



✉ Correspondence author

S. S. Ivasiv

stanislav.ivasiv.mknus.2022@lpnu.ua

Article received 06.10.2023 p.

Article accepted 26.10.2023 p.

UDK 004.4

УДК 004.4

В. М. Теслюк, С. С. Івасів

Національний університет "Львівська політехніка", м. Львів, Україна

## СИСТЕМА ДЛЯ РОЗПІЗНАВАННЯ ЕЛЕМЕНТІВ ОДЯГУ ТА ЇХ КОЛЬОРІВ НА ЗОБРАЖЕННІ

У роботі запропонована система розпізнавання елементів одягу та їх кольорів, розроблена мовою програмування Java у вигляді вебзастосунку, із використанням архітектури Model-View-Controller (MVC). Для зручності розроблення та складання проєкту використано систему автоматичного збирання Gradle. Система оперує принципами штучних нейронних мереж на платформі Clarifai, детально розглядаючи процес виявлення об'єктів на зображенні та ефективність роботи системи в умовах реального використання. Взятю InceptionV2 як базову модель з оброблення зображень, використано техніку FPN для аналізу зображень з різними масштабами, зі зміною розміру зображень до 512 пікселів, і здійснено навчання за допомогою стохастичного градієнтного спуску із використанням техніки виділення важких прикладів (негативів). Застосовано алгоритм зрізу по медіані для визначення домінантного кольору елементу одягу; визначення домінантного кольору кожного елемента одягу розглянуто окремо, з акцентом на точність розпізнавання, яка становить 100 %. Реалізовано підхід для визначення найближчого кольору із назвою на основі домінантного кольору. Підхід ґрунтується на обчисленні евклідової відстані між двома точками у тривимірному просторі, де здійснюється перебір 140 кольорів з назвою колірної моделі RGB для визначення кольору, найближчого за назвою до домінантного. Проте під час пошуку найближчого кольору з назвою виявлено певні обмеження, які призводять до пониження точності до 60 %. Вплив різних факторів, таких як освітлення та якість зображення, детально розглянуто в контексті їх впливу на результат роботи системи. Інтерфейс користувача представлено як інтуїтивний інструмент для взаємодії із системою, що дає змогу перевіряти розпізнані елементи одягу та визначені кольори. Окрім того, додатково надано рекомендації з основних колірних комбінацій, які може використати користувач для поліпшення свого стилю одягу. У роботі наведено приклад застосування системи на реальному зображенні, візуально продемонстровано результати та описано якість розпізнавання елементів одягу та їх кольорів. Однією із ключових особливостей запропонованої системи є її гнучкість і можливість масштабування. Якість розпізнавання можна поліпшувати надалі за допомогою додаткового тренування моделі на більших наборах даних. У роботі висвітлено питання оптимізації процесів оброблення даних та подальшого аналізу отриманих результатів. Оцінено час та складність виконання алгоритмів пошуку домінантного кольору та пошуку найближчого кольору з назвою. Наукова новизна результатів дослідження полягає в тому, що вперше розроблено комплексну систему розпізнавання елементів одягу та їх кольорів із використанням ШНМ та реалізовано підхід пошуку найближчого кольору з назвою, де під комплексністю системи розуміємо повний цикл оброблення зображення в одній системі – визначення розташування елементів одягу, класифікація елементів одягу, визначення домінантного кольору елементів одягу та визначення найближчого кольору з назвою для подальшого візуального відображення усієї інформації користувачу та надання основних колірних комбінацій для виправлення власних кольорів, чи для того, щоб упевнитись у правильності колірної комбінації. Практична значущість результатів дослідження полягає у розробленні структури та алгоритмів роботи застосунку, програмній реалізації системи, яка змінює підхід до вибору та перевірки колірних комбінацій. Такий підхід дає користувачу чіткі назви кольорів, згідно з якими можна створювати власні, правильні за комбінаціями кольорів, набори одягу або якісно здійснювати пошук елементів одягу у мережі за отриманими кольорами. Також систему можна використати як засіб для розмітки "даних" або, іншими словами, для підготовки матеріалу, який буде використовуватись для тренування моделей машинного навчання чи нейронних мереж. Перспективи розвитку передбачають інтеграцію з іншими системами, розширення бази даних зображень для підвищення точності та використання додаткових джерел даних для розширення функціоналу системи.

**Ключові слова:** Clarifai; домінантний колір; RGB; колірні комбінації; модель машинного навчання; Feature Pyramid Network (FPN).

### Вступ / Introduction

Нині слово "стиль" трапляється всюди. Його вживають, говорячи про культуру, архітектуру, літературу, мовознавство, музику та менеджмент. Все ж сучасний стиль у всіх напрямках диктують бренди, виробники, відомі люди і сам теперішній час.

Якщо говорити про стиль одягу, сформульовано таке визначення – певна акцентованість ансамблю (костюму в широкому сенсі), продиктована такими ознаками (або їх сукупністю): віком, статтю, професією, соціальним статусом, належністю до субкультури, особистим смаком людини, епохою життя суспільства, національністю, релігій-

ною належністю, доречністю, функціональністю, способом життя та індивідуальними особливостями. Хоча сучасники трактують і бачать стиль по-своєму, все ж він має чіткий визначник і цим визначником є етикет.

