# Система обнаружения лиц с использованием MB-LBP на микроконтроллере с ограниченными ресурсами

*М.А. Мельников, М.Л. Рысин*

*МИРЭА – Российский технологический университет, Москва*

**Аннотация:** Рассматривается алгоритм Виолы-Джонса с признаками MB-LBP для обнаружения лиц и его реализация на базе микроконтроллера STM32. Предлагаются методы оптимизации данного алгоритма для встраиваемых систем с ограниченными ресурсами. Описано создание аппаратно-программной системы обнаружения лиц.
**Ключевые слова:** обнаружение лиц, микроконтроллер, встраиваемые системы, алгоритм Виолы-Джонса, признаки MB-LBP, оптимизация классификаторов, оптимизация интегрального изображения, SIMD инструкции.

## Введение

Всё большее распространение получают биометрические методы идентификации личности, используемые в самых разных сферах жизни. Из них наиболее распространенным является идентификация по лицу, активно применяемая в самых разных информационных системах (например, широким спросом пользуется технология Face ID в смартфонах iPhone). Однако алгоритмы идентификации не могут работать обособленно от алгоритмов обнаружения, быстро локализующих области изображений, в которых располагается исследуемый объект [1]. Именно задача обнаружения лиц будет рассмотрена в данной статье.

Использование микроконтроллеров в качестве аппаратной основы системы распознавания лиц позволит снизить себестоимость её производства за счет сокращения элементной базы (значительная часть периферийных устройств уже размещена внутри микроконтроллера) и сократить энергопотребление во время работы.

Одним из эффективных архитектурных решений, подходящих для данной системы, является использование многоядерных микроконтроллеров — одно ядро отвечает за процесс обнаружения и локализации лица, второе проводит процесс идентификации в выделенном участке изображения [2].

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

## Ожидаемый результат

Использование микроконтроллеров, в свою очередь, влечет за собой значительные сложности реализации обнаружения лиц в силу серьезной ограниченности аппаратных ресурсов: тактовой частоты процессора и объема оперативной памяти. Положение усугубляется, если задача обнаружения лиц должна быть решена в реальном времени – при обработке непрерывного потока кадров, поступающих от видеокамеры.

В данной работе будет рассмотрено решение задачи обнаружения лиц в реальном времени на маломощном микроконтроллере.

Создаваемая программно-аппаратная система должна обрабатывать непрерывный цветной видеопоток с частотой обработки не менее, чем 1 кадр в секунду. Вывод должен быть представлен в цветовом формате с отображением обнаруженных лиц в виде прямоугольных рамок. Не предполагается использование внешних модулей оперативной и постоянной памяти.

## Аппаратная часть системы

В качестве основы системы нами был выбран микроконтроллер STM32F407VET6 со следующими характеристиками: 32-х битное ядро Cortex-M4 с максимальной тактовой частотой 168 МГц, 196 Кбайт оперативной памяти (включая CCM), 1 Мбайт флеш-памяти [3]. Такой выбор обоснован популярностью микроконтроллеров STM32 (как следствие, развитыми средствами разработки и подробной документацией), а также наличием минимально необходимой для разрабатываемой системы периферии внутри.

Микроконтроллер расположен на плате разработки JL32-F4VE, к которой подключены модуль видеокамеры на базе OV7670 и TFT дисплей под управлением ILI9341 (состав компонентов системы и схема подключения представлены на рис. 1 и 2).

