

Università degli Studi di Padova – Dipartimento di Ingegneria Industriale

Corso di Laurea in Ingegneria dell'Energia

***Relazione per la prova finale
«Realizzazione di una Termocamera
con Raspberry Pi IV»***

Tutor universitario: Prof. Luigi Alberti

Laureando: *Aleksandar Stupić*

Padova, 15/03/2024

- Cos'è una termocamera?

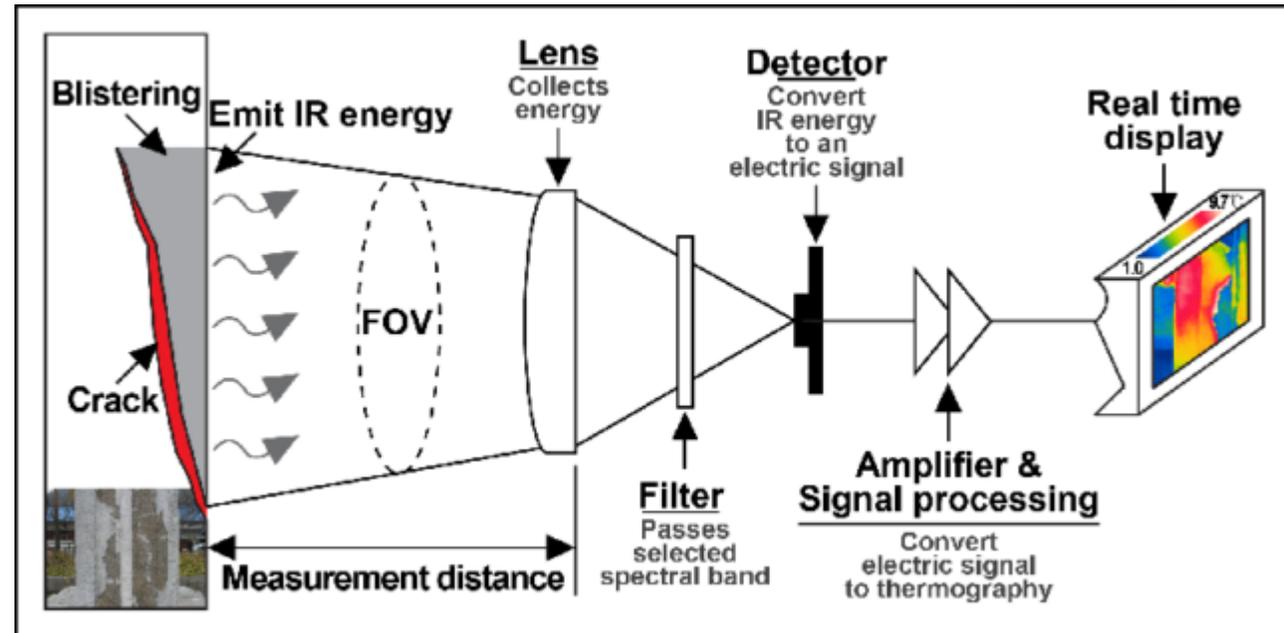
Termocamera: strumento per la rilevazione di modelli termici sotto forma di immagine.

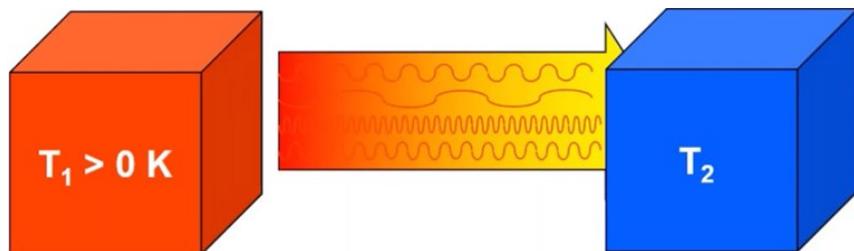


- Come funziona?

