

Application of Convolutional Neural Networks for Recognizing Long Structural Elements of Rails in Eddy-Current Defectograms

E. V. Kuzmin¹, O. E. Gorbunov², P. O. Plotnikov², V. A. Tyukin², V. A. Bashkin¹

DOI: [10.18255/1818-1015-2020-3-316-329](https://doi.org/10.18255/1818-1015-2020-3-316-329)

¹P. G. Demidov Yaroslavl State University, 14 Sovetskaya str., Yaroslavl 150003, Russia.

²Center of Innovative Programming, NDDLlab, 144 Soyuznaya str., Yaroslavl, 150008, Russia.

MSC2020: 68T07

Research article

Full text in Russian

Received July 20, 2020

After revision August 3, 2020

Accepted September 9, 2020

To ensure traffic safety of railway transport, non-destructive test of rails is regularly carried out by using various approaches and methods, including 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 in defectograms. This article is devoted to the problem of recognizing images of long structural elements of rails in eddy-current defectograms. Two classes of rail track structural elements are considered: 1) rolling stock axle counters, 2) rail crossings. Long marks that cannot be assigned to these two classes are conditionally considered as defects and are placed in a separate third class. For image recognition of structural elements in defectograms a convolutional 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 30 x 140 points.

Keywords: nondestructive testing; eddy-current testing; rail flaw detection; automated analysis of defectograms; neural networks.

INFORMATION ABOUT THE AUTHORS

Egor V. Kuzmin correspondence author	orcid.org/0000-0003-0500-306X . E-mail: kuzmin@uniyar.ac.ru , kuzminev@nddlab.com Professor, Doctor of Science.
Oleg E. Gorbunov	orcid.org/0000-0001-6274-9971 . E-mail: gorbunovoe@nddlab.com General Director, PhD.
Petr O. Plotnikov	orcid.org/0000-0001-5687-7969 . E-mail: plotnikovpo@nddlab.com Production Engineer.
Vadim A. Tyukin	orcid.org/0000-0001-9149-7435 . E-mail: tyukinva@nddlab.com Head of Software Development.
Vladimir A. Bashkin	orcid.org/0000-0002-2534-1026 . E-mail: bashkinva@nddlab.com Professor, Doctor of Science.

For citation: E. V. Kuzmin, O. E. Gorbunov, P. O. Plotnikov, V. A. Tyukin, and V. A. Bashkin, "Application of Convolutional Neural Networks for Recognizing Long Structural Elements of Rails in Eddy-Current Defectograms", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 316-329, 2020.

Применение свёрточных нейронных сетей для распознавания длинных конструктивных элементов рельсов на вихретоковых дефектограммах

Е. В. Кузьмин¹, О. Е. Горбунов², П. О. Плотников², В. А. Тюкин², В. А. Башкин¹

DOI: [10.18255/1818-1015-2020-3-316-329](https://doi.org/10.18255/1818-1015-2020-3-316-329)

¹Ярославский государственный университет им. П. Г. Демидова, ул. Советская, 14, г. Ярославль, 150003 Россия.

²ООО «Центр инновационного программирования», NDDLlab, ул. Союзная, 144, г. Ярославль, 150008 Россия.

УДК 004.032.26

Научная статья

Полный текст на русском языке

Получена 20 июля 2020 г.

После доработки 3 августа 2020 г.

Принята к публикации 9 сентября 2020 г.

Для обеспечения безопасности движения на железнодорожном транспорте регулярно проводится неразрушающий контроль рельсов с применением различных подходов и методов, включая методы 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 длинных конструктивных элементов железнодорожных рельсов по дефектограммам многоканальных вихретоковых дефектоскопов. Рассматриваются два класса конструктивных элементов рельсового пути: 1) счётчики осей подвижного состава, 2) пересечения рельсовых путей. Длинные отметки, которые не могут быть отнесены к этим двум классам, условно считаются дефектами и выносятся в отдельный третий класс. Для распознавания образов конструктивных элементов на дефектограммах применяется свёрточная 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 30 на 140 точек.

Ключевые слова: неразрушающий контроль рельсов; вихретоковая дефектоскопия; обнаружение дефектов; автоматический анализ дефектограмм; нейронные сети.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Егор Владимирович Кузьмин
автор для корреспонденции

orcid.org/0000-0003-0500-306X.
E-mail: kuzmin@uniyar.ac.ru, kuzminev@nddlab.com
профессор, доктор физ.-мат. наук.

Олег Евгеньевич Горбунов

orcid.org/0000-0001-6274-9971. E-mail: gorbunovoe@nddlab.com
генеральный директор, канд. физ.-мат. наук.

Петр Олегович Плотников

orcid.org/0000-0001-5687-7969. E-mail: plotnikovpo@nddlab.com
инженер-технолог.

Вадим Александрович Тюкин

orcid.org/0000-0001-9149-7435. E-mail: tyukinva@nddlab.com
руков. сектора разработки.

Владимир Анатольевич Башкин

orcid.org/0000-0002-2534-1026. E-mail: bashkinva@nddlab.com
профессор, доктор физ.-мат. наук.

Для цитирования: Е. В. Кузьмин, О. Е. Горбунов, П. О. Плотников, В. А. Тюкин, and В. А. Башкин, "Application of Convolutional Neural Networks for Recognizing Long Structural Elements of Rails in Eddy-Current Defectograms", *Modeling and analysis of information systems*, vol. 27, no. 3, pp. 316-329, 2020.

