

REALISASI Program (Masih belum beres)

Objek deteksion

```
from __future__ import print_function
import cv2 as cv
import argparse

def detectAndDisplay(frame):
    frame_gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
    frame_gray = cv.equalizeHist(frame_gray)

    #-- Detect fishs
    fishes = fish_cascade.detectMultiScale(frame_gray)
    for (x,y,w,h) in fishes:
        center = (x + w//2, y + h//2)
        frame = cv.ellipse(frame, center, (w//2, h//2), 0, 0, 360,
(255, 0, 255), 4)

        fishROI = frame_gray[y:y+h,x:x+w]
        #-- In each fish, detect head
        head = head_cascade.detectMultiScale(fishROI)
        for (x2,y2,w2,h2) in head:
            head_center = (x + x2 + w2//2, y + y2 + h2//2)
            radius = int(round((w2 + h2)*0.25))
            frame = cv.circle(frame, eye_center, radius, (255, 0, 0
), 4)

    cv.imshow('Capture - Fish detection', frame)

parser = argparse.ArgumentParser(description='Code for Cascade
Classifier tutorial.')
parser.add_argument('--fish_cascade', help='Path to fish cascade.',
default='data/haarcascades/haarcascade_frontalfish_alt.xml')
parser.add_argument('--head_cascade', help='Path to head cascade.',
default='data/haarcascades/haarcascade_eye_tree_eyeglasses.xml')
parser.add_argument('--camera', help='Camera devide number.',
type=int, default=0)
args = parser.parse_args()

fish_cascade_name = args.fish_cascade
head_cascade_name = args.head_cascade

fish_cascade = cv.CascadeClassifier()
head_cascade = cv.CascadeClassifier()

#-- 1. Load the cascades
```

```

if not fish_cascade.load(cv.samples.findFile(fish_cascade_name)):
    print('--(!)Error loading fish cascade')
    exit(0)
if not head_cascade.load(cv.samples.findFile(head_cascade_name)):
    print('--(!)Error loading head cascade')
    exit(0)

camera_device = args.camera
#-- 2. Read the video stream
cap = cv.VideoCapture(camera_device)
if not cap.isOpened():
    print('--(!)Error opening video capture')
    exit(0)

while True:
    ret, frame = cap.read()
    if frame is None:
        print('--(!) No captured frame -- Break!')
        break

    detectAndDisplay(frame)

    if cv.waitKey(10) == 27:
        break

```

Smoothing

```

import sys
import cv2 as cv
import numpy as np

# Global Variables

DELAY_CAPTION = 1500
DELAY_BLUR = 100
MAX_KERNEL_LENGTH = 31

src = None
dst = None
window_name = 'Smoothing Demo'

def main(argv):
    cv.namedWindow(window_name, cv.WINDOW_AUTOSIZE)

    # Load the source image

```

```

imageName = argv[0] if len(argv) > 0 else 'lele.jpg'

global src
src = cv.imread(cv.samples.findFile(imageName))
if src is None:
    print ('Error opening image')
    print ('Usage: smoothing.py [image_name -- default
../data/lele.jpg] \n')
    return -1

if display_caption('Original Image') != 0:
    return 0

global dst
dst = np.copy(src)
if display_dst(DELAY_CAPTION) != 0:
    return 0

# Applying Homogeneous blur
if display_caption('Homogeneous Blur') != 0:
    return 0

## [blur]
for i in range(1, MAX_KERNEL_LENGTH, 2):
    dst = cv.blur(src, (i, i))
    if display_dst(DELAY_BLUR) != 0:
        return 0
## [blur]

# Applying Gaussian blur
if display_caption('Gaussian Blur') != 0:
    return 0

## [gaussianblur]
for i in range(1, MAX_KERNEL_LENGTH, 2):
    dst = cv.GaussianBlur(src, (i, i), 0)
    if display_dst(DELAY_BLUR) != 0:
        return 0
## [gaussianblur]

# Applying Median blur
if display_caption('Median Blur') != 0:
    return 0

## [medianblur]
for i in range(1, MAX_KERNEL_LENGTH, 2):
    dst = cv.medianBlur(src, i)

```

```

        if display_dst(DELAY_BLUR) != 0:
            return 0
    ## [medianblur]

    # Applying Bilateral Filter
    if display_caption('Bilateral Blur') != 0:
        return 0

    ## [bilateralfilter]
    # Remember, bilateral is a bit slow, so as value go higher, it
    takes long time
    for i in range(1, MAX_KERNEL_LENGTH, 2):
        dst = cv.bilateralFilter(src, i, i * 2, i / 2)
        if display_dst(DELAY_BLUR) != 0:
            return 0
    ## [bilateralfilter]

    # Done
    display_caption('Done!')

    return 0

def display_caption(caption):
    global dst
    dst = np.zeros(src.shape, src.dtype)
    rows, cols, _ch = src.shape
    cv.putText(dst, caption,
               (int(cols / 4), int(rows / 2)),
               cv.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255))

    return display_dst(DELAY_CAPTION)

def display_dst(delay):
    cv.imshow(window_name, dst)
    c = cv.waitKey(delay)
    if c >= 0 : return -1
    return 0

if __name__ == "__main__":
    main(sys.argv[1:])

```

Raspberry PROGRAM

```
import picamera
import time

camera = picamera.PiCamera()
camera.capture('example.jpg')

camera.vflip = True

camera.capture('example2.jpg')

camera.start_recording('examplevid.h264')
time.sleep(5)
camera.stop_recording()
```

REALISASI KABEL



Realisasi kemasan

