

# Использование сверточных сетей для решения задачи детектирования и распознавания государственных регистрационных знаков транспортных средств

А.Н. Даниленко<sup>1</sup>, А.С. Гуженко<sup>1</sup>

<sup>1</sup>Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

**Аннотация.** В настоящей работе представлена реализация системы обнаружения и распознавания государственных регистрационных знаков (ГРЗ) транспортных средств на цифровых изображениях. Проведен сравнительный анализ методов детектирования, по результатам которого сделан вывод о построении модуля детектирования на основе свёрточной нейронной сети. На базе нейронной сети GoogleNet разработана собственная архитектура сети. Сделан обзор подходов к распознаванию ГРЗ на цифровых изображениях, определено лучшее решение для модуля распознавания. Разработанная система РГРЗ способна локализовать номера, в которых используются буквы латинского алфавита и арабские цифры. Произведено тестирование системы детектирования и распознавания ГРЗ, показавшее устойчивость к изменению различных факторов, затрудняющих обнаружение и распознавание номера. Точность реализованной системы составила не менее 94%.

## 1. Введение

Автоматическое распознавание государственных регистрационных знаков (РГРЗ) является частой темой исследований [1] и имеет множество практических применений, таких как: автоматический сбор платы за проезд, обеспечение соблюдения правил дорожного движения, контроль доступа на частные территории и мониторинг дорожного движения. Классический процесс РГРЗ обычно состоит из следующих этапов:

- предобработка изображения (сглаживание, бинаризация, выделение контуров);
- поиск и выделение рамки ГРЗ;
- обработка изображения с рамкой ГРЗ (выравнивание прямоугольника с ГРЗ, фильтрация, бинаризация);
- сегментирование изображения;
- классификация символов [16].

## 2. Материалы и методы

### 2.1 Задача детектирования с помощью сверточных нейронных сетей

С 1980-х годов изображения начали обрабатывать нейронными сетями. К тому же времени относится и первый взлет свёрточных нейронных сетей, специальной архитектуры, которая отлично подходит для обработки именно таких входов, как цифровые изображения [2].

В целом, это редкий пример области, в которой нейронные сети никогда полностью не пропадали из виду. Однако после начала революции глубокого обучения прогресс в обработке изображений тоже резко ускорился. В 2009–2010 годах глубокие свёрточные сети выиграли ряд соревнований по распознаванию символов и даже и распознаванию видео с камер слежения. Кроме того, в 2009 году появились первые реализации нейронных сетей на графических процессорах, что дало огромный импульс всем связанным со свёрточными сетями исследованиям. Свёрточные нейронные сети – это весьма широкий класс архитектур, основная идея которых состоит в повторном использовании одних и тех же частей нейронной сети для работы с разными небольшими, локальными участками входного изображения. Стоит отметить, что применения нейронных сетей позволяет лучше решать большинство задач обработки изображений [3].

### *2.2 Свёрточная нейронная сеть YOLO*

YOLO (You Only Look Once) – новый подход к решению задачи детектирования объектов, предложенный в 2015 году Джозефом Редмоном и другими учеными [4]. Данный метод позволяет обрабатывать изображение за одну стадию (проход) начиная от пикселей входного изображения и заканчивая координатами ограничивающих прямоугольников и их вероятностями.

YOLO объединяет отдельные компоненты обнаружения объектов в единую свёрточную сеть, которая одновременно предсказывает координаты ограничивающих прямоугольников и вероятности принадлежности объекта к каждому из известных классов. Используя данный подход, можно обучать нейронную сеть на изображениях любых размеров и напрямую оптимизировать производительность детектирования. Эта унифицированная модель имеет несколько преимуществ по сравнению с традиционными методами обнаружения объектов.

Во-первых, архитектура YOLO обеспечивает работу в режиме реального времени при сохранении высокой средней точности. Во-вторых, YOLO при предсказании анализирует изображение целиком. В отличие от методов, использующих скользящие окна или основанных на локальных предположениях, YOLO использует признаки всего изображения для прогнозирования каждого ограничивающего прямоугольника. Таким образом, подобный подход целиком описывает полный образ изображения, а также все объекты находящиеся на нём.

