

М. А. Резницкий¹, Л. В. Аршинский²

¹ *Группа компаний «Эфко», г. Алексеевка, Российская Федерация*

² *Иркутский государственный университет путей сообщения*

КОНТРОЛЬ СОСТОЯНИЯ ЭЛЕМЕНТОВ ВЕРХНЕГО СТРОЕНИЯ ПУТИ ПО ЕГО ИЗОБРАЖЕНИЯМ

Аннотация. В статье рассматриваются один из подходов к решению задачи автоматизированного обнаружения дефектов верхнего строения железнодорожного пути на основе методов обработки изображений. Вся задача разбивается на ряд последовательных подзадач: предобработка изображений, обнаружение рельсовых дефектов, поиск рельсовых креплений, выявление креплений, подозрительных на дефект. Для предобработки использовался подход, основанный на использовании матричных фильтров для операции свёртки. Решение задачи поиска рельсовых дефектов выполнялось на основе алгоритма поиска областей, отличных от заданного цвета внутри заданной прямоугольной области. Рельсовые крепления выявлялись на основе адаптированного к особенностям предметной области алгоритма сопоставления по шаблону. Ввиду разнообразия дефектов было принято решение разделять изображения на два класса: с дефектами и без с последующей передачей изображений, подозрительных на дефект для ручной обработки. Для этого изображения делятся на четыре группы:

- промахи; на таких изображениях креплений либо нет совсем, либо они не полностью попали в ракурс ввиду неточностей алгоритма поиска креплений;
- реальные дефекты; на таких изображениях крепления либо отсутствуют ввиду неисправности, либо присутствует иная поломка;
- идеальные крепления; здесь изображениях крепления не содержат дефектов и полностью видны;
- прочее; на таких изображениях присутствуют белые пятна, мусор, крепления могут быть засыпаны камнями, закрыты иными объектами и другое.

Все неидеальные изображения считаются подозрительными на дефект. Для их разделения с идеальными использовалась сверточная нейронная сеть. Обсуждаются вопросы обучения такой сети.

Ключевые слова: железная дорога, верхнее строение пути, свёрточная нейронная сеть, обработка изображений, обнаружение дефектов

М. А. Reznitskiy¹, L. V. Arshinskiy²

¹ *EFKO Group of Companies, Alekseevka, the Russian Federation*

² *Иркутский государственный университет путей сообщения*

CONTROL THE STATE OF ELEMENTS OF THE PERMANENT WAY IN HIS IMAGES

Abstract: The article considers one of the approaches to solving the problem of automated detection of defects of the upper structure of the railway track on the basis of image processing methods. The whole problem is broken into a series of sequential subtasks: image preprocessing, detection of rail defects, rail fasteners search, identification of fasteners that are suspected to defect. For preprocessing, an approach based on the use of matrix filters for the convolution operation was used. The solution of the problem of finding rail defects was performed on the basis of the algorithm for finding areas other than a given color within a given rectangular area. Rail fasteners were identified on the basis of the pattern matching algorithm adapted to the characteristics of the subject area. In view of the variety of defects, it was decided to divide the images into two classes: with and

without defects, followed by the transmission of images suspicious of the defect for manual processing. The images are divided into four groups:

- mistakes; in such images of bonds or not at all, or they are not completely caught in the perspective due to inaccuracies of the algorithm for finding bonds;
- real defects; on such images of fastening or are absent in view of malfunction, or there is other breakage;
- perfect staples; here the staples images are free of defects and are fully visible;
- other; in such images there are white spots, debris, bonds can be covered with stones, covered with other objects and more.

All non-ideal images are considered suspicious of the defect. A convolutional neural network was used to separate them from the ideal ones. The issues of training such a network are discussed

Keywords: image processing, railway, upper structure of the railway track, convolutional neural network, defect detection.

Введение. Железная дорога представляет собой зону повышенной опасности по различным причинам, одной из которых являются возможные повреждения в верхнем строении пути. Для контроля наличия повреждений на путях используют оснащённые специальным оборудованием для высокочастотной видеосъёмки поезда, которые с некоторой периодичностью проходят по железнодорожным путям. В настоящее время во всём мире полученные данные обрабатываются вручную. Специалисты ежегодно просматривают терабайты снимков путей с целью поиска, фиксации и последующего устранения всех возможных дефектов. В связи с этим возникает задача автоматизации процесса поиска дефектов с целью улучшения качества обработки и освобождения человеческих ресурсов.

Естественным способом решения данной задачи является компьютерная обработка фотографий верхнего строения пути технологиями машинного зрения. Область компьютерного зрения является молодой, разнообразной и динамично развивающейся наукой. В настоящее время одним из самых перспективных направлений являются свёрточные нейронные сети (англ. convolution neural network, CNN) [1-3], входящие в состав технологии глубинного обучения (англ. deep learning) [4] и эффективно решающие задачи классификации объектов на изображении [5]. Технология нейронных сетей получила широкое распространение в таких задачах, как сегментация и классификация изображений, дополненная реальность, медицинская диагностика, автопилотирование автомобилей и многое другое (см. напр. [6]).

Важной особенностью задачи выявления дефектов по изображениям является значительное превышение стоимости ошибки 2 рода (принятие дефекта за его отсутствие) над стоимостью ошибки 1 рода (принятие отсутствия дефекта за его наличие). Поэтому главной целью является достоверное выявление тех изображений, на которых точно дефектов нет. В силу большого разнообразия типов возможных дефектов, рассматриваются только задачи поиска рельсов, шпал и креплений, с последующим анализом рельсов и креплений на предмет наличия дефекта, а именно:

- разработка и реализация алгоритмов сегментации изображений с целью выделения зон, относящихся к рельсам, шпалам и креплениям;
- создание обучающей выборки для CNN, состоящей из фрагментов с изображениями креплений, разделённых на две категории – изображения без дефектов и подозрительные на дефекты. Разделение выборки на обучающую и тестовую;
- эмпирический поиск оптимальной конфигурации (структуры) CNN с целью минимизации вероятности ошибки второго рода.

Вся задача разбивается на ряд последовательных этапов:

1. Предварительная обработка изображений.
2. Обнаружение рельсовых дефектов.
3. Задача поиска рельсовых креплений.
4. Выявление креплений, подозрительных на дефект.

Выделение рельсов на изображении. Первым шагом при работе с изображениями является предварительная обработка с целью улучшения качества исходных данных, устранения шумов и засвеченных пикселей. Здесь использовался подход, основанный на использовании матричных фильтров для операции свёртки [7]. Ядро свёртки заполняется по Гауссовскому (нормальному) закону распределения.

