Исследование производительности нейросетевых компонент в SLAM-алгоритмах на встраиваемых устройствах

Киселёв Михаил Николаевич, группа 21.Б07-мм

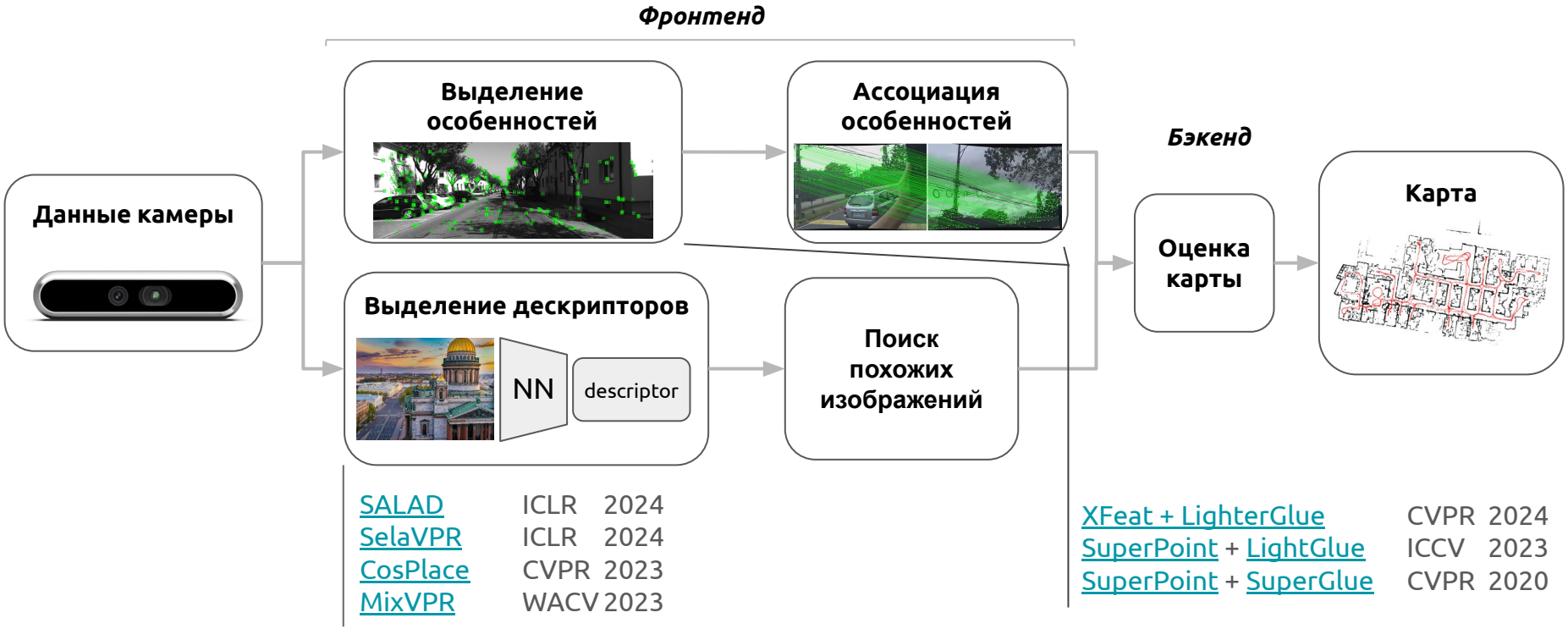
***Научный руководитель**: Старший преподаватель кафедры СП, к.т.н. Ю. В. Литвинов*

***Консультант**: Руководитель исследовательской группы, лаборатория “ИИ для автономных систем”, АНОО ВО «Сколковский институт науки и технологий», А. В. Корнилова*

***Рецензент:** ведущий научный сотрудник, МФТИ, д.ф.-м.н. С. Э. Парсегов*

СПбГУ 2025

Введение. Принцип работы SLAM



Нейросетевые методы более **точны**,
но более **требовательны** к вычислительным ресурсам

Цель и задачи

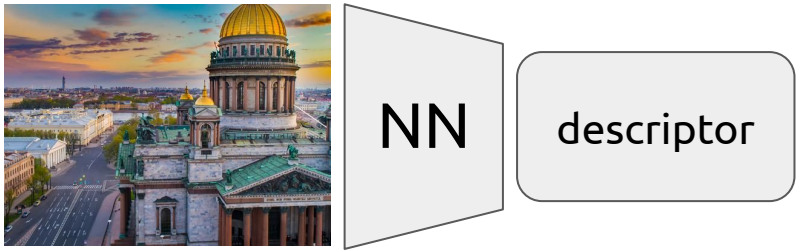
Цель: Выполнить исследование производительности популярных методов *выделения и ассоциации особенностей* и *выделения дескрипторов*, основанных на нейронных сетях, на одноплатных компьютерах с аппаратными ускорителями для нейронных сетей, имеющихся в наличии у Лаборатории

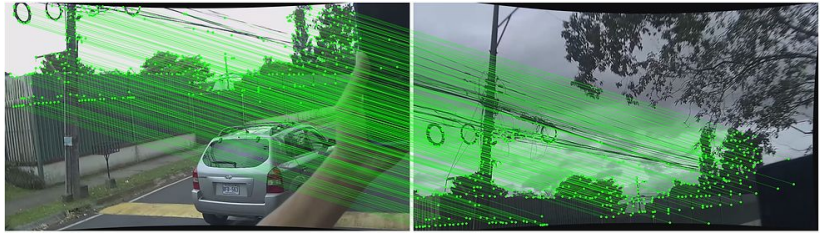
Задачи:

- Выполнить обзор
  - методов выделения и ассоциации особенностей и выделения глобальных дескрипторов, основанных на нейронных сетях
  - распространенных встраиваемых платформ с ускорителями для запуска нейронных сетей
  - техник квантования нейронных сетей
- Реализовать инструмент для измерения производительности алгоритмов поддерживающий запуск на выбранных платформах
- Произвести измерения производительности алгоритмов, а также их квантованных версий на имеющихся у Лаборатории платформах

Обзор. Нейросетевые компоненты

- Методы выделения дескрипторов:
  - SALAD 2024
  - SelaVPR 2024
  - CosPlace 2023
  - EigenPlaces 2023
  - MixVPR 2023
  - NetVLAD 2015
- Методы выделения и ассоциации особенностей
  - XFeat + LighterGlue 2024
  - SuperPoint + LightGlue 2023
  - SuperPoint + SuperGlue 2020

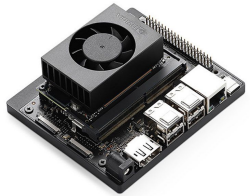




Обзор. Встраиваемые платформы

Производитель	Модель	Ускоритель	Заявленная Производительность	Формат
Nvidia Jetson	Nano	Nvidia GPU	0.47 TFLOPS	SBC
	Xavier AGX		32 TOPS	SBC
	Orin NX		100 TOPS	SBC
Intel	NCS2	VPU	4 TOPS	USB-модуль
	Altera Series	FPGA	10-40 TOPS	PCI-модуль, чип
Rockchip	RK3588	NPU	6 TOPS	SBC
Hailo	Hailo 8	TPU	26 TOPS	SBC, M.2-, PCI-модуль
Google	Coral	TPU	4 TOPS	SBC, USB-модуль, чип
Qualcomm	Hexagon	TPU	15 TOPS	чип
Huawei	Atlas	TPU	16 TOPS	SBC, PCI-модуль

Nvidia Jetson Xavier AGX

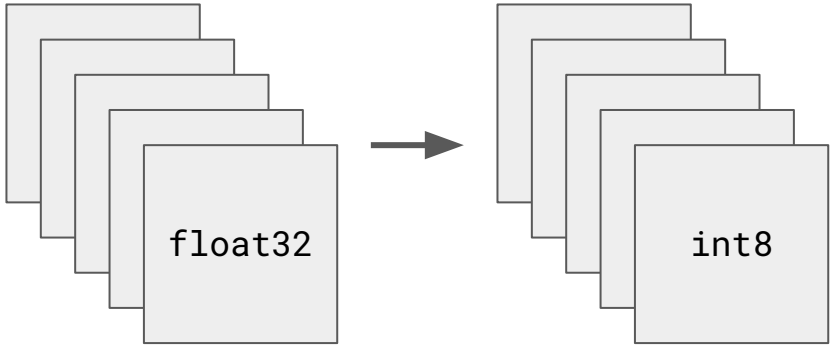


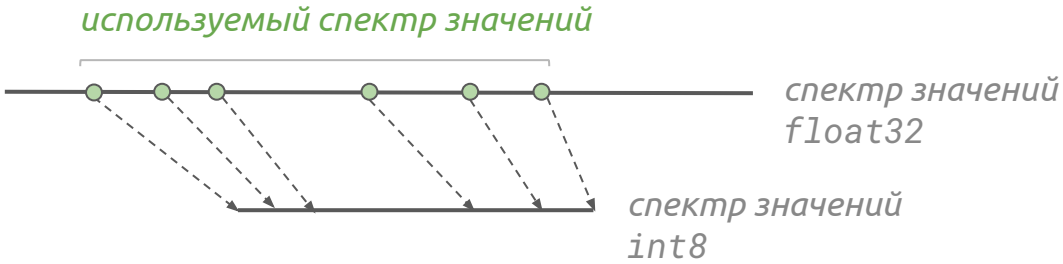
Jetson Orin NX

Nvidia Jetson Nano

Orange Pi 5

Техники квантования





- Квантование с обучением
  - требует процесса обучения
- Динамическое пост-тренировочное
  - не поддерживается выбранными платформами
- **Статическое пост-тренировочное**
  - не требует процесса обучения
  - поддерживается Jetson Orin NX, Orange Pi 5

Требования к инструменту

- Измерение производительности выбранных алгоритмов выделения глобальных дескрипторов и выделения и ассоциации локальных особенностей
- Поддержка преобразования моделей в формат RKNN и запуска на NPU Orange Pi 5
- Поддержка квантования и запуска квантованных моделей

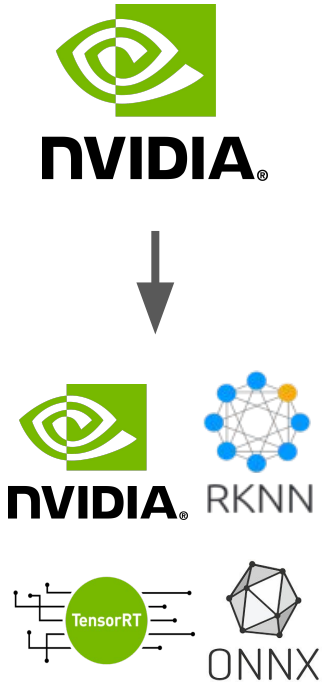
Реализация. aero-vloc как основа решения

aero-vloc[1] поддерживает:

- запуск большей части рассматриваемых алгоритмов
- запуск на спутниковых снимках
- использование Nvidia GPU
- простое расширение новыми методами

