

---

# **VHH Core Package: Automatic Video Analysis Framework (vhh\_core)**

*Release 1.0.0*

**Daniel Helm**

**Jun 08, 2020**

**CONTENTS**

<b>1</b>	<b>System Overview</b>	<b>2</b>
<b>2</b>	<b>Process Pipeline</b>	<b>3</b>
<b>3</b>	<b>Package Overview</b>	<b>4</b>
<b>4</b>	<b>Setup instructions</b>	<b>5</b>
<b>5</b>	<b>Parameter Description</b>	<b>6</b>
<b>6</b>	<b>API Description</b>	<b>7</b>
6.1	Configuration class . . . . .	7
6.2	Sbd class . . . . .	7
6.3	Stc class . . . . .	8
6.4	Video class . . . . .	8
6.5	VhhRestApi class . . . . .	9
6.6	MainController class . . . . .	10
<b>7</b>	<b>Indices and tables</b>	<b>11</b>
<b>8</b>	<b>References</b>	<b>12</b>
	<b>Index</b>	<b>13</b>

Detecting cinematographic techniques such as Shot Boundary Detection (SBD), Shot Type Classification (STC) and Camera Movements Classification (CMC) is a fundamental task for automatic film archival. Therefore, a software framework is developed to process historical films related to the time of the liberation phase of Nazi concentration camps during the Second World War [[Visual History of the Holocaust](https://www.vhh-project.eu/)<sup>1</sup> (VHH)]. This framework is able to detect and classify SBDs ([vhh\\_sbd](https://github.com/dahe-cvl/vhh_sbd)<sup>3</sup>), STCs ([vhh\\_stc](https://github.com/dahe-cvl/vhh_stc)<sup>4</sup>) and CMCs ([vhh\\_cmc](https://github.com/dahe-cvl/vhh_cmc)<sup>5</sup>) by using deep learning-based as well as optical flow-based approaches [XX][XX][XX]. This documentation is separated into the following sections: In Section *System Overview* and *Process Pipeline* an overview of the code structure as well as the automatic annotation process pipeline is demonstrated. Furthermore, the package structure and the setup and usage description is visualized in Section *Package Structure* and *Setup Instructions*. Finally, this documentation closes with an API description of all classes, modules and their members in Section *API Description*.

---

<sup>1</sup> <https://www.vhh-project.eu/>

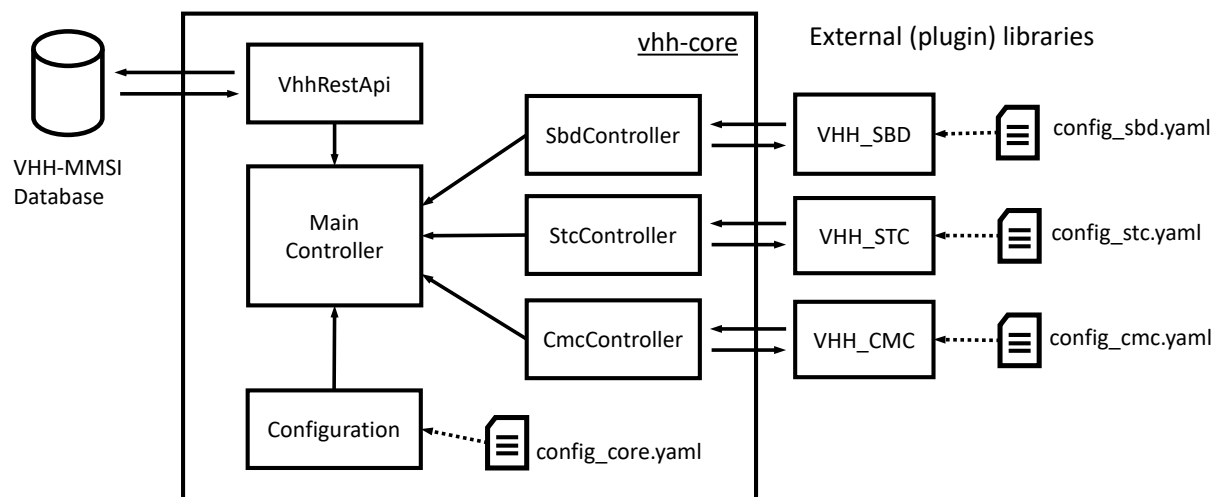
<sup>3</sup> [https://github.com/dahe-cvl/vhh\\_sbd](https://github.com/dahe-cvl/vhh_sbd)

