STAR: Screen Time and Actor Recognition in Video Content

Tomas Kerepecky[1,2][0009−0008−2240−4993], Filip Sroubek[1][0000−0001−6835−4911], Barbara Zitova[1][0000−0003−0110−3741], and Jan Flusser[1][0000−0003−3747−9214]

[1] Institute of Information Theory and Automation, The Czech Academy of Sciences, Pod Vodarenskou vezi 4 CZ-18200, Prague, Czechia, kerepecky@utia.cas.cz
[2] Faculty of Nuclear Sciences and Physical Engineering, CTU in Prague, Brehova 78/7, CZ-11519, Prague, Czechia

Abstract. Accurately measuring the duration of actors' presence in videos is a challenging task that goes beyond actor recognition. We propose the STAR pipeline, the new model designed to analyze the time performers appear on screen across diverse video content, including movies and TV shows. The proposed model has been successfully deployed and tested by the Czech TV infrastructure provider. Our pipeline uses machine learning techniques for shot detection, face detection, tracking, recognition, and introduces a novel shot-based method for calculating screen time. We present extensive experiments proving the robustness and real-time performance of our approach. Alongside the pipeline, we introduce the STAR dataset to address the need for high-quality benchmarks in evaluating screen time models, now available for download.

Keywords: Screen Time · Actor Recognition · Video Analysis · Computer Vision · Machine Learning · Star Dataset.

1 Introduction

The rapid growth of online video content is highlighted by significant increases in both quantity and diversity, spanning movies, TV shows, and surveillance footage. Over 19 million titles are now listed on Internet Movie Database (IMDb), and the global TV and film industry's revenue is expected to reach tens of billions of dollars in 2024 [16,24].

Actors are central to many videos, influencing narratives and audience engagement. Efficiently recognizing these actors and analyzing their screen time offers valuable insights. This aids viewers in identifying content with their favorite performers. It also enhances video content management and indexing for applications in journalism, security, and targeted advertising across various commercial sectors.

Actor identification in videos is challenging, tackled through various methods ranging from utilizing text cues such as scripts and subtitles [20,6,31,11,18,23] to purely image-based techniques [1,29,25,10]. Generative appearance models have also been explored for detecting and naming actors in movies [9]. Advancements

in machine learning have led to significant improvements in face detection and recognition [30,3,21,14,28,4,15], yet applying these to videos poses challenges due to factors such as variable lighting, changing camera angles, and occlusions. DeepStar project, for instance, presented a unique approach for reference-free identification of starring characters [10].

Building on this and other existing methodologies, our work introduces a robust approach to the actor recognition problem, focusing not just on identifying actors but also on accurately calculating their screen time. Research on this topic is limited. To the best of our knowledge, no precise and verifiable benchmark dataset or baseline model for evaluating screen time accuracy is available; only fan-based or educational efforts like those documented by [27,17,26,2].

Contributions (1) We present an automated actor recognition pipeline, establishing a baseline for screen time calculation. (2) We propose a novel approach to screen time calculation by grouping frames into meaningful segments called shots, and calculating screen time on them. The shot-based method significantly enhances overall accuracy. (3) We introduce the STAR dataset, a per-frame, human-labeled dataset specifically designed to evaluate screen time, which is now available for download. This is the first benchmark dataset that quantifies screen time with respect to two visibility criteria: face and body.

Calculating screen time represents a novel and impactful domain in video analysis with numerous practical applications. Our pipeline, developed and successfully tested by the Czech TV infrastructure provider České Radiokomunikace a.s. (CRA), effectively analyzes actors' screen time in TV shows and movies, enhancing metadata for TV and film databases. Additionally, our approach holds significant potential for journalism, where it can aid in detailed analysis by efficiently searching through news footage and other broadcast videos.

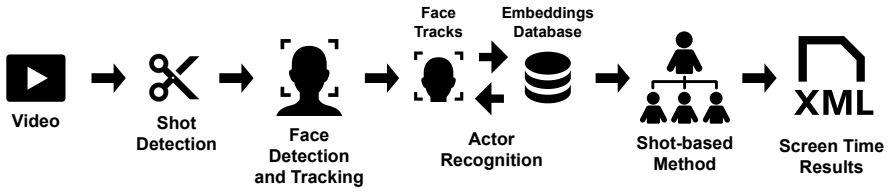

Fig. 1. STAR Pipeline: The process starts with shot detection, followed by face detection and tracking to form face tracks. These tracks are matched against an actor database for recognition. Based on the face track analysis, the shot-based method calculates screen time results for both known and unmatched face tracks.

