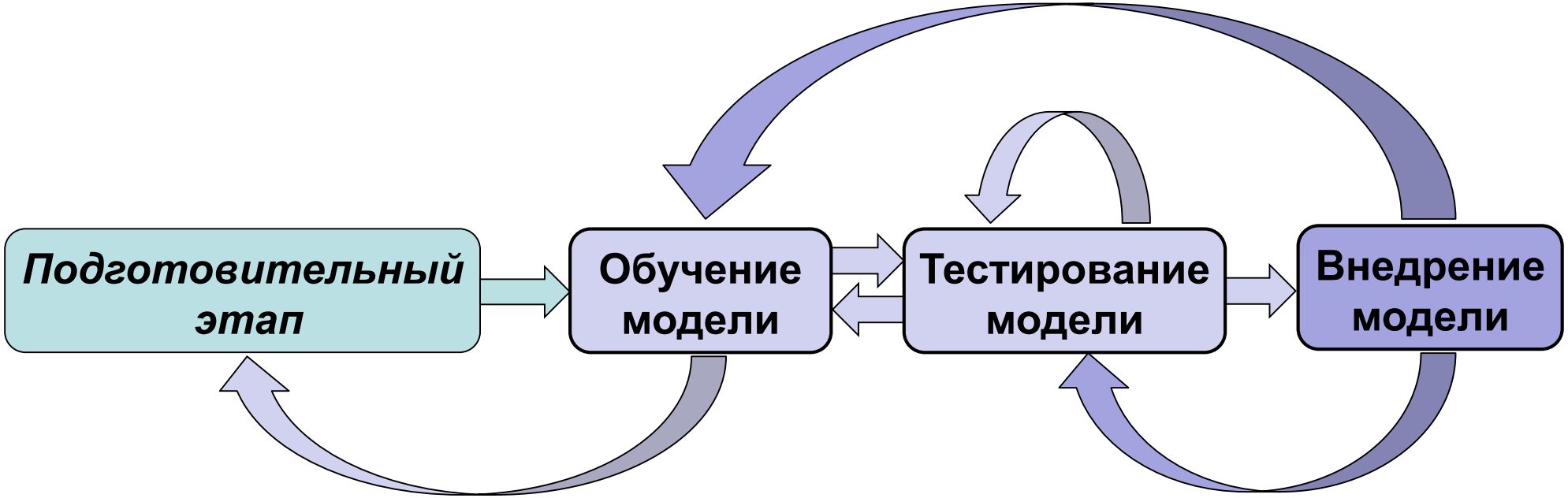
Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского
Институт информационных технологий, математики и механики

Бенчмаркинг глубоких моделей на RISC-V-процессорах

И.С. Мухин, Ю.А. Родимков, Е.П. Васильев,
В.Д. Волокитин, А.К. Сидорова, Е.А. Козинов,
И.Б. Мееров, В.Д. Кустикова

Введение

- Цикл решения задач с использованием глубоких моделей:



Обзор литературы

Подготовка данных

Подготовка тестовой инфраструктуры

Разработка топологии

Обучение модели

Верификация модели

Определение качества решения задачи

Анализ производительности

Оптимизация модели

Многократный вывод

Цели

- ***Цель*** – сравнить производительность вывода нескольких широко известных моделей глубокого обучения на RISC-V-процессорах и оценить возможности использования тензорных компиляторов* на примере Apache TVM

- **** Тензорные компиляторы (tensor compiler)*** – инструменты, которые предназначены для ускорения вывода нейронных сетей за счет оптимизации модели на разных уровнях представления и компиляции под конкретное устройство

Сравниваемые фреймворки

- **OpenVINO toolkit** – инструмент для эффективного развертывания нейронных сетей
  - Собирается, но не оптимизирован и не распараллелен для RISC-V
- **Apache TVM** – активно развивающийся тензорный компилятор с открытыми исходными кодами
  - Вывод оптимизирован для запуска на RISC-V
- **TensorFlow Lite** – библиотека для развертывания глубоких нейросетевых моделей на мобильных устройствах и микроконтроллерах
  - Оптимизирована под RISC-V за счет интеграции низкоуровневой библиотеки примитивов XNNPACK

