

ПРОГРАММНО-АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ РАБОТЫ СИСТЕМЫ МАШИННОГО ЗРЕНИЯ

Герасименко Г.В.

Белорусский государственный университет информатики и радиоэлектроники
Институт информационных технологий,
г. Минск, Республика Беларусь

Скудняков Ю.А. -к.т.н., доцент

Для компьютерного распознавания объектов естественного происхождения разработано программно-алгоритмическое обеспечение (ПАО) функционирования системы машинного зрения (СМЗ), использование которого позволяет обеспечить высокие оперативность и точность распознавания субъектов.

Весьма актуальной задачей современного общества является распознавание субъектов различных категорий. Для решения данной задачи в настоящее время уже получены определенные результаты. Однако они требуют своего дальнейшего совершенствования по ряду важных показателей качества: высоких значений точности, оперативности, надежности, которые зависят от множества различных факторов и видов обеспечения СМЗ.

Данная работа посвящена разработке и использованию ПАО для эффективной работы СМЗ [1].

В качестве платформы функционирования ПАО СМЗ в работе выбран простой одноплатный компьютер, построенный на базе ARM процессора и операционной системы Raspbian, реализованной на основе Debian GNU/Linux.

Для решения задачи записи потокового видео с камеры была выбрана открытая библиотека OpenCV, которая поддерживается в системах Linux и может работать с библиотеками, позволяющими записывать, конвертировать и передавать цифровые аудио- и видеозаписи в различных форматах, таких как V4L2, DC1394 и FFmpeg. Также в OpenCV есть возможность приведения изображений к градации серого и черно-белому виду по порогу, а также одинаковому разрешению.

Для решения задачи машинного обучения использована открытая программная библиотека TensorFlow, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов [2-3]. Основной API для работы с библиотекой реализован для Python, также существуют реализации для R, C Sharp, C++, Haskell, Java, Go и Swift. TensorFlow 9 ноября 2015 года был открыт для свободного доступа. TensorFlow является системой машинного обучения Google Brain второго поколения. В то время как эталонная реализация работает на единичных устройствах, TensorFlow может работать на многих параллельных процессорах, как CPU, так и GPU. TensorFlow доступна для 64-разрядных Linux, macOS, Windows, и для мобильных вычислительных платформ, включая Android и iOS. Вычисления TensorFlow выражаются в виде потоков данных через граф состояний.

Для упрощения работы с библиотекой TensorFlow использована открытая нейросетевая библиотека Keras, написанная на языке Python. Она представляет собой надстройку над фреймворками DeepLearning4j, TensorFlow и Theano. Нацелена на оперативную работу с сетями глубинного обучения, при этом спроектирована так, чтобы быть компактной, модульной и расширяемой.

Keras предоставляет высокоуровневый, более интуитивный набор абстракций, который делает простым формирование нейронных сетей, независимо от используемой в качестве вычислительного бэкенда библиотеки научных вычислений.

Для получения возможностей захвата и обработки изображений необходимо подключить библиотеки OpenCV: `import cv2`.

Для загрузки изображения в буфер используем следующие функции:

```
image = cv2.imread(«./путь/к/изображению.расширение»); cv2.imshow("Image", image); cv2.waitKey(0); cv2.destroyAllWindows()
```

