**Jetson nano Environment**

*Tài liệu này được thực hiện để hướng dẫn cách xây dựng môi trường cho jetson nano phục vụ việc lập trình.(Các thành phần trong tài liệu này được liệt kê dựa trên thời điểm tài liệu này được tạo ra-15/12/2020- và có thể thay đổi trong tương lai).*

1. **Chuẩn bị:**

Jetson Nano Developer Kit: B01 hoặc A02 tùy theo nhu cầu sử dụng.

Màn hình LCD: Màn Hình Waveshare 7 Inch HDMI Capacitive Touch Screen LCD (B). Có thể sử dụng các màn hình LCD có sẵn khác với giao tiếp HDMI.

1 module wifi để kết nối wifi: có thể sử dụng module USB hoặc module “Intel Wireless AC8265”.

Nguồn: Nguồn Power Adaptor AC-DC 5V 5A. Có thể dùng nguồn khác tuy nhiên lưu ý về dòng và chế độ cung cấp điện của jetson nano.

Thẻ micro: khuyến khích từ 32GB trở lên.

1 Đầu đọc thẻ nhớ

Chuột và phím kết nối(cái này cũng không bắt buộc lắm vì mình thường dùng remote desktop để làm việc trên này)

1. **Thực hiện cài hệ điều hành cho jetson nano:**

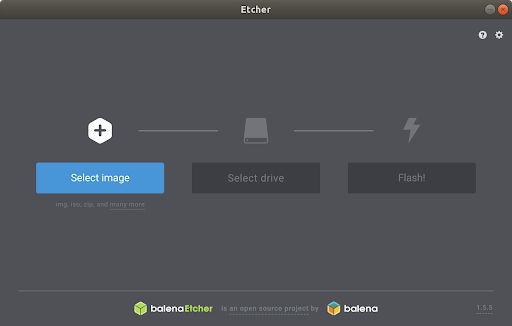
Download hệ điều hành từ trang chủ jetson nano tại đường dẫn bên dưới

<https://developer.nvidia.com/embedded/downloads>

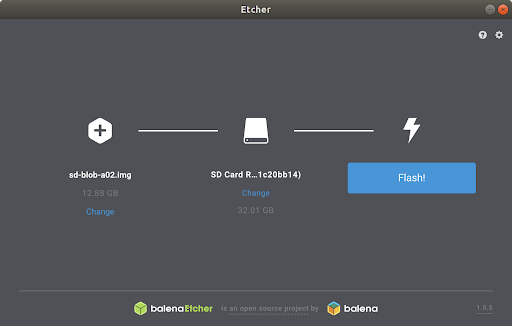
Download phần mềm ghi thẻ nhớ tại đây

<https://www.balena.io/etcher/>

Giao diện của chương trình ghi thẻ:

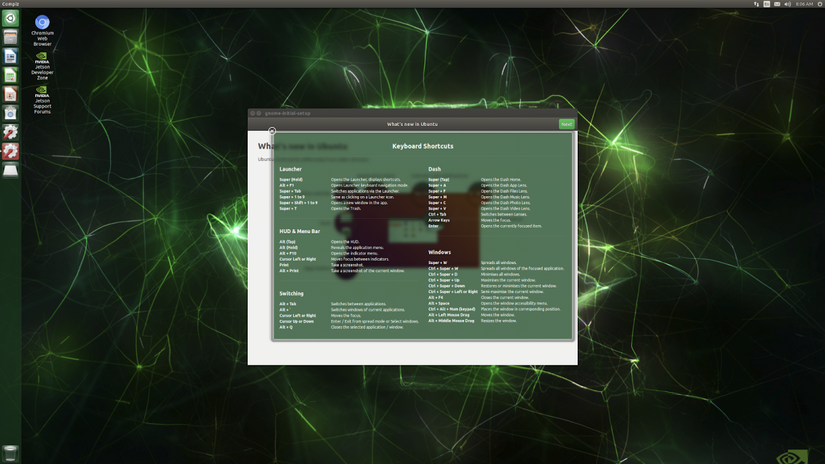


Ghi hệ điều hành vừa cài vào thẻ nhớ bằng phần mềm ghi thẻ. Chọn select image, chỉ đường dẫn đến file img vừa tải về từ trang chủ jetson nano và chọn Flash. Chờ đến khi Flash xong thì rút thẻ nhớ ra. Lưu ý, thẻ sau khi Flash xong sẽ không hiện trên giao diện Windows explore nữa.



Cắm thẻ vào khe thẻ nhớ trên jetson nano và cắm nguồn để khởi động cho lần đầu tiên. Lưu ý, ở lần khởi động đầu tiên, chúng ta nên kết nối đầy đủ màn hình, chuột và bàn phím để thao tác.

Trong lần khởi động đầu tiên, jetson nano sẽ yêu cầu xác thực điều khoản, cài đặt múi giờ, tên user, và kiểu bàn phím,… Sau khi hoàn tất chúng ta sẽ có giao diện như thế này.



Vậy là đã cài xong hệ điều hành cho jetson nano.

Tài liệu cài hệ điều hành này có sử dụng ảnh và tham khảo từ trang: <https://viblo.asia>

Tại đường dẫn: <https://viblo.asia/p/huong-dan-cai-dat-moi-truong-cho-jetson-nano-phuc-vu-nghien-cuu-machine-learning-3Q75w8VBKWb>

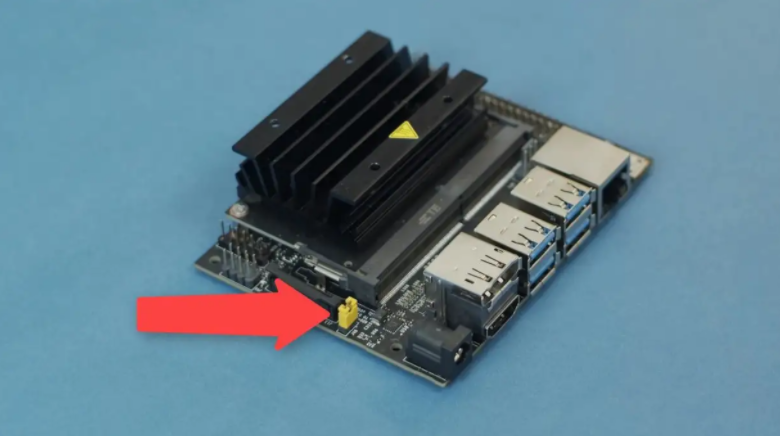
1. **Một số thiết lập:**

Thiết lập nguồn:

Việc này sẽ cần thiết khi chúng ta sử dụng jetson nano ở các dự án cần tiết kiệm điện nếu không thì chúng ta có thể bỏ qua bước này do ban đầu chúng ta đã sử dụng nguồn chuyên dụng cho jetson nano.

Jetson nano hỗ trợ 2 chế độ nguồn điện:

10W: **sudo nvpmodel -m 0**. (Chế độ này được thiết lập mặc định. Lưu ý, ở chế độ này 2 chân J48 phải được nối với nhau bằng 1 jumper thường thì khi mua jetson nano sẽ có sẵn jumper này tuy nhiên cũng nên kiểm tra kĩ như hình bên dưới).



