Generating Audio Descriptions for Videos Using Contrastive Language-Image Pre-training (CLIP) Interrogation

Julia Chen

Thomas Jefferson High School for Science and Technology

Mobile Web Application Development Research Lab

Mr. Kosek

May 14, 2025

Table of Contents

1. Abstract 3

2. Introduction 4

3. Background 4

3.1 Neural Networks 4

3.1.1 Convolutional Neural Networks 5

3.1.2 Transformers 5

3.1.3 Contrastive Language-Image Pre-training 6

3.2 Hierarchical Clustering 7

3.3 Python 7

3.4 JPG File Format 8

4. Methodology 8

4.1 Metric of Success 8

4.2 Libraries 8

4.3 Algorithm Specification 9

5. Discussion 11

5.1 Results 11

5.2 User Interface Testing 13

5.3 Limitations 14

5.4 Unsuccessful Approaches 14

6. Conclusion 15

7. References 16

Abstract

Audio descriptions are a form of narration that provide blind and low vision individuals information about key visual elements of a video. This project aims to develop a machine learning model that generates audio descriptions to improve the accessibility of videos. The model analyzes the frames of a video by quantifying their complexity using JPG image size. Hierarchical clustering is performed on the JPG image sizes to identify the most representative frames. These frames are processed by the Contrastive Language-Image Pre-training (CLIP) Interrogator which generates descriptions for each of the selected frames. The descriptions are then added to the video in text and audio. The limitations of this model include lengthy processing time and inaccurate descriptions generated by the CLIP Interrogator model.

Introduction

Audio descriptions are a form of narration that provide blind and low vision individuals information about key visual elements of a video. Currently, since audio descriptions must be transcribed manually, many videos do not have audio descriptions and are thus inaccessible to people with visual impairments. Applications to generate alt text for images have been developed (Shen et al., 2024); however, minimal published research has been conducted to investigate the techniques that can be used to create an application that generates audio descriptions for videos.

Therefore, I aim to create a machine learning algorithm that generates audio descriptions of videos for blind and low vision individuals. The audio descriptions should be concise, accurate, and generated in a timely manner to improve the accessibility of videos.

This paper will introduce information on neural networks, clustering, Python, and the JPG file format, which are topics relevant to the development of the model. Subsequently, I will describe the specification and rationale of the developed algorithm. I will then present the results I achieved and discuss the successes and limitations of the model, as well as areas of future improvement.

Background

3.1 Neural Networks

Neural Networks (NNs) are a class of machine learning (ML) models that emulate the activity of the brain. NNs consist of interconnected nodes, known as neurons, that when given an input, aim to optimize the output to match the label, which is the provided output. During the training process, the NN minimizes the error of its outputs, known as the loss, compared to the true outputs. This is performed through back propagation, an algorithm that uses

gradient descent to modify the weights and biases of the neurons (O'Shea & Nash, 2015). The NN modifies the existing weights using (1), where w_{old} is the original value of the weight, λ is the learning rate, E is the total error, and w_{new} updated value of the weight.

$$w_{new} = w_{old} - \lambda \frac{\partial E}{\partial w_{old}} \quad (1)$$

The biases are also modified similarly using (2), where b_{old} is the original value of the bias and b_{new} is the updated value of the bias.

$$b_{new} = b_{old} - \lambda \frac{\partial E}{\partial b_{old}} \quad (2)$$

Training concludes after the weights and biases have converged or after a predefined number of epochs. Following training, the model can be used to classify unseen instances (O'Shea & Nash, 2015).

3.1.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specific type of NN designed to recognize patterns within images. It differs from the traditional NN in that it is more suited to process large images that would otherwise be expensive in terms of memory for a traditional NN due to the large size of the network that would be needed to process large images. The CNN applies many convolution operations on to the input image, which can extract the most important features of the image and better inform the classification of the image (O'Shea & Nash, 2015).

3.1.2 Transformers

Transformers are a type of NN that is designed to perform machine translation and natural language processing tasks. Transformers are unique from other NNs in that they are based on an attention mechanism, which allows them to focus on particular parts of the text input. Additionally, Transformers process the input data concurrently, which makes them very

time efficient and also enables them to make connections between data that is spatially distant (Shankar, 2023).

