



DEPARTAMENTO
DE TELEMÁTICA
Y ELECTRÓNICA

2025

User Manual

Nvidia Jetson

Nano

Waveshare Jetson Nano 4GB
Development Kit E



INDEX

1.	Description Jetson Nano	4
2.	Jetson Nano Components View	5
3.	Jetson Pack	6
4.	Initial Setup Requirements	7
5.	Compatible Models	7
5.1	Pre-trained models	7
5.2	Jetson GPI Python Libray	9
6.	Initialization and Implementation in LINUX	9
6.1	Install SDK Manager	9
6.2	Open SDK Manager	10
7.	Access credentials	11
8.	Graphical Interface Visualization	12
8.1	HDMI Display	12
8.2	SSH Display	12
8.3	USB Display + VNC Client.....	13
9.	Internet connection.....	14
9.1	Connecting via Ethernet	14
9.2	Connection via USB	14
9.3	Connection via WIFI cards.....	14
10.	Camera Interface	15
10.1	GStreamer.....	15
10.2	OpenCV Library.....	15

INDEX OF FIGURES

Figure 1. Visual Jetson Nano Components	5
Figure 2. Jetson Nano Overview	6
Figure 3. SDK Manager Install	9
Figure 4. SDK Manager Download.....	10
Figure 5. Nivida SDK Manager	10
Figure 6. Jetson Nano Recognition	11

1. Description Jetson Nano

The Nvidia Jetson Nano is a low-cost, high-performance development computer designed to apply Artificial Intelligence – AI and deep learning to edge computing devices. Its design is unique and aimed at students and developers who want to implement solutions by computer vision, neural networks and robotics without the need for expensive hardware.

There are several models, however, this manual focuses on the **Development Kit E** model of the kit developed by **Waveshare**, which contains the following features:

- ❖ **Manufacturer:** Waveshare
- ❖ **GPU:** 128 CUDA MaxWell cores
- ❖ **CPU:** Quad-core ARM Cortex-A57
- ❖ **RAM:** 4GB LPDDR4
- ❖ **Ports:** HDMI, USB 3.0, USB 2.0, GPIOs
- ❖ **Storage:** microSD
- ❖ **Power Consumption:** 5W – 10W
- ❖ **Num Procesadores:** 4
- ❖ **Dimensions:** 6.9 x 4.5 x 4.5 cm
- ❖ **Weight:** 1.19 kg
- ❖ **Wireless Connection Type:** Bluetooth
- ❖ **Cooling fan**
- ❖ **Compatible devices:** Stereo camera (dual camera for 3D vision or depth sensing), sensor, personal computer, smartphone, keyboard, mouse, external storage devices, display devices

The AI algorithms **that** can be run are those related to image classification, object detection, segmentation, and speech processing.

To support the models that are going to be used in the Jetson Nano it is advisable to use a **64GB UHS-I // II SD microSD** card, although you can also use the 32GB. This microSD card must contain the image of the operating system to be used. The JetPack image must be installed and then loaded onto the card to be inserted into the Jetson Nano. In the section "[Initiation and start-up](#)" the steps are explained in detail.

If you want to look at the components in more detail, see the [WaveShare page](#).

