Term Project

The objective in the term project is to evaluate students' understanding in containerization technique and best practice using Dockerfile and deploy it via Kubernetes orchestrator.

By completing all the practical lab and up until this point, student should be able to:

In conceptual level,

- Understand the benefits of containerizing an application
- Understand the container landscape and the role played by different components in the landscape.
- Understand the differences between high level container runtime interface (CRI) such as containerd, low level container runtime such as runc.
- Understand about container life cycle.

In practice,

- Using DeepStream SDK
- Search the right container base image from Dockerhub.
- Write a Dockerfile
- Build a container image
- Perform container image task such as build, list, tag, pull, and push image to remote container registry such as Dockerhub
- Describe and write Kubernetes resources file in YAML format
- Deploy Kubernetes YAML resources using kubectl CLI tool
- Deploy Kubernetes YAML resources using Lens graphical user interface

General Info

Project Topic Selection

- It is best if your application has a web frontend or dashboard to visualize the result. However, it is not necessary. It can be a direct video output from the program.
- We have pre-selected several project ideas in the next page, or you could pick one for inspiration from https://developer.nvidia.com/embedded/community/jetson-projects
- It is not necessary to select a project that can utilize the DeepStream or TensorRT AI processing power on the Xavier NX. It can even be a simple computer-vision based project using framework such as OpenCV.

Project Demo Criteria

Each team will be given a duration of <> minutes to present their work. It is up to the team leader to delegate the tasks among team members.

- Show and explain the Dockerfile
- Show the Pods/Services or any other K8s resources that you created using Lens for the project during live demo

Folder Structure

- Dockerfile
- <your_app>-deployment.yaml
- <your_app>-service.yaml
- <your_app>-<k8s resource>.yaml
- Report.docx

Tips

Containerization (When writing docker file)

- Use ` opendatacam:v3.0.2-xavier` as your base image when writing Dockerfile. It already contains CUDA library inside.
- If you don't know how to write Dockerfile, you can start by logging in an opendatacam container, then execute all the installation process inside the container. Then record all the commands you entered, and convert it to 'RUN...' command in Dockerfile.

Python

Some program might be coded using Python3, in that case and if you are using `opendatacam` as your container base, install the right package manager for Python3 such as

```
apt install python3-pip
pip3 install --upgrade pip

# Install requirements
pip3 install -r requirements.txt
```

Video Stream Location

You can get RTSP live stream for video of different type, e.g. traffic, people by referring to the Appendix section.

Project Ideas

1. People Counting
2. Mask Detection
3. Body Pose Detection
4. Hand Pose Detection
5. Fall Detection

Project 1: People counting in real time

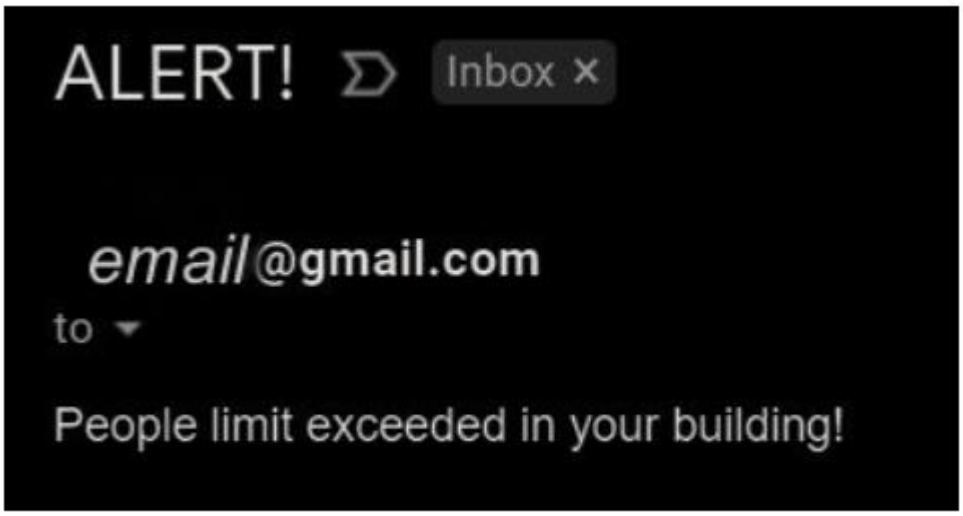
Project source code: https://github.com/saimj7/People-Counting-in-Real-Time

