

Анализ сигналов Signal Analysis

©Кузьмин Е. В., Горбунов О. Е., Плотников П. О., Тюкин В. А., Башкин В. А., 2018

DOI: 10.18255/1818-1015-2018-6-667-679

УДК 004.032.26

Применение нейронных сетей для распознавания конструктивных элементов рельсов на магнитных и вихретоковых дефектограммах

Кузьмин Е. В., Горбунов О. Е., Плотников П. О., Тюкин В. А., Башкин В. А.

Поступила в редакцию 1 октября 2018

После доработки 23 ноября 2018

Принята к публикации 30 ноября 2018

Аннотация. Для обеспечения безопасности движения на железнодорожном транспорте регулярно проводится неразрушающий контроль рельсов с применением различных подходов и методов, включая методы магнитной и вихретоковой дефектоскопии. Актуальной задачей является автоматический анализ больших массивов данных (дефектограмм), которые поступают от соответствующего оборудования. Под анализом понимается процесс определения по дефектограммам наличия дефектных участков наряду с выявлением конструктивных элементов рельсового пути. Данная статья посвящена задаче распознавания образов конструктивных элементов железнодорожных рельсов по дефектограммам многоканальных магнитных и вихретоковых дефектоскопов. Рассматриваются три класса конструктивных элементов рельсового пути: 1) болтовой стык с прямым или скошенным соединением рельсов, 2) электроконтактная сварка рельсов и 3) алюмотермитная сварка рельсов. Образы, которые не могут быть отнесены к этим трем классам, условно считаются дефектами и выносятся в отдельный четвертый класс. Для распознавания образов конструктивных элементов на дефектограммах применяется нейронная сеть, реализованная в рамках открытой библиотеки TensorFlow. С этой целью каждая выделенная для анализа область дефектограммы преобразуется в графический образ в градации серого цвета размером 20 на 39 пикселей.

Ключевые слова: неразрушающий контроль рельсов, магнитная и вихретоковая дефектоскопия, обнаружение дефектов, автоматический анализ дефектограмм, нейронные сети

Для цитирования: Кузьмин Е. В., Горбунов О. Е., Плотников П. О., Тюкин В. А., Башкин В. А., "Применение нейронных сетей для распознавания конструктивных элементов рельсов на магнитных и вихретоковых дефектограммах", *Моделирование и анализ информационных систем*, **25:6** (2018), 667–679.

Об авторах:

Кузьмин Егор Владимирович, orcid.org/0000-0003-0500-306X, д-р физ.-мат. наук, профессор кафедры теоретической информатики, Ярославский государственный университет им. П.Г. Демидова, ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: kuzmin@uniyar.ac.ru, kuzminev@nddlab.com

Горбунов Олег Евгеньевич, orcid.org/0000-0001-6274-9971, канд. физ.-мат. наук, генеральный директор, ООО «Центр инновационного программирования», NDDLAb, ул. Союзная, 144, г. Ярославль, 150008 Россия, e-mail: gorbunovoe@nddlab.com

Плотников Петр Олегович, orcid.org/0000-0001-5687-7969, инженер-технолог, ООО «Центр инновационного программирования», NDDLAb, ул. Союзная, 144, г. Ярославль, 150008 Россия, e-mail: plotnikovpo@nddlab.com

Тюкин Вадим Александрович, orcid.org/0000-0001-9149-7435, руков. сектора разработки, ООО «Центр инновационного программирования», NDDLAb, ул. Союзная, 144, г. Ярославль, 150008 Россия, e-mail: tyukinva@nddlab.com

Башкин Владимир Анатольевич, orcid.org/0000-0002-2534-1026, д-р физ.-мат. наук, профессор кафедры теоретической информатики, Ярославский государственный университет им. П.Г. Демидова, ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: bashkinva@nddlab.com

Введение

Для обеспечения безопасности движения на железнодорожном транспорте регулярно проводится неразрушающий контроль рельсов с применением различных подходов и методов, включая методы магнитной и вихретоковой дефектоскопии. Актуальной задачей является автоматический анализ [3–5] больших массивов данных (дефектограмм), которые поступают от соответствующего оборудования. Под анализом понимается процесс определения по дефектограммам наличия дефектных участков наряду с выявлением конструктивных элементов рельсового пути.

Данная статья посвящена задаче распознавания образов конструктивных элементов железнодорожных рельсов по дефектограммам магнитных и вихретоковых дефектоскопов. Статья является логическим продолжением работ [1, 2], посвященных определению порогового уровня шума рельсов. В этой работе мы ограничимся рассмотрением обобщения реального устройства в виде 10-разрядного вихретокового дефектоскопа с 10-ю каналами данных (анализ данных магнитных дефектоскопов проводится аналогичным образом). Каналы данных соответствуют физическим датчиками, которые последовательно располагаются на поверхности рельса перпендикулярно направлению движения дефектоскопа. Значения амплитуд сигналов каждого канала регистрируются дефектоскопом в виде натуральных чисел от 1 до 1024 (в 10-разрядном случае).

При автоматическом анализе дефектограммы обычно разбиваются на фрагменты, которые, например, могут соответствовать 50-метровым участкам рельсового пути, т. е. при снятии показаний 10-канального дефектоскопа с каждого миллиметра пути блок анализа представляет собой матрицу размером 10 строк на 50000 столбцов, где элемент матрицы — это значение амплитуды сигнала соответствующего канала данных.

Интерес представляют амплитуды только полезных сигналов. Сигнал считается полезным (и подлежит дальнейшему анализу), если отклонение его значения от среднего значения всех сигналов (в рамках одного канала) как минимум в два раза превосходит пороговый уровень шума рельсов. Как и ранее [1, 2], под пороговым уровнем шума (одного канала) будем понимать отклонение $Level$ от среднего значения μ сигналов рассматриваемого фрагмента дефектограммы (данные с 50-метрового участка), при котором сигналы со значениями амплитуд из диапазона $[\mu - Level; \mu + Level]$ являются шумом рельсов. Например, на рис. 1 в виде линий отсечки показаны пороговые уровень шума рельсов и уровень начала полезных сигналов для данных одного канала вихретокового дефектоскопа на 50-метровом участке рельсового пути.

