Automatic Thumbnail Selection for Soccer Videos using Machine Learning

Andreas Husa
SimulaMet, Norway

Cise Midoglu
SimulaMet, Norway

Malek Hammou
SimulaMet, Norway

Steven A. Hicks
SimulaMet, Norway

Dag Johansen
UIT The Artic University of Norway

Tomas Kupka
Forzasys AS, Norway

Michael A. Riegler*
SimulaMet, Norway

Pål Halvorsen†‡
SimulaMet, Norway

ABSTRACT

Thumbnail selection is a very important aspect of online sport video presentation, as thumbnails capture the essence of important events, engage viewers, and make video clips attractive to watch. Traditional solutions in the soccer domain for presenting highlight clips of important events such as goals, substitutions, and cards rely on the manual or static selection of thumbnails. However, such approaches can result in the selection of sub-optimal video frames as snapshots, which degrades the overall quality of the video clip as perceived by viewers, and consequently decreases viewership, not to mention that manual processes are expensive and time consuming. In this paper, we present an automatic thumbnail selection system for soccer videos which uses machine learning to deliver representative thumbnails with high relevance to video content and high visual quality in near real-time. Our proposed system combines a software framework which integrates logo detection, close-up shot detection, face detection, and image quality analysis into a modular and customizable pipeline, and a subjective evaluation framework for the evaluation of results. We evaluate our proposed pipeline quantitatively using various soccer datasets, in terms of complexity, runtime, and adherence to a pre-defined ruleset, as well as qualitatively through a user study, in terms of the perception of output thumbnails by end-users. Our results show that an automatic end-to-end system for the selection of thumbnails based on contextual relevance and visual quality can yield attractive highlight clips, and can be used in conjunction with existing soccer broadcast pipelines which require real-time operation.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Activity recognition and understanding*; **Video summarization**.

*Also affiliated with UIT The Artic University of Norway
†Also affiliated with Oslo Metropolitan University, Norway
‡Also affiliated with Forzasys AS, Norway

MMSys '22, June 14–17, 2022, Athlone, Ireland

ACM ISBN 978-1-4503-9283-9/22/06.
https://doi.org/10.1145/3524273.3528182

KEYWORDS

blur detection; deep learning; image quality; logo detection; object detection; shot boundary detection; soccer; sports analysis; thumbnail generation; user survey; video

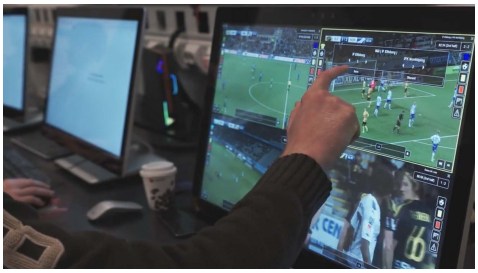
ACM Reference Format:
Andreas Husa, Cise Midoglu, Malek Hammou, Steven A. Hicks, Dag Johansen, Tomas Kupka, Michael A. Riegler, and Pål Halvorsen. 2022. Automatic Thumbnail Selection for Soccer Videos using Machine Learning. In *13th ACM Multimedia Systems Conference (MMSys '22), June 14–17, 2022, Athlone, Ireland.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3524273.3528182

1 INTRODUCTION

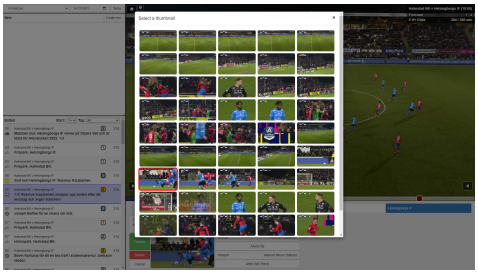
Sports broadcasting and streaming are immensely popular, and the interest in viewing videos from sports events grows day by day. Today, live streaming of sports events generates most of the video traffic and is replacing live broadcasting on TV [38]. For example, 3.572 billion viewers tuned in to watch the 2018 FIFA World Cup [9], and as of 2020, soccer had a global market share of about 45% of the $500 billion sports industry [34]. However, the availability of content and the large number of games make systems for extracting highlights and providing summaries in real- or near real-time increasingly important. The generation of video summaries and highlight clips from sports games is of tremendous interest for broadcasters, as a large percent of audiences prefer to view only the main events in a game.

The state-of-the-art systems for generating soccer highlight clips consist of several manual operations. A typical tagging center is shown in Figure 1, where important events such as goals, substitutions, and cards are tagged and annotated before being published into individual clips. A typical pipeline consists of a detection phase where the video is marked with an event such as a goal or card (Figure 1a), relevant frames are cut to generate a highlight clip, and in a second "refinement" phase, the highlight clip is further improved by customized trimming, insertion of additional textual descriptions and tags, and the selection/update of a thumbnail (Figure 1b).

A thumbnail is an image representing a video. For soccer, thumbnails are frequently used in web pages where various highlight clips are presented in gallery form [2, 8], and serve as the first impression meant to attract people to view a highlight clip. As highlight clips summarize certain important events in a soccer game, the challenge is to find appropriate thumbnails for each clip (and not a single thumbnail for the overall game). Thumbnails need to be



(a) An event is detected, classified and annotated.



(b) A thumbnail is selected for the highlight clip.

Figure 1: A live tagging operation by people in a cumbersome, error-prone, and tedious manual process.

selected carefully to be eye-catching and should properly represent the event in the highlight clip, as unattractive thumbnails can cause low engagement (highlight clips might go unwatched due to non-appealing thumbnails). Manual selection can potentially yield appropriate thumbnails, but since the selection operation is time-consuming and expensive, image quality is often not considered extensively. In this sense, automating the thumbnail selection process has the potential to both save resources and improve quality.

In this work, our goal is to replicate the potential performance of manual thumbnail selection with an automated system, which is much faster (applicable in real-time), cheaper, and quantitative (documentable and reproducible). This will not only save resources for the top soccer leagues which already rely on manual selection, but also enable similar services for less resource-capable leagues where automation is the only alternative. We develop an AI-based solution to identify appropriate thumbnail candidates by checking the video frames for logos, scene boundaries, faces, and analyzing image quality. Our proposed approach considers relevance to video content along with visual quality and aesthetic metrics, so that the resulting thumbnail is adequately representative of the video clip. In particular, we make the following contributions:

- We propose a modular and customizable automatic thumbnail selection pipeline, which integrates pre-processing (options for trimming, down-scaling, and down-sampling), logo detection, close-up shot detection, face detection, and image quality analysis (image quality prediction and blur detection). The modular and lightweight implementation allows for the agile integration and benchmarking of various Machine Learning (ML) models from literature in each module[1]. This pipeline can be controlled via a dashboard with a user-friendly Graphical User Interface (GUI).
- We evaluate our proposed pipeline quantitatively using various soccer datasets, in terms of system performance (complexity and runtime) as well as adherence to a pre-defined ruleset, under different configurations for components.
- We run a subjective evaluation (user study) involving 42 participants, to evaluate the performance of the proposed pipeline qualitatively. The subjective evaluation campaign is conducted using a novel survey framework for crowdsourced feedback collection on multimedia assets called HOST-ATS, which is a plug-and-play system component for any future studies as well[2].
- We additionally run a small benchmark study with a state-of-the art generic (non soccer-specific) thumbnail selector.
- We provide a discussion of the generalizability of our approach, along with the limitations and pitfalls of our pipeline, and suggest potential improvements and future work topics.

Overall, the novelty of our work includes: the combination of various independently applicable approaches in an end-to-end automatic thumbnail selection pipeline, with an overarching goal and a generalized ruleset; corresponding quantitative performance and complexity analyses; a novel user study framework for the qualitative evaluation of different thumbnail candidates; as well as the insights gained from the study and subsequent discussions. Our automatic thumbnail selection system is able to reduce production costs by automating a traditionally complex and labor-intensive task, accompanied by end-user validation. Our approach is applicable to various other sports broadcasts, such as skiing, handball, or ice hockey, and presents a viable potential to impact future sports productions.

The rest of this paper is structured as follows. In Section 2, we provide background information and an overview of related work. In Section 3, we describe our proposed thumbnail selection pipeline in detail. In Section 4, we present the results from our quantitative analysis of the proposed pipeline. In Section 5, we introduce a user study framework, and present the results from our qualitative analysis of the proposed thumbnail selection pipeline through subjective evaluation. In Section 6, we discuss a number of relevant aspects including limitations and potential future work. In Section 7, we conclude the paper.

