

УДК 159.938.25 + 004.05

## УЛУЧШЕНИЕ НЕЙРОСЕТЕВОГО ДЕТЕКТОРА ОТРЕЗКОВ ПУТЕМ ДОБАВЛЕНИЯ ИНЖЕНЕРНЫХ ПРИЗНАКОВ

© 2021 г. Л. А. Ерлыгин<sup>1,2,\*</sup>, Л. М. Тепляков<sup>2</sup><sup>1</sup> Московский физико-технический институт (государственный университет),  
141701 Долгопрудный, Институтский пер., д. 9, Московская обл., Россия<sup>2</sup> Институт проблем передачи информации им. А.А. Харкевича Российской академии наук,  
127051 Москва, Большой Каретный переулок, д. 19, Россия

\*E-mail: erlygin.la@phystech.edu

Поступила в редакцию 28.09.2020 г.

После доработки 16.10.2020 г.

Принята к публикации 02.11.2020 г.

В данной работе рассматривается задача детектирования прямолинейных контуров (отрезков). В недавних исследованиях этой задачи было показано, что нейросетевые методы способны превосходить классические алгоритмы по качеству, однако высокая вычислительная сложность нейросетей стоит на пути внедрения их в приложения на замену быстрым классическим алгоритмам. В данной работе предлагается способ объединения нейросетевого и классического подходов к задаче для компенсации их недостатков путем подачи на вход нейросети не только изображения, но и результата работы классического алгоритма и комбинации предсказаний на выходе. Эта модификация “упрощает” задачу, стоящую перед нейросетью, и позволяет добиться высокого качества с использованием простой быстрой нейросети. Предложенный метод позволяет добиться 0.72 и 0.66 F1 на наборах Wireframe и York соответственно при скорости 14 FPS на CPU.

**Ключевые слова:** искусственные нейронные сети, детектирование отрезков, детектирование контуров, Unet, Sanny

**DOI:** 10.31857/S0235009221010042

### ВВЕДЕНИЕ

Детектирование отрезков – важный шаг во многих системах компьютерного зрения. При помощи детектирования отрезков решаются задачи позиционирования роботов (Bista et al., 2016), распознавания дорожной разметки (Huang et al., 2020) и документов (Zhukovsky et al., 2017).

В литературе встречаются несколько формулировок задачи детектирования отрезков, различающиеся по способу представления результатов детектирования – набора отрезков на изображении. Встречаются представления в виде: множества пар вершин (Grompone et al., 2012); графа вершин (Huang et al., 2018), в котором ребра соответствуют отрезкам, на одной вершине может быть инцидентно несколько ребер; классификации каждого пикселя изображения на принадлежность отрезку (Xue et al., 2019). В нашей работе мы используем последний способ представления отрезков.

Традиционно для решения задачи детектирования отрезков используются так называемые инженерные алгоритмы (Duda, Hart, 1972; Grompone et al., 2012). Они реализуют некий формальный критерий наличия отрезка на изображении, сформулированный, как правило, на основе градиента

изображения (Grompone et al., 2012) или образа изображения в пространстве Хафа (Duda, Hart, 1972).

Сегодня для детектирования отрезков набирают популярность нейросетевые методы (Lin et al., 2020; Xue et al., 2019). В отличие от рассмотренных инженерных алгоритмов, в нейросетевых подходах не производится формализация понятия “отрезок”, а детектирующий алгоритм строится путем обучения нейросети на большом количестве примеров.

Существенным недостатком нейросетевых методов, несмотря на их высокое качество, является низкая скорость работы. Так, в недавней работе (Xue et al., 2019) скорость нейросети составляет 6–10 FPS на дорогостоящем GPU, что, к примеру, соответствует обработке одного кадра на несколько метров движения беспилотного автомобиля.

В данной работе предлагается простой способ объединения нейросетевого и классического подходов к задаче для компенсации недостатков нейросети, путем подачи на вход небольшой нейросети не только изображения, но и результата работы классического алгоритма, что позволяет “упростить” задачу, стоящую перед нейросетью; и комбинации предсказаний на выходе для увеличения точности предсказания. Предложенный метод позволяет добиться 0.72 и 0.66 F1 на набо-

рах Wireframe и York (Huang et al., 2018; Coughlan, Yuille, 2003) соответственно при 14 FPS на CPU и 63 FPS на GPU.

Существующие подходы к детектированию отрезков можно разделить на две категории – инженерные и нейросетевые. В инженерных подходах определяется формальное понятие “отрезок” и на изображении ищутся объекты, удовлетворяющие этому определению; тогда как в нейросетевых подходах формализация понятия “отрезок” не производится, а детектирующий алгоритм строится путем обучения нейросети на наборе примеров.