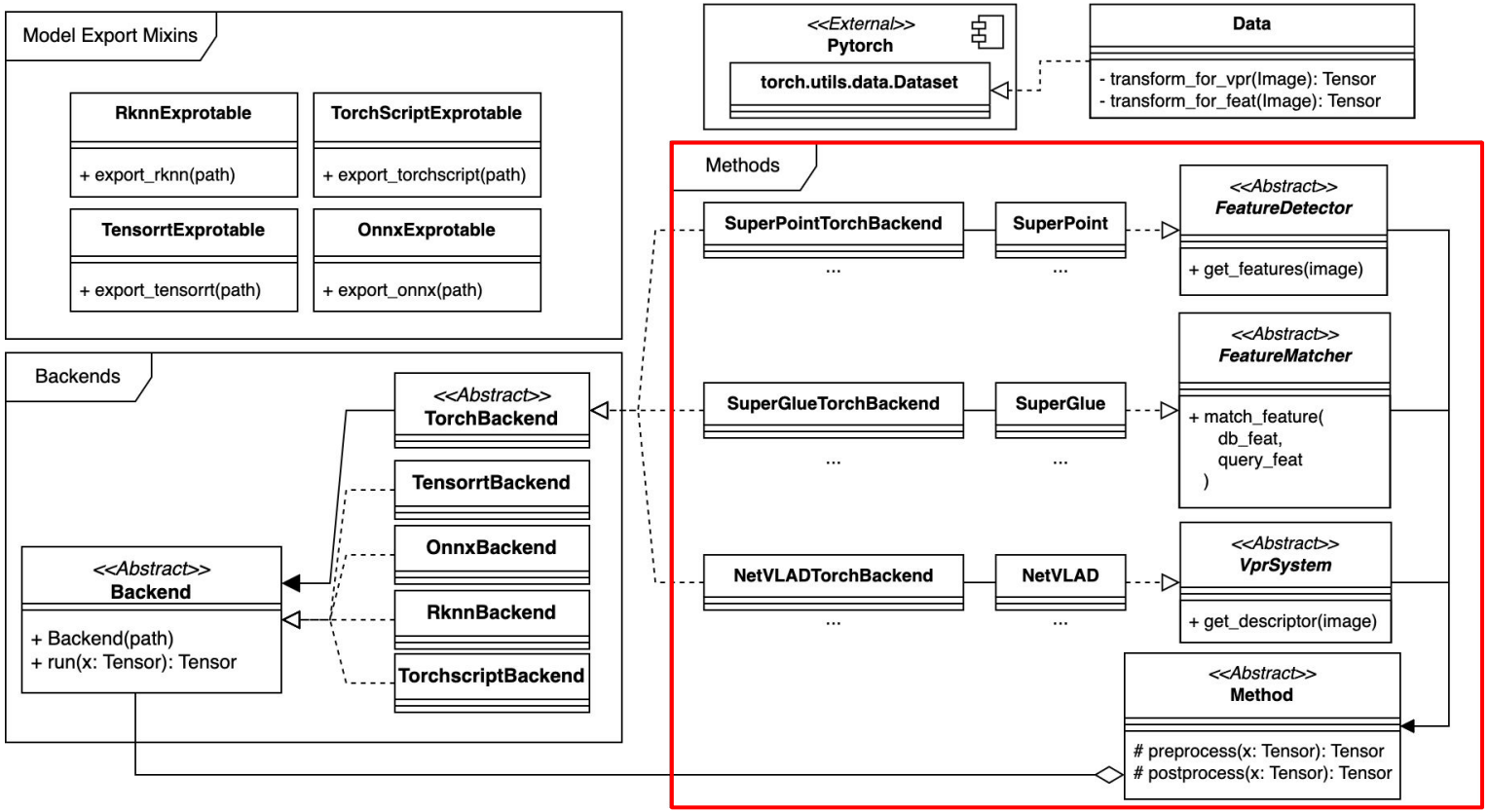
Не поддерживает:

- запуск на не спутниковых наборах данных
- измерение времени выполнения
- использование других ускорителей
- квантование моделей



[1] И. Москаленко. 2024. aero-vloc. URL: https://github.com/prime-slam/aero-vloc. (Дата обращения: 24.05.2025)