5W: **sudo nvpmodel -m 1**. (Cho các dự án cần tiết kiệm điện)

Cài desktop nhẹ hơn:

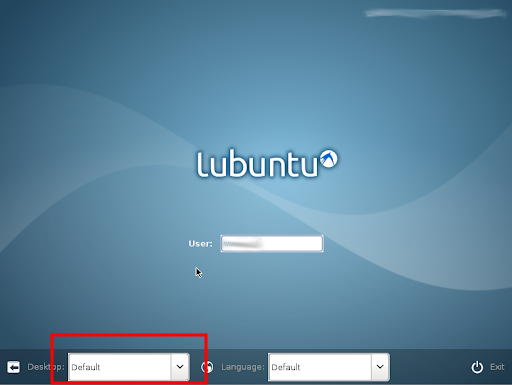
Desktop hiện tại của jetson nano dùng rất sướng, tuy nhiên nếu chúng ta không làm việc thường xuyên trên desktop này thì có thể cài desktop khác nhẹ hơn hoặc gỡ luôn cũng được, để tận dụng ram cho chuyện khác-mấy cái sướng sướng thường tốn kém mà.

Mở terminal gõ:

**sudo apt install lxde lxdm**

Các bạn có thể nhìn thấy hộp yêu cầu lựa chọn display manager gdm3 hay lxdm. Chọn lxdm nhé.

Tiếp theo bạn cần khởi động lại máy tính. Sau khi khởi động lại bạn sẽ nhìn thấy giao diện đăng nhập như hình dưới. Chọn LXDE ở phần Desktop và đăng nhập.



Riêng mình nhận thấy giao diện này khá gần gủi với windows.

Gỡ desktop cũ (Không bắt buộc). Bạn có thể gỡ Ubuntu Desktop và gdm3 bằng các câu lệnh sau.

1. **jtop - Xem trạng thái hệ thống**

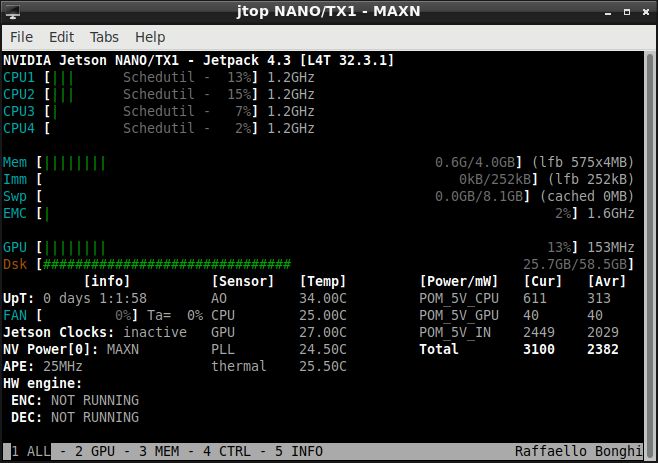
cài đặt pip

**sudo apt-get install python-pip**

**sudo -H pip install -U jetson-stats**

sử dụng

**sudo jtop**



1. **Cài đặt putty để thực hiện thao tác dòng lệnh từ máy khác.**

Trên máy linux cài đặt putty bằng lệnh:

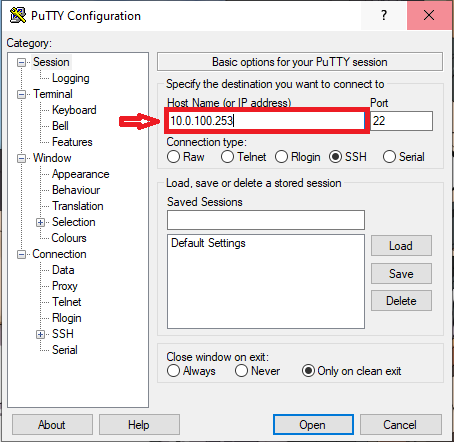
**sudo apt install putty**

Trên máy window download và cài đặt putty từ đường dẫn : <http://www.putty.org/>

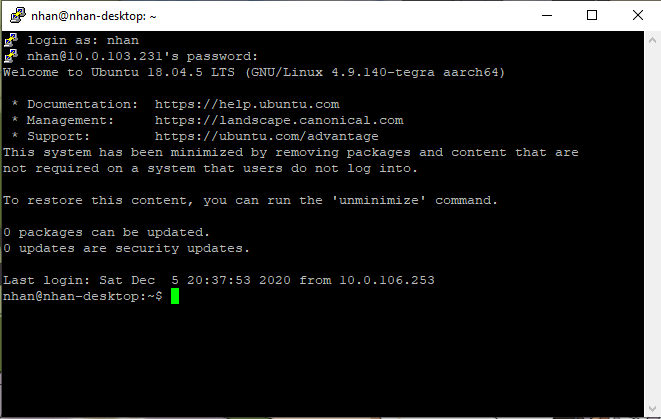
Trên jetson nano lấy ip bằng lệnh:

**Ifconfig**

Mở putty trên windows ta được giao diện như hình dưới:



Nhập ip của máy jetson nano và đăng nhập vào ta sẽ được giao diện sau.



Từ đây ta có thể sử dụng jetson nano từ giao diện dòng lệnh của putty.

1. **Sử dụng remote desktop của Windows.**

Cài đặt RDP trên máy jetson nano:

Update hệ thống bằng lệnh:

**sudo apt-get update**

Cài XRDP bằng lệnh:

**sudo apt-get install xrdp**

Cài giao diện đồ họa xfce4:

**sudo apt-get install xfce4**

**sudo apt-get install xfce4-terminal**

Cấu hình để chạy RDP trên jetson nano:

**cd /etc/xrdp**

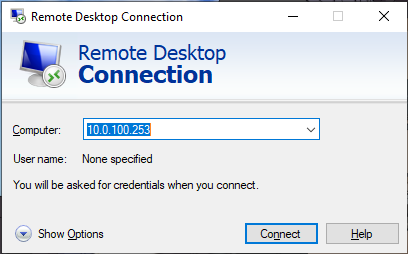
**echo xfce4-session > ~/.xsession**

khởi động lại XRDP:

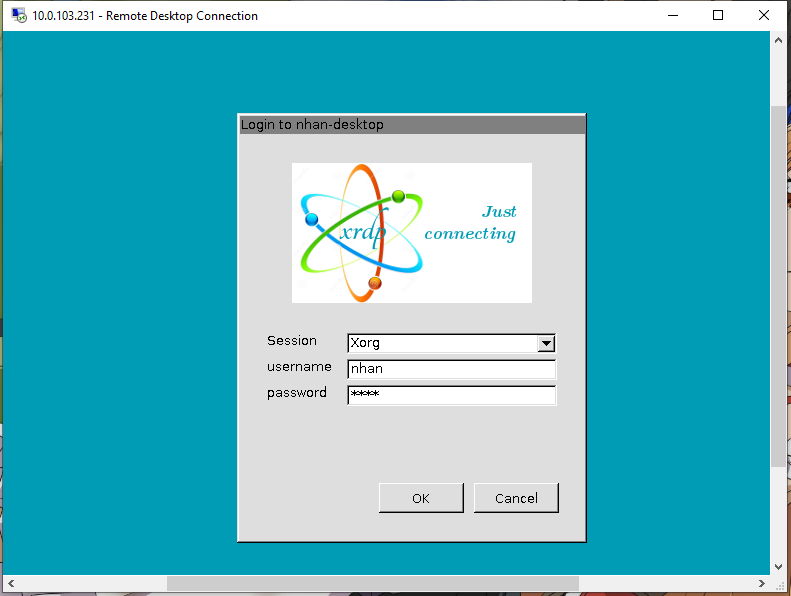
**service xrdp restart**

Lấy ip máy jetson nano bằng lệnh ifconfig.

