2D Image Processing

Open-set still-to-video face identification project

Report

Ivan Zherebiatnikov

December 29, 2021

During the work on this program, many different algorithms were used.

1. **Tracking.** In the version of OpenCV that I worked with, only three types of trackers worked out of the box. They are KCF, CSRT, and MIL. However, all trackers except KCF are very unreliable. For example, when camera changes, they still somehow track the face from the previous frame, but there already isn't one on the current frame. Taking that into consideration, I decided to choose KCF tracker. However, you can still change the type of tracker by specifying `tracker_type` argument in `FaceRecognition()` function in "main.cpp".
2. **Face detection**. I tried to use Haar cascade classifiers that come with OpenCV and also dlib face detector. "haarcascade_frontalface_alt.xml" and "haarcascade_frontalface_alt2.xml" work the best, and I decided to choose the latter one.
3. **Face recognition**. I programmed many approaches. The first one is using BoW method with different classifiers: Bayesian Classifier, Logistic Regression and SVM Classifier. However, the results seemed completely random so I had to give on that idea. The second one is to calculate face landmarks and use pre-trained face recognition model from dlib library that tells whether the person on two images is the same or not. It works much slower, yet the results are still random. So I decided to use 2NN method to find good matches between the descriptors from the videoframe and the descriptors from the database. The only somewhat reliable metric turned out to be the number of good matches. I also tried total and average distance between good matches. However, the results were again really, really bad.

Thus I had to finetune two hyperparameters: the number of good matches that guarantees that the person is present in the database and the parameter of ratio between the distances to the closest and the second-closest neighbor in 2NN algorithm.

Since there are way too many metrics too look after, I had to choose what values of this hyperparameters work the best by using my own subjective judgment. These values turned out to be 9 and 0.65 respectively.

There also arose an issue with evaluating the quality metrics for face detection and tracking. We are required to calculate FPR, however, in this problem the number of true negatives cannot be defined. So instead I decided to calculate precision. For the same metrics for evaluating face recognition for every class I calculated the metrics for the binary classification problem, where the person from this class is a positive sample and all other people are negative samples.

The following is the direct copy of statistics calculated when running the command "build\bin\Release\faceID", which processes all videos available in the "test" directory:

DETECTION AND TRACKING STATISTICS

Precision - 0.996671.
Recall - 0.991113.
FNR - 0.00888657.

########## STATISTICS ON CLASSES ##########

Christian Bale:
Precision - 0.998047.
Recall - 0.998047.
FNR - 0.00195312.

Harrison Ford:
Precision - 0.927386.
Recall - 0.995546.
FNR - 0.00445434.

Jackie Chan:
Precision - 0.796247.
Recall - 1.
FNR - 0.

Linus Sebastian:
Precision - 0.735484.
Recall - 0.895288.
FNR - 0.104712.

Marques Brownlee:
Precision - 1.
Recall - 0.898438.
FNR - 0.101562.

Ryan Gosling:
Precision - 0.601156.
Recall - 0.984227.
FNR - 0.0157729.

Steve Harvey:
Precision - 1.
Recall - 0.757098.
FNR - 0.242902.

