Advantech EI-52 installation guide to Edge Insights for Vision

This page will guide you through steps to configure Edge Insights for Vision on ESH, install on target device, pull container from docker hub, and run benchmark application with AI models.

- Advantech EI-52 installation guide to Edge Insights for Vision
 - Step 1 : Prepare for target OS system
 - Step 2 : Configure Edge Insight for Vision package on ESH
 - Step 3 : Install on target device
 - Step 4 : Pull docker image from docker hub & run smart city demo
 - Step 5 : Run benchmark application with AI models (yolov3 & yolov4).
 - Step 6: Run benchmark application with other AI models
- (Optional) SOP for using VPU in Docker Container
 - Step 1 : Download HDDL driver package
 - Step 2 : Install HDDL Driver
 - Step 3 : Using VPU in Docker Container
- (Optional) SOP for turn on Turbo Mode
 - <u>Step 1 : Configuration to get best CPU performance</u>
 - Step 2 : Configuration to get best GPU performance

Step 1: Prepare for target OS system

Make sure your target system has a fresh installation of Ubuntu that corresponds to the version of Edge Insights for Vision that you downloaded. Here I use Ubuntu20.04 LTS as example.

*URL to download Ubuntu 20.04 LTS ISO image:

<u>https://ubuntu.com/download/desktop/thank-you?version=20.04.2.0&architecture=amd64</u> (<u>https://ubuntu.com/download/desktop/thank-you?version=20.04.2.0&architecture=amd64</u>)

Step 2 : Configure Edge Insight for Vision package on ESH

 Go to Intel ESH website: <u>https://software.intel.com/content/www/us/en/develop/topics/iot/edge-</u> <u>solutions/vision-recipes.html (https://software.intel.com/content/www/us/en/develop/topics/iot/edge-</u> <u>solutions/vision-recipes.html)</u>

• Configure and download required components:

IoT Developer Program	(intel	$\hat{\boldsymbol{r}}$		⊕ පු Q
Overview	Featured Components	Use Cases	Documentation	
		Ur	nit (VPU).	0
	Configure & Do	ownload		
	Get Started Doc	umentation		

• Select "Customize download"

Hands-on guide to Edg	e I × Intel® Edge Software Hui× +					- •
← → C	O A https://software.intel.com/iot/ed	gesoftwarehub/download/home/visioninsights		ť	2	
ntel.				() SUPPORT	2 🕀 USA	(ENGLISH)
Edge Software Hub / Edg	e Insights for Vision					
Intel® Download a re	Edge Software Hub	installation from pre-validated components.				
Ed	lge Insights for Vision		Customize gewnload	Download recommended configuration		
Comp Imple	outer vision and deep learning inference for app ment using a containerized architecture or a st	lications at the edge, optimized for Intel [®] architecture and-alone runtime.				
	Related resources	2 Learn more about Edge Insights for Vision	D Product documentation	12 Hardware and System Requirements		
8	Madulas	Reference Implementation - Multi-Camera Dataction	of Social Distancing		•	
	View pre-validated modules available	Intel [®] Distribution of OpenVINO [™] toolkit 2021.4 in a	Container		0	
	with this solution.	Intel® Distribution of OpenVINO" toolkit 2021.4 Run	itime		•	
		Deep Learning (DL) Streamer Pipelines			١	

• Select Openvino Download version: 2021.3 for Target System OS

Target

Select version and target OS information.

Download Version :	2021.3	~
Target System OS :	Ubuntu 20.04 LTS	~

Next 🕨

1 of 8

• Select the Multi-Camera Detection if need to detect social distancing

2 of 8	 Back 	Next
--------	--------------------------	------

Grayed-out components cannot be removed, as previous selections depend on them

Reference Implementation - Multi-Camera Detection of Social Distancing Version 1.4.0	(i
Creates an end-to-end pipeline to detect the presence of COVID-19 preventive measures, such as the social distancing using computer vision inference.	

.