Етикет – сукупність правил поведінки, що регулює зовнішні прояви людських взаємин (поведінка в громадських місцях, поводження з оточенням, форми звертання і привітання, манери поведінки, одяг тощо). Етикет регламентує поведінку людини в певних ситуаціях. А сам етикет зачіпає не тільки правила поведінки, а й правила підбору одягу відповідно до ситуації, або ж сфери діяльності чи перебування особи. А перше, що спадає на думку, коли говорять про одяг, – це кольори. Тут етикет також важливий, адже чітко трактує підбір кольорів. Зважаючи на це, питання підбору кольорів та стилю одягу залишається відкритим і в певних випадках не завжди вичерпним.

Перше завдання – визначення розташування елемента одягу на зображенні, а серед підводних каменів варто зазначити, що об'єкти на зображенні можуть міститись у різних розмірах та масштабах, що, своєю чергою, потребує використання методів для виявлення та визначення об'єктів у разі зміни їх масштабу. І ще одна проблема – перекриття об'єктів, адже у реальних сценах об'єкти можуть перекривати один одного, а це ускладнює визначення розташування на зображенні.

Наступне завдання полягає у класифікації елемента одягу, розпізнаного на зображенні; тут важливу роль відіграватиме достатня кількість даних для навчання та збалансованість класів (класи повинні мати приблизно однакову кількість прикладів), інший чинник – залежність від освітлення та кута, з якого відбувається захоплення зображення.

Завдання визначення домінантного кольору має свої аспекти, серед яких:

- відсутність чітких домінантних кольорів, тобто зображення можуть бути “зашумлені”;
- вибір самого підходу до визначення домінантного кольору, який впливатиме на точність результатів.

У завданні визначення найближчого кольору з назвою можна натрапити на суб'єктивність сприйняття кольорів, яка ґрунтується на індивідуальній відмінності в сприйнятті відтінків і насиченості, також є неоднозначність назв кольорів, що потребує вибору системи назв кольорів.

З цього можна зробити висновок, що тема актуальна і застосування моделей та ІТ засобів для повного або часткового вирішення поставлених технічних завдань щодо розпізнавання елементів одягу та їх кольорів на зображенні є доцільним.

*Об'єкт дослідження* – процес збирання, збереження та опрацювання користувацького зображення з використанням ШНМ.

*Предмет дослідження* – моделі штучних нейронних мереж для класифікації типу об'єкта та визначення його розташування, алгоритми визначення домінантного кольору.

*Мета роботи* – розроблення комплексної системи для розпізнавання елементів одягу та їх кольорів.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

- здійснити огляд літературних джерел за тематикою дослідження;
- розробити застосунок, який використовує ШНМ для класифікації елемента одягу та визначення його розташування на користувацькому зображенні і встановлює колір кожного розпізнаного елемента одягу.

*Матеріали і методи дослідження.* У дослідженні застосовано такі методи: метод теорії алгоритмів – для оцінювання складності алгоритмів; метод об'єктно-орієнтованого підходу – для розроблення застосунку.

Для ШНМ використовують окремі розділи лінійної алгебри, математичного аналізу, основ оптимізації та теорії ймовірності.

Для організації пошуку домінантного кольору та пошуку найближчого кольору з назвою застосовано застави алгоритмів і структур даних.

Застосунок розроблено мовою Java в інтегрованому середовищі розроблення IntelliJ IDEA з використанням системи автоматичного збирання Gradle.

Графік побудовано за допомогою бібліотеки matplotlib у середовищі Jupyter Notebook.

*Аналіз останніх досліджень та публікацій.* Завдання визначення місця розташування елементів на зображенні одягу та визначення типу об'єкта розглянуто у багатьох статтях. Зокрема, робота “Глибинне навчання згорткової нейронної мережі для класифікації на прикладі набору даних Fashion-MNIST” [5] дає змогу зрозуміти процес класифікації зображень. У статті описано створення згорткової нейронної мережі (CNN) для класифікації зображень одягу. Для побудови CNN автор використав стандартні їх параметри, такі як швидкість навчання, функції активації та функції втрат. Зокрема, він описує, як створити набір даних, візуалізувати дані, побудувати модель, відобразити результати та здійснити передбачення.

Наступна робота – “Як навчити модель визначення об'єктів, використовуючи Keras” [6]. У статті увагу сконцентровано на задачі об'єктного детектування та формалізації різних методів для розв'язання цієї задачі, зокрема методів R-CNN, Fast R-CNN, Faster R-CNN, YOLO та SSD. Автор розглядає основні компоненти моделі об'єктного детектування, такі як функція витрат, оптимізатор та метрики. У роботі описано процес підготовки даних, такий як створення анотацій об'єктів та генерація даних. Описано побудову та тренування моделі з використанням бібліотеки Keras, зокрема розглянуто архітектуру моделі та процес тренування.

Наведені статті допомогли уточнити засади класифікації та детектування об'єктів на зображеннях і надали важливі відомості для розроблення моделей, які можуть впевнено розпізнавати та визначати об'єкти на зображеннях одягу.

Наступним елементом є визначення домінантного кольору. Тут можна застосувати різні підходи. Найпростіший спосіб визначення домінантного кольору на зображенні – це знайти колір, якому відповідає найбільша кількість пікселів на зображенні. Спершу визначають розмір зображення (ширину та висоту), далі здійсню-

ють перебір усіх пікселів та підраховують кількість пікселів кожного кольору і вибирають колір з найбільшою кількістю пікселів.