Реализация. Архитектура решения

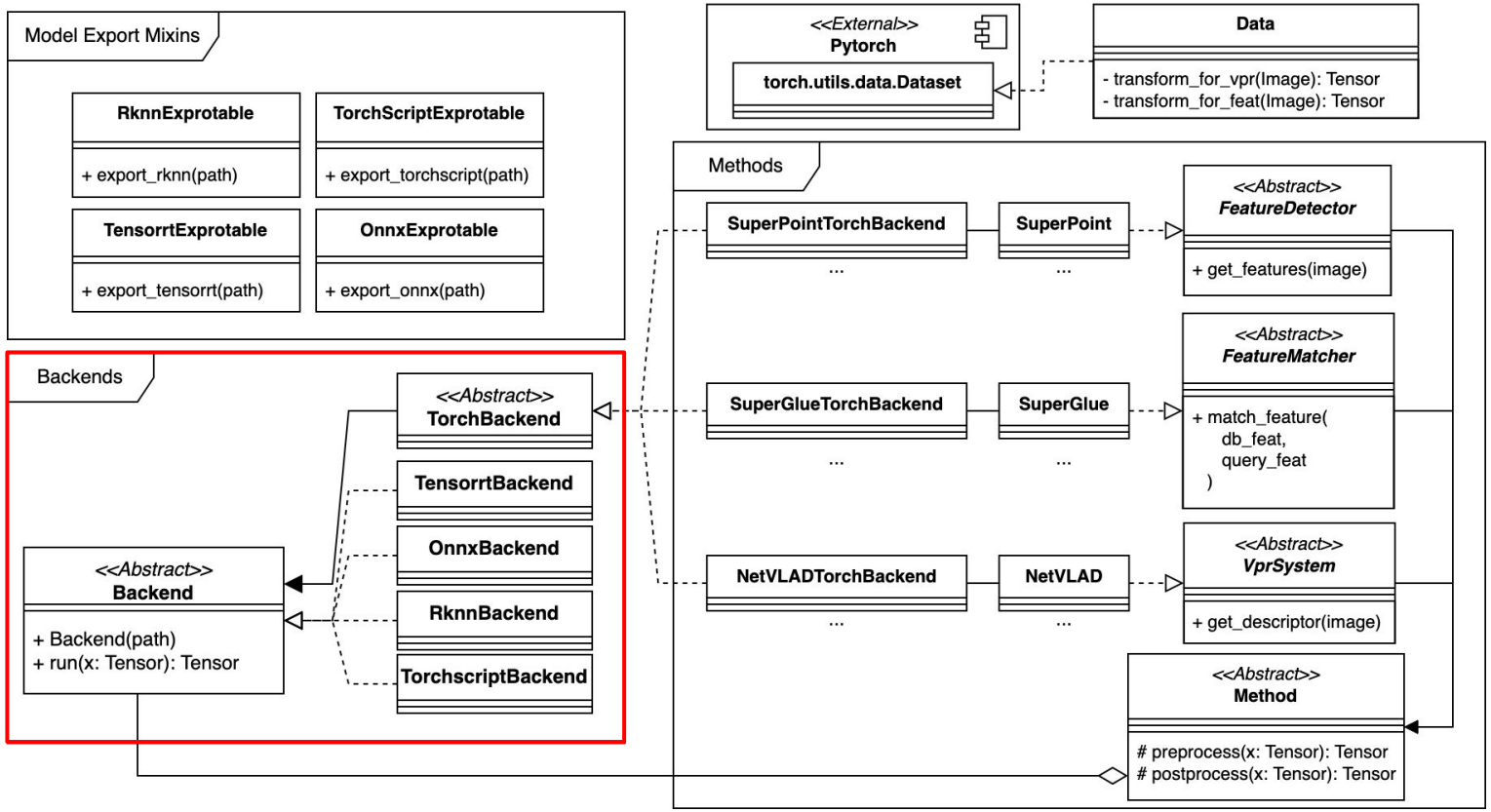


flake8 Black

RKNN TensorRT

ONNX PyTorch

Реализация. Архитектура решения

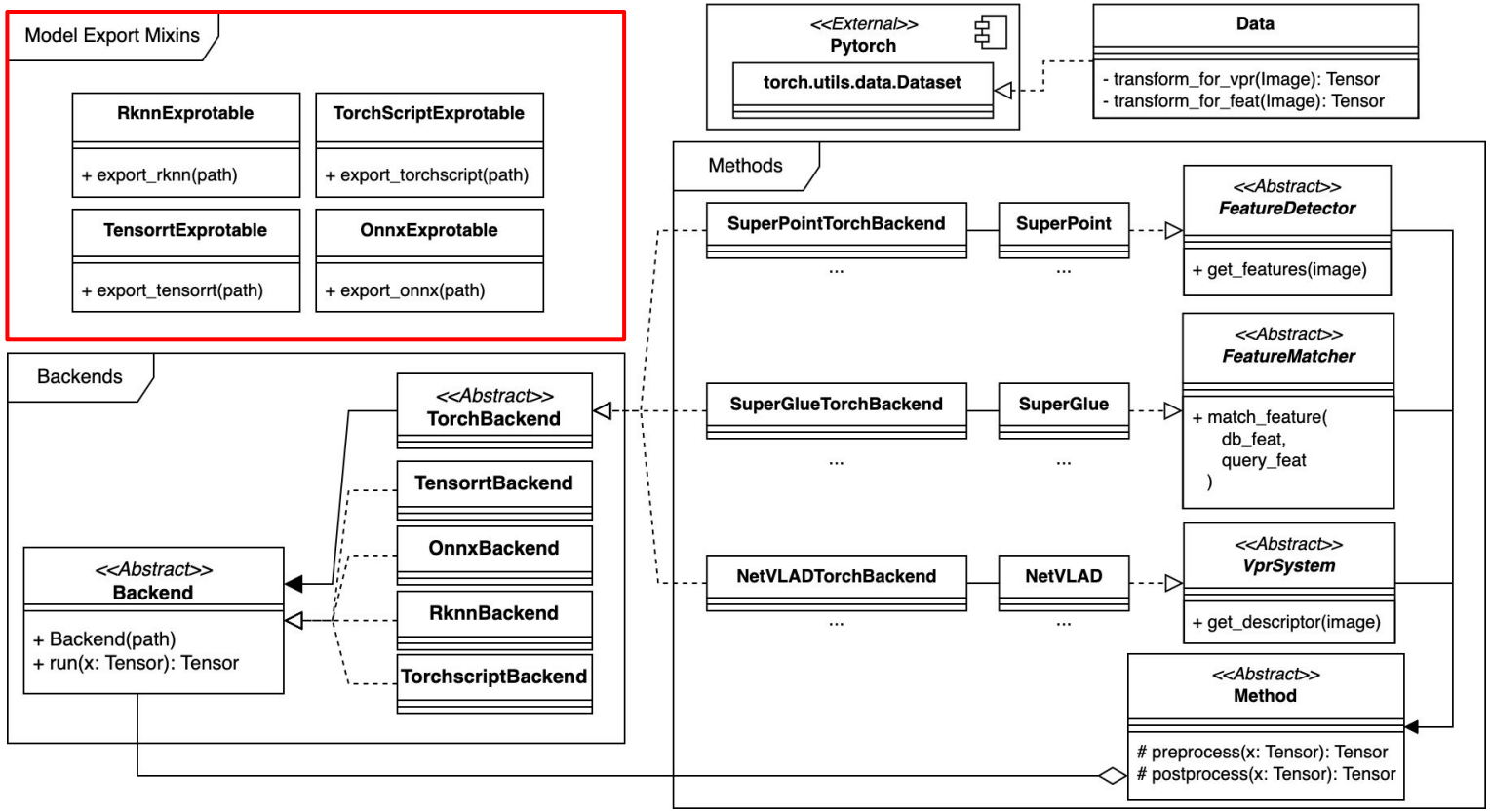


flake8 Black

RKNN TensorRT

ONNX PyTorch

Реализация. Архитектура решения



flake8 Black

RKNN TensorRT

ONNX PyTorch

Эксперимент

RQ1: Как квантование в тип данных `int8` влияет на производительность алгоритмов?