Постановка задачи классификации изображений

- Задача классификации изображений состоит в том, чтобы поставить в соответствие изображению класс объектов, содержащихся на этом изображении
- ***Входные данные:***
  - Трехканальное изображение I в формате RGB, которое представляется трехмерной матрицей с пространственными размерами w и h (ширина и высота)
  - Каждый элемент – интенсивность пикселя в диапазоне от 0 до 255 (или от 0 до 1, если выполнена нормировка)
- ***Выходные данные:***
  - Вектор вещественных значений
  - Длина вектора = число категорий изображений
  - Каждый элемент вектора – достоверность принадлежности изображения определенному классу

Тестовая инфраструктура

CPU, RAM	CPU TH1520 (4 RISC-V 64GCV cores C910) 1.848Ghz, 8 GB
Операционная система	Debian 12, kernel 5.10.113-g7b352f5ac2ba
Фреймворки для компиляции и конвертации моделей (выполняется на x86)	OpenVINO toolkit 2023.3 (from pip repository) Apache TVM 0.15 (built from source codes) with llvm 10.0 (from Ubuntu apt repository) TensorFlow 2.14.0 (from pip repository) TensorFlow Lite 2.14.0 (from pip repository)
Фреймворки для вывода на RISC-V	OpenVINO toolkit 2023.3 (built from source codes) Apache TVM 0.15 (built from source codes) TensorFlow Lite 2.14.0 (built from source codes)

Тестовые модели

- ***DenseNet-121*** и ***GoogleNet-v4***

- **Источник:** репозиторий OpenVINO – Open Model Zoo (OMZ) [https://github.com/openvinotoolkit/open_model_zoo]
- **Исходный формат:** TensorFlow
- **Конвертация и компиляция моделей:**
  - Формат OpenVINO: Model Converter из OMZ
  - Формат Apache TVM: Python-пакет tf2onnx + конвертер из ONNX в Apache TVM из Deep Learning Inference Benchmark [https://github.com/itlab-vision/dl-benchmark]
  - Формат TensorFlow Lite: конвертер из ONNX в TensorFlow Lite из Deep Learning Inference Benchmark

Тестовые данные

- Валидационная выборка набора данных ImageNet [https://www.image-net.org] (размер – 50 000 изображений)

- **Анализ качества классификации:**
  - Первые 1 000 изображений из выборки

- **Анализ производительности:**
  - Случайные 32 изображения из выборки
  - При необходимости используются повторно
- ***Примечание:*** конкретные данные не влияют на вычислительную сложность

