POLITECNICO DI TORINO
Repository ISTITUZIONALE

Movie Lens: Discovering and Characterizing Editing Patterns in the Analysis of Short Movie Sequences

*Original*
Movie Lens: Discovering and Characterizing Editing Patterns in the Analysis of Short Movie Sequences / Vacchetti, B., Cerquitelli, T.. - 13804:(2023), pp. 660-675. (17th European Conference on Computer Vision (ECCV 2022) Tel-Aviv 23-27 Ottobre 2022) [10.1007/978-3-031-25069-9_42].

*Availability:*
This version is available at: 11583/2971433 since: 2022-09-21T12:06:00Z

*Publisher:*
Springer

*Published*
DOI:10.1007/978-3-031-25069-9_42

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*
Springer postprint/Author's Accepted Manuscript

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-031-25069-9_42

(Article begins on next page)

27 November 2026

# MOVIE LENS: Discovering and Characterizing Editing Patterns in the Analysis of Short Movie Sequences

Anonymous ECCV submission

Paper ID 12

**Abstract.** Video is the most widely used media format. Automating the editing process would impact many areas, from the film industry to social media content. The editing process defines the structure of a video. In this paper, we present a new method to analyze and characterize the structure of 30-second videos. Specifically, we study the video structure in terms of sequences of shots. We investigate what type of relation there is between what is shown in the video and the sequence of shots used to represent it and if it is possible to define editing classes. To this aim, labeled data are needed, but unfortunately they are not available. Hence, it is necessary to develop new data-driven methodologies to address this issue. In this paper we present MOVIE LENS, a data driven approach to discover and characterize editing patterns in the analysis of short movie sequences. Its approach relies on the exploitation of the Levenshtein distance, the K-Means algorithm, and a Multilayer Perceptron (MLP). Through the Levenshtein distance and the K-Means algorithm we indirectly label 30 seconds long movie shot sequences. Then, we train a Multilayer Perceptron to assess the validity of our approach. Additionally the MLP helps domain experts to assess the semantic concepts encapsulated by the identified clusters. We have taken out data from the Cinescale dataset. We have gathered 23 887 shot sequences from 120 different movies. Each sequence is 30 seconds long. The performance of MOVIE LENS in terms of accuracy varies (93% - 77%) in relation to the number of classes considered (4-32). We also present a preliminary characterization concerning the identified classes and their relative editing patterns in 16 classes scenario, reaching an overall accuracy of 81%.

**Keywords:** Sequence Analysis, Movie Editing, K-Means, Multilayer Perceptron, Levenshtein Distance

## 1 Introduction

Videos have been deeply studied in the past using machine learning techniques. These studies mainly focus on image, video, or text data. However, a little attention has been devoted to studying the editing process of a video that defines its narrative structure.

Editing consists of joining individual shots together to form a scene. How long each shot is, which shot comes before and which after, are all elements that

strongly influence the viewer's perception. In other words, editing helps define the narrative and mood of a film, along with other elements such as the setting, the soundtrack, and so on. However, while these elements are perceived by our senses, the audience only notices editing when it is poorly done. For this reason, it has been called invisible art. According to Walter Murch [15], editing a film is like telling a story. If there is a great story, but the narrator tells it in the wrong rhythm and focuses on the wrong parts, it has no impact. If the film is poorly edited, the viewer will be less engaged in the story. Worse, there is a possibility that the viewer will not understand the story at all. This is because editing, among other elements, determines the structure of a film, which consequently affects the mood of the narrative style. Practically speaking, editing is a process of cutting and chaining together individual clips that have no meaning on their own to create a meaningful video.

In the past, some studies, such as [25], have analyzed the challenges of automating the entire video editing process. Some successful efforts have been made to solve some tasks of the process. One example is the autoEdit library, which is based on the research presented in [4]. AutoEdit receives as input a video file and gives as output the text extracted from the people speaking in the video. Then, the user has to select a part of the text, and the algorithm selects the corresponding clip from the video and cuts it. Automating video processing could bring interesting and useful benefits, especially considering that it is the most commonly used media format [31]. For example, a deeper understanding of the narrative structures implemented in videos could lead to the automatic creation of more semantically meaningful videos on social media. It could also speed up the post-editing phase of a video production. Another practical implementation could be an integration with text-image translation, such as DALLE [19]. This type of model is able to generate photo-realistic images from a textual description. This type of network could be used to transform scripts into storyboards. However, to do this, they need to understand the classification of shots and editing patterns.