Найденные полезные сигналы всех каналов группируются в отметки, которые представляют собой секции (дефектограммы) длиной 157 мм. Отметки формируются таким образом, чтобы иметь максимальное суммарное амплитудное отклонение по всем каналам на 157-миллиметровом участке дефектограммы среди близлежащих полезных сигналов. Также важным условием при формировании отметки является необходимость того, чтобы центр отметки совпадал с возможным центром реального конструктивного элемента, который может быть представлен на дефектограмме в виде записи данных. Алгоритм нахождения центра отметки в данной статье не рассматривается.

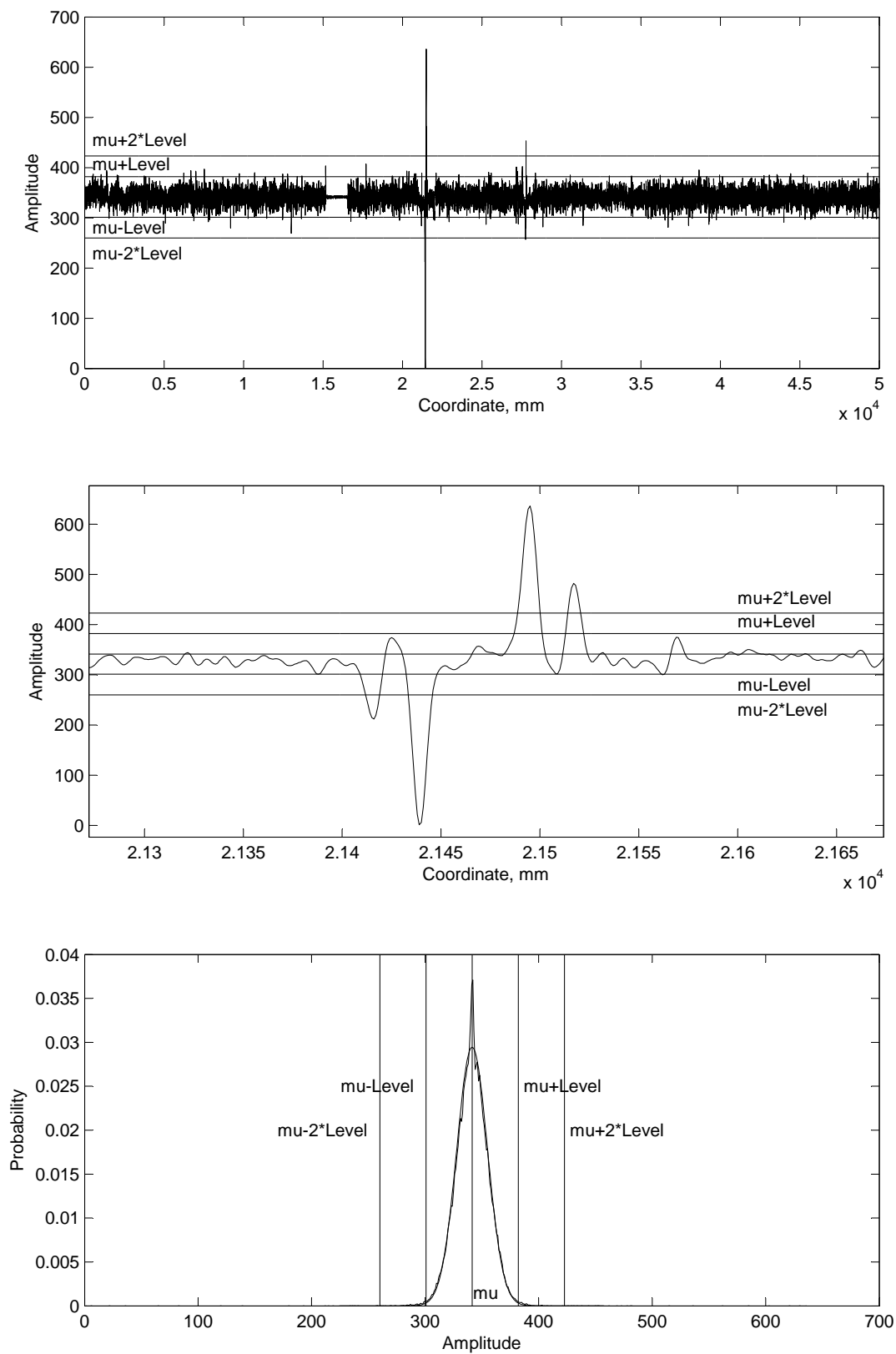


Рис. 1. Данные канала дефектоскопа с линиями отсечки полезных сигналов

Fig. 1. Data of one testing system channel with useful signal threshold lines

В данной статье рассматриваются следующие три класса конструктивных элементов рельсового пути: 1) болтовой стык с прямым или скошенным соединением рельсов, 2) электроконтактная сварка рельсов и 3) алюмотермитная сварка рельсов. Отметки, которые не могут быть отнесены к этим трем классам, условно считаются дефектами и выносятся в отдельный (четвертый) класс отметок.

Для распознавания образов конструктивных элементов на вихретоковых дефектограммах применяется нейронная сеть, реализованная в рамках открытой библиотеки TensorFlow (в состав которой входит библиотека Keros) [9]. С этой целью каждая отметка преобразуется в графический образ в градации серого цвета размером 20 на 157 пикселей с последующим построчным сжатием до размера 20 на 39 пикселей. Полносвязная нейронная сеть, состоящая из входного (780 нейронов), внутреннего (800 нейронов) и выходного (4 нейрона) слоев, обучается на базе эталонных образов указанных четырех классов (всего около 50 тыс. образцов).

Обученная нейронная сеть в составе аппаратно-программного комплекса рельсовой дефектоскопии успешно используется на практике для распознавания образов конструктивных элементов рельсового пути.

