

Заключение. Анализ вероятности банкротства организации — одно из признанных направлений финансового анализа, а общая цель финансового анализа — оказание помощи в выработке обоснованных и эффективных управленческих решений. Оценка вероятности банкротства может проводиться менеджментом самой организации для выявления слабых мест и предотвращения банкротства. А также анализ нередко нужен банкам, инвесторам, прочим действительным и потенциальным контрагентам и кредиторам [2]. Автоматизация расчёта вероятности банкротства с использованием сервисов Google позволит упростить процедуру и получать достоверную и обработанную информацию.

Список цитируемых источников

1. Савицкая, Г. В. Комплексный анализ хозяйственной деятельности предприятия : учебник / Г. В. Савицкая. — М. : Инфра-М, 2018. — 352 с.
2. Анализ и диагностика финансово-хозяйственной деятельности предприятий : учебник / Под ред. проф. В. Я. Позднякова — М. : Инфра-М, 2018. — 190 с.

УДК 004.032.26

П. П. Люцко, Е. Г. Шапович, А. И. Калько

Учреждение образования «Барановичский государственный университет», Барановичи,
Республика Беларусь

СИСТЕМА ОБНАРУЖЕНИЯ АЛКОГОЛЬНОГО ОПЬЯНЕНИЯ

Введение. В современном мире искусственные интеллектуальные системы завоевали большую популярность, благодаря богатым возможностям и эффективности использования. Существует множество задач, где необходимо практическое разрешение с использованием интеллектуальных систем уже сейчас: экономика и бизнес, робототехника, геологоразведка, математика, биофизика, авионика, безопасность и охранные системы, медицина и многие другие [1]. Такое обилие областей применения, говорит о том, что интеллектуальные системы — это уникальный набор для решения вопросов анализа и обработки большого объема данных, решения задач разного уровня сложности.

Исследование дорожного движения показывает, что усталость водителя может быть фактором, способствующим до 20 %, а из-за употребления алкоголя — около 31 % всех дорожно-транспортных происшествий [2].

Суть работы заключается в разработке система обнаружения алкогольного опьянения с использованием нейронной сети.

Основная часть. В качестве исходных данных к приложению выступают фотографии лица пьяных и трезвых людей в формате jpg и png, в количестве от 100 до 1000.

В качестве выходной информации будут выступать:

- модель нейронных сетей;
- библиотеки, используемые в построении нейронной сети;
- набор данных для обучения нейронных сетей;
- подробное пояснение каждого шага обучения нейронных сетей;
- результаты обучения нейронных сетей.

Для реализации была выбрана среда программирования Visual Code.

В качестве нейронной сети выбрана сверточная нейронная сеть с одним входным слоем, двумя сверточными и двумя субдискретизирующими. Размерность входного слоя 512×512 , первого сверточного слоя 150×150 , второго слоя подвыборки 75×75 . Данные слои состоят из 32-х карт признаков. Размерности второго слоя свертки и субдискретизирующего слоя 37×37 и 18×18 соответственно.

Структура модели нейронной сети представлена на рисунке 1.

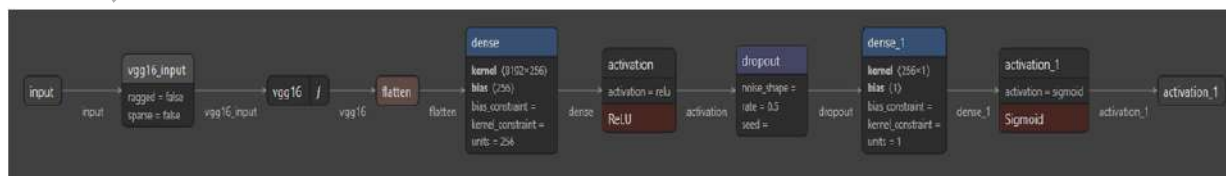


Рисунок 1 — Структура нейронной сети

Процесс обучения предполагает наличие лишь входных данных. Алгоритмы обучения нейронных сетей корректируют весовые коэффициенты таким образом, чтобы нейронная сеть могла из схожих по некоему принципу данных на «входе» выдать результат, обнаруживающий другие взаимосвязи и закономерности между этими данными. В процессе обучения происходит выделение параметров, характерных для моделей обучающего материала, и дальнейшее объединение этих моделей в группировки по схожим признакам.

Процесс обучения делится на несколько шагов:

1. Значения пикселя каждого изображения поступают на нейроны входного слоя.
2. Они проходят в скрытые слои, где они умножаются на веса, складываются, смещаются и поступают в соответствующую функцию активации.
3. На выходе из функции Relu извлекается вероятность для каждого класса.
4. После результат сравнивается с целевой переменной, делается расчёт ошибки.
5. Алгоритм оптимизации стремится уменьшить ошибку и соответствующим образом меняет веса.
6. Процесс повторяется, но уже с новыми весами.