Quattro passaggi:

1. Rilevamento a infrarossi
2. Conversione in segnale elettrico
3. Elaborazione dell'immagine
4. Visualizzazione

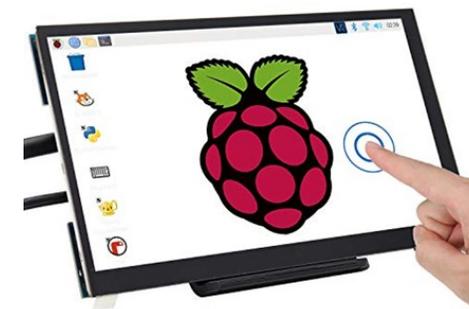




- Studio teorico della termografia
- Studio dei principi di analisi dei dati rilevati da una termocamera
- Distinta dei materiali necessari
- Procedimento per la realizzazione di una termocamera
- Studio del processo di programmazione di una termocamera con Raspberry Pi IV
- Studio della rilevanza pratica di una termocamera



$$T = \sqrt[4]{\frac{R}{\sigma \varepsilon}}$$



Spettro di radiazione elettromagnetica:

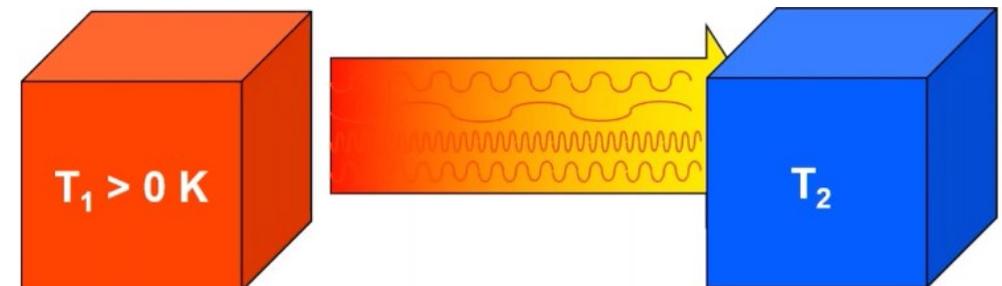
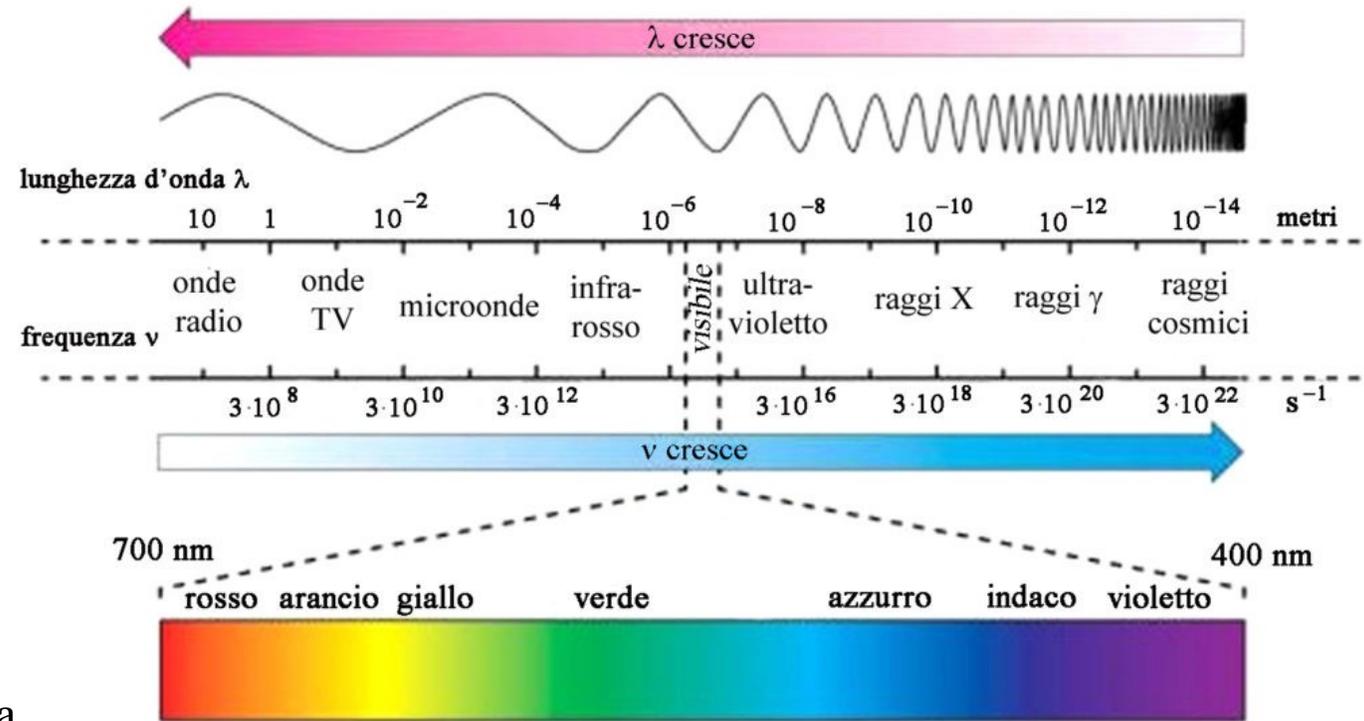
- Spettro visivo: 0,4 μm – 0,7 μm
- Spettro Infrarosso: 0,7 μm – 100 μm

