RGB Camera Installation

Koushik K
Master's Thesis

December 2025

1 Hardware Required

Laptop with Windows, Luxonis Oak-D-Pro Wide Robotics Camera as in Fig 1 and an inbox cable.

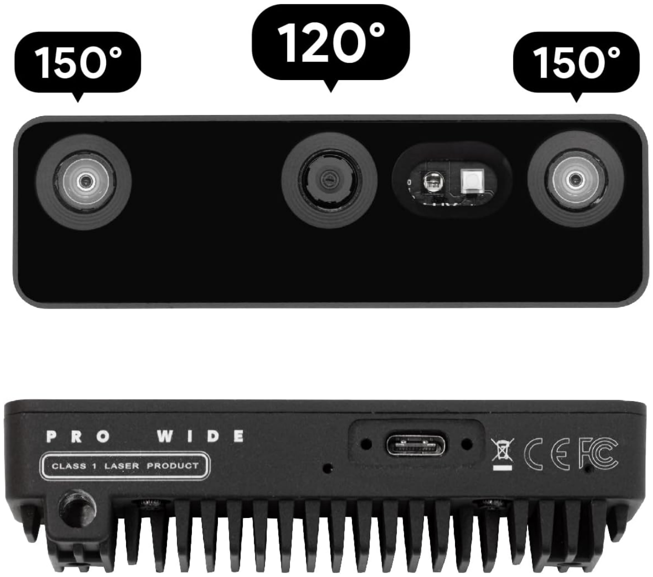

Figure 1: RGB camera with FoV

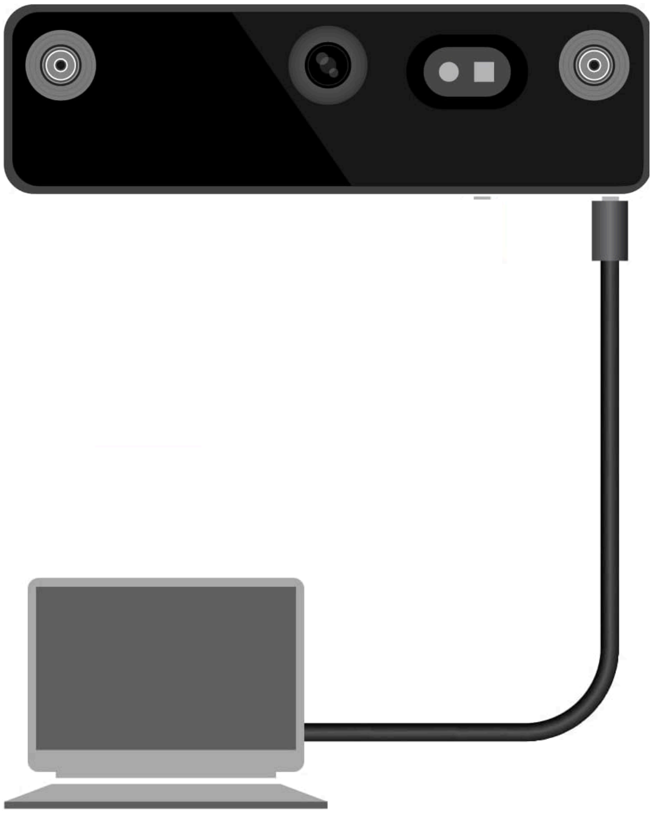

Figure 2: RGB camera connected with laptop

2 Steps to Install

Install DepthAI v3 via CMD in a Windows laptop.

```
git clone https://github.com/luxonis/depthai-core.git
cd depthai-core
# Create and source venv
python -m venv venv
.\venv\Scripts\Activate.bat
# Install requirements
python examples\python\install_requirements.py
```

Open CMD and change the location to the

```
cd depthai-core\examples\python\Camera
```

Create a Python file to capture images with timestamps as the filename for example ”20251202_034904_629072.jpg”

```
depthai-core\examples\python\Camera  notepad camera_run.py
```

and copy the capturing script from Appendix A

3 Deploying the Setup

Connect the laptop and camera as shown in Fig 2 and run the script.

```
depthai-core\examples\python\Camera  python camera_run.py
```

The captured image looks like

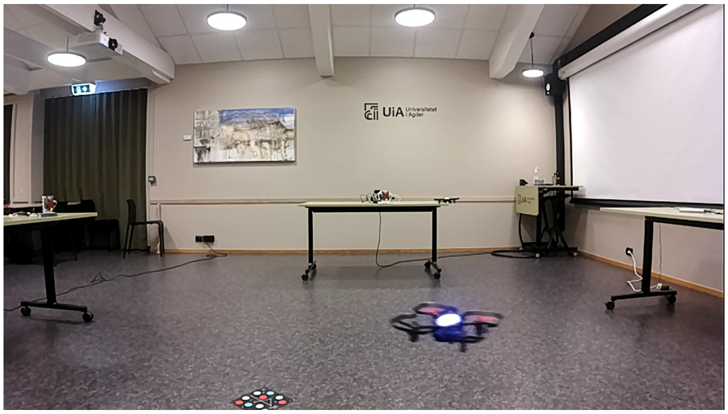

Figure 3: Output of the camera with name as ”20251018_182747_136799.jpg”

4 Appendix

A Python Script for RGB Frame Capture using DepthAI

```python
#!/usr/bin/env python3

import cv2
import depthai as dai
import os
import time
from datetime import datetime

# === USER CONFIG ===
ROTATE_180 = False     # Set to True if camera is upside down
TARGET_FPS = 10        # Desired save rate (frames per second)
# ===================

# Create a new folder for this run
RUN_DIR = "./captures/" + datetime.now().strftime("%Y%m%d_%H%M%S")
os.makedirs(RUN_DIR, exist_ok=True)
print(f"[INFO] Saving images to: {RUN_DIR}")

# Create pipeline
with dai.Pipeline() as pipeline:
    # Define source and output
    cam = pipeline.create(dai.node.Camera).build()
    videoQueue = cam.requestOutput((1280, 720)).createOutputQueue()  # use 720p for better
        ↪ aspect

    # Connect to device and start pipeline
    pipeline.start()
    # FPS limiter
    target_frame_time = 1.0 / TARGET_FPS
    last_time = 0

    while pipeline.isRunning():
        videoIn = videoQueue.get()
        assert isinstance(videoIn, dai.ImgFrame)
        frame = videoIn.getCvFrame()

        # Optional rotation
        if ROTATE_180:
            frame = cv2.rotate(frame, cv2.ROTATE_180)

        # Show preview
        cv2.imshow("video", frame)

        # FPS limiter for saving
        now = time.time()
        if now - last_time >= target_frame_time:
            timestamp = datetime.now().strftime("%Y%m%d_%H%M%S_%f")
            filename = os.path.join(RUN_DIR, f"{timestamp}.jpg")
            cv2.imwrite(filename, frame)
            last_time = now

        if cv2.waitKey(1) == ord("q"):
            break
```