

**О. С. ДОГАДАЙЛО, В. Б. УСПЕНСЬКИЙ**

### **ВИБІР АРХІТЕКТУРИ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ АВТОВОДІЯ ТРАНСПОРТНОГО ЗАСОБУ**

В даній роботі розглядається задача вибору архітектури нейронної мережі для створення автопілоту. Був створений автопілот для віртуального транспортного засобу, що може рухатися визначеним маршрутом та реагувати на різні сигнали світлофора. Вибрана архітектура, а саме згортоква нейронна мережа, має високу ефективність в задачі розпізнавання зображень. Автопілот складається з двох згорткових нейронних мереж, одна розпізнає маршрут руху, інша розпізнає сигнали світлофора. Через велику кількість шумів, фотографії для розпізнавання світлофорів були оброблені для підсилення червоного каналу та занулення зеленого і синього, що допомогло в розпізнаванні червоного та жовтого кольорів. Як середовище для навчання нейронних мереж та перевірки працездатності автопілоту, в цілому, була створена двовимірна гра з видом зверху. Даний автопілот, на відміну від інших, використовує лише зображення для орієнтації в просторі. Тестування моделі автопілоту показало майже 100 % точність на розпізнаванні маршруту та сигналів світлофора. Позитивний результат тестування показав, що автопілот може виконувати керування в простому середовищі і це дає можливість ускладнити робоче середовище. Використовуючи тільки зображення, такий автопілот є дешевшим за існуючі. Актуальність даної роботи ґрунтується на дослідженнях збільшення кількості транспортних засобів та шкідливих викидів в атмосферу у майбутньому. У статті розглянуто літературні джерела, обґрунтування вибору архітектури нейронної мережі, опис програмної реалізації, показані результати проведення тестування, у висновках вказаний можливий напрямок розвитку даної теми.

**Ключові слова:** автопілот, задача автокермування, автоматичне керування, нейронна мережа, згортоква нейронна мережа, розпізнавання зображень.

**А. С. ДОГАДАЙЛО, В. Б. УСПЕНСКИЙ**

### **ВЫБОР АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ ДЛЯ АВТОВОДИТЕЛЯ ТРАНСПОРТНОГО СРЕДСТВА**

В данной работе рассматривается задача выбора архитектуры нейронной сети для создания автопилота. Был создан автопилот для виртуального транспортного средства, которое может двигаться по определенному маршруту и реагировать на разные сигналы светофора. Выбранная архитектура, а именно сверточная нейронная сеть, имеет высокую эффективность в задаче распознавания изображений. Автопилот состоит из двух сверточных нейронных сетей, одна распознает маршрут движения, другая распознает сигналы светофора. Из-за большого количества шумов фотографии для распознавания светофоров были обработаны для усиления красного канала и зануления зеленого и синего, что помогло в распознавании красного и желтого цветов. Как среда для обучения нейронных сетей и проверки работоспособности автопилота в целом, была создана двумерная игра с видом сверху. Этот автопилот, в отличие от других, использует только изображения для ориентировки в пространстве. Тестирование модели автопилота показало почти 100 % точность на распознавании маршрута и сигналов светофора. Положительный результат тестирования показал, что автопилот может выполнять управление в простой среде, что позволяет усложнить рабочую среду. Используются только изображения, такой автопилот дешевле существующих. Актуальность данной работы основывается на исследованиях увеличения количества транспортных средств и вредных выбросах в атмосферу в будущем. В статье рассмотрены литературные источники, обоснование выбора архитектуры нейронной сети, описание программной реализации, показаны результаты проведения тестирования, в выводах указано возможное направление развития данной темы.

**Ключевые слова:** автопилот, задача автовождения, автоматическое управление, нейронная сеть, сверточная нейронная сеть, распознавание изображений.

