

Disciplina: Tópicos Avançados

Turma: Análise e Desenvolvimento de Sistemas

Nome: Giovanna Rosa Borges Pereira e Ruy Guilherme Bastos

Sistema de Detecção e Contagem de Pessoas

O presente sistema foi desenvolvido com o intuito de ser utilizado para a detecção e contagem de pessoas em assembleias estudantis, facilitando a contabilização dos participantes. Realizamos a incrementação da detecção das mãos dos integrantes, de maneira a tornar o sistema um contabilizador de votos. Desse modo, automatizamos as quantificações de presentes e votantes durante as assembleias estudantis.

Para isso, desenvolvemos o código em Python, utilizando as bibliotecas OpenCV e MediaPipe.

Estrutura do Código

O código disponibilizado no [GitHub](#) possui um arquivo intitulado “[requirements.txt](#)”, no qual apresenta as dependências utilizadas para o desenvolvimento do projeto. O arquivo “[README.md](#)” apresenta uma breve descrição do que se refere o sistema, além de uma lista com as dependências utilizadas.

O arquivo “[app.py](#)” consiste na main do projeto. O código foi estruturado da seguinte maneira:

1. Importação das bibliotecas OpenCV e MediaPipe.
 - O OpenCV foi utilizado para realizar a captura do vídeo da câmera, enquanto o Mediapipe realizou a detecção e o rastreamento dos rostos e mãos.
2. Configuração das funcionalidades utilizadas pelo MediaPipe.
 - mp.solutions.face_detection: configuração para detecção dos rostos.
 - mp.solutions.hands: configuração para detecção e rastreamento das mãos.
 - mp.solutions.drawing_utils: configuração para a exibição dos landmarks das mãos e dos rostos.
3. Configuração e inicialização dos detectores de rostos e mãos.

- `model_selection=1`: configuração para detecção de rostos em distâncias maiores. Indicamos utilizar “0” para rostos mais próximos.
 - `min_detection_confidence=0.5`: definição do mínimo de nível de confiança para considerar uma detecção válida. No caso, declaramos 50% tanto para os rostos, quanto para as mãos.
 - `min_tracking_confidence=0.5`: definição do mínimo de nível de confiança para rastrear o movimento das mãos.
4. Captura do vídeo.
- Utilização do OpenCV para acessar a webcam ou alguma outra câmera. Por padrão, utilizamos “0”, por representar a webcam do notebook. Para o caso de câmeras externas, utilize “1”, “2” e assim por diante.
 - `cam.isOpened()`: verifica se a câmera foi aberta corretamente. Caso não, ele exibirá a frase “Erro ao Acessar a Câmera”.
5. Looping dos frames do vídeo.
- `cam.read()`: lê cada frame do vídeo em tempo real. Caso o sistema apresente algum erro nessa função, ele exibirá a frase “Erro ao Capturar Vídeo”.
 - `cv2.cvtColor`: converte o frame de BGR, que é o formato padrão do OpenCV, para RGB, que é o formato esperado pelo MediaPipe.
6. Detecção dos rostos e mãos.
- `face_detection.process`: processa o frame para realizar a detecção de rostos.
 - `hands.process`: processa o frame para realizar a detecção e rastreamento das mãos.
7. Contabilização dos rostos e mãos.
- `resultado_faces.detections`: cria uma lista dos rostos detectados, caso haja algum. A quantificação desses resultados são armazenadas no `contador_faces`.
 - `mp_drawing.draw_detection`: desenha um retângulo ao redor do rosto detectado no frame.
 - `resultados_maos.multi_hand_landmarks`: cria uma lista das mãos detectadas, caso haja alguma. A quantificação desses resultados são armazenados no `contador_maos`.
 - `mp_drawing.draw_landmarks`: desenha as landmarks e conexões entre os pontos da mão detectada.
8. Adição da legenda no vídeo.

- cv2.putText: adiciona um texto no frame exibido. No caso, optamos por adicionar dois textos, um de “Rostos detectados: x” e outro de “Mãos detectadas: x”, sendo x a quantidade de rostos e mãos identificadas no frame.
9. Exibição do resultado do vídeo.
- cv2.imshow: exibe o vídeo processado em uma nova janela.
 - cv2.waitKey(1): aguarda 1 milissegundo por uma tecla. Configuramos a tecla “q” para ser utilizada como chave para encerrar o programa.
10. Liberação de recursos.
- cam.release(): libera a câmera.
 - cv2.destroyAllWindows(): fecha as janelas abertas pelo OpenCV.
 - face_detection.close() e hands.close(): libera os recursos usados pelos detectores do MediaPipe.

Referências

Alguns materiais foram utilizados para o desenvolvimento do sistema, tais como:

- <https://www.youtube.com/watch?v=1x6rjyqfH3o&list=PLTYLKz3zyxKopRT1cTB4S4asbvUCaVxnF&index=7>
- <https://www.youtube.com/watch?v=Var4pdJnIXQ>
- <https://www.youtube.com/watch?v=vKiKaErHfpw>
- https://www.youtube.com/watch?v=xMc-ZL5mOig&list=PLWyN7K28ZraQHPw_Pd_T6IIFZ1XHOi5tpP

Problemas Encontrados

Identificamos os seguintes problemas durante o desenvolvimento do código:

- Detecção de ambas as mãos da mesma pessoa (para a contagem de votos apenas uma mão deve ser considerada)
- Detecção simultânea de no máximo duas mãos por vez, independente do número de pessoas
- Melhorias na detecção de falsos positivos
- Distância da câmera e a pessoa pode interferir na detecção

Sugestões de Melhorias

Como sugestões de melhorias e soluções para os problemas encontrados, destacamos:

- Melhorar a distância de detecção de rostos e faces
- Contabilização de apenas uma mão levantada por integrante
- Melhorar falsos positivos
- Impressão dos dados coletados (emitir print do sistema com a votação final ou arquivo de texto com os integrantes e votos contabilizados, diretamente pelo sistema)
- Melhoria no design do sistema (Cor da fonte/landmarks, fonte, posicionamento, etc)