3 of 8		Next	•

Grayed-out components cannot be removed, as previous selections depend on them

Intel [®] Distribution of OpenVINO [™] toolkit 2021.3 in a Container Version 2021.3	()
The Intel® Distribution of OpenVINO™ toolkit optimizes inferencing on your edge IoT device by extending workloads across Intel® hardware.	
This component builds Docker* images for the OpenVINO toolkit.	
This component requires Docker* CE to run. Selecting this component will add Docker* CE to your download package.	
Intel [®] Distribution of OpenVINO [™] toolkit 2021.3 Runtime Version 2021.3	(i)
Intel® Distribution of OpenVINO [™] toolkit 2021.3 Runtime Version 2021.3 The Intel® Distribution of OpenVINO [™] toolkit optimizes inferencing on your edge IoT device by extending workloads across Intel® hardware.	()

• Select other tools if needed

Intel Tools

Grayed-out components cannot be removed, as previous selections depend on them

Deep Learning (DL) Streamer Pipelines Version 1.4.0	i
Intel Threading Building Blocks is used by OpenVINO™ Inference Engine for inference on CPU and DL Streamer uses Inference Engine as a backend for executing inference.	
ONNX* Runtime	í
Version 1.7.0	
ONNX* Runtime lets you inference using deep learning models that are written	
initialite works not directly supported by openvirto.	

Interoperability

5 of 8 🛛 📢 Back

Next 🕨

Grayed-out components cannot be removed, as previous selections depend on them

EdgeX* Foundry Version 1.3	()
The EdgeX Foundry* framework provides data and communications interoperability between devices and applications at the IoT edge.	

Container Engines & Orchestration

Grayed-out components cannot be removed, as previous selections depend on them

Docker Compose* Version 1.24.0	i
Docker* Compose is a tool for defining and running multi-container Docker* applications. It lets you run and manage multiple containers at the same time with a single command. This module requires Docker* CE to run. Selecting this module will add Docker* CE to your download package.	
Docker Community Edition (CE)* Version 20.10.5	(j)
Docker* is a container framework. It lets you package an application and its dependencies in a virtual container, making it easier to create, deploy, run, and manage applications.	\checkmark
K3s* (Lightweight Kubernetes*) Version 1.20.4	(j)
Cloud Edge Connectors 7 of 8	Next 🕨

Grayed-out components cannot be removed, as previous selections depend on them



• Download the package and copy the .zip file to target device and copy the product key

Download

Downloads a small installer file. Once the download is complete, copy and run from your target device.



Yes, I would like to subscribe to Edge Software Hub product communications and stay connected by email and telephone. I understand that I can unsubscribe at any time.

Included Components

Intel[®] Distribution of OpenVINO[™] toolkit 2021.3 Runtime

Docker Community Edition (CE)*

Intel® Distribution of OpenVINO™ toolkit 2021.3 in a Container

Intel [®] Edge Software Hub Download a recommended configuration or create a customized installation from pre-validated
Get Edge Insights for Vision
Product Key. Copy
You will be prompted to enter your product key during installation.
Installing Edge Insights for Vision Once your download is complete, copy the zip file to your target system and install it:
Step 1: Prepare your target system with the target OS.
Step 2: Copy the zip file to your target system.
Step 3: Extract and install the software.

See the Get Started Guide 12 for detailed instructions.

Support

For Edge Software Hub issues, contact our support team 12

Step 3 : Install on target device

• Copy the Edge Insights for Vision .zip File to the Target System

	J · · · J · · · .	J			- 5	J = = =		
<	> ∢ û Home	Downloads					۹ :	
Ø	Recent							
企		Desktop	Documents	Downloads	Music	Pictures	Public	Templates
	Desktop							
D	Documents		zip	3				
÷	Downloads	Videos	edge_ insights_	Examples				
99	Music		vision.zip					
٥	Pictures							
×	Videos							
	Trash							
+	Other Locations							
						"edge_insights_	vision.zip" se	elected (10.6 MB)

• Extract the Edge Insights for Vision Software

<	> 🔸 🏠 Home	edge_insights_vision	٩	:	
Ø	Recent				
ŵ	Home	edgesoftw edgesoftw readme txt			
	Desktop	are are configurati			
D	Documents	on.xml			
∻	Downloads				
99	Music				
٥	Pictures				
H	Videos				
1	Trash				
+	Other Locations				
			"edgesoftware	" selected (10.7 MB)

- Install the Edge Insights for Vision Software
- Run these commands

```
$ cd edge_insights_vision/
```

- \$ sudo chmod 775 edgesoftware
- \$./edgesoftware install

• Type the product key at the prompt:



• When the installation is complete, you will see the message Installation of package complete and the installation status for each module.



