Term Report

Computer Visualization

Contents

1 Introduction

In this final project for the Advanced Course on Computer Visualization, I will aim to create a dashboard for visualizing the pickled data provided in the UnweaveNet active stories repository available on Github at this link: https://github.com/willprice/activity-stories. I chose to visualize this dataset for the purpose of my research which is video summarization and the UnweaveNet paper introduces video summarization as a related work.

This report will start by an explanation of the UnweaveNet paper [2] and the components of the provided dataset. The last part of this report will constitute of the explanation of each element of visualization available in this dashboard, why this form was used and how they were regrouped to get a better and effective comprehension of the dataset.

2 Description of the dataset

2.1 Definitions

[2] introduces a new conceptualization of a video. Instead of a succession of frames, a video is considered as a story, which is constituted of woven activity threads. But what is an activity thread? [2] gives the example of a morning routine such as the following: after waking up, you first decide to make your coffee by turning on the coffee machine. While the coffee is preparing, you start making toasts. As soon as the coffee is done, you stop making toasts to get the coffee. In this example, we can see two distinct activities: making coffee and making toasts. An activity can be paused and later resumed, like the "making coffee" activity in the example. As we read this example, we unfold two main narratives. The concept of unweaving that is introduced in [2] is similar to this process: as the video is processed, when it detects a new activity, it creates a new thread and at each processed clip, it updates the representation of existing activity threads or creates a new one.

Another important definition is the definition of a *story*. This is more simple: videos of activities are referred to as *activity stories*. The dataset I am working on is an ensemble of activity stories from the Epic Kitchen dataset [1], one of the major dataset of egocentric videos with annotations of the actions. The EPIC-KITCHENS activity stories is a dataset containing all the annotated stories issued from EPIC-KITCHENS-100 for the training of the UnweaveNet, a model to automatically unweave videos.

2.2 Dataset components

Each row of the dataset contains information about an annotated story. There are 5 attributes: **video_id**, **clip_frame_idxs**, **thread_clip_idxs**, **id** and **split**. Before giving the direct description of the attributes, it is essential to understand how the data is formed. The annotation of the data is such that the annotator has been given a portion of a video - *a story* - to annotate. This story is constituted of clips (10 clips per story). Each clip is a succession of frames that are indexed. When doing the annotation, the annotator had to group the clips that belong to the same activity thread together. After analysis, there are one to three threads per story. During the annotation the clips are still kept in their chronological order in the threads.

To sum up: each story is a portion of video. It is constitued of one or more threads that are themselves constituted of clips (an ensemble of frames).

Now here is the description of the attributes of the dataset of each row.

- **video_id**: corresponds to the ID of the video in the EPIC KITCHENS dataset which the story is a portion of.
- **clip_frame_idxs**: 2D array to retrieve the list of frames indices corresponding to the j-th clip of the i-th thread with $clip\_frame\_idxs[i][j]$.

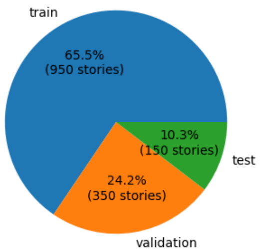
- **thread_clip_idxs**: 2D array. Given the i-th clip of the story, we can retrieve the index of the thread $thread\_idx$ the i-th clip belongs to and at which position $thread\_clip\_idx$ the clip is in the thread with $thread\_clip\_idxs[i] = [thread\_idx, thread\_clip\_idx]$
- **id**: unique identifier for the story
- **split**: indicates which split of the dataset it belongs to: train, val (for validation) or test.

3 Dashboard visualization

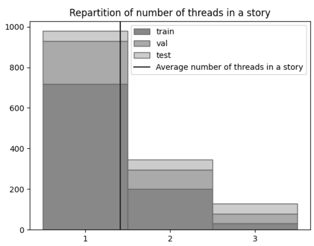
For the visualization of the data, I decided to do a dashboard, since there is a lot of information that can make it hard to understand what is inside the data. Mainly, since the data has been used for the training of a neural network, it is important to understand how the data is divided in different splits for training, validation and testing. But I also wanted to know the data for one video and what parts of each video is annotated. This part is the most obscure and hard to understand. This is why I decided to separate the view of the overall dataset and the view of the information specific to one video.