Є й інший підхід, описаний у роботі “Домінантні кольори на зображенні з використанням кластеризації k-середніх” [17]. Автор починає зі зчитування зображення та його перетворення з формату BGR у RGB, щоб забезпечити правильну інтерпретацію кольорів. Зображення перетворюється на одновимірний масив пікселів, де кожен піксель представлений трьома значеннями (R, G, B). Використовуючи бібліотеку scikit-learn, автор застосовує алгоритм k-середніх з визначеною кількістю кластерів. Домінантні кольори визначаються на основі центрів кластерів. Ці центри і є домінуючими кольорами на зображенні. Результатом роботи алгоритму є масив із RGB значеннями домінуючих кольорів. Загалом, стаття та наведений код надають зрозумілий огляд та інструкції щодо використання алгоритму k-середніх для визначення домінуючих кольорів на зображенні.

Ще одне завдання – знайти найближчий колір з назвою. Для цього можна застосувати алгоритм k-NN [3]. k-найближчих сусідів (k-NN) – це алгоритм машинного навчання, який можна використовувати для вирішення проблеми пошуку найближчого кольору для певної точки в просторі RGB. Ключова ідея цього методу полягає в тому, що у кольорів, ближчих до точки, більша ймовірність бути схожими на неї. Спочатку потрібно мати набір даних, що містить кольори з назвами (мітками) та їхні RGB-представлення. Цей набір даних буде використовуватися для навчання моделі k-NN. Параметр “k” визначає, скільки найближчих сусідів буде враховано під час прийняття рішення. Зазвичай вибирають різні значення “k” та оцінюють, які з них працюють найкраще для конкретного завдання. Модель k-NN не потребує в явному вигляді “навчання”, оскільки це лінива модель, але набір даних повинен бути завантажений у неї для подальшого використання. Модель k-NN зберігатиме цей набір даних для подальших порівнянь. Для точки без назви в просторі RGB буде обчислено відстань від неї до кожної точки з навчального набору даних і вибрано “k” точок із найменшими відстанями. Знайшовши “k” найближчих сусідів, визначають, яких класів (назв кольорів) найбільше серед цих сусідів. Клас, який трапляється найчастіше, вважається передбаченим класом для точки без назви. Передбачений клас є найближчим кольором з назвою для певної точки без назви.

Переваги методу k-NN – його простота і придатність для різних видів даних. Однак у нього є недоліки – висока обчислювальна складність, особливо за великого обсягу даних, а також необхідність належного вибору значення “k” для досягнення найкращої точності.

Отже, виконавши аналіз, варто зазначити, що необхідно створити систему, яка об’єднає описані або еквівалентні підходи, що вирішуватимуть проблему розпізнавання елементів одягу та їх кольорів на зображенні.

## Результати дослідження та їх обговорення / Research results and their discussion

На рис. 1 зображено структуру розробленої системи у вигляді блок-схеми.

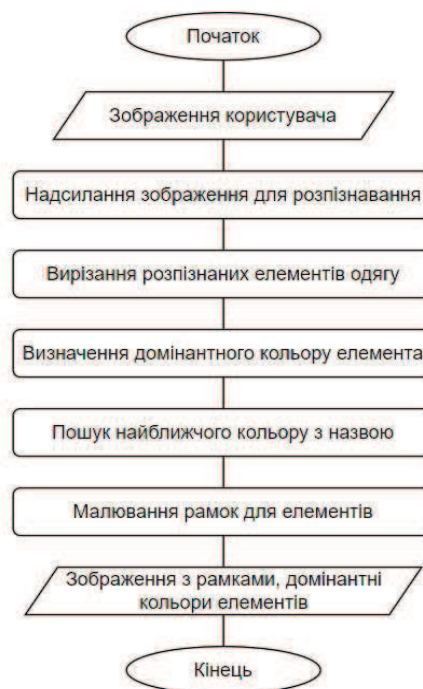


Рис. 1. Блок-схема повного циклу оброблення зображення /  
Flowchart of the complete image processing cycle

*Визначення об’єктів на зображенні та їх класифікація.* Застосовано платформу Clarifai [8], яка надає можливість скористатись готовою моделлю для розпізнавання елементів одягу на зображенні. Для використання моделі зареєстровано акаунт та створено власний проект у межах платформи, що генерує унікальний API ключ. Згідно із документацією Clarifai [9] отримано клієнт, розроблений на технології gRPC для мови програмування Java. Використовувана модель для розпізнавання елементів одягу має назву "apparel-detection", а її версія "1ed35c3d176f45d69d2aa 7971e6ab9fe". Вказана модель дає змогу визначати місцезнаходження елементів одягу на зображенні. Таких елементів є 34, до них належать: сумка, ремінь, метелик, браслет, сукня, сережки, окуляри, рукавички, шпилька для волосся, шапка, пов’язка, панчішно-шкарпеткові вироби, комбінезон, рукавиці, намисто, краватка, верхній одяг, штани, шпилька/брошка, кишень, каблучка, комбінезон, шарф, взуття, шорти, спідниця, шкарпетки, сонцезахисні окуляри, підтяжки, купальники, затискач для краватки, топ, жилет, годинник. Проте розроблена система функціонуватиме з трьома елементами: верхній одяг, штани, взуття. Вхідне зображення повинне містити людей, зображених у розмірі від 3/4 до повного зросту, обличчям вперед або під кутом, із різноманітним фоном від простого, у стилі каталогу, до жвавих вуличних сцен. Моделі показує гірші результати детектування та класифікації елементів одягу, якщо їй передають такі типи зображень: зображення, обрізані близько до окремої частини одягу людини (тобто плечі та голова або лише тулуб), зображення окремого одягу великим планом (де більшу частину зображення займає одяг чи

елемент одягу) і зображення одягу, який не на людині. Система передає зображення платформі у вигляді байтів, а у відповідь платформа передає системі серіалізовані дані, які містять розпізнані елементи та їхнє розташування на зображенні.