In this paper we propose a novel data-driven methodology, named Movie Lens . It allows identifying editing patterns within video structures and characterizing their main properties. To this aim, we focus on sequences of shot scale classes. The shot scale is defined in relation to what is shown in the camera field of view (see Section 3 for further details). We are interested in investigating what kind of correlation exists between sequences of shot classes and what is actually shown in the corresponding video. We selected shot class sequences from the Cinescale dataset [21]. However, the sequences obtained were not labeled. To label them, we have used a novel method based on a joint approach that relies on K-Means and the Levenshtein distance. As a result, we assign a label for each sequence without directly analyzing the content of the sequences. Then we then train and test a classifier using the original sequences and the newly obtained labels.

Specifically, Movie Lens introduces the following contributions: (i) a data-driven methodology to cluster short video sequences in homogeneous and well-

separated groups; (ii) a machine learning algorithm capable of classifying video sequences assessing the robustness of the clustering analysis. Furthermore, it helps domain experts to easily identify a semantic label for each group of short video sequences. (iii) Additionally we present a preliminary characterization conducted on a real set of data characterized by 23 887 sequences from 120 different movies.

This paper is organized as follows. In Section 2 we analyze the existing state-of-the-art on the subject. In Section 3 we focus on the data used, while in Section 4 we present the Movie Lens methodology. Section 5 discusses some preliminary results obtained by Movie Lens on a large set of real-data. Section 6 concludes the paper with an in-depth discussion on the weaknesses of the proposed approach and how to address them in our future research studies.

## 2 Related Works

In recent years, videos and films have been analyzed using different techniques and for different purposes. Most of these studies are related to computer vision and image classification, but other branches of research focus on other aspects. One task that has been addressed in the past using different approaches is the classification of movie images [2] [22] [28] [27]. It is a classification that assigns a shot type to an image based on what is in the camera's field of view. Most of these studies rely on convolutional neural networks (CNN) [23] to predict the shot scale of an image. This type of classification can be more or less fine-grained. Some studies also incorporate camera motion classes [8]. In computer vision, there have been significant studies on translating text into images. These algorithms, such as [19], are able to reproduce a more or less accurate image based on a text description. A rather different, but still interesting approach is the work proposed in [18]. Here the authors use natural language processing techniques on the IMDB dataset to perform movie sentiment analysis.

In terms of studies dealing with the editing structure of videos and films, there have been some developments. The first study on this topic dates back to 2002 ([13]). Here, the authors attempted to identify editing rules and patterns. They emphasized the central importance of editing in videos. Their focus was on editing speed and on a preliminary version next shot type prediction with three classes. Recently, in [3], the authors have analyzed how the concatenation of shot types affects viewer attention. Another interesting study is proposed in [10]. Here, the authors present an automatic censoring method that detects inappropriate visual content and classifies it as "Universal," "Universal Adult," or "Adult". In [33], the authors perform video retrieval. They use recording sequences in combination with temporal information and visual features to find a specific video in a large collection. Also in [1], video retrieval is performed using textual data along with other movie metadata such as script, editing speed, release year, and genre of the movie. In [24], the authors instead perform movie genre classification by implementing a methodology based on convolutional neural networks. [32] is also about movie genre classification, but the approach here

is different. Representative key frames are extracted from trailer clips and genre classification analysis is performed using these key frames. In [16], the authors focus on film structure by analyzing the time interval from one cut to the next. They examine how film genre affects shot duration. In [26], the authors analyze what kind of relationship exists between the director of a film and the shots used in the film. In automated video editing, there have been advances in various aspects of the entire process. In [4], the authors develop a method for cutting a video depending on the part of the dialog that the transcript text receives as input. In [20], the focus is not on classifying frames or videos into recording types, but on video segmentation, i.e., splitting a video into individual clips. In [17], the authors present a preliminary methodology that mimics the editing structure of movies. In this study, three shot typologies are considered: close up, medium shot, and long shot. Another interesting study dealing with automatic video editing is presented in [30]. Here, the focus is on the typology of videos to be processed in multi camera environments. For a more comprehensive overview, we recommend the following surveys [14] [31] [25]. All of these studies could yield even more surprising results if they integrated even rudimentary narrative editing patterns.

Editing structure influences how the story is perceived. Some studies address specific aspects of editing structure or use automated editing in controlled environments. Differently from all cited works, MOVIE LENS focuses on discovering and characterizing editing patterns to better understand how the sequence of shots changes in relation to what is shown on screen. Only a few research studies, address our research issue, such as [17] [30] [3], however with a different methodology and a different problem characterization. Differently than [17] [30] [3], MOVIE LENS considers more shot classes and integrates a novel strategy to automatically discover editing patterns and how to classify them easily.

## 3 Data




Fig. 1: The 8 types of shot considered in this study.