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

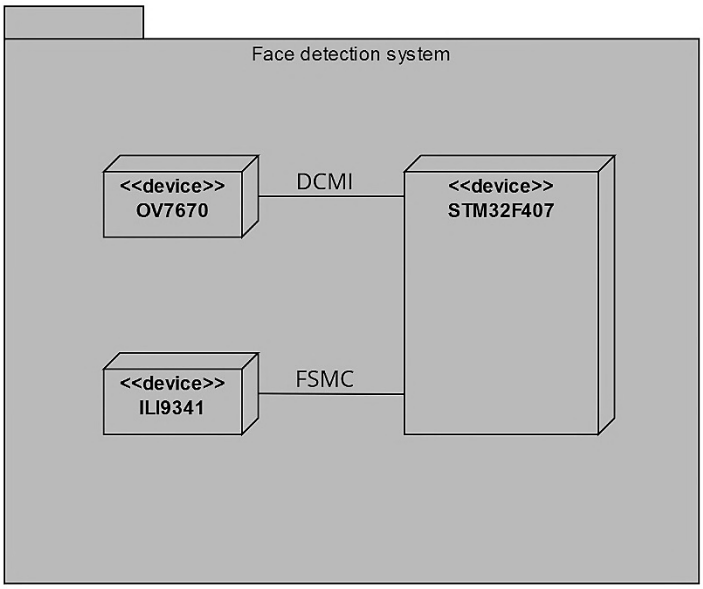


Рис. 1. – Диаграмма развёртывания системы обнаружения лиц

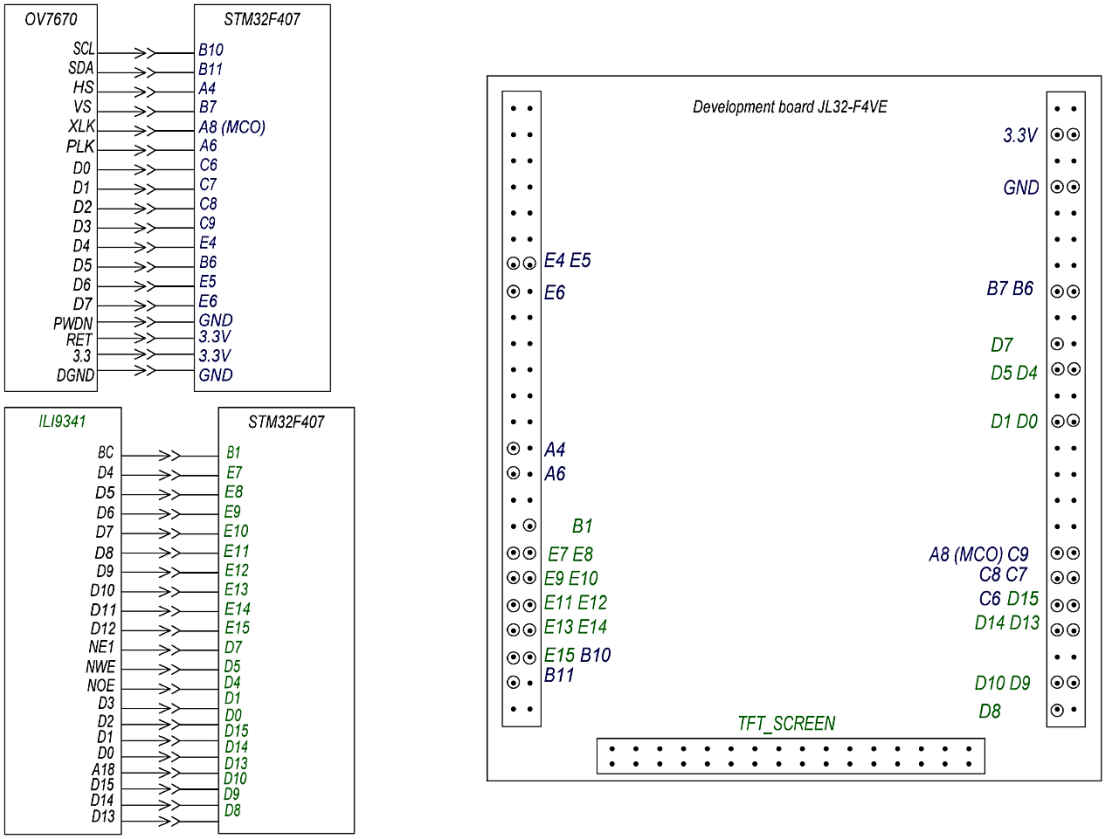


Рис. 2. – Схема подключения внешних устройств к STM32

Взаимодействие с видеокамерой организовано через протокол I2C (используется для отправки управляющих команд) и DCMI (синхронная

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

параллельная шина данных, позволяющая оперировать с данными, получаемыми от цифровой видеокамеры, на высоких скоростях вплоть до 54 Мбайт/с). Работа с TFT дисплеем организована посредством FSMC (контроллер взаимодействия со внешними модулями памяти или дисплеями по параллельной шине данных) [4].

Алгоритм обнаружения лиц и драйверы внешней периферии (видеокамеры и дисплея) были реализованы на языке С с изолированием от аппаратной части, посредством использования управляющих структур с указателями функций (таким образом, организовано внедрение зависимостей) [5]. Это позволяет быстро портировать данную кодовую базу на любую платформу, для которой создан компилятор языка С, и создать модульные тесты. Для работы с внутренней периферией STM32 используется библиотека HAL (Hardware Abstraction Library). Видеокамере были заданы следующие параметры: формат поступающих данных RGB565 QCIF (разрешение: 172 x 144); частота отправки данных 16 МГц (тактовый сигнал поступает от микроконтроллера по пину MCO); автоматический баланс белого.

