

УДК 004.93

DOI: 10.15587/1729-4061.2019.164789

## Розробка модифікованих методів навчання нейромереж для розв'язання задачі розпізнавання учасників дорожнього руху

Є. М. Федорченко, А. О. Олійник, О. О. Степаненко, Т. А. Зайко,  
С. І. Шило, А. О. Свириденко

Розроблено модифікації простого генетичного алгоритму для розпізнавання образів. У запропонованій модифікації Альфа-Бета на етапі відбору особин до нової популяції особини ранжуються за показником пристосованості, далі випадковим чином визначається кількість пар – певна кількість найпристосованіших особин, та стільки ж найменш пристосованих. Найпристосованіші особини формують підмножину  $B$ , найменш пристосовані – підмножину  $W$ . Обидві підмножини входять в множину пар  $V$ . Число особин, що можуть бути обрані в пари, знаходиться в діапазоні 20–60 % від загальної кількості особин. У модифікації Альфа-Бета фіксована в порівнянні з оригінальною версією простого генетичного алгоритму було додано можливість виникнення двох мутацій, додано фіксовану точку схрещення, а також змінено відбір особин для схрещення. Це дозволяє підвищити показник точності у порівнянні з базовою версією простого генетичного алгоритму. У модифікації Фіксована встановлено фіксовану точку схрещення. В схрещенні приймає участь половина генів – гени що відповідають за кількість нейронів на шарах, значення інших генів завжди передаються нащадкам від однієї з особин. Також, на етапі мутації випадковим чином відбуваються мутації з використанням методу Монте-Карло.

Розроблені методи програмно реалізовано для вирішення задачі розпізнавання учасників дорожнього руху (автомобілів, велосипедів, пішоходів, мотоциклів, вантажівок). Також було проведено порівняння показників використання модифікацій простого генетичного алгоритму та визначено кращий підхід вирішення задачі розпізнавання учасників дорожнього руху. Було встановлено, що розроблена модифікація Альфа-Бета показала кращі результати у порівнянні з іншими модифікаціями при вирішенні задачі розпізнавання учасників дорожнього руху. При застосуванні розроблених модифікацій отримано наступні показники точності: Альфа-Бета – 96.90 %, Альфа-Бета фіксована – 95.89 %, фіксована – 85.48 %. Крім того, при застосуванні розроблених модифікацій скорочується час підбору параметрів нейромоделі, зокрема при використанні модифікації Альфа-Бета витрачається лише 73,9 % часу базового методу, при використанні модифікації Фіксована – 91,1 % часу базового генетичного методу

Ключові слова: розпізнавання образів, генетичний алгоритм, еволюційний алгоритм, нейронні мережі, Python, OpenCV, Keras

## 1. Вступ

Кількість областей, в яких застосовується розпізнавання образів, постійно зростає, це пов'язано з розвитком технологій, методів, програмних рішень, бібліотек, обчислювальної техніки, а також з необхідністю автоматизації чи контролю процесів без участі людини.

Одною з таких областей є дорожня безпека, що включає багато вузьких напрямків: розпізнавання номерних знаків, визначення завантаженості вулиць, розпізнавання учасників руху в безпілотному транспорті та інше.

Однак на сьогодні повністю не вирішені проблеми якості розпізнавання. Наприклад, в системі Autopilot Tesla було виявлено вразливість – система у більшості випадків не може розпізнати велосипед, вона визначає велосипед як людину чи невеликий автомобіль, відповідно, система може прийняти рішення внаслідок яких можлива загроза життю велосипедиста [1].

Також відомі вразливості в розпізнаванні учасників дорожнього руху в таких системах як Mazda Smart City Brake Support [2], автопілот Waymo [3].

Можна сказати, що розпізнавання велосипедів та інших учасників дорожнього руху є достатньо актуальною задачею. Розпізнавання учасників дорожнього руху може бути використано в системі контролю та/або забезпечення безпеки дорожнього руху чи для дослідження актуальності створення спеціалізованих велосипедних доріжок, тощо.

## 2. Аналіз літературних даних та постановка проблеми

У роботі [4] наведено дослідження з вирішення задачі розпізнавання велосипедів на відеопотоці за допомогою об'єднання таких підходів: метод гістограм орієнтованих градієнтів (Histogram of Oriented Gradients – HOG) [5, 6], метод опорних векторів (Support Vector Machine – SVM) [7], каскадний класифікатор (метод Віоли-Джонса) [8, 9]. Наведено результати виконання розпізнавання велосипедів за різноманітних погодних умов та стверджується, що розроблений у роботі [4] підхід може бути використаний у системах розпізнавання реального часу.

Втім, використання поєднання методів HOG та SVM не є досить ефективним, оскільки це дозволяє розпізнавати достатньо великі об'єкти, які не завжди присутні на відео в режимі реального часу, та через те, що необхідно достатньо багато часу на обчислення складних параметрів. Використання каскадного класифікатора є більш ефективним з точки зору часу розпізнавання. Проте, даний метод має помилки в розпізнаванні – через застосування різних масштабів та розміру скануючого вікна один об'єкт може бути розпізнано як два.

Загалом, низька якість зображення, погодні умови та різноманітне освітлення суттєво ускладнюють використання методів [4–9]. Розроблений підхід [4] є досить комплексним та складним, кожен метод [5–9] потребує достатньо обчислювальних та часових ресурсів, що впливає на отримуваних результати та ефективність використання таких методів на практиці. Доцільнішим підходом до реалізації розпізнавання велосипедів було б використання нейронних мереж.

Так, у [10] наведено дослідження аналізу потоку машин на CCTV (системи відеоспостереження) зображеннях, що засновується на вирішенні задачі розпізнавання машин за допомогою нейронних мереж Faster R-CNN (faster region based convolutional neural network) [11] та SSD (single shot multibox detector) [12].

Загалом, SSD мережі здатні швидко здійснювати розпізнавання, але часто помиляються при розпізнаванні невеликих об'єктів, тоді як Faster R-CNN працюють повільніше, але здатні надавати більшу точність.

У роботі [12] виконано тестування вказаних навчених нейронних мереж на зображеннях низької якості та при різних погодних умовах. В результаті дослідження отримано, що розглянуті нейронні мережі не здатні розпізнати всі машини, при чому, при поганих погодних умовах (дощ, сніг, туман) показник точності значно падає. Тож, при навчанні нейронних мереж для розпізнавання машин слід було використати більше різноманіття навчальної вибірки. Крім того в дослідженні реалізується розпізнавання тільки машин, хоча поняття “трафік” охоплює декілька видів транспорту (мотоцикли, вантажівки, тощо).

У [13] розглянуто вирішення задачі розпізнавання машин за допомогою навчання згорткової нейронної мережі [14], що базується на підкріпленій помилці навчання та зразках схильних до помилок. Тобто, при навчанні згорткової нейронної мережі пропонується використовувати помилки навчання поточної стадії, в якості навчальних даних на наступній стадії. Було зроблено порівняння розпізнавання методом гістограм орієнтованих градієнтів зі згортковими нейронними мережами, одна з яких була навчена стандартним способом, а друга – запропонованим авторами, в результаті якого запропонований у [13] підхід показав кращі показники (зокрема, точність розпізнавання склала 83,91 %).

Однак не наведено прикладів виконання розпізнавання автомобілів, та через невеликий об'єм навчальної вибірки, що була використана у дослідженні, не можна з повною впевненістю говорити про те, що було досягнуто значного покращення навчання загорткової нейронної мережі.

