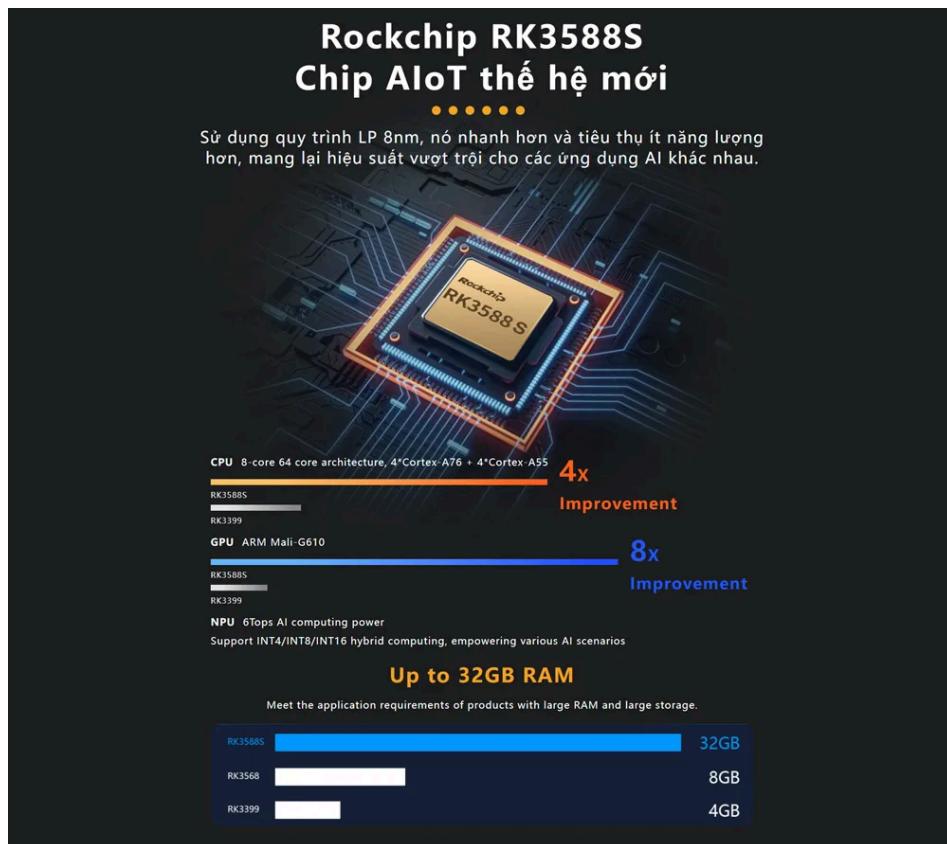


# AI on Orange Pi 5B with NPU Rockchip RK3588S

## 1. Giới thiệu về OrangePi 5B:

- Mua từ OrangePi.vn: <https://orangeipi.vn/shop/orange-pi-5b-16gb-ram128gb-emmc>



## 1 NPU là gì?

NPU là một loại chip chuyên dụng được thiết kế để tăng tốc độ xử lý các tác vụ liên quan đến trí tuệ nhân tạo (AI), đặc biệt là học máy sâu (deep learning).

NPU hoạt động tương tự như **CPU** (Central Processing Unit) và **GPU** (Graphics Processing Unit), nhưng được tối ưu hóa để xử lý các phép toán ma trận và vectơ phức tạp thường được sử dụng trong các mô hình học máy sâu.

Các chip NPU này được sản xuất dưới dạng vật lý và được tích hợp trực tiếp vào các thiết bị như **điện thoại thông minh**, máy tính, PC. Tuy nhiên, cũng có những trường hợp NPU được mô phỏng trên phần mềm, sử dụng các tài nguyên phần cứng hiện có như CPU hoặc GPU.



NPU là một loại chip chuyên dụng được thiết kế để tăng tốc độ xử lý các tác vụ liên quan đến trí tuệ nhân tạo (AI)

- Chạy được nhiều OS: OrangePi OS (Droid)、OrangePi OS (Arch)、OrangePi OS (OH)、Debian11、Ubuntu22.04、Ubuntu20.04、Android12
- Ứng dụng của OrangePi

Orange Pi 5B có thể được áp dụng rộng rãi cho máy tính bảng, điện toán biên (EDGE Computing), trí tuệ nhân tạo (Artificial Intelligence), điện toán đám mây, AR/VR, bảo mật thông minh, nhà thông minh và các lĩnh vực khác, bao gồm các ngành công nghiệp IoT khác nhau.



