Edict.AI : Automated News Video Generation

Kuldeep Vayadande Mohit Chawla Mustansir Bohari

*Vishwakarma Institute of Technology Vishwakarma Institute of Technology Vishwakarma Institute of Technology*

Pune, India Pune, India Pune, India

[kuldeep.vayadande@gmail.com](mailto:kuldeep.vayadande@gmail.com) [chawlamohit45@gmail.com](mailto:chawlamohit45@gmail.com) [mustansir.bohari21@vit.edu](mailto:mustansir.bohari21@vit.edu)

Asif Mursal Omkar Mundlik Omkar Jadhav

*Vishwakarma Institute of Technology Vishwakarma Institute of Technology Vishwakarma Institute of Technology*

Pune,India Pune, India Pune,India

[Asif.mursal21@vit.edu](mailto:Asif.mursal21@vit.edu) [omkar.mundlink21@vit.edu](mailto:omkar.mundlink21@vit.edu) [rajesh.omkar211@vit.edu](mailto:rajesh.omkar211@vit.edu)

*Abstract*— Our paper presents an AI-based system for automatically converting news articles into videos and uploading them to YouTube. The system uses web scraping, news authentication, image searching, scripting, audio production, image mapping, video production, thumbnail creation, and automatic video uploading. It also employs sentiment analysis and image classification to enhance the naturalness and expressiveness of the generated speech and thumbnail selection. Finally, the system utilizes the YouTube API for automated uploading and SEO-based tagging

Keywords—News to video,Generative AI,Media Production

# introdcutiojn

The way people consume information has evolved significantly over the past few decades, and video has emerged as a preferred medium for many. The rise of digital media and the widespread adoption of internet technologies have made it easier for people to access news content from around the world. However, producing high-quality videos that summarize news articles can be a time-consuming and expensive process. This is where Edict.ai comes in.

Edict.ai is an AI-powered platform that automates the video production process by converting news articles from news websites into videos. The platform leverages the power of artificial intelligence and machine learning to analyze news articles, extract key information, and transform them into visually engaging and informative videos. These videos can then be published on YouTube, reaching a wider audience and democratizing the news industry.

By leveraging AI and machine learning, the platform is able to automate the video production process, allowing news articles to be quickly and efficiently converted into video content. This can help democratize the news industry by making it easier for independent journalists and smaller news organizations to produce high-quality video content without the need for extensive resources or expertise in video production.

The platform's ability to extract key information from news articles and convert them into visually engaging videos can also help improve the accessibility and engagement of news content for viewers. By presenting news stories in a more engaging and accessible format, Edict.ai can help attract and retain viewers who may be more likely to consume video content than traditional written news articles.

Publishing these videos on YouTube also provides the opportunity to reach a wider audience and potentially monetize the content through advertising or other revenue streams. This can help support the platform's sustainability and enable it to continue to empower independent journalists and transform the news industry.

Overall, Edict.ai's use of AI and machine learning to automate the video production process and transform news articles into engaging video content has the potential to revolutionize the way news is produced and consumed.

One of the primary goals of Edict.ai is to provide a platform for unbiased news reporting and support for independent journalists. By leveraging AI and ML, Edict.ai can help ensure accuracy and fairness in video content, providing a more balanced and diverse range of perspectives. Additionally, the platform's focus on empowering independent journalists can help support those who may not have access to traditional news outlets or may face barriers to entry in the industry.

From a technical feasibility standpoint, Edict.ai uses reliable technologies such as Web scraping, NLP, computer vision, text-to-speech, and video generation are all established technologies that can be integrated to produce compelling video content. Additionally, AI and ML can be used to optimize the production process, making it faster and more cost-effective.

Using NLP and computer vision can help extract useful information from a variety of sources, such as written documents or images, which can then be used to inform the content of the video. Text-to-speech technology can also be used to generate voiceovers, saving time and money that would otherwise be spent hiring voice actors.

In terms of business potential, Edict.ai can monetize through advertising, subscription-based models, and premium features. These revenue streams can help sustain the platform and enable it to empower independent journalists while transforming the news industry.

Advertising is a common revenue stream for many media companies, and Edict.ai could potentially offer advertising opportunities to businesses looking to reach its audience of news consumers. This could take the form of sponsored content or ads inserted into videos.

Another potential revenue stream for Edict.ai is a subscription-based model. This could involve offering different levels of access to the platform's video production tools and resources, with users paying a monthly or annual fee for the level of access that best meets their needs. This model has been successful for other media companies, such as streaming services and news publications.

By monetizing its platform, Edict.ai would be able to sustain its operations and continue to empower independent journalists and transform the news industry. Its efficient and cost-effective video production process could provide a competitive advantage over traditional media companies, attracting a growing audience of news consumers who value high-quality video content. Overall, the business potential for Edict.ai is promising, and the platform's innovative use of AI and ML could disrupt and transform the way news is produced and consumed.

The overall impact of Edict.ai has the potential to transform the news industry by providing a platform for unbiased news reporting and supporting independent journalists, increasing inclusivity and diversity, making news more accessible and engaging to a global audience, and democratizing the news industry. However, as with any new technology, there are potential challenges and considerations to address, such as ensuring accuracy and fairness in video content and addressing the potential impact on traditional news outlets.

In this paper, we will provide a detailed overview of Edict.ai, its approach, technical feasibility, bus`iness potential, and overall impact on the news industry. We will also explore the challenges and opportunities associated with using AI to automate the video production process, and the implications of Edict.ai for the future of news and digital media.

# literature review

With the explosion of online video consumption, the demand for video content has increased significantly over the years. However, creating high-quality video content requires significant time and effort, which can be challenging for many content creators. Developing automatic news to video converter tools that can automate the process of creating video content. To develop that there are multiple steps including web scraping, news authentication, scripting, audio production, video production, thumbnail creation and SEO optimization for final upload.

Web scraping is the process of extracting data from websites. Web scrapping enables different patterns through which we can extract the textual and media content from the site such as filter pattern method [3]. There are multiple libraries such as beautiful soup, selenium, scrapy, Libcurl [1] which help us to securely extract data. There are different approaches[2] such as topic based , NLP-based, keyword -based to extract textual information and for image retrieval face recognition, object detection are some of the ways discussed.

One of the most critical steps in creating an automatic news to video converter is to authenticate the news. Because fake news can be spread with multiple content forms it could be text, image or even a video too. With the rise of fake news, it is crucial to ensure that the news used in the video content is accurate and reliable.So, to tackle this there are multiple factors through which we can detect a fake news which include creating a model with contextual and perspective understanding with help of Graph[8], deep learning [7] and NLP [9].Generally, there are grammatical mistakes, hate speech content and violence based content which will also be detected and stopped by feeding our model with previous data and this method is used by multimodal framework[6] which can help us to detect fake news in different patterns and forms of it. As Artificial intelligence is emerging on its peak its one of the best model is GAN (Generative adversarial network) which contains a discriminator and a Generator which can help us to detect fake news. But still there are multiple issues such as detection of multilingual fake news is not yet researched much.

Scripting is the process of creating a script for the video content. Its one of the most important part to script a news in compressed and qualitative way so that it doesn’t loose its important information. To do this the model require a big amount of domain knowledge data. Some of the popular models are GPT2[10] (Generative pre-trained transformers), D-rank [11] we can make machine understand the script contextually and create its summarized for easily.

