

OpenCV на RISC-V

**Милащенко Максим,
Младший инженер по разработке ПО
искусственного интеллекта,
YADRO**



Почему мы говорим об OpenCV на RISC-V?

План доклада

- **Знакомство с OpenCV**
- **Оптимизации OpenCV под RISC-V**
- **Сборка OpenCV для RISC-V архитектур**
- **Результаты производительности модулей Core и Imgproc на Lichee Pi 4A RISC-V**



OpenCV

- **Открытая библиотека компьютерного зрения**
- **Традиционные алгоритмы компьютерного зрения +
Глубокое обучение (AI)**
- C++/Python/Java/JavaScript/...
- Windows/Linux/macOS/iOS/Android



Модули OpenCV

core. Основные функциональности
imgproc. Обработка изображений
imgcodecs. Чтение и запись файлов изображений
videoio. Видео ввод/вывод
highgui. Графический интерфейс высокого уровня
video. Видеоанализ
calib3d. Калибровка камеры и 3D-реконструкция
features2d. 2D Features Framework
objdetect. Обнаружение объектов
dnn. Модуль глубоких нейронных сетей
ml. Машинное обучение
flann. Кластеризация и поиск в многомерных пространствах
photo. Алгоритмы обработки фотографий
stitching. Склейка изображений
gapi. Графовый API



SIMD/ Vector Extensions / Intrinsics

- **SIMD-128** – SSE (x86_64), NEON (ARM), VSX (POWER), MSA (MIPS), WASM/JS
- **SIMD-256** – AVX (x86_64), LASX (Loongson)
- **SIMD-512** – AVX512 (x86_64)
- **SCALABLE** – RVV (RISC-V), SVE (ARM)

Intrinsics examples:

```
__m128  _mm_add_pd(__m128, __m128)
__mm256 _mm256_add_ps(__m256, __m256)
_m512  _mm512_add_epi64(_m512i, _m512i)
vint8m1_t vadd_vv_i8m1(vint8m1_t,
vint8m1_t, size_t)
```

Single Instruction, Multiple Data

8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8

16 16 16 16 16 16 16 16

32 32 32 32

64 64

128

128-bit register

Универсальные интринсики в OpenCV

Raw intrinsics `#ifdef __SSE2__`
 `#ifdef __AVX__`
 `#ifdef __AVX512__`
 `#ifdef __ARM_NEON__`

Universal intrinsics `#ifdef CV_SIMD128`
 `#ifdef CV_SIMD256`
 `#ifdef CV_SIMD512`

Wide intrinsics `#ifdef CV_SIMD`

Scalable intrinsics `#ifdef CV_SIMD_SCALABLE`

```
for (int i = 0; i < len; <...>)
{
    i += 2;
    _m128i a = _mm_add_epi64(b + c);

    i += 2;
    v_uint64x2 a = b + c;

    i += 4;
    v_uint64x4 a = b + c;

    i += v_uint64::nlanes;
    v_uint64 a = b + c;

    i += Vtraits<v_uint64>::vlanes();
    v_uint64 a = b + c;
}
```

RISC-V RVV support in OpenCV (fixed size)

RVV 0.7.1

```
struct v_int32x4
{
    explicit v_int32x4(vint32m1_t v)
        : val(v) {}
    v_int32x4(int v0, int v1, int v2, int v3)
    {
        int v[] = {v0, v1, v2, v3};
        val = vle_v_i32m1(v, 4);
    }
    vint32m1_t val;
};
```

T-Head 1.12

RVV 1.0

```
struct v_int32x4
{
    explicit v_int32x4(vint32m1_t v)
    {
        vse32_v_i32m1(val, v, nlanes);
    }
    operator vint32m1_t() const
    {
        return vle32_v_i32m1(val, nlanes);
    }
    int val[4]; // vint32m1_t is "sizeless"
                // - can not be stored
};
```

T-Head 2.6.1 (0.7 and 1.0)

LLVM/Clang
RISC-V Collab @rvv_next
Syntacore (LLVM)

Сборка OpenCV RISC-V

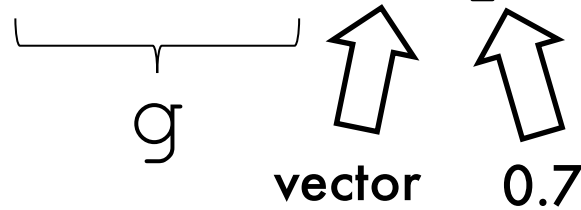
Используемый для анализа:

- T-Head toolchain 1.12.0 (GCC 8.4.0)
 - 20210618, no sources, no binaries
 - RVV 0.7.1, old intrinsics (riscv-vector.h)

Возможно использовать:

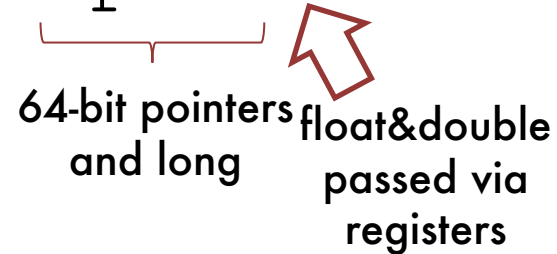
- T-Head toolchain 2.6.0 (GCC 10)
 - 20220906, sources, binaries
 - RVV 0.7.1 and 1.0, new intrinsics (riscv_vector.h)
- RISC-V Collab GCC (*rvv_next* branch)
 - Sources
 - RVV 1.0, new intrinsics
- LLVM/Clang
 - RVV 1.0, new intrinsics, 0.11 since LLVM 16
- Syntacore (LLVM/Clang + GCC)
 - RVV 1.0, new intrinsics (only in LLVM)

```
-march=rv64imafdcv0p7
```



- **Default:** -march=rv64imafdc (rv64gc)
- **Vector:** rv64gcv / **0.7:**

```
rv64gcv0p7 -mabi=lp64d
```

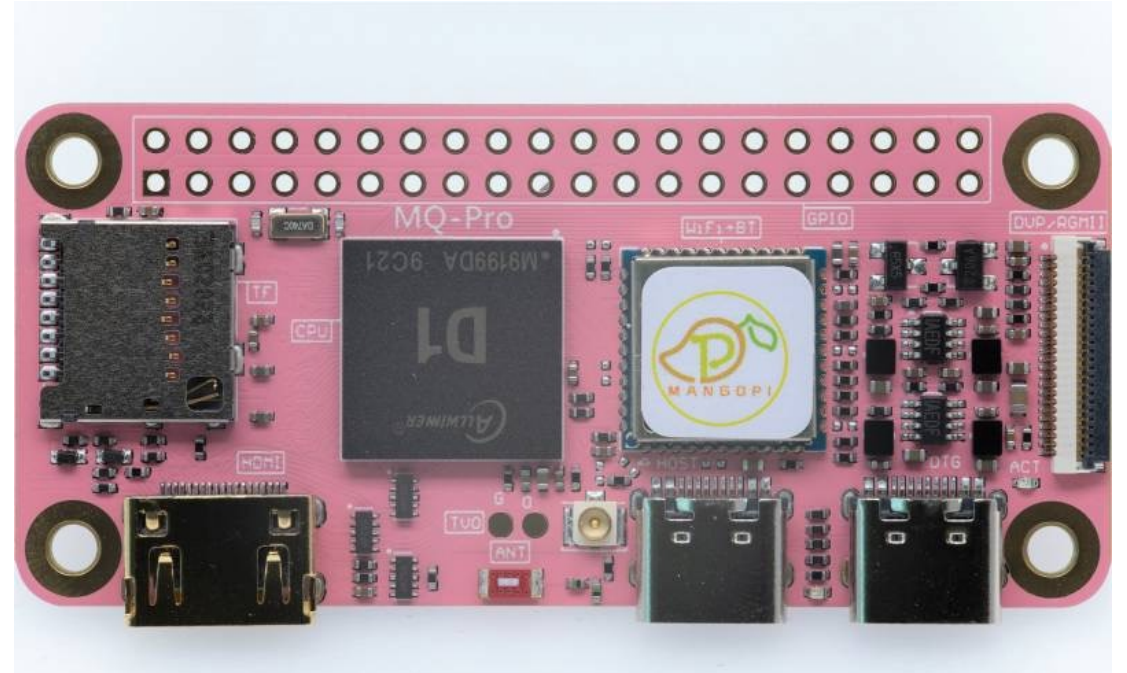


- **Default:** -mabi=lp64d
- **Other:** lp64, lp64v
- **Dynamic linker/loader:**
 - ld-linux-riscv64xthead-lp64d.so
 - ld-linux-riscv64-lp64v.so
 - ld-linux-riscv64-lp64d.so

```
./app: cannot execute: required file not found
```