2. Jetson Nano Components View

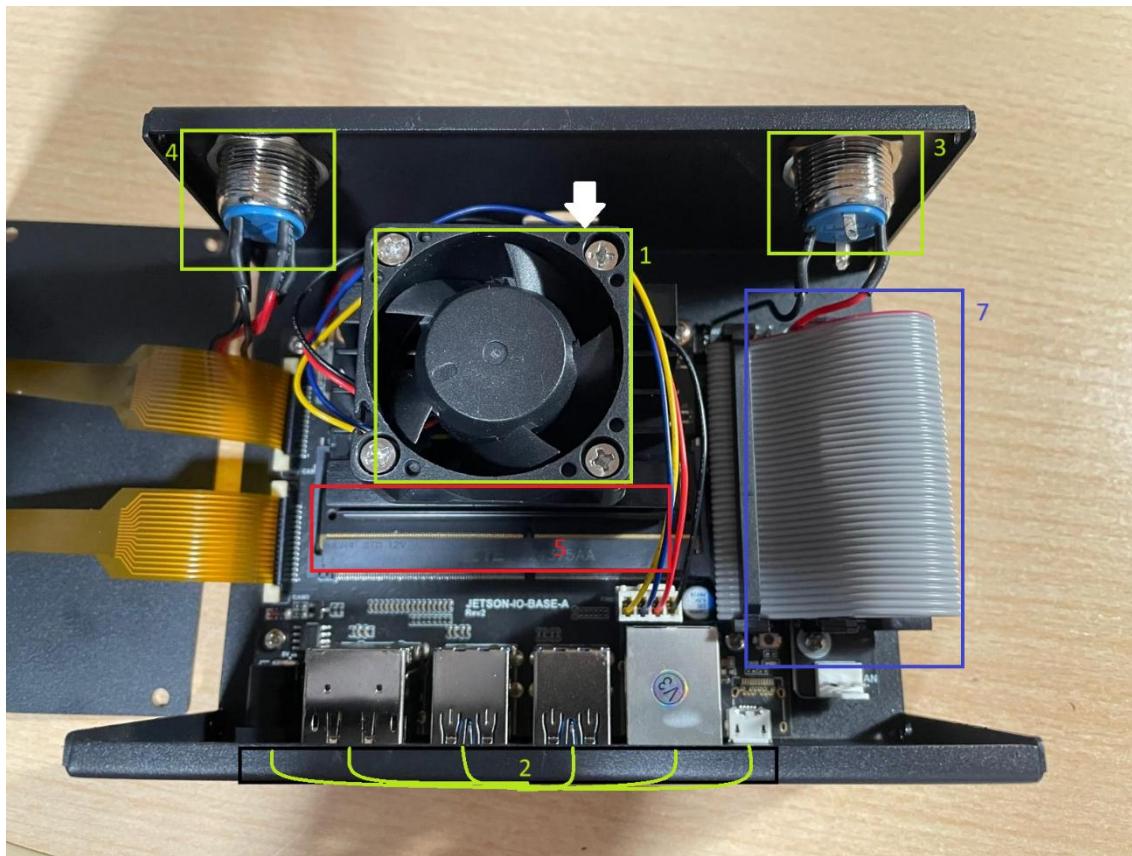


Figure 1. Visual Jetson Nano Components

An aerial view of the Jetson Nano is indicated without the protective outer shell. The components listed in the image are as follows:

1. **Fan**: Used to cool the processor and prevent overheating.
2. **Connection ports**: USB; HDMI, Power supply, Ethernet and auxiliary USBs.
3. **Power**: power button.
4. **Restart**: equivalent to doing a sudo system reboot.
5. **Motherboard**: where the microSD is located at the height of the white arrow.
6. **Yellow bands**: they are responsible for connecting the camera
7. **Pin port**

The view of the implanted camera on the outside together with the housing is the general view of the device.



Figure 2. Jetson Nano Overview

3. Jetson Pack

Nvidia Jetpack is a software package that includes all the dependencies and packages necessary to be able to develop in Nvidia Jetson Nano. It greatly facilitates the development of AI, computer vision, and GPU computing applications. This Jetson Pack contains the following components:

- ❖ **BSP – Board Support Package:** software that allows the operating system and programs to interact correctly with the Jetson Nano's hardware.
- ❖ **S.O Linux:** Version of Ubuntu Linux optimized for Jetson Nano.
- ❖ **CUDA:** **Required** for the GPU to perform advanced calculations. It significantly speeds up AI tasks.
- ❖ **cuDNN** – Cuda Deep Neural Networks: Optimized library to run deep neural network models faster.
- ❖ **TensorRT:** Software that optimizes the performance of deep learning models when programmed with other libraries. Ex: TensorFlow.
- ❖ **Libraries Media processing and computer vision:** Working with images, videos, and camera sensors.

The Waveshare Jeton Nano has **Jetpack version 4.6** installed on the internal **eMMC chip card**.

4. Initial Setup Requirements

The initial components required to be able to configure the Jetson Nano are as follows:

- ❖ **NIC AC825 module:** to be able to perform the interface part. This is necessary if we want to do the configuration through WiFi. If the USB Cable is used, it is not necessary.
- ❖ **64GB UHS-I // II SD microSD:** or its default of 32GB.
- ❖ **SD or SD/micro card adapter/reader.**
- ❖ **JetsonPack image:** dump it into the microSD card.
- ❖ **Power supply adapter (recommended):** 5V, 4Amps.
- ❖ **Cable USB.**
- ❖ **Wi-Fi router and Internet connection:** if you want to configure it via WiFi.