1. Формирование образа отметки

После того как по полезным сигналам на дефектограмме выделена отметка в виде матрицы амплитуд размером 10 строк (по числу каналов) на 157 столбцов (длина отметки в мм), предполагается преобразование отметки в графический образ размером 20 на 157 пикселей в градации серого цвета с последующим построчным сжатием образа до размера 20 на 39 пикселей.

Отметим, что все эти преобразования делаются для того, чтобы свести хорошо известную задачу распознавания рукописных цифр в поле размером 28 на 28 точек к текущей задаче распознавания конструктивных элементов рельсов на дефектограмме. Во многом размер и конфигурация выбранной нейронной сети продиктована результатами работы [6].

Удвоение количества строк в образе отметки. Первым шагом в изменении представления отметки является добавление в матрицу амплитуд по одной строке для каждого канала. При этом в нечетной строке остаются только те амплитуды, которые больше среднего значения μ амплитуд сигналов (одного канала) в текущем фрагменте дефектограммы, а также превосходят пороговый уровень полезных сигналов $\mu + 2 * \text{Level}$. Остальные значения в нечетной строке устанавливаются в ноль. Аналогичным образом, в четной строке остаются только те амплитуды, которые меньше порогового уровня полезных сигналов $\mu - 2 * \text{Level}$. Остальные значения в четной строке также устанавливаются в ноль. В результате имеем матрицу амплитуд полезных сигналов с их дифференциацией и последующим построчным размещением по направлению отклонения от среднего значения (соответствующего канала) размером 20 строк на 157 столбцов.

Переход к 8-битовым значениям. Следующим шагом является представление текущего образа отметки в виде картинки в градации серого цвета. Для этого необходимо перейти к 8-битовым значениям матрицы. При этом осуществляется перевод амплитуд сигналов в децибелы с числом $2 * \text{Level}$, т. е. уровнем начала полезных

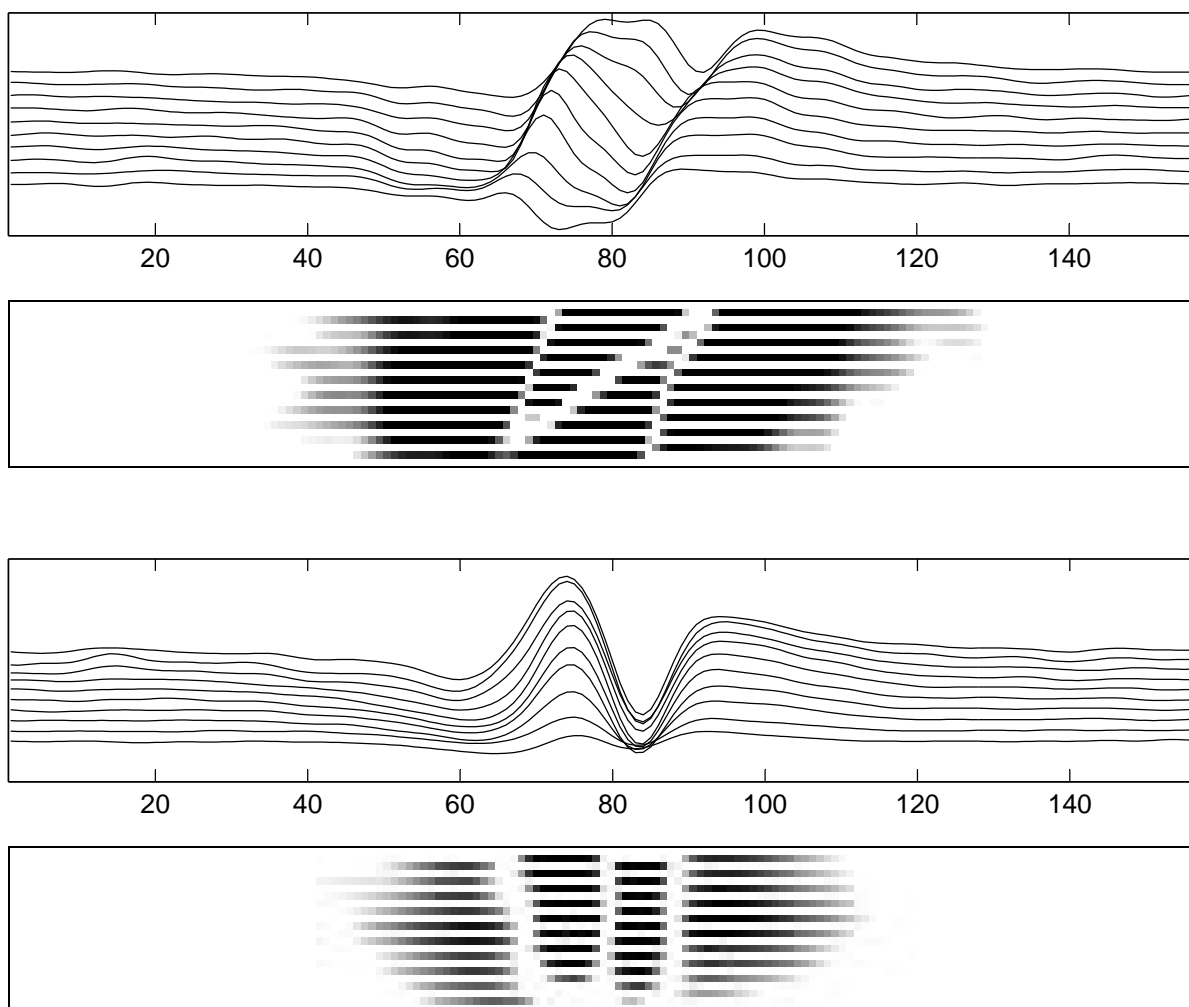


Рис. 2. Скошенный рельсовый стык и прямой рельсовый стык

Fig. 2. Two types of rail bolted joints

сигналов (в каждом канале свой уровень), в качестве опорной/исходной величины. Полученное значение проецируется на шкалу 0–255 простым умножением на 10 и присваиванием 255 всем элементам, которые превосходят это число.

