

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет**  
**імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

**ДОПУЩЕНО ДО ЗАХИСТУ**  
Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.  
\_\_\_\_\_ Ю. П. Кондратенко  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

**СИСТЕМА РОЗПІЗНАВАННЯ ДОРОГОЦІННИХ  
КАМЕНІВ З ВИКОРИСТАННЯМ КОМП'ЮТЕРНОГО  
ЗОРУ**

Спеціальність 122 «Комп'ютерні науки»

**122 – КРМ – 601.21810102**

*Виконав студент 6-го курсу, групи 601*  
\_\_\_\_\_ *В. О. Балутін*  
«19» лютого 2024 р.

*Керівник: канд. техн. наук, доцент*  
\_\_\_\_\_ *Є. В. Сіденко*  
«19» лютого 2024 р.

**Миколаїв – 2024**

**Чорноморський національний університет ім. Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

Освітньо-кваліфікаційний рівень **магістр**

Галузь знань **12 «Інформаційні технології»**

*(шифр і назва)*

Спеціальність **122 «Комп'ютерні науки»**

*(шифр і назва)*

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.

\_\_\_\_\_ Ю. П. Кондратенко

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**З А В Д А Н Н Я**  
**на кваліфікаційну роботу магістра**

**Балутіну Володимирі Олександровичу**

1. Тема кваліфікаційної роботи магістра «Система розпізнавання дорогоцінних каменів з використанням комп'ютерного зору».

Керівник роботи Сіденко Євген Вікторович, доцент кафедри інтелектуальних інформаційних систем, канд. техн. наук, доцент

Затв. наказом Ректора ЧНУ ім. Петра Могили від «02» лютого 2024 р. № 21

2. Строк подання студентом роботи 16 лютого 2024 р.

3. Вхідні (початкові) дані до роботи: датасет зображень дорогоцінного каміння.

Очікуваний результат роботи: програмний застосунок, який використовує натреновані моделі для класифікації дорогоцінного каміння.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

– дослідження теоретичних засад створення рекомендаційних систем та здійснення аналізу існуючих систем розпізнавання дорогоцінного каміння;

– обґрунтування вибору інструментальних засобів розробки програмного забезпечення, яке використовує комп'ютерний зір для класифікації дорогоцінних каменів;

– розробка та здійснення програмної реалізації застосунку для класифікації дорогоцінних каменів.

5. Перелік графічного матеріалу: презентація, рисунки, таблиці.

6. Завдання до спеціальної частини: Аналіз виробничих факторів на робочому місці та техніка безпеки.

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	д-р біол. наук Л. І. Григор'єва	
Методична частина	канд. техн. наук, доцент Є. В. Сіденко	

Керівник роботи канд. техн. наук, доцент Є. В. Сіденко  
(наук. ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Завдання прийнято до виконання В. О. Балутін  
(прізвище та ініціали)

\_\_\_\_\_ (підпис)

Дата видачі завдання « 07 » листопада 2023 р.

## КАЛЕНДАРНИЙ ПЛАН

### виконання магістерської кваліфікаційної роботи

Тема: «Система класифікація дорогоцінних каменів з використанням комп'ютерного зору»

№	Найменування роботи	Початок	Закінчення	Примітки
1	Визначення керівника і теми КРМ. Подання заяви на затвердження теми КРМ	01.09.2023	10.10.2023	Виконано
2	Отримання завдання на виконання КРМ	11.10.2023	01.11.2023	Виконано
3	Складання календарного плану на період виконання КРМ	02.11.2023	10.11.2023	Виконано
4	Огляд літератури за темою дослідження	11.11.2023	26.11.2023	Виконано
5	Проходження передатестаційної практики, збір та аналіз матеріалів до КРМ	27.11.2023	23.12.2023	Виконано
6	Аналіз предметної області та розробка технічного завдання. Моделювання результатів	25.12.2023	12.01.2024	Виконано
7	Опис фахової частини КРМ, зокрема розробка системи розпізнавання дорогоцінних каменів за допомогою комп'ютерного зору.	13.01.2024	25.01.2024	Виконано
8	Розробка спеціальної частини з охорони праці та методичної частини	26.01.2024	02.02.2024	Виконано
9	Перший попередній захист КРМ на засіданні комісії кафедри	29.01.2024	29.01.2024	Виконано
10	Корегування роботи за результатами попереднього захисту	30.01.2024	05.02.2024	Виконано
11	Доробка та остаточне оформлення КРМ	06.02.2024	11.02.2024	Виконано
12	Другий попередній захист КРМ на засіданні комісії кафедри	12.02.2024	12.02.2024	
13	Подання КРМ, її електронної копії та інших документів (відгуку, рецензії) до захисту	19.02.2024	20.02.2024	
14	Захист КРМ перед екзаменаційною комісією (ЕК)	26.02.2024	27.02.2024	
№	Найменування роботи	Початок	Закінчення	Примітки

Розробив студент Балутін В.О.  
(прізвище та ініціали)

\_\_\_\_\_ (підпис)

Керівник роботи канд. техн. наук, доцент Є. В. Сіденко  
(наук. ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

« 12 » листопада 2023 р.

## АНОТАЦІЯ

до кваліфікаційної роботи магістра  
студента групи 601 ЧНУ ім. Петра Могили

**Балутіна Володимира Олександровича**

на тему: **“СИСТЕМА РОЗПІЗНАВАННЯ ДОРОГОЦІННИХ КАМЕНІВ З  
ВИКОРИСТАННЯМ КОМП'ЮТЕРНОГО ЗОРУ”**

**Актуальність** даного дослідження полягає в ефективності результатів застосування комп'ютерного зору з точки зору точності і часу. На додаток цей підхід не вимагає підготовки зразків або знищення матеріалів, як це іноді роблять гомологи при ідентифікації дорогоцінних каменів, що робить інтеграцію CNN у цю сферу досить привабливою.

**Об'єктом** дослідження є процес розпізнавання дорогоцінних каменів.

**Предметом** дослідження є методи та алгоритми комп'ютерного зору для розпізнавання дорогоцінного каміння.

**Метою** дослідження є розпізнавання дорогоцінних каменів за допомогою сучасних технологій комп'ютерного зору.

В результаті виконання роботи було створено систему розпізнавання дорогоцінного каміння, натреновано та протестовано 3 моделі комп'ютерного зору. Також було виконано їх порівняння.

Дана робота складається з 5 розділів. Кожен розділ відповідно присвячений: аналізу предметної області, математичним моделям і методам, використаним у магістерській роботі, моделюванню і проектуванню системи планування і оптимізації маршрутів, аналізу отриманих результатів, охороні праці, методичній частині магістерської роботи. Загальний обсяг роботи – 97 сторінок. Магістерська кваліфікаційна робота містить один додаток, 42 рисунків, 1 таблицю і посилання на 45 літературних джерел.

**Ключові слова:** дорогоцінні камені, комп'ютерний зір, згорткові нейронні мережі.

## **ABSTRACT**

to the master's qualification work by the student of the group 601 of Petro Mohyla  
Black Sea National University

**Balutin Volodymyr**

### **“SYSTEM OF PRECIOUS STONE RECOGNITION USING COMPUTER VISION”**

A relevance of this research lies in the results of the computer vision approach in terms of accuracy and time. Additionally, this approach does not require sample preparation or material destruction, as gemologists sometimes do when identifying precious stones, making the integration of CNN into this field quite attractive.

An object of the study is the process of recognizing precious stones.

A subject of the study is the methods and algorithms of computer vision for the recognition of precious stones.

The aim of the research is to recognize precious stones using modern computer vision technologies. As a result of the work, a system for recognizing precious stones was created, and 3 computer vision models were trained and tested. Their comparison was also performed.

This work consists of six sections. Each section is devoted to: analysis of the subject area, mathematical models and methods used in the master's thesis, modeling and designing and optimization system of routes, analysis of the obtained results, labor protection, methodological part of the master's thesis.