Trasmissione del calore per irraggiamento:

- Tutti i corpi con temperatura sopra lo zero assoluto ($T > -273,15\text{ }^\circ\text{C}$) emettono energia termica per irraggiamento
- L'energia si trasmette tra i corpi tramite onde elettromagnetiche con diverse lunghezze d'onda secondo la formula:

$$c = \lambda f$$

Dove
 c = velocità di propagazione della luce
 λ = lunghezza d'onda
 f = frequenza



Tipologie di analisi:

- Quantitativa
- Qualitativa

$$T = \sqrt[4]{\frac{R}{\sigma \varepsilon}}$$

Dove

T = Temperatura

R = Intensità della radiazione infrarossa

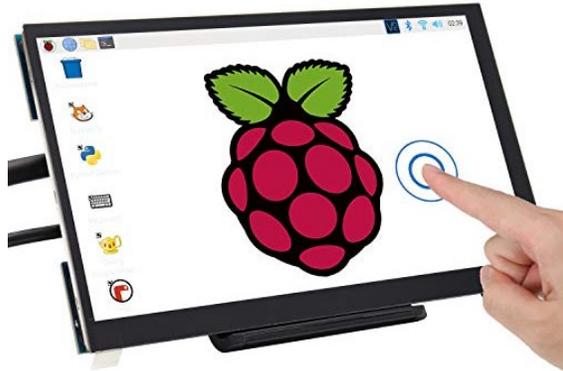
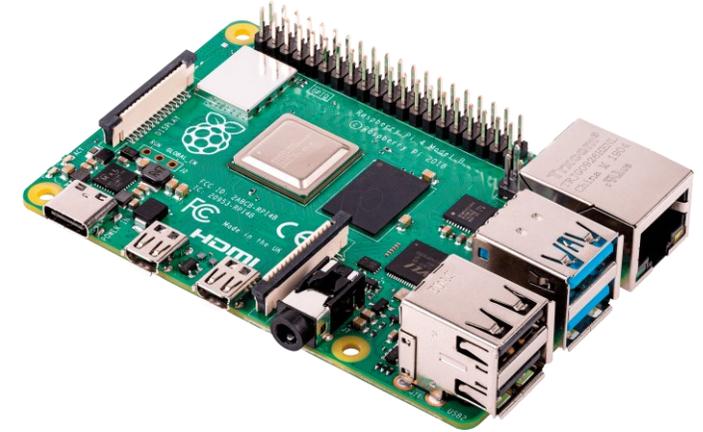
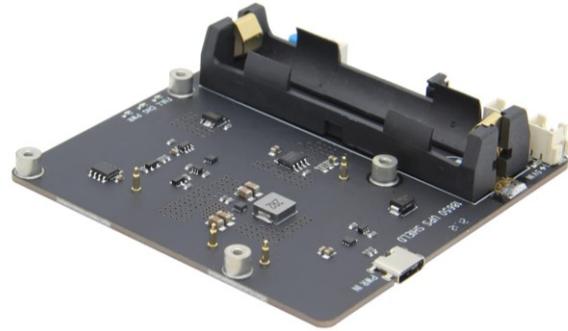
σ = costante universale (Stefan-Boltzman)

