

Р. Бевзюк, О. Ляшенко

Харківський національний університет радіоелектроніки, Харків, Україна

СИСТЕМА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ДЛЯ АВТОМАТИЗАЦІЇ СКЛАДІВ З ВИКОРИСТАННЯМ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ

Анотація. **Актуальність.** З збільшенням онлайн продаж в світі, зростає потреба в системах автоматизації складів. При великій кількості товарів постає питання розпізнавання схожих за виглядом товарів, але які мають різні характеристики. В зв'язку з цим використання елементів штучного інтелекту та побудові систем комп'ютерного зору для великих складських підприємств є необхідністю. **Метою роботи** є побудова системи розпізнавання різноманітних товарів за допомогою згорткових нейронних мереж. **Об'єктом дослідження** є процеси побудови та навчання системи розпізнавання товарів на складі. **Предметом дослідження** є методи розпізнавання об'єктів на базі нейронної мережі з використанням CPU та GPU. **Висновок.** Побудована система розпізнавання об'єктів на базі згорткових нейронних мереж в середовищі MatLab. Проведено експерименти з використанням CPU та GPU для навчання нейронної мережі. Отримані результати показали, що підвищення якості розпізнавання необхідно збільшувати тестову вибірку.

Ключові слова: нейрона мережа, машинне навчання, розпізнавання об'єктів, CPU, GPU.

Вступ

Однією з основних причин швидкого зростання глобального ринку автоматизації складів є значне зростання електронної комерції та вимоги до виконання замовлень, які створюються в результаті (швидка доставка, відстеження тощо). Існує підвищена потреба в системах, здатних обробляти великі обсяги замовлень із кількох позицій.

Найбільша проблема в реалізації повністю автономних систем виконання замовлень полягає в ідентифікації та обробці сторонніх предметів [1–3]. Сучасні методи сегментації зображення забезпечують достатню точність для обробки сторонніх об'єктів для операцій вибору та розміщення. Ці методи сегментації зображення дозволяють ідентифікувати та розрізнити геометричні форми, краї, поверхні та інші характеристики, достатні для надійної обробки сторонніх об'єктів. Однак цих ознак недостатньо для ідентифікації сторонніх предметів. Це обмеження технології зумовлює необхідність присутності фізичного працівника, який забезпечує ідентифікацію сторонніх предметів у процесі виконання замовлення. Для реалізації повної автоматизації система виконання замовлень повинна мати можливість автономно розпізнавати об'єкт, який обробляється, щоб об'єкт можна було транспортувати до відповідного місця призначення.

Постановка задачі

Розробити систему, яка може розпізнавати сторонні об'єкти та обробляти їх, важко через безліч візуальних явищ, які відбуваються в реальному світі, наприклад оклюзію об'єктів або світлові ефекти. Відрізнити два об'єкти, які майже однакові за формою та кольором, але є абсолютно різними, є однією з найскладніших проблем, яку можна вирішити за допомогою сучасних технологій комп'ютерного зору.

Приклад цієї ситуації наведено на рис. 1 та 2, де два майже однакові на вигляд продукти все ще мають різний вміст.



Рис. 1. Паприка копчена (мелена)

Рис. 2. Паприка (мелена)

Щоб розробити надійну систему обробки замовлень, багато з цих складних або незвичайних випадків потрібно правильно визначити. Щоб досягти цього, властивості, унікальні для кожного елемента, повинні бути знайдені та використані для правильної класифікації кожного елемента. У деяких із найскладніших випадків, коли майже не видно візуальної різниці, залишається унікальна властивість, яку можна використовувати для правильної класифікації: штрих-код. Кожен товар, який обробляється на складах, містить штрих-код, який у певний момент сканується фізичним працівником, щоб визначити його призначення під час процесу повторного збирання замовлення.

Основна увага в системі розпізнавання об'єктів буде зосереджена на використанні штрих-коду як унікальної властивості для правильної ідентифікації об'єктів. Якщо штрих-код кожного об'єкта можна знайти та надійно розпізнати, кожен об'єкт можна ідентифікувати та правильно обробити. Незважаючи на те, що існують стандартні вказівки щодо розміщення штрих-кодів, певно, не всі товари відповідають цим правилам. Дуже цікаво дослідити, чи існує корисний зв'язок між характеристиками об'єкта та розташуванням штрих-коду. Якщо такий зв'язок існує, його

можна використовувати для дуже ефективного підбору та сканування товарів на складі, усуваючи потребу людей у ручній ідентифікації об'єктів.

