UNIVERSIDADE FEDERAL DE SÃO CARLOS
CCET - Centro de Ciências Exatas e Tecnologias

Estudando a eficiência de algoritmos de reconhecimento facial

Sebastião Venâncio Guimarães Neto

São Carlos, 17 de fevereiro de 2023

Introdução:

Neste experimento, foram utilizados dois algoritmos para detecção facial, um utilizando a biblioteca opencv juntamente com um algoritmo Haar Cascade para a detecção, e outro utilizando opencv com Mediapipe, biblioteca para reconhecimento de imagens criada pela Google, com o objetivo de comparar o tempo de execução real em diferentes condições.

Procedimento Experimental:

Para realizar este experimento, foi utilizado um notebook com processador Intel Core i5-8625U com 8GB de memória RAM, um SSD de 128GB e uma placa de vídeo MX110 2GB conectado à energia, em modo alto desempenho do Windows 10. Sendo assim, antes de iniciar as medidas, o computador foi reiniciado e, em seguida, apenas o VSCODE foi aberto.

As medidas foram realizadas utilizando a função time (time python nome.py) e o valor considerado foi o tempo real de execução, uma vez que os algoritmos não aguardam nenhuma resposta do usuário, fato que o torna mais adequado.

Assim, tais medições foram feitas com o vídeo em tempo real da webcam para os dois algoritmos, sendo três medidas para cada uma das situações analisadas, com 300 frames cada uma, as quais são:
vídeo totalmente preto, vídeo com apenas um rosto e vídeo com 2 rostos.

Por fim, com os dados obtidos foi possível calcular a média para chegar a um valor mais preciso e concluir o experimento.

Apresentação dos resultados:

Medidas feitas com o vídeo totalmente preto:

	Tempo Real 1 [s]	Tempo Real 2 [s]	Tempo Real 3 [s]	Média [s]
CV2 + Haar	13,498	13,679	13,691	13,623
CV2 + MP	14,019	12,997	13,041	13,352

Medidas feitas com apenas um rosto:

	Tempo Real 1 [s]	Tempo Real 2 [s]	Tempo Real 3 [s]	Média [s]
CV2 + Haar	20,882	18,270	18,761	19,304
CV2 + MP	14,606	13,147	12,911	13,555

Medidas feitas com dois rostos:

	Tempo Real 1 [s]	Tempo Real 2 [s]	Tempo Real 3 [s]	Média [s]
CV2 + Haar	23,301	24,084	23,545	23,643
CV2 + MP	13,199	13,170	13,093	13,154

Conclusão:

Dados os resultados acima, pode-se analisar uma semelhança no tempo de execução para o caso do vídeo totalmente preto, enquanto que nos demais o algoritmo com Mediapipe se sobressaiu. Além disso, durante a execução dos algoritmos, foi possível notar que o Mediapipe mostrou uma melhor fluidez no vídeo apresentado, além de um melhor desempenho ao reconhecer a face, indicando também olhos, nariz, orelhas e bocas, enquanto que, para o Haar Cascade, seria necessário adicionar mais arquivos .xml para adicionar essas funcionalidades de reconhecimento de partes da face, fato que diminuiria a eficiência desse, o qual não já não mostra tanta fluidez na execução do vídeo.

Pode-se indicar essa falta de fluidez do Haar no tratamento frame a frame, o qual aparenta atrasar a execução, visto que é feito explicitamente a conversão de imagem para cinza e a consulta ao arquivo .xml para fazer a detecção da face, além de utilizar um loop para desenhar o retângulo em cada face.

Portanto, fica evidente que é mais viável utilizar o algoritmo Mediapipe para o reconhecimento facial, visto que esse método manteve um tempo de execução muito semelhante para cada caso e mais eficiente em relação ao outro.

Bibliografia:

1. https://www.hashtagtreinamentos.com/controlar-webcam-com-python#:~:text=Inicialmente%20n%C3%B3s%20vamos%20importar%20a,e%20o%20c%C3%B3digo%20vai%20continuar.
2. https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python-90f89d7c0f81
3. https://www.hashtagtreinamentos.com/reconhecimento-facial-no-python