Введение

Для обеспечения безопасности движения на железнодорожном транспорте регулярно проводится неразрушающий контроль рельсов с применением различных подходов и методов, включая методы вихретоковой дефектоскопии. Актуальной задачей является автоматический анализ [1–3] 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 является логическим продолжением работ [4–6], посвященных задаче распознавания образов дефектов и конструктивных элементов железнодорожных рельсов по дефектограммам магнитных и вихретоковых дефектоскопов. Для классификации выделенных конструктивных элементов применяются нейронные сети [7, 8].

В этой работе рассматривается обобщение реального устройства в виде 10-разрядного вихретокового дефектоскопа с 15 каналами данных (на один рельс). Каналы данных соответствуют физическим датчикам, которые последовательно располагаются на поверхности рельса перпендикулярно направлению движения дефектоскопа. Значения амплитуд сигналов каждого канала регистрируются дефектоскопом в виде целых чисел от -512 до 511 (в 10-разрядном случае).

Интерес представляют амплитуды только полезных сигналов. Сигнал считается полезным (и подлежит дальнейшему анализу), если отклонение от нуля как минимум в два раза превосходит пороговый уровень шума рельсов. Отрицательное и положительное пороговые значения уровня шума определяются отдельно для каждого канала данных по алгоритму из статей [5, 6].

Найденные полезные сигналы всех каналов группируются в отметки, которые подлежат дальнейшей классификации. Ранее в статье [4] решалась задача распознавания небольших конструктивных элементов (длиной до 157 мм) следующих трёх типов: 1) болтовой стык с прямым или скошенным соединением рельсов, 2) электроконтактная сварка рельсов и 3) алюминотермитная сварка рельсов.

Данная статья посвящена задаче распознавания записей длинных конструктивных элементов рельсового пути. Выделяются два класса объектов для распознавания: 1) записи мест установки счётчиков осей подвижного состава («axle counter»), 2) записи мест пересечений рельсовых путей («crossing»). Размер записей этих объектов может варьироваться от 420 мм до 3220 мм в длину.

Длинные отметки, которые не могут быть отнесены к этим двум классам, условно считаются дефектами и выносятся в отдельный (третий) класс отметок.

Для распознавания образов конструктивных элементов на вихретоковых дефектограммах была задействована свёрточная нейронная сеть, реализованная в рамках открытой библиотеки TensorFlow (в состав которой входит библиотека Keras) [9]. С этой целью каждая выделенная длинная отметка преобразуется в графический образ в градации серого цвета размером 30 на 140 точек.

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

1. Формирование образа отметки

При автоматическом анализе дефектограммы обычно разбиваются на фрагменты, которые, например, могут соответствовать 50-метровым участкам рельсового пути, т. е. при снятии показаний 15-канального дефектоскопа с каждого миллиметра пути блок анализа представляет собой матрицу размером 15 строк на 50000 столбцов, где элемент матрицы — это значение амплитуды сигнала соответствующего канала данных. Пример графического представления 50-метрового блока анализа приведён на рис. 1 (15-канальные данные с 50-метрового участка одного рельса). В этом фрагменте данных присутствуют записи места пересечения рельсовых путей, места расположения счётчика

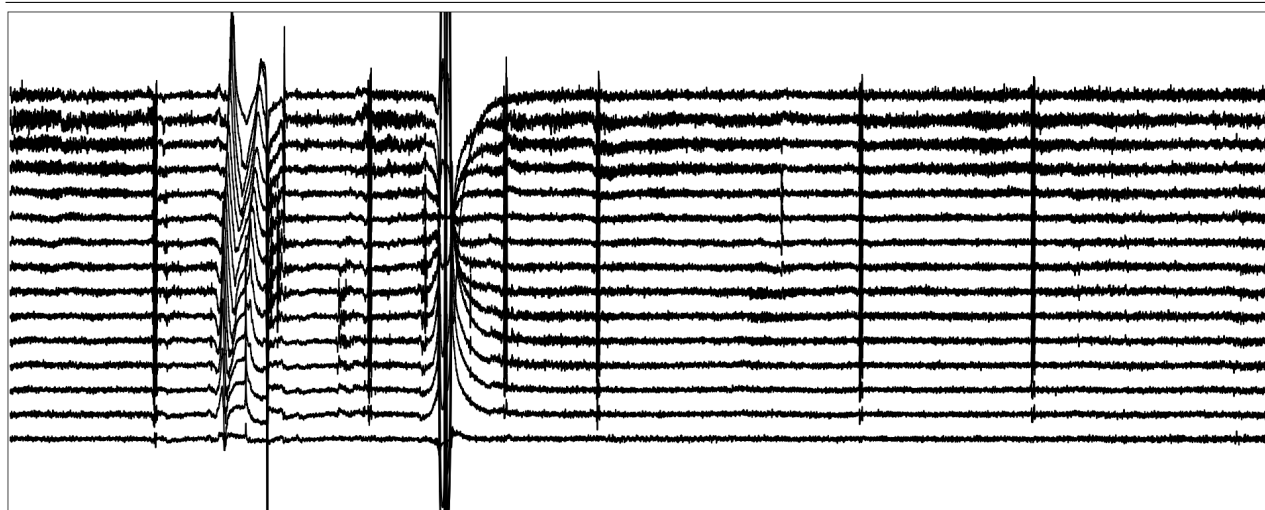


Fig. 1. Eddy-current data (15 channels).
An example of an analysis block (50 m)

Рис. 1. Вихретоковые данные по 15 каналам.
Пример 50-метрового блока анализа

осей подвижного состава, а также записи шести сварных алюминотермитных рельсовых стыков. Более крупно место пересечения путей показано на рис. 2 (см. верхний пример). Типичный вид записи счётчика осей представлен на рис. 3. Запись сварного алюминотермитного стыка можно найти в статье [4] или в несколько искажённом виде (из-за наличия концентрации напряжения в стыке «pumping») на рис. 4.