Машинне навчання

Машинне навчання – це галузь, яка зосереджена на розробці алгоритмів, які роблять прогнози на основі даних. Завдання машинного навчання має на меті ідентифікувати (вивчити) функцію $f: X \rightarrow Y$, що відображає вхідну область X (даних) на вихідну область Y (можливих прогнозів) [4]. Функції f вибираються з різних класів функцій, залежно від типу алгоритму навчання, який використовується. Для завдання класифікації точність системи зазвичай вибирається як міра ефективності, де точність визначається як пропорція, для якої система правильно створює вихідні дані. Досвід, якому піддаються алгоритми машинного навчання, це набори даних. Ці набори даних містять набір прикладів, які використовуються для навчання та тестування цих алгоритмів.

Алгоритми машинного навчання можна в основному класифікувати на три категорії за типом наборів даних, які використовуються як досвід. Ці категорії включають контрольоване навчання, неконтрольоване навчання та навчання з підкріпленням. Системи контрольованого навчання використовують мічені набори даних $(x, y) \in X \times Y$, де x представляє точку даних, а y відповідне істинне передбачення для x . Цей навчальний набір пар введення-виведення використовується для пошуку детермінованої функції, яка відображає будь-який вхід на вихід, прогнозуючи майбутні спостереження вводу-виводу, мінімізуючи помилки, наскільки це можливо. Системи неконтрольованого навчання використовують немарковані набори даних для навчання системи. Метою неконтрольованого навчання є отримання структури з немаркованих даних шляхом дослідження подібності між парами об'єктів і зазвичай пов'язане з оцінкою щільності або групуванням даних. Системи навчання з підкріпленням відчувають не фіксований набір даних, а петлю зворотного зв'язку між системою та її досвідом [5]. Розглядається динамічне середовище, в якому як дані спостерігаються трійки стан-дія-винагорода. Метою навчання з підкріпленням є відображення ситуацій у діях з метою максимізації винагороди. Існують інші навчальні системи, які є комбінацією двох категорій, наприклад напівкероване навчання, яке використовує як позначені, так і немарковані дані [6].

Існує велика різноманітність завдань, які можна вирішити за допомогою машинного навчання. Двома популярними завданнями машинного навчання є регресійний аналіз і класифікація. У регресійному аналізі зв'язок між змінними є наближеним для успішного прогнозування значення за певних вхідних даних. Ця задача розв'язується виведенням функції $f: R^n \rightarrow R$, що відповідає даним [7]. Регресійний аналіз можна використовувати, наприклад, для прогнозування майбутніх цін на акції у світі торгівлі. Під час класифікації машину просять визначити категорію n , до якої належить певний вхід. Задачу можна розв'язати, вивішивши функцію $f: R^n \rightarrow \{1, \dots, n\}$ [8]. Популярною проблемою класифікації є розпізнавання об'єктів для інтелектуальних

систем. Класифікація може бути використана, наприклад, для класифікації об'єктів на складі, щоб визначити правильне призначення кожного об'єкта, за допомогою сучасного розпізнавання об'єктів із використанням алгоритмів глибокого навчання [5].

Правила розміщення штрих-коду

Існує багато програм, які можуть знаходити та зчитувати штрих-коди зі складних сцен за допомогою камери та використовувати їх для ідентифікації об'єктів. Різниця в освітленні, позах і спотворенні перспективи може ускладнити ідентифікацію штрих-коду, але існують алгоритми, які пропонують адаптовані рішення для цих складних сцен. Найцікавішим є випадок, коли штрих-код приховано в результаті пози продукту, і його неможливо ідентифікувати в результаті цього без зміни положення об'єкта. Складський контейнер складається з предметів, на яких видно штрих-код, а також інших предметів зі штрих-кодами, прихованих за іншими поверхнями.

Відомо, що існує певна схема розміщення штрих-коду на предметах різного розміру та форми. Більшість товарів, які обробляються на складах, відповідають інструкціям щодо розміщення символів штрих-кодів. Ці настанови містять загальні принципи, які застосовуються, обов'язкові правила та рекомендації щодо розміщення символу штрих-коду на певних типах упаковок.

Ці вказівки можуть виявитися дуже зручними для прогнозування правильного розташування штрих-коду, коли штрих-код приховано. Однак у багатьох випадках розміщення штрих-коду на товарах може порушувати правила. Хоча не всі предмети відповідають цим правилам розміщення штрих-кодів, дуже цікаво дослідити, чи існує корисний зв'язок між такими характеристиками об'єкта, як розмір або форма, і розташуванням штрих-коду. Якщо цей зв'язок існує, його можна використовувати для дуже ефективного вибору та сканування товарів на складі, усуваючи потребу в ідентифікації об'єкта за допомогою людини. Машинне навчання буде впроваджено для отримання цих моделей розміщення штрих-кодів на об'єктах різних форм і розмірів. Аналізуючи форму, текстуру та набору інших можливих функцій пропонується нова структура розпізнавання об'єктів, яка повинна надійно визначати розташування штрих-коду об'єктів.