Показатели качества

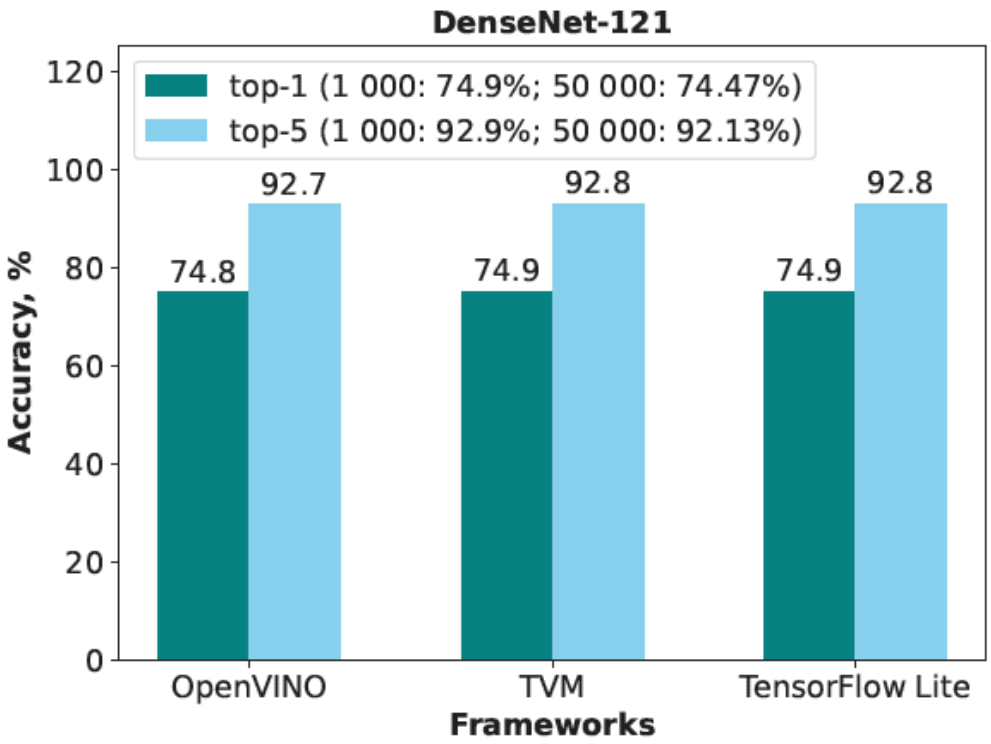
- ***Точность top-K*** – отношение числа правильно проклассифицированных изображений к общему их количеству
  - N – количество категорий изображений
  - Выход модели – вектор достоверностей $p^j = \left(p_1^j, p_2^j, \dots, p_N^j\right)$ для каждого изображения $I_j, j = \overline{1, S}$ в выборке, где p_i^j – достоверность того, что изображение I_j принадлежит классу i
  - Если среди K наибольших достоверностей $p_{i_1^j}^j, p_{i_2^j}^j, \dots, p_{i_K^j}^j$ присутствует достоверность, соответствующая искомому классу, то изображение проклассифицировано корректно

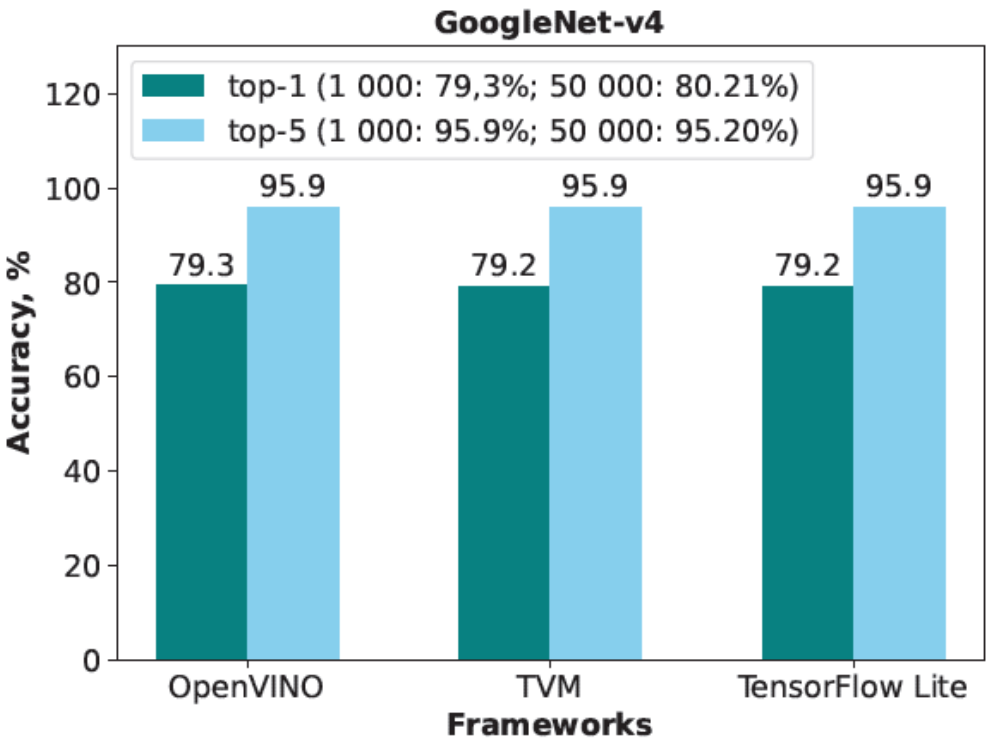
Показатели производительности

- **Эксперимент:**
  - Множество входных примеров разбивается на пачки
  - Прямой проход = решение задачи для пачки
  - Запросы на прямой проход по сети выполняются последовательно, следующий запрос выполняется после завершения предыдущего
  - Число запросов (итераций) = 1 000
  - Для каждого запроса измеряется продолжительность его выполнения

