



# JETSON AI CERTIFICATIONS

Yet Another Virtual Room  
Jetson AI Ambassador



# JETSON AI CERTIFICATIONS

Обо мне:

Я много лет работаю старшим преподавателем Южного Федерального университета и преподаю курсы по многим предметам, включая «Технологии машинного обучения».

Кроме того, я преподаю курс AI в Samsung IT Academy  
[https://myitacademy.ru/ai/ai\\_lectures/](https://myitacademy.ru/ai/ai_lectures/)

В настоящее время пишу кандидатскую диссертацию по AI/ML.



# JETSON AI CERTIFICATIONS

Идея.

Я хотел разработать практически полезное приложение, которым мог бы пользоваться я сам и любой человек. Мне, в моей преподавательской деятельности необходима возможность замены заднего фона при общении через программы удаленного общения. К сожалению не все программы могут это делать. Например MS Teams for linux или Discord не реализуют такой функциональности.

Поэтому я решил получить полезное велью от от покупки Nvidia Jetson Nano и в качестве проекта написать приложение, которое это делает.

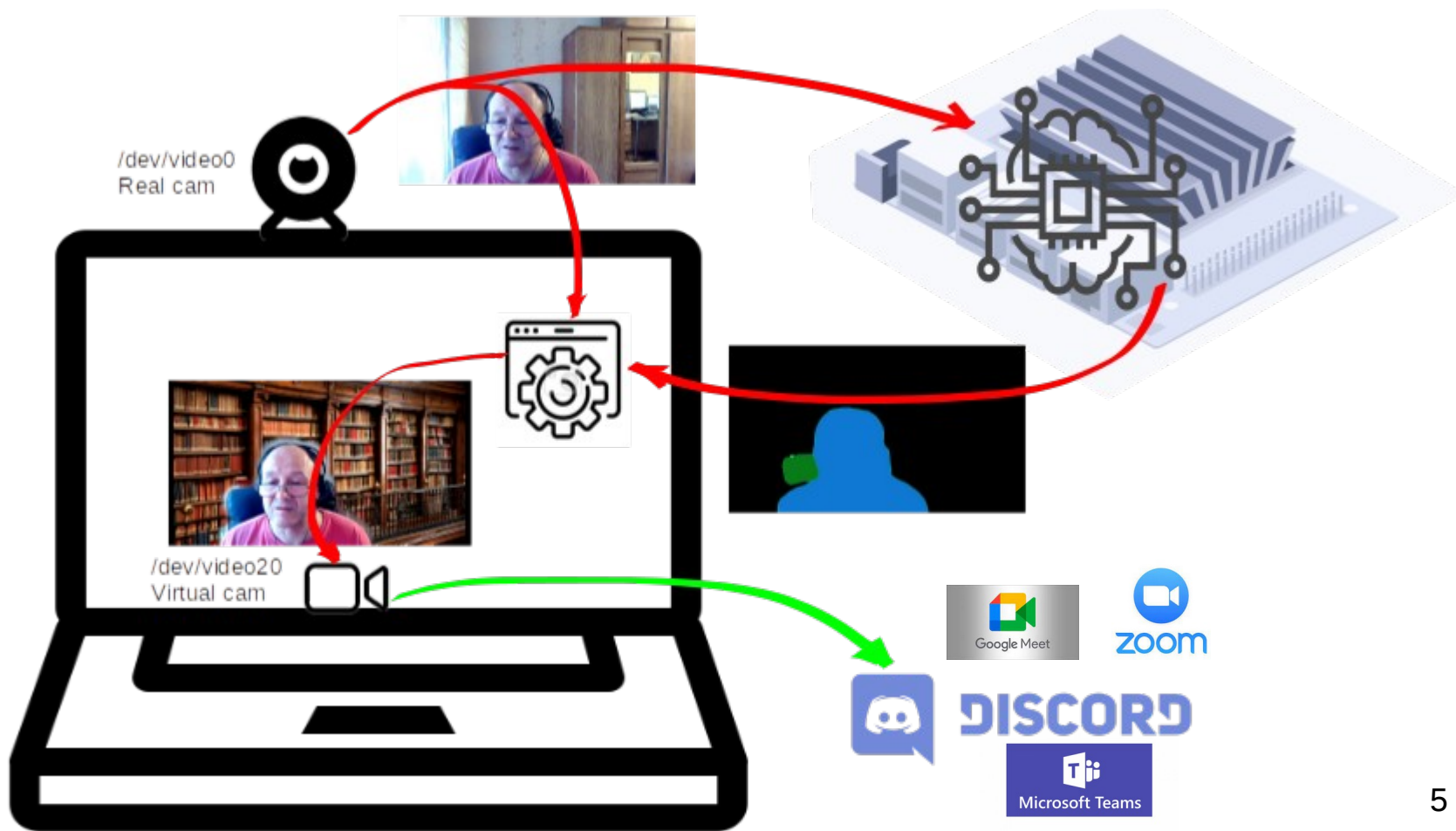


# JETSON AI CERTIFICATIONS

Задачи.

- 1) Захват изображений с веб-камеры как серию изображений
- 2) Семантическая сегментация объектов на изображениях
- 3) Генерация маски для изображения по цветам
- 4) Генерация нового изображений по маскированным изображениям из (1)
- 5) Передать изображения из (4) в виртуальное (aka fake, loopback и т. д.) видеоустройство v4l2loopback
- 6) Настроить программное обеспечение для удаленной связи, чтобы использовать видеоустройство loopback в качестве моей веб-камеры.

# JETSON AI CERTIFICATIONS





# JETSON AI CERTIFICATIONS

## Вызовы.

- Мой ноутбук, как и большинство ноутбуков, слишком медленный для потоковой сегментации - нужно передать обработку на другое устройство
