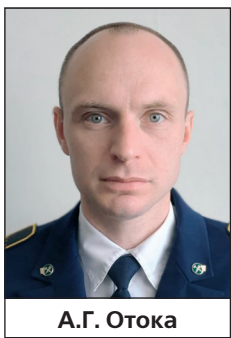


# ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В РАСПОЗНАВАНИИ ИНДИКАЦИЙ ПРИ МАГНИТОПОРОШКОВОМ КОНТРОЛЕ ДЕТАЛЕЙ ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА



А.Г. Отока



Р.Р. Ика

В статье рассмотрены вопросы применения сверточных нейронных сетей для обнаружения и классификации дефектов, выявляемых при магнитопорошковом контроле деталей и составных частей подвижного состава железных дорог. Предложен вариант алгоритма построения нейронной сети на базе существующей среды программирования Google Colab, библиотек Tensor Flow и Keras и языка программирования Python.

*Ключевые слова:* магнитопорошковый контроль, детали железнодорожного транспорта, магнитопорошковые индикации

EDN: AEOTJK

Целью неразрушающего контроля (НК) технических объектов является не только обнаружение дефектов, но и оценка потенциальной опасности последних. Распознавание дефектов разного типа существенно повышает информативность контроля, позволяет оптимизировать технологические процессы за счет реализации так называемого активного контроля.

В настоящее время наиболее динамично развивающимся направлением при решении задач НК является анализ дефектоскопической информации в виде изображений, причем распознавание дефектов сводится к визуальному распознаванию изображения обнаруженного дефекта человеком-оператором. Такое распознавание имеет ряд естественных недостатков:

субъективизм, малая надежность, зависимость от компетентности оператора и т.д.

Достоверность и объективность оценки полученных результатов непосредственно зависят от используемых средств и способов регистрации, правильный выбор которых обеспечивает технико-экономическую эффективность контроля. Применение телевизионных и фотографических средств для регистрации результатов НК с высокой точностью повышает надежность и достоверность обнаружения имеющихся на поверхности изделия дефектов, а также позволяет получить данные о степени опасности дефектов.

В настоящее время для решения задач НК применяются автоматизированные системы обработки

**Отока Александр Генрикович**, аспирант кафедры «Вагоны» Белорусского государственного университета транспорта (БелГУТ), начальник пункта технического обслуживания вагонов Гомель-пассажирский Гомельского вагонного депо РУП «Гомельское отделение Белорусской железной дороги», г. Гомель, Республика Беларусь. Область научных интересов: повышение эффективности и достоверности методов неразрушающего контроля на железнодорожном транспорте, автоматизация магнитопорошковой дефектоскопии при ремонте деталей и составных частей колесных пар. Автор более 50 научных работ. Имеет два патента на изобретения.

**Ика Ричард Робертович**, инженер-конструктор ОАО «СтанкоГомель», г. Гомель, Республика Беларусь. Область научных интересов: применение искусственного интеллекта для решения различных задач в промышленном секторе экономики, изучение возможности использования сверточных нейронных сетей при поиске и идентификации дефектов в условиях проведения неразрушающего контроля магнитопорошковым методом различных объектов контроля.

**Холодилов Олег Викторович**, доктор технических наук, профессор кафедры «Вагоны» Белорусского государственного университета транспорта (БелГУТ), г. Гомель, Республика Беларусь. Область научных интересов: материаловедение, трение и износ в машинах, неразрушающий контроль и техническая диагностика. Автор более 250 научных работ. Имеет более 10 патентов на изобретения.

изображений, машинного зрения. В качестве приемника изображения используются различные видеоконтрольные устройства. Программное обеспечение, входящее в состав этих систем, позволяет с разной степенью объективности проводить количественный анализ изображения дефектов.

Применение специализированных методов и алгоритмов автоматического распознавания изображений для систем при реализации различных методов НК позволяет существенно повысить их эффективность. Одними из таких алгоритмов в настоящее время являются алгоритмы, основанные на использовании искусственных нейронных сетей (НС) (neural network – NN) [1–3].

Области применения искусственных НС весьма разнообразны. Нейронные сети нашли применение в самых различных сферах деятельности, таких как экономика и бизнес; медицина и здравоохранение; робототехника; охранные системы; промышленность химическая, нефтеперерабатывающая; энергетика [4].

Традиционные методы НК, такие как радиационный, ультразвуковой, магнитный и др., имеют свои особенности и ограничения и требуют при интерпретации результатов участия опытных специалистов. В этой связи возникает потребность в новых методах, которые могут снизить роль человеческого фактора и повысить точность и эффективность НК.

Использование НС в НК позволяет решить такие задачи, как автоматизация и увеличение скорости контроля; повышение точности; возможность обрабатывать различные типы данных; обучение на основе имеющихся данных.