Построчное сжатие. Заключительный шаг – построчное сжатие графического образа отметки до размера 20 строк на 39 столбцов. Для этого в каждой строке из каждых идущих подряд четырех элементов (по направлению от концов строки к середине) выбирается максимальный, который и будет являться элементом в новой битовой матрице. Единственное исключение – центральные элементы строки. В этом случае максимальное значение выбирается из пяти центральных элементов.

Ниже приведена функция на языке Python 3, формирующая графический образ отметки в виде bmp-матрицы размером 20 строк на 157 столбцов в градации серого цвета. В этой функции `EC[0:49999,0:9]` – текущий фрагмент анализируемых исходных вихретоковых данных, т. е. фрагмент дефектограммы длиной 50 метров

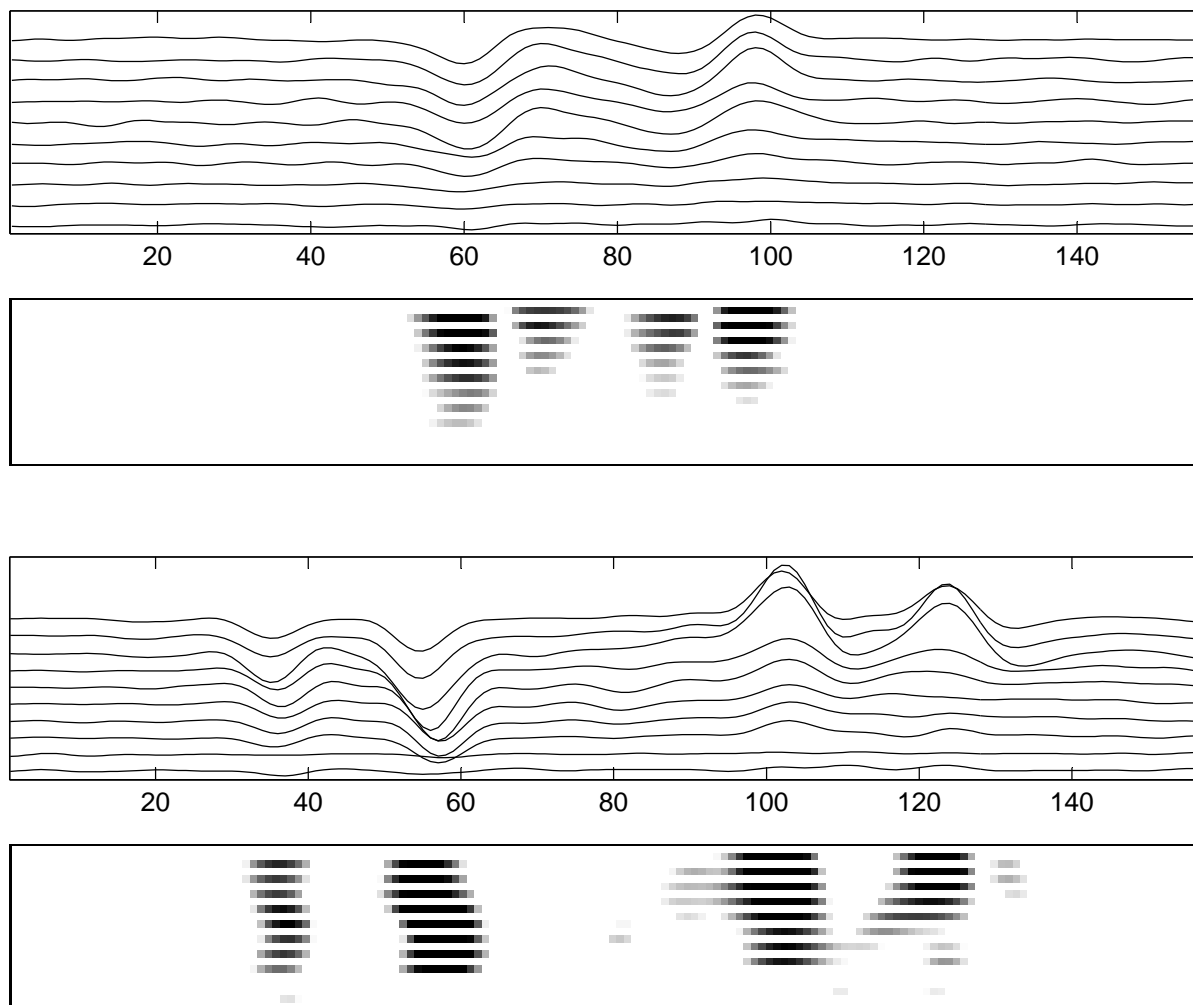


Рис. 3. Электродуговая и алюмотермитная сварки рельсов
 Fig. 3. Flash butt weld and aluminothermic weld

с данными от 10 каналов, $\text{Threshold}[0:9,0:1]$ – массив пороговых значений амплитуд полезных сигналов для 10 каналов в соответствии с направлением отклонения от среднего значения, centr – координата (номер строки в ЕС) середины текущей отметки, $A[0:19,0:156]$ – искомая (изначально состоящая полностью из нулей) bmp-матрица размером 20 строк на 157 столбцов в градации серого цвета, которая возвращается в качестве результата работы функции.

```
def Image_Preparation(EC, Threshold, centr, A):
    (ec_row, ec_col) = EC.shape
    start = int(centr - 78)
    finis = int(centr + 78)
    for i in range(1, 10 + 1):
        n = int(0)
        for j in range(start, finis + 1):
            n = int(n + 1)
            if j >= 1 and j <= ec_row:
```