У [15] розглядається вирішення задачі розпізнавання серед натовпу кожного пішохода окремо. Основна мета дослідження здійснення розпізнавання пішохода на чорно-білих зображеннях у різноманітних ситуаціях, детектування пішохода на зображенні, навіть при частковому перекритті, та з високою точністю локалізувати його. Також, за мету ставиться визначення скільки пішоходів присутньої на зображенні.

Підхід, розроблений у даній статті, заснований на масштабо-незалежному розширенні моделі неявної форми (Implicit Shape Model – ISM). Авторам вдалося реалізувати всі поставлені цілі, однак точність розпізнавання пішоходів дорівнює 71.3 % та не є задовільною. Також, варто відзначити, що розроблений метод працює лише із зображеннями. Проте, метод має великий потенціал та може в результаті подальшого розвитку розвиватися в напрямку розпізнавання образів та трекінгу.

У [16] розглядається вирішення задачі розпізнавання пішоходів за допомогою поєднання декількох нейронних мереж. Запропонована система

складається з двох підсистем: головна система детектування пішоходів для генерації всіх детектувань та система семантичної сегментації для покращення результатів. Система детектування пішоходів в свою чергу складається з генератору «кандидатів» пішоходів та системи класифікації. Для того щоб надавати більше інформації про координати об'єкта мережі класифікації, пропонується використання нового методу «м'яких позначень», який відносить об'єкт до всіх класів. Для реалізації системи класифікації було втілено ідею «збірного навчання». Також пропонується методика «м'якого відкидання» для поєднання висновків усіх мереж класифікації та мережі семантичної сегментації з висновком мережі генерації «кандидатів».

Розпізнавання пішоходів було протестовано на чотирьох популярних наборах зображень: набір Caltech Pedestrian, набір INRIA, набір ETH, набір KITTI. На перших трьох наборах було отримано найвищі показники точності розпізнавання. Також вказується на значну швидкість роботи розробленого підходу.

У [17] розглядається реалізація розпізнавання пішоходів, з використанням комбінації методу гістограм орієнтованих градієнтів і методу опорних векторів та згорткової нейронної мережі. Мета роботи полягає в тому, щоб сконструювати семантичні регіони, що представляють інтерес, для того щоб добути об'єкт переднього плану, для зменшення помилок пов'язаних з помилками розпізнавання фону. Спочатку за допомогою згорткової нейронної мережі, навченої за набором Caltech Pedestrian, генерується теплова мапа за вхідним зображенням. Далі, семантичні регіони інтересу добуваються з теплової мапи за допомогою морфологічної обробки зображення. В решті, семантичні регіони інтересу розділяють все зображення на фон та передній план, для полегшення роботи детекторів які прийматимуть рішення.

Відмічається, що за допомогою семантичних регіонів інтересу, робота детекторів в різній ступені поліпшується. Втім, використання поєднання методів HOG та SVM не є оптимальним, оскільки це дозволяє розпізнавати достатньо великі об'єкти, які не завжди присутні на відео в режимі реального часу та через те, що необхідно достатньо багато часу на обчислення складних параметрів.

В роботі [18] розглянуто реалізацію системи розпізнавання транспортних засобів, що відносяться до 7 класів (мотоцикл, авто, пікап, автобус, вантажівка, вантажівка з причепом, вантажівка з декількома причепами) на CCTV відео. У реалізації було використано навчену глибоку згорткову нейронну мережу (Deep Convolutional Neural Network – DCNN).

Запропонована система алгоритмічно складається з двох задач: локалізації та класифікації. Спочатку виконується локалізація, за допомогою генерації регіонів незалежних від класу, для кожного кадру, потім використовується глибока згорткова нейронна мережа для добутку описів ознак для кожного регіону. Врешті, за допомогою методу опорних векторів (Support Vector Machines – SVM), для кожного регіону, добути описи ознак порівнюються із шаблонами, та здійснюється оцінка відповідності і класифікація. Точність розпізнавання транспортних засобів відрізняється для кожного класу, але

загалом показник точності знаходиться в межах 92–95 %. Показник точності розпізнавання помітно зменшується при завантаженому трафіку, за дощу та туману, при низькій якості відео.

Недоліком даної розробки є використання глибокої згорткової нейронної мережі оскільки її обчислення вимагає багато обчислювальних ресурсів, що не є прийнятним. Використання даної нейронної мережі можливе при наявності потужного процесору (CPU), або графічного процесору (GPU) NVIDIA, в поєднанні з такими середовищами роботи зі штучним інтелектом як CUDA, Caffe, Torch. Також, в системі виявлені помилки розпізнавання, при невеликому масштабі, тоді об'єкт одного класу розпізнається як об'єкт іншого класу, наприклад “вантажівка” розпізнається як “вантажівка з причепом”, тощо.

У статті [19] описується метод “an end-to-end” розпізнавання рухомих об'єктів з відеопотоку в реальному часі, їх класифікації за заданим категоріям відповідно до властивостей на основі зображень і подальшого їх відстеження. Розпізнавання рухомих об'єктів відбувається з використанням піксельної різниці між послідовними кадрами зображення. До метриці класифікації застосовуються ці об'єкти з тимчасовим обмеженням узгодженості, щоб класифікувати їх за трьома категоріями: людина, транспортний засіб або фон. Після класифікації об'єкти відслідковуються шляхом об'єднання тимчасових відмінностей і зіставлення з шаблоном.

Двома ключовими елементами, які роблять цю систему стійкою, є система класифікації, заснована на тимчасовій узгодженості, і система відстеження, заснована на комбінації тимчасової різниці і зіставлення кореляцій. Система ефективно об'єднує прості знання предметної області про класи об'єктів зі статистичними показниками в тимчасовій області для класифікації цільових об'єктів.

Перевагою даного методу є те, що він дозволяє безперервне відстеження, незважаючи на оклюзію і припинення руху об'єкта, запобігає «дрейф» шаблонів на фонову текстуру і забезпечує надійне відстеження, не потребуючи спеціального фільтру, такого як фільтр Калмана.

Основним недоліком даного методу була неправильна класифікація, яка виникала у випадку, якщо кілька об'єктів знаходяться близько один до одного. А також дрібні об'єкти часто не розпізнаються, як стабільні в часі об'єкти.

В роботі [20] вирішується проблема виявлення та відстеження рухомих об'єктів у відеопотоці, отриманому з рухомої повітряної платформи. Досліджуваний метод спирається на графічне представлення рухомих об'єктів, що дозволяє виводити і підтримувати динамічний шаблон кожного переміщеного об'єкта шляхом забезпечення їх тимчасової когерентності. Динамічний шаблон разом з графічним поданням, що використовується в даному підході, дозволяє охарактеризувати траєкторії об'єктів як оптимальний шлях у графі. Запропонований трекер дозволяє вирішувати часткові оклюзії, зупиняти і рухатися в дуже складних ситуаціях. У роботі представлені результати на ряді різних реальних послідовностей. Метою відстеження та виявлення об'єктів є встановлення відповідності між об'єктами або частинами об'єктів у послідовних кадрах і вилучення тимчасової інформації про такі

об'єкти, як траєкторія, положення, швидкість і напрямок. Відстеження виявленого кадру об'єкта за кадром у відео є важливим і важким завданням.

Недоліком даної роботи являється те, що треба враховувати, якщо загальний відеосигнал містить дуже великий обсяг інформації, важливо отримувати і обробляти тільки невелику кількість релевантних даних, особливо в випадках, де повинна бути досягнута продуктивність з частотою кадрів в реальному часі.