**O. S. DOHADAILO, V. B. USPENSKYI**

### **CHOOSING A NEURAL NETWORK ARCHITECTURE FOR A VEHICLE AUTOPILOT**

In the paper the task of choosing a neural network architecture for creating an autopilot is considered. An autopilot was created for a virtual vehicle that can move along a defined route and respond to various traffic lights. The selected architecture, namely a convolutional neural network, has high efficiency in the task of image recognition. Autopilot consists of two convolutional neural networks, one of which recognizes the driving route, the other recognizes traffic light signals. Due to the large amount of noise, the traffic light recognition photos were processed to enhance the red channel and null the green and blue, which helped in the recognition of red and yellow colors. As an environment for training neural networks and testing the performance of the autopilot in general, a two-dimensional game with a top view was created. The autopilot model showed almost 100% accuracy in recognizing the route and traffic lights in the test environment. The positive test result showed that the autopilot can perform control in a simple environment and this gives the opportunity to complicate the operating environment. The proposed autopilot uses only images to navigate in space, which distinguishes it from the other existing autopilots and, in particular, makes it cheaper. The relevance of this work is based on studies of the increase in the number of vehicles and harmful emissions into the atmosphere in the future. In the paper the literary sources are analyzed, the rationale for choosing a neural network architecture is explained, the software implementation is described, the results of testing are shown, and the possible direction of development of this topic is indicated in the conclusions.

**Key words:** autopilot, auto driving task, automatic control, neural network, convolutional neural network, image recognition.

**Вступ.** Нині сфера використання *інформаційних технологій* безперервно розширюється. *Системи інтелектуального управління* – це новий клас комп'ютеризованих систем, які мають на меті моделювання та аналіз інтелектуальних завдань, а також підтримку діяльності людини за їх вирішенням. Підвищена складність систем інтелектуального управління визначається, по-перше, складністю аналізу поведінки суттєво нелінійних об'єктів управління та, по-друге, складністю синтезу алгоритмів управління такими об'єктами. Додатковим чинником складності є взаємодія людини із комп'ютеризованими засобами управління. Тому дослідження концептуальних та прикладних аспектів створення подібних систем, а також факторів, що визначають їх ефективне та надійне функціонування, є важливою та актуальною науково-технічною проблемою.

**Аналіз останніх досліджень.** Основною метою інтелектуалізації є забезпечення можливості управління

системами, що адаптуються у динамічно змінному середовищі при забезпеченні необхідних показників якості, швидкодії та стійкості. Для досягнення цієї мети необхідне вдосконалення існуючих та створення нових підходів, концепцій, технологій, методів та моделей інтелектуального управління [1].

Одним з напрямків застосування системи інтелектуального управління є *автоматизація керування автомобілем*. Ідея самокерованого автомобіля така ж, як і винахід транспортного засобу. До останніх років самокерований автомобіль досяг рівня лабораторного застосування завдяки розвитку *сенсорної технології*, обчислювальної техніки та мобільного Інтернету. Самокеровані автомобілі, які з'являються в пробній версії *безпілотних* автомобілів, представляють найвищий рівень досягнень у цій галузі [2].

Мотивації для впровадження *автоматичного керування (АК)* є різноманітними і враховують, серед іншого, екологічні, демографічні, соціальні та економічні аспекти. За статистикою, 68 % населення ЄС, включаючи асоційовані держави, проживає в міських районах, а у шести європейських містах проживає понад три мільйони мешканців [3]. За прогнозами *Всесвітньої організації охорони здоров'я (ВОЗ)*, ще до 2030 року майже третина населення світу житиме в містах. Це виправдовує потребу в технологіях, які підтримуватимуть міську мобільність. АК пропонує чудове рішення завдяки своїй здатності оптимізувати транспортні потоки, зменшуючи затори та аварії. Таким чином, скорочення споживання палива, а отже, зменшення вуглекислого газу та інших шкідливих викидів, може бути значним [4]. Також слід очікувати значного підвищення безпеки дорожнього руху та комфорту за рахунок уникнення людських помилок керування. Це може підвищити реакцію невпевнених і літніх водіїв і забезпечити їх включення в мобільність. У певному сенсі можна говорити про «персоналізований» транспорт, де необхідний рівень автоматизації може бути індивідуально адаптований відповідно до потреб замовника. Це означає, що час, коли водій не приділяє повної уваги процесу керування, може бути використано, наприклад, для відпочинку, або для роботи, що підвищить комфорт для співробітників, які часто перебувають у русі. Тому можна очікувати, що АК спричинить підвищення продуктивності [5].