### *2.3 Структура нейронной сети*

Для построения архитектуры нейронной сети разработчики во многом ориентировались на модель GoogleNet для классификации изображений. Конфигурация YOLO реализована как свёрточная нейронная сеть, где первоначальные свёрточные слои нейронной сети извлекают признаки из входного изображения, а полносвязные слои прогнозируют вероятности и координаты вывода. Сеть имеет 24 свёрточных слоя, за которыми следуют 2 полносвязных слоя. Вместо исходных модулей, которые используются в GoogleNet, используются каскады свёрточных и прореживающих слоёв. Выходной слой нейронной сети представляет собой тензор размерностью  $7 \times 7 \times 30$  [4]. Учитывая преимущества нейронных сетей в задаче распознавания образов, было принято решение использовать свёрточные нейронные сети в данной работе для решения задачи детектирования и распознавания номерных знаков транспортных средств.

### *2.4 Предлагаемый метод детектирования и распознавания ГРЗ*

Обучение нейронной сети для одновременного решения задачи детектирования и распознавания ГРЗ требует огромного количества размеченных данных – то есть большой обучающей выборки, в которой будут размечены не только области нахождения данных объектов, но и символы, которые должны достаточно часто повторяться. Из-за отсутствия подходящей обучающей выборки для решения задачи распознавания ГРЗ были обучены две нейронные сети на имеющихся наборах данных. Предлагаемый метод значительно упростит процесс РГРЗ, сведя его к следующим этапам:

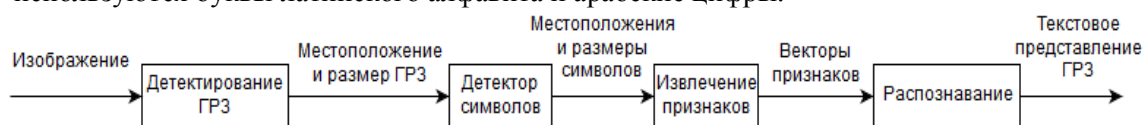
1. детектирование и вырезание рамки ГРЗ;
2. детектирование символов;
3. сортировка символов.

Для разработки системы РГРЗ необходимо спроектировать и обучить следующие модели машинного обучения:

1. детектор ГРЗ, принимающий на вход изображение ТС и возвращающий координаты обрамляющего прямоугольника;
2. детектор символов, принимающий на вход изображение прямоугольника ГРЗ и возвращающий координаты обрамляющих прямоугольников для символов, оставляющих ГРЗ.

Предлагаемая схема распознавания ГРЗ представлена на рисунке 1.

Разрабатываемая система РГРЗ должна быть способна локализовать номера, в которых используются буквы латинского алфавита и арабские цифры.



**Рисунок 1.** Процесс распознавания ГРЗ на изображении.

### 2.5 Выбор и описание архитектуры используемой нейронной сети

Для выбора подходящей архитектуры СНС для детектирования ГРЗ был проведен анализ наиболее распространенных архитектур НС. В таблице 1 приведен сравнительный анализ нескольких архитектур классификаторов по следующим параметрам:

- точность классификации на наборе данных ImageNet (метрика Top-1 accuracy);
- количество параметров НС, миллионов (метрика Params);
- количество операций сложения и умножения, миллионов (метрика MAdds);
- время классификации цветного изображения размером 224\*224 пиксела на одном ядре процессора Qualcomm Snapdragon 821 MSM 8996;
- Pro в миллисекундах (метрика CPU).

**Таблица 1.** Сравнение архитектур классификаторов.

Архитектура НС	Top-1 accuracy	Params	MAdds	CPU
MobileNetV1	70.6	4.2	575	113
ShuffleNet(1.5)	71.5	3.4	292	-
ShuffleNet(x2)	73.4	5.4	524	-
NasNet-A	74.0	5.3	564	183
MobileNetV2	72.0	3.4	300	75
MobileNetV2(1.4)	74.7	6.9	585	143

В таблице 2 приведен сравнительный анализ нескольких архитектур детекторов по следующим параметрам:

1. Средняя точность на наборе данных COCO (метрика mAP).
2. Количество параметров НС, миллионов (параметр Params).
3. Количество операция сложения и умножения, миллиардов (параметр MAdds).
4. Время обработки одного цветного изображения из набора данных COCO на на одном ядре процессора Qualcomm Snapdragon 821 MSM 8996.
5. Pro в миллисекундах (метрика CPU).

