UNIVERSIDADE FEDERAL DE SÃO PAULO
INSTITUTO DE CIÊNCIA E TECNOLOGIA

Projeto Final Redes Neurais

Controlador de Mídia por Gestos com a Mão

Professor : Marcos Quiles
Richard Rangel do Nascimento Junior - 156575

Introdução

O intuito desse projeto é desenvolver um sistema em tempo real que utiliza uma webcam para reconhecer gestos específicos com a mão e traduzi-los em comandos diretos. A ideia desse projeto é utilizar a webcam para identificar os gestos e a partir disso enviar comandos e controlar a mídia por meio dos sinais identificados na mão. Então é utilizado a webcam por meio da biblioteca cv2 da opencv para capturar os frames e com o modelo Yolo8 é identificados os gestos e então é executado comandos por meio do código no python para controlar a mídia

O objetivo de tudo isso é dar maior conforto ao usuário que não terá mais a necessidade de ficar indo na aba, precisando mexer no mouse, selecionar a página, com esse projeto tudo fica mais prático necessitando apenas fazer um gesto e pronto.

Trabalhos Relacionados

Há o artigo ‘Real-Time Hand Gesture-Based System for Human-Machine Interaction’ publicado em 2023 que é semelhante, contudo utiliza YOLOv5 enquanto nesse projeto é utilizado o YOLOv8 que é uma versão mais moderna.

Tambem há outro artigo chamado ‘Deep Learning-Based Hand Gesture Recognition System and Design of a Human–Machine Interface’ que para essa tarefa utiliza outras redes neurais para buscar o melhor resultado e com uma maior variedade de gestos

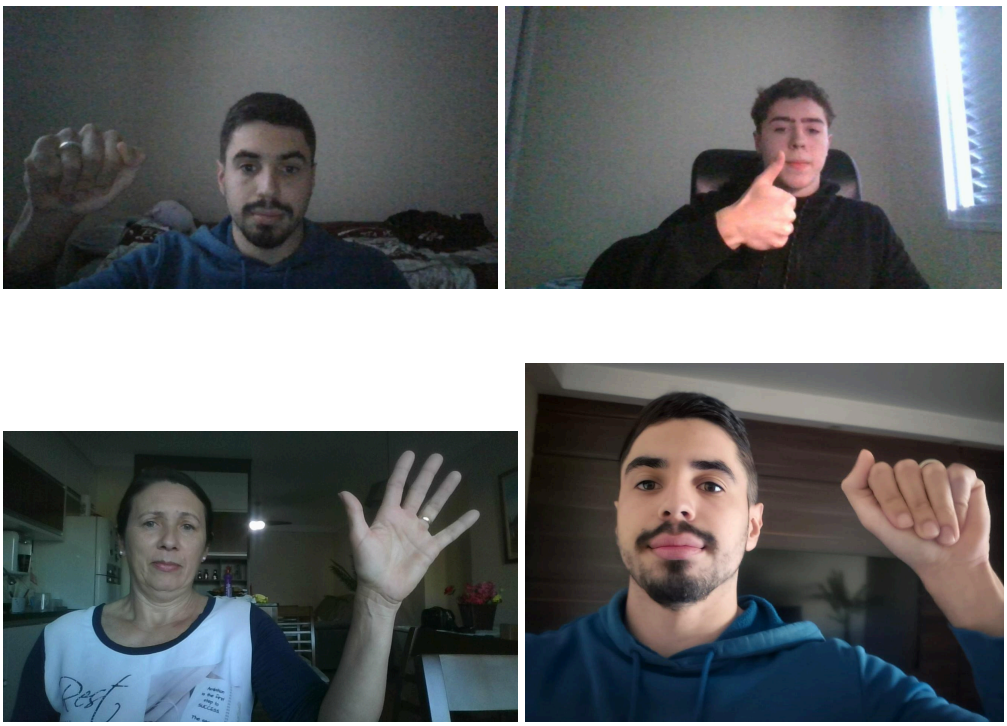
Tecnologias Utilizadas

Para o desenvolvimento desse projeto foram utilizadas algumas tecnologias para apoiar o desenvolvimento dele. O roboflow é uma plataforma online utilizada para gerenciar o dataset, fazer o labeling e pré-processar o dataset de imagens de gestos, aplicando um resize. Além disso, o Python foi utilizado para criar o código, executando localmente como no google colab.