3.1 Overview of the complete dataset

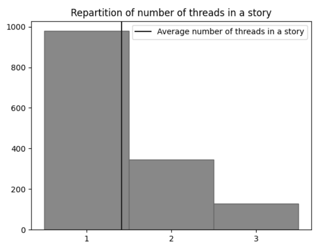
The overview of the complete dataset mainly consists of visualizing the repartitions of the stories in the dataset and per split. For that I created two different visualizations: first one is a pie chart and the second one is histograms.



(a) Distribution of the stories in the different splits

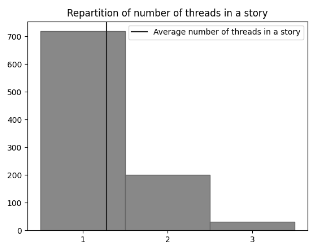


(b) Stacked histogram of number of threads per story

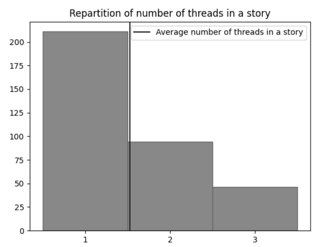


(c) Histogram of number of threads per story

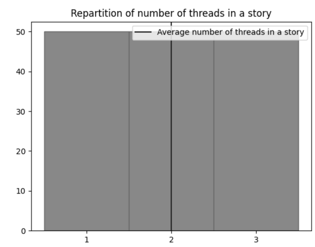
Figure 1: Graphs for the overall view of the dataset



(a) Number of threads per story in *train* split



(b) Number of threads per story in *validation* split



(c) Number of threads per story in *test* split

Figure 2: Separated histograms for number of threads per story

The pie chart shows the percentage of the data corresponding to each split. Meanwhile, the histograms show a distribution of stories containing a certain number of threads. After visualization of the data, we can see that each story has between 1 and 3 threads. The splits are constituted such that the train split 2a and validation splits 2b are similar: the distribution of stories with a certain number of threads is similar. With the view of separated histograms per split, we can clearly see the intent of the test split to have an equal number of stories containing one, two and three threads 2c.

I decided to do two views for the distribution of the number of threads per story in the dataset: one with separated histograms 2 and another one with stacked histograms 1b. A stacked histogram is great to have a quick visual comprehension of the data. However, it is less readable to have the exact numbers, hence I also did a separated histograms view. Also, on these histograms, I added a "mean line", that show the average number of threads per story. This line is computed on the whole dataset for the stacked histogram, and not for each split as it would lead to less readability.

3.2 Display of information for a specific video

The second part of the dataset corresponds to the information we can get on the story itself. After analysis, I checked that stories from the same video all belong to the same split. For user interface purpose, since there are a lot of videos, I decided to split the dashboard into the three different splits. To select the different information we want on a specific video, I created similar pages for each split, which contain the list of videos corresponding to the split. However, such way of doing creates the problem of needing to know beforehand to which split the video belongs to. I decided that this could be changed by searching for the different split pages. An ideal way to check the information on a specific video would be to add a research bar but I lack the skills to implement it.

3.2.1 Aggregation of the information for each story

Before showing the information for a video, I aggregated the information for each story belonging to the same video. A story is actually an ensemble of clips and each clip is said to belong to thread 1 or 2, etc. I first created a visualization of the story with a "timeline". It is a 1D visualization of the frame indices. There is a colored bar at each frame index that has been annotated. The color of the bar depends of the thread it belongs to. Hence, on this timeline we can see the frame indices corresponding to each thread and how the threads are woven.

After plotting many timelines, I realized that knowing the information on the story alone is not enough interesting. Hence, I decided to create a visualization on the video scale.

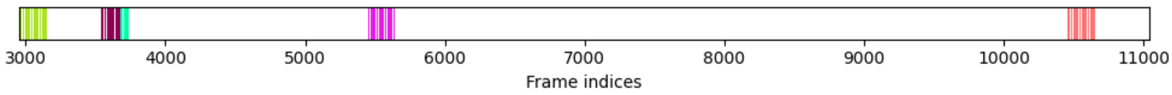
3.2.2 Aggregation of the information for each video

It is more interesting to know which frames of the complete video are annotated. I extended the visualization for one story to a video itself 3. It is easy to retrieve the data for one video since each story contains the video_id of the video they are a portion of as an attribute.

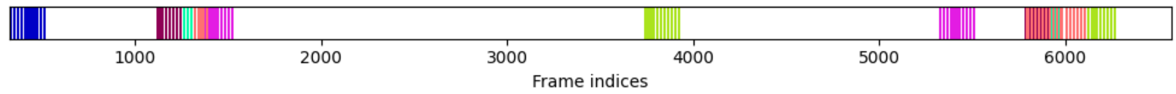
There is one major problem with the extension of the visualization: the colors are created randomly to make sure that they are different. However, though the colors are indeed different, it is complicated to have them be distinct colors and sometimes it seems like two different clips belong to the same thread when it is not the case.