До цієї функції також входить вирізання розпізнаних елементів одягу з головного зображення і зберігання кожного елемента одягу як окремого зображення для подальшого опрацювання.

*Визначення домінантного кольору.* Для визначення домінантного кольору застосовано реалізацію під назвою ColorThief [10], яку запропонував розробник Локеш Дакар. Цей спосіб пошуку домінантного кольору ґрунтується на алгоритмі зрізу по медіані.

Алгоритм зрізу по медіані містить такі кроки:

Крок 1. Вибір опорного кольору: один піксель (колір) зі списку вибирають як опорний колір. Це може бути будь-який піксель зі списку.

Крок 2. Поділ на дві частини: Система ділить список на дві частини: одну, де кольори менші від або дорівнюють опорному кольору, та іншу, де кольори більші від опорного. Вона робить це, порівнюючи компоненти (R, G, B) кожного пікселя з компонентами опорного кольору.

Крок 3. Визначення потрібного підписку. Система визначає, в якому підписку міститься медіанний колір. Якщо кількість кольорів в меншому підписку (кольори менші або дорівнюють опорному) менша від або дорівнює  $k$ , то медіанний колір у цьому підписку. В іншому разі медіанний колір у більшому підписку.

Крок 4. Рекурсивний виклик. Система продовжує викликати алгоритм медіанного зрізу для нового підписку, в якому міститься медіанний колір, застосовуючи рекурсію.

Крок 5. Система повторює цей процес (кроки 1–4) до досягнення базового випадку, коли в списку лише один колір, і цей колір є медіанним.

У результаті під час розроблення системи застосовано портовану версію [19] реалізації ColorThief для мови програмування Java.

*Пошук найближчого кольору з назвою.* Після знаходження домінантного кольору для елемента одягу система містить змінну, яка представляє колір у колірній моделі RGB. RGB дає змогу закодувати 16 777 216 кольорів, а система повинна надавати текстове подання кольору, зрозуміле для пересічної людини.

Для 140 кольорів є визначені назви [12], тому в роботі написано функцію, яка знаходить найближчий реальний колір способом порівняння попередньо отриманого домінантного кольору зі списком основних кольорів колірної моделі RGB. На вхід функції система передає масив з трьох цілих чисел, які представляють колір у колірній моделі RGB. Застосовано та заповнено значеннями структуру даних Map, де ключ є назвою кольору, а значення – комбінація з трьох чисел, яка представляє колір у RGB. Оголошено змінну, яка відповідатиме за результат обчислень у межах циклу, спочатку вона має велике значення як заглушка. Також оголошено змінну, яка міститиме назву найближчого реального кольору, початкове її значення “null”. Система отримує значення кольору із Map і записує його в окремий масив. Далі здійснюється пошук найближчого кольору за формулою (1)

$$\text{CalculatedValue} = \frac{1}{\sqrt{(a[0]-b[0])^2 + (a[1]-b[1])^2 + (a[2]-b[2])^2}}, \quad (1)$$

де *CalculatedValue* – значення, яке вказує на близькість кольору (що ближчі кольори між собою, то менше значення набуває змінна),  $a$  – масив з трьох елементів, що є визначеним домінантним кольором,  $b$  – масив із трьох елементів, який є одним з 140 кольорів колірної схеми RGB, які мають назву.

Система перевіряє, чи знайдене значення близькості кольорів є меншим за попереднє значення змінної, яка містить значення близькості кольорів. У разі істинності умови змінюється значення близькості кольорів на щойно обчислене, у випадку хибності залишається попереднє значення близькості кольорів. Заповнюється значення назви кольору завдяки отриманню ключа поточного елемента. У результаті отримано структуру даних Map, де ключ – це назва найближчого кольору, а значення – масив, що представляє комбінацію RGB найближчого кольору.

*Повний цикл оброблення зображення.* Здійснено опис послідовності дій, які виконуються у повному циклі оброблення зображення:

Спершу передається у функцію користувачке зображення. Створюється об’єкт для поміщення в нього gRPC каналу, а також заглушка, яка містить унікальний користувачкий номер акаунта Clarifai. Із використанням функції клієнта gRPC здійснюється надсилання зображення користувача через кодування його BASE64, тобто надсилання зображення користувача на платформу Clarifai у вигляді набору байтів. Створюється об’єкт клієнта Clarifai, який містить відповідь від платформи. Створюється список, який міститиме майбутні об’єкти (елементи одягу із зображення). Записується користувачке зображення у об’єкти типу File. Передається записане користувачке зображення у вигляді файлу в сутність BufferedImage, яка використовується для оброблення зображення. Визначається ширина та висота користувачкого зображення у пікселях. Створюється окремий об’єкт для зображення, на якому буде обведено елементи одягу у вигляді рамок. Створюється об’єкт класу Graphics2D [13], який дасть змогу додавати прямокутники (рамку) на користувачке зображення. Створюється кисть завтовшки 3 пікселі та передається об’єкту Graphics2D. Створюється змінна, яка слугуватиме лічильником, що відповідатиме за колір рамки.