Dentre as principais bibliotecas para executar esse projeto está a ultralytics(criadora do Yolov8 que foi utilizado como modelo), a openCV, mais especificamente a cv2, que é responsável pelo uso da webcam e capturar os frames e por último a spotify, que conecta o código python com a WEB API Spotify para ser possível executar os comandos.

Desenvolvimento

Para criar a base de dados foi tirado fotos tanto pela webcam, pois seria por ela que seria visualizado os sinais depois, como pelo celular, para ter uma melhor qualidade de imagens. Há aproximadamente 98 imagens de cada classe. Para dar variedade a base, ela foi constituída de imagens de 3 pessoas diferentes e com diferentes níveis de iluminação e diferentes fundos, visando uma maior generalização.



Amostra de imagens que foram tiradas e para montar a base de dados

Após ter todas as imagens da base começa o processo de labeling, para fazer tal tarefa foi utilizado o roboflow. A partir disso em cada imagem foi feito caixas nas mãos para o modelo poder identificar os sinais futuramente e cada gesto recebia a atribuição da classe, após fazer isso com todas as imagens manualmente foi feito o processo de separação entre treino(70%), validação(20%) e teste(10%) e então o pré-processamento padronizando o tamanho das imagens para 640x640.

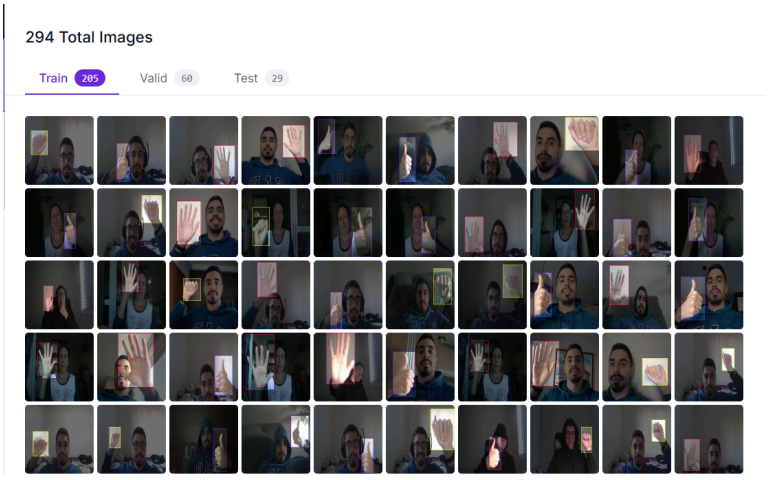


imagem da base tratada no roboflow

Com a base já pronta se inicia a parte do trabalho no Google Colab, a ideia de ir para esse ambiente é aproveitar o maior poder computacional que há lá com a GPU e assim acelerar o processo de treinamento, então com a api do roboflow a base é baixada no ambiente.

Usando a biblioteca da ultralytics é então importado o modelo Yolov8n.pt . Esse foi o modelo escolhido pois é o último YOLO lançado pela ultralytics e ele tem excelente desempenho, como a tarefa não é algo tão complexo e os sinais nas mãos são diferentes foi usado a versão nano que já atende a complexidade do problema, além disso, foi utilizado uma versão pré-treinada do Yolov8n (indicado pelo '.pt' no nome).

Então com isso o modelo foi treinado por 50 épocas, as medidas avaliativas dele foram boas, com precision, recall e mAP50 muito próximos de 1,já o mAP50-95 variou de acordo com a classe, o comando previous com o melhor resultado, já que é um gesto mais único e não pode ter muitas alterações, e o comando next sofreu mais com o pior resultado por poder ter diferentes variações da mesma coisa. Assim, foi utilizado o arquivo 'best.pt', referente ao melhor desempenho do modelo obtido durante o treinamento, para a próxima etapa.