На основе анализа показателей точности и скорости работы наиболее распространенных архитектур было принято решение использовать архитектуру MobileNetV2+SSDLite для решения задачи детектирования ГРЗ. Основные преимущества данной архитектуры:

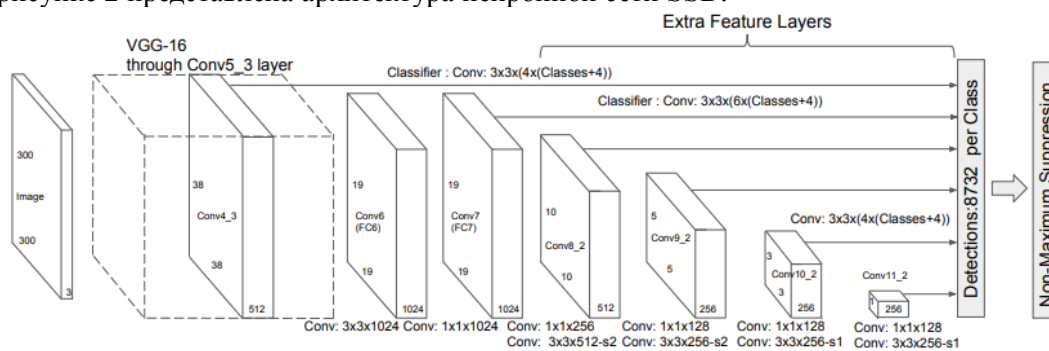
1. в качестве классификатора используется MobileNetV2, позволяющая при относительно небольшом количестве параметров получить высокую точность классификации;
2. комбинация MobileNetV2 с детектором SSDLite представляет собой самое быстрое на текущий момент решение для детектирования объектов на изображении.

**Таблица 2.** Сравнение архитектур детекторов.

Архитектура HC	mAP	Params	MAdds	CPU
SSD300	23.2	36.1	35.2	>1000
SSD512	26.8	36.1	99.5	>1000
YOLOv2	21.6	50.7	17.5	>1000
MobileNetV1+SSDLite	22.2	5.1	1.3	270
MobileNetV2+SSDLite	22.1	4.3	0.8	200

SSDLite-архитектура для задачи детектирования объекта, использующая MobileNetV2 в свёрточной части, превосходит известный детектор реального времени YOLOv2 по точности на датасете MS COCO, при этом показывая в 20 раз большую скорость и в 10 раз меньший размер [5].

На рисунке 2 представлена архитектура нейронной сети SSD.



**Рисунок 2.** SSD-архитектура.

На вход системе поступает изображение любого размера, которое приводится к размеру 300 на 300 пикселей и подается на вход нейронной сети. Начальные свёрточные слои нейронной сети извлекают признаки из изображения, которые затем обрабатываются процедурой детектирования.

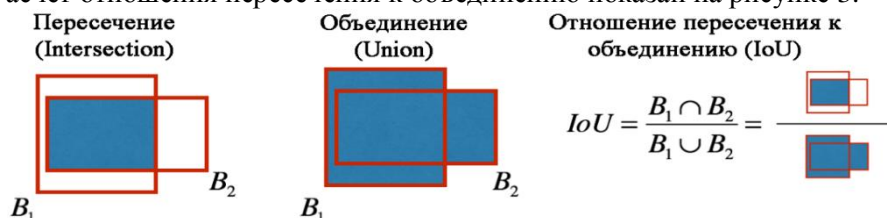
Многие свёрточные сети предсказывают координаты ограничивающих прямоугольников, используя полносвязные слои на основе извлечения признаков с помощью свёрточных слоёв. Вместо предсказания абсолютных значений координат и размеров ограничивающих прямоугольников, построенная нейронная сеть, используя только свёрточные слои, предсказывает смещение и степень доверия для опорных областей (anchor box).

Поскольку слой предсказания является свёрточным, смещения предсказываются в каждом месте на карте признаков. Использование смещений вместо координат облегчает обучение нейронной сети. Аналогичный принцип используется в модуле локальных предсказаний Faster R-CNN.

