

О ПОДХОДЕ К ДЕТЕКТИРОВАНИЮ ДВИЖЕНИЯ ПЕШЕХОДОВ МЕТОДОМ ГИСТОГРАММ НАПРАВЛЕННЫХ ГРАДИЕНТОВ

М. В. Бобырь¹ [0000-0002-5400-6817], Н. А. Милостная² [0000-0002-3779-9165],
Н. И. Храпова³ [0000-0001-7947-1427]

¹⁻³Юго-Западный государственный университет, Курская обл., Курск, 305040,
Россия

¹maxbobyrg@gmail.com, ²nat_mil@mail.ru, ³KhrapovaNI@yandex.ru

Аннотация

Рассмотрен подход к автоматическому распознаванию движения людей на пешеходном переходе. Он включает две основные процедуры, для каждой из которых приведены команды программного кода на языке программирования C# с помощью библиотеки компьютерного зрения EMGU. На первой процедуре с помощью комбинации методов гистограмм направленных градиентов и опорных векторов осуществляется детектирование пешеходов. Вторая процедура позволяет считывать кадры из видеопоследовательности и обрабатывать их. Данный подход позволяет детектировать движения людей на пешеходном переходе без использования специализированных нейронных сетей. При этом метод, предложенный в статье, продемонстрировал достаточную надежность распознавания движения людей, что свидетельствует о его применимости в реальных условиях.

Ключевые слова: распознавание движения пешеходов, EMGU, гистограммы направленных градиентов, метод опорных векторов

ВВЕДЕНИЕ

Детектирование движения людей на пешеходном переходе является важной задачей для повышения безопасности дорожного движения и улучшения работы систем управления дорожным движением. В рамках дуальных технологий подобные системы могут использоваться в задачах определения перемещений людей внутри помещения [1]. Существует несколько подходов и алгоритмов для решения этой задачи, которые условно делятся на следующие группы.

К первой группе относятся модели, основанные на классических методах компьютерного зрения, в частности, модели фонового вычитания [1] и модели обнаружения краев и построения на их основе контуров объектов.

Ко второй группе следует отнести методы машинного обучения [2], использующие алгоритм Хаара или метод анализа признаков гистограмм направленных градиентов (HOG – Histogram of Oriented Gradients) в сочетании с машиной опорных векторов (SVM – Support Vector Machine).

Третья группа включает такие подходы к детектированию объектов, основанные на моделях глубокого обучения, как сверточные нейронные сети (CNN) [3] или рекуррентные нейронные сети (RNN).

Алгоритмы первой группы с помощью метода фонового вычитания выделяют движущиеся объекты за счет вычитания текущего кадра из предыдущего с помощью различных фильтров, таких как приближенный медианный метод, метод скользящего среднего, адаптивная гауссовская модель [4] или экспоненциально взвешенное скользящее среднее [5]. Данная задача упрощается, если перед вычитанием фона уменьшить количество информации путем применения фильтра Канни для детектирования границ объектов [6, 7].

Алгоритмы второй группы распознавания объектов (например, лиц и тел людей) используют каскадные классификаторы, например, Хаара [8, 9]. Также для решения задач детектирования объектов используется комбинация методов гистограмм направленных градиентов с методом опорных векторов.

HOG – это дескриптор особых точек, используемый в компьютерном зрении и обработке изображений для распознавания объектов. Основная идея HOG заключается в следующем. Вместо вычисления каждого отдельного направления градиента для каждого пикселя изображения пиксели группируются в маленькие ячейки, и для каждой группы ячеек вычисляется направление градиента, показывающее ориентацию ячейки на изображении. Далее величины градиента суммируются для каждой группы ячеек, например, окна размером 3×3 пикселя. Процесс повторяется для всех групп ячеек изображения. На основе направлений градиентов делается вывод о классификации объекта.

SVM – это алгоритм машинного обучения, используемый для классификации данных. Комбинация детекторов HOG+SVM позволяет разделять объекты на

классы («пешеход» и «не пешеход»). При этом SVM стремится найти гиперплоскость, которая наилучшим образом разделяет данные в многомерном пространстве признаков.

Алгоритмы третьей группы, такие как сверточные нейронные сети, позволяют извлекать признаки на изображениях, причем свертка – это процесс применения фильтра к изображению, например, окну 3×3, позволяющая выделять границы объектов на изображении. Одним из примеров применения CNN для решения задач данного класса служит использование глубокой нейросети AlexNet [10]. Рекуррентные нейронные сети используются для анализа последовательных данных, таких как временные ряды или текст. Они предназначены для обработки последовательных данных и учитывают временные зависимости между текущим и предыдущим кадрами видео [11, 12].

Поставим задачу разработки программного кода детектирования перемещения людей на пешеходном переходе. В ходе решения этой задачи особое внимание необходимо уделить организации двух процедур, а именно, считывания всех кадров из загруженного видеофайла и обнаружения пешеходов на каждом из загруженных кадров.

Таким образом, выбор алгоритма детектирования движения людей на пешеходном переходе зависит от конкретных условий использования подобных систем, например, таких как требуемая надежность распознавания объектов и доступные вычислительные ресурсы. Традиционные методы на основе компьютерного зрения эффективны в обычных условиях, однако методы на основе глубокого обучения, хотя более сложные и ресурсоемкие, но обеспечивают повышенную точность к изменениям внешней среды.

МЕТОДОЛОГИЯ

Рассмотрим процесс детектирования движения людей на пешеходном переходе с помощью открытой библиотеки обработки изображений EMGU. Программный код для реализации этой процедуры реализован на языке программирования C#. Для обработки видеoinформации использованы следующие переменные: `capture` – для хранения захватываемых кадров изображений с видеокамеры; `frames` – для хранения количества кадров в видео и `framesCurrent` – для

хранения текущего кадра; fps используется для указания количества видеок кадров, воспроизводимых в единицу времени, например, за секунду.