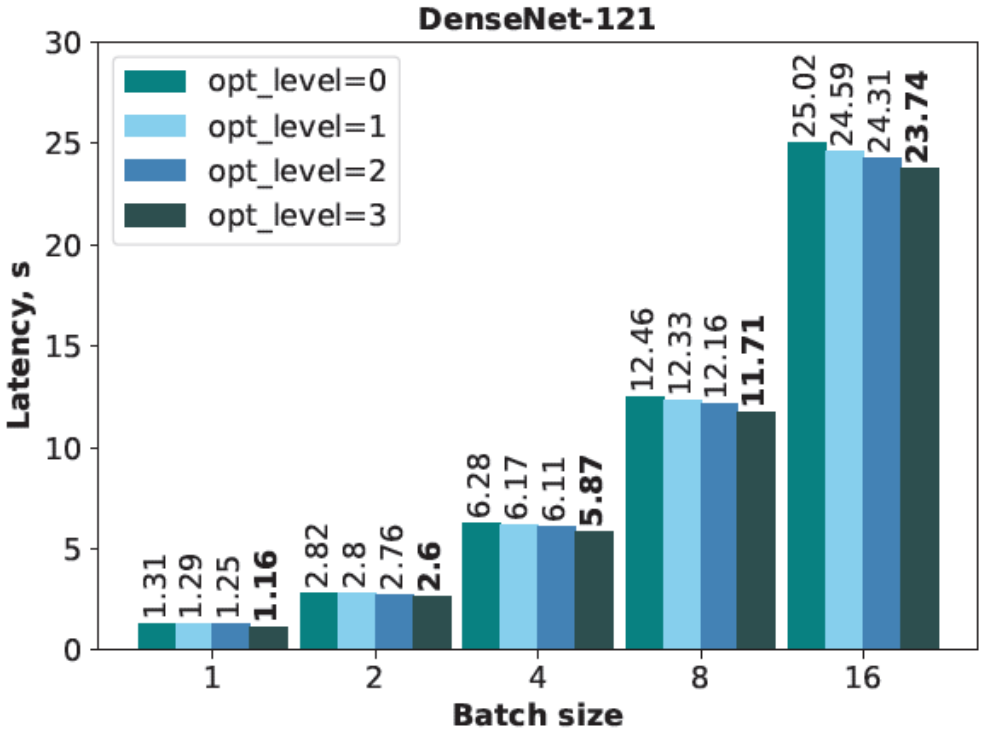
- ***Латентность*** (Latency) – медиана времен выполнения запросов