## Реализация алгоритма

Из самых популярных в настоящее время методов обнаружения лиц можно выделить: метод Виолы-Джонса, гистограммы ориентированных градиентов (HOG) и свёрточные нейронные сети [6]. В силу ограниченности ресурсов для реализации обнаружения лиц был использован алгоритм Виолы-Джонса. Он позволяет быстро обнаружить несколько объектов с жесткой геометрией в кадре и обладает малым количеством ложных срабатываний (с точки зрения разработчиков алгоритма).

Идея алгоритма Виолы-Джонса состоит в том, чтобы объединить «слабые» классификаторы (вычисляемые с помощью признаков) в «сильные» классификаторы (далее, будем называть их стадиями), которые, в свою

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

очередь, образуют каскад, способный обнаружить лицо [7]. При анализе области изображения совершается последовательное вычисление стадий, начиная с самой простой (наименее затратной по вычислительным ресурсам). В том случае, если хотя бы одна стадия выдаст отрицательный результат (сумма «слабых» классификаторов меньше порога стадии), то анализируемая область изображения отбрасывается (рис. 3). Таким образом, мы можем быстро исключать участки изображения, на которых не находится лицо. При этом признаки считаются с помощью интегрального изображения за константное время.

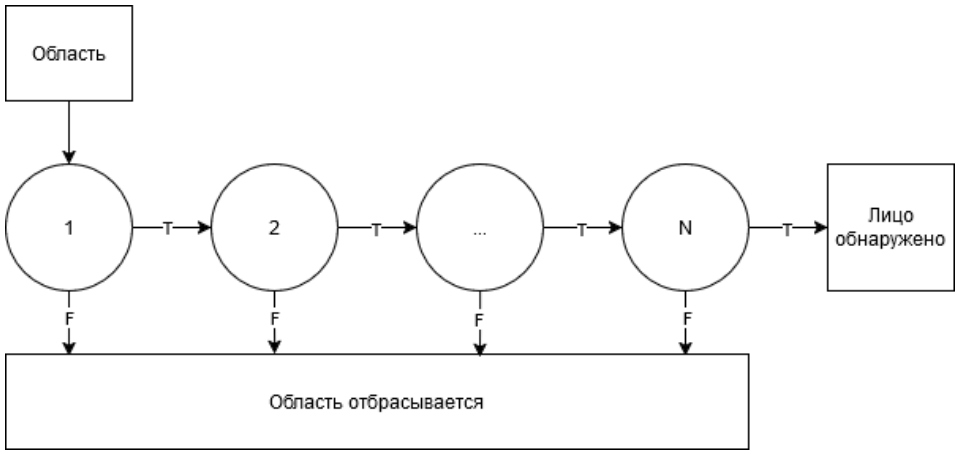


Рис. 3. – Схема работы каскада из N «сильных» классификаторов (стадий)

В классическом алгоритме Виолы-Джонса используются признаки Хоара, однако в данном случае выбор пал на более эффективные признаки MB-LBP [8]. Это улучшенный вариант LBP (local binary pattern), который может быть применен для описания окрестности пикселя только размером 3 на 3 пикселя. Итоговое значение LBP представляет собой 8-битное число – центральный пиксель сравнивается с окрестными. Если яркость окрестного пикселя больше или равна, чем у центрального, то соответствующий бит равен единице, иначе ноль. MB-LBP, напротив, может быть применен для бóльших по площади окрестностей, чем в LBP, оперируя не пикселями, а их множествами (рис. 4). В рамках текущей задачи, это позволяет масштабировать признак и находить лица различного размера. Также стоит

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

отметить, что MB-LBP позволяет оперировать только с целочисленными значениями, что значительно ускоряет процесс анализа, совершаемого на маломощном микроконтроллере.


Рис. 4. – «Слабый» классификатор, вычисляемый с помощью признака MB-LBP и стадия комбинации «слабых» классификаторов

Общий алгоритм обнаружения лиц работает следующим образом (рис. 5):

1. Считывается кадр.

2. Вычисляется интегральное изображение исходного кадра.

3. Берется квадратная область изображения.

