**testing process of the Camera device of a laptop**

1. **Introduction**:  
   This document describes the design of PC webcam test assignment for Test Automation Engineer position at Notraffic. The objective of this test is to demonstrate technical abilities in programming, designing, and requirements testing coverage, and process automation.
2. **Scope**:

test setup:

* 1. Generate requirements
  2. Design test setup
  3. Create test setup (HW and SW)

testing will be focusing on:

* Sanity, basic functionality
* Performance
* requirements and edge cases
* Quality and user experience

1. **Objectives**:

* Verify Ubuntu OS compatibility with Lenovo ideaPad 5 15IIL05 model webcam ([Lenovo IdeaPad\_5\_15IIL05 specs](https://psref.lenovo.com/syspool/Sys/PDF/IdeaPad/IdeaPad_5_15IIL05/IdeaPad_5_15IIL05_Spec.PDF)).
* Smooth user experience usage
* Common apps compatibility

1. **Implementation Milestones**:
   1. Verify [defined] system requirements
   2. Design lab setup
   3. Test setup initialization and validation
      1. See ' *Test\_Automation\_Engineer\_Assignment\_Technical\_Development.docx*' for setup development.
   4. Setup development (Ubuntu 20.04 OS PC)
      1. Verify Python and drivers installations
      2. Verify communication with webcams (built-in and usb connected)
   5. Writing test automation code
      1. Initial daft test code
         1. Sanity tests for both camera and python
      2. Adding advanced tests based on requirements
      3. Code review and improvements
2. **Test Cases**:
3. Functional (sanity) Tests:
   1. Verify video device is available
   2. Verify driver version
      1. Verify driver can be upgraded/downgraded
   3. Screenshot capturing
   4. Video recording
   5. Live video stream
   6. Save recording to local machine with supported file formats
4. Performance Tests:
   1. Image pixel resolution with different settings
   2. Image brightness/darkness settings
   3. Image quality
   4. Recording latency
      1. Verify video length is as expected
      2. Verify image taken within expected delay
   5. Video/image saving in all supported formats
   6. Common apps compatibility (Teams, Slack, Google Meet, Facebook, etc.)
5. Color, sharpness:
   1. Video fps rate (packets loss)
   2. Video and image sharpness
   3. Video and image pixels
      1. Test all supported resolutions
6. Camera security
   1. Camera privacy shutter coverage (under different lighting conditions, include sunlight and flashes)
   2. Settings permissions
      1. Test modifications as admin
      2. Test modifications as limited-access user
7. Edge cases Tests (based on camera specs):
   1. Max resolution pixel limit
      1. Test for different parameters; brightness, blurriness, contrast, saturation, size, fps, hue
   2. Min resolution pixel limit
      1. Test for different parameters; brightness, blurriness, contrast, saturation, size, fps, hue
   3. Max screenshots per sec
   4. Long-term video usage
   5. Camera exposure (under different lighting conditions)
8. Negative testing, intended failures
   1. Over resolution pixel limit
   2. Unsupported pixels resolutions
      1. Bellow min
      2. Above max
      3. Unsupported structures (ex 41x465px)
   3. Non-existing path file saving
   4. Unsupported file formats saving
   5. overload testing (running streaming/screenshot requests at the same time [w python threads])
9. **HW** performance and durability

* *Assuming these tests can be done using external camera, such that the computer [of the built-in webcam] is not affected*.
  1. Electrical measurements
     1. Voltage consumption
     2. current draw
     3. power usage
  2. Temperature tests
     1. Edge tests: min/max temp functionality
     2. Camera HW shape (material leak)
  3. Vibration (motion) tests
     1. Camera shall be functional during vibrations
     2. Camera shall be functional after vibrations
     3. Camera HW completion (cracks, breaks, lenses durability)
  4. Dust, humidity, moisture
     1. Corrosion test
  5. Conducted emissions test (FCC compliant)
     1. RF Power
     2. Performance under RF noise (from other RF Tx devices)
  6. Different USB connections performance
     1. USB 2.0
     2. USB 3.0
     3. USB hub (different types)

1. **Test setup**:

**HW**:

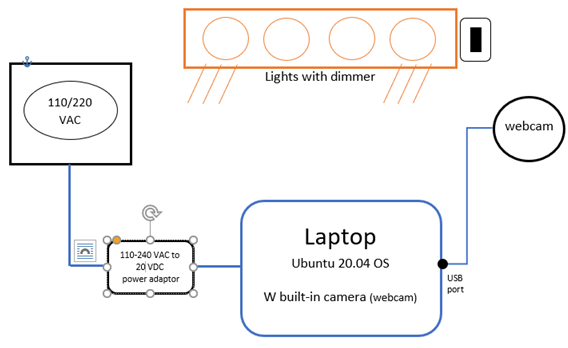
1. 220/110VAC supply
2. Electrical cables with 12V power adaptor for laptop computer
3. Ubuntu 20.04 laptop with built-in webcam and external usb-connected webcam.
4. Adjustable lighting conditions
   1. lights with dimmer
   2. daytime/night conditions
5. usb-connected webcam (if built-in webcam does not exist, such as desktop)

Optional:

1. 12VDC power supply for outdoor testing

Assuming usb-connected webcam, electronics lab equipment can be used for electrical measurements

1. DMM
2. Oscilloscope
3. Spectrum Analyzer
   1. Current probe
4. Signal (functional) generator; Appendix B



**SW**:

Ubuntu 20.04 installations:

* Webcam driver
* Docker
  + Ubuntu 20.04 Docker container with the following packages and apps:
    - Python 3.9 or later
      * Pycharm
      * Flask
    - v4l-utils
    - ffmpeg
    - ffprobe
    - fswebcam
    - gstreamer
* Optional installation
  + Pycharm
  + python3-distutils (for installing Python packages)

Test Execution Log:

\* Example for only the first (sanity) test that will be executed.