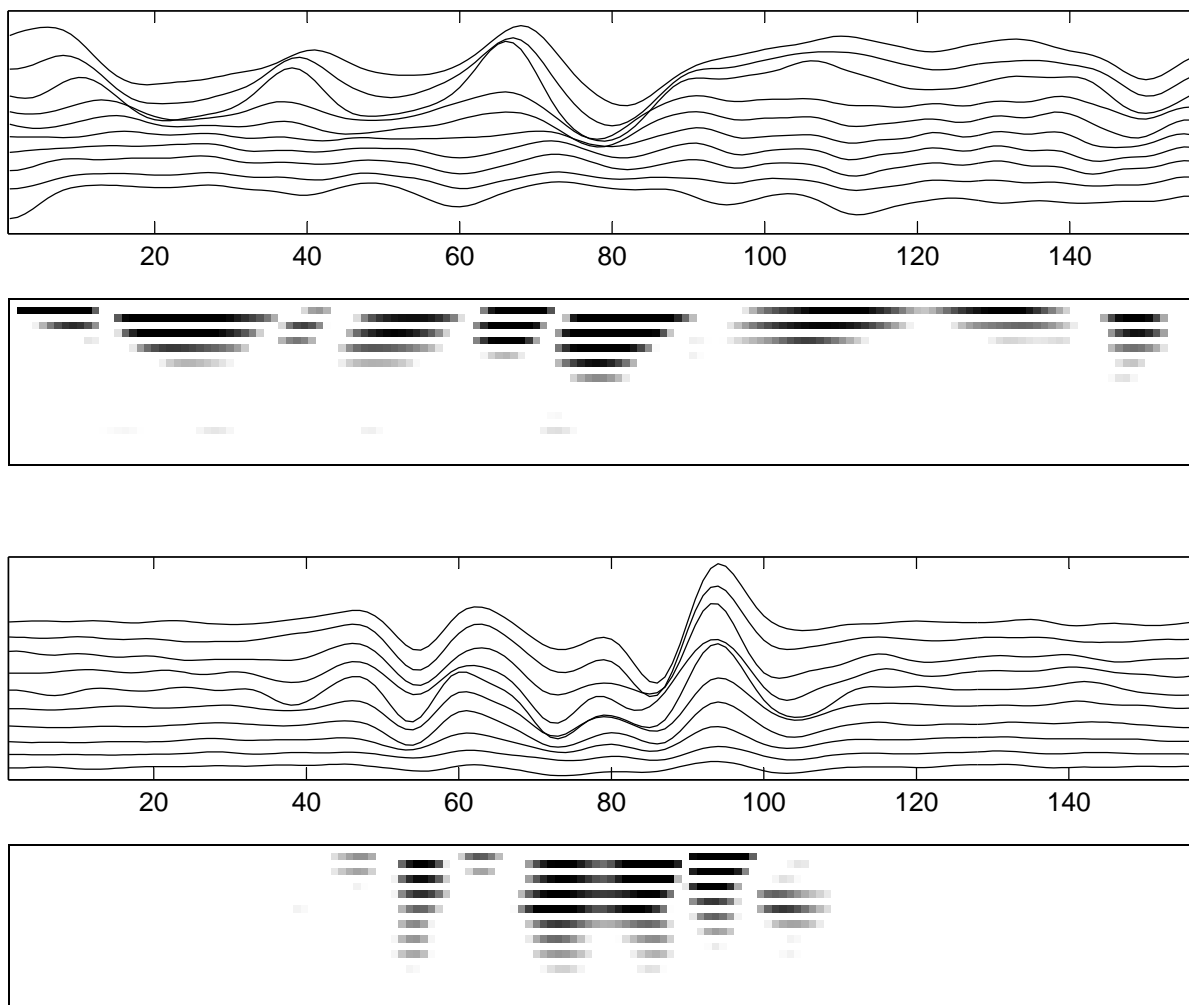


Рис. 4. Дефекты на поверхности головки рельса

Fig. 4. Defects of a rail head surface

```

if EC[j - 1, i - 1] > Threshold[i - 1, 1 - 1]:
    mu = (Threshold[i - 1, 1 - 1] + Threshold[i - 1, 2 - 1]) / 2
    rslt = (EC[j - 1, i - 1] - mu) / (Threshold[i - 1, 1 - 1] - mu)
    rslt = 20 * math.log10(rslt)
    rslt = 10 * rslt
    rslt = min(rslt, 255)
    rslt = max(rslt, 1)
    A[i * 2 - 1, n - 1] = int(rslt + 0.5)
if EC[j - 1, i - 1] < Threshold[i - 1, 2 - 1]:
    mu = (Threshold[i - 1, 1 - 1] + Threshold[i - 1, 2 - 1]) / 2
    rslt = (EC[j - 1, i - 1] - mu) / (Threshold[i - 1, 2 - 1] - mu)
    rslt = 20 * math.log10(rslt)
    rslt = 10 * rslt
    rslt = min(rslt, 255)
    rslt = max(rslt, 1)
    A[i * 2 - 1, n - 1] = int(rslt + 0.5)

```

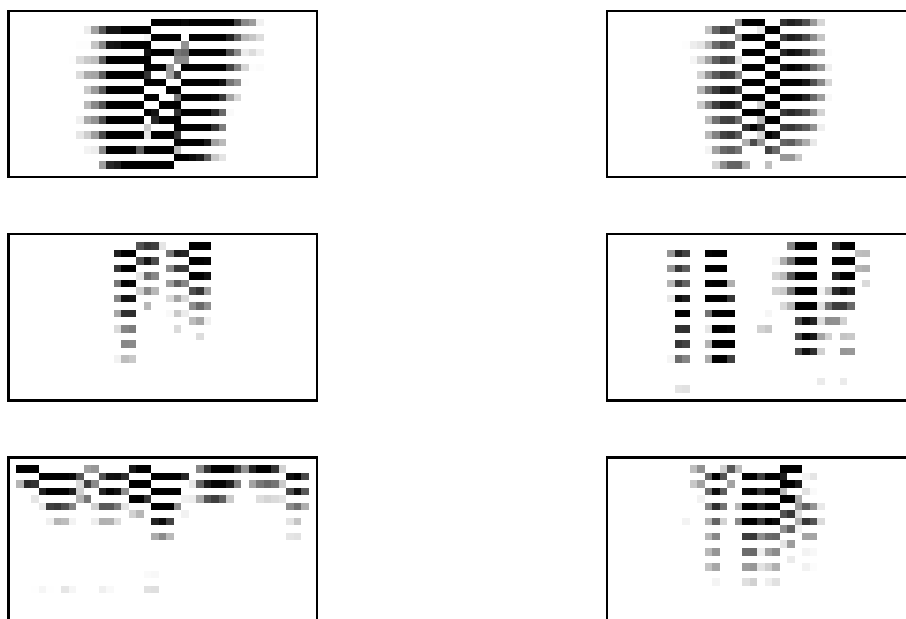


Рис. 5. Сжатые графические образы конструктивных элементов и дефектов
Fig. 5. Shrunk graphic images of structural elements and defects