3.1.3 Contrastive Language-Image Pre-training

Contrastive Language-Image Pre-training (CLIP) is a model developed by OpenAI that learns visual concepts through natural language descriptions. CLIP associates text descriptions with images by embedding the text and images to the same vector space so they are directly comparable. Since the images and text are represented in the form, as vectors, the model can make connections between images and their associated text descriptions (Klingler, 2024).

Network Architecture

CLIP's architecture of two neural networks, a convolutional neural network (CNN) and a transformer. The CNN is responsible for image encoding, the process of extracting the most important information from an image, and the transformer is responsible for encoding the semantic meaning of the text description. The output of each of the neural networks is a vector in the shared vector space (Klingler, 2024).

Model Training

CLIP is trained on a labeled dataset containing 400 million image-text pairs. Among these pairs, there are positive pairs, a pair in which the text accurately describes the image, and negative pairs, a pair in which the text does not accurately describe the image. During the training process, CLIP uses contrastive loss to optimize the vector embeddings, which maximizes the similarity between positive pairs and minimizes the similarity between negative pairs (Klingler, 2024).

The metric to measure the similarity between vectors is cosine similarity; vectors that have a small angle between them are considered similar and vectors that have a large angle between them are considered dissimilar. The loss function penalizes the model for incorrectly matching pairs and rewards the model when it correctly matches an image-text pair (Klingler, 2024).

3.2 Hierarchical Clustering

Hierarchical clustering is a type of unsupervised ML model that clusters data points using similarity and creates a tree structure that represents the hierarchy. It begins by assigning each data point to a cluster and creates larger clusters by combining the smaller clusters based on similarity. The clusters with the most similarity are combined at each step until all the data points have been combined into one large cluster. There are many similarity metrics that can be used, some of which include the minimum distance between any two points of the clusters, the maximum distance between any two points of the clusters, the average distance between every pair of two points in the clusters, and Ward's method, which is based on the increase in squared error. The clusters created by the model can be visualized with a dendrogram, a tree-like diagram that indicates the clusters that have been merged at each step (Shetty & Singh, 2021).

3.3 Python

Python is a high-level, object-oriented programming language that has many built in data structures and libraries. These characteristics, along with its readability make it a popular choice for developing ML algorithms (What is Python?).

3.4 JPG File Format

JPG is a compressed file format that depends on the frequency representation of an image. The Discrete Cosine Transform (DCT) is used to convert the image's data into a frequency representation. The size of the JPG image represents the complexity of the image; a minor change in an image causes a minor change in the JPG file size of the image, and a large change in the content in an image corresponds to larger changes in the JPG file size of an image. Thus, the change in the JPG file size of an image can be used as a metric for the change in complexity of an image (Raid et al., 2014).

Methodology

4.1 Metric of Success

This project will be considered successful if a model to generate audio descriptions is created and the audio descriptions are (1) concise, (2) accurate, and (3) generated in a timely manner. An audio description will be considered concise if its length does not exceed the length of the segment of video it intends to describe. An accurate audio description is one that correctly describes the visual content shown in the video and makes note of the primary subjects present in the frame. An audio description that is generated in a timely manner is one whose processing time does not exceed the length of the video input. For example, if a two minute video is received as an input to the algorithm, the algorithm should take no longer than 2 minutes to generate accompanying audio descriptions.

4.2 Libraries

The core functionality of this project is built upon several Python libraries that implement and manage key tasks, including video frame extraction, frame selection,

text-to-audio generation, and audio concatenation. The libraries used in this project to perform these tasks include OpenCV 2, scipy, gTTS, and pydub.