Для считывания видео использован следующий код:

```
01. capture = new VideoCapture(openFileDialog1.FileName);
```

```
02. Mat m = new Mat();
```

```
03. capture.Read(m);
```

```
04. pictureBox.Image = m.Bitmap;
```

На первой строке кода создаётся объект класса `VideoCapture`, который используется для захвата видео, например, директива `openFileDialog1.FileName` представляет путь к видеофайлу, который хранится на персональном компьютере. Вторая строка кода создает объект `Mat`, который используется для хранения одного кадра видео. Третья строка кода считывает следующий кадр из видеофайла, хранящегося в объекте `capture`, и записывает его в переменную `m` класса `Mat`. Четвертая строка кода позволяет выводить считанное изображение в `pictureBox` – специальную область на экране программы для вывода распознанных изображений.

Далее необходимо получить информацию о считанном видео с помощью следующих строк программного кода:

```
05. fps = capture.GetCaptureProperty(Emgu.CV.CvEnum.CapProp.Fps);
```

```
06. frames = capture.GetCaptureProperty(Emgu.CV.CvEnum.CapProp.Frame-  
Count);
```

```
07. framesCurrent = 1;
```

Пятая строка кода получает значение частоты кадров из объекта `capture`. Шестая строка кода позволяет определить общее количество кадров в видео. И на седьмой строке кода инициализируется начальное значение переменной `framesCurrent`.

Реализация вышеуказанных семи строк программного кода позволяет открыть в окне программы только первый кадр загруженного видео (см. рис. 1).

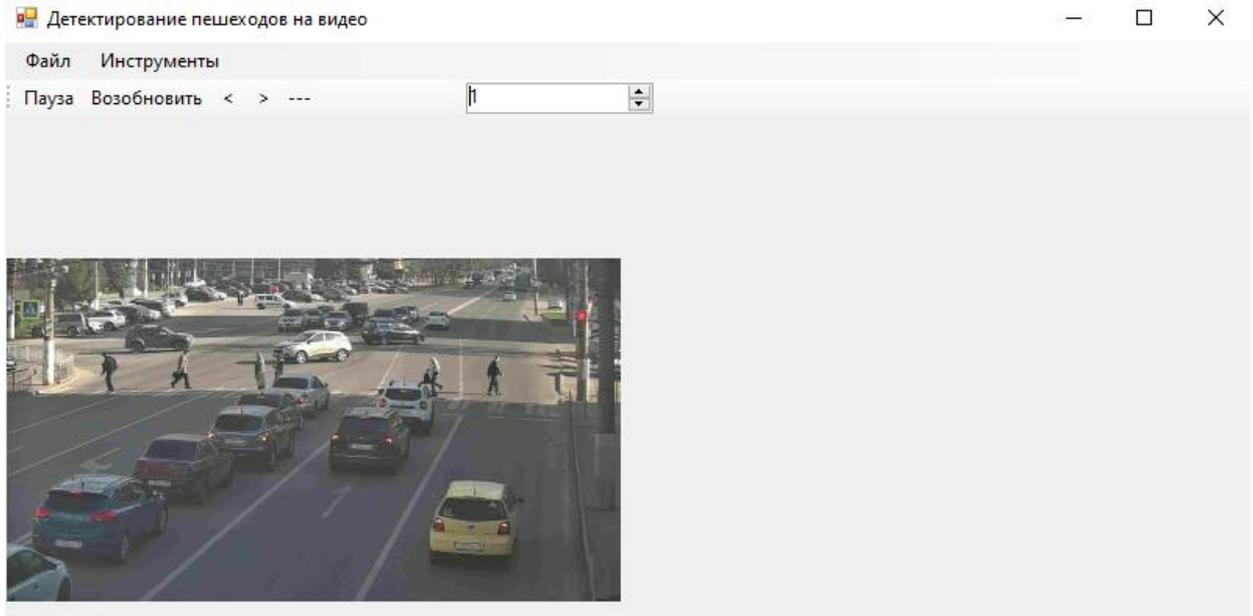


Рис. 1. Загрузка первого кадра (г. Курск, пересечение улиц Карла Маркса и Хуторской)

1.1. Процедура поиска пешеходов на перекрестке

Первоначально создадим метод Find для поиска людей на пешеходном перекрестке с помощью следующей директивы:

```
08. private Image<Bgr, byte> Find(Image<Bgr, byte> image)
```

Данный метод принимает аргумент типа Image<Bgr, byte>, который представляет собой изображение в цветовом пространстве BGR (синий, зеленый, красный) с 8-битным порогом хранения цветных пикселей и возвращает картинку того же формата с найденными пешеходами на ней. Для поиска пешеходов создаем объект regions класса MCvObjectDetection, который будет содержать информацию о распознанных объектах на изображении (координаты и их размеры)

```
09. MCvObjectDetection[] regions;
```

Затем используем метод обнаружения людей на изображении с помощью алгоритма Гистограммы направленных градиентов [13, 14]:

```
10. using(HOGDescriptor descriptor = new HOGDescriptor()) {
```

```
11.   descriptor.SetSVMDescriptor(HOGDescriptor.GetDefaultPeopleDetector());
```

```
12.   regions = descriptor.DetectMultiScale(image); }
```

В данном коде HOGDescriptor является классом библиотеки EmguCV, который реализует алгоритм HOG для обнаружения объектов на изображении. На одиннадцатой строке кода применен SVM-детектор [15, 16], который используется для обнаружения людей, при этом метод GetDefaultPeopleDetector() возвращает предварительно обученный детектор, используемый для распознавания людей. На двенадцатой строке кода в массив regions сохраняются распознанные пешеходы на изображении с помощью метода многомасштабного обнаружение объектов DetectMultiScale(image) на изображении.