В данной статье рассматривается задача классификации (распознавания) длинных отметок размером от 420 мм до 3220 мм, которые характеризуются продолжительными участками (от 100 мм) амплитудного отклонения от порогового значения. Наличие таких участков позволяет построчно сжимать графический образ отметки в 3–23 раза без потери её характерного рисунка. Главным образом это касается записей пересечений рельсовых путей, длина которых (и, соответственно, продолжительность участков сильного амплитудного отклонения) зависит как от угла пересечения, так и от сложности рельсовых конструкций.

После того как по полезным сигналам на дефектограмме выделена отметка в виде матрицы амплитуд размером 15 строк (по числу каналов) на n столбцов (длина отметки в мм), предполагается преобразование отметки в графический образ размером 30 на n точек в градации серого цвета с последующим построчным сжатием образа до размера 30 на 140 точек.

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 нуля и превосходят положительный пороговый уровень полезных сигналов. Остальные значения в нечетной строке устанавливаются в ноль. В четной строке остаются абсолютные значения только тех амплитуд, которые меньше отрицательного порогового уровня полезных сигналов. Остальные значения в четной строке также устанавливаются в ноль. В результате имеем матрицу размером 30 строк на n столбцов амплитуд полезных сигналов с их дифференциацией и последующим построчным размещением по направлению отклонения от нуля.

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 абсолютным значением соответствующего уровня полезных сигналов 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 30 строк на 140 столбцов.

Ниже приведена функция на языке Python 3, формирующая графический образ отметки в виде bmp-матрицы размером 30 строк на 140 столбцов в градации серого цвета. В этой функции EC[0:49999,0:14] – это матрица анализируемых исходных вихретоковых данных, т. е. фрагмент дефектограммы длиной 50 метров с данными от 15 каналов, Threshold[0:14] – список положительных и отрицательных пороговых значений амплитуд полезных сигналов для 15 каналов, start и end – координаты начала и конца текущей отметки, B[0:29,0:139] – искомая (изначально состоящая полностью из нулей) bmp-матрица размером 30 строк на 140 столбцов в градации серого цвета, которая возвращается в качестве результата работы функции.

```
def Image_Preparation(EC, Threshold, B, start, end):
    lngth = end - start + 1
    divrslt = lngth / 140
    A = np.zeros((30, lngth))
    for j in range(0, 15):
        thplus = Threshold[j].thplus
        thmins = Threshold[j].thmins
        n = int(0)
        for i in range(start, end + 1):
            ec = EC[i, j]
            if ec > thplus:
                rslt = int(10 * 20 * math.log10(ec / thplus) + 0.5)
                rslt = max(min(rslt, 255), 1)
                A[j * 2, n] = rslt
            if ec < thmins:
                rslt = int(10 * 20 * math.log10(ec / thmins) + 0.5)
                rslt = max(min(rslt, 255), 1)
                A[j * 2 + 1, n] = rslt
            n += 1
    for j in range(0, 30):
        for i in range(0, 140):
            mx = 0
            for m in range(int(i * divrslt), int((i + 1) * divrslt)):
                if mx < A[j, m]:
                    mx = A[j, m]
            B[j, i, 0] = mx
```

На рис. 2 и 3 представлены примеры специального графического изображения в виде bmp-матриц 30 строк на 140 столбцов в градации серого цвета для вихретоковых записей мест пересечения рельсовых путей («crossing») размером 2435 мм и 1361 мм и счётчиков осей («axle counter») размером 2206 мм и 3220 мм соответственно.

Пример условного дефекта – концентрация напряжения в области сварного алюминотермитного рельсового стыка («rumping»). Специальное графическое представление записей этого вида условного дефекта размером 1243 мм и 1323 мм показано на рис. 4.

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).