Відмінність даної роботи від інших автопілотів в тому, що для прийняття рішень використовуються лише дані з камери, а в інших відомих автопілотах наприклад, *компанії Tesla*, використовується велика кількість різноманітних датчиків, які підвищують вартість автопілоту.

**Постановка задачі.** Розробка архітектури *нейронної мережі* для задачі автокерування, реалізація програми для керування об'єктом, а саме – переміщення за маршрутом та розпізнавання сигналів світлофора.

Вхідні дані:

- фото певних частин екрану;

Вихідні дані:

- напрямок руху об'єкта (наліво, прямо або направо);
- наявність руху загалом (їхати, зупинитися).

В задачі переміщення за маршрутом є три типи руху:

- прямо;
- наліво;
- направо.

Для руху *прямо* треба створити тренувальну множину, в якій є фото прямих фрагментів маршруту, але прямі лінії повинні знаходитися майже по центру.

Для руху *наліво* та *направо* треба створити тренувальну множину, в якій є фото прямих фрагментів дороги дуже віддалених від центру та під надто великим кутом.

В задачі розпізнавання сигналів світлофорів є два типи дій:

- рух;
- зупинка.

Для руху треба створити тренувальну множину, в якій є фото зеленого сигналу світлофора та повної відсутності світлофорів.

Для *зупинки* треба створити тренувальну множину з фото червоного та жовтого сигналів світлофора.

Критерієм працездатності автопілоту є швидкодія і точність прийняття рішень на тестовій множині та у реальному часі.

**Вибір архітектури нейронної мережі.** Для задачі розпізнавання використана *згорткова нейронна мережа (ЗНМ)*. Згорткова нейронна мережа – це тип *моделі глибокого навчання* для обробки даних, які мають сітку, яка аналогічна структурі організації зорової кори тварин [7, 8, 9], і призначена для автоматичного та адаптивного вивчення *просторової ієрархії функцій*, від низьких – до візерунків високого рівня. ЗНМ – це математична конструкція, яка зазвичай складається з трьох типів шарів (або будівельних блоків): шарів згортки, об'єднання та повністю пов'язаних шарів. Перші два, шари згортки та об'єднання, виконують вилучення загальних ознак, тоді як третій, повністю пов'язаний шар, відображає вилучені об'єкти в кінцевий результат, наприклад, класифікацію (рис. 1) Рівень згортки відіграє ключову роль у ЗНМ, який складається з набору математичних операцій, таких

як згортка, спеціалізований тип лінійної операції. У цифрових зображеннях значення пікселів зберігаються в двовимірній (2D) сітці, тобто в масиві чисел, і невелика сітка параметрів, що називається ядром (оптимізований екстрактор функцій), застосовується до кожної позиції зображення, що робить ЗНМ (CNN) високо ефективною для обробки зображень, оскільки функція може виникнути в будь-якому місці зображення [10]. Оскільки один шар передає свій вихід на наступний шар, витягнуті об'єкти можуть ієрархічно та поступово ставати все більш складними. Процес оптимізації таких параметрів, як ядра, називається навчанням, яке виконується таким чином, щоб мінімізувати різницю між виходами та основними мітками істини за допомогою алгоритму оптимізації, який називається зворотним поширенням і градієнтним спуском, серед інших [6, 9].