Robot và Trí tuệ nhân tạo



Máy tính và màn hình thông minh



Điện toán cận biên



Điện toán đám mây



AR/VR



Bảo mật thông minh



Nhà thông minh

## 2. Install OS

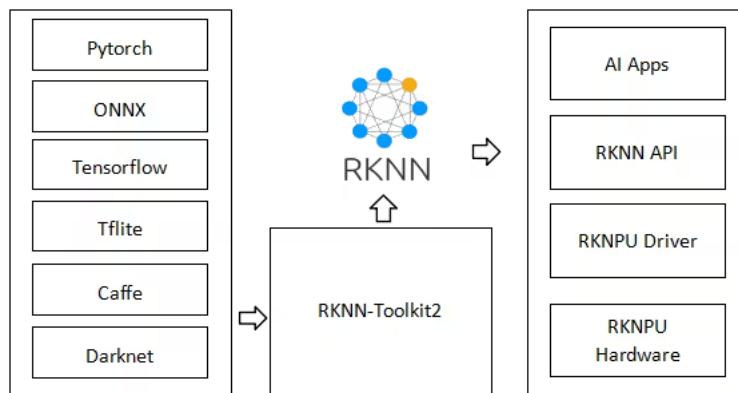
- OS được cung cấp và build sẵn bởi OrangePi.vn. Mua hàng tại đây để được hỗ trợ tốt nhất bước đầu. Sau đó pro rồi thì nghịch thoải mái.
- Có Armbian GUI: <https://drive.google.com/file/d/1criCuJLnyUOrc-z-zKc1kiEkgzAB6pde/view?usp=sharing>
- Armbian Không GUI: [https://drive.google.com/file/d/1BnVlg4uOcqQwlwbJ0J2\\_mtaLpNoDfVJ/view?usp=sharing](https://drive.google.com/file/d/1BnVlg4uOcqQwlwbJ0J2_mtaLpNoDfVJ/view?usp=sharing)
- Flash bằng Balena : <https://etcher.balena.io/#download-etcher>
- Cắm thẻ nhớ, theo hướng dẫn nhập username, kết nối wifi, chọn timezone, language và BOOM! Giao diện hiện ra. Chi tiết: <https://www.youtube.com/watch?v=EC1wOCGxLTI>

## 3. Bài toán

- Nhận diện lửa cháy bằng Pi 5B Yolov11
- Chạy LLM trên Pi 5B: Llama8B

## 4. Export weight bằng máy tính X86

- Vì sao phải export?



```

path: data # dataset root dir
train: train/images # train images (relative to 'path')
val: val/images # val images (relative to 'path')
test: test/images # test images (relative to 'path')

# Classes
names: ['smoke', 'fire'] # Replace with your actual class names

# Counts
nc: 2 # number of classes
train_count: 14122
val_count: 3099
test_count: 4306

```

```

pip install ultralytics
yolo task=detect mode=train model=yolo1n.pt data=data.yaml epochs=10 imgsz=640 plots=True

```

```
yolo export model=ok.pt format=rknn name=rk3588
```

- Có thể tải weight sẵn tại: [https://docs.radxa.com/en/rock5/rock5c/app-development/rknn\\_ultralytics](https://docs.radxa.com/en/rock5/rock5c/app-development/rknn_ultralytics)

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
1/10	1.36G	1.901	2.937	1.627	5	480: 100%   883/883 [01:33<00:00, 9.45it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100%   97/97 [00:10<00:00, 8.98it/s]
	all	3099	3932	0.306	0.316	0.219 0.0853
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
2/10	1.41G	1.945	2.146	1.651	17	480: 100%   883/883 [01:27<00:00, 10.14it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100%   97/97 [00:08<00:00, 11.45it/s]
	all	3099	3932	0.372	0.364	0.279 0.114
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
3/10	1.4G	1.913	2.003	1.636	23	480: 100%   883/883 [01:23<00:00, 10.52it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100%   97/97 [00:09<00:00, 10.75it/s]
	all	3099	3932	0.422	0.416	0.376 0.165
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
4/10	1.41G	1.821	1.856	1.567	7	480: 100%   883/883 [01:24<00:00, 10.45it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100%   97/97 [00:08<00:00, 11.53it/s]
	all	3099	3932	0.518	0.439	0.439 0.201
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
5/10	1.4G	1.762	1.716	1.513	14	480: 100%   883/883 [01:23<00:00, 10.53it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100%   97/97 [00:08<00:00, 11.38it/s]
	all	3099	3932	0.572	0.475	0.498 0.245
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
6/10	1.4G	1.699	1.607	1.474	20	480: 100%   883/883 [01:23<00:00, 10.60it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100%   97/97 [00:08<00:00, 11.16it/s]
	all	3099	3932	0.609	0.522	0.549 0.274
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
7/10	1.4G	1.639	1.506	1.431	7	480: 100%   883/883 [01:26<00:00, 10.22it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100%   97/97 [00:09<00:00, 10.47it/s]
	all	3099	3932	0.621	0.557	0.592 0.303
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
8/10	1.4G	1.592	1.438	1.407	15	480: 100%   883/883 [01:26<00:00, 10.18it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100%   97/97 [00:09<00:00, 10.71it/s]
	all	3099	3932	0.65	0.558	0.605 0.324
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
9/10	1.4G	1.56	1.389	1.368	18	480: 49%   430/883 [00:42<00:46, 9.66it/s]