The total volume of work is 97 pages. The master's qualification work 42 figures, 1 table, and references to 45 literary sources.

Keywords: gems, computer vision, convolutional neural networks

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ.....	7
1.1 Опис предметної сфери .....	7
1.2 Огляд та аналіз наявних аналогів та публікацій .....	15
Висновок до розділу 1 .....	17
2 МОДЕЛІ ТА МЕТОДИ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ .....	18
2.1 Методи для вирішення задачі .....	18
2.2 Технології розробки системи.....	37
Висновок до розділу 2 .....	43
3 ОПИС ВХІДНИХ ДАНИХ, СТВОРЕННЯ СИСТЕМИ ТА ТРЕНУВАННЯ МОДЕЛЕЙ .....	44
3.1 Опис вхідних даних .....	44
3.2 Завантаження та аугментація датасету .....	44
3.3 Створення моделі на архітектурі VGG16 .....	51
Висновок до розділу 3 .....	58
ВИСНОВКИ.....	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	60

## ПЕРЕЛІК СКОРОЧЕНЬ

ВДТ	– Візуальний дисплейний термінал
ПЕОМ	– Персональні електронні обчислювальні машини
КПО	– Коефіцієнт природної освітленості
ПК	– Персональний комп'ютер
ЕОМ	– Електронна обчислювальна машина
ССБП	– Система стандартів безпеки праці
МБІ	– Морфологічний будівельний індекс
CNN	– Згоркова нейронна мережа;
FCN	– Повна згоркова мережа;
MFCNN	– Багатофункціональної згорткової Нейронної Мережі
GAF	– Глобальний модуль уваги



## ВСТУП

Дорогоцінні камені завжди викликали захоплення та захват. Вони не лише втілюють в собі природну красу, але і мають величезне історичне, культурне та економічне значення. З прогресом технологій та зростанням попиту на ці благородні камені, виникає необхідність удосконалення методів їхньої класифікації.

Ідентифікація дорогоцінних каменів в наш час і досі є трудомісткий процес, який потребує складних та дорогих інструментів, які не всі гомологи можуть собі дозволити. Шляхом уважного спостереження за дорогоцінним камінням під збільшенням, гомологи виявляють візуальні характеристики, такі як колір, прозорість, блиск, біре-бахрома, щоб полегшити відділення дорогоцінних каменів.

**Актуальність.** В останні роки, великої популярності здобувають згорткові нейронні мережі, тому багато галузей по всьому світу намагаються інтегрувати їх у свою роботу. На додаток до результатів підходу комп'ютерного зору з точки зору точності і часу, цей підхід не вимагає підготовки зразків або знищення матеріалів, як це іноді роблять гомологи при ідентифікації дорогоцінних каменів, що робить інтеграцію CNN у цю сферу досить привабливою.

**Об'єктом дослідження** є процес розпізнавання дорогоцінних каменів.

**Предметом дослідження** є методи та алгоритми комп'ютерного зору для розпізнавання дорогоцінного каміння.

**Мета дослідження** – розпізнавання дорогоцінних каменів за допомогою сучасних технологій комп'ютерного зору.

Досягнення поставленої мети обумовлює необхідність вирішення наступних **завдань:**

– провести оцінку та порівняння різних методів класифікації, зокрема тих, що базуються на комп'ютерному зорі та машинному навчанні. Визначити їхню точність та швидкість, а також можливість використання в різних умовах;

- розробка та вдосконалення алгоритмів: розробити нові алгоритми класифікації, які враховують специфічні властивості дорогоцінних каменів. Вдосконалити існуючі методи для досягнення вищої точності та надійності результатів;
- розробити та здійснити систему яка класифікує дорогоцінні камені за допомогою комп'ютерного зору.

**Методологічною основою** дослідження є інтегрований підхід, що поєднує геммологічні техніки з принципами комп'ютерного зору та методами машинного навчання. Основні аспекти методології визначаються через використання геммологічних знань для аналізу характеристик каменів та комп'ютерного зору для автоматизованого оброблення та аналізу візуальних ознак. Машинне навчання, зокрема використання нейронних мереж, стає ключовим елементом для автоматичної класифікації. Важливо також враховувати експертні оцінки та залучення фахівців для підтвердження результатів. Методи розробляються з урахуванням можливостей інтеграції у виробничі та торговельні процеси, з метою створення ефективного та надійного інструменту для класифікації дорогоцінних каменів.

**Практичне значення** отриманих результатів полягає в тому, що Практичне значення запропонованих методів класифікації дорогоцінних каменів полягає в покращенні точності, швидкості та ефективності геммологічного аналізу. Використання комп'ютерного зору та машинного навчання дозволяє автоматизувати процес визначення характеристик каменів, роблячи його менш часо та ресурсномістким.

Ці методи можуть знайти практичне застосування у промислових лабораторіях та ювелірних підприємствах, де потрібно швидко та надійно класифікувати різні типи дорогоцінних каменів. Автоматизація такого процесу може сприяти підвищенню продуктивності та зниженню ймовірності помилок, що важливо для забезпечення якості та конкурентоспроможності у сучасному ринковому середовищі.

Крім того, вдосконалені алгоритми та технології можуть знайти використання в галузі наукових досліджень та навчання, сприяючи подальшому розвитку гомології та внесенню нових знань у сферу обробки дорогоцінних матеріалів. Такий підхід відкриває перспективи для розвитку інноваційних методів та технологій у галузі дорогоцінного каміння.

**Структура магістерської роботи.** Відповідно до мети, завдань і предмета дослідження, магістерська робота містить основну, методичну та спеціальну частини. Основна частина магістерської роботи складається із вступу, трьох розділів, висновку, списку використаних джерел та \_\_ додатків. Загальний обсяг магістерської роботи – 97 сторінок, із них основного тексту основної частини – 60 сторінок, методичної частини – 10 сторінок, спеціальної – 17 сторінок. Кількість використаних джерел – 45.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ

### 1.1 Опис предметної сфери

Дорогоцінний камінь — це шматок мінерального кристала, який після огранювання або полірування використовується для виготовлення ювелірних виробів чи інших прикрас. Однак деякі гірські породи (такі як лазурит, опал і обсидіан) і іноді органічні матеріали, які не є мінералами (такі як бурштин, гагат і перли), також використовуються для ювелірних виробів і тому часто також вважаються дорогоцінним камінням. Більшість дорогоцінних каменів тверді, але деякі м'які мінерали використовуються в ювелірних виробках через їх блиск або інші фізичні властивості, які мають естетичну цінність. Рідкість і погана відомість є іншими характеристиками, які надають цінності дорогоцінним каменям.

Знайдена в усьому світі промисловість кольорових дорогоцінних каменів (це означає будь-що, крім діамантів) наразі оцінюється приблизно в 10–12 мільярдів доларів США.

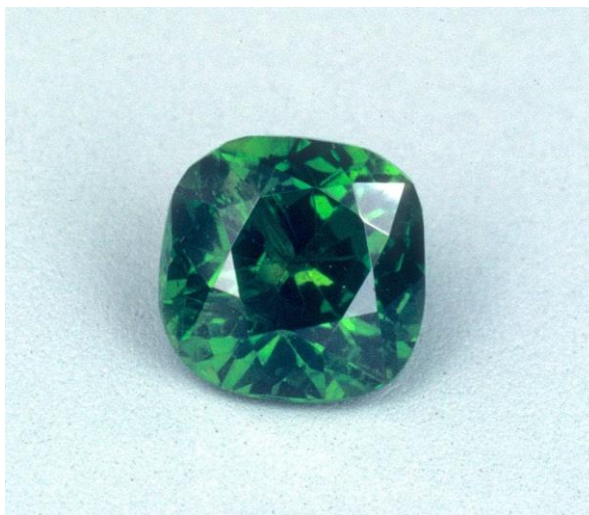


Рисунок 1.1 – Дорогоцінний камінь

Крім ювелірних виробів, з найдавніших часів головними видами мистецтва розкоші були вигравірувані дорогоцінні камені та різьблені вироби з твердого каменю, такі як кубки. Експерт з дорогоцінних каменів — гемолог, виробник

дорогоцінних каменів — лапідарист або огранщик; алмазна фреза називається діамантером.