Алгоритм гистограммы направленных градиентов состоит из шести этапов. На первом этапе осуществляется извлечение градиентов. Изображение разбивается на маленькие ячейки (обычно 8×8 пикселей), и для каждой из них вычисляются горизонтальные и вертикальные градиенты. Градиенты представляют направление и интенсивность изменения яркости в ячейке [7]. На втором этапе происходит создание гистограмм. Для каждой ячейки строится гистограмма направлений градиентов, содержащая бины, представляющие различные направления градиентов. И после этого гистограммы объединяются в блоки (по умолчанию 2×2 пикселя). На третьем этапе для уменьшения влияние изменений освещения применяется нормализация изображения. После этого на четвертом шаге создаётся дескриптор. При этом все нормализованные блоки объединяются в один вектор, который называется «дескриптором HOG», и этот дескриптор содержит информацию о текстуре и форме объекта. Далее осуществляется обучение классификатора, например, с помощью метода опорных векторов, который определяет, является ли область изображения искомым объектом, – в данном случае пешеходом. На заключительном, шестом этапе происходит обнаружение объектов. Если классификатор считает, что окно содержит искомый объект, то оно считается обнаруженным пешеходом и выделяется рамкой.

После обнаружения всех пешеходов на изображении выделим их на изображении с помощью красной рамки:

```
13. foreach(MCvObjectDetection pesh in regions) {  
14.   image.Draw(pesh.Rect, new Bgr(Color.Red), 3);  
15.   CvInvoke.PutText(image, "Пешеход", new Point(pesh.Rect.X, pesh.Rect.Y),  
                        Emgu.CV.CvEnum.FontFace.HersheyPlain, 1,
```

```
new MCvScalar(255, 255, 255), 2); }
```

```
16. return image;
```

Указанный код выполняет обработку массива `regions`, который содержит обнаруженные объекты – распознанных пешеходов на изображении. В цикле `foreach` (13-я строка) перебирается каждый элемент массива `regions`. Каждый элемент `resh` представляет собой распознанного пешехода. На четырнадцатой строке кода вокруг области с распознанным пешеходом рисуется прямоугольник красного цвета с толщиной линии 3 пикселя. С помощью пятнадцатой строки кода рядом с красным прямоугольником добавляется текст белого цвета, отображающий надпись «Пешеход». Шестнадцатая строка кода возвращает в переменную `image` обработанное изображение с найденными пешеходами и нарисованными вокруг них прямоугольниками красного цвета и надписями белого цвета.

Таким образом, метод `Find(Image<Bgr, byte> image)` возвращает новое изображение с распознанными пешеходами, выделенными красными прямоугольными рамками с надписями «Пешеход».

1.2. Процедура считывания кадров из видеоизображения и их обработка

Данный метод позволяет выводить исходное и обработанное изображения с помощью метода `Find` в `pictureBox` на форме компьютерного приложения. Программный код данного метода имеет следующий вид:

```
17. private async void ReadFrames(){
```

```
18.     Mat m = new Mat();
```

```
19.     while(play && framesCurrent < frames){
```

```
20.         framesCurrent += 1;
```

```
21.             capture.SetCaptureProperty(Emgu.CV.CvEnum.CapProp.PosFrames, framesCurrent);
```

```
22.             capture.Read(m);
```

```
23.             pictureBox1.Image = m.Bitmap;
```

```
24.             pictureBox2.Image = Find(m.ToImage<Bgr, byte>()).Bitmap;
```

```
25.             toolStripLabel1.Text = $"{framesCurrent} / {frames}";
```

```
26.    await Task.Delay(1000 / Convert.ToInt16(fps));}}
```

Метод `ReadFrames()` использует асинхронный режим работы, поскольку объявлен с ключевым словом `async`, и позволяет приложению выполнять другие задачи в то время, пока текущая операция ожидает завершения [17]. Логика этого метода заключается в следующем.

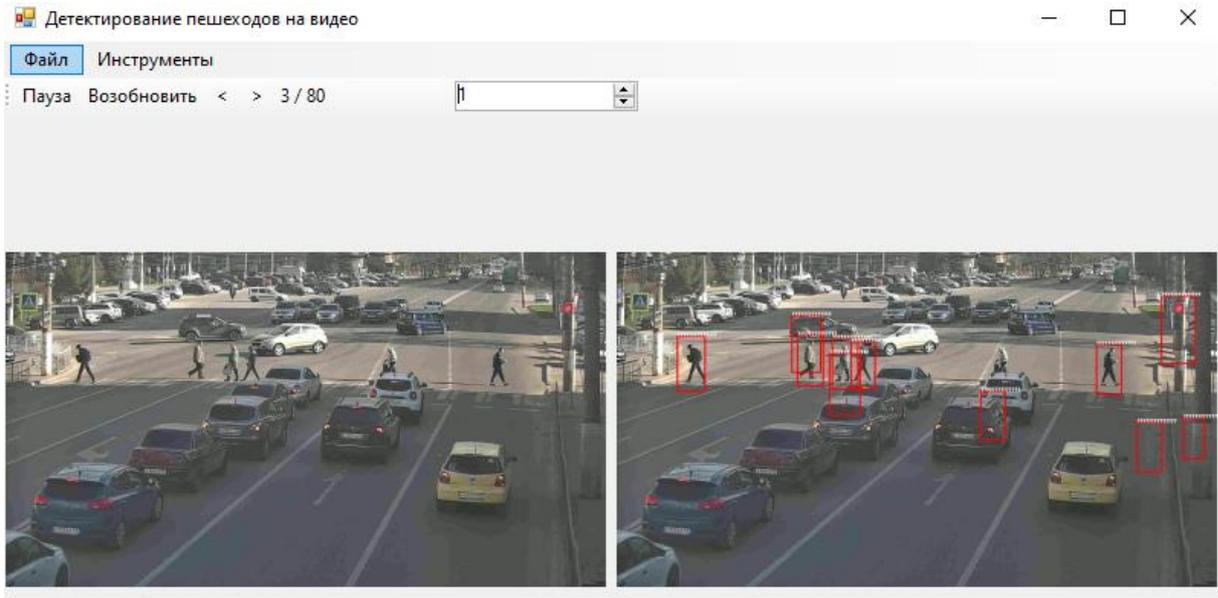
Цикл `while` перебирает все кадры введенного видеоизображения (условие `framesCurrent < frames`) или останавливается при нажатии на кнопку «Пауза» (см. рис. 1). В процессе реализации цикла `while` выполняются следующие действия. Во-первых, осуществляется увеличение счетчика текущих кадров `framesCurrent`. На двадцать первой строке кода свойство `PosFrames` получает и записывает текущую позицию (количество прочитанных кадров) в видеофайле или видеопотоке в переменную `framesCurrent`. Далее на двадцать второй строке кода осуществляется считывание кадра из видеоизображения в объект `m`. Затем на двадцать третьей и двадцать четвертой строках кода осуществляется помещение изображений в `pictureBox1` и `pictureBox2`. В `pictureBox1` хранится исходное изображение видеоизображения, в `pictureBox2` помещается обработанное видеоизображения с распознанными пешеходами на нём с помощью метода `Find`. Далее в метку `toolStripLabel1` записываются два значения номер текущего кадра `framesCurrent` и их общее количество `frames`. И на последней строке этого метода делается задержка в 1 секунду с помощью функции `await`.