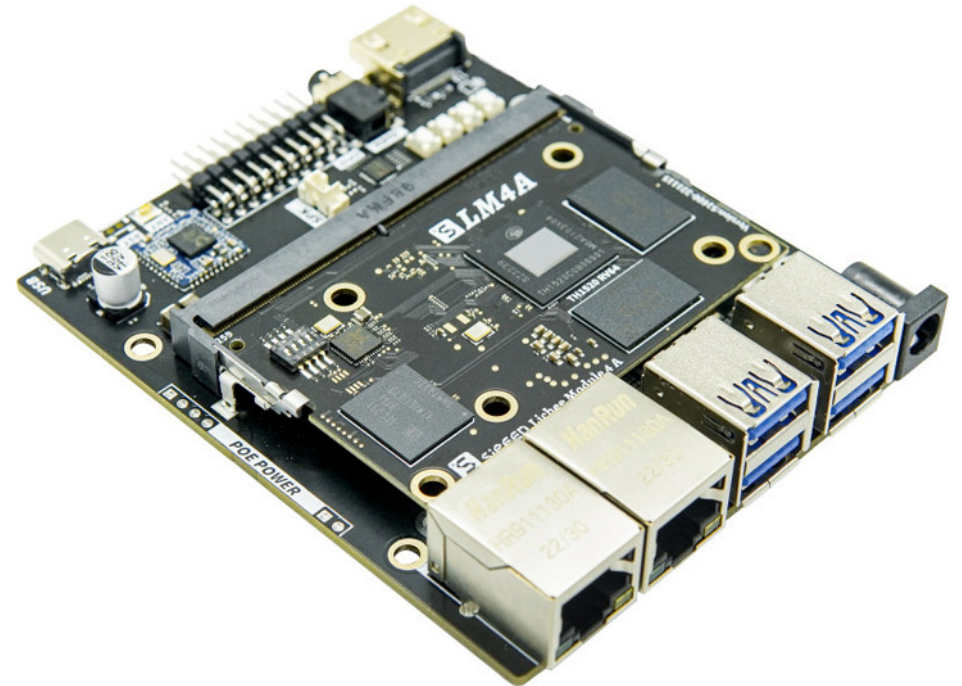
Mango Pi MQ Pro (HW)

- D1, C906 Core, RISC-V core up to 1GHz
- 512MB or 1GB DDR3/DDR3L
- USB-OTG Type-C
- USB-HOST Type-C
- 40Pin RPI-expand
- 24Pin DVP/RGMII connector
- mini HDMI connector
- TF card
- RTL8723ds Wi-Fi/BT module
- 20Pins DSI/CTP/LVDS FPC connector
- Audio OUT pads
- onboard Wi-Fi/BT ant
- 6.5x3cm size