4. Вычисляется первая стадия – суммируются все значения входящих в неё «слабых» классификаторов (признаков MB-LBP) и сравниваются с её пороговым значением. В том случае, если сумма меньше порога, то область отбрасывается, как не содержащая в себе лица (производится переход к следующей области).

5. Последовательно вычисляются оставшиеся стадии.

6. Если все стадии выдали положительный результат, то считается, что в анализируемой области присутствует лицо.

7. Производится переход к следующей области изображения.

8. После анализа всех областей найденные позиции ближайших лиц объединяются.

9. Вывод итоговых позиций на экран в виде рамок.

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

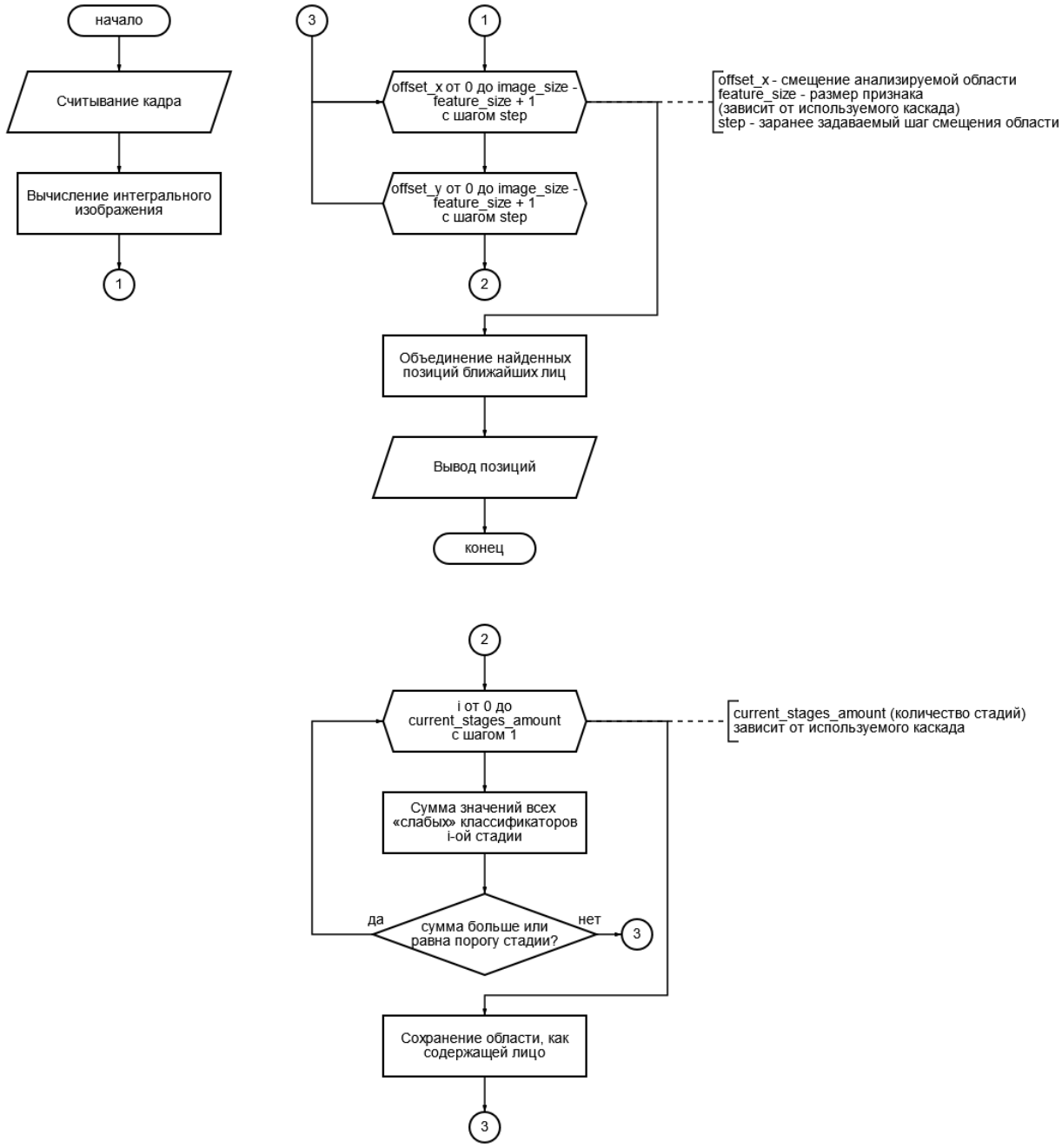


Рис. 5. – Алгоритм обнаружения лиц на одном кадре

В качестве обученных каскадов классификаторов (для обнаружения лиц во фронт) используются файлы из библиотеки OpenCV: lbpcascade_frontalface.xml (20 стадий; минимальное окно – 25 х 25 пикселей) и lbpcascade_frontalface_improved.xml (улучшенный вариант, эффективно отсеивающий области без лица на ранних стадиях; 19 стадий; минимальная область – 45 х 45 пикселей) [9].