Протягом століть люди використовували терміни "дорогоцінний" та "півдорогоцінний" для опису дорогоцінних каменів. До цього часу ви все ще час від часу можете почути ці терміни. Однак ця класифікація дорогоцінних каменів має так багато винятків, що вона не має жодної реальної цінності. Наприклад, діаманти традиційно вважалися дорогоцінними каменями, але деякі з них продаються за \$100 за карат. Ви можете побачити їх (з достатнім збільшенням) як акцентні камені на недорогих прикрасах. З іншого боку, гранатові камені традиційно вважалися півдорогоцінними каменями, але деякі продаються за більше ніж \$1,000 за карат, в десять разів дорожче, ніж низькоякісний діамант.

Також досить часто камені поділяють на дві інші категорії: діаманти та кольорові камені. Деякі дилери використовують скорочення "колір" для кольорових каменів. Усі камені, окрім діамантів, належать до категорії "кольорових каменів". Гемологи та дилери каменів використовують цю систему класифікації каменів з двох основних причин.

По-перше, для обробки діамантів потрібні спеціальні інструменти, оскільки ці камені твердіше за всі інші. З небагатьох винятків, ці інструменти не підходять для обробки кольорових каменів. Тому гемологам потрібні два різних набори інструментів для обробки обох типів каменів.

По-друге, діаманти та кольорові камені добуваються та розповсюджуються по-різному. Діаманти - один з небагатьох дорогоцінних каменів з постійним постачанням. Проте діамантова промисловість вводить загальну громадськість у віру, що діаманти неймовірно рідкісні. Насправді існують кольорові камені, які набагато рідше, ніж діаманти.

Чому це відбувається? Поблизу контролює продажі та маркетинг діамантів. Ті, хто відповідають за це, обережно не допускають залиття ринку. Таким чином, діаманти зберігають свою вартість. Тим часом відмінна реклама також переконала

громадськість у тому, що діаманти - це переважні дорогоцінні камені для обручальних кілець.

У цій системі класифікації каменів усі діаманти, незалежно від безбарвних або кольорових, все ще вважаються діамантами. Однак безбарвні діаманти та фантазійні кольорові діаманти - ті, що показують будь-який колір, крім світло-жовтого - мають різні системи оцінки.

Так само всі кольорові камені вважаються кольоровими каменями, навіть якщо екземпляр безбарвний. Наприклад, гемологи все ще класифікуватимуть безбарвний сапфір або топаз як кольоровий камінь. Однак, на відміну від діамантів, безбарвний екземпляр оцінюється так само, як і кольоровий екземпляр.



Рисунок 1.2 – Синтетичні камені

Природні камені, звичайно, формуються у природі. Синтетичні камені вирощуються в лабораторіях.

Природні та синтетичні матеріали можуть мати однакові властивості, але все ж мають значні відмінності. Основна різниця полягає в рідкості. Природний дорогоцінний камінь зазвичай формується мільйони років. Крім того, багато людей

вважають, що природні камені мають естетичні якості, які не зустрічаються в масово вироблених матеріалах. Ще одна різниця - це вартість. Оскільки природні дорогоцінні камені рідше і формуються довше, вони цінніші, ніж їх синтетичні аналоги. З цієї причини розрізнення між природними та синтетичними каменями є важливою навичкою для гемологів.

Камені, створені у лабораторіях, можуть бути поділені на додаткові підкатегорії.

Термін "синтетичний" відноситься до матеріалів, які відтворюють свої природні аналоги. Наприклад, синтетичні емальди, сапфіри та спінели мають фізичні та оптичні властивості природних варіантів. У багатьох випадках розрізнення між природними та синтетичними каменями є досить складним завданням. Матеріали, які не мають природних аналогів, називаються гомостворюваними. До цієї категорії належать лабораторно створені гарнети з гадолінієм галієм (GGG) та гарнети з ітрієм та алюмінієм (YAG). Протягом тривалого часу гемологи вважали кубічний цирконій (CZ) гомоствореним дорогоцінним каменем. Однак було виявлено дрібні кристали CZ, недостатньо великі для використання як дорогоцінні камені, в природі як мінерал бадделейїт. Це відкриття означає, що CZ є синтетичним каменем, а не гомоствореним.

Симулянт або імітаційний камінь - це будь-який матеріал, представлений як "схожий" на інший дорогоцінний камінь. Наприклад, природний білий топаз може бути проданий як імітація діаманта. З іншого боку, його також можуть продавати як справжній топаз. CZ, описаний як кубічний цирконій в рекламі ювелірних виробів, не є імітацією. У порівнянні, CZ, представлений як "схожий" на діамант, є імітацією.

Завжди є виняток або два. Отже, ось вони. Не всі мінерали є безбарвними у своєму чистому стані. Гранат - один з найочевидніших прикладів. Крім того, існує кілька видів та сортів гранатів. Усі гранати мають одну і ту ж структуру та багато схожостей у їх хімічному складі. Однак у них є відмінності в хімії. Кожна з цих відмінностей відповідає новому виду гранату.

Ось ілюстрація, яка не зовсім науково точна, але все ж допомагає пояснити, як різняться гранати. Подивіться на свою руку і уявіть, що це модель молекули гранату. Усі гранати мають одну й ту ж структуру, схожу на форму вашої руки, і майже той самий хімічний склад. Останні суглоби ваших пальців представляють собою окремі атоми. Хоча більшість атомів залишаються тими ж, різні атоми можуть знаходитися в цих суглобах. Якщо змінити атоми (хімію), то змінюється вид. Це правило. Однак можна побачити, що форма вашої руки не змінилася, а також жодна з інших основних характеристик. Тому різні види все ще є гранатами.



Рисунок 1.3 – Схема штучної нейронної мережі

Звичайні червоні гранати можуть бути альмандиновими або піроповими гранатами. Обидва ці види гранату глибокого червоного кольору. Однак у кожного з них трохи відмінний хімічний склад. Найчистіший альмандиновий гранат, що коли-небудь був знайдений, містить приблизно 20% піропу. Найчистіший піроп також містить приблизно 20% альмандину. Кожен з цих каменів також містить дрібну кількість інших видів гранату. Коли гемологи повинні назвати гранат, вони називають його за відсотковим співвідношенням основного компонента. Як ви можете побачити, це не завжди чітке розмежування. Подумайте: якщо найчистіший піроповий гранат, коли-небудь знайдений, містить лише 80% піропу, то є багато

2024 р. Балутін В. О. 122 – КРМ – 601.21810102



більше зразків, які ближчі до 50% піропу. Більшість гемологів просто описують більшість гранатів як суміш альмандину і піропу.

Деякі суміші гранатів мають відмінний набір характеристик. Наприклад, родоліт складається приблизно з 70% піропу і 30% альмандину. Те, що робить його виразним, - це його фіолетове забарвлення, оскільки його два основні компоненти червоні. Ця якість настільки виразна, що родоліт вважається варієтетом гранату.

Гранати ніколи не зустрічаються у чистому вигляді, а завжди у поєднаннях один з одним. Наприклад, більшість гранатів гемологічної якості належить до серії альмандин-піроп-шпесартит. Альмандин, піроп та шпесартит - це індивідуальні види гранату й завжди знаходяться разом. Вид гранату, який складає більшість, дає назву каменю. Такий тип суміші, де види певного каменю завжди знаходяться разом, називається серією у твердому стані.

Ще один підхід до класифікації дорогоцінних каменів полягає в розділенні їх на органічні та неорганічні. Органічні камені - це камені, у формуванні яких брали участь живі організми. Камінь янтар, наприклад, почався як сік дерев. Різні молюски створюють перлини. Тому ці матеріали вважаються органічними дорогоцінними каменями.