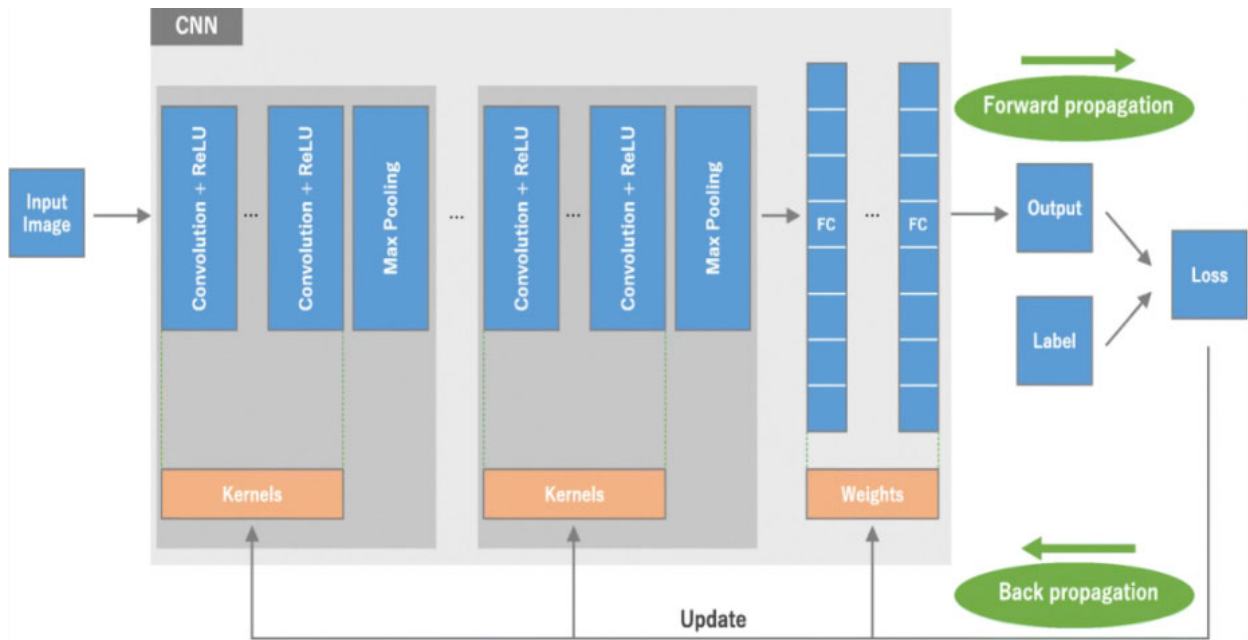


Рис. 1 – Огляд згорткової нейронної мережі.

**Програмна реалізація та результати.** Комп'ютерна програма складається з двох класів нейронних мереж, які відповідають за розпізнавання лінії маршруту та сигналів світлофора. Вони агрегують свої методи прийняття рішень головному класу *Game*, який відповідає за малювання ігрової сцени, а саме: фон, автомобілі і лінії маршруту (рис. 2).

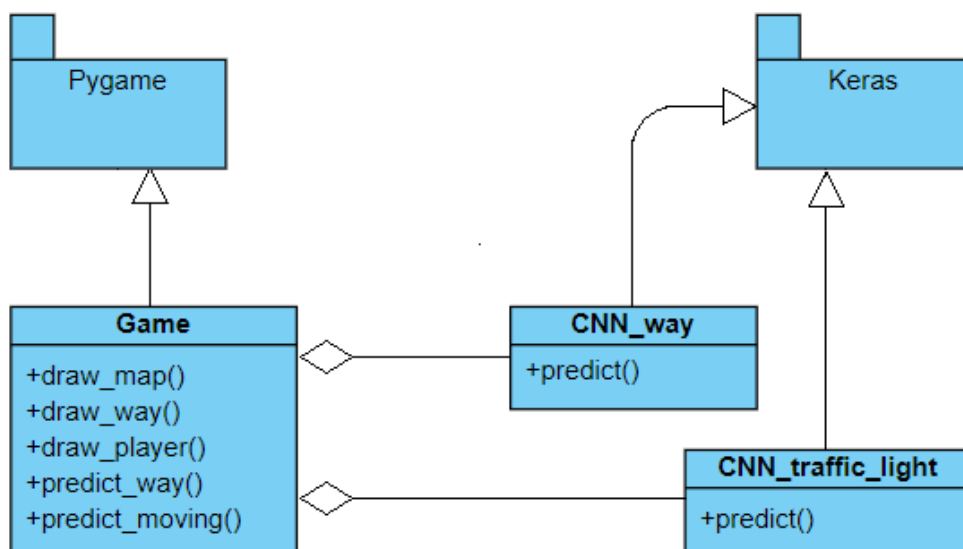


Рис. 2 – Діаграма класів.