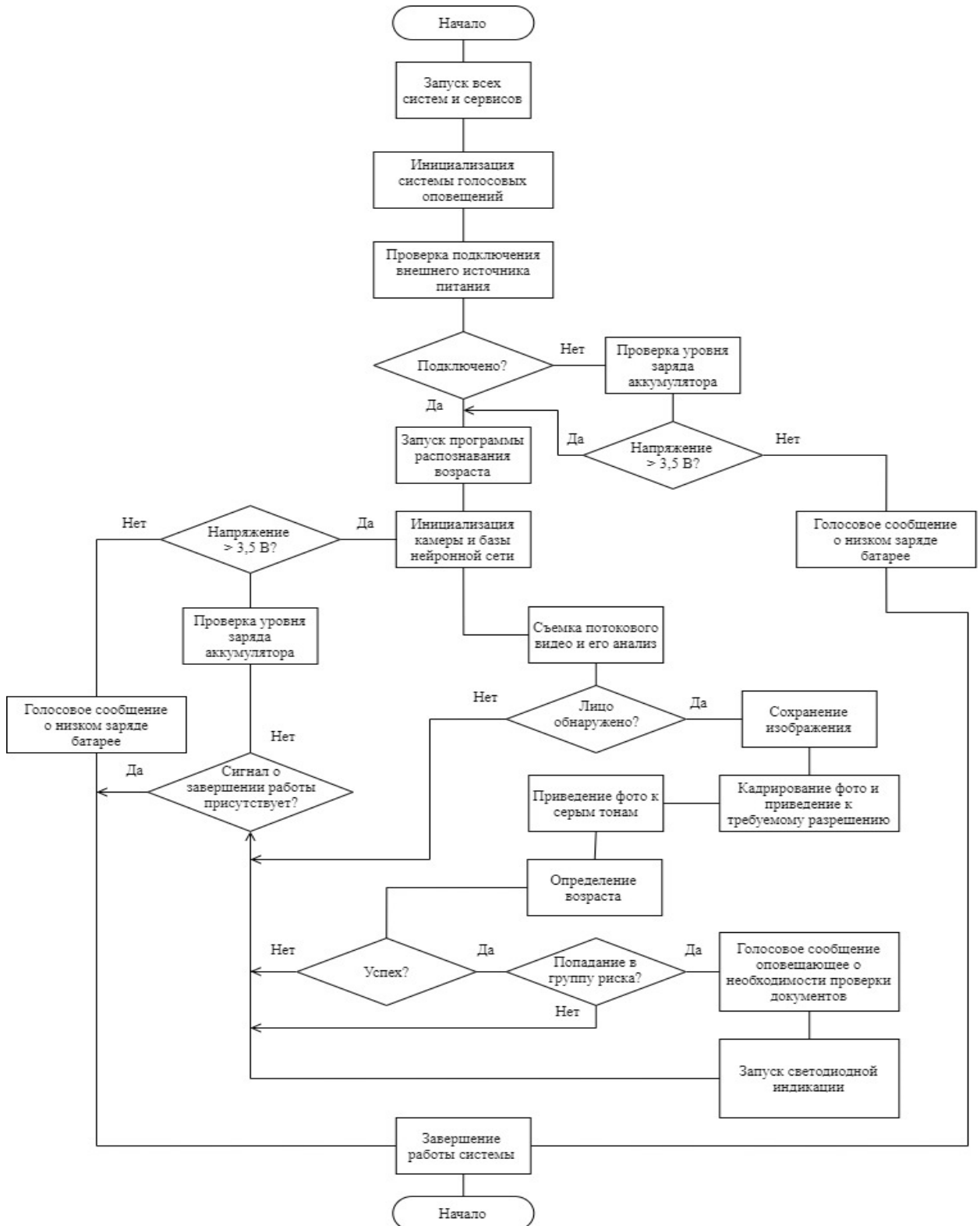
При чтении таким способом изображение находится в цветовом пространстве BGR (синий, зеленый, красный), для перехода в цветовое пространство RGB необходимо либо поменять местами 1-й канал (R — красный) с 3-м каналом (B — синий), либо поменять цветовое пространство на RGB при помощи функции `cv2.cvtColor`, которая в свою очередь имеет возможность преобразовывать изображения в градации серого. Для этого в функцию необходимо передать конвертируемое изображение и параметр `COLOR_BGR2GRAY`, который позволит перевести изображение из пространства BGR в пространство градаций серого:

```
gray image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

После распознавания изображение приводится к требуемому разрешению (кадрирование) и осуществляется следующей функцией:
`cropped = self.crop_face (frame, face, margin=50, size=self.face_size).`

Перед тем, как отдавать найденные лица в детектор пола/возраста, их необходимо нормализовать, т.е. привести лица разного масштаба, по-разному повернутые и наклонённые, к одному стандартному виду.

Логическая часть ПАО разработана в виде алгоритма работы СМЗ для оценки возраста человека, схема которого представлена на рисунке 1.



57-я научная конференция аспирантов, магистрантов и студентов БГУИР, 2021 г.

Рисунок 1 – Алгоритм работы СМЗ для оценки возраста человека

Реализация разработанного алгоритма осуществлена с помощью программного средства на языке *Python*.

Апробация ПАО СМЗ в реальных условиях показала хорошую эффективность с точки зрения получения требуемых значений оперативности, точности и надежности работы СМЗ.

Список использованных источников:

1. Компьютерное зрение: технологии, рынок, перспективы [Электронный ресурс]. – Режим доступа : <http://www.tadviser.ru/>. – Дата доступа: 26.06.2019.
2. Способ распознавания изображений лиц и система для его осуществления [Электронный ресурс] Патент / Freepatent. – Режим доступа: <patent-2541136.pdf>. – Дата доступа: 20.08.2020.
3. Универсальный подход к любой задаче машинного обучения [Электронный ресурс]. – Режим доступа: <http://datareview.info/>. – Дата доступа: 12.11.2020.