

ТЕХНИЧЕСКИЕ НАУКИ



УДК 004

Нейронная сеть для обнаружения дефектов на транспортных средствах

В.Ф. Будзинский

Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

Аннотация. Рассмотрены системы, применяемые для обнаружения дефектов на кузовах автомобилей, проведено их сравнение. Проанализированы функциональность, поддержка и доступность систем. В результате система Cognex была признана наиболее подходящей, несмотря на ее дороговизну и отсутствие поддержки на русском языке. Авторами разработан собственный прототип нейронной сети для обнаружения дефектов на транспортном средстве с использованием библиотеки TensorFlow. Приводится пример кода, который определяет архитектуру нейронной сети, настраивает модель для обучения и оценивает ее производительность на тестовом наборе данных. Нейронная сеть может обнаруживать дефекты на изображениях кузова автомобиля размером 256x256x3 и выводить единственное значение, указывающее на наличие дефекта. Авторы полагают, что прототип может быть дополнительно доработан и интегрирован в существующие системы и сервисы в будущем.

Ключевые слова: нейронная сеть, обнаружение дефектов, машинное обучение, автомобильная промышленность

Neural Network for Vehicle Defect Detection

Vladislav F Budzinskiy

Don State Technical University (Rostov-on-Don, Russian Federation)

Abstract. The paper discusses the importance of defect detection on vehicle bodies and compares several existing systems used for this purpose. The comparison is based on functionality, support and availability, with the Cognex system being considered the most suitable despite its high cost and lack of support in Russian. As a result, an in-house prototype of a neural network for vehicle defect detection using TensorFlow library has been developed. The article provides a code example that defines the architecture of the neural network, configures the model for training and evaluates its performance on a test dataset. The neural network can detect defects on 256x256x3 car body images and output a single value indicating the presence of a defect. The authors believe that the prototype can be further developed and integrated into the existing systems and services in the future.

Keywords: neural network, defect detection, machine learning, automotive industry.

Введение. Автомобильная промышленность — одна из крупнейших и наиболее быстро развивающихся отраслей в мире. За последние годы производство автомобилей значительно возросло. С увеличением количества автомобилей растет и потребность в обнаружении дефектов в них. Выявление дефектов имеет решающее значение для обеспечения безопасности водителя и пассажиров. Цель данной работы — путем анализа и сравнения нескольких систем для обнаружения дефектов на кузове автомобиля выявить их преимущества и недостатки и создать свою собственную систему.

Основная часть. Анализ существующих систем. Для анализа были определены следующие признаки, по которым можно сравнивать данные системы: функциональность, поддержка и доступность. В качестве тестовых образцов выбраны три зарубежные системы: UVIScan PRO, Cognex и Uveye [1–3]. При исследовании функциональности рассматривались способности системы к поиску дефектов, учитывалось и то, как она встроена, в каком виде отображаются результаты, имеются ли возможности для дополнительной настройки. Важным критерием сравнения была поддержка. Проверялось наличие поддержки пользователей в России, а также возможность работы с системой на территории РФ. Другой ключевой параметр — это доступность, то есть стоимость, необходимость в специальных знаниях или навыках, требования к качеству загружаемых фотографий, устройства, необходимые для работы с системой.

Функциональность. Отличительной особенностью Cognex является возможность многогранной настройки системы под конкретные нужды конечного пользователя. Эта система имеет множество дополнительных

функций, которые могут быть настроены под конкретные требования пользователя. У UVIScan PRO и Uveye меньше возможностей настройки по сравнению с Cognex.

Поддержка. Позвонив в службу поддержки, мы обнаружили, что эти системы не предназначены для использования на территории Российской Федерации. Сотрудники общаются только на английском языке, есть сложности с приобретением этих систем. Документация на русском языке отсутствует.

Доступность. Поскольку ни одна из систем не предназначена для использования на территории России, стоимость их достаточно высока. Все, кроме Cognex, предполагают абонентскую систему оплаты. В контексте краткосрочного частного использования такая система достаточно выгодна, но для компаний, работающих на рынке купли-продажи автомобилей, более выгодной выглядит разовая покупка.

Лидером в этом сравнении является система Cognex по функциональности, поддержке и доступности. Однако у нее есть существенные недостатки: высокая стоимость внедрения и отсутствие поддержки в Российской Федерации. В связи с чем авторы статьи разработали собственную систему обнаружения дефектов на транспортных средствах.

Реализация прототипа нейронной сети. После проведения сравнительного анализа стало ясно, что системы по обнаружению дефектов на транспортных средствах широко используются в США и странах Европы, но в России они применяются редко. Было принято решение самим реализовать прототип нейронной сети для обнаружения дефектов на транспортных средствах для дальнейшей доработки и внедрения в существующие смежные системы и сервисы.

Для нейронной сети была использована библиотека TensorFlow, которая является программной библиотекой с открытым исходным кодом, разработанной командой Google Brain для построения и обучения моделей машинного обучения, включая нейронные сети.