Step 4 : Pull docker image from docker hub & run smart city demo

• To get the video streaming file for demo

```
$ wget -0 ~/Downloads/NewVideo2.mp4 \
    https://github.com/incluit/OpenVino-For-SmartCity\
    /raw/master/data/NewVideo2.mp4
```

- Pull image from Docker Hub
- \$ docker pull docker_hub_destination/openvino:2021.3_developer_models

NOTE: "docker_hub_destination" should be changed to existing resource locations: sertek or synnexgrp or wpig or wt1com

• Run smart city demo

```
$ docker run -it -v /tmp/.X11-unix:/tmp/.X11-unix \
    -e DISPLAY=$DISPLAY -v ~/Downloads:/mnt \
    --device /dev/dri:/dev/dri \
    --group-add=$(stat -c "%g" /dev/dri/render*) \
    --rm docker_hub_destination/openvino:2021.3_developer_models
$ cd /home/openvino/
$ ./smartcity_demo.sh
```

NOTE: "docker_hub_destination" should be changed to existing resource locations: sertek or synnexgrp or wpig or wt1com



Step 5 : Run benchmark application with AI models (yolov3 & yolov4)

• Run Benchmark_app to measure the performance on device

```
$ docker run -it -v /tmp/.X11-unix:/tmp/.X11-unix \
    -e DISPLAY=$DISPLAY -v ~/Downloads:/mnt \
    --device /dev/dri:/dev/dri \
    --group-add=$(stat -c "%g" /dev/dri/render*) \
    --rm docker_hub_destination/openvino:2021.3_developer_models
$ cd /home/openvino/
$ python3 run_command.py
```

NOTE: "docker_hub_destination" should be changed to existing resource locations: sertek or synnexgrp or wpig or wt1com

• Benchmark results:

• Intel® Celeron® 6305E

Al Model	Dovico	Throughput(FPS)			
Al Wouei	Device	FP32	Throughput(FPS) FP16 IN 1.54 5. 14.62 30 0.72 2. 6.84 13	INT8	
volo v2 tf	CPU	1.54	1.54	5.59	
y010-v5-ti	GPU	7.43	14.62	30.03	
volo v4 tf	CPU	0.72	0.72	2.12	
y010-V4-LI	GPU	3.36	<mark>6.84</mark>	13.83	

Intel[®] Core[™] i3-1115G4E

	Device	Throughput(FPS)		
Allviouei		FP32	FP16	INT8
volo v2 tf	CPU	3.49	3.43	12.66
y010-v5-ti	GPU	7.53	Throughput(FPS) FP16 IN 3.43 12. 14.75 29. 1.51 4.6 6.85 13.	29.42
volo v4 tf	CPU	1.51	1.51	4.68
y010-v4-ti	GPU	3.34	6.85	13.79

• Intel[®] Core[™] i5-1145G7E

		Throughput(FPS)				
Al Model	Device	FP32	FP16	INT8		
	CPU	5.38	5.43	20.47		
yolo-v3-tf	GPU	11.39	22.24	47.59		
	CPU	2.57	2.56	8.19		
yolo-v4-tf	GPU	4.71	9.77	21.18		

<Model name: YOLOv3(FP32/FP16/INT8); Target device: CPU>

	<pre>cmd:python3 /opt/intel/openvino/deployment_tools/tools/benchmark_tool/benchmark_app.py -m /opt/intel/ openving models/public/uplo_v3-tf/FP32/uplo_v3-tf.xml -d CPU -api async -t 60</pre>
	openvente_houses/particle/goto-valien/hase/goto-valien/kme -a ero -ape agine -e oo
Į	MKLDNNPlugin version 2.1
ł	[INFO] Read network took 627.51 ms
1	[INFO] Load network took 966.85 ms
ł	[Step 10/11] Measuring performance (Start inference asynchronously, 4 inference requests using 4 stre
ł	ams for CPU, LINITS: 00000 MS duration)
1	Latercy 720.22 ms
1	Throughout: 5.38 FPS
1	<pre>cmd:python3 /opt/intel/openvino/deployment tools/tools/benchmark tool/benchmark app.py -m /opt/intel/</pre>
	openvino_models/public/yolo-v3-tf/FP16/yolo-v3-tf.xml -d CPU -api async -t 60
	MKLDNNPlugin version 2.1
	LINFO J Kead network took 321.03 Ms
	[INFO] Lodo network took 1000.20 MS
	(step 19/11) neusining performance (start characterice asynchronously, 4 anterence requests using 4 streams for CPU. Limits: 60000 ms duration)
	[INFO] First inference took 299.34 ms
	Latency: 734.35 ms
	Throughput: 5.43 FPS
	<pre>cmd:python3 /opt/intel/openvino/deployment_tools/tools/benchmark_tool/benchmark_app.py -m /opt/intel/</pre>
	openvino_models/public/yolo-v3-tf/FP16-INT8/yolo-v3-tf.xml -d CPU -api async -t 60
	MKLDNNPlugin version 2.1
	[INFO] Read network took 178.49 ms
	[INFO] Load network took 969.19 ms
	[Step 10/11] Measuring performance (Start inference asynchronously, 4 inference requests using 4 stre
1	ams for CPU, limits: 60000 ms duration)
5	LIARD J FUTST UNFERCE TOOK 81.29 AS
	Catelly: 19014 MS



