

DOI10.37882/2223–2966.2022.07.10

ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ ДЛЯ ОПРЕДЕЛЕНИЯ ПОВРЕЖДЕНИЙ ПОКРЫТИЯ РАЗЛИЧНЫХ ПОВЕРХНОСТЕЙ

TRAINING OF A NEURAL NETWORK TO DETERMINE DAMAGE TO THE COATING OF VARIOUS SURFACES

T. Ziyazetdinov
B. Latypov
A. Muhetdinov

Summary. This article describes the creation and operation of a neural network that can detect visible damage to various surfaces and be used to prevent the consequences of fatigue failure of various kinds of structures due to cracks due to external and internal factors of influence.

Keywords: neural network, damage, surface, dataset, machine learning, sampling.

Зиязетдинов Тимур Рифович

ФГБОУ ВО «Уфимский государственный нефтяной
технический университет»
ziyazetdinov.timur@gmail.com

Латыпов Булат Фиделевич

ФГБОУ ВО «Уфимский государственный нефтяной
технический университет»
Linksmtv@gmail.com

Мухетдинов Амир Русланович

ФГБОУ ВО «Уфимский государственный нефтяной
технический университет»
amir.muhetdinov@yandex.ru

Аннотация. Данная статья описывает создание и работу нейронной сети, которая способна определять видимые повреждения различных поверхностей и может использоваться для предотвращения последствий усталостного разрушения разного рода конструкций из-за возникновения трещин вследствие внешних и внутренних факторов воздействия.

Ключевые слова: нейронная сеть, повреждение, поверхность, датасет, машинное обучение, выборка.

Введение

В настоящее время ежегодно растет объем строительства разного рода зданий и сооружений. Вместе с тем растет объем работ по диагностике и поддержанию конструкций в надлежащем состоянии, поскольку это напрямую влияет на обеспечение должных условий работы и жизни человека. Данная работа рассматривает внедрение ПО на основе нейронных сетей в структуру диагностики и анализа качества покрытий для увеличения эффективности за счет снижения времени необходимого для обработки информации, благодаря использованию вычислительных мощностей компьютеров и других видов ЭВМ.

1. Описание датасета

Для обучения нейросети был выбран датасет, который состоит из фотографий поверхностей, часть которых имеют явные признаки внешнего разрушения, кон-

кретно — трещины (рис. 1). Выборка состоит из порядка 40000 цветных фотографий разрешением 300*300 пикселей. В целях проверки работы программы фотографии были разбиты на три группы — обучающую, тестовую и валидационную.

2. Описание кода

Нейросеть написана на базе языка программирования Python с использованием различных библиотек, таких как tensorflow, numpy, pandas и других [1].

Написание НС происходило в следующей последовательности шагов:

1. Определение классов фотографий (с трещинами / без трещин), указание путей к выборкам (рис. 2).
2. Отображение соотношения количества фотографий каждого из классов (рис. 3).
3. Использование метода flow_from_dataframe для создания обучающего, тренировочного и валидационного датасетов (рис. 4).

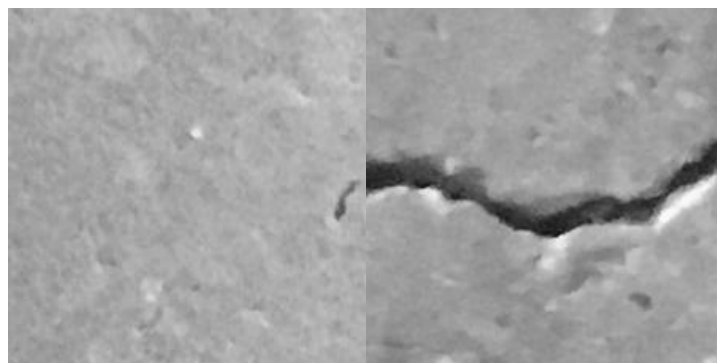


Рис. 1. Пример фотографий из обучающей выборки

```
train_img_path=pathlib.Path(train_data_path)
train_images=list(train_img_path.glob('*/*'))

positive=pathlib.Path('C:/Users/ OneDrive/Рабочий стол/data/training/positive')
positive=list(positive.glob('*'))
negative=pathlib.Path('C:/Users/ OneDrive/Рабочий стол/data/training/negative')
negative=list(negative.glob('*'))
```