- Для нормального функционирования программы скорость предсказания должна быть менее 50 мс. Для Jetson это слишком быстро - необходима оптимизация
- Благодаря распределенному способу работы потребуется максимально возможный асинхронный и отказоустойчивый программный код.



# JETSON AI CERTIFICATIONS

## Программное и аппаратное обеспечение.

- Ноутбук (client)
  - Dell 5491 i5-8300H 2.3Gh/2\*8G/SSD256
  - OpenSUSE Leap 15.3, 5.3.18-59.19-default
  - Python 3.6.13 on bare metal
- Jetson Nano (server)
  - Jetson B01 NANO 2G
  - Ubuntu 18.04.6 LTS, 4.9.201-tegra
  - Python 3.6.9 on bare metal



# JETSON AI CERTIFICATIONS

## Реализация.

Клиентское приложение состоит из двух модулей **ui.py** и **main.py**. Исходный код <https://github.com/d-yacenko/YAVR/tree/main/client>

- В **ui.py** нет ничего интересного - это просто графический пользовательский интерфейс, написанный на tkinter, работающий с конфигурационным файлом и запускающий с выбранными параметрами **main.py**
- **main.py** выполняет более важные задачи
  - загружает модуль v4l2loopback (необходимо настроить sudoers для текущего пользователя)
  - кадровый захват с веб-камеры и передача кадров с помощью HTTP-запроса на веб-сервер Jetson для сегментации. Здесь нужно ограничить запрос по времени, чтобы добиться работы в реальном времени, и учесть возможность того, что сервер не предоставит результат.
  - получает изображение маски с сервера
  - применяет маску для удаления фона, накладывает статический или динамический фон, применяет специальные эффекты





# JETSON AI CERTIFICATIONS

## Реализация сервера.

Source code - <https://github.com/d-yacenko/YAVR/tree/main/server> . Серверная часть приложения состоит из двух модулей **server.py** и **segmentation.py**