Использование опорных областей немного уменьшает точность нейронной сети, но значительно увеличивает показатель полноты, что позволяет увеличить результат работы системы детектирования в целом [6]. Для нахождения опорных областей использовался метод К-средних на обучающем наборе данных. Количество опорных областей равно 4 выбрано как компромисс между показателем полноты и сложностью модели.

Выходной слой нейронной сети объединяет в себе как признаки с высоким разрешением, так и признаки с низким. Аналогичный подход используется в архитектуре нейронной сети ResNet [7]. Таким образом, работа с расширенной картой признаков позволяет детектировать как крупные, так и относительно небольшие изображения объектов. Данный подход позволяет

увеличить качество детектора. Выходом нейронной сети является карта признаков размером  $N \times N \times 24$ . Каждая ячейка содержит в себе 6 прогнозируемых параметров для каждой из 4 опорных областей. Параметр  $p$  показывает значение достоверности, которое представляет собой отношение пересечения к объединению между предсказанной областью и истинным значением. Расчёт отношения пересечения к объединению показан на рисунке 3.

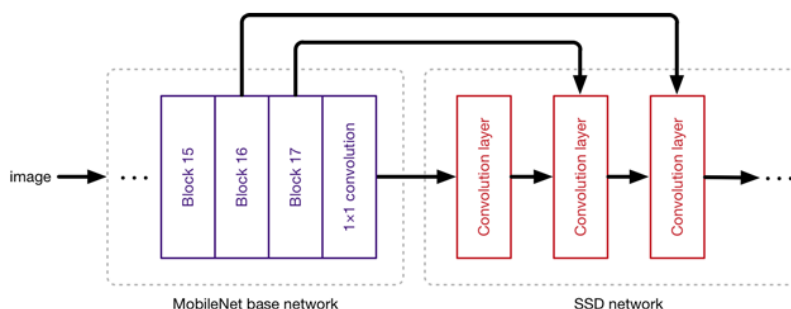


**Рисунок 3.** Вычисление отношения пересечения к объединению.

Также для каждой опорной области предсказывается вероятность её принадлежности к одному из классов, в нашем случае это либо класс «ГРЗ» (на этапе детектирования ГРЗ)/«символ» (на этапе распознавания текста), либо класс «фон».

Одной из проблем, возникающих при детектировании объектов, является множественное обнаружение одного и того же объекта или же ложные срабатывания детектора. Для того, чтобы отфильтровать лишние обнаружения и оставить только область, наиболее точно описывающую данный объект, применяется алгоритм подавления не максимумов (Non-Maximum Suppression). Алгоритм подавления не максимумов отбрасывает объекты, вероятность появления которых ниже некоторого заданного порогового значения. Значение порога может выбираться в зависимости от качества работы нейронной сети, процедуры обучения, и различных факторов характерных для входных изображений. Это позволяет адаптировать подсистему детектирования под различные условия эксплуатации. Далее алгоритм подавления не максимумов выбирает наиболее вероятное детектирование в определенной области и фильтрует другие детектирования того же класса, сильно пересекающиеся с выбранным.

В данной работе рассматривается архитектура SSDLite, использующая в сверточных слоях MobileNetV2 в качестве извлекателя признаков, схема работы которой представлена на рисунке 4.



**Рисунок 4.** Схема архитектуры SSDLite + MobileNetV2.

В предложенной архитектуре на вход каждого сверточного слоя (Convolution layer) нейронной сети SSDLite помимо выходного значения последнего слоя базовой сети MobileNet подаются также результаты нескольких предыдущих слоев. Работа слоев MobileNet в данном случае заключается в преобразовании пикселей входного изображения в признаки, которые описывают содержание изображения для последующего детектирования. SSD разработан независимо от базовой сети, поэтому он может работать практически со всем, включая MobileNet. Более того, в MobileNet + SSD используется вариант, называемый архитектуры SSDLite, который использует разделенные по глубине слои вместо обычных сверток для обнаружения объектов на изображении. Использование SSDLite поверх MobileNet2 позволяет

легко получать результаты распознавания в реальном времени даже на мобильном устройстве (от 30 FPS – количество кадров, обрабатываемых в секунду).

## 2.6 Обучение детектора ГРЗ

В данной работе обучение детектора ГРЗ было проведено в 2 этапа с применением методики обучения с переносом (transfer learning):

1. предварительное обучение на наборе данных общего назначения;
2. дообучение на наборе данных, содержащем только изображения ГРЗ.