Спочатку треба було зробити програму, яка б згенерувала навчальну множину даних. Програма робить фото певної частини екрану, де знаходиться об'єкт керування, перетворює його у вектор та зберігає у файлі разом зі значенням напрямку руху.

Для навчання нейронної мережі пересуватися заданим маршрутом, програма створила більше 14 тисяч фото, завдяки яким можна починати навчання. Для створення моделі було використано *бібліотеку Keras* та її клас *Sequential*. Модель представляє собою мережу із чотирьох шарів згортки та двох шарів звичайної нейронної мережі (рис. 3).

```
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 98, 98, 32)	896
max_pooling2d (MaxPooling2D)	(None, 49, 49, 32)	0
conv2d_1 (Conv2D)	(None, 47, 47, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 23, 23, 64)	0
conv2d_2 (Conv2D)	(None, 21, 21, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 10, 10, 128)	0
conv2d_3 (Conv2D)	(None, 8, 8, 128)	147584
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 128)	0
dropout (Dropout)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 512)	1049088
dense_1 (Dense)	(None, 3)	1539

```

=====
Total params: 1,291,459
Trainable params: 1,291,459
Non-trainable params: 0

```

Рис. 3 – Архітектура нейронної мережі.

Для запобігання перенавчання, була включена функція відключення нейронів. Функції активації були вибрані «relu». Після навчання була перевірена точність, вона складала ~ 99 % (рис. 4). Можна побачити, що знадобилася лише одна епоха навчання для того, щоб точність була максимальною.

```

history = model.fit_generator(
Epoch 1/5
120/120 [=====] - 163s 1s/step - loss: 0.0775 - accuracy: 0.9687 - val_loss: 0.0639 - val_accuracy: 0.9953
Epoch 2/5
120/120 [=====] - 153s 1s/step - loss: 0.0045 - accuracy: 0.9986 - val_loss: 0.1208 - val_accuracy: 0.9920
Epoch 3/5
120/120 [=====] - 146s 1s/step - loss: 3.2555e-04 - accuracy: 0.9999 - val_loss: 0.1201 - val_accuracy: 0.9920
Epoch 4/5
120/120 [=====] - 147s 1s/step - loss: 3.2759e-05 - accuracy: 1.0000 - val_loss: 0.1249 - val_accuracy: 0.9920
Epoch 5/5
120/120 [=====] - 147s 1s/step - loss: 6.2299e-06 - accuracy: 1.0000 - val_loss: 0.1276 - val_accuracy: 0.9920

```

Рис. 4 – Результати навчання нейронної мережі.

Далі йде навчання нейронної мережі, яка відповідає за розпізнавання сигналів світлофорів. Для розпізнавання світлофорів було зроблено понад 23 тисячі фото як тренувальна множина. Архітектура нейронної мережі складається з чотирьох шарів згортки та чотирьох шарів звичайної нейронної мережі (рис. 5).

```

cmd
<1> cmd
=====
Layer (type)                Output Shape                Param #
=====
conv2d (Conv2D)              (None, 98, 98, 32)         896
max_pooling2d (MaxPooling2D) (None, 49, 49, 32)         0
conv2d_1 (Conv2D)            (None, 47, 47, 64)         18496
max_pooling2d_1 (MaxPooling2D) (None, 23, 23, 64)         0
conv2d_2 (Conv2D)            (None, 21, 21, 128)        73856
max_pooling2d_2 (MaxPooling2D) (None, 10, 10, 128)        0
conv2d_3 (Conv2D)            (None, 8, 8, 128)          147584
max_pooling2d_3 (MaxPooling2D) (None, 4, 4, 128)         0
dropout (Dropout)            (None, 4, 4, 128)         0
flatten (Flatten)            (None, 2048)               0
dense (Dense)                 (None, 512)                1049088
dense_1 (Dense)               (None, 256)                131328
dense_2 (Dense)               (None, 32)                 8224
dense_3 (Dense)               (None, 2)                  66
=====
Total params: 1,429,538
Trainable params: 1,429,538
Non-trainable params: 0
=====
cmd.exe*[64]:3308

```

Рис. 5 – Архітектура нейронної мережі.

Декілька перших спроб навчити нейронну мережу були невдалими. «Шуми» на зображенні заважали правильно ідентифікувати сигнали світлофора, нейронна мережа бачила їх там, де їх не було. Ця проблема була вирішена підсиленням червоного та зведенням до нуля зеленого і синього каналів кожного пікселя на зображенні. Таким чином вдалося виділити червоний та жовтий сигнали світлофора, це полегшило розпізнавання потрібної інформації та призвело до 100 % точності на тестовій множині (рис. 6).