Следующая функция на языке Python 3 сжимает графический образ отметки (bmp-матрицу A) до размера 20 строк на 39 столбцов. Происходит подготовка итоговой матрицы B (сжатого в четыре раза по строкам графического образа отметки, полученного на предыдущей стадии).

```
def Image_Compression(A, B):
    for i in range(1, 20 + 1):
        mxA = 0
        mxA = max(mxA, A[i - 1, 77 - 1])
        mxA = max(mxA, A[i - 1, 78 - 1])
        mxA = max(mxA, A[i - 1, 79 - 1])
        mxA = max(mxA, A[i - 1, 80 - 1])
        mxA = max(mxA, A[i - 1, 81 - 1])
        B[i - 1, 20 - 1, 1 - 1] = int(mxA)
    for j in range(1, 19 + 1):
        mxA = 0
        mxA = max(mxA, A[i - 1, j * 4 - 3 - 1])
        mxA = max(mxA, A[i - 1, j * 4 - 2 - 1])
        mxA = max(mxA, A[i - 1, j * 4 - 1 - 1])
        mxA = max(mxA, A[i - 1, j * 4 - 0 - 1])
        B[i - 1, j - 1, 1 - 1] = int(mxA)
    mxA = 0
    mxA = max(mxA, A[i - 1, (40 - j) * 4 - 2 - 1])
    mxA = max(mxA, A[i - 1, (40 - j) * 4 - 1 - 1])
    mxA = max(mxA, A[i - 1, (40 - j) * 4 - 0 - 1])
    mxA = max(mxA, A[i - 1, (40 - j) * 4 + 1 - 1])
    B[i - 1, (40 - j) - 1, 1 - 1] = int(mxA)
```

На рис. 2, 3, 4 представлены примеры скошенного и прямого болтовых рельсовых стыков, электроконтактной и алюмотермитной сварок рельсов и дефектов поверхности головки рельса в виде вихретоковых данных (фрагментов вихретоковых дефектограмм) и построенных графических образов (bmp-матриц в градации серого цвета) соответственно. На этих рисунках в графических образах отметок интенсивность серого цвета возрастает с увеличением значения элемента bmp-матрицы, т. е. от 0 (чистый белый цвет) до 255 (полный черный цвет). Рис. 5 содержит сжатые построчно (в четыре раза) графические образы приведенных конструктивных элементов и дефектов.

2. Построение и применение нейронной сети

Подготовка данных для обучения нейронной сети. Для обучения нейронной сети был подготовлен набор данных, состоящий из 48028 графических образов (файлов в формате bmp) размером 20 строк на 39 столбцов в градации серого цвета. Этот набор данных содержит в себе следующие три выборки, каждая из которых разбита на четыре класса – три класса файлов для конструктивных элементов и один для условных дефектов. Обучающая выборка (training set) используется непосредственно для обучения нейронной сети. С помощью проверочной выборки (validation set) в процессе обучения сети осуществляется оценка его качества. Итоговая оценка качества обучения сети после завершения процесса обучения проводится на тестовой выборке (test set).

В подготовленном наборе данных наряду с bmp-файлами, которые были созданы по реальным образам конструктивных элементов и условных дефектов, представленным на вихретоковых дефектограммах, содержатся также и искусственные образцы, сгенерированные на основе реальных по принципу симметрии. При построении искусственного образца в графическом образе реальной отметки (bmp-матрице) меняются местами четные и нечетные строки соответствующих каналов, а их данные (элементы строк) переписываются в обратном порядке.

В обучающей выборке находятся 16287 реальных и 8323 сгенерированных образца конструктивных элементов трех классов и 11417 реальных образцов условных дефектов. Проверочная выборка содержит 4098 сгенерированных образцов конструктивных элементов трех классов и 1903 реальных образца условных дефектов. Тестовая выборка составлена из 4098 сгенерированных образцов конструктивных элементов трех классов и 1902 реальных образцов условных дефектов. Таким образом, проверочная и тестовые выборки занимают лишь по 12,5% от общего числа подготовленных для обучения графических файлов. А все входящие в них образцы конструктивных элементов являются сгенерированными. Наличие в обучающей выборке почти всех реальных образцов связано с необходимостью «запоминания» нейронной сетью всех подтвержденных экспертами образов конструктивных элементов, которые были найдены на реальных вихретоковых дефектограммах.

Создание нейронной сети. Следующий код на языке Python 3 с опорой на библиотеки TensorFlow и Keras позволяет сформировать нейронную сеть, состоящую из трех слоев, расположенных последовательно друг за другом. Входной слой содержит 780 входных нейронов, внутренний слой имеет 800 нейронов и выходной слой

состоит всего лишь из 4-х нейронов (по числу классов распознавания). Все нейроны внутреннего слоя соединяются со всеми нейронами входного слоя, а все нейроны выходного слоя – со всеми нейронами внутреннего слоя. Начальные значения весовых коэффициентов нейронов назначаются случайным образом в соответствии с нормальным распределением вероятностей. Функция активации нейронов внутреннего слоя – ReLu (Rectified Linear Unit $f(x) = \max(0, x)$). В качестве функции активации нейронов выходного слоя используется нормализованная экспоненциальная функция SoftMax, при которой сумма значений всех выходных нейронов равняется 1. Эта функция позволяет выдавать результат в виде набора вероятностей принадлежности распознаваемого объекта тому или иному классу (искомый класс определяется по большему значению из набора).

Слой регуляризации служит для борьбы с проблемой переобучения сети (адаптации весов только к обучающей выборке), который выключает нейроны внутреннего слоя из процесса обучения с вероятностью 0,25 (на один нейрон) при каждом новом объекте, подаваемом на сеть для обучения.