В настоящее время искусственные НС уже нашли применение в таких видах НК как радиационный [4;5], акустический (ультразвуковой, акустико-эмиссионный) [6;7], вихретоковый [8], тепловой [9], капиллярный [10–12].

Случаев применения НС для анализа и классификации дефектов при магнитно-порошковом контроле (МПК) авторам неизвестно.

В работах [13;14] нами была рассмотрена возможность использования систем видеонаблюдения и нейросетевых технологий при МПК колесных пар. Применение специализированного программного обеспечения и НС позволяет проводить анализ изображений, на которых имеется участок с высокой интенсивностью свечения люминесцентной суспензии с возможностью выдачи сообщения оператору о наличии дефекта.

Скопление магнитного порошка над дефектом приводит к изменению интенсивности его свечения, что может использоваться в качестве параметра для обучения НС.

В настоящее время существует большое число подходов к распознаванию дефектов различного типа. Построены автоматизированные системы, использующие для их классификации статистические методы машинного обучения. Такое распознавание имеет ряд естественных недостатков: субъективизм, малая надежность, зависимость от компетентности оператора и т.д.

Другим подходом к распознаванию дефектов может стать использование НС. В отличие от статистических методов, НС применяются для решения трудно формализуемых задач, для которых сложно найти точный алгоритм решения.

Разновидностью НС, используемых при анализе сложных объектов, являются сверточные НС (англ. convolutional neural network, CNN), которые могут эффективно использоваться для распознавания и классификации изображений [15–18].

Свое название архитектура сети получила из-за наличия операции свертки, суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свертки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения.

Сверточная НС обычно представляет собой чередование сверточных слоев (convolution layers), субдискретизирующих слоев (subsampling layers) и при

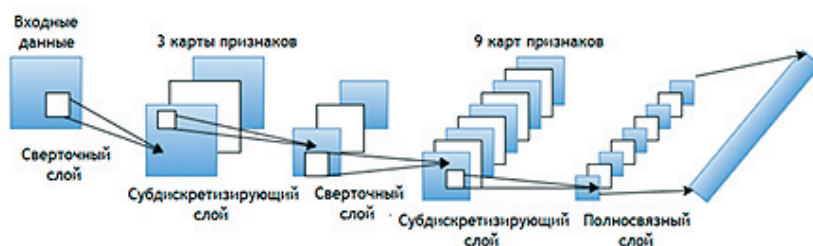


Рис. 1. Пример архитектуры сверточной НС для задачи классификации изображения

наличии полносвязных слоев (fully-connected layer) на выходе (рис. 1). Все три вида слоев могут чередоваться в произвольном порядке.

Основная идея сверточных НС заключается в том, что при обучении НС на основе исследуемых изображений одного класса, она самостоятельно формирует наборы признаков (так называемые карты признаков), которые в полной мере характеризуют этот самый класс, и в то же время, отделяют их от других изображений. Все это реализуется в сети, с помощью так называемых сверточных слоев, которые составляют карты признаков изображения, и слоев подвыборки, которые уменьшают масштаб изображения. Чередование данных слоев позволяет составить карты признаков, с помощью которых изображения классифицируются.

На данный момент не существует обоснованных рекомендаций для выбора архитектуры сверточных НС. Сеть можно собрать из блоков, описанных выше, почти в любом порядке и любых размеров.

Основной особенностью и преимуществом НС перед традиционными алгоритмами — это возможность обучения. В процессе обучения НС способна выявлять сложные зависимости между входными и выходными данными, а также выполнять обобщение. То есть, при успешном обучении, сеть позволит получить результат на основании данных, которые отсутствовали в обучающей выборке, а также в случае представления неполных и/или «зашумленных», частично искаженных данных.

### Цель работы

Показать возможность использования сверточных НС при проведении МПК и разработать алгоритм построения НС на базе уже существующей сред программирования и библиотек.

### Методика исследования. Используемое оборудование

Для реализации рассматриваемых при МПК алгоритмов анализа и классификации дефектов можно использовать одну из существующих библиотек.

Комплект программного обеспечения сверточных НС имеет четыре основных составляющих, показанных на рис. 2.

Эти популярные библиотеки были выбраны исходя из их возможности поддержки различных платформ и языков программирования, а также совместимости между собой с практической точки зрения.

Также в нашем случае был использован ряд других библиотек для написания кода, но которые напрямую не связаны с разработкой НС.

Из-за своей доступности и простоты в качестве платформы (среды программирования) был выбран Google Colab — облачный сервис Jupyter в виде программы-блокнота для записи, передачи и запуска кода в Google CoLaboratory, доступный с любой точки мира для написания кода или создания документации.

В качестве языка программирования был выбран Python — высокоуровневый язык программирования, отличающийся эффективностью, простотой и универсальностью использования.

Цифровую обработку полученных изображений можно представить в виде ряда алгоритмов: 1. Бинаризация изображения. 2. Выделение фрагментов изображения. 3. Определение формы магнитопорошкового следа. 4. Определение площади следа и его линейных размеров.