In addition to the timelines, there are also pieces of information such as the number of threads and the number of stories from the same video that are written.

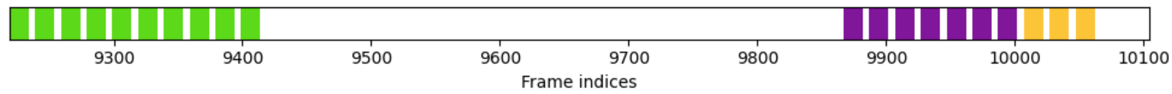
An ideal visualization would add the option to zoom in on the visualization of the complete timeline and have the story id displayed when we zoom in. However I do not have the skills for this since the figures were all created using matplotlib.



(a) P16_01



(b) P30_09



(c) P37_102

Figure 3: Examples of timelines

4 Conclusion

The current dashboard still needs to be perfected but other libraries for its creation should be considered. The use of a dashboard can help the user to select what they want to see from the data more easily and understand the different splits of the dataset. The interaction permits a better selection of precise data but even more interactions could be added to have data of another level.

A Screenshots of the dashboard

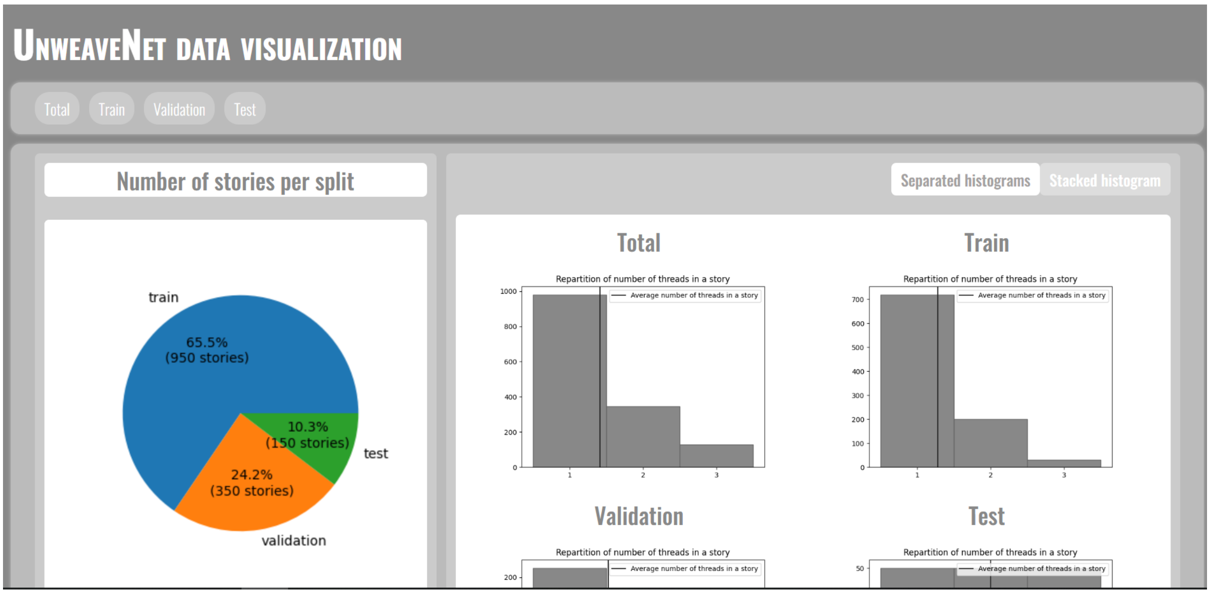


Figure 4: Screenshot of the complete dataset overview (separated histograms tab)