Далі здійснюється перевірка точності та назви елемента. “Проходять” ті елементи, точність яких понад 60 %, а сам об’єкт повинен бути або туфлями, або верхнім одягом (піджак, куртка), або штанами. Здійснюється розрахунок крайніх точок елемента на зображенні. Створюється масив, який містить два елементи, сам масив – це ліва верхня точка рамки навколо розпізнаного елемента одягу. Здійснюється обчислення висоти та ширини рамки, яка буде навколо об’єкта на зображенні. Записується назва вирізаного елемента з зображення за допомогою виклику функції для вирізання елемента із зображення. Створюється масив елементів, де масив є комбінацією RGB, а сам масив заповнюється із викликанням функції для визначення домінантного кольору вирізаного елемента. Створюється структура даних Map, яка має ключ типу String та значення типу “Масив Int”, де ключ – це назва найближчого кольору з

назвою, а масив –комбінація трьох значень, які представляють колір у RGB. Ця структура заповнюється із викликанням функції пошуку найближчого кольору з назвою. Здійснюється виклик функції для рисування рамок навколо елементів одягу, в цю функцію передаються об'єкт класу Graphics2D, а також лічильник, масив лівої верхньої точки, довжина та ширина рамки. Збільшується значення лічильника, який відповідає за колір рамки. Створюється об'єкт елемента одягу, який містить назву елемента, точність, домінуючий колір (комбінація RGB), назву найближчого кольору з назвою та найближчий колір з назвою (комбінація RGB). Створений об'єкт додається у список.

Генерується унікальне ім'я для зображення з рамками навколо об'єктів. Це зображення записується у відповідну директорію. У результаті отримано об'єкт зображення, який містить зображення з рамками та список елементів на зображенні з їх властивостями.

*Побудова моделі на основі ШНМ.* Інформація про структуру мережі:

- один етап детектора об'єктів – застосований алгоритм використовується для виявлення об'єктів на зображеннях за один раз без необхідності додаткового аналізу;
- FPN (Feature Pyramid Network) – метод оброблення зображень, який допомагає покращити точність виявлення об'єктів, особливо коли об'єкти різних розмірів. У цьому випадку FPN використовується для покращення роботи одноетапного детектора. Основна ідея FPN полягає у тому, щоб побудувати ієрархію піраміди ознак з різними масштабами, яка містить інформацію про об'єкти на різних рівнях деталізації [11, 18]. Архітектуру FPN зображено на рис. 2;

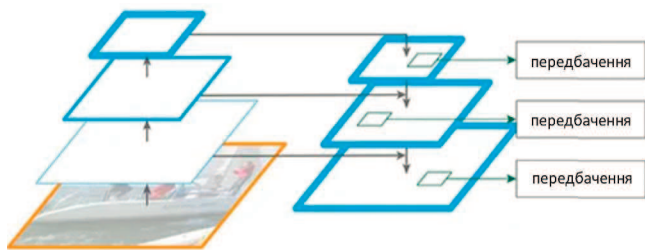


Рис. 2. Архітектура FPN / FPN Architecture

- InceptionV2 – модель глибокого навчання, яка використовується як базовий блок для аналізу зображень та визначення ознак об'єктів на зображеннях. Основні особливості InceptionV2: глибока архітектура з модулями Inception для виявлення ознак на різних масштабах, використання блоків ResNet для покращення навчання глибоких мереж, а також використання операцій Batch Normalization і ReLU для підвищення швидкості й точності навчання [2, 4];
- зображення змінюються до  $\text{min\_side}=512$  – перед тим, як зображення аналізують, їх розмір зменшують так, щоб мінімальна сторона дорівнювала 512 пікселів;
- масштаби шарів передбачення варіюються від  $1/8$  до  $1/64$  – алгоритм використовує кілька

шарів для передбачення об'єктів різних розмірів, причому це шари різних масштабів. Розміри об'єкта на зображенні можуть бути різними і кожен шар спеціалізується на передбаченні об'єктів на конкретному масштабі – це дає змогу підвищити точність;

- ШНМ тренувалась за допомогою SGD з використанням Online Hard Negative Mining (OHNM) – під час навчання було застосовано стохастичний градієнтний спуск із технікою виділення важких прикладів (негативів), що допомогло моделі швидше натренуватись. OHNM дає змогу покращити навчання моделі, зменшуючи кількість легко класифікованих негативних зразків (false negatives) і зосереджуючи навчання на найскладніших негативних зразках (hard negatives) [7, 14].

Отже, застосовано InceptionV2 як базову модель з оброблення зображень, використано техніку FPN для аналізу зображень із різними масштабами, зі зміною розміру зображень до 512 пікселів, і здійснено навчання за допомогою стохастичного градієнтного спуску із використанням техніки виділення важких прикладів (негативів).