У роботі [21] розглянуто використання Switchable Deep Network (SDN) для розпізнавання пішоходів. У даній роботі вирішуються актуальні проблеми розпізнавання пішоходів, а саме помилки у розпізнаванні фону, велика кількість варіацій вигляду пішоходів через зміну положення та інших факторів. Однією з поширених проблем систем розпізнавання пішоходів є те, що система може розпізнати деякий об'єкт як пішохід. Наприклад, форма і зовнішній вигляд «стовбура дерева» або «залізного стовпа» в певній точці зору схожі на пішоходів. Тому для вирішення цієї проблеми в даній роботі було запропоновано використати обмежену машину Больцмана (Restricted Boltzmann Machine – RBM) для розпізнавання пішоходів. Використання SDN поліпшує стандартну згорткову нейронну мережу шляхом додавання декількох обмежувальних слоїв, які створюються з новою обмеженою машиною Больцмана, що дозволяє в результаті автоматично вивчати як низькорівневі особливості, так і високорівневі частини (наприклад "голова", "ноги", і т.п.) пішохода.

Однак вираш в продуктивності завдяки використанню Switchable Deep Network був незначним. Розроблений метод потребував достатньо багато часу для розпізнавання складних об'єктів і питання про те, як розпізнати пішохода для досягнення оптимальної роботи, все ще залишається відкритим питанням.

Завдання розпізнавання пішоходів має деякі специфічні особливості. Однією з таких особливостей є необхідність коректного виявлення пішоходів різного розміру. Різний розмір пішоходів може бути обумовлений як різної віддаленістю від детектора, так і різним ростом і статурою самих людей (наприклад, дитина і баскетболіст). Можливість розпізнавання багаторозмірних об'єктів значно покращить детектор [22]. Наприклад, в разі використання детектора в автомобілі, коректне виявлення пішоходів не тільки поблизу від нього, а й на більшій відстані дозволить краще контролювати ситуацію на дорозі і приймати своєчасні рішення.

Найчастіше детектор [22] обчислює скалярний добуток між навченим шаблоном (чутливим полем) і регіоном зображення, який розпізнається. Для коректної роботи детектора розмірність навченого шаблону повинна збігатися з розмірністю об'єкта для розпізнавання. Існує два основні методи розв'язання задачі розпізнавання багатомасштабних об'єктів.

Перший описаний у роботі [23] який передбачає навчання одного класифікатора і зміна розміру вхідних зображень, так, щоб класифікатор підходив для всіх можливих розмірів об'єктів. Даний метод найбільш точний, але вимагає великої кількості обчислень, оскільки вимагає обчислення ознак при кожній зміні розміру вхідного зображення.

Недоліком першого методу є те, що розмір регіону кандидата буде змінюватися до розміру, з яким працює мережа і відповідно ознаки будуть обчислюватися не для всього зображення, а тільки для деякого регіону.

Другий метод [24] полягає в навчанні декількох класифікаторів, які будуть підходити для розпізнавання всіх розмірів об'єктів при незмінному розмірі вхідного зображення. Цей метод дозволяє уникнути повторних обчислень ознак, але якість розпізнавання досить низька, так як для кожного з можливих розмірів об'єкта потрібно свій класифікатор.

Недоліком другого методу є те, що відбувається сильна розбіжність розмірів розпізнаного об'єкта і зразка.

Аналіз літературних джерел [4, 10, 13, 15–18] дозволяє стверджувати, що проведення досліджень щодо підвищення точності розпізнавання є доволі актуальною задачею, а також показав, що існують ще невирішені проблеми, зокрема:

- людська оклюзія. Оскільки люди з'являються в різних і непередбачуваних фонах, оклюзія може статися в будь-який час. Таким чином, для досягнення високої продуктивності при виявленні людини в процесі розпізнавання, проблема оклюзії повинна бути ефективно оброблена;

- людська артикуляція. Зовнішність людини може бути надзвичайно різноманітною: зміною положення, відстані або точкою зору камери. Таким чином, алгоритми розпізнавання в процесі виявлення людини повинні враховувати цей аспект, щоб зробити систему більш надійною і точною;

- фоновий шум. Зміни в контексті через погодні умови, зміни освітленості та складні фони є найважливішими причинами для виявлення несправності або виявлення пропуску, що також в деяких випадках суттєво знижує точність розпізнавання;

- час обробки. Час обробки є жорсткою вимогою для багатьох реальних додатків для роботи в реальному часі. Сучасні підходи до розпізнавання людини, машини все ще потребують чіткого вдосконалення, щоб задовольнити цю вимогу та скоротити час розпізнавання.

Такі проблеми в деяких випадках суттєво знижують точність розпізнавання, що ускладнює застосування відомих методів на практиці.

Наявність зазначених недоліків обумовлює необхідність розробки еволюційних методів побудови розпізнавальних моделей, що дозволяють виконувати розпізнавання автотранспортних засобів з більш високою точністю розпізнавання за менший час.

### **3. Мета і задачі дослідження**

Мета дослідження – розробка математичного забезпечення для розв'язання задачі розпізнавання учасників дорожнього руху (автомобілів, велосипедів, пішоходів, мотоциклів, вантажівок) на основі модифікацій простого генетичного методу навчання нейромереж. Це надасть можливість розпізнавання зображень низької якості та при різних погодних умов.

Для досягнення поставленої мети вирішувалися наступні задачі:

- розробка модифікацій простого генетичного методу навчання нейромереж;
- здійснити програмну реалізацію розроблених модифікацій простого генетичного методу;
- провести дослідження ефективності розроблених модифікацій простого генетичного методу.

#### **4. Розробка модифікацій простого генетичного методу навчання нейромереж. Визначення впливу параметрів навчання нейронних мереж**

У розроблених модифікаціях простого генетичного методу для підвищення ефективності еволюційного пошуку запропоновано нові евристичні процедури, зокрема додано можливість виникнення двох мутацій, зокрема, з використанням методу Монте-Карло, модифіковано оператори відбору та схрещування. Це дозволяє підвищити показник точності у порівнянні з базовою версією простого генетичного алгоритму. Запропоновані модифікації простого генетичного алгоритму полягають в наступному.

У запропонованій модифікації Альфа-Бета для схрещення в кожній генерації обирається різна кількість пар для схрещення, при чому в парі одна особина відноситься до найприспособаніших, а друга до найменш приспособованих особин. Також випадковим чином може виникнути дві мутації (базова та подвоююча) або одна мутація (базова): за методом Монте-Карло – генерується випадкове число, 0 або 1. Якщо випадає 0, то виникає одна мутація, якщо випадає 1 – виникає дві мутації.

Послідовність дій даної модифікації простого генетичного алгоритму схожа на базову його версію, проте має певні відмінності. На етапі відбору особин до нової популяції  $P_n$  особини ранжуються за показником приспособованості, далі випадковим чином визначається кількість пар – певна кількість найприспособаніших особин, та стільки ж найменш приспособованих. Найприспособаніші особини формують підмножину  $V$ , найменш приспособовані – підмножину  $W$ . Обидві підмножини входять в множину пар  $V$ . Число особин, що можуть бути обрані в пари, знаходиться в діапазоні 20 – 60 % від загальної кількості особин. Такі характеристики діапазону особин обрано в результаті чисельних експериментів та досліджень розробленої модифікації генетичного методу. Решта нової популяції  $P_n$  отримується за рахунок схрещення обраних особин ( $K$ ).

$$P_n = (V, K_j), \quad (1)$$

де  $j=1, \dots, (N-V)$ .

Також, в запропонованій модифікації випадковим чином може виникнути дві ( $\mu_1, \mu_2$ ) або одна мутація ( $\mu$ ), тоді як в базовій версії алгоритму, випадковим чином виникає одна мутація. Причому в ситуації, коли виникає дві мутації, – один ген може мутувати двічі.