OpenCV 2 Library

OpenCV 2 (also known as cv2) is a library frequently used for machine learning and computer vision tasks, and includes features that can be used to process images and videos. Among these include the ability to extract the individual frames that form a video (Culjak et al., 2012). This particular feature of the OpenCV 2 library is used in this project to extract the frames of the given input video so they can be analyzed by the CLIP Interrogation algorithm.

scipy Library

scipy is a library designed to carry out fundamental scientific algorithms, including optimization, integration, interpolation, eigenvalue problems, algebraic equations, and differential equations. It also includes classification algorithms, such hierarchical clustering, which are typically used for organism classification in biology (Virtanen et al., 2020). Hierarchical clustering is used in this project to create clusters of frames and to select a frame that represents the cluster. This is performed to reduce the total number of frames that the CLIP Interrogation algorithm must generate descriptions for.

gTTS Library

gTTS is a library designed by Google that uses Google Translate's text-to-speech API to generate spoken audio when provided with a text input (gTTS, 2024). In this project, gTTS is used to convert the text descriptions generated by the CLIP Interrogation algorithm into audio so they can be heard by the user.

pydub Library

pydub is a library that can be used to manipulate audio files, including slicing, concatenating, and editing (Ghadge, 2024). After the descriptions are converted to audio using gTTS, the pydub library is used to concatenate the individual audio files together and to ensure that the audio descriptions align with the visuals in the video.

4.3 Algorithm Specification

To generate audio descriptions, the algorithm selects individual frames of a video to analyze with the CLIP Interrogation algorithm.

To separate a video into its constituent frames, I used Python's OpenCV 2 library. I chose to examine every fourth frame of the video to reduce processing time, as examining every frame is time intensive and does not significantly improve the performance of the algorithm.

I used the JPG image size of each frame to quantify the information represented in the image so clustering can be performed to further select the frames to be analyzed by the CLIP Interrogation algorithm. The JPG image size is a metric that can be used to approximately determine the similarity between two images. Other more granular image similarity measurement techniques were not used because they are less time efficient and because such a degree of granularity is not necessary for the frame selection being performed. Typically, image similarity techniques, such as directly comparing the pixels of images to determine similarity, are used for keyframe selection in video compression. The purpose of the frame selection in this project is to select one important frame that is representative of a segment of video, as opposed to maintaining the quality of the video while reducing the number of frames, thus a high degree of granularity is not required.

After quantifying the data represented by an image, hierarchical clustering was performed on the data with frame number and JPG image size as attributes. The frame number and JPG image sizes were normalized to take on values between 0 and 1 so they are weighted equally when hierarchical clustering is performed. A threshold value of 1 was selected to determine the clusters created by the hierarchical clustering algorithm that would be used. This value was chosen through limited experimentation.

After the clusters are determined, the frame in the center of each cluster with respect to the frame number is chosen as the frame to represent the cluster. Subsequently, these frames are processed by the CLIP Interrogation algorithm to generate associated descriptions. The caption model used is the blip-large and the CLIP model used is ViT-L-14/openai. To ensure concise descriptions, the description used is the first clause of the description generated by the CLIP Interrogation algorithm. The description is then displayed on the screen for the duration of the cluster and read aloud by an auto generated voice.

Discussion

5.1 Results

To test the performance of the algorithm, I created a video, myVideo.mp4. This video tests the model's ability to generate accurate audio descriptions in a variety of circumstances, such as when a new subject is being introduced into the frame and when the camera pans quickly.

The video was separated into its constituent frames, which were subsequently quantified using their JPG image sizes. The hierarchical clustering algorithm produced the dendrogram in Figure 1, and with a threshold value of 1, the hierarchical clustering algorithm created four clusters.

