

ДОДАТОКА

Текстпрограми

Main.py

```
import os
import socket
import sys
import cv2
import io
import mysql.connector
from mysql.connector import Error

def video():
    face_cascade_db = cv2.CascadeClassifier(cv2.data.harcascades +
"haarcascade_frontalface_default.xml")
    eye_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
"haarcascade_eye.xml")

    cap = cv2.VideoCapture(0)
    while True:
        success, img = cap.read()
        img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade_db.detectMultiScale(img_gray, 1.1, 19)
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
            img_gray_face = img_gray[y:y + h, x:x + w]
            eyes = eye_cascade.detectMultiScale(img_gray_face, 1.1, 19)
            for (ex, ey, ew, eh) in eyes:
                cv2.rectangle(img, (x + ex, y + ey), (x + ex + ew, y + ey + eh), (255, 0,
0), 2)
            cv2.imshow('rez', img)
        if cv2.waitKey(1) & 0xff == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
    return img

def create_connection(host_name, user_name, user_password):
    connection = None
```

```

try:
    connection = mysql.connector.connect(
host=host_name,
user=user_name,
passwd=user_password
    )
print("Connection to MySQL DB successful")
except Error as e:
print(f"The error '{e}' occurred")
return connection

connection = create_connection("localhost", "root", "")
def execute_read_query(connection, query):
    cursor = connection.cursor()
    result = None
    try:
        cursor.execute(query)
        result = cursor.fetchall()
    return result
except Error as e:
print(f"The error '{e}' occurred")

select_users = "SELECT * from users"
users = execute_read_query(connection, select_users)

for user in users:
print(user)

with io.open('/path/to/file.txt', encoding='utf-8') as file:
for line in file:
if bicubic in line:
perenos()

def perenos ()
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM) # For UDP

udp_host = socket.gethostname() # Host IP
udp_port = 12345 # specified port to connect

#print type(sock) =====> 'type' can be used to see type
# of any variable ('sock' here)

sock.bind((udp_host,udp_port))

while True:

```

```
print "Waiting for client..."
data,addr = sock.recvfrom(1024)      #receive data from client
print "Received Messages:",data," from",addr
cap.release()
cv2.destroyAllWindows()
```

Client.py

```
import socket
```

```
sock =socket.socket(socket.AF_INET,socket.SOCK_DGRAM)# For UDP
```

```
udp_host=socket.gethostname()      # Host IP
```

```
udp_port=12345                      # specified port to connect
```

```
msg ="Hello Python!"
```

```
print"UDP target IP:",udp_host
```

```
print"UDP target Port:",udp_port
```

```
sock.sendto(msg,(udp_host,udp_port))
```

ДОДАТОК Б
Демонстраційні матеріали

Слайд 1

ТИТУЛЬНИЙ ЛИСТ РОБОТИ

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Факультет Автоматики і комп'ютеризованих технологій
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки

Атестаційна робота
**«Система дистанційного аналізу теплових властивостей
технічних об'єктів – програмна частина»**

Виконав:
ст. гр. КТРСм-19-1
Сухов Владислав Олександрович

Керівник:
Доц. Письменецький В.О.

Слайд 2

МЕТА ТА ЗАДАЧІ АТЕСТАЦІЙНОЇ РОБОТИ

МЕТА ТА ЗАДАЧІ АТЕСТАЦІЙНОЇ РОБОТИ

Об'єкт дослідження - процес дистанційного аналізу теплових властивостей технічних об'єктів.

Предмет дослідження –Теплові властивості технічних об'єктів.

Метою атестаційної роботи - розробка програмного забезпечення для дистанційного аналізу теплових властивостей об'єктів

Для досягнення мети атестаційної роботи **необхідно:**

- виконати огляд та аналіз проблеми дистанційного аналізу теплових властивостей технічних об'єктів;
- проаналізувати існуючі системи дистанційного аналізу теплових властивостей технічних об'єктів;
- розглянути особливості вхідних даних;
- проаналізувати сучасні апаратні та програмні засоби аналізу теплових властивостей технічних об'єктів;
- розробити алгоритм роботи програми ;
- розробити програмне забезпечення;
- провести тестування програмного забезпечення;
- провести аналіз та надати рекомендації щодо використання результатів роботи.