Метод обучения сети – метод стохастического градиентного спуска в модификации Adam. Мера ошибки при обучении – перекрестная энтропия (для четырех категорий). Метрика оценки результатов обучения – точность (ассигасу, отношение количества правильно классифицированных объектов к общему числу классифицируемых объектов).

```
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dropout, Flatten, Dense
img_width, img_height = 39, 20 # Размеры изображения
input_shape = (img_height, img_width, 1) # Размерность тензора
model = Sequential() # создание последовательной модели нейронной сети
model.add(Flatten(input_shape=input_shape)) # добавление входного слоя нейронов
# добавление полносвязного внутреннего слоя нейронов
model.add(Dense(800, input_dim=780, activation='relu', kernel_initializer='normal'))
model.add(Dropout(0.25)) # добавление слоя регуляризации
# добавление выходного слоя из четырех нейронов
model.add(Dense(4, activation='softmax', kernel_initializer='normal'))
# компиляция модели
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Обучение нейронной сети. Ниже приводится код на языке Python 3, с помощью которого происходит обучение нейронной сети для распознавания конструктивных элементов и условных дефектов. Данные для обучения, проверки и тестирования подаются порциями по 32 изображения из соответствующих каталогов (для этого задействованы генераторы изображений библиотеки Keras с размером мини-выборки, равным 32). При этом осуществляется нормализация данных – значения элементов bmp-матриц делятся на 255 (приводятся к вещественным числам от 0 до 1). Количество эпох обучения – 400.

Время, затраченное на обучение описанной модели нейронной сети, составило около 3-х часов. Обучение сети проводилось на компьютере с центральным процессором Intel Core i7-4770K CPU 3.50 GHz и оперативной памятью 8 GB. В результате обучения была достигнута следующая точность (ассигасу):

- на данных для обучения – 99,83%,
- на данных для проверки – 99,45%,
- на тестовых данных – 99,23%.

```
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
epochs = 400      # количество эпох обучения
batch_size = 32   # размер мини-выборки
nb_train_samples = (8112 + 11417 + 6640 + 9858) # количество изображений для обучения
nb_validation_samples = (1352 + 1903 + 1104 + 1642) # кол-во изображений для проверки
nb_test_samples = (1352 + 1902 + 1104 + 1642) # кол-во изображений для тестирования
# создание генератора изображений с нормализацией данных
datagen = ImageDataGenerator(rescale=1. / 255)
# генерация данных для обучения (на основе изображений из каталога)
train_generator = datagen.flow_from_directory(
    train_dir, # каталог с данными для обучения
    target_size=(img_height, img_width), # размер изображения
    color_mode='grayscale', # изображение в градации серого цвета
    batch_size=batch_size, # размер мини-выборки
    class_mode='categorical')
# генерация данных для проверки (на основе изображений из каталога)
val_generator = datagen.flow_from_directory(
    val_dir, # каталог с данными для проверки
    target_size=(img_height, img_width), # размер изображения
    color_mode='grayscale', # изображение в градации серого цвета
    batch_size=batch_size, # размер мини-выборки
    class_mode='categorical')
# генерация данных для тестирования (на основе изображений из каталога)
test_generator = datagen.flow_from_directory(
    test_dir, # каталог с данными для тестирования
    target_size=(img_height, img_width), # размер изображения
    color_mode='grayscale', # изображение в градации серого цвета
    batch_size=batch_size, # размер мини-выборки
    class_mode='categorical')
# обучение модели нейронной сети с помощью генераторов данных
model.fit_generator(
    train_generator,
    steps_per_epoch=nb_train_samples // batch_size,
    epochs=epochs, # количество эпох обучения
    validation_data=val_generator,
    validation_steps=nb_validation_samples // batch_size)
# оценка качества обучения на тестовых данных
scores = model.evaluate_generator(test_generator, nb_test_samples // batch_size)
```

Распознавание графических образов конструктивных элементов. Отметим, что в каждом из каталогов `train_dir`, `val_dir` и `test_dir` данные (картинки) четырех классов находились в соответствующих папках: для графических образов алюмотермитной сварки была отведена папка `Alt`, для образов электроконтактной сварки – папка `Wfb`, для болтовых стыков – `Gap`, а для условных дефектов – `Def`. Поэтому при обучении нейронной сети классы образцов получили имена в соответствии с названиями этих папок.

С применением следующего кода на языке Python 3 происходит распознавание графических образов четырех классов с помощью обученной нейронной сети. На вход сети подается подготовленный и нормализованный графический образ `V` найденной на дефектограмме отметки, выходом является название класса, к которому принадлежит исследуемый образ. Нейронная сеть выдает набор значений вероятностей принадлежности объекта к соответствующему классу. Класс объекта выбирается по максимальному значению из этого набора.

