MovieNet: A Holistic Dataset for Movie Understanding *Supplementary Material*

Qingqiu Huang*, Yu Xiong*, Anyi Rao, Jiaze Wang, and Dahua Lin

CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong
{hq016, xy017, ra018, dhlin}@ie.cuhk.edu.hk
jzwang@link.cuhk.edu.hk

Overview of Supplementary Material

In this supplementary material, we provide overall details about MovieNet, including data, annotation, experiments and the toolbox. The content is organized as follows:

(1) We provide details about the content of particular data and how to collect and clean them in Sec. 1:

– **Meta Data.** The list of meta data is given followed by the content of these meta data. See Sec. 1.1.
– **Movie.** The statistics of the movies are provided. See Sec. 1.2
– **Subtitle.** The collection and post-processing procedure of obtaining and aligning subtitles are given. See Sec. 1.3.
– **Trailer.** We provide the process of selecting and processing the trailers. See Sec. 1.4.
– **Script.** We automatically align the scripts to movies. The details of the method will be presented. See Sec. 1.5.
– **Synopsis.** The statistics of synopsis will be introduced. See Sec. 1.6.
– **Photo.** The statistics and some examples of photo will be shown. See Sec. 1.7.

(2) We demonstrate annotation in MovieNet with the description about the design of annotation interface and workflow, see Sec. 2.

– **Character Bounding Box and Identity.** We provide step by step procedure of collecting images and annotating the images with a semi-automatic algorithm. See Sec. 2.1.
– **Cinematic Styles.** We present the analytics on cinematic styles and introduce the workflow and interface of annotating cinematic styles. See Sec. 2.2.
– **Scene Boundaries.** We demonstrate how to effectively annotate scene boundaries with the help of an optimized annotating workflow. See Sec. 2.3.
– **Action and Place Tags.** We describe the procedure of jointly labeling the action and place tags over movie segments. The workflow and interface are presented. See Sec. 2.4.

* Equal contribution

- **Synopsis Alignment.** We provide the introduction of an efficient coarse-to-fine annotating workflow to align a synopsis paragraph to a movie segment. See Sec. 2.5.
- **Trailer Movie Alignment.** We introduce a automatic approach that align shots in trailers to the original movies. This annotation facilitate tasks like trailer generation. See Sec. 2.6.

(3) We set up several benchmarks on our MovieNet and conduct experiments on each benchmark. The implementation details of experiments on each benchmark will be introduced in Sec. 3:

- **Genre Classification.** Genre Classification is a multi-label classification task build on MovieNet genre classification benchmark. See details at Sec. 3.1.
- **Cinematic Styles Analysis.** On MovieNet cinematic style prediction benchmark, there are two classification tasks, namely *scale classification* and *movement classification*. See Sec. 3.2 for implementation details.
- **Character Detection.** We introduce the detection task as well as model, implementation details on MovieNet character detection benchmarks. See Sec. 3.3.
- **Character Identification.** We further introduce the challenging benchmark setting for MovieNet character identification. See details in Sec. 3.4.
- **Scene Segmentation.** The scene segmentation task is a boundary detection task for cutting the movie by scene. The details about feature extraction, baseline models and evaluation protocols will be introduced in Sec. 3.5.
- **Action Recognition.** We present the task of multi-label action classification task on MovieNet with the details of baseline models and experimental results. See Sec. 3.6.
- **Place Recognition.** Similarly, we present the task of multi-label place classification task on MovieNet. See Sec. 3.7.
- **Story Understanding.** For story understanding, we leverage the benchmark MovieNet segment retrieval to explore the potential of overall analytics using different aspects of MovieNet. The experimental settings and results will be found in Sec. 3.8.

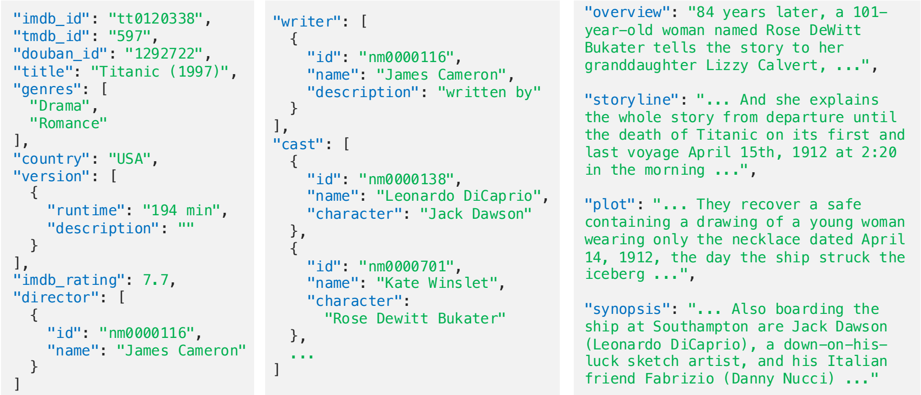
(4) To manage all the data and provide support for all the benchmarks, we build up a codebase for managing MovieNet with handy processing tools. Besides the codes for the benchmarks, we would also release this toolbox, the features of this tool box are introduced in Sec. 4

1 Data in MovieNet

MovieNet contains various kinds of data from multiple modalities and high-quality annotations on different aspects for movie understanding. They are introduced in detail below. And for comparison, the overall comparison of the data in MovieNet with other related dataset are shown in Tab. 1.

Table 1: Comparison between MovieNet and related datasets in terms of data.

	movie	trailer	photo	meta	genre	script	synop.	subtitle	plot	AD
MovieScope [4]	-	5,027	5,027	5,027	13K	-	-	-	5,027	-
MovieQA [36]	140	-	-	-	-	-	-	408	408	-
LSMDC [31]	200	-	-	-	-	50	-	-	-	186
MovieGraphs [39]	51	-	-	-	-	-	-	-	-	-
AVA [16]	430	-	-	-	-	-	-	-	-	-
MovieNet	1,100	60K	3.9M	375K	805K	986	31K	5,388	46K	-



```
"imdb_id": "tt0120338",
"tmdb_id": "597",
"douban_id": "1292722",
"title": "Titanic (1997)",
"genres": [
  "Drama",
  "Romance"
],
"country": "USA",
"version": [
  {
    "runtime": "194 min",
    "description": ""
  }
],
"imdb_rating": 7.7,
"director": [
  {
    "id": "nm0000116",
    "name": "James Cameron"
  }
]
```

```
"writer": [
  {
    "id": "nm0000116",
    "name": "James Cameron",
    "description": "written by"
  }
],
"cast": [
  {
    "id": "nm0000138",
    "name": "Leonardo DiCaprio",
    "character": "Jack Dawson"
  },
  {
    "id": "nm0000701",
    "name": "Kate Winslet",
    "character":
      "Rose Dewitt Bukater"
  },
  ...
]
```

```
"overview": "84 years later, a 101-
year-old woman named Rose DeWitt
Bukater tells the story to her
granddaughter Lizzy Calvert, ...",

"storyline": "... And she explains
the whole story from departure until
the death of Titanic on its first and
last voyage April 15th, 1912 at 2:20
in the morning ...",

"plot": "... They recover a safe
containing a drawing of a young woman
wearing only the necklace dated April
14, 1912, the day the ship struck the
iceberg ...",

"synopsis": "... Also boarding the
ship at Southampton are Jack Dawson
(Leonardo DiCaprio), a down-on-his-
luck sketch artist, and his Italian
friend Fabrizio (Danny Nucci) ..."
```

Fig. 1: A sample of metadata from the movie *Titanic*.

1.1 Meta Data

MovieNet contains meta data of $375K$ movies. Note that the number of metadata is significantly large than the movies provided with video sources (*i.e.* $1,100$) because we belief that metadata itself can support various of tasks. It is also worth noting that the metadata of all the $1,100$ selected movies are included in this metadata set. Fig. 1 shows a sample of the meta data, which is from *Titanic*. More details of each item in the meta data would be introduced below.

- **IMDb ID.** IMDb ID is the ID of a movie in the IMDb website[1]. IMDb ID is usually a string begins with "tt" and follows with 7 or 8 digital numbers, *e.g.* "tt0120338" for the movie *Titanic*. One can easily get some information of a movie from IDMb with its ID. For example, the homepage of *Titanic* is "https://www.imdb.com/title/tt0120338/". The IMDb ID is also taken as the ID of a movie in MovieNet.
- **TMDb ID.** TMDb ID is the ID of a movie in the TMDb website[2]. We find that some of the content in TMDb is of higher-quality than IMDb. For example, TMDb provides different versions of trailers and higher resolution

[1] https://www.imdb.com/

[2] https://www.themoviedb.org/

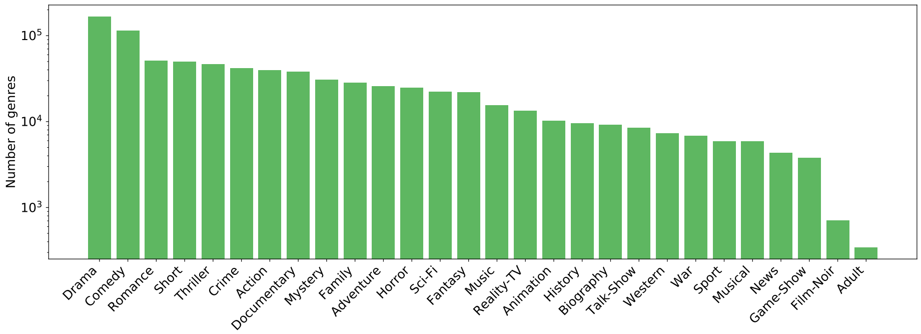


Fig. 2: Statistics of genres in metadata. It shows the number of genres for each genre category (y-axis in log-scale).

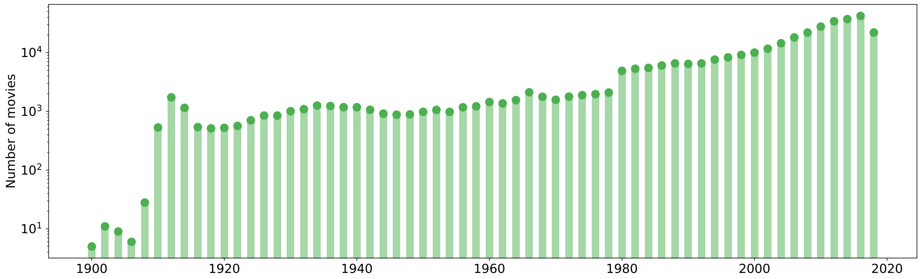


Fig. 3: Distribution of release date of the movies in metadata. It shows the number of movies in each year (y-axis in log-scale). Note that the number of movies generally increases as time goes by.

posters. Therefore, we take it as a supplement of IMDb. TMDb provides APIs for users to search for information. With the TMDb ID provided in MovieNet, one can easily get more information if needed.

- **Douban ID.** Douban ID is the ID of a movie in Douban Movie[3]. We find that for some Asian movies, such as those from China and Japan, IMDb and TMDb contains few information. Therefore, we turn to a Chinese movie website, namely Douban Movie, for more information of Asian movies. We also provide Douban ID for some of the movies in MovieNet for convenience.
- **Version.** For movie with over one versions, *e.g.* normal version, director's cut, we provide runtime and description of each version to help researchers align the annotations with their own resources.
- **Title.** The title of a movie following the format of IMDb, *e.g.*, *Titanic (1997)*.
- **Genres.** Genre is a category basedon similarities either in the narrative elements or in the emotional response to the movie, *e.g.*, comedy, drama. There are totally 28 unique genres from the movies in MovieNet. Fig. 2 shows the distribution of the genres.

[3] https://movie.douban.com/

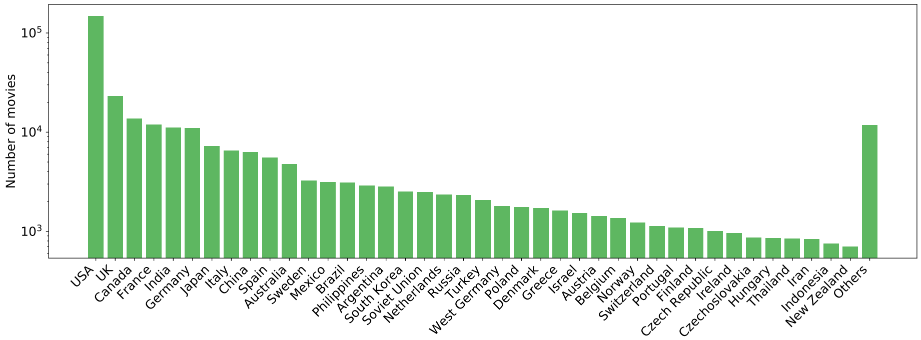


Fig. 4: The countries that the movies belong to in metadata. Here we show top 40 countries with the left as "Others". The number of movies (y-axis) is in log-scale.