В ході досліджень було визначено структуру повної системи розпізнавання об'єктів. Було вирішено, що систему буде створено з використанням згорткової нейронної мережі (ЗНМ). Тоді для цього методу локалізації штрих-коду потрібно побудувати схему розміщення штрих-коду на продуктах за допомогою згорткової нейронної мережі, щоб визначити, на якій поверхні товару лежить штрих-код.

Щоб автоматизувати фазу повторного генерування системи виконання замовлень, нещодавно розроблена структура розпізнавання об'єктів має задовольнити двом важливим вимогам:

1. Система повинна досягати 40% точності передбачення локалізації штрих-коду: оскільки системи виконання замовлень обробляють величезну

кількість продуктів, нові системи бачення повинні прагнути до принаймні 40% точності передбачення. У подальшому можна покращити, додавши більше даних для навчання, або запровадивши більш оптимальні методи навчання.

2. Система повинна бути масштабованою до дуже великих наборів продуктів: спроектована система бачення повинна мати можливість обробляти поточні набори продуктів, які складаються в порядку 107 різних продуктів.

Визначення параметрів

Щоб побудувати структуру локалізації штрих-коду, спочатку визначаються вхід, вихід і визначення класу. Складські підприємства розробляють рішення автоматизації для різних ринків: посилок, складів і аеропортів, що означає, що об'єкти, з якими ці рішення обробляють, зазвичай мають велику різноманітність за формою, кольором, вагою та іншими фізичними властивостями. Важливо розуміти, що в цьому проєкті акцент робиться на складському сегменті. Крім того, продукти вибираються зі стандартизованих категорій, оскільки це дозволяє уникнути неоднозначності щодо визначення та тлумачення класу. Сфера дії цього дослідження обмежена елементами категорії "Коробки", для яких будується доказ концепції. Після створення робочої структури до неї можна додати інші категорії продуктів.

Створений набір даних складається із зображень елемента категорії коробки, взятих із вигляду зверху збоку. Приклади цих зображень з окремими предметами в різних позах можна знайти на рис. 3 та 4.



Рис. 3. Коробка пластівців із видимим штрих-кодом у верхній частині



Рис. 4. Коробка печива з прихованим штрих-кодом

І для коробки пластівців, і для коробки з печивом система локалізації штрих-коду повинна мати можливість визначати розташування штрих-коду.

Зображення будуть імпортовані у застосунок як дані RGB з роздільною здатністю 400x300. Таким



Вхідні зображення

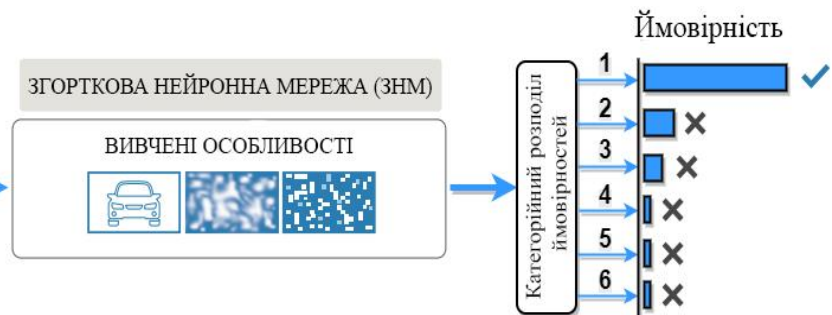


Рис. 6. Визначення класу для об'єкта у формі коробки

чином, вхідні дані складаються з матриці дизайну $X \in R^{N \times 400 \times 300 \times 3}$, де «N» представляє кількість зразків у наборі даних.

Оскільки згорткові нейронні мережі є типом контрольованого навчання, для етапу навчання рfспjсеуге потрібні дані з мітками. Для ефективного маркування всіх даних вибрано таке визначення класу:

$$c_n = \{1, 2, 3, 4, 5, 6\}$$

У цьому визначенні класу кожне з чисел представляє одну з шести граней об'єкта розміром з коробку. Це визначення класу завжди застосовується з точки зору камери, рис. 5.

Для всіх випадків, грань яка знаходиться найближче до камери, визначається як верхня грань (с=1), звідти визначаються інші обличчя.

Грань, протилежна верхній грані, є нижньою (с=3). Гранями, що прилягають до верхньої та нижньої граней, є передня (с=2), задня (с=4), права (с=5) і ліва грані (с=6).