2 BACKGROUND AND RELATED WORK

Soccer video production systems can incorporate research findings from different fields such as object detection [17, 23, 26], shot boundary detection [18, 42], event detection and clasification [5, 13, 20, 21, 24, 27, 28, 30, 35], and event clipping [4, 33, 36, 40]. In this work, our particular focus is on automatic thumbnail selection. As a representative snapshot, thumbnails capture the essence of a video and provide the first impression to the viewers. A good thumbnail makes the video clip more attractive to watch [19, 31]. Various components from the above fields can facilitate the identification of appropriate images that best describe the events in a video sequence and are of high quality.

[1]The proposed pipeline is not limited to the models currently presented in this paper, alternative models can be added with relative ease.

[2]Live deployment of the HOST-ATS subjective evaluation (user study) framework: https://host-ats.herokuapp.com

There is not much work on automatic thumbnail selection specifically for sports videos, but there are a number of related works that focus on thumbnail selection in general. Song et al. [31] propose a generic model called "Hecate" for selecting thumbnails automatically. Their framework uses a video as input, and filters the frames that are qualified as low-quality such as blurry, dark or uniform-colored frames. This is calculated with and decided upon via a threshold value, and not through ML. The framework also filters frames that are related to fading, dissolving or wiping effects in the video, identifying these through a shot boundary detection model. In a second step, frames that are near duplicates are discarded, and finally, frames with highest aesthetic quality are selected. This can be done by selecting the frame from a cluster with the smallest difference value (i.e., the frame that has the least change from the other frames in the same cluster, or the mean frame), where clusters are frames that have visual similarities. The aesthetic quality can be calculated by using a model that assigns a beauty score to a given image. This model has been trained by a set of images that have been annotated with subjective aesthetic scores. Vasudevan et al. [37] present a query-adaptive video summarization model which picks frames from a given video that are relevant to the given query. The model also has the possibility to output a single frame as a thumbnail. The query is a text of what content the end-user would like the frame to contain (e.g., in our context, it could be "soccer" or "goal").

A number of Image Quality Analysis (IQA) models for predicting the subjective and/or objective quality of an image have also been proposed. The model by Jongyoo [14] was tested for detecting distortion types on images, primarily white noise and blur. This is a no-reference image quality assessment (NR-IQA) model meant to be used for assessment without reference images. In some practical scenarios, there may be no reference image, then it can be useful to have a general IQA.

As our focus is on soccer, and there can be a lot of motion in soccer videos resulting in several frames ending up being blurry, blur detection is an important field of research. When selecting a single frame from a video, it is important to avoid images that are too blurry for aesthetic reasons, but also keeping images which are informative and representative of the event. It should be possible to see what is happening in the image, and too much blur could avoid that. In this context, the blur detection operator Laplacian from the OpenCV library [3] is also relevant. The operator outputs a value for a given image and predicts the presence of blur with a score, where the higher the score is, the less blur is predicted.

3 THUMBNAIL SELECTION PIPELINE

Related work indicates that a good thumbnail is relevant to the corresponding video, and appears interesting and attractive in terms of content and image quality [16, 19, 31]. In this regard, we center our proposed thumbnail selection pipeline around 3 key principles, namely relevance, content, and image-quality.

Relevance: The thumbnail that our proposed pipeline selects will be a frame from the video it is supposed to represent (i.e., no external images are considered related to the clip by the automated system). Video frames from the highlight clip will be used as input to our pipeline, so the output image is relevant to the video as it is a frame around the event annotation.

Content: Highlight clips are usually presented in a gallery as a grid, where each thumbnail appears in a small size (e.g., 200 pixels) on the screen. It could be difficult for viewers to understand the contents of the thumbnail if the image displays a long-distance shot of the soccer field. Therefore, we resolve to use close-up shots in our pipeline. Close-up shots are usually frames showing the soccer players, spectators (audience), and managers. There could also be frames from the replay of the event with shots that are closer than the default long-distance shot. If a frame is identified as a close-up shot, it will have a higher priority in the thumbnail selection process. We would also like to omit graphics such as the logo transitions appearing before replays. So, if a frame is identified as containing a logo, it will not be used as a thumbnail.

Image quality: It is possible that there are frames in a video which appear aesthetically unpleasing or unclear to the human eye. For instance, images that are blurry, dark, and/or fading are not usable as thumbnails. Therefore, we propose to undertake image quality analysis as the final filter in our pipeline.

Category	No	Rule
Relevance	1	The thumbnail should be a frame from the video clip itself.
Content	2	The frame should be a close-up shot of people.
	3	The frame should contain a face.
	4	The frame should not contain graphics (e.g., logo).
	5	The frame should not contain visuals of a fading transition.
Image Quality	6	The frame should not be blurry.
	7	The frame should not be dark.

Table 1: Thumbnail selection rules for our proposed pipeline.

Based on the above observations, we devise a number of thumbnail selection goals, which are listed in Table 1. These rules do not mean that our definition of a good thumbnail is universal, but rather establish a framework which is tailored for soccer videos. Our pipeline consists of 3 steps: pre-processing, content analysis and priority assignment, and image quality analysis. Figure 2 presents these steps and the corresponding components in our pipeline. A video clip (sequence of video frames) is fed as input to the pipeline, and the final output is an image, which is a frame from the video clip, as the suggested thumbnail.

3.1 Step 1: Pre-processing

In the pre-processing step, the sequence of frames[3] can be trimmed, down-sampled, and/or down-scaled. *Trimming* refers to the cutting of a desired number of seconds from the beginning and/or end of the sequence. It is also possible to define a time interval of interest with respect to an event annotation. This allows for increasing the relevance of the thumbnail candidates to the central event in the clip, by ensuring that images come from around the event annotation timestamp. *Down-sampling* refers to the extraction of a lower number of frames as a subset from the full set of frames in the sequence. It allows for decreasing the number of frames that are considered further on in the pipeline, consequently decreasing the amount of processing time. *Down-scaling* refers to the reduction of image resolution (changing the resolution of individual frames) in terms of a percentage. Table 8 shows the influence of down-scaling on accuracy. Each of these operations are optional, with

[3]Sequence of frames refers to the frames in the original input (video clip). E.g., a 30 second video clip at 30fps would yield 900 frames.

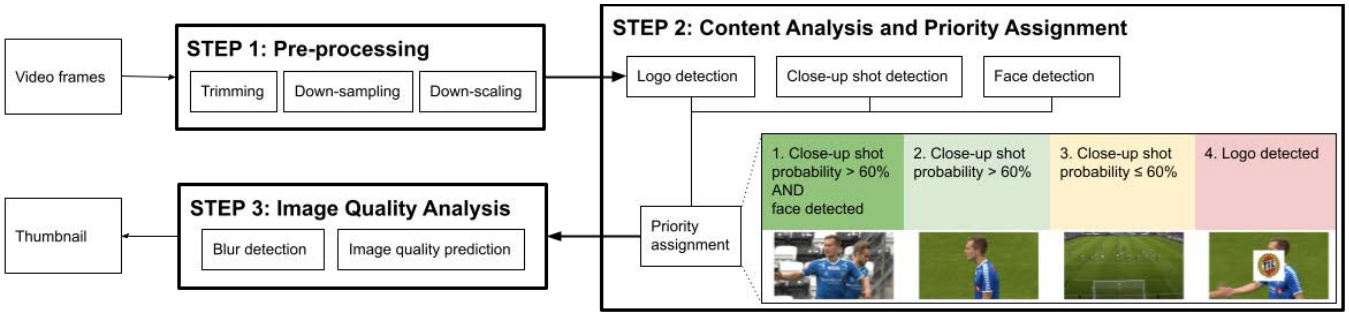


Figure 2: Proposed automatic thumbnail selection pipeline.

configurable parameters. The set of frames remaining after the pre-processing step is passed to the next step.

3.2 Step 2: Content Analysis and Priority Assignment

In the second step, the contents of the frames passed on from the first step are analyzed according to rules 2 – 5 from Table 1. For this purpose, independent modules for logo detection, close-up shot detection, and face detection are used, after which priorities are assigned to each frame and the frames are filtered accordingly.

3.2.1 Logo Detection. The logo detection module is for detecting logos and other graphics that might appear in the video frames. These are frames we would like to eliminate from the thumbnail selection. As an alternative model for this module, we train a convolutional neural network (CNN) based on the architecture presented by Surma [32]. This is a general image classification model that can classify images based on a given dataset. The output of the model is a probability score between 0 and 1, where an output of 0.5 or above indicates that the image contains a logo. Alternatively, the model by Ocampo [25] can also be used, which in turn is based on a model proposed by Jongyoo et al. [14]. The logo detection model and the threshold value are configurable parameters.

