HOST-ATS: Automatic Thumbnail Selection with Dashboard-Controlled ML Pipeline and Dynamic User Survey

Andreas Husa
SimulaMet, Norway

Cise Midoglu
SimulaMet, Norway

Malek Hammou
SimulaMet, Norway

Pål Halvorsen[*][†]
SimulaMet, Norway

Michael A. Riegler[‡]
SimulaMet, Norway

ABSTRACT

We present HOST-ATS, a holistic system for the automatic selection and evaluation of soccer video thumbnails, which is composed of a dashboard-controlled machine learning (ML) pipeline, and a dynamic user survey. The ML pipeline uses logo detection, close-up shot detection, face detection, image quality prediction, and blur detection to automatically select thumbnails from soccer videos in near real-time, and can be configured via a graphical user interface. The web-based dynamic user survey can be employed to qualitatively evaluate the thumbnails selected by the pipeline. The survey is fully configurable and easy to update via continuous integration, allowing for the dynamic aggregation of participant responses to different sets of multimedia assets. We demonstrate the configuration and execution of the ML pipeline via the custom dashboard, and the agile (re-)deployment of the user survey via Firebase and Heroku cloud service integrations, where the audience can interact with configuration parameter updates in real-time. Our experience with HOST-ATS shows that an automatic thumbnail selection system can yield highly attractive highlight clips, and can be used in conjunction with existing soccer broadcast practices in real-time.

CCS CONCEPTS

• **Computing methodologies** → **Video summarization**; Machine learning; • **Human-centered computing** → **Empirical studies in interaction design**; **User studies**; • **General and reference** → **Empirical studies**.

KEYWORDS

crowdsourcing; blur detection; dashboard; deep learning; graphical user interface; image quality; logo detection; object detection; shot boundary detection; soccer; survey; thumbnail generation; user study; video

MMSys '22, June 14–17, 2022, Athlone, Ireland
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9283-9/22/06.
https://doi.org/10.1145/3524273.3532908

ACM Reference Format:
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, 7 pages. https://doi.org/10.1145/3524273.3532908

1 INTRODUCTION

Sports broadcasting is immensely popular, and the interest in viewing videos from sports games grows day by day. Consequently, the amount of worldwide content, such as video footage and audio commentaries, is enormous and rapidly growing. The huge availability of content and the number of games makes it increasingly important to design systems for extracting highlights in real- or near real-time. As a large percent of audiences prefer to view only the main events in a game, the generation of highlight clips and video summaries is of tremendous interest for broadcasters.

For an end-to-end soccer video production system capable of delivering game highlights, existing work in the areas of event detection [6, 10, 17, 18, 20, 23, 25, 30] and event clipping [5, 29, 31, 34] can be complemented by a thumbnail selection operation. A thumbnail is an image representing a video. Thumbnails can be used in galleries where various highlight clips are presented, and serve as the first impression meant to attract people to view a clip[1]. Thumbnails need to be selected carefully to be eye-catching and to properly represent the event in the highlight clip. However, the thumbnail selection operation is time-consuming and expensive as there are many frames in a video to select among, and image quality is often not considered extensively due to time limitations and costs. Therefore, automating the thumbnail selection process has the potential to both save resources and improve quality.

We present HOST-ATS, a holistic system for the automatic selection and evaluation of soccer video thumbnails. HOST-ATS is composed of: (1) a dashboard-controlled Machine Learning (ML) pipeline, and (2) a dynamic user survey. The pipeline [9] uses ML to automatically select thumbnails for soccer videos in near real-time, and can be configured via a Graphical User Interface (GUI). It combines logo detection, close-up shot detection, face detection, image quality prediction, and blur detection. HOST-ATS also includes a dynamic user survey for evaluating the results of the pipeline qualitatively (through user studies). This web-based survey is fully configurable and easy to update via Continuous Integration (CI), allowing for the dynamic aggregation of participant responses to different sets of multimedia assets.

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

[1] Example highlight galleries from the Norwegian Eliteserien: https://highlights.eliteserien.no/ and Swedish Allsvenskan: https://highlights.allsvenskan.se/

We demonstrate the configuration and execution of the HOST-ATS pipeline via the custom dashboard, and the agile (re-)deployment of the HOST-ATS user survey via Firebase[2] and Heroku[3] cloud service integrations. The audience can interact with configuration parameter updates in real-time. Overall, our experience with HOST-ATS shows that an automatic thumbnail selection system can yield highly attractive highlight clips, and can be used in conjunction with existing soccer broadcast practices in real-time.