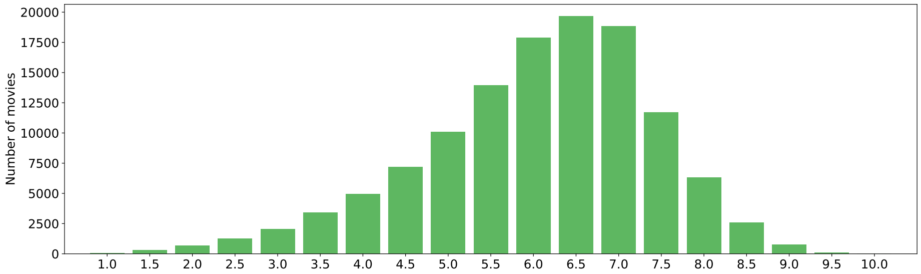
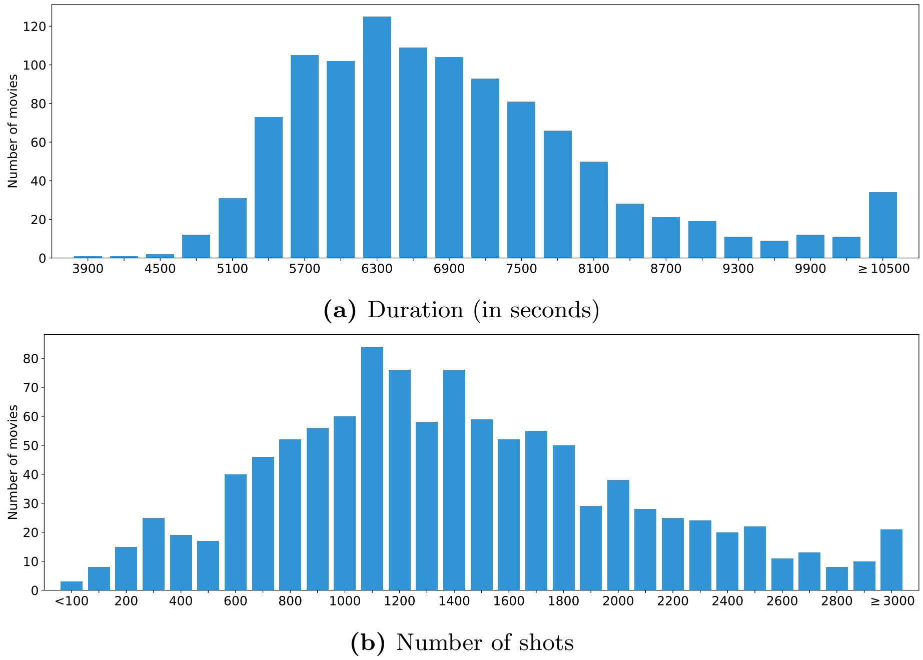


Fig. 5: Distribution of score ratings in MovieNet metadata.

- **Release Date.** Release Date is the date when the movie published. Fig 3 shows the number of the movies released every year, from which we can see that the number of movies continuously grows every year.
- **Country.** Country refers to the country where the movie produced. The top-40 countries of the movies in MovieNet are shown in Fig. 4.
- **Version.** A movie may have multiple versions, *e.g.*, director's cut, special edition. And different versions would have different runtimes. Here we provide the runtimes and descriptions of the movies in MovieNet.
- **IMDb Rating.** IMDb rating is the rating of the movie uploaded by the users. The distribution of different ratings are shown in Fig. 5.
- **Director.** Director contains the director's name and ID.
- **Writer.** Writer contains the writer's name and ID.
- **Cast.** A list of the cast in the movie, each of which contains the actor/actress's name, ID and character's name.
- **Overview.** Overview is a brief introduction of the movie, which usually covers the background and main characters of the movie.
- **Storyline.** Storyline is a plot summary of the movie. It is longer and contains more details than the overview.



(a) Duration (in seconds)

(b) Number of shots

Fig. 6: Distribution of duration and number of shots for the 1, 100 movies in MovieNet.

- **Wiki Plot.** Wiki Plot is the summary of the movie from Wikipedia and is usually longer than overview and storyline.

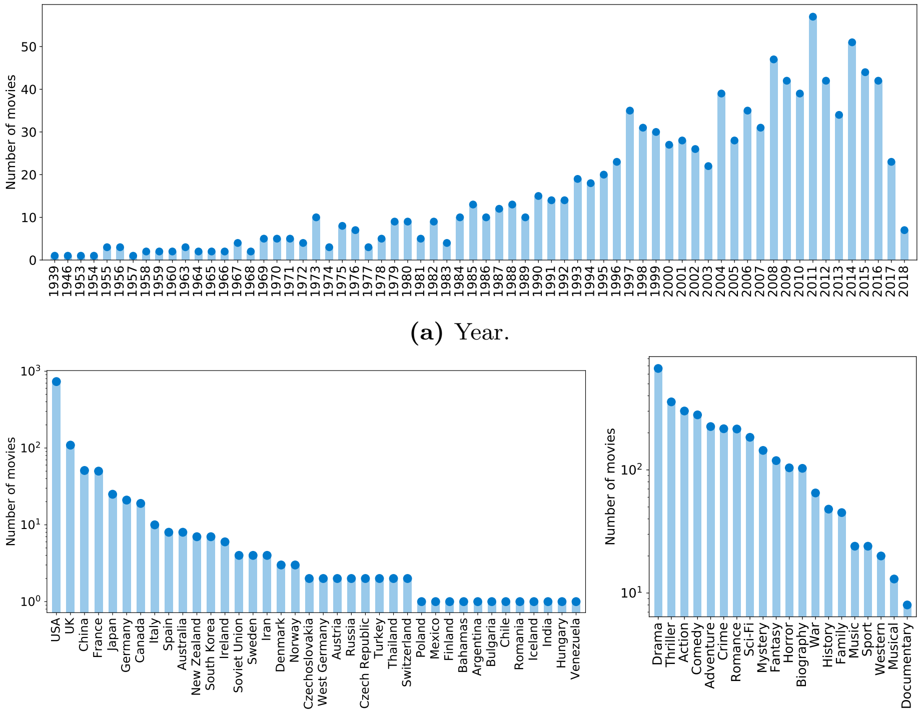
1.2 Movie

As we introduced in our paper, there are 1, 100 movies in MovieNet. Here we show some statistics of the 1, 100 movies in Fig. 6, including the distributions of runtime and the shot number. As mentioned in Sec. 1.1, in addition to these 1, 100 movies, we also provided metadata for other movies as much as we can. This also apply for other data like trailer and photo, and we would not clarify it in the next sections.

It is mentioned in the paper that we select the movie that covers a wide range of years, countries and genres. The distribution of these data are shown in Fig. 7. We can see that the movies are diversity in terms of year, country and genre.

Feature Representation. To play with a long video is nontrivial for the current deep learning framework and computational power. For the convenience of research, we propose multiply ways of feature representations for a movie.

- **Shot-based visual feature.** For most of the task, *e.g.* genre classification, shot-based representation is an efficient representation. A shot is a series of



(a) Year.

(b) Country.

(c) Genre.

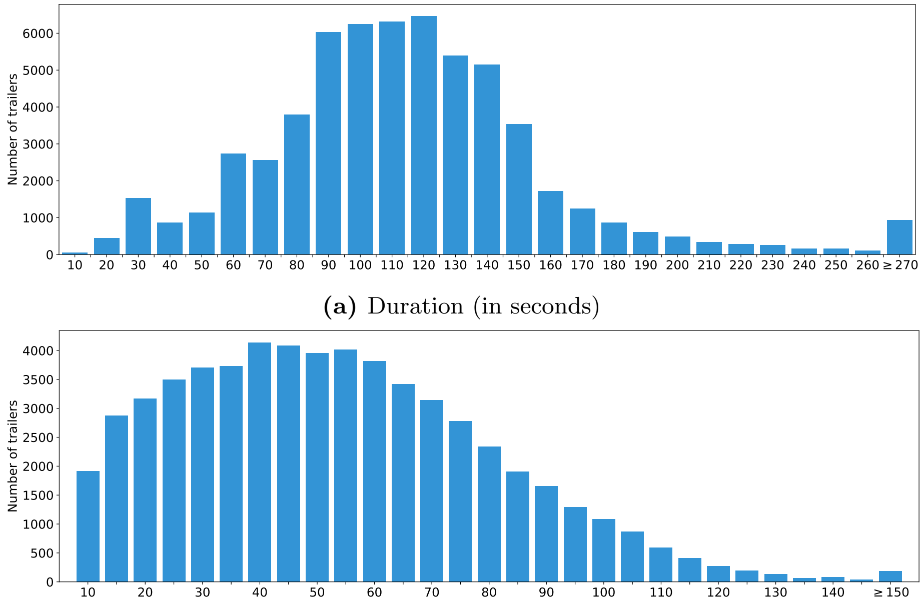
Fig. 7: Distribution of release year, countries and genres for the 1, 100 movies in MovieNet (y-axis of country and genre in log scale).

frames that runs for an uninterrupted period of time, which can be taken as the smallest visual unit of a movie. So we use shot-based representation for movies in our MovieNet. Specifically, we first separate each movie into shots with a shot detection tool [35]. Then, we sample three key frames and extract visual features using models pre-trained on ImageNet.

- **Audio feature.** For each shot, we also cut the audio wave within this shot and then extract audio feature [7] as the supplementary of visual feature.
- **Frame-based feature.** For those tasks like action recognition that need consider motion information, we also provide frame-based feature for these tasks.

1.3 Subtitle

For each movie from MovieNet, we provide an English subtitle that aligned with the movie. It is often the case that the downloaded subtitle is not aligned with the video because usually a movie has multiple versions, *e.g. director's cut* and *extended*. To make sure that each subtitle is aligned with the video source, before



(a) Duration (in seconds)

(b) Number of shots

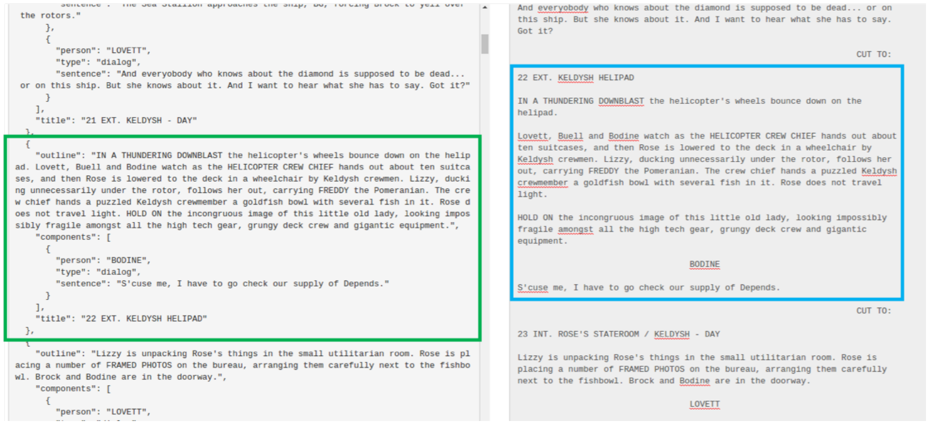
Fig. 8: Distribution of duration and number of shots for the trailers in MovieNet.

manually checking, we make the following efforts: (1) For the subtitle extracted from original video or downloaded from the Internet, we first make sure the subtitles are complete and are English version (by applying regular expression). (2) Then we clean the subtitle by removing noise such as HTML tags. (3) We leverage the off-the-shelf tool[4] that transfers audio to text and matches text with the subtitle to produce a shift time. (4) We filtered out those subtitles with a shift time surpasses a particular threshold and download another subtitle. After that, we manually download the subtitles that are still problematic, and then repeat the above steps.

Particularly, the threshold in step(4) is set to 60s by the following observations: (1) Most of the aligned subtitles have a shift within 1 seconds. (2) Some special cases, for example, a long scene without any dialog, would cause the tool to generate a shift of a few seconds. But the shift is usually less than 60s. (3) The subtitles that do not align with the original movies are usually either another version or crawled from another movie. In such cases, the shift will be larger than 60s.

After that, we ask annotators to manually check if the auto aligned subtitles are still misaligned. It turns out that the auto alignment are quite effective that few of the subtitles are still problematic.

[4] https://github.com/smacke/subsync



```
        "sentence": "The Sea Stallion approaches the ship, BG, forcing Brock to yell over
 the rotors."
      },
      {
        "person": "LOVETT",
        "type": "dialog",
        "sentence": "And everyobody who knows about the diamond is supposed to be dead...
 or on this ship. But she knows about it. And I want to hear what she has to say. Got it?"
      }
    ],
    "title": "21 EXT. KELDYSH - DAY"
  },
  {
    "outline": "IN A THUNDERING DOWNBLAST the helicopter's wheels bounce down on the helip
ad. Lovett, Buell and Bodine watch as the HELICOPTER CREW CHIEF hands out about ten suitca
ses, and then Rose is lowered to the deck in a wheelchair by Keldysh crewmen. Lizzy, ducki
ng unnecessarily under the rotor, follows her out, carrying FREDDY the Pomeranian. The cre
w chief hands a puzzled Keldysh crewmember a goldfish bowl with several fish in it. Rose d
oes not travel light. HOLD ON the incongruous image of this little old lady, looking impos
sibly fragile amongst all the high tech gear, grungy deck crew and gigantic equipment.",
    "components": [
      {
        "person": "BODINE",
        "type": "dialog",
        "sentence": "S'cuse me, I have to go check our supply of Depends."
      }
    ],
    "title": "22 EXT. KELDYSH HELIPAD"
  },
  {
    "outline": "Lizzy is unpacking Rose's things in the small utilitarian room. Rose is pl
acing a number of FRAMED PHOTOS on the bureau, arranging them carefully next to the fishbo
wl. Brock and Bodine are in the doorway.",
    "components": [
      {
        "person": "LOVETT",
```

```
And everyobody who knows about the diamond is supposed to be dead... or on
this ship. But she knows about it. And I want to hear what she has to say.
Got it?

                                                                  CUT TO:

22 EXT. KELDYSH HELIPAD

IN A THUNDERING DOWNBLAST the helicopter's wheels bounce down on the
helipad.

Lovett, Buell and Bodine watch as the HELICOPTER CREW CHIEF hands out about
ten suitcases, and then Rose is lowered to the deck in a wheelchair by
Keldysh crewmen. Lizzy, ducking unnecessarily under the rotor, follows her
out, carrying FREDDY the Pomeranian. The crew chief hands a puzzled Keldysh
crewmember a goldfish bowl with several fish in it. Rose does not travel
light.

HOLD ON the incongruous image of this little old lady, looking impossibly
fragile amongst all the high tech gear, grungy deck crew and gigantic
equipment.

                                  BODINE

S'cuse me, I have to go check our supply of Depends.
                                                                  CUT TO:

23 INT. ROSE'S STATEROOM / KELDYSH - DAY

Lizzy is unpacking Rose's things in the small utilitarian room. Rose is
placing a number of FRAMED PHOTOS on the bureau, arranging them carefully
next to the fishbowl. Brock and Bodine are in the doorway.

                                  LOVETT
```

Fig. 9: Here we show an example of the parsed script. The left block shows the formatted script snippet and the right block shows the corresponding raw ones of *"Titanic"*.