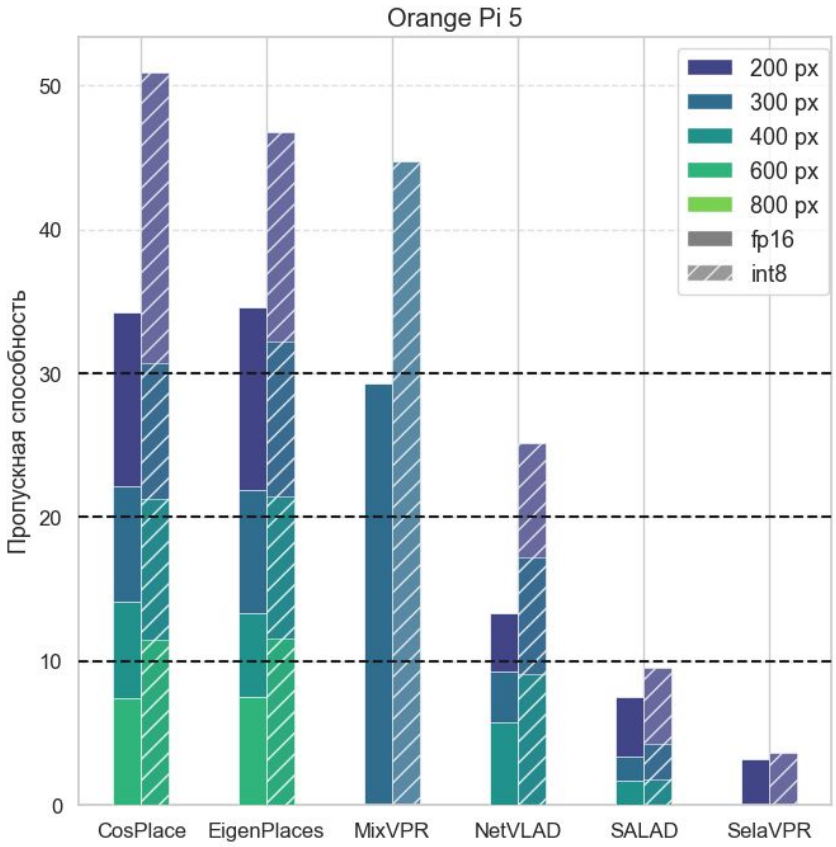
RQ2: Какие методы выделения глобальных дескрипторов и выделения и ассоциации локальных особенностей наиболее производительные на рассмотренных платформах?

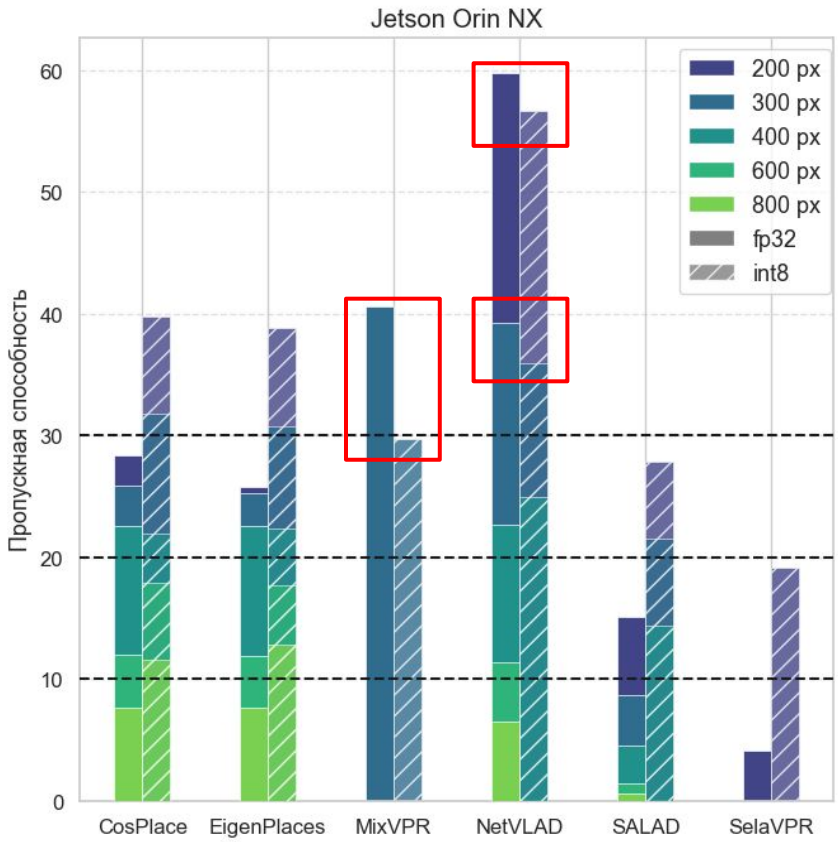
Метрика: средняя пропускная способность (в кадрах в секунду)
Набор данных: St Lucia (3013 изображений)
30 запусков по 1000 изображений + 30 изображений для прогрева

