



**Universidade Federal de Campina Grande – UFCG**

### **MISSÃO 4.0: Hand Tracking**

**RELATÓRIO:**Realizar o reconhecimento de números utilizando os dedos da mão.

Autora: Melry Kétury de Carvalho Mendes

Campina Grande-PB, março de 2024

[melry.ketury@estudante.ufcg.edu.br](mailto:melry.ketury@estudante.ufcg.edu.br)

## INTRODUÇÃO

O reconhecimento de números utilizando os dedos da mão é uma aplicação interessante da visão computacional e do processamento de imagens. Esta aplicação utiliza a biblioteca OpenCV (Open Source Computer Vision Library), uma biblioteca de código aberto amplamente utilizada para processamento de imagens e visão computacional. Descreveremos o desenvolvimento de uma aplicação de reconhecimento de números utilizando os dedos da mão com OpenCV e o Mediapipe.

## OBJETIVO

O objetivo deste projeto é desenvolver uma aplicação capaz de reconhecer números através dos dedos da mão em tempo real, utilizando uma webcam como entrada de vídeo. A aplicação deve ser capaz de identificar os dedos levantados em uma mão e interpretá-los como números de 0 a 5 da mão esquerda, caso utilize a mão direita o código deverá ser modificado pois o polegar já mudaria o sentido.

## CÓDIGO:

```
import cv2
import mediapipe as mp

video = cv2.VideoCapture(0)

hand = mp.solutions.hands
Hand = hand.Hands(max_num_hands=1)
mpDraw = mp.solutions.drawing_utils

while True:
    check, img = video.read()
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = Hand.process(imgRGB)
    handPoints = results.multi_hand_landmarks
    h, w, _ = img.shape
    pontos = []
    if handPoints:
        for points in handPoints:
            mpDraw.draw_landmarks(img, points, hand.HAND_CONNECTIONS)
            for id, cord in enumerate(points.landmark):
                cx, cy = int(cord.x * w), int(cord.y * h)
                #cv2.putText(img, str(id), (cx, cy + 10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 0, 0), 2)
                pontos.append((cx, cy))
    dedos = (8, 12, 16, 20)
    contador = 0
    if pontos:
        if pontos[4][0] < pontos[2][0]:
            contador += 1
        for x in dedos:
```

```

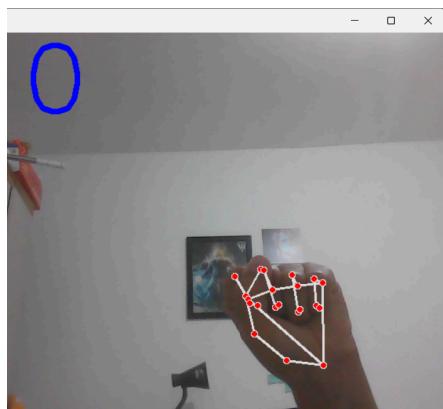
        if pontos[x][1] < pontos[x - 2][1]:
            contador += 1

cv2.putText(img, str(contador), (100, 100),
cv2.FONT_HERSHEY_SIMPLEX, 4, (255, 0, 0), 5)
print(contador)
cv2.imshow("image", img)
cv2.waitKey(1)

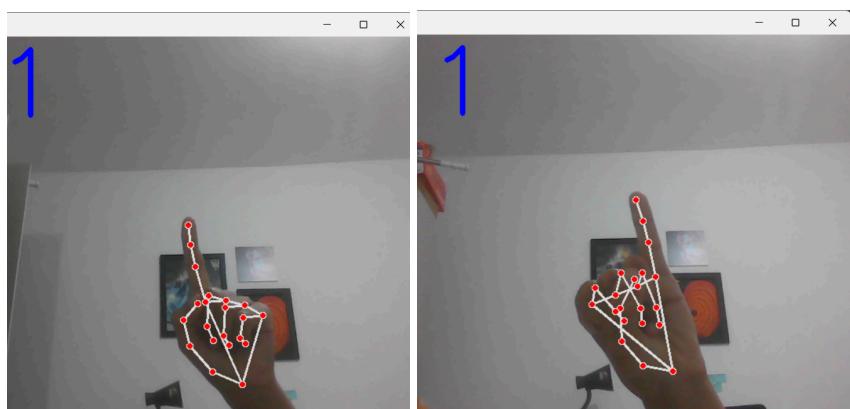
```