Враховуючи перспективу камери, визначення тлумачиться та ілюструється як:

$$c_n = \{1, 2, 3, 4, 5, 6\} = \{\text{верх, спереду, знизу, ззаду, справа, зліва}\}$$

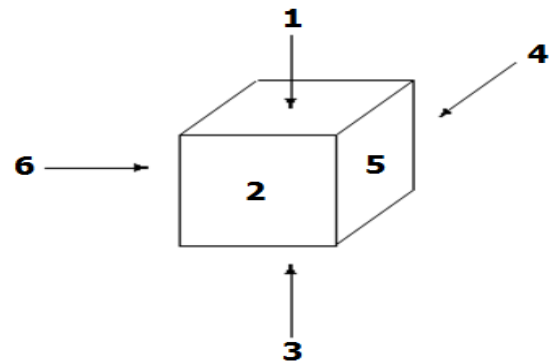


Рис. 5. Пропонована структура розпізнавання об'єктів із визначенням класу

Вихідні дані застосунку складаються з категоричного вектора розподілу ймовірностей, який визначає ймовірність наявності штрих-коду на певній грані. Можливий результат можна визначити за допомогою:

$$c_n = \{1, 2, 3, 4, 5, 6\} = \{0.8, 0.1, 0.05, 0.02, 0.02, 0.01\}$$

Цей результат означає, що з імовірністю 80% штрих-код буде розміщено на верхній стороні, 10% – на лицьовій стороні тощо. Повний каркас зображено на рис. 6.

Згорткова нейронна мережа

Згорткові нейронні мережі наразі є одним із найвідоміших алгоритмів глибокого навчання із зображеннями. У той час як для традиційного машинного навчання відповідні функції потрібно витягувати вручну, глибоке навчання використовує необроблені зображення як вхідні дані для вивчення певних функцій. ЗНМ складаються з вхідного та вихідного рівня та кількох проміжних рівнів між входом і виходом. Прикладами проміжних шарів є згорткові шари, шари максимального об'єднання та повністю зв'язані шари [9].

Архітектури ЗНМ відрізняються кількістю та типом шарів, реалізованих для конкретного застосування. Для безперервних відповідей мережа повинна включати рівень регресії в кінці мережі, тоді як для категоричних відповідей система повинна включати функцію класифікації та рівень. Нейрони в кожному шарі ЗНМ розташовані в тривимірному порядку та перетворюють тривимірний вихід із тривимірного вхідного. Для нашої конкретної програми вхідний рівень містить зображення як 3D-вхідні дані зі значеннями висоти, ширини та RGB як розмірів. Далі на згортковому шарі нейрони прикріплюються до областей зображення та перетворюються на тривимірний результат.

Конфігурації ЗНМ складаються з безлічі прихованих шарів. У кожному шарі обсяги активації змінюються за допомогою диференційованих функцій. Існують чотири основні типи шарів, які використовуються для побудови конфігурацій ЗНМ:

1. Згортковий рівень – згорткові фільтри використовуються для отримання карти активації з вхідних даних.

2. Rectified Linear Unit Layer (ReLU) – фільтрує від'ємні значення, щоб отримати лише додатні значення для набагато швидшого часу навчання.

3. Рівень об'єднання – виконує нелінійну низьку дискретизацію та скорочує кількість параметрів для простішого виведення.



Рис. 7. Конфігурація рівня згорткової нейронної мережі.

Розроблена мережа складається з 13 рівнів із різними розмірами фільтрів

Використання графічного процесора значно скорочує час навчання, і це можна зробити за допомогою Matlab без додаткового програмування. Однак для паралельних обчислень необхідний графічний процесор NVidia на основі CUDA, щонайменше 3.0. Існує також можливість використовувати кілька графічних процесорів, що ще більше зменшує час обробки. Нарешті, хмарні обчислення GPU розглядають використання хмарних ресурсів для обчислювальної потужності. Написаний код MATLAB можна покращити для цілей хмарних обчислень. У рамках цієї роботи обчислення, як на CPU, так і на GPU розглядаються для дослідження впливу використання GPU на зменшення часу обчислення.

4. Повністю зв'язаний рівень – обчислює оцінки ймовірності класу шляхом виведення вектора розмірів C , де C є кількістю класів. Усі нейрони підключені до цього шару.

Створення ЗНМ в Matlab

Структура, описана в цій роботі, розроблена в Matlab. Сам застосунок призначений для роботи на настільному комп'ютері. Використовуються два набори інструментів Matlab: Parallel Computing і Neural Network Toolboxes. Ці набори інструментів забезпечують алгоритми згорткового навчання, попередньо запрограмовані рівні та згорткові мережі, а також можливість паралельного обчислення для значного скорочення часу обчислення.