Багато продавців часто продають імітаційні дорогоцінні камені, зазвичай за набагато меншу ціну, ніж коштували б камені, які вони імітують. Доти, покупці усвідомлюють, що вони придбають "фальшиві перлини" або "фальшиві діаманти", наприклад, немає нічого неетичного в цьому. Однак продаж імітаційного каменю як справжнього - це неетично.

Використання термінів "дорогоцінний" для певного набору типів каменів та "півдорогоцінний" для всіх інших каменів може бути оманливим, оскільки це припускає, що дорогоцінні камені мають вроджену вищу цінність. Професійні гемологи більше не використовують ці терміни. Якщо ви зустрічаєте ці описи, будьте обережні, як покупець.

Традиційна класифікація на Заході, яка сходить до стародавніх греків, починається з розрізнення дорогоцінних і напівкоштовних предметів; подібні

відмінності проводяться в інших культурах. У сучасному користуванні дорогоцінним камінням є смарагд, рубін, сапфір і діамант, а всі інші дорогоцінні камені є напівкоштовними. Ця відмінність відображає рідкість відповідних каменів у стародавні часи, а також їхню якість: усі вони напівпрозорі з тонким кольором у своїй найчистішій формі, за винятком безбарвного алмазу, і дуже тверді, з твердістю від 8 до 10 за шкалою Мооса. . Інші камені класифікуються за кольором, напівпрозорістю та твердістю. Традиційне розрізнення не обов'язково відображає сучасні цінності; наприклад, хоча гранати відносно недорогі, зелений гранат під назвою цаворит може бути набагато ціннішим, ніж смарагд середньої якості. Ще один традиційний термін для напівкоштовних дорогоцінних каменів, який використовується в історії мистецтва та археології, - твердий камінь. Використання термінів «дорогоцінне» та «напівдорогоцінне» в комерційному контексті, мабуть, вводить в оману, оскільки передбачає, що певні камені є більш цінними, ніж інші, хоча це не відображається у фактичній ринковій вартості, хоча загалом це було б правильним якщо йдеться про бажаність.

У наш час дорогоцінні камені ідентифікують геммологи, які описують дорогоцінні камені та їхні характеристики, використовуючи спеціальну термінологію в галузі геммології. Перша характеристика, за якою геммолог ідентифікує дорогоцінний камінь, це його хімічний склад. Наприклад, алмази виготовлені з вуглецю (C), а рубіни з оксиду алюмінію (Al<sub>2</sub>O<sub>3</sub>)

Багато дорогоцінних каменів є кристалами, які класифікуються за їх кристалічною системою, такою як кубічна, тригональна або моноклінна. Інший термін, який використовується, — звичка, форма, у якій зазвичай зустрічається дорогоцінний камінь. Наприклад, алмази, які мають кубічну кристалічну систему, часто зустрічаються у вигляді октаєдрів.

Дорогоцінні камені поділяються на різні групи, види та різновиди. Наприклад, рубін є червоним різновидом різновиду корунду, тоді як будь-який інший колір корунду вважається сапфіром. Іншими прикладами є смарагд (зелений), аквамарин (блакитний), червоний берил (червоний), гошеніт

(безбарвний), геліодор (жовтий) і морганіт (рожевий), які є різновидами мінерального виду берилу.

Дорогоцінні камені характеризуються показником заломлення, дисперсністю, питомою вагою, твердістю, спайністю, руйнуванням і блиском. Вони можуть демонструвати плеохроїзм або подвійне заломлення світла. Вони можуть мати люмінесценцію та характерний спектр поглинання. Матеріал або дефекти в камені можуть бути присутніми у вигляді включень.

Дорогоцінні камені також можна класифікувати з точки зору їх «води». Це визнана оцінка блиску, прозорості або «блискучості» дорогоцінного каменю. Дуже прозорі дорогоцінні камені вважаються «першою водою», тоді як дорогоцінні камені «другої» або «третьої води» мають меншу прозорість.

Точна класифікація дорогоцінних каменів є критично важливою для галузі драгоцінного каміння та прикрас, оскільки ідентифікація є важливим першим етапом при оцінці і оцінці будь-якого дорогоцінного каменю. Наразі ідентифікація дорогоцінного каменю визначається за допомогою комбінації візуальних спостережень та спектрохімічного аналізу.

Шляхом уважного спостереження за дорогоцінними каменями неозброєним оком та при збільшенні, геммологи виявляють візуальні характеристики, такі як колір, прозорість, блиск, тріщини, клеймовини, включення, плеохроїзм, феномен та двоякопроменевість, для полегшення розділення дорогоцінних каменів .

Цей процес є складним, оскільки багато дорогоцінних каменів мають спільний колір та характеристики. Ідентифікація та класифікація часто супроводжуються використанням геммологічних інструментів, таких як рефрактометри, поляризатори та коноскопи , ручні спектроскопи , дихроскопи та ультрафіолетове світло для дослідження оптичних властивостей дорогоцінних каменів.

Вимірювання фізичних властивостей, таких як конкретна густина (також відома як відносна густина), надає додаткову інформацію щодо ідентифікації дорогоцінного каменю. З появою нових синтетичних дорогоцінних каменів та

технік обробки були введені в геммологічні лабораторії все більш складні інструменти з потужними спектроскопічними, флуоресцентними або хімічними аналітичними здібностями. До таких інструментів входять інфрачервоні спектрометри, Раман і люмінесцентні спектрометри, ультрафіолетово-видимі спектрометри, катодолюмінесценція, енергодисперсійні рентгенівські флуоресцентні спектрометри, лазерно-абляційні індуктивно-зв'язані плазмово-мас-спектрометри та флуоресцентні спектрометри.

## **1.2 Огляд та аналіз наявних аналогів та публікацій**

У роботі Самалі Амарасеката та Равінди Мегама було запропоновано наступну підхід: методика використовує мікроскопічне зображення дорогоцінного каменю, отримане за допомогою геммологічного мікроскопа. Згорточна нейронна мережа (CNN) навчена класифікувати тип дорогоцінного каміння, тоді як такі характеристики, як тип, колірна палітра, форма та вага, використовуються для оцінки вартості. Ця робота створює систему, яка здатна класифікувати та оцінювати вартість чотирьох типів дорогоцінних каменів, а саме блакитного сапфіра, жовтого сапфіра, аметисту та котячого ока. Результати показують, що запропонована методика дозволила класифікувати дорогоцінні камені з найвищою точністю 87% для жовтих сапфірів і 77% для синіх сапфірів. Жовті сапфіри показали найвищу точність категоризації кольорів, що можна пояснити високою контрастністю доступних зображень. Таким чином, можна зробити висновок, що якість вихідного зображення важлива для правильного визначення точного кольору дорогоцінного каменю.

Щоб отримати приблизну оцінку дорогоцінного каменю, оцінювач вручну переглядає та аналізує кожен дорогоцінний камінь окремо та створює звіт із зазначенням вартості. Оскільки цей процес займає багато часу, запропонована методика дозволяє автоматизувати це завдання оцінки вартості за допомогою підходу машинного навчання. Хоча на Шрі-Ланці існує понад сімдесят різновидів дорогоцінних каменів, у промисловості зазвичай зустрічаються приблизно

двадцять типів каменів. Для експериментів, проведених у цій статті, розглядаються лише чотири типи дорогоцінних каменів, які широко доступні в місцевій промисловості. Якість оригінального зображення важлива для правильного визначення точного кольору дорогоцінного каменю. Розширення цього дослідження для ідентифікації та оцінки вартості інших типів дорогоцінних каменів можливе за допомогою великої кількості наборів даних із зображеннями гарної якості.