Оба этапа обучения проводились на виртуальной машине Google Compute Engine.

Первый этап обучения проводился с использованием набора данных COCO [8]. Основная цель этого этапа – научить НС выделять низкоуровневые признаки изображения, такие как линии, углы, свойства текстур, простейшие геометрические фигуры. Для обучения использовался фреймворк глубокого обучения Tensorflow, процесс обучения 48 часов на указанной выше виртуальной машине. По результатам этого обучения модель имела среднюю точность (mAP), равную 20.9.

Поскольку набор данных COCO содержит объекты, принадлежащие 91 различному классу, выходной слой предобученной НС содержит 92 нейрона (91 нейрон соответствует выходным классам, и 1 нейрон соответствует классу «фон»). Однако детектор ГРЗ должен локализовывать только объекты, принадлежащие классу «ГРЗ». Для этого из НС, полученной после первого этапа, был удален выходной слой, и вместо него добавлен новый слой, содержащий только 2 нейрона (1 для класса «ГРЗ» и 1 для фона). Согласно методике transfer learning, остальные веса НС не фиксируются, что позволит скорректировать их на следующем этапе для корректного детектирования ГРЗ.

Второй этап обучения проводился с использованием следующих наборов данных:

- UFPR-ALPR, содержащий 4500 цветных изображений ТС с бразильскими ГРЗ размером 1920\*1080 пикселей, полученных с использованием камер GoPro Hero4 Silver, Huawei P9 Lite и iPhone 7 Plus [9]. На каждую из камер было снято 1500 изображений, 900 из которых – автомобили с серыми ГРЗ, 300 – автомобили с красными ГРЗ, 300 – мотоциклы с серыми ГРЗ [9, 10].
- European License Plates, содержащий 10000 фотографий ТС стран Евросоюза разного размера [6]. Типы ГРЗ также варьируются.
- База данных сайта avto-nomer.ru, содержащая 10000 фотографий российских ТС и 5500 фотографий ТС стран СНГ [11].
- Таким образом, итоговый набор данных состоит из 30000 фотографий ТС, принадлежащих разным странам. Этот набор был случайным разбит на 3 части следующим образом:
  - 6000 изображений (20% от общего количества) составляют тестовый набор, используемый для итогового тестирования детектора;
  - 3000 изображений (10%) составляют валидационный набор, который используется для проверки точности детектора в процессе обучения и принятия решения о корректировке параметров НС;
  - 21000 изображений (70%) используется непосредственно для обучения НС.

В качестве целевой метрики использовалось Intersection over Union (IoU):

$$IoU = \frac{area(Rdet \cap Rgt)}{area(Rdet \cup Rgt)},$$

где Rdet – предсказанный обрамляющий прямоугольник, Rgt – истинный обрамляющий прямоугольник, area – функция вычисления площади [12].

Процесс обучения был запущен на той же виртуальной машине, изначальное количество итераций обучения (эпох) равно 1000, был применен критерий ранней остановки процесса обучения при отсутствии роста целевой метрики на протяжении 10 эпох. Обучение было остановлено данным критерием на 47 эпохе, величина IoU на валидационном наборе на этот момент равна 0.93. Обучение детектора ГРЗ заняло 19 часов.

## 2.7 Обучение детектора символов

Архитектура MobileNetV2+SSDLite хорошо себя показала в задаче детектирования ГРЗ, поэтому было решено использовать ее в задаче детектирования символов. Поскольку данный детектор должен локализовывать изображения с достаточно простыми признаками (символы и цифры), обучение было проведено в один этап, без применения transfer learning.

Обучение было произведено на том же наборе данных, что и второй этап обучения детектора ГРЗ (30000 изображений ТС разных стран с разными типами ГРЗ). Из каждого изображения ТС был вырезан обрамляющий прямоугольник ГРЗ.

Поскольку детектор символов работает с большим количеством классов (26 символов латинского алфавита, 10 цифр, фон, всего 37 классов), чем детектор ГРЗ, было решено разбить исходный набор данных в следующих пропорциях:

- 6000 изображений (20%) составляют тестовый набор;
- 6000 изображений (20%) составляют валидационный набор;
- 18000 изображений (60%) составляют тренировочный набор.