For our experiment, we have used the Cinescale dataset [21]. To be more precise, only its labels were used. Cinescale is a dataset containing 120 different

movies realized by six different directors. It is a dataset used to perform cinematographic shot classification. From each movie a frame has been sampled and labeled at every second. The label assigned to each sampled frame corresponds to its shot class. The shot classes considered in this study are the following: (i) Class 0) Foreground Shot (FS): a shot that contains elements of different shot classes. For instance camera movements fall under this category (128 335 frames); (ii) Class 1) Extreme Close Up (ECU): a shot that focuses on details, such as the eyes of the subject, what the character is holding and so on (3 367 frames); (iii) Class 2) Close Up (CU): a shot focused on the subject's face, it shows the actor from the shoulder up. It can also be used to focus the viewer's attention on some detail like objects or hands (83 682 frames); (iv) Class 3) Medium Close Up (MCU): the subject figure is shown from the upper half of its torso (252 639 frames); (v) Class 4) Medium Shot (MS): only the upper half of a human subject is shown (78 053 frames); (vi) Class 5) Medium Long Shot (MLS): the human figure is shown from the knee up (89 450 frames); (vii) Class 6) Long Shot (LS): the human figure occupies the totality of the frame height or 2/3 (49 788 frames); (viii) Class 7) Extreme Long Shot (ELS): the human figure is absent or occupies less than a third of the screen height (7 118 frames). Figure 1 shows the different types of shot classes considered in this study.

The extracted video sequences were 30 frames long, which corresponds to 30 seconds, since in the Cinescale dataset one frame is sampled every second. We focused on 30-second sequences for the following reasons. Not only do scenes in movies vary in length, but even when they are of similar length, the number of shots used varies. Therefore, to simplify the problem, we decided to use 30-second sequences. Usually scenes do not last that long, but it is rare for a scene to be shorter than 30 seconds. In this time interval, there is also a chance to capture some patterns, while if smaller sequences are chosen, there is a risk that the resulting sequences cannot describe anything in particular. On the other hand, if a larger time interval is chosen, there is a chance that more scenes will be compressed into a single one. Cinescale has two other classes besides the shot type classes. One contains opening and closing titles, the other undefined frames. After removing the sequences that contained these shots, the resulting dataset contained 23 887 sequences. The next section will explain the methodology in more detail.

## 4 Methodology

Movie Lens performs two main analytic building blocks: the *Label Estimation Phase* and the *Editing Patterns Analysis*. The aim of the first task is to label each short movie sequence in the selected portion of data. In order to do that, first we run an analysis based on a distance metric on every sequence. The analysis focuses on how much every sequence is similar to some fixed reference sequences. At the end of this operation we use those distances as points' coordinates. Every point models a sequence. The points are then grouped through a clustering algorithm. Each group should theoretically model a type of sequence. As a result

of the first step, MOVIE LENS defines a cluster identifier for each short movie sequence. To help the domain expert to define a semantic label to each group, we have introduced a second phase named Editing Pattern Analysis. The aim of the second step is to evaluate the results of the Label Estimation Phase through the analysis of the performance yielded by a classifier on a portion of the dataset. After training the model, we test it on the remaining part of the dataset. The obtained results need to be assessed by domain experts. Specifically, in this final step MOVIE LENS allows the domain expert to analyze which patterns characterize each class and which ones are the most common misclassification errors. The domain experts are asked to assess the quality of the classes identified by MOVIE LENS . This analysis is performed manually on a subset of short movie sequences for each class along with the label defined by the classifier. Selected sequences include sequences that are classified correctly or misclassified.

Figure 2 shows the main analytic building blocks of MOVIE LENS . Further details of each task are provided in the next subsections.

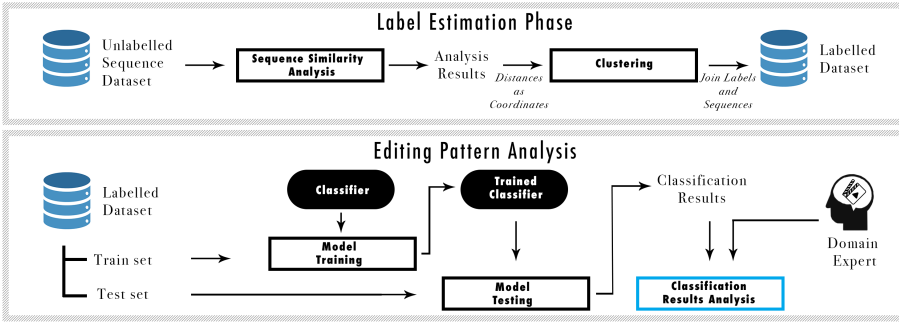



Fig. 2: Overview of the MOVIE LENS 's methodology.