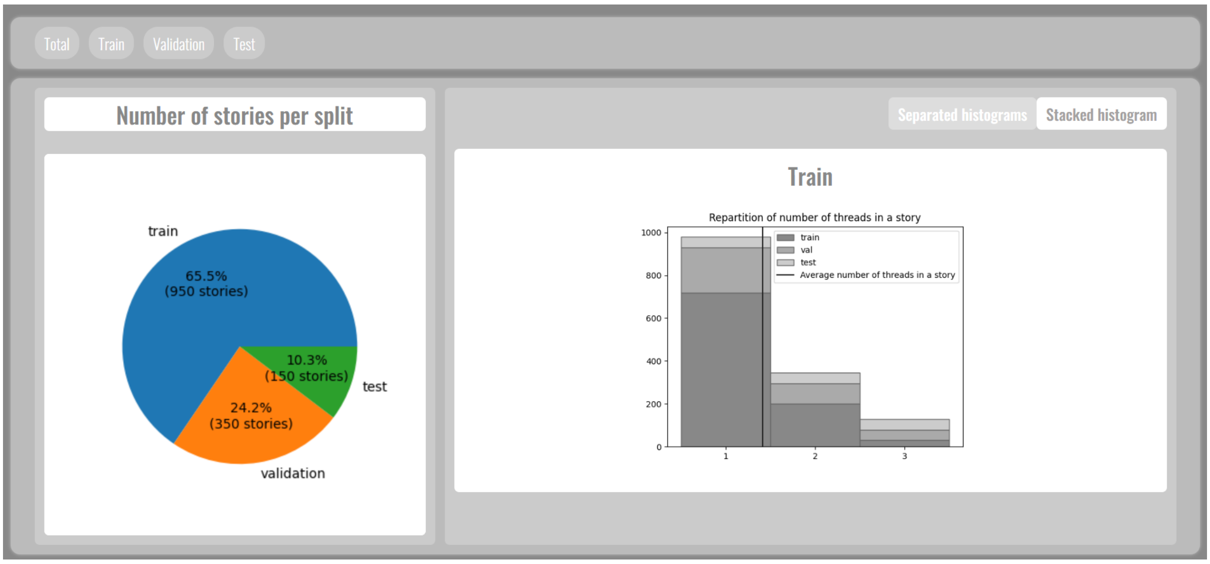


Figure 5: Screenshot of the complete dataset overview (stacked histograms tab)

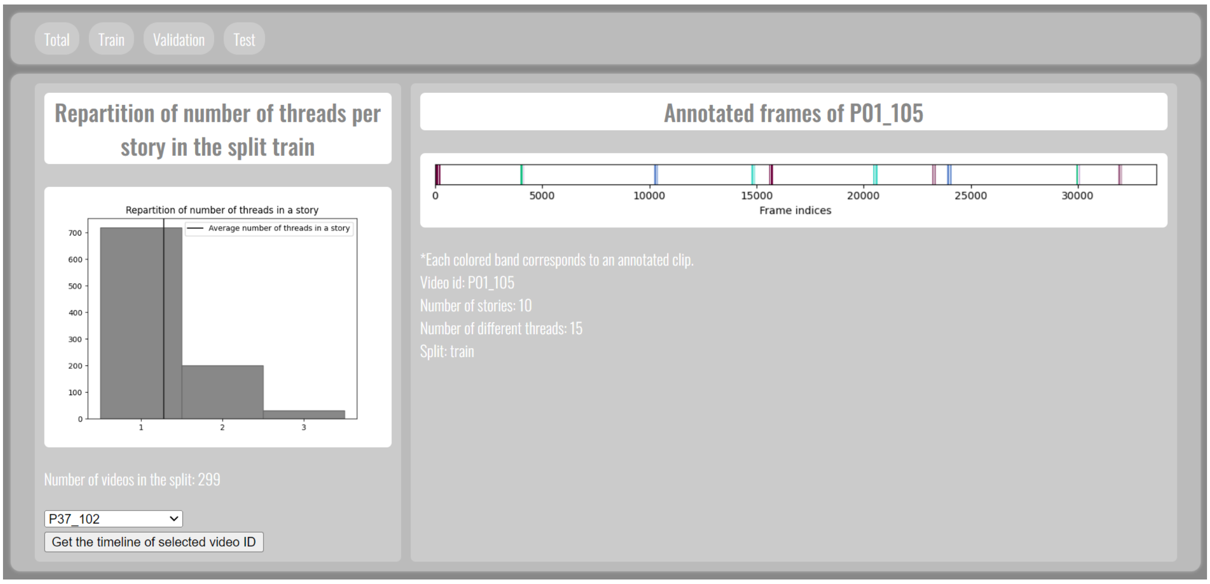


Figure 6: Screenshot of a split dataset page

B Github link for code repository

The complete code for the dashboard is available at the following repository: `https://github.com/mirarzf/Computer-Visualization`.

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2021.

[2] Will Price, Carl Vondrick, and Dima Damen. Unweavenet: Unweaving activity stories. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022.