3.2.2 Close-up Shot Detection. The close-up shot detection module is used to decide whether a video frame depicts a scene coming from a wide-angle camera (zoomed-out, long-distance shot), or a close-up (zoomed-in) shot. Images classified as close-up shots are prioritized in thumbnail selection. Our pipeline supports the use of the image classification model by Surma [32] in this module. The threshold value for model certainty is a configurable parameter. Unlike the logo detection module, images with a probability below this threshold are not omitted, but receive a lower priority.

3.2.3 Face Detection. Face detection is used to detect the appearance of a face on a given image, as well as where the face appears on the image. In our context, we consider a thumbnail image to be more relevant if there is a face appearing in it. Traditionally, face detection models rely on frontal views, and cannot detect a person's head from behind. It is possible to consider images with faces turned to an angle (such as a person's head from the side or behind) to be just as relevant. However, models for detecting such phenomena are more complex and less accurate, so in this work we only consider frontal face views for the sake of simplicity and accuracy. Our pipeline currently supports 4 alternative models for this module. Haar cascade [39] is an object detection model which is fast, but tends to be prone to false positive detection, compared to other models [29]. This algorithm can be run in real-time, making it possible to detect objects in live video streams. It is possible to train the model for detecting other objects as well as faces. It is capable of detecting objects regardless of their scale and position in an image. Dlib [15] uses features extracted by histogram of oriented gradients (HOG), and passes them through a support vector machine (SVM). HOG counts the occurrences of gradient orientation on fragments of the picture. The method can be helpful for finding shapes in the picture. We support MTCNN [41] where a CNN obtains candidate windows, filters out the false positive candidates, and performs a facial landmark detection. The DNN [3] face detector in OpenCV is a Caffe model which is based on the Single Shot-Multibox Detector (SSD) and uses ResNet-10 architecture as a backbone. DNN is faster, has more detections and is more accurate overall. Agarwal [1] compared the performance of Dlib, Haar, and MTCNN, concluding that Dlib and MTCNN perform much better for face detection in terms of accuracy, but use more time than Haar for processing. Any one of the models can be used in our pipeline for face detection, specified via configuration parameters.

3.2.4 Priority Assignment. As shown in Figure 2, the results from the logo detection, close-up shot detection, and face detection modules are passed to a priority assignment module before being filtered. Our pipeline uses 4 priority levels:

(4) Images that are classified by the logo detection module as containing a logo are assigned to priority level 4, the lowest priority, and dropped[4]. Images that are classified by the logo detection module as not containing a logo proceed ahead.
(3) Images that are classified by the close-up shot detection module as containing a close-up shot are sorted in descending order of their probability score (i.e., image with the highest probability score comes first). Images with a score below the threshold value (default=75%) are assigned to priority level 3. Images with a score above the threshold value proceed ahead. Rankings are preserved in both cases.

[4]If all the images are classified as containing a logo, they are all assigned to the priority level 1 and passed to the image quality prediction module. In this case, the thumbnail presented as the overall result of the framework will be an image with a logo, which, despite being undesirable, is preferred over no output.

(2) Images that are classified by the face detection module as not containing any faces are assigned to priority level 2.
(1) Images that are classified as having at least one face are sorted according to the pixel size of the biggest face detected on the image in descending order, and assigned to priority level 1.

Note that increasing the threshold for the probability score from the close-up shot detection module can increase the adherence to the established thumbnail selection rules (more rigid enforcement of rule 2 from Table 1) and decrease latency (as the face detection module will take shorter if fewer images are passed on from the close-up shot detector), but also carries the risk of omitting potentially viable thumbnail candidates by assigning more images to the lower priority levels. Another trade-off is related to the final sorting of images which have been assigned to priority level 1, as sorting by face size vs. sorting by close-up probability emphasize different preferences. We propose that this choice be made depending on the accuracy of the models used in the face detection and close-up shot detection modules. For instance, in Section 4, we sort the images assigned to level 1 by face size in descending order, since the face detection model Dlib has better accuracy than the close-up shot detection model Surma, especially on bigger faces.

3.3 Step 3: Image Quality Analysis (IQA)

The assignment of images to priority levels in the second step gives us a sorted list of thumbnail candidates. Then, one by one from the top, each candidate goes through the third step. The iteration for selecting a thumbnail candidate starts at the highest priority level, and follows the image order prescribed in the previous step. As mentioned above, we prefer sorting images by the size of the largest face detected in them at this level. If there are no images assigned to the higher priority level, the iteration skips to the next priority level. During the iteration process, an image quality predictor [25] will be run on each image. It is supposed to predict the quality of an image by calculating its blur and distortion. If the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) score from the quality predictor is below a threshold, the image is chosen as the output thumbnail. The image quality prediction module is only ran until the first image to satisfy this condition, and this image will become the final thumbnail. For instance, in Figure 2, the sample image in the fourth category would be omitted due to having a logo, and not passed to the third step. The sample image in the first category would not be omitted, and has a high chance of being selected as the final thumbnail in the third step.

4 EXPERIMENTS AND RESULTS

4.1 Datasets

In this work, we make use of 2 in-house video datasets generated from the Norwegian and Swedish men's elite soccer leagues[5], focusing on goal events. Table 2 presents the distribution of samples in our datasets into training, validation, and test splits. Table 3 presents the corresponding image resolutions per dataset, for the results presented in this section[6].

Dataset	Class	Train.	Val.	Test	Total
E1	Logo	561	240	223	1024
	No logo	4281	1355	1388	7024
E2	Close-up	564	109	29	702
	No close-up	772	158	38	968
E3	Face (various)	-	-	56	56
	Face (audience)	-	-	32	32
	No face (various)	-	-	33	33
A1	Logo	-	-	69	69
	No logo	-	-	111	111

Table 2: Distribution of samples per dataset: training, validation, and test splits.

Dataset	Original	Training	Testing
E1	960×540	200×120	200×120
E2	960×540	960×540	960×540
E3	960×540	-	960×540
A1	1280×720	-	1280×720

Table 3: Image resolution per dataset: original, used in training, and used in testing / thumbnail selection.

The **Eliteserien** dataset [36] consists of 300 clips of goal events scored in the Norwegian Eliteserien. Most of these clips start 25 seconds before the goal event and end 50 seconds after the event. **Eliteserien for logo detection (E1):** For logo detection, we use the annotations made in [36] (logo images from the Norwegian Eliteserien), and augment these with logos from the English Premier League [7]. Overall, this dataset contains 1,024 images classified as logos and 7,024 images classified as background. **Eliteserien for close-up shot detection (E2):** For close-up shot detection, we create an image dataset containing frames that are extracted from the Eliteserien dataset. This dataset consists of 2 classes, "close-up shot" and "no-close-up shot". Images classified as close-up shots often contain players and managers, where others include wide views of the pitch, spectators, etc. Shots of spectators celebrating are often not as close as those of players celebrating. Blurriness or image noise are not taken into account when considering which class an image belongs to. Overall, this dataset contains 702 images classified as close-up shots and 968 images classified as no-close-up shots. **Eliteserien for face detection (E3):** For face detection, we create an image dataset containing frames extracted from the Eliteserien dataset, consisting of 2 parts: "Various" and "Audience". The "Various" part contains 56 images with one or more faces, and 33 images without any faces. The images not containing faces are either long distance shots, meaning the faces are small, or have heads turned away from the camera. The second part of the dataset, "Audience", is comprised of images showing the audience. It has a total of 32 images. The distance from the camera to the audience varies, but the images mostly contain faces that are smaller than the faces appearing in the "Various" dataset. The second part of the dataset is generated for the purpose of determining if a model is good at detecting small faces.

The **Allsvenskan** dataset consists of goal clips from the Swedish Allsvenskan. This dataset is similar to the Eliteserien, but there

[5]The reason we create our own datasets is that each module in our pipeline requires different ground truth labels (e.g., close-up shot vs. long distance shot for one module, logo vs. no-logo for another). Therefore, multiple datasets, or a large dataset with multiple ground truth dimensions is needed. However, there is no publicly available dataset we can use for all modules in our pipeline.

[6]As our pipeline is configurable, different image resolutions can be used for training and testing (different from the original assets and/or different between training and testing). The influence of resolution ("scale") is further investigated in Tables 6 and 8.

is more variation in video clip length. Most of the clips start approximately 25 seconds before the goal event and end about 60-80 seconds after the goal event. There are 233 clips in the dataset.
Allsvenskan for logo detection (A1): For testing the logo detection module, we annotate a number of video clips from the Allsvenskan dataset. Overall, this dataset contains 180 images, where 69 images contain a logo and 111 images do not contain a logo. We do not use this dataset for training, but only for testing how the logo detection module performs on completely unseen data (see Section 4.3).