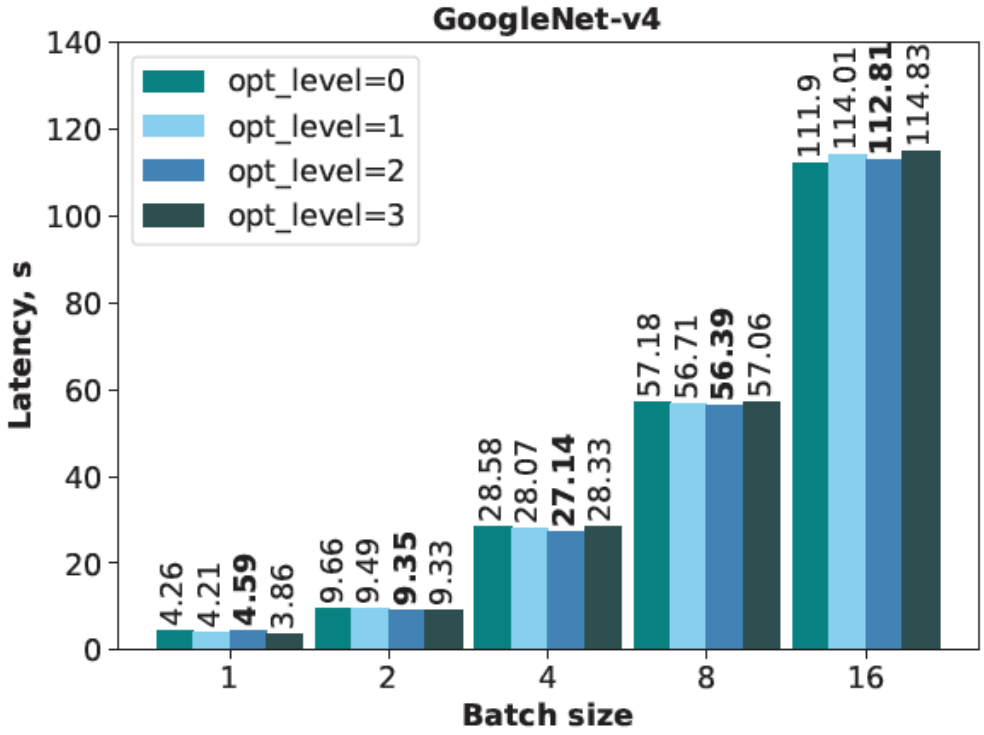
Сравнение качества классификации



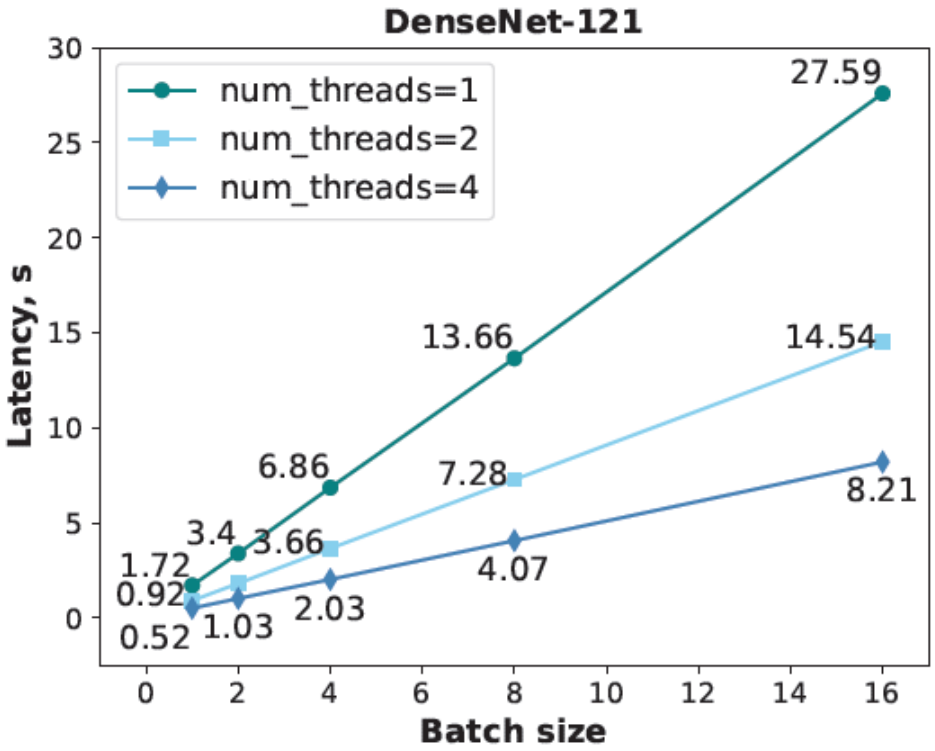