Після застосування оператора мутації нащадки включаються до нового покоління  $P_n$ . Схрещення та мутації проводяться до тих пір, поки не буде створено нове покоління  $P_n$  розміру  $N$  (1).

У другій запропонованій модифікації Альфа-Бета фіксована для схрещення в кожній генерації обирається різна кількість пар для схрещення, при чому в парі одна особина відноситься до найприспособаніших, а друга до найменш пристосованих особин. Також, випадковим чином може виникнути дві мутації (базова та подвоююча) або одна мутація (базова): за методом Монте-Карло – генерується випадкове число, 0 або 1. Якщо випадає 0, то виникає одна мутація, якщо випадає 1 – виникає дві мутації. Та встановлено фіксовану точку схрещення – в схрещенні приймає участь перша половина генів – гени що відповідають за кількості нейронів на шарах, значення інших генів завжди передаються нащадкам від однієї з особин.

Послідовність дій даної модифікації схожа на послідовність дій модифікації Альфа-Бета, проте, операція схрещення відмінна. З підмножин  $B$  та  $W$  випадково вибирається по особині  $M$  та  $F$ , які мають певний набір генів:

$$M = \left( (n_1^1, \dots, n_m^1), m^1, f_{act}^1, f_{opt}^1 \right),$$

$$F = \left( (n_1^2, \dots, n_m^2), m^2, f_{act}^2, f_{opt}^2 \right).$$

Таким чином, оператор схрещення можна виразити таким чином:

$$C^* = \left( \left( \frac{1}{2}M \times \frac{1}{2}F \right), \frac{1}{2}F \right). \quad (2)$$

Результатом схрещення особин є два нащадки  $K_1, K_2$ , які можна записати так:

$$K_1 = \left( R \left[ (n_1^1, \dots, n_m^1), (n_1^2, \dots, n_m^2) \right], m^2, f_{act}^2, f_{opt}^2 \right), \quad (3)$$

$$K_2 = \left( R \left[ (n_1^1, \dots, n_m^1), (n_1^2, \dots, n_m^2) \right], m^2, f_{act}^2, f_{opt}^2 \right),$$

де  $R$  – функція вибору випадкової величини;  $m^2, f_{act}^2, f_{opt}^2$  – значення генів що передались від особини  $F$ .

Розроблена модифікація Фіксована полягає у тому, що встановлено фіксовану точку схрещення: в схрещенні приймає участь половина генів – гени, що відповідають за кількості нейронів на шарах, значення інших генів завжди передаються нащадкам від однієї з особин. Також, на етапі мутації випадковим чином виникає дві мутації (базова та подвоююча) або одна мутація (базова): за

методом Монте-Карло – генерується випадкове число, 0 або 1. Якщо випадає 0, то виникає одна мутація, якщо випадає 1 – виникає дві мутації.

Послідовність дій даної модифікації схожа на послідовність дій модифікації Альфа-Бета, проте має певну відмінність – відбір особин до нової популяції відбувається як у базовій версії генетичного алгоритму  $P_n = T + L + K$ , та схрещення відбувається як у другій модифікації (2), (3).

Таким чином, створено модифіковані генетичні методи для підбору параметрів навчання нейронних мереж – Альфа-Бета, Альфа-Бета фіксована, Фіксована.

## 5. Програмна реалізація розроблених модифікацій простого генетичного методу

Програмна реалізація розроблених модифікацій простого генетичного методу включає в себе різні програми: розпізнавання образів з використанням генетичних методів, каскадного класифікатора, навчання нейронної мережі, тестування розпізнавання нейронної мережі, підбору параметрів при навчанні нейронних мереж з використанням модифікацій простого генетичного алгоритму (рис. 1).

Програма реалізація розпізнавання образів складається з кількох модулів, пов'язаних методами, тому структурну схему програми можна зобразити наступним чином (рис. 2).