```

cmd
c:\Python\game\model_light_traffic.py:101: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  history = model.fit_generator(
Epoch 1/4
236/236 [=====] - 306s 1s/step - loss: 0.0657 - accuracy: 0.9764 - val_loss: 4.7703e-05 - val_accuracy: 1.0000
Epoch 2/4
236/236 [=====] - 302s 1s/step - loss: 0.0019 - accuracy: 0.9994 - val_loss: 6.0772e-06 - val_accuracy: 1.0000
Epoch 3/4
236/236 [=====] - 301s 1s/step - loss: 0.0015 - accuracy: 0.9995 - val_loss: 2.5205e-05 - val_accuracy: 1.0000
Epoch 4/4
236/236 [=====] - 313s 1s/step - loss: 0.0041 - accuracy: 0.9993 - val_loss: 1.5274e-05 - val_accuracy: 1.0000
user@DESKTOP-15CNC1D c:\Python\game
* |
cmd.exe*[64]:3308

```

Рис. 6 – Результати навчання нейронної мережі.

Далі модель була перевірена в реальних умовах. В програмі є режим при якому можна протестувати дану модель (рис. 7).

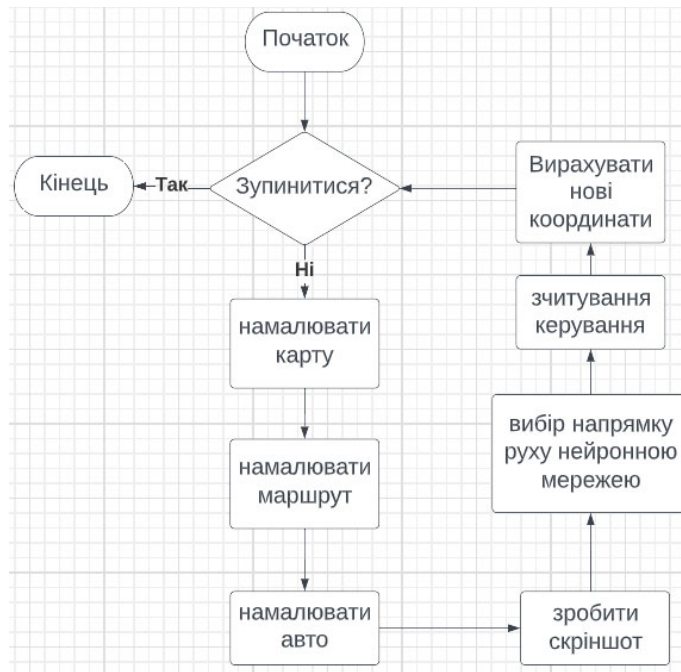


Рис. 7 – Загальна логіка роботи програми.

Перевірка автопілоту в реальному часі (рис. 8).

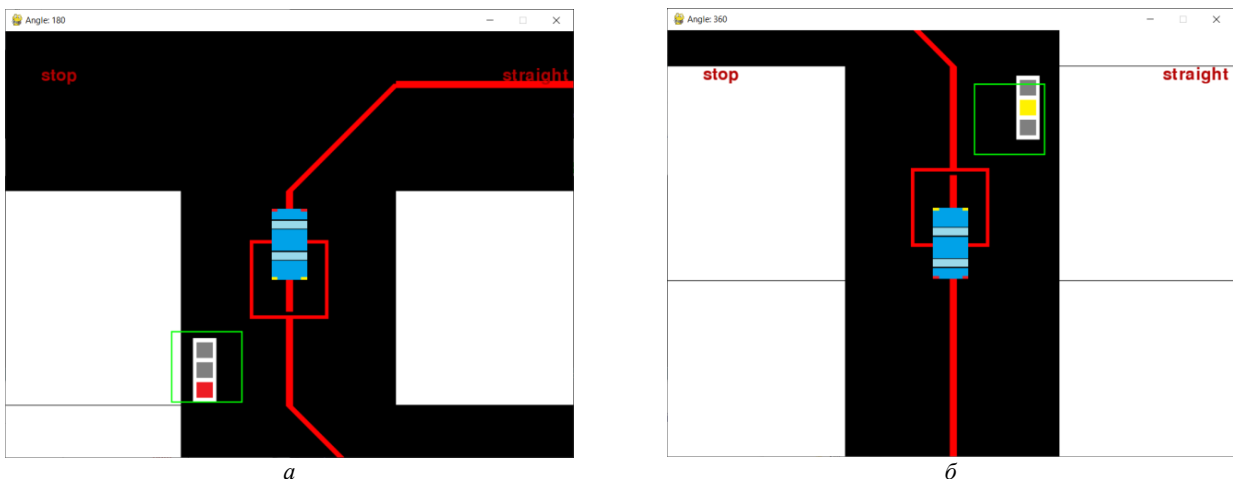


Рис. 8 – Приклади роботи автопілоту: а – автопілот реагує на червоний сигнал світлофора; б – автопілот реагує на жовтий сигнал світлофора.

**Висновки.** Будь-яка галузь людської діяльності йде шляхом автоматизації, і керування транспортним засобом не є виключенням. Як показують деякі дослідження, близько третини населення Землі, вже скоро буде жити у містах. Такий розвиток подій тягне за собою велику кількість проблем, які потребують вирішення. Збільшення населення міст призведе до збільшення кількості автомобілів, а це, в свою чергу, призведе до підвищення рівня викидів вуглекислого газу та заторів на дорогах. Описані вище проблеми можуть бути частково або повністю вирішені впровадженням автоматичних систем керування, які дозволять оптимізувати роботу транспорту. В результаті виконання даної роботи були проведені дослідження та оптимізація нейронних мереж, використані можливості згорткової нейронної мережі, та створена програма-автопілот для двовимірної гри, а саме – для розпізнавання світлофорів та напрямку руху. Згорткова нейронна мережа показала свою ефективність у даному середовищі з розпізнавання об'єктів на зображеннях. Успіх у використанні даної архітектури і типу нейронних мереж, говорить про можливість ускладнення середовища, а саме: додання різних видів перешкод, таких як інші автомобілі, пішоходи, стовпи та інше, і навчання автопілоту коректно реагувати на них. Позитивні результати у подальших дослідженнях можуть призвести до розробки більш дешевих систем автопілоту, що в свою чергу зробить переміщення дорогами більш безпечнішим, скоротить викиди вуглекислого газу та зменшить затори.