2 STAR Pipeline

The STAR pipeline, as shown in Figure 1, is a framework designed to identify actors in videos and calculate their screen time. Initially, the video is divided into shots to improve efficiency and ensure precise actor time analysis. The pipeline then alternates between using face detection and face tracking to reduce the computational load.

Detected and tracked faces are linked across frames to form face tracks, which represent an actors' presence throughout the video. Each face track generates a faceprint—a numerical representation of facial features—used to compare against a database of actor faceprint embeddings to identify matches.

For unmatched face tracks, the shot-based method comes into play. This process clusters similar face tracks and determines the screen time for recognized actors on a per-shot basis.

2.1 Shot Detection

The shot detection module, or cut detection, splits videos into units known as shots, each a series of connected images from a single camera that depict a continuous event. We use SceneDetect library [19], which identifies frame changes indicating new shots using a content-based strategy and a predefined threshold. This process is critical for organizing videos into manageable parts and for accurate screen time calculations.

2.2 Face Detection and Tracking

After segmenting the video into shots, our pipeline uses a dual method to detect faces by deploying a MTCNN face detection algorithm [30] every nth frame and activating a dlib correlation tracker [5] on intervening frames where detection is off. Faces across frames are linked by calculating the spatial Euclidean distance between the centroids of newly detected and previously seen faces, determining whether to continue an existing track or start a new one. This method ensures consistent tracking of each actor throughout the video. The system also holds tracks temporarily for faces that disappear, using a predefined period to check for reappearance. When a face track ends, due to disappearance or a video shot boundary, it is saved, and the median faceprint embedding from all corresponding bounding boxes is calculated.

2.3 Actor Recognition with Faceprint Embeddings

Faceprint embeddings are numerical vectors representing faces in a low-dimensional space. These embeddings enable the quantitative comparison and analysis of faces, invariant to ordinary face rotation, facial expression, age, and partial occlusions like glasses. They are generated using the machine learning model which employs ResNet100 architecture [12], pretrained with ArcFace loss [4].

In contrast to the traditional SoftMax loss function, ArcFace introduces a margin between classes, enhancing actor class separability. The pretrained ArcFace model extracts deep features from aligned facial images and maps them onto a 512-dimensional hypersphere. Each face is represented as a vector, where the geodesic distance on the hypersphere correlates with visual similarity. This similarity between faces is quantified using cosine similarity, allowing for more precise differentiation and recognition of actors.

Following the faceprint encoding process, the matching phase compares each median faceprint from the face track against a pre-existing database of known actors' faceprints, seeking matches with the highest similarity that meet or exceed a predefined threshold. Faceprints for videos are computed online during processing, while those for actor databases may be precomputed offline.

2.4 Shot-based Method

Shot-based method consists of two parts: clustering and screen time calculation. Rather than quantifying actor recognition on a per-frame basis, it is assessed on a per-shot basis, which logically groups frames into meaningful segments reflecting continuous action or presence.

In the clustering phase, hierarchical clustering from the scipy library [22] groups unmatched face tracks with those from recognized actors that have similar features, enhancing the identification process for initially unrecognized actors. The shot-based screen time calculation employs heuristics that assume an actor remains in the shot for its entire duration. For actors who are sufficiently long in the shot, all frames from those shots are attributed to them. This approach is not only computationally efficient but also aligns with the narrative and visual continuity of the video content, making it a logical method for estimating screen time.

An alternative method, the face-only mode, quantifies actor presence strictly based on frames where the actor is directly detected or tracked. However, the shot-based mode, remains the default due to its higher abstraction level and alignment with how video content is typically structured and interpreted.

Ultimately, the pipeline outputs results in XML format and produces a video showcasing the detected actors.

3 STAR Dataset

The STAR dataset, detailed in Table 1, includes 18 one-minute clips from Czech TV shows 'Krejzovi' [7] and 'Zoo' [8], annotated for main actor appearances with respect to two visibility criteria: face and body. This dataset, featuring a mix of action, non-action, comedic, and dramatic content, is suggested as a benchmark in actor recognition and screen time analysis. It is available for download at https://star.kerepecky.eu or on Kaggle [13] under the same name as the paper.