4.2 Implementation Details

Our automatic thumbnail selection pipeline was implemented using Python `v3.9.7`, Tensorflow `v2.6.0`, Keras `v2.6.0`, cv2 `v4.5.3`, Dlib `v19.22.1`, mtcnn `v0.1.0` and Imquality `v1.2.7` on a DGX-2 server. This server is well suited for heavy computational and memory operations. However, it is not necessary to use a machine that is exceptionally suited for heavy operations to run the pipeline. The image classifier model by Surma [32] which is used for logo detection and close-up shot detection was trained on the same server, with the same versions of Keras and Tensorflow along with Matplotlib `v3.4.3`, Livelossplot `v0.5.4` and Efficientnet `v1.1.1`. The codebase for our pipeline is publicly accessible as an open-source software repository under[7], and further artifact details are provided in the reproducibility Appendix. A demonstration of our pipeline is presented in [12].

The datasets for logo detection and close-up shot detection are split into training, validation and test sets. The distribution of the samples are presented in Table 2. To evaluate the performance of the classifier-based modules in our pipeline (logo detection, close-up shot detection, face detection), we use accuracy, precision, and recall as metrics. For the image quality prediction module, we use the BRISQUE score, which is a score indicating image quality. A smaller score indicates better perceptual quality [22].

4.3 Logo Detection Performance

We train the Surma [32] model on the E1 dataset. The results are displayed in Table 4, where the first row corresponds to a threshold value of 0.5 (default value). The model achieves an accuracy of 0.994 on the E1 test split. Upon inspection, we see that most of the "no-logo" images have a probability score below 0.01, indicating that the model strongly (and correctly) identifies that the image does not contain a logo, most of the time. The precision is higher than the recall, indicating that there are more false negatives than false positives. To make the model have higher recall and maybe higher overall accuracy, it is possible to consider adjusting the threshold value. It would be reasonable to assume that the accuracy of the model would be improved by lowering the threshold value. For instance, an image with over 0.1 probability is likely to contain a logo, so we test how the threshold value influences the accuracy, using 0.1 and 0.05 in addition to the default value of 0.5. The results for a threshold value of 0.1 are given in the second row of Table 4. Even though the threshold value is changed by a large percentage from 0.5 to 0.1, the accuracy of the model is not affected considerably. Precision is lower, as the number of false positives increases, but the recall is improved. We argue that having high recall is more important than having high precision, as classifying images which do not actually contain a logo to contain a logo (false positive) causes the pipeline to be more conservative, and possibly drop more thumbnail candidates, rather than cause it to allow for rule 4 in Table 1 to be broken. The third row of Table 4 displays the results for an even lower threshold value, namely 0.05. The recall remains the same, but the precision is worse. We therefore conclude that adjusting the threshold value further than 0.1 is not necessary for better accuracy.

Dataset	Threshold	Accuracy	Precision	Recall
Eliteserien (E1)	0.5	0.994	0.986	0.969
	0.1	0.995	0.969	0.996
	0.05	0.993	0.952	0.996
Allsvenskan (A1)	0.5	0.717	1.000	0.261
	0.1	0.761	1.000	0.377
	0.05	0.794	1.000	0.464

Table 4: Performance of the logo detection module using the model by Surma [32], on the test splits from the Eliteserien (E1) and Allsvenskan (A1) datasets containing 1611 **and** 180 **images, respectively.**

Overall, from the first part of Table 4, we see that the logo detection module performs adequately well on the Eliteserien dataset (E1 test split), as it has been trained on a part of this dataset (E1 training split). However, for this module to be used as part of an automatic thumbnail selection pipeline which is generalizable across different datasets (and correspondingly, different soccer leagues), we would like to see if it performs adequately for different datasets as well. For this purpose, we use the unseen Allsvenskan dataset (A1 test split). The results are displayed in the second part of Table 4. On the Allsvenskan dataset, the model is able to detect fewer than half of the logos. The images not containing a logo are all predicted correctly, irrespective of the threshold value. The accuracy of the model for the Allsvenskan set is not as high as for the Eliteserien dataset, where it was 0.99. However, it did not predict any false positives. We observe that a change in the threshold value from 0.5 to 0.1 improves performance for the Allsvenskan dataset as well. The precision is 100% for both cases, but the recall is improved for 0.1 as the threshold value. As we are interested in improving the recall, we further test with 0.05 as for the Eliteserien dataset, and see that there actually is an improvement when the threshold is lowered further. However, since this dataset is smaller than the Eliteserien dataset, we have lower confidence in this conclusion and resolve to use a threshold value of 0.1 in the final pipeline.

4.4 Close-up Shot Detection Performance

The close-up shot detection module is trained on the E2 dataset consisting of "close-up shot" and "no close-up shot" classes. The binary image classification by Surma [32] originally yields an accuracy of 0.82 on the test split when the probability threshold is set to 0.5. For the close-up shot detection task, we would like to prioritize precision over recall, because there is a relatively high chance that false positives can end up being the final thumbnail. Increasing the precision will prevent false positives from being prioritized. We test with different threshold values to observe the changes in the performance of the close-up shot detection module, the results are

[7] https://github.com/simula/host-ats

displayed in Table 5. Based on the trade-off between precision and accuracy, we resolve to use a threshold of 0.75 in the final pipeline.

Dataset	Threshold	Accuracy	Precision	Recall
Eliteserien (E2)	0.8	0.761	1.000	0.448
	0.75	0.866	0.955	0.724
	0.7	0.881	0.920	0.793
	0.6	0.836	0.750	0.931
	0.5	0.821	0.730	0.931

Table 5: Performance of the close-up shot detection module using the model by Surma [32], on the test split from the Eliteserien (E2) dataset containing 67 images.

4.5 Face Detection Performance

The face detection module can be run with different face detection models. In this section, we consider Dlib [15], MTCNN [41], and Haar cascade [39]. According to Agarwal [1], Haar cascade is faster but less accurate, where Dlib is slower but more accurate. To evaluate the performance of the module, we use precision achieved on the E3 dataset, using the original versions of images ("Full" scale) and 50% down-scaled[8] versions ("Half" scale). Table 6 presents the results in terms of precision and processing time per image.

Dlib: From the "Various" set of 89 images, the Dlib model detected 55 faces in total when the images were full-scaled. The Dlib model had 53 face detections on the same images when half-scaled. There were a few false positive face detections on the "Various" set, where 2 false detections were made on full-scaled and 1 on half-scaled. Both false detections were on a face-shaped skeleton logo. On the "Audience" set, the Dlib had a total of 153 detections on full-scaled and only 2 on half-scaled. A false positive face detection appeared here as well that was a logo with a face of a dog. The trend for the Dlib when doing face detection on half-scaled images was that it could not detect smaller faces. When half-scaling the images, the faces are covering half the amount of pixels compared to the original. This makes the Dlib overlook all the smaller faces.

MTCNN: Overall, MTCNN has higher recall than the other models for all classes on the dataset. The precision on the "Various" dataset is better than Haar cascade, but not as good as Dlib. The model is the best at detecting smaller faces, as it has the highest recall on the "Audience" set. It is also able to detect faces from the side (at a 90 degree angle). However, the mean processing time per image for MTCNN is 0.96*s* on half-scale images and 1.53*s* on full-scale images. This is about 19 times slower than Haar cascade and 3 times slower than Dlib.

Haar cascade: On the "Audience" set, Haar cascade has comparable precision to the other models, and higher recall than Dlib. But, it has significantly lower precision on the "Various" dataset. It detects more faces than Dlib, but not as much as MTCNN. The model is prone to false positives: on full-scaled images, 39% of the detections are false.

Final pipeline: After evaluating different models, we try to make a decision for the model to use in our final pipeline. In the context of automatic thumbnail selection for soccer video clips, we prioritize high precision and speed. The reason we prioritize precision over recall for the face detection task is because faces appear often and redundantly in soccer videos, so it is not essential that our model manages to detect *all* faces. However, it is essential that the model be certain *when* it does detect a face, because there is a higher probability that this image will be selected as the final thumbnail, according to the prioritization rules shown in Figure 2. Regarding speed, we see that a clear benefit of down-scaling is the reduction in processing time. For 50% down-scaling, up to 0.2*s* can be saved per image. For instance, if the pipeline is running face detection on 7 frames from a single video clip, 50% down-scaling decreases the time required by the face detection module using the Dlib model from 3.9*s* to 2.5*s*.