Пример изображения из набора данных St. Lucia

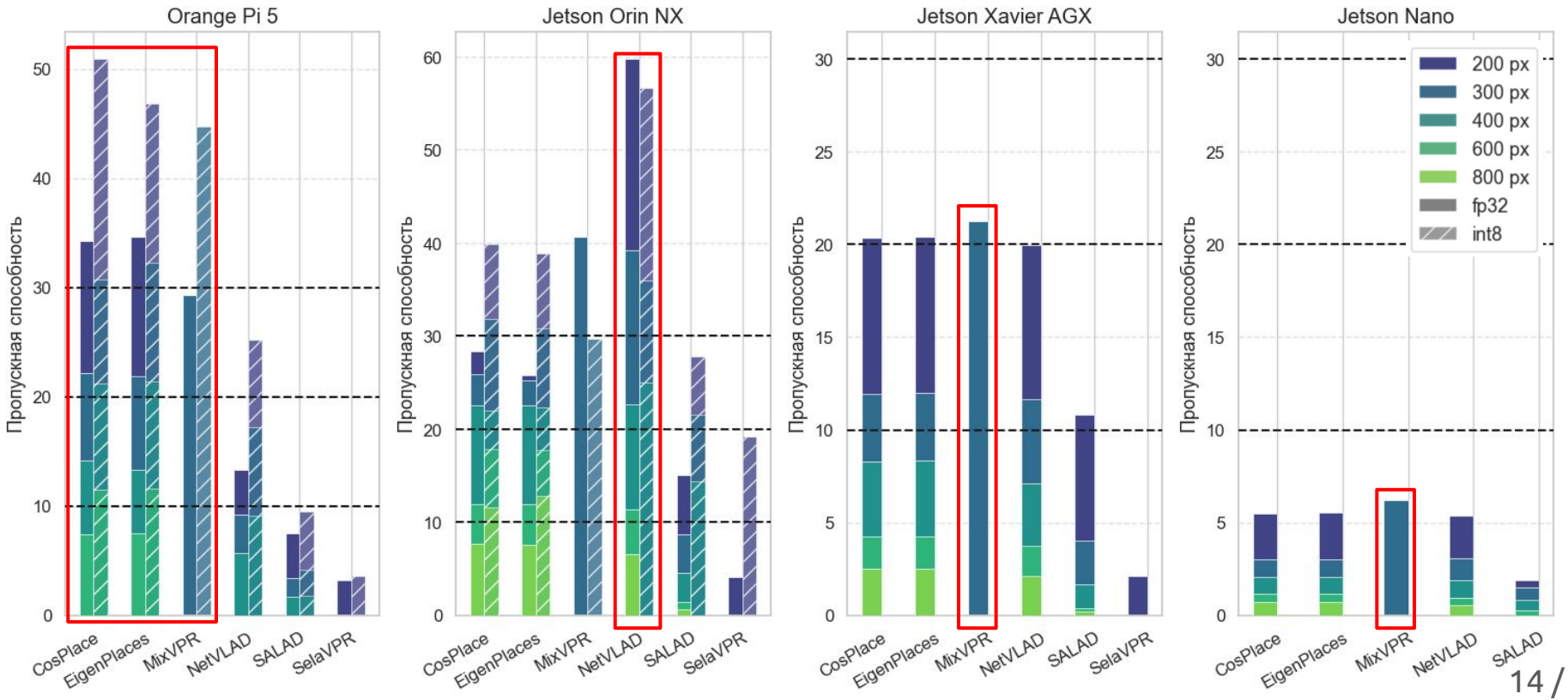
Устройство	Заявленная производительность	ОС	Версия ПО поддержки ускорителя	Версия Python	Версия PyTorch
Jetson Orin NX	100 TOPS	JetPack 6.2	CUDA 12.6	3.10	2.5.0
Jetson Xavier AGX	30 TOPS	JetPack 5.1	CUDA 11.8	3.8	2.0.0
Jetson Nano	0.47 TFLOPS	JetPack 4.6	CUDA 10.2	3.6	1.11.0
Orange Pi 5	6 TOPS	Ubuntu 22.04	RKNNLite 2.3.0	3.10	2.5.0-cpu