Наличие такого рода алгоритмов, применительно к изображениям различного качества, позволяет в автоматическом режиме сформировать представление об объекте контроля и выдать заключение о пригодности объекта контроля к эксплуатации. Методика иссле-

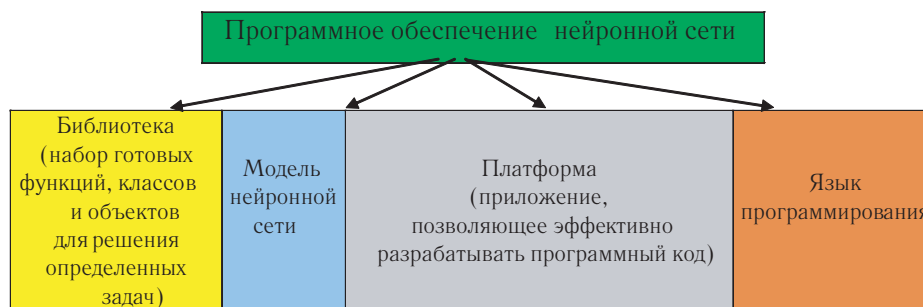


Рис. 2. Структура сверточной НС

дования заключается в последовательной обработке цифровых изображений с различными характеристиками (светочувствительность, разрешение, контрастность магнитопорошкового следа, УФ-освещенность объекта контроля, общая освещенность рабочего места и др.) и выявлении особенностей для каждого изображения.

Определение реакций алгоритмов на те или иные характеристики изображений позволят выявить уязвимые места самих алгоритмов или же их сильные стороны. Алгоритмы автоматической обработки цифровых изображений могут быть реализованы в виде программного пакета и задействованы при обработке результатов [12;22;23].

В нашем эксперименте для обучения НС был подготовлен набор входных данных, состоящий из 60 изображений (файлов в формате \*.jpg).

Выбор формата изображения зависит от конкретной задачи и требований к его качеству. Многие модели машинного обучения адаптированы для работы с изображениями в формате \*.jpg. Обработка изображений в таком формате может быть оптимизирована для их быстрого чтения и загрузки. Поэтому использование того или иного формата не принципиально и формат \*.jpg был выбран нами в силу своей популярности. Причем какой бы формат мы ни использовали, сверточная НС (СНС) осуществляет дальнейшее сжатие данных на фотографии. Отсюда и независимость как таковая от формата представления изображения.

В качестве изображений использовались фотографии деталей и составных частей вагонов, сделанные авторами в процессе проведения МПК в одинаковых

условиях равномерной люминесцентной освещенности, созданной УФ-источником излучения. Этот набор данных содержит в себе выборку, состоящую из двух классов изображений:

- изображения, содержащие дефект «def». Дефект представляет скопление магнитного порошка над естественной трещиной или искусственным дефектом с уже известными размерами в виде заводской вставки;

- изображения, содержащие фон («fop»), оставленный после проведения МПК с использованием люминесцентной суспензии в условиях отсутствия дефектов на поверхности. Фон представляет собой равномерное распределение частиц магнитного порошка на участке объекта контроля после операции намагничивания и нанесения люминесцентной суспензии.

На рис. 3 приведена схема эксперимента, который состоял из двух этапов: подготовительного и основного. На подготовительном этапе осуществлялся непосредственно сам МПК выбранных объектов, а на основном этапе происходила цифровая регистрация изображений и их дальнейшая обработка.

Объект контроля в процессе МПК, освещался при помощи УФ-источника излучения (Ultrafire WF-501B UV, Китай) с длиной волны 395 нм. С помощью цифровой фотокамеры с разрешением 64 Мп (фокусное расстояние  $F=150$  мм) регистрировали изображение полученных индикаций на участке объекта контроля, после чего полученное цифровое изображение загружали на персональный компьютер для дальнейшей обработки (см. рис. 3). Освещенность контролируемой поверхности видимым светом измерялась циф-

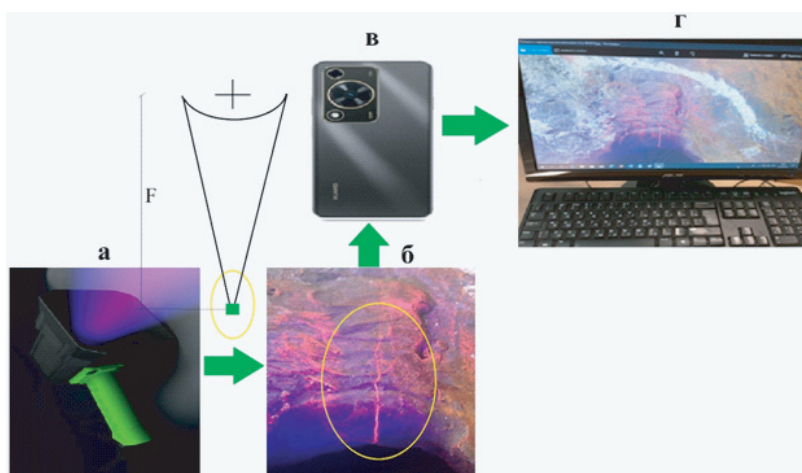


Рис. 3. Условная схема эксперимента: а – УФ-источник излучения; б – изображение участка объекта контроля после проведения МПК; в – цифровая камера; г – компьютер с программным обеспечением



ровым мультиметром (MASTECH MS-8229, Китай) и не превышала 20 лк.