При визначенні конфігурації мережевого рівня важливо зазначити, що не існує точної формули для оптимальної конфігурації рівня. Замість цього найкращий підхід – це метод проб і помилок, коли кілька конфігурацій шарів досліджуються та порівнюються, щоб побачити, наскільки добре вони працюють. Попередньо навчені мережі, можна використовувати як початкову конфігурацію рівня. Однак у цьому випадку, переслідуються власні реалізації конфігурації шару. Вибрану конфігурацію шару показано на рис. 7. Навчання глибокої згорткової мережі, зображеної на рис. 7, може зайняти різну кількість часу, залежно від розміру набору даних і доступної потужності обробки. Для навчання ЗНМ доступні три варіанти обчислень, і вибір найбільш оптимального є вирішальним залежно від кількості доступного часу для вирішення конкретного завдання:

1. Обчислення на основі CPU.

2. Обчислення на основі GPU.

3. Хмарні обчислення GPU.

Варіанти обчислень на основі CPU є найбільш простими та доступними. Однак оскільки центральний процесор обчислює завдання в послідовній конфігурації, навчання мережі займає найдовше часу за допомогою цього методу.

Навчання ЗНМ

Під час імпорту позначеного набору даних 75%/25% розподіляються між даними навчання та даними перевірки. Це означає, що 75% даних використовується для навчання мережі, а 25% даних використовується для перевірки мережі. Після досягнення достатньої точності перевірки можна використати додатковий набір тестів, щоб побачити, наскільки добре працює мережа. Перша конфігурація виконується на основі обчислювальної моделі ЦП. Використовується процесор Intel Core i7 720QM з чотирма фізичними та чотирма віртуальними ядрами. Хід навчання та результат показані на рис. 8:



Рис. 8. ЗНМ навчальний запуск 1, з використанням центрального процесора. Точність перевірки 16% досягається за час навчання приблизно 431 хвилину

Незважаючи на те, що використовується CPU має чотири фізичні ядра, за час, що минув, можна відразу помітити, що навчання з CPU все одно займає надзвичайно багато часу, і його неможливо точно налаштувати в рамках цього проекту. Навіть якщо центральний процесор складається з кількох ядер і є найкращим у лінійці, велика тривалість обробки залишається проблемою. Дивлячись на результати тесту, точність перевірки в 16% є дуже недостатньою, і схоже, що ЗНМ не дізнався жодних цінних функцій, які можна було б використати для правильної класифікації. Змінюючи кількість функцій, які обчислюють згорткові шари, і розмір фільтра, який використовує згортковий шар, система додатково налаштована для підвищення точності перевірки. Однак з другого запуску на основі GPU беруться обчислення. Для наступних прогонів використовується графічний процесор NVidia GTX1060 і процесор Intel i7-6700HQ. Результати другого запуску показані на рис. 9.

Що відразу можна відзначити, так це те, що час обчислень зменшився з 431 хвилини до приблизно 1 хвилини. Це означає, що час обчислення зменшився приблизно в 400 разів. Це принаймні дозволяє ефективно налаштувати ЗНМ. Дивлячись на результати перевірки, досягнуто точності 40,16%. У той час як точність навчального набору збільшується приблизно до 100%, точність перевірки продовжує коливатися близько 40%.

Порівнюючи втрату під час навчання з втратою підтвердження, можна побачити, що хоча навчальний набір справді збігається до нуля, набір підтвердження залишається приблизно 1,75. Подальше тонке налаштування виконується для ЗНМ, щоб дослідити, чи може зміна швидкості навчання, розміру мінібатча чи епохи, або конфігурацій і налаштувань шару ще більше покращити ЗНМ. Це робиться кілька разів поспіль, показуючи деякі з цікавих результатів на рис. 10-12.