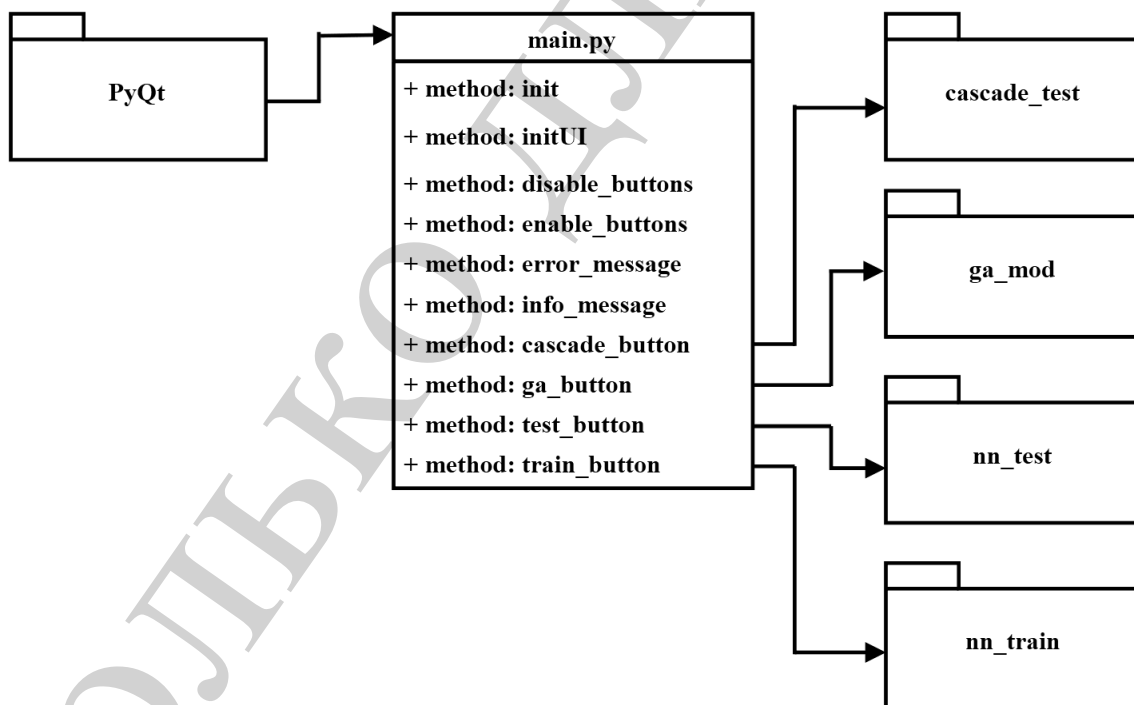


Рис. 1. Схема розробленого програмного комплексу

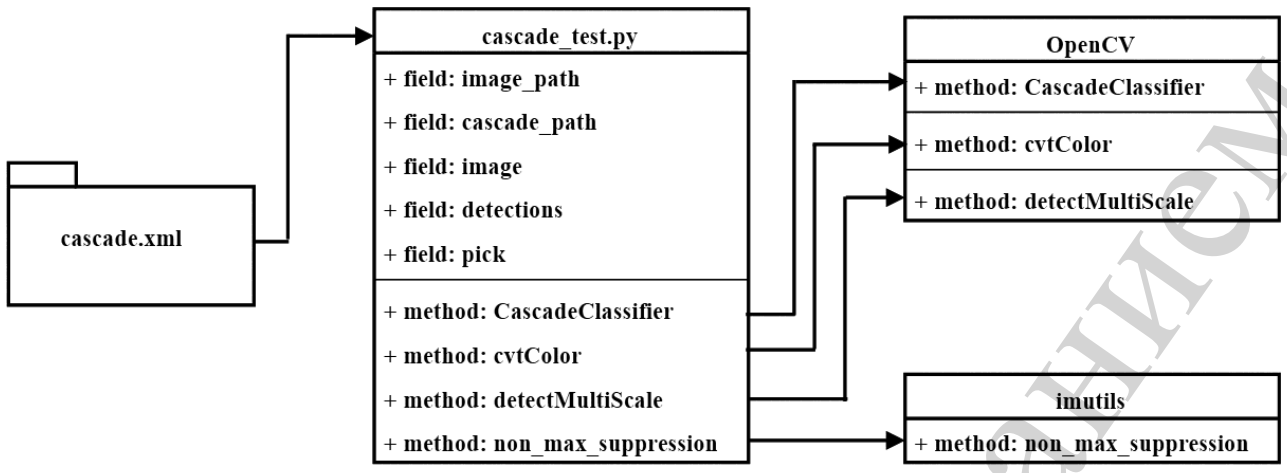


Рис. 2. Схема програми тестування каскадного класифікатора

Програмну реалізацію підбору параметрів навчання нейронних мереж за допомогою модифікованих генетичних методів можна зобразити наступним чином (рис. 3).

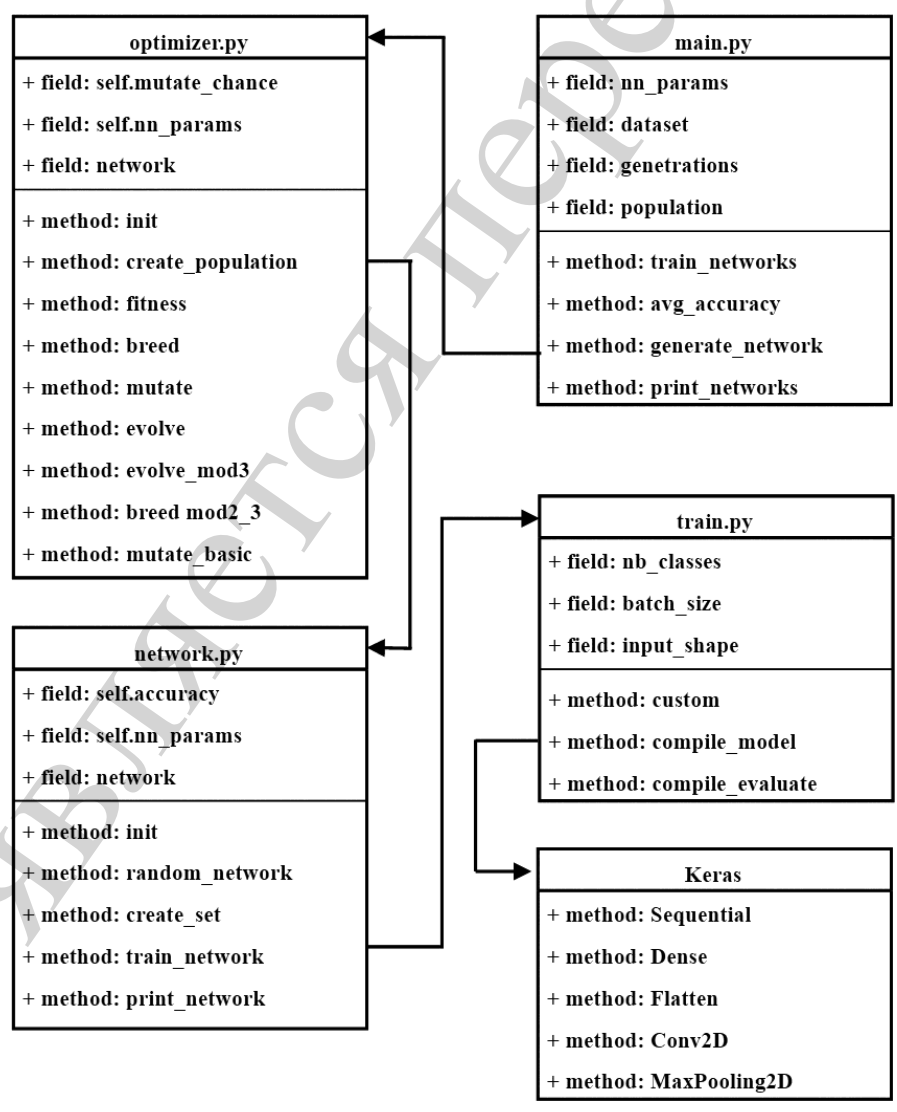


Рис. 3. Схема програми підбору параметрів для навчання нейронних мереж

Програмну реалізацію навчання згорткової нейронної мережі можна зобразити наступним чином (рис. 4).

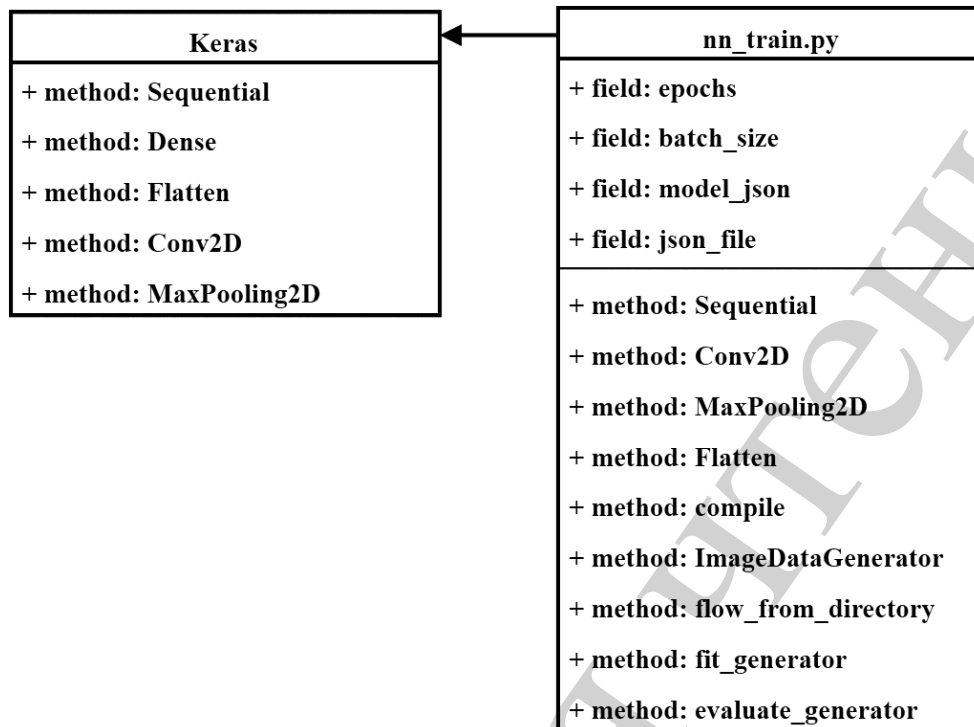


Рис. 4. Схема програми навчання нейронної мережі

Розроблена програмна реалізація являє собою набір програм для розпізнавання образів, який містить файли (скрипти, модулі) для розпізнавання образів.

Головний модуль програми – main.py, з якого виконується виклик програм перелічених вище. Даний модуль містить такі методи:

- initUI – метод створення вікна програми;
- disable\_buttons – метод в якому обмежується можливість виклику програм;
- enable\_buttons – метод в якому надається можливість виклику програм;
- error\_message – метод виведення вікна помилки;
- info\_message – метод виведення вікна повідомлення;
- cascade\_button – метод виклику програми тестування детектування каскадного класифікатора;
- ga\_button – метод виклику програми підбору параметрів при навчання нейронних мереж;
- test\_button – метод виклику програми тестування розпізнавання нейронною мережею;
- train\_button – метод виклику програми навчання нейронної мережі.