<Model name: YOLOv4(FP32/FP16/INT8); Target device: CPU>

<pre>cmd:python3 /opt/intel/openvino/deployment_tools/tools/benchmark_tool/benchmark_app.py -m /opt/intel/ openvino_models/public/yolo-v4-tf/FP32/yolo-v4-tf.xml -d CPU -api async -t 60</pre>
MKLDNNPlugin version 2.1
[INFO] Read network took 645.99 ms
[INFO] Load network took 1016.38 ms
[Step 10/11] Measuring performance (Start inference asynchronously, 4 inference requests using 4 stre ams for CPU, limits: 60000 ms duration)
[INFO] First inference took 607.19 ms
Latency: 1550.43 ms
Throughput: 2.57 FPS
cmd:python3 /opt/intel/openvino/deployment_tools/tools/benchmark_tool/benchmark_app.py -m /opt/intel/ openvino_models/public/yolo-v4-tf/FP16/yolo-v4-tf.xml -d CPU -api async -t 60
MKLDNNPlugin version 2.1
[INFO] Read network took 331.68 ms
[INFO] Load network took 1101.35 ms
[Step 10/11] Measuring performance (Start inference asynchronously, 4 inference requests using 4 stre ams for CPU, limits: 60000 ms duration)
[INFO] First inference took 604.96 ms
Latency: 1542.77 ms
Throughput: 2.56 FPS
cmd:python3 /opt/intel/openvino/deployment_tools/tools/benchmark_tool/benchmark_app.py -m /opt/intel/ openvino_models/public/yolo-v4-tf/FP16-INT8/yolo-v4-tf.xml -d CPU -api async -t 60
MKLDNNPlugin version 2.1
[INFO] Read network took 191.63 ms
[INFO] Load network took 1082.36 ms
[Step 10/11] Measuring performance (Start inference asynchronously, 4 inference requests using 4 stre
ams for CPU, limits: 60000 ms duration)
[INFO] First inference took 200.93 ms
Latency: 481.49 ms
Throughput: 8.19 FPS



Step 6: Run benchmark application with other AI models

If you wish to run the benchmark with other AI models, run the following command and change the "route_to_model" to the destination of the model file and "model.xml" to the model file name.

```
$ python3 \
   /opt/intel/openvino/deployment_tools/tools/benchmark_tool/benchmark_app.py \
   -m /mnt/path_to_model/model.xml \
   -d CPU -api async -t 60
```

NOTE: There is some pre-build model in the "/opt/intel/openvino_models/public" you can try to use these models.

Below is an example using resnet-50-tf public model

```
$ python3 \
   /opt/intel/openvino/deployment_tools/tools/benchmark_tool/benchmark_app.py \
   -m /opt/intel/openvino_models/public/resnet-50-tf/FP16-INT8/resnet-50-tf.xml \
   -d CPU -api async -t 60
```

(Optional) SOP for using VPU in Docker Container

Step 1: Download HDDL driver package

You can use the official Intel[®] Distribution of OpenVINO[™] toolkit packages from trusted resources.

See more on the <u>product page (https://software.intel.com/content/www/us/en/develop/tools/openvino-</u> toolkit/choose-download.html).