Основным элементом рассматриваемого изображения является рельс. Относительно рельса выделяются другие элементы изображения: шпалы и крепления. При этом исходим из следующего:

- примерно известна ширина рельса;
- примерно известны координаты рельса по оси Ox ;
- известно, что рельс белый.

Для решения задачи поиска рельсов был разработан алгоритм поиска областей заданного цвета на изображении – метод *search_def* (поиск областей, подозрительных на дефекты) Результатом работы метода являются координаты всех прямоугольных областей, описанных вокруг непересекающихся областей белого цвета, произвольных по форме. Однако таких областей на изображении может быть больше двух, в тех случаях, когда:

- изображение рельса разбито на несколько областей в стыке или при коррозии;
- на исходных изображениях присутствуют объекты белого цвета, не являющиеся рельсом, например, белый гравий.

Возникает задача идентификации из множества областей именно тех, которые соответствуют левому и правому рельсам. Для этого используется несколько параметров:

- минимальная высота области рельса;
- минимальная и максимальная ширина области рельса;
- горизонтальные диапазоны для левого и правого рельсов, в один из которых должна попадать область.

Совокупность этих параметров используются в качестве критерия для идентификации областей рельсов. Затем области достраиваются до прямоугольника по высоте изображения. Координаты найденных прямоугольников считаются координатами рельсов. В случае отсутствия результатов поиска для одного или обоих рельсов, данный кадр передаётся на ручную проверку пользователю, и не участвует в дальнейшей обработке.

Поиск рельсовых дефектов. Для решения задачи поиска рельсовых дефектов был разработан алгоритм поиска областей, отличных от заданного цвета, внутри заданной прямоугольной области на изображении, основанный на вышеупомянутом методе *search_def* [8]. Результатом его работы являются координаты всех прямоугольных областей, описанных вокруг произвольных по форме непересекающихся областей не белого цвета. Однако такие области могут включать не только дефекты, но и рельсовые стыки. Возникает задача идентификации областей стыков и дефектов (что есть что). Особенностью стыков является их экранная форма. Пример стыка изображён на рис. 1. Как видно, стык представляет собой узкий строго прямоугольный фрагмент изображения, заполненный пикселями тёмного цвета. Совокупность этих параметров и используются в качестве критерия для разделения.



Рис. 1. Пример рельсового стыка

Рассмотрим работу алгоритма в целом (блок-схема показана на рис. 2).

Пусть *left*, *right*, *bot* и *top* – координаты вершин прямоугольной области изображения;

Color – цвет области, которую необходимо найти;

Range – допустимый цветовой диапазон.

Функция *расстояние* (пиксель, цвет) находит расстояние в цветовом пространстве от текущего пикселя с координатами (R, G, B) до точки с координатами $(R_{\text{цв}}, G_{\text{цв}}, B_{\text{цв}})$ заданной переменной *Color*. Это расстояние сравнивается с заданным диапазоном *range*.

Метод *search_def* обнаруживает сегменты изображения, отличные от заданного цвета. При этом на вход подаётся не всё изображение, а область *Region*. Два блока сравнения с участием функции *расстояние* (пиксель, цвет) работают таким образом, что формируют области с цветом, не совпадающим с заданным (то есть выполняется поиск областей, не принадлежащих цветовому диапазону). Далее такие области рассматриваются как подозрительные на дефект.

Шаги алгоритма:

1. Поиск областей белого цвета на фотографии методом *search_def*.
2. Для областей каждого из рельсов выполняется расчёт координат минимальной прямоугольной области, включающая в себя все пиксели данной области.
3. Идентификация областей рельсов.
4. Если область хотя бы для одного из рельсов не найдена, передача кадра на ручную проверку пользователю. Исключение кадра из списка для дальнейшей обработки. Завершение работы алгоритма.
5. Поиск областей, отличных от белого цвета, внутри областей рельсов.
- 5.1. Идентификация каждой из найденных областей как стыка или дефекта.

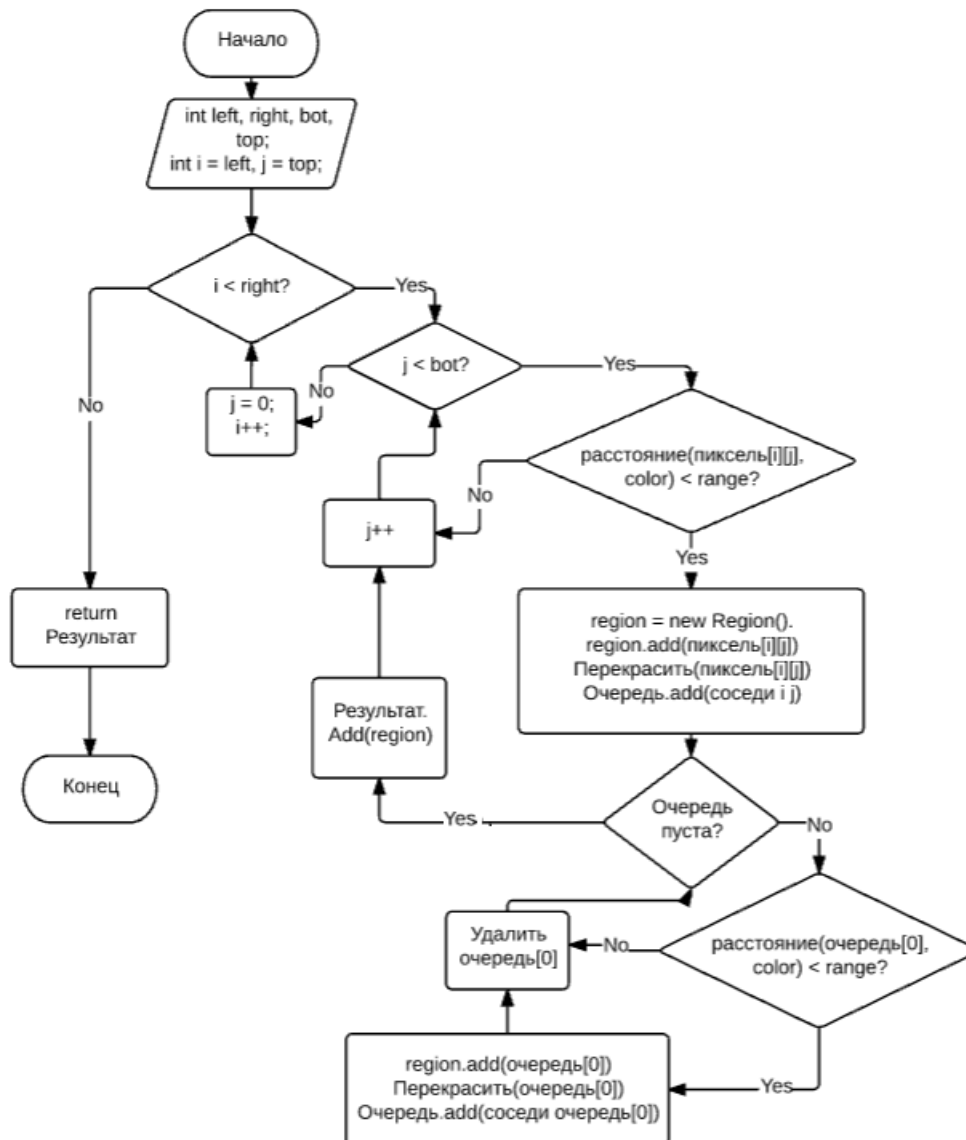


Рис. 2. Блок-схема алгоритма поиска областей заданного цвета.