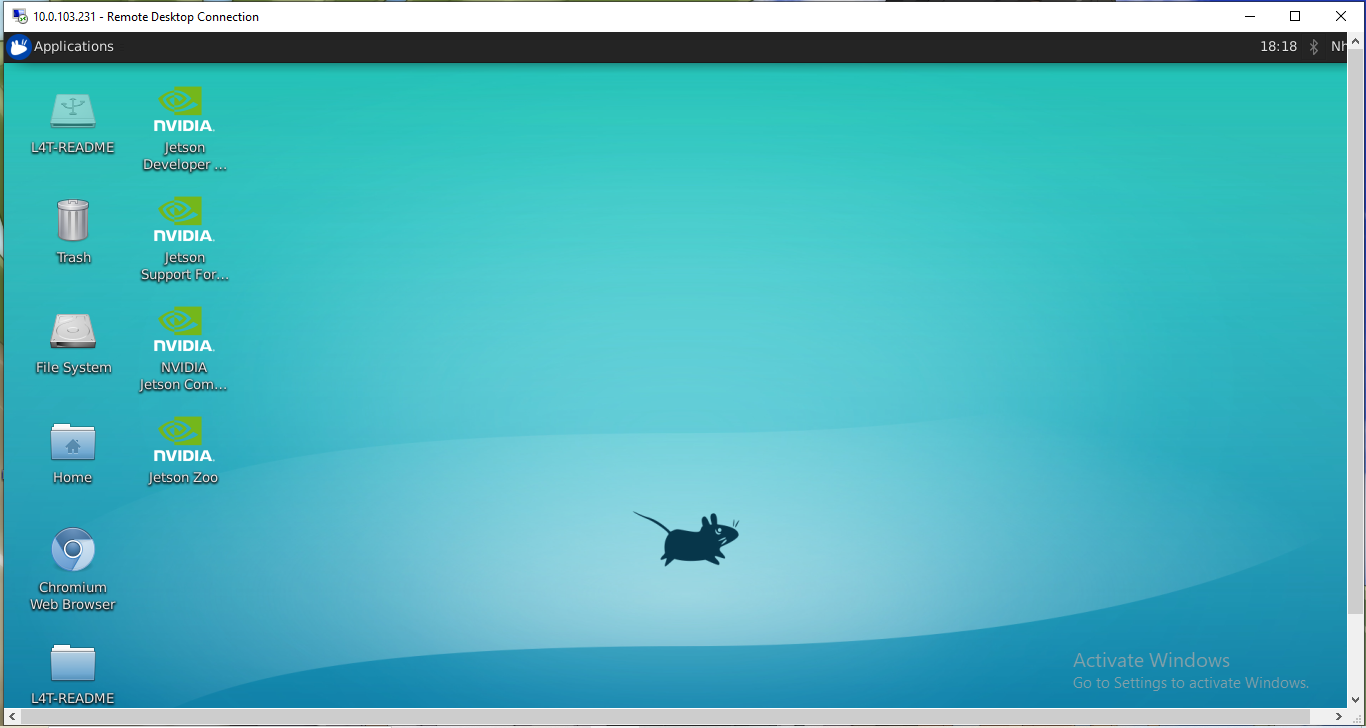
Mở Remote desktop trên Windows và nhập ip của máy jetson nano vào.



Sau đó sẽ xuất hiện cửa sổ đăng nhập như sau:



Nhập username và password của máy jetson nano, chúng ta sẽ có giao diện như thế này:



Từ đây chúng ta có thể thao tác trên jetson nano từ máy khác. Điều này có thể giúp chúng ta có thể tương tác với jetson trong cùng môi trường làm việc quen thuộc.

Lưu ý: Jetson nano và máy trạm phải được kết nối cùng 1 lớp mạng lan có thể là mạng dây hoặc wifi.

1. **Cài đặt các thư viện cần thiết cho lập trình:**

Opencv:

Opencv là 1 thư viện mã nguồn mở lập trình thị giác máy tính. Hỗ trợ các hàm xử lý hình ảnh.

Ở gói jetpack 4.4.1(hệ điều hành mà chúng ta Flash vào jetson nano lúc đầu) đã tích hợp sẵn opencv 4.1.0 nên chúng ta chỉ cần gọi ra và sử dụng thôi.