### *Инженерные методы*

Среди инженерных методов можно выделить алгоритмы, основанные на преобразовании Хафа (Duda, Hart, 1972; Kiryati et al., 1991; Bandera et al., 2006). Как правило, такие алгоритмы состоят из следующих шагов: выделение контуров при помощи, например, алгоритма Canny (Canny, 1986), после которого пиксели отрезков обнаружены с высокой полнотой, но, возможно, низкой точностью; детектирование прямых на контурном изображении при помощи преобразования Хафа; разделение задетектированных прямых на отрезки с использованием контурного изображения.

Другая группа инженерных методов основана на так называемых “областях определения отрезка”, LSR (Line Support Regions) (Burns et al., 1986; Grompone et al., 2012). LSR – это область изображения с похожими ориентациями градиентов пикселей, которая потенциально содержит отрезок. Методы из этой группы выделяют множество LSR на изображении, а затем – определяют, в каких из них содержится отрезок. В работе (Burns et al., 1986) предполагается, что отрезку соответствует “скачок” в интенсивности пикселей. Для проверки, содержит ли LSR искомый “скачок”, пиксели LSR представляются как точки в трехмерном пространстве (две координаты и интенсивность) и аппроксимируются плоскостью. На основании полученных параметров аппроксимации делается вывод о наличии отрезка в данной LSR. В более новом методе (Grompone et al., 2012) вокруг каждой LSR описывается прямоугольник, а далее оценивается правдоподобие присутствия отрезка в прямоугольнике. Для этого подсчитывается доля пикселей внутри прямоугольника, для которых ориентация градиентов совпадает с ориентацией прямоугольника. Затем оценивается вероятность пронаблюдать такие доли на случайном изображении шума.

Рассмотренный алгоритм (Grompone et al., 2012) является одним из самых популярных детекторов отрезков – именно с ним производится сравнение большинства нейросетевых методов

(Huang et al., 2018), которые рассматриваются в следующем разделе.

### *Нейросетевые методы*

В последние годы для решения многих задач компьютерного зрения, например, классических задач классификации и детектирования объектов (в виде описанных прямоугольников) применяются нейронные сети (Krizhevsky et al., 2012; Ronneberger et al., 2015). Задача детектирования отрезков не стала исключением (Huang et al., 2018; Yichao et al., 2019; Xue et al., 2019).

В работе (Huang et al., 2018) задача детектирования отрезков декомпозируется на две подзадачи: поиск концов отрезков и попиксельная классификация пикселей на принадлежность отрезку. Для решения этих подзадач используются две независимые сети. Затем предсказания отдельных нейросетей объединяются для построения отрезков в виде пар точек. Для этого подсчитывается количество положительно классифицированных пикселей вдоль прямой, соединяющей пары задетектированных концов и, если таких пикселей достаточно много, считается, что между концами есть отрезок.

Схожий метод представлен в работе (Yichao et al., 2019). В этом подходе нейросеть, как и в (Huang et al., 2018), детектирует точки пересечения отрезков. После чего на основе точек пересечения предлагаются гипотезы отрезков и классифицируются нейросетью, в отличие от (Huang et al., 2018), где для верификации гипотез используется эвристика.

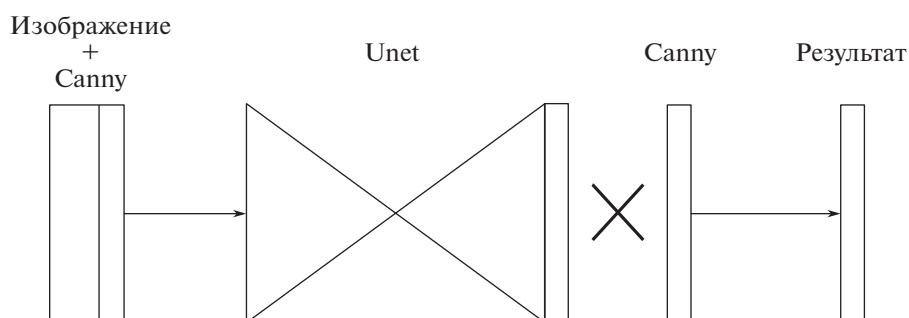
Другой подход к задаче детектирования отрезков использует тот факт, что задачу поиска отрезков можно свести к задаче сегментации изображения (Xue et al., 2019), где отрезкам соответствуют границы сегментов.