5.2. В случае наличия хотя бы одного дефекта, передача кадра на ручную проверку пользователю.

Результат представлен на рис. 3.

Задача идентификации креплений. К особенностям данной задачи можно отнести:

- координаты креплений по оси Ox можно рассчитать относительно рельсов;
- координаты рельсов по оси Oy неизвестны;
- крепления размещаются на шпалах и должны быть выделены на них.

При этом также возникают проблемы идентификации:

- крепление может быть засыпано камнями;
- крепления вообще может не быть в случае дефекта;
- шпала, на которой расположен рельс, также может быть засыпана камнями.

В связи с этими и другими особенностями было решено осуществлять поиск не сразу креплений, а фрагментов изображений, представляющих собой пересечения шпал с рельсами с внешних по отношению к рельсам сторон, и только потом обнаруживать крепления.

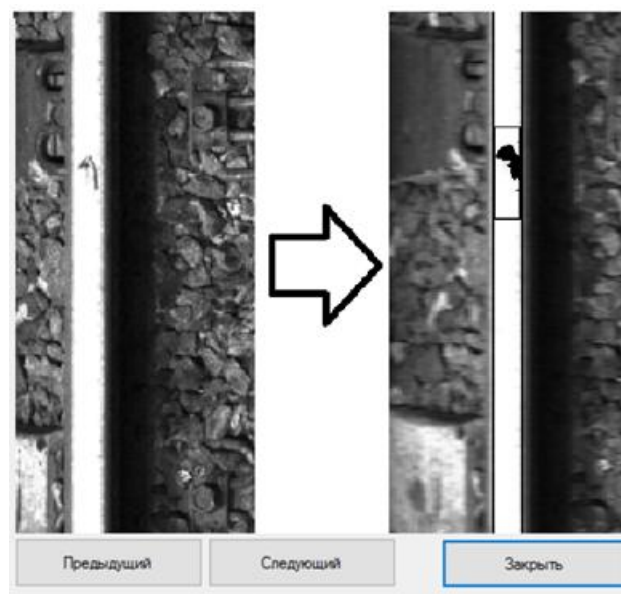


Рис. 3. Результат работы алгоритма

Основная идея алгоритма поиска креплений основана на применении операции свёртки для полос вдоль рельсов с внешних сторон (рис. 4).

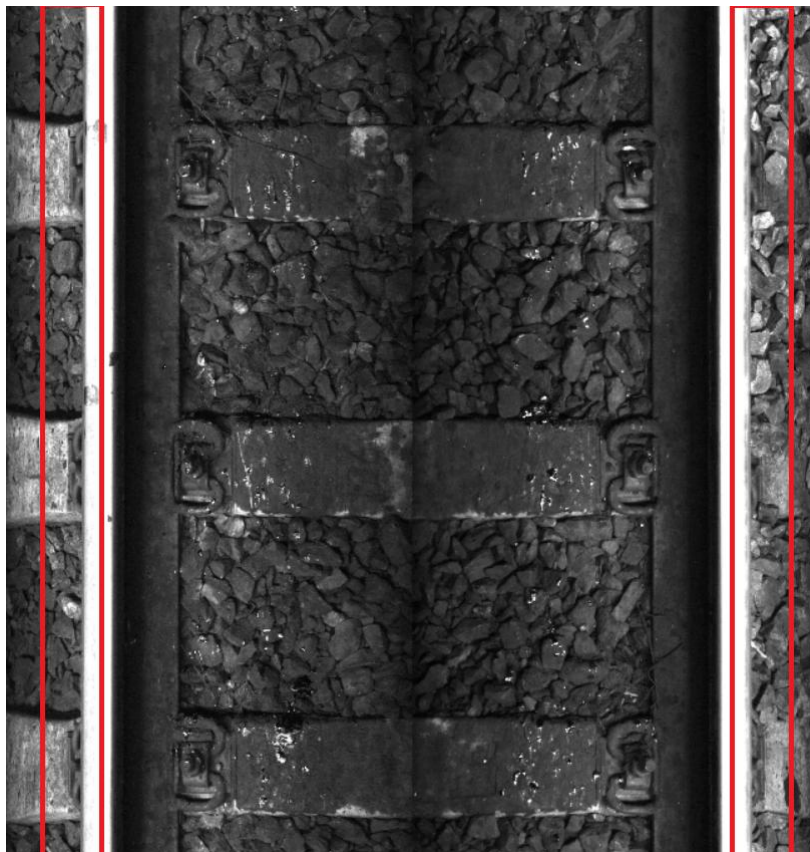


Рис. 4. Пример полос вдоль рельсов с внешних сторон

В качестве ядер свёртки используются фрагменты изображений – «шаблоны» скреплений. В качестве применяемой математической операции – среднее квадратическое отклонение текущего изображения от шаблона. Задача заключается в том, чтобы найти такие позиции ядер свёртки, при которых фрагмент изображения под ядром свёртки даст наименьшее отклонение от него. Пример таких полос изображён на рис. 4 (выделено красными прямоугольниками). Примеры изображений-ядер свёртки показаны на рис. 5.

Ввиду большого разнообразия шаблонов, свёртка для каждого рельса применяется по нескольку раз для разных шаблонов. Шаблоны формируются для левого и правого рельсов отдельно, при этом чем более репрезентативна выборка шаблонов, тем больше вероятность того, что найдётся такой шаблон, при котором отклонение для какой-то позиции ядра свёртки окажется ниже порогового значения, что и будет сигналом к тому, что данная позиция соответствует искомому фрагменту изображения.