1.4 Trailer

There are $60K$ trailers from $33K$ unique movies in MovieNet. The statistics of the trailers are shown in Fig. 8, including the distributions of runtime and shot number.

Besides some attractive clips from the movie, which we name as content-shots, trailers usually contains exract shots to show some important information, *e.g.*, the name of the director, release date, *etc.*. We name these shots as *info-shots*. Info-shots are quite different from other shots since they contain less visual content. For most of the tasks with trailers, we usually focus on content-shots only. Therefore, it is necessary for us to distinguish info-shots and content-shots. We develop a simple approach to tackling this problem.

Given a shot, we first use a scene text detector [37] to detect the text on each frame. Then we generate a binary map of each frame, where the areas covered by the text bounding boxes are set to 1 and others are set to 0. Then we average all the binary maps of a shot and get a heat map. By average the heat map we get an overall score s to indicate how much text detected in a shot. The shot whose score is higher than a threshold α and a the average contrast is lower than β is taken as a info-shot in MovieNet. Here we take the contrast into consideration by the observation that info-shots usually have simple backgrounds.

1.5 Script

We provide aligned scripts in MovieNet. Here we introduce the details of script alignment. As mentioned in the paper, we align the movie script to movies by automatically matching the dialog with subtitles. This process is introduced below.

Algorithm 1 Script Alignment

```
INPUT: $\mathbf{S} \in \mathbb{R}^{M \times N}$
  $R \leftarrow Array(N)$
  $val \leftarrow Matrix(M, N)$
  $inds \leftarrow Matrix(M, N)$
  for $col \leftarrow 0, N-1$ do
      for $row \leftarrow 0, M-1$ do
          $a \leftarrow val[row, col-1] + \mathbf{S}[row, col]$
          $b \leftarrow val[row-1, col]$
          if $a > b$ then
              $inds[row, col] \leftarrow row$
              $val[row, col] \leftarrow a$
          else
              $inds[row, col] \leftarrow inds[row-1, col]$
              $val[row, col] \leftarrow b$
          end if
      end for
  end for
  $index \leftarrow M-1$
  for $col \leftarrow N-1, 0$ do
      $index \leftarrow inds[index, col]$
      $R \leftarrow index$
  end for
OUTPUT: $R$
```

Particularly, a movie script is a written work by filmmakers narrating the storyline and dialogs. It is useful for tasks like movie summarization. To obtain the data for these tasks, we need to align scripts to the movie timelines.

In the preprocessing stage, we develop a script parsing algorithm using regular expression matching to format a script as a list of scene cells, where scene cell denotes for the combination of a storyline snippet and a dialog snippet for a specific event. An example is shown in Fig. 9. To align each storyline snippet to the movie timeline, we choose to connect dialog snippet to subtitle first. To be specific, we formulate script-timeline alignment problem as an optimization problem for dialog-subtitle alignment. The idea comes from the observation that dialog is designed as the outline of subtitle.

Let dig_i denote the dialog snippet in i^{th} scene cell, sub_j denote the j^{th} subtitle sentence. We use TF-IDF [28] to extract text feature for dialog snippet and subtitle sentence. Let $f_i = \text{TF-IDF}(dig_i)$ denote the TF-IDF feature vector of i^{th} dialog snippet and $g_j = \text{TF-IDF}(sub_j)$ denote that of j^{th} subtitle sentence. For all the M dialog snippets and N subtitle sentences, the similarity matrix $\mathbf{S}$ is given by

$$s_{i,j} = \mathbf{S}(i, j) = \frac{f_i^T g_j}{|f_i||g_j|}$$

Table 2: Comparison on the statistics of wiki plot with that of synopsis.

	# sentence/movie	# word/sentence	# word/movie
wiki plot	26.2	23.6	618.6
synopsis	98.4	20.4	2004.7

For j^{th} subtitle sentence, we assume the index of matched dialog snippet i_j should be smaller than i_{j+1}, which is the index of matched dialog for $(j+1)^{th}$ subtitle sentence. By taking this assumption into account, we formulate the dialog-subtitle alignment problem as the following optimization problem,

$$\begin{aligned} \max_{i_j} \quad & \sum_{j=0}^{N-1} s_{i_j,j} \\ \text{s.t.} \quad & 0 \leq i_{j-1} \leq i_j \leq i_{j+1} \leq M-1. \end{aligned} \tag{1}$$

This can be effectively solved by dynamic programming algorithm. Let $L(p,q)$ denote the optimal value for the above optimization problem with $\mathbf{S}$ replaced by its submatrix $\mathbf{S}[0,\ldots,p,0,\ldots,q]$. The following equation holds,

$$L(i,j) = \max\{L(i,j-1), L(i-1,j) + s_{i,j}\}$$

It can be seen that the optimal value of the original problem is given by $L(M-1,N-1)$. To get the optimal solution, we apply the dynamic programing algorithm shown in Alg. 1. Once we obtain the connection between a dialog snippet and a subtitle sentence, we can directly assign the timestamp of the subtitle sentence to the script snippet who comes from the same scene cell as the dialog snippet. Fig. 10 shows the qualitative result of script alignment. It illustrates that our algorithm is able to draw the connection between storyline and timeline even without human assistance.

1.6 Synopsis

Here we show some statistics of synopsis in Tab. 2 and wordcloud visualization in Fig. 11. Compared to the wiki plot, we can see that synopsis is a higher-quality textual source which contains richer content and longer descriptions.

1.7 Photo

As we introduced in the paper, there are $3.9M$ photos from 7 types in MovieNet. Here we show the percentage of each type in Fig. 12. Also some samples are shown in Fig. 13.

2 Annotation in MovieNet

To achieve high-quality annotations, we have made great effort on designing the workflow and labeling interface, the details of which would be introduced below.

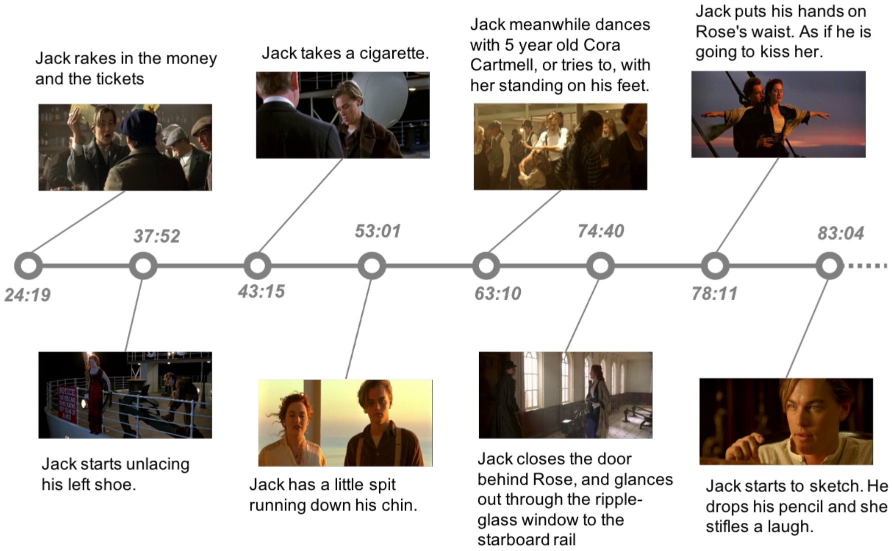

Jack rakes in the money and the tickets
Jack takes a cigarette.
Jack meanwhile dances with 5 year old Cora Cartmell, or tries to, with her standing on his feet.
Jack puts his hands on Rose's waist. As if he is going to kiss her.
24:19
37:52
43:15
53:01
63:10
74:40
78:11
83:04
Jack starts unlacing his left shoe.
Jack has a little spit running down his chin.
Jack closes the door behind Rose, and glances out through the ripple-glass window to the starboard rail
Jack starts to sketch. He drops his pencil and she stifles a laugh.

Fig. 10: Qualitative result of script alignment. The example comes from the movie *Titanic*. Each node marked by a timestamp is associated with a matched storyline snippet and a snapshot image.

2.1 Character Bounding Box and Identity

Workflow and Interface. Annotation of the character bounding box and identity follows six steps. (1) We first randomly choose $758K$ key frames from the provided movies. Here the key frames are extracted by average sampling three frames each shot. Then the annotators are asked to annotate the bounding box of the characters in the frames, after which we get $1.3M$ character bounding boxes. (2) With the $1.3M$ character bounding boxes, we train a character detector using implementations from [5,6]. Specifically, the detector is a Cascade R-CNN [2] with feature pyramid [22], using a ResNet-101 [18] as backbone. We find that the detector can achieve a 95% mAP. (3) Since the identities in different frames within one shot are usually duplicated, we choose only one frame from each shot to annotate the identities of the characters. We apply the detector to the key frames for identity annotation. Since the detetor performs good enough, we only manually clean the false positive boxes in this step. Resulting in $1.1M$ instances. (4) To annotate the identities in a movie is a challenging task due to the large variance in visual appearances. We develop a semi-automatic system for the first step of identity annotation to reduce cost. We first get the portrait of each cast from IMDb or TMDb, some of which are shown in Fig. 14. (5)We then extract the face feature with a face model trained on MS1M [17] and extract the body feature with a model trained on PIPA [43]. By calculating the feature similarity of the portrait and the instances in the movie, we sort the candidate list for each cast. And the annotator is then asked to determine whether each candidate

Fig. 11: Word cloud of synopsis corpus in MovieNet.

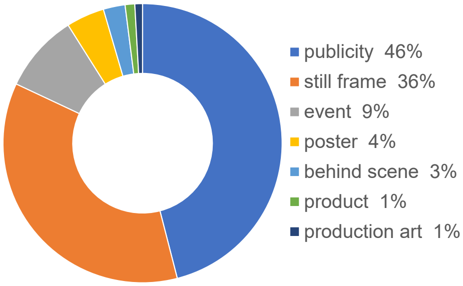


Fig. 12: Percentage of different photo types in MovieNet.

is the cast or not. Also, the candidate list would update after each annotation, which is similar to active learning. The interface is shown in Fig. 15. We find that this semi-automatic system can highly reduce the annotation cost. (6) Since the semi-automatic system may introduce some bias and noise, we further design a step for cleaning. At this step, the frames are demonstrated in time order and the annotating results at the first step are shown. The annotator can clean the results with temporal context.

Statistics and Samples. Here we show some statistics of the character annotation in Fig. 16, including the size distribution of bounding boxes and the distribution of instance number. From the statistics we can see that the number of character instance is a long-tail distribution. However, for those famous actors like *Leonardo Dicaprio*, they have much more character instances than others. Some samples are also shown in Fig. 14. We can see that MovieNet contains large-scale and diverse characters, which would be helpful for the researches on character analysis.

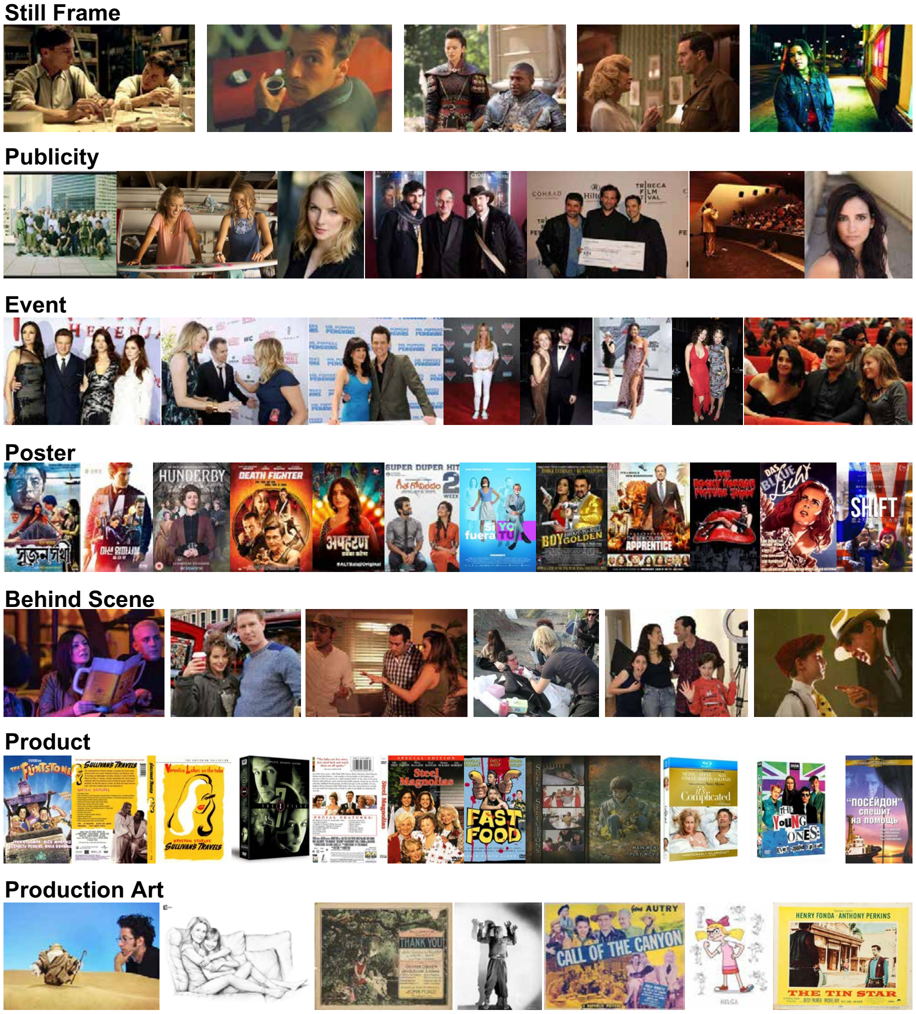


Fig. 13: Samples for different types of photos in MovieNet.