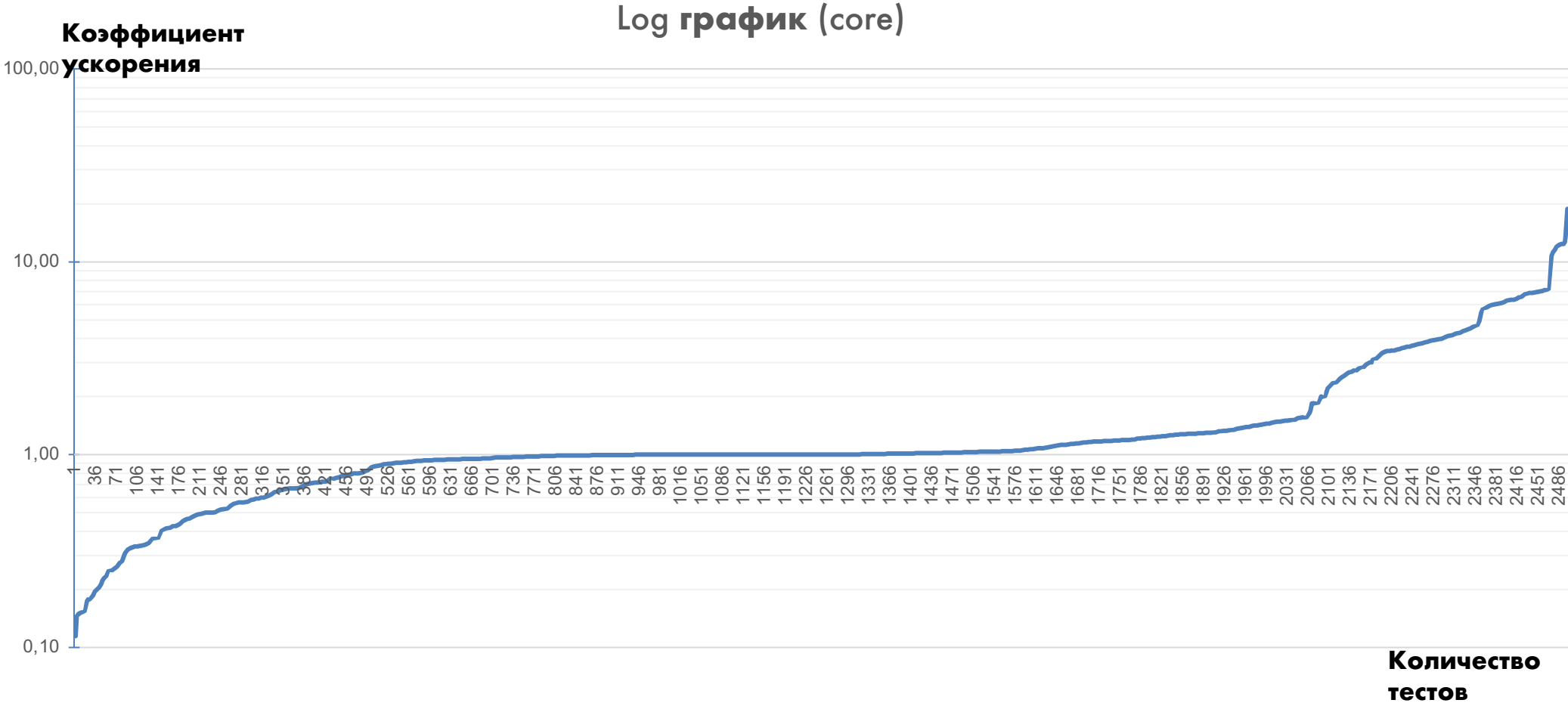


Lichee PI 4A (HW)

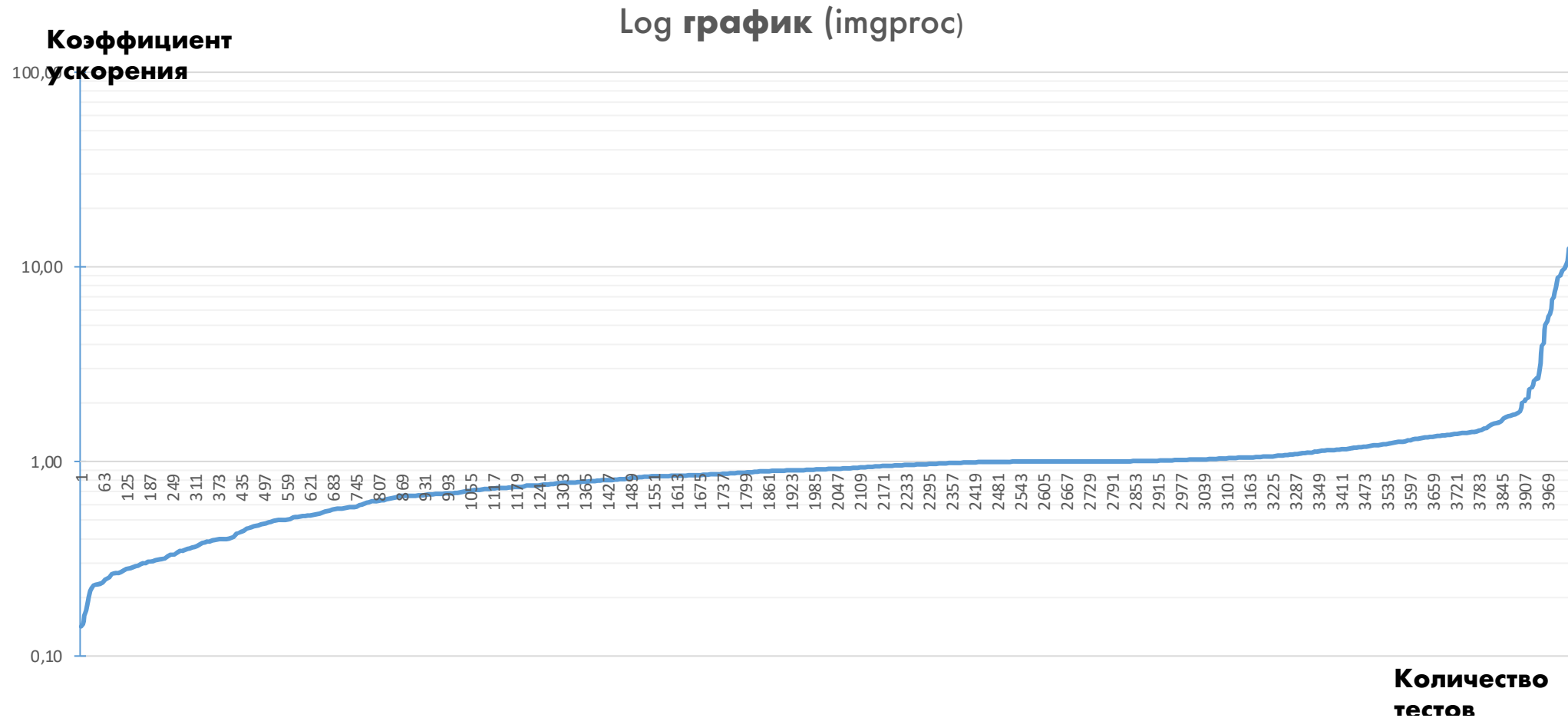
- Shadow 1520
- RISC-V 64GCV C910*4@2GHz
- 64KB I, 64KB D Cache; 1MB L2 Cache
- 8GB or 16GB 64bits LPDDR4
- NPU processor
- TF card
- USB3.0 x 4 and USB2.0 x 1
- 2 x Gigabit Ethernet ports