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

Для экономии оперативной памяти исходные файлы классификаторов OpenCV были конвертированы в бинарные файлы. Для описания процесса конвертации рассмотрим структуру классификаторов OpenCV.

В исходных файлах «слабые» классификаторы состоят из двух составляющих (листинг 1):

- internalNodes – массив следующих значений: номер узла; индекс узла, к которому происходит переход (в нашем случае переход не происходит); индекс прямоугольника; маска (8 значений), полученная в результате обучения классификатора;
- leafValues – два возвращаемых значения.

Прямоугольники перечислены в конце файла. Они обозначены в виде координат верхнего левого угла, ширины и высоты. Для вычисления значения «слабого» классификатора применяется признак MB-LBP в области прямоугольника, индекс которого находится в internalNodes. Прямоугольник в данном случае обозначает верхнюю левую область признака. Далее, с помощью следующей строки вычисляется итоговое значение «слабого» классификатора: masks[score >> 5] & (1 << (score & 31)) ? leafValues[0] : leafValues[1], где score – значение признака MB-LBP.

Листинг 1. Исходная структура «слабого» классификатора

```
<weakClassifiers>
  <_>
    <internalNodes>
      0 -1 26 -1 -1 -17409 -1 -1 -1 -1 -1</internalNodes>
    <leafValues>
      -9.9726462364196777e-001 -3.8938775658607483e-001</leafValues></_>
...
```

Для конвертации в бинарные файлы все значения исходных классификаторов были помещены в массив структуры lbp_feature языка C с использованием Little-Endian кодировки и 32-битной адресации (листинг 2). Далее, данный массив был записан в итоговый бинарный файл так, что

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

сначала в памяти идут стадии (ссылки на «слабые» классификаторы заменены относительным адресом – смещением относительно начала бинарного файла), а затем «слабые» классификаторы. Таким образом, мы получаем бинарные данные, которые не требуют предварительной обработки для использования – они уже представлены в виде массива языка С.

Листинг 2. Исходная структура «слабого» классификатора

```
typedef struct
{
  lbp_feature *features;
  uint8_t features_amount;
  int16_t threshold;
} stage;
typedef struct
{
  lbp_feature_rectangle rectangles[LBP_FEATURE_RECTANGLES_AMOUNT];
  lbp_feature_rectangle **scaled_rectangles;
  uint8_t scaled_rectangles_amount;
  int32_t masks[LBP_FEATURE_MASKS_AMOUNT];
  int16_t left_value;
  int16_t right_value;
} lbp_feature;
```

При этом предварительно все дробные значения в исходных файлах были конвертированы в целочисленные значения следующим образом: int(float(n) * 5000). Так мы ускоряем работу алгоритма на маломощном ядре микроконтроллера и сокращаем размер итогового бинарного файла. По итогу размер бинарного файла lbpcascade_frontalface.bin составляет 11 КБ (у исходного – 50,6 КБ), а у lbpcascade_frontalface_improved.bin 11,8 КБ (у исходного – 52,8 КБ). Далее, данный файл был сконвертирован в объектный с помощью утилиты arm-none-objcopy, загружен во флеш-память STM32 и помещен в CCM (Core Coupled Memory). Это раздел памяти, к которому ядро имеет эксклюзивный доступ, и, таким образом, мы ускоряем работу с

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

классификаторами, активно использующимися во время работы алгоритма обнаружения лиц.

При реализации алгоритма возникла серьезная проблема – в оперативную память не помещалось исходное изображение (необходимое для вывода) и интегральное, требующее большой объем памяти. В классическом виде количество бит, приходящихся на один пиксель в интегральном изображении можно рассчитать так [10]:

$$(2^{L_{ii}} - 1) \geq (2^{L_i} - 1)WH \quad (1)$$

где $L_{ii}$ – количество бит в пикселе интегрального изображения; $L_i$ - количество бит в пикселе исходного изображения; $W$ – ширина исходного изображения; $H$ – высота исходного изображения.

