. In future iterations, the ability to translate the summarized text into different languages directly from the extension could be explored. Additionally, the concept of Transcript Summarizer can be extended to other streaming services, expanding its utility and reach beyond YouTube.

## References


[1.] L. Agnihotri, K. V. Devara, T. McGee, and N. Dimitrova, "Summarization of video programs based on closed captions," in *Storage and Retrieval for Media Databases* vol. 4315, pp. 599-607.

[2.] M. Ajmal, M. H. Ashraf, M. Shakir, Y. Abbas, and F. A. Shah, "Video summarization: techniques and classifcation," in *-ternational conference on computer vision and graphics*, Springer, 2012, pp. 1–13.

[3.] N. Alok, K. Krishan, and P. Chauhan, "Deep learning-based image classifer for malaria cell detection," in *Machine learning for healthcare applications*, pp. 187–197, 2021.

[4.] M. Z. Alom *et al.*, "A state-of-the-art survey on Deep Learning theory and architectures," *Electronics (Basel)*, vol. 8, no. 3, p. 292, 2019.

[5.] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, 2021.

[6.] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Unsupervised video summarization via at-ten-tion-driven adversarial learning," in *International conference on multimedia modeling*, Springer, pp. 492–504, 2020.

[7.] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "AC-SUM-GAN: Connecting actor-critic and generative adversarial networks for unsupervised video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3278–3292, 2021.

[8.] N. Archana and N. Malmurugan, "RETRACTED ARTICLE: Multi-edge optimized LSTM RNN for video summarization," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 5, pp. 5381–5395, 2021.

[9.] M. Barbieri, L. Agnihotri, and N. Dimitrova, "Video summarization: methods and landscape," in *Internet Multimedia Management Systems IV*, 2003.

[10.] M. Basavarajaiah and P. Sharma, "Survey of compressed domain video summarization techniques," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–29, 2020.

[11.] M. Basavarajaiah and P. Sharma, "GVSUM: generic video summarization using deep visual features," *Multimed. Tools Appl.*, 2021.

[12.] Y. Bengio, "Learning deep architectures for AI," *Found. Trends® Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009. https://doi.org/10.1561/2200000006

[13.] H. Binol, M. K. Niazi, C. Elmaraghy, A. C. Moberly, and M. N. Gurcan, "Automated video summarization and label as-signment for otoscopy videos using deep learning and natural language processing," *Medical imaging 2021: imaging informatics for healthcare, research, and applications*, vol. 11601, pp. 153–158, 2021.

[14.] D. Brezeale and D. J. Cook, "Automatic video classifcation: a survey of the literature," *IEEE Trans Syst Man Cybern C*, vol. 38, no. 3, pp. 416–430, 2008.

[15.] J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Mach. Learn. Appl.*, vol. 6, no. 100134, p. 100134, 2021.

[16.] H. S. Chang, S. Sull, and S. U. Lee, "Efcient video indexing scheme for content-based retrieval," *IEEE Trans Circuits Syst Video Technol*, vol. 9, no. 8, pp. 1269–1279, 1999.

[17.] V. Chasanis, A. Likas, and N. Galatsanos, "Efcient video shot summarization using an enhanced spectral clustering ap-proach," in *International conference on artifcial neural networks*, Springer, pp. 847–856, 2008..

[18.] P. Chauhan, H. L. Mandoria, and A. Negi, "Deep residual neural network for plant seedling image classifcation," in *Agricultural informatics: automation using the IoT and machine learning*, pp. 131–146, 2021.

[19.] P. Chauhan, H. L. Mandoria, A. Negi, and R. S. Rajput, "Plant diseases concept in smart agriculture using deep learning," in *Advances in Environmental Engineering and Green Technologies*, IGI Global, pp. 139–153, 2021.

[20.] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 1251–1258, 2017.

[21.] C. Chootong, T. K. Shih, A. Ochirbat, W. Sommool, and Y.-Y. Zhuang, “An attention enhanced sentence feature network for subtitle extraction and summarization,” *Expert Syst. Appl.*, vol. 178, no. 114946, p. 114946, 2021.

[22.] W.-S. Chu, Y. Song, and A. Jaimes, “Video co-summarization: Video summarization by visual co-occurrence,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp 3584–3592, 2015.