ε = emissività del materiale

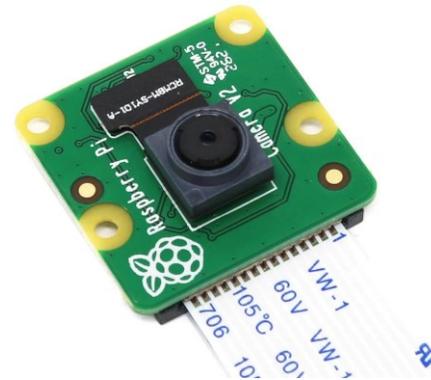
Analisi qualitativa:

- Utilizzo della formula con valori costanti eccetto R
- R si misura rilevando la radiazione IR ad onda lunga (7-14 micron) che causa una diminuzione della resistenza elettrica di ogni cella
- Valore costante $\sigma = 5.67 \times 10^{-8} \text{ W/m}^2 \text{ K}^4$
- In natura l'emissività della maggior parte dei materiali solidi e liquidi $\varepsilon = 0,95$

Materiale	ε
Corpo "nero"	1
Pelle umana	0,98
acqua	0,98
amianto	0,95
ceramica	0,95
fango	0,95
cemento	0,95
tessuto	0,95
ghiaia	0,95
carta	0,95
plastica	0,95
gomma	0,95
legno	0,95
rame (ossidato)	0,68
acciaio inox	0,1
rame (pulito)	0,02
Alluminio (pulito)	0,05



- Raspberry Pi modello 4B con 4GB
- Kingston Scheda microSD 128 GB
- Geekworm Rpi 4 UPS X703
- Freenove 5 Inch Touchscreen Monitor
- Waveshare MLX90640 IR Array con 32x24 Poxels 55°
- Pi camera V2.1 con Sensore IMX219
- Termocamera Hikmicro E01 (IR 240x240) di controllo



1. Acquisto e assemblaggio delle componenti

2. Configurazione del microcontrollore Raspberry Pi IV

- scaricare il sistema operativo Raspberry Pi OS sulla carta SD
- collegare il sistema UPS alla scheda madre



3. Collegamento schermo

- montarlo sopra il microcontrollore

4. Configurazione del sensore Waveshare MLX90640, 32x24 IR Array

- collegare il sensore al microcontrollore sui PIN: (2) 5 V, (3) SDA, (5) SCL, e (6) GND
- installare le librerie richieste dal sensore: le librerie di adafruit, I2C-tools, smbus, RPI.GPIO.