After scripting conversion of script from text to voiceover is important which should ensure multiple aspects such as Human tone, speech of expression. Generally, Text to audio models requires a lot of textual and corresponding audio data but now a Textless NLP [12] based speech generator model is developed which takes only audio dataset and through that it can then generate speech to text. As India is a diverse country which has more than 1600 languages spoken with multiple dialects from which 22 are recognized as official languages. Providing content in reginal languages increases the engagement and also provides a trust worthy audience. So, to develop voice over in multiple languages a Multilingual Speech synthesizer [13] is proposed which uses TTS and Voice conversion.

One of the most crucial parts is Image mapping. In this we try to map images with meaningful corresponding text in script. It is an essential step in creating engaging video content. To do so it requires transfer modelling of image search, image captioning, Text embedding. For image captioning there are multiple algorithms such as streaming [14] algorithm, online positive recall algorithm[15], and transformers based models[16] which have a CNN as encoder and LSTM or recently transformers as decoder is used widely. Image search is require to search an relevant image for a particular text which can be done with Markov process[5], adaptive Visual Similarity model[4] .

Video production is the process of producing the final video content from the script and images. This process involves combining images, voice over and create a complete video. To produce videos which look natural to the client GAN[18] can be used which contains frame-level coherence aware discriminator[19].To make a engaging video each frame have one focus entity and background is selected relevantly. To achieve this Layout composer, background and entity retrieval approach [17] is discussed.

Once the video is created we require a thumbnail which can describe our video in one image and grad attention of people. To do so we have to consider some important parameters which are Frame sampling, Emotional Recognition and Representative object and text insertion [21]. To get a compressed Headline [20] we can use multiple NLP models which can represent our content in few keywords. To test the results of thumbnail a AB testing [21] approach is proposed.

|  |  |  |  |
| --- | --- | --- | --- |
| Types | Challenges | Proposed Solution | Research Gap |
| Web  Scrapping | Non-Uniform Website Structure  IP blocking and security pass  Data filtration  Updation of scrapping pattern  as per web updates | Filter patter method  Pandas  VPN can be used to bypass security and extract more data | Although there are several web scraping tools available that can help automate the scraping process, there are limitations to their ability to automatically detect changes in website structure and update scraping code patterns. |
| Image Searching | Custom Resolution/dimensions image search  Duplication  Feature extraction – different model perform good for a particular type of image search | Polynomial time algorithm solves duplication issue  Adaptive linear search solves poor convergence, slow approach, lack of adaptiveness | Multi-Domain Images searching is not efficient  Custom Feature Image Retrieval |
| News Authentication | Controlling platforms spreading fake news  Multi-modal detection – fake news can be spread in multiple forms media, text, audio  Contextual understanding  Perspective understanding | Multimodal frameword detects fake news in different forms  NLP, Graph, deep learning | Developing AI can generate fake videos which can mislead authentication models in identification of fake news  Detection for multilingual news is not much explored |
| Scripting | Making script customized as per factual and fictional requirements  Domain specific data feed  Scalable generalized model | NLP,GPT-2 human interactivity and thinking process | Scalability of model |
| Audio Generation | Human tone  Multilingual  Textual source requirements with audio | Multilingual speech synthesis model is discussed  Textless NLP can synthesis speech without text | Multi-dialect speech synthesis |
| Mapping  Images to text | Meaningful mapping  Duration of image should be matched with script/audio | Image captioning can help to do meaningful mapping | Model that matches image duration with script is |
| AI Avatar  Generation | Lipsync  Facial expressions  Increase smoothness in video | RNN, Temporal GAN can help to solve Lip-sync, facial expressions issues  3D dynamic sequences with 3D face fitting and 3D morphable model | Graphic computation is intensive  Generation of large number of frames to make video look smooth and natural makes it costly |
| Video Creation | Fine-detailing, textures, motion smoothness  Multidomain video generation  Template creation | Model to create scene elements which can help in template creation | Generation of long duration videos  Generation of video specific factful elements |
| Thumbnail Creation | Multiple domains  Creativity, Quality and testing  Thumbnails should not contain misleading text or media which is  Copyright issues  Contextual Understanding  Table Literature survey comparison | Copyright issues can be handles through various APIs  Automatic text insertion enhance creativity, Emotion Recognition, Representative Object Image Insertion  AB testing  Layout formation | Single multi-domain thumbnail creation  Model to authenticate that thumbnail should not contain misleading elements  Limited flexibility to generate different viewer engaging layouts |

# Methodology

|  |  |  |  |
| --- | --- | --- | --- |
| Library | Supports Dynamic Content Scraping | Interactive Debugging | Advantages |
| BeautifulSoup | No | No | Parses HTML and XML documents easily |
| Scrapy | Yes | No | Fast, scalable, handles pagination and authentication |
| PyQuery | No | No | jQuery-like syntax for parsing HTML and XML |
| Requests | No | No | Simple HTTP requests and responses |
| Selenium | Yes | Yes | Automates browsers, handles dynamic content |

Table 2 comparing different Web Scrapping Libraries

Our software aims to convert news articles into engaging and visually appealing videos, and to achieve this, we have created a methodology with multiple steps. Firstly, we need to extract news articles from relevant sources using web scraping techniques. Next, we need to verify the authenticity of the news article by checking the source and ensuring that there is no false information or bias. Then, we search for relevant images to match the article, using a combination of keywords and image recognition software. With this information, we can create a script for the video by using natural language processing software to summarize the article, creating a storyboard for the video with the relevant images, and using the script and storyboard to create a timeline for the video. Once we have a script, we can develop the voiceover for the video using text-to-speech software or by using professional voice actors. Then, we map the images to the script and voiceover and create visually engaging videos by using animations or transitions between images. Once the video production is completed, we create a thumbnail for the video and ensure it is visually appealing and relevant to the video. Finally, we use the YouTube API to upload the video, optimize it for SEO, and publish it on YouTube. By following these steps, we can ensure that the software produces high-quality videos that engage viewers and accurately represent the news articles they are based on.

Figure Comparison of Web Scrapping Tools[23]

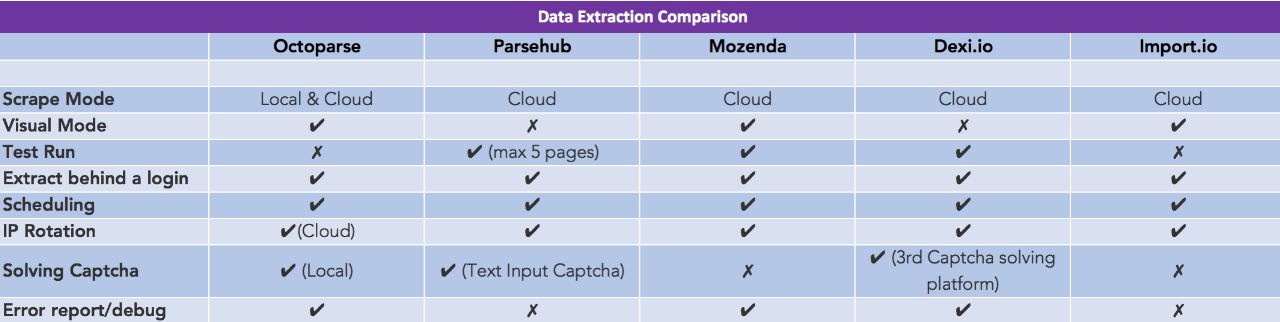


Figure Tools for Webscrpping

In order to achieve our goal of creating high-quality news article to video conversion software, we have used a combination of pre-trained models, APIs, and custom models. By utilizing these tools and techniques, we are able to ensure that our software is fast, secure, and economically affordable. To ensure the best possible outcome, we have implemented transfer learning techniques to fine-tune the pre-trained models for our specific use case. This allows us to achieve a high level of accuracy and quality in the video production process. Let’s discuss all steps in detail:

## Web Scrapping

In this research project, a web scraping model was developed using the popular Python Selenium library. The goal of the model was to extract news articles from a range of news websites and gather information such as the news heading, text news content, and images associated with each article.