### 4.1 Label Estimation Phase

This analytic building block performs the similarity analysis and models its results through a clustering algorithm. Specifically, we have used the Levenshtein distance [7] to compute the sequences' similarities and differences, and the K-Means algorithm[11] to indirect model editing patterns. The Levenshtein distance measures the similarity between sequences by counting the minimum number of changes (substitution, insertions, and deletions) needed to convert one sequence to another. The selection of the distance measure has been guided by the data semantic. Specifically, the shot type classes on Cinescale are defined through numerical values. From label 1 to 7 the higher the class number the wider shot we are considering. However, the foreground shot class, identified by the number 0, is not defined by the shot scale. Since the numerical magnitude

of a class is not reliable to compute more traditional distances, such as the Euclidean distance [12], we have chosen a different approach. We have considered our shot sequences as strings and used the Levenshtein distance. However, also the Levenshtein distance has some limitations, because it does not characterize the changes made. However in our scenario it is important to know also to what class the frame has been converted to in order to match the sequence. Hence, instead of using the Levenshtein distance to directly measure the differences between sequences, we took a slightly different approach. First, we have defined 8 reference sequences artificially, one for each shot type (e.g., close up). All reference sequences were 30 frames long, with each frame belonging to the same shot type. Then, we have measured the Levenshtein distance among the short video sequences extracted from the Cinescale dataset and each reference sequence. After computing the distances between each sequence and the 8 reference ones, we have used those distances as coordinates. In this way each sequence corresponds to an 8-dimensional point in the multi-dimensional space. Afterwards we have clustered all the points and labeled them according to which region of the 8-dimensional space they occupy. Among the different clustering techniques we have decided to rely on the K-Means algorithm. We have decided to exploit this specific clustering technique because it converges quickly while providing good results [6]. The only parameter it requires is the number of clusters. To find a reliable value for the input parameter, a joint strategy based on the elbow graph method and Ward's method - i.e., a hierarchical clustering metric - has been adopted. Both these strategies focus on identifying the optimal number of clusters by minimizing the within-cluster variance. We have run different simulations with a varying number of groups. For each 8-dimensional point, the cluster analysis identifies the group of similar sequences to which the corresponding short movie sequence belongs to.

### 4.2 Editing Patterns Analysis

This phase has a double purpose. On the one hand it assesses the robustness of the clustering analysis. On the other hand it helps the domain experts to analyze emerging patterns from the clustering groups. In the editing pattern analysis we split our dataset into train and test sets. The classifier is trained on the shot sequences with the labels defined through the previous step. Then, the performance of the classifier has been evaluated through the most common classification quantitative metrics (see Section 4.3 for further detail). For the classifier choice we have decided to rely on a Multilayer Perceptron (MLP), by adapting the model [1] proposed for sentence classification to short movie sequences classification. After training the MLP we evaluate its performance by analyzing the classification results on the test set. This evaluation step has a double purpose. This evaluation step on the one hand allows understanding if the results obtained from the labeling phase are robust. On the other hand it

[1] the Multilayer Perceptron for sentence classification can be retrieved from a GitHub repository [29]

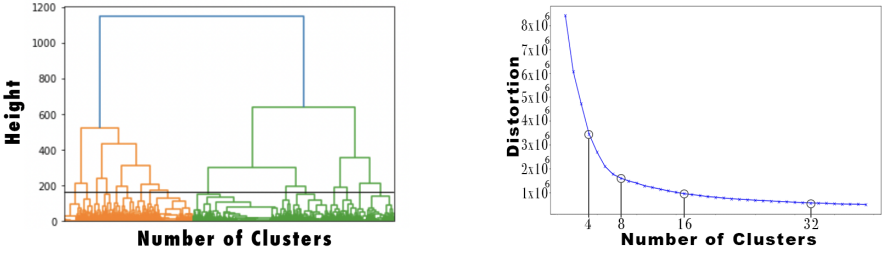
allows domain experts to easily evaluate the main characteristics of each defined class. In fact, after training and evaluating the model, we analyze the patterns characterizing the different classes. To this aim the domain experts are asked to verify the correctness of the classification results and to extract the main characteristics of each sequence type. For each class, we selected 5% of short movie sequences stratified with respect to the class cardinality and the distribution of misclassified sequences. The domain expert verifies what type of correlation there is among the identified patterns and what is shown in the video. Section 5 offers more insight on this aspect.

### 4.3 Technical Details

We used a traditional K-Means procedure to cluster the 8-dimensional points representative of the acquisition sequences. After a large set of experiments we have defined the following Movie Lens configuration. The number of initial centroids of the clusters is 20. The MLP has an input layer, a hidden layer, and an output layer. The number of units of the input layer and the hidden layer are 128 and 64 respectively. The number of layers of the output layer depends on the number of classes considered. The optimizer used is the adaptive Nesterov moment estimation (Nadam). The activation function for both the input and hidden layer is the Hyperbolic tangent activation function (tanh), while for the output layer the activation function is the softmax function. The loss function used is the categorical cross entropy. The experiments were conducted on a HPC system with 4 CPU allocated. The code was implemented in Python with the support of the following libraries: Numpy, Keras, Tensorflow, Scikit-learn and Pandas. The metrics considered in this study are the following: (i) accuracy: number of correct predictions over the total amount of predictions; (ii) recall: the ratio of true positives identified over the sum of true positives and false negatives; (iii) precision: the ratio of true positives identified over the total of positives, both true and false; (iv) f1-score: the weighted harmonic mean of precision and recall.