2 BACKGROUND

Automated soccer video production systems can cover research fields such as object detection [14, 19, 22], shot boundary detection [15, 36], event detection and classification [6, 10, 17, 18, 20, 23, 25, 30], and event clipping [5, 29, 31, 34]. In this context, thumbnail selection refers to the task of finding an appropriate thumbnail image for event highlight clips.

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 [16, 26]. There is not much work on thumbnail selection specifically for sports videos, but there are a number of studies that target thumbnail selection in general. Song et al. [26] 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 model has been trained by a set of images that have been annotated with subjective aesthetic scores. Vasudevan et al. [32] 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"). Our analysis of such models has shown that generic approaches might not necessarily work well for soccer highlight clips [9].

3 HOST-ATS DASHBOARD-CONTROLLED ML PIPELINE

The first component of HOST-ATS is an ML-based pipeline which identifies appropriate images to be used as thumbnails, by checking for logos, scene boundaries, faces, and analyzing image quality. This pipeline selects attractive thumbnails by considering visual quality and aesthetic metrics along with relevance to video content, thus making the resulting thumbnails more representative of the video.

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 [13, 16, 26]. In this regard, we centered our pipeline around 3 key principles, namely relevance, content, and image-quality.

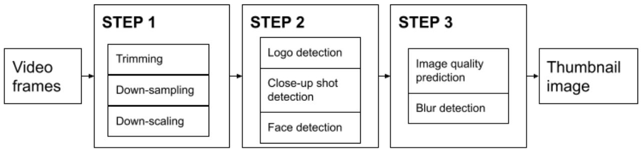


Figure 1: HOST-ATS automatic thumbnail selection pipeline.

- **Relevance:** The thumbnail that our pipeline selects is a frame from the video it is supposed to represent. Along with ensuring relevance, this bypasses the time it would take to find external images, such as graphics.
- **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 image if the thumbnail displays a long-distance shot of the soccer field. Therefore, our pipeline prioritizes close-up shots. Close-up shots are usually frames showing the soccer players, spectators, and managers. There could also be frames from the replay of the goal 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 individually 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, our pipeline undertakes image quality prediction as the final filter.

The end-to-end execution of our pipeline consists of 3 steps: (1) pre-processing, (2) content analysis and priority assignment, and (3) image quality analysis. Figure 1 presents these steps and the corresponding components in our framework. A video clip (sequence of video frames) is fed as input to the framework, and the final output is an image, which is a frame from the video, as the suggested thumbnail.

3.1 Step 1. Pre-processing

In this step, the input video is trimmed with respect to the start-end of the clip or an event annotation, it is down-sampled to extract a certain number of frames (default=50[4]) and frames are optionally down-scaled in terms of resolution. Each of the remaining frames are candidates for being the final thumbnail, and are fed into the next step. The left column in Figure 2 presents the configuration parameters related to this step in the dashboard GUI.

