

РАЗРАБОТКА СИСТЕМЫ РАСПОЗНАВАНИЯ ПЕШЕХОДОВ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ ДЛЯ ЕЕ ПРИМЕНЕНИЯ В КОНЦЕПЦИИ УМНОГО ГОРОДА

Я. С. Яскеляйнен¹, А. А. Хакимов^{1*}, М. С. А. Мутанна¹,
А. С. А. Мутханна¹, Р. В. Киричѐк¹

¹ СПбГУТ, Санкт-Петербург, 193232, Российская Федерация

* Адрес для переписки: abdukadir94@gmail.com

Аннотация

В данной статье рассматривается вопрос о возможности распознавания пешеходов в режиме реального времени на конечных устройствах при помощи микрокомпьютера для автоматизации процесса сигнализации светофора и повышения мобильности системы. **Предмет исследования.** Статья посвящена краткому анализу существующих систем обнаружения пешеходов, с последующей разработкой собственной системы на основе изученных, и ее дальнейшему внедрению на микрокомпьютер Raspberry Pi. **Метод.** Проведен натуральный эксперимент по обнаружению пешеходов на изображении с последующим определением их нахождения у пешеходного перехода. **Основные результаты.** Проведен анализ ряда популярных систем обнаружения пешеходов с точки зрения быстродействия, точности определения, и возможности внедрения на микрокомпьютеры. Даны рекомендации по выбору системы, условиям применения и улучшения характеристик. **Практическая значимость.** Разработанная система автоматизирует работу светофора через внедрение дополнительных функций (отслеживание позы пешехода, перемещения пешеходов вдоль участка проезжей части), обеспечивая безопасность участников дорожного движения, что позволяет улучшить уровень городской среды в целом.

Ключевые слова

Интернет Вещей, IoT, deep learning, распознавание образов, обнаружение пешеходов, Raspberry Pi, openCV, tensorflow, гистограммы направленных градиентов, HOG, каскады Хаара.

Информация о статье

УДК 004.77

Язык статьи – русский.

Поступила в редакцию 27.07.18, принята к печати 03.09.18.

Ссылка для цитирования: Яскеляйнен Я. С., Хакимов А. А., Мутанна М. С. А., Мутханна А. С. А., Киричѐк Р. В. Разработка системы распознавания пешеходов в режиме реального времени для ее применения в концепции Умного города // Информационные технологии и телекоммуникации. 2018. Том 6. № 3. С. 123–136.

DEVELOPMENT OF REAL TIME PEDESTRIAN RECOGNITION SYSTEM AND ITS APPLICATION IN THE CONCEPT OF SMART CITY

I. Jaskelainen, A. Khakimov^{1*}, M. S. A. Mutanna¹,
A. S. A. Muthanna¹, R. Kirichek¹

¹ SPbSUT, St. Petersburg, 193232, Russian Federation

* Corresponding author: abdukadir94@gmail.com

Abstract—This article discusses the possibility of real time pedestrians recognition on end devices using a microcomputer for automation of traffic lights signaling process and system mobility improvement. **Subject research.** This article is devoted to a brief analysis of existing systems for pedestrians detection and subsequent development of new system based on the studied ones and its further implementation on the Raspberry Pi microcomputer. **Method.** Was conducted full-scale experiment to detect pedestrians on image with the subsequent determination of their location at a crosswalk. **Core results.** A number of popular pedestrian detection systems have been analyzed in terms of speed, accuracy of determination and possibility of implementation on microcomputers. Are given recommendations on the choice of the system, conditions of implementation and characteristic improvements. **Practical relevance.** The developed system automates the working process of traffic light through by the means of introduction of additional functions (tracking of pedestrian's posture, moving pedestrians along roadway section), ensuring the safety of road users, which allows to improve the level of urban environment generally.

Keywords—Internet of Things, IoT, deep learning, pattern recognition, pedestrian detection, Raspberry Pi, openCV, TensorFlow, HOG, Haar cascades.

Article info

Article in Russian.

Received 27.07.18, accepted 03.09.18.

For citation: Jaskelainen I., Khakimov A., Mutanna M. S. A., Muthanna A. S. A., Kirichek R.: Development of Real Time Pedestrian Recognition System and Its Application in the Concept of Smart City // Telecom IT. 2018. Vol. 6. Iss. 3. pp. 123–136 (in Russian).

Введение

В рамках концепции Интернета вещей одним из перспективных направлений развития является «Умный город». Фундаментальным принципом «умного города» является внедрение информационных технологий и объектов IoT в городскую среду.