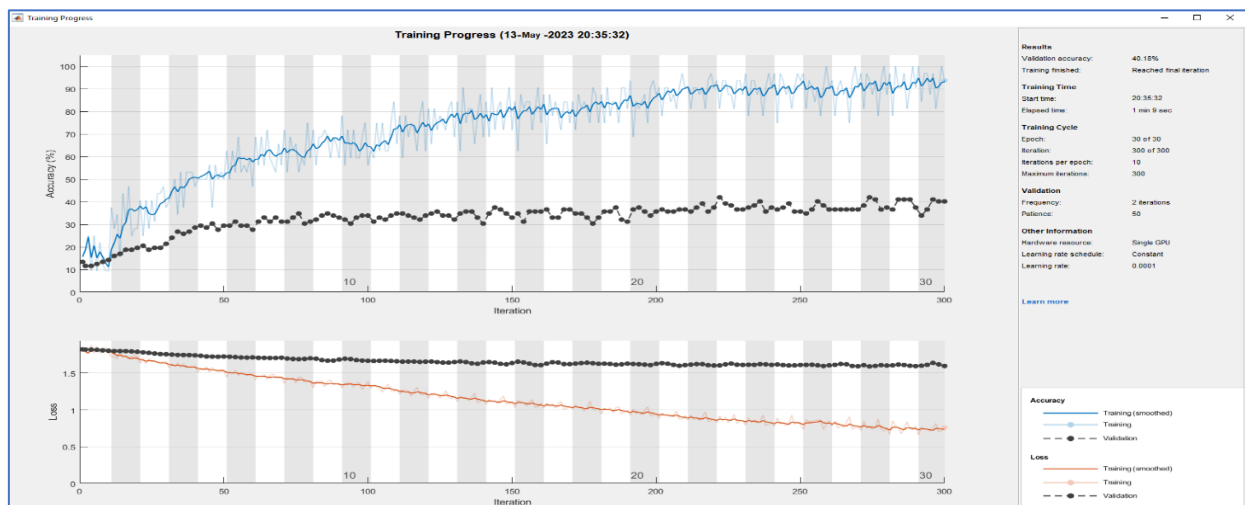


Рис. 9. Тренування ЗНМ 2 з використанням GPU. Параметри системи вже дещо налаштовані. За рахунок зменшення розміру фільтра та кількості фільтрів точність перевірки 40,18% досягається з часом навчання 69 секунд

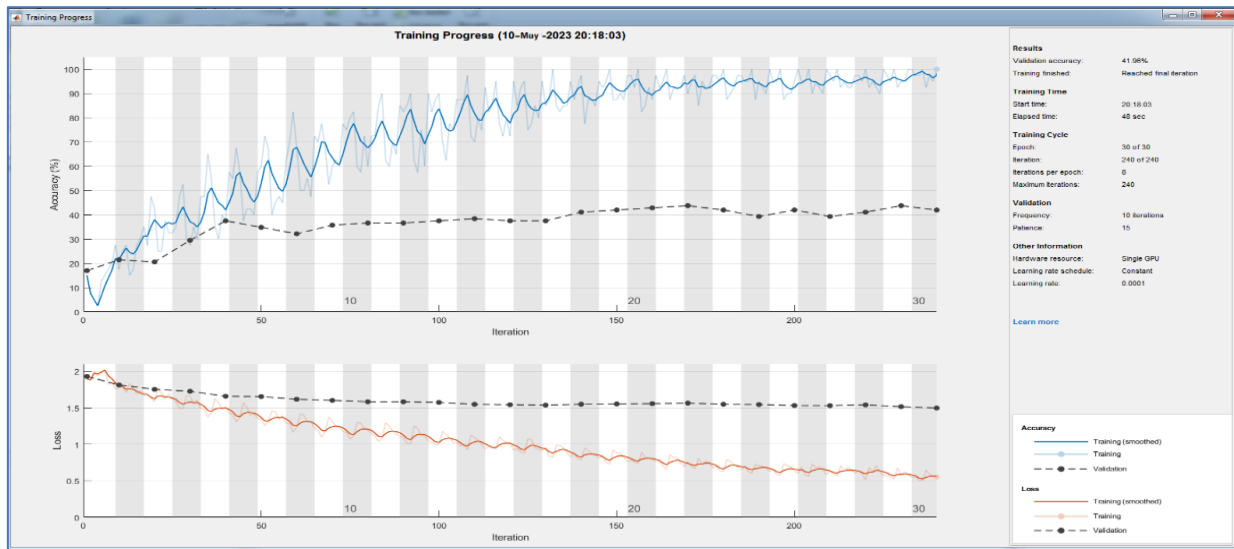


Рис. 10. Навчальний запуск ЗНМ (з GPU) зі зміненими темпами навчання та розмірами фільтрів шарів. Точність перевірки 41,96% досягається за час навчання 48 секунд