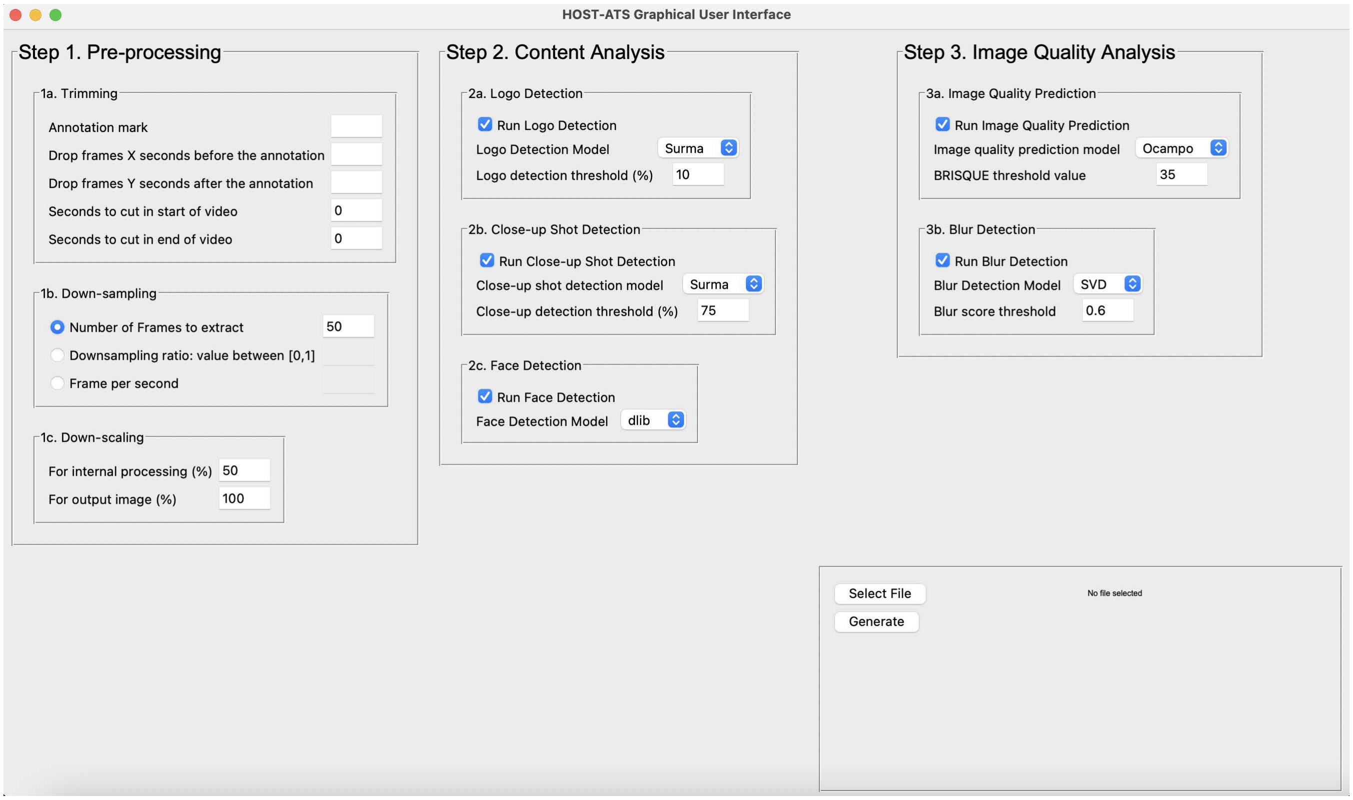
3.2 Step 2. Content Analysis

In this step, the contents of the candidate frames are analyzed. For this purpose, independent modules for logo detection, close-up shot

[2] https://firebase.google.com/

[3] https://heroku.com/

[4] The default number of frames might seem for a full soccer game, i.e., 90 minute video, but the pipeline is optimized for shorter highlight clips, which are traditionally no longer than 90 seconds. The number of frames is a configurable parameter.



HOST-ATS Graphical User Interface

Step 1. Pre-processing

1a. Trimming

Annotation mark	
Drop frames X seconds before the annotation	
Drop frames Y seconds after the annotation	
Seconds to cut in start of video	0
Seconds to cut in end of video	0

1b. Down-sampling

- ● Number of Frames to extract: 50
- ○ Downsampling ratio: value between [0,1]
- ○ Frame per second

1c. Down-scaling

For internal processing (%)	50
For output image (%)	100

Step 2. Content Analysis

2a. Logo Detection

- ☑ Run Logo Detection
- Logo Detection Model: Surma
- Logo detection threshold (%): 10

2b. Close-up Shot Detection

- ☑ Run Close-up Shot Detection
- Close-up shot detection model: Surma
- Close-up detection threshold (%): 75

