Project Assignment

In order to finish this course, you need to complete an implementation project that will cover many topics discussed in this course. It can be done in groups of 4 people. Your project must be fully functional (I should be able to test your system) and its performance will be tested by you in a live evaluation competition at the end of the semester, where all teams compete against each other and against a baseline system. As a prerequisite to submit the project, you will also have to perform a design of the architecture and the GUI, and to carefully evaluate a selected content analysis method for a single video.

Content-Based Video Retrieval System

The main goal of the assignment is to develop a *content-based video retrieval system* that can be used for finding small video segments of interest (with a duration of a few seconds) in a moderately large dataset of 100 videos. More specifically, we will focus on the use case of the *Video Browser Showdown* (VBS) and *Known-Item Search* queries (KIS) – see Course Introduction and https://www.videobrowsershowdown.org.

V3C Dataset

We will use a small part of the V3C-1 dataset (https://link.springer.com/chapter/10.1007/978-3-030-05710-7_29). More specifically, we will use a subset of 100 videos, consisting of the video files themselves (in MP4 containers, with different resolutions) and meta-data (basically upload descriptions used for those vieos by the authors). You can download the videos here:

Link: http://www.itec.aau.at/~klschoef/ivadl/V3C100.zip

ZIP password: IVADL2024V3C100:(Video&Metadata)

Visual Content Analysis (Backend)

You need to process and analyze the videos with different methods that we have learned in this course:

1. Detect shot boundaries and extract keyframes of shots. (e.g., with OpenCV)
2. Analyze the content of shots or keyframes with neural networks, e.g.:
    a. Convolutional Neural Networks
    b. Region-based CNNs
    c. Vision Transformers
    d. ...
3. Store analysis results in in an appropriate database and/or indexing system.
4. Transcode videos to small versions so that they can be used with your system for interactive playback/inspection (e.g., ffmpeg).

What methods you choose is entirely up to you, but you must do the analysis on your own (with your own scripts and your own databases). You do not need to *train* neural networks on your own,

you can use existing architectures pretrained on meaningful datasets and just use them for inference with the extracted keyframes/shots (which is fast enough on CPUs too).

Database, Index, GUI

The results of the video analysis (stored in a database and/or index) should be used by your video search system frontend with a graphical user interface (GUI) that integrate features that allow to

1. Interactively search for video content (e.g., keyframes and shots), either by a text-query, content filtering, browsing, similarity search, and/or recommendations.
2. Inspect found items and play the corresponding video content.
3. Send a selected item (e.g., keyframe) to the DRES (Distributed Retrieval Evaluation Server) for evaluation (see details below).

   Examples of sophisticated video search systems that support such features are summarized in the videos here: https://videobrowsershowdown.org/teams/vbs2024-systems/. A more detailed description of VBS systems is available in this publication: https://link.springer.com/article/10.1007/s00530-023-01143-5.

4. To test your system, we will use a live in-class evaluation session with a running instance of the Distributed Retrieval Evaluation Server (DRES) that will be used (1) to present queries to you, and (2) to collect and evaluate your submissions. We will do about 10-15 tasks and for each task you will get a maximum of 5 minutes to answer a task. The faster you find the correct item and the less mistakes you make, the more points your will get. In particular, you will get 100 points if you correctly solve a task in the first second, with a linear decrease to 50 points for correctly solving a task in the last second, and a penalty of 10 points for every wrong submission. Therefore, only make submissions to DRES if you are sure that the found segment is in the target segment presented for the task!

Communication with DRES

The DRES server is freely available on GitHub (https://github.com/dres-dev/DRES) and supports an HTTP-like API that is specified via OpenAPI (https://github.com/dres-dev/DRES/blob/master/doc/oas-client.json)[1]. In order to communicate with DRES the first action is a *login* function call with a valid user and password[2]. Next, you should query for actively running evaluation session at DRES with *client/evaluation/list* and save the evaluationId of the corresponding session (for your local instance of DRES this will just be the first one, for the instance that will be running at https://vbs.videobrowsing.org it will be “IVADL2024”). The most important API function is *submit*, which needs the evaluationId as first parameter, the task name as the second parameter, and an array of (for us just a single) *ApiClientAnswer* items that consist of a text (can be null), the media item (video ID[3]), the media collection name (“V3C100”), the start time in milliseconds, and the end time in milliseconds (can be the same as start). Please note that

[1] A more convenient view of the OpenAPI specification can also be seen here: https://editor.swagger.io/?url=https://raw.githubusercontent.com/dres-dev/DRES/master/doc/oas-client.json.