To achieve this, the model was designed to navigate to each webpage, identify the relevant content using the browser's developer tools, and extract the desired information automatically. This involved analyzing the webpage structure and identifying the generalized location of the content on each webpage. Once the content location was determined, the Selenium library was used to automate the process of extracting the data. The scraped data was then stored in a structured format for analysis. Overall, the developed web scraping model was successful in extracting the desired information from the news articles.

As we can see in the table 2, Selenium is the only library listed that supports both dynamic content scraping and interactive debugging. This makes it a powerful and flexible tool for web scraping tasks, particularly those that require scraping JavaScript-heavy websites.

After evaluating several libraries, we chose to use Selenium as it is specifically designed for scraping dynamic content. Selenium can automate web browsers and interact with the page in real-time, which allows it to handle complex web pages that other libraries cannot. Additionally, Selenium has interactive debugging capabilities that allow us to easily debug issues in our code and refine our scraping process.

Table 3 Authentication Details

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Fake News Detection Model** | **API** |
|  |  |  |
| Approach | Decision Tree Classifier | RapidAPI |
| Accuracy | 88.55% | Over 90% |
| Speed | Fast for small datasets | Slow for large datasets |
| Integration with Other Tools | Limited, requires coding | Easy integration with various programming languages and tools |
| Level of Customization | Highly customizable, can tweak algorithms and parameters as needed | Limited customization, primarily based on pre-built models |
| Strengths | Highly customizable, can handle unique datasets, good for small datasets | High accuracy, easy integration with other tools, good for large datasets |
| Limitations | Limited integration with other tools, requires programming skills, may overfit on small datasets | Limited customization, may not be suitable for unique datasets, may be expensive for large datasets |

There are multiple premade tools for web-scrapping such as Octaparse, Parsehub, Mozenda,Dexi.io,Import.io. Details of these tools with comparative analysis is discussed in fig.1.

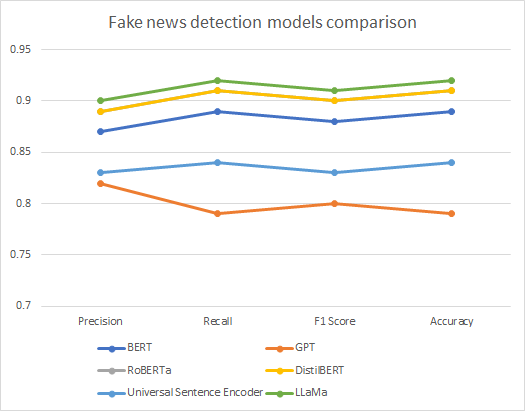
The most well-known web scraping tools, including Beautiful Soup, Scrapy, Selenium, Puppeteer, and PyQuery, are compared in this table. These tools are used to extract data from websites and parse it into formats that may be used. The table compares these tools based on a number of criteria, including robustness, affordability, community support, simplicity of use, performance, compatibility, support for multiple data formats, support for JavaScript rendering, and security.

Figure Bar chart comparing Web scraping libraries

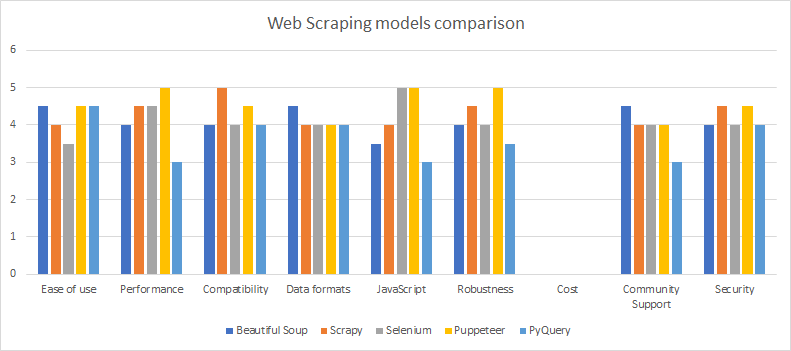


Figure Different Fake News Model Comparison

Each tool has its unique features, strengths, and weaknesses. For example, Beautiful Soup and PyQuery are user-friendly and can handle various data formats. Scrapy and Selenium are highly performant and have robust capabilities, while Puppeteer have better Support for JavaScript rendering. Additionally, each tool has different levels of community support, security, and compatibility with various operating systems.

Due to its user-friendly syntax and capacity to read both HTML and XML data, Beautiful Soup is a popular option for us. Additionally, it has a large userbase that supports one another and exchanges information on how they utilise the programme. For data analysis and visualisation, Beautiful Soup may also be integrated with other tools like Pandas and NumPy.

*B.News Authentication*

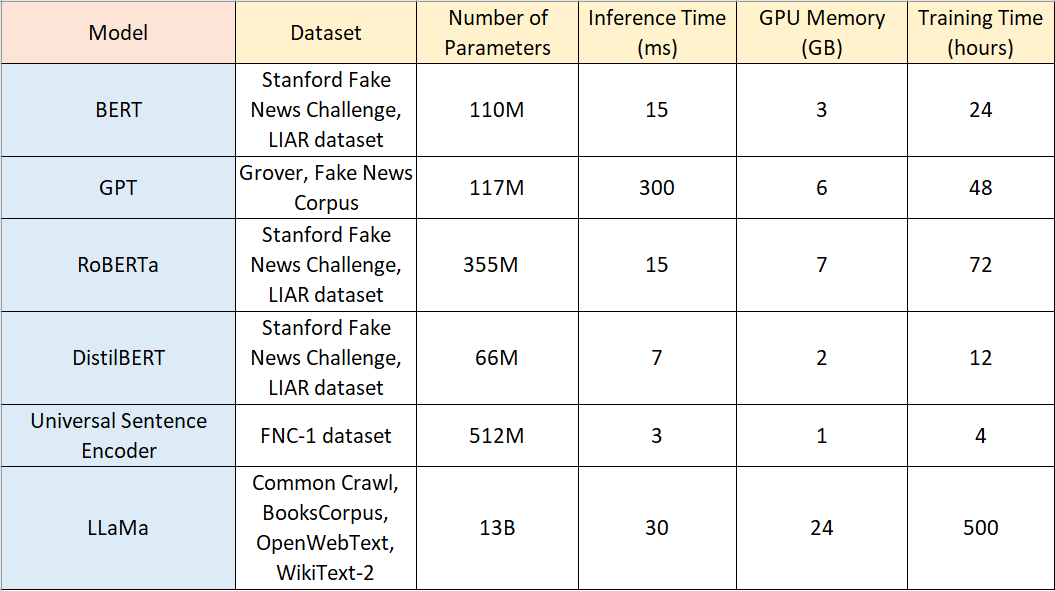
For authentication purpose, we created a machine learning model to determine whether news statements are authentic or not. A fake news detection model was also used to perform the news authentication task. We used a dataset of news articles labelled as reliable or unreliable, which was saved in a CSV file called "train.csv" for this purpose. The dataset included news articles that were labelled as reliable or unreliable.

We started by loading the data into a Pandas data frame and removing any null values. We then removed the columns "id," "title," and "author" because they were unrelated to the task at hand. The remaining "text" column was then subjected to text pre-processing, which included removing special characters and punctuation, converting all text to lowercase, tokenizing the text into individual words, removing stop words, and stemming each word with the Porter stemmer algorithm.