2c. Face Detection

- ☑ Run Face Detection
- Face Detection Model: dlib

Step 3. Image Quality Analysis

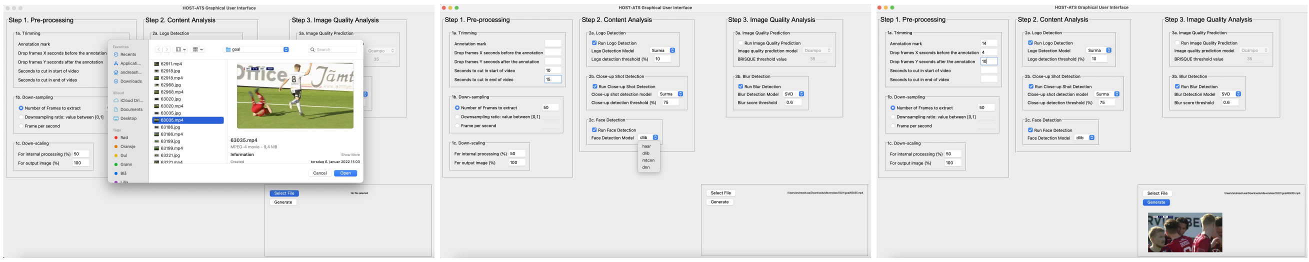
3a. Image Quality Prediction

- ☑ Run Image Quality Prediction
- Image quality prediction model: Ocampo
- BRISQUE threshold value: 35

3b. Blur Detection

- ☑ Run Blur Detection
- Blur Detection Model: SVD
- Blur score threshold: 0.6

Figure 2: HOST-ATS dashboard.



(a) Input video clip selection

(b) Model configuration

(c) Output

Figure 3: HOST-ATS dashboard details.

detection, and face detection are used, after which priorities are assigned to each frame. The middle column in Figure 2 presents the configuration parameters related to this step in the dashboard GUI.

Logo Detection. The logo detection module is used to detect logos that appear in the video frames. When a logo appears in a soccer video, it usually indicates that a graphic is about to appear. These are frames we would like to eliminate from the thumbnail selection. For logo detection, the pipeline can use a Convolutional Neural Network (CNN) based on the architecture presented by Surma [28]. This is a general image classification model that can train and 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. The closer the number is to 1 or 0, the more certain the model is of the input being in the predicted class. Alternatively, the pipeline can use the model by Ocampo [21] for logo detection, which is based on model proposed by Jongyoo [11]. The logo detection model and the threshold value are configurable parameters in the dashboard GUI as seen in Figure 3b.

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, i.e., medium or long-distance shot), or a close-up (zoomed-in) shot. Images classified as “close-up shot”s are prioritized in thumbnail selection. The image classification model by Surma [28] can be used in this module. The model certainty threshold value is a configurable parameter in the dashboard GUI. The images with a close-up shot probability below the set threshold are not ignored, as in the case of logo detection, but receive a lower priority.

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 the context of our work, 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 can be more complex and less accurate.

Our pipeline integrates 4 alternative models for face detection. Haar cascade [33] is an object detection model which is fast, but which tends to be prone to false positive detection, compared to other models [24]. 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 [12] uses features extracted by Histogram of Oriented Gradients (HOG), and passes them through a Support Vector Machine (SVM). Histogram of oriented gradients (HOG) counts the occurrences of gradient orientation on fragments of the image. The method can be helpful for finding shapes in an image. MTCNN [35] where a CNN obtains candidate windows, filters out the false positive candidates, and performs a facial landmark detection. The DNN [4] face detector in OpenCV is a Caffe model which is based on the Single Shot-Multibox Detector (SSD) and uses ResNet-10 architecture as its backbone. DNN is faster, has more detection and is more accurate. Agarwal [3] 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. All models can be used in our automatic thumbnail selection pipeline as alternative face detectors which can be selected via configuration parameters. Other models, such as You Only Look Once (YOLO) by [22] can also be integrated in the future, to address the challenges associated with existing models such as sensitivity to facial orientation.

Priority Assignment. The results from the logo detection, close-up shot detection, and face detection modules are passed through priority assignment before filtering. The reasons for the priority levels are to be able to use several metrics concurrently in the evaluation of candidate images, and to control how greedy we would like our framework to be in enforcing our thumbnail selection rules. Figure 4 presents the decision tree used for determining the priority levels.

3.3 Step 3. Image Quality Analysis (IQA)

The iteration for selecting a thumbnail candidate starts at the highest priority level, and follows the image order prescribed in the previous step. The pipeline sorts 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.