During the annotation process, a human annotator compared the actors appearing in each frame with the images in the reference database to identify

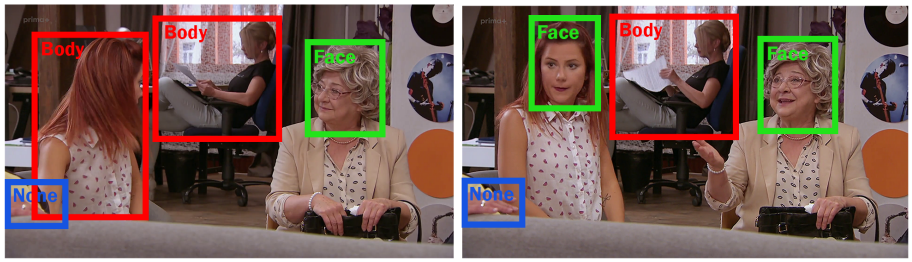

Fig. 2. An actor is labeled as 'Face' (green box) if recognizable by the face. Labeled as 'Body' (red box) if the face is occluded, but recognizable by the body, hair, or clothing. Not labeled (blue box) if the actor is not present or not distinguishable.

Table 1. STAR dataset

Video Name	Frames	Shots	Actors	Keywords
Krejzovi-bear	1500	22	3	comedy, humor, hospital, animal
Krejzovi-bullrun	1500	33	2	action, fear, ranch, animal
Krejzovi-goat	1500	10	3	comedy, surprise, garden, animal
Krejzovi-videochat	1500	14	5	drama, shame, home, family
Krejzovi-workshop	1500	12	2	comedy, letdown, workshop, inventor
Krejzovi-coach	1500	19	8	comedy, boredom, office, colleagues
Krejzovi-argument	1500	24	4	drama, conflict, office, boss
Krejzovi-crying	1500	17	6	comedy, tension, office, grandma
Krejzovi-family	1500	28	9	comedy, unease, dining-room, neighbor
Zoo-bedroom	1500	15	2	neutral, private, bedroom, couple
Zoo-boss	1500	15	2	neutral, calm, office, boss
Zoo-cafeteria	1500	22	4	drama, irony, cafeteria, colleagues
Zoo-kitchen	1500	20	3	neutral, warmth, kitchen, kid
Zoo-patrol	1500	9	2	action, nervous, zoo, thief
Zoo-twins	1500	17	5	neutral, social, bar, twins
Zoo-tender	1500	20	6	drama, tension, zoo, group
Zoo-silver-wig	1500	16	5	drama, fear, bar, singer
Zoo-black-mask	1500	12	6	drama, doubt, bar, singer

matches. Visibility is categorized as either "Face" or "Body" based on the recognizability of the actor. As demonstrated in Figure 2, if the actor's face is clearly visible and identifiable, the annotator records this appearance as "Face." If the face is obscured or partially visible but the actor can still be identified through other distinguishing features like hair or clothing, the visibility is recorded as "Body." The actual annotations do not include bounding boxes, as they are redundant for screen time calculation; Figure 1 includes them only to aid reader understanding. Instead, annotations are formatted in a structured data table,

specifying each actor, the shot, frame range, visibility type, and duration of visibility, as exemplified below:

```
Actor;Shot;Frame_start;Frame_end;Type;Duration
Ivana_Korolova;1;1;26;Body;26
Ludmila_Molinova;1;1;103;Face;103
...
Ivana_Korolova;1;56;103;Face;48
```

4 Experimental Results

4.1 Actor Presence in STAR dataset

We evaluates how effectively our pipeline can identify the presence of actors in videos from the STAR dataset. This challenge is similar to searching for videos that feature a specific actor; therefore, we are not concerned with screen time in this experiment. We compiled a database of 7,500 faces including 43 actors present in the analyzed videos. Identifying actor presence in each video translates to a binary classification task for each actor in the face database. Actors correctly identified in videos are marked as true positives, while those absent and correctly not detected are labeled as true negatives. We face two primary errors: false negatives, where present actors are overlooked, and false positives, where actors not in the video are mistakenly identified. The latter is particularly problematic for aforementioned video search tasks, as it leads to incorrect recommendations on streaming platforms, where the goal is to provide relevant rather than overwhelming content. Thus, we aim to maximize accurate identifications with zero false positives, prioritizing precision and specificity, potentially at the cost of recall.

Table 2. Validation dataset: evaluation metrics for actor presence recognition in 'Krejzovi' videos. TP, TN, FP, FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively. FP are set to zero; therefore, precision is not calculated.