МОДЕЛИРОВАНИЕ ПРОЦЕССА РАСПОЗНАВАНИЯ ЛЮДЕЙ НА ПЕШЕХОДНОМ ПЕРЕХОДЕ

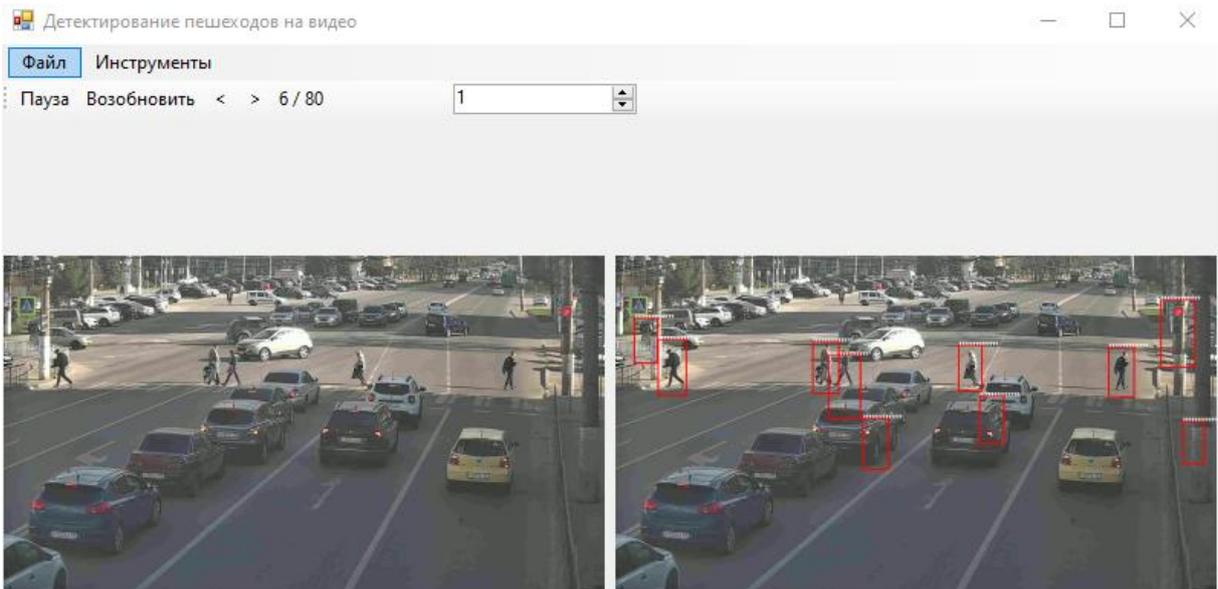
Метод детектирования перемещения людей на пешеходном переходе реализован в среде Microsoft Visual Studio на языке программирования C#, с использованием пакета Emgu CV v. 4.1.1.3497. Для экспериментов использовались персональный компьютер Intel(R) Core(TM) i5-8600K CPU 3,60 ГГц, RAM 16 ГБ, и операционная система Win10. В процессе эксперимента определялась надежность распознавания движущихся пешеходов [18].

Визуализация процесса детектирования движущихся пешеходов представлена на рисунке 2. Причем первый кадр показан на рисунке 1.

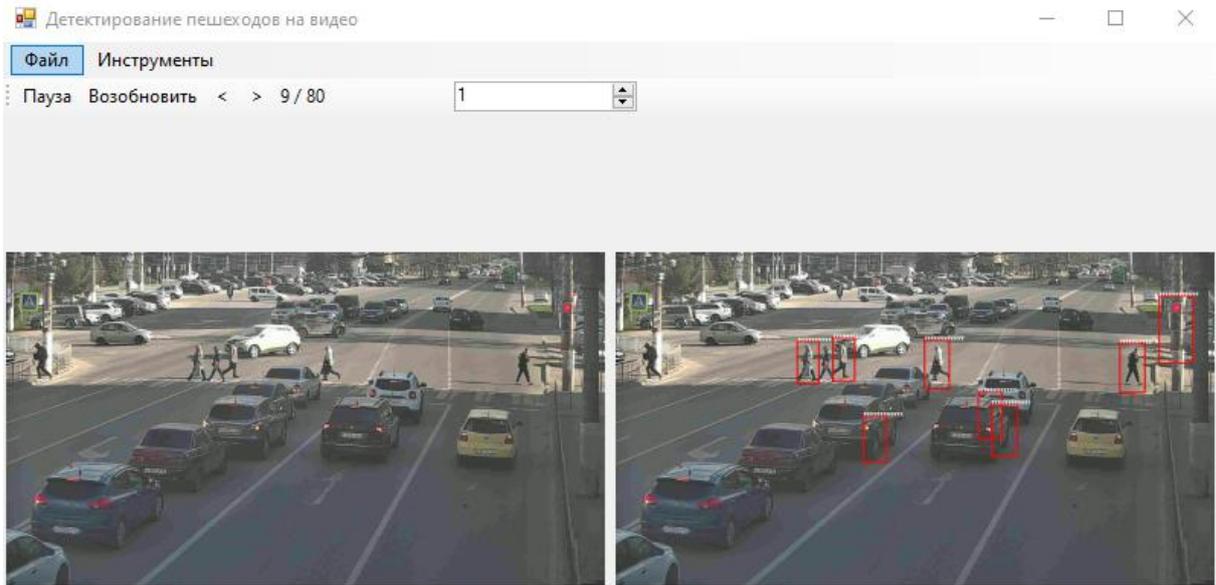
a)