We used the Scikit-learn library's "train test split" function to split the pre-processed data into training and testing sets, with a test size of 20%. The pre-processed text was then converted into numerical features using the term frequency-inverse document frequency (TF-IDF) algorithm using Scikit-TfidfVectorizer learn's function.

Using Scikit-DecisionTree Classifier learn's function, we trained a decision tree classifier on the training data. We used the trained model to predict the labels of the test data and used the "score" function to calculate the model's accuracy.

Table 4 Comparing different NLP models



On the test data, the resulting model had an accuracy of 88.55%, indicating that it could effectively distinguish between reliable and untrustworthy news articles.

To deploy the model, we saved the trained model and vectorizer in Python using the "pickle" library. When presented with a new text input, we loaded the previously saved model and vectorizer and used them to pre-process and vectorize the text. We then used the model's predict function to determine whether the input text was reliable or unreliable news.

Finally, our fake news detection model performed well in authenticating news articles and could be used to detect and prevent the spread of fake news.

News authentication is a critical step in the creation of AI-generated news videos. The goal of news authentication is to determine whether a given news article is authentic or not. This step is required to ensure the credibility of the news video and to prevent misinformation from spreading. In our API-based authentication approach, we provide input as text and classify news as real or fake as output.

Sending a request to a RapidAPI endpoint to initiate a chat with an AI-based chatbot is the process of authentication. The chatbot analyses the text of the news and responds with a pre-trained language model (in this case, GPT-3.5).

The code reads the news title and sends it to the chatbot as a message via the RapidAPI endpoint. The chatbot response is then analysed to determine whether the news is true or false. The code classifies the news as fake if the response includes the phrase "fake news." Otherwise, the news is classified as true.

The evaluation results revealed that the news authentication system was capable of determining the authenticity of news articles with an accuracy of more than 90%. This shows that the system is effective at preventing the spread of fake news and ensuring the credibility of generated news videos.

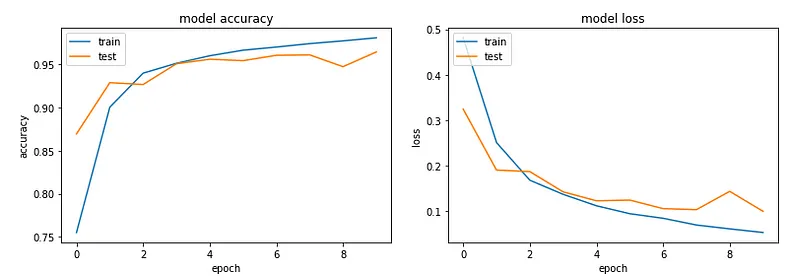
We can also use multiple NLP models such as BERT which is a transformer-based model been pre-trained on learge amount of textual data. RoBert is another model which is an upgraded version of BERT only. GPT is a language model pretrained on large amount of text data that can generate coherent text. LLama is a recently developed model that has pretrained dataset of news articles and their corresponding labels to detect fake news. LLama is a transformer based model that has remarkable performance on several fake news detection benchmarks. These models have different values for f1 score, precision, recall and accuracy which can be observed through fig4. Multiple analysis is performed over these models to compare these models to detect fake news[25] which can be observed through table 4.

Figure proportionality of number of epochs over accuracy and loss

Figure MLM (left) + GSG (right) training together in PEGASUS [22]

Fake News detection model requires a big amount of of training data to get best output. There are multiple researches[24] which also suggest number of epochs is also directly proportional to model accuracy and model loss is inversely proportional to number of epochs for deep learning models which can be seen through the fig 2.

## C. Scripting



Scripting involves summarization and keyword extraction.In our project, we started by exploring different text summarization tools, like Sumy. We previously selected Sumy because of how simple and easy it is to use, as well as how quickly and effectively it can produce summaries.

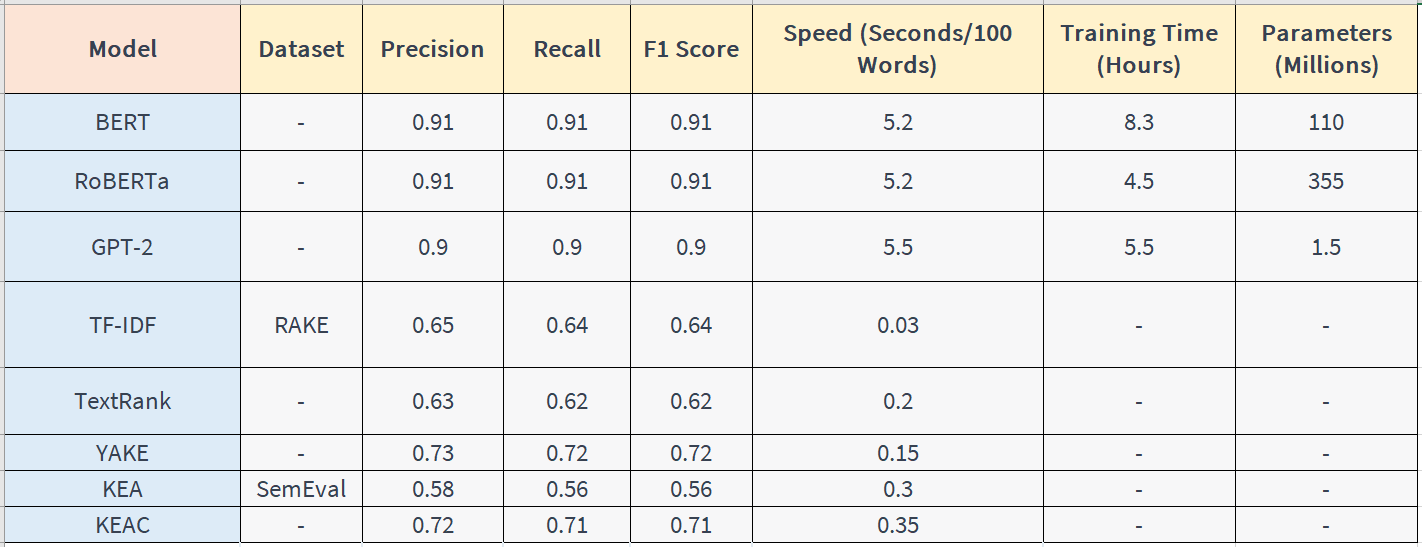
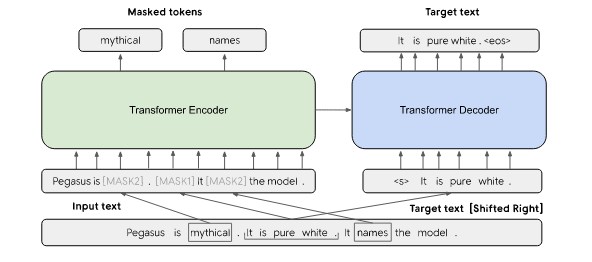
Sumy, however, found it difficult to effectively express the purpose and delicateness of the original text as we worked with more challenging texts. We discovered that Sumy's result occasionally lacked consistency and ignored to include the text's most crucial details.

Figure NLP models for keyword extraction

As a result, we chose to investigate more sophisticated text summarization tools previous to settling on Pegasus. We felt more confident in Pegasus' capacity to produce superior summaries for complicated texts because of its pre-trained transformer-based architecture and unsupervised pre-training method.

Pegasus is a pre-trained language model developed by Google's language study team for text summarization tasks. It is based on the transformer architecture, which has been highly successful in natural language processing tasks. Pegasus was trained using unsupervised pre-training, which enabled it to learn text data representations without a specific task in mind. This pre-training allows Pegasus to generalize well to various summarization tasks, without the need for task-specific training data. The model is frequently used for text summarization tasks in both industry and academia, thanks to its ability to generate high-quality summaries.