- **server.py** представляет собой реализацию веб сервера, где в обработчике post запроса получается входное изображение и запускается модуль **segmentation.py** для его обработки и получения маски. Здесь используются потоки для оптимизации.
- **segmentation.py** — наиболее интересный модуль во всем проекте. Для сегментации была выбрана pretrained network deeplabv3\_mobilenet\_v3\_large из проекта torchvision. Эта сеть показала наибольшую скорость сегментации. Я решил не обучать собственную сеть, т.к. я вряд ли бы получил более высокую скорость, и именно скорость критически важна в приложении. Кроме того был предпринят еще ряд методов для ускорения inference:
  - Использование неблокирующей pinned memory для ускорения переноса данных GPU → CPU
  - Понижение разрядности данных перед переносом на CPU
  - Вычисление argmax также на GPU
  - Очистка памяти на GPU после inference
  - Note. По ряду причин необходимо было сохранять input image и сгенерированную маску как файл. Для ускорения чтения / записи предохранения storage от выжигания был использован ramdisk. Для его монтирования в /etc/fstab была добавлена запись: **tmpfs /RAM tmpfs size=10m 0 0**



# JETSON AI CERTIFICATIONS

## Результаты.

- Было создано MVP приложения для бюджетного роеалтайм потоковой обработки видео
- Исходный код опубликован - <https://github.com/d-yacenko/YAVR.git>
- Видео демонстрацию можно посмотреть тут: <https://drive.google.com/file/d/1VlaSLGa5P0YtwAdNDjHJI9W7D7Tcl3PS/view?usp=sharing>
- Получившееся приложение полезно мне, имеет самостоятельную ценность, а также может быть реализовано как стартап
- Во время работы приложения нагрузка на оборудование и температура компонентов не высоки (см следующий слайд)

# JETSON AI CERTIFICATIONS

Laptop

```
d.yacenko@notebook...thon/VirtualDesktop
top - 23:29:50 up 1 day, 13:43, 9 users, load average: 0,83, 0,78, 0,59
Tasks: 390 total, 2 running, 383 sleeping, 0 stopped, 5 zombie
%Cpu(s): 9,8 us, 0,5 sy, 0,0 ni, 89,2 id, 0,0 wa, 0,0 hi, 0,5 si, 0,0 st
MiB Mem : 15796,45+total, 1427,371 free, 9050,219 used, 5318,867 buff/cache
MiB Swap: 0,000 total, 0,000 free, 0,000 used. 4513,113 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
32114	d.yacenko+	20	0	2706000	724132	54408	S	61,13	4,477	2:47.20	python3
24409	d.yacenko+	20	0	20,643g	209148	108064	S	8,970	1,293	1:43.38	chrome
2065	d.yacenko+	20	0	2179480	439868	387728	S	2,990	2,719	93:23.44	X
4155	d.yacenko+	20	0	16,500g	153824	93660	S	0,997	0,951	38:31.94	chrome
30318	root	0	-20	0	0	0	I	0,997	0,000	0:02.39	kworker/u17:2-rb_allocator
2389	d.yacenko+	20	0	184832	6372	4916	S	0,664	0,039	0:00.63	dconf-service
4108	d.yacenko+	20	0	16,910g	612276	224884	S	0,664	3,785	78:04.63	chrome
7746	d.yacenko+	20	0	5484776	277988	22172	S	0,664	1,719	1:31.22	java