<sup>4</sup> [https://github.com/dahe-cvl/vhh\\_stc](https://github.com/dahe-cvl/vhh_stc)

<sup>5</sup> [https://github.com/dahe-cvl/vhh\\_cmc](https://github.com/dahe-cvl/vhh_cmc)

## SYSTEM OVERVIEW

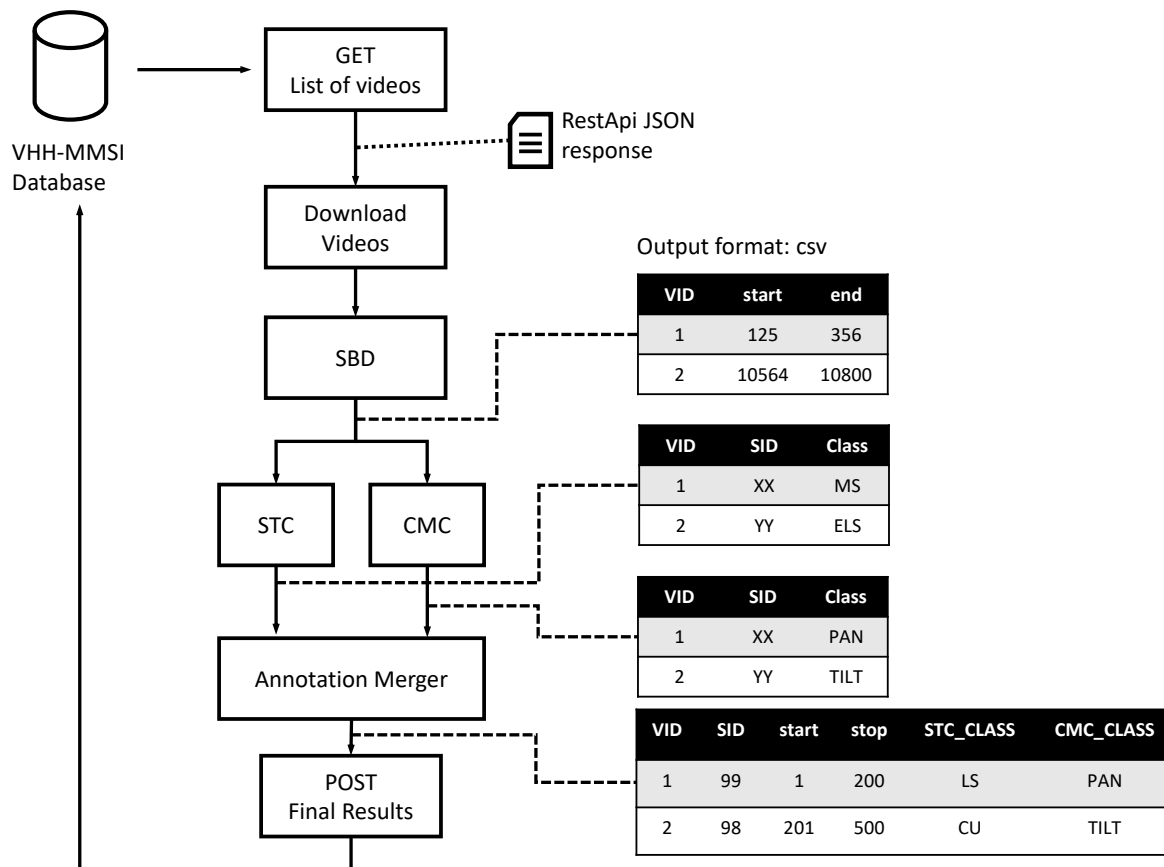
The software architecture is visualized in the below figure. The `vhh_core`<sup>2</sup> is divided in multiple classes. The core class is called *MainController* and administers the external plugin modules for automatic shot analysis in historical films: `vhh_sbd`<sup>3</sup> (SBD), `vhh_stc`<sup>4</sup> (STC) and `vhh_cmc`<sup>5</sup> (CMC). Moreover, central functionality such as the communication with the VHH-MMSI system is handled in this class. The class *VhhRestApi* includes methods to communicate with the VHH-MMSI database via RestAPI endpoints. A various number of parameters defined and set in the `config_core.yaml` file is managed by the class *Configuration*. Furthermore, each shot analysis plugin can be configured with separate configuration files stored in the `./config` folder in this repositories.