Сбор, обмен и передача данных между городскими инфраструктурами, уменьшение издержек рабочих процессов благодаря автоматизации процессов, а также налаживание обратной связи между администрацией и горожанами для благоустройства среды, являются основными целями концепции Умного города, которые в конечном итоге должны повысить уровень жизни граждан [1].

Задачи, решаемые Умным городом обширны, одной из них является управление городским транспортом и дорожным трафиком. Уже сейчас в некоторых

городах как в тестовом, так и в эксплуатационном режиме, внедрены умные беспилотные автобусы, информирование и оповещение водителей в режиме реального времени, адаптивное управление городскими светофорами на основе машинного обучения для разгрузки дорожного трафика и т. д. Однако, помимо транспорта одним из участников движения также является пешеход [3, 4].

Обеспечение безопасности пешеходов и повышение качества обслуживания через интеллектуальную систему автоматизации процессов, являются основными задачами для улучшения пешеходной инфраструктуры.

В настоящее время на оживленных магистралях устанавливают, как правило, автоматически переключающиеся светофоры. Но часто применяется и вариант, когда светофор переключается после нажатия специальной кнопки и разрешает переход в течение определенного времени после этого. Но что если светофор сам, в автоматическом режиме будет распознавать человека, намеревающегося перейти дорогу? Это позволит не только избавить человека от необходимости вручную активировать возможность переходить проезжую часть, но и проинформировать водителей приближающихся транспортных средств о намерениях пешехода. Причем не только с помощью традиционного сигнала светофора, но и через систему навигаторов, умных приборных панелей, лобовых стекол с дополненной реальностью. Более того, помимо распознавания как одного, так и нескольких пешеходов, «умный» светофор, может «наблюдать» за успешным прохождением объектов проезжей части, т. е. из точки «А» (одна сторона проезжей части) в точку «Б» (противоположная сторона проезжей части) и в случае возникновения чрезвычайной ситуации или ДТП, система распознавания сможет проанализировать ситуацию и оперативно сообщить в службу спасения, что существенно поможет выиграть время в сложных ситуациях [2, 4].

Данный тип светофоров, может так же учитывать загруженность как дорожного, так и пешеходного трафика, «общаясь» с другими подобными устройствами или центральной интеллектуальной системой, и на основе полученных данных принимать те или иные решения.

Постановка задачи

Компьютерное зрение и распознавание образов является неотъемлемой частью искусственного интеллекта, в работе которого используются Deep Learning, Big Data и облачные сервисы для анализ данных. Облака предоставляют огромную вычислительную мощность, но для передачи видео, даже после сжатия, требуются каналы с очень высокой пропускной способностью. Отсутствие полного покрытия городской местности мобильными сетями нового поколения, необходимость хранения огромных объемов данных, издержки полосы пропускания, загруженность сетевого трафика, а также возможные проблемы с безопасностью и конфиденциальностью данных заставляют переосмысливать подходы к применению облачных вычислений и компьютерного зрения в условиях городской среды.

В настоящий момент видеoinформацию возможно собирать и анализировать на уровне устройства или небольшого мобильного объекта. В свою очередь конечное устройство, должно оснащаться датчиками, сенсорами, графическими ускорителями и производительным железом, способное передавать видеопоток

с высоким разрешением и минимальными задержками, и способствующее оптимальной работе системы в целом. На верхнем уровне находится операционная система, на которой реализована программная часть по распознаванию объектов на изображении, с использованием того или иного подхода [5, 6].

Одни из подходов основаны на реализации известных алгоритмов по распознаванию через высокоуровневые языки программирования с подключаемыми библиотеками в набор которых входит функции по обработке матриц, векторов, классов принадлежности. Другие системы используют готовые фреймворки с открытым исходным кодом, позволяющий модифицировать модули под текущие задачи. Общим у данных подходов является наличие набора данных для обучения, который обрабатывается согласно заданному алгоритму и происходит обучение модели, которая затем получая поток новых данных может классифицировать тот или иной объект на изображении.

На текущий момент имеется большое количество созданных систем по распознаванию образов, включая человеческие фигуры.

Одна из таких систем – распознавание движущихся пешеходов (*Walking pedestrian recognition* [10]). В статье описывается метод обнаружения, отслеживания и окончательного распознавания пешеходов, пересекающих траекторию движущегося наблюдателя. Осуществляется комбинация подходов, основанных на данных и моделях. Процесс первоначального обнаружения основан на слиянии анализа текстуры, на основе моделей, в частности, с геометрическими особенностями пешеходов и обратном-перспективном отображении (бинокулярное зрение). Кроме того, анализы движения движений конечностей анализируются для определения исходных объектных гипотез. Отслеживание квазиригидной части тела выполняется с помощью различных алгоритмов, которые были успешно использованы для отслеживания седанов, грузовиков, мотоциклов и пешеходов. Окончательная классификация получается путем временного анализа процесса ходьбы.