2.2 Cinematic Styles

We annotated the commonly used two kinds of cinematic tags of a shot [15]. Shot scale depict the portion of subject within the frames in a shot, while shot movement describe the camera movement or the lens change of a shot.

Shot Scale. Shot scale has 5 categories (as shown in Fig. 17): (1) *extreme close-up shot*: it only shows a very small part of a subject, *e.g.*, an eye or a mouth of a person; (2) *close-up shot*: it concentrates on a relatively small part of a subject, *e.g.*, the face of the hand of a person; (3) *medium shot*: it contains part of a subject *e.g.*, a figure from the knees or waist up; (4) *full shot*: it includes

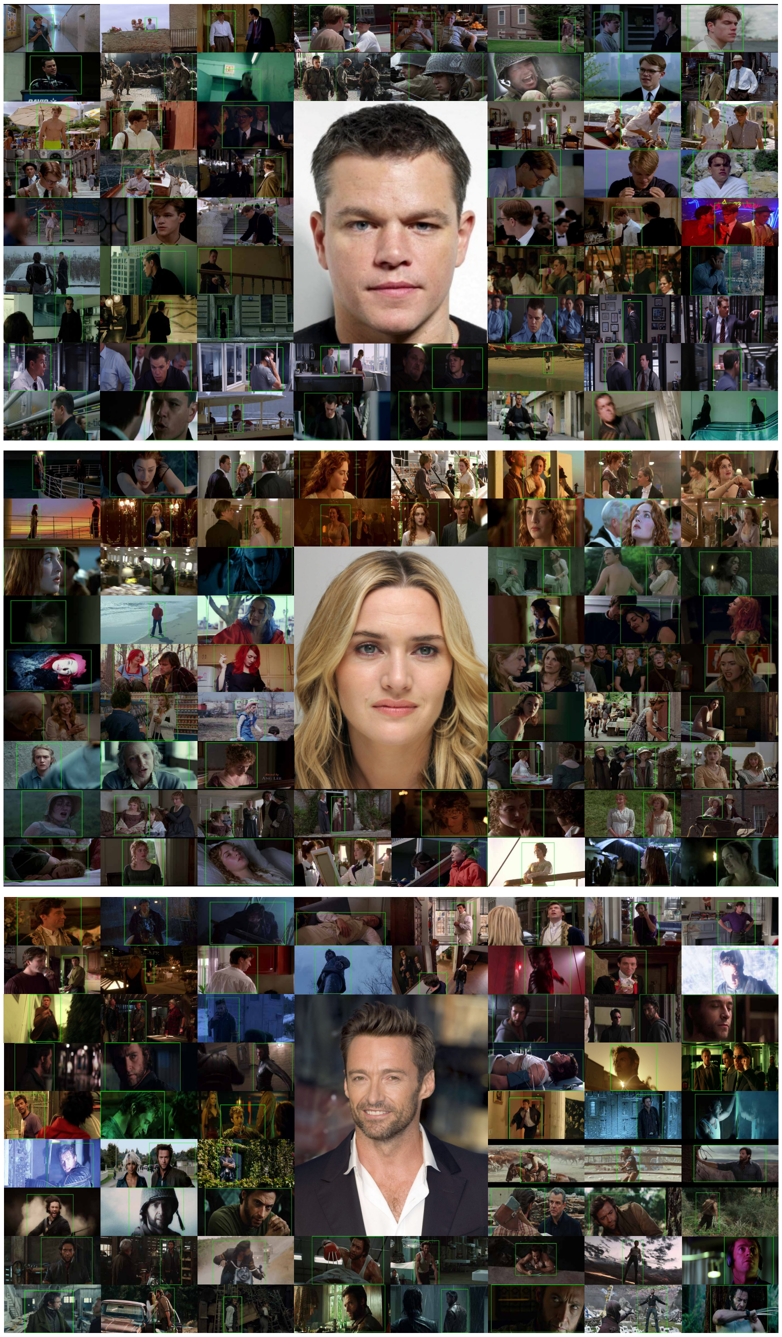


Fig. 14: Samples of the character annotations in MovieNet with portrait in the center.

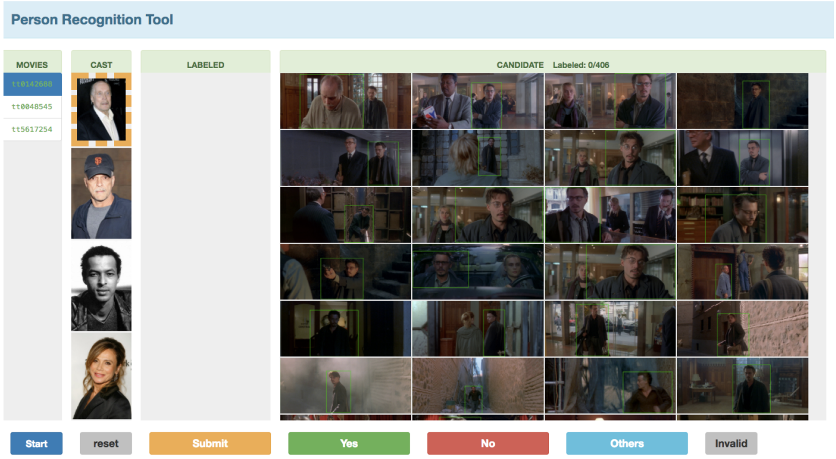


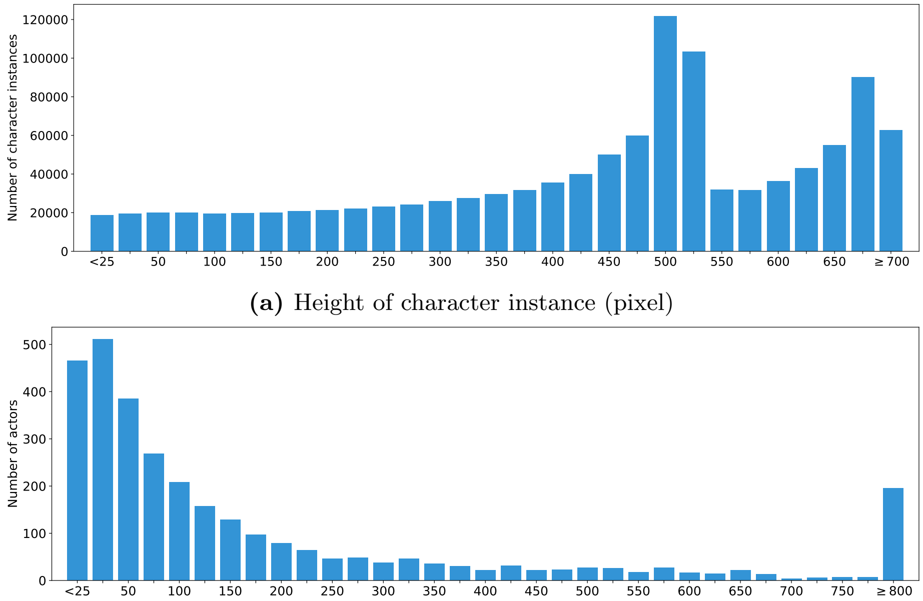
Fig. 15: Annotation interface of Character Identity (Stage 1). From left to right, they are (1) the movie list, (2) the cast list of the selected movie shown by their portraits, (3) the labeled samples annotated as the selected cast, which would be helpful for annotating more hard samples, and (4) the candidates of the selected cast, which is generated by our algorithm considering both face feature and body feature. The annotator can label positive samples (by clicking "Yes") or negative samples (by clicking "No"). After several iterations, when they are familiar to all the cast in the movie, they can label the characters belong the credit cast list by clicking "Others".

the full subject; (5) *long shot*: it is taken from a long distance and the subject is very small within the frames.

Shot Moviement. Shot movement has 4 categories (as shown in Fig. 18): (1) *static shot*: the camera is fixed but the subject is flexible to move; (2) *pans and tilts shot*: the camera moves or rotates; (3) *zoom out shot*: the camera zooms out for pull shot; (4) *zoom in shot*: the camera zooms in for push shot.

Annotation Categories. Compared with the definition in [15], we simplify the cinematic styles to make the annotation affordable. But we make sure the simplified categories are enough for most applications. For example, other cinematic styles like lighting are also important aspects. But the standard of lighting is hard to develop and we are now working on that with movie experts.

Annotation Workflow. To ensure the quality of the annotation, we make efforts as follows, (1) Instead of asking annotators to cut shot from movie or trailers and annotate their labels simultaneously, we cut shots from movies and trailers with the off-the-shelf method [35]. It is going to mitigate annotators' burdens considering the shot detection is well solved. (2) All the annotators went through a training phase first from cinematic professionals. And they are not allowed to annotate until they pass the professional test. They use our web-based annotation tool, as shown in Fig. 19. Each task is annotated three times to ensure a high annotation consistency.



(a) Height of character instance (pixel)

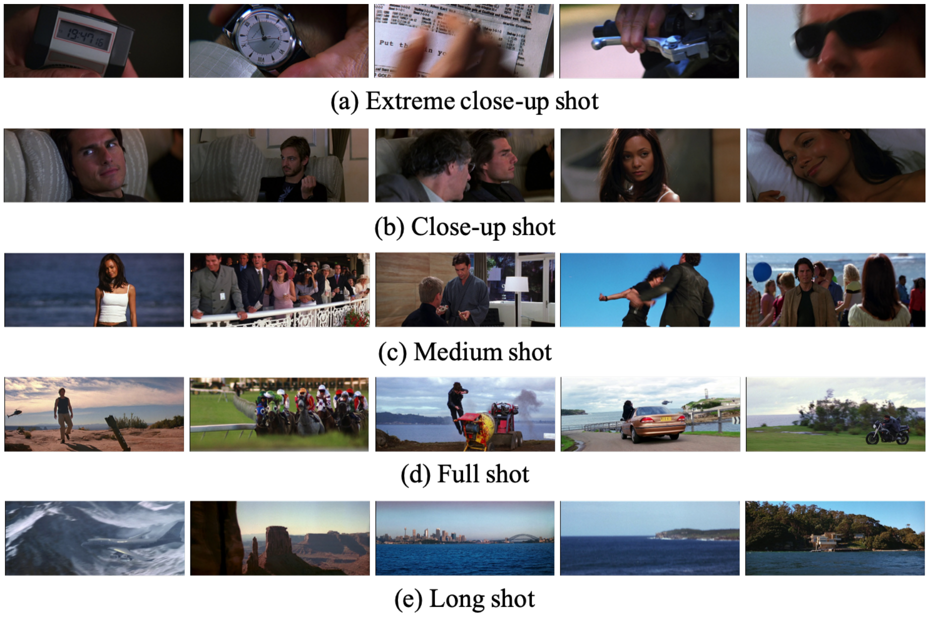
(b) Number of character instance

Fig. 16: Statistics of character bounding box and identity annotation, including the height of character instance and the number of character instance.

Sample. We show five samples of each category of shot scale in one movie *Mission Impossible*. As we can see from Fig. 17. Extreme close-up shot reveals the detailed information about characters or objects. Close-up shot reveals information about characters' faces and their emotions. Medium shot shows character involved activities. Long shot shows the scenes setting of a character or an object. Full shot shows the landscapes that set up the movie. Additionally we show four examples of different movement type shots, as shown in Fig. 18.

2.3 Scene Boundary

Annotation Workflow. To increase the efficiency of annotating procedure, as well as to improve the label quality, we propose the following annotating strategy. It would be a prohibitive amount of work if the annotators are asked to go through the movies frame by frame. In this work, we adopt a shot-based approach, based on the observation that a shot is a part of exactly one scene. Hence, we consider a scene as a continuous subsequence of shots, and consequently the scene boundaries can be selected from shot boundaries. Also note that shot detection, as a task, is already well solved. Specifically, for each movie, we first divide it into shots using off-the-shelf methods [35]. This shot-based approach greatly simplifies and speeds up the annotation process.



(a) Extreme close-up shot

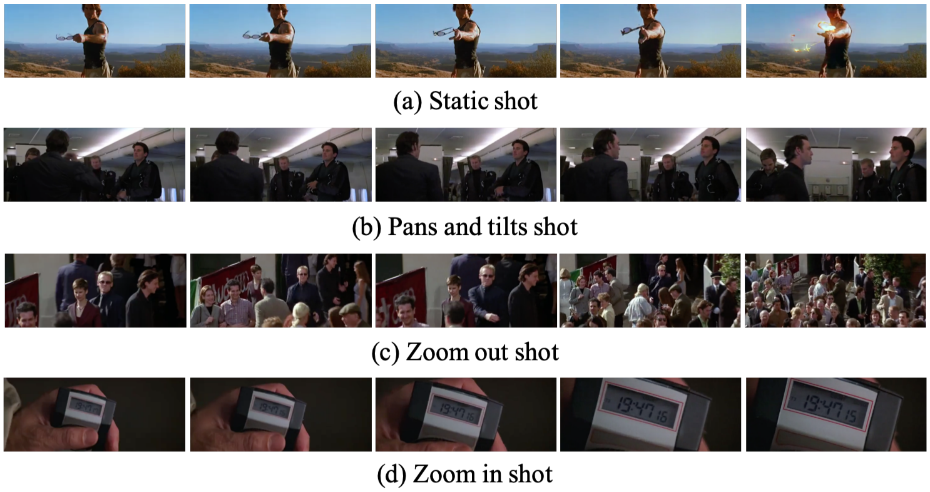
(b) Close-up shot

(c) Medium shot

(d) Full shot

(e) Long shot

Fig. 17: Examples of the cinematic style (scale type) of shots in *Mission Impossible*. From (a) to (e), they are extreme close-up shot, close-up shot, medium shot, full shot and long shot.



(a) Static shot

(b) Pans and tilts shot

(c) Zoom out shot

(d) Zoom in shot

Fig. 18: Examples of the cinematic style (movement type) of shots in *Mission Impossible*. From (a) to (d), they are static shot, pans and tilts shot, zoom out shot, and zoom in shot.