## МЕТОД

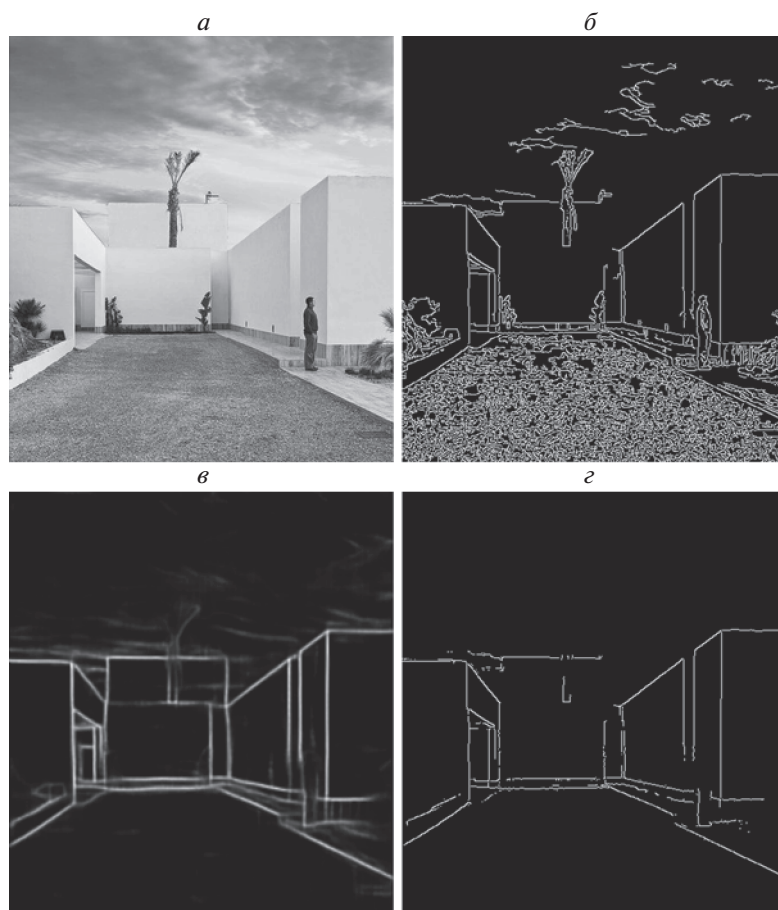
На рис. 1 изображена структура алгоритма, а на рис. 2 проиллюстрирован пример его работы. Помимо ставшей классической для подобных задач (Xue et al., 2019) нейросети архитектуры U-net (Ronneberger et al., 2015), предлагаются две следующие модификации.

Результат работы детектора Canny подается на вход нейросети вместе с изображением. Мы полагаем, что это позволяет упростить задачу, стоящую перед нейросетью, а значит, использовать меньше вычислений при эксплуатации и меньше данных при обучении. Границы, детектируемые Canny (рис. 2,б), являются полезными признаками для задачи детектирования отрезков, а значит их предварительное извлечение позволяет “сэкономить” вычисления внутри нейросети.

Результаты работы Canny (рис. 2,б) и нейросети (в) перемножаются для получения финального



**Рис. 1.** Иллюстрация предложенного метода. При помощи детектора Canny детектируются всевозможные контуры, затем четырехканальное изображение (RGB + Canny) подается нейросети, которая детектирует отрезки. Наконец, результат работы нейросети и Canny перемножаются для точной локализации отрезков.



**Рис. 2.** Пример работы предложенного метода. *a* – изображение; *б* – результат работы детектора Canny; *в* – результат работы нейросети; *г* – объединение результатов работы детектора Canny и нейросети.

результата (*г*). Детектор Canny детектирует всевозможные контуры с пиксельной точностью локализации; нейросеть детектирует искомые отрезки, однако точность локализации невысока – ширина детекций нейросети (*в*) составляет несколько пикселей. Их объединение позволяет получить точно локализованные контуры (преимущество Canny), которые при этом соответствуют искомому типу контуров – отрезкам (преимущество нейросети).

## ЭКСПЕРИМЕНТЫ

### *Архитектура и параметры обучения*

Мы использовали простую нейросеть типа U-Net (Ronneberger, et al., 2015) с 2 max-pooling слоями и с 16 сверточными фильтрами на первом сверточном слое, количество которых удваивается после каждого max-pooling слоя. Размер входного изображения  $256 \times 256$ .