## Список літератури

1. Pavlov V., Shepetukha Y., Melnikov S., Volkov A. Intelligent Control: Approaches, Results and Prospects of Development // *Kibernetika i vychislitel'naâ tehnika*. – 2017. – С. 30 – 48.
2. Zhao J., Liang B., Chen Q. The key technology toward the self-driving car // *International Journal of Intelligent Unmanned Systems*. – 2018. – С. 2 – 20.
3. ERTRAC, EPoSS and SMART GRIDS, European Roadmap Electrification of Road Transport, 2nd Edition. – 2012. – С. 46.
4. Meyer G., Deix S. Research and Innovation for Automated Driving in Germany and Europe // *Road Vehicle Automation*. – 2014. – С. 71 – 81.
5. Meyer G., Dokic J., Müller B. Elements of a European Roadmap on Smart Systems for Automated Driving // *Road Vehicle Automation 2*. – 2015. – С. 153 – 159.
6. Yamashita R., Nishio M., Do R. K. G., Togashi K. Convolutional neural networks: an overview and application in radiology // *Insights into Imaging*. – 2018. – С. 611 – 629.
7. Ketkar N. *Deep Learning with Python: A Hands-on Introduction*, 1st ed. – Apress, 2017. – 243 с.
8. Manaswi N. K. Understanding and Working with Keras // *Deep Learning with Applications Using Python*. – CA : Apress, 2018. – С. 31 – 43.
9. Prangishvili A., Namicheishvili O., Ramazashvili M. Convolutional Neural Networks // *Works of Georgian Technical University*. – 2020. – С. 33 – 56.
10. Joshi P. *Python Machine Learning Cookbook* O'Reilly Media; 1st edition. – 2016. – 366 с.

## References (transliterated)

1. Pavlov V., Shepetukha Y., Melnikov S., Volkov A. Intelligent Control : Approaches, Results and Prospects of Development. *Kibernetika i vychislitel'naâ tehnika*. 2017, pp. 30–48.
2. Zhao J., Liang B., Chen Q. The key technology toward the self-driving car. *International Journal of Intelligent Unmanned Systems*. 2018, pp. 2–20.
3. ERTRAC, EPoSS and SMART GRIDS, European Roadmap Electrification of Road Transport, 2nd Edition. 2012, 46 p.
4. Meyer G., Deix S. Research and Innovation for Automated Driving in Germany and Europe. *Road Vehicle Automation*. 2014, pp. 71–81.