Annotation Interface. We also developed a web-based annotation tool, as shown in Fig. 20 to facilitate human annotators to determine whether a scene

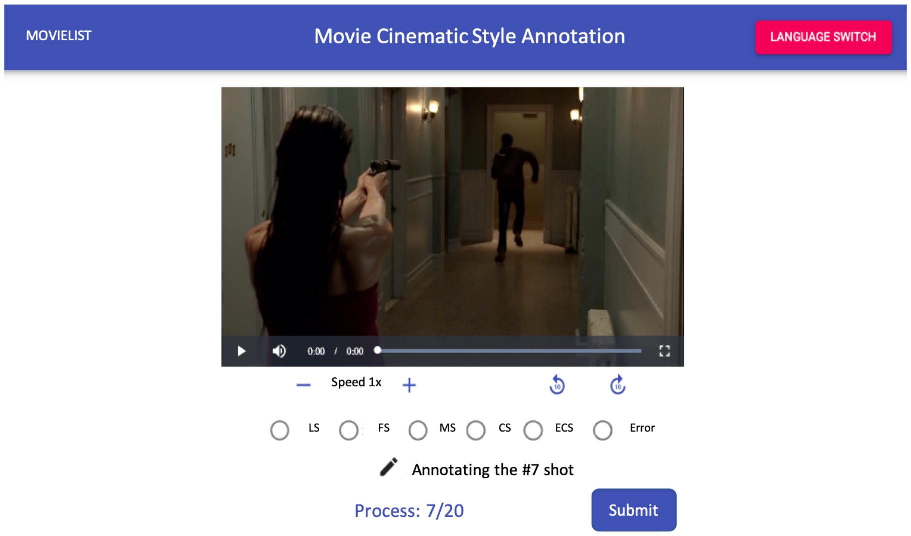


Fig. 19: Annotation interface of cinematic styles.

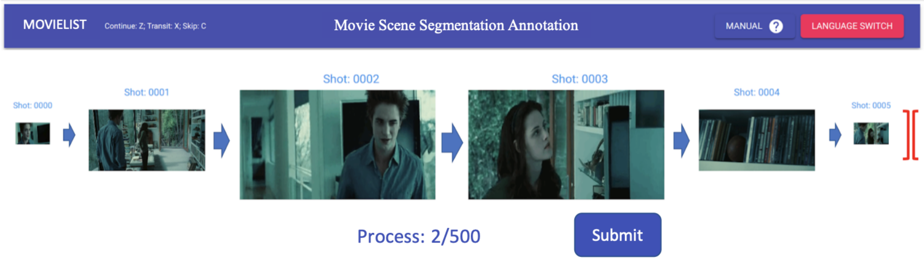


Fig. 20: Annotation interface of scene boundary.

transit or not between each pair of shots. On the web UI, annotators can watch two shots placed in the center, together with the frames preceding and succeeding these shots, respectively on the left and right sides. Annotators are required to make a decision as to whether the two shots belong to different scenes after watching both shots. The preceding and succeeding frames also provide a useful context.

Annotation Quality. A movie contains about $1K$ to $2K$ shots. It takes about one hour for an annotator to work through a whole movie, which can cause the difficulty to focus. To mitigate such issues, we cut each movie into 3 to 5 chunks (with overlaps), each containing about 500 to 700 shots. Then it only takes about 15 minutes to annotate one chunk. We found that it is much easier for annotators to focus when they work on chunks instead of the entire movie. All the annotators went through a training phase to learn how to use the annotation tool and how to handle various ambiguous cases before they work on the annotation

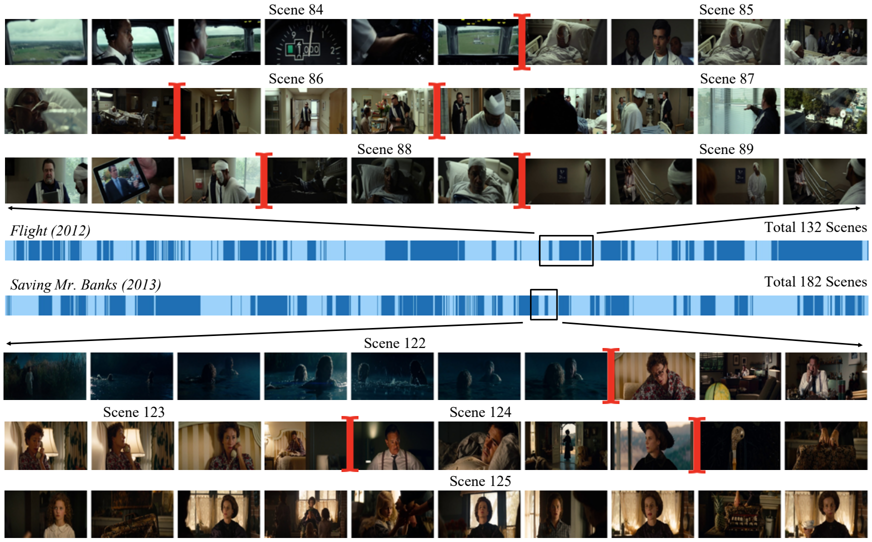


Fig. 21: Examples of the annotated scenes from two movies. The two lines in the middle correspond to the whole movie time line where the dark blue and light blue regions represent different annotated scenes, while the representative frames sampled from some scenes are also shown.

tasks. We asked annotators to make careful decisions between each pair of shots. If an annotator is not sure about certain cases, they can choose unsure as an answer and skip. The collection constitutes two rounds. In the first round, we dispatch each chunk of movies to three independent annotators so that we can check consistency. In the second round, inconsistent annotations, *i.e.* the results collected from three annotators do not agree, will be re-assigned to two additional annotators. For such cases, we will get five results for each chunk.

Samples. We show two samples of scene boundaries in Fig. 21. Segmenting scenes is challenging since visual cues are not enough to recognize the scene boundaries. For example, in the first movie *Flight* scene 84 to scene 89 and the second movie *Saving Mr. Banks* scene 123 to scene 124, most of the frames look very similar to each other. Additional semantic information such as character, action, audio are needed to make the right prediction. The difficulty of segmenting vary among different scenes. Some easy cases are also listed, *e.g.* the scene boundary between scene 122 and scene 123 is easy to recognize since visual changes are obvious.

2.4 Action and Place Tags

In this section, we introduce the annotating procedure of action and place tags in MovieNet. We develop an interface to jointly label action and place tags. The detailed workflow and introduction of interface will be expanded as follows.

Table 3: Detailed statistics of MovieNet action/place tags.

	Train	Val	Test	Total
# Action Clip	23747	7543	9969	41259
# Action Tag	26040	8071	10922	45033
# Place Clip	8101	2624	2975	13700
# Place Tag	11410	3845	4387	19642

Annotation Workflow. We split each movie into segments according to scene boundaries. Each scene video lasts around 2 min. We manually annotated tags of place and action for over each segment. For place annotation, each segment is annotated with multiple place tags, *e.g.*, *deck*, *cabin*, that cover all the places appear in this video. While for action annotation, we ask the annotators to first detect sub-clips that contain people and actions. Here they are asked to annotate the boundary that cover an uninterrupted human actions. Then they are asked to assign multiple action tags to each sub-clip to describe the actions within it. We have made the following efforts to keep tags diverse and informative: (1) We encourage the annotators to create new tags and (2) Tags that convey little information for story understanding, *e.g.*, stand, sit, walk, watch, listen and talk, are excluded. Note that in AVA [16], there are a large amount of this kind of actions, but here we choose to ignore these tags. This makes our dataset focus on the actions that more related to story telling. Finally, we merge the tags and filtered out 80 action classes and 90 place classes with a minimum frequency of 25 as the final annotations. In total, there are 13.7K segments with 19.6K scene tags and 41.3K action clips with 45K action tags. The detailed statistics are shown in Tab. 3. The distribution of action and place tags are shown in Fig. 22 and Fig. 23 respectively.

Annotation Quality. As mentioned above, the annotators can not only choose pre-defined tags but also create tags as they will. Before the annotation procedure

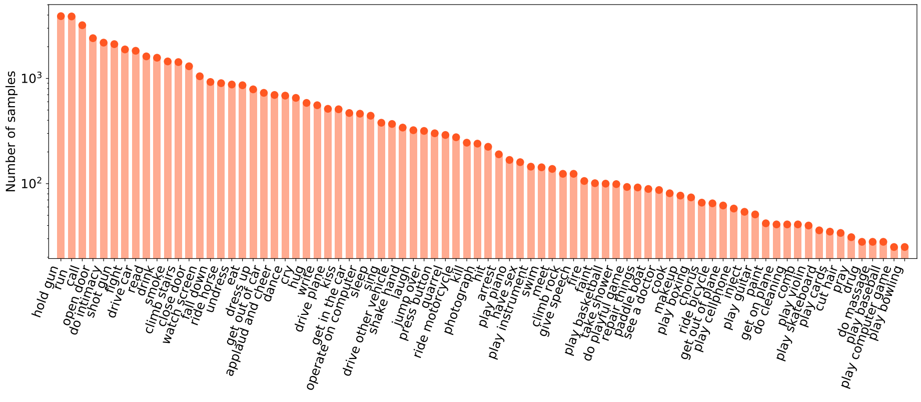


Fig. 22: Distribution of action annotations in MovieNet (y-axis in log scale).

starts, we create a pre-defined list of action and place tags by the following steps: (1) We collect the tags from previous works. The action tags are collected from datasets like AVA [16], Hollywood2 [26], *etc.*and the place tags are from datasets like Places [45], HVU [9], *etc.*. (2) We leverage GoogleNLP tools to detect verbs that stand for actions and nouns that stand for location. Then we manually choose some useful action and place tags into the list. (3) We randomly annotate a few hours of videos to collect tags before we ask the annotators to do so.

Besides, in case that there are uncovered tags, we highly recommend the annotators to create tags by themselves. During annotation stage, we gradually collect and merge the created tags into our pre-defined list for improving the annotation efficiency.

Annotation Interface. We provide annotators with easy-to-use annotation interface shown in Fig. 24. At the beginning of annotating tags, annotators are able to get familiar with all the pre-defined tags by clicking the button "ACT. LABEL" and "PLACE LABEL" in the menu bar. Then they can carry out action and place annotation at "Action Annotation Zone" and "Place Annotation Zone" respectively. For the convenience of annotators and also for the consideration of annotation quality, we provide a function of replaying history action annotation that enable the annotators to replay what they just labeled and refine the original annotations. We also provide snapshot of shot keyframes to help the annotators quickly grasp the rough content of the current video. To help annotate temporal action boundaries, we provide workspace for annotators to set timestamps by moving forward or backward at a minimum stride of 0.1 seconds. By the observation that shot boundaries are often action boundaries, we provide shortcuts to set action boundary as shot boundary. The above strategies are beneficial for improving annotation quality and efficiency.

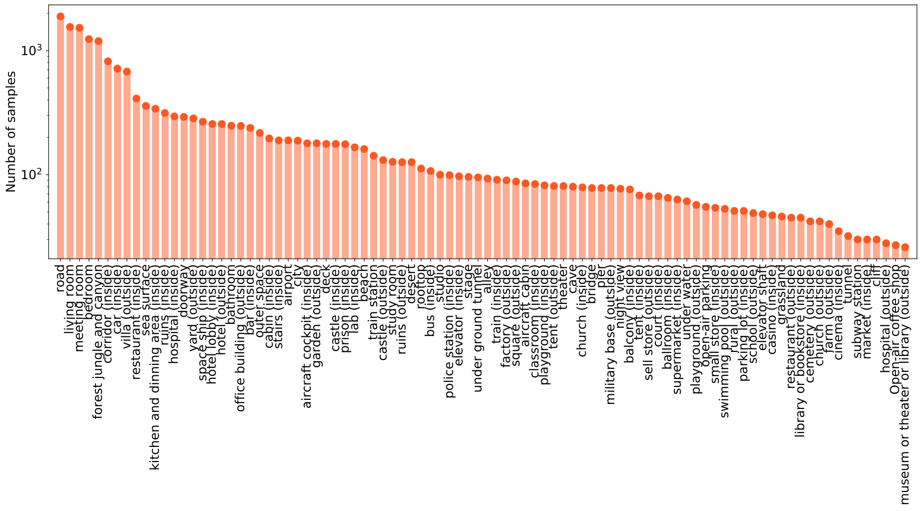
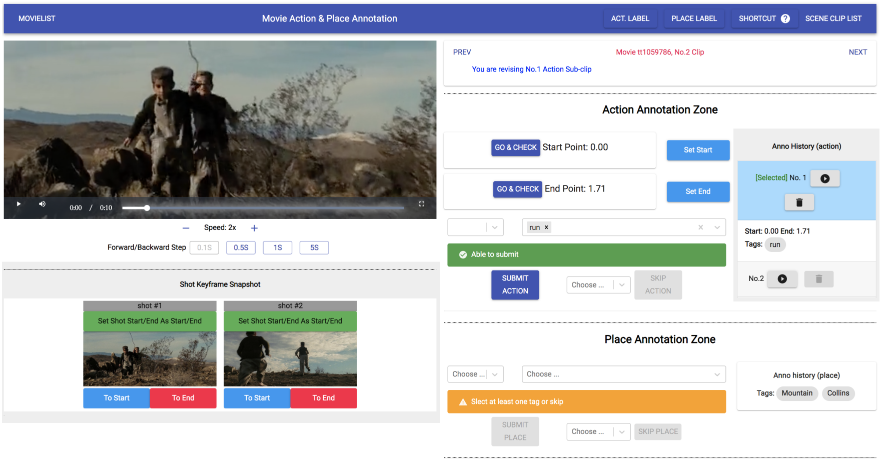


Fig. 23: Distribution of place annotations in MovieNet (y-axis in log scale).



Movie tt1059786, No.2 Clip

You are revising No.1 Action Sub-clip

Start Point: 0.00

End Point: 1.71

Able to submit

Start: 0.00 End: 1.71

Tags: run

Slect at least one tag or skip

Tags: Mountain Collins

Fig. 24: Annotation interface of the action and place tagging in MovieNet.