Используя формулу выше, получаем, что количество бит, необходимое для хранения интегрального изображения в наших условиях (разрешение 172 х 144; 16 бит на пиксель) равно 32-м. Тогда итоговый размер интегрального изображения можно посчитать по следующей формуле:

$$W \cdot H \cdot L_i i \quad (2)$$

$$(172 \cdot 144) \cdot 32/8 = 96{,}75 \text{ Кбайт} \quad (3)$$

Учитывая размер исходного изображения – 48,4 Кбайта и то, что нам доступно всего 132 Кбайта обычной оперативной памяти (мы не можем единообразно использовать обычное пространство SRAM и CCM, размер которой равен 64-м Кбайтам), можно сделать вывод, что имеющихся ресурсов микроконтроллера недостаточно. Было принято решение сократить размер итогового интегрального изображения посредством вычислений с переполнением. Для этого необходимо выполнение следующего равенства:

$$L_{ii} = \left\lceil log_2((2^{L_i} - 1){S_{max}}^2 + 1) \right\rceil \quad (4)$$

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

где $S_{max}$ – максимальная длина стороны квадрата исследуемой области, $L_i$ – количество бит на пиксель в исходном изображении.

Чтобы подобрать оптимальные величины, удовлетворяющие условию (4), сформулируем многокритериальную задачу оптимизации:

$$\begin{cases} min\{f_1(\vec{x}), f_2(\vec{x})\} \\ \vec{x} = (x_1, x_2) \\ f_1 = log_2((2^{x_1} - 1){x_2}^2 + 1) \\ f_2 = -x_2 + 0 \cdot x_1 \\ f_1(\vec{x}) \le 16 \\ x_1 \in [3,8] \\ x_2 \in [45,144] \\ sin(\pi \cdot x_i) = 0, i = 1,2 \end{cases} \quad (5)$$

Нормализуем критерии (6) и применим взвешенную сумму для получения целевой функции оптимизации (7). Веса равны 0.5, так как $f_1$ и $f_2$ равнозначны.

$$f_i^* = \frac{f_i - f_{i(min)}}{f_{i(max)} - f_{i(min)}} \quad (6)$$

где $f_{i(min)}$ и $f_{i(min)}$ – минимальное и максимальное значения критерия в заданном диапазоне.

$$F = 0{,}5 \cdot \left(\frac{{f_1}^*}{8{,}54304} - 1{,}61432\right) + 0{,}5 \cdot \left(\frac{{f_2}^*}{99} - 1{,}45455\right) \to min$$

$$\begin{cases} min\{{f_1}^*(\vec{x}), {f_2}^*(\vec{x})\} \\ \vec{x} = (x_1, x_2) \\ {f_1}^* = log_2\left((2^{x_1} - 1){x_2}^2 + 1\right) \\ {f_2}^* = -x_2 + 0 \cdot x_1 \\ {f_1}^*(\vec{x}) \le 16 \\ x_1 \in [3,8] \\ x_2 \in [45,144] \\ sin(\pi \cdot x_i) = 0, i = 1,2 \end{cases} \quad (7)$$

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

В результате решения методом градиентного спуска при параметрах $\varepsilon = 1e-8, h = 0.001, \overrightarrow{x_0} = (3,45)$ получаем результат: $\vec{x} = (3,96.75617)$ [11]. Тогда оптимальными величинами являются: максимальная площадь анализируемой области — 96 х 96, количество бит в пикселе исходного изображения — 3 (используются старшие биты значения яркости исходного пикселя). Итоговый размер оптимизированного интегрального изображения:

$$(172 \cdot 144) \cdot 16/8 = 48{,}38 \text{ Кбайт} \quad (8)$$

За счет сокращения размера интегрального изображения мы можем предпринять ещё одну меру для оптимизации работы алгоритма обнаружения лиц на STM32 — использование векторных вычислений (SIMD операций). SIMD операция позволяет выполнить одну машинную инструкцию для нескольких наборов данных. В частности, нас интересует инструкция UQADD16, которая позволяет за один такт сложить два 16-и битных числа [12]. Для её использования применим не классический метод вычисления интегрального изображения, а использующий кумулятивную сумму столбца и имеющий потенциал к параллелизму [13]:

$$S(x,y) = i(x,y) + S(x,y-1) \quad (9)$$

$$S(x+1,y) = i(x+1,y) + S(x+1,y-1) \quad (10)$$

$$ii(x,y) = ii(x-1,y) + S(x,y) \quad (11)$$

$$ii(x+1,y) = ii(x,y) + S(x+1,y) \quad (12)$$

где $ii(x,y)$ – яркость пикселя в интегральном изображении; $i(x,y)$ - яркость пикселя в исходном изображении; $S$ — кумулятивная сумма столбца. Этапы (9) и (10) могут быть вычислены параллельно с помощью инструкции UQADD16 (листинг 3).