```

import numpy
classes = ['alt', 'def', 'gap', 'wfb']
B /= 255 # нормализация матрицы B
B = numpy.expand_dims(B, axis=0) # переход к 'линейному' массиву данных
preds = model.predict(B) # классификация входного объекта
if classes[preds.argmax()] == 'def': # расшифровка результата классификации
    mark_result = 'условный дефект'
if classes[preds.argmax()] == 'wfb': # расшифровка результата классификации
    mark_result = 'электроконтактная сварка'
if classes[preds.argmax()] == 'alt': # расшифровка результата классификации
    mark_result = 'алюмотермитная сварка'
if classes[preds.argmax()] == 'gap': # расшифровка результата классификации
    mark_result = 'болтовой стык'

```

Заключение. По результатам обучения можно сделать вывод, что выбранная конфигурация модели нейронной сети является удачной и вполне достаточной для решения поставленной задачи распознавания конструктивных элементов. Построение глубоких (сверточных) нейронных сетей не требуется, так как, с одной стороны, удалось провести явное сжатие исходного графического образа отметки до относительно небольшого размера без существенного влияния на результат распознавания отметок, с другой стороны, образы конструктивных элементов имеют жесткую простую структуру и характерный рисунок, четко выраженный для соответствующего класса. При этом ни масштабирования, ни поворота, ни сдвигов ключевых признаков на изображении не происходит, что делает простую последовательную полносвязную 3-слойную модель нейронной сети достаточной для решения данной задачи распознавания.

Список литературы / References

- [1] Кузьмин Е. В., Горбунов О. Е., Плотников П. О., Тюкин В. А., “Об определении уровня полезных сигналов при расшифровке магнитных и вихретоковых дефектограмм”, *Модел. и анализ информ. систем*, **24**:6 (2017), 760–771; [Kuzmin E. V., Gorbunov O. E., Plotnikov P. O., Tyukin V. A., “On Finding a Threshold of Useful Signals in the Analysis of Magnetic and Eddy Current Defectograms”, *Modeling and Analysis of Information Systems*, **24**:6 (2017), 760–771, (in Russian).]
- [2] Кузьмин Е. В., Горбунов О. Е., Плотников П. О., Тюкин В. А., “Эффективный алгоритм определения уровня полезных сигналов при расшифровке магнитных и вихретоковых дефектограмм”, *Модел. и анализ информ. систем*, **25**:4 (2018), 382–387; [Kuzmin E. V., Gorbunov O. E., Plotnikov P. O., Tyukin V. A., “An Efficient Algorithm for Finding a Threshold of Useful Signals in the Analysis of Magnetic and Eddy Current Defectograms”, *Modeling and Analysis of Information Systems*, **25**:4 (2018), 382–387, (in Russian).]
- [3] Марков А. А., Кузнецова Е. А., *Дефектоскопия рельсов. Формирование и анализ сигналов. Кн. 1. Основы*, КультИнформПресс, СПб., 2010; [Markov A. A., Kuznetsova E. A., *Rails flaw detection. Formation and analysis of signals. Book 1. Principles*, KultInformPress, St. Petersburg, 2010, (in Russian).]
- [4] Марков А. А., Кузнецова Е. А., *Дефектоскопия рельсов. Формирование и анализ сигналов. Кн. 2. Расшифровка дефектограмм*, Ультра Принт, СПб., 2014; [Markov A. A., Kuznetsova E. A., *Rails flaw detection. Formation and analysis of signals. Book 2. Data interpretation*, Ultra Print, St. Petersburg, 2014, (in Russian).]
- [5] Тарабрин В. Ф., Зверев А. В., Горбунов О. Е., Кузьмин Е. В., “О фильтрации данных при автоматической расшифровке дефектограмм АПК «АСТРА»”, *В мире неразрушающего контроля*, **64**:2 (2014), 5–9; [Tarabrin V. F., Zverev A. V., Gorbunov O. E.,

Kuzmin E. V., "About Data Filtration of the Defectogram Automatic Interpretation by Hardware and Software Complex "ASTRA"", *NDT World*, **64**:2 (2014), 5–9, (in Russian).]

- [6] Simard P. Y., Steinkraus D., Platt J. C., "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis", *7th International Conference on Document Analysis and Recognition (ICDAR-2003)*, 2-Volume Set, (3–6 August 2003, Edinburgh, Scotland, UK), IEEE Computer Society, 2003, 958–962.
- [7] Goodfellow I., Bengio Y., Courville A., *Deep Learning*, MIT Press, 2016.
- [8] Chollet F., *Deep Learning with Python*, Manning Publications Co., 2018.
- [9] TensorFlow <https://www.tensorflow.org/>.

Kuzmin E.V., Gorbunov O.E., Plotnikov P.O., Tyukin V.A., Bashkin V.A., "Application of Neural Networks for Recognizing Rail Structural Elements in Magnetic and Eddy Current Defectograms", *Modeling and Analysis of Information Systems*, **25**:6 (2018), 667–679.

DOI: 10.18255/1818-1015-2018-6-667-679

Abstract. To ensure traffic safety of railway transport, non-destructive test of rails is regularly carried out by using various approaches and methods, including magnetic and eddy current flaw detection methods. An automatic analysis of large data sets (defectograms) that come from the corresponding equipment is an actual problem. The analysis means a process of determining the presence of defective sections along with identifying structural elements of railway tracks on defectograms. This article is devoted to the problem of recognition of rail structural element images in magnetic and eddy current defectograms. Three classes of rail track structural elements are considered: 1) a bolted joint with straight or beveled connection of rails, 2) a butt weld of rails, and 3) an aluminothermic weld of rails. Images that cannot be assigned to these three classes are conditionally considered as defects and are placed in a separate fourth class. For image recognition of structural elements in defectograms a neural network is applied. The neural network is implemented by using the open library TensorFlow. To this purpose each selected (picked out) area of a defectogram is converted into a graphic image in a grayscale with size of 20 x 39 pixels.

Keywords: nondestructive testing, magnetic and eddy current testing, rail flaw detection, automated analysis of defectograms, neural networks

On the authors:

Egor V. Kuzmin, orcid.org/0000-0003-0500-306X, doctor of science, associate professor, P.G. Demidov Yaroslavl State University, 14 Sovetskaya str., Yaroslavl, 150003 Russia, e-mail: kuzmin@uniyar.ac.ru, kuzminev@nddlab.com

Oleg E. Gorbunov, orcid.org/0000-0001-6274-9971, PhD, general director, Center of Innovative Programming, NDDLlab, 144 Soyuznaya str., Yaroslavl, 150008, Russia, e-mail: gorbunovoe@nddlab.com

Petr O. Plotnikov, orcid.org/0000-0001-5687-7969, production engineer, Center of Innovative Programming, NDDLlab, 144 Soyuznaya str., Yaroslavl, 150008, Russia, e-mail: plotnikovpo@nddlab.com

Vadim A. Tyukin, orcid.org/0000-0001-9149-7435, head of software development, Center of Innovative Programming, NDDLlab, 144 Soyuznaya str., Yaroslavl, 150008, Russia, e-mail: tyukinva@nddlab.com

Vladimir A. Bashkin, orcid.org/0000-0002-2534-1026, doctor of science, associate professor, P.G. Demidov Yaroslavl State University, 14 Sovetskaya str., Yaroslavl, 150003 Russia, e-mail: bashkinva@nddlab.com