Первая строка кода, `import tensorflow as tf`, импортирует библиотеку TensorFlow в текущее окружение Python.

Затем в коде определяется функция `create_model`, которая создает модель нейронной сети с определенной архитектурой. Архитектура включает в себя входной слой, свёрточные слои, плоский слой и плотные слои. Эта функция возвращает модель.

```
def create_model(input_shape):
    model = tf.keras.Sequential([
        # Input layer
        tf.keras.layers.InputLayer(input_shape=input_shape),
        # Convolutional layers
        tf.keras.layers.Conv2D(16, (3,3), padding='same', activation='relu'),
        tf.keras.layers.MaxPooling2D(pool_size=(2,2)),
        tf.keras.layers.Conv2D(32, (3,3), padding='same', activation='relu'),
        tf.keras.layers.MaxPooling2D(pool_size=(2,2)),
        tf.keras.layers.Conv2D(64, (3,3), padding='same', activation='relu'),
        tf.keras.layers.MaxPooling2D(pool_size=(2,2)),
        # Flatten layer
        tf.keras.layers.Flatten(),
        # Dense layers
        tf.keras.layers.Dense(128, activation='relu'),
        tf.keras.layers.Dropout(0.5),
        tf.keras.layers.Dense(1, activation='sigmoid')
    ])
    return model
```

Функция `model.compile` используется для настройки модели для обучения. Параметр `optimizer` задает алгоритм оптимизации, используемый во время обучения, `loss` задает функцию потерь, используемую для оценки производительности модели, а `metrics` задает метрику оценки, используемую для отслеживания производительности модели во время обучения.

```
model = create_model(input_shape=(256,256,3))
```

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Функция `model.fit` используется для обучения модели на наборе данных, обучающих изображений и меток. Параметр `epochs` задает количество итераций набора данных для обучения в процессе обучения. Параметр

validation_data используется для оценки производительности модели на проверочном наборе данных во время обучения.

```
model.fit(training_images, training_labels, epochs=10, validation_data=(validation_images, validation_labels))
```

Наконец, функция model.evaluate используется для оценки работы обученной модели на тестовом наборе данных. Потери при тестировании и точность выводятся на консоль.

```
test_loss, test_accuracy = model.evaluate(test_images, test_labels)
print(f'Test loss: {test_loss}')
print(f'Test accuracy: {test_accuracy}')
```

Здесь определяется архитектура нейронной сети, которая состоит из сверточных слоев, объединяющих слоев, сглаживающего слоя и плотных слоев. Нейронная сеть принимает изображения размером 256x256x3 (для RGB-изображений), и на выходе получается одно значение, которое указывает, содержит ли изображение дефект или нет. Для обучения модели используется бинарная функция потерь кросс-энтропии и оптимизатор Адама.

Чтобы использовать этот код для обнаружения дефектов на кузове автомобиля по фотографии, необходимо подготовить набор изображений автомобилей с дефектами и без них, разделить набор на обучающий, проверочный и тестовый. Также необходимо предварительно обработать изображения, изменив их размер до 256x256, нормализовав значения пикселей и преобразовав их в тензоры.

После подготовки набора данных и предварительной обработки изображений можно использовать метод fit() для обучения модели на обучающих данных и метод evaluate() для оценки модели на тестовых данных. Обучение предполагает 10 эпох, данный параметр может быть изменён для увеличения точности, но должен быть установлен после нескольких тестовых запусков, так как слишком большое количество эпох может привести к переобучению, когда кривая очень точно подстроится под точки, но пропадёт обобщающая способность. Переменные test_loss и test_accuracy дают представление о том, насколько хорошо модель работает на тестовом наборе.

Заключение. На основе анализа существующих систем для обнаружения дефектов на транспортных средствах был разработан прототип нейронной сети, который можно подстроить под конкретные нужды, дополнив его или изменив входящий датасет. Данная нейронная сеть может быть частью систем объявлений о купле-продаже автомобилей, автосервисов, служить для предварительной оценки стоимости автомобиля и расчета страховых выплат при авариях.

Одной из главных целей авторов статьи было создание доступной по стоимости разработки и обслуживания системы, которую возможно будет успешно применять на территории Российской Федерации. Существует возможность в дальнейшем настроить API соединение с данной нейронной сетью и разместить её на сервере для одновременного доступа большого количества пользователей.

Библиографический список

1. Система для обнаружения дефектов UVIScan PRO. *Secure One*. URL: <https://www.uviscan.com/products/uviscan-pro> (дата обращения: 01.04.2023).
2. Система для обнаружения дефектов Cognex. *Cognex*. URL: <https://www.cognex.com/solutions/cognex-tire-solutions> (дата обращения: 01.04.2023).
3. Система для обнаружения дефектов Uveye. *Uveye*. URL: <https://www.uveye.com/products/> (дата обращения: 01.04.2023).

Об авторе:

Будзинский Владислав Феликсович, студент кафедры «Информационные технологии» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), honpro99@gmail.com

About the Author:

Vladislav F Budzinskiy, student of the Information Technology Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), honpro99@gmail.com