```
root@dyacenko-desktop:/home/dyacenko
top - 22:29:48 up 21 days, 7:17, 4 users, load average: 0,56, 0,41, 0,18
Tasks: 265 total, 1 running, 264 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6,6 us, 8,8 sy, 0,0 ni, 83,8 id, 0,0 wa, 0,4 hi, 0,4 si, 0,0 st
KiB Mem : 4051080 total, 114136 free, 3546264 used, 390680 buff/cache
KiB Swap: 2025536 total, 1451624 free, 573912 used. 323912 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5226	dyacenko	20	0	12,281g	2,592g	667972	S	46,9	67,1	62:46.67	python3
738	root	20	0	0	0	0	S	9,6	0,0	31:15.51	nvmmap-bz
25410	root	20	0	10352	3640	2848	R	1,0	0,1	0:02.14	top
1101	root	-51	0	0	0	0	S	0,7	0,0	194:06.86	irq/74-tegradc.
5293	root	20	0	0	0	0	S	0,7	0,0	29:19.33	nvgpu_channel_p
835	root	-51	0	0	0	0	S	0,3	0,0	4:55.58	irq/69-host_syn
6553	gdm	20	0	495936	9148	1944	S	0,3	0,2	65:54.74	gsd-color
23865	root	20	0	0	0	0	S	0,3	0,0	0:04.35	kworker/1:2
24376	root	20	0	0	0	0	S	0,3	0,0	0:00.38	kworker/u8:2
24585	root	20	0	0	0	0	S	0,3	0,0	0:00.47	kworker/u8:0

```
d.yacenko@notebook...thon/VirtualDesktop
Adapter: ISA adapter
Package id 0: +59.0°C (high = +100.0°C, crit = +100.0°C)
Core 0: +59.0°C (high = +100.0°C, crit = +100.0°C)
Core 1: +58.0°C (high = +100.0°C, crit = +100.0°C)
Core 2: +57.0°C (high = +100.0°C, crit = +100.0°C)
Core 3: +57.0°C (high = +100.0°C, crit = +100.0°C)

BAT0-acpi-0
Adapter: ACPI interface
in0: +8.28 V
curr1: +0.00 A

pch_cannonlake-virtual-0
Adapter: Virtual device
temp1: +62.0°C

notebook:/home/d.yacenko/work/python/VirtualDesktop #
```

```
root@dyacenko-desktop:/home/dyacenko
No passwd entry for user 'sensors'
dyacenko@dyacenko-desktop:~$ sudo su
root@dyacenko-desktop:/home/dyacenko# 120OpenMi
bash: 120OpenMi: command not found
root@dyacenko-desktop:/home/dyacenko#
root@dyacenko-desktop:/home/dyacenko# sensors
thermal-fan-est-virtual-0
Adapter: Virtual device
temp1: +42.8°C

root@dyacenko-desktop:/home/dyacenko#
```

Jetson nano



# JETSON AI CERTIFICATIONS

## Выводы.

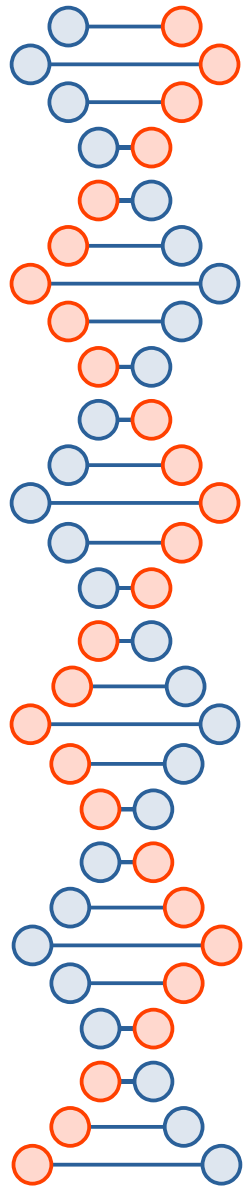
- Устройства уровня Jetson Nano можно использовать для не очень больших реальных задач AI/ML.
- Частично Jetson Nano может работать как edge устройство, например, на стыке IoT и AI.



# JETSON AI CERTIFICATIONS

## Мотивация.

- Почему я начал заниматься этим проектом? Мне стало интересно изучить возможности Jetson nano и мои возможности AI/ML на специализированном оборудовании.
- Почему я получаю сертификат Jetson AI Ambassador? В первую очередь мне, как преподавателю сертификат пригодится как подтверждение статуса. Во-вторых, возможно, программа Ambassador даст мне возможность получить другое оборудование Nvidia для исследований - например, Xavier NX или какое-то другое.



Thanks for attention!