Model	Image Scale	Precision (Various)	Precision (Audience)	Time per Image
Dlib	Full	0.96	0.99	0.55*s*
	Half	0.98	0.50	0.36*s*
MTCNN	Full	0.85	0.94	1.53*s*
	Half	0.91	0.93	0.96*s*
Haar	Full	0.61	0.91	0.08*s*
	Half	0.74	0.94	0.05*s*

Table 6: Performance of the face detection module using different models (Dlib, MTCNN, and Haar cascade) and different image scales (full size 960×540 and half size 480×270), on the test splits from the "Various" and "Audience" datasets.

MTCNN has the highest recall and the second highest precision on the "Various" set. It can also detect smaller faces, and faces from a side angle unlike the other models. However, MTCNN uses significantly more time than the other models, and the absolute time it takes for going through, e.g., 7 images per video clip is not feasible for our context. Haar cascade has performed well on detecting small faces, and is the fastest model (can process a video 7 times faster than the Dlib for instance). However, it tends to have many false positives. Dlib has the highest precision on the "Various" set and it is faster while processing down-scaled images. The disadvantage is that it cannot detect smaller faces, but we consider the ability to detect bigger faces to be more important.

We therefore resolve to use the Dlib model as the default model in the face detection module in our pipeline, along with 50% down-scaling of the video frames. The MTCNN model is added as a configurable (non-default) option. As the Haar cascade is very fast compared to Dlib and MTCNN, it is also added as a configurable option. If the automatic thumbnail selector is intended to process a large number of video clips in near-real-time, this model might be more suitable, as it allows for prioritizing speed over precision.

4.6 Image Quality Prediction

The last step in our pipeline comprises image quality analysis on the thumbnail candidates, in the order that the images appear after priority assignment in the previous step, followed by filtering and final thumbnail selection. We run the model by Ocampo [25] for image quality prediction, which is based on model proposed by Jongyoo [14]. The model outputs a BRISQUE score, where the lower the score, the better the predicted quality of the image. The image quality predictor is tested on 65 frames from a single video clip from the Eliteserien dataset.

Influence of compression: While testing the model, we also investigate if the model predicts lower quality when the input image has been saved with lower quality using the imwrite function

[8] In this context, down-scaling refers to the reduction in the resolution of (and correspondingly, number of pixels in) the images, and not the encoding quality.

from cv2. With the imwrite function, it is possible to choose the percentage of how much data from the image is going to be preserved. From Table 7, it is possible to see the accuracy of the model for predicting lower quality on actual lower quality images. For the test set of 65 images, it did not achieve 95% accuracy before the chosen image quality was about 50%. Another relevant aspect is how fast the image quality predictor model processes the images when they contain less data. The model by Ocampo has a tendency to consume a lot of time for processing a single image. From Table 7 it is possible to see the mean processing time per image with respect to preserved quality. Preserving less data on an image does not seem to make the image faster to process for the image quality predictor model, even though the storage size is reduced to a great extent.

Preserved Quality	Accuracy	Mean Processing Time	Mean Size
100%	NA	2.95*s*	128 KB
70%	0.72	2.82*s*	51 KB
60%	0.89	2.72*s*	42 KB
50%	0.95	2.65*s*	35 KB
40%	0.97	2.65*s*	32 KB
5%	1.00	2.28*s*	13 KB

Table 7: Influence of compression: accuracy of the Ocampo model [25] in predicting if the image has lower quality, along with mean processing time per image and mean storage size per image, with respect to preserved image quality.

Influence of down-scaling: What seems to reduce the processing time is reducing the scale of the image. The intermediary conclusion is that processing time is dependent on the amount of pixels and not necessarily the amount of data the image contains. We have tested what the image quality predictor [25] predicts for different degrees of down-scaling (Table 8). This is to know how the model behaves when it receives a down-scaled image, as we want the processing of an image to be fast in our pipeline. However, even though the processing might be faster, we want to make sure that the model is able to predict on down-scaled images with similar performance compared to the images in original scale. The output thumbnail of our pipeline can be the original or down-scaled version of the image. In the latter case, the BRISQUE score used in the final step will be referring to the down-scaled version of the image.

Image Scale	Accuracy	Mean Processing Time	Mean Size
960×540	NA	2.95*s*	128 KB
800×450	0.88	1.63*s*	42 KB
600×338	0.86	1.16*s*	27 KB
500×281	0.84	0.73*s*	21 KB
300×169	0.92	0.34*s*	9 KB
100×56	0.85	0.06*s*	1.9 KB

Table 8: Influence of down-scaling: accuracy of the Ocampo model [25] in predicting if the down-scaled image has better quality than the original image with 960×540 resolution, along with mean processing time per image and mean storage size per image, with respect to image scale.

Assessing the image quality predictor, many of the images receiving a high score (low quality) are often images that are blurry, noisy and even with generic graphics. Ranking the images with lower scores does not seem necessary as an image with 10 in score can look as good as an image with 30 in score to the human eye. The difference is not significant enough as we want to take the runtime into account as well. Eliminating the undesired images altogether seems to be the most helpful course of action with this model.

4.7 Complexity Analysis

In order to evaluate the complexity of our pipeline, we refer to 3 metrics. The **computational requirements** referring to hardware and software requirements for the complete end-to-end execution of the pipeline were already discussed in Section 4.2. The **model size** refers to the size of the ML models we use in terms of storage. Table 9 presents respective sizes of various models of the pipeline on disk (in terms of storage). Finally, the **execution time** refers to the duration it takes for certain components in the pipeline, or the pipeline itself, to perform their operations. For example, the time it takes for the pipeline to select a thumbnail after receiving a video as input is an aspect to consider when considering the efficiency of the pipeline. There have been certain decisions made during the implementation of the pipeline rendering it more time-efficient, but less thorough in search for a good thumbnail. Using the image quality predictor BRISQUE [25] takes about 1 second per image. If we want to receive an image quality score for all frames from the video clip, it could take more time. However, it is ideal if the complete pipeline does not use more than a few seconds to select a thumbnail for a given video clip. Table 11 presents the average execution time per video clip for each module in the pipeline, as well as the overall pipeline in an end-to-end fashion. Here, we use 37 clips from the Allsvenskan dataset, with an average duration of 77 seconds. As a benchmark for execution time, we run the state-of the-art thumbnail selector called Hecate from Song et al. [31] that has an average end-to-end execution time of 3.4*s*. Running our proposed pipeline on the same video clips uses minimum 3.8*s* as shown in Table 11.

Module	Model	Size
Logo detection	Surma [32]	10530KB
Close-up shot detection	Surma [32]	10530KB
Face detection	Dlib [15]	7365KB
	MTCNN [41]	2256KB
	Haar [39]	930KB
Image quality prediction	Ocampo [25]	147KB

Table 9: Complexity analysis: model size on disk (all models trained on the respective Eliteserien dataset).

4.8 Comparison with a State-of-the-Art Model

In order to benchmark the actual thumbnail selections by our pipeline, we present a visual comparison with the state-of-the-art thumbnail selector Hecate from Song et al. [31] in Table 10 for 4 sample video clips[9]. Hecate does not prioritize close-up shots of people, or images that may be more temporally-relevant (e.g., frames closer to a certain event annotation). Therefore, although the selected images might score well on objective metrics such as blur and darkness, they do not appear relevant, i.e., the model does not necessarily capture the semantics of a particular event. For example, Hecate often selects far away shots with no indication of

[9]Functional code could not be obtained for similar generic thumbnail selectors Vasudevan et al. [37] from https://github.com/arunbalajeev/query-video-summary and Agrawal from https://github.com/sumeetag/thumbnails-for-videos.

the goal event, compared to the close-up shots of the ball inside the goal or players cheering as selected by our proposed pipeline.

Table 10: Comparison with the state-of-the art thumbnail selector Hecate from Song et al. [31].

5 SUBJECTIVE EVALUATION

5.1 User Study Framework

`Huldra` is an open-source framework for collecting crowdsourced feedback on multimedia assets [10]. This framework allows for the collection of participant responses in a storage bucket hosted on the cloud, from where they can be retrieved in real-time by survey organizers, using credentials, immediately after the first interaction of each participant. As part of our automatic thumbnail selection system, we customize the Huldra framework and call it HOST-ATS. Figure 3 presents screenshots of the main pages of HOST-ATS.

The survey begins with the home page (Figure 3a) which allows users who already have a universally unique identifier (UUID) to complete their responses if they have closed the browser involuntarily, or decided to continue later. In both cases, their information remains saved in the browser's local storage. However, if the participant does not have a UUID, they must complete a registration form (Figure 3b) where we ask for participant information regarding age, gender, video editing experience, and soccer fandom (mandatory), as well as a free form text field if the participant has other relevant comments to add (optional). Details regarding the registration page input fields are given in [11]. After successful login or registration, the user is redirected to the background page (Figure 3c) which introduces the context of the study and presents directions for use with a simple figure. The core of the framework lies in the ranking of multimedia assets. This functionality is provided by the case page

Figure 3: Screenshots from the HOST-ATS user study.