Другая система Pedestrian Detection OpenCV [7], использует библиотеку OpenCV поставляемую с предварительно подготовленной моделью HOG + Linear SVM, которая может использоваться для обнаружения одновременно нескольких пешеходов на изображениях. Особое внимание уделено применению NMS (*Non-Maximum Suppression* – подавление не-максимумов, которое означает, что пикселями границ объявляются точки, в которых достигается локальный максимум градиента в направлении вектора градиента. Данный метод основан на детекторе границ Канни (John F. Canny; 1953). Использование NMS позволяет перекрывать, перекрещивающиеся ограничивающие поля, оставляя правильным окончательным обнаружением, избегая дубликатов.

Существуют уже и готовые коммерческие системы, например, Accuware Sentinel¹. Sentinel обнаруживает движение людей по заданной местности и отслеживают их движения. Обработка видео происходит с камер видеонаблюдения, развернутых вокруг контролируемой зоны, система распознает людей и собирает последовательность их местоположений с течением времени. Так же система может назначать индивидуальные метки пользователям, пока те находятся в зоне

¹ Accuware Inc. 2015–2018. URL: <https://www.accuware.com/support/sentinel-introduction/>

видеонаблюдения. Для внедрения данной системы необходимы камеры, установленные в помещении для передачи видеопотока, мощный кластер или вычислительное облако, инструментальная панель, а также API для интеграции дополнительных возможностей и приложений.

Каждая из этих систем имеет хорошие показатели и отличные результаты работы по обнаружению объектов. Однако они обнаруживают всех пешеходов на изображении.

В рамках данной работы для решения задачи необходима система, которая будет распознавать только тех пешеходов, которые намереваются перейти проезжую часть, одним из признаков является наличие рядом с человеком пешеходного перехода, поэтому должен быть распознан и сам пешеходный переход. В своей работе я изучил особенности работы некоторых таких систем и попытался создать систему по распознаванию пешеходов на пешеходной зоне для мобильной и компактной платформы с минимальными ресурсами.

В экспериментах для создания прототипа системы был использован микрокомпьютер Raspberry Pi 3 Model B, с установленным Linux дистрибутивом, Python 3.5.2 и библиотеками OpenCV, NumPy, а также open-source библиотекой TensorFlow. Видеопоток для анализа и распознавания объектов в режиме реального времени был перенаправлен с некоторых веб-камер города, на кадрах которого находится пешеходный переход.

Формирование требований

Главная задача системы – распознавание пешехода, намеревающегося перейти дорогу, все остальные объекты неудовлетворяющие данному условию, должны либо игнорироваться, либо не распознаваться вовсе. В противном случае, система будет распознавать каждого пешехода вне зависимости от его намерений перейти дорогу, и «выдавать» зеленый свет, тем самым затрудняя движение транспортных средств, вплоть до их блокировки. Исходя из вышесказанного система должна уметь не только распознавать пешехода, но и определять его местонахождение в определенной зоне – начале пешеходного перехода (перед «зеброй»). Таким образом система должна распознавать и сам пешеходный переход.

После того, как система успешно распознала пешехода и «поняла», что он намеревается перейти проезжую часть, происходит обратный отсчет до выполнения стандартной светофорной сигнализации, сообщающей водителям транспортных средств о переходящем дорогу пешеходе.

Основные требования:

- распознавание пешехода, находящегося определенное время в непосредственной близости от «зебры»;
- запуск процедуры перехода проезжей части;
- повтор данных шагов через определенный интервал времени.

Проектирование системы

Существует множество моделей распознавания человека на изображении или потоковом видео. Каждая из них имеет свои преимущества и недостатки, среди них повышенные требования к производительности устройства, скорость

обработки кадров, условия нормализации данных, точность определения объекта и его классификации, и т. д.

Обнаружение объектов методами HOG и Хаара

Библиотека OpenCV уже включает в себя много встроенных функций и обученных моделей для работы по обнаружению объектов на изображении или видео. Одними из таких функций являются алгоритмы каскадов Хаара и гистограммы направленных градиентов HOG с применением методов опорных векторов (SVM).

HOG (гистограмма направленных градиентов) – это дескрипторы особых точек, которые используются в компьютерном зрении и обработке изображений с целью распознавания объектов. Данная техника основана на подсчете количества направлений градиента в локальных областях изображения. Метод обнаружения пешеходов с помощью HOG системы предложен Н. Далалом и Б. Триггсом в их статье «Гистограммы ориентированных градиентов для обнаружения людей», опубликованной в 2005 году.