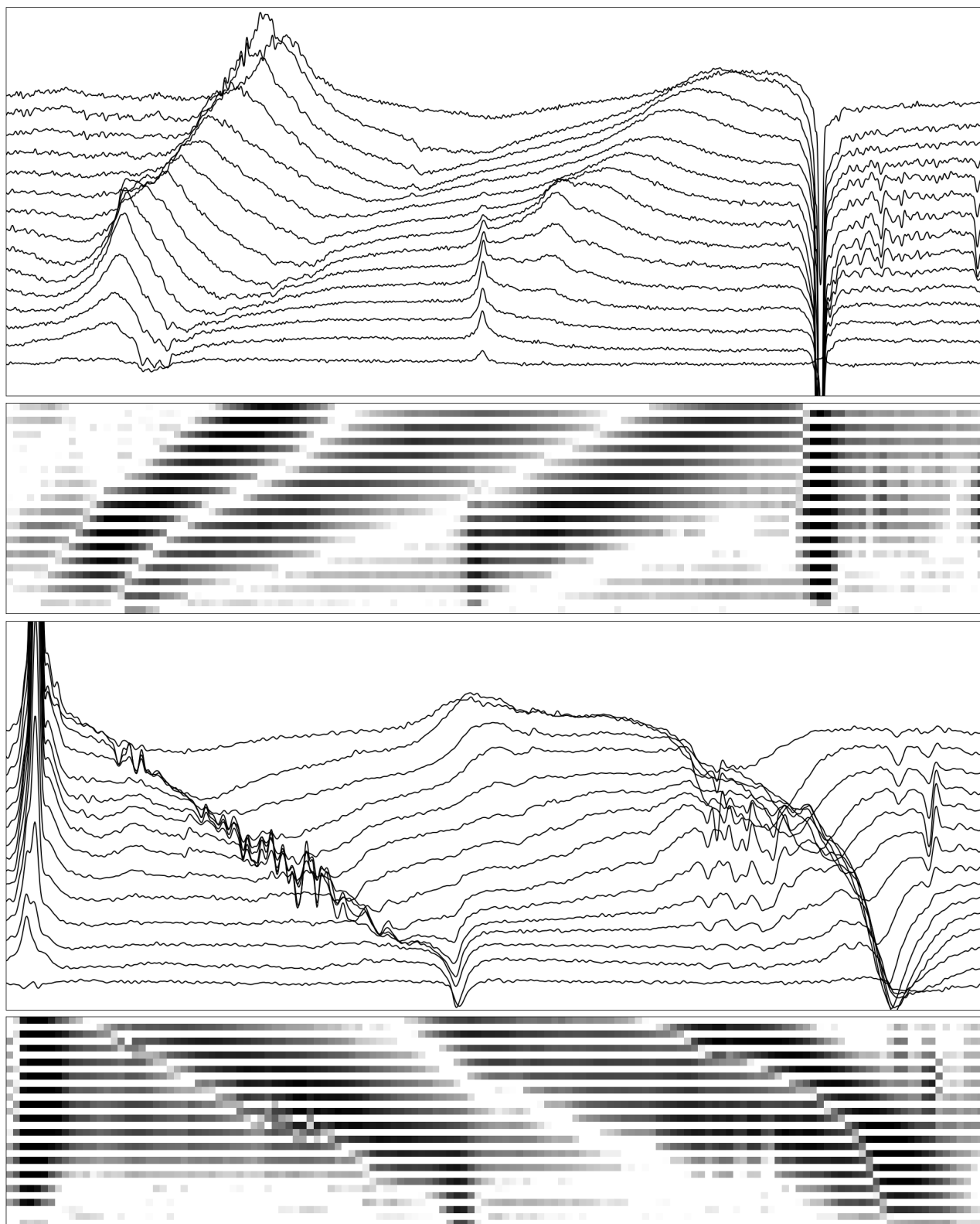


Fig. 2. Eddy-current records and its special graphic representation for crossings

Рис. 2. Вихретоковые записи пересечений путей и их специальное графическое представление

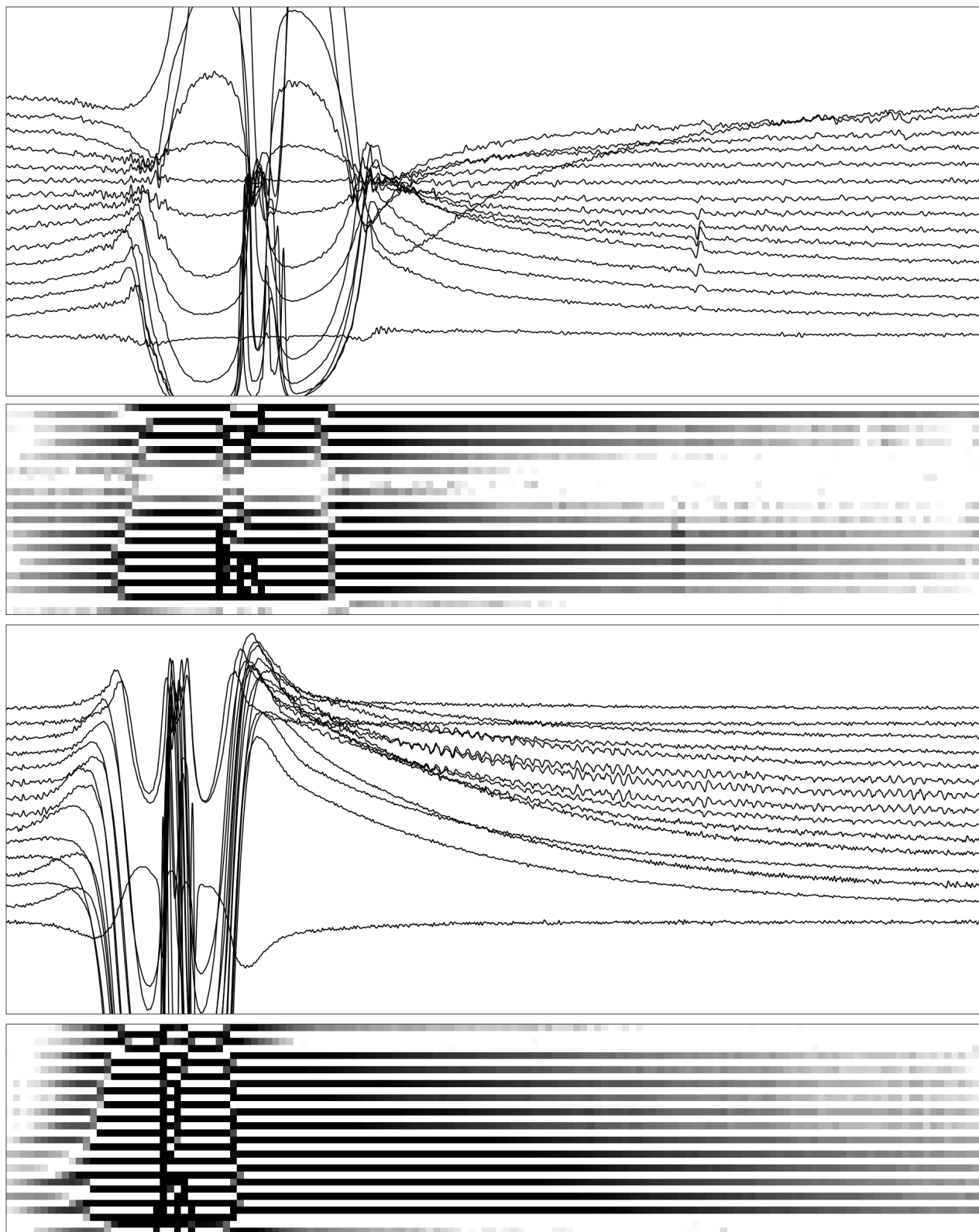


Fig. 3. Eddy-current records and its special graphic representation for axle counters