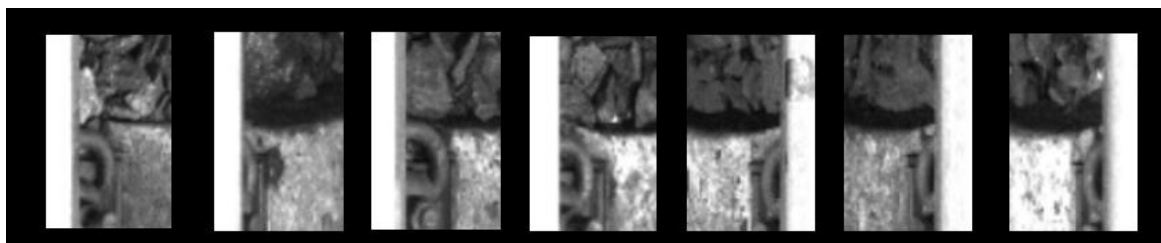


Рис. 5. Примеры шаблонов

Однако увеличение объёма выборки шаблонов порождает увеличение длительности работы алгоритма, поэтому необходимо наложить ограничение на их количество. В текущей конфигурации было решено использовать по 30 шаблонов для левого и правого рельсов.

Для нормализации результатов рассчитывается максимальное значение отклонения среди всех полученных. Для каждого отклонения производится преобразование: $value = 1 - value/global_max$, где $value$ – результат свёртки шаблона с фрагментом исходного изображения, $global_max$ – рассчитанное максимальное отклонение. Для итогового значения выполня-

ется зависимость: чем ближе число к единице, тем больше текущий фрагмент изображения похож на шаблон.

Описанный алгоритм основан на использовании принципов выделения эталонов, именуемых цифровыми отпечатками. Он и в дальнейшем будет называться алгоритмом поиска по шаблону [9, 10]. Аналогичные подходы используются, например, в задачах корреляционно-экстремальной привязки [11].

Результаты применения тридцати свёрток для некоторого изображения приведены на рис. 6, где по оси Ox – вертикальная координата пикселя, к которому прикладывается ядро свёртки, по оси Oy – нормированная величина отклонения.

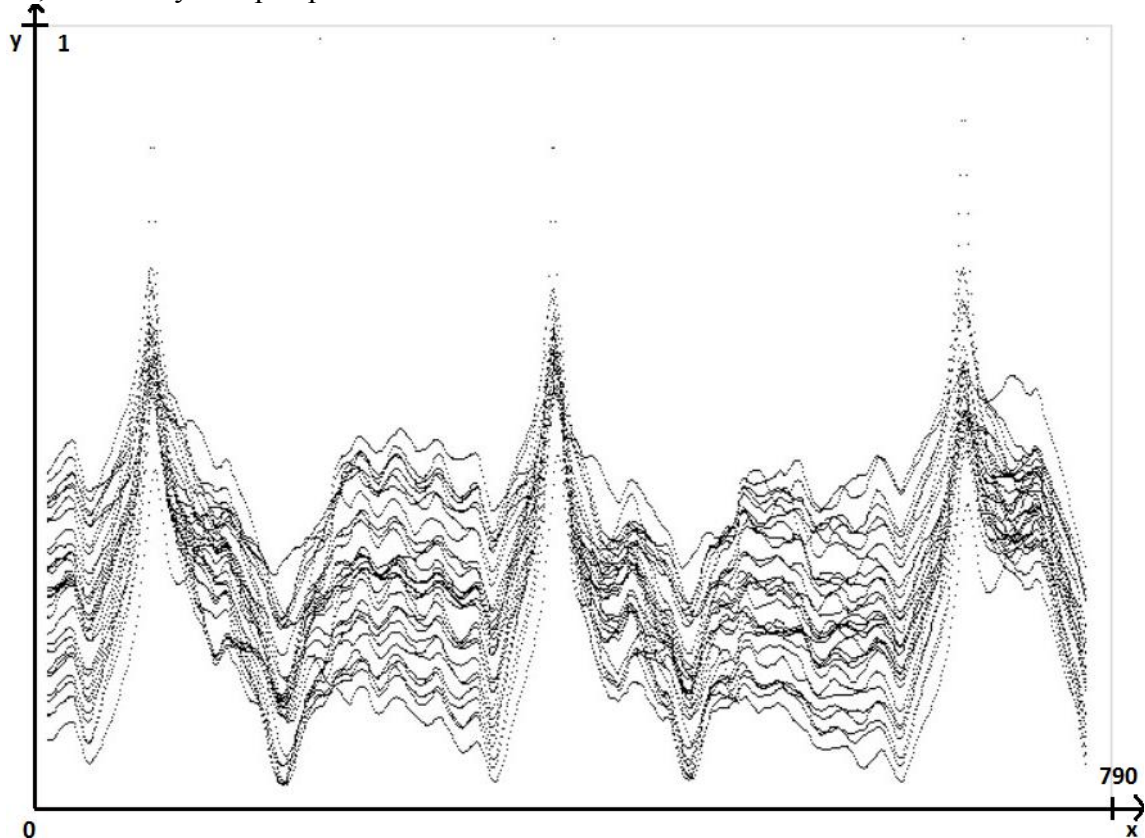


Рис. 6. Пример результатов применения тридцати свёрток

Для усреднения результатов используется некоторая функция $g(x)$. Итоговое отклонение для i -ой позиции по оси Oy рассчитывается по формуле:

$$R_i = g^{-1} \left(\frac{\sum_{i=1}^n g(x_i)}{n} \right);$$

где n – количество шаблонов для данного рельса; x_i – результат свёртки для j -го шаблона.

В качестве функции $g(x)$ необходимо использовать такую функцию, чтобы те свёртки, которые дали малое значение отклонения, оказывали на результат большее влияние, чем остальные. Поскольку величины нормированы, то искомую закономерность можно определить так: чем ближе число к единице, тем больше оно влияет на значение R_i . Ввиду репрезентативности выборки шаблонов, это правило означает: если данный фрагмент изображения «похож» на какие-то шаблоны и «непохож» на другие шаблоны, то он с большой вероятностью является искомым, поэтому при усреднении необходимо учитывать результаты «похожих» шаблонов в большей степени.

Было проведено сравнение эффективности работы алгоритма для функций $x, x^3, e^{\sqrt{x}}, e^{10x}, e^{100x}$. Наглядное сравнение перечисленных функций для двух изображений

представлено на рис. 7. По результатам было принято решение использовать функцию $g(x) = e^{\sqrt{x}}$.