Програма розпізнавання образів з використанням каскадного класифікатора складається з двох файлів: cascade\_test.py та cascade.xml.

Файл cascade\_test.py – головний файл програми що використовує каскадний класифікатор для розпізнавання, в якому відбувається зчитування зобра-

ження та файлу каскадного класифікатора, відбувається обробка зображення та розпізнавання, і вивід результуючого зображення. Скрипт використовує такі методи:

- CascadeClassifier – метод зчитування каскадного класифікатора з бібліотеки OpenCV;
- imread – метод зчитування зображення з бібліотеки OpenCV;
- datetime.now – метод визначення поточного часу. Використовується для обчислення часу витраченого на розпізнавання;
- cvtColor – метод переведення зображення в інший колірний простір з бібліотеки OpenCV;
- detectMultiScale – метод розпізнавання об'єкту з бібліотеки OpenCV;
- non\_max\_suppression – метод що дозволяє зменшити похибку розпізнавання з бібліотеки imutils.

Файл cascade.xml – навчений каскадний класифікатор для розпізнавання об'єкту.

Програма підбору параметрів при навчанні нейронної мережі складається з таких файлів: main.py, network.py, optimizer.py, train.py, які містяться в теці ga\_mod.

Файл main.py – головний файл програми, де відбувається ініціалізація значень параметрів що будуть використовуватися в процесі навчання нейронних мереж, виконується виклик навчання нейронних мереж. Скрипт має такі методи:

- train\_networks – викликає метод train скрипту network.py та відображає прогрес навчання нейронних мереж;
- avg\_accuracy – підраховується середній показник точності навчених мереж;
- generate\_network – викликає методи навчання нейронних мереж та методи підбору параметрів зі скрипту optimizer.py;
- print\_networks – виконується вивід інформації про нейронні мережі з кращими показниками;
- main – відбувається ініціалізація значень параметрів що будуть використовуватися в процесі навчання нейронних мереж, та виклик методу generate.

Файл network.py призначений для зберігання інформації про нейронну мережу. Містить такі методи:

- init – виконується ініціалізація мережі;
- random\_network – виконується заповнення генів мережі випадковими значеннями;
- create\_set – встановлює відповідній мережі значення згенерованих генів;
- train\_network – викликає метод навчання нейронних мереж train\_and\_score зі скрипту train.py;
- print\_network – виконує вивід інформації про показники навченої нейронної мережі.

Файл `optimizer.py` містить реалізацію модифікованого простого генетичного алгоритму для підбору оптимальних параметрів навчання нейронної мережі. Містить такі методи:

- `init` – виконується ініціалізація параметрів для підбору параметрів;
- `create_population` – створення популяції нейронних мереж;
- `breed` – виконується схрещення двох нейронних мереж (модифікація Альфа-Бета);
- `breed_mod2_3` – виконується схрещення двох нейронних мереж (модифікації Альфа-Бета фіксована та Фіксована);
- `mutate` – виконується оператор мутації до генів нейронної мережі (дві мутації – базова та подвоююча);
- `mutate_basic` – виконується оператор мутації до генів нейронної мережі (одна мутація – базова);
- `evolve` – виконується відбір нейронних мереж в наступне покоління (модифікації Альфа-Бета та Альфа-Бета фіксована);
- `evolve_mod3` – виконується відбір нейронних мереж в наступне покоління (модифікація Фіксована).

Файл `train.py` призначений для завантаження навчальної вибірки та безпосередньо навчання нейронної мережі. Містить наступні методи:

- `custom` – виконується завантаження навчальної вибірки;
- `compile_model` – виконується створення структури нейронної мережі;
- `compile_evaluate` – виконується навчання нейронної мережі.

Програма навчання нейронної мережі складається з файлу `nn_train.py`. В даному скрипті відбувається завантаження навчальної вибірки та навчання нейронної мережі та її збереження до пам'яті комп'ютера. Архітектура навченої нейронної мережі зберігається до файлу `model.json`, а ваги мережі записуються до файлу `model.h5`. Скрипт використовує такі методи:

- `compile` – відбувається збірка нейронної мережі;
- `ImageDataGenerator` – відбувається генерація навчального набору з навчальної вибірки;
- `flow_from_directory` – відбувається перетворення зображень для подальшого використання;
- `fit_generator` – відбувається робота із зображеннями в процесі навчання;
- `evaluate_generator` – відбувається робота із зображеннями в процесі авто-тестування розпізнавання;
- `save_weights` – запис ваг нейронної мережі до файлу `model.h5`.

Програма розпізнавання образів з використанням нейронної мережі складається з файлу `nn_test.py`, який використовує файли архітектури мережі `model.json` та файл її ваг `model.h5`. В даному скрипті виконується зчитування вхідного зображення, переведення його у відповідний формат, та подання нейронній мережі для розпізнавання. Скрипт використовує такі методи:

- `model_from_json` – відбувається завантаження архітектури нейронної мережі з файлу `model.json`;
- `load_weights` – відбувається завантаження ваг нейронної мережі з файлу `model.h5`;

- compile – відбувається збірка нейронної мережі;
- load\_img – завантаження зображення для розпізнавання;
- predict – виконується розпізнавання по вхідному зображенню.

В процесі навчання нейронних мереж було задіяно декілька вибірки, які використовувалися для навчання та тестування розпізнавальних моделей за допомогою розроблених модифікацій простого генетичного методу. Кожна вибірка містить 5 класів: пішохід, велосипед, мотоцикл, авто, вантажівка. Всі вибірки представляють собою набори зображень, що містять відповідний об'єкт. Кожна з навчальних вибірок складається з 1000000 зображень, кожна перевірна зі 400000 зображень.

## 6. Експерименти та результати дослідження використання модифікацій простого генетичного алгоритму

Дослідження впливу значень параметрів при навчанні нейронних мереж на показник точності та час виконання навчання проводилося на комп'ютері з процесором Intel Core i5 7400 з тактовою частотою 3 ГГц, з об'ємом оперативної пам'яті 8 ГБ.

Для навчання нейронних мереж використовувалась навчальна вибірка, яка складалася із 1000000 зображень, розділених на 5 класів та.

Було проведено 4 експерименти, в яких було використано розроблені модифікації генетичного алгоритму та безпосередньо простий генетичний алгоритм. Всі експерименти були проведені з однаковими параметрами: кількість поколінь – 5, розмір популяції – 20, коефіцієнт мутації – 0.01. Вхідні дані – вибірка зображень 5 класів (пішохід, велосипед, мотоцикл, авто, вантажівка). Результати експериментів з порівняння простого генетичного методу та розроблених модифікацій наведено у табл. 1.

Таблиця 1

Результати підборів параметрів за допомогою простого генетичного алгоритму

Версія ГА	Час навчання		Точність на тестових даних	Точність авто-тестування
	Фактичний	У відсотках до базової версії ГА		
Базова	8 днів, 8 хвилин, 39 секунд	100 %	89.45 %	93.89 %
Модифікація Альфа-Бета	5 днів, 22 години, 27 хвилин, 5 секунд	73,9 %	92.12 %	96.90 %
Модифікація Альфа-Бета фіксована	8 днів, 12 годин, 4 хвилини, 58 секунд	106,2 %	90.02 %	95.89 %
Модифікація Фіксована	7 днів, 7 годин, 16 хвилин, 43 секунди	91,1 %	92.48 %	95.72 %

Виходячи результатів, наведених в табл. 1, можна зробити висновок, що використання модифікації Альфа-Бета генетичного алгоритму є кращим підходом для досягнення вищої точності розпізнавання за менший час.

