

Package: ovml (via r-universe)

October 14, 2026

Title Machine Learning Tools for Volleyball

Version 0.1.7

Description Image and video machine learning tools, for application to volleyball analytics.

Depends R (>= 3.3.0)

Imports assertthat, magick, magrittr, ovideo (>= 0.18.0), ovml.common, reticulate, torch

Suggests testthat

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Remotes openvolley/ovideo, openvolley/ovml.common

Config/pak/sysreqs cmake make libmagick++-dev gsfonts libavfilter-dev libicu-dev libjpeg-dev libpng-dev libuv1-dev libxml2-dev libssl-dev python3 libx11-dev zlib1g-dev

Repository <https://openvolley.r-universe.dev>

Date/Publication 2026-01-16 02:01:30 UTC

RemoteUrl <https://github.com/openvolley/ovml>

RemoteRef HEAD

RemoteSha 6abc6346f402dde3ebb591ec7d221ab897a6572a

Contents

ovml	2
ovml_install_onnxruntime	2
ovml_onnx_inspect	2
ovml_onnx_versions	3
ovml_yolo	3
ovml_yolo_detect.default	4

ovml_yolo_detect.ovml_yolo	4
ovml_yolo_detect_video	6
ovml_yolo_onnx	7

Index	8
--------------	----------

ovml	ovml
------	-------------

Description

Image and video machine learning tools, for application to volleyball analytics.

Author(s)

Maintainer: Ben Raymond <ben@untan.gl>

Authors:

- Adrien Ickowicz

Other contributors:

- openvolley.org [originator]

ovml_install_onnxruntime	<i>Install Python dependencies for ONNX inference</i>
--------------------------	---

Description

Install Python dependencies for ONNX inference

Usage

ovml_install_onnxruntime(gpu = FALSE)

ovml_onnx_inspect	<i>Inspect ONNX model input/output names and shapes</i>
-------------------	---

Description

Inspect ONNX model input/output names and shapes

Usage

ovml_onnx_inspect(onnx_file)

ovml_onnx_versions	<i>List available ONNX model versions</i>
--------------------	---

Description

List available ONNX model versions

Usage

ovml_onnx_versions()

ovml_yolo	<i>Construct YOLO network</i>
-----------	-------------------------------

Description

Construct YOLO network

Usage

ovml_yolo(version = 4, device = "cuda", weights_file = "auto", class_labels)

Arguments

version	integer or string: one of • 3 : YOLO v3 • 4 : YOLO v4 • "4-tiny" : YOLO v4-tiny • "4-mvb" : an experimental network trained specifically to detect (only) vol-leyballs • "4-tiny-mvb" : the v4-tiny version of the same • 7 or "7-tiny" : YOLO v7 or v7-tiny
device	string: "cpu" or "cuda"
weights_file	string: either the path to the weights file that already exists on your system or "auto". If "auto", the weights file will be downloaded if necessary and stored in the directory given by <code>ovml_cache_dir()</code>
class_labels	character: the class labels used for network training. If missing or NULL, these default to <code>ovml_class_labels("coco")</code> for all models except "mvb" models, which use <code>ovml_class_labels("mvb")</code>

Value

A YOLO network object

References

<https://github.com/pjreddie/darknet>, <https://github.com/WongKinYiu/yolov7>

Examples

```
## Not run:
dn <- ovml_yolo()
img <- ovml_example_image()
res <- ovml_yolo_detect(dn, img)
ovml_ggplot(img, res)

## End(Not run)
```

ovml_yolo_detect.default

Detect objects in image using a YOLO network

Description

Detect objects in image using a YOLO network

Usage

```
## Default S3 method:
ovml_yolo_detect(net, ...)
```

ovml_yolo_detect.ovml_yolo

Processing of a video file requires that ffmpeg be installed on your system. [ovideo::ov_install_ffmpeg\(\)](#) can help with this on Windows and Linux.

Description

Processing of a video file requires that ffmpeg be installed on your system. [ovideo::ov_install_ffmpeg\(\)](#) can help with this on Windows and Linux.