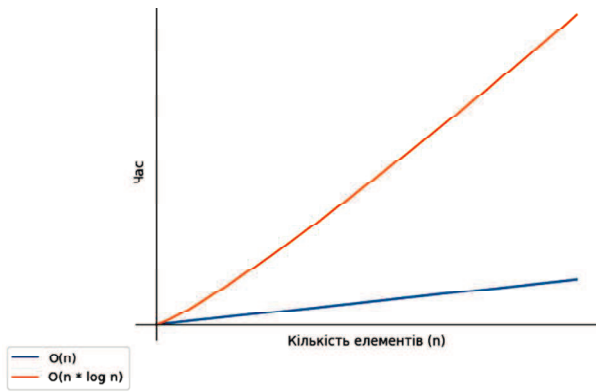
*Точність моделі, складність та час виконання алгоритмів.* Точності моделі досягнуто завдяки тестуванню на вибірці із 50 зображень, які чітко відповідають вимогам щодо вмісту зображень (вимоги описано у розділі “Визначення об'єктів на зображенні та їх класифікація”). Внаслідок цього об'єкти на 49 зображеннях розпізнано правильно, на зображенні 50 була часткова помилка. Відповідно, за ідеальних умов (вміст зображення відповідає вимогам), точність становить майже 98 %. Проте з погіршенням вмісту зображення, на противагу вимогам, точність розпізнавання стрімко знижується або ж розпізнавання не відбувається.

Для розпізнавання елементів і їх розташування час залежить від розміру зображення, яке буде надсилатись для розпізнавання. Сама ж модель розпізнає елементи на зображенні за менший проміжок часу, ніж одна секунда. Наприклад, зображення розміром 16 мегабайт з швидкістю інтернету 50 мб/с надсилатиметься 2,56 с.

Щодо алгоритму зрізу по медіані для визначення домінуючого кольору, то тут визначається середня часова складність алгоритму як  $O(n * \log n)$ , тобто логарифмічна. Логарифмічна складність алгоритму ґрунтується на використанні алгоритмів сортування “швидке сортування”, де у найкращому випадку часова складність  $O(n \times \log n)$  [16], та сортування структури даних “купа”, де середня часова складність сортування  $O(n \times \log n)$  [15]. Середній час отримання домінуючого кольору в поточній реалізації – 0,712 мілісекунд.

Якщо ж вести мову про алгоритм пошуку найближчого кольору з назвою, то його часова складність оцінюється як  $O(n)$ , тобто лінійна, адже здійснюється перебір всіх кольорів з назвами для знаходження кольору, найближчого за назвою до домінуючого. На практиці процес пошуку відбувається без затримки у часі, якщо говорити про мілісекунди [1].

Рис. 3 містить графічне відображення часової складності  $O(n \times \log n)$  для алгоритму визначення домінуючого кольору та часової складності  $O(n)$  для алгоритму пошуку найближчого кольору з назвою.



**Рис. 3.** Графіки часової складності алгоритмів  $O(n)$  та  $O(n * \log n)$  / Time Complexity Charts of Algorithms  $O(n)$  and  $O(n * \log n)$

*Візуальна демонстрація.* Для перевірки функціонування системи надіслано зображення для розпізнавання на ньому елементів одягу та їх кольорів. Результати опрацювання зображення подано на рис. 4.

На рис. 4 у блоці “Recognized picture” можна побачити рамки навколо елементів одягу. Розпізнавання здійснено успішно, розпізнані всі необхідні об’єкти, адже на програмному рівні було задано, що потрібно розпізнати такі елементи одягу, як взуття, брюки та верхній одяг (піджак). Відтак здійснено перехід до блока “Your combination”, в якому перший рядок є

заголовком таблиці, а всі наступні – записами, кожен з яких відповідає розпізаному елементу одягу. Розглянемо перші два записи, які відповідають за розпізане взуття: у стовпці “Accuracy” можемо побачити, що зі 100 % точністю розпізані об’єкти є взуттям; стовпці “Recognized color RGB” та “Recognized color VIEW” містять доміантний колір розпізаного елемента одягу і подають інформацію у візуальному форматі сприйняття кольору та у числовому форматі, який відповідає колірній моделі RGB; наступні стовпці “Closest color to real RGB”, “Closest color to real Name” та “Closest color to real VIEW” демонструють інформацію про найближчий колір до доміантного, але той, який має назву (згідно з таблицею основних кольорів у колірній моделі RGB, де близько 140 кольорів з назвами), відповідно перший стовпець містить числове відображення у колірній моделі RGB, другий – текстове представлення назви кольору, а третій – візуальне відображення найближчого кольору. Щодо туфель варто відзначити, що після розпізання було представлено кольори, близькі до тих, які є на зображенні, проте варто враховувати зони, в яких охоплено задній фон, а також тінь та положення туфель, що не опрацьовується розробленим підходом. Розпізнати кольори верхнього одягу (піджака) та брюк вдалось краще – як доміантного кольору, так і найближчого до нього кольору з назвою.



**Рис. 4.** Результат розпізнавання елементів одягу та їх кольорів / Recognition results of clothing items and their colors

*Основні колірні комбінації.* Після оброблення зображення користувач матиме доступ до таблиці поєднань кольорів чоловічих костюмів із туфлями. Подальше розроб-

лення передбачає автоматичне відображення правильності колірної комбінації користувача, проте на цьому етапі користувач самостійно ознайомлюється з таблицею.