[2] An instance of DRES will be running at https://vbs.videobrowsing.org/ beginning in the second half of June 2024 (due to ongoing other international evaluations it cannot be provided earlier, unfortunately). Every group will be provided with a username and a password upon request to me by email. However, the effort for a successful communication with DRES is **not to be underestimated**, therefore it is recommended to install and test it on your own machine, already earlier.

[3] Video name without file extension.

for the latter data you need to know the frame-rate of the video (it is advisable to store it in the database).

Prerequisites, Submission, Live-Evaluation

As a prerequisite for the project implementation, you need to:

- Design your system architecture and create screen mockups of your GUI until **June 1, 2024**. The system architecture should reveal what content analysis methods you are going to use, what deep learning framework, and what data storage/database you will use. The screen mockups should demonstrate the visual structure of your search system, the available features, and the implementation technology for the GUI. Moreover, please do not forget to mention your group members!
- Create an evaluation of the performance for one of your content analysis methods (e.g., a CNN) for five semantics in one selected video in terms of *Recall* and *Precision* with *weighted averaging* (including a *confusion matrix*) until **June 30, 2024**. This can be five different content classes if you use a CNN, five object classes if you use a region-based CNN, or five different queries if you use CLIP, for example. Please note that for this evaluation you will have to annotate your selected video with the corresponding semantics.

You need to submit your implementation project with all sources (backend, frontend), data (models, keyframes, etc.), and a documentation via Moodle until the end of the semester (deadline **July 11, 2024**).

For the live evaluation competition, which we will do via Zoom, I provide two alternative dates. Each group/team must participate in exactly one of them:

July 08, 2024, at 08:15:

https://us02web.zoom.us/j/88539538334?pwd=LzZpRit5T2hJM25sUXV2VllvSXk3dz09
Meeting ID: 885 3953 8334
Passcode: 687137

July 12, 2024, at 10:00:

https://us02web.zoom.us/j/86763958641?pwd=OWhHUFYzS1dEeG9OamhpR1BaMUo3dz09
Meeting ID: 867 6395 8641
Passcode: 696563

During this Zoom session I will create tasks for the session “IVADL2024” at the DRES instance running at https://vbs.videobrowsing.org and you will use the Remote-Viewer of DRES to see when a task starts and how it looks like. You will try to solve the task as fast as possible and submit the correct segment to the DRES instance as fast as possible and DRES will score your submission.

Moreover, I will ask each group/team individually to share their screen and demonstrate me their system.

Grading of the Implementation Project

The grading for the project will be based on the following aspects:

a. 55%: Achieved score during the live evaluation competition with DRES at the end of the semester (see above) in relation your team size. Please note that you will get 0% for this part (and hence fail) if your communication with DRES does not work.
b. 35%: Sophistication of your system in relation to your team size:
    i. How good is your shot boundary detection?
    ii. Which video content analysis features do you integrate?
    iii. Which search features do you provide?
    iv. How accurate are your search results?
    v. How easy-to-use is your system?
    vi. How structured is your architecture?
c. 10%: Documentation
    i. System architecture.
    ii. Code

Reasons for Failing

Your implementation project will be graded negatively if:

- You do not meet the project prerequisites (architecture design, screen mockups, evaluation of five semantics).
- You significantly deviate from your design (architecture and screen mockups) in the actual project.
- You do not achieve at least 50% for the project implementation (see aspects above).
- Your system is not able to participate in the live evaluation competition.
- Your system is not an integrated system that allows to enter a query, then perform an actual search, present the results, and send the segment of a selected result to DRES if the user wants to.
- Your architecture, frontend, analysis, or selected set of features is too similar to another group.
- Your system is not demonstratable to me, or not testable by me.