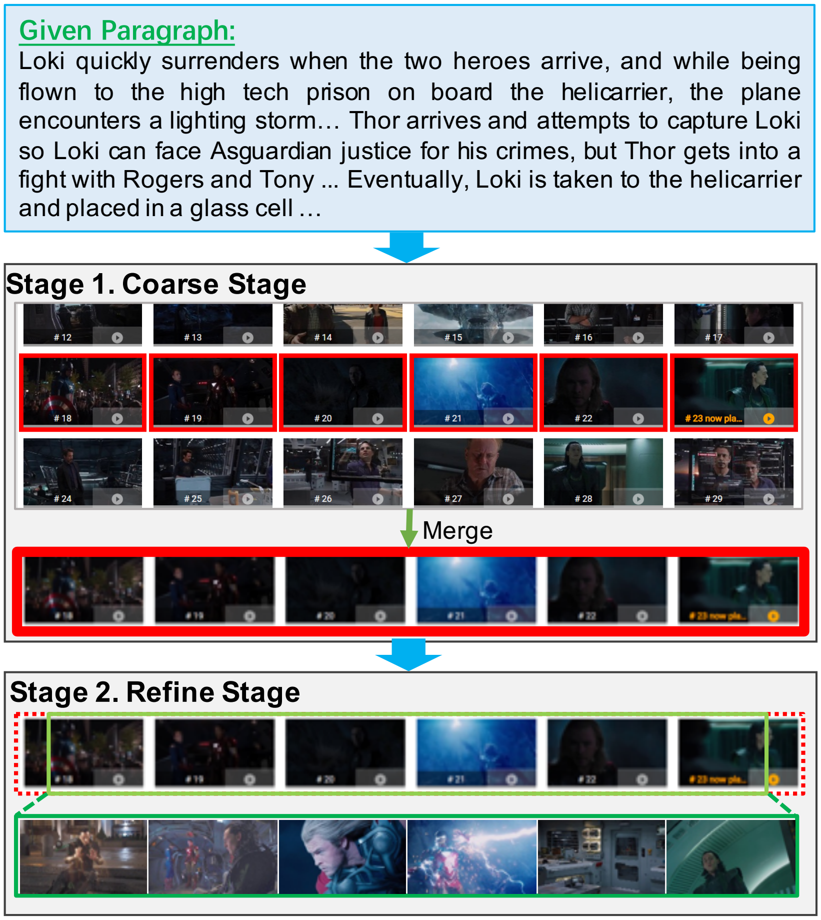
Table 4: Comparison of action and place tags with related datasets. For AVA, the action tags are annotated every second, hence the number of tags are larger than expected (see *). So for AVA, we show the statistics after merging the person instance into tracklet for fair comparison, resulting in 116K character tracklet (each tracklet is taken as a clip) and 360K action tags.

Dataset	dura.(h)	action clip	action tag	place clip	place tag	source
Hollywood2 [26]	21.7	1.7K	1.7K	1.2K	1.2K	movie
MovieGraphs [39]	93.9	7.6K	23.4K	7.6K	7.6K	movie
AVA [16]	107.5	116K	360K(1.58M*)	-	-	movie
SOA [29]	-	308K	484K	173K	223K	web video
HVU [9]	-	481K	1.6M	367K	1.5M	web video
MovieNet	214.2	41.3K	45.0K	13.7K	19.6K	movie

Dataset Comparison. Here we compare our annotated tags with other datasets with action and place tagging. The comparison is shown in Tab. 4. Note that for fair comparison, we merge the tags of AVA because they are annotated every second. The de-duplication is done by merging the tags within the same character tracklet. After de-duplication, the number of tags is $360K$ in AVA. But most of them are common actions like stand. There are $116K$ tracklet with $426K$ bboxes in AVA. In a word, AVA is comparable to MovieNet in spatial temporal action recognition, but MovieNet can support much more research topics.

2.5 Synopsis Alignment

To support the movie segment retrieval task, we manually associate movie segments and synopsis paragraphs. In this section, we will present the following

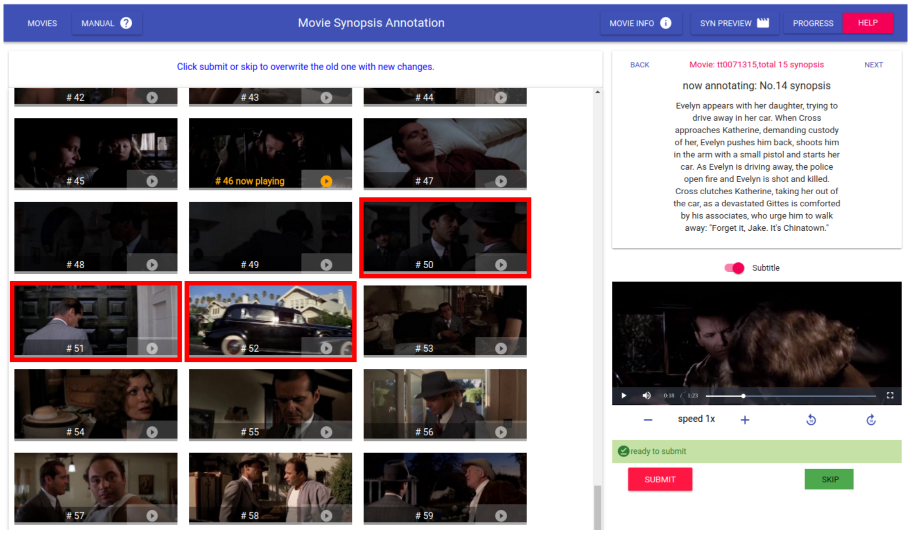

Given Paragraph:

Loki quickly surrenders when the two heroes arrive, and while being flown to the high tech prison on board the helicarrier, the plane encounters a lighting storm… Thor arrives and attempts to capture Loki so Loki can face Asguardian justice for his crimes, but Thor gets into a fight with Rogers and Tony ... Eventually, Loki is taken to the helicarrier and placed in a glass cell …

Fig. 25: Example of the annotating procedure for the movie *The Avengers*. At the coarse stage, annotator chooses consecutive clips. At the refine stage, boundaries are refined.

details about the movie-synopsis alignment, including annotation interface and workflow.

Annotation Workflow. After collecting synopses from IMDb, we further filtered out synopses with high quality, *i.e.* those contain more than 50 sentences, for annotating. Then we develop a coarse-to-fine procedure to effectively align each paragraph to its corresponding segment. (1) At the coarse stage, each movie is split into N segments, each lasting few minutes. Here we set $N = 64$. For each synopsis paragraph, we ask the annotators to select K consecutive clips that cover the content of the whole synopsis paragraph. (2) At the refine stage, the annotators are asked to refine the boundaries that make the resulting segment better aligned with the synopsis paragraph.



Click submit or skip to overwrite the old one with new changes.

Movie: tt0071315,total 15 synopsis

now annotating: No.14 synopsis

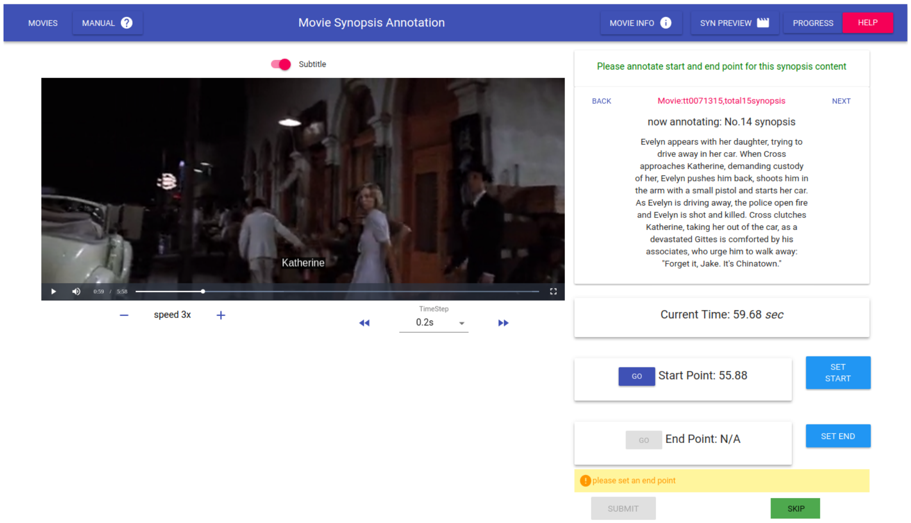
Evelyn appears with her daughter, trying to drive away in her car. When Cross approaches Katherine, demanding custody of her, Evelyn pushes him back, shoots him in the arm with a small pistol and starts her car. As Evelyn is driving away, the police open fire and Evelyn is shot and killed. Cross clutches Katherine, taking her out of the car, as a devastated Gittes is comforted by his associates, who urge him to walk away: "Forget it, Jake. It's Chinatown."

ready to submit

Fig. 26: User interface for the coarse stage. The left part display 64 movie clips; the upper right panel shows the synopsis paragraph; the bottom right player plays the selected movie clip.

To ensure the quality of the annotation, we make the following efforts. (1) The synopsis paragraphs from the same movie will be assigned to the same annotator. To ensure they are familiar with the movie, we provide the annotators with detailed overview of the movie, including character list, plot, reviews, *etc.*. (2) Each synopsis paragraph is dispatched to 3 annotators. Then, we only keep those annotations with high consistency, *i.e.*, those with high temporal IoU among all the 3 annotations. Finally, 4208 paragraph-segment pairs are obtained.

Annotation Interface. The movie-synopsis alignment is collected in a coarse-to-fine manner shown in Fig. 25. We develop an on-line interface to carry out these two stages. The interface of the coarse stage is shown in Fig. 26. At the beginning of annotating each movie, annotators are required to browse the overview of this movie, which is available at "MOVIE INFO" and "SYN PREVIEW" in the menu bar. Then annotators select a subsequence of N consecutive clips that cover the corresponding synopsis description shown in the text panel. After receiving the annotations, the back-end server will merge the consecutive clips into a whole segment for refine stage. As shown in Fig. 27, at the refine stage, annotators adjust the temporal boundaries of the resultant segments. We allow annotators to set timestamps at the current playback location as start or end timestamps. To enable fine adjustment, users are able to control the video player by moving forward or backward at a minimum stride of 0.1 seconds.



Please annotate start and end point for this synopsis content

Movie:tt0071315,total15synopsis

now annotating: No.14 synopsis

Evelyn appears with her daughter, trying to drive away in her car. When Cross approaches Katherine, demanding custody of her, Evelyn pushes him back, shoots him in the arm with a small pistol and starts her car. As Evelyn is driving away, the police open fire and Evelyn is shot and killed. Cross clutches Katherine, taking her out of the car, as a devastated Gittes is comforted by his associates, who urge him to walk away: "Forget it, Jake. It's Chinatown."

TimeStep: 0.2s

Current Time: 59.68 *sec*

Start Point: 55.88

End Point: N/A

please set an end point

Fig. 27: User interface for the refine stage. The left part displays the merged movie segment; the upper right panel shows the synopsis paragraph; the bottom right buttons are for adjusting the boundaries.

2.6 Trailer Alignment.

To facilitate tasks like trailer generation, we provide an automatic process for matching a shot in a trailer to its movie counterpart. The process is introduced below.

Consider each shot in a trailer as a sequence of images $s^T = \{s_1^T, \dots, s_M^T\}$ where M is the number of frames in this shot. For a movie, we define it as a sequence of frame $s^M = \{s_1^M, \dots, s_N^T\}$ where N is the number of frame in the movie. To locate the position of the trailer shot in the movie is to find the most similar sub-sequence of the movie with the shot sequence. Let $Sim(s_i^T, s_j^M)$ denotes the similarity of i-th frame in trailer shot and j-th frame in the movie. The solution is to find the sub-sequence in the movie that maximize the similarities:

$$\begin{aligned} j^* = \operatorname*{argmax}_j \quad & \sum_{i=1}^{N} Sim(s_i^T, s_{j+i-1}^M) \\ \text{s.t.} \quad & j \le N - M + 1. \end{aligned} \tag{2}$$

To obtain the similarity between two frames, we resort to features from low-level to high-level. For each frame, we extract GIST feature [11] and feature from *pool*5 layer of ResNet-50 [18] that pre-trained on ImageNet [34]. We choose to use low-lwvel feature like GIST feature because we observe that most of the frames from trailers are alike with the original ones in movies, only with slightly changes in terms of color, size, lighting, boundary, *etc.*. For some harder cases like cropped or carefully edited shots, we find high-level features like ResNet feature works. Hence, the similarity can be obtained by

$$Sim(s_a, s_b) = cosine(f_a^{gist}, f_b^{gist}) + cosine(f_a^{imagenet}, f_b^{imagenet}).$$

where f^{gist} stands for GIST feature while $f^{imagenet}$ stands for feature from ResNet.

Solving the optimization problem would result in the required alignment results. For those aligned shots with low optimized similarity score, we set it as misaligned shot. By observation, those misaligned shots are mostly transition shots or shots that only exist in the trailer. We then manually checked the alignment result to make sure the alignment results are accurate.

3 Experiments

In this section, we introduce the detailed information for each benchmark. Note that unless specified, the 1100 movies are split according to a ratio of $3 : 1 : 1$. The annotations are split according to the result of the split of movies, hence there are no overlapping movies among train, val and test sets for each task.

3.1 Genre Classification

Here we would introduce the benchmark setting, evaluation metrics, implementation details of the baseline models and show more experiment results.

Benchmark Setting. There are total 28 unique genres in MovieNet. Some rare genres, e.g. Adult, are ignored. And we further remove non-visual genres, e.g. News, to obtain a list with 21 genres. For image-based genre classification, we use the $3.9M$ photos of MovieNet as our data source. Since not all types of photos are related to the genres, e.g. publicity, here we only take 4 of them to build the benchmark for genre classification, namely poster, still frame, product and production art, resulting in $1.6M$ photos left. The $1.6M$ images are split into train, validation and test set that contains $1.1M$, $160K$ and $321K$ images respectively. For video-based genres classification, we take trailers for experiments. We sample $32K$ trailers containing at least one of the 21 genres. They are split as train, validation and test set contains $22.5K$, 3.2K and $6.4K$ videos respectively.