В ходе эксперимента для получения качественных фотографий полученных индикаций после проведения МПК соблюдались такие параметры как равномерная УФ-освещенность поверхности объекта контроля (расстояние выдерживалось за счет стационарного штатива с возможностью фиксации УФ-источника излучения) и одинаковое расстояние от цифровой фотокамеры до объекта контроля.

Магнитопорошковому контролю подвергались следующие детали: маятниковая подвеска автосцепного устройства с продольной и поперечными трещинами, втулка шпинтона с продольной трещиной, фрагмент цельнокатаного колеса с поперечной трещиной на внутренней части обода, ролик подшипника с искусственным дефектом в виде вставки, гайка шестигранная с фланцем поводка тележки с продольной трещиной).

По набору входных данных (рис. 4) можно предсказать результат, например, фоновое свечение индикатора или свечение дефекта (трещина, волосовина, ложные индикации по рискам и др.).

Итоговая оценка качества обучения НС после завершения процесса обучения проводилась на тестовом наборе данных, который также представлял подборку изображений, не входящих в первоначальный датасет<sup>1</sup>, с магнитопорошковыми индикациями на упомянутых выше деталях.

Учитывая небольшое количество имеющихся изображений, важно было получить максимально хороший результат при распознавании дефекта, вероятность которого можно оценить в процентах. Наиболее оптимальным методом обучения является правило «80 на 20», т. е. 80% всех изображений НС будет использоваться в качестве материалов для обучения, а оставшиеся 20% используются для проверки правильности работы НС. Это правило используется

<sup>1</sup>Датасет — совокупность данных, систематизированных в определенном формате (в нашем случае датасетом является набор фотографий в формате «\*.jpg»).

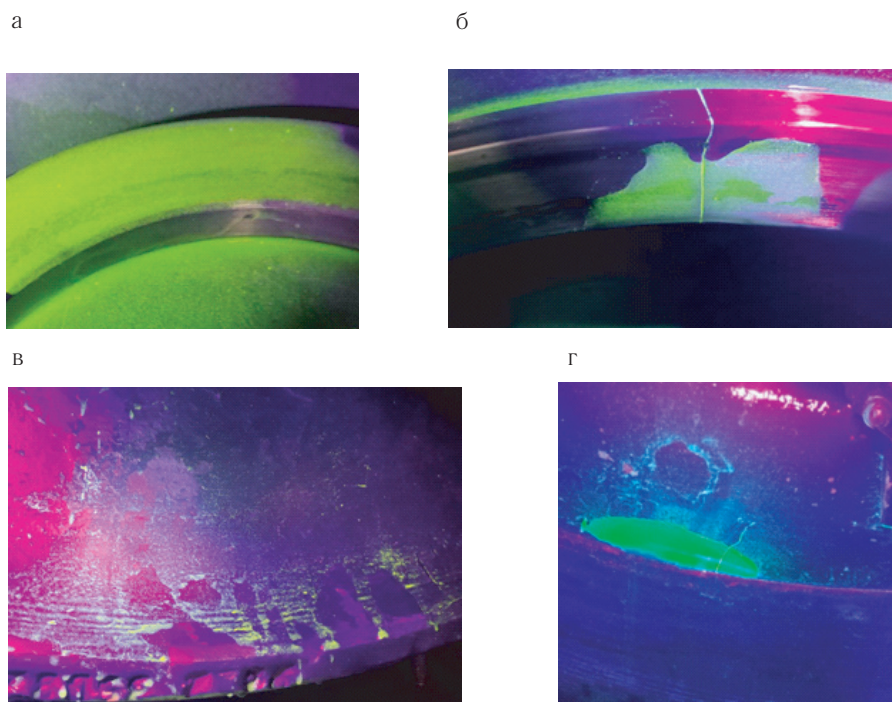


Рис. 4. Пример используемых изображений для формирования входных данных при обучении НС: а – фоновое свечение суспензии КСФ-12 на поверхности упорного кольца подшипника буксового узла; б – свечение дефекта (трещины) на поверхности упорного кольца подшипника буксового узла; в – фоновое свечение суспензии КСФ-12 на поверхности цельнокатаного колеса колесной пары; г – свечение дефекта (трещины) на поверхности цельнокатаного колеса колесной пары

практически для всех НС, от которых ожидают определенный результат на выходе [24–27].