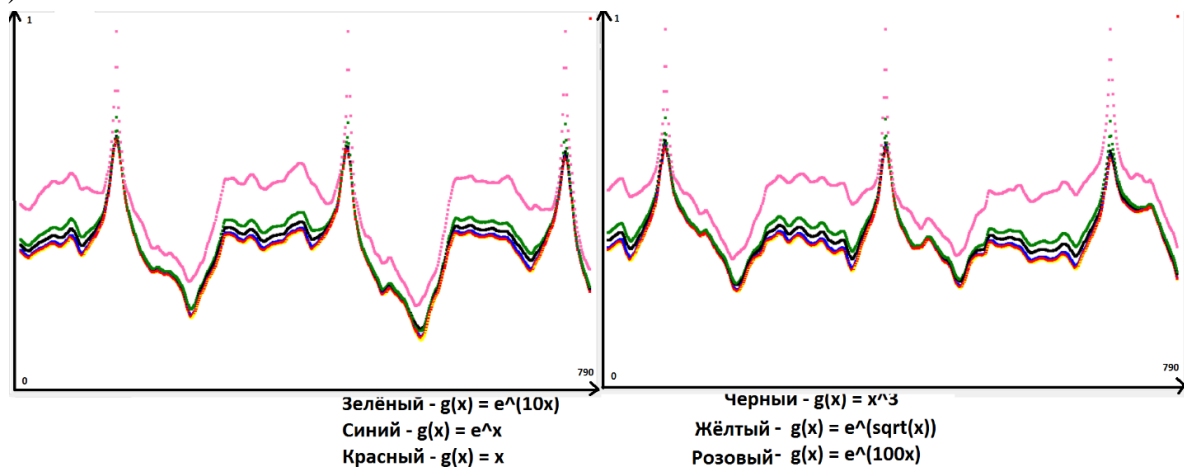


Рис.7. Сравнение результатов усреднения для различных $g(x)$

Следующая задача заключается в поиске локальных максимумов по усреднённым отклонениям. Проблема заключается в том, что на исходном изображении искомым фрагментам может быть либо два, либо три, следовательно, и количество локальных максимумов, которые нужно найти, заранее неизвестно. С учётом этих особенностей был разработан следующий алгоритм выделения локальных максимумов, состоящий из следующих шагов:

1. Исходный диапазон вдоль рельса по оси Oy разбивается на 3 части с небольшими пересечениями.

2. В каждом диапазоне находится максимальное значение и позиция свёртки по оси Oy , при котором оно было достигнуто.

3. Для пар диапазонов (первый, второй) и (второй, третий):

3.1. Если расстояние между позициями меньше 100 пикселей, из двух максимумов выбирается наибольшее значение и соответствующая ему позиция.

В результате работы данного алгоритма получается две или три позиции свёртки, которые принимаются за позиции искомым фрагментам изображения.

С учетом этого выполняются шаги (рис. 8):

1. Загрузка шаблонов для поиска нужных областей вдоль рельсов.

2. Применение алгоритма поиска креплений для каждого шаблона. Полученные массивы *results* сохраняются в списки *res_left* и *res_right* для левого и правого рельса соответственно.

3. Нормализация и усреднение результатов по всем шаблонам.

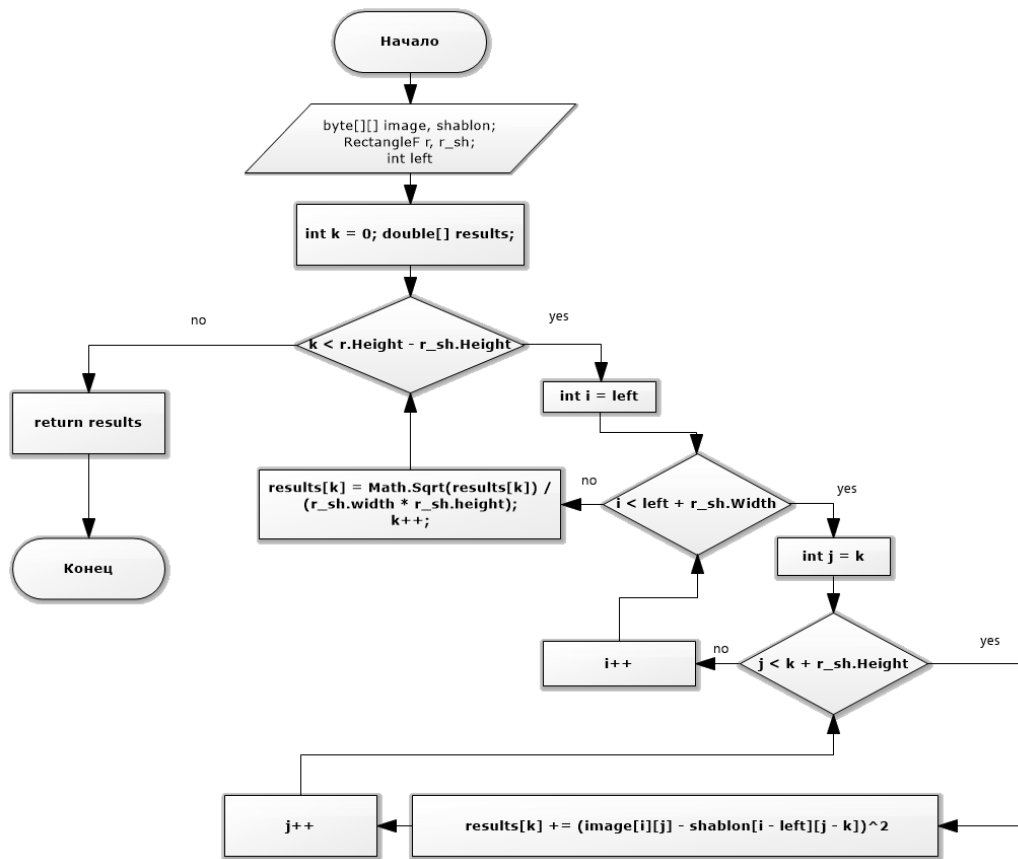


Рис. 8. Блок-схема алгоритма поиска по шаблону.

4. Поиск локальных максимумов для левого и правого рельса.

5. Расчёт координат скреплений на изображении исходя из полученных координат пересечений рельсов со шпалами.

Поиск скреплений без дефектов. В силу разнообразия дефектов и связанными с эти трудностями идентификации, было принято решение отказаться от их поиска, а разделять изображения с дефектами и без дефектов. Изображения без дефектов пропускаются, а подозрительные на дефект передаются для ручной обработки оператором. Предполагается, что уже одно это существенно снизит нагрузку на оператора и в значительной степени автоматизирует его труд. Для решения такого класса задач хорошо подходят методы искусственного интеллекта, а именно нейронные сети. Основная идея искусственных нейронных сетей заключается в обучении компьютера самостоятельно находить какие-то зависимости и закономерности, которые называют признаками, в исходных данных. В работе использована технология свёрточных нейронных сетей (CNN).