Evaluation Metric. Genre classification is a multi-label classification problem. We use mAP, recall@0.5 and precision@0.5 as evaluation metrics. Here 0.5 is the threshold to get the final prediction, which means that the final score (between 0 and 1) above 0.5 would be set as positive while the others would be set as negative.

Implementation Details. The models are trained with BCE Loss. The input size is set to 224×224 and we use SGD as optimizer. For the video-based model, we get 8 clips, each with 3 frames, on training. At the inference stage, we would predict the score of all the clips and average them to get the final prediction.

More Results. Here we show the per-genre results of the ResNet-50 model in Tab. 5. We can see that *animation* achieve the highest accuracy. This is

Table 5: Per-genre performance of genre classification.

	R@0.5	P@0.5	AP
Drama	79.42	71.16	79.95
Comedy	48.65	68.61	68.81
Thriller	14.50	64.98	49.80
Action	22.21	73.96	54.60
Romance	14.02	71.93	49.27
Horror	8.76	70.03	35.51
Crime	39.30	74.12	49.25
Documentary	4.79	85.49	21.03
Adventure	24.72	75.24	53.06
Sci-Fi	14.51	81.35	44.14
Family	27.11	82.55	52.19
Fantasy	13.51	69.83	39.12
Mystery	7.76	76.42	39.70
Biography	0.04	100.00	9.13
Animation	74.09	93.16	86.45
History	12.52	82.90	34.41
Music	27.24	89.04	47.13
War	12.80	86.27	34.41
Sport	21.99	94.97	39.59
Musical	4.45	73.58	22.88
Western	51.93	88.89	73.99

reasonable since the characteristic of animation is significant. And the AP of *Biography* and *Documentary* is much lower since these genres are determined by higher semantic elements.

3.2 Cinematic Style Analysis

Dataset Split. The MovieNet cinematic style prediction benchmark contains $46K$ shots coming from $8K$ trailers where $26K$ for training, $7K$ for validation and $13K$ for testing.

Details of baseline models for cinematic style prediction. We implement TSN [40] (3 segments) and I3D [3] with different backbones ResNet-18, ResNet-34, ResNet-50 [18]. The results are shown in the Tab. 6. We observe that 2D models achieve better results as the network becoming deeper both in terms of scale and movement classification. Deeper 3D models are a bit of over fitting on movement since the performance drops a little when the network becomes deeper.

Details of TSN+R^{3}Net. As we point out in the paper, the subject is very important for cinematic style analysis. Thus a subject-based method is proposed to solve the problem. First, we adopt R^{3}Net [8] to get the saliency map of each frame in the shot. Then, each video clip passes through a two-branch classification network, one branch is for video clips, the other branch is for video saliency clips. For movement classification, the whole image and the background image are used

Table 6: Baselines for MovieNet cinematic style prediction.

Method	Backbone	Scale-Accuracy	Move-Accuracy
TSN [40]	ResNet18	79.31	68.02
	ResNet34	82.73	69.91
	ResNet50	84.08	70.46
I3D [3]	ResNet18	76.79	78.45
	ResNet34	82.10	**82.17**
	ResNet50	82.70	81.97
TSN+R^{3}Net [8]	ResNet50	**87.58**	80.65

Table 7: Character identification: comparison of MovieNet and related datasets that used as the training datasets in this benchmark.

Dataset	ID	instance
Market [44]	1,501	32K
CUHK03 [21]	1,467	28K
CSM [19]	1,218	127K
MovieNet	**3,087**	**1.1M**

Table 8: Performance of different methods for character identification in MovieNet.

Train Data	cues	Method	mAP
Market [44]	body	r50-softmax	4.62
CUHK03 [21]	body	r50-softmax	5.33
CSM [19]	body	r50-softmax	26.21
MovieNet	body	r50-softmax	32.81
	body+face	LP [47]	8.29
	body+face	two-step [25]	63.95
	body+face	PPCC [19]	**75.95**

as the inputs. For scale classification, the whole image and the subject image are used as the inputs of the two branches. The features from the two branches are concatenated and pass through a fully-connected layer to get the final prediction.
Implementation Details. We take cross-entropy loss for the classification. We train these models for 60 epochs with mini-batch SGD, where the batch size is set to 128 and the momentum is set to 0.9. The network weights are initialized with pretrained models from ImageNet [34]. The initial learning rate is 0.001 and the learning rate will be divided by 10 at the 20th and 40th epoch.

3.3 Character Detection

Dataset Split. We collect 1.3M bounding boxes from 758K keyframe images. The 758K images are split into train and test set with 692K and 66K respectively.
Evaluation Metric Following one of the most popular benchmark for object detection – COCO [23], we use the AP from 0.5 to 0.95 with a stride 0.05, namely mAP, as our evaluation metric.
Implementation Details. We take a Faster R-CNN with ResNet-50 as backbone, and train on three different datasets, COCO [23], CalTech [10] and our MovieNet-PDet to show the large domain gap between movie and other data source. And then we also try more models on MovieNet, including a single-stage model, namely RetinaNet, and a more powerful model, *i.e.* Cascade R-CNN [2] with ResNeXt-101 [41] backbone and feature pyramid [22].

Table 9: Comparison of MovieNet scene segmentation with related datasets.

Dataset	# Scene	Duration(hour)	Source
OVSD [33]	300	10	MiniFilm
BBC [1]	670	9	Documentary
MovieNet	42K	633	Movie

3.4 Character Identification

Dataset Split. We annotate identities of more than $1.1M$ instances of $3K$ identities. In the MovieNet cahracter identification benchmark. They are split into train, val, test set with 2088 identities with $639.9K$ instances, 821 identities with $336.6K$ instances and 876 identities with $364.2K$ instances respectively.
Benchmark Setting. The Character identification task is to search for all the instances of a character in a movie with just one portrait. To enable the task, we download a portrait from homepage of each credited cast, which will serve as the query portraits for the character identification tasks.
Evaluation Metric. We use mAP that average the AP on each query as the evaluation protocol.
Baseline Results. The results of character identification are shown in Tab. 8. The character identification task is similar to conventional person ReID task, however our dataset is much more challenging and larger than theirs. So in order to show the domain gap, we train ResNet-50 with softmax loss on three person/character identification dataset, namely, Market [44], CUHK03 [21] and MovieNet shown in Tab. 7. From the results, we see that due to the large domain gap, current ReID datasets cannot support the researches on character analysis in Movies. We also adopt methods – LP [47], PPCC [19] for comparison.
Implementation Details. The character identification task need to utilize both face feature and body feature. Here the face features are extracted by a ResNet-101 trained on MS1M [17] and the body features are extracted by a ResNet-50 trained by MovieNet identity annotation or other ReID dataset. The cues in Tab. 8 means that we retrieve the character by only the features mentioned by cues. The *Two-Step* method means that we would first retrieve by face features, and then add some instances with high confidence to the query set, after which we would do set-to-set retrieval by body features. This is widely used in the WIDER Challenge [25]. LP [47] means the naive label propagation method, which would be affected by noise and get a poor performance. And PPCC [19] improves LP by developing a competitive consensus scheme, which is the current state-of-the-art for character identification.

3.5 Scene Segmentation

Dataset Split. The overall $42K$ scenes are split into train, val, test sets with $25K$, $8.9K$ and $8.1K$ scene segments respectively. There are no overlapped movies. The comparison of our dataset with other related scene segmentation datasets are shown in Tab. 9.

Table 10: Baseline results for scene segmentation.

Training Data	Method	AP (↑)	M_{iou} (↑)
OVSD [33]	Grouping [33]	0.170	0.301
	MS-LSTM	**0.313**	**0.387**
BBC [33]	Siamese [1]	0.268	0.358
	MS-LSTM	**0.334**	**0.379**
MovieNet-SSeg	Grouping [33]	0.336	0.372
	Siamese [1]	0.358	0.396
	MS-LSTM (Audio only)	0.210	0.341
	MS-LSTM (Character only)	0.213	0.348
	MS-LSTM (Action only)	0.227	0.368
	MS-LSTM (Place only)	0.442	0.421
	MS-LSTM	**0.465**	**0.462**

Evaluation Metrics. We take two commonly used metrics: (1) Average Precision (AP). Specifically in our experiment, it is the mean of AP of detected scene boundary for each movie. (2) M_{iou}: a weighted sum of intersection of union of a detected scene boundary with respect to its distance to the closest ground-truth scene boundary.

Baseline Models. We reproduce Grouping [33] and Siamese [1] according to their papers. For our baseline multi-semantic LSTM (MS-LSTM), we extract audio, character, action and scene feature from each shot as follows,

- **Audio feature.** We apply NaverNet [7] pretrained on AVA-ActiveSpeaker dataset [32] to separate speech and background sound, and stft [38] to get their features respectively in a shot with 16K Hz sampling rate and 512 windowed signal length, and concatenate them to obtain audio features.
- **Character feature.** We firstly take the advantage of Faster-RCNN [30] pretrained on MovieNet character detection benchmark to detect character instances. And then we use a ResNet50 trained on MovieNet character identification benchmark to extract character features.
- **Action feature.** We utilize TSN [40] with AVA dataset [16] pretraining to get *action* features.
- **Place feature.** We take ResNet50 [18] with Places dataset [46] pretraining on key frame images of each shot to get place features.

More Results. From Tab. 10, we observe that (1) Benefited from large scale and high diversity, models trained on MovieNetSSeg achieve more than 40% improvement in performance. Specifically, Grouping [33] improves 98% from 0.170 to 0.336, Siamese [1] improves 34% from 0.268 to 0.358, MS-LSTM improves 39% from 0.334 to 0.465. (2) Multiple semantic elements are important for scene segmentation, which highly raise the performance. Jointly using audio, character, action and place information surpass any single element.

Table 11: Results of action classification.

Method	mAP
TSN [40]	14.17
I3D [3]	20.69
SlowFast [12]	**23.52**

Table 12: Performance of different methods for place classification.

Method	mAP
I3D [3]	7.66
TSN [40]	**8.33**

Implementation Details. We take cross entropy loss for the binary classification. We train these models for 30 epochs with SGD optimizer. The initial learning rate is 0.01 and the learning rate will be divided by 10 at the 15th epoch.

3.6 Action Recognition

We conduct experiments of action recognition on MovieNet action recognition benchmark. This task aims at predicting multiple action tags of a given video.

Dataset Split. The whole dataset is randomly split as train, val, test set with 23747, 7543, 9969 video clips respectively, without overlapping movies.

Loss and Metrics. For training all models, we use binary cross-entropy loss as the loss function. For multi-label classification evaluation, we use mAP as the evaluation protocol.

Implementation of TSN. We adopt TSN [40] as one of our baseline models. To be specific, the TSN2D model adopt ResNet50 [18] as backbone. We sample 3 segments for each video and the consensus function is simply *Average*. We set batch size as 32 and dropout rate as 0.5. The model is trained using SGD for 100 epochs with an initial learning rate 0.01, momentum 0.9 and weight decay 0.0005. The learning rate is divided by 10 at the 60 and 90 epoch.

Implementation of I3D. For I3D [3], we adopt ResNet-I3D [3] with depth 50 as the baseline model. The inflate style is set to $3 \times 1 \times 1$ and the input length is 32 with stride 2. We set batch size as 8 and dropout rate as 0.5. The model is trained using SGD for 100 epochs with an initial learning rate 0.01, momentum 0.9 and weight decay 0.0001. The learning rate is divided by 10 at the 60 and 90 epoch.

Implementation of SlowFast. For SlowFast Network [12], the backbone is I3D with ResNet50. We set τ to 8, α to 8 and β to 1/8. The input length is 32 with stride 2. We set batch size as 8 and dropout rate as 0.5. The model is also trained 100 epochs using a half-period cosine schedule [24] of learning rate decaying with n-th iteration learning rate as $0.5\eta[cos(\frac{n}{n_{max}}\pi) + 1]$. n_{max} is the max iteration number, η is the basic learning rate set as 0.2.

Analysis. The experimental results are shown in Tab. 11. We see that the performance of TSN is the lowest while SlowFast is the best and outperform other baselines by a large margin. To further analysis the performance, we show per-class AP score of SlowFast Network in Fig. 28.

3.7 Place Recognition

We conduct experiments of place recognition on MovieNet place recognition benchmark. The task aims at predicting multiple place tags of a given video.

Dataset Split. The whole dataset is randomly split as train, val, test set with 8101, 2624, 2975 clips respectively, without overlapping movies.

Loss and Metrics. For training all models, we use binary cross-entropy loss as the loss function. For multi-label classification evaluation, we use mAP as the evaluation protocol.

Implementation of TSN. We adopt the same TSN structure as in MovieNet-Action, except that we use 12 segments here instead of 3. The training scheme is also the same as TSN in MovieNet-Action except that the batch size is changed to 8.

Implementation of I3D. We use the same I3D model and the same training scheme of I3D in MovieNet-Action.

Analysis. The experimental results are shown in Tab. 12. The backbone weights are adopted from ImageNet pretrained model. We see that the performance of TSN outperforms I3D probably because 3d convolution is harder to learn and the I3D model suffers over-fitting. We do not use SlowFast Network as one of the baseline models because SlowFast can not leverage the pretrain weight from ImageNet and our dataset is not large enough to support training SlowFast from scratch. To further analysis the performance, we show per-class ap score of TSN Network in Fig. 29.