## 5 Preliminary Results

The following is a description of the initial experiments we conducted to evaluate the quality of Movie Lens. The experiments were performed to show: (i) the quality of the identified partition by the K-Means algorithm; (ii) if the identified patterns hold any insight in relation to what is happening in the video. We tested different configurations to perform its evaluation and configuration. Thanks to Ward's hierarchical clustering technique and the Elbow method, we were able to correctly configure the K-Means algorithm. Figure 3a 3b shows the obtained results. The height parameter in Ward's dendrogram grows with the intra-cluster variance. Table 1 shows the performance of the MLP with 4, 8, 16 and 32 classes, in terms of accuracy, f1-score macro average and f1-score weighted average.




(a) Ward's Dendrogram. (b) Elbow Method

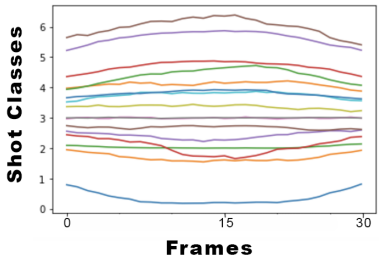
Fig. 3: Different strategies to identify the optimal number of clusters.

Table 1: Performance of the MLP classifier with a different amount of classes considered.

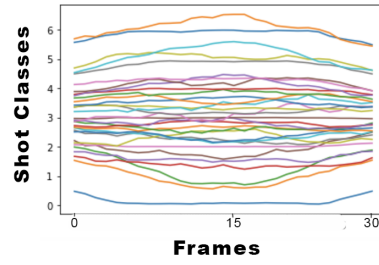
| number of classes | overall accuracy | macro average | weighted average |
|---|---|---|---|
| 4 | 93 | 93 | 93 |
| 8 | 88 | 88 | 88 |
| 16 | 81 | 79 | 81 |
| 32 | 77 | 72 | 77 |

With 4 classes our classifier achieves a good performance, however if we take a look at the elbow graph we can see that the number of classes can be incremented. With 8 classes the classifier reaches a slightly lower performance but still satisfying. However, since our sequences are composed of eight elements and our labels are defined in relation to how much every sequence is distant from the 8 artificial sequences, we wanted to see if by increasing the number of classes considered we were able to find more fine-grained editing patterns. Hence we tested MOVIE LENS with also 16 and 32 classes. Figures 4a and 4b show the average pattern per class, i.e. centroid identified in these last two scenarios. By analyzing the patterns in Figure 4a we can see that centroids modeling editing pattern groups are well-separated with respect to the ones in Figure 4b. Thus, fine-grained partitions obtained with 32 classes are too detailed and present overlapped centroids (i.e., some types of frames in the sequences are very similar).

Hence, in the following we detail the result obtained with 16 classes. The experiments presented here were performed using a stratified K-fold cross-validation with a K value of 10. With 16 classes, the MLP achieves a training accuracy of 88% after 50 epochs and a validation accuracy between 78% and 83%, with an average value of 81.2%. Table 2 shows precision, recall, and f-1 score of the average model.




(a) Editing patterns with 16 classes.




(b) Editing patterns with 32 classes.

Fig. 4: These patterns, one for each class, are obtained by averaging all the sequences belonging to the same group.

Table 2: Precision, recall and f1-score of the MLP with 16 labels.

| Labels | Semantic Label | Precision | Recall | f1-Score | Support |
|---|---|---|---|---|---|
| 0 | CER | 66% | 61% | 63% | 130 |
| 1 | CER | 73% | 79% | 76% | 109 |
| 2 | N | **99%** | **97%** | **98%** | 177 |
| 3 | CER | 76% | 72% | 74% | 101 |
| 4 | CER | 88% | 86% | 87% | 140 |
| 5 | ED | 96% | 64% | 77% | 77 |
| 6 | CCI | 85% | 82% | 83% | 197 |
| 7 | CCI | **97%** | **95%** | **96%** | 369 |
| 8 | N | 67% | 61% | 64% | 127 |
| 9 | N | 69% | 83% | 75% | 237 |
| 10 | CCI | **99%** | **95%** | **97%** | 155 |
| 11 | ED | 79% | 71% | 75% | 21 |
| 12 | ED | 87% | 92% | 89% | 118 |
| 13 | N | 56% | 77% | 65% | 119 |
| 14 | N | 67% | 57% | 62% | 136 |
| 15 | N | 77% | 75% | 76% | 175 |
| accuracy | | | | 81% | 2388 |
| macro avg | | 80% | 78% | 79% | 2388 |
| weighted avg | | 82% | 81% | 81% | 2388 |

From the classification report shown in Table 2 we can see that the proposed model performs particularly well in some classes. In a first analysis, with the help of a domain expert we found some insightful patterns allowing us to define three semantic labels (as shown in Table 2): (1) *Character-Environment Relationship (CER)* , (2) *Environment Descriptions (ED)*, and (3) *Character-Character Interaction (CCI)* discussed in the following subsections. Furthermore, an additional label, named *undefined (N)* has been defined to group all ill-defined classes.