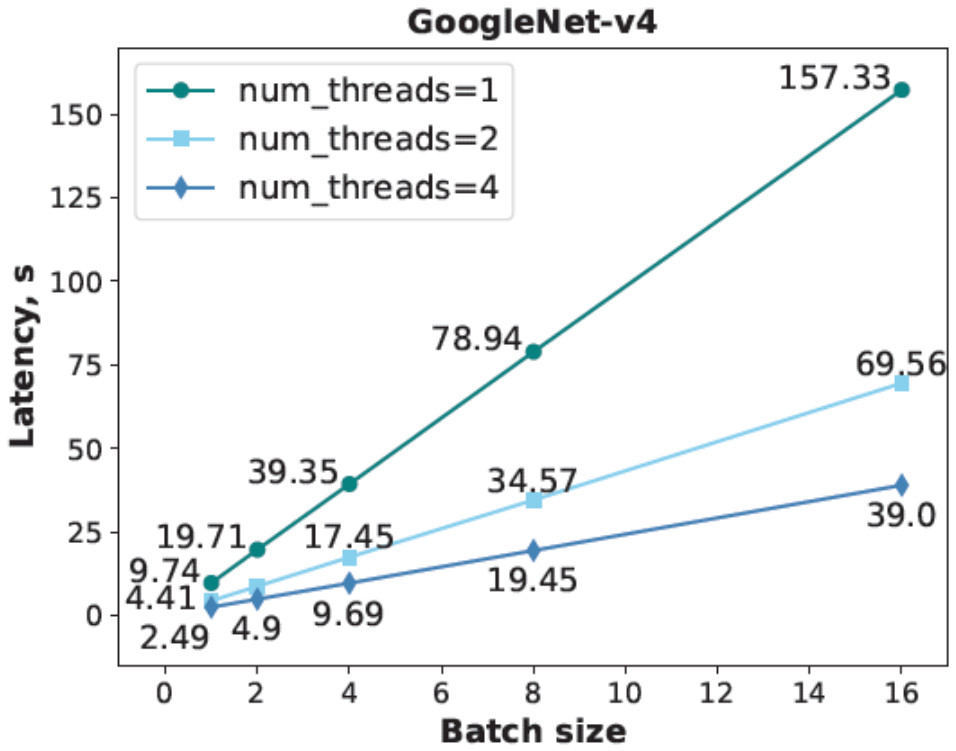
Подбор оптимальных параметров запуска. Apache TVM