(Figure 3d) which is composed of 3 vertical columns as follows: The left column presents a sample video clip showing a goal event. We use the npm package React player [6] for playback, which offers an off-the-shelf component for playing a variety of multimedia. The middle column presents two alternative thumbnails for the video clip. Both of which could be viewed larger if needed. The user ranks the alternatives simply by clicking on one of the thumbnails. Once a thumbnail is clicked, it is displayed immediately on the top in the right column (Figure 3e).

In order to have complete and consistent responses for our study, we make it mandatory to rank a case before proceeding to the next. Users can later go backwards and revisit their answers or change them. After finishing the ranking, the participants are invited to fill out a feedback form (Figure 3f) about the aspects that they deem important in a thumbnail. They can mark from a list of alternatives, as well as suggest other facets. We also give them the option to provide additional comments and feedback in a text field input.

We use the HOST-ATS framework to run 2 user study iterations, the first including 9 cases, and the second including 13 cases. In each case, participants compare 2 alternative thumbnail selection methods. Table 12 presents the investigation aspects relevant to the alternative thumbnails for each case. The selection of cases allows us to compare among different thumbnail selection methods, as well as to gain deeper insights into viewer expectations from thumbnails, which will potentially help us improve our framework (e.g., rules and assumptions described in Section 3 and Table 1) as future work.

5.2 User Study Findings

Figures 4 and 5 present the overall results from the iterations of our user study, in terms of the number of votes for different options per case, where "HOST-ATS" indicates the thumbnail selection by

Step 1: Pre-processing		Step 2: Content Analysis and Priority Assignment					Step 3: IQA	Overall
# frames	Frame extraction	Loading models	Logo detection	Close-up shot detection	Face detection		Quality prediction	Total
50	1.767*s*	0.064*s*	0.249*s*	0.223*s*	Dlib	2.669*s*	1.156*s*	6.221*s*
					Haar	0.288*s*		3.805*s*
100	1.935*s*	0.064*s*	0.423*s*	0.387*s*	Dlib	5.383*s*	1.429*s*	9.449*s*
					Haar	0.572*s*		4.917*s*

Table 11: Complexity analysis: average execution time per clip for each module in the framework, for 37 video clips from the Allsvenskan dataset (average video clip duration: 77*s*). All frames are 50% down-scaled. "Loading time" refers to the loading of the logo detection and close-up shot detection models.

Table 12: Investigation aspects for cases in the user study. Evaluated thumbnail selection alternatives are HOST-ATS (H), manual selection (M), and static selection (S).

Case	Investigation	Method		
		H	M	S
70396	Players celebrating: close-up vs. medium distance	✓	✓	
75861	Player close-up vs. player celebration	✓	✓	
76465	Players celebrating: scorer visible vs. not visible	✓	✓	
80540	Player close-up vs. player celebration	✓	✓	
73375	Player close-up vs. players celebrating	✓		✓
77648	Players celebrating: scorer visible vs. not visible	✓		✓
77651	Players celebrating vs. audience	✓		✓
79994	Players celebrating vs. goal shot	✓		✓
80077	Players celebrating vs. attack action	✓		✓
75143	Player close-up vs. players celebrating	✓		✓
76358	Player close-up vs. attack action	✓		✓
76394	Long distance vs. close-up shot	✓		✓
76407	Player close-up vs. player celebration	✓		✓
76422	Player close-up vs. long distance shot	✓		✓
76821	Player close-up vs. audience	✓		✓
78438	Players celebrating vs. audience	✓		✓
78888	Player close-up vs. players celebrating	✓		✓
79285	Players celebrating vs. long distance shot	✓		✓
79588	Player close-up: players from different teams	✓		✓
79606	Player close-up: front view vs. back view	✓		✓
79789	Player close-up vs. players celebrating	✓		✓
80136	Long distance shot vs. audience	✓		✓

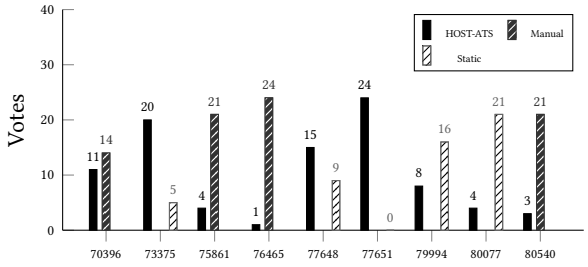


Figure 4: Case answers for the first user study.

our framework, "Static" indicates the thumbnail selection method wherein a frame from the video clip at a fixed timestamp (e.g., 5 seconds after playback start) is assigned as the thumbnail, and "Manual" indicates manual thumbnail selection undertaken by human operators. The study received 27 responses for the first iteration, and 21 responses for the second iteration, with a total of 42 responses remaining after filtering. We make the following general observations.

Manual selection is preferred over HOST-ATS (4 out of 4 pairwise comparisons), i.e., a human operator can select better thumbnails than our automated system. For case `70396`, it is a close call, where the medium distance shot of player celebration is preferred over the close-up shot of player celebration. For case `75861`, manual selection showing multiple players is preferred by a larger margin against a single player close-up. For case `76465`, manual selection where the scorer is visible in front view is preferred by a larger margin against a celebration where the scorer is only visible in back view. For case `80540`, manual selection showing multiple players in celebration from the back view is preferred by a larger margin against a close-up view of players where a face is visible more distinctly but there is no celebration action. These are very valuable insights for us to improve our thumbnail selection rules (Table 1) with more details regarding the joint evaluation of face detection and close-up shot detection results, as well as including context-aware priorities with respect to celebrations and action content (e.g., crowded celebration scene preferred over frontal or side view of fewer players with less action content).

HOST-ATS outperforms static selection in most cases (11 out of 18 pairwise comparisons). For cases `73375`, `77648`, `77651`, `76407`, `76422`, `76821`, `78438`, `78888`, `79285`, `79588`, and `79606`, we see that the rules enforcing higher image quality, preference of close-up shots to long distance shots, and frontal face view are in line with viewer preferences. Cases where the static selection is preferred include the following: `79994` (clear view of the goal shot, despite being medium/long distance preferred over blurry image of players celebrating), `80077` (attack scene preferred over celebration scene with partial obstruction in image), `75143` (goal scorer's single celebration, despite blur in image, preferred over multiple-player celebration where scorer is not visible), `76358` (attack action from medium distance preferred over close-up shot of player), `76394` (long distance shot of field preferred over close-up shot with partial obstruction in image), `79789` (goal scorer from back view preferred over multiple-player celebration with slight blur), and `80136` (long distance shot of the field preferred over blurry audience celebration). These cases show that additional rule adjustments are needed to take image blurriness, partial obstructions, action content, and celebration context (e.g., goal scorer) into account.

Viewers consider high image quality, player faces, and action content as the most important aspects of a thumbnail, followed by close-up shots, cheering, and the absence of logo transitions (Figure 6). These confirm our initial thumbnail selection rules, and give further insights regarding possible additional rules (e.g., detection of action content and cheering context).

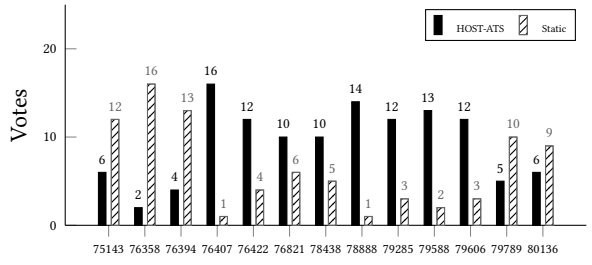


Figure 5: Case answers for the second user study.

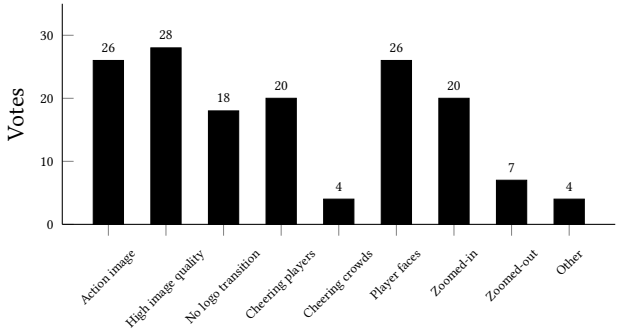


Figure 6: Feedback form answers for question F1 [11] (first and second user study combined).