Подход по обнаружению объектов, основанный на признаках Хаара предложен Полом Виолой и Майклом Джонсом в статье «Быстрое обнаружение объектов с использованием расширенного каскада простых функций», опубликованном в 2001 году. Этот подход широко используется для обнаружения лиц.

Признак Хаара состоит из смежных прямоугольных областей. Они позиционируются на изображении, далее суммируются интенсивности пикселей в областях, после чего вычисляется разность между суммами. Эта разность и будет значением признака с определенным размером, определенным образом спозиционированного на изображении.

OpenCV включает встроенную функциональность, обеспечивающую обнаружение объекта на основе каскада Хаара. В официальной репозитории OpenCV доступны предварительно подготовленные модели для обнаружения человеческой фигуры в полный рост, верхней части и нижней частей человеческой фигуры.

Применение HOG в потоковом видео с использованием Open CV 3.4 на тестовом оборудовании, оснащенном Intel Core i7 6700, 8 Гб памяти, nVidia Geforce GTX 960 4 Гб видеопамати показывает приблизительно 140–160 миллисекунд на кадр (что приблизительно эквивалентно 6 кадрам в секунду).

Применение каскадов Хаара на той же программно-аппаратной части показывает примерно 90–100 миллисекунд на кадр (что приблизительно составляет 11 кадров в секунду) (рис. 1).

Во время экспериментов были отмечены следующие недостатки:



Рис. 1. Применение каскадов Хаара (входят в библиотеку OpenCV)

- модель не очень хорошо обнаруживают людей в разных позах, если только несколько моделей не используются для обнаружения людей в каждой позе;
- модель справляется с обнаружением людей с фронтального и заднего вида, объекты, находящиеся сбоку определяются хуже;
- модель восприимчива к определению нечеловеческих объектов как людей (можно избежать, определив пороговые значения минимального размера окна обнаружения);
- возможны повторные обнаружения и дублирование обнаружений;
- граница обнаружения, обеспечиваемая моделью HOG, не плотно прилегает к обнаруженному человеку. Это может существенно затруднить обнаружение человека рядом с пешеходным переходом, или распознать двух пешеходов, как одного;
- мерцание во время обнаружения. Довольно часто наблюдается, что человек, обнаруженный в одном кадре, не обнаруживается в следующем кадре.

Очевидно, что использовать данные модели для обработки видео в режиме реального времени на компактных устройствах, в силу низкой скорости обработки кадров и высоких требований к производительности не представляется возможным. Возможными вариантами решения могут быть покадровая съемка, с последующим обнаружением объекта на каждом из кадров или задействование графических ускорителей.

Обнаружение пешеходов с помощью библиотеки TensorFlow

TensorFlow – это API с открытым исходным кодом от Google, который широко используется для решения задач машинного обучения, связанных с Deep Learning. В свою очередь Tensorflow имеет библиотеку Object Detection, созданную на основе Tensorflow с возможностью обучения моделей и оценки обнаружения объектов. Для нашего проекта мы будем использовать Tensorflow Detection Model Zoo, которая представляет собой набор предварительно подготовленных моделей, совместимых с Tensorflow Object Detection API. Она состоит из 16 моделей обнаружения объектов, прошедших предварительную подготовку по набору данных COCO (*Common Object in Context*). COCO – это крупномасштабный набор данных обнаружения объектов, сегментации и меток. Эти модели способны обнаруживать более 80 типов объектов, включая людей. Для небольшого микрокомпьютера с небольшими вычислительными мощностями будем использовать SSD Mobilenet V1 COCO Model (данная модель может быть также использована и для мобильных платформ под управлением Android) [7].

Эта модель достаточно хорошо работает при обнаружении близких объектов, занимающих большое пространство обзора (например, человека, стоящего перед камерой), однако хуже справляется с задачей обнаружения, когда на видео, содержится большое количество людей, каждый из которых занимает небольшое пространство на картинке. При отсутствии GPU ускорении время кадра изменялось в диапазоне 60–80 миллисекунд, что составляет приблизительно 15 кадров в секунду. Более продвинутые модели, дают большую точность обнаружения объектов, однако более требовательны к производительности устройства [11].

Для повышения производительности нашей компактной системы будет использоваться дополнительная библиотека YOLO, которая может работать с различными моделями по обнаружению объектов в реальном времени. Данные модели прошли обучение с двумя различными наборами данных: VOC 2007+2012 и COCO. Благодаря широкому диапазону доступных вариантов выбрана версия, наиболее подходящая для наших требований и возможностей. Наиболее компактный и нетребовательный вариант – Tiny YOLO, который может работать быстро даже на смартфонах или Raspberry Pi [12].