Листинг 3. Исходная структура «слабого» классификатора

```
union {
```

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

```
  uint16_t parts[2];
  uint32_t full;
} row_sum = { 0 };

for (uint16_t y = 1; y < image_size.height; y++)
{
  row_sum.full = UQADD16(
    row_sum.full,
    *(uint32_t*)(integral_image + x + y * image_size.width)
  );

  GET_PIXEL(integral_image, x, y) = GET_PIXEL(integral_image, x - 1, y) +
    row_sum.parts[0];
  GET_PIXEL(integral_image, x + 1, y) = GET_PIXEL(integral_image, x, y) +
    row_sum.parts[1];
}
```

## Заключение

Таким образом, нам удалось достичь средней частоты работы системы не менее 3-х кадров в секунду (табл. 1). При этом микроконтроллер выполняет не только алгоритм обнаружения лиц, но и вывод итогового цветного кадра (с обозначением обнаруженных лиц) на дисплей в реальном времени (рис. 6).

Таблица № 1. Результаты измерения временных затрат

| | Вычисление интегрального изображения | Анализ всех областей кадра | Общее время обработки кадра |
|---|---|---|---|
| Среднее время, мс | 8,69 | 309,76 | 319,95 |

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

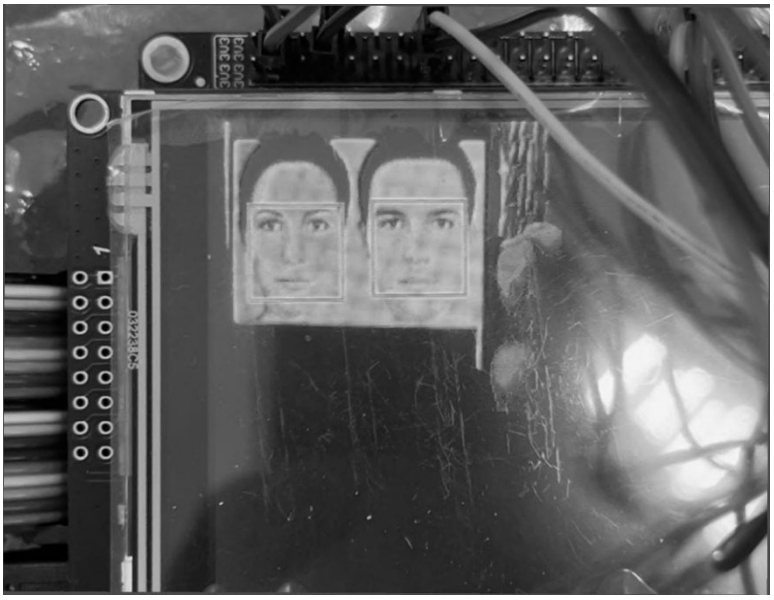

Рис. 6. – Демонстрация работы системы

Следующим шагом в развитии предложенной системы может стать замена, в порядке импортозамещения, микроконтроллера STM32F407VET6 на отечественный К1921ВК01Т, базирующийся на ядре ARM Cortex-M4F со встроенной ОЗУ объемом в 192 Кбайт и таковой частотой 100 МГц [14].

## Литература

1. Li L., Mu X., Li S., Peng H. A Review of Face Recognition Technology // IEEE Access. 2020. Vol. 8. P. 139110.

2. Beningo J. Embedded Software Design: A Practical Approach to Architecture, Processes, and Coding Techniques. – Linden: Apress, 2022. – 463 p.

3. RM00900. Reference Manual. STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs. – URL: st.com/resource/en/reference_manual/rm0090-stm32f405415-stm32f407417-stm32f427437-and-stm32f429439-advanced-armbased-32bit-mcus-stmicroelectronics.pdf (дата обращения: 20.03.2025).

4. TFT LCD interfacing with the high-density STM32F10xxx (FSMC) (AN2790). – URL: st.com/content/ccc/resource/technical/document/application_note/85/ad/ef/0f/a3/a6

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

/49/9a/CD00201397.pdf/files/CD00201397.pdf/jcr:content/translations/en.CURD00201397.pdf (дата обращения: 20.03.2025).