RQ1. Влияние квантования на пропускную способность

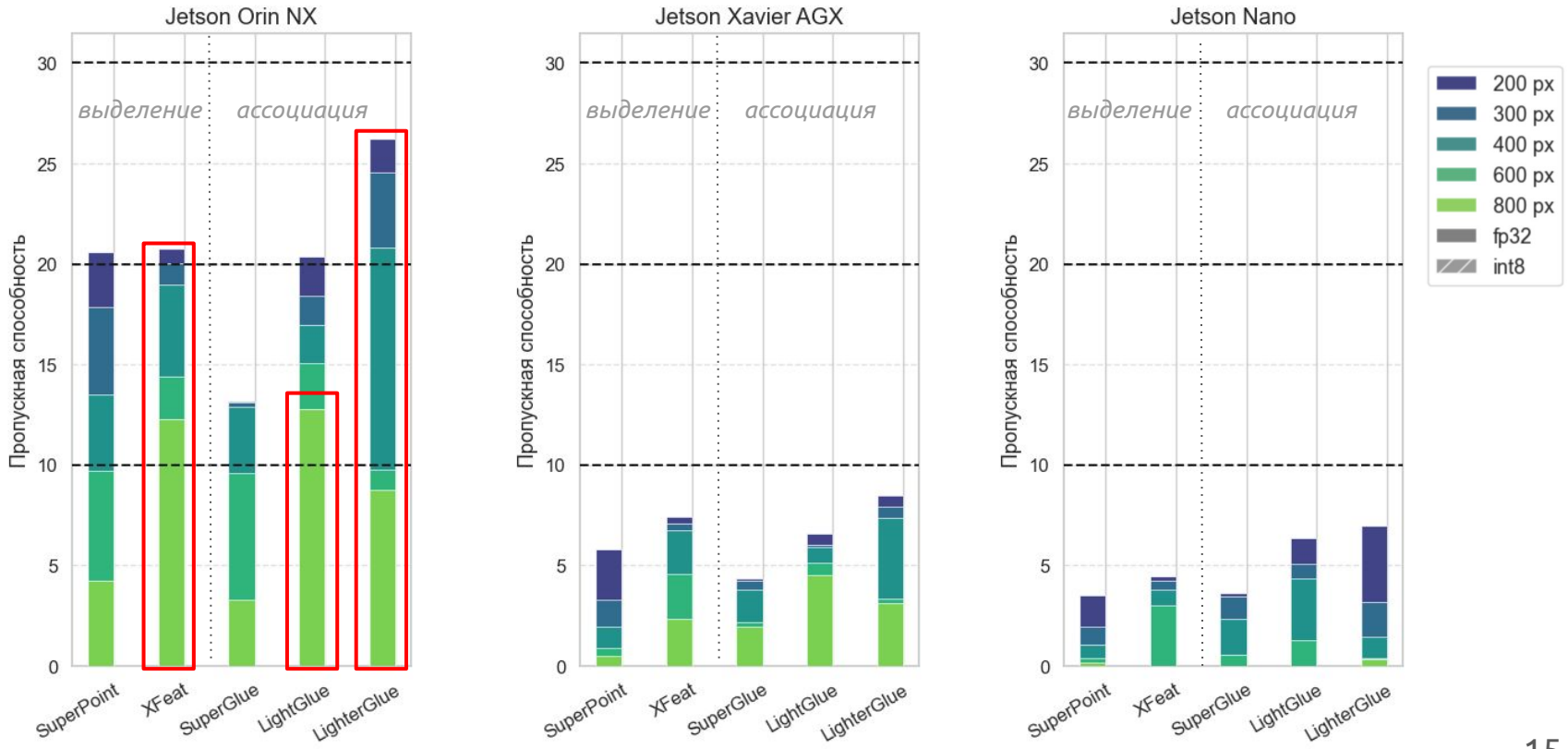




RQ2. Наиболее производительные методы. 1/2



RQ2. Наиболее производительные методы. 2/2



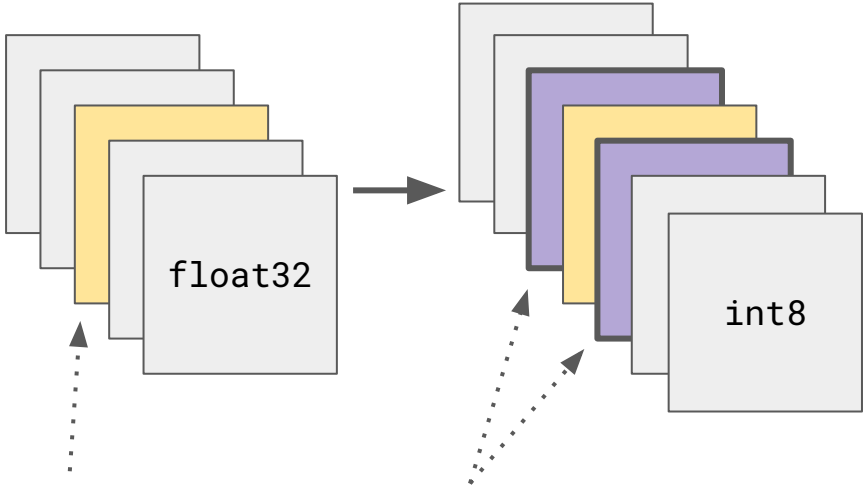
Заключение

- Выполнен обзор
  - методов выделения и ассоциации особенностей и выделения глобальных дескрипторов, основанных на нейронных сетях;
  - распространенных встраиваемых платформ с ускорителями для запуска нейронных сетей;
  - техник квантования нейронных сетей
- Модифицирован инструмент aero-vloc для измерения времени при запуске на произвольных наборах данных
- Выполнены измерения производительности оригинальных и квантованных версий выбранных методов на платформах Nvidia Jetson Orin NX 16, Orange Pi 5, Nvidia Jetson Nano, Nvidia Jetson Xavier AGX

Исходный код доступен в репозитории
https://github.com/true-real-michael/nn-slam-benchmark