Рис. 2. Распознавание пешеходов с использованием TensorFlow

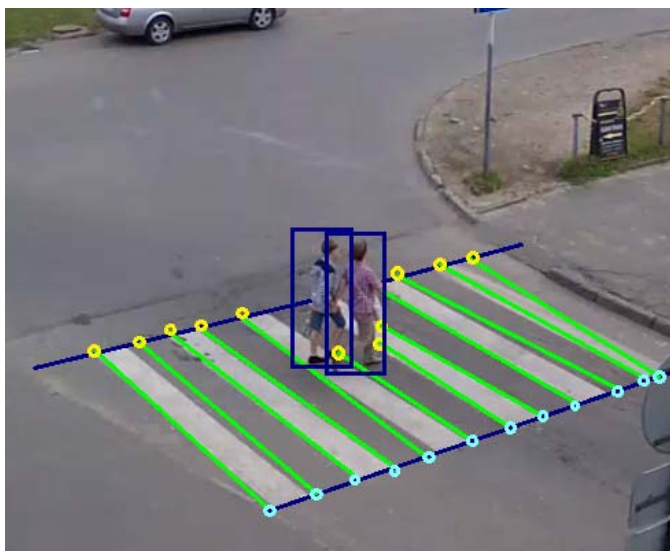


Рис. 3. Распознавание пешеходных переходов с использованием OpenCV

Один из способов решить данную задачу является определение системой, находятся ли пешеходы в заданной области заданное время, если условие выполняется, запускается таймер, визуальнo информирующий водителей транспортных средств о скором переходе пешеходами данной проезжей части.

Одним из преимуществ модели YOLO является, то что системе достаточно «взглянуть» на изображение лишь раз. Данная модель накладывает на изображение сетку, разделяя его на ячейки. Каждая ячейка пытается предсказать координаты зоны обнаружения с оценкой уверенности для этих полей и вероятностью классов. Затем оценка уверенности для каждой зоны обнаружения умножается на вероятность класса, чтобы получить окончательную оценку.

Используемая модель Tensorflow – YOLO может распознать 80 классов. Демонстрационный запуск системы будет захватывать видеопоток с камеры, оценивать его и отображать результаты в простом окне со своими прогнозами. Процесс оценки кадровый и может занимать некоторое время в зависимости от оборудования, на котором он запущен (рис. 2).

После того, как система способна распознавать пешеходов, необходимо усовершенствовать ее для того, чтобы она определяла только тех пешеходов, которые собираются перейти проезжую часть.

В качестве специальной зоны наиболее практично использовать реально существующие пешеходные зоны, обозначенные на тротуаре ярким цветом (как правило былым или желтым), именуемые «зебры».

При создании и обучении модели по определению пешеходных переходов используются те же библиотеки и программные среды. Разница заключается лишь в наборе данных для обучения (см. рис. 3).

После того как система научилась распознавать пешеходный переход (рис. 4), мы можем добавить простую функцию, которая будет определять расстояние от центра пешеходного перехода до заданного объекта, в нашем случае это обнаруженный пешеход, выделенный синим боксом. Если в течение заданного интервала времени данная функция будет возвращать число в пределах одного и того же значения, будем считать, что объект находится у перехода в ожидании перехода проезжей части.

Совместно с имеющимися моделями определения пешехода и пешеходного перехода на изображении, будет использоваться оценка человеческой позы в режиме реального времени. Когда пешеход намеревается перейти дорогу, как правило он останавливается перед «зеброй» в ожидающей позе. Данная поза не сильно меняется из кадра в кадр, и она отличается от поз перемещающегося человека. Данную особенность можно использовать для подтверждения того, что обнаруженный на изображении человек хочет перейти проезжую часть, а не просто шел мимо пешеходного перехода.