Минимизируемая функция потерь – бинарная кросс-энтропия (Huang, et al., 2018). Оптимиза-

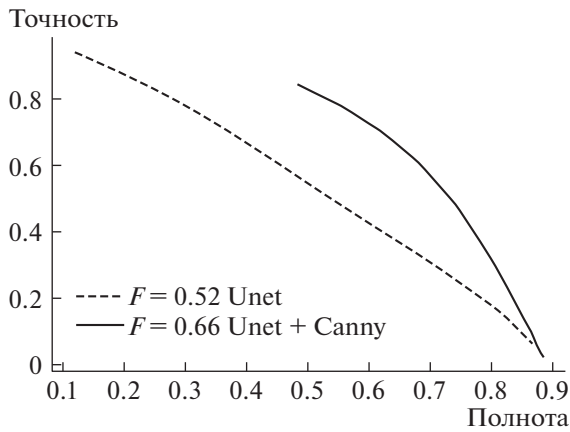


Рис. 3. Кривая точность – полнота для предложенного метода на наборе York и сравнение с обычным Unet.

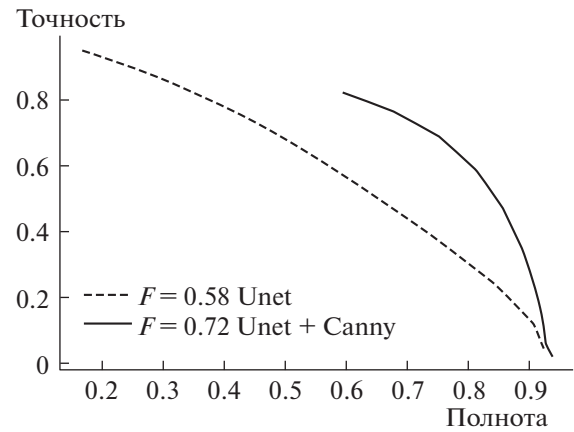


Рис. 4. Кривая точность – полнота для предложенного метода на наборе Wireframe и сравнение с обычным Unet.

цию проводили при помощи алгоритма Adam (Kingma, Ba, 2014).

Обучение производили с размером батча 32. Оно останавливалось, если в течение 30 эпох не наблюдалось уменьшения значения функции потерь на валидационном наборе. Скорость обучения уменьшалась в 2 раза, если значение функции потерь на валидационном наборе не уменьшалось в течение 15 эпох. Из-за стохастической природы обучения приведенные далее количественные результаты являются медианным значением по пяти повторениям обучения нейросети.

#### Наборы данных и оценка качества

Для анализа предложенного метода использовались популярные для задачи (Yichao et al., 2019; Xue et al., 2019) наборы Wireframe (Huang, et al., 2018) (обучение и тестирование) и York (Coughlan, Yuille, 2003) (только тестирование).

Качество оценивалось при помощи кривой точности – полноты (precision – recall). Для построения кривой истинные пиксели (через которые проходит некий отмеченный на изображении отрезок) сопоставлялись с задетектированными пикселями (для которых предсказание нейросети выше некоего порога), причем сопоставление

считалось успешным, если расстояние между пикселями не превосходило 1% от диагонали изображения (Huang et al., 2018). Это сопоставление позволяет вычислить точность и полноту. Такая метрика является стандартной для задачи детектирования отрезков (Yichao et al., 2019; Xue et al., 2020; Xue et al., 2019).

Для построения скалярной оценки качества работы алгоритма по кривой точности – полноты вычислялась F1 оценка для каждой точки кривой, и затем бралось максимальное значение F1.

## РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Численный анализ предложенного метода приведен на рис. 3 и 4. Предложенные модификации существенно повышают качество детектирования по сравнению с Unet без модификаций.

Сравнение с другими методами и скорость работы представлены в табл. 1. Численные результаты методов цитируются по работам (Grompone et al., 2012; Xue et al., 2020; Yichao et al., 2019; Huang et al., 2018). Объективности сравнения могут помешать различия в программном и аппаратном обеспечении, использованном для замеров. Мы полагаем, что связанные с этим различия незначительны.

Таблица 1. Сравнение предложенного метода с другими методами

Методы	F1 Wireframe	F1 York	FPS CPU	FPS GPU
Unet + Canny	0.72	0.66	14	63
Unet	0.58	0.52	14	63
LSD	0.62	0.60	50	–
HAWP	0.80	0.65	–	29.5
L-CCN	0.77	0.62	–	15.6
(Huang et al., 2018)	0.72	0.61	–	2.24

Примечание: Численные результаты методов цитируются по работам (Grompone et al., 2012; Xue et al., 2020; Yichao et al., 2019; Huang et al., 2018). Объективности сравнения могут помешать различия в программном и аппаратном обеспечении, использованном для замеров. Мы полагаем, что связанные с этим различия незначительны.