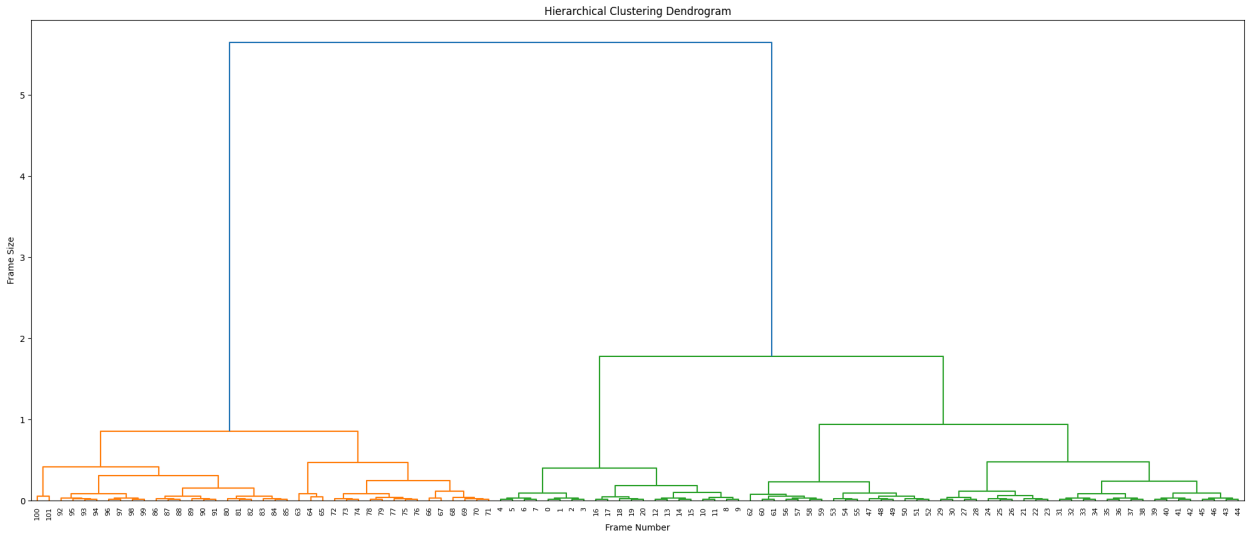

Fig. 1. Dendrogram produced by hierarchical clustering algorithm on the JPG image sizes of myVideo.mp4

Following clustering, the algorithm selected the centermost frame with respect to frame number from each cluster to serve as the keyframes. The keyframes are shown in Figure 2, along with their associated descriptions generated by the CLIP Interrogator.