6 DISCUSSION

6.1 Blur Detection as a Framework Component

There can be a lot of motion in soccer games, resulting in video frames being blurry. When selecting a frame from a video clip as a thumbnail, it is important to avoid images that are too blurry, both for aesthetic reasons and also for keeping the information content and representativity of the thumbnail high. It should be possible to see what is happening in the image, at least in the foreground.

Running blur detection on frames from soccer videos has its challenges. Firstly, blur detection models do not necessarily capture the essence of what is really perceived as blur by humans, within a certain context. There might be objects or regions in a frame that are more important to the human eye than others. For instance, a frame where the celebrating soccer player is clear but the background is blurry might be discarded falsely by a blur detection model due to supposed high presence of blur, whereas the frame could actually be considered non-blurry, in terms of the object of interest in the scene. Secondly, blur scores for frames should be considered relatively to other frames from the same video and not in absolute terms. Overall image quality can vary from video to video, which leads to a different viewer tolerance for what is perceived as a blurry frame for a given video clip.

We integrate and experiment with alternative blur detection models in our automatic thumbnail selection pipeline. One of these is the Laplacian operator from OpenCV Library [3]. Figure 7 presents the blur scores according to [3] of different frames from the same video. In Figure 7d, we can see a close-up shot of a player which is clear, but the background is blurry. This frame could have served as a perfectly adequate thumbnail, but scores a high presence of blur, relative to the rest of the frames tested. In contrast, the frames in Figures 7a and 7b) score a low presence of blur. These frames have a clear background but the players in the foreground are a bit blurry as they are in motion. This discrepancy demonstrates the first challenge mentioned above, namely that the blur detection model predicts higher blur the larger the total area covered by blur is, irrespective of regions of interest or layout. Another example for this challenge is given by Figures 7e and 7f. These frames are from the same second and the same camera. The player appearing in Figure 7e is less blurry than the players appearing in Figure 7f. However, the background is more clear in the latter compared to the former. Yet another challenge is to identify why seemingly similar scenes can receive different blur scores, such as Figure 7c receiving a score which indicates much more blur than Figures 7a and 7b.

By using a blur detection module in our pipeline, we would like to promote frames that look clearer, and consequently improve the thumbnail selection process. However, the frames that existing blur detection models predict as being very clear might not be what we prefer to have as thumbnails. It seems that the direct use of existing blur detection models might not be the right approach for capturing what viewers actually perceive as a degradation within the context of soccer highlight clips. Therefore, the blur detection module is an optional component in our pipeline (which can be configured to use either of 2 alternative models, or not activated at all), and a topic for further investigation.

6.2 Limitations

Our proposed automatic thumbnail selection system has a number of limitations. One of the limitations of our approach is that the thumbnails selected by our pipeline can be very similar to each other, especially for similar video clips, due to our fixed ruleset. For instance, if player scenes are prioritized over audience scenes in the ruleset, such frames will be more likely to be selected for any given video clip. Along with our insights from the user studies described in Section 5, this motivates our current work concerning a finer, more elaborate ruleset. We are currently working on moving away from a plain decision tree based approach towards a more complex approach based on custom scores for each candidate frame, considering more diverse aspects (action content, face angle, context such as celebration, etc.). Despite its good performance, our pipeline is limited in the sense that its sole focus is on goal events, not considering other soccer events such as player substitutions, yellow and red cards. Our experiments are also limited in terms of the scale of the datasets we have used for training and evaluation (which are mainly composed of 2 leagues from Norway and Sweden, with a relatively low number of usable samples in the second dataset).

6.3 Possible Improvements and Future Work

There are a number of possible improvements which constitute potential future work directions:

- **Blur detection:** As shown in the previous section, blur detection is challenging to integrate directly into an automatic thumbnail selection pipeline. In order to identify blur in line with viewer preferences, foreground-background detection

and weighted analysis using a grid to identify important pixels in an image might be required.

- **Integration of subjective findings:** It is necessary to further investigate what makes a good thumbnail for viewers (in general, for soccer events, and in particular, for goals). In this respect, we have made note of the following: action content as a higher priority over face detection, blur detection together with contextual information, relaxing of the close-up shot requirement depending on content and blur, detection of partial obstructions in thumbnail candidates.
- **Model performance:** Different models can be used in the individual framework components for logo detection, close-up shot detection, face detection, and image quality prediction, and benchmarked, similar to what we have demonstrated in Section 4 for face detection. For instance, YOLO [26] is another promising object detection model that we plan to integrate and test in future versions of our pipeline.
- **Generalizability:** Cross-dataset analysis of our proposed pipeline, involving larger open soccer datasets, can yield better insights into the generalizability of our assumptions and findings.
- **Extended complexity analysis:** Tests with different video clip durations, and even full live streams can give a better idea of the real-time performance of our pipeline.
- **Different events:** Extending our automatic thumbnail selection framework to other soccer events, such as player substitutions, yellow and red cards can lead to a more practically relevant and complete system. Developing different versions focusing on other sports is also possible.
- **Alternative pipeline:** An alternative approach to our complete pipeline would be to train an ML model on a large dataset of thumbnail images manually selected by experts, and use this model directly, in a single step, on video clips. Such an idea could be preferable to our current pipeline, which needs to use multiple modules to explicitly enforce certain rules. However, the logic behind the selection would remain black box, and the underlying rules might not be possible to explicitly document and reproduce. Secondly, there exists no big-enough dataset to train such a model yet, which might need a lot of data[10]. In this respect, there is a need for open datasets shared between the industry and academia, for facilitating research in this direction.

7 CONCLUSION

In this paper, we present an automatic thumbnail selection system for soccer videos called HOST-ATS, which uses ML to deliver representative thumbnails with high relevance to the video content and high visual quality in near real-time. HOST-ATS comprises a software framework that leverages logo detection, close-up shot detection, face detection, image quality prediction, and blur detection into an automatic thumbnail selection pipeline, and a user study framework for subjective evaluation and validation. We evaluate the proposed pipeline quantitatively using various soccer datasets, as well as qualitatively, through subjective user studies.

(a) Blur score: 11.72 (b) Blur score: 34.24

(c) Blur score: 3.75 (d) Blur score: 1.82

(e) Blur score: 1.76 (f) Blur score: 28.65

Figure 7: Results using the OpenCV Laplacian operator [3] on selected frames (higher score indicates less blur).

Our work combines various independently applicable approaches in an end-to-end system, with an overarching goal and a generalized ruleset. The modular and lightweight implementation of the thumbnail selection pipeline allows for the agile integration and benchmarking of various ML methods from literature in each module. The subjective evaluation campaign using a novel survey framework for crowdsourced feedback collection yields a number of insights into viewer preferences. The components of the HOST-ATS system, namely the pipeline (as controlled by a dashboard with a GUI) and the user study framework can be demonstrated.

Our results show that an automatic end-to-end system for the selection of thumbnails based on contextual relevance and visual quality can yield highly attractive thumbnails, and can be used in conjunction with existing soccer video production pipelines which require real-time operation. However, the results also indicate that some of our initial rules can be reconsidered and adjusted, as viewers might have different preferences based on context. Nevertheless, our proposed pipeline is shown to work as intended, following the specified rules and priorities, and can run efficiently. This work is therefore a good starting point for even better future automatic thumbnail selection systems.

ACKNOWLEDGMENTS

This research was funded by the Research Council of Norway, project number 327717 (AI-producer).

[10]There might also be a need for separate datasets, or different annotations within the same dataset, for different *events* (e.g., thumbnails that experts tend to select for goal event clips might be different from those for card or substitution event clips), and different *leagues* (different jerseys, players, etc. might confuse models trained on one league and tested on another).

REFERENCES

[1] Vardan Agarwal. 2021. *Face Detection Models: Which to Use and Why?* https://towardsdatascience.com/face-detection-models-which-to-use-and-why-d263e82c302c

[2] Allsvenskan. 2022. Highlights. https://highlights.allsvenskan.se/.

[3] Gary Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[4] Chen-Yu Chen, Jia-Ching Wang, Jhing-Fa Wang, and Yu-Hen Hu. 2008. Motion Entropy Feature and Its Applications to Event-Based Segmentation of Sports Video. *EURASIP Journal on Advances in Signal Processing* 2008 (2008). https://doi.org/10.1155/2008/460913

[5] Anthony Cioppa, Adrien Deliege, Silvio Giancola, Bernard Ghanem, Marc Van Droogenbroeck, Rikke Gade, and Thomas B. Moeslund. 2020. A Context-Aware Loss Function for Action Spotting in Soccer Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/CVPR42600.2020.01314

[6] Pete Cook. 2021. react-player. https://www.npmjs.com/package/react-player.