---

<sup>2</sup> [https://github.com/dahe-cvl/vhh\\_core](https://github.com/dahe-cvl/vhh_core)

## PROCESS PIPELINE



## PACKAGE OVERVIEW

The following list gives an overview of the folder structure of this python repository:

*name of repository:* vhh\_core

- **ApiSphinxDocumentation/**: includes all files to generate the documentation as well as the created documentations (html, pdf). The results are stored in the build folder (e.g. xx/build/latex/vhh\_core.pdf)
- **config/**: This folder includes the required configuration file. For each plugin (sbd, stc, cmc, ...) there is one subdirectory holding the corresponding configuration file.
- **Demo/**: This folder includes a all scripts to run this application. Furthermore, scripts to setup the folder structure (video and results storage) as well as to setup the environment for running this application.
- **Develop/**: This folder includes additional scripts used during developing phase of this application. (e.g. the document\_builder script to generate the package documentation).
- **README.md**: This file gives a brief description of this repository (e.g. link to this documentation)
- **requirements.txt**: this file holds all python dependencies and is needed to install the package in your own virtual environment
- **setup.py**: this script is needed to install the stc package in your own virtual environment

## SETUP INSTRUCTIONS

This package includes a `setup.py` script and a `requirements.txt` file which are needed to install this package for custom applications. The following instructions have to be done to use this library in your own application:

### Requirements:

- Ubuntu 18.04 LTS
- CUDA 10.1 + cuDNN
- python version 3.6.x

### Create a virtual environment:

- create a folder to a specified path (e.g. `/xxx/vhh_core/`)
- `python3 -m venv /xxx/vhh_core/`

### Activate the environment:

- `source /xxx/vhh_core/bin/activate`

### Checkout vhh\_core repository to a specified folder:

- `git clone https://github.com/dahe-cvl/vhh\_core`

### Install the stc package and all dependencies:

- change to the root directory of the repository (includes `setup.py`)
- `python setup.py install`

---

**Note:** You can check the success of the installation by using the command `pip list`. This command should give you a list with all installed python packages and it should include `vhh_core`

---

---

**Note:** Currently there is an issue in the `setup.py` script. Therefore the pytorch libraries have to be installed manually by running the following command: `pip install torch==1.5.0+cu101 torchvision==0.6.0+cu101 -f https://download.pytorch.org/whl/torch\_stable.html`

---

## PARAMETER DESCRIPTION

Documenting single YAML file.

**DEBUG\_FLAG** This parameter is used to activate or deactivate the debug mode.

**PEM\_PATH** This parameter specifies the path to the certificate file needed to provide vhh-mmsi api access.

**ROOT\_URL** This parameter specifies the root url for the RestApi endpoints.

**VIDEO\_DOWNLOAD\_PATH** This parameter specifies the video download path.

**CLEANUP\_FLAG** This parameter is used to activate/deactivate the clean-up mode. The clean-up mode deletes all downloaded videos and the corresponding generated results.

**RESULTS\_ROOT\_DIR** This parameter specifies the results path for each individual plugin.

**SBD\_CONFIG\_FILE** This parameter specifies the configuration path (.yaml) for the shot boundary detection plugin module

**STC\_CONFIG\_FILE** This parameter specifies the configuration path (.yaml) for the shot type classification plugin module

**CMC\_CONFIG\_FILE** This parameter specifies the configuration path (.yaml) for the camera movement classification module



## API DESCRIPTION

This section gives an overview of all classes and modules in *vhh\_core* as well as a brief configuration parameter description.

### 6.1 Configuration class

**class** `Configuration.Configuration` (*config\_file: str*)

Bases: `object`

This class is needed to read the configuration parameters specified in the `configuration.yaml` file. The instance of the class is holding all parameters during runtime.

---

**Note:** e.g. `./config/CORE/config.yaml`

the `yaml` file is separated in multiple sections `config['Development']` `config['Security']` `config['VhhCore']` `config['ApiEndpoints']` `config['PluginConfigs']`

whereas each section should hold related and meaningful parameters.

---

**loadConfig()**

Method to load configurables from the specified configuration file

### 6.2 Sbd class

**class** `Sbd.Sbd` (*config=None*)

Bases: `object`

This class includes the interfaces and methods to use the plugin package SBD.

**run** (*video\_instance\_list=None*)

This method is used to run the shot boundary detection task

**Parameters** `video_instance_list` – parameter must hold a list of video objects (Class-type: `Video`)

## 6.3 Stc class

**class** `Stc.Stc` (*config=None*)

Bases: `object`

This class includes the interfaces and methods to use the plugin package STC.

