

KV260 AI Example

<https://xilinx.github.io/kria-apps-docs/home/build/html/index.html> の Smart Camera を試す .

セットアップ

cf. [Booting Kria Starter Kit Linux](#)

Ubuntu 22.04 な SD カードを用意しておく .

KV260 で Ubuntu 22.04 を 使う には Boot FW の アップデート が必要 (cf. <https://wasa-labo.com/wp/?p=1056>)

Ubuntu 22.04 起動後は必要なパッケージのセットアップ .

snap を使うか apt の ppa リポジトリを使うか選択すること .

apt を使うことにしてすすめる .

```
sudo add-apt-repository ppa:xilinx-apps
sudo add-apt-repository ppa:ubuntu-xilinx/sdk
sudo apt update
sudo apt upgrade
```

Docker もインストール .

cf. <https://docs.docker.com/engine/install/ubuntu/>

以前にインストールされていたバージョンを削除して

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

必要なライブラリをインストール

```
sudo apt-get install ca-certificates curl gnupg lsb-release
```

GPG 鍵をセットして

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
```

リポジトリを登録

```
echo ¥
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu ¥
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

で , 最後にセットアップ

```
sudo apt update
sudo apt upgrade
sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

動作確認しておしまい .

```
sudo docker run hello-world
```

ユーザ権限で Docker 使えるように Docker 用のグループを作ってユーザを追加

```
sudo groupadd docker
sudo usermod -a -G docker $USER
```

最後に zocl ドライバのインストール

```
sudo apt install xrt-dkms
```

ドライバがロードできてるか確認してみる

```
ubuntu@kria: $ lsmod |grep zocl
zocl                204800  0
```

Smart Camera を試す

まずは , アプリケーションファームウェアをセットアップする .

```
sudo apt install xlnx-firmware-kv260-smartcam
```

デスクトップ環境を終了する . これで HDMI 接続してるディスプレイの gdm が終了するはず .

```
sudo xutil desktop_disable
```

ちなみに , デスクトップ環境を再度有効にする場合は ,

```
sudo xutil desktop_enable
```

らしい .

準備したら , kv260-smartcam をロードする

```
sudo xutil unloadapp # こちらは何か別のアプリ用のファームウェアをロードしていた場合
sudo xutil loadapp kv260-smartcam
```

loadapp すると

```
ubuntu@kria: $ sudo xutil loadapp kv260-smartcam
kv260-smartcam: loaded to slot 0
```

という感じ . UART でみてるコンソールには ,

```
[ 210.854187] OF: overlay: WARNING: memory leak will occur if overlay removed, property:
/fpga-full/firmware-name
[ 210.864389] OF: overlay: WARNING: memory leak will occur if overlay removed, property:
```

```
/fpga-full/resets
[ 211.164188] zocl-drm axi:zyxclmm_drm: IRQ index 8 not found
[ 211.240274] debugfs: Directory '4-003c' with parent 'regmap' already present!
```

とか出力された .

SmartCamera アプリは Docker コンテナで提供されている .

```
docker pull xilinx/smartcam:2022.1
```

で pull して , 実行 .

```
docker run ¥
--env="DISPLAY" ¥
-h "xlnx-docker" ¥
--env="XDG_SESSION_TYPE" ¥
--net=host ¥
--privileged ¥
--volume="$HOME/.Xauthority:/root/.Xauthority:rw" ¥
-v /tmp:/tmp ¥
-v /dev:/dev ¥
-v /sys:/sys ¥
-v /etc/vart.conf:/etc/vart.conf ¥
-v /lib/firmware/xilinx:/lib/firmware/xilinx ¥
-v /run:/run ¥
-it xilinx/smartcam:2022.1 bash
```

Docker イメージが起動して bash プロンプトが表示される .

動作確認用の動画をダウンロードする . チュートリアルでは ,

- [Facedet / RefineDet AI Task](#)
- [ADAS SSD AI Task](#)

が例示している . 1920x1080 を指定して Free Download をクリックしてダウンロード .
それぞれ

```
ffmpeg -i input-video.mp4 -c:v libx264 -pix_fmt nv12 -vf scale=1920:1080 -r 30 output.nv12.h264
```

として変換 . output.nv12.h264 が生成されるので , KV260 上の Ubuntu の /tmp に scp する .
/tmp は , Docker コンテナでも /tmp にマウントされている .

アプリ実行用に Jupyter Lab 環境をセットアップする . セットアップ用の

```
smartcam-install.py
```

を実行すると ,

```
root@xlnx-docker: # smartcam-install.py
Info: Notebook files under /opt/xilinx/kv260-smartcam/share/notebooks are copied to
/root/notebooks/smartcam.
```

と , /root/notebooks/smartcam に必要なファイルがコピーされる .
で , Jupyter Lab を起動

```
jupyter-lab --notebook-dir=/root/notebooks/smartcam --allow-root --ip='*' &
```

ブラウザでアクセスするとカメラから取り込んだ画像が HDMI に出力されるなどがみれる .

コマンドラインでもアプリを実行できる . たとえば ,

```
smartcam --file /tmp/output.nv12.h264 -i h264 -W 1920 -H 1080 -r 30 --target file -a ssd
```

実行すると out.h264 ができる . 実行結果はこんな感じ .

AI のモデルの設定は [Customizing the AI Models used in the application](#) に詳しい .