[7] Adrien Deliège, Anthony Cioppa, Silvio Giancola, Meisam J. Seikavandi, Jacob V. Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B. Moeslund, and Marc Van Droogenbroeck. 2020. SoccerNet-v2 : A Dataset and Benchmarks for Holistic Understanding of Broadcast Soccer Videos. arXiv:2011.13367 [cs.CV]

[8] Eliteserien. 2022. Highlights. https://highlights.eliteserien.no/.

[9] FIFA.com. 2018. More than half the world watched record-breaking 2018 World Cup. https://www.fifa.com/worldcup/news/more-than-half-the-world-watched-record-breaking-2018-world-cup

[10] Malek Hammou, Cise Midoglu, Steven A. Hicks, Andrea Storås, Saeed Shafiee Sabet, Inga Strümke, Michael A. Riegler, and Pål Halvorsen. 2022. Huldra: A Framework for Collecting Crowdsourced Feedback on Multimedia Assets. In *13th ACM Multimedia Systems Conference (MMSys '22), June 14–17, 2022, Athlone, Ireland*. ACM, New York, NY, USA. https://doi.org/10.1145/3524273.3532887

[11] Andreas Husa. 2022. *Automated Thumbnail Selection for Soccer Videos with Machine Learning*. Master's thesis. University of Oslo, Oslo, Norway.

[12] Andreas Husa, Cise Midoglu, Malek Hammou, Pål Halvorsen, and Michael A. Riegler. 2022. HOST-ATS: Automatic Thumbnail Selection with Dashboard-Controlled ML Pipeline and Dynamic User Survey. In *13th ACM Multimedia Systems Conference (MMSys '22), June 14–17, 2022, Athlone, Ireland*. ACM, New York, NY, USA. https://doi.org/10.1145/3524273.3532908

[13] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-Scale Video Classification with Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1725–1732. https://doi.org/10.1109/CVPR.2014.223

[14] Jongyoo Kim, Anh-Duc Nguyen, and Sanghoon Lee. 2019. Deep CNN-Based Blind Image Quality Predictor. *IEEE Transactions on Neural Networks and Learning Systems* 30, 1 (2019), 11–24. https://doi.org/10.1109/TNNLS.2018.2829819

[15] Davis King. 2021. dlib C++ Library. http://dlib.net/. Last accessed 2022-01-24.

[16] Ryan Knott. 2021. *What Are Video Thumbnails and Why Do They Matter?* https://www.techsmith.com/blog/what-are-video-thumbnails/

[17] Jacek Komorowski, Grzegorz Kurzejamski, and Grzegorz Sarwas. 2019. FootAndBall: Integrated player and ball detector. *CoRR* abs/1912.05445 (2019). arXiv:1912.05445 http://arxiv.org/abs/1912.05445

[18] Harilaos Koumaras, Georgios Gardikis, George Xilouris, Evangelos Pallis, and Anastasios Kourtis. 2006. Shot boundary detection without threshold parameters. *J. Electronic Imaging* 15 (4 2006), 020503. https://doi.org/10.1117/1.2199878

[19] Thomas J Law. 2021. The Perfect YouTube Thumbnail Size and Best Practices. https://www.oberlo.com/blog/youtube-thumbnail-size.

[20] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. 2019. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.

[21] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. 2018. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In *Proceedings of the European Conference Computer Vision (ECCV)*.

[22] MATLAB. 2021. *brisque (R2021a)*. https://se.mathworks.com/help/images/ref/brisque.html

[23] Pier Luigi Mazzeo, Marco Leo, Paolo Spagnolo, and Massimiliano Nitti. 2012. Soccer Ball Detection by Comparing Different Feature Extraction Methodologies. *Advances in Artificial Intelligence* 2012 (2012), 12. https://doi.org/10.1155/2012/512159

[24] Olav Andre Nergård Rongved, Markus Stige, Steven Alexander Hicks, Vajira Lasantha Thambawita, Cise Midoglu, Evi Zouganeli, Dag Johansen, Michael Alexander Riegler, and Pål Halvorsen. 2021. Automated Event Detection and Classification in Soccer: The Potential of Using Multiple Modalities. *Machine Learning and Knowledge Extraction* 3, 4 (2021), 1030–1054. https://doi.org/10.3390/make3040051

[25] Ricardo Ocampo. 2021. *Deep CNN-Based Blind Image Quality Predictor in Python*. https://towardsdatascience.com/deep-image-quality-assessment-with-tensorflow-2-0-69ed8c32f195

[26] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2015. You Only Look Once: Unified, Real-Time Object Detection. *CoRR* abs/1506.02640 (2015). arXiv:1506.02640 http://arxiv.org/abs/1506.02640

[27] Olav A. Nergård Rongved, Steven A. Hicks, Vajira Thambawita, Håkon K. Stensland, Evi Zouganeli, Dag Johansen, Cise Midoglu, Michael A. Riegler, and Pål Halvorsen. 2021. Using 3D Convolutional Neural Networks for Real-time Detection of Soccer Events. *International Journal of Semantic Computing* 15, 02 (2021), 161–187. https://doi.org/10.1142/S1793351X2140002X

[28] Olav A. Nergård Rongved, Steven A. Hicks, Vajira Thambawita, Håkon K. Stensland, Evi Zouganeli, Dag Johansen, Michael A. Riegler, and Pål Halvorsen. 2020. Real-Time Detection of Events in Soccer Videos using 3D Convolutional Neural Networks. In *Proceedings of the IEEE International Symposium on Multimedia (ISM)*. 135–144. https://doi.org/10.1109/ISM.2020.00030

[29] Adrian Rosebrock. 2021. *OpenCV Haar Cascades*. https://www.pyimagesearch.com/2021/04/12/opencv-haar-cascades/

[30] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. 568–576.

[31] Yale Song, Miriam Redi, Jordi Vallmitjana, and Alejandro Jaimes. 2016. To Click or Not To Click: Automatic Selection of Beautiful Thumbnails from Videos. arXiv:1609.01388 [cs.MM]

[32] Greg Surma. 2018. *Image Classifier - Cats vs Dogs*. https://gsurma.medium.com/image-classifier-cats-vs-dogs-with-convolutional-neural-networks-cnns-and-google-colabs-4e9af21ae7a8

[33] Dian Tjondronegoro, Yi-Ping Phoebe Chen, and Binh Pham. 2003. Sports video summarization using highlights and play-breaks. In *Proceedings of ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR)*. 201–208. https://doi.org/10.1145/973264.973296

[34] Torrens University Australia. 2020. Why the Sports Industry is Booming in 2020 (and which key players are driving growth). https://www.torrens.edu.au/blog/why-sports-industry-is-booming-in-2020-which-key-players-driving-growth

[35] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6450–6459. https://doi.org/10.1109/CVPR.2018.00675

[36] Joakim Olav Valand, Haris Kadragic, Steven Alexander Hicks, Vajira Lasantha Thambawita, Cise Midoglu, Tomas Kupka, Dag Johansen, Michael Alexander Riegler, and Pål Halvorsen. 2021. AI-Based Video Clipping of Soccer Events. *Machine Learning and Knowledge Extraction* 3, 4 (2021), 990–1008. https://doi.org/10.3390/make3040049

[37] Arun Balajee Vasudevan, Michael Gygli, Anna Volokitin, and Luc Van Gool. 2017. Query-adaptive Video Summarization via Quality-aware Relevance Estimation. arXiv:1705.00581 [cs.CV]

[38] Vimeo Livestream Blog. 2022. Streaming Stats - 47 Must-Know Live Video Streaming Statistics. https://livestream.com/blog/62-must-know-stats-live-video-streaming.

[39] P. Viola and M. Jones. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. I–I. https://doi.org/10.1109/CVPR.2001.990517

[40] Hossam M. Zawbaa, Nashwa El-Bendary, Aboul Ella Hassanien, and Ajith Abraham. 2011. SVM-based soccer video summarization system. In *Proceedings of the World Congress on Nature and Biologically Inspired Computing*. 7–11. https://doi.org/10.1109/NaBIC.2011.6089409

[41] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *CoRR* abs/1604.02878 (2016). arXiv:1604.02878 http://arxiv.org/abs/1604.02878

[42] Matko Šarić, Dujmić Hrvoje, and Baričević Domagoj. 2008. Shot Boundary Detection in Soccer Video using Twin-comparison Algorithm and Dominant Color Region. *Journal of Information and Organizational Sciences* 32 (06 2008).