Рис. 3. Вихретоковые записи счётчиков осей и их специальное графическое представление

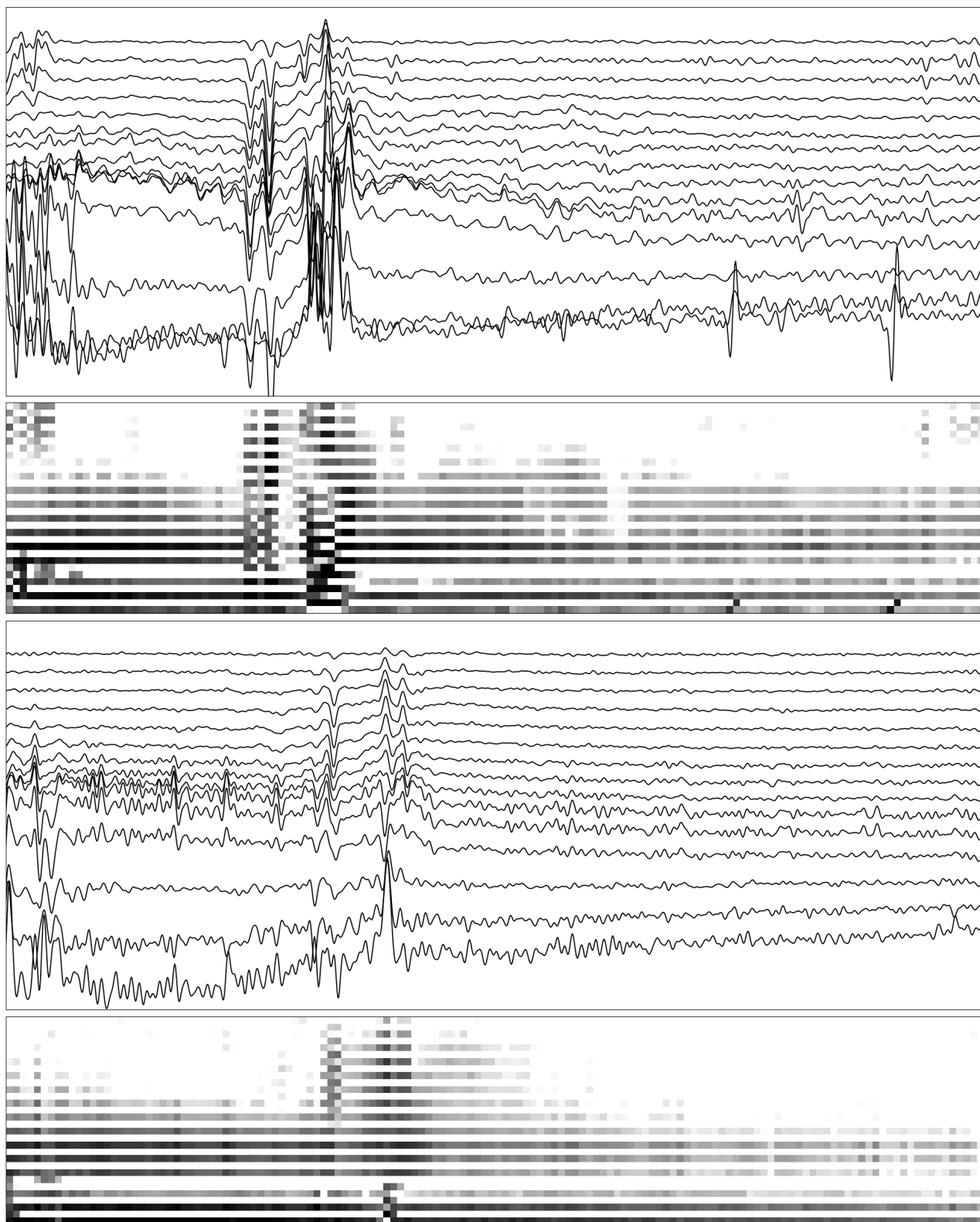


Fig. 4. Eddy-current records and its special graphic representation for pumpings

Рис. 4. Вихретоковые записи условных дефектов и их специальное графическое представление

2. Построение и применение нейронной сети

Подготовка данных для обучения нейронной сети. Для обучения нейронной сети был подготовлен набор данных, состоящий из 3720 графических образов (файлов в формате bmp) размером 30 строк на 140 столбцов в градации серого цвета. Этот набор данных содержит в себе следующие три выборки, каждая из которых разбита на три класса — два 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, представленных на вихретоковых дефектограммах, содержатся также и искусственные образцы, сгенерированные на основе реальных по принципу зеркальной симметрии. При построении искусственного образца в графическом образе реальной отметки (bmp-матрице) элементы каждой строки переписываются в обратном порядке.

В обучающей выборке находятся 1860 реальных и 930 сгенерированных образцов условных дефектов и конструктивных элементов двух классов. Проверочная выборка содержит 463 сгенерированных образца. Тестовая выборка составлена из 467 сгенерированных образцов. Таким образом, проверочная и тестовые выборки в совокупности занимают 25% от общего числа подготовленных для обучения графических файлов. А все входящие в них образцы конструктивных элементов являются сгенерированными. Наличие в обучающей выборке всех реальных образцов связано с необходимостью «запоминания» нейронной сетью всех подтвержденных экспертами образцов конструктивных элементов, которые были найдены на реальных вихретоковых дефектограммах.