Слайд 3

ВХІДНІ ДАНІ ПРОГРАМИ

Вхідні данні програми

- Зображення з камери



Зображення з тепловізора

Слайд 4

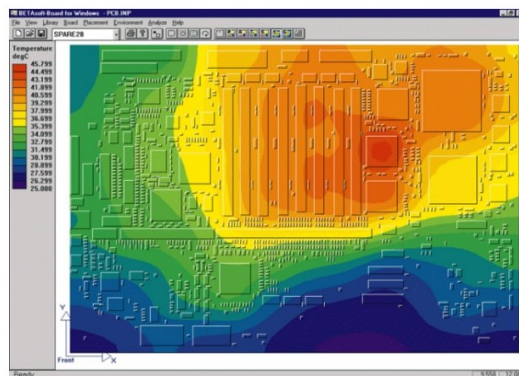
ПРИКЛАДИ ВИМІРЮВАННЯ ТЕПЛА



Слайд 5

ПРОГРАМНІ ЗАСОБИ АНАЛІЗУ ТЕПЛОВИХ ВЛАСТИВОСТЕЙ ОБ'ЄКТІВ

Програмні засоби аналізу теплових властивостей об'єктів

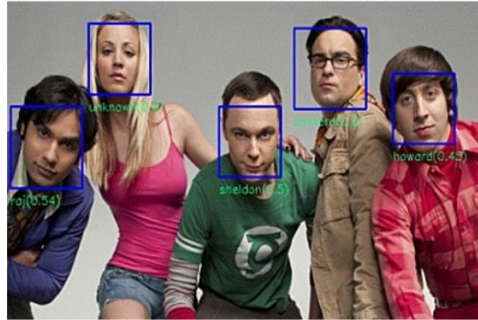


Програма BETASoft-Board

Слайд 6

OPEN CV

Open CV



Приклад роботи Open CV

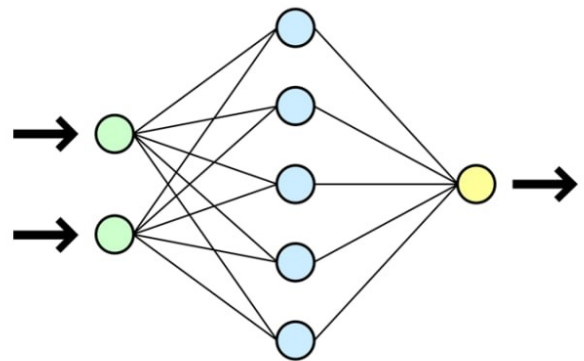
Слайд 7

ВАРІАНТИ ОБРОБКИ ДАНИХ

Варіанти обробки даних



База даних



Двошарова нейронна мережа

Слайд 8

РЕЗУЛЬТАТИ РОБОТИ ПРОГРАМИ

Результати роботи програми



Приклад результату роботи програми з живим об'єктом



Приклад результату роботи програми з технічним об'єктом

Слайд 9

ПЕРЕВАГИ ТА НЕДОЛІКИ

Переваги та недоліки

Переваги

- Система може працювати з різним типом об'єктів
- Система працює автоматизовано
- Система має мобільність

Недоліки

- Для коректної роботи системи потрібно щоб об'єкт був у центрі камери
- Система не може працювати з об'єктами що рухаються на великій швидкості
- Система може некоректно працювати якщо у зоні роботи є об'єкт з вищою температурою

Слайд 10

ВИСНОВКИ

ВИСНОВКИ

В магістерській атестаційній роботі розглянуто проблему вимірювання теплових властивостей об'єктів дистанційно.

Проведено аналіз існуючих моделей систем дистанційного вимірювання теплових властивостей об'єктів, виявлено їх основні компоненти. У функції даних систем входить вимірювання теплових властивостей об'єктів за допомогою отриманої з сенсорів інформації. Розглянуто найбільш поширені методи вимірювання температури об'єктів, найліпшим з яких являється тепловізор. Визначено основні функції системи.

Спроектовано систему дистанційного вимірювання теплових властивостей об'єктів, її інформаційну модель. Така система зчитує данні з зображень з камери та виявляє що за об'єкт знаходиться на зображенні за допомогою бази даних. Опіраючись на наявну інформацію вона здатна виявити температуру об'єкта за допомогою кількості заявлених пікселів у робочій області.

Розроблено базу даних у яку додається інформація про об'єкт.

У якості апаратної платформи для системи дистанційного вимірювання теплових властивостей об'єктів було обрано Raspberry Pi 4B, у якому наявні необхідні датчики, та доступ до мережі Інтернет. Це забезпечує компактність та мобільність разом з необхідною розрахунковою здатністю.

Було проведено тестування пристрою для коректування коду та виявлення недоліків системи.

У подальшому, для покращення роботи системи необхідно використовувати більш дорогі та ресурсо-доступні компоненти. Це потрібно для того щоб система могла більш швидко реагувати та обробляти зображення. Також для коректної праці системи необхідно щоб досліджуваний об'єкт знаходився у центрі кадра та у робочій зоні не було зайвих об'єктів з висою температурою.