Usage

```
## S3 method for class 'ovml_yolo'
ovml_yolo_detect(
  net,
  image_file,
  conf = 0.6,
  nms_conf = 0.4,
  classes,
  batch_size = 4,
  ...
)
```

Arguments

net	yolo: as returned by ovml_yolo()
image_file	character: path to one or more image files, or a single video file (mp4, m4v, or mov extension)
conf	scalar: confidence level
nms_conf	scalar: non-max suppression confidence level
classes	character: vector of class names, only detections of these classes will be returned
batch_size	integer: the number of images to process as a batch. Increasing batch_size will make processing of multiple images faster, but requires more memory
...	: currently ignored

Value

A data.frame with columns "image_number", "image_file", "class", "score", "xmin", "xmax", "ymin", "ymax"

See Also

[ovml_yolo\(\)](#)

Examples

```
## Not run:
dn <- ovml_yolo()
img <- ovml_example_image()
res <- ovml_yolo_detect(dn, img)
ovml_ggplot(img, res)

## End(Not run)
```

ovml_yolo_detect_video

Run YOLO inference on a video (sampled frames + timestamps)

Description

This is a convenience wrapper around `ovideo::ov_video_frames()` and `ovml_yolo_detect()`, adding time-windowing, frame sampling and timestamps.

Usage

```
ovml_yolo_detect_video(
  net,
  video_file,
  start_time = 0,
  end_time,
  duration,
  fps_extract = 5,
  fps_video,
  conf = 0.6,
  nms_conf = 0.4,
  classes,
  batch_size = 4,
  outdir,
  keep_frames = FALSE,
  format = "jpg",
  jpg_quality = 1,
  debug = FALSE
)
```

Arguments

net	YOLO network as returned by <code>ovml_yolo()</code> .
video_file	Path to a video file.
start_time	Numeric seconds from the start of the video (default 0).
end_time	Numeric seconds from the start of the video (optional).
duration	Numeric seconds (optional). If end_time is missing, duration is used.
fps_extract	Numeric. Frames per second to extract (sampling rate). If NULL, all frames are extracted.
fps_video	Numeric. Video framerate. If missing, inferred via <code>av::av_video_info()</code> . Used only to compute timestamps when fps_extract is NULL.
conf, nms_conf, classes, batch_size	Passed to <code>ovml_yolo_detect()</code> .
outdir	Output directory for extracted frames. If missing, a temp dir is used.

keep_frames	Logical; if FALSE, extracted frames are deleted on exit.
format	"jpg" or "png" (passed to <code>ovideo::ov_video_frames()</code>).
jpg_quality	Numeric 1-31 (passed to <code>ovideo::ov_video_frames()</code> if format="jpg").
debug	Logical; passed to <code>ovideo::ov_video_frames()</code> .

Value

A list with elements: detections, frames, meta.

See Also

`ovml_yolo()`

Examples

```
## Not run:
library(ovideo)
v <- ov_example_video(1)
dn <- ovml_yolo()
out <- ovml_yolo_detect_video(dn, v, start_time = 0, duration = 3, fps_extract = 5)
head(out$detections)

## End(Not run)
```

ovml_yolo_onnx	Create an ONNX-backed YOLO detector (Ultralytics export)
----------------	--

Description

Create an ONNX-backed YOLO detector (Ultralytics export)

Usage

```
ovml_yolo_onnx(
  version = "11n-coco-nms",
  weights_file = "auto",
  providers = c("CPUExecutionProvider")
)
```

Arguments

providers	Execution providers in priority order, e.g. <code>c("CUDAExecutionProvider", "CPUExecutionProvider")</code> .
onnx_file	Path to an exported ONNX model (recommended: exported with <code>nms=True</code>).
class_labels	Character vector of class labels (index -> label).
input_size	Model input size (typically 640).

Index

ovideo::ov_install_ffmpeg(), [4](#)
ovideo::ov_video_frames(), [6](#), [7](#)
ovml, [2](#)
ovml-package (ovml), [2](#)
ovml_cache_dir(), [3](#)
ovml_install_onnxruntime, [2](#)
ovml_onnx_inspect, [2](#)
ovml_onnx_versions, [3](#)
ovml_yolo, [3](#)
ovml_yolo(), [5–7](#)
ovml_yolo_detect(), [6](#)
ovml_yolo_detect.default, [4](#)
ovml_yolo_detect.ovml_yolo, [4](#)
ovml_yolo_detect_video, [6](#)
ovml_yolo_onnx, [7](#)