Video Name	TP	FN	FP	TN	Precision	Recall
Krejzovi-bear	3	0	-	7497	-	100%
Krejzovi-bullrun	2	0	-	7498	-	100%
Krejzovi-goat	3	0	-	7497	-	100%
Krejzovi-videochat	5	0	-	7495	-	100%
Krejzovi-workshop	2	0	-	7498	-	100%
Krejzovi-family	8	1	-	7491	-	88.9%
Krejzovi-coach	7	1	-	7492	-	87.5%
Krejzovi-crying	5	1	-	7494	-	83.3%
Krejzovi-argument	3	1	-	7496	-	75%

Table 3. Testing dataset: evaluation metrics for actor presence recognition in 'Zoo' videos. TP, TN, FP, FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively.

Video Name	TP	FN	FP	TN	Precision	Recall
Zoo-bedroom	2	0	0	7498	100%	100%
Zoo-boss	2	0	0	7498	100%	100%
Zoo-cafeteria	4	0	0	7496	100%	100%
Zoo-kitchen	3	0	0	7497	100%	100%
Zoo-patrol	2	0	0	7498	100%	100%
Zoo-twins	5	0	0	7495	100%	100%
Zoo-tender	5	1	0	7494	100%	83.3%
Zoo-silver-wig	4	1	0	7495	100%	80%
Zoo-black-mask	4	2	0	7494	100%	66.7%

Fig. 3. Actor presence detection challenges in STAR dataset: Missing face detection in 'Krejzovi-family', 'Krejzovi-crying', 'Krejzovi-argument', and 'Zoo-silver-wig' highlights issues with occlusions. High facial recognition threshold caused mismatches in 'Krejzovi-coach' and 'Zoo-black-mask'. Eventually, the same identity was incorrectly assigned to twin actors in 'Zoo-tender'.

For our validation set, we use videos from the TV show 'Krejzovi,' setting the facial recognition threshold at 0.40 to ensure zero false positives. As indicated in Table 2, this setup correctly classifies all actors with 100% recall in 5 out of 9 videos. Failures, as detailed in Figure 3, are attributed to factors such as completely obscured faces or partially visible faces, which our system fails

to detect. In the video 'Krejzovi-coach,' the high facial recognition threshold prevents the recognition of the actor.

We use the TV show 'Zoo' as a testing set. Table 3 shows that our model maintained high precision, accurately classifying all actors in 6 out of 9 videos. Issues arose in 'Zoo-black-mask' due to the strict facial recognition threshold. In 'Zoo-silver-wig' and 'Zoo-tender,' challenges were intentionally introduced in the STAR dataset to test the limits of the facial detection and recognition model, with items such as microphones and wigs obstructing facial features, and twin actors being misclassified as a single person.

While minimizing false positives is crucial for streaming platforms to provide accurate recommendations, a different approach is required for surveillance footage and news broadcasting, where limiting false negatives is more important. Missing a person's appearance in surveillance could overlook a security threat or key evidence, and in news broadcasting, it could lead to misinformation. Therefore, we prioritize recall over precision in these cases, even if it results in more false positives.

4.2 Shot-based Calculation Analysis

As described in Section 2.4, screen time calculation through our pipeline operates in two modes. The face-only mode calculates screen time based solely on the number of frames where an actor's face is detected or tracked. The shot-based mode employs heuristics assuming the actor remains in the shot for the entire duration. We first compare the performance of these modes on 1-minute clips from the movie 'The Hitman's Bodyguard.' Selected frames from both scenes are shown in the first row of Figure 4. Scene 1 represents a non-action part of the movie, while Scene 2 represents an action scene. Comparisons of the results from these clips are shown in the second row of Figure 4. The shot-based approach significantly enhances the accuracy of screen time detection, achieving near-perfect results in regular scenes. However, even with this noticeable improvement, action clips still face significant errors due to reduced face detection accuracy caused by blur and occlusion in fast scenes.

Occasionally in movies, an actor may leave a scene before the shot ends or enter in the middle, which can potentially overestimate their screen time. However, this is relatively rare, and when calculating screen time over longer videos, this aspect is marginal. For fast-paced videos, where shots are very short, it is even less significant. For demonstration, Table 4 compares ground truth screen time per-frame versus per-shot for a non-action scene from 'The Hitman's Bodyguard' movie. For two actors, the screen time remains the same. Screen time for actor Tine Joustra is 1% higher when calculating per-shot, as her full body was occluded by the closing door during the initial frames, which is not counted in the per-frame mode.