У роботі Йана Чоу та Карлоса Рейеса було представлено методологію на основі комп'ютерного зору для автоматичної класифікації на основі зображень 2042 навчальних зображень і 284 невидимих (тестових) зображень, розділених на 68 категорій дорогоцінних каменів. Ряд методів виділення ознак (33, включаючи колірні гістограми в просторі RGB, HSV і CIELAB, локальний бінарний шаблон, текстуру Хараліка та властивості матриці спільного появи рівня сірого) використовувався в поєднанні з різними алгоритмами машинного навчання (логістична регресія, лінійна регресія). Дискримінантний аналіз, K-найближчий сусід, Дерево рішень, Випадковий ліс, Наївний Байєс та опорна векторна машина). Також було досліджено класифікацію глибокого навчання за допомогою ResNet-18 і ResNet-50. Оптимальна комбінація була забезпечена алгоритмом Random Forest із гістограмою кольорів RGB із восьми бінами та функціями локального бінарного шаблону з точністю 69,4% на невидимих зображеннях; алгоритмам було потрібно 0,0165 с для обробки 284 тестових зображень. Ці результати порівнювалися з трьома експертами-геммологами з принаймні 5-річним досвідом ідентифікації дорогоцінних каменів, які отримали точність від 42,6% до 66,9% і витратили 42–175 хвилин на класифікацію тестових зображень. Як і очікувалося, людям-експертам знадобилося набагато більше часу, ніж алгоритмам комп'ютерного зору, які, крім того, забезпечили, хоч і незначну, вищу точність. Незважаючи на те, що ці експерименти включали відносно невелику кількість зображень, перевага комп'ютерного зору над людським узгоджується з тим, що було повідомлено в

інших областях дослідження, і це надихає для подальшого вивчення застосування в геммології та суміжних областях.

У 1988 Еммануелем Фритсчем та Джорджем Россманом була запропонована техніка маркування для бурштину, де для ідентифікації використовуються обробка зображень і машинне навчання. Частини бурштину ідентифікуються та позначаються одним із 30 класів кольорів або одним із 20 класів геометричної форми. Експериментальні результати показали, що цей метод ефективний для класифікації неорганічних форм у межах вибраної геометричної фігури сім, якщо існує велика неоднозначність між органічними формами. Представлено метод виявлення синтетичних дорогоцінних каменів із застосуванням обробки зображень, де дорогоцінний камінь освітлюється за допомогою лазерного променя для створення візерунка відбиття/заломлення на екрані. Потім знімається зображення шаблону відображення, щоб визначити множину білих плям. Крім того, координати цих білих плям використовуються для визначення того, синтетичний дорогоцінний камінь чи ні.

### **Висновки до розділу 1**

У результаті проведеного дослідження встановлено, що встановлено, що сьогодні у багатьох компаніях фіксується інтегрування штучного інтелекту за для підвищення продуктивності та скорочення штату. Це робить розробку системи більш актуальною, так як програм по розпізнаванню дорогоцінних каменів у відкритому доступі немає.

Для реалізації системи, будуть розглянуті методи машинного навчання та комп'ютерного зору. В якості мови програмування було обрано мову python через наявність великої кількості бібліотек.

## 2. МОДЕЛІ ТА МЕТОДИ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

### 2.1 Методи для вирішення задачі

Машинне навчання (МН) - машинне навчання, яке часто скорочується як ML, є підмножиною штучного інтелекту (ШІ), яка зосереджена на розробці комп'ютерних алгоритмів, які автоматично вдосконалюються завдяки досвіду та використанню даних. Простіше кажучи, машинне навчання дозволяє комп'ютерам навчатися на основі даних і приймати рішення чи прогнози, не будучи на це явно запрограмованим.

За своєю суттю, машинне навчання полягає в створенні та впровадженні алгоритмів, які полегшують ці рішення та прогнози. Ці алгоритми розроблено для покращення їх продуктивності з часом, стаючи точнішими та ефективнішими, оскільки вони обробляють більше даних.

У традиційному програмуванні комп'ютер дотримується набору заздалегідь визначених інструкцій для виконання завдання. Однак у машинному навчанні комп'ютеру надається набір прикладів (даних) і завдання для виконання, але комп'ютер повинен визначити, як виконати завдання на основі наданих прикладів.

Наприклад, якщо ми хочемо, щоб комп'ютер розпізнавав зображення кішок, ми не надаємо йому конкретних інструкцій щодо того, як виглядає кіт. Натомість ми надаємо йому тисячі зображень котів і дозволяємо алгоритму машинного навчання з'ясувати загальні шаблони та особливості, які визначають кота. З часом, коли алгоритм обробляє більше зображень, він стає кращим у розпізнаванні котів, навіть якщо йому пред'являються зображення, яких він ніколи раніше не бачив.

Ця здатність навчатися на основі даних і вдосконалюватися з часом робить машинне навчання неймовірно потужним і універсальним. Це рушійна сила багатьох технологічних досягнень, які ми бачимо сьогодні, від голосових помічників і систем рекомендацій до безпілотних автомобілів і прогнозової аналітики.

Машинне навчання включає в себе різні ключові компоненти. По-перше, це дані, які є основою для ефективного використання цієї технології. Дані можуть бути структурованими або неструктурованими і використовуються для виявлення патернів.

Другий компонент - моделі, які навчаються на тренувальних даних і застосовуються для прогнозування або класифікації нових даних. Третій компонент - алгоритми навчання, які визначають спосіб, яким модель навчається на основі даних і як вона здійснює передбачення. Четвертий - функції втрат і оптимізації, які визначають різницю між передбаченими і фактичними значеннями і спосіб, яким модель адаптується для мінімізації цієї різниці. Нарешті, тренування та тестування використовуються для оцінки ефективності моделі на нових даних.

Машинне навчання поділяється на кілька типів, включаючи навчання з учителем, де модель навчається на позначених даних; навчання без учителя, де модель працює з непозначеними даними для виявлення структур; і навчання з підкріпленням, де модель вчиться взаємодіяти з оточенням і отримує винагороди за свої дії.

Машинне навчання використовується в різних галузях, таких як розпізнавання обличчя, медична діагностика, фінанси, технічне обслуговування, автономні автомобілі, геммологія та багато інших.

У контексті класифікації дорогоцінного каміння, машинне навчання може бути використане для автоматичного розпізнавання та сортування каменів за їхніми характеристиками.

Під час навчання для навчання під наглядом системи піддаються впливу великої кількості даних із мітками, наприклад зображень рукописних цифр із примітками, що вказують, якому номеру вони відповідають. Якщо навести достатню кількість прикладів, система навчання під наглядом навчиться розпізнавати кластери пікселів і фігур, пов'язані з кожним числом, і зрештою зможе розпізнавати рукописні цифри, здатні достовірно розрізняти числа 9 і 4 або 6 і 8.



Однак для навчання цих систем зазвичай потрібні величезні обсяги позначених даних, причому деякі системи мають бути піддані мільйонам прикладів, щоб виконати завдання.

Як наслідок, набори даних, які використовуються для навчання цих систем, можуть бути величезними: Open Images Dataset Google містить близько дев'яти мільйонів зображень, його позначене відеорепозіторій YouTube-8M посилається на сім мільйонів позначених відео та ImageNet, одна з перших баз даних такого типу, що містить понад 14 мільйонів категоризованих зображень. Розмір навчальних наборів даних продовжує зростати, а Facebook оголосив, що зібрав 3,5 мільярда зображень, загальнодоступних в Instagram, використовуючи хештеги, прикріплені до кожного зображення як мітки. Використання одного мільярда цих фотографій для навчання системи розпізнавання зображень забезпечило рекордний рівень точності – 85,4% – за тестом ImageNet.

Трудомісткий процес маркування наборів даних, які використовуються під час навчання, часто виконується за допомогою краудворкінг-сервісів, таких як Amazon Mechanical Turk, який надає доступ до великого фонду недорогої робочої сили, розкиданої по всьому світу. Наприклад, ImageNet створювали протягом двох років майже 50 000 людей, в основному набраних через Amazon Mechanical Turk. Однак підхід Facebook до використання загальнодоступних даних для навчання систем може забезпечити альтернативний спосіб навчання систем за допомогою мільярдних наборів даних без накладних витрат на ручне маркування.

