



Teste Saturação da câmera

Sobre o tema na documentação:

Determinar níveis de luminosidade (bom, médio, ruim).

O problema de ser em tempo real é que essa funcionalidade não existe no sensor
então teria que criar. Essas métricas são de processamento de imagem a câmera não tem isso nativamente, então teria que criar uma função de processamento de imagem.

Para calcular a saturação e interpretar as métricas de luminosidade e nitidez com base nas imagens processadas:

<https://prod-files-secure.s3.us-west-2.amazonaws.com/c5c56f8e-8abf-4750-a27e-28a1d1406c2e/420f8e5b-d11e-407e-952e-f4b90ca730b0/test.py>

```
from statistics import mean
from python3_scripts import cassandra.py
from glob import glob
import cv2
import os

def processamento_imagem(image_path):
    # Carregar a imagem
    img = cv2.imread(image_path)
```

```

if img is None:
    print(f"Erro ao carregar a imagem: {image_path}")
    return None, None, None

# Converter para o espaço de cores HSV
hsv_img = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

# Calcular a saturação média da imagem
saturacao_media = hsv_img[:, :, 1].mean()

# Calcular o nível de embaciamento e luminosidade
embacamento = cassandra.get_blurriness_level(img)
luminosidade = cassandra.get_lighting_level(img)

return saturacao_media, embacamento, luminosidade

# Caminho do diretório de imagens
diretorio_imagens = "./images_sensor"

# Iterar sobre os diretórios de imagens
for c in [name for name in os.listdir(diretorio_imagens) if os.path.isdir(os.path.join(diretorio_imagens, name))]:
    files = glob(f"{diretorio_imagens}/{c}/*")
    print(f'===== {c.upper()} =====')
    print("")

# Inicializar listas para as métricas
saturacao_levels = []
blurriness_levels = []
lighting_levels = []

for img_file in files:
    # Processar cada imagem
    saturacao, embacamento, luminosidade = processamento_imagem(img_file)

    if saturacao is not None:
        saturacao_levels.append(saturacao)

```

```

        blurriness_levels.append(embaciamento)
        lighting_levels.append(luminosidade)

# Calcular médias das métricas
if saturacao_levels:
    saturacao_media = mean(saturacao_levels)
    embaciamento_media = mean(blurriness_levels)
    luminosidade_media = mean(lighting_levels)

# Interpretar métricas de luminosidade
if luminosidade_media <= 100:
    status_luminosidade = "escura"
elif 100 < luminosidade_media <= 150:
    status_luminosidade = "normal"
else:
    status_luminosidade = "saturada"

# Interpretar métricas de nitidez
if embaciamento_media <= 300:
    status_embaciamento = "embaçada"
else:
    status_embaciamento = "normal"

print("Nível de Saturação: %.2f" % saturacao_media)
print("Nível de Luminosidade: %.2f (%s)" % (luminosidade_media, status_luminosidade))
print("Nível de Nitidez: %.2f (%s)" % (embaciamento_media, status_embaciamento))
else:
    print("Nenhuma imagem encontrada.")

print("")

```