5. STM32FaceDetection. Исходный код системы обнаружения лиц. – URL: github.com/MatveyMelnikov/STM32FaceDetection (дата обращения: 20.03.2025).

6. Хахина А.М, Тельнова Т.В. Методы обнаружения лиц // Научные известия. 2022. № 27. С. 27 – 28.

7. Viola P., Jones M. Rapid Object Detection using a Boosted Cascade of Simple Features // Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001 (08 – 14 dec. 2001). pp. 1 – 5.

8. Liao S., Zhu X., Lei Z., Zhang L., Li S.Z. Learning Multi-scale Block Local Binary Patterns for Face Recognition // Proceedings of the 2007 international conference on Advances in Biometrics (27 – 29 aug. 2007). Seoul, pp. 2 – 4.

9. OpenCV: Open Source Computer Vision Library. – URL: github.com/opencv/opencv (дата обращения: 20.01.24).

10. Belt H. Storage Size Reduction for the Integral Image. Technical Report TN-2007/00784 // Philips Research Eindhoven. 2007. P. 8.

11. Нестеров Ю.Е. Методы выпуклой оптимизации. – М.: МЦНМО, 2010. – 281 с.

12. STM32 Cortex-M4 MCUs and MPUs programming manual (PM0214).. – URL: st.com/resource/en/programming_manual/pm0214-stm32-cortexm4-mcus-and-mpus-programming-manual-stmicroelectronics.pdf (дата обращения: 20.03.2025).

13. Ehsan S., Clark A., Rehman N., McDonald-Maier K. Integral Images: Efficient Algorithms for Their Computation and Storage in Resource-Constrained Embedded Vision Systems // Sensors (MDPI). 2015. P. 16809.

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

14. Микроконтроллер 1921ВК01Т1. – URL: niiet.ru/product/1921вк01т1 (дата обращения: 20.03.2025).

**References**

1. Li L., Mu X., Li S., Peng H., IEEE Access, 2020, vol. 8, 139110 p.

2. Beningo J. Embedded Software Design: A Practical Approach to Architecture, Processes, and Coding Techniques. Linden, Apress, 2022. 463 p.

3. RM00900. Reference Manual. STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs. URL: st.com/resource/en/reference_manual/rm0090-stm32f405415-stm32f407417-stm32f427437-and-stm32f429439-advanced-armbased-32bit-mcus-stmicroelectronics.pdf (accessed 03/20/2025).

4. TFT LCD interfacing with the high-density STM32F10xxx (FSMC) (AN2790). URL: st.com/content/ccc/resource/technical/document/application_note/85/ad/ef/0f/a3/a6/49/9a/CD00201397.pdf/files/CD00201397.pdf/jcr:content/translations/en.CD00201397.pdf (accessed 03/20/2025).

5. STM32FaceDetection. Source code of face detection system. URL: github.com/MatveyMelnikov/STM32FaceDetection (accessed 03/20/2025).

6. Khakhina A.M., Telnova T.V. Nauchnye izvestiya. 2022. №27. pp. 27-28.

7. Viola P., Jones M. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001 (08 – 14 dec. 2001). pp. 1 – 5.

8. Liao S., Zhu X., Lei Z., Zhang L., Li S.Z. Proceedings of the 2007 international conference on Advances in Biometrics (27 – 29 aug. 2007). Seoul. 2007. pp. 2 – 4.

9. OpenCV: Open Source Computer Vision Library. URL: github.com/opencv/opencv (accessed 03/20/2025).

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025

10. Belt H. Technical Report TN-2007/00784. Philips Research Eindhoven, 2007, 8 p.

11. Nesterov Yu.E. Metody vypukloj optimizacii [Convex optimization methods]. Moskva. MCzNMO, 2010. 281 p.

12. STM32 Cortex-M4 MCUs and MPUs programming manual (PM0214). URL: st.com/resource/en/programming_manual/pm0214-stm32-cortexm4-mcus-and-mpus-programming-manual-stmicroelectronics.pdf (accessed 03/20/2025).

13. Ehsan S., Clark A., Rehman N., McDonald-Maier K. Sensors (MDPI), 2015, vol. 15, 16809 p.

14. Microcontroller 1921VK01T1. URL: niiet.ru/product/1921%d0%b2%d0%ba01%d1%821 (accessed 03/20/2025).

**Дата поступления: 24.02.2025**

**Дата публикации: 25.04.2025**

© Электронный научный журнал «Инженерный вестник Дона», 2007–2025