### 5.1 Character-Environment Relationship

The classes in this first semantic group are 0, 1, 3 and 4, as shown in Table 2. Class 0 contains mainly medium close ups and medium long shots. On class 0 the MLP was able to reach an f1-score of 63%. It is one of the lowest scores in terms of performance and it is mainly due to the presence of two types of sequences. The first one consists of dialogues where there is not too much involvement with the characters or in their reaction. The focus of the viewer in these sequences is divided between the characters and the environment in which the scene is set. The other main sequence branch is also centered around characters and the environment, but represents it from a different perspective. In these sequences there is the camera that follows the subject exploring or moving through the scene environment. Some sequences show a mixture of both patterns. For instance two characters may be talking about a friend of theirs in jail and then we see a flashback in which we follow the third character while he is being arrested.

Also class 1 (f1-score=76%) encapsulates sequences focused on the relationship between characters and the environment. These sequences contain mainly close ups mixed with wider shots. We have found one relevant pattern. The analyzed clips showed mainly how the characters react to changes in the environment. The close ups are used to see the character facial expression, while the wider shots are used to show what is happening. Whether the character interacts with the environment producing a change to which he reacts to or the environment changes by itself they describe action and reaction between the character and the environment. Class 3 (f1-score=74%) is conceptually similar. The typologies of shot used are similar, although in different proportions. Also here what is shown is the relationship between character and environment. The difference from the previous class is that here the close ups do not show only the character reactions but also objects of the surrounding environment. By using close up to show also details of the environment changes the viewer's perception of the story. Yet another class that focuses on the relationship between character and environment is class 4 (f1-score=87%). However, here the mood is different. Usually they represent a unique shot, with a fixed shot or a camera movement. Here the characters are presented with the environment itself, whether they are doing nothing or talking to each other. Most of the analyzed sequences depicted scenes in an internal setting, as shown in Figure 5.

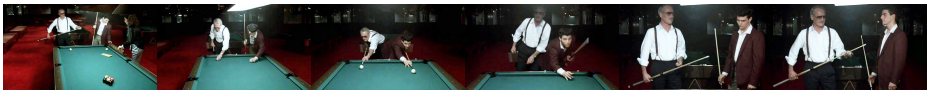


Fig. 5: Character and Environment Introduction. For visualization purposes we show 1 frame every 5.



Fig. 6: Environment Sequence. For visualization purposes we show 1 frame every 3.

### 5.2 Environment Description

The classes in this semantic group are 5, 11 and 12 (see Table 2). These groups focus on the environment description, what changes mainly is the shot scale used. For instance, class 12 (f1-score=89%) encapsulates sequences that are mainly composed of medium long shots. This class contains mainly sequence shots. Sequence shot, which is not to not be confused with sequence of shots or shots sequence, is a sequence composed of a unique clip. In other words there are no cuts. These sequence shots are either stationary, or follow at a fixed distance a character that moves around the environment. Sequences belonging to class 11 (f1-score=75%) contain mainly long shots or extreme long shots. Here the focus is completely on the Environment, if there are subjects on the screen, the viewer's attention dedicated to them is minimal. Figure 6 shows an example of this type of sequence taken from "Bande à Part". Class 5 (f1-score=77%) uses more narrow shots. compared to class 11. Nonetheless also these are environment descriptive sequences. They tend to be sequence shots that describe the environment. If there are characters framed, the viewer gives a little more attention to them. However there is no emphasis on what the characters are doing and the main focus remains on the environment.

### 5.3 Character-Character Interaction



Fig. 7: Dialogue Sequence. For visualization purposes we show 1 frame every 3.

This main category, including groups 6, 7, and 10, as shown in Table 2, focuses on dialogue among characters. Class 7 (f1-score=96%) contains mainly medium close up sequences. This is the class with the highest amount of sequences. All of

these sequences are dialogues. Unfortunately all of these sequences look similar so it is difficult to extract more meaningful patterns (more on this in section 6). Figure 7 shows a sequence belonging to this category.