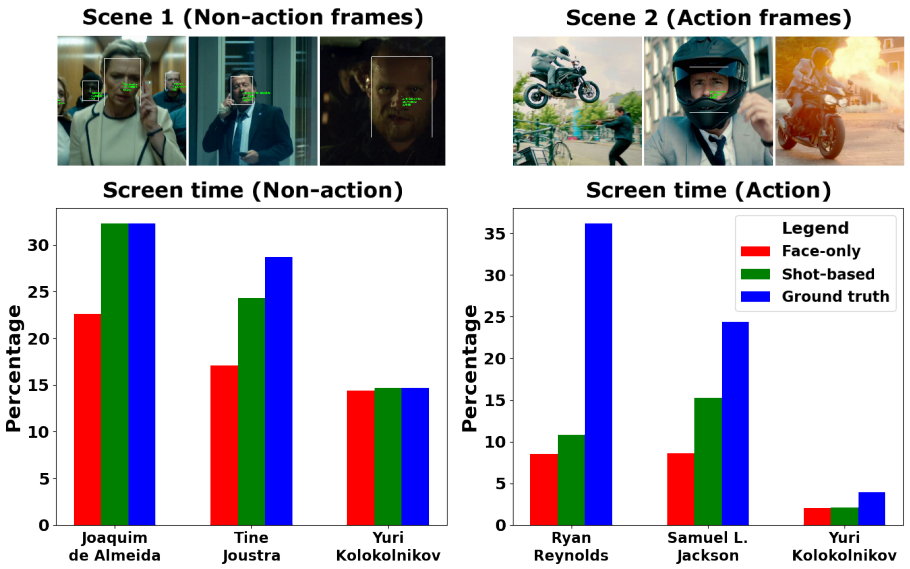

Fig. 4. Comparing methods of calculating screen time in scenes from 'The Hitman's Bodyguard.' The first row illustrates frames from non-action and action scenes. The second row demonstrates improvements in screen time calculation using a shot-based method over a face-only mode.

Table 4. Ground-truth (GT) screen time in non-action scenes from 'The Hitman's Bodyguard' annotated either per-frame or per-shot.

	Joaquim de Almeida	**Tine Joustra**	**Yuri Kolokolnikov**
GT per-frame	32.3%	28.7%	14.7%
GT per-shot	32.3%	29.7%	14.7%

4.3 Screen Time in STAR Dataset

In the TV industry, both the precision and speed of actor analysis are crucial for effective application. We investigate the impact of the face detection frequency on actor screen time accuracy and processing speed using videos from the STAR dataset.

As detailed in Section 2.2, our method alternates between a face detection algorithm applied every nth frame and a correlation tracker during intervening frames. This strategy balances computational load, with the face detection component being the most resource-intensive. Figure 5 illustrates that the average frames per second (FPS) across all STAR dataset videos varies with detection frequency. Real-time operation on an NVIDIA GeForce RTX 2080 Ti is feasible when the detector is activated every 9-10 frames, as indicated by the yellow band in the figure.

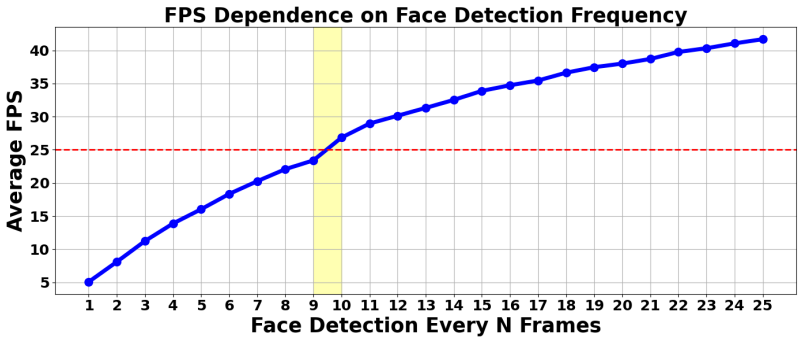

Fig. 5. Processing speed, measured as average frames per second (FPS), across the STAR dataset videos varies with face detection frequency. The yellow band highlights the smallest detection interval within which real-time processing is achievable.