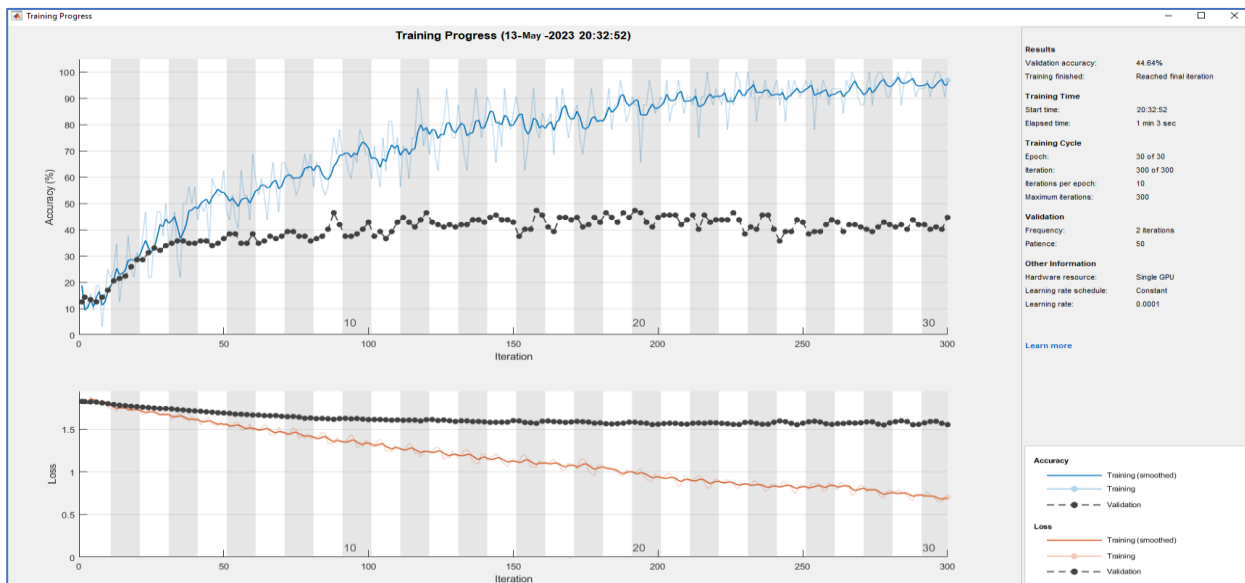


Рис. 11. Навчальний запуск ЗНМ (з GPU) зі зміненими темпами навчання та розмірами фільтрів шарів. Точність перевірки 44,64% досягається за час навчання 63 секунди

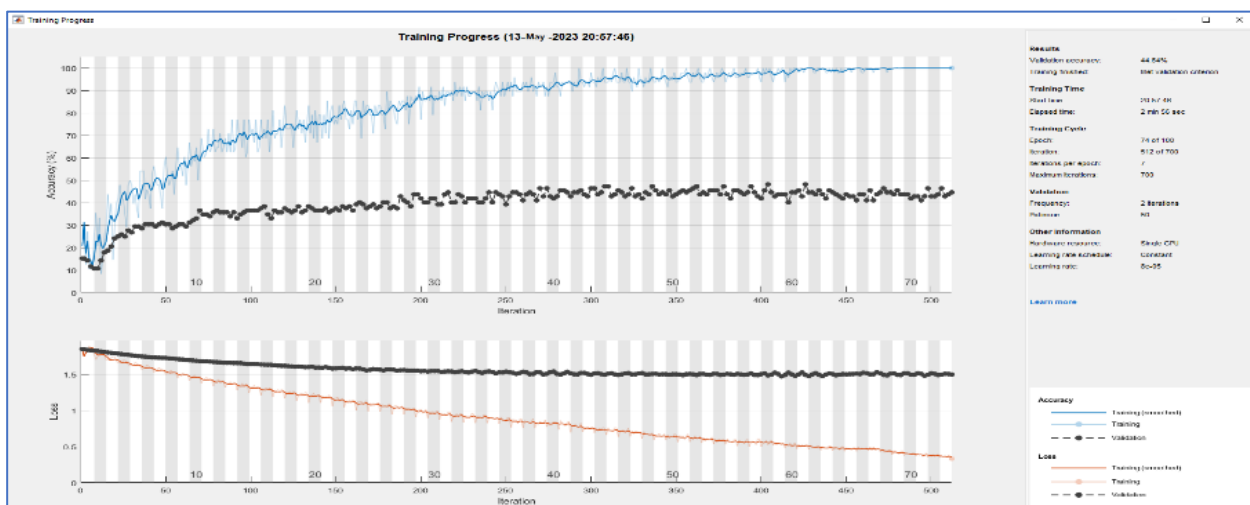


Рис. 12. Навчальний запуск ЗНМ (з GPU) зі зміненими темпами навчання та розмірами фільтрів шарів. Точність перевірки 44,64% досягається за час навчання 2 хвилини 56 секунд

Після проведення кількох десятків тестів було зроблено висновок, що точність перевірки набору даних становить приблизно 45% після точного налаштування системи. Точність навчального набору зростає до 100%, але набір перевірки продовжує коливатися на рівні 45%. З цих результатів можна зробити висновок, що з поточним розміром набору даних близько 1000 зображень ЗНМ не може достатньо розрізнити екземпляри даних. Ці результати є ідеальною ілюстрацією випадку, коли мережа не узагальнює достатньо добре, щоб не побачити екземпляри даних. ЗНМ переповнив дані, як видно зі 100% точності на навчальному наборі, а налаштування швидкості навчання та розмірів пакетів, здається, не мають жодного додаткового ефекту після точного налаштування, щоб досягти точності близько 45%. Висновок, який можна зробити, такий: для навчання ЗНМ потрібен більший набір даних, щоб зменшити переобладнання та досягти вищої точності.