Предложенный метод сравним с аналогичными методами по качеству на наборе wireframe, воспроизводит наилучшее качество для набора York и устанавливает наилучшую скорость на GPU, которая превосходит инженерный метод LSD на CPU.

### ВЫВОД

В данной работе рассматривается задача детектирования отрезков в виде классификации

пикселей изображения на принадлежность отрезку. Предложен метод построения быстрого и точного детектора на основе нейросети типа Unet и классического детектора границ Canny. Показано, что качество детектирования, составившее 0.72 F1 для набора Wireframe и 0.66 для набора York, сопоставимо с аналогами, а скорость работы 14 FPS на CPU и 66 FPS на GPU превосходит представленные в литературе методы.

## Improvement of a line segment detector based on a neural network by adding engineering features

L. A. Erlygin<sup>a,b,#</sup> and L. M. Teplyakov<sup>b</sup>

<sup>a</sup> Moscow Institute of Physics and Technology (State University),  
141701 Dolgoprudny, Institutsky per., 9, Moscow region, Russia

<sup>b</sup> Institute for Information Transmission Problems. A.A. Kharkevich Russian Academy of Sciences,  
127051 Moscow, Bolshoi Karetny lane, 19, Russia

<sup>#</sup>E-mail: erlygin.la@phystech.edu

In this work, we study the problem of line segment detection. In the recent studies of this problem, it was shown that neural network-based methods are able to outperform classical algorithms in accuracy, yet high computational complexity of neural networks limits their usage in real-world applications. In our work, we propose to fuse a neural network with a classical algorithm to compensate their flaws, by (i) feeding into the network not only an image, but also classical algorithm's output and (ii) by combining their predictions. We expect these modifications to simplify the problem faced by the neural network, resulting in higher accuracy in smaller network. Proposed method provides 0.72 and 0.66 F – measure on Wireframe and York datasets respectively with 14 FPS on CPU.

*Keywords:* artificial neural networks, line segment detection, contour detection, Unet, Canny

### REFERENCES

- Bandera A., Pérez-Lorenzo A., Bandera J.M., Sandoval F. Mean shift based clustering of hough domain for fast line segment detection. *Pattern Recognition Letters*. 2006. P. 578–586.
- Bista S.R., Giordano P.R., Chaumette F. Appearance-based indoor navigation by IBVS using line segments. *IEEE Robotics and Automation Letters*. 2016. P. 423–430.
- Burns J.B., Hanson A.R., Riseman E.M. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986. P. 425–455.
- Canny J. A Computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986. P. 679–698.
- Coughlan J.M., Yuille A.L. Manhattan world: orientation and outlier detection by bayesian inference. *Neural Comput.* 2003. P. 1063–1088.
- Duda R.O., Hart P.E. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*. 1972. P. 11–15.
- Grompone von Gioi R., Jakubowicz J., Morel J.M., Randall G. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2010. P. 722–732.
- Grompone von Gioi R., Jakubowicz J., Morel J.M., Randall G. LSD: A line segment detector. *Image Processing On Lin*. 2012. P. 35–55.
- Huang J., Wang Z., Liang H. Lane marking detection based on segments with upper and lower structure. *International Journal of Pattern Recognition and Artificial Intelligence*. 2020.
- Huang K., Wang Y., Zhou Z., Ding T., Gao S., Ma Y. Learning to parse wireframes in images of man-made environments. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018. P. 626–635.
- Kingma D.P., Ba J.L. Adam: A Method for stochastic optimization, 2014.
- Kiryati N., Eldar Y., Bruckstein A.M. A probabilistic hough transform. *Pattern Recognition*. 1991. P. 303–316.
- Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. 2012. P. 1097–1105.
- Lin Y., Silvia L., Pinte S. Deep hough-transform line priors, 2020.
- Ronneberger O., Fischer P., Tomar B. U-Net: convolutional networks for biomedical image segmentation. 2015.
- Xue N., Bai S., Wang F., Xia G., Wu T., Zhang L., Torr P. Learning regional attraction for line segment detection. *IEEE transactions on pattern analysis and machine intelligence*. 2019.
- Xue N., Wu T., Bai S., Wang F.D., Xia G.S., Zhang L. Holistically-attracted wireframe parsing, 2020.
- Yichao Z., Haozhi Q., Ma Y. End-to-end wireframe parsing. 2019.
- Zhukovsky A., Nikolaev D., Arlazarov V., Postnikov V., Polevoy D., Skoryukina N., Povolotsky M. Segments graph-based approach for document capture in a smartphone video stream. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). 2017. P. 337–342.