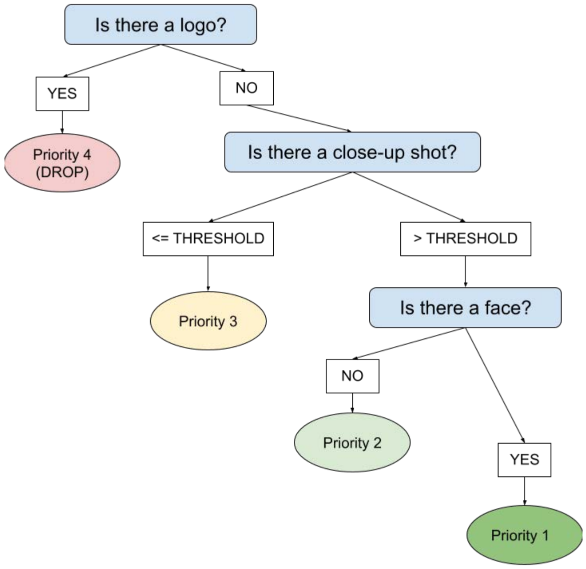


Figure 4: HOST-ATS priority assignment decision tree.

During the iteration process, an image quality predictor [21] can 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 above the threshold, the image will not be chosen as the output thumbnail. The image quality prediction module is followed by a dedicated blur detection module, which integrates 2 alternative models. Singular Value Decomposition (SVD) is inspired by code from Su [27], and calculated using Numpy. It has a default threshold value of 0.60. The Laplacian using OpenCV highlights regions of an image containing rapid intensity changes. Too low thresholds lead to the incorrect marking of images as blurry when they are not, and too high thresholds lead to the marking of images that are actually blurry as not blurry. The image quality prediction module is only ran until the first image to satisfy this condition, and the image will become the final thumbnail.

3.4 Implementation Details and Demonstration

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 operations and heavy memory operations. However, it is not necessary to use a machine that is exceptionally suited for heavy operations to run the thumbnail generator. The image classifier model by Surma [28] which is used for logo detection and close-up shot detection was trained on the same server as well. For training the image classifier, the same versions of Keras and Tensorflow were used, as well as Matplotlib `v3.4.3`, Livelossplot `v0.5.4` and Efficientnet `v1.1.1`. The logo detection and close-up shot detection models used by the framework are saved in Hierarchical Data Format (HDF) format, and the Haar cascade face detection model is saved in Extensible Markup Language (XML) format. These models and the complete codebase for HOST-ATS are publicly accessible as an open-source software repository under[5].

The dashboard GUI consists of a window with all the interactions in one page. Figure 2 shows the page displayed by the GUI. All configuration parameters in the pipeline are possible to modify from this page. None of the parameters in steps 1, 2 or 3, which are presented as subcategories in Figure 2, are mandatory. If no input is provided for a specific parameter, the default values of the pipeline will be run. The "Generate" button executes the pipeline, and the output image is displayed under the button as seen in the Figure. Sample videos presenting the dashboard layout and consecutive executions with different configuration parameters can be found under[6] and[7]. A deeper analysis of the influence of configuration parameters, providing insights on how practitioners can customize the framework for best results, can be found in [9].

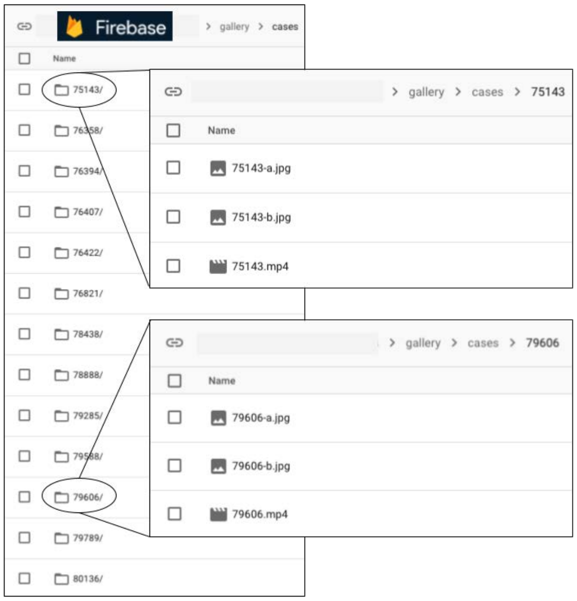
4 HOST-ATS DYNAMIC USER SURVEY

In order to evaluate the performance of the pipeline in terms of end-user perception, HOST-ATS provides a dynamic survey which can be used to conduct user studies. For instance, each video clip for which a thumbnail is selected can be presented as a "case", where survey participants are asked to compare the thumbnail selected by our pipeline, and a thumbnail selected by other methods (e.g., static frame selection method used by the industry today).