**run** ()

This method is used to run the shot type classification task.

## 6.4 Video class

**class** `Video.Video` (*config=None*)

Bases: `object`

This class represents a video object.

**cleanup** ()

This method is used to cleanup all data related to the corresponding video ID. It deletes the generated results of the sbd, stc and cmc plugin as well as the downloaded video file.

**create\_video** (*vid, originalFileName, url, download\_path*)

This method is used to fill all properties of a video.

### Parameters

- **vid** – This parameter must hold a valid video id.
- **originalFileName** – This parameter must hold the original filename of a video
- **url** – This parameter must hold the download url of the video.
- **download\_path** – This parameter must hold the download path in the local storage.

**download** (*rest\_api\_instance=None*)

This method is used to download the video into the local storage path.

**Parameters** **rest\_api\_instance** – This parameter must hold a valid VhhRestApi object.

**Returns** This method returns the download status (true ... successfully downloaded OR false ... download failed).

**is\_downloaded** ()

This method is used to check if a video is already downloaded.

**Returns** This method returns a boolean flag (true ... video already downloaded OR false ... video does not exist).

**printInfo** ()

This method summarizes all properties of this object and print it to the console.

## 6.5 VhhRestApi class

**class** VhhRestApi.VhhRestApi (*config=None*)

Bases: object

This class includes the interfaces and methods to use the vhh restAPI interfaces provided by MaxRecall.

**downloadVideo** (*url, file\_name, video\_format*)

This method is used to download a video from the Vhh-MMSI system.

**Parameters**

- **url** – this parameter must hold a valid restApi endpoint.
- **file\_name** – This parameter represents the filename on the local storage.
- **video\_format** – This parameter must hold a valid video format extension (e.g. m4v)

**Returns** This method returns a boolean flag which includes the state of the download process (true ... successfully finished OR false ... download failed)

**getAutomaticResults** (*vid*)

This method is used to get all automatic generated results from the VhhMMSI system.

**Parameters** **vid** – This parameter must hold a valid video identifier.

**Returns** This method returns the results (payload) as json format.

**getListofVideos** ()

This method is used to get a list of all available videos in the VHH-MMSI system.

**Returns** This method returns a list of video objects (Class-type: Video) which holds all video specific meta-data.

**getRequest** (*url*)

This method is used to send a get request to the Vhh-MMSI system.

**Parameters** **url** – this parameter must hold a valid restApi endpoint.

**Returns** This method returns the original response including header as well as payload.

**postAutomaticResults** (*vid, results\_np*)

This method is used to post the automatic generated results to the VhhMMSI system.

**Parameters**

- **vid** – This parameter must hold a valid video identifier.
- **results\_np** – This parameter must hold a numpy array including the automatic generated results.

**postRequest** (*url, data\_dict*)

This method is used to send a post request to the Vhh-MMSI system.

**Parameters**

- **url** – this parameter must hold a valid restApi endpoint.
- **data\_dict** – this parameter must hold a valid list of dictionaries with the specified fields (see RestApi documentation).

**Returns** This method returns the original response including header as well as payload

## 6.6 MainController class

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

## REFERENCES

## C

`cleanup()` (*Video.Video method*), 8  
*Configuration* (class in *Configuration*), 7  
`create_video()` (*Video.Video method*), 8

## D

`download()` (*Video.Video method*), 8  
`downloadVideo()` (*VhhRestApi.VhhRestApi method*),  
 9

## G

`getAutomaticResults()` (*VhhRestApi.VhhRestApi method*), 9  
`getListofVideos()` (*VhhRestApi.VhhRestApi method*), 9  
`getRequest()` (*VhhRestApi.VhhRestApi method*), 9

## I

`is_downloaded()` (*Video.Video method*), 8

## L

`loadConfig()` (*Configuration.Configuration method*),  
 7

## P

`postAutomaticResults()`  
 (*VhhRestApi.VhhRestApi method*), 9  
`postRequest()` (*VhhRestApi.VhhRestApi method*), 9  
`printInfo()` (*Video.Video method*), 8

## R

`run()` (*Sbd.Sbd method*), 7  
`run()` (*Stc.Stc method*), 8

## S

*Sbd* (class in *Sbd*), 7  
*Stc* (class in *Stc*), 8

## V

*VhhRestApi* (class in *VhhRestApi*), 9  
*Video* (class in *Video*), 8