Висновки

Розглядаючи запропоновану структуру розпізнавання об'єктів у світлі визначених вимог і обмежень, можна зробити висновок, що досягнуто достатньої точності 44,64%, що перевищує вимогу 40%. Логічний висновок, який можна зробити з результатів навчання, полягає в тому, що набір даних має бути різко збільшений, щоб запобігти переобладнанню даних. Крім того, набори даних усіх інших категорій продуктів повинні бути додані до ЗНМ, щоб

структура була реалізована на складі. Хоча генерація та збір цих даних може зайняти багато часу, з результатів, досягнутих на сьогоднішній день, можна побачити, що навіть з невеликими наборами даних показані багатообіцяючі результати.

Під час навчання ЗНМ необхідна реалізація GPU. Навчання ЗНМ лише за допомогою процесора може зайняти місяці для повного набору даних категорії продуктів. Логічним є висновок, що тонке налаштування системи під час цього процесу неможливе в цьому часовому режимі. Масштабованість, здається, не є проблемою при реалізації обчислень GPU, і, крім того, можна реалізувати хмарні обчислення, щоб ще більше скоротити час навчання під час навчання дуже великих наборів даних. Коли досягається достатня точність класифікації ЗНМ, можна застосувати розумні стратегії комплектування, коли робот-комплектатор намагається класифікувати продукти, показуючи грані продукту вздовж вектора ймовірності класифікації. Цей метод забезпечує негайний вибір, коли продукт класифіковано правильно, і вторинний вибір у разі неправильної класифікації, коли робот намагається сканувати обличчя відповідно до вектора ймовірності класифікації, доки не буде знайдено штрих-код. Це гарантує мінімальні або повну відсутність витрат на помилки, оскільки продукт завжди сканується. У цьому випадку точність класифікації ЗНМ корелює з тим, наскільки швидко сканер штрих-кодів класифікує продукти з першої спроби, де підвищення точності означає швидший вибір.

СПИСОК ЛІТЕРАТУРИ

1. Kuchuk, H., Kovalenko, A., Ibrahim, B.F. and Ruban, I. (2019), "Adaptive compression method for video information", *Int. Journal of Advanced Trends in Computer Science and Engineering*, 8(1), pp. 66-69, DOI: <http://dx.doi.org/10.30534/ijatcse/2019/1181.22019>
2. Коваленко А. А., Кучук Г. А. Методи синтезу інформаційної та технічної структур системи управління об'єктом критичного застосування. *Сучасні інформаційні системи*. 2018. Т. 2, № 1. С. 22–27. DOI: <https://doi.org/10.20998/2522-9052.2018.1.04>
3. Свиридов А. С., Коваленко А. А., Кучук Г. А. Метод перерозподілу пропускної здатності критичної ділянки мережі на основі удосконалення ON/OFF-моделі трафіку. *Сучасні інформаційні системи*. 2018. Т. 2, № 2. С. 139–144. DOI: <https://doi.org/10.20998/2522-9052.2018.2.24>
4. R. Bekkerman, M. Bilenko, J. Langford, *Scaling Up Machine Learning*, Cambridge University Press, January 2012
5. Mitchell, T. (1997). *Machine Learning*. McGraw Hill.
6. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Cambridge: MIT Press
7. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2017), "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, Vol. 60, No. 6
8. Alexander Toshev, Christian Szegedy (2014), "DeepPose: Human Pose Estimation via Deep Neural Networks", DOI: <https://doi.org/10.1109/CVPR.2014.214>
9. I. Mykhailichenko, H. Ivashchenko, O. Barkovska and O. Liashenko, "Application of Deep Neural Network for Real-Time Voice Command Recognition," *2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek)*, Kharkiv, Ukraine, 2022, pp. 1-4, doi: 10.1109/KhPIWeek57572.2022.9916473.

Received (Надійшла) 17.03.2023

Accepted for publication (Прийнята до друку) 16.05.2023

Object recognition system for component automation using a convergular neural network

Roman Bevzyuk, Oleksii Lyashenko

Abstract. Topicality. With the increase in online sales in the world, the need for warehouse automation systems is growing. With a large number of products, the question arises of recognizing products that are similar in appearance, but have different characteristics. In this regard, the use of elements of artificial intelligence and the construction of computer vision systems for large warehouse enterprises is a necessity. **The purpose of the work** is to build a system for recognizing various goods using convolutional neural networks. **The object** of the study is the process of building and training a system for recognizing goods in a warehouse. **The subject** of the research is object recognition methods based on a neural network using CPU and GPU. **Conclusion.** An object recognition system was built based on convolutional neural networks in the MatLab environment. Experiments were conducted using CPU and GPU for neural network training. The obtained results showed that to increase the quality of recognition, it is necessary to increase the test sample.

Keywords: neural network, machine learning, object recognition, CPU, GPU.