У табл. 2 наведено результати розпізнавання образів при використанні різних підходів для навчання розпізнавальних моделей.

Таблиця 2

Порівняння результатів розпізнавання образів при використанні різних підходів для навчання розпізнавальних моделей

Підхід \ Характеристика	Час навчання	Точність на тестових даних	Точність авто-тестування
Каскадний класифікатор	8 днів, 15 годин, 36 хвилин	90.95 %	–
Повнозв'язна нейронна мережа (100 класів, без використання ГА)	1 година, 4 хвилини, 19 секунд	1.2 %	24.93 %
Згорткова нейронна мережа (100 класів, без використання ГА)	1 година, 59 хвилин, 34 секунди	1.8 %	47.14 %
Повнозв'язна нейронна мережа (100 класів, з використанням ГА)	22 години, 47 хвилин, 33 секунди	1.4 %	30.52 %
Згорткова нейронна мережа (100 класів, з використанням ГА, 4 шари)	4 дні, 16 годин, 59 хвилини, 21 секунда	2.1 %	48 %
Згорткова нейронна мережа (100 класів, з використанням ГА, 6 шарів)	6 днів, 12 годин, 51 хвилина, 42 секунди	2.39 %	53.69 %
Згорткова нейронна мережа (5 класів, з використанням ГА, 3 шари)	8 днів, 8 хвилин, 39 секунд	89.45 %	93.89 %
Згорткова нейронна мережа (5 класів, з використанням модифікації ГА Альфа-Бета, 3 шари)	5 днів, 22 години, 27 хвилин, 5 секунд	92.12 %	96.90 %
Згорткова нейронна мережа (5 класів, з використанням модифікації ГА Альфа-Бета фіксована, 3 шари)	8 днів, 12 годин, 4 хвилини, 58 секунд	90.02 %	95.89 %
Згорткова нейронна мережа (5 класів, з використанням модифікації ГА Фіксована, 3 шари)	7 днів, 7 годин, 16 хвилин, 43 секунди	92.48 %	95.72 %



Виходячи з таблиці 5, можна зробити висновок, що зменшення кількості класів для навчання зі 100 до 5 значно підвищило показник точності, проте час виконання підбору дещо збільшився. Також, порівнюючи результати отримані при виконанні підбору параметрів з використанням простого генетичного алгоритму та його модифікацій, можна зробити висновок, що кращим підходом до оптимізації процесу підбору параметрів навчання нейронної мережі є використання модифікації генетичного алгоритму Альфа-Бета.

## **6. Обговорення результатів дослідження використання модифікацій простого генетичного алгоритму**

В результаті дослідження була розроблена модифікація простого генетичного алгоритму – Альфа-Бета. Використання даної модифікації дозволяє пришвидшити виконання підбору параметрів навчання нейронних мереж, та підвищити показник точності. Доведено, що підвищення кількості мутацій та підбір в пари різних особин надає більшу різноманітність комбінацій генів, що призводить до кращих показників за менший час.

Як видно з таблиць 1 та 2, розроблені модифікації простого генетичного методу дозволяють підвищити швидкість синтезу розпізнавальних моделей відносно базової версії ГА у (зокрема, при використанні модифікації Альфа-Бета витрачається лише 73,9 % часу базового методу, при використанні модифікації Фіксована – 91,1 % часу базового генетичного методу). Це дозволило зменшити час оброблення експериментального масиву даних за допомогою модифікованого алгоритму Альфа-Бета на 58 годин у порівнянні з базовим методом. Як видно з таблиці 5, найкращим методом за швидкістю рішення обраної задачі є повнозв'язна нейронна мережа, яка має наступні характеристики: 1 година, 4 хвилини, 19 секунд, проте точність розпізнання у неї складає лише 24.93%. Повнозв'язна нейронна мережа з використанням ГА має наступні характеристики: час навчання 22 години, 47 хвилин, 33 секунди, проте точність розпізнавання складає лише 30.52 %. Розроблені модифікації простого генетичного методу дозволили отримати точність розпізнавання на тестових даних 95–96 %, що є досить прийнятним результатом.

Такі результати обумовлюються використанням розроблених евристичних процедур, зокрема мутацій з використанням методу Монте-Карло, модифікованих операторів відбору та схрещування. Це дозволило підвищити показник точності та зменшити час оптимізації за допомогою розроблених модифікацій простого генетичного методу у порівнянні з його базовою версією.

Виходячи з результатів розпізнавання можна зробити висновок, що навчена нейронна мережа здатна з досить високою точністю визначати належність об'єкта до певного класу.

Таким чином, розроблені модифікації простого генетичного алгоритму дозволяють підвищити точність розпізнавання та зменшити час навчання. Це досягається за рахунок використання у розроблених модифікованих методах нових евристичних процедур, зокрема додано можливість мутацій з використанням методу Монте-Карло, модифіковано оператори відбору та

схрещування. Це дозволило підвищити показник точності у порівнянні з базовою версією простого генетичного алгоритму.

Недоліком модифікацій простого генетичного алгоритму, розроблених та досліджених у цій роботі, є необхідність витрачання великого часу (декілька діб) при обробці великих масивів даних, що при розв'язанні деяких практичних завдань є неприпустимим. Таким чином обмеженнями на використання розроблених модифікацій є невеликі обсяги оброблюваних даних.

Розвиток даного дослідження може бути пов'язаний з усуненням зазначених недоліків, обумовлених практичним порогом використання розроблених модифікацій. Для цього доцільно розробити їх паралельну реалізацію, що дасть можливість суттєво (в рази) збільшити швидкість роботи методів. Супутні проблеми, які можуть виникнути при розробленні паралельних модифікацій простого генетичного методу, пов'язані з необхідністю планування ресурсів паралельної комп'ютерної системи та зі збільшенням вимог до апаратного забезпечення, задіяного в процесі генетичної оптимізації.

## **7. Висновки**

1. Розроблено три модифікації простого генетичного алгоритму для підбору параметрів навчання нейронних мереж – Альфа-Бета, Альфа-Бета фіксована, Фіксована. У розроблених модифікаціях, на відміну від простого генетичного алгоритму, використовуються запропоновані евристичні процедури, зокрема мутація з використанням методу Монте-Карло, модифіковані оператори відбору та схрещування. Це дозволило підвищити показник точності та зменшити час оптимізації за допомогою розроблених модифікацій простого генетичного методу у порівнянні з його базовою версією.

2. Здійснено програмну реалізацію, що використовує розроблені модифікації простого генетичного алгоритму для підбору параметрів навчання нейронних мереж та розпізнавання учасників дорожнього руху. При застосуванні розроблених модифікацій отримано наступні показники:

– Альфа-Бета – точність 96.90 %, час підбору – 5 днів, 22 години, 27 хвилин, 5 секунд;

– Альфа-Бета фіксована – точність 95.89 %, час підбору – 8 днів, 12 годин, 4 хвилини, 58 секунд;

– Фіксована – точність 85.48 %, час підбору – 7 днів, 7 годин, 16 хвилин, 43 секунди.

3. Проведено порівняння показників модифікацій простого генетичного алгоритму, та визначено що модифікація Альфа-Бета є кращим підходом вирішення задачі розпізнавання учасників дорожнього руху. Оскільки дана модифікація дозволила отримати кращий показник точності за менший час підбору.