Avvio di programma

Cattura dell'immagine termica

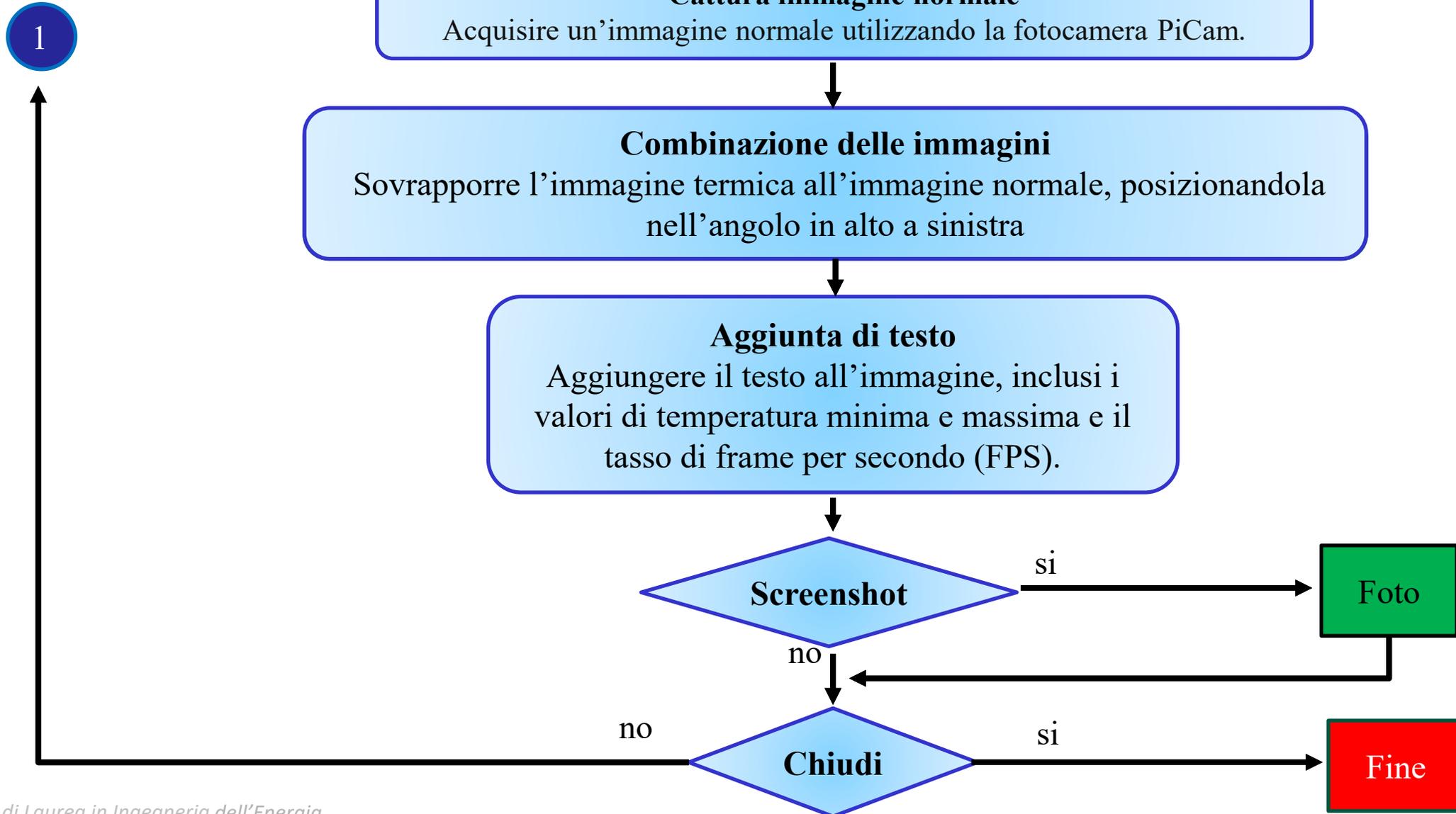
Acquisire un'immagine termica utilizzando la fotocamera termica MLX90640

Elaborazione immagine termica

Trasformare l'immagine termica in una rappresentazione a colori utilizzando la mappa dei colori RGB(8.8.8).

Aggiungere una croce all'immagine per indicare la posizione della temperatura massima.