Щоб зрозуміти навчання з підкріпленням, можна подумати про те, як хтось може навчитися грати в комп'ютерну гру старої школи вперше, коли вони не знайомі з правилами або як контролювати гру. Хоча вони можуть бути зовсім новачками, зрештою, дивлячись на взаємозв'язок між кнопками, які вони натискають, тим, що відбувається на екрані та їхніми результатами в грі, їх продуктивність ставатиме все кращою.

Прикладом навчання з підкріпленням є мережа Deep Q від Google DeepMind, яка перемогла людей у багатьох старовинних відеоіграх. Система отримує пікселі з кожної гри та визначає різну інформацію про стан гри, наприклад відстань між об'єктами на екрані. Потім він розглядає, як стан гри та дії, які він виконує в грі, пов'язані з результатом, який він досягає.

Потім зібрані дані поділяються на більшу частину для навчання, скажімо, близько 70%, і меншу частку для оцінки, скажімо, що залишилися 30%. Ці оціночні дані дозволяють протестувати навчену модель, щоб побачити, наскільки добре вона працюватиме на даних реального світу.

Перед початком навчання зазвичай також буде етап підготовки даних, під час якого виконуватимуться такі процеси, як дедуплікація, нормалізація та виправлення помилок.

Наступним кроком буде вибір відповідної моделі машинного навчання з широкого асортименту доступних. Кожен має сильні та слабкі сторони залежно від типу даних, наприклад, деякі підходять для обробки зображень, деякі для тексту, а деякі для чисто числових даних.

Прогнози, зроблені за допомогою навчання під наглядом, поділяються на два основні типи: класифікація, коли модель позначає дані як попередньо визначені класи, наприклад ідентифікація електронних листів як спам або не спам, і регресія, коли модель передбачає певні постійні значення, наприклад ціни на житло. .

Протягом багатьох циклів гри система зрештою будує модель того, які дії максимізують рахунок за яких обставин, наприклад, у випадку відеогри Breakout, де потрібно перемістити весло, щоб перехопити м'яч.

Основні етапи використання машинного навчання для класифікації дорогоцінного каміння виглядають наступним чином:

- збір даних: збираємо великий обсяг даних про дорогоцінні камені, включаючи інформацію про колір, прозорість, фактуру, форму та інші важливі характеристики. Ці дані мають бути ретельно проаналізовані геммологами;

- підготовка даних: проведемо обробку та підготовку даних, включаючи їхню нормалізацію, видалення зайвої інформації та розподіл на навчальний та тестовий набори;
- вибір моделі: вибираємо модель машинного навчання, яка найкраще відповідає вашій задачі класифікації. Для цього можуть використовуватися такі моделі, як нейронні мережі, методи опорних векторів (SVM), рішучі дерева тощо;
- тренування моделі: використовуємо навчальний набір даних для тренування обраної моделі. Модель навчається розпізнавати патерни та залежності між характеристиками та класами каменів;
- тестування та оцінка: перевіримо ефективність моделі на тестовому наборі даних. Визначте точність, чутливість та специфічність моделі;
- впровадження: після успішного тренування та тестування моделі впроваджуйте її для автоматизованої класифікації нових дорогоцінних каменів на практиці.

Використання машинного навчання дозволяє ефективно враховувати велику кількість факторів та взаємозв'язків, які можуть бути складні для аналізу вручну. Застосування цієї технології може полегшити та прискорити процес класифікації дорогоцінних каменів, забезпечуючи високу точність і об'єктивність результатів.

Комп'ютерний зір – це галузь штучного інтелекту (ШІ), яка зосереджена на розробці та реалізації алгоритмів та моделей, які дозволяють комп'ютерам розуміти і інтерпретувати візуальну інформацію з зображень або відео. Головною метою комп'ютерного зору є забезпечення комп'ютерам можливості "бачити" і розуміти світ навколо них, аналізувати графічні дані та взаємодіяти з оточенням на основі візуальної інформації.

Людське бачення включає наші очі, але воно також включає все наше абстрактне розуміння концепцій і особистий досвід через мільйони взаємодій, які ми мали з зовнішнім світом. Донедавна комп'ютери мали дуже обмежені можливості самостійного мислення. Комп'ютерний зір — це нещодавня галузь

технології, яка зосереджується на відтворенні людського зору, щоб допомогти комп'ютерам ідентифікувати й обробляти речі так само, як це роблять люди.

Завдяки нещодавнім розробкам у таких сферах, як штучний інтелект і обчислювальні можливості, сфера комп'ютерного зору досягла значного прогресу, щоб стати більш проникливою в повсякденне життя. Очікується, що до 2030 року ринок комп'ютерного зору наблизиться до 41,11 мільярда доларів, а середній річний темп зростання (CAGR) становитиме 16,0% між 2020 і 2030 роками.

Якби вам сказали назвати певні речі, які ви знайдете в парку, ви випадково згадали б такі речі, як трава, лавка, дерева тощо. Це дуже просте завдання, яке будь-яка людина може виконати миттєво. Однак існує дуже складний процес, який відбувається в глибині нашої свідомості.

Людський зір включає наші очі, але також включає все наше абстрактне розуміння понять і особистий досвід через мільйони взаємодій, які ми мали із зовнішнім світом. Донедавна комп'ютери мали дуже обмежені можливості самостійного мислення. Комп'ютерний зір — це нещодавня галузь технології, яка зосереджується на відтворенні людського зору, щоб допомогти комп'ютерам ідентифікувати й обробляти речі так само, як це роблять люди.

Завдяки нещодавнім розробкам у таких сферах, як штучний інтелект і обчислювальні можливості, сфера комп'ютерного зору досягла значного прогресу, щоб стати більш проникливою в повсякденне життя. Очікується, що до 2030 року ринок комп'ютерного зору наблизиться до 41,11 мільярда доларів, а середній річний темп зростання (CAGR) становитиме 16,0% між 2020 і 2030 роками.

Комп'ютерне бачення в штучному інтелекті спрямоване на розробку автоматизованих систем, які можуть інтерпретувати візуальні дані (наприклад, фотографії чи кінофільми) так само, як це роблять люди. Ідея, що лежить в основі комп'ютерного зору, полягає в тому, щоб навчити комп'ютерам інтерпретувати та розуміти зображення на піксельній основі. Це основа поля комп'ютерного зору. Щодо технічної сторони речей, комп'ютери намагатимуться отримувати візуальні дані, керувати ними та аналізувати результати за допомогою складних програм.

Обсяг даних, який ми створюємо сьогодні, величезний – 2,5 квінтільйона байтів даних щодня. Таке зростання даних виявилось одним із рушійних факторів розвитку комп'ютерного зору.

Для комп'ютерного зору потрібні величезні обсяги інформації. Повторні аналізи даних виконуються, доки система не зможе розрізнити об'єкти та ідентифікувати візуальні елементи. Глибоке навчання, специфічний вид машинного навчання, і згорткові нейронні мережі, важлива форма нейронної мережі, є двома ключовими техніками, які використовуються для досягнення цієї мети.

За допомогою попередньо запрограмованих алгоритмічних структур система машинного навчання може автоматично дізнаватися про інтерпретацію візуальних даних. Модель може навчитися розрізнити схожі зображення, якщо їй надати достатньо великий набір даних. Алгоритми дозволяють системі навчатися самостійно, щоб вона могла замінити людську працю в таких завданнях, як розпізнавання зображень.

Згорткові нейронні мережі допомагають зрозуміти моделі машинного та глибокого навчання, розділяючи візуальні елементи на менші частини, які можна позначати тегами. За допомогою тегів він виконує згортки, а потім використовує третинну функцію для надання рекомендацій щодо сцени, яку спостерігає. З кожним циклом нейронна мережа виконує згортки та оцінює правдивість своїх рекомендацій. І саме тоді він починає сприймати та ідентифікувати картинку, як людина.