Отметим, что указанное количество графических образцов значительно меньше того объёма данных, который использовался в работе [4] при обучении нейронной сети для распознавания сварок и рельсовых стыков. Это связано с тем, что на вихретоковых дефектограммах записи счётчиков осей и элементов стрелочных переводов встречаются намного реже, чем записи рельсовых стыков и алюминотермитных и электроконтактных сварок. Однако даже такое относительно небольшое число образцов позволило обучить нейронную сеть, успешно применяемую на практике.

Создание нейронной сети. Следующий код на языке Python 3 с использованием библиотек TensorFlow и Keras позволяет сформировать свёрточную нейронную сеть, состоящую из чередующихся двух слоёв подвыборки и двух слоёв свёртки, полносвязного внутреннего слоя из 32 нейронов, слоя регуляризации и полносвязного выходного слоя, состоящего всего лишь из 3 нейронов (по числу классов распознавания).

Начальные значения весовых коэффициентов нейронов назначаются случайным образом в соответствии с нормальным распределением вероятностей. Функция активации нейронов внутренних слоёв — 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 трёх категорий). Метрика оценки результатов обучения — точность (ассигасу, отношение количества правильно классифицированных объектов к общему числу классифицируемых объектов).

Построенная свёрточная нейронная сеть имеет относительно небольшой объём — во время обучения настраиваются 52139 параметров (обучаемых весов).

```
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Conv2D, MaxPooling2D
from tensorflow.python.keras.layers import Activation, Dropout, Flatten, Dense
img_width, img_height = 140, 30 # размеры изображения
input_shape = (img_height, img_width, 1) # размерность тензора
model = Sequential() # создание последовательной модели нейронной сети
# добавление слоя свёртки с размером ядра 3x3; 16 карт признаков
model.add(Conv2D(16, (3, 3), input_shape=input_shape))
model.add(Activation('relu')) # функция активации ReLU
# добавление слоя подвыборки; выбор максимального значения из квадрата 2x2
model.add(MaxPooling2D(pool_size=(2, 2)))
# добавление слоя свёртки с размером ядра 3x3; 8 карт признаков
model.add(Conv2D(8, (3, 3)))
model.add(Activation('relu')) # функция активации ReLU
# добавление слоя подвыборки; выбор максимального значения из квадрата 2x2
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten()) # добавление полносвязного внутреннего слоя,
model.add(Dense(32)) # состоящего из 32-х нейронов
model.add(Activation('relu')) # функция активации ReLU
model.add(Dropout(0.25)) # добавление слоя регуляризации
# добавление выходного слоя из трёх нейронов
model.add(Dense(3, 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 30 минут. 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 (accuracy):

- на данных для обучения — 100%,
- на данных для проверки — 96,88%,
- на тестовых данных — 96,43%.

```
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
epochs = 400 # количество эпох обучения
batch_size = 32 # размер мини-выборки
nb_train_samples = (1014 + 822 + 954) # количество изображений для обучения
nb_validation_samples = (168 + 136 + 159) # количество изображений для проверки
nb_test_samples = (170 + 138 + 159) # количество 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) трёх классов находились в соответствующих каталогах: для графических образов счётчиков осей был отведен каталог `Axle`, для образов мест пересечения рельсовых путей — каталог `Cross`, а для условных дефектов — `Pump`. Поэтому при обучении нейронной сети классы образцов получили имена в соответствии с названиями этих каталогов.

С применением следующего кода на языке Python 3 происходит распознавание графических образов трёх классов с помощью обученной нейронной сети. На вход сети подается подготовленный и нормализованный графический образ B 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 = ['axle', 'cross', 'pump']
B /= 255 # нормализация матрицы B
B = numpy.expand_dims(B, axis=0) # вспомогательное увеличение размерности

```