В качестве целевой метрики используется средняя точность (mAP).

Процесс обучения был запущен на той же виртуальной машине, изначально количество эпох равно 100, применен критерий ранней остановки обучения при отсутствии роста целевой метрики в течение 5 эпох. Обучение было остановлено данным критерием на 18 эпохе, величина mAP на этот момент была равна 0.54. Время обучения составило 5 часов.

## 3. Результаты и обсуждение

Было разработано клиент-серверное приложение, экранная форма которого представлена на рисунке 7.

Для тестирования системы распознавания ГРЗ использовалось 6000 изображений, отложенных перед этапом обучения детектора ГРЗ.

В данном наборе содержатся ГРЗ следующих стран:

- Россия, всего 2000 изображений разных типов;
- Евросоюз, 2000 изображений разных стран;
- СНГ, 1000 изображений разных стран;
- Бразилия, 1000 изображений.

ID записи	Изображение	ГРЗ	Точность	Дата и время	IP камеры
a4712e61-88af-4337-acb8-67b1a37bbc2f		939_PH 61	96%	23.04.2019 19:30:08	192.168.99.21
86d40e66-6df4-4275-b2e1-2f7e95bdcf8f		5561_BO-7	87%	23.04.2019 19:25:31	192.168.99.21
3db3e482-84a1-11e7-8c9e-379d7074c978		777HHH 01	94%	23.04.2019 17:20:35	192.168.99.21
1b1be037-846e-11e7-8c9e-9795137ce39e		AA0020 XC	89%	23.04.2019 16:23:02	192.168.99.21
074e9007-9804-43cc-915f-ac0c4509266d		y602AB 777	95%	23.04.2019 15:24:07	192.168.99.21

Рисунок 7. Экранная форма разработанного приложения.

Кроме того, в тестовый набор было добавлено 1000 номеров США для проверки обобщаемости нейронных сетей на новые типы.

Для тестирования детектора ГРЗ использовались следующие метрики:

- IoU;
- полнота (Recall) – метрика, показывающая долю ГРЗ, найденных в общем наборе данных;



- среднее время детектирования (DT), в миллисекундах.
  - Для тестирования детектора символов использовались следующие метрики:
    - средняя точность детектирования (mAP);
    - полнота (Recall2);
    - среднее время детектирования (DT2) в миллисекундах.
  - Для тестирования всей системы ГРЗ использовались следующие метрики:
    - доля номеров, в которых корректно распознано более 5 символов (5C);
    - доля номеров, распознанных полностью (AC).
  - Для тестирования использовался экземпляр виртуальной машины Google Compute Engine со следующими характеристиками:
    - 1 процессор Intel Xeon Silver 4116;
    - 16 Гб RAM;
    - 100 Гб дискового пространства на SSD.
- Результаты тестирования приведены в таблице 3.

**Таблица 3.** Сравнение архитектур классификаторов.

Набор данных	Количество изображений	IoU	Recall	DT	mAP	Recall2	DT2	5C	AC
РФ	2000	0.95	1	85	0.52	0.99	103	0.99	0.98
ЕС	2000	0.97	1	86	0.49	0.98	110	0.99	0.95
СНГ	1000	0.95	1	85	0.51	0.99	105	0.99	0.94
Бразилия	1000	0.94	1	85	0.52	0.99	106	0.99	0.99
США	1000	0.91	0.99	92	0.48	0.91	119	0.97	0.86

#### 4. Заключение

В работе были рассмотрены основные этапы построения системы распознавания ГРЗ на цифровых изображениях. Был проведен сравнительный анализ подходов к построению систем распознавания номерных знаков. Проанализированы достоинства и недостатки различных алгоритмов детектирования и выравнивания, а также подходы к построению признакового пространства. В ходе анализа была выбрана наиболее подходящая архитектура нейронной сети, а также оптимальный метод обучения для разных классов задач.

Осуществлена разработка подсистемы детектирования и распознавания ГРЗ на цифровых изображениях, а также реализовано их взаимодействие в рамках единой системы. Для обучения и тестирования системы были проанализированы и отобраны наборы данных, максимально приближенные к условиям реального мира.

В ходе работы также был реализован web-интерфейс для работы с результатами распознавания и последующей интеграции с другими системами.