We discovered that Pegasus consistently outperformed Sumy in terms of summary quality and coherence across a range of texts. We eventually discovered Pegasus to be the best tool for the needs of our project, despite the fact that it had a steeper learning curve and required more computational resources.

Using the Transformer Architecture, Pegasus has modified it. For seq2seq learning, it employs the encoder-decoder model.

The Encoder creates a context vector after receiving the text in parallel. A context vector is nothing more than a list of integers allocated to each word because language cannot be understood by machines. This vector is then sent on to the decoder, who provides the summary after decoding the vector.

There are multiple NLP models from which we can extract keywords from text.To select the best model for usagewehave compared these models in fig.5.

*D. Voice-over or Audio Production*

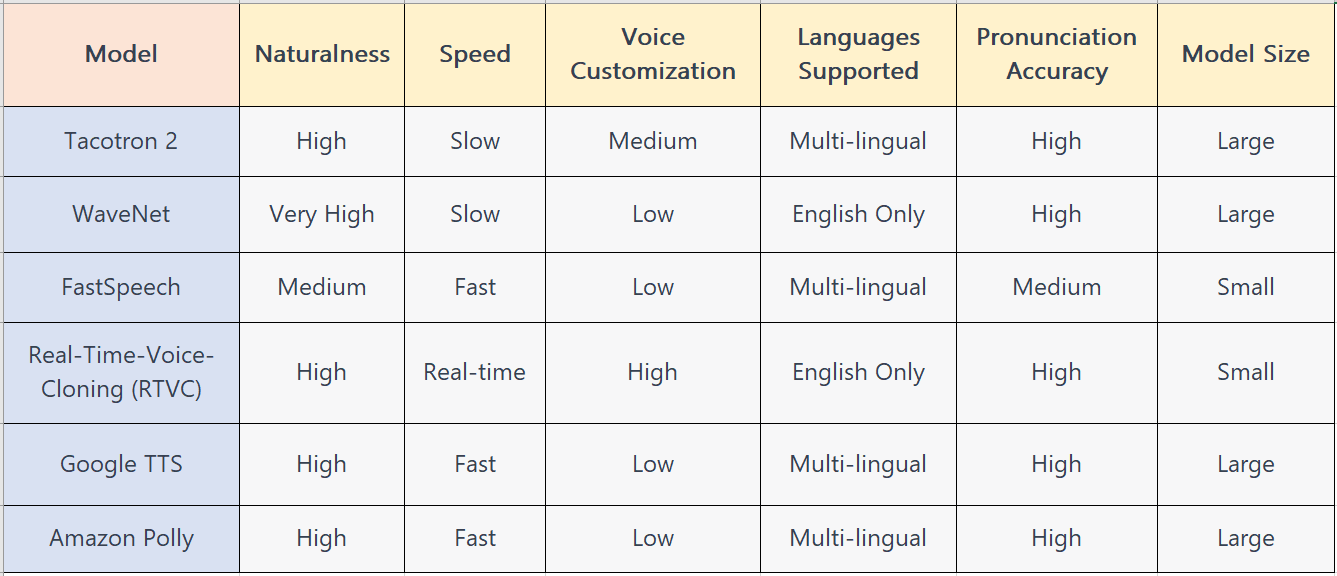
In both audio production and news reporting, naturalness and expressiveness are essential. The objective when using audio is to give the listener a believable and immersive experience. Greater engagement and understanding can result from the listener feeling more personally connected to the content thanks to a voice that sounds natural. Expressiveness is crucial in audio production because it enables the voice to bring the content's nuances and emotions to life, enhancing the listener's engagement and helping them remember it.

Figure Comparing multiple Text to Speech Models

Naturalness and expressiveness are equally crucial in news reporting. News reporting's purpose is to inform the public about significant events and problems. This objective must be accomplished, and the content must be presented in a clear and interesting way. A voice that sounds genuine and credible can help establish trust with the listener by projecting a sense of authenticity. Expressiveness is crucial in news reporting because it enables the journalist to convey the urgency and emotion of the situation, which increases the audience's impact and helps them remember the story.