```
preds = model.predict(B) # классификация входного объекта
if classes[preds.argmax()] == 'axle': # расшифровка результата классификации
    mark_result = 'axle_counter' # счётчик осей
if classes[preds.argmax()] == 'cross': # расшифровка результата классификации
    mark_result = 'crossing' # место пересечения путей
if classes[preds.argmax()] == 'pump': # расшифровка результата классификации
    mark_result = 'pumping' # условный дефект
```

На рис. 5 и 6 изображены карты признаков, которые были получены на соответствующих свёрточных слоях нейронной сети для графического образа места пересечения рельсовых путей, приведённого на рис. 2 (см. верхний пример) (для наглядности максимальное значение точки соответствует чистому белому цвету, а минимальное — полному чёрному; размер карт первого слоя составляет 28x138 точек, второго слоя — 12x67). Карты признаков для графического образа счётчика осей (рис. 3, верхний пример) показаны на рис. 7 и 8.

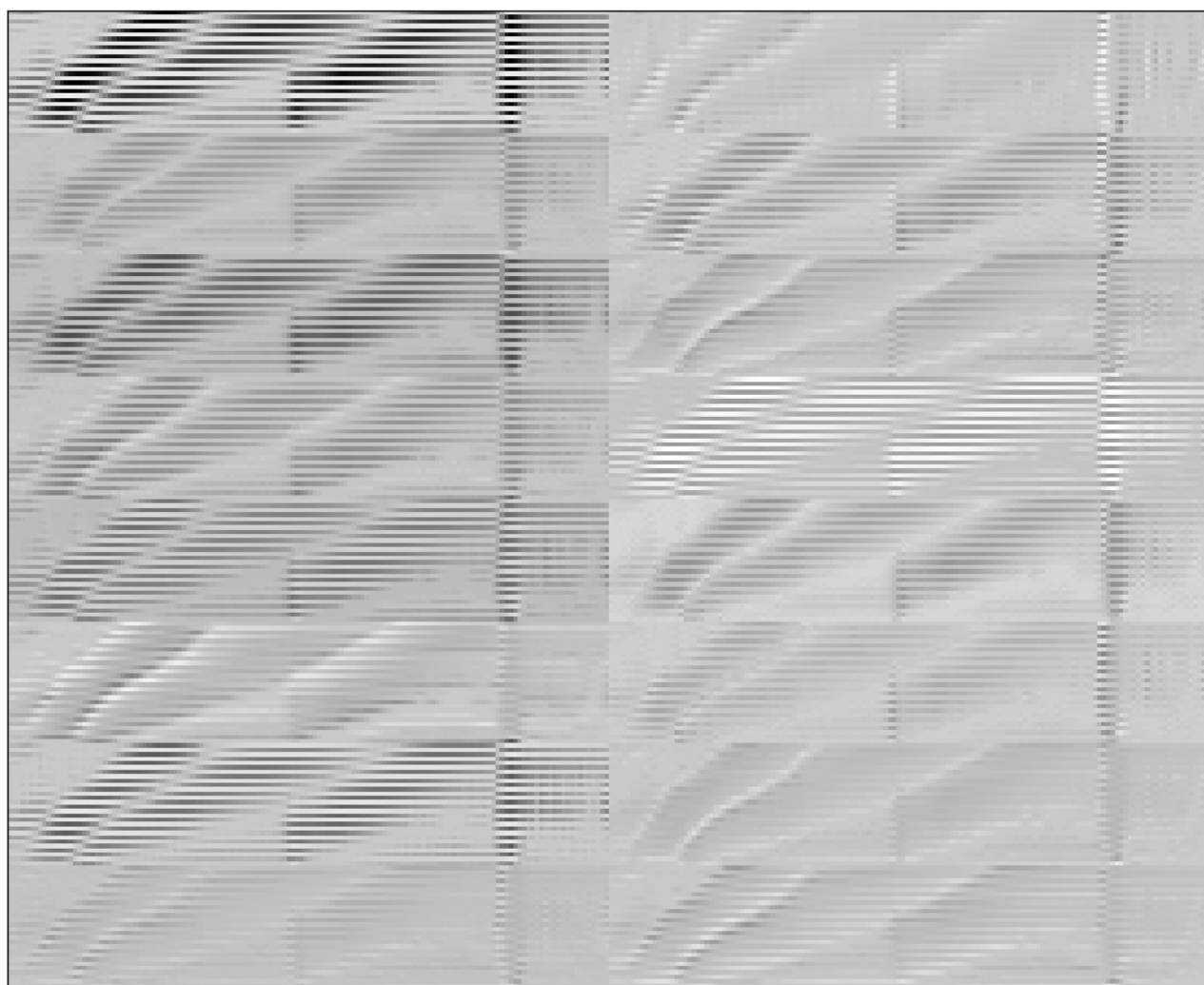


Fig. 5. Feature maps for the “crossing” in the first convolutional layer of the neural network

Рис. 5. Карты признаков для «пересечения» на первом свёрточном слое нейронной сети



Fig. 6. Feature maps for the “crossing” in the second convolutional layer of the neural network

Рис. 6. Карты признаков для «пересечения» на втором свёрточном слое нейронной сети

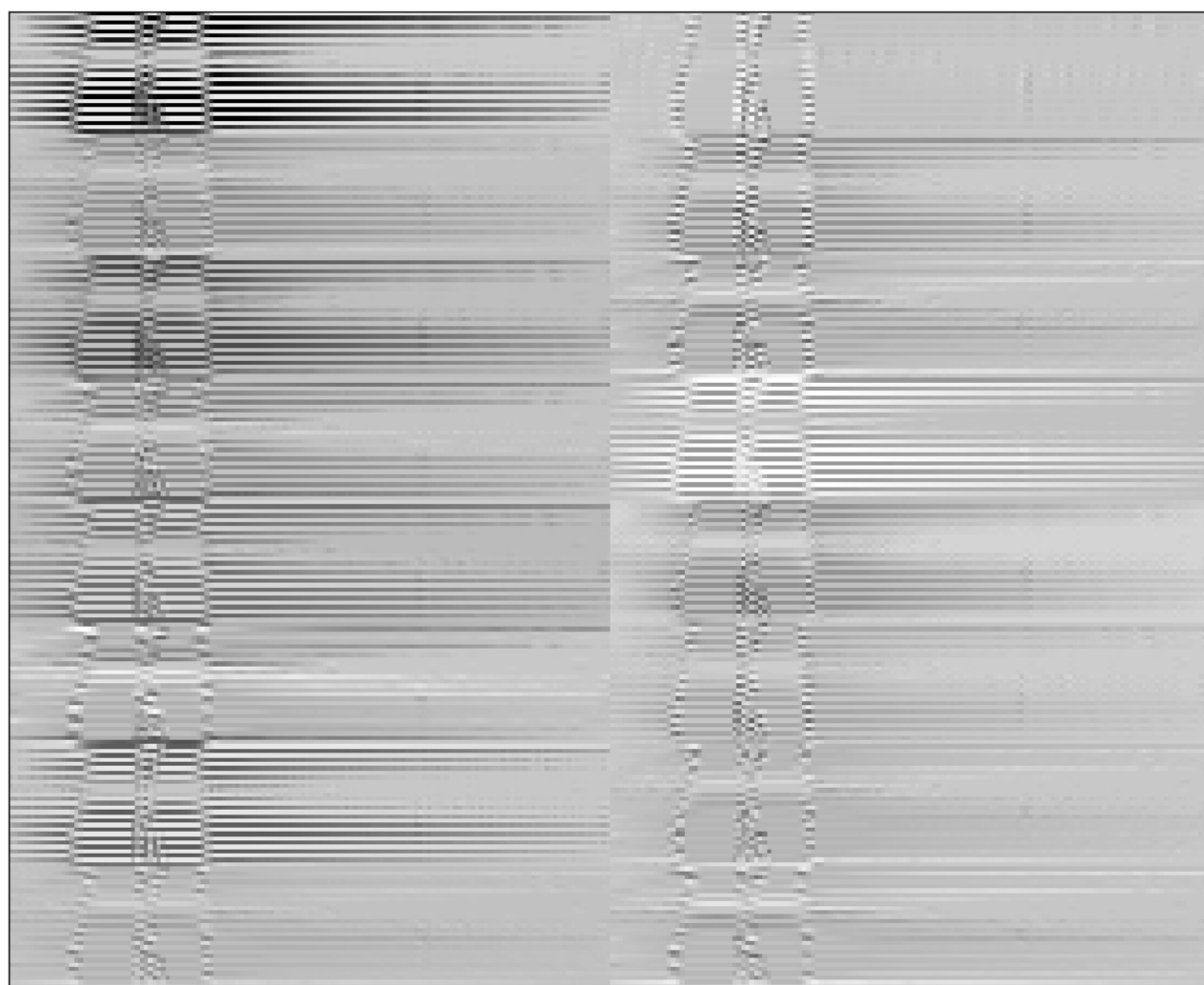


Fig. 7. Feature maps for the “axle counter” in the first convolutional layer of the neural network

Рис. 7. Карты признаков для «счётчика осей» на первом свёрточном слое нейронной сети

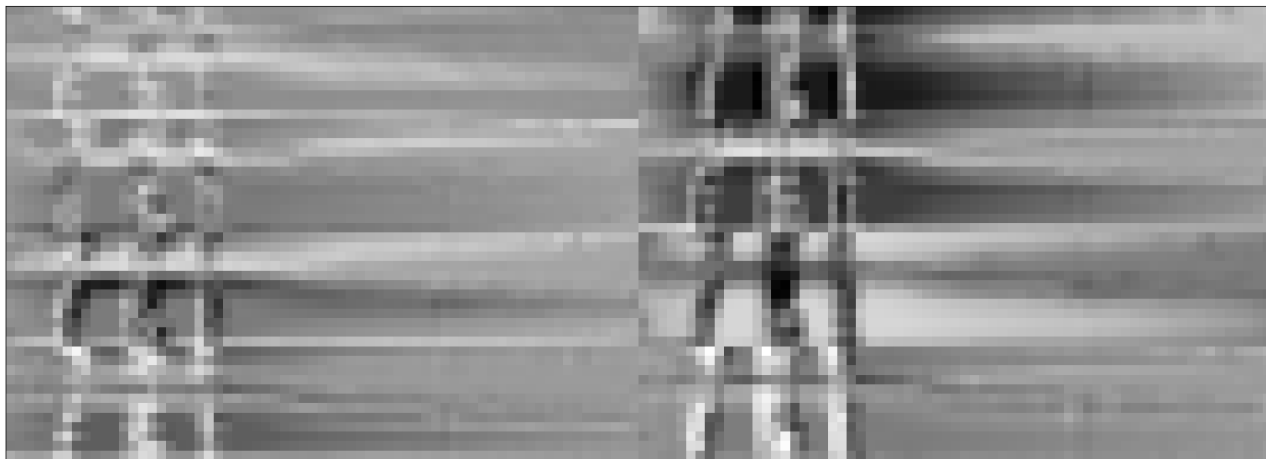


Fig. 8. Feature maps for the “axle counter” in the second convolutional layer of the neural network

Рис. 8. Карты признаков для «счётчика осей» на втором свёрточном слое нейронной сети

Заклучение