Функция Relu в Python является наиболее распространенным выбором функции активации в мире глубокого обучения. Данная функция обеспечивает самые лучшие результаты и в то же время очень эффективна с точки зрения вычислений.

Далее на рисунке 2 представлен процесс обучения нейросети.

После анализа нейросеть может вынести вердикт к какому классу относится изображение (рисунок 5).

После подключения уже обученной модели и прогона изображения, можно получить информацию (карты признаков) о том, как нейросеть выделяет признаки на нем. На рисунке 4 можно увидеть, как нейросеть выделила основные черты лица, а также некоторые другие признаки, которые в дальнейшем будут исключены.

Из рисунка 3 видно, что при обучении оптимизатор изменяет веса, в следствии чего потери, при обучении, уменьшаются. По окончании обучения выводится точность прогноза, которой удалось достичь (рисунок 3).

На рисунке 5 приведено 3 строки: первая показывает сколько было обработано изображений; вторая указывает на класс, к которому было отнесено анализируемое изображение; третья, показывает классы, к которым нейронная сеть ссылается и их индексы. Метка “Sober” означает, что человек на изображении трезвый, а “Drunk” — в состоянии алкогольного опьянения.

```
Epoch 1/30
34/34 [-----] - 2s 36ms/step - loss: 0.7276 - accuracy: 0.4706 - val_loss: 0.6886 - val_accuracy: 0.5000
Epoch 2/30
34/34 [-----] - 1s 39ms/step - loss: 0.6891 - accuracy: 0.5882 - val_loss: 0.6553 - val_accuracy: 0.5000
Epoch 3/30
34/34 [-----] - 1s 29ms/step - loss: 0.5823 - accuracy: 0.7647 - val_loss: 0.5654 - val_accuracy: 0.7857
Epoch 4/30
34/34 [-----] - 1s 29ms/step - loss: 0.3928 - accuracy: 0.8676 - val_loss: 0.4227 - val_accuracy: 0.7143
Epoch 5/30
34/34 [-----] - 1s 32ms/step - loss: 0.2857 - accuracy: 0.8824 - val_loss: 0.7690 - val_accuracy: 0.7143
Epoch 6/30
34/34 [-----] - 1s 32ms/step - loss: 0.1391 - accuracy: 0.9412 - val_loss: 0.5958 - val_accuracy: 0.7857
Epoch 7/30
34/34 [-----] - 1s 30ms/step - loss: 0.1582 - accuracy: 0.9559 - val_loss: 0.4893 - val_accuracy: 0.8571
Epoch 8/30
34/34 [-----] - 1s 28ms/step - loss: 0.2129 - accuracy: 0.8971 - val_loss: 0.6105 - val_accuracy: 0.7857
Epoch 9/30
34/34 [-----] - 1s 29ms/step - loss: 0.1283 - accuracy: 0.9559 - val_loss: 0.8710 - val_accuracy: 0.7143
Epoch 10/30
34/34 [-----] - 1s 28ms/step - loss: 0.1001 - accuracy: 0.9706 - val_loss: 0.8368 - val_accuracy: 0.8571
Epoch 11/30
34/34 [-----] - 1s 29ms/step - loss: 0.0370 - accuracy: 0.9853 - val_loss: 1.8327 - val_accuracy: 0.7857
Epoch 12/30
34/34 [-----] - 1s 29ms/step - loss: 0.0881 - accuracy: 0.9706 - val_loss: 1.0887 - val_accuracy: 0.8571
Epoch 13/30
34/34 [-----] - 1s 29ms/step - loss: 0.1190 - accuracy: 0.9706 - val_loss: 1.1558 - val_accuracy: 0.7857
Epoch 14/30
34/34 [-----] - 1s 29ms/step - loss: 0.1650 - accuracy: 0.9706 - val_loss: 2.1189 - val_accuracy: 0.7857
Epoch 15/30
34/34 [-----] - 1s 29ms/step - loss: 0.0842 - accuracy: 0.9559 - val_loss: 1.4485 - val_accuracy: 0.7143
Epoch 16/30
34/34 [-----] - 1s 29ms/step - loss: 0.0225 - accuracy: 1.0000 - val_loss: 1.1213 - val_accuracy: 0.7143
Epoch 17/30
34/34 [-----] - 1s 29ms/step - loss: 0.0208 - accuracy: 1.0000 - val_loss: 2.0236 - val_accuracy: 0.7143
Epoch 18/30
34/34 [-----] - 1s 29ms/step - loss: 0.0021 - accuracy: 1.0000 - val_loss: 2.1742 - val_accuracy: 0.7143
Epoch 19/30
34/34 [-----] - 1s 32ms/step - loss: 0.0054 - accuracy: 1.0000 - val_loss: 2.8574 - val_accuracy: 0.7143
Epoch 20/30
34/34 [-----] - 1s 30ms/step - loss: 0.1342 - accuracy: 0.9412 - val_loss: 0.4982 - val_accuracy: 0.9286
Epoch 21/30
34/34 [-----] - 1s 31ms/step - loss: 0.1340 - accuracy: 0.9559 - val_loss: 0.4463 - val_accuracy: 0.8571
Epoch 22/30
34/34 [-----] - 1s 28ms/step - loss: 0.0403 - accuracy: 0.9853 - val_loss: 0.8555 - val_accuracy: 1.0000
Epoch 23/30
```