4.1 Survey Framework

`Huldra` [8] is a framework for collecting crowdsourced feedback on multimedia assets. 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. The framework is completely configurable through the use of a single configuration file, which allows for the deployment of custom surveys in a very short amount of time.

Huldra uses 3 third-party integrations. **Google Cloud Platform (GCP)** is a suite of cloud computing services that runs on Google infrastructure [1]. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Huldra uses a GCP S3 bucket for cloud storage, both for storing the multimedia assets and the participant responses. Access to the bucket is maintained through Firebase integration, where credentials can be specified as environment variables in the Heroku configuration (configuration of access to the storage bucket can also be undertaken using the `config.json` file in the codebase, but this is not advisable due to privacy reasons, unless a public bucket is used on purpose). **Firebase** is a platform for creating mobile and web applications [2]. Used in connection with the GCP S3 bucket, Firebase allows Huldra to fetch the multimedia assets to be used in the study seamlessly from, and write participant responses to, configurable locations in this bucket. Huldra uses **Heroku** as a cloud Platform as a Service (PaaS) for web deployment, which supports the triggering of automatic deployments from a GitHub repository. Applications deployed on the web via Heroku can be accessed via a URL of the form: `https://<application-name>.herokuapp.com`. All of the Huldra third-party integration functionalities can be used with free plans as well as paid plans: minimum requirements are a free Google account for GCP and Firebase, and free hobby dynos for a Heroku personal application.



Firebase > gallery > cases

□ Name
□ 75143/
□ 76358/
□ 76394/
□ 76407/
□ 76422/
□ 76821/
□ 78438/
□ 78888/
□ 79285/
□ 79588/
□ 79606/
□ 79789/
□ 80136/

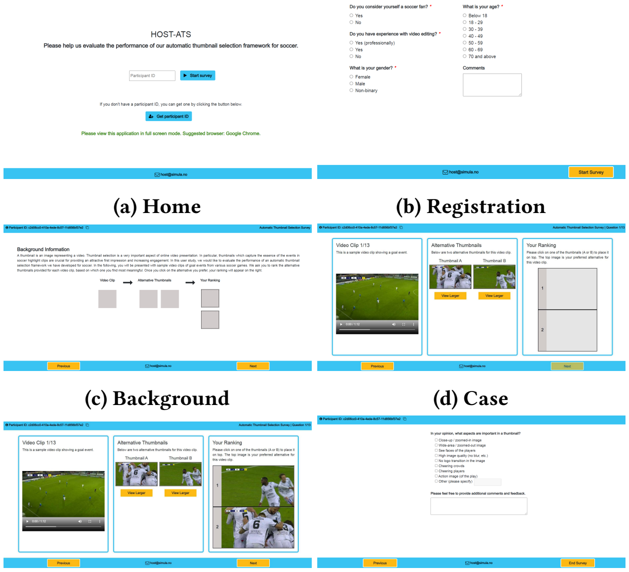
> gallery > cases > 75143

□ Name
□ 75143-a.jpg
□ 75143-b.jpg
□ 75143.mp4

> gallery > cases > 79606

□ Name
□ 79606-a.jpg
□ 79606-b.jpg
□ 79606.mp4

Figure 5: Firebase integration and folder structure.



HOST-ATS
Please help us evaluate the performance of our automatic thumbnail selection framework for soccer.

Participant ID ▶ Start survey

If you don't have a participant ID, you can get one by clicking the button below.

Get participant ID

Please view this application in full screen mode. Suggested browser: Google Chrome.

host@simula.no

(a) Home

Do you consider yourself a soccer fan? *
○ Yes
○ No

Do you have experience with video editing? *
○ Yes (professionally)
○ Yes
○ No

What is your gender? *
○ Female
○ Male
○ Non-binary

What is your age? *
○ Below 18
○ 18 - 29
○ 30 - 39
○ 40 - 49
○ 50 - 59
○ 60 - 69
○ 70 and above

Comments

host@simula.no Start Survey

(b) Registration

Participant ID: c2d06cc0-410a-4ede-8c57-11d856bf37e2 Automatic Thumbnail Selection Survey

Background Information
A thumbnail is an image representing a video. Thumbnail selection is a very important aspect of online video presentation. In particular, thumbnails which capture the essence of the events in soccer highlight clips are crucial for providing an attractive first impression and increasing engagement. In this user study, we would like to evaluate the performance of an automatic thumbnail selection framework we have developed for soccer. In the following, you will be presented with sample video clips of goal events from various soccer games. We ask you to rank the alternative thumbnails provided for each video clip, based on which one you find most meaningful. Once you click on the alternative you prefer, your ranking will appear on the right.