По результатам обучения можно сделать вывод, что выбранная конфигурация модели нейронной сети является удачной и вполне достаточной для решения поставленной задачи распознавания длинных конструктивных элементов рельсового пути при неразрушающем контроле поверхности головки рельсов с помощью вихретоковых дефектоскопов.

References

- [1] A. A. Markov and E. A. Kuznetsova, *Rails flaw detection. Formation and analysis of signals. Book 1. Principles*. St. Petersburg: KultInformPress, 2010.
- [2] A. A. Markov and E. A. Kuznetsova, *Rails flaw detection. Formation and analysis of signals. Book 2. Data interpretation*. St. Petersburg: Ultra Print, 2014.
- [3] V. F. Tarabrin, A. V. Zverev, O. E. Gorbunov, and E. V. Kuzmin, “About Data Filtration of the Defectogram Automatic Interpretation by Hardware and Software Complex ASTRA”, *NDT World*, vol. 64, no. 2, pp. 5–9, 2014.
- [4] E. V. Kuzmin, O. E. Gorbunov, P. O. Plotnikov, V. A. Tyukin, and V. A. Bashkin, “Application of Neural Networks for Recognizing Rail Structural Elements in Magnetic and Eddy Current Defectograms”, *Automatic Control and Computer Sciences*, vol. 53, no. 7, pp. 628–637, 2019.
- [5] E. V. Kuzmin, O. E. Gorbunov, P. O. Plotnikov, and V. A. Tyukin, “An Efficient Algorithm for Finding the Level of Useful Signals on Interpretation of Magnetic and Eddy Current Defectograms”, *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 867–870, 2018.
- [6] E. V. Kuzmin, O. E. Gorbunov, P. O. Plotnikov, and V. A. Tyukin, “Finding the Level of Useful Signals on Interpretation of Magnetic and Eddy-Current Defectograms”, *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 658–666, 2018.
- [7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [8] F. Chollet, *Deep Learning with Python*. Manning Publications Co., 2018.
- [9] *TensorFlow*. [Online]. Available: <https://www.tensorflow.org/>.