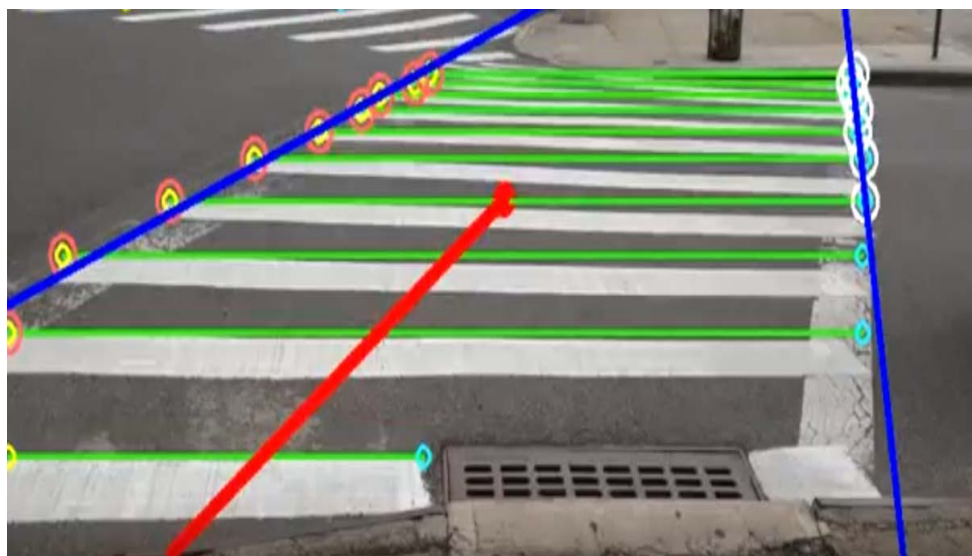


Рис. 4. Распознавание пешеходных переходов с использованием OpenCV

Существующие алгоритмы оценки позы человека относятся к компьютерному зрению и так же представлены в многочисленных библиотеках и моделях. Общей особенностью данных алгоритмов является обнаружение, где находятся суставы основного тела, для нашего проекта этого вполне достаточно. Для осуществления данной возможности используется модель PoseNet совместно с TensorFlow [8] (рис. 5).

Модель PoseNet можно использовать для оценки одной позы или нескольких поз, существует версии алгоритмов, которые могут обнаружить одного или нескольких людей на изображении или видео.

На первом этапе оценки позы изображение подается через предварительно подготовленную модель. PoseNet поставляется с несколькими различными версиями модели, каждая из которых соответствует архитектуре MobileNet v1 с определенным множителем. Чтобы начать работу, модель должна быть загружена с контрольной точки, с указанным коэффициентом.

Коэффициент может иметь значения: 1.01, 1.0, 0.75 или 0.50. По умолчанию – 1.01. Чем больше значение, тем больше размер слоев и более точная модель в ущерб скорости, и напротив, меньшее значение, увеличивает скорость за счет точности определения. Для мобильных устройств рекомендуется модель с коэффициентом 0.50

Алгоритм нескольких поз более сложный и медленный, чем алгоритм одиночной позы, в тоже время, если на изображении появляются несколько человек, их обнаруженные ключевые точки с меньшей вероятностью будут связаны с неправильной позой. Данная модель использует алгоритм быстрого алчного декодирования, который возвращает оценку, представляющую собой список, каждый элемент которого имеет показатель «вероятности» позы и массив ключевых точек, индексированных идентификатором, состоящий из номера и положения [9, 10].

В результате прототип системы основывается на взаимодействии следующих моделей:

- 1) Распознавание пешеходной дорожки и сопряженной с ней участка тротуара, на котором будут располагаться пешеходы;
- 2) Распознавание пешеходов;
- 3) Распознавание поз пешеходов, для дополнительной верификации и отслеживания перемещения.

Таким образом отфильтровываются заведомо ненужные объекты, т. е. все те, кто не находится на пешеходном переходе или рядом с ним. Тем самым система не будет реагировать на каждого пешехода, а лишь на желающего перейти проезжую часть. Другим преимуществом уменьшения сектора обзора, является повышение производительности и точности распознавания.

После распознавания пешехода на пешеходной зоне, система начинает стандартный обратный отсчет, по истечении которого, проезжую часть можно переходить.

Результаты проведенных экспериментов

На текущий момент система состоит из трех моделей:

- 1) Распознавание пешеходов;
- 2) Распознавание пешеходного перехода;



Рис. 5. Распознавание поз пешеходов с использованием модели PoseNet в Tensorflow

3) Распознавание позы пешеходов.

В конечном итоге планируется: объединить данные модели в одну систему, внедрение дополнительной логики для принятия решений, эмуляция сигнализации для проведения тестов. В качестве эмуляции сигнализации светофора может быть использованы простые светодиоды красного и зеленых цветов, расположенные на плате расширения. При распознавании в видеопотоке пешехода(ов) намеревающегося перейти проезжую часть, будет срабатывать таймер, по истечению которого, один из диодов будет становиться красным (сигнал для водителей), второй зеленым (сигнал для пешеходов).

В табл. 1 показаны результаты тестов днем, плотность пешеходного трафика разная.

Таблица 1.

