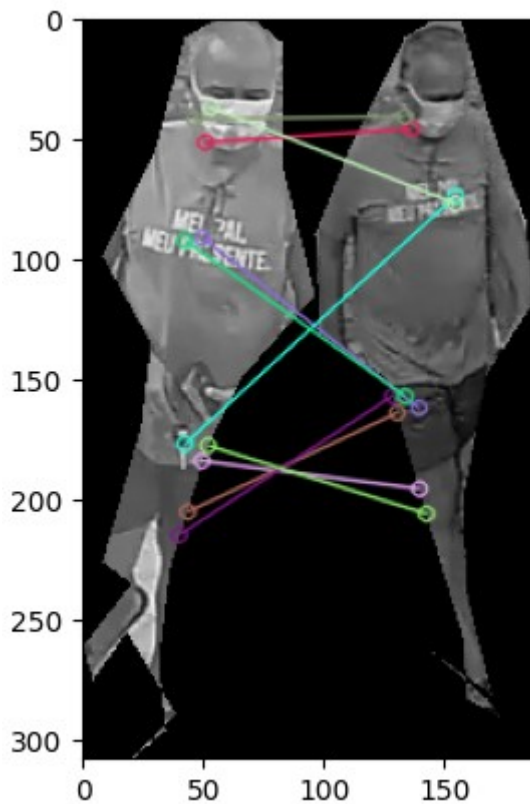


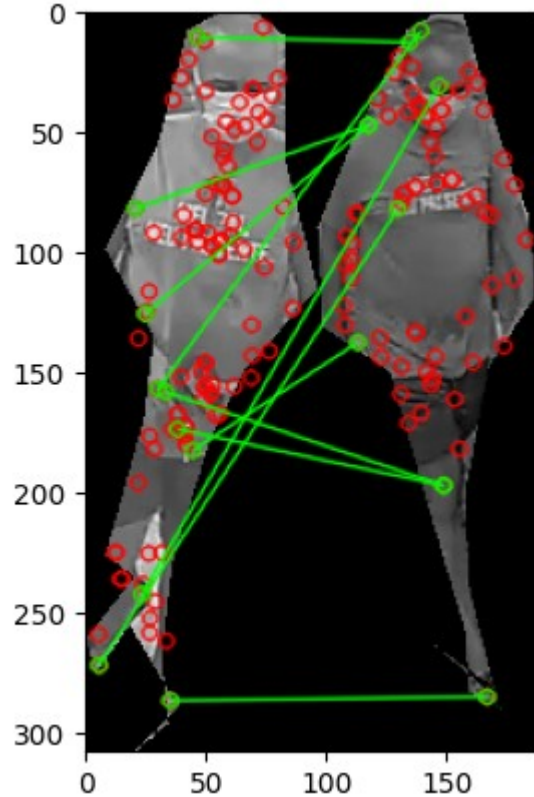
Pesquisando no Google por “opencv image similarity”, retorna-se o site “Stack Overflow” nesse link: [“https://stackoverflow.com/questions/11541154/checking-images-for-similarity-with-opencv”](https://stackoverflow.com/questions/11541154/checking-images-for-similarity-with-opencv). A resposta do “Sam” conta com 3 métodos de comparação, sendo o “Feature matching” o mais eficiente para o trabalho de comparar pessoas.

O código foi retirado da documentação do OpenCV no seguinte link: [“https://www.docs.opencv.org/master/dc/dc3/tutorial_py_matcher.html”](https://www.docs.opencv.org/master/dc/dc3/tutorial_py_matcher.html). A página explica tanto o método de força bruta, como a partir de uma biblioteca.