Extended intervals between face detections can lead to missed actor identifications, reducing screen time accuracy (Figure 6). We assessed the mean absolute error (MAE) between our pipeline results and ground-truth (GT) annotations for both subsets of the STAR dataset. Errors were measured for the GT actor's face visibility and GT for any visible part of the actor. The 'Krejzovi' videos showed smaller discrepancies and maintained high accuracy despite infrequent detections. In contrast, 'Zoo' videos experienced increases in MAE for full-body screen time. This is especially true in challenging videos like 'Zoo-patrol' and 'Zoo-silver-wig', where actor detection based on face recognition leads to errors when faces are not fully visible. However, MAE curves increase very slowly, demonstrating the high robustness of the proposed approach.

For the STAR/Krejzovi subset, the lowest errors are achieved by using the face detector on every frame, yielding face/full-body MAE values of 3.82%/6.05%. Table 5 details screen time percentages calculated using our pipeline and GT annotation for each actor. The results for the STAR/Zoo subset are in Table 6.

We assume that the identities of the actors are already known, either from video metadata or from a previous actor presence detection task. Therefore, we use smaller, more targeted face databases – specifically, 24 actors for 'Krejzovi' videos. This adjustment allows us to lower the face recognition threshold from 0.4 to 0.3, effectively recognizing all actors in 'Krejzovi-coach', unlike the failures highlighted in Section 4.1. Nonetheless, limitations of the face detection system in identifying actor presence, as described in Section 4.1, results in some instances of zero screen time, such as in 'Krejzovi-family'.

It is worth mentioning that for some actors the shot-based method reaches full screen presence, even without a body recognition component. For example, in 'Krejzovi-family', our system suggests a full body presence of 51.2% for the actor "Hybnerova". However, a limitation of our approach is that the shot-based mode might slightly overestimate screen time in some instances, as observed with a 1% increase for actor 'Stastny' in 'Krejzovi-goat'.

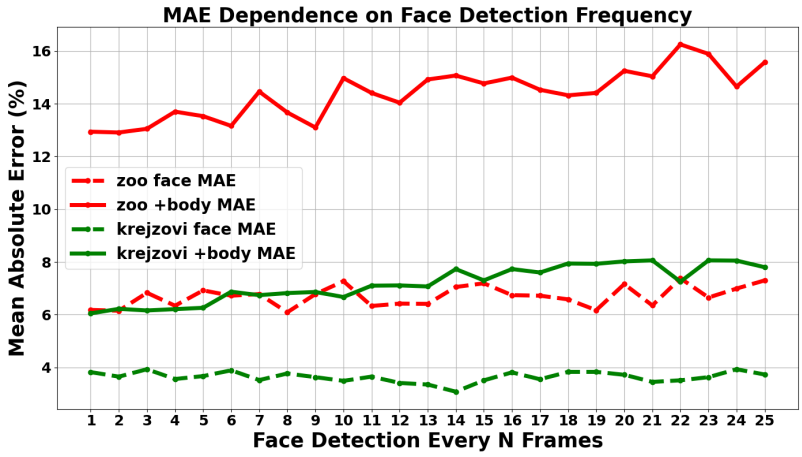

Fig. 6. Mean Absolute Error (MAE) trends showing the impact of face detection frequency on actor recognition accuracy for 'Zoo' and 'Krejzovi' video subsets. Errors are divided into 'face' based on ground truth face visibility, and '+body' based on ground truth face and body visibility.

5 Acknowledgments

This work was supported by the Czech Science Foundation GA 24-10069S. We also thank FTV Prima, spol. s.r.o. for providing video episodes from TV shows Krejzovi [7] and ZOO [8], which were essential for building our dataset. Gratitude is also extended to České Radiokomunikace a.s. (CRA) for their support in the development and application of the discussed code.

6 Conclusion

We introduced the STAR pipeline and dataset to accurately measure actor screen time in videos. The pipeline combines shot detection, face detection and tracking, actor recognition, and a novel shot-based method, showing significant accuracy improvements.

Extensive testing demonstrated robustness of our approach and real-time performance. The shot-based method effectively handles occlusions, achieving error rates as low as 3.82% for videos from STAR dataset. This pipeline, successfully tested by the Czech TV infrastructure provider, offers a scalable solution for media content management.

The STAR dataset, with detailed annotations, sets a standard for evaluating screen time models. Our new dataset is publicly available, promoting further research in this field. Future work will enhance face detection and recognition algorithms and incorporate more contextual information, such as body or clothing recognition, to improve screen time estimation.