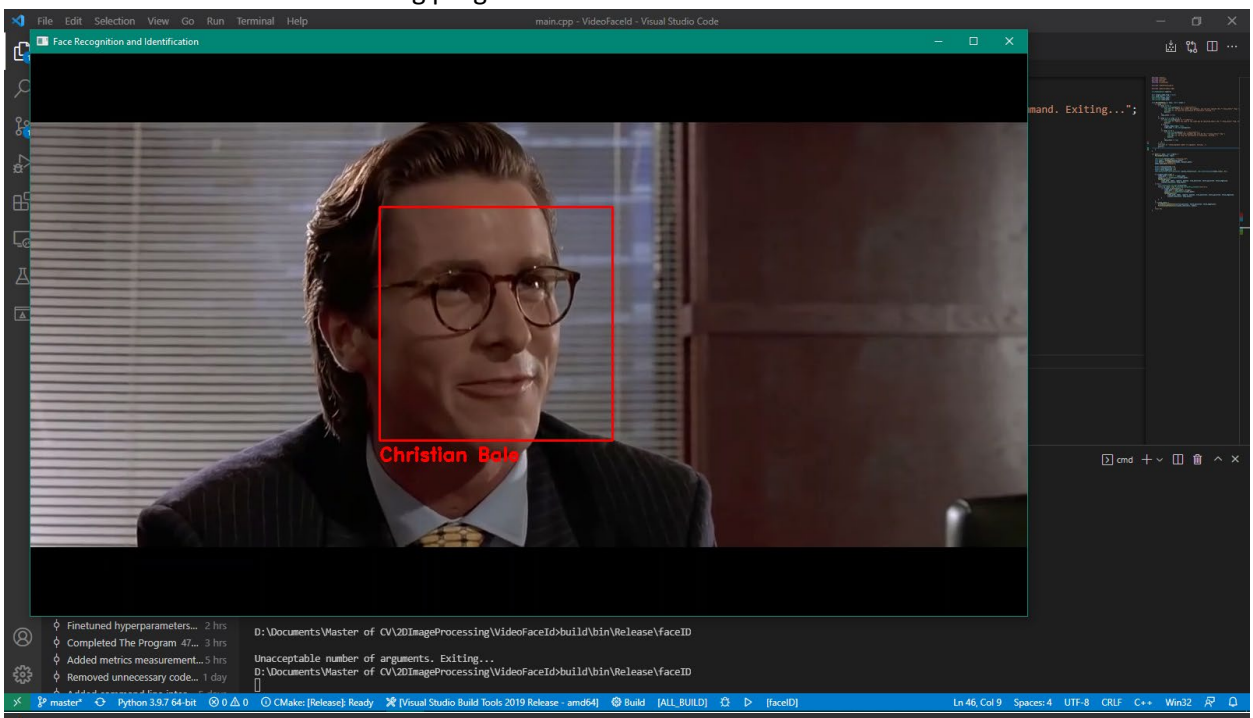
Tobey Maguire:
Precision - 1.
Recall - 0.985876.
FNR - 0.0141243.

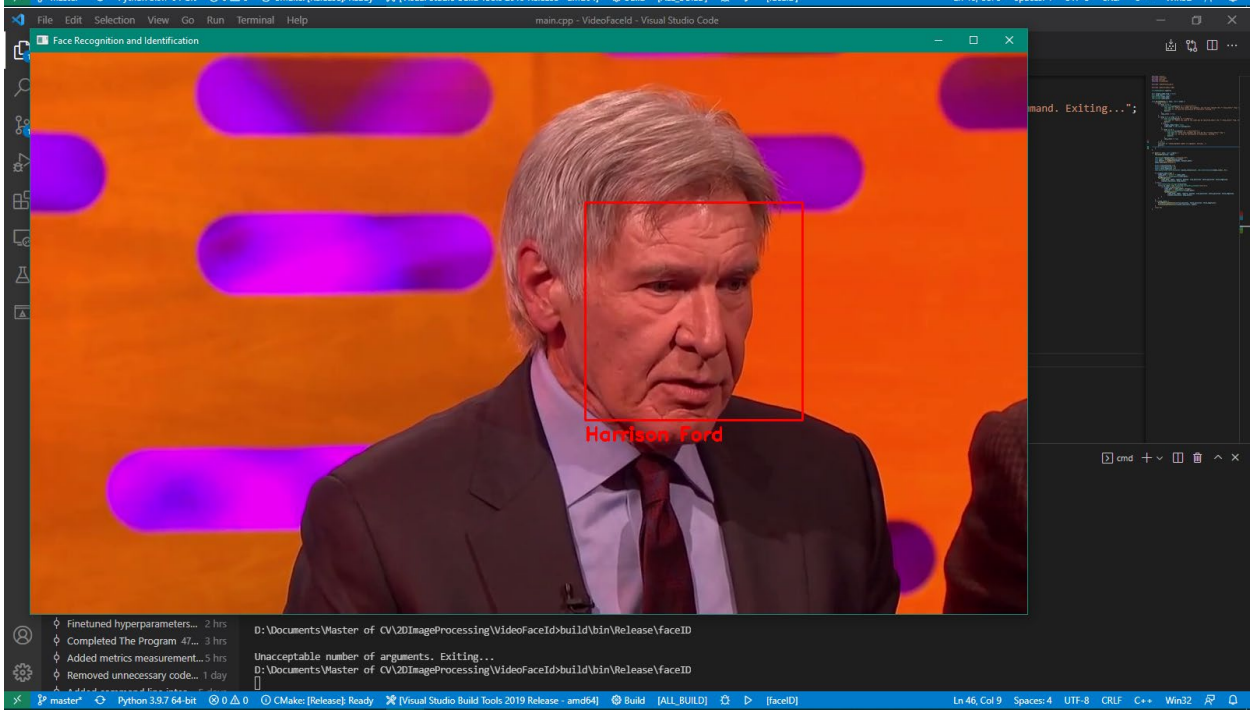
Will Smith:
Precision - 0.540373.
Recall - 1.
FNR - 0.