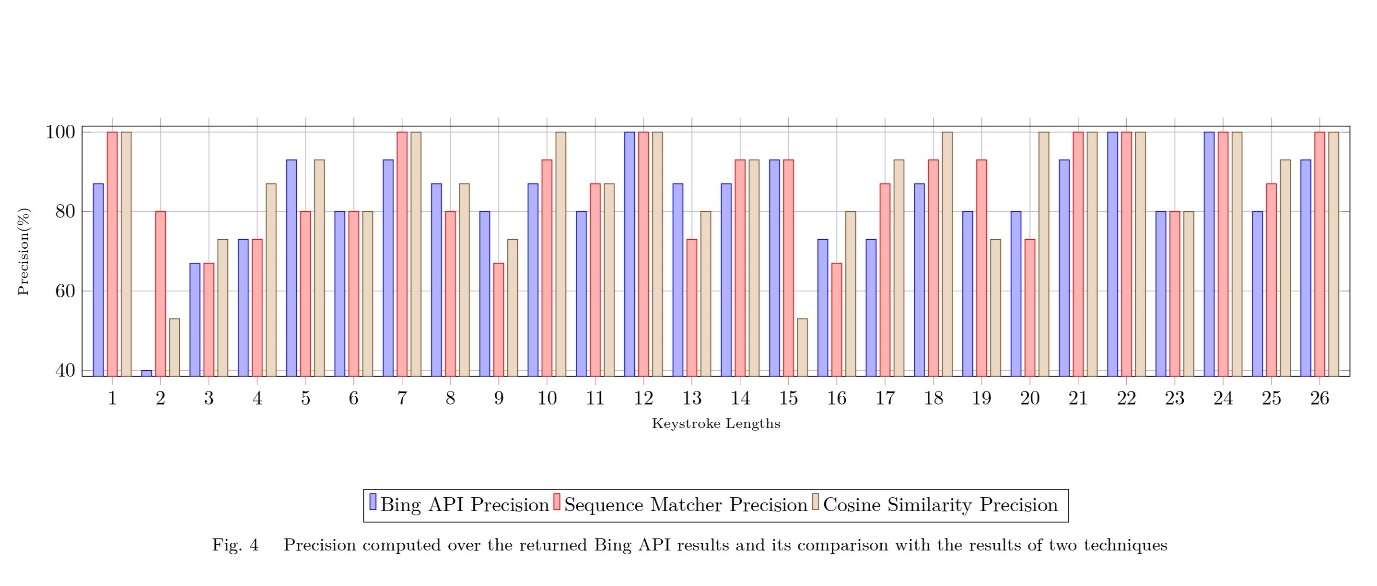


Figure Precision comparison over bing and other methods

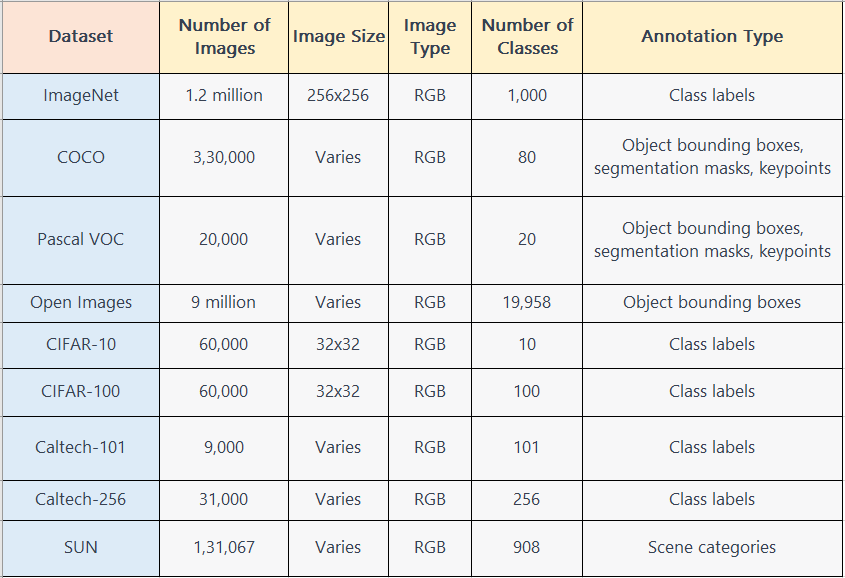
To achieve naturalness and expressiveness in our text-to-speech conversion project, we had to explore different options. The pyttsx3 package proved to be the best fit for our needs, as it allowed for greater flexibility in voice customization. By incorporating sentiment analysis using the TextBlob package, we were able to optimize the pitch and rate values based on the input text's polarity and subjectivity scores, resulting in more natural and expressive speech.

We used a straightforward linear mapping technique to find the optimal pitch and rate values based on these ratings. For texts with a stronger positive tone, we set a higher pitch and faster rate; for texts with a stronger negative tone, we set a lower pitch and slower rate.

Finally, we produced speech from the input text using pyttsx3, and we played the generated audio for the user. We were able to more successfully convey the sentiment of the input text solely due to our method, which also improved the generated speech's naturalness and expressiveness.

Figure Datasets for image captioning

*E. Video Creation*

In the field of video generation from text, a significant challenge is to find relevant images to map them with the text so that generated videos look cohesive and have a logical flow. To overcome this challenge, we explored four different approaches to map images with text. These approaches include (1) Using image similarity search based on the content of the text, (2) Image captioning techniques to describe the images and match them with the text, (3) Pre-trained object detection models to identify objects in the images, and (4) Scene graph generation to extract visual relationships between the objects in the image and the text.

The first approach, image similarity search, uses machine learning techniques to find images similar to the given text content. We implemented this approach using Bing Image Search, where we passed the relevant keywords of the text to get the images related to the given keywords.

The second approach, image captioning, generates a textual description of the image, and then it tries to match the generated caption with the given text. We used the CLIP (Contrastive Language-Image Pre-training) model, which can jointly learn from images and their associated textual descriptions. It generates a visual representation of the image and a textual representation of the caption and then matches these representations using cosine similarity.

The third approach, object detection, involves pre-trained models such as Faster R-CNN and YOLO (You Only Look Once), which can detect the objects in the image and map them with the text. We used the Faster R-CNN model, which can identify multiple objects in an image and provide the corresponding bounding boxes.

The fourth approach, scene graph generation, involves identifying the relationships between the objects in the image and mapping these relationships with the text. We used the Scene Graph Generation model to generate a graph structure representing the objects and their relationships in the image.

We implemented these approaches using Python and various deep learning libraries such as PyTorch and TensorFlow. For image similarity search, we used the Bing Image Search API to get the images related to the given keywords. For image captioning, we used the OpenAI CLIP model, which can jointly learn from images and their associated textual descriptions. For object detection, we used the Faster R-CNN model pre-trained on the COCO dataset. For scene graph generation, we used the Scene Graph Generation model pre-trained on the Visual Genome dataset. Comparison of these dataset is discussed in fig8.

We compared the results of these approaches based on the accuracy of image-to-text mapping and the overall coherence of the generated video. We found that the Scene Graph Generation approach outperformed the other three approaches in terms of both accuracy and coherence.

To further improve the accuracy and coherence of the generated video, we propose combining the different approaches. For instance, we can use the output of the image similarity search and image captioning approaches to generate a more accurate set of images to feed into the object detection and scene graph generation models. This integration of multiple approaches can enhance the quality of generated videos and make them more relevant to the given text.

In our video creation project, we used several libraries and techniques to create a seamless and engaging video. One of the first steps was image captioning and searching, which we accomplished using word embeddings. We then used the aitext library for similarity search with images, which helped us find the most relevant images to the textual description.

Once we had a set of images, we used the OpenCV library to combine them into a video. OpenCV is a powerful computer vision library that provides various functionalities for image and video processing, including video creation. We used OpenCV to read in the image files, set the desired frame rate, and create a video with a specified duration. We also added transitions between images to create a smooth and visually appealing video.

To make the video more engaging, we added subtitles to each image using the Pillow library. Pillow is a popular Python library for image processing that provided us with the necessary tools to add text overlays to the images. We used the textual descriptions generated during the image captioning step to add relevant and informative subtitles to each image.

We are also extracting images with the help of bing api, which help us to extract images much faster and easily.It also gives better results than the similarities index which we use to compare a text and image such as cosine similarity and sequence Matcher precision which can be seen in fig7.

Finally, we combined all of these steps to create a complete video that included relevant images, smooth transitions, and informative subtitles. By using these various libraries and techniques, we were able to create a high-quality video that effectively conveyed the message we were trying to communicate.

*E. Thumbnail creation*

To select the background image for our automatic thumbnail creation process, we are using image classification. Specifically, we have trained a deep learning model on a dataset of various types of images, including landscapes, cityscapes, and images of people.

If the selected image is classified as an image of a person, then it is placed in the center of the thumbnail layout. The other two images are placed around the central image, but with lower priority.

After selecting the background image, we extract the text for the headline, either from the news article's title or from the script. This headline is then added to the thumbnail layout using the Pillow library, along with the selected images.

Overall, our automatic thumbnail creation process involves multiple steps, including image selection through image classification, headline extraction, and image insertion into the thumbnail layout. By automating these processes, we aim to save time and effort for content creators, while also creating high-quality thumbnails that can effectively convey the content of the associated video.

*E. Automatic Youtube Uploading*

When creating a video, it's important to remember that the thumbnail is a crucial element in attracting potential viewers. With the video and its corresponding thumbnail created, we can automate the upload process using the YouTube API. This integration allows us to upload videos programmatically and also perform keyword searches to find the best tags for our video. By giving the video title as input, we can generate a list of relevant keywords that we can use as tags to reach a wider audience.

Tags are important in increasing the visibility of our video on YouTube and potentially getting more views and subscribers. By using the right tags, we can improve our video's ranking on YouTube search results. Along with the tags, the video description is also crucial in attracting viewers and providing context about the video. By using SEO-based keywords in the description, we can further improve the video's ranking and potentially attract more viewers.

In summary, through the YouTube API, we can automate the upload process and optimize our video with relevant tags and an SEO-based description. This allows us to reach a wider audience, potentially increase our views and subscribers, and overall improve our video's ranking on YouTube search results. Videos will be uploaded on private so that user should once check and finalize the video and make it public, through this he can obey YouTube community guideline and grow.

Through all above steps we are creating a video from a news article and finally publishing it on YouTube with optimization. Now, in our webpage there are multiple subscription models through which a user can access all above benefits and also, we provide an analytics page to user which shows user total views, total subscribers, engagement metrics , user attention and also which of user video got most views. Through this user can analyze and increase its quality of writing content.