Video Clip → Alternative Thumbnails → Your Ranking

Previous host@simula.no Next

(c) Background

Participant ID: c2d06cc0-410a-4ede-8c57-11d856bf37e2 Automatic Thumbnail Selection Survey | Question 1/13

Video Clip 1/13
This is a sample video clip showing a goal event.

Alternative Thumbnails
Below are two alternative thumbnails for this video clip.
Thumbnail A Thumbnail B
View Larger View Larger

Your Ranking
Please click on one of the thumbnails (A or B) to place it on top. The top image is your preferred alternative for this video clip.
1
2

Previous host@simula.no Next

(d) Case

Participant ID: c2d06cc0-410a-4ede-8c57-11d856bf37e2 Automatic Thumbnail Selection Survey | Question 1/13

Video Clip 1/13
This is a sample video clip showing a goal event.

Alternative Thumbnails
Below are two alternative thumbnails for this video clip.
Thumbnail A Thumbnail B
View Larger View Larger

Your Ranking
Please click on one of the thumbnails (A or B) to place it on top. The top image is your preferred alternative for this video clip.
1
2

Previous host@simula.no Next

(e) Case (ranked)

Participant ID: c2d06cc0-410a-4ede-8c57-11d856bf37e2

In your opinion, what aspects are important in a thumbnail?
□ Close-up / zoomed-in image
□ Wide-area / zoomed-out image
□ See faces of the players
□ High image quality (no blur, etc.)
□ No logo transition in the image
□ Cheering crowds
□ Cheering players
□ Action image (of the play)
□ Other (please specify)

Please feel free to provide additional comments and feedback.

Previous host@simula.no End Survey

(f) Feedback

Figure 6: Screenshots of the HOST-ATS user study.

[5] HOST-ATS repository: https://github.com/simula/host-ats

[6] HOST-ATS Dashboard-Controlled ML Pipeline - Part 1/2: https://youtu.be/HHMCdMucorI

[7] HOST-ATS Dashboard-Controlled ML Pipeline - Part 2/2: https://youtu.be/VZQaEy2VauQ

For HOST-ATS, we customize the Huldra framework, specify the multimedia assets in Firebase as presented in Figure 5, and use custom participant registration and feedback form questions. A running instance can be found under[8]. Figure 6 presents screenshots of the main pages. The survey begins with the home page (Figure 6a) 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 6b) 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). After successful login or registration, the user is redirected to the background page (Figure 6c) which introduces the context of the study and shows directions for use with a simple figure.

The core of the framework is the ranking of multimedia assets which can be easily updated using the Firebase integration. This functionality is provided by the case page (Figure 6d) which is composed of 3 vertical columns. The left column presents a sample video clip showing a goal event. HOST-ATS uses the npm package React player [7] 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 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 6e). In order to have complete and consistent responses for our study, it is 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 6f) about the aspects that they deem important in a thumbnail. They can mark from a list of alternatives, as well as suggest other facets. They also have the option to add additional comments and feedback in a text field input.

User studies conducted with HOST-ATS allow for the comparison of different thumbnail selection methods, as well as provide deeper insights into viewer expectations from thumbnails, which can potentially help improve the selection mechanisms of the ML pipeline itself (e.g., rules and assumptions which motivate the use of various modules) as future work. [9] presents a preliminary analysis of the qualitative results obtained from the survey responses of 42 participants.

4.2 Deployment and Demonstration

Assuming that the third-party service accounts are already established, the below steps can be used to run a HOST-ATS instance. The instance is automatically updated via CI after every push to the repository branch connected to the Heroku instance. It can also be updated (without re-deployment) by updating the multimedia assets in the Firebase bucket, allowing for multiple versions of the survey to be run without any code changes. More information on system setup and reproducibility aspects can be found in [9].

