

Package: ovml (via r-universe)

August 21, 2024

Title Machine Learning Tools for Volleyball

Version 0.1.6

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, torch

Suggests testthat

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Remotes openvolley/ovideo, openvolley/ovml.common

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

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

RemoteRef HEAD

RemoteSha 3469e998101aebb02a3b24e223fc8a5a1bd4beff

Contents

ovml	2
ovml_yolo	2
ovml_yolo_detect	3

Index	5
--------------	----------

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

Description

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

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 <ul style="list-style-type: none"> • 3 : YOLO v3 • 4 : YOLO v4 • "4-tiny" : YOLO v4-tiny • "4-mvb" : an experimental network trained specifically to detect (only) volleyballs • "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	<i>Detect objects in image using a YOLO network</i>
------------------	-----------------------------------------------------

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

```
ovml_yolo_detect(
  net,
  image_file,
  conf = 0.6,
  nms_conf = 0.4,
  classes,
  batch_size = 4,
  ...
)
```

Arguments

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

Index

ovideo::ov_install_ffmpeg(), 3
ovml, 2
ovml_cache_dir(), 2
ovml_yolo, 2
ovml_yolo(), 3, 4
ovml_yolo_detect, 3