Feature

Log

End Time	In	Out	Total Inside
18-08-2020 20:46	1	1	4
	2	2	
	3	3	
	4		
	5		
	6		
	7		

Real time alert



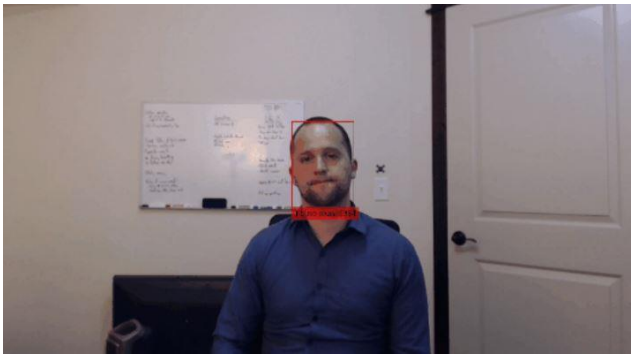
Project 2: Mask Detection

Project source code: https://github.com/bdtinc/maskcam#viewing-the-live-video-stream

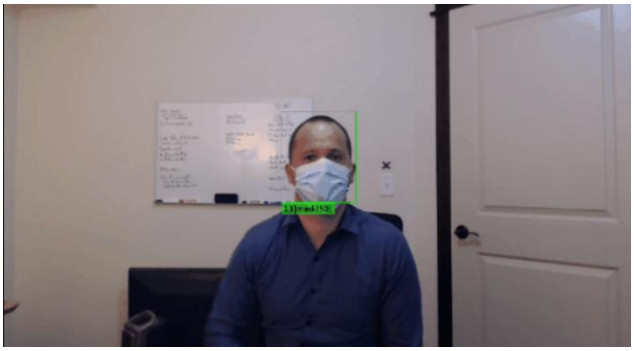
References using the source code: https://collabnix.com/building-a-real-time-crowd-face-mask-detection-system-on-nvidia-jetson-nano/