В результате работы программы для автоматизированной сегментации изображений было получено множество файлов формата .png с изображением скреплений (рис. 9). Эти данные необходимо вручную разделить на две категории – изображения без дефектов и изображения, подозрительные на дефекты.

После этого изображения делятся на четыре группы:

- промахи. На таких изображениях скреплений либо нет совсем, либо они не полностью попали в ракурс ввиду неточностей алгоритма поиска скреплений; примеры даны на рис. 9 а, б;

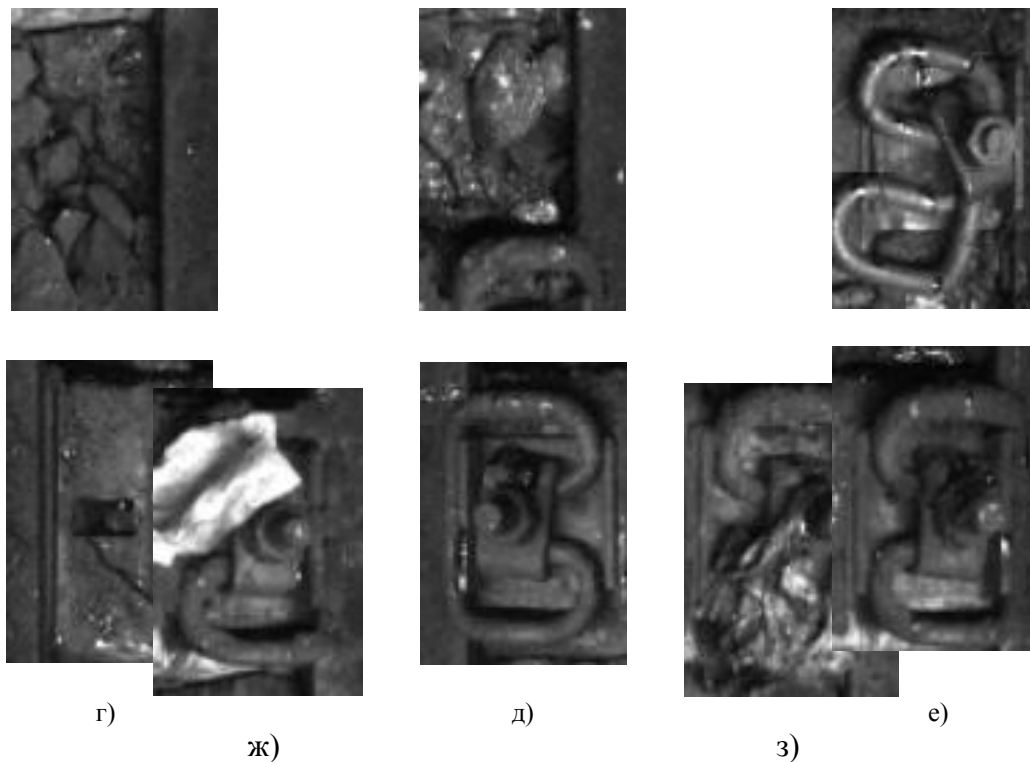
- реальные дефекты. На таких изображениях скрепления либо отсутствуют ввиду неисправности, либо присутствует иная полочка; примеры – на рис. 9 в, г;

- идеальные скрепления. На таких изображениях скрепления не содержат дефектов и полностью видны; рис. 9 д, е;

а)

б)

в)



а, б – промахи; в, г – реальные дефекты; д, е – идеальные скрепления; ж, з - прочее

Рис. 9. Примеры изображений скреплений.

– прочее. На таких изображениях присутствуют белые пятна, мусор, скрепления могут быть засыпаны камнями, закрыты иными объектами и другое: рис. 9 ж, з.

Для обучения CNN в качестве категории изображений без дефектов использовались только идеальные скрепления. Все остальные группы рассматривались как подозрительные на дефекты и передавались для ручной проверки.

В ходе экспериментов были опробованы три конфигурации сетей: А, В и С, где

А – это сеть C1 S1 C2 S2 C4 C6;

В – это C1 S1 C2 S2 C3 S3 C4 C6;

С – это C1 S1 C2 S2 C3 S4 C4 C5 C6.

Здесь C1 – свёрточный слой с ядром размером 5 на 5 пикселей и шагом 1. На входе одно изображение, на выходе 32 карты признаков; C2 – свёрточный слой с ядром размером 5 на 5 пикселей и шагом 1. На входе 32 карты признаков, на выходе 64 карты признаков; C3 – свёрточный слой с ядром размером 3 на 3 пикселей и шагом 1. На входе 64 карты признаков, на выходе 128 карт признаков; S1, S2 и S3 – слои подвыборки с ядром 2 на 2 пикселя и шагом 2; C4 – полносвязный слой размером 1024 нейрона; C5 – полносвязный слой размером 256 нейронов; C6 – полносвязный слой размером 2 нейрона, в которых будет находиться распределение вероятностей того, что исходное изображение принадлежит первому или второму классу. Это выходной слой сети.

В качестве функции активации выбрана ReLU.

В результате ручной классификации фрагментов скреплений была получена выборка из 1932 данных, по 966 данных на каждый из двух классов объектов. Эта выборка была разделена на обучающую в размере 1344 изображений и тестовую в размере 588 изображений.

В таблице 1 приведено сравнение результатов для каждой из трёх конфигураций с использованием двух функций оптимизации: градиентный спуск [12] и ADAM [13]. Изображения подаются в сеть наборами (англ. batch) по 64 за одну итерацию. Для каждого набора из обучающей выборки измеряется точность ответов сети до обучения. Для тестовой выборки обучения после измерения точности не происходит.