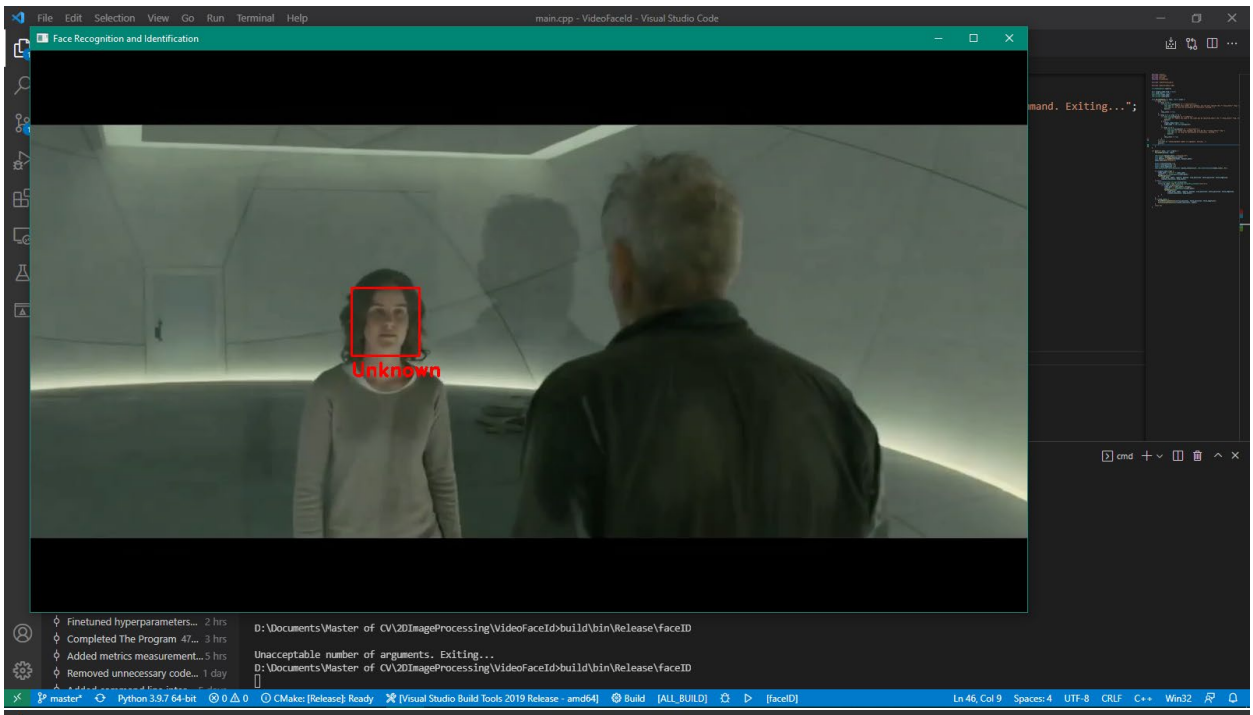
William Dafoe:
Precision - 1.
Recall - 1.
FNR - 0.