Or you can download a small archive of HDDL driver package from Amazon Web Services*. Available releases for Ubuntu* 18.04:

- 2020.2 (https://storage.openvinotoolkit.org/drivers/vpu/hddl/2020.2/hddl ubuntu18 1076.tgz)
- 2020.3 (https://storage.openvinotoolkit.org/drivers/vpu/hddl/2020.3/hddl ubuntu18 1167.tgz)
- 2020.3.1 (https://storage.openvinotoolkit.org/drivers/vpu/hddl/2020.3.1/hddl ubuntu18 1409.tgz)
- 2020.3.2 (https://storage.openvinotoolkit.org/drivers/vpu/hddl/2020.3.2/hddl ubuntu18 1651.tgz)
- 2020.4 (https://storage.openvinotoolkit.org/drivers/vpu/hddl/2020.4/hddl_ubuntu18_1229.tgz)
- <u>2021.1 (https://storage.openvinotoolkit.org/drivers/vpu/hddl/2021.1/hddl_ubuntu18_1380.tgz)</u>
- 2021.2 (https://storage.openvinotoolkit.org/drivers/vpu/hddl/2021.2/hddl ubuntu18 1509.tgz)
- 2021.3 (https://storage.openvinotoolkit.org/drivers/vpu/hddl/2021.3/hddl ubuntu18 1636.tgz)
- 2021.4 (https://storage.openvinotoolkit.org/drivers/vpu/hddl/2021.4/hddl_ubuntu18_1701.tgz)

Step 2 : Install HDDL Driver

• Run following commands on the host machine with HDDL device from <archive_extract_folder>/hddl:

```
$ sudo -i
$ source setupvars.sh
$ ./install_IVAD_VPU_dependencies.sh
$ reboot
$ sudo -i
$ source setupvars.sh && ./bin/hddldaemon -d
```

NOTE: Please don't close the terminal which runing the hddldaemon.

Step 3 : Using VPU in Docker Container

• Run Benchmark_app with VPU

```
$ docker run -it -v /tmp/.X11-unix:/tmp/.X11-unix \
    -e DISPLAY=$DISPLAY -v ~/Downloads:/mnt \
    --device /dev/dri:/dev/dri \
    --group-add=$(stat -c "%g" /dev/dri/render*) \
    --rm docker_hub_destination/openvino:2021.3_developer_models
$ cd /home/openvino/
$ python3 \
    /opt/intel/openvino/deployment_tools/tools/benchmark_tool/benchmark_app.py \
    -m /opt/intel/openvino_models/public/resnet-50-tf/FP16/resnet-50-tf.xml \
    -d HDDL -api async -t 60
```

NOTE: "docker_hub_destination" should be changed to existing resource locations: sertek or synnexgrp or wpig or wt1com

(Optional) SOP for turn on Turbo Mode

refer to the following guide for performance optimization.

Step 1 : Configuration to get best CPU performance

• Press "Delete" to go to BIOS Setting



• Go to "Advanced" tab and select the "Power & Performance"



• Go to "CPU - Power Management Control"

Advanced	Aptio Setup — AMI	
 Power & Performance CPU - Power Management Control GT - Power Management Control 		CF Op
		++: † ↓: Ent +/- F1:

 Select the "Boot performance mode" modify it to "Turbo Performance" (Default:Max Non-Turbo Performance)

Advanced	Aptio Setup – AMI	
CPU – Power Management Control Boot performance mode Intel(R) SpeedStep(tm) Turbo Mode C states	[Max Non-Turbo Performance] [Enabled] [Enabled] [Disabled]	Select the that the B starting fo
	Boot performance mode Max Battery Max Non-Turbo Performance Turbo Performance	<pre>+: Select 1: Select nter: Select +/-: Change F1: General F2: Previous F3: Optimize F4: Save & E ESC: Evit</pre>

Step 2 : Configuration to get best GPU performance

• Go to "Advanced" tab and select the "Power & Performance"



• Go to "GT - Power Management Control"

Advanced	Aptio Setup – AMI	
 Power & Performance CPU - Power Management Control GT - Power Management Control 		CF Op
		→+: †↓: Ent
		+/ F1

• Select the "Disable Turbo GT frequency" modify it to "Disable"(Default:Disable)

	Advanced Aptio Setup - AMI		
	GT – Power Management Control RC6(Render Standby) Maximum GT frequency Disable Turbo GT frequency	[Disabled] [Default Max Frequency] [Disabled]	Enabled: Dj frequency. frequency i
N. J.		Disable Turbo GT frequency — bled abled	Select So Select It Enter: Select +/-: Change C F1: General H F2: Previous F3: Optimized F4: Save & Ex ESC: Exit

• Press "F4" to Save the configuration.