1



5. Programmazione della Termocamera

- Importare le librerie board e busio richieste dalla libreria di adafruit per ottenere i dati dal sensore termico
- Importare le librerie time per ottenere una lettura del framerate della camera e numpy per vari calcoli matematici.
- Importare le librerie PiCamera2 e adafruit_mlx90640 per ottenere un output rispettivamente dalla videocamera e dal sensore termico
- Importare librerie PIL (a.k.a. Pillow), tkinter e cv2 per processare i dati del sensore in un immagine e per fare la GUI del programma.
- Inizializzare i vari oggetti relativi alla libreria di adafruit, PiCamera2 e tkinter.

```
import time, board, busio
import numpy as np
import adafruit_mlx90640
import datetime as dt
import cv2
from picamera2 import PiCamera2 as PiCam
from tkinter import *
from PIL import Image, ImageTk
```

```
mlx = adafruit_mlx90640.MLX90640(i2c)
```

```
Cam = PiCam()
```

```
app = Tk()
```

- Definire le costanti relative alla frequenza del sensore termico, varie impostazioni della videocamera, la grandezza della nostra GUI e la risoluzione della termocamera.
- Trovare la prima funzione, "td_to_img", per convertire un vettore di temperature in un vettore di numeri
- Definire la funzione "takeImage" per salvare un'immagine in memoria.
- Definire la funzione "getImg", per prendere i dati dal sensore termico e trasformarli in un immagine termica con una croce grigia sul punto con la temperature più alta.
- Ottenere un'immagine dalla videocamera e sovrapporre le due immagini in modo PIP (Picture in Picture)
- Aggiungere in testo la temperatura massima, la temperatura minima e gli FPS.

```
def td_to_img(f,tmax,tmin): #Function to convert  
    norm = np.uint8((f - tmin)*255/(tmax-tmin))  
    return norm
```

```
mlx.getFrame(tframe) # read MLX temperatures into frame var  
t_img = (np.reshape(tframe,mlx_shape)) # reshape to 24x32  
tmax = tframe.max() #Get maximum temperature  
tmin = tframe.min() #Get minimum temperature  
ta_img = td_to_img(t_img, tmax, tmin) #Get array with color values
```

```
img = cv2.applyColorMap(ta_img, cv2.COLORMAP_JET) #Apply color map to temperature array and get back a image  
img = cv2.resize(img, (480,360), interpolation = cv2.INTER_CUBIC) #Resize and interpolate image
```