Таблица 1. Сравнение результатов обучения CNN по точности для различных конфигураций сети

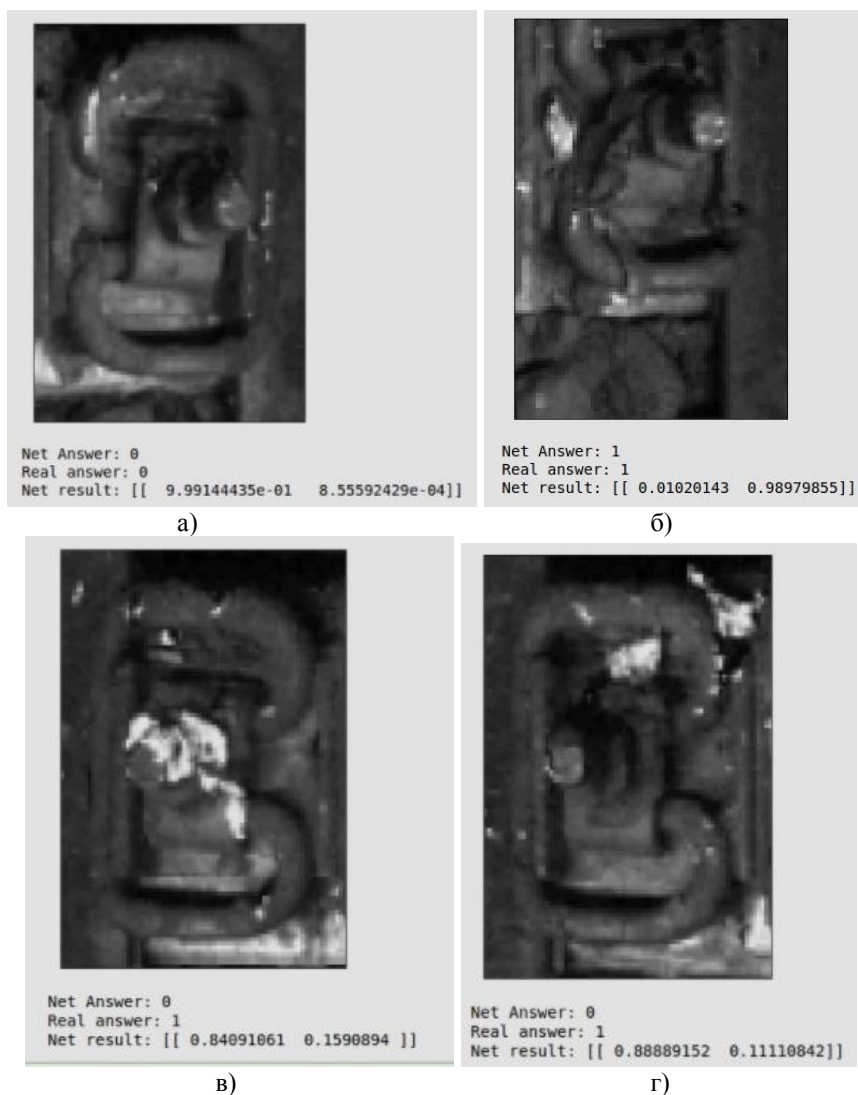
Обучающая выборка						
Объём	Конф. А.	Конф. А.	Конф. В.	Конф. В.	Конф. С.	Конф. С.

выборки	Gradient	Adam	Gradient	Adam	Gradient	Adam
64	0,53	0,56	0,5	0,47	0,53	0,49
128	0,47	0,66	0,44	0,56	0,56	0,56
192	0,47	0,38	0,53	0,59	0,44	0,53
256	0,5	0,62	0,57	0,53	0,62	0,72
320	0,38	0,5	0,41	0,56	0,59	0,53
384	0,53	0,66	0,41	0,72	0,59	0,66
448	0,47	0,43	0,56	0,88	0,78	0,72
512	0,59	0,56	0,53	0,66	0,53	0,59
576	0,47	0,66	0,72	0,84	0,56	0,69
640	0,5	0,87	0,63	0,63	0,56	0,84
704	0,56	0,625	0,63	0,62	0,59	0,53
768	0,47	0,66	0,59	0,72	0,78	0,75
832	0,59	0,69	0,63	0,75	0,75	0,66
896	0,66	0,81	0,78	0,78	0,75	0,75
960	0,44	0,91	0,41	0,65	0,59	0,91
1024	0,5	0,72	0,56	0,78	0,63	0,59
1088	0,67	0,56	0,53	0,94	0,69	0,88
1152	0,66	0,88	0,72	0,91	0,75	0,75
1216	0,59	0,88	0,69	0,78	0,66	0,81
1280	0,66	0,91	0,59	0,88	0,63	0,81
Тестовая выборка						
1376	0,69	0,84	0,69	0,81	0,56	0,78
1440	0,66	0,88	0,66	0,81	0,5	0,78
1504	0,62	0,75	0,63	0,88	0,62	0,75
1568	0,69	1,0	0,69	0,78	0,69	0,84
1632	0,67	0,81	0,56	0,84	0,53	0,94
1696	0,49	0,84	0,59	0,81	0,59	0,91
1760	0,56	0,72	0,53	0,94	0,53	0,75
1824	0,34	0,88	0,5	0,91	0,68	0,94
1888	0,44	0,91	0,59	0,91	0,49	0,84
Итоговая точность для тестовой вы- борки	0,575	0,835	0,601	0,837	0,539	0,825

Из таблицы видно, что при любой из перечисленных конфигураций функция оптимизации ADAM показывает существенно лучшие результаты, чем градиентный спуск.

Если рассматривать результаты сети для трёх конфигураций с использованием функции оптимизации ADAM, можно сказать, что увеличение гиперпараметров сети не оказывает влияния на итоговую точность ответов. Такой результат является классическим примером *недообучения сети*, в результате которого часть весов сети не влияет на её ответы. Для решения такой проблемы приходится увеличивать объём обучающей выборки.

На рис. 10 а, б показаны правильные ответы сети на оба класса объектов; на рис. 10 в, г показаны неправильные ответы сети на оба класса объектов. Для проверки использовалась конфигурация А, метод оптимизации ADAM. Входные изображения взяты из тестовой выборки.



а, б – верные ответы сети; в, г – неверные ответы сети
Рис. 10. Примеры ответов сети на данные из тестовой выборки

Заключение. По итогу ручной проверки результатов обработки 3000 данных, представляющих собой участок в 3 км путей, разработанные алгоритмы показали следующие показатели качества:

- алгоритм поиска рельсов успешно определяет координаты рельсов на 98 % исходных данных;

- алгоритм поиска креплений успешно определяет координаты креплений на 90 % исходных данных.

Данные показатели являются приемлемыми, поскольку ложные срабатывания алгоритмов устранимы при дальнейшей обработке.