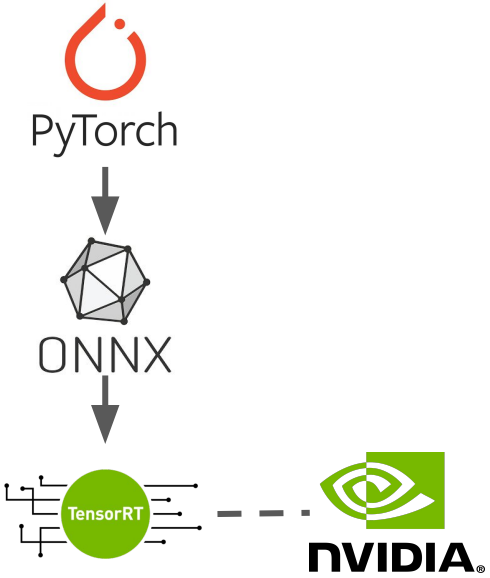
Дополнительные материалы

Особенности квантования

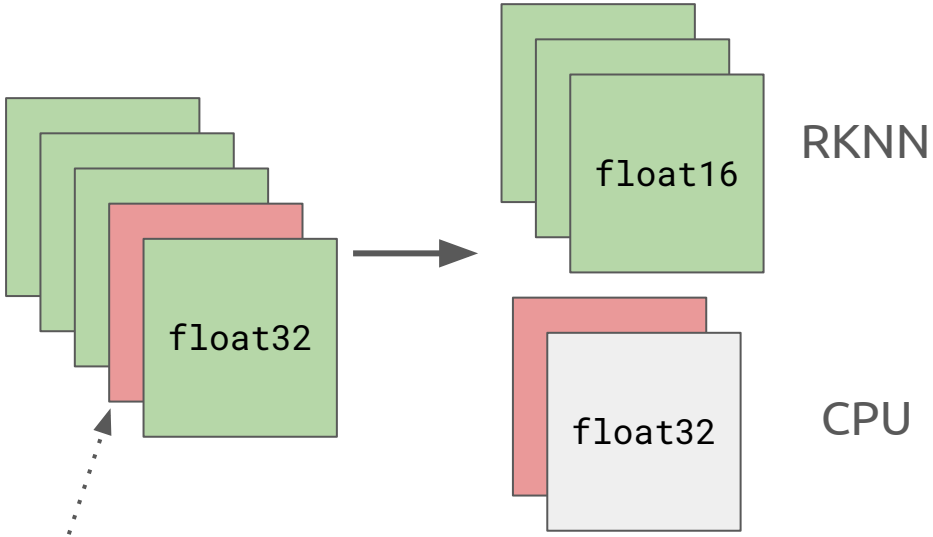


только float32

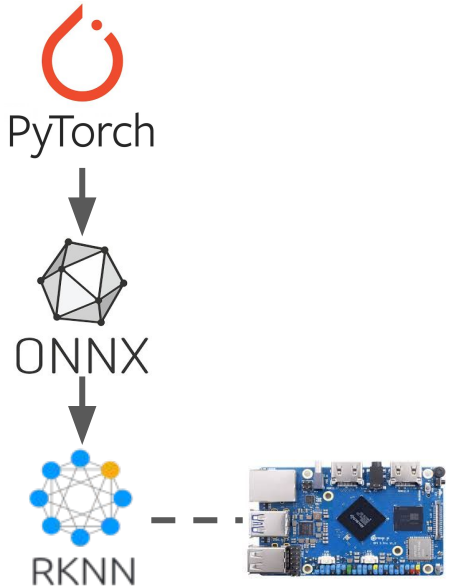
преобразование типов



Особенности формата RKNN

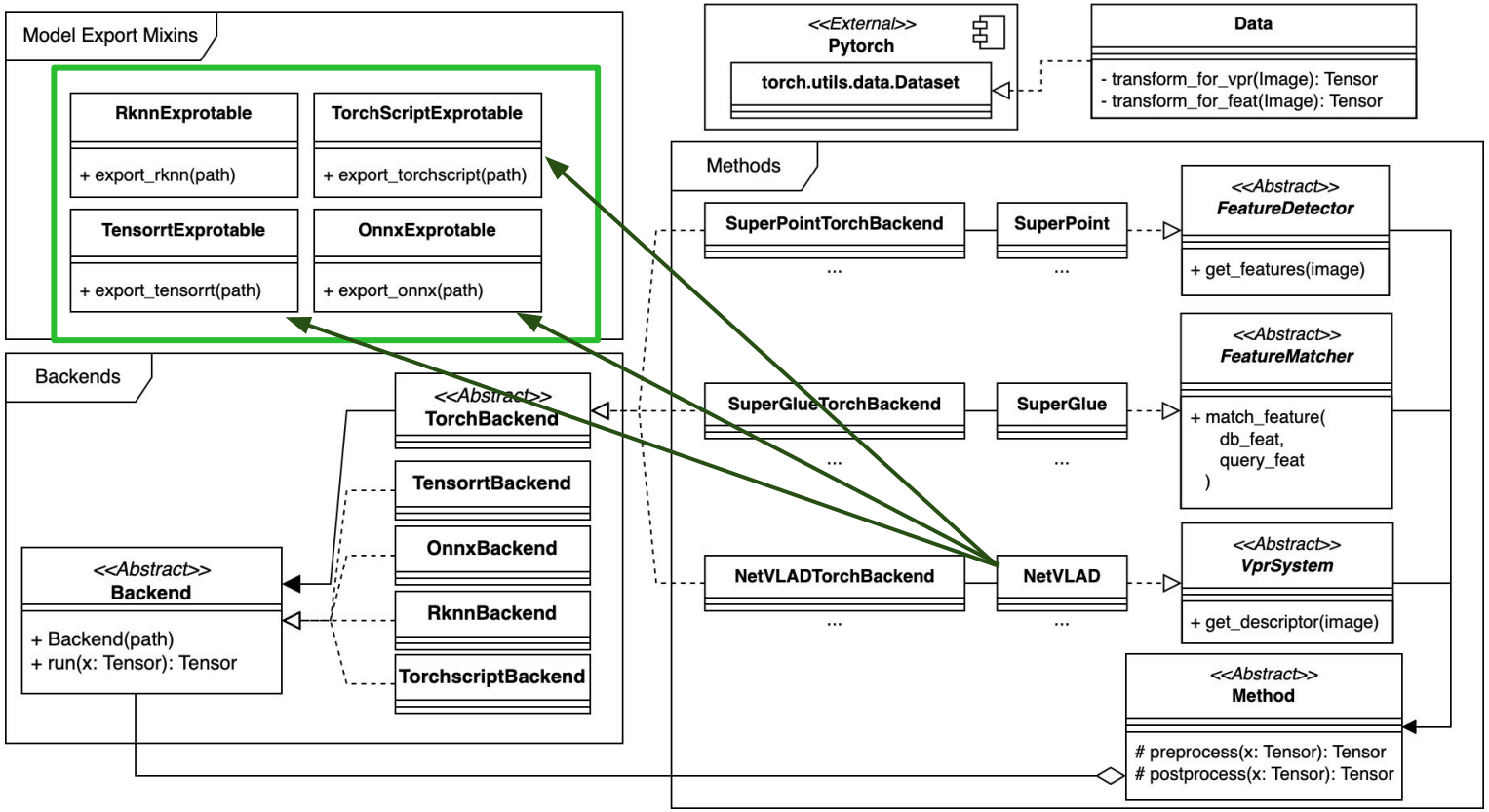


неподдерживаемый слой



Orange Pi 5

Реализация. Архитектура решения



flake8 Black

RKNN TensorRT

ONNX PyTorch