## EXECUÇÃO DO CÓDIGO:

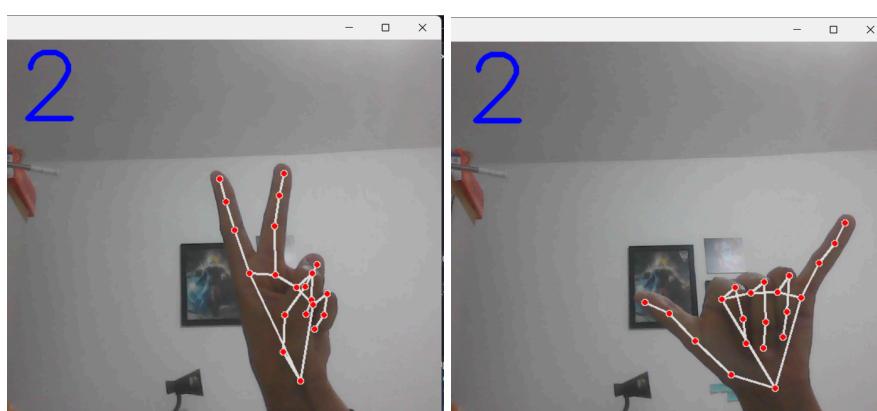
-Número 0:



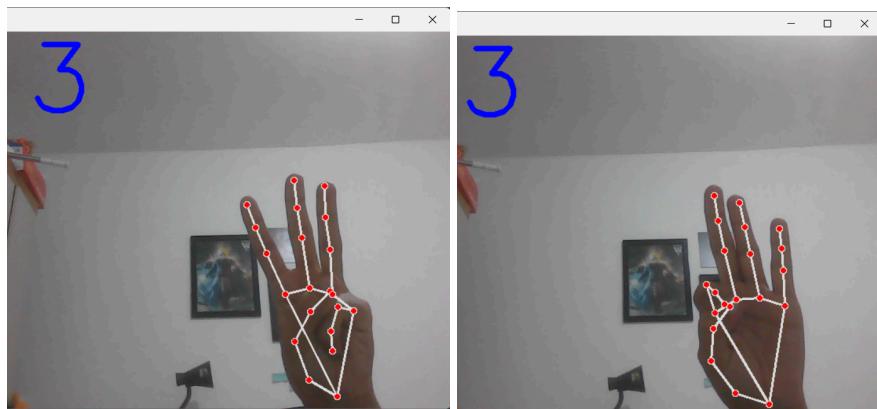
-Número 1:



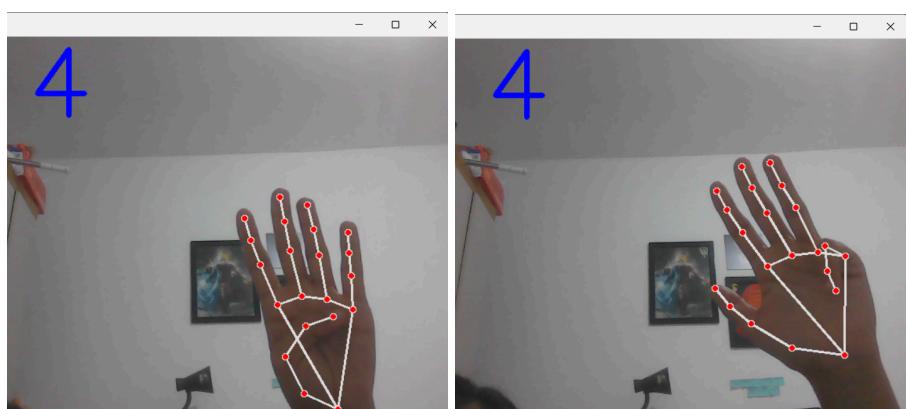
-Número 2:



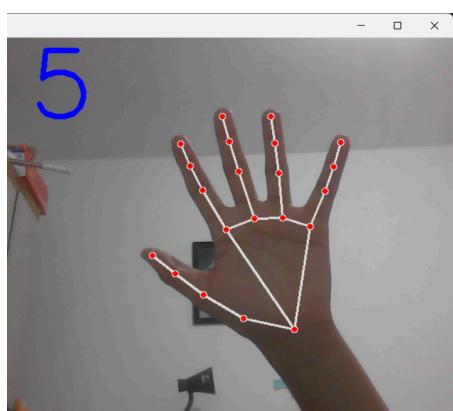
-Número 3:



-Número 4:



-Número 5:



## EXPLICAÇÃO DO CÓDIGO:

### 1. Importações de Bibliotecas:

-Import cv2: Importa a biblioteca OpenCV, que é usada para trabalhar com imagens e vídeos.

-Import mediapipe as mp: Importa a biblioteca MediaPipe, que fornece soluções pré-treinadas para tarefas de visão computacional, como detecção de mãos. A instalação da biblioteca MediaPipe, fornece modelos pré-treinados para detecção e rastreamento de mãos

### 2. Configuração da Câmera:

-vídeo = cv2.VideoCapture(0): Inicializa a captura de vídeo da webcam do computador.

### 3. Inicialização do Modelo de Detecção de Mão:

-hand = mp.solutions.hands: Cria um objeto para a detecção de mãos da MediaPipe.

-Hand = hand.Hands(max\_num\_hands=1): Inicializa o modelo de detecção de mãos, configurando para detectar apenas uma mão por vez.

#### 4. Loop Principal:

-while True:: Inicia um loop infinito que executa o código continuamente até que seja interrompido.

#### 5. Captura de Vídeo e Processamento:

-check, img = video.read(): Captura um quadro de vídeo da webcam.

-imgRGB = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB): Converte o quadro capturado para o formato RGB.

-results = Hand.process(imgRGB): Processa o quadro capturado para detectar mãos.

-handPoints = results.multi\_hand\_landmarks: Obtém os pontos das mãos detectadas no quadro.

#### 6. Desenho dos Pontos da Mão:

-for points in handPoints:: Itera sobre as mãos detectadas.

-mpDraw.draw\_landmarks(img, points, hand.HAND\_CONNECTIONS): Desenha os pontos das mãos no quadro capturado.

#### 7. Contagem dos Dedos:

-for id, cord in enumerate(points.landmark):: Itera sobre os pontos de cada mão.

-pontos.append((cx, cy)): Armazena as coordenadas (x, y) de cada ponto da mão.

-.Calcula quantos dedos estão levantados usando as coordenadas dos pontos das mãos.

#### 8. Exibição do Resultado:

cv2.putText(img, str(contador), (100, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 4, (255, 0, 0), 5): Exibe o número de dedos levantados no quadro capturado.

#### 9. Mostrando o Quadro Atualizado:

-cv2.imshow("image", img): Mostra o quadro atualizado com os pontos das mãos e a contagem de dedos.

-cv2.waitKey(1): Aguarda um pequeno período de tempo para permitir que o OpenCV exiba o quadro.

## CONCLUSÃO:

Desenvolvimento da aplicação de reconhecimento de números utilizando os dedos da mão com OpenCV e MediaPipe demonstra o potencial da visão computacional e do aprendizado de máquina para criar aplicações interativas e intuitivas. A aplicação desenvolvida pode ser utilizada em vários cenários, desde jogos educacionais até interfaces.