1. **Assets:** Set up the necessary folder structure in the S3 bucket, prepare and upload the multimedia assets corresponding to your desired "case"s.
2. **Codebase:** Clone the Huldra repository.
3. **Configuration:** Update configuration parameters in the `config.json` file as needed, to customize your instance.

4a. **Deployment (Heroku):** Enter the Firebase connection parameters in your app's Heroku configuration. Deploy the relevant branch of your repository. The survey will be accessible at `https://<app-name>.herokuapp.com` by default.

4b. **Deployment (local):** Enter the Firebase connection parameters in your local environment variables. Run `npm install` and `npm start` to start your local server. The survey will be accessible at `localhost:3000` by default.

5 **Outputs:** Retrieve participant response files from the bucket.

5 CONCLUSION

Thumbnail selection is a very important aspect of online video presentation. Thumbnails are needed to capture the essence of a video clip and provide a good first impression to viewers, making clips more attractive to watch. Traditional solutions for soccer highlight clips, which display important events such as goals and cards, rely on the static or manual selection of thumbnails. However, static approaches can result in the selection of sub-optimal video frames as snapshots, which degrades the overall quality of the clip as perceived by the viewers, consequently decreasing viewership, and manual approaches are expensive.

In this demonstration, we present an automatic thumbnail selection system for soccer called HOST-ATS, which comprises a ML pipeline to deliver representative thumbnails with high relevance to the video content and high visual quality in near real-time, and a dynamic user survey for evaluating the selected thumbnails qualitatively. The HOST-ATS ML pipeline leverages logo detection, close-up shot detection, face detection, image quality prediction, and blur detection. The HOST-ATS user survey allows for the evaluation of this pipeline through subjective user studies. Our demonstration will show that an automatic system for the selection of thumbnails based on contextual relevance and visual quality, complemented with a tool for validation studies, can yield highly attractive highlight thumbnails, and can be used in conjunction with existing soccer broadcast practices. Automating the traditionally complex and labor-intensive task of thumbnail selection can reduce production costs. It will also enable the research community to conduct further studies with minimal programming knowledge and time investment, studies which provide insights into various factors influencing the perception of soccer video clips by viewers. We hope that our holistic system for automatic thumbnail selection and evaluation can facilitate both academic and practical applications in the domain of multimedia content generation and presentation. The HOST-ATS system is applicable to various other sports broadcasts, such as skiing, handball, or ice hockey, and presents a viable potential to impact future sports productions.

ACKNOWLEDGMENTS

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

[8] https://host-ats.herokuapp.com

REFERENCES

[1] 2022. Cloud Computing Services - Google Cloud. https://cloud.google.com/.

[2] 2022. Firebase. https://firebase.google.com/.

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

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

[5] 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

[6] Anthony Cioppa, Adrien Deliège, Silvio Giancola, Bernard Ghanem, Marc Van Droogenbroeck, Rikke Gade, and Thomas B. Moeslund. 2019. A Context-Aware Loss Function for Action Spotting in Soccer Videos. *CoRR* abs/1912.01326 (2019). arXiv:1912.01326 http://arxiv.org/abs/1912.01326

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

[8] Malek Hammou, Cise Midoglu, Steven Alexander Hicks, Andrea Storås, Saeed Shafiee Sabet, Inga Strümke, Michael Alexander Riegler, and Pål Halvorsen. 2022. Huldra: A Framework for Collecting Crowdsourced Feedback on Multimedia Assets. In *Proceedings of the ACM Multimedia Systems Conference (MMSys)*.

[9] 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. https://doi.org/10.1145/3524273.3528182

[10] 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

[11] 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

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

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

[14] 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

[15] 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

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

[17] 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)*.

[18] 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)*.

[19] 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

[20] 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

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

[22] 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

[23] 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

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

[25] 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.

[26] 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]

[27] Bolan Su, Shijian Lu, and Chew Lim Tan. 2011. Blurred Image Region Detection and Classification. 1397–1400. https://doi.org/10.1145/2072298.2072024

[28] 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

[29] 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

[30] 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

[31] 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

[32] 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]

[33] 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)*. https://doi.org/10.1109/CVPR.2001.990517

[34] 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

[35] 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

[36] 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).