**Date**: 04/02/2025

**Test Environment** (Lab, computer, setup): Personal PC

**HW**: Integrated Camera: Integrated C.

**SW version**: xx.xx.xx

**Driver name & Version**: ‘uvcvideo’, 5.15.85

**Prerequisites**: see appendix A for apps and installations

**Test logs and images**: /home/ubuntu/Documents/test\_results\_02\_04\_2025\_\_12\_57\_30/

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case # | Test Name | Test objectives | Test Steps manual, (Pytest name for automated) | Test Date | Pass / Fail | Bug # (or N/A for Pass)  [link to Jira] | Comments |
| 1 | Capture\_frame | The camera shall capture image using Ubuntu Terminal request | 1) Open Ubuntu Terminal  2) run the command: “ffmpeg -f video4linux2 -i /dev/video0 -frames 1 /home/ubuntu/Documents/frame.jpg”  3) Navigate to ‘/home/ubuntu/Documents/’  4) Verify frame.jpg was created  5) Verify frame.jpg can be opened and the frame image is as expected. | 4/2/2025 | Pass | N/A |  |
| 2 |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |
| ... |  |  |  |  |  |  |  |
| 49 |  |  |  |  |  |  |  |
| 50 |  |  |  |  |  |  |  |

**Bug Example:**

**Bug #: 770**

**Title**: external webcam image created with unsupported resolution

**Description**: external webcam captured videos and images with unsupported pixel resolution also it was set to capture with supported resolution (with multiple file formats).

|  |  |
| --- | --- |
| Resolution set | Resolution captured |
| 320x180 | 320x176 |
| 424x240 | 432x240 |
| 848x480 | 864x480 |
| 960x540 | 960x544 |

**Date**: 04/02/2025

**Test Environment** Lenovo IdeaPad 5 15IIL05

**Prerequisites**: see appendix A for apps and installations

**Webcam HW**: UVC Camera (046d:081a)

**Webcam SW**: xx.xx.xx

**Driver name & Version**: ‘uvcvideo’, 5.15.85

**Test logs and images**: /home/ubuntu/Documents/test\_results\_02\_04\_2025\_\_12\_57\_30/

Steps to reproduce:

1. power up the Ubuntu computer
2. connect the UVC Camera via usb port
3. open Terminal
4. verify supported frame resolution supported. Type:  
   v4l2-ctl --list-formats-ext  
   *supported\_resolutions = ["1280x720", "320x180", "320x240", "352x288", "424x240", "640x360", "640x480", "848x480", "960x540"]*
5. navigate to a path where images can be saved
6. Capture image, type:   
   ffmpeg -f video4linux2 -video\_size 320x180 -i /dev/video2 -frames 1 filename.jpg
7. Verify image actual resolution, type:  
   ffmpeg -i filename.jpg 2>&1 | grep "Stream #0" | sed -n 's/.\*, \([0-9]\+x[0-9]\+\).\*/\1/p'
8. Output: 320x176, which is not equal to 320x180

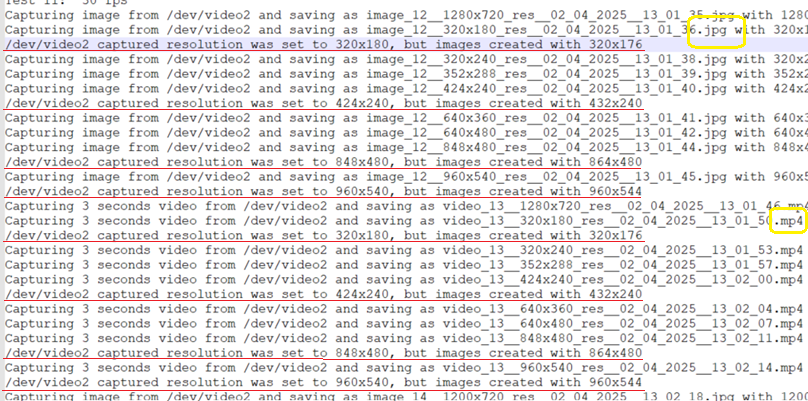
Acording to device specs, these (and few more) resolutions are supported

**Command**: *ffmpeg -f v4l2 -list\_formats all -i /dev/video2*

**Info**: 640x480 160x120 176x144 **320x176** 320x240 352x288 432x240 544x288 640x360 752x416 800x448 800x600 864x480 960x544 960x720 1024x576 1184x656 1280x720 1280x960

**ffmpeg** Ubuntu package UVC Camera do support its resolution specs, which are not listed in **v4l2-ctl** supported resolutions.

**Screenshots**:

  
*(lines 930-954 in report\_04\_02\_2025\_\_12\_57.txt report)*

**Attachments**:

* C:\Users\USER\Documents\NoTraffic\_Assignment\Results\test\_results\_02\_04\_2025\_\_12\_57\_30
* /home/ubuntu/Documents/test\_results\_02\_04\_2025\_\_12\_57\_30/

**Appendix A:**

Ubuntu 20.04 OS with the following installations/steps required:

sudo add-apt-repository universe

sudo apt update

sudo apt install v4l-utils

sudo apt install snap

sudo apt install python3-distutils

sudo apt install v4l-utils

sudo apt install ffmpeg

sudo apt install vlc -y

pycharm-community edition

**Appendix B:**

Function generator for powering up the camera outside the usb power; there are several useful scenarios for testing external usb powered cameras, such as Overvoltage, Voltage spikes (while inserting to the usb port).