Результат плотности пешеходного трафика

№ теста	Реальное число	Количество подсчитанных	Точность, %
1	3	3	100
2	9	8	88,8
3	18	15	83

В зависимости от плотности пешеходного трафика, результаты тестов не всегда сходятся к единому значению, общий показатель распознавания человеческой фигуры начинает падать при большом количестве людей, очевидно, что, используя более производительное железо, разработанное для конкретной задачи, либо использование графических ускорителей позволит свести к минимуму количество потерь.

В табл. 2 представлены результаты тестов на тех же самых пешеходных участках при низкой освещенности.

Таблица 2.

Результат пешеходный участков при низкой освещенности

№ теста	Реальное число	Количество подсчитанных	Точность, %
1	4	3	75 %
2	7	4	57 %
3	11	6	54 %

Уровень освещенности зависит от времени суток, расположения тротуара и пешеходного перехода в освещаемой зоне. Это в конечном итоге отражается на детализации объектов на изображении. При сильном зашумлении изображения системе становится распознать объект. Возможные варианты решения: использование более точных моделей в условиях плохой освещенности, автоматическая подсветка пешеходного перехода и прилегающей зоны, использование инфракрасных камер.

В результате был проведен эксперимент на платформе микрокомпьютера Raspberry Pi 3, для того чтобы показать, что данные системы возможно разворачивать не только на мощных вычислительных кластерах, но и на конечных

устройствах. Разумеется, скорость обработки изображения при этом существенно падает. Возможные варианты решения – это использование графических процессоров и ускорителей, совместно с программным обеспечением, разработанного для них (*Nvidia Kuda, TensorFlow GPU* и пр.) При этом необходимо повышать качество обнаружения объектов и поз людей, независимо от того где расположена камера обнаружения, устройство должно быть универсальным. Еще одной проблемой которую необходимо решить – это более точное определение пешеходов, которые хотят перейти проезжую часть, от пешеходов, идущих по тротуару рядом с пешеходным переходом. Подобные ложные срабатывания могут привести к заторам и пробкам на дорогах. Возможный вариант решения: распределить работу по обнаружению пешеходов между датчиками присутствия, камерой, которая будет захватывать изображение для обработки, и микрокомпьютером, согласовывающий все действия, производящий вычисления и обработку изображения, а также содержащий программные модули для дальнейшего запуска. Датчики присутствия при этом должны располагаться в непосредственной близости от проезжей части и фиксировать только тех людей, которые попали в зону их действия.

Наибольшее количество ДТП с участием пешеходов и транспортных средств случается в темное время суток, в условиях плохой видимости из-за погодных условий, плохой освещенности зоны пешеходного перехода. Данную проблему можно решить, путем активирования противотуманного и люминесцентного подсвечивания проезжей части дороги, во время пересечения ее пешеходом, после того как сработала система распознавания. В случае, когда ни один из пешеходов не пересекает проезжую часть, данная функция может быть неактивна или находиться в режиме ожидания с низким энергопотреблением, необходимого лишь для нормального движения транспортных средств по проезжей части. Тем самым избегается простой городских энергосистем, и излишняя освещенность на «пустых» участках дороги.

Заключение

Предложена интеллектуальная система по обнаружению людей в пешеходной зоне является улучшение качества обслуживания транспортного и пешеходного трафика, повышение безопасности, автоматизация процессов с возможностью интеграции в другие городские системы, таким образом улучшая городскую среду в целом.

Создан прототип на платформе микрокомпьютера Raspberry Pi 3, для того чтобы показать, что данные системы возможно разворачивать не только на мощных вычислительных кластерах, но и на конечных устройствах.

В перспективе данная система должна выполнять следующие задачи:

- обеспечивать автоматическую подсветку пешеходного перехода;
- повышение безопасности пешехода через его отслеживание вдоль всего перехода (внедрение системы меток объектов);
- повышение безопасности пешехода через отслеживание его положения во время перехода проезжей части;
- связь системы с городской спасательной службой («Безопасный город»).