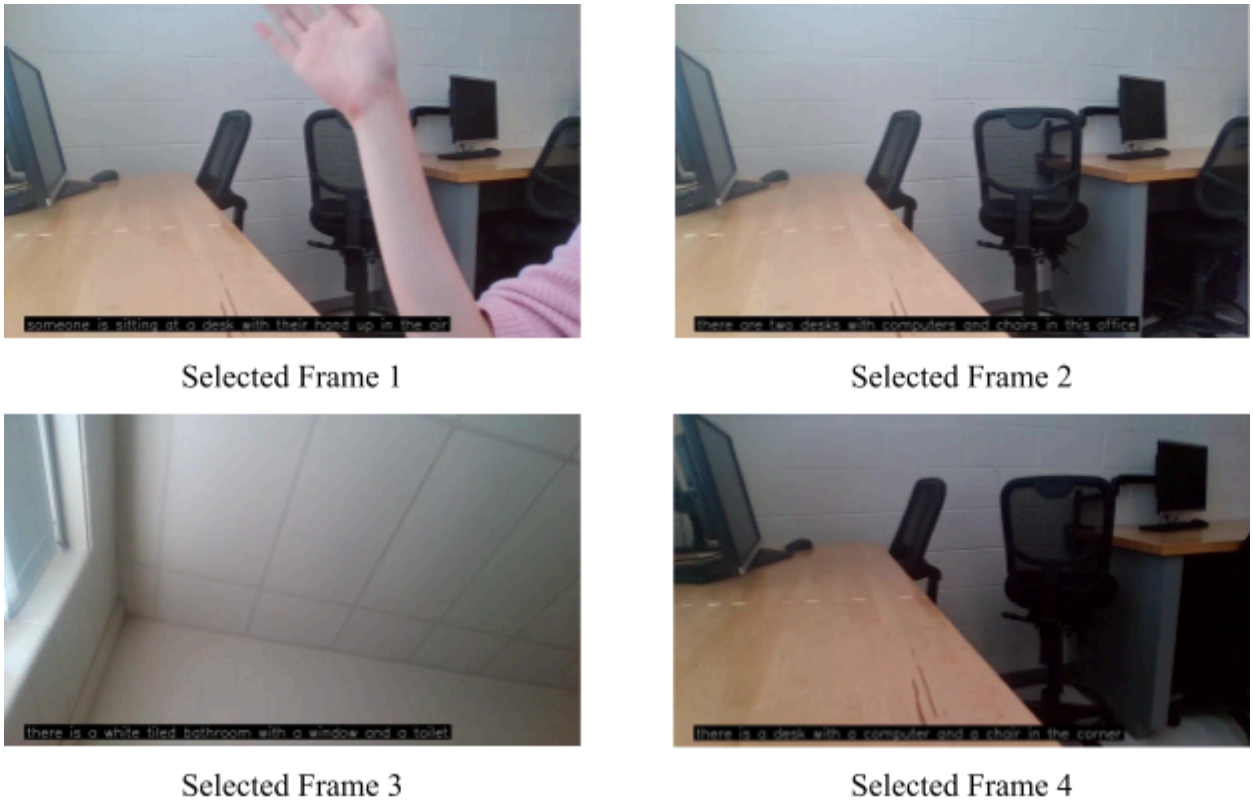

Fig. 2. Frames selected as keyframes for myVideo.mp4 by the hierarchical clustering algorithm with associated audio descriptions

Each description was displayed on the screen for the duration of the cluster the associated keyframe was selected from and was read aloud accordingly by an auto generated voice. The algorithm was able to generate accurate descriptions for Selected Frame 1 (“someone is sitting at a desk with their hand up in the air”), 2 (“there are two desks with computers and chairs in this office”), and 4 (“there is a desk with a computer and chair in the corner”). These descriptions were considered accurate because they correctly describe the primary subjects present in the frame. For Selected Frame 3, the algorithm generated a description,“there is a white tiled bathroom with a window and toilet.” This is not accurate, as the frame depicts a shot of the ceiling and windows. This indicates that the training data could be modified so the model can generate accurate descriptions for a wider range of environments and subjects.

5.2 User Interface Testing

I asked several peers to view two videos that the algorithm generated audio descriptions for. One of these was myVideo.mp4 and the other was a clip from "Antarctica: Home at the End of the Earth," a nature documentary (National Geographic, 2024).

After viewing the videos, the users commented on the accuracy of the descriptions. They noted that the descriptions were inaccurate at times and the algorithm often misidentified subjects. For example when generating descriptions for a clip from "Antarctica: Home at the End of the Earth," the CLIP Interrogator misidentified penguins as dolphins or seals. Additionally the CLIP Interrogator also stated that there were particular subjects in frame when they were not. For example, when describing a shot of the ocean from "Antarctica: Home at the End of the Earth," the algorithm stated there were surfers and boats in the ocean when these subjects were not present in the frame.

Users also commented on the phrasing of the descriptions. The majority of descriptions began with the words "there is," and users indicated that the sentence structure of the descriptions could include more variation.

The feedback indicated that the primary improvements that could be made to the algorithm involved the content and phrasing of the descriptions. These could be improved by modifying the model, specifically the model's training data.

5.3 Limitations

The CLIP Interrogator algorithm attempts to create a description that, if provided to a text-to-image model, would have generated the given image. Since this is the goal of the algorithm, rather than generating concise audio descriptions, extraneous information is generated. The extraneous information is removed before the audio description is created;

however, the process of generating audio descriptions could be optimized in terms of time and accuracy by training a CLIP model designed specifically for generating audio descriptions.

The algorithm is also quite time expensive. For a two minute long video clip, the algorithm took 24 minutes to generate audio descriptions. This is an average of 12 seconds of processing per second of video. The CLIP Interrogator took four minutes generating descriptions for the chosen frames. 10 minutes were spent on adding the descriptions to the video as subtitles, and 10 minutes were spent on adding the audio descriptions to the video. The most meaningful improvements to runtime could be made by optimizing the processes of adding the descriptions to the video in text and audio.