Табл. Основні колірні комбінації / Basic color combinations

	Чорний костюм	Сірий костюм	Синій костюм	Коричневий костюм	Білий костюм
Чорні туфлі	+	+	+	–	–
Коричневі туфлі	–	+	+	+	+
Бордові туфлі	–	+	+	+	–
Білі туфлі	–	–	–	–	+

**Обговорення результатів дослідження.** Оцінивши результат основної функції, варто відзначити, що точність розпізнавання елементів одягу на користувачькому зображенні становить 98 % за умови, що вміст зображення відповідає описаним вимогам моделі. У визначенні домінантного кольору зони з елементом одягу досягнуто 100 % точності. Точність підбирання найближчого реального кольору на основі домінантного кольору – 60 %, адже не враховується область зображення, яка не є елементом одягу. Після виконання дій користувач отримує опрацьоване зображення з рамками навколо розпізнаних елементів одягу, а також таблицю із переліком розпізнаних елементів одягу, їхніх домінантних кольорів та найближчого реального кольору. Далі користувачу надається джерело для перевірки комбінації кольорів – це таблиця основних колірних комбінацій згідно з етикетом.

Отже, за результатами досліджень можна сформулювати наукову новизну та практичну цінність роботи.

**Наукова новизна отриманих результатів дослідження:** вперше розроблено комплексну систему розпізнавання елементів одягу та їх кольорів з використанням ШНМ та реалізовано підхід пошуку найближчого кольору з назвою, де під комплексністю системи розуміємо повний цикл оброблення зображення в одній системі – визначення розташування елементів одягу, класифікацію цих елементів, визначення домінантного кольору елементів одягу і найближчого кольору з назвою для подальшого візуального відображення усієї інформації користувачу та надання основних колірних комбінацій для виправлення власних кольорів, чи для того, щоб упевнитись у правильності колірної комбінації.

**Практична значущість результатів дослідження** полягає у розробленні структури та алгоритмів роботи застосунку, програмній реалізації системи, яка змінює підхід до вибору та перевірки колірних комбінацій. Такий підхід дає користувачу чіткі назви кольорів, згідно з якими можна створювати власні, правильні за комбінаціями кольорів набори одягу або якісно здійснювати пошук елементів одягу в мережі за отриманими кольорами. Також систему можна використати як засіб для розмітки “даних”, або, інакше кажучи, для підготовки матеріалу, який буде використовуватись для тренування моделей машинного навчання чи нейронних мереж.

## Висновки / Conclusions

Розроблено структуру системи розпізнавання елементів одягу та їх кольорів.

Для визначення об'єктів на зображенні та їх класифікації застосовано ШНМ платформи Clarifai.

Для визначення домінантного кольору використано алгоритм зрізу по медіані.

Розроблено підхід для пошуку найближчого кольору з назвою.

Розроблено комплексний застосунок для розпізнавання елементів одягу та їх кольорів, який використовує мову Java, наведено результати функціонування застосунку, проаналізовано точність моделі, складність та час виконання алгоритмів.

Систему можна розширювати такими способами:

- застосувати моделі машинного навчання чи нейронні мережі для точнішого визначення найближчого реального кольору (кольору з назвою);
- ввести зображення сторінки з демонстрацією/рекомендацією правильно підібраних кольорів;
- описувати механізм, який би дав змогу визначити патерни на одязі (стрічка, клітинка, рисунок);
- організувати базу даних із колірними комбінаціями одягу знаменитостей, завдяки якій можна буде демонструвати схожість або надавати рекомендації.

## References

- [1] Ahmad, I. (2020). 40 Algorithms Every Programmer Should Know: Hone your problem-solving skills by learning different algorithms and their implementation in Python (Illustrated ed.). Packt Publishing, 382 p.
- [2] Alamsyah, A., Saputra, M. A. A., & Masrury, R. A. (2019). Object Detection Using Convolutional Neural Network To Identify Popular Fashion Product. *Journal of Physics: Conference Series*, 1192, 012040. <https://doi.org/10.1088/1742-6596/1192/1/012040>
- [3] Albon, C. (2018). Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning (1st ed.). O'Reilly Media, 364 p. ISBN-10: 9781491989388, ISBN-13: 978-1491989388.
- [4] Brital, A. (2021). Inception V2 CNN Architecture Explained. Retrieved from: <https://medium.com/@AnasBrital98/inception-v2-cnn-architecture-explained-128464f742ce>
- [5] Brownlee, J. (2020 a). Machine Learning Mastery. Deep Learning CNN for Fashion-MNIST Clothing Classification. Retrieved from: <https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-fashion-mnist-clothing-classification>
- [6] Brownlee, J. (2020 b). Machine Learning Mastery. How to Train an Object Detection Model with Keras. Retrieved from: <https://machinelearningmastery.com/how-to-train-an-object-detection-model-with-keras>
- [7] Chu, J., Guo, Z., & Leng, L. (2018). Object Detection Based on Multi-Layer Convolution Feature Fusion and Online Hard Example Mining. *IEEE Access*, 6, 19959–19967. <https://doi.org/10.1109/ACCESS.2018.2815149>
- [8] Clarifai (2013). A leading platform for building AI-powered software solutions. Retrieved from: <https://www.clarifai.com/company/about>
- [9] Clarifai documentation (2013). Retrieved from: <https://docs.clarifai.com/api-guide/api-overview>
- [10] Dhakar, L. (2011). Color Thief. Retrieved from: <https://lokeshdhakar.com/projects/color-thief>

- [11] Hui, J. (2018). Understanding Feature Pyramid Networks for Object Detection (FPN). Retrieved from: <https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c>
- [12] RapidTables (2023). RGB Color Codes Chart. Retrieved from [https://www.rapidtables.com/web/color/RGB\\_Color.html](https://www.rapidtables.com/web/color/RGB_Color.html).
- [13] Schildt, H. (2021). Java: The Complete Reference, Twelfth Edition (12th ed.). McGraw Hill. ISBN-10: 1260463419, ISBN-13: 978-1260463415.
- [14] Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training Region-based Object Detectors with Online Hard Example Mining. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1, 761–69. <https://doi.org/10.48550/arXiv.1604.03540>
- [15] Sourabh, S. K., & Chakraborty, S. (2009). Empirical Study on the Robustness of Average Complexity & Parameterized Complexity Measure for Heapsort Algorithm. *International Journal of Computational Cognition*, 7(4), 1.
- [16] Taiwo, O. E., Christianah, A. O., Oluwatobi, A. N., & Aderonke, K. A. (2020). Comparative study of two divide and conquer sorting algorithms: quicksort and mergesort. *Procedia Computer Science*, 171, 2532–2540.
- [17] Thakkar, S. (2018). Dominant colors in an image using k-means clustering. Retrieved from: <https://medium.com/buzzrobot/dominant-colors-in-an-image-using-k-means-clustering-3c7af4622036>
- [18] Tsang, S. (2019). FPN – Feature Pyramid Network (Object Detection). Retrieved from: <https://towardsdatascience.com/review-fpn-feature-pyramid-network-object-detection-262fc7482610>
- [19] Woltmann, S. (2014). Color Thief by Lokesh Dhakar – A Fast Java Implementation. Retrieved from: <https://github.com/SvenWoltmann/color-thief-java>

**V. M. Teslyuk, S. S. Ivasiv**

*Lviv Polytechnic National University, Lviv, Ukraine*

## SYSTEM FOR RECOGNIZING CLOTHING ITEMS AND THEIR COLORS IN AN IMAGE

The paper introduces a system for recognizing clothing items and their colors, developed using the Java programming language in the form of a web application, employing the Model-View-Controller (MVC) architecture. For development and project assembly convenience, the Gradle automatic build system was utilized. The system operates based on artificial neural network principles on the Clarifai platform, thoroughly examining the process of object detection in images and the system's real-world performance. InceptionV2 serves as the base model for image processing, incorporating the FPN technique to analyze images at different scales, resizing images up to 512 pixels, and training using stochastic gradient descent with hard negative mining. The system employs a median cut algorithm to determine the dominant color of clothing items. Determining the dominant color of each clothing item is done separately, with a focus on achieving 100 % recognition accuracy. An approach to determining the nearest color name based on the dominant color is implemented. This approach is based on calculating the Euclidean distance between two points in a three-dimensional space, iterating through 140 colors with RGB color model names to find the nearest color name. However, there are certain limitations in finding the nearest color name that result in a reduced accuracy of 60 %. The impact of various factors, such as lighting and image quality, is thoroughly examined in the context of their influence on the system's performance. The user interface is designed as an intuitive tool for interacting with the system, allowing users to check recognized clothing items and identified colors. Additionally, recommendations on basic color combinations that users can employ to enhance their clothing style are provided. The work includes an example of the system's application on a real image, visually demonstrating the results and describing the quality of clothing item and color recognition. One of the key features of the presented system is its flexibility and scalability. Recognition quality can be further improved through additional model training on larger datasets. The paper also addresses data processing and result analysis optimization. The novelty of the research results lies in the development of a comprehensive clothing item and color recognition system using artificial neural networks and the implementation of an approach to finding the nearest color with name. Completeness of the system refers to the entire image processing cycle within a single system, including clothing item location determination, clothing item classification, dominant color determination, and finding the nearest color with name for visual display to the user, along with providing basic color combinations for color correction or reassurance of color combinations' correctness. The practical significance of the research results lies in the development of the application's structure and algorithms, the software implementation of the system, which changes the approach to selecting and verifying color combinations. This approach provides users with clear color names, allowing them to create their own accurate color combinations for clothing sets or effectively search for clothing items online based on received colors. The system can also be used as a "data labeling" tool for preparing materials used in training machine learning models or neural networks. Future prospects include integration with other systems, expanding the image database to improve accuracy, and utilizing additional data sources to enhance the system's functionality.

**Keywords:** clarifai; dominant color; RGB; color combinations; machine learning model; Feature Pyramid Network (FPN).

### Інформація про авторів:

**Теслюк Василь Миколайович**, д-р техн. наук, професор, завідувач кафедри автоматизованих систем управління.

Email: vasyli.m.teslyuk@lpnu.ua; <https://orcid.org/0000-0002-5974-9310>

**Івасів Станіслав Степанович**, магістр кафедри автоматизованих систем управління.

Email: stanislav.ivasiv.mknus.2022@lpnu.ua; <https://orcid.org/0009-0007-3406-7376>

**Цитування за ДСТУ:** Теслюк В. М., Івасів С. С. Система для розпізнавання елементів одягу та їх кольорів на зображенні.

*Український журнал інформаційних технологій*. 2023. Т. 5, № 2. С. 25–32.

**Citation APA:** Teslyuk, V. M., & Ivasiv, S. S. (2023). System for recognizing clothing items and their colors in an image. *Ukrainian Journal of Information Technology*, 5(2), 25–32. <https://doi.org/10.23939/ujit2023.02.025>