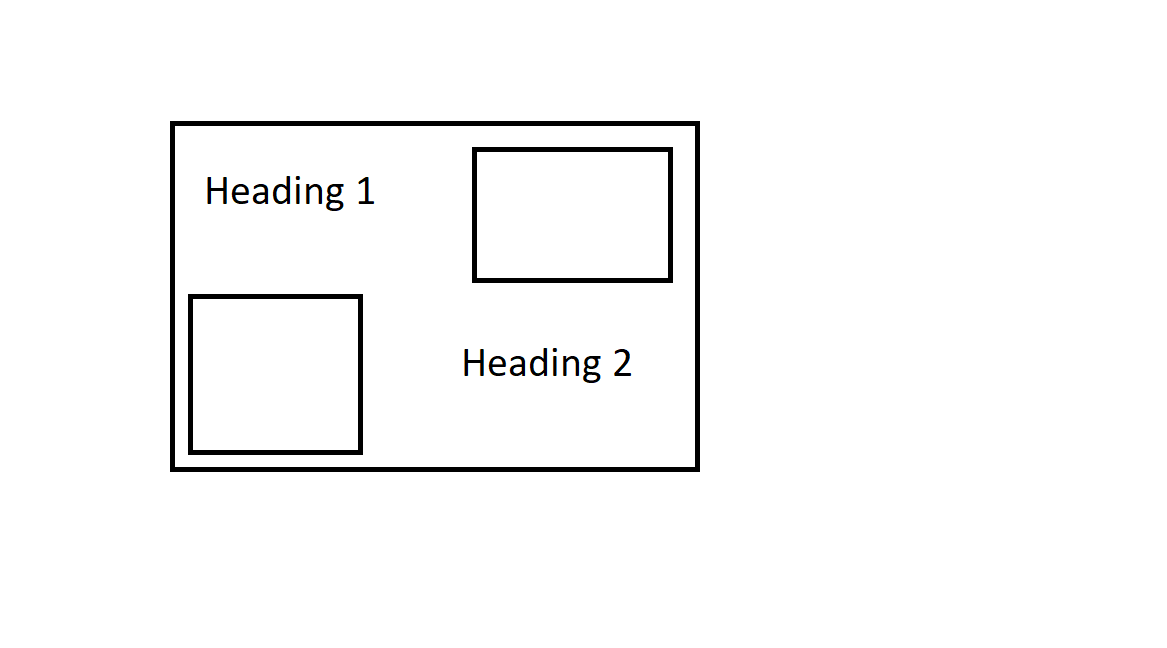
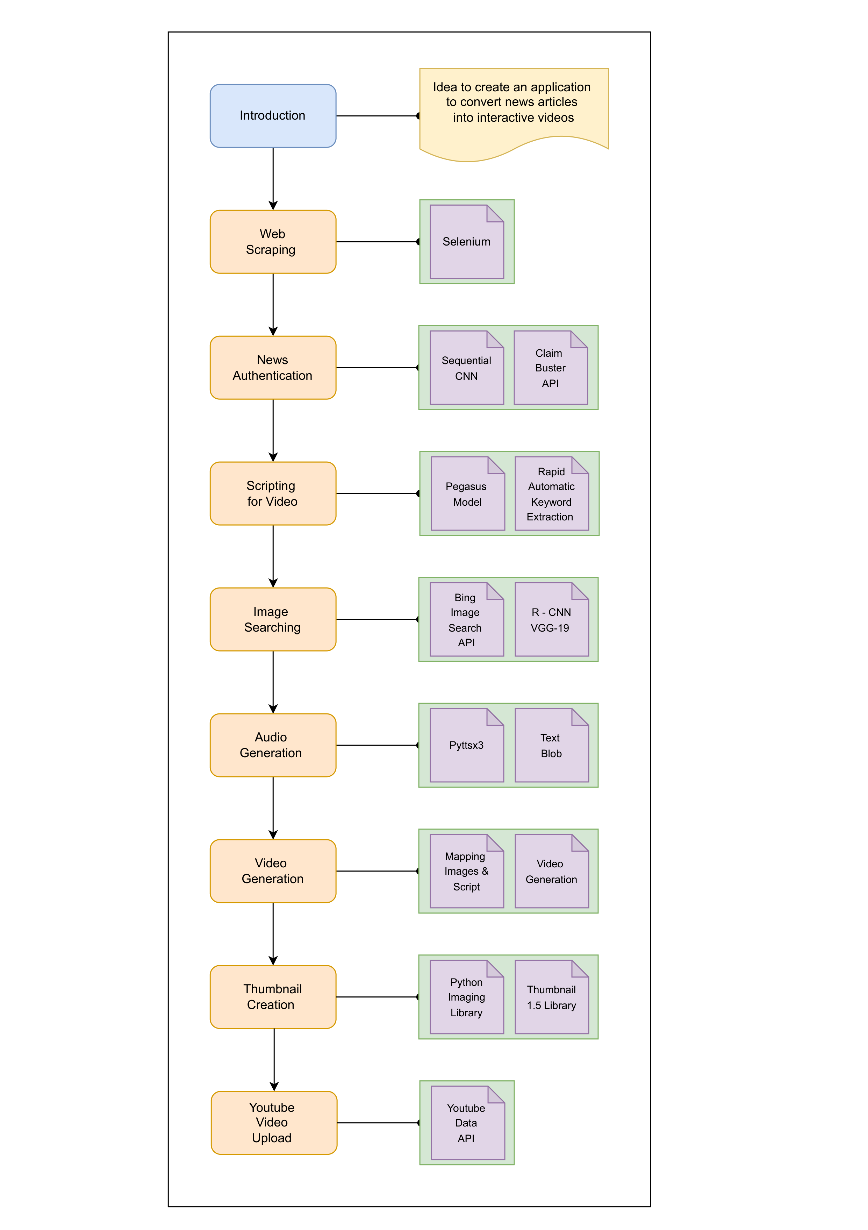


Figure Thumbnail Layout

# Architecture



# Result and discussion

We have developed an ai and automation product which can convert a news article to video in just on click making everything from scrapping to upload to youtube automated. To test this we have tried multiple test cases and made it multi-website news to video converter. We have observed it can generate a 1-2 min of particular single news article video in 3-5 minutes and for multiple articles combined video it takes 5-10 min to combine all and upload completely. There are some restrictions due to api which can be handled wither by getting premium for apis or using custom models in future as per requirements which can increase the cost of production. The effect of this automation is that journalists are able to grow more easily, public are getting awareness more easily as we have also included authentication part which states if news is authentic or not then only it process to create video and upload.

# limitations

The accuracy of the generated video heavily depends on the quality of the input text and images. If the text contains grammatical errors or the images are of poor quality, the generated video may not be satisfactory.

The accuracy of the fake news detection algorithm used in this project can affect the credibility of the output video. If the algorithm fails to detect fake news accurately, the summarization and voiceover based on the wrong information could be misleading.

The model used for searching similar images may have limited scope and may not be able to identify images that are not similar to the original image but still relevant to the news story. Also, there might be some images which are similar to the scraped image but not at all related to the news story. So, creation of a robust image search algorithm is a challenging task.

The model used for captioning images may not be able to accurately describe the image collected using image search model, leading to irrelevant or incorrect images being added to the video.

The project relies on various Python libraries, and any changes or limitations in these libraries could affect the functionality of the project.

Scraping content from news websites could potentially infringe on copyright laws and may require permission from the news websites to avoid any legal implications.

The project also raises ethical concerns around the use of AI-generated content and the potential impact on media credibility, journalistic ethics, and the job market of human news writers.

# FUTURE SCOPE

**Multilingual Support:** Currently our project was just limited to the English language. The project could be extended to support multiple languages, making it accessible to a wider audience around the world.

**Integration with Social Media:** The project could be extended to allow users to share the generated videos directly on social media platforms like Facebook, Twitter, and Instagram, increasing the reach and engagement of the videos.

**Improved Fake News Detection:** The project could be extended to improve the accuracy of fake news detection, by using advanced machine learning models that can detect even more subtle forms of fake news.