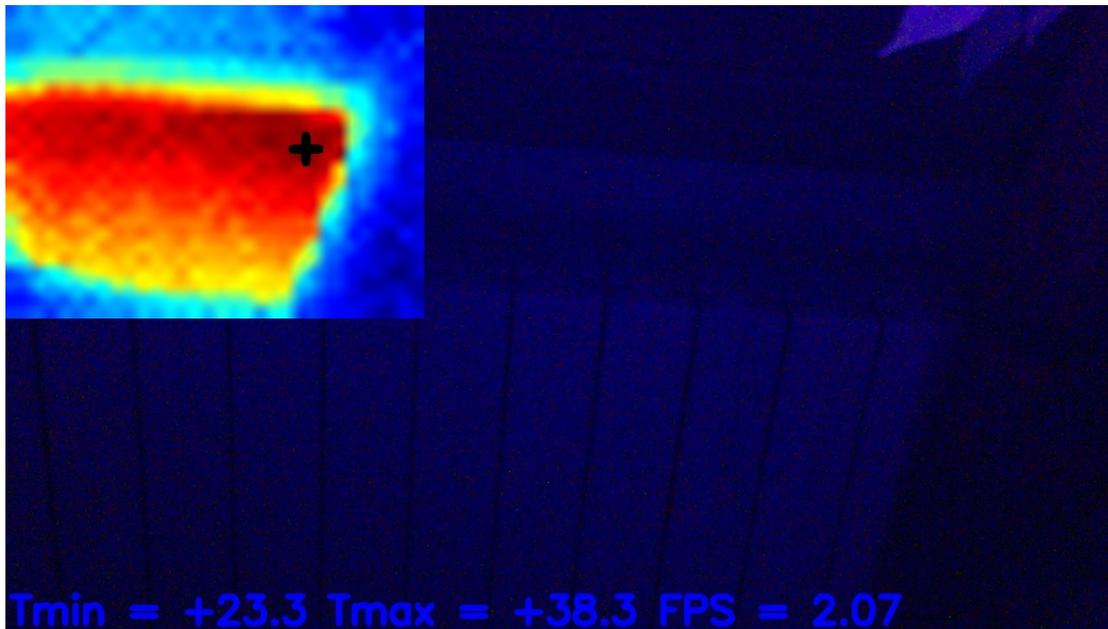
- Nella funzione "open_camera" modificare l'immagine per aggiungerla alla nostra GUI tkinter.
- Aggiungere i pulsanti per chiudere il programma e fare screenshot
- Aggiungere un bottone invisibile che si avvia una volta avviata la GUI per far partire il programma.
- Avviare la GUI.

```
def open_camera(): #Function to start the program and keep it running
    frame = getImg() #Get the frame from the getImg function
    resizedFrame = cv2.resize(frame, (640, 360), interpolation = cv2.INTER_AREA)
    opencv_image = cv2.cvtColor(resizedFrame, cv2.COLOR_BGR2RGBA) #Change the color
    captured_image = Image.fromarray(opencv_image) #Convert the image to a Pillow image
    photo_image = ImageTk.PhotoImage(image = captured_image) #Convert image to a Tkinter image
    label_widget.photo_image = photo_image #Set the label photo image to the image
    label_widget.configure(image = photo_image) #Update the label with the image
    label_widget.after(10, open_camera) #Recursively call the function
```

```
Button(app, text = 'Fai uno screenshot', command = lambda: takeImage(getImg())).pack(side = LEFT) #Add button to save pictures to the
left of the tkinter window
Button(app, text = 'Chiudi', command = lambda: app.quit()).pack(side = RIGHT) #Add button to quit the app to the right of the tkinter
window
buttonstart = Button(app, text = 'Avvia Cam', command = open_camera()) #On application start this runs to enter the open_camera function
for the first time

app.mainloop() #Start the tkinter app and thus the whole program
```

6. Test di funzionamento e confronto con una termocamera in commercio HIKMICRO E01, 96x96 IR





Durante il COVID19 in strutture pubbliche:

- Ospedali
- Aeroporti
- Stazioni ferroviarie



www.dii.unipd.it

Nelle industrie e nel privato per rilevare anomalie:

- Perdite energetiche
- Impianti elettrici
- Impianti termici
- Impianti fotovoltaici
- Sistemi meccanici
- Edilizia (infiltrazione di acqua, isolamento,...)



- Il confronto con la termocamera in commercio ha dimostrato l'accuratezza della termocamera realizzata. L'emissività corrisponde a $\varepsilon = 0,95$ e la risoluzione a $\pm 1 \text{ }^\circ\text{C}$.
- Utilizzando solo un sensore 32 x 24 pixels, l'immagine è sgranata e non interpretabile.
- Bisogna interpolare l'immagine per aumentare la risoluzione (in questo caso di 15 volte, arrivando a 480 x 360 pixels) e di conseguenza si abbassa la frequenza del numero di frame al secondo da circa 12 a 2-3.
- Per migliorare la visibilità e distinzione precisa dei corpi nell'immagine, si necessita l'aggiunta di una videocamera normale. Aggiungendo la videocamera, in modalità picture in picture si può individuare con precisione il punto più caldo nell'inquadratura.
- I risultati migliori si ottengono con la tavolozza di colori RGB (8.8.8).