Результаты тестов производительности Lichee Pi 4A (scalar vs vector, RVV 0.7.1)



Результаты тестов производительности Lichee Pi 4A (scalar vs vector, RVV 0.7.1)



Размытие изображений

Gaussian blur, 3x3, 8UC1

Size	Scalar (ms)	Vector (ms)	SpeedUP
640x480	0,69	0,52	1,33
1280x720	1,99	1,52	1,31

Gaussian blur, 3x3, 32FC1

Size	Scalar (ms)	Vector (ms)	SpeedUP
640x480	1,94	1,96	0,99
1280x720	6,16	5,80	1,06

Median blur, 8UC1

Size	Scalar (ms)	Vector (ms)	SpeedUP
640x480	27,63	7,54	3,66
1280x720	93,66	18,75	5,00

Median blur, 32FC1

Size	Scalar (ms)	Vector (ms)	SpeedUP
640x480	23,94	4,27	5,61
1280x720	71,83	12,69	5,66

Заключение

- **Уже сегодня библиотека полностью функциональна на RISC-V и имеет достаточную производительность, полученную без глубокой переработки оптимизаций**
- **Реализованный в OpenCV подход позволяет библиотеке быть переносимой, что играет важную роль в адаптации под RISC-V**
- **Библиотека продолжает развиваться, улучшая оптимизации под RISC-V и расширяя поддержку новых версий инструкций RVV**



Ссылки

- OpenCV | <https://github.com/opencv/opencv>
- OpenCV docs | <https://docs.opencv.org/>
- T-Head toolchain repository | <https://github.com/T-head-Semi/xuantie-gnu-toolchain>
- T-Head DL framework for RISC-V | <https://github.com/T-head-Semi/csi-nn2>
- Lichee Pi Official Page | <https://wiki.sipeed.com/hardware/zh/lichee/th1520/lp4a.html>
- Mango Pi Official page | <https://mangopi.org/mqpro>
- Mango Pi awesome collection | <https://github.com/boosterl/awesome-mango-pi-mq-pro>
- Armbian | <https://www.armbian.com/mangopi-mq-pro/>
- Mango Pi MQ Pro | <https://drive.yadro.com/s/6TPfmkgS7nKgGkL>
- Old RVV 0.7 toolchain | <https://drive.yadro.com/s/FSRs2f4zXtbc4qd>
- RVV Intrinsic doc | <https://github.com/riscv-non-isa/rvv-intrinsic-doc>
- RISC-V arch&abi | <https://www.sifive.com/blog/all-aboard-part-1-compiler-args>