Литература

1. Кучерявый А. Е., Прокопьев А. В., Кучерявый Е. А. Самоорганизующиеся сети. СПб.: Любавич, 2011. 312 с.
2. Клейнберг Дж., Тардос Е. Алгоритмы: разработка и применение. СПб.: Питер, 2016. 800 с.
3. Minichino, J., Howse, J. Learning OpenCV 3 Computer Vision with Python. Mumbai: Packt Publishing, 2015. 268 p.
4. Navneet, D., Triggs, B. Histograms of Oriented Gradients for Human Detection // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
5. Doyle, L. How to Deal with Networking IoT Devices. URL: <https://www.itworld.com/article/3258993/internet-of-things/how-to-deal-with-networking-iot-devices.html>
6. Kurlekar, S. Embedded vision in IoT. URL: <https://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/Embedded-vision-in-IoT>
7. Rosebrock, A. Deep learning on the Raspberry Pi with OpenCV. URL: <https://www.pyimagesearch.com/2017/10/02/deep-learning-on-the-raspberry-pi-with-opencv/>
8. Papandreou, G., Zhu, T., Chen, L.-C., Gidaris, S., Tompson, J., Murphy, K. PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model // Lecture Notes in Computer Science. 2018. Vol. 11218. Pp. 282–299.
9. Romih, T., Malajner, M., Gleich, D., Planinsic, P. Educational Computer Vision System for Object Detection and Tracking // 50th International Symposium ELMAR, 2008.
10. Curio, C., Edelbrunner, J., Kalinke, T., Tzomakas, C., Von Seelen W. Walking Pedestrian Recognition // 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, 1999.
11. Koucheryavy, A., Muthanna, A., Prokopiev, A. The Mixed Telemetry/Image USN in the Overload Conditions // 16th Conference on Advanced Communication Technology (ICACT). 2014.
12. Гимадинов Р. Ф., Мутханна А. С., Кучерявый А. Е. Кластеризация в мобильных сетях 5G. случай частичной мобильности // Информационные технологии и телекоммуникации. 2015. № 2 (10). С. 44–52.

References

1. Koucheryavy, A., Prokopiev, A., Koucheryavy, Y. Self-Organizing Networks. SPb.: Lyubavich, 2011. 312 p.
2. Kleinberg, J., Tardos, É. Algorithm Design. Pearson, 2005. 864 p.
3. Minichino, J., Howse, J. Learning OpenCV 3 Computer Vision with Python. Mumbai: Packt Publishing, 2015. 268 p.
4. Navneet, D., Triggs, B. Histograms of Oriented Gradients for Human Detection // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
5. Doyle, L. How to Deal with Networking IoT Devices. URL: <https://www.itworld.com/article/3258993/internet-of-things/how-to-deal-with-networking-iot-devices.html>
6. Kurlekar, S. Embedded vision in IoT. URL: <https://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/Embedded-vision-in-IoT>
7. Rosebrock, A. Deep learning on the Raspberry Pi with OpenCV. URL: <https://www.pyimagesearch.com/2017/10/02/deep-learning-on-the-raspberry-pi-with-opencv/>
8. Papandreou, G., Zhu, T., Chen, L.-C., Gidaris, S., Tompson, J., Murphy, K. PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model // Lecture Notes in Computer Science. 2018. Vol. 11218. pp. 282–299.
9. Romih, T., Malajner, M., Gleich, D., Planinsic, P. Educational Computer Vision System for Object Detection and Tracking//50th International Symposium ELMAR, 2008.
10. Curio, C., Edelbrunner, J., Kalinke, T., Tzomakas, C., Von Seelen W. Walking Pedestrian Recognition // 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, 1999.
11. Koucheryavy, A., Muthanna, A., Prokopiev, A. The Mixed Telemetry/Image USN in the Overload Conditions // 16th Conference on Advanced Communication Technology (ICACT). 2014.
12. Gimadinov, R., Muthanna, A., Koucheryavy, A. Clustering in Mobile Network 5G based on Partial Mobility // Telecom IT. 2015. Vol. 2 (10). pp. 44–52. (in Russian).

- Яскеляйнен Яан Сергеевич*** – магистр, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, jsklnnjn@gmail.com
- Хакимов Абдукодир Абдукаримович*** – аспирант, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, abdukadir94@gmail.com
- Мутанна Мохаммед Салех Али*** – аспирант, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, muthanna@mail.ru
- Мутханна Аммар Салех Али*** – кандидат технических наук, ассистент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, ammarexpress@gmail.com
- Киричек Руслан Валентинович*** – доктор технических наук, доцент, СПбГУТ, Санкт-Петербург, 193232, Российская Федерация, kirichek@sut.ru
- Jaskelainen Ian*** – Undergraduate, SPbSUT, St. Petersburg, 193232, Russian Federation, jsklnnjn@gmail.com
- Khakimov Abdukodir*** – Postgraduate student, SPbSUT, St. Petersburg, 193232, Russian Federation, abdukadir94@gmail.com
- Mutanna Mohammed*** – Postgraduate student, SPbSUT, St. Petersburg, 193232, Russian Federation, muthanna@mail.ru
- Muthanna Ammar*** – Candidate of Engineering Sciences, Assistant, SPbSUT, St. Petersburg, 193232, Russian Federation, ammarexpress@gmail.com
- Kirichek Ruslan*** – Doctor of Engineering Sciences, Associate Professor, SPbSUT, St. Petersburg, 193232, Russian Federation, akouch@mail.ru