Table 5. Screen time calculation for videos in the STAR/Krejzovi dataset. The percentages represent the visibility of each actor based on three criteria: 'face' for ground-truth visibility of the face, '+body' for ground-truth visibility of any part of the actor, and 'ours' for screen time calculated using our pipeline.

workshop	face	+body	ours	**bullrun**	face	+body	ours
Postranecky	61.7%	61.7%	57.9%	Hruska	22.7%	22.7%	20.3%
Suchanek	80.7%	87.9%	87.8%	Stastny	24.9%	55.9%	16.9%
bear	face	+body	ours	**goat**	face	+body	ours
Tomicova	71.7%	71.7%	71.7%	Tomicova	64.7%	72.3%	72.2%
Mrazik	61.2%	68.7%	61.1%	Marysko	63.0%	72.3%	72.2%
Stastny	52.7%	54.1%	55.9%	Stastny	85.1%	90.7%	91.7%
family	face	+body	ours	**coach**	face	+body	ours
Korolova	7.6%	15.3%	0.0%	Korolova	43.1%	50.7%	48.5%
Zapletal	6.5%	15.3%	4.3%	Stastna	9.3%	21.0%	14.1%
Blazek	19.1%	66.9%	21.5%	Revai	40.3%	43.1%	39.7%
Hrachovcova	53.6%	62.7%	62.2%	Pletankova	20.7%	20.7%	21.1%
Hybnerova	47.0%	51.2%	51.2%	Kocianova	25.7%	26.5%	27.3%
Halouzkova	22.2%	30.9%	24.1%	Novotny	47.7%	52.5%	53.2%
Sadlonova	10.8%	15.3%	6.0%	Boucek	59.4%	61.2%	61.7%
Stastny	22.9%	26.1%	24.9%	Stepan	20.1%	25.4%	27.3%
Plankova	39.1%	55.9%	42.8%				
videochat	face	+body	ours	**crying**	face	+body	ours
Polisenska	45.5%	50.9%	45.5%	Korolova	46.0%	48.6%	48.6%
Korolova	43.3%	48.5%	45.5%	Novotny	47.7%	55.0%	47.6%
Hybnerova	49.9%	54.3%	49.9%	Pletankova	0.0%	37.7%	0.0%
Postranecky	34.7%	48.1%	49.3%	Kocianova	44.0%	47.7%	43.9%
Plankova	67.3%	77.2%	72.8%	Molinova	37.7%	37.7%	37.7%
				Stepan	12.3%	14.7%	12.3%
argument	face	+body	ours				
Kocianova	53.7%	59.3%	62.7%				
Pletankova	0.2%	0.2%	0.0%				
Korolova	54.8%	63.7%	64.5%				
Revai	45.3%	46.5%	41.8%				

Table 6. Screen time calculation for videos in the STAR/Zoo dataset. The percentages represent the visibility of each actor based on three criteria: 'face' for ground-truth visibility of the face, '+body' for ground-truth visibility of any part of the actor, and 'ours' for screen time calculated using our pipeline.

boss	face	+body	ours	**patrol**	face	+body	ours
Nesvacilova	16.3%	23.6%	19.3%	Novotny	19.1%	71.9%	43.4%
Gransky	67.1%	77.9%	65.5%	Sobotka	31.9%	96.6%	36.3%
kitchen	face	+body	ours	**bedroom**	face	+body	ours
Klus	45.3%	48.3%	48.2%	Klus	80.1%	84.6%	70.9%
Buresova	49.7%	49.7%	49.7%	Buresova	83.8%	100.0%	93.9%
Klusova	50.3%	82.1%	50.3%				
twins	face	+body	ours	**silver-wig**	face	+body	ours
CernaL	35.9%	38.0%	14.3%	Buresova	34.1%	34.1%	0.0%
Buresova	47.5%	47.5%	47.4%	Brumovska	32.5%	60.9%	37.3%
Razlova	28.3%	52.3%	35.1%	Razlova	9.3%	16.4%	12.7%
Barta	30.2%	30.2%	30.2%	Barta	6.9%	10.0%	10.0%
CernaB	37.0%	43.9%	38.0%	Tomicova	23.7%	60.9%	23.7%
tender	face	+body	ours	**black-mask**	face	+body	ours
CernaL	26.1%	26.5%	0.0%	Bilina	10.1%	19.0%	10.1%
Necas	39.9%	47.3%	37.9%	Buresova	29.3%	45.5%	19.5%
Nesvacilova	36.9%	55.3%	47.3%	Pechackova	49.9%	62.1%	56.9%
Hudeckova	20.1%	24.6%	25.1%	Razlova	6.6%	20.9%	3.7%
Gransky	42.9%	44.1%	47.1%	Benoni	3.7%	3.7%	3.7%
CernaB	25.9%	26.1%	25.2%	Barta	10.9%	10.9%	10.9%
cafeteria	face	+body	ours				
Genzer	30.5%	33.1%	33.1%				
Novakova	51.6%	77.6%	53.7%				
Novotny	17.1%	33.1%	27.5%				
Zilkova	57.5%	75.3%	60.1%				