5. Meyer G., Dokic J., Müller B. Elements of a European Roadmap on Smart Systems for Automated Driving. *Road Vehicle Automation 2*. 2015, pp. 153–159.
6. Yamashita R., Nishio M., Do R. K. G., Togashi K. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*. 2018, pp. 611–629.
7. Ketkar N. *Deep Learning with Python : A Hands-on Introduction, 1st ed.* Apress, 2017. 243 p.
8. Manaswi N. K. Understanding and Working with Keras. *Deep Learning with Applications Using Python*. CA, Apress, 2018, pp. 31–43.
9. Prangishvili A., Namicheishvili O., Ramazashvili M. Convolutional Neural Networks. *Works of Georgian Technical University*. 2020, pp. 33–56.
10. Joshi P. *Python Machine Learning Cookbook O'Reilly Media, 1st edition*. 2016. 366 p.

Надійшла (received) 17.09.2022

## Відомості про авторів / Сведения об авторах / Information about authors

**Догдайло Олександр Сергійович** – магістр, студент, Національний технічний університет «Харківський політехнічний інститут», м. Харків; тел.: (098) 396-25-81; e-mail: oleksandr.dohadailo@infiz.khpi.edu.ua.

**Догдайло Олександр Сергеевич** – магістр, студент, Национальный технический университет «Харьковский политехнический институт», г. Харьков; тел.: (098) 396-25-81; e-mail: oleksandr.dohadailo@infiz.khpi.edu.ua.

**Dohadailo Oleksandr Serhiiovych** – master, student, National Technical University «Kharkiv Polytechnic Institute», Kharkiv; tel.: (098) 396-25-81; e-mail: oleksandr.dohadailo@infiz.khpi.edu.ua.

**Успенський Валерій Борисович** – доктор технічних наук, професор, доцент кафедри математичного моделювання процесів та систем, Національний технічний університет «Харківський політехнічний інститут», м. Харків; тел.: (057) 707-64-54; e-mail: valerii.uspenskyi@khpi.edu.ua.

**Успенский Валерий Борисович** – доктор технической наук, профессор, доцент кафедры математического моделирования процессов и систем, Национальный технический университет «Харьковский политехнический институт», г. Харьков; тел.: (057) 707-64-54; e-mail: valerii.uspenskyi@khpi.edu.ua.

**Uspenskyi Valerii Borysovych** – Doctor of Technical Sciences, Professor, Docent of the Department of Computer Modeling of Processes and Systems, National Technical University «Kharkiv Polytechnic Institute», Kharkiv; tel.: (057) 707-64-54; e-mail: valerii.uspenskyi@khpi.edu.ua.