По итогу обучения точность ответов CNN составила порядка 83 %. Такой результат не является приемлемым ввиду высокой стоимости ошибки второго рода. Для улучшения этого результата необходимо как увеличение объема тестовой и обучающей выборок данных, так и дальнейший эмпирический поиск эффективной конфигурации CNN.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. LeCun, Y. Convolutional Networks for Images, Speech, and Time-Series, in Arbib / LeCun, Y. Bengio ; MIT Press, 1995 – 14с.
2. Simard, P.Y. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis / P.Y. Simard, D. Steinkraus, J. Platt // International Conference on Document

3. Kozin, N.E. Gradual learning the radial neural networks / N.E. Kozin, V.A. Fursov // *Computer Optics*. – 2004. – № 26. – P. 138-141.
4. LeCun, Y. Gradient Based Learning Applied to Document Recognition / Y. LeCun, L. Bottou, P. Haffner – IEEE Press, 1998. – P. 46.
5. Хайкин С. Нейронные сети: полный курс, 2-е издание : Пер. с англ. Н.Н. КуССуль, А.Ю. Шелестова; под ред. Н.Н. КуССуль – М.: Издательский дом «Вильямс», 2006. – 1104 с.
6. Колмогорцев М.Г. Осипова В.Э. Разработка модели определения межпоездных интервалов с использованием аппарата искусственных нейронных сетей // *Известия Транссиба*. Т4(28), 2016. – С. 87-93.
7. Фисенко В.Т., Фисенко Т.Ю. Компьютерная обработка и распознавание изображений. – Санкт-Петербург: Изд-во СПбГУ ИТМО, 2008. – 281с.
8. Резницкий М.А. Решение задачи анализа верхнего строения пути по его изображениям // *Информационные технологии и проблемы математического моделирования сложных систем*. – Вып.16. – Иркутск: ИрГУПС, 2016. – С. 15-20/
9. Jahne B. Digital image processing. 6th ed. Springer, 2005. – 639 с.
10. Hall, E. L. Computer Image Processing and Recognition, Academic Press, New York, 1979. – 811с.
11. Белоглазов И.Н., Тарасенко В.П. Корреляционно-экстремальные системы. Советское радио, 1974. 392 с.
12. Борисов Е.С. О методах обучения многослойных нейронных сетей прямого распространения – <http://mechanoid.kiev.ua/neural-net-backprop2.html> (дата обращения 05.11.2018)
13. Adam: A method for stochastic optimization – <https://arxiv.org/abs/1412.6980> (Дата обращения 05.11.2018)

REFERENCES

1. LeCun, Y. Convolutional Networks for Images, Speech, and Time-Series, in Arbib / LeCun, Y. Bengio ; MIT Press, 1995 – 14с.
2. Simard, P.Y. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis / P.Y. Simard, D. Steinkraus, J. Platt // *International Conference on Document Analysis and Recognition (ICDAR)*, IEEE Computer Society. – Los Alamitos. – 2003. – P. 958-962.
3. Kozin, N.E. Gradual learning the radial neural networks / N.E. Kozin, V.A. Fursov // *Computer Optics*. – 2004. – № 26. – P. 138-141.
4. LeCun, Y. Gradient Based Learning Applied to Document Recognition / Y. LeCun, L. Bottou, P. Haffner – IEEE Press, 1998. – P. 46.
5. Khaykin S. Neyronnyye seti: polnyy kurs. 2-e izdaniye [Neural networks: full course. 2nd edition]: Per. s angl. N.N. Kussul. A.Yu. Shelestova; pod red. N.N. Kussul – M.: Izdatelskiy dom «Viliams». 2006. – 1104 p. (in Russian).
6. Kolmogortsev M.G. Osipova V.E. Razrabotka modeli opredeleniya mezhpoyezdnykh intervalov s ispolzovaniyem apparata iskusstvennykh neyronnykh setey [Development of a model for determining inter-train intervals using artificial neural networks] // *Izvestiya Transsiba*. Vol.4(28). 2016. – pp. 87-93. (in Russian).
7. Fisenko V.T., Fisenko T.Yu. Kompyuternaya obrabotka i raspoznavaniye izobrazheniy [Computer processing and image recognition]. – Sankt-Peterburg: Izd-vo SPBGU ITMO. 2008. – 281p. (in Russian).
8. Reznitskiy M.A. Resheniye zadachi analiza verkhnego stroyeniya puti po ego izobrazheniyam [Solving the problem of analysis of the upper structure of the path by its images] // *Informatsionnyye tekhnologii i problemy matematicheskogo modelirovaniya slozhnykh sistem*. – Vol.16. – Irkutsk: IrGUPS. 2016. – pp. 15-20. (in Russian).
9. Jahne B. Digital image processing. 6th ed. Springer, 2005. – 639 с.

10. Hall, E. L. Computer Image Processing and Recognition, Academic Press, New York, 1979. – 811с.
11. Beloglazov I.N., Tarasenko V.P. Korrelyatsionno-ekstremalnyye sistemy [Correlation-extreme systems]. Sovetskoye radio. 1974. 392 p. (In Russian).
12. Borisov E.S. O metodakh obucheniya mnogosloynnykh neyronnykh setey pryamogo rasprostraneniya [Methods of training multilayered neural networks of direct distribution] – <http://mechanoid.kiev.ua/neural-net-backprop2.html> (05.11.2018) (in Russian).
13. Adam: A method for stochastic optimization – <https://arxiv.org/abs/1412.6980> (05.11.2018).

Информация об авторах

Михаил Александрович Резницкий – инженер-программист группы компаний «Эфко», г. Алексеевка, e-mail: natz94@mail.ru.

Леонид Вадимович Аршинский – д.т.н., доцент, заведующий кафедрой «Информационные системы и защита информации» ФГБОУ ВО «Иркутский государственный университет путей сообщения», г. Иркутск, e-mail: arsh@irgups.ru.

Authors

Mikhail Aleksandrovich Reznitskiy – engineer-programmer of the EFKO Group of Companies, e-mail: natz94@mail.ru

Leonid Vadimovich Arshinskiy – Doctor of Technical Science, head of the Department of Information System and Information Security, Irkutsk State Transport University, Irkutsk, e-mail: arsh@irgups.ru

Для цитирования

Резницкий М.А., Аршинский Л.В. Контроль состояния элементов верхнего строения пути по его изображениям // Информационные технологии и математическое моделирование в управлении сложными системами: электрон. науч. журн. – 2019. – №1. – С. 82-95 – Режим доступа: <http://ismm-irgups.ru/toma/toma/12-2019>, свободный. – Загл. с экрана. – Яз. рус., англ. (дата обращения: 25.03.2019)

For citation

Reznitskiy M.A., Arshinskiy L.V. Kontrol sostoyaniya elementov verkhnego stroyeniya puti po ego izobrazheniyam [Control the state of elements of the permanent way in his images]. // Informacionnye tehnologii i matematicheskoe modelirovanie v upravlenii slozhnyimi sistemami: ehlektronnyj nauchnyj zhurnal [Information technology and mathematical modeling in the management of complex systems: electronic scientific journal], 2019, No. 1, P. 82-95. [Accessed 25/03/19]