Steps to building OpenFace library for linux using the dpkg package manager. Please be aware that the first step is only needed because of the limited ram capacity of the raspberry pi thus perform it only if you are doing this in a machine that has a small amount of ram such as 1GB. The following step will increase the swap size allowed between ram and storage. Furthermore you may want to change the swap size back after you are done as it will allow for more data to be written to the storage and if you are using a micro sd, that is not such a good thing.

- The steps below explain the procedure, all the information was taken from: https://www.pyimagesearch.com/2017/05/01/install-dlib-raspberry-pi/
    - `sudo nano /etc/dphys-swapfile`
    - CONF_SWAPSIZE=100 => CONF_SWAPSIZE=1024; and save the changed file (Ctrl + O, Ctrl + X , Y, Enter)
    - `sudo /etc/init.d/dphys-swapfile stop`
    - `sudo /etc/init.d/dphys-swapfile start`
- `sudo apt-get update --fix-missing`
- `sudo apt-get upgrade`
- `sudo apt-get install build-essential cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev libboost-all-dev libtbb-dev libopenblas-dev libeigen3-dev default-jdk ant`
- check if python2 is installed
    - call python or python2, if you call the prior then check the python version
- install pip2 like this:
    - `wget https://bootstrap.pypa.io/get-pip.py`
    - `sudo python2 get-pip.py`
- with pip2 install the following:
    - `sudo pip2 uninstall numpy`
    - `sudo pip2 install numpy`
    - `sudo pip2 install pandas`
    - `sudo pip2 install scipy`
    - `sudo pip2 install scikit-learn`
    - `sudo pip2 install scikit-image`
- Now we build and install OpenCv, these steps were taken from the official page of OpenCv on how to build it:
    - First step includes cloning opencv to your working directory, do not forget to delete this after you install it since the build will take a lot of space:
        - `cd ~/<my_working _directory>`
        - `git clone https://github.com/opencv/opencv.git`
    - The next step is to build OpenCv from source using make
        - `cd opencv`
        - `mkdir release`
        - `cd release`
        - `cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..`
        - `make`
        - `sudo make install`

- You need to get and build dlib 18.16 as per direction from openface page
    - `wget http://dlib.net/files/dlib-18.16.tar.bz2`
    - `tar xvf dlib-18.16.tar.bz2`
    - `cd dlib-18.16/python_examples`
    - `mkdir build`
    - `cd build`
    - `cmake ../../tools/python`
    - `cmake --build . --config Release`
    - `sudo cp dlib.so /usr/local/lib/python2.7/dist-packages`
- Next you can test if OpenCv and dlib have been installed correctly by importing them in python and simply running an empty code with the module imports only to see if python can find them.
    - import cv2
    - import dlib
- The next step is to install torch on our system. These steps were taken directly from the torch website, thus for more information got to www.torch.ch .
    - `git clone https://github.com/torch/distro.git ~/torch --recursive`
    - `cd ~/torch`
    - `bash install-deps`
    - `./install.sh`
    - `source ~/.bashrc`
- Through the installation process of torch luarocks was also installed, thus we will use that to install several other packages
    - `luarocks install dpnn`
    - `luarocks install nn`
    - `luarocks install optim`
    - `luarocks install csvigo`
    - There are other packets however they are needed for training the DNN which will not be used, however if needed check the OpenFace page mentioned in the ReadMe
- The next step and last is to get the OpenFace library and install it
    - `git clone https://github.com/cmusatyalab/openface.git`
    - `cd openface`
    - `sudo python2 setup.py install`
    - `./models/get-models.sh`
- Apperantely there is a problem with ARM processors and the model thus you need to do some changes. You can follow this issue (https://github.com/cmusatyalab/openface/issues/42) and use their ASCII model.