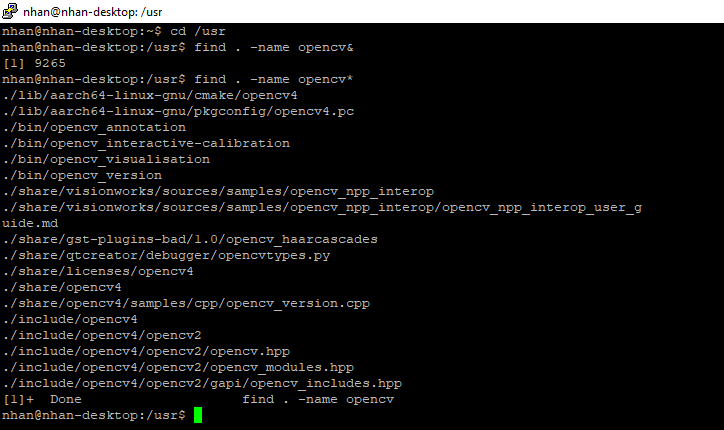
Để kiểm tra opencv đã được cài đặt chưa chúng ta thực hiện những bước sau:

Chuyển đến thưc mục usr:

**cd /usr**

Tìm tìm những thứ có liên quan đến opencv:

**find . -name opencv\***



Trong thư mục include hiện đang có opencv4:

Để sử dụng thư viện opencv cho source chúng ta chỉ cần include các header file cần thiết và link và cmake như sau.

Vd trong source cần sử dụng:

#include <opencv2/opencv.hpp>

Và trong file CMakeLists.txt chúng ta liên kết thư viện với file nguồn như sau:

cmake\_minimum\_required(VERSION 3.9)

set(PROJECT\_NAME TestOpenCVCmake)

project(${PROJECT\_NAME})

set(CMAKE\_CXX\_STANDARD 11)

find\_package(OpenCV REQUIRED)

include\_directories(${OpenCV\_INCLUDE\_DIRS})

add\_executable(face\_rec dnn\_face\_recognition\_ex.cpp)

target\_link\_libraries(face\_rec ${OpenCV\_LIBS} )

Với dnn\_face\_recognition\_ex.cpp là file .cpp mà chúng ta lập trình. face\_rec là tên file thực thi sau khi build, các config khác là thuộc Cmake.

Hoặc nếu các bạn muốn cài 1 phiên bản mới hơn của opencv có thể làm theo hướng dẫn từ trang:

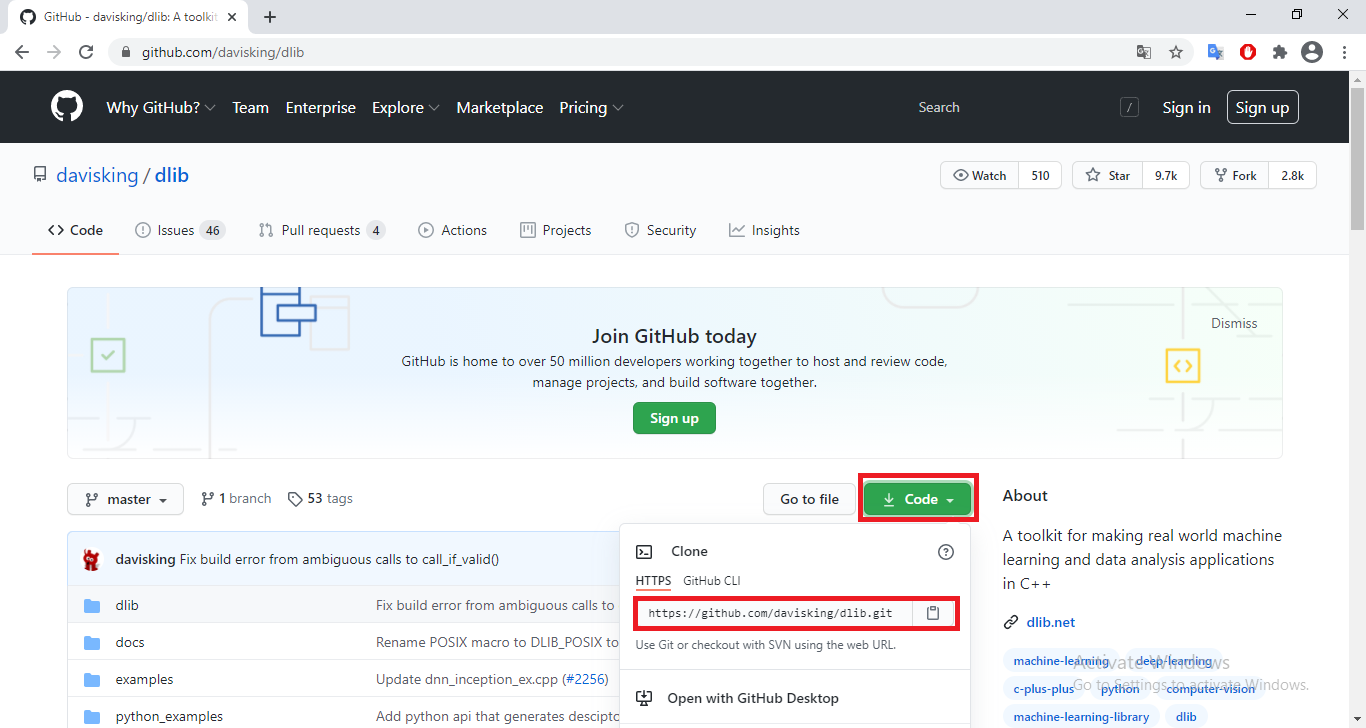
<https://pivietnam.com.vn/huong-dan-cai-dat-thu-vien-opencv-va-tensorflow-tren-he-dieu-hanh-ubuntu-jetson-nano-pivietnam-com-vn.html>

Dlib và ncnn:

Dlib là 1 thư viện mã nguồn mở được thực hiện bởi Davisking. Các hàm hỗ trợ tính toán machine leaning và computer science.

Với các thư viện ncnn và dlib thì chúng ta có thể tải về từ trang github của nhà phat hành như sau.

Lấy đường dẫn clone từ trang git



Bằng giao diện dòng lệnh gõ:

**git clone** [**https://github.com/davisking/dlib.git**](https://github.com/davisking/dlib.git)

Chúng ta có thể để soure này ở bất cứ đâu nhưng khuyến khích để chung với project của chúng ta

Build và thêm Dlib vào source.

Thêm các header file cần thiết sau:

#include <dlib/dnn.h>

Trong file cmake cùng cấp với source chúng ta cấu hình như sau:

cmake\_minimum\_required(VERSION 3.9)

set(PROJECT\_NAME TestOpenCVCmake)

project(${PROJECT\_NAME})

set(CMAKE\_CXX\_STANDARD 11)

add\_subdirectory(/home/nhan/dlib/dlib dlib\_build)

add\_executable(main test\_utraFace.cpp)

target\_link\_libraries(main dlib::dlib)

Với test\_utraFace.cpp là file source của chúng ta, main là tên file thực thi sẽ sinh ra. /home/nhan/**dlib**/**dlib** là đường dẫn tới thư mục dlib **bên trong** thư mục dlib mà chúng ta clone từ git về (lưu ý là có 2 thư mục dlib cha và con. Trong trường hợp chúng ta tải trực tiếp từ trang dlib.org thì sẽ là /home/nhan/**dlib.xx.xx**/**dlib**, 2 cặp số xx.xx là phiên bản).

Sau đó chúng ta build source bằng lệnh cmake:

cmake .

make .