Class 10 (f1-score=97%) instead identifies a more specific pattern. In this class there are a lot of close ups. Usually there are two, three at most characters involved in this type of sequence. The interaction between characters is one to one. Instead class 15 (f1-score=76%) contains dialogues among more characters. Usually there is the main character talking while the others listen or vice versa. The interactions among characters are one to many. In fact this class contains wider shots compared to class 10.

Class 6 (f1-score=83%) is a class that contains a lot of foreground shots and narrow shots. Even if the sequences are a little noisy in this class there are dialogues, not too centered on the character's facial emotional reaction but also his physical reaction. To follow the character's physical reaction we follow him at a fixed distance using camera movements, labeled on Cinescale as foreground shots.

### 5.4 Undefined Classes

Classes 8-9-13-14-15 (see Table 2) are ill-defined since no specific and no common meaningful patterns can be identified. For classes 8 (f1-score=64%), 13 (f1-score=65%) and 14 (f1-score=62%) we were not able to extract any type of knowledge. This is probably related to the fact that these classes share similar shot types.

For class 2 (f1-score=98%) the issue is different. This class contains mainly foreground shots. Foreground shots usually describe camera movements in which the shot scale changes. However there is no difference if the shot scale changes from a close up to a medium shot or from a medium shot to a long shot. However this type of information impacts the viewer's perception. Generally speaking this class encapsulates sequences that contain only camera movements, but for now it is not possible to have a more fine-grained characterization. For this reason we have chosen to include this group of sequences in the undefined classes.

### 5.5 Misclassified Sequences

We have conducted a preliminary characterization also on the wrongly classified sequences. Here we have reported the analysis conducted on the 5 main misclassification errors, identified through the confusion matrix (not reported here to lack of space). Specifically, misclassified patterns belong to classes 5-0-6-8-14 predicted as 13-9-15-13-9, respectively. These errors mainly arise since the wrong predicted classes are ill-defined, thus the classifier is not so robust to clearly distinguish some editing patterns. A large and comprehensive data set should be considered so that the classifier can better model each specific group and perform the classification easily and correctly.

## 6 Discussion

This paper proposes a data-driven approach to effectively address editing pattern characterization. From our preliminary characterization, we were able to discern some interesting fine-grained editing patterns. Although these preliminary results are promising, there is room for improvements. Some research directions to be considered are discussed in the following: (1) A *more fine-grained shot type classification* modeling more cinematographic shot categories. This would mean having classes for camera movements, rather than the general class for foreground shots. Also, a distinction between close ups of a character's face and objects would allow for more insightful analysis. (2) A *characterization of single clip versus multiple clips*. It should categorize sequences obtained from a single clip from those composed of different video files. It could improve the MOVIE LENS overall performance. (3) *Dataset enrichment with video metadata*. Additional movie metadata, such as the starting and ending point of single scenes, could help the editing pattern characterization. (4) *Extension of the* MOVIE LENS *methodology with new ML algorithms*. The proposed approach could be enriched with other deep learning methods. For example the LSTM [9] networks or graph neural networks [5] could be integrated to take into account the temporal dimension in the analytics pipeline.

## References

1. Bain, M., Nagrani, A., Brown, A., Zisserman, A.: Condensed movies: Story based retrieval with contextual embeddings. CoRR **abs/2005.04208** (2020), https://arxiv.org/abs/2005.04208
2. Bak, H.Y., Park, S.B.: Comparative study of movie shot classification based on semantic segmentation. Applied Sciences **10**, 3390 (05 2020). https://doi.org/10.3390/app10103390
3. Benini, S., Savardi, M., Balint, K., Kovacs, A., Signoroni, A.: On the influence of shot scale on film mood and narrative engagement in film viewers. IEEE Transactions on Affective Computing **13**(2), 592–603 (2022). https://doi.org/10.1109/taffc.2019.2939251
4. Berthouzoz, F., Li, W., Agrawala, M.: Tools for placing cuts and transitions in interview video. ACM Transactions on Graphics **31**, 1–8 (07 2012). https://doi.org/10.1145/2185520.2335418
5. Bloemheuvel, S., van den Hoogen, J., Jozinovic, D., Michelini, A., Atzmueller, M.: Multivariate time series regression with graph neural networks. CoRR **abs/2201.00818** (2022), https://arxiv.org/abs/2201.00818
6. Chakraborty, S., Nagwani, N., Dey, L.: Performance comparison of incremental k-means and incremental dbscan algorithms. International Journal of Computer Applications (0975 – 8887) **27**, 975–8887 (08 2011)
7. Haldar, R., Mukhopadhyay, D.: Levenshtein distance technique in dictionary lookup methods: An improved approach. Computing Research Repository - CORR (01 2011)
8. Hasan, M.A., Xu, M., He, X., Xu, C.: Camhid: Camera motion histogram descriptor and its application to cinematographic shot classification. IEEE Transactions on Circuits and Systems for Video Technology **24**(10), 1682–1695 (2014). https://doi.org/10.1109/TCSVT.2014.2345933