Рисунок 2 — Обучение нейросети

```
Epoch 30/30
34/34 [=====]
d:\Projects\VSCode\PTH\try1\main.py:1
scores = model.evaluate_generator(t
Точность: 50.00%
```

Рисунок 3 — Точность прогноза по окончании процесса обучения

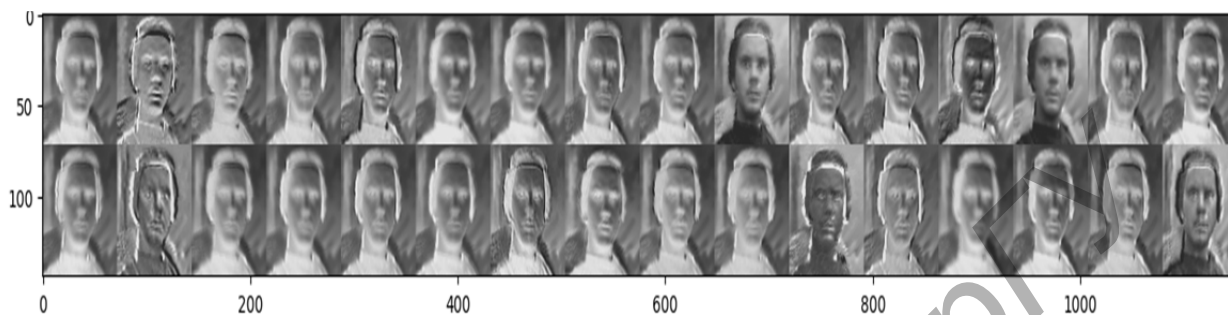


Рисунок 4 — Карта признаков

```
1/1 [=====] - 2s 2s/step
[[1.]]
{'Drunk': 0, 'Sober': 1}
```

Рисунок 5 — Результат проверки изображения

Заключение. В ходе исследования была разработана нейронная сеть для обнаружения алкогольного опьянения по изображению.

Разработанная нейронная сеть выдаёт приемлемые результаты, а также имеет возможность к развитию за счёт открытого доступа к моделям обученных нейронных сетей. Оно позволяет с вероятностью выше 50 % спрогнозировать в каком состоянии находится человек: трезвости или алкогольном опьянении. Таким образом приложение может быть востребованным в таких сферах как: контроль персонала, маркетинг, правовая и других сферах, где производится контроль персонала или общественности.

Улучшить приложение можно путём добавления большого количества новых изображений для нейронной сети и её дальнейшее переобучение.

Список цитируемых источников

1. Шапович, Е. Г. Интеллектуальная система для распознавания диатомовых водорослей с применением нейронных сетей / Е. Г. Шапович // Векторы инновационного развития : материалы I Междунар. науч.-практич. конф., Барановичи, 11 дек. 2020 г. : в 2 ч. / М-во образования Респ. Беларусь, Баранович. гос. ун-т, редкол.: В. В. Климук (гл. ред.) [и др.]. — Барановичи : БарГУ, 2020. — Ч. 1. — С. 52—54.
2. Всемирный доклад о предупреждении дорожно-транспортного травматизма / пер. с англ. — М. : Издательство «Весь Мир», 2004. — 280 с.

УДК 004.942

П. П. Люцко, Н. Ю. Кондратчик

*учреждение образования «Барановичский государственный университет», Барановичи,
Республика Беларусь*

КОМПОНОВКА И РАЗНЕСЕНИЕ ИЗДЕЛИЯ В ПРОГРАММНОМ ПАКЕТЕ BLENDER

Введение. В настоящее время Blender пользуется большой популярностью среди бесплатных 3D-редакторов в связи с его быстрым стабильным развитием и технической поддержкой. В программе есть всё необходимое для создания качественных работ не только в игровой индустрии и области спецэффектов, создании анимации, но и возможности для моделирования деталей, которые можно скомпоновать в общую сборку с последующим разнесением компонентов и 3D-печатью.