5.4 Unsuccessful Approaches

Prior to performing clustering on the JPG image sizes for frame selection, I attempted to use You Only Look Once Version 11 (YOLOv11) to perform frame selection, where a new keyframe would be selected when YOLOv11 detected a change in the primary subjects of the image. However, YOLOv11's detections were often sensitive to negligible changes between the frames, leading me to investigate alternative methods for keyframe selection that were less sensitive to minor changes across frames of video.

After using JPG image size to quantify each frame, I attempted to use the K-Means clustering algorithm to create clusters from which I could select the keyframes from. However, K-Means clustering did not perform well for the task of identifying keyframes because it clustered the frames that had a similar JPG image size, but were not necessarily adjacent in the video. Additionally, I also attempted to determine the k-value using the elbow method and the

silhouette score; however, both of these methods were ineffective at determining a reasonable number of keyframes for the given video.

Conclusion

This study investigated the various techniques that could be used to develop a ML algorithm to generate audio descriptions for videos and proposed a possible implementation of this algorithm.

The implementation consists of selecting keyframes by quantifying the frames of a video by extracting their JPG image size and performing hierarchical clustering on the image sizes to select the most representative frames. The CLIP Interrogation algorithm processes these frames to generate audio descriptions, which are then applied to the video.

Future work could further optimize the approaches proposed by this study, such as optimizing the runtime, using a CLIP model designed specifically to generate audio descriptions, developing an algorithm that varies the threshold for hierarchical clustering based on the input video, or developing an algorithm that generates descriptions based on segments of video rather than analyzing only the constituent frames.

References

Culjak, I., Abram, D., Pribanic, T., Dzapo, H., & Cifrek, M. (2012). A brief introduction to OpenCV. *2012 Proceedings of the 35th International Convention MIPRO*. https://ieeexplore.ieee.org/document/6240859

Ghadge, M. (2024, September 23). *Exploring the Pydub library: A comprehensive guide to audio manipulation in Python*. Medium. https://mjghadge9007.medium.com/exploring-the-pydub-library-a-comprehensive-guide-to-audio-manipulation-in-python-46a09c96f69b

gTTS. (2024, November 10). Python Package Index. https://pypi.org/project/gTTS/

Klingler, N. (2024, September 27). *CLIP: Contrastive language-image pre-training*. Viso.ai. https://viso.ai/deep-learning/clip-machine-learning/

National Geographic. (2024, April 11). *Antarctica: Home at the end of the earth (Full episode) | incredible animal journeys* [Video]. YouTube. https://www.youtube.com/watch?v=eS6a6btDK8M&t=163s

O'Shea, K., & Nash, R. (2015, December 2). *An introduction to convolutional neural networks*. https://arxiv.org/pdf/1511.08458

Raid, A.M, Khadr, W.M, El-dosuky, M.A, & Ahmed, W. (2014). Jpeg image compression using discrete cosine transform - A survey. *International Journal of Computer Science & Engineering Survey*, *5*(2), 39-47. https://doi.org/10.5121/ijcses.2014.5204

Shankar, A. (2023, September 27). *Understanding Google's "Attention Is All You Need" paper and its groundbreaking impact*. Medium. https://alok-shankar.medium.com/understanding-googles-attention-is-all-you-need-paper-and-its-groundbreaking-impact-c5237043540a

Shen, Y., Zhang, H., Shen, Y., Wang, L., Shi, C., Du, S., & Tao, Y. (2024). AltGen: AI-Driven Alt Text Generation for Enhancing EPUB Accessibility. *arXiv preprint arXiv:2501.00113*.

Shetty, P., & Singh, S. (2021). Hierarchical clustering: A survey. *International Journal of Applied Research*, *7*(4), 178-181. https://doi.org/10.22271/allresearch.2021.v7.i4c.8484

Stancin, I., & Jovic, A. (2019). An overview and comparison of free python libraries for data mining and big data analysis. *International Convention on Information and Communication Technology, Electronics and Microelectronics*. https://doi.org/10.23919/mipro.2019.8757088

Virtanen, P., Gommers, R., & Oliphant, T. E. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*, 261-272. https://doi.org/10.1038/s41592-019-0686-2

What is Python? Executive summary. (n.d.). Python. https://www.python.org/doc/essays/blurb/