If Jetpack 4.6 is desired ≠ , the image will be flashed in the microSD, otherwise the software is already installed on the internal card of the eMMC chip

5. Compatible Models

Many of the models that can be used or created from scratch are compatible with frameworks widely used in Python. These frameworks supported by the Jetson Nano are **TensorFlow, PyTorch, Caffe and MXNet**.

5.1 Pre-trained models

Included with installing JetsonPack are a variety of AI models that enable **image classification, object detection, segmentation, and audio processing**.

When you initialize the Jetson Nano, the interface provides you with the following models to download to use, i.e. they have already been trained. The models are as follows:

1. Image recognition

- AlexNet
- GoogleNet V1, V12
- ResNet V18, V50, V101, V152
- VGG-16, VGG-19
- Inception – v4

2. Object Detection

- SSD-MobileNet V1, V2
- Pednet

- MultiPed
- FaceNet
- DetectNet-COCO-Dog
- DetectNet-COCO-Bottle
- DetectNet-COCO-Chair
- DetectNet-COCO-Airplane

3. Semantic segmentation

- FCNN-Resnet18-Cityscapes-512x256
- FCNN-Resnet18-Cityscapes-1024x512
- FCNN-Resnet18-Cityscapes-2048x1024
- FCNN-Resnet18-DeepScene-576x320
- FCNN-Resnet18-DeepScene-864x480
- FCNN-Resnet18-MHP-512x320
- FCNN-Resnet18-MHP-640x360
- FCNN-Resnet18-Pascal-VOC-320x320
- FCNN-Resnet18-Pascal-VOC-512x320
- FCNN-Resnet18-SUN-RGBD-512x400
- FCNN-Resnet18-SUN-RGBD-640x512
- FCNN-Alexnet-Cityscapes-SD
- FCNN-Alexnet-Cityscapes-HD
- FCNN-Alexnet-Aerial-FPV
- FCNN-Alexnet-Pascal-VOC
- FCNN-Alexnet-Synthia-CVPR
- FCNN-Alexnet-Synthia-Summer-SD
- FCNN-Alexnet-Synthia-Summer-HD

4. Image Processing

- Deep-Homography-COCO
- Super-Resolution-BSD500

All these models, as indicated above, are used only for the function of classification, detection or segmentation. These models are already trained and cannot be retrained or modified. In case you want to train a model from scratch, you can choose to use the **Jetson GPIO Python Library**

5.2 Jetson GPIO Python Library

The **Jetson GPIO Python Library** is a library that allows you to make AI models from the start. It is useful for coding and creating models, including the training part of these. In addition, it is compatible with common sensors and peripherals, including many that are used in other on-chip systems such as AdaFruit or Raspberry Pi.

Its import into Python files, after its previous installation, can be done by:

```
import Jetson.GPIO as GPIO
```

For the correct and complete configuration consult the following link:
<https://github.com/NVIDIA/jetson-gpio>

6. Initialization and Implementation in LINUX

You will then work with a **Linux** system, as it is the operating system compatible with the Jetson Nano, thus making the configuration much more accessible. It should be remembered that you could also work with Windows but using various tools such as **PuTTY** or **WinSCP**.

5.1 Install SDK Manager

This section is aimed at formatting the **microSD** in case there was nothing pre-installed. SDK Manager is a tool provided by Nvidia to install, update and manage the software packages necessary for proper operation.

Installation requires removing the microSD card. If the microSD card comes inside the Jetson Nano, it must be removed and then the software dumped into it. The location of the microSD card may vary depending on the model, but in this kit it is located at the bottom. See the Jetson [Nano Components View section](#).

Before performing the installation, a previous account must have been created. For installation click [here](#).