## 5. Run on OrangePi 5B

<https://gist.github.com/ZlodeiBaal/847569b24bb338566680ab2a8f22c6b2/raw/58fa8278ac9492ab4c244e7da>

```
https://github.com/Prashant2804/Fire-and-smoke-detection-using-yolov11
```

```
sudo apt install python3-pip  
sudo apt install python3.10-venv
```

```
python3 -m venv yolov11_env  
source ./yolov11_env/bin/activate  
pip3 install ultralytics  
pip3 install rknn-toolkit-lite2  
pip3 install python-telegram-bot  
https://github.com/airockchip/rknn-toolkit2/raw/refs/heads/master/rknpu2/runtime/Linux/librknn\_api/aarch64/librknrt.so
```

```
from ultralytics import YOLO  
  
# Load the exported RKNN model  
rknn_model = YOLO("./ok_rknn_model")  
  
# Run inference  
results = rknn_model("https://ultralytics.com/images/bus.jpg")
```

```
import asyncio  
import cv2  
import telegram  
from ultralytics import YOLO  
import datetime  
import threading  
import requests  
from PIL import Image  
import numpy  
  
# Load the exported RKNN model  
model = YOLO("./ok_rknn_model")  
my_token = "7534858637:AAEwGQU6Ryp9LxfK7F7h63JGJkkES1DrWwk"  
bot = telegram.Bot(token=my_token)  
  
def get_chat_id():  
    url = f"https://api.telegram.org/bot{my_token}/getUpdates"  
    print(requests.get(url).json())  
  
def alert(img):  
    global last_alert  
    cv2.putText(img, "ALARM!!!!", (10, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)  
    # New thread to send telegram after 15 seconds  
    if (last_alert is None) or (  
        (datetime.datetime.utcnow() - last_alert).total_seconds() > 5):
```

```

last_alert = datetime.datetime.utcnow()
cv2.imwrite("alert.png", cv2.resize(img, dsize=None, fx=0.2, fy=0.2))
thread = threading.Thread(target=send_telegram)
thread.start()
return

def send_telegram():
    try:
        asyncio.run(bot.sendPhoto(chat_id=1246123900, photo=open("alert.png", "rb"), caption="Có chay",))
    except Exception as ex:
        print("Can not send message telegram ", ex)

    print("Send sucess")

# establish and open webcam feed
last_alert = datetime.datetime.utcnow()

cap = cv2.VideoCapture("Sequence 01_1.mp4")

if not cap.isOpened():
    print("Cannot open camera")
    exit(1)

while True:
    ret, frame = cap.read()
    if not ret:
        print("Cannot read camera")
        exit(2)

    # pass frame through model
    frame_resized = cv2.resize(frame, (800, 640))
    # img_bgr = frame_resized
    r = model(frame_resized)[0]

    detects = r.to_json()
    if len(detects)>0:

        # Plot results image
        frame_resized = r.plot()
        alert(frame_resized)

        cv2.imshow('Stream', frame_resized)
        # Break loop on 'q' for quit
        if cv2.waitKey(1) == ord('q'):
            break

```

## 5. Sample LLM

- <https://github.com/thanhtantran/RKLLM-Gradio> (của bác chủ cung cấp)

```

python3 -m venv rkllm_env
source ./rkllm_env/bin/activate
git clone https://github.com/c0zaut/rkllm-gradio && cd rkllm-gradio

```

```
python3 -m pip install --no-cache-dir --upgrade -r requirements.txt
```

```
cd models  
wget https://huggingface.co/c01zaut/Llama-3.2-3B-Instruct-rk3588-1.1.1/resolve/main/Llama-3.2-3B-Instruct-rk3588-w8a
```

```
cd ..  
python3 rkllm_server_gradio.py
```

#### 6. Món quà tặng từ OrangePi.vn

- Code MIAI\_ORANGEPI
- Hiệu lực đến 20/3
- Giảm 100K khi mua 5B hàng tại Orangepi.vn