3.8 Story Understanding

We conduct experiments of movie-synopsis retrieval on MovieNet synopsis alignment dataset. To be specific, the task is to search a relevant movie segment given

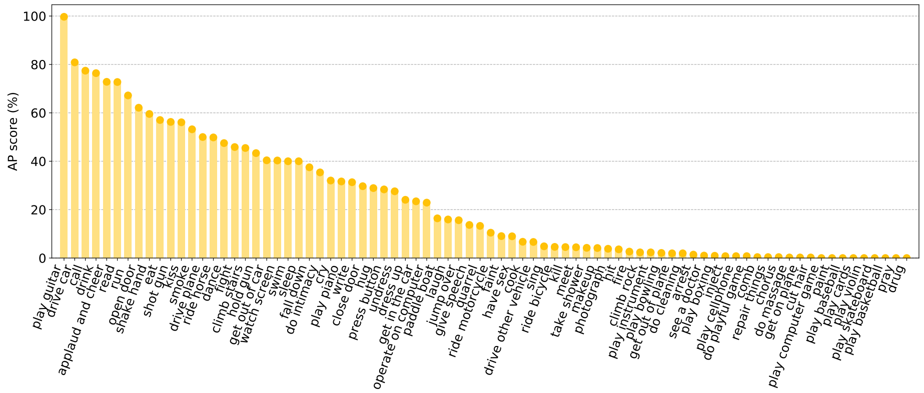


Fig. 28: Per-class AP score of SlowFast Network for action recognition task on MovieNet (sorted according to the performance of each class).

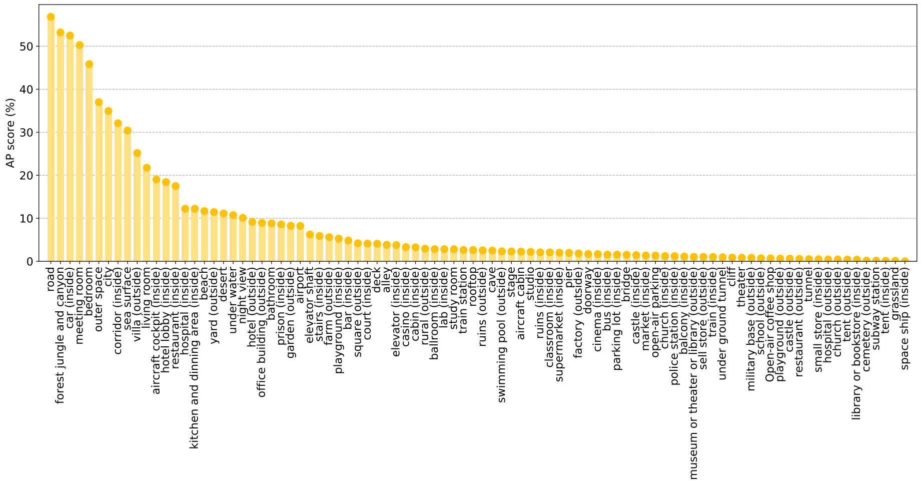


Fig. 29: Per-class AP score of TSN for place recognition task on MovieNet (sorted according to the performance of each class).

Table 13: Results of movie segment retrieval. Here, G stands for global appearance feature, S for subtitle feature, A for action, P for character and C for cinematic style.

Method	Recall@1	Recall@5	Recall@10	MedR
Random	0.11	0.54	1.09	460
G	3.16	11.43	18.72	66
G+S	3.37	13.17	22.74	56
G+P	12.76	42.98	53.97	8
G+S+A	5.22	13.28	20.35	52
S+A+P	12.62	27.21	49.40	11
G+S+A+P	18.50	43.96	55.50	7
G+S+A+P+C	18.72	44.94	56.37	7
MovieSynAssociation [42]	**21.98**	**51.03**	**63.00**	**5**

a synopsis paragraph as query. The extended results are show in Tab. 13. and the details would be introduced below.

Dataset Split. After dataset split by movies, we obtain train, val, test set with 2422, 867, 919 samples respectively.

Evaluation Metrics. For retrieval task, we adopt two metric to measure the performance, namely, Recall@K and MedR. (1) Recall@K: the fraction of ground truth movie segments that have been ranked in top K; (2) MedR: the median rank of ground truth movie segments.

Baseline Models. We adopt VSE [14] as the base model in our experiments. In VSE model, we gradually add four kinds of nodes into baseline model, namely, *appearance*, *subtitle*, *action* and *character*. For each node, we extract a sequence of visual features from movie segment and a sequence of text features from synopsis paragraph. The detail of feature extraction will be introduced in the next section. In VSE, the input video features and paragraph features are first transformed with two-layer MLPs. For appearance, subtitle and action nodes, we first obtain the embedding of segment and paragraph by taking the average of the output sequence features. Then, the similarity score between segment and paragraph is computed by applying cosine similarity between two embeddings. For cast feature, we obtain the similarity score by applying *KuhnMunkres (KM)* algorithm [20] between the output sequence features from segment and paragraph. For training, we use the pairwise ranking loss with margin α shown below,

$$\begin{aligned}\mathcal{L}(S;\boldsymbol{\theta}) = &\sum_{i}\sum_{j\neq i} max(0, S(Q_j, P_i) - S(Q_i, P_i) + \alpha) \\ &+ \sum_{i}\sum_{j\neq i} max(0, S(Q_i, P_j) - S(Q_i, P_i) + \alpha) \quad (3)\end{aligned}$$

where $S(Q_j, P_i)$ denotes the similarity score between j^{th} segment Q_j and i^{th} paragraph P_i. $\boldsymbol{\theta}$ denotes the model parameters.

Feature Extraction. The process of extracting input features for different modalities and different nodes are presented below.

- **Appearance feature from movie segment.** Appearance feature consists of a sequence of features extracted from each shot. For each shot, we extract the feature from *pool5* layer of ResNet-101 [18].
- **Appearance features from synopsis paragraph.** For paragraph, the appearance feature is represented as a sequence of Word2Vec [27] embeddings extracted from each sentence.
- **Subtitle feature from movie segment.** We also use Word2Vec embedding to extract features for subtitles in each shot. When adding to the model, we directly concatenate the subtitle feature of each shot to the appearance feature of each shot.
- **Action feature from movie segment.** The action features come from feature concatenated by TSN [40] pre-trained on AVA [16] and on MovieNet action recognition benchmark. We extract the action feature on each shot when there are actors appear in this shot.
- **Action feature from synopsis paragraph.** We detect verbs using part-of-speech tagging provided by GoogleNLP[5]. We select 1000 verbs with the highest frequency from the synopses corpus, and then retain those corresponding to visually observable actions, *e.g. run.* Action verbs are then represented by Word2Vec embeddings.
- **Character feature from movie segment.** We leverage the detector trained on MovieNet character detection benchmark to detect character instance in every shot. Then we use ResNet50 pretrained on PIPA [43] to extract the body feature and face feature as representation.

[5] https://cloud.google.com/natural-language/

– **Character feature from synopsis paragraph.** We detect all the named entities (*e.g. Jack*) using StanfordNer [13]. With the help of IMDb, we can retrieve a portrait for each named character and thus obtain facial and body features using ResNet50 pre-trained on PIPA. This allows character nodes to be matched to the character instances detected in the movie.

Add Cinematic Style. As mentioned in the paper, cinematic style can help distinguish whether a node is important in a particular shot. Hence we design a module that take cinematic style and the node itself as input and produce an attention on this node. For each element in a shot, we concatenate its feature and the probability of the cinematic style as input, then this feature is passed through a MLP to produce a single attention score. This score is later used as the weight of output embedding.

Using Graph Formulation. We implement the algorithms in [42], both Event Flow Module and Character Interaction Module. The results are the combination of the two modules. That being said, we leverage the two modules to model spatial and temporal graph relations respectively. The difference between our implementation with theirs are the feature we used are different (see the Feature Extraction section).

Implementation Details. We train all the embedding networks using SGD with learning rate 0.001. The batch size is set to 16 and the margin α in pair-wise ranking loss is set to 0.2.

4 Toolbox

In this section, we introduce the toolbox designed for MovieNet, it will be released with the dataset and corresponding benchmark codes. The toolbox are mainly comprised of the following parts:

– **Crawlers.** The crawler for downloading metadata, subtitle and other useful data will be provided.
– **Preprocessing.** The preprocessing tools would provide functions that efficiently process multi-media resources. For example, extract audio waves, cut movies, resize movies.
– **Data generators.** The tools for generating the data, for example, shot detection will be included. Besides, we will also provide hany tool for users to align their own movies with ours, if needed.
– **Data Parser.** The parsers are designed to easily access the required matadata or data. For example, to visualize the character bounding box or to read the genres of a movie.

References

1. Baraldi, L., Grana, C., Cucchiara, R.: A deep siamese network for scene detection in broadcast videos. In: 23rd ACM International Conference on Multimedia. pp. 1199–1202. ACM (2015) 30, 31

2. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018) 12, 29
3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017) 28, 29, 32
4. Cascante-Bonilla, P., Sitaraman, K., Luo, M., Ordonez, V.: Moviescope: Large-scale analysis of movies using multiple modalities. arXiv preprint arXiv:1908.03180 (2019) 3
5. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Change Loy, C., Lin, D.: Hybrid task cascade for instance segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 12
6. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019) 12
7. Chung, J.S.: Naver at activitynet challenge 2019–task b active speaker detection (ava). arXiv preprint arXiv:1906.10555 (2019) 7, 31
8. Deng, Z., Hu, X., Zhu, L., Xu, X., Qin, J., Han, G., Heng, P.A.: R3net: Recurrent residual refinement network for saliency detection. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. pp. 684–690. AAAI Press (2018) 28, 29
9. Diba, A., Fayyaz, M., Sharma, V., Paluri, M., Gall, J., Stiefelhagen, R., Van Gool, L.: Holistic large scale video understanding. arXiv preprint arXiv:1904.11451 (2019) 22, 23
10. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE transactions on pattern analysis and machine intelligence **34**(4), 743–761 (2011) 29
11. Douze, M., Jégou, H., Sandhawalia, H., Amsaleg, L., Schmid, C.: Evaluation of gist descriptors for web-scale image search. In: Proceedings of the ACM International Conference on Image and Video Retrieval. pp. 1–8 (2009) 26
12. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6202–6211 (2019) 32
13. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd annual meeting on association for computational linguistics. pp. 363–370. Association for Computational Linguistics (2005) 36
14. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: A deep visual-semantic embedding model. In: Advances in neural information processing systems. pp. 2121–2129 (2013) 34
15. Giannetti, L.D., Leach, J.: Understanding movies, vol. 1. Prentice Hall Upper Saddle River, New Jersey (1999) 14, 16
16. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., et al.: Ava: A video dataset of spatio-temporally localized atomic visual actions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6047–6056 (2018) 3, 21, 22, 23, 31, 35

17. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: European Conference on Computer Vision. pp. 87–102. Springer (2016) 12, 30
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 12, 26, 28, 31, 32, 35
19. Huang, Q., Liu, W., Lin, D.: Person search in videos with one portrait through visual and temporal links. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 425–441 (2018) 29, 30
20. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955) 35
21. Li, W., Zhao, R., Xiao, T., Wang, X.: Deepreid: Deep filter pairing neural network for person re-identification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 152–159 (2014) 29, 30
22. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017) 12, 29
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014) 29
24. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 32
25. Loy, C.C., Lin, D., Ouyang, W., Xiong, Y., Yang, S., Huang, Q., Zhou, D., Xia, W., Li, Q., Luo, P., et al.: Wider face and pedestrian challenge 2018: Methods and results. arXiv preprint arXiv:1902.06854 (2019) 29, 30
26. Marszałek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR 2009-IEEE Conference on Computer Vision & Pattern Recognition. pp. 2929–2936. IEEE Computer Society (2009) 22, 23
27. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013) 35
28. Ramos, J., et al.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning. vol. 242, pp. 133–142. Piscataway, NJ (2003) 10
29. Ray, J., Wang, H., Tran, D., Wang, Y., Feiszli, M., Torresani, L., Paluri, M.: Scenes-objects-actions: A multi-task, multi-label video dataset. In: The European Conference on Computer Vision (ECCV) (September 2018) 23
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 91–99. Curran Associates, Inc. (2015) 31
31. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3202–3212 (2015) 3
32. Roth, J., Chaudhuri, S., Klejch, O., Marvin, R., Gallagher, A., Kaver, L., Ramaswamy, S., Stopczynski, A., Schmid, C., Xi, Z., et al.: Ava-activespeaker: An audio-visual dataset for active speaker detection. arXiv preprint arXiv:1901.01342 (2019) 31
33. Rotman, D., Porat, D., Ashour, G.: Optimal sequential grouping for robust video scene detection using multiple modalities. International Journal of Semantic Computing **11**(02), 193–208 (2017) 30, 31

34. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015) 26, 29
35. Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., Trancoso, I.: Temporal video segmentation to scenes using high-level audiovisual features. IEEE Transactions on Circuits and Systems for Video Technology **21**(8), 1163–1177 (2011) 7, 16, 17
36. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: Movieqa: Understanding stories in movies through question-answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016) 3
37. Tian, Z., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: European conference on computer vision. pp. 56–72. Springer (2016) 9
38. Umesh, S., Cohen, L., Nelson, D.: Fitting the mel scale. In: 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258). vol. 1, pp. 217–220. IEEE (1999) 31
39. Vicol, P., Tapaswi, M., Castrejon, L., Fidler, S.: Moviegraphs: Towards understanding human-centric situations from videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018) 3, 23
40. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Val Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: ECCV (2016) 28, 29, 31, 32, 35
41. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017) 29
42. Xiong, Y., Huang, Q., Guo, L., Zhou, H., Zhou, B., Lin, D.: A graph-based framework to bridge movies and synopses. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019) 34, 36
43. Zhang, N., Paluri, M., Taigman, Y., Fergus, R., Bourdev, L.: Beyond frontal faces: Improving person recognition using multiple cues. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4804–4813 (2015) 12, 35
44. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person reidentification: A benchmark. In: Proceedings of the IEEE international conference on computer vision. pp. 1116–1124 (2015) 29, 30
45. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1452–1464 (2017) 22
46. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1452–1464 (2018) 31
47. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation (2002) 29, 30