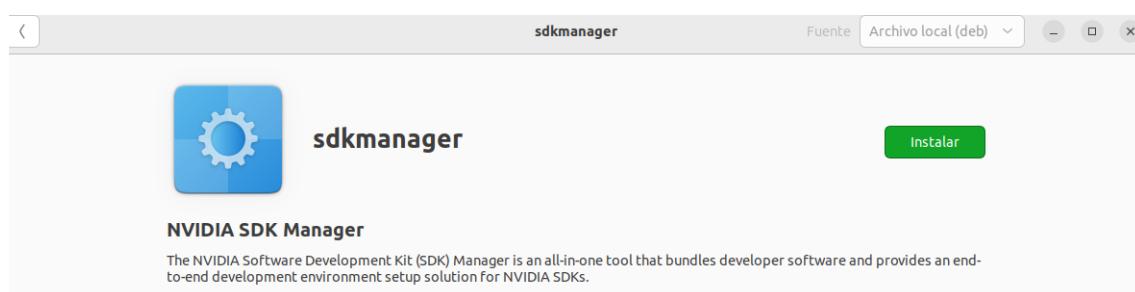


Figure 3. SDK Manager Install

When it has been downloaded, run the following command to be able to install it completely.

```
sudo dpkg -i sdkmanager_2.2.0-12028_amd64.deb (or the version you have)
```

```
cipd@CIPD:~/Descargas$ sudo dpkg -i sdkmanager_2.2.0-12028_amd64.deb
[sudo] contraseña para cipd:
Leyendo la base de datos ... 233410 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar sdkmanager_2.2.0-12028_amd64.deb ...
Desempaquetando sdkmanager (2.2.0-12028) sobre (2.2.0-12028) ...
Configurando sdkmanager (2.2.0-12028) ...
Procesando disparadores para mailcap (3.70+nmu1ubuntu1) ...
Procesando disparadores para gnome-menus (3.36.0-1ubuntu3) ...
Procesando disparadores para desktop-file-utils (0.26-1ubuntu3) ...
Procesando disparadores para hicolor-icon-theme (0.17-2) ...
```

Figure 4. SDK Manager Download

In case there is an error due to file dependency failure, run the following command to resolve the issue.

```
sudo apt --fix-broken install
```

6.2 Open SDK Manager

After the installation is complete, open SDK Manager from the terminal by typing.

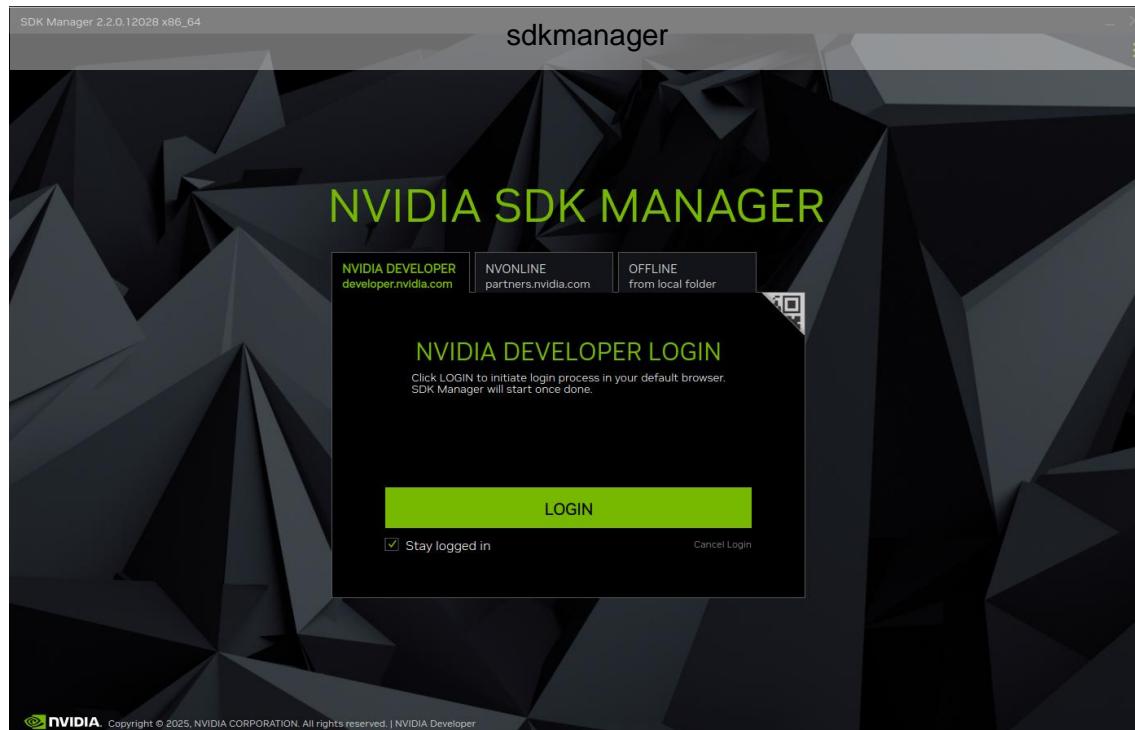


Figure 5. Nivida SDK Manager

You must enter the email, and the username used for its creation in the previous step. Once the Jetson Nano is successfully created and connected to the pc using USB and the power connector, the result should be as follows.

