

```

import time
import cv2
from datetime import datetime, time
import numpy as np
import time as time2

ap = argparse.ArgumentParser()
ap.add_argument("-v", "--video", help="path to the video file")
ap.add_argument("-a", "--min-area", type=int, default=500, help="minimum area size")
ap.add_argument("-t", "--tracker", type=str, default="csrt", help="OpenCV object tracker type")
args = vars(ap.parse_args())

# extract the OpenCV version info
(major, minor) = cv2.__version__.split(".")[ :2]
# if we are using OpenCV 3.2 or an earlier version, we can use a special factory
# function to create the entity that tracks objects
if int(major) == 3 and int(minor) < 3:
    tracker = cv2.Tracker_create(args["tracker"].upper())
    #tracker = cv2.TrackerGOTURN_create()
# otherwise, for OpenCV 3.3 or newer,
# we need to explicitly call the respective constructor that contains the tracker
# object:
else:
    # initialize a dictionary that maps strings to their corresponding
    # OpenCV object tracker implementations
    OPENCV_OBJECT_TRACKERS = {
        "csrt": cv2.TrackerCSRT_create,
        "kcf": cv2.TrackerKCF_create,
        "boosting": cv2.legacy.TrackerBoosting_create,
        "mil": cv2.TrackerMIL_create,
        "tld": cv2.legacy.TrackerTLD_create,
        "medianflow": cv2.legacy.TrackerMedianFlow_create,
        "mosse": cv2.legacy.TrackerMOSSE_create
    }
# grab the appropriate object tracker using our dictionary of
# OpenCV object tracker objects
tracker = OPENCV_OBJECT_TRACKERS[args["tracker"]]()

```