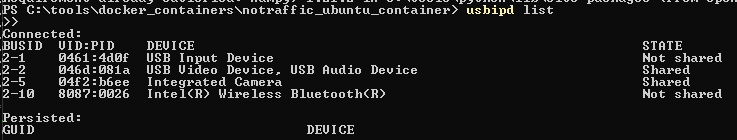
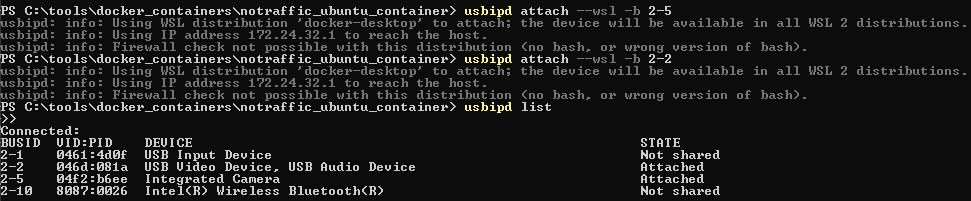
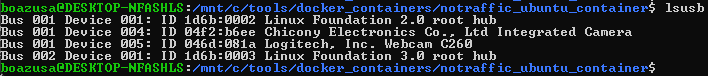
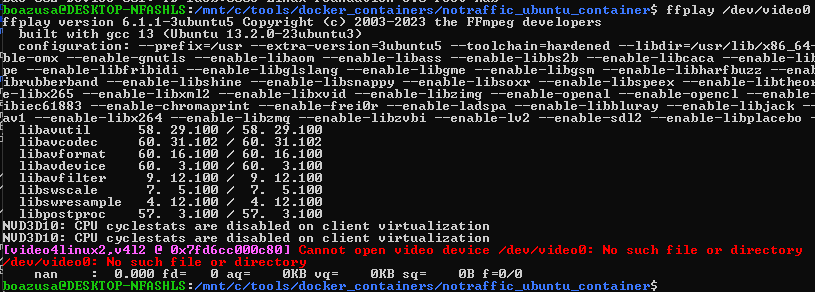
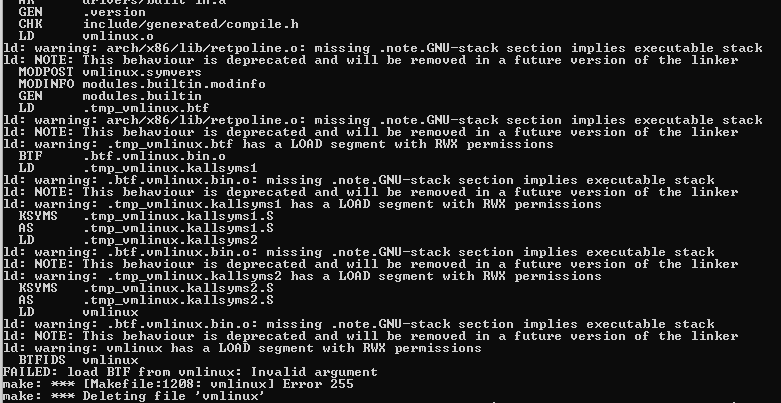
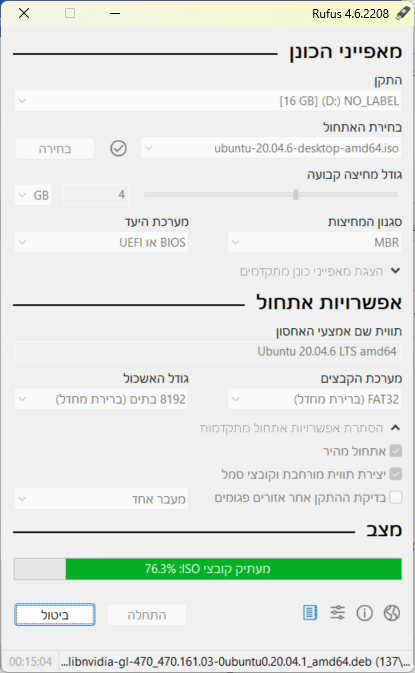
Technological steps tried to access cameras from wsl on windows machine:

1. usbipd & lsusb bind method
   1. Installed 'usbipd-win-4.4.0' on windows to enable cameras to be viewed in wsl (Windows Subsystem for Linux).
      1. Attached and shared them on windows  
         Before:  
           
         bind:  
         

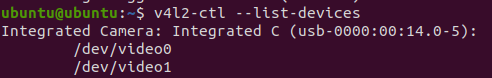
attach (final step):  


* 1. Installed 'lsusb' on wsl.
     1. Verify they are viewed in wsl:  
        
  2. However, this is not enough for having the cameras viewed as '/dev/video\*'  
     

1. Modifying the wsl kernel to startup the cameras drivers within the wsl startup.
   1. followed the ['Linux Tips - Record Video from USB Camera in WSL (2022)](https://www.youtube.com/watch?v=t_YnACEPmrM)' tutorial, which seems to work on the video, but it failed at the vmlinux image.  
      failed at:  
      
   2. the result was that no '/dev/video\*' showed up as well.
2. Running windows script from wsl workaround.
   1. Installing 'opencv-python' for cameras control
   2. Launching cmd (windows command-line) inside wsl, using the '/mnt/c/Windows/System32/cmd.exe /C' command, enabled powering up the camera.
      1. Cons: it is not really ubuntu script that runs the cameras, it is 'Ubuntu powered windows-python'.
      2. Cannot be ran from inside a container that is built in wsl (even when I mounted the wsl path to the container).  
           
         (directly from wsl)
3. Run Ubuntu from USB (without installation, on my Windows PC).
   1. Download Ubuntu 20.04 from <https://releases.ubuntu.com/20.04/ubuntu-20.04.6-desktop-amd64.iso>
   2. Download rufus from <https://rufus.ie/he/>
   3. Create bootable Ubuntu 20.04 USB
   4. Reboot the PC while pressing the F12 key while startup.
      1. Select "Try Ubuntu" option
   5. Install Pyton and testing libraries (pytest, v4l2-ctl)  
      
   6. after launching Ubuntu (‘try Ubuntu’ version), I installed v4l-utils (sudo apt install v4l-utils), and I was able to view the camera as a device (v4l2-ctl --list-devices).
   7. First draft of python controlled webcam seems to be working.

Generating high-level design document that defines the testing procedure:

4-f



4-g