9. He, Z., Gao, S., Xiao, L., Liu, D., He, H., Barber, D.: Wider and deeper, cheaper and faster: Tensorized lstms for sequence learning (12 2017)
10. Jani, K., Chaudhuri, M., Patel, H., Shah, M.: Machine learning in films: an approach towards automation in film censoring. Journal of Data, Information and Management **2** (03 2020). https://doi.org/10.1007/s42488-019-00016-9
11. Juang, B.H., Rabiner, L.: The segmental k-means algorithm for estimating parameters of hidden markov models. IEEE Transactions on Acoustics, Speech and Signal Processing **38**(9), 1639–1641 (Sep 1990)
12. Liberti, L., Lavor, C., Maculan, N., Mucherino, A.: Euclidean distance geometry and applications. SIAM Review **56** (05 2012). https://doi.org/10.1137/120875909
13. Matsuo, Y., Amano, M., Uehara, K.: Mining video editing rules in video streams. pp. 255–258 (01 2002). https://doi.org/10.1145/641007.641058
14. Mogadala, A., Kalimuthu, M., Klakow, D.: Trends in integration of vision and language research: A survey of tasks, datasets, and methods. J. Artif. Int. Res. **71**, 1183–1317 (sep 2021). https://doi.org/10.1613/jair.1.11688, https://doi.org/10.1613/jair.1.11688
15. Murch, W.: In the Blink of an Eye. Silman-James Press (2001)
16. Nothelfer, C., DeLong, J., Cutting, J.E.: Shot structure in hollywood film (2009)
17. Podlesnyy, S.: Towards data-driven automatic video editing (07 2019)
18. Qaisar, S.: Sentiment analysis of imdb movie reviews using long short-term memory (11 2020). https://doi.org/10.1109/ICCIS49240.2020.9257657
19. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation (2021). https://doi.org/10.48550/ARXIV.2102.12092, https://arxiv.org/abs/2102.12092
20. Ren, J., Shen, X., Lin, Z., Měch, R.: Best frame selection in a short video. In: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 3201–3210 (2020). https://doi.org/10.1109/WACV45572.2020.9093615
21. Savardi, M., Kovács, A.B., Signoroni, A., Benini, S.: Cinescale: A dataset of cinematic shot scale in movies. Data in Brief **36** (2021)
22. Savardi, M., Signoroni, A., Migliorati, P., Benini, S.: Shot scale analysis in movies by convolutional neural networks. pp. 2620–2624 (10 2018). https://doi.org/10.1109/ICIP.2018.8451474
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014), http://arxiv.org/abs/1409.1556
24. Simões, G., Wehrmann, J., Barros, R., Ruiz, D.: Movie genre classification with convolutional neural networks. pp. 259–266 (07 2016). https://doi.org/10.1109/IJCNN.2016.7727207
25. Soe, T.H.: Automation in video editing: Assisted workflows in video editing. In: AutomationXP@CHI (2021)
26. Svanera, M., Savardi, M., Signoroni, A., Kovács, A.B., Benini, S.: Who is the film's director? authorship recognition based on shot features. IEEE MultiMedia **26**(4), 43–54 (2019). https://doi.org/10.1109/MMUL.2019.2940004
27. Vacchetti, B., Cerquitelli, T.: Cinematographic shot classification with deep ensemble learning. Electronics **11**(10), 1570 (2022)
28. Vacchetti, B., Cerquitelli, T., Antonino, R.: Cinematographic shot classification through deep learning. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC). pp. 345–350 (2020). https://doi.org/10.1109/COMPSAC48688.2020.0-222
29. Walters, A.: Sentence classification, https://github.com/lettergram/sentence-classification

30. Wu, H.Y., Santarra, T., Leece, M., Vargas, R., Jhala, A.: Joint attention for automated video editing. In: ACM International Conference on Interactive Media Experiences. p. 55–64. IMX '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3391614.3393656, https://doi.org/10.1145/3391614.3393656
31. Zhang, X., Li, Y., Han, Y., Wen, J.: Ai video editing: a survey (12 2021). https://doi.org/10.20944/preprints202201.0016.v1
32. Zhou, H., Hermans, T., Karandikar, A., Rehg, J.: Movie genre classification via scene categorization. pp. 747–750 (10 2010). https://doi.org/10.1145/1873951.1874068
33. Zhou, J., Zhang, X.P.: Automatic identification of digital video based on shot-level sequence matching. In: Proceedings of the 13th Annual ACM International Conference on Multimedia. p. 515–518. MULTIMEDIA '05, Association for Computing Machinery, New York, NY, USA (2005). https://doi.org/10.1145/1101149.1101265, https://doi.org/10.1145/1101149.1101265