Source Code *GUI*:

```
from picamera.array import PiRGBArray
from picamera import PiCamera
from time import sleep, time
from datetime import datetime
import cv2 as cv
import sys, os
import io
from PyQt5.QtCore import QTimer, Qt
from PyQt5.QtGui import QImage, QPixmap
from PyQt5.QtWidgets import QApplication, QDialog, QMessageBox
from PyQt5.uic import loadUi

class veinviewer(QDialog):
	def __init__(self):
		super(veinviewer,self).__init__()
		loadUi('UI_testframe.ui',self)
		self.cam_H = 480
		self.cam_W = 320
		self.rgb_result = None
		self.start = 0
		self.save_button.clicked.connect(self.save)
		self.quit_button.clicked.connect(self.quit)
		self.start_button.clicked.connect(self.start_video)
	def quit(self):
		self.timer.stop()
		self.camera.close()
		os.system("sudo shutdown -h now")
	def setup_camera(self):
		self.camera.resolution = (self.cam_H, self.cam_W)
		#self.camera.framerate = 32
		self.camera.brightness = 40
		self.camera.contrast = 90
		self.camera.rotation = 270
```

```
                        self.camera.sharpness = 50
                        sleep(0.1)
                def start_video(self):
                        if self.start == 0 or self.start == 2:
                    if self.start == 0:
                        self.start_button.setText('Stop')
                        self.image = None
                        self.save_button.show()
                        self.camera = PiCamera()
                        self.setup_camera()
                        self.rawCapture = PiRGBArray(self.camera, size=(self.cam_H,
self.cam_W))
                        self.start = 1
                        self.timer = QTimer(self)
                        self.timer.timeout.connect(self.update_frame)
                        self.timer.start(0)
                     if self.start == 2:
                        self.save_button.show()
                        self.start_button.setText('Stop')
                        self.timer.start(0)
                        self.start = 1
                        elif self.start == 1:
                    self.start_button.setText('Start')
                    self.timer.stop()
                    self.start = 2
                def update_frame(self):
                        self.rawCapture.truncate(0)
                        self.camera.capture(self.rawCapture, format="rgb",
use_video_port=True)
                        self.image = self.rawCapture.array
                        self.gray = cv.cvtColor(self.image, cv.COLOR_BGR2GRAY)
                        clahe = cv.createCLAHE(clipLimit = 2.0, tileGridSize = (8, 8))
                        self.contrast_gray = clahe.apply(self.gray)
                        self.denoise = cv.blur(self.contrast_gray, (5, 5))
```

```
        ridge_filter = cv.ximgproc.RidgeDetectionFilter_create(cv.CV_32FC1,
1, 1, 3 )
        self.ridge = ridge_filter.getRidgeFilteredImage(self.denoise)
        kernel = cv.getStructuringElement(cv.MORPH_ELLIPSE, (5, 5))
        self.morph_erode = cv.erode(self.ridge, kernel, iterations = 1)
        self.tresh = cv.adaptiveThreshold(self.morph_erode, 255,
cv.ADAPTIVE_THRESH_MEAN_C, cv.THRESH_BINARY_INV, 11, 2)
        self.gaussian = cv.GaussianBlur(self.tresh, (5, 5), 0)
        kernel1 = cv.getStructuringElement(cv.MORPH_ELLIPSE, (5, 5))
        self.morph_dilate = cv.dilate(self.gaussian, kernel1, iterations = 1)
        self.cvt_rgb = cv.cvtColor(self.morph_dilate, cv.COLOR_GRAY2RGB)
        self.rgb_result = cv.addWeighted(self.cvt_rgb, 0.6, self.image, 0.4, 0)
        self.displayImage(self.rgb_result, 1)
    def save(self):
        dt = datetime.now()
        dt2 = dt.strftime('%d%m%y-%H%M%S')
        str_save = dt2 + '.png'
        cv.imwrite('/home/pi/WISUDA18/0original ' + str_save, self.image)
        cv.imwrite('/home/pi/WISUDA18/1gray '+ str_save, self.gray)
        cv.imwrite('/home/pi/WISUDA18/2clahe ' + str_save, self.contrast_gray)
        cv.imwrite('/home/pi/WISUDA18/3denoise ' + str_save, self.denoise)
        cv.imwrite('/home/pi/WISUDA18/4ridge ' + str_save, self.ridge)
        cv.imwrite('/home/pi/WISUDA18/5morph_erode ' + str_save,
self.morph_erode)
        cv.imwrite('/home/pi/WISUDA18/7treshold ' + str_save, self.tresh)
        cv.imwrite('/home/pi/WISUDA18/8gaussian ' + str_save, self.gaussian)
        cv.imwrite('/home/pi/WISUDA18/9morph_dilate ' + str_save,
self.morph_dilate)
        cv.imwrite('/home/pi/WISUDA18/92cvt_rgb ' + str_save, self.cvt_rgb)
        cv.imwrite('/home/pi/WISUDA18/93final ' + str_save, self.rgb_result)
        QMessageBox.information(self, 'Information', "You have succsesfully
save all your images with filename:\n" + str_save, QMessageBox.Ok, QMessageBox.Ok)

        self.timer.stop()
```

```
            self.start = 2
            self.save_button.hide()
            self.start_button.setText("start")
            self.displayImage(self.rgb_result, 1)
        def displayImage(self, img, window = 1):
            qformat = None
            if len(img.shape)==3:
                if img.shape[2]==4:
                    qformat=QImage.Format_RGBA8888
                else:
                    qformat=QImage.Format_RGB888
            outImage = QImage(img, img.shape[1], img.shape[0], img.strides[0],
qformat)
            outImage = outImage.rgbSwapped()
            if window == 1:
                self.video_read.setPixmap(QPixmap.fromImage(outImage))
                self.video_read.setScaledContents(True)

if __name__ == '__main__':
        app=QApplication(sys.argv)
        window=veinviewer()
        window.setWindowFlags(Qt.FramelessWindowHint)
        window.setGeometry(0, 0, 480, 320)
        window.show()
        sys.exit(app.exec_())
```