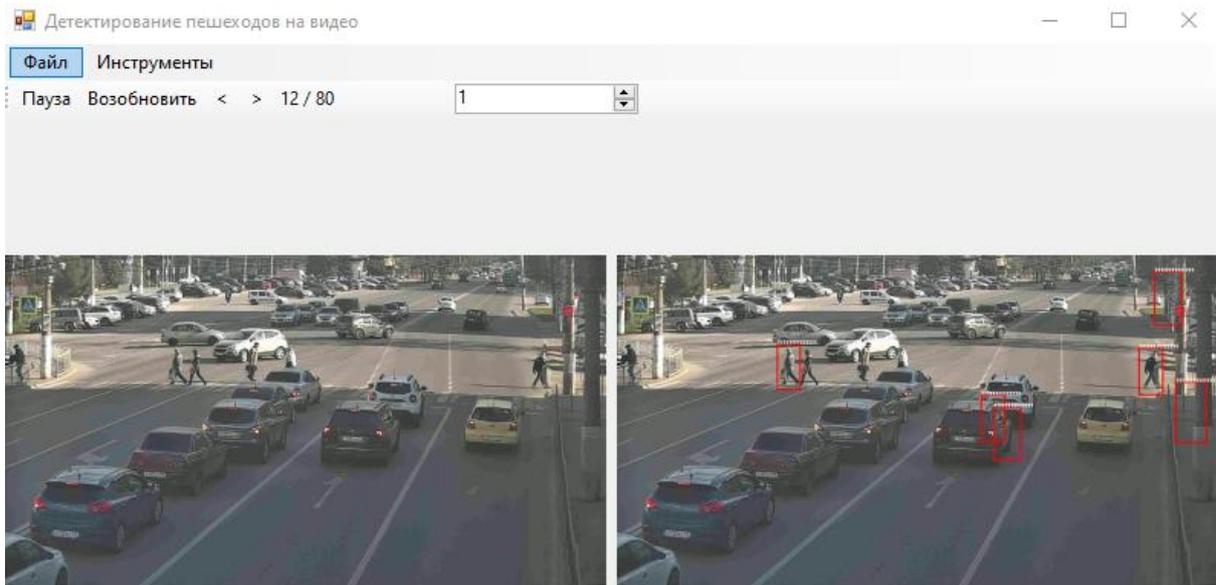
б)



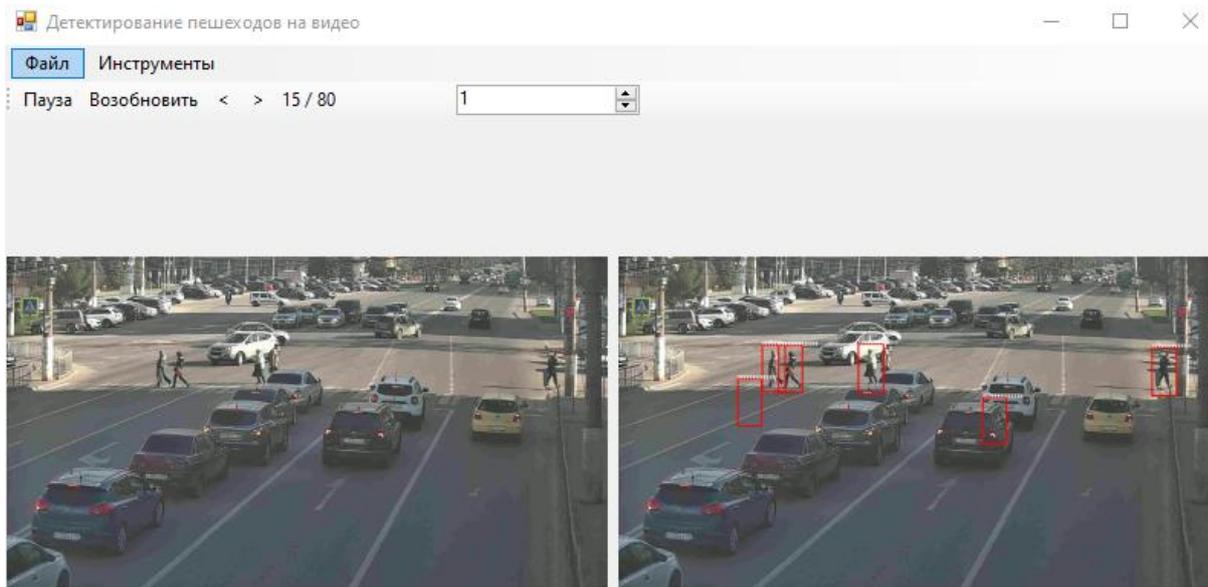
в)



г)



д)



е)