## **Подяки**

Роботу виконано в рамках науково-дослідної теми " Методи і засоби прийняття рішень для оброблення даних в інтелектуальних системах розпізнавання

образів " (№ державної реєстрації 0117U003920) кафедри програмних засобів Запорізького національного технічного університету.

### Література

1. Доросинский Л. Г. Распознавание изображений, нейронные сети и генетические алгоритмы // Успехи современного естествознания. 2011. № 10. С. 87–88.
2. Эксперт робототехники: «Никогда не используйте автопилот Tesla рядом с велосипедистами!». URL: <https://itc.ua/news/ekspert-robototekniki-nikогда-ne-ispolzuyte-avtopilot-tesla-ryadom-s-velosipedistami/>
3. Mazda I-Activsense. URL: <http://mazda.ua/ru/showroom/cx-5/i-activsense/>
4. Waymo. URL: <https://waymo.com/>
5. Zhang Y., Ling Q. Bicycle Detection Based On Multi-feature and Multi-frame Fusion in Low-Resolution Traffic Videos // arXiv. 2017. URL: <https://arxiv.org/pdf/1706.03309.pdf>
6. Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection // 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). 2005. doi: <https://doi.org/10.1109/cvpr.2005.177>
7. Encyclopedia of artificial intelligence / J. R. R. Dopico, J. Dorado, A. Pazos (Eds.). IGI Global, 2009. doi: <https://doi.org/10.4018/978-1-59904-849-9>
8. Support vector clustering / Ben-Hur A., Horn D., Siegelmann H. T., Vapnik V. // Journal of Machine Learning Research. 2001. Vol. 2. P. 125–137.
9. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features // Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. 2001. doi: <https://doi.org/10.1109/cvpr.2001.990517>
10. Viola P., Jones M. J. Robust Real-Time Face Detection // International Journal of Computer Vision. 2004. Vol. 57, Issue 2. P. 137–154. doi: <https://doi.org/10.1023/b:visi.0000013087.49260.fb>
11. Urban traffic flow analysis based on deep learning car detection from CCTV image series / Peppas M. V., Bell D., Komar T., Xiao W. // ISPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2018. Vol. XLII-4. P. 499–506. doi: <https://doi.org/10.5194/isprs-archives-xxii-4-499-2018>
12. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / Ren S., He K., Girshick R., Sun J. // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. Vol. 39, Issue 6. P. 1137–1149. doi: <https://doi.org/10.1109/tpami.2016.2577031>
13. SSD: Single Shot MultiBox Detector / Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A. C. // Computer Vision – ECCV 2016. 2016. P. 21–37. doi: [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
14. Xuze Z., Shengsuo N., Teng H. A CNN Vehicle Recognition Algorithm based on Reinforcement Learning Error and Error-prone Samples // IOP Conference Series: Earth and Environmental Science. 2018. Vol. 153. P. 032052. doi: <https://doi.org/10.1088/1755-1315/153/3/032052>

15. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks // In NIPS. 2012. P. 1097–1105.
16. Leibe B., Seemann E., Schiele B. Pedestrian Detection in Crowded Scenes // 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). 2005. doi: <https://doi.org/10.1109/cvpr.2005.272>
17. Fused Deep Neural Networks for Efficient Pedestrian Detection / Du X., El-Khamy M., Morariu V. I., Lee J., Davis L. // arXiv. 2018. URL: <https://arxiv.org/pdf/1805.08688.pdf>
18. Pedestrian Detection with Semantic Regions of Interest / He M., Luo H., Chang Z., Hui B. // Sensors. 2017. Vol. 17, Issue 11. P. 2699. doi: <https://doi.org/10.3390/s17112699>
19. Adu-Gyamf Y. O. Automated Vehicle Recognition with Deep Convolutional Neural Networks // Iowa State University Digital Repository. 2017. 12 p. URL: [https://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=1182&context=ccee\\_pubs](https://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=1182&context=ccee_pubs)
20. Lipton A. J., Fujiyoshi H., Patil R. S. Moving target classification and tracking from real-time video // Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV98 (Cat. No.98EX201). 1998. doi: <https://doi.org/10.1109/acv.1998.732851>
21. Cohen I., Medioni G. Detecting and tracking moving objects for video surveillance // Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. PR00149). 1999. doi: <https://doi.org/10.1109/cvpr.1999.784651>
22. Switchable Deep Network for Pedestrian Detection / Luo P., Tian Y., Wang X., Tang X. // 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014. doi: <https://doi.org/10.1109/cvpr.2014.120>
23. Tombari F., Salti S., Di Stefano L. Performance Evaluation of 3D Keypoint Detectors // International Journal of Computer Vision. 2013. Vol. 102, Issue 1-3. P. 198–220. doi: <https://doi.org/10.1007/s11263-012-0545-4>
24. Human Tracking Using Convolutional Neural Networks / Fan J., Xu W., Wu Y., Gong Y. // IEEE Transactions on Neural Networks. 2010. Vol. 21, Issue 10. P. 1610–1623. doi: <https://doi.org/10.1109/tnn.2010.2066286>
25. What is the best multi-stage architecture for object recognition? / Jarrett K., Kavukcuoglu K., Ranzato M. A., LeCun Y. // 2009 IEEE 12th International Conference on Computer Vision. 2009. doi: <https://doi.org/10.1109/iccv.2009.5459469>
26. Development of the indicator set of the features informativeness estimation for recognition and diagnostic model synthesis / Oliinyk A., Subbotin S., Lovkin V., Leoshchenko S., Zaiko T. // 2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET). 2018. doi: <https://doi.org/10.1109/tcset.2018.8336342>
27. Oliinyk A. A., Subbotin S. A. A stochastic approach for association rule extraction // Pattern Recognition and Image Analysis. 2016. Vol. 26, Issue 2. P. 419–426. doi: <https://doi.org/10.1134/s1054661816020139>

28. Oliinyk A. O., Zayko T. A., Subbotin S. O. Synthesis of Neuro-Fuzzy Networks on the Basis of Association Rules // *Cybernetics and Systems Analysis*. 2014. Vol. 50, Issue 3. P. 348–357. doi: <https://doi.org/10.1007/s10559-014-9623-7>

29. Development of the method for decomposition of superpositions of unknown pulsed signals using the second-order adaptive spectral analysis / Stepanenko A., Oliinyk A., Deineha L., Zaiko T. // *Eastern-European Journal of Enterprise Technologies*. 2018. Vol. 2, Issue 9 (92). P. 48–54. doi: <https://doi.org/10.15587/1729-4061.2018.126578>

30. Stratified model of the internet of things infrastructure / Alsayaydeh J. A. J., Shkarupylo V., Bin Hamid M. S., Skrupsky S., Oliinyk A. // *Journal of Engineering and Applied Sciences*. 2018. Vol. 13, Issue 20. P. 8634–8638.

31. Development of stratified approach to software defined networks simulation / Shkarupylo V., Skrupsky S., Oliinyk A., Kolpakova T. // *Eastern-European Journal of Enterprise Technologies*. 2017. Vol. 5, Issue 9 (89). P. 67–73. doi: <https://doi.org/10.15587/1729-4061.2017.110142>

32. Kolpakova T., Oliinyk A., Lovkin V. Improved method of group decision making in expert systems based on competitive agents selection // 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON). 2017. doi: <https://doi.org/10.1109/ukrcon.2017.8100388>

33. Evolutionary Method for Solving the Traveling Salesman Problem / Oliinyk A., Fedorchenko I., Stepanenko A., Rud M., Goncharenko D. // 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). 2018. doi: <https://doi.org/10.1109/infocommst.2018.8632033>