Разработанная система позволяет решать все основные классы задач, такие как детектирование и классификация, согласно требованиям, предъявляемым на практике.

Проведено успешное тестирование системы детектирования и распознавания ГРЗ, показавшее устойчивость к изменению различных факторов, затрудняющих обнаружение и распознавание номера. Точность реализованной системы составила не менее 94%, что является отличным результатом. Таким образом, разработанная в ходе выполнения данной работы система распознавания ГРЗ на цифровых изображениях может применяться для решения задач на практике.

#### 5. Литературf

- [1] T.-Y. Lin. Microsoft COCO: Common Objects in Context / T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. Lawrence Zitnick, P. Dollár // Системные требования: Adobe Acrobat Reader DC, arXiv: 1405.0312, 2015 (31.03.2019).



- [2] Логическая модель базы данных [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F\\_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C\\_%D0%B1%D0%B0%D0%B7%D1%8B\\_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) (22.03.2019).
- [3] Sandler, M. MobileNetV2: Inverted Residuals and Linear Bottlenecks / M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.C. Chen // Системные требования: Adobe Acrobat Reader DC, arXiv:1801.04381, 2018 (23.03.2019).
- [4] Li, B. Component-based license plate detection using conditional random field model / B. Li, B. Tian, Y. Li, D. Wen // IEEE Trans. Intell. Transp. Syst. – 2013. – Vol. 14(4). – P. 1690-1699.
- [5] Yu, S. A novel license plate location method based on wavelet transform and EMD analysis / S. Yu, B. Li, Q. Zhang, C. Liu, M. Meng // Pattern Recogn. – 2015. – Vol. 48(1). – P. 114125.
- [6] Chang, S. Automatic license plate recognition / S. Chang, L. Chen, Y. Chung, S. Chen // IEEE Trans. Intell. Transp. Syst. – 2004. – Vol. 5(1). – P. 4253.
- [7] Du, S. Automatic license plate recognition (alpr): A state-of-the-art review / S. Du, M. Ibrahim, M. Shehata, W. Badawy // IEEE Trans. Circuits Syst. Video Technol. – 2013. – Vol. 23(2). – P. 311-325.
- [8] Giannoukos, I. Operator context scanning to support high segmentation rates for real time license plate recognition / I. Giannoukos, C.-N. Anagnostopoulos, V. Loumos, E. Kayafas // Pattern Recogn. – 2010. – Vol. 43(11). – P. 38663878.
- [9] Zhou, W. Principal visual word discovery for automatic license plate detection / W. Zhou, H. Li, Y. Lu, Q. Tian // IEEE Trans. Image Process. – 2012. – Vol. 21(9). – P. 4269-4279.
- [10] Anagnostopoulos, C. A license plate-recognition algorithm for intelligent transportation system applications / C. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, E. Kayafas // IEEE Trans. Intell. Transp. Syst. – 2006. – Vol. 7(3). – P. 377-392.
- [11] Yuan, Y. A robust and efficient approach to license plate detection / Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu, N. Komodakis // IEEE Trans. Image Process. – 2017. – Vol. 26(3). – P. 1102-1114.
- [12] Hsu, G. Application-oriented license plate recognition / G. Hsu, J. Chen, Y. Chung // IEEE Trans. Veh. Technol. – 2013. – Vol. 62(2). – P. 552-561.

## Use of convolution networks to solve the problem of detection and recognition of state registration signs of vehicles

A.N. Danilenko<sup>1</sup>, A.S. Guzhenko<sup>1</sup>

<sup>1</sup>Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

**Annotation.** This paper presents the implementation of a system for the detection and recognition of state registration plates (RGRZ) of vehicles in digital images. A comparative analysis of the detection methods was carried out. It led to the selection of the convolutional neural network for the detection module. The customized network architecture was developed on the basis of the neural network GoogleNet. An overview of approaches for RGRZ in digital images was made, the best solution for the recognition module was determined. The developed RGRZ system can identify numbers that use letters of the Latin alphabet and Arabic digits. Testing of the detection and recognition system of the vehicle license plates were carried out. It proved the reliability of the system and its ability to perform well, even in conditions that might prevent the detection and recognition of license plates. The accuracy of the implemented system was at least 94%.