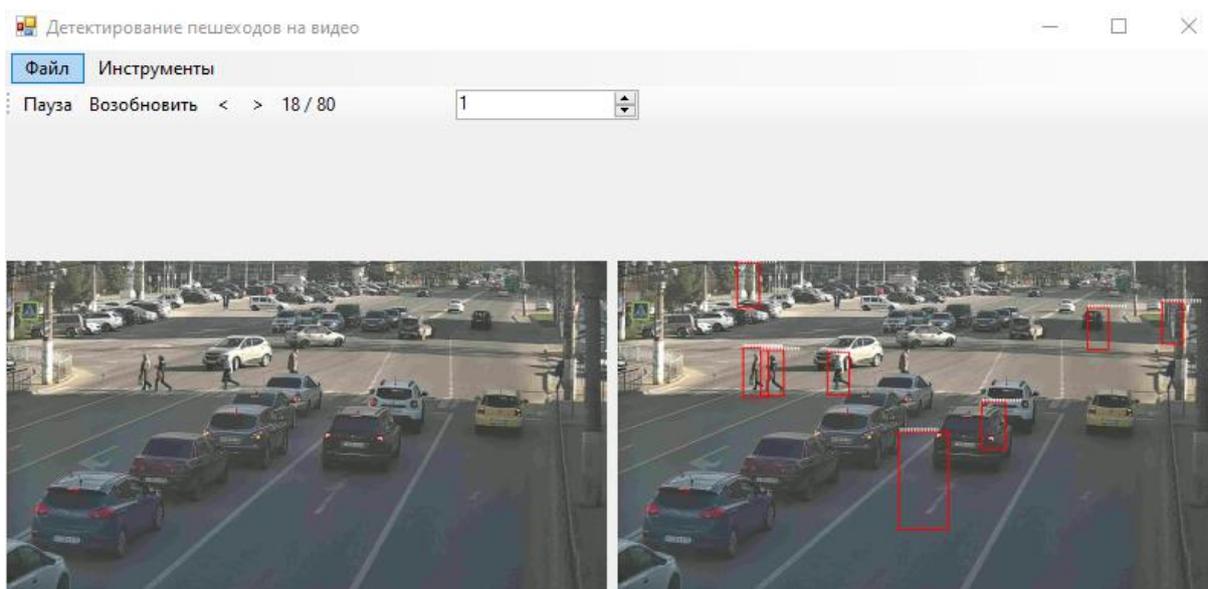


Рис. 2. Детектирование движения пешеходов: а – 3-й кадр; б – 6-й кадр; в – 9-й кадр; г – 12-й кадр; д – 15-й кадр; е – 18-й кадр

Эксперименты повторялись 100 раз для каждого из изображений, по которому необходимо было детектировать движения пешеходов. Результаты вычислительного эксперимента для шести изображений (см. рис. 2) сведены в таблицу 1.

Таблица 1. Анализ детектирования движения пешеходов

Эксперимент	Количество обнаруженных объектов		
	Правильно обнаруженные объекты	Ложно обнаруженные объекты	Пропущенные объекты
Рис. 2а	5	6	1
Рис. 2б	5	5	0
Рис. 2в	4	4	2
Рис. 2г	2	4	4
Рис. 2д	4	2	0
Рис. 2е	3	5	0
	TP	FP	FN
Сумма	23	26	7

Надежность результатов эксперимента вычислялась по формуле

$$N = \frac{TP}{TP + FP + FN},$$

где TP – количество верно обнаруженных объектов; FP – количество ложно обнаруженных объектов; FN – количество пропущенных объектов.

С учетом данных таблицы 2 надежность

$$N = \frac{23}{23 + 26 + 7} = 0.411,$$

т. е. составила 41,1%, что говорит о невысокой надежности метода распознавания движения пешеходов на основе комбинации методов гистограммы направленных градиентов и опорных векторов, что объясняется тем, что видеокамеры установлены далеко от пешеходного перехода.

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ И ЗАКЛЮЧЕНИЕ

Анализ моделирования процесса детектирования движения людей на пешеходном переходе показал, что:

1. Предложенный метод детектирования движения пешеходов, основанный на комбинации методов гистограммы направленных градиентов и опорных векторов, без использования специально обученных нейронных сетей или им подобных обеспечивает детектирование практически всех объектов, относящихся к классу людей.

2. Надежность рассмотренного подхода, основанного на комбинации методов гистограммы направленных градиентов и опорных векторов, составила порядка 41%.

3. В ходе экспериментальных исследований установлено, что предложенный метод распознавания движения людей на пешеходном переходе применим для обработки цветowych изображений, что делает возможным его использование в системах типа «Умный город».

4. Необходим поиск новых решений, позволяющих увеличить надежность рассмотренного подхода.

Благодарности

Работа выполнена при поддержке гранта РФФ 24-21-00055.

СПИСОК ЛИТЕРАТУРЫ

1. Фатихов Ч.И., Григорян К.А. Разработка системы сбора данных о перемещении людей внутри помещения // Электронные библиотеки. 2022. Т. 25, № 1. С. 87–102. <https://doi.org/10.26907/1562-5419-2022-25-1-87-102>

2. Ямиков Р.Р., Григорян К.А. Анализ и разработка конвейера MLOps для развертывания моделей машинного обучения // Электронные библиотеки. 2022. Т. 25, № 2. С. 177–196. <https://doi.org/10.26907/1562-5419-2022-25-2-177-196>

3. Байрамов А.И., Фасхутдинов Т.Р., Тимергалин Д.М. [и др.]. Оценка усталости человека методом анализа фотографий лица с помощью сверточных нейронных сетей // Электронные библиотеки. 2021. Т. 24, № 4. С. 582–603. <https://doi.org/10.26907/1562-5419-2021-24-4-582-603>.

4. *Поташников А.М., Власюк И.В., Аугсткальн Я.А.* Анализ методов детектирования движущихся объектов различного типа на видеоизображении // *Фундаментальные проблемы радиоэлектронного приборостроения*. 2017. Т. 17, № 4. С. 1201-1204.