References

1. Aggarwal, A., Pandya, Y., Ravindranathan, L.A., Ahire, L.S., Sethu, M., Nandy, K.: Robust actor recognition in entertainment multimedia at scale. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 2079–2087 (2022)
2. Chopade, A.: Screen time calculation (2024), https://github.com/AbhishekChopade/Screen_Time_Calculation, accessed May 17, 2024
3. Deng, J., Guo, J., Ververas, E., Kotsia, I., Zafeiriou, S.: Retinaface: Single-shot multi-level face localisation in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5203–5212 (2020)
4. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4690–4699 (2019)
5. Dlib: Modern c++ toolkit for machine learning algorithms and complex software development (2024), http://dlib.net/, accessed March 24, 2024
6. Everingham, M., Sivic, J., Zisserman, A.: Hello! my name is... buffy–automatic naming of characters in tv video. In: BMVC. vol. 2, p. 6 (2006)
7. FTV Prima, s.s.: Krejzovi (2018), television series produced by FTV Prima, spol. s.r.o.
8. FTV Prima, s.s.: Zoo (2022), television series produced by FTV Prima, spol. s.r.o.
9. Gandhi, V., Ronfard, R.: Detecting and naming actors in movies using generative appearance models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3706–3713 (2013)
10. Haq, I.U., Muhammad, K., Ullah, A., Baik, S.W.: Deepstar: Detecting starring characters in movies. IEEE Access **7**, 9265–9272 (2019)
11. Haurilet, M.L., Tapaswi, M., Al-Halah, Z., Stiefelhagen, R.: Naming tv characters by watching and analyzing dialogs. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1–9 (2016)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
13. Kerepecký, T.: Star Screen Time and Actor Recognition in Videos. https://www.kaggle.com/datasets/tomaskerepecky/star-screen-time-and-actor-recognition-in-videos (2024)
14. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 212–220 (2017)
15. Meng, Q., Zhao, S., Huang, Z., Zhou, F.: Magface: A universal representation for face recognition and quality assessment. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14225–14234 (2021)
16. Needham, C.: Internet movie database (imdb) (2024), https://www.imdb.com/pressroom/stats/, accessed: Mar. 23, 2024
17. ninewheels0: Screen time breakdown episode-by-episode: Hawkeye. IMDb (2024), https://m.imdb.com/list/ls565685241/, accessed: May 10, 2024
18. Parkhi, O.M., Rahtu, E., Cao, Q., Zisserman, A.: Automated video face labelling for films and tv material. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(4), 780–792 (2018)
19. PySceneDetect: Shot change detection tool for videos (2024), https://www.scenedetect.com/, accessed March 24, 2024

20. Satoh, S., Kanade, T.: Name-it: Association of face and name in video. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 368–373 (1997)
21. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
22. SciPy: Fundamental algorithms for scientific computing in python (2024), https://scipy.org/, accessed March 24, 2024
23. Sivic, J., Everingham, M., Zisserman, A.: "who are you?"-learning person specific classifiers from video. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1145–1152 (2009)
24. Statista: Statista film and tv video industry statistics (2024), https://www.statista.com/markets/417/topic/476/tv-video-film, accessed: Mar. 23, 2024
25. Tapaswi, M., Bäuml, M., Stiefelhagen, R.: "knock! knock! who is it?" probabilistic person identification in tv-series. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2658–2665 (2012)
26. TheAILearner: Calculating screen time of an actor using deep learning (2024), https://theailearner.com/2019/10/14/calculating-screen-time-of-an-actor-using-deep-learning/, accessed May 17, 2024
27. University, S.: Counting actor screen time in movies (2024), https://cs230.stanford.edu, accessed May 17, 2024
28. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5265–5274 (2018)
29. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: CVPR 2011. pp. 529–534 (2011)
30. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters **23**(10), 1499–1503 (2016)
31. Zhang, Y.F., Xu, C., Lu, H., Huang, Y.M.: Character identification in feature-length films using global face-name matching. IEEE Transactions on Multimedia **11**(7), 1276–1288 (2009)