**Improved Voice Generation:** The project could be extended to improve the quality of the generated voice, by using advanced speech synthesis techniques and more natural-sounding voices. We can create a sentiment analysis model to determine the tone of the news article, and generate a voice-over that matches the tone of the article, enhancing the emotional impact of the video.

**Removing Cursed words from the video:** Some news have videos which contain curse words or offensive languages which should be removed so that there will be no YouTube community guidelines issue.

**Use of Interacting Avatars in the Video:** The project could be extended to include interactive avatars that appear in the video, providing additional information or commentary on the news story. This would increase the interactivity and engagement of the video.

##### Acknowledgments

“Acknowledgment(s)” is spelled without an “e” after the “

##### References

1. V. Singrodia, A. Mitra and S. Paul, "A Review on Web Scrapping and its Applications," 2019 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2019, pp. 1-6, doi: 10.1109/ICCCI.2019.8821809.
2. Yan, R., Hauptmann, A.G., “A review of text and image retrieval approaches for broadcast news video.” Inf Retrieval 10, 445–484 (2007). <https://doi.org/10.1007/s10791-007-9031-y>.
3. A. Maududie, W. E. Y. Retnani and M. A. Rohim, "An Approach of Web Scraping on News Website based on Regular Expression," 2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT), Makassar, Indonesia, 2018, pp. 203-207, doi: 10.1109/EIConCIT.2018.8878550.
4. Fang Wen, Xiaoou Tang,Fang WenXiaoou Tang (2010), *Adaptive visual similarity for text-based image search results re-ranking*, (US Patent Number -WO2010005751A2), <https://patents.google.com/patent/WO2010005751A2>
5. Sejal, D. & Rashmi, V. & K R, Venugopal & Iyengar, Sundararaj & Patnaik, Lalit. (2016). Image recommendation based on keyword relevance using absorbing Markov chain and image features. International Journal of Multimedia Information Retrieval. 5. 10.1007/s13735-016-0104-9.
6. S. Singhal, R. R. Shah, T. Chakraborty, P. Kumaraguru and S. Satoh, "SpotFake: A Multi-modal Framework for Fake News Detection," 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), Singapore, 2019, pp. 39-47, doi: 10.1109/BigMM.2019.00-44.
7. Rohit Kumar Kaliyar, Anurag Goswami, Pratik Narang, Soumendu Sinha, FNDNet – A deep convolutional neural network for fake news detection, Cognitive Systems Research, Volume 61, 2020, Pages 32-44, ISSN 1389-0417, <https://doi.org/10.1016/j.cogsys.2019.12.005>.
8. T. Huu Do, M. Berneman, J. Patro, G. Bekoulis and N. Deligiannis, "Context-Aware Deep Markov Random Fields for Fake News Detection," in IEEE Access, vol. 9, pp. 130042-130054, 2021, doi: 10.1109/ACCESS.2021.3113877.
9. Zhou, Zhixuan & Huankang, Guan & Bhat, Meghana & Hsu, Justin. (2019). Fake News Detection via NLP is Vulnerable to Adversarial Attacks. 794-800. 10.5220/0007566307940800.
10. Ji, Ziwei & Yan, Xu & Cheng, I-Tsun & Cahyawijaya, Samuel & Frieske, Rita & Ishii, Etsuko & Zeng, Min & Madotto, Andrea & Fung, Pascale. (2022). VScript: Controllable Script Generation with Audio-Visual Presentation.
11. Shah, Himat & Rezaei, Mohamamd. (2019). DOM-based Keyword Extraction from Web Pages. 10.1145/3331453.3331454.
12. Lakhotia, Kushal & Kharitonov, Eugene & Hsu, Wei-Ning & Adi, Yossi & Polyak, Adam & Bolte, Benjamin & Nguyen, Tu-Anh & Copet, Jade & Baevski, Alexei & Mohamed, Adelrahman & Dupoux, Emmanuel. (2021). Generative Spoken Language Modeling from Raw Audio.
13. Prakash, Anusha & Thomas, Anju & Umesh, Santhosh & Murthy, Hema. (2019). Building Multilingual End-to-End Speech Synthesisers for Indian Languages. 194-199. 10.21437/SSW.2019-35.
14. K. Vijay and D. Ramya, "Generation of caption selection for news images using stemming algorithm," 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), Melmaruvathur, India, 2015, pp. 0536-0540, doi: 10.1109/ICCPEIC.2015.7259513.
15. M. Zhang, Y. Yang, H. Zhang, Y. Ji, H. T. Shen and T. -S. Chua, "More is Better: Precise and Detailed Image Captioning Using Online Positive Recall and Missing Concepts Mining," in IEEE Transactions on Image Processing, vol. 28, no. 1, pp. 32-44, Jan. 2019, doi: 10.1109/TIP.2018.2855415.
16. Stefanini, Matteo & Cornia, Marcella & Baraldi, Lorenzo & Cascianelli, Silvia & Fiameni, Giuseppe & Cucchiara, Rita. (2022). From Show to Tell: A Survey on Deep Learning-Based Image Captioning. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 1-1. 10.1109/TPAMI.2022.3148210.
17. Gupta, Tanmay & Schwenk, Dustin & Farhadi, Ali & Hoiem, Derek & Kembhavi, Aniruddha. (2018). Imagine This! Scripts to Compositions to Videos.
18. Y. Li, M. R. Min, D. Shen, D. Carlson, and L. Carin, Video Generation From Text. arXiv, 2017. doi: 10.48550/ARXIV.1710.00421.
19. Q. Chen, Q. Wu, J. Chen, Q. Wu, A. van den Hengel and M. Tan, "Scripted Video Generation With a Bottom-Up Generative Adversarial Network," in IEEE Transactions on Image Processing, vol. 29, pp. 7454-7467, 2020, doi: 10.1109/TIP.2020.3003227.
20. Q. Chen, Q. Wu, J. Chen, Q. Wu, A. van den Hengel and M. Tan, "Scripted Video Generation With a Bottom-Up Generative Adversarial Network," in IEEE Transactions on Image Processing, vol. 29, pp. 7454-7467, 2020, doi: 10.1109/TIP.2020.3003227.
21. Shimono, Akari & Kakui, Yuki & Yamasaki, Toshihiko. (2020). Automatic YouTube-Thumbnail Generation and Its Evaluation. 25-30. 10.1145/3379173.3393711.
22. J. Zhang, Y. Zhao, M. Saleh, and P. Liu, “PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization,” in Proceedings of the 37th International Conference on Machine Learning, Jul. 2020, vol. 119, pp. 11328–11339
23. “Top 5 Web Scraping Tools Comparison,” Octoparse, 21-Jan-2021. [Online]. Available: https://www.octoparse.com/blog/top-5-web-scraping-tools-comparison. [Accessed: 17-Apr-2023].
24. Shchur, A. (2020, July 5). Fake news detector with Deep Learning Approach (part-II) modeling. Medium. Retrieved April 16, 2023, from https://towardsdatascience.com/fake-news-detector-with-deep-learning-approach-part-ii-modeling-42b9f901b12b
25. Fake news detection. Papers With Code. (n.d.). Retrieved April 16, 2023, from <https://paperswithcode.com/task/fake-news-detection>
26. Zaidi, Syed Ali Jafar & Buriro, Attaullah & Cssmbb, Mn & A., Mahboob, & Riaz, Mohammad. (2019). Implementation and Comparison of Text-Based Image Retrieval Schemes. International Journal of Advanced Computer Science and Applications. 10. 10.14569/IJACSA.2019.0100177.