### Результаты исследований и их обсуждение

Последовательность написания кода на языке Python, с помощью которого происходило обучение НС для распознавания и классификации изображений состоит из следующих операций:

1. Импорт нужных нам библиотек в среду разработки (рис. 5).

2. Подключение облачного хранилища, в которое предварительно были загружены изображения

(рис. 6). Так как разработка производится в облачном сервисе, необходимо изображения в него загрузить. Реализовать данную задачу можно разными способами, в том числе подключить облачное хранилище в среду разработки.

3. Создание тестового и проверочного датасета (рис. 7).

4. Кэширование датасета, основной функцией которого является ускорение процесса извлечения данных (рис. 8).

5. Создание и компилирование<sup>2</sup> модели НС (рис. 9). Для нашей задачи хорошо подошла последовательная архитектура НС, состоящая из блоков свертки

<sup>2</sup>Без компилятора любой код на языке программирования будет для компьютера просто текстом — он не распознает команды и не сможет их выполнить

```
import matplotlib.pyplot as plt # импорт библиотеки для визуализации данных
import numpy as np # импорт библиотеки позволяющая ускорить работу с многомерными массивами и матрицами
import PIL # импорт библиотеки для обработки изображений
import tensorflow as tf # импорт библиотеки TensorFlow

from tensorflow import keras # импорт API (keras)
from tensorflow.keras import layers # импорт функций из TensorFlow и keras
from tensorflow.keras.models import Sequential # импорт функций из TensorFlow и keras
```

Рис. 5. Ввод библиотек в среду разработки Google Colab

```
from google.colab import drive # обращаемся к функции drive
drive.mount('/content/gdrive', force_remount=True) # подключаем облачное хранилище
```

Рис. 6. Подключение облачного хранилища Google Drive

```
batch_size = 32 # размер мини-выборки

# ниже задаем размер изображения
img_width = 180
img_height = 180

# создаем тренировочный датасет (датасет - массив данных)
train_ds = tf.keras.utils.image_dataset_from_directory(
    dataset_dir, # каталог данных
    validation_split = 0.2, # функция разделения изображений
    subset="training", # вызываемый метод (тренировка)
    seed=123,
    image_size=(img_height, img_width),
    batch_size = batch_size)

# создаем проверочный датасет
val_ds = tf.keras.utils.image_dataset_from_directory(
    dataset_dir,
    validation_split=0.2,
    subset="validation", # вызываемый метод (проверка)
    seed=123,
    image_size=(img_height, img_width),
    batch_size = batch_size)
```

Рис. 7. Код созданного тестового и проверочного датасетов

с максимальным объединяющим слоем в каждом из них, и с полносвязным слоем на 128 единиц, вместе с нелинейной активационной функцией Relu.

6. Обучение НС (рис. 10). Для обучения НС указывалось количество эпох<sup>3</sup> обучения и использовался метод fit. Метод fit используется для возвращения объекта, в котором хранятся значения функций потерь для тренировочной и проверочной выборки (датасета).

7. Проверка качества обучения НС, с использованием библиотеки Matplotlib. Определение точности и потерь<sup>4</sup> НС по статистическим данным в виде графиков.

По результатам обучения НС были получены зависимости точности и потерь навыков обучения и проверки НС (рис. 11).

Видно, что к 10-й эпохе точность навыков обучения и проверки НС растет, а потери навыков обуче-

ния и проверки снижаются. Это говорит о том, что НС к 10-й эпохе начинает хорошо распознавать как тренировочный, так и проверочный датасет. Можно считать, что обучение НС прошло успешно.

В противном случае, как это часто бывает при разработке НС, потери точности при обучении снижаются, однако потери точности при проверке растут.

Это говорит о том, что НС хорошо справляется с изображениями, используемыми при ее тренировке, и плохо справляется с изображениями, используемыми для ее проверки. Это проблема называется overfitting (с англ. переобучение).

Для решения данной проблемы существуют общепринятые методики, такие как:

- увеличение датасета, либо путем поиска новых изображений, либо при помощи техники, которая называется «аугментация датасета». Аугментация — это прием, который позволяет расширить дата-

<sup>3</sup>Эпоха — один цикл обучения НС на массиве данных.

<sup>4</sup>Точность НС представляет показатель того, насколько правильно НС классифицирует изображения. Потери НС — показатель того, насколько далеко предсказуемое значение находится от желаемого выходного значения. Чем больше значение показателя потерь, тем хуже справляется НС со своей задачей и наоборот.

```
# кэшируем датасет
AUTOTUNE = tf.data.AUTOTUNE
train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

Рис. 8. Кэширование датасета

```
# создаем модель СНС
num_classes = len(class_names)

# ниже прописываем слои СНС и используем функцию relu
model = Sequential([
    layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])

# компилируем модель
model.compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['accuracy'])
```

Рис. 9. Создание модели сверточной НС

```
epochs = 10 # количество эпох обучения
# вызываем метод fit
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs)

# выводим статистические данные в виде графиков (точность и потери)
# данный код можно найти в документации по работе с TensorFlow
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs_range = range(epochs)
plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')
plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```

Рис. 10. Обучение НС

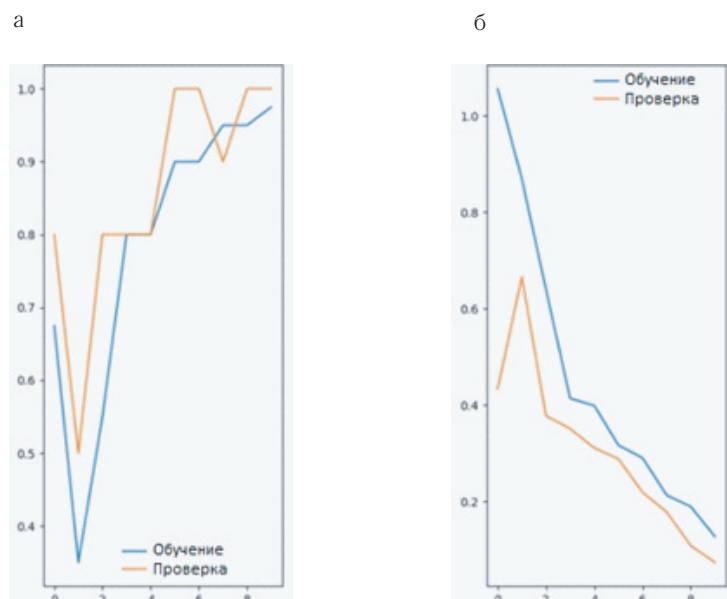


Рис. 11. Зависимость точности (а) и потерь (б) навыков обучения (тренировочный процесс) и проверки НС

сет в несколько раз путем несложных манипуляций с изображениями, а именно, картинки в датасете каким-либо образом поворачивают, отображают, искажают и т.д.

- регуляризация, это довольно простой математический прием, позволяющий повысить точность модели и уменьшить ненужные отклонения. Как правило в TensorFlow для этого используются методы Dropout или Batch Normalization [28–30].

Ручная проверка обученной модели НС (загрузка нового изображения, которое отсутствовало в датасетах (тестовых и проверочных) производится следующим образом. Указывается путь на изображение, находящийся в Google Drive и прописывается код, загружающий изображение в модель НС (рис. 12).

По итогам обучения НС с задачей классификации дефекта (трещины) и фона на образцах были получены следующие результаты.



```
# загружаем изображения и Google Drive
test = pathlib.Path('/content/gdrive/MyDrive/test/def/6e0c98d1-6734-405b-9108-cf67d781c261.jpg')

# загружаем изображение в модель НС
img = tf.keras.utils.load_img(test, target_size=(img_height, img_width))
img_array = tf.keras.utils.img_to_array(img)
img_array = tf.expand_dims(img_array, 0)
predictions = model.predict(img_array)
score = tf.nn.softmax(predictions[0])
# выводим результат проверки изображения на экран
print("На изображении {} {:.2f}% вероятность".format(
    class_names[np.argmax(score)],
    100 * np.max(score)))
img.show()
```

Рис. 12. Проверка работы НС

а

```
[ ] # загружаем изображения и Google Drive
test = pathlib.Path('/content/gdrive/MyDrive/test/fon/15d62bb4-23ed-48bf-b836-e1b10c4d3b63.jpg')

# загружаем изображение в модель НС
img = tf.keras.utils.load_img(test, target_size=(img_height, img_width))
img_array = tf.keras.utils.img_to_array(img)
img_array = tf.expand_dims(img_array, 0)
predictions = model.predict(img_array)
score = tf.nn.softmax(predictions[0])
# выводим результат проверки изображения на экран
print("На изображении {} {:.2f}% вероятность".format(
    class_names[np.argmax(score)],
    100 * np.max(score)))
img.show()
```

1/1 [=====] - 0s 42ms/step  
На изображении fon (82.33% вероятность)

б

```
[12] # загружаем изображения и Google Drive
test = pathlib.Path('/content/gdrive/MyDrive/test/def/6e0c98d1-6734-405b-9108-cf67d781c261.jpg')

# загружаем изображение в модель НС
img = tf.keras.utils.load_img(test, target_size=(img_height, img_width))
img_array = tf.keras.utils.img_to_array(img)
img_array = tf.expand_dims(img_array, 0)
predictions = model.predict(img_array)
score = tf.nn.softmax(predictions[0])
# выводим результат проверки изображения на экран
print("На изображении {} {:.2f}% вероятность".format(
    class_names[np.argmax(score)],
    100 * np.max(score)))
img.show()
```

1/1 [=====] - 0s 43ms/step  
На изображении def (97.98% вероятность)

Рис. 13. Результаты работы НС по определению магнитопорошковых индикаций фонового свечения (а) и дефекта (б)


На рис. 13 показано, как НС определила с точностью 97,98% изображение, содержащее дефект и с точность 82,33% изображение, содержащее фон.

С учетом использования в работе только набора данных в виде двух классов изображений «def» и «fon», средняя точность НС составила 90,16%, что является приемлемым результатом при небольшой выборке в 60 фотографий.

## Заключение

Таким образом, выбранная авторами конфигурация модели НС является удачной и вполне работоспособной при работе с распознаванием индикаций при проведении МПК. В дальнейшем в целях усовершенствования НС в датасет можно загружать изображения с ложными магнитопорошковыми индикациями, вызванными

чрезмерной концентрацией магнитного индикатора в используемом объеме суспензии, а также наличием задигов, рисок и других повреждений механического характера.

Предложенный авторами алгоритм работы НС можно взять за основу при идентификации магнитопорошковых индикаций в ходе выполнения автоматического или автоматизированного МПК деталей подвижного состава с использованием специализированных стендов и видеосистем. 

## Литература

1. Alex Krizhevsky Ilya Sutskever Geoffrey E. Hinton ImageNet Classification with Deep Convolutional Neural Networks //Advances in Neural Information Processing Systems 25; Edited by: F. Pereira and C.J. Burges and L. Bottou and K.Q. Weinberger (NIPS 2012), pp. 1097-1105.
2. Jaswal, Deepika & Vishvanathan, Sowmya & Kp, Soman. Image Classification Using Convolutional Neural Networks. International Journal of Scientific and Engineering Research. 5, pp. 1661-1668. 10.14299/ijser.2014.06.002.
3. Chauhan, Rahul & Ghanshala, Kamal & Joshi, R. (2018). Convolutional Neural Network (CNN) for Image Detection and Recognition, pp. 278-282. 10.1109/ICSCCC.2018.8703316.
4. Назаренко, С. Ю. Применение искусственных нейронных сетей в радиационном неразрушающем контроле / С. Ю. Назаренко, В. А. Удод. - Текст : непосредственный // Дефектоскопия. - 2019. - № 6. - С. 54-70.
5. Поляков, А. В. О применении сверточных нейронных сетей при решении задачи неразрушающего контроля изделий / А. В. Поляков. - Текст : непосредственный // Успехи современной науки. - 2017. - Т. 1, № 5. - С. 200-210.
6. Стенюшкин, В. В. Применение искусственных нейронных сетей в задачах ультразвукового неразрушающего контроля / В. В. Стенюшкин, Г. Ю. Гуськов, В. В. Шишкин. - Текст : непосредственный // Теоретические и практические аспекты развития отечественного авиастроения: тезисы Всероссийской научно-технической конференции. - Ульяновск : УлГТУ. - 2016. - С. 43-45.
7. Возможность применения искусственных нейронных сетей для распознавания АЭ образов / С. К. Фомичев и [др.]. - Текст : непосредственный // J. of Mech. Eng. NTUU «Kyiv Polytechnic Institute». - 2011. - Т. 1, № 61. - С. 199-203.
8. Применение нейронных сетей для распознавания конструктивных элементов рельсов на магнитных и вихретоковых дефектограммах / Е. В. Кузьмин и [др.]. - Текст : непосредственный // Моделирование и анализ информационных систем. - 2018. - № 25(6). - С. 667-679.
9. Вавилов, В. П. Активный тепловой контроль композиционных материалов с использованием нейронных сетей / В. П. Вавилов, Д. А. Нестерук. - Текст : непосредственный // Дефектоскопия. - 2011. - № 10. - С. 10-18.
10. Huipeng Yu, Maodong KANG, Chengyang Ding et al. An automated fluorescent defect detection system in precision castings by convolutional neural network, 15 September 2023, PREPRINT (Version 1) available at Research Square. - doi:org/10.21203/rs.3.rs-3337794/v1.
11. Холичев, Д. Д. Исследование возможности применения нейронных сетей для решения задач капиллярной дефектоскопии / Д. Д. Холичев, Анваржон Ганижон угли Хоназаров. - Текст : непосредственный // Ресурсосберегающие технологии в контроле, управлении качеством и безопасности : сборник научных трудов XI Международной конференции школьников, студентов, аспирантов, молодых учёных «Ресурсоэффективные системы в управлении и контроле: взгляд в будущее», 8-10 ноября 2022 г., г. Томск. - Томск : Изд-во ТПУ. - 2023. - С. 259-263.
12. Чесноков, Д. В. Разработка алгоритмов автоматической обработки цифровых изображений в капиллярной дефектоскопии: магистерская диссертация / Д. В. Чесноков; Национальный исследовательский Томский политехнический университет (ТПУ), Институт неразрушающего контроля (ИНК), Кафедра физических методов и приборов контроля качества. - Томск, 2016. - 124 с. - Текст : непосредственный.
13. Отока, А. Г. Автоматизированный магнитопорошковый контроль колесных пар с использованием систем видеонаблюдения и нейросетевых технологий / А. Г. Отока, О. В. Холодилов, П. А. Дашук. - Текст : непосредственный // Механика. Исследования и инновации: международный сборник научных трудов. - Вып. 16. - Гомель: БелГУТ, 2023. - С. 153-158.

14. Отока, А. Г. Перспективы автоматизации магнитопорошкового контроля колесных пар с использованием систем видеонаблюдения и нейросетевых технологий / А. Г. Отока, О. В. Холодилов. - Текст : непосредственный // Транспортная наука и инновации: материалы международной научно-практической конференции, посвященной юбилею СамГУПС. - Самара, 2023. - С. 51-57 с.
15. Картер, Джейд Сверточные нейросети / Джейд Картер. - Москва : Литрес. - URL: <https://www.litres.ru/book/dzheyd-karter/svertochnye-neyroseti-70965715/> (дата обращения: 30.12.2024). - Текст : электронный.
16. Michael Nielsen Neural Networks and Deep Learning. - URL: <http://neuralnetworksanddeeplearning.com/index.html> (дата обращения: 30.12.2024). - Текст : электронный.
17. Сверточные нейронные сети (CNN). - URL: <https://yourtodo.ru/ru/posts/svertochnnye-nejronnyie-seti-cnn/> (дата обращения: 30.12.2024). - Текст : электронный.
18. Singh, S.A., Desai, K.A. Automated surface defect detection framework using machine vision and convolutional neural networks / J. Intell Manuf 34, 1995-2011 (2023). - doi:10.1007/s10845-021-01878-w.
19. Все символы в TensorFlow 2. - URL: [https://www.tensorflow.org/api\\_docs/python/tf/all\\_symbols](https://www.tensorflow.org/api_docs/python/tf/all_symbols) (дата обращения: 24.05.2024). - Текст : электронный.
20. Документация Keras 3 API. - URL: <https://ke-ras.io/api/> (дата обращения: 24.05.2024). - Текст : электронный.
21. Словарь IT. Keras, 2023. - URL: <https://blog.skillfactory.ru/glossary/keras/> (дата обращения: 24.05.2024). - Текст : электронный.
22. Применение нейронных сетей для распознавания конструктивных элементов рельсов на магнитных и вихретоковых дефектограммах / Е. В. Кузьмин, О. Е. Горбунов и [др.]. - Текст : непосредственный // Моделирование и анализ информационных систем. - 2018. - Т. 25, № 6. - С. 667-679. - doi: 10.18255/1818-1015-2018-6-667-679.
23. Сикорский, О. С. Обзор сверточных нейронных сетей для задачи классификации изображений / О. С. Сикорский. - Текст : непосредственный // Новые информационные технологии в автоматизированных системах. - 2017. - № 20. - С. 37-42.
24. Парасич, А. В. Формирование обучающей выборки в задачах машинного обучения / А. В. Парасич, В. А. Парасич, И. В. Парасич. - Текст : непосредственный // Обзор. Информационно-управляющие системы. - 2021. - № 4. - С. 61-70. - doi:10.31799/1684-8853-2021-4-61-70.
25. Свирина, А. Р. Классификация изображений с помощью сверточных нейронных сетей / А. Р. Свирина. - Текст : непосредственный // Молодой ученый. - 2024. - № 24 (523). - С. 96-99.
26. Разделение данных на обучающую и тестовую выборки в Python: подробное руководство и практические примеры. - URL: <https://zdrons.ru/veb-program-mirovanie/razdelenie-dannyh-na-obuchajushhuju-i-testovuju-vyborki-v-python-po-drobnое-rukovodstvo-i-prakticheskieprimery/?ysclid=m43xw297g132297176> (дата обращения: 30.12.2024). - Текст : электронный.
27. Обучающая и тестовая выборки (training set & testing set). - URL: [https://vk.com/wall-210601538\\_290?ysclid=m43x8vbedg564952696](https://vk.com/wall-210601538_290?ysclid=m43x8vbedg564952696) (дата обращения: 30.12.2024). - Текст : электронный.
28. Garbin, C., Zhu, X. & Marques, O. Dropout vs. batch normalization: an empirical study of their impact to deep learning. Multimed Tools Appl 79, 12777–12815 (2020). <https://doi.org/10.1007/s11042-019-08453-9>.
29. Guangyong, Chen & Chen, Pengfei & Shi, Yujun & Hsieh, Kim & Liao, Benben & Zhang, Shengyu. (2019). Rethinking the Usage of Batch Normalization and Dropout in the Training of Deep Neural Networks. 10.48550/arXiv.1905.05928.
30. Hinton, Geoffrey & Srivastava, Nitish & Krizhevsky, Alex & Sutskever, Ilya & Salakhutdinov, Ruslan. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint. arXiv.