Result

Before wearing mask

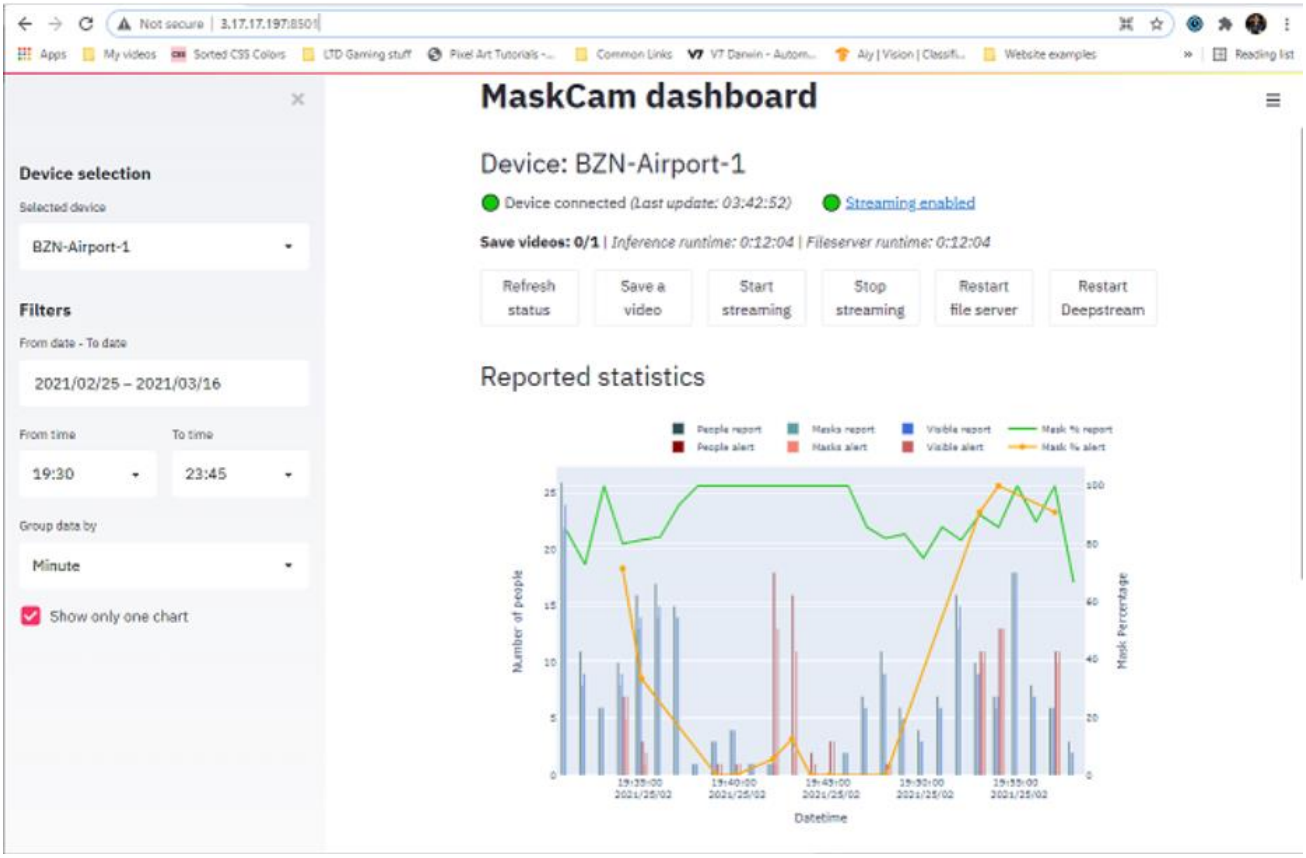


Before Wearing Mask



After Wearing Mask

Web Dashboard

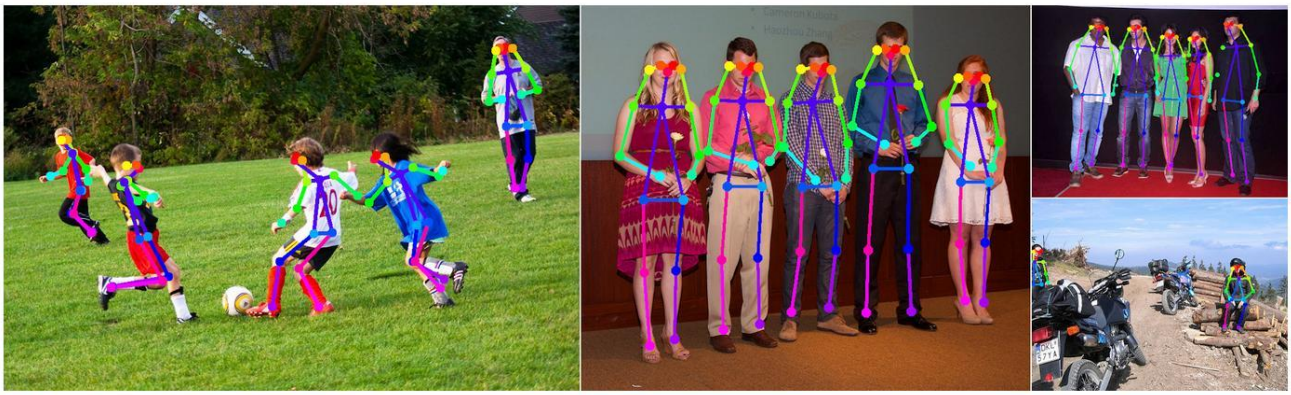


Project 3: Body Pose Detection

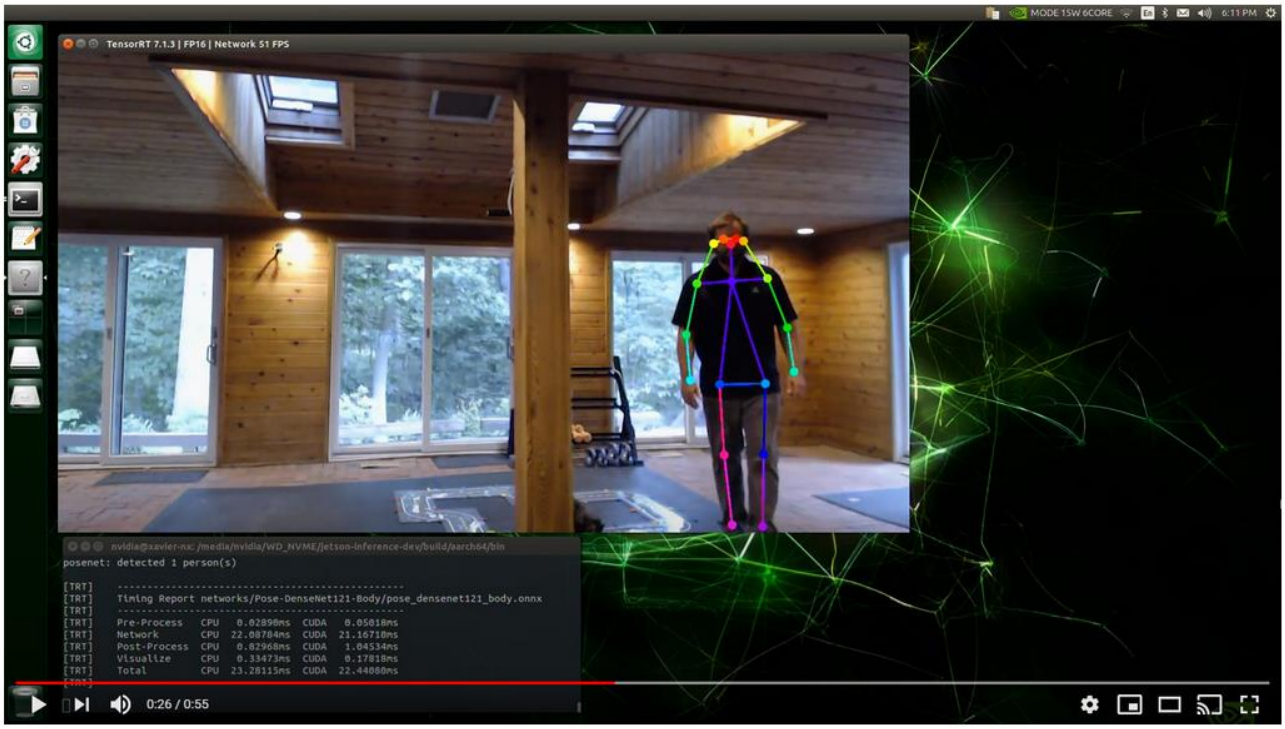
Project source code: https://github.com/dusty-nv/jetson-inference/blob/master/docs/posenet.md

Result

Detect from image

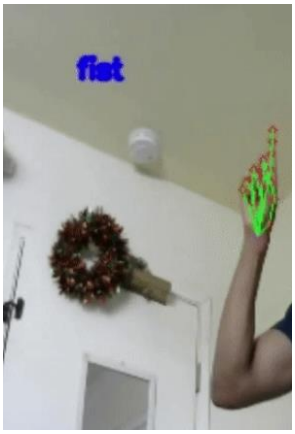


From video:



Project 4: Hand Pose Detection

Project source code: https://github.com/NVIDIA-AI-IOT/trt_pose_hand



Project 5: Fall Detection

Project source code: https://github.com/JustinhoCHN/fall-detection

Fall not detected

Fall detected

References

Find other Jetson related projects here:

https://developer.nvidia.com/embedded/community/jetson-projects

Appendix

RTSP Server Stream Location

For Body Pose

rtps:<user>@<pass>:…

For Traffic

For …