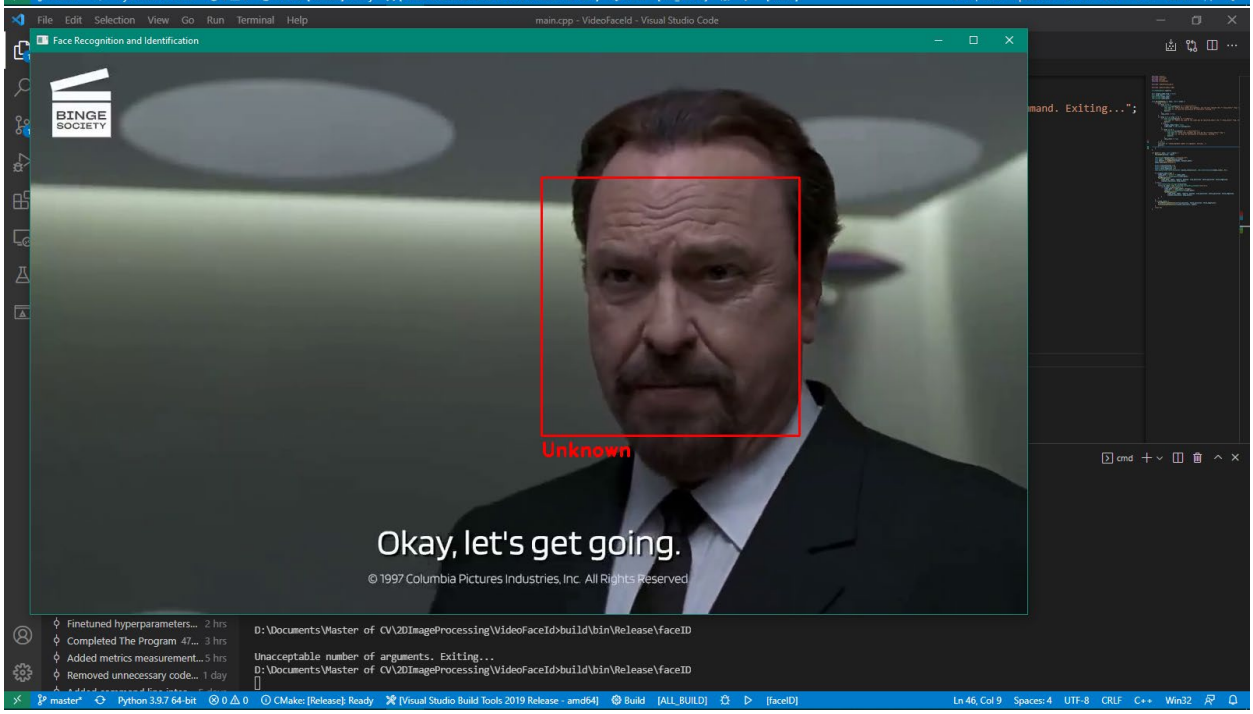
Unknown:
Precision - 0.959404.
Recall - 0.778241.
FNR - 0.221759.

Screenshots of the working program:






File Edit Selection View Go Run Terminal Help
main.cpp - VideoFaceId - Visual Studio Code
Face Recognition and Identification
Unknown
mand. Exiting...";
cmd
Finetuned hyperparameters... 2 hrs
Completed The Program 47... 3 hrs
Added metrics measurement... 5 hrs
Removed unnecessary code... 1 day
D:\Documents\Master of CV\2DImageProcessing\VideoFaceId>build\bin\Release\faceID
Unacceptable number of arguments. Exiting...
D:\Documents\Master of CV\2DImageProcessing\VideoFaceId>build\bin\Release\faceID
master* Python 3.9.7 64-bit 0 0 CMake: [Release]: Ready [Visual Studio Build Tools 2019 Release - amd64] Build [ALL_BUILD] [faceID]
Ln 46, Col 9 Spaces: 4 UTF-8 CRLF C++ Win32


File Edit Selection View Go Run Terminal Help
main.cpp - VideoFaceId - Visual Studio Code
Face Recognition and Identification
BINGE SOCIETY
Unknown
Okay, let's get going.
© 1997 Columbia Pictures Industries, Inc. All Rights Reserved.
mand. Exiting...";
cmd
Finetuned hyperparameters... 2 hrs
Completed The Program 47... 3 hrs
Added metrics measurement... 5 hrs
Removed unnecessary code... 1 day
D:\Documents\Master of CV\2DImageProcessing\VideoFaceId>build\bin\Release\faceID
Unacceptable number of arguments. Exiting...
D:\Documents\Master of CV\2DImageProcessing\VideoFaceId>build\bin\Release\faceID
master* Python 3.9.7 64-bit 0 0 CMake: [Release]: Ready [Visual Studio Build Tools 2019 Release - amd64] Build [ALL_BUILD] [faceID]
Ln 46, Col 9 Spaces: 4 UTF-8 CRLF C++ Win32