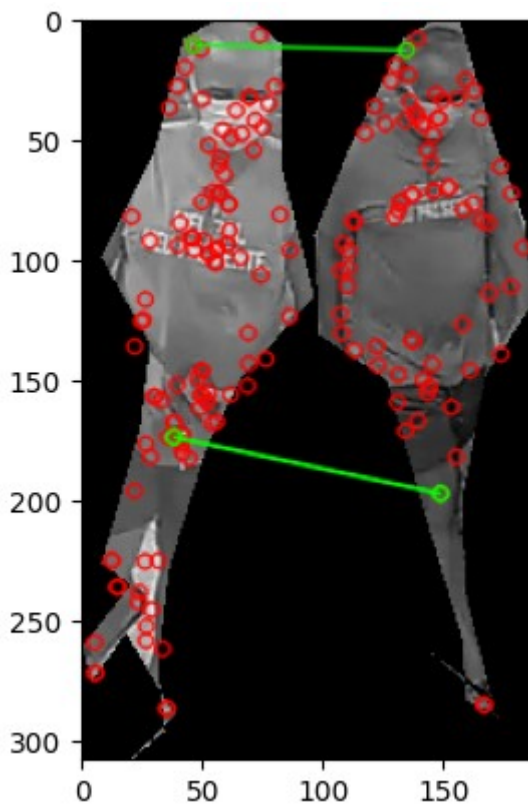
Força bruta



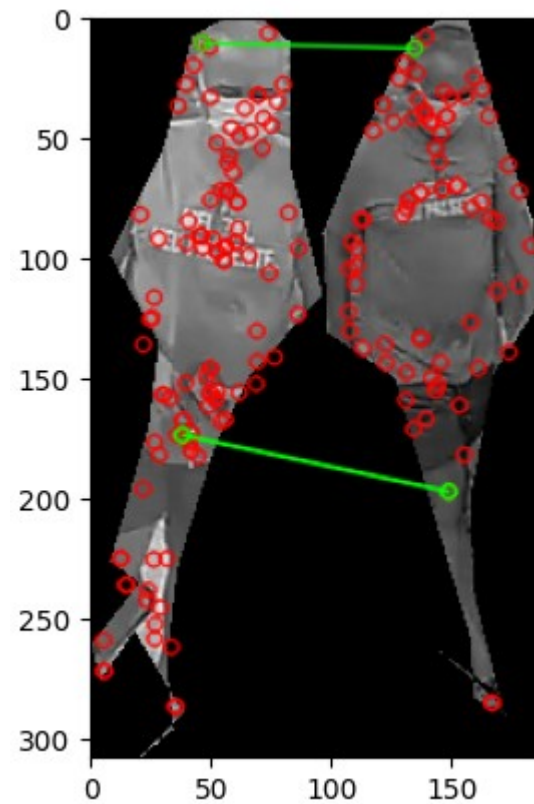
Biblioteca FLANN (checks=1)



Biblioteca FLANN (checks=50)



Biblioteca FLANN (checks=100)



O número em checks implica em quantidade de recursões. Quanto maior o número, mais preciso o algoritmo fica.

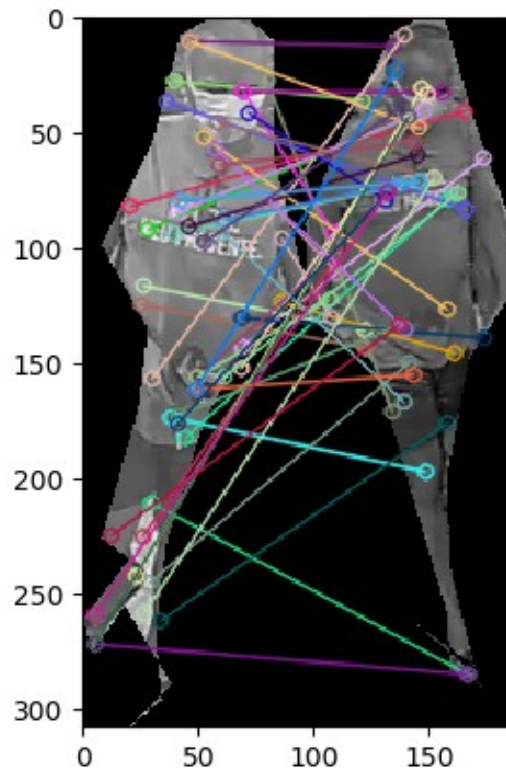
Outra linha de pesquisa forneceu uma solução baseado no histograma das imagens. Isso foi no site 'GeeksforGeeks' no link: "<https://www.geeksforgeeks.org/measure-similarity-between-images-using-python-opencv/>". O código apresenta um valor de diferença entre as imagens. Então é possível usar um valor limite aproximado que indica que uma pessoa é diferente da outra. O valor que coloquei foi 5000. Se o for menor que isso, é provável que seja a mesma pessoa.

Outro link de pesquisa "<https://www.pyimagesearch.com/2017/06/19/image-difference-with-opencv-and-python/>" destaca as partes da imagem que são diferentes. O problema é que as imagens precisam ter o mesmo tamanho e formato. Então não funciona para comparar as imagens recortadas de cada pessoa.



Os dois ângulos (Câmera 01 e câmera 02) mostram imagens totalmente diferentes, com exceção da data, parte do horário e nome da câmera. A pessoa alvo está em posições diferentes nas imagens. Então essa semelhança não é considerada. Seria necessário recortar todas as pessoas em uma imagem com o mesmo tamanho e compará-las.

Em outra pesquisa, no site "<https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/>" foi alcançado um resultado semelhante à força bruta e à biblioteca FLANN. O autor do site mostra a técnica SIFT. O resultado possui várias linhas se cruzando. Isso mostra que o algoritmo está confuso ao identificar pontos similares.



Conclusão: O melhor algoritmo, por enquanto, foi do site “<https://www.geeksforgeeks.org/measure-similarity-between-images-using-python-opencv/>”, que utiliza um histograma e calcula a distância euclidiana das imagens e fornece um valor flutuante. Sendo assim, pode ser possível fornecer um valor limite para definir se uma pessoa é ou não igual a outra.