Agora nesta etapa, não é mais utilizado o Google Colab e é criado um script no python para realizar a tarefa, nesta etapa ela é composta por duas partes, a primeira um código utilizando pyautogui e outro utilizando o spotipy.

Tratando primeiro das semelhanças entre eles, em ambos foi utilizado a biblioteca cv2 para ter acesso a webcam e assim ser possível capturar os frames para ser possível avaliar os gestos. Para avaliar eles, é carregada a rede Yolo já treinada para poder fazer a avaliação dos frames. Então é avaliado frame a frame se há a presença dos gestos e quando identificado, o comando referente a ele é executado. Com relação a isso, há um tempo de 'cooldown'(espera) de dois segundos entre executar um próximo comando captado, evitando que fosse executado um comando várias vezes repetidas enquanto era para acontecer uma vez só.

Falando sobre as diferenças, na primeira parte, para executar os comandos, foi utilizado a biblioteca pyautogui, que é uma ferramenta para automação,permitindo que os scripts controlem o mouse e o teclado para automatizar interações com a interface gráfica do usuário (GUI), exatamente como um ser humano faria. Contudo, há uma grande limitação que com essa abordagem, para poder controlar a mídia, é necessário que a aba da mídia esteja como principal para poder receber o comando, se não estiver não funciona. Essa abordagem é mais geral pois funciona com qualquer tipo de site, contudo é limitado, e visando superar essa adversidade foi criado a segunda parte.

Nessa segunda parte foi utilizado a biblioteca spotipy para se conectar com a Web API do Spotify e assim superar a limitação da versão anterior feita. Assim, é possível controlar o Spotify de qualquer dispositivo que esteja em funcionamento, então foi superado o problema anterior e expandido as possibilidades também, pois agora não só se pode estar fora da aba, como também controlar o Spotify de outros dispositivos como celular ou televisão. Contudo para isso acontecer tem se agora só uma mídia disponível para controlar.

Desafios encontrados

As partes mais difíceis estavam em grande parte relacionadas com a criação e preparação da base de dados,pois foi um trabalho totalmente manual, foram 294 fotos tiradas e então todas passaram pelo processo de labeling. Além disso, lidar com a API do Spotify tem um grau de complexidade, pois para poder executar todos os comandos é necessário definir um escopo bem específico para permitir isso.

O modelo também enfrenta algumas dificuldades às vezes para identificar o gesto devido a iluminação do ambiente e com relação ao gesto de next, que é feito pelo 'joinha', pois ele pode ser de diferentes formas, seja com a mão de frente ou de lado.

Resultados Finais

Os resultados obtidos atingiram o objetivo inicial, após testar diversas vezes e com pessoas diferentes, em todas as vezes o programa estava funcionando conforme o esperado, enviando os comandos certos e tudo sendo executado sem bugs, apenas dificuldades que já foram explicadas anteriormente.

Referências

SARFRAZ, Muhammad et al. Real-Time Hand Gesture-Based System for Human-Machine Interaction. Electronics, v. 12, n. 19, p. 4125, 2023. Disponível em: https://www.mdpi.com/2079-9292/12/19/4125. Acesso em: 18 jul. 2025.

KUMAR, Pradeep; SHESHADRI, H. S.; PRASAD, S. R. K. Deep Learning-Based Hand Gesture Recognition System and Design of a Human–Machine Interface. In: 2nd INTERNATIONAL CONFERENCE ON ELECTRONICS AND RENEWABLE SYSTEMS (ICEARS), 2023, Tuticorin. Índia: IEEE, 2023. p. 1100-1104. DOI: 10.1109/ICEARS56399.2023.10085347.

https://roboflow.com

https://docs.ultralytics.com/models/yolov8/#overview