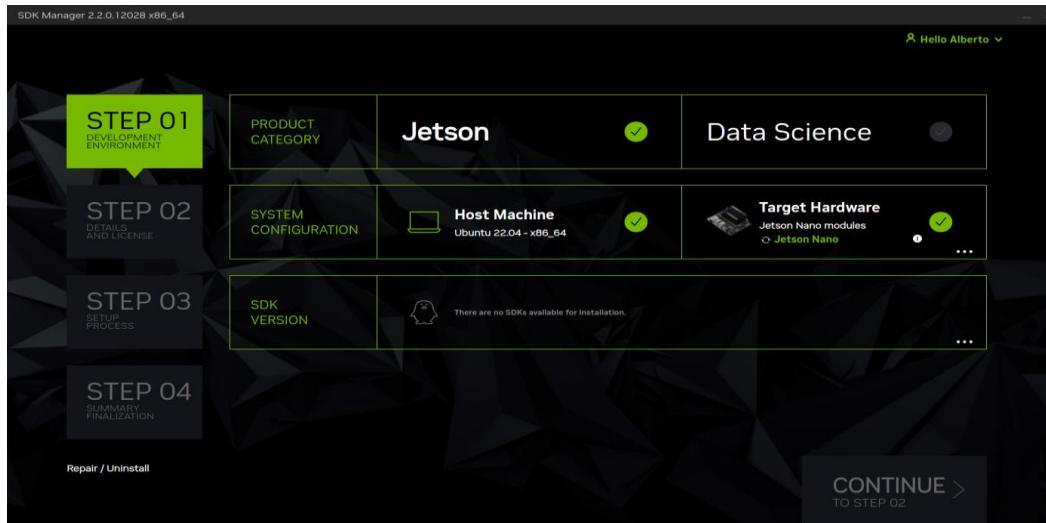


Figure 6. Jetson Nano Recognition

As you can see, there are no versions for its installation since in this **Waveshare Jetson Nano** it is already pre-installed. If there are installations, they will tell you in step 3. If this is your case, click [here](#) to perform the installation correctly.

It should be remembered that the pre-installation comes on the kit's internal eMMC card, specifically with the **Jetpack version 4.6**.

7. Access credentials

The associated credentials to access within the Jetson Nano may vary, so it is recommended to check the official product page first. In this kit developed by Waveshare, the login credentials are as follows:

- **Username:** jetson
- **Password:** jetson

8. Graphical Interface Visualization

Simply explained, the Jetson Nano is a computer in itself, but the kit comes without any screen where you can view the graphical interface. Various ways or methods can be used for viewing it and entered the device's own terminal.

8.1 HDMI Display

The easiest and most optimal way is to have an external monitor and use the HDMI port input. In this way, all the content will be transmitted through the port and the graphical interface can be displayed on the monitor.

On the other hand, to be able to interact within the interface, a mouse and keyboard are necessary. Both must be connected via USB to the Jetson Nano. It is advisable to connect all the necessary connections before turning on the machine, since being a Linux system, it may not recognize them.

8.2 SSH Display

The second method is to use the **SSH** (Secure Shell) method. First, you need to connect the Jetson Nano to your PC or laptop via USB. Once connected, you must identify the IP address of the Jetson Nano (to be accessed) by the command:

```
ifconfig (can also be used ip a)
```

Look for the section that indicates eth0 or wlan. Otherwise, the connection associated with the usb0 field must be searched. Once the IP address has been identified, the following command must be used to access it:

```
ssh user@IP_JETSON_NANO
```

In the user part you must put the user corresponding to the jetson session. Please refer to the [credentials](#) for this section.