5. Скользящая_средняя
URL: https://ru.wikipedia.org/wiki/Скользящая_средняя (доступ свободный: дата обращения 19.06.2024)

6. *Canny J.* A computational approach to edge detection // *IEEE Trans Pattern Anal. Mach. Intell.* 1986. Т8(6). Р. 679–698.

7. *Бобырь М.В., Архипов А.Е., Горбачев С.В., Цао Ц., Бхаттачарья С.* Не-четко-логические методы в задаче детектирования границ объектов // *Информатика и автоматизация*. 2022. Т. 21, №2. С. 376–404.
<https://doi.org/10.15622/ia.21.2.6>

8. *Raj R., Rajiv P., Kumar P., Khari M., Verdú E., Crespo R. G., Manogaran G.* Feature based video stabilization based on boosted HAAR Cascade and representative point matching algorithm // *Image and Vision Computing*. 2020. Т101.
<https://doi.org/10.1016/j.imavis.2020.103957>

9. *Бурмистров А.В.* Распознавание объектов на изображениях с использованием базовых средств языка Python и библиотеки OpenCV // *Научное обозрение. Технические науки*. 2021. № 5. С. 15–19.

10. *Gou Y., Li Q., Yao R., Chen J., Zhao H., Zhang Z.* Ice accretion existence and three-dimensional shape identification based on infrared thermography detection // *Infrared Physics and Technology*. 2023. Т135.
<https://doi.org/10.1016/j.infrared.2023.104972>

11. *Zhou Z., Yu X., Chen X.* Object Detection in Drone Video with Temporal Attention Gated Recurrent Unit Based on Transformer // *Drones*. 2023. Vol. 7, No. 7.
<https://doi.org/10.3390/drones7070466>

12. *Буров С.А., Титов К.Б., Иванов А.Р.* Программа прогнозирования координат движения целей на основе рекуррентной искусственной нейронной сети. Свидетельство о государственной регистрации программы для ЭВМ № 2021613317 РФ. 05.03.2021.

13. Ярошевич П.В., Богуш Р.П. Алгоритм классификации изображений парковочных мест автостоянки на основе гистограмм ориентированных градиентов и метода опорных векторов // Компьютерная оптика. 2017. Т. 41. № 1. С. 110–117. <https://doi.org/10.18287/2412-6179-2017-41-1-110-117>

14. Побережник В.И. Классификация BIM структур с помощью гистограммы направленных градиентов и метода опорных векторов // Приложение математики в экономических и технических исследованиях. 2020. Т. 10, № 1. С. 140–146.

15. Rahman M., Zhou Yu., Wang Sh., Rogers Ja. Wart Treatment Decision Support Using Support Vector Machine // International Journal of Intelligent Systems and Applications. 2020. Vol. 12, No. 1. P. 1–11. <https://doi.org/10.5815/ijisa.2020.01.01>

16. Присухина И.В., Борисенко Д.В. Машинная классификация режима работы электрической рельсовой цепи методом опорных векторов // Омский научный вестник. 2018. Т. 162, №6. С. 126–130. <https://doi.org/10.25206/1813-8225-2018-162-126-130>

17. Как на самом деле Async/Await работают в C#. Часть 6. Анализ результатов компиляции асинхронных вызовов URL: <https://habr.com/ru/articles/792782/> (доступ свободный: дата обращения 19.06.2024).

18. Бобырь М.В., Храпова Н.И., Архипов А.Е., Милостная Н.А. Программа детектирования движения объектов на пешеходном переходе. Свидетельство о государственной регистрации программы для ЭВМ № 2024660940 РФ. опубли. 14.05.2024.

ON THE APPROACH TO DETECTING PEDESTRIAN MOVEMENT USING THE METHOD OF HISTOGRAMS OF ORIENTED GRADIENTS

M. V. Bobyr¹ [0000-0002-5400-6817], N. A. Milostnaya² [0000-0002-3779-9165],

N. I. Khrapova³ [0000-0001-7947-1427]

¹⁻³Southwest State University, Kursk region, Kursk, 305040, Russia

¹maxbobyr@gmail.com, ²nat_mil@mail.ru, ³KhrapovaNI@yandex.ru

Abstract

An approach to automatically recognizing the movement of people at a pedestrian crossing presented in the article. This approach includes two main procedures, for each of which program code commands are given in the C# programming language using the EMGU computer vision library. In the first procedure, pedestrian detection is carried out using a combination of directional gradient histogram and support vector methods. The second procedure allows you to read frames from a video sequence and process them. This approach allows detecting the movements of people at a pedestrian crossing without using specialized neural networks. At the same time, the method proposed in the article demonstrated sufficient reliability of human movement recognition, which indicates its applicability in real conditions.

Keywords: *Pedestrian Motion Recognition, EMGU, Histogram of Oriented Gradients, Support Vector Machine*

REFERENCES

1. *Fatihov Ch.I., Grigorjan K.A.* Razrabotka sistemy sbora dannyh o peremeshhenii ljudej vnutri pomeshhenija // Russian digital library. 2022. T. 25, № 1. S. 87–102. <https://doi.org/10.26907/1562-5419-2022-25-1-87-102>
2. *Jamikov R.R., Grigorjan K.A.* Analiz i razrabotka konvejera MLOps dlja razvertyvanija modelej mashinnogo obuchenija // Jelektronnye biblioteki. 2022. T. 25, № 2. S. 177–196. <https://doi.org/10.26907/1562-5419-2022-25-2-177-196>
3. *Bajramov A.I., Fashutdinov T.R., Timergalin D.M.* [i dr.]. Ocenka ustalosti cheloveka metodom analiza fotografij lica s pomoshh'ju svertochnyh nejronnyh setej // Jelektronnye biblioteki. 2021. T. 24, № 4. S. 582–603. <https://doi.org/10.26907/1562-5419-2021-24-4-582-603>.

4. Potashnikov A.M., Vlasjuk I.V., Augstkal'n Ja.A. Analiz metodov detektirovaniya dvizhushhihsja ob#ektov razlichnogo tipa na videoizobrazhenii // Fundamental'nye problemy radioelektronnoho priborostroenija. 2017. T. 17, № 4. S. 1201–1204.