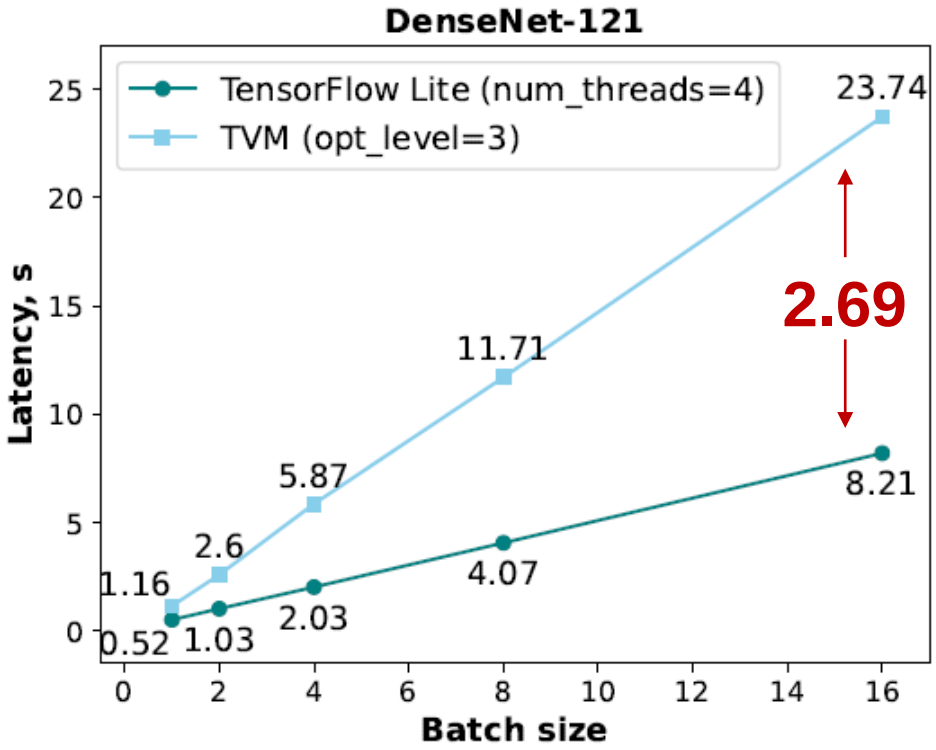


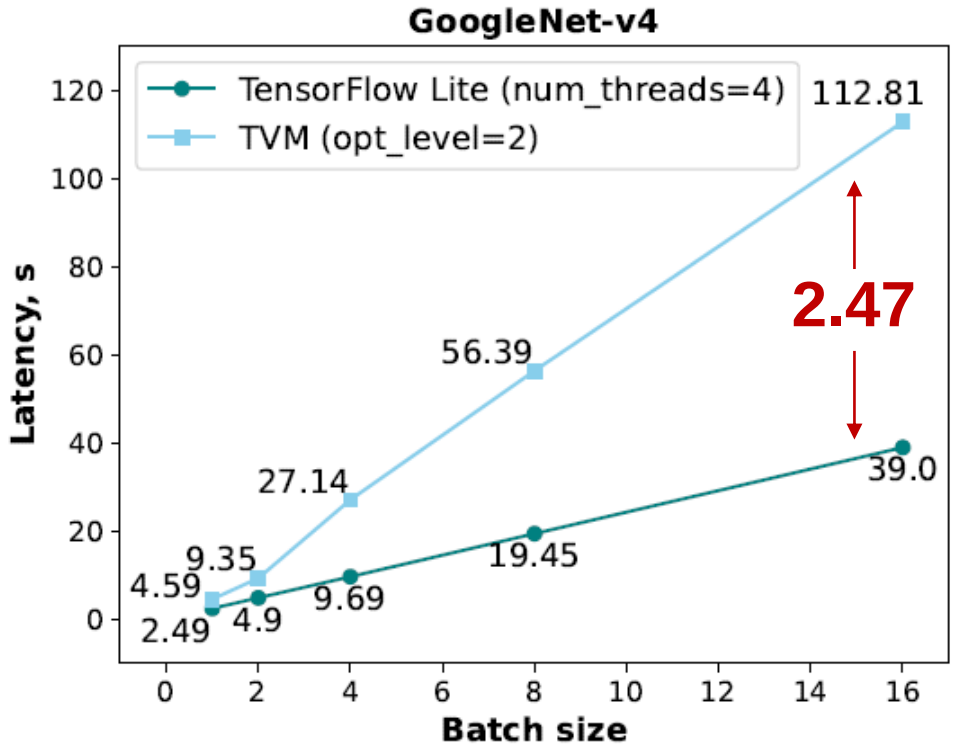
Подбор оптимальных параметров запуска. TensorFlow lite





Лучшие показатели производительности





- OpenVINO toolkit быстро работает на x86-архитектурах, но отстает на 2 порядка на RISC-V

Заключение

- Производительность вывода глубоких моделей на RISC-V-устройствах соответствует их текущим возможностям
- Выполнен анализ производительности вывода и качества классификации на RISC-V для широко известных глубоких моделей DenseNet-121 и GoogleNet-v4
  - Качество классификации на x86 и RISC-V отличается не более, чем на 0.1%, хорошо соотносится с референсными значениями
  - Apache TVM проигрывает TensorFlow Lite в среднем в 2.58 раза
  - OpenVINO отстает на 2 порядка от TensorFlow Lite
- Исходная версия OpenVINO является неоптимизированной и последовательной, сейчас ведется разработка OpenVINO под RISC-V
- Apache TVM активно развивается, и в перспективе сможет приблизиться к TensorFlow Lite