 This method only works if the Jetson Nano has a WIFI module installed and supports SSH connection. SSH settings can only be activated by accessing the device from the inside.

8.3 USB Display + VNC Client

The third method, like the previous one, uses the USB connection, but in this case a VNC server for Linux is used. **VNC** or Virtual Network Computing is a communication protocol used to remotely use another computer over a network connection. It works by sending images of the remote computer's screen allowing interaction with keyboard and mouse. For them, the following must be installed:

- **VNC Server:** Installed on the Jetson Nano.
- **VNC Client (Viewer):** on the computer through which you want to access remotely.
- **Network connection:** using IP and port.

To access the Jetson Nano terminal, after having connected the USB run the following commands:

```
ls /dev/ttyUSB* o ls /dev/ttyACM*
```

A set of numeric characters will appear that we will call NUM. Then run to access via Serial UART to the terminal:

```
sudo screen /dev/ttyUSB0 NUM
```

You can also write minicom instead of screen. These are two packages required to access. Remember to install them, if they are not, by running:

```
sudo apt install screen / sudo apt install minicom
```

Once inside the Jetson terminal, the VNC server is installed. Inside the Jetson Nano there are several readmes, and one of them explains all the steps and commands to be executed on how it should be installed.

After installing the server, in the Linux system terminal of the personal computer, the client must be installed in order to be able to view. There are several clients who are:

- **Remmina (Linux)**
- **TightVNCServer**
- **X11VNC**
- **RealVCN**
- **TigerVNC**

In this case, it will be explained how to install Reminna, although in the other cases it will be similar. The following steps must be executed below:

```
sudo apt update && sudo apt install remmina -y
```

```
vcnserver :1 (desde la terminal de la Jetson para iniciar el servidor)
```

Open the application from the personal computer terminal and select VNC connection type and server **IP_JETSON_NANO:1**.

9. Internet connection

The Jetson Nano can be configured to the Internet using several types of connection. This connection is necessary if you need to communicate with other devices, save data in a database, manage packages or updates.

9.1 Connecting via Ethernet

The development kit has a built-in Ethernet cable, which can be used to plug directly into a router or signal repeater.

In case there are two options due to location, lack of resources or space that is not accessible, WiFi-Ethernet adapters can also be used.

This option is optimal for greater connectivity.

9.2 Connection via USB

There are WIFI-USB adapters that are similar to Ethernet, with the consideration that the type of cable used for the connection changes. It can be used in cases where Ethernet connections are not available on the Jetson Nano.

9.3 Connection via WIFI cards

This option can only be implemented on 4GB versions, not 2GB versions. To do this, a WIFI module would be needed, which is a card to be able to support wireless connection.

This card must be installed in the **M.2 Key E slot**.

10. Camera Interface

As explained, the Waveshare Develop Kit contains a camera which is integrated in the Jetson Nano. This camera is useful in case there's some need of recording from videos to take images. There are two ways to access it.

10.1 GStreamer

GStreamer is an open-source multimedia framework designed efficiently to handle audio and video streams.

The next command is written in the CLI to try the functionality of the camera.

```
gst-launch 1.0 nvarguscamerasrc ! noverlaysink
```

This option will open the camera in the monitor, and you can see what's recording. It may be possible that the camera is recording at an undesired angle. This happens because the initial setup was wrong and it was turned upside down, so instead of ridding it again, you can use the parameter [flip_method](#).

If you want to record a video instead just write:

```
gst-launch-1.0 nvarguscamerasrc ! videoconvert ! xvimagesink
```

There are also attributes to specify the width and height of the image and the path where it can be saved.

10.2 OpenCV Library

OpenCV is a library very useful for access to cameras and using images. In this case we have a CSI (Camera Serial Interface) due it's directly embedded. This is an example of code to access the camera.

```
import cv2

cap = cv2.VideoCapture("nvarguscamerasrc ! video/x-raw(memory:NVMM), width=1280, height=720, framerate=30/1,
format=NV12 ! nvvidconv ! video/x-raw, format=BGRx ! videoconvert ! video/x-raw, format=BGR ! appsink",
cv2.CAP_GSTREAMER)

while cap.isOpened():

    ret, frame = cap.read()

    if not ret:
        break

    cv2.imshow("CSI Camera", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):

        break

cap.release()
cv2.destroyAllWindows()
```