5. Moving average

URL: https://en.wikipedia.org/wiki/Moving_average (date of access 19.06.2024)

6. Canny J. A computational approach to edge detection // IEEE Trans Pattern Anal. Mach. Intell. 1986. Vol. 8. No. 6. P. 679–698.

7. Bobyr M.V., Arhipov A.E., Gorbachev S.V., Cao C., Bhattachar'ja S. Nechetko-logicheskie metody v zadache detektirovaniya granic obektov // Informatika i avtomatizacija. 2022. T. 21(2). S. 376–404. <https://doi.org/10.15622/ia.21.2.6>

8. Raj R., Rajiv P., Kumar P., Khari M., Verdú E., Crespo R. G., Manogaran G. Feature based video stabilization based on boosted HAAR Cascade and representative point matching algorithm // Image and Vision Computing. 2020. T101. <https://doi.org/10.1016/j.imavis.2020.103957>

9. Burmistrov A.V. Raspoznavanie ob#ektov na izobrazhenijah s ispol'zovaniem bazovyh sredstv jazyka Python i biblioteki OpenCV // Nauchnoe obozrenie. Tehniceskie nauki. 2021. № 5. S. 15–19.

10. Gou Y., Li Q., Yao R., Chen J., Zhao H., Zhang Z. Ice accretion existence and three-dimensional shape identification based on infrared thermography detection // Infrared Physics and Technology. 2023. T135. <https://doi.org/10.1016/j.infrared.2023.104972>

11. Zhou Z., Yu X., Chen X. Object Detection in Drone Video with Temporal Attention Gated Recurrent Unit Based on Transformer // Drones. 2023. Vol. 7, No. 7. <https://doi.org/10.3390/drones7070466>

12. Burov S.A., Titov K.B., Ivanov A.R. Programma prognozirovanija koordinat dvizhenija celej na osnove rekurrentnoj iskusstvennoj nejronnoj seti. Svidetel'stvo o gosudarstvennoj registracii programmy dlja JeVM № 2021613317 RF. 05.03.2021

13. Jaroshevich P.V., Bogush R.P. Algoritm klassifikacii izobrazhenij parkovochnyh mest avtostojanki na osnove gistogramm orientirovannyh gradientov i metoda opornyh vektorov // Komp'juternaja optika. 2017. T. 41. № 1. S. 110–117. <https://doi.org/10.18287/2412-6179-2017-41-1-110-117>.

14. *Poberezhnik V.I.* Klassifikacija BIM struktur s pomoshh'ju gistogrammy napravlennyh gradientov i metoda opornyh vektorov // Prilozhenie matematiki v jekonomicheskikh i tehniceskikh issledovanijah. 2020. T. 10, № 1. S. 140–146.

15. *Rahman M., Zhou Yu., Wang Sh., Rogers Ja.* Wart Treatment Decision Support Using Support Vector Machine // International Journal of Intelligent Systems and Applications. 2020. Vol. 12, No. 1. P. 1–11. <https://doi.org/10.5815/ijisa.2020.01.01>

16. *Prisuhina I.V., Borisenko D.V.* Mashinnaja klassifikacija rezhima raboty jektricheskoy rel'sovoj cepi metodom opornyh vektorov // Omskij nauchnyj vestnik. 2018. T. 162, №6. S. 126–130.

<https://doi.org/10.25206/1813-8225-2018-162-126-130>.

17. Kak na samom dele Async/Await rabotayut v C#. CHast' 6. Analiz rezul'tatov kompilyacii asinhronnyh vyzovov.

URL: <https://habr.com/ru/articles/792782/> (date of access 19.06.2024).

18. *Bobyр M.V., Hrapova N.I., Arhipov A.E., Milostnaja N.A.* Programma dektirovaniya dvizhenija obektov na peshehodnom perehode. Svidetel'stvo o gosudarstvennoj registracii programmy dlja JeVM № 2024660940 RF. opubl. 14.05.2024.

СВЕДЕНИЯ ОБ АВТОРАХ



БОБЫРЬ Максим Владимирович – Доктор технических наук, профессор. Профессор кафедры «Программная инженерия», ФГБОУ ВО «Юго-Западный государственный университет». Области исследований: интеллектуальные системы управления, адаптивные нейро-нечеткие системы вывода, распознавание и обработка изображения, робототехнические системы.

Maxim V. BOBYR – Doctor of technical sciences, professor. Professor of the Department of Software Engineering, Southwest State University. Research areas: intelligent control systems, adaptive neuro-fuzzy inference systems, image recognition and processing, robotic systems.

email: maxbobyр@gmail.com

ORCID: 0000-0002-5400-6817



МИЛОСТНАЯ Наталья Анатольевна – Доктор технических наук. Ведущий научный сотрудник кафедры «Программная инженерия», ФГБОУ ВО «Юго-Западный государственный университет». Области исследований: методы оптимизации на основе динамического программирования, робототехнические и автоматизированные системы управления сложными объектами.

Natalya A. MILOSTNAYA – Doctor of technical Sciences. Leading Researcher, Department of Software Engineering, Southwest State University. Research areas: optimization methods based on dynamic programming, robotic and automated control systems for complex objects.

email: nat_mil@mail.ru

ORCID: 0000-0002-3779-9165



ХРАПОВА Наталья Игоревна – аспирант кафедры «Программная инженерия», ФГБОУ ВО «Юго-Западный государственный университет». Область исследования: распознавание сложных объектов с использованием нечеткой логики.

Natalia Igorevna KHRAPOVA – Postgraduate student of the Department of Software Engineering at the Southwest State University. Research area: recognition of complex objects using fuzzy logic.

email: KhrapovaNI@yandex.ru

ORCID: 0000-0001-7947-1427

Материал поступил в редакцию 15 июня 2024 года