Рис. 2. Команда определения пути к обучающим выборкам

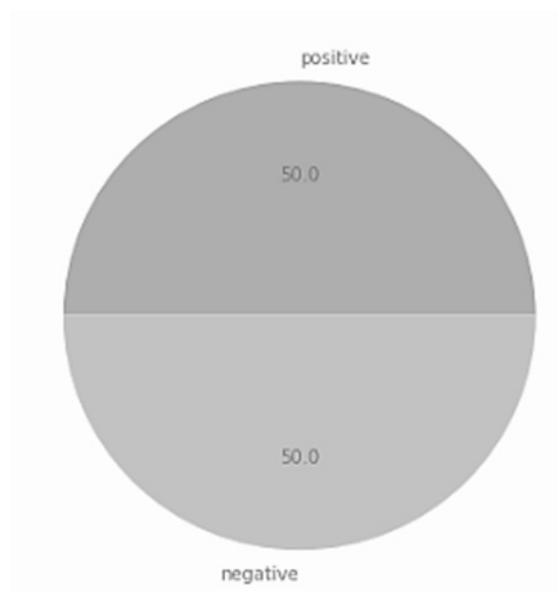


Рис. 3. Круговая диаграмма, отражающая соотношение фотографий двух классов в выборке

```

BATCH_SIZE=32
INPUT_SHAPE=(180,180)

datagen1=ImageDataGenerator(rescale=1./255,horizontal_flip=True,zoom_range=0.4,shear_
range=0.3)
datagen2=ImageDataGenerator(rescale=1./255)

train_ds=datagen1.flow_from_directory(train_img_path,shuffle=True,target_size=INPUT_S
HAPE,batch_size=BATCH_SIZE,class_mode='sparse')

validation_ds=datagen2.flow_from_directory(validation_img_path,shuffle=True,target_si
ze=INPUT_SHAPE,batch_size=BATCH_SIZE,class_mode='sparse')

prediction_imgs=list(prediction_img_path.glob('*'))

```

Рис. 4. Команда формирования выборок для обучения

```

def get_data(path):
    X=[]
    for i in path:
        img=load_img(i,target_size=(180,180),color_mode='rgb')
        img=img_to_array(img)
        img=img/255
        X.append(img)
    X=np.asarray(X)
    return X

test_ds=get_data(prediction_imgs)

```

Рис. 5. Команда создания функции для формирования валидационного датасета

```

INPUT_SHAPE=INPUT_SHAPE+(3,)

cnn=Sequential()
cnn.add(layers.Conv2D(filters=64,kernel_size=(4,4),activation='relu',input_shape=INPU
T_SHAPE))
cnn.add(layers.MaxPool2D(pool_size=(2,2)))
cnn.add(layers.Conv2D(filters=128,kernel_size=(4,4),activation='relu'))
cnn.add(layers.MaxPool2D(pool_size=(2,2)))
cnn.add(layers.Conv2D(filters=64,kernel_size=(4,4),activation='relu',))
cnn.add(layers.MaxPool2D(pool_size=(2,2)))
cnn.add(layers.Conv2D(filters=64,kernel_size=(4,4),activation='relu'))
cnn.add(layers.MaxPool2D(pool_size=(2,2)))
cnn.add(layers.Dropout(0.3))
cnn.add(layers.Flatten())
cnn.add(layers.Dense(units=128,activation='relu'))
cnn.add(layers.Dropout(0.4))
cnn.add(layers.Dense(units=6,activation='softmax'))

```

Рис. 6. Структура модели обучаемой нейросети

```

cnn.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])

history=cnn.fit(train_ds,steps_per_epoch=100,epochs=10,validation_data=validation_ds)