Комп'ютерний зір схожий на вирішення головоломки в реальному світі. Уявіть, що у вас є всі ці фрагменти мозаїки, і вам потрібно зібрати їх, щоб сформувати справжнє зображення. Саме так працюють нейронні мережі всередині комп'ютерного зору. Завдяки серії фільтрів і дій комп'ютери можуть об'єднати всі частини зображення, а потім думати самостійно. Однак комп'ютеру не просто дається головоломка із зображенням – скоріше, його часто годують тисячами зображень, які навчають його розпізнавати певні об'єкти.

Наприклад, замість того, щоб навчати комп'ютер шукати гострі вуха, довгі хвости, лапи та вуса, які складають kota, програмісти завантажують і передають у комп'ютер мільйони зображень котів. Це дозволяє комп'ютеру розуміти різні риси kota та миттєво розпізнавати його.

Основні аспекти комп'ютерного зору:

- розпізнавання об'єктів: метою є виявлення та класифікація об'єктів на зображеннях або відео. Це може включати розпізнавання облич, автомобілів, тварин, предметів меблів і т. д.;
- сегментація областей: розділення зображення на окремі області чи об'єкти для докладного аналізу. Наприклад, виділення контурів облич або різних частин зображення;
- відслідковування рухомих об'єктів: аналіз руху та відслідковування рухомих об'єктів на відео. Застосовується у сферах відеоспостереження та робототехніки;
- розпізнавання обличчя: виявлення та ідентифікація осіб на зображеннях чи відео. Використовується у системах безпеки, відеоспостереженні, аутентифікації тощо;
- розпізнавання рукописного тексту: оптичне розпізнавання символів та тексту на зображеннях, що дозволяє комп'ютерам аналізувати написаний текст.
- 3D-реконструкція: створення тривимірних моделей об'єктів або оточення на основі 2D-зображень;
- аналіз медичних зображень: використовується для діагностики за допомогою аналізу медичних зображень, таких як рентгени, знімки МРТ чи комп'ютерної томографії;
- аугментована реальність: застосування комп'ютерного зору для накладання віртуальних об'єктів на реальний світ;

Методи комп'ютерного зору включають в себе використання глибокого навчання (зокрема нейронних мереж), традиційні методи обробки зображень, використання фільтрів та дескрипторів, а також техніки геометричної обробки.

Комп'ютерний зір є ключовою технологією для численних застосувань у різних галузях, від промисловості та медицини до розваг та наукових досліджень.

У контексті класифікації дорогоцінного каміння, комп'ютерний зір використовується для автоматизованого розпізнавання візуальних ознак каменів на зображеннях.

Основні етапи використання комп'ютерного зору для класифікації дорогоцінного каміння можуть виглядати наступним чином:

- збір зображень: зберіть велику кількість зображень дорогоцінних каменів, що включають різноманітні варіації колірив, форм, текстур та інших важливих характеристик;

- попередня обробка зображень: Виконайте попередню обробку зображень, таку як зменшення розміру, корекція кольору, нормалізація, щоб підготувати їх для аналізу комп'ютерним зором.

- використання нейронних мереж: застосуйте методи машинного навчання, зокрема нейронні мережі, для навчання комп'ютера розпізнавати патерни на зображеннях, пов'язані з характеристиками каменів.

- тренування моделі: використовуйте навчальний набір зображень для тренування моделі комп'ютерного зору. Модель навчається розпізнавати та класифікувати камені з урахуванням їхніх візуальних ознак.

- тестування та оцінка: перевірте ефективність моделі на тестовому наборі зображень, визначаючи її точність та надійність в класифікації.

- впровадження: після успішного тренування та тестування впровадьте модель для автоматизованої класифікації нових зображень дорогоцінного каміння.

Використання комп'ютерного зору дозволяє аналізувати великі обсяги даних та вивчати візуальні патерни, що може бути важко або неможливо зробити вручну. Це спрощує та автоматизує процес класифікації дорогоцінного каміння, забезпечуючи об'єктивні та консистентні результати.

Згортова нейронна мережа (ConvNet/CNN) - це алгоритм глибокого навчання, який призначений для обробки вхідних зображень. Вона визначає

вагомість та значення зміщення для різних аспектів чи об'єктів на зображенні, щоб розрізнити їх між собою. Попередня обробка, необхідна для ConvNet, є набагато меншою порівняно з іншими алгоритмами класифікації. У відміню від простіших методів, де фільтри встановлюються вручну, ConvNets можуть самостійно навчати ці фільтри та характеристики під час навчання.

Архітектура ConvNet подібна до способу, як нейрони підключаються у людському мозку, і була натхненна організацією зорової кори. Кожен окремий нейрон реагує лише на обмежену область поля зору, відому як рецептивне поле. Набір таких полів перекривається, щоб охопити всю візуальну область.

Що таке штучні нейронні мережі? Нейронні мережі, також відомі як штучні нейронні мережі або імітовані нейронні мережі, є підмножиною машинного навчання та лежать в основі алгоритмів глибокого навчання. Їх назва та структура навіяні людським мозком, імітуючи те, як біологічні нейрони передають сигнали один одному.

Штучні нейронні мережі (ШНМ) складаються з шарів вузлів, що відповідають вхідному шару, одному або декільком прихованим шарам і вихідному шару. Кожен вузол або штучний нейрон з'єднаний з іншими і має відповідну вагу та поріг. Коли вихідний сигнал одного вузла досягає порогового значення, цей вузол активується шляхом перевищення даних на наступному рівні мережі. В іншому випадку дані не будуть передані на наступний рівень мережі.

Штучні нейронні мережі містять штучні нейрони, які називаються одиницями. Ці блоки розташовані в серії шарів, які разом утворюють всю штучну нейронну мережу в системі. Рівень може мати лише дюжину одиниць або мільйони одиниць, оскільки це залежить від того, наскільки складні нейронні мережі будуть потрібні для вивчення прихованих шаблонів у наборі даних. Як правило, штучна нейронна мережа має вхідний рівень, вихідний рівень, а також приховані шари. Вхідний рівень отримує дані із зовнішнього світу, які нейронна мережа повинна проаналізувати або дізнатися про них. Потім ці дані проходять через один або кілька прихованих шарів, які перетворюють вхідні дані в дані, які є цінними для



вихідного рівня. Нарешті, вихідний рівень забезпечує вихід у формі відповіді штучних нейронних мереж на надані вхідні дані.

У більшості нейронних мереж блоки пов'язані між собою від одного рівня до іншого. Кожне з цих зв'язків має ваги, які визначають вплив однієї одиниці на іншу. Коли дані передаються від одного блоку до іншого, нейронна мережа дізнається все більше і більше про дані, що в кінцевому підсумку призводить до виведення з вихідного рівня.

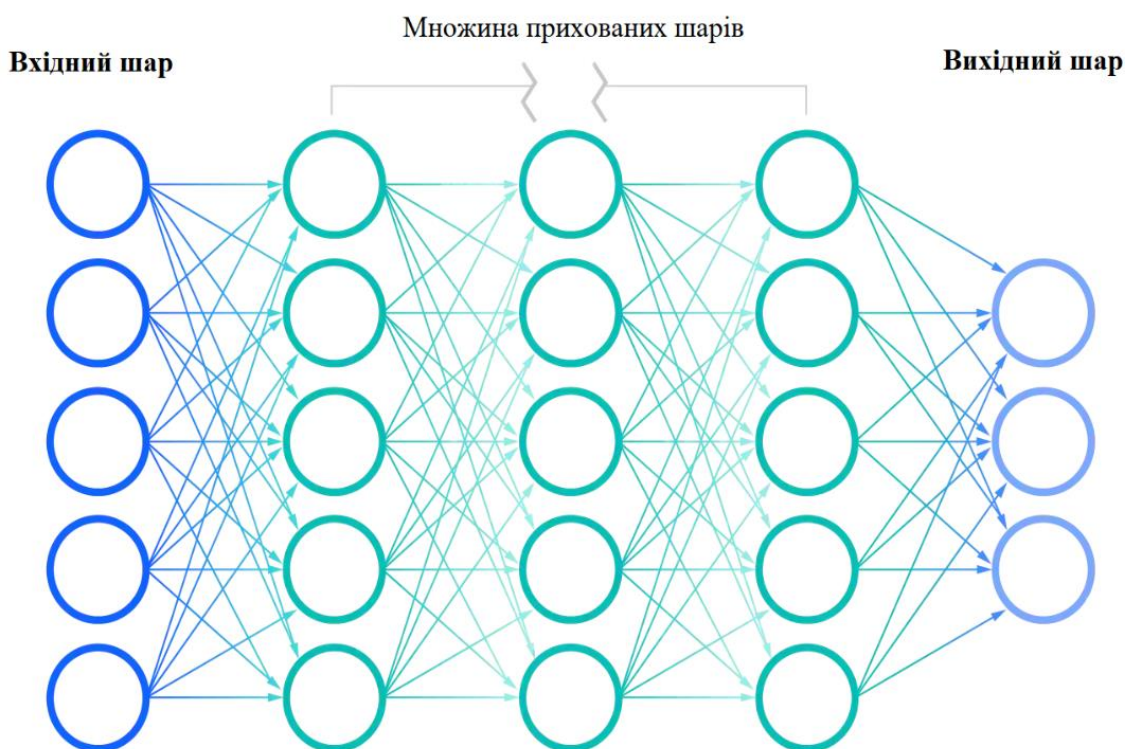


Рисунок 2.1 – Схема штучної нейронної мережі

Нейронні мережі покладаються на навчальні дані, щоб навчатися та підвищувати свою точність з часом. Однак, як тільки ці алгоритми навчання точно налаштовані, вони стають потужними інструментами в інформатиці та штучному інтелекті, забезпечуючи високу швидкість класифікації та кластеризації даних.

Завдання розпізнавання голосу чи зображення можуть тривати хвилини замість годин порівняно з ідентифікацією вручну експертами-людьми. Однією з найвідоміших нейронних мереж є пошуковий алгоритм Google.

Як працюють нейронні мережі? Подумайте про кожен окремий вузол як про власну модель лінійної регресії, що складається з вхідних даних, вагових коефіцієнтів, зміщення (або порогу) і вихідних даних. Формула виглядає приблизно так:

$$\sum_{i=1}^m w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

Рисунок 2.2 – Загальний вигляд формули вузла

$$\text{output} = f(x) = \begin{cases} 1 & \text{if } \sum w_1 x_1 + b \geq 0 \\ 0 & \text{if } \sum w_1 x_1 + b < 0 \end{cases}$$

Рисунок 2.3 – Загальний вигляд формули вихідного сигналу

Штучні нейронні мережі навчаються за допомогою навчального набору. Наприклад, припустімо, що ви хочете навчити ШНМ розпізнавати кота. Потім йому показують тисячі різних зображень кішок, щоб мережа навчилася ідентифікувати кота. Після того, як нейронна мережа достатньо навчена використовувати зображення котів, вам потрібно перевірити, чи може вона правильно ідентифікувати зображення котів. Це робиться шляхом того, що ШНМ класифікує надані зображення, вирішуючи, чи є вони зображеннями котів чи ні. Вихідні дані, отримані ШНМ, підтверджуються наданим людиною описом того, чи є зображення зображенням кота чи ні. Якщо ШНМ ідентифікує неправильно, тоді

використовується зворотне розповсюдження для коригування всього, що він дізнався під час навчання. Зворотне розповсюдження виконується шляхом точного налаштування ваг з'єднань в одиницях ШНМ на основі отриманої частоти помилок. Цей процес триває до тих пір, поки штучна нейронна мережа не зможе правильно розпізнати kota на зображенні з мінімальною можливістю помилок.

Які є типи штучних нейронних мереж? Нейронна мережа прямого зв'язку. Нейронна мережа прямого зв'язку є однією з найпростіших штучних нейронних мереж. У цій ШНМ надані дані або вхідні дані переміщуються в одному напрямку. Він входить до ШНМ через вхідний рівень і виходить через вихідний рівень, тоді як приховані шари можуть існувати або не існувати. Таким чином, нейронна мережа прямого зв'язку має хвилю, що поширюється лише попереду, і зазвичай не має зворотного поширення.

Згортка нейронної мережі. Згортка нейронної мережі має деякі подібності до прямої нейронної мережі, де зв'язки між одиницями мають ваги, які визначають вплив однієї одиниці на іншу. Але CNN має один або більше ніж один згортковий шар, який використовує операцію згортки на вході, а потім передає отриманий результат у формі виводу на наступний рівень. CNN має програми для обробки мови та зображень, що особливо корисно для комп'ютерного зору.

Модульна нейронна мережа: модульна нейронна мережа містить набір різних нейронних мереж, які працюють незалежно для отримання результату без взаємодії між собою. Кожна з різних нейронних мереж виконує різні підзавдання, отримуючи унікальні вхідні дані порівняно з іншими мережами. Перевага цієї модульної нейронної мережі полягає в тому, що вона розбиває великий і складний обчислювальний процес на менші компоненти, таким чином зменшуючи його складність, водночас одержуючи необхідний результат.

Радіальна базисна функція Нейронна мережа: радіальні базисні функції – це ті функції, які враховують відстань точки відносно центру. Функції RBF мають два рівні. На першому шарі вхідні дані зіставляються з усіма радіальними базисними функціями прихованого шару, а потім вихідний рівень обчислює вихідні дані на

наступному кроці. Мережі радіальних базисних функцій зазвичай використовуються для моделювання даних, які представляють будь-яку базову тенденцію або функцію.

Повторювана нейронна мережа: повторювана нейронна мережа зберігає вихідні дані шару та передає їх назад на вхід, щоб краще передбачити результат шару. Перший рівень у RNN дуже подібний до прямої нейронної мережі, а рекурентна нейронна мережа запускається після обчислення результату першого рівня. Після цього шару кожен блок запам'ятує деяку інформацію з попереднього кроку, щоб він міг діяти як комірка пам'яті під час виконання обчислень.

$$\text{Cost Function} = \text{MSE} = \frac{1}{2m} \sum_{i=1}^m (\hat{y} - y)^2$$

Рисунок 2.4 – Загальний вигляд формули MSE

Зрештою, мета полягає в тому, щоб мінімізувати нашу функцію витрат, щоб забезпечити правильність визначення для будь-якого даного спостереження.

Оскільки модель коригує ваші ваги та зміщення, вона використовує функцію вартості та навчання підкріплення, щоб досягти точки конвергенції або локального мінімуму. Процес, за допомогою якого алгоритм коригує ваші вагові коефіцієнти, є градієнтним спуском, який дозволяє змінювати напрямок моделі, у якій ви хочете зменшити помилки (або мінімізувати вартість функції). З кожним навчальним прикладом параметри моделі налаштовуються таким чином, щоб вони збільшувалися до мінімуму.

Рівень може містити лише десятки або мільйони одиниць, залежно від того, наскільки складними повинні бути нейронні мережі, щоб вивчати приховані шаблони в наборі даних. Штучна нейронна мережа зазвичай має вхідний рівень,

вихідний рівень і приховані шари. Вхідний рівень отримує дані із зовнішнього світу, які нейронна мережа повинна проаналізувати або вивчити. Ці дані потім передаються через один або кілька прихованих рівнів, які перетворюють вхідні дані в дані, які є цінними для вихідного рівня. Нарешті, вихідний рівень забезпечує вихід у формі відповіді штучної нейронної мережі на задані вхідні дані.