```

Рис. 7. Команда компилирования модели нейросети

```

Epoch 1/10
100/100 [=====] - 37s 372ms/step - loss: 0.7491 - accuracy:
0.5481 - val_loss: 0.6560 - val_accuracy: 0.4997
Epoch 2/10
100/100 [=====] - 31s 306ms/step - loss: 0.9415 - accuracy:
0.6569 - val_loss: 0.6914 - val_accuracy: 0.4988
Epoch 3/10
100/100 [=====] - 30s 301ms/step - loss: 0.7011 - accuracy:
0.4899 - val_loss: 0.7082 - val_accuracy: 0.4997
Epoch 4/10
100/100 [=====] - 30s 300ms/step - loss: 0.6968 - accuracy:
0.4859 - val_loss: 0.6929 - val_accuracy: 0.5003
Epoch 5/10
100/100 [=====] - 30s 300ms/step - loss: 0.6956 - accuracy:
0.4953 - val_loss: 0.7123 - val_accuracy: 0.4997
Epoch 6/10
100/100 [=====] - 29s 287ms/step - loss: 0.6958 - accuracy:
0.5041 - val_loss: 0.6896 - val_accuracy: 0.5002
Epoch 7/10
100/100 [=====] - 29s 286ms/step - loss: 0.5882 - accuracy:
0.6644 - val_loss: 0.2800 - val_accuracy: 0.9001
Epoch 8/10
100/100 [=====] - 29s 288ms/step - loss: 0.1999 - accuracy:
0.9384 - val_loss: 0.0369 - val_accuracy: 0.9671
Epoch 9/10
100/100 [=====] - 30s 295ms/step - loss: 0.0877 - accuracy:
0.9722 - val_loss: 0.0122 - val_accuracy: 0.9816
Epoch 10/10
100/100 [=====] - 29s 288ms/step - loss: 0.1089 - accuracy:
0.9722 - val_loss: 0.1167 - val_accuracy: 0.9846

```

Рис. 8. Результаты обучения нейросети

```

plt.imshow(test_ds[8000])
plt.xlabel(labels[predict[8000]])
plt.xticks([])
plt.yticks([])
plt.show()

```

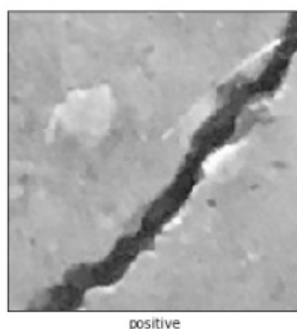


Рис. 9. Команда вывода фотографии с трещиной и предсказания наличия повреждений

```
plt.imshow(test_ds[50])
plt.xlabel(labels[predict[50]])
plt.xticks([])
plt.yticks([])
plt.show()
```

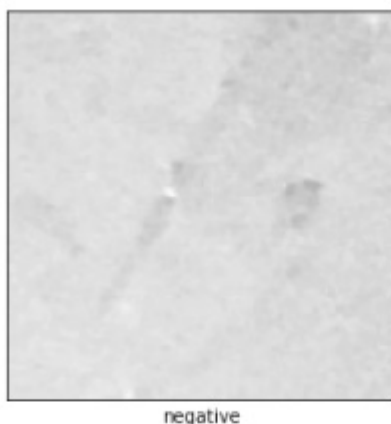


Рис. 10. Команда вывода фотографий без трещины и предсказания наличия повреждений

4. Создание функции загрузки фотографий из датасетов в систему нейросети (рис. 5).
5. Описание модели создаваемой нейросети, добавление сверточных и полносвязных слоев, указание количества и конфигурации ядер (рис. 6).
6. Компиляция и обучение нейросети. Указание количества эпох и шагов обучения. В нашей модели в качестве оптимизатора использовался алгоритм «Adaptive Moment Estimation», в качестве функции потерь использовалась функция `sparse_categorical_crossentropy` [2]. Результаты обучения получали в виде процентов точности определения наличия повреждений (рис. 7).

По итогам обучения полученная точность составила 98,46% (рис. 8).

7. Проверка точности обучения нейросети путем ручного выбора фотографии из выборки с трещиной (рис. 9) и без (рис. 10).

ВЫВОД

По результатам выполненной работы была получена ИС на основе обученной нейросети, способная отличать изображения целой поверхности от поверхности, на которой имеются видимые повреждения, например, трещины. В конце обучения точность работы нейросети достигла 98,46%. В перспективе написанная ИС может быть использована в качестве инструмента контроля, диагностики и анализа состояния поверхностей строений жилищного и производственного фонда.

ЛИТЕРАТУРА

1. Гудфеллоу Я. Глубокое обучение: книга / пер с англ. А.А. Слинкина. — 2-е изд., — М.: Издательство «ДМК Пресс». — 2018. — 652 с.
2. Николенко С. Глубокое обучение: книга — СПб.: Издательство «Питер». — 2018. — 480 с.

© Зиязетдинов Тимур Рифович (ziyazetdinov.timur@gmail.com),

Латыпов Булат Фиделевич (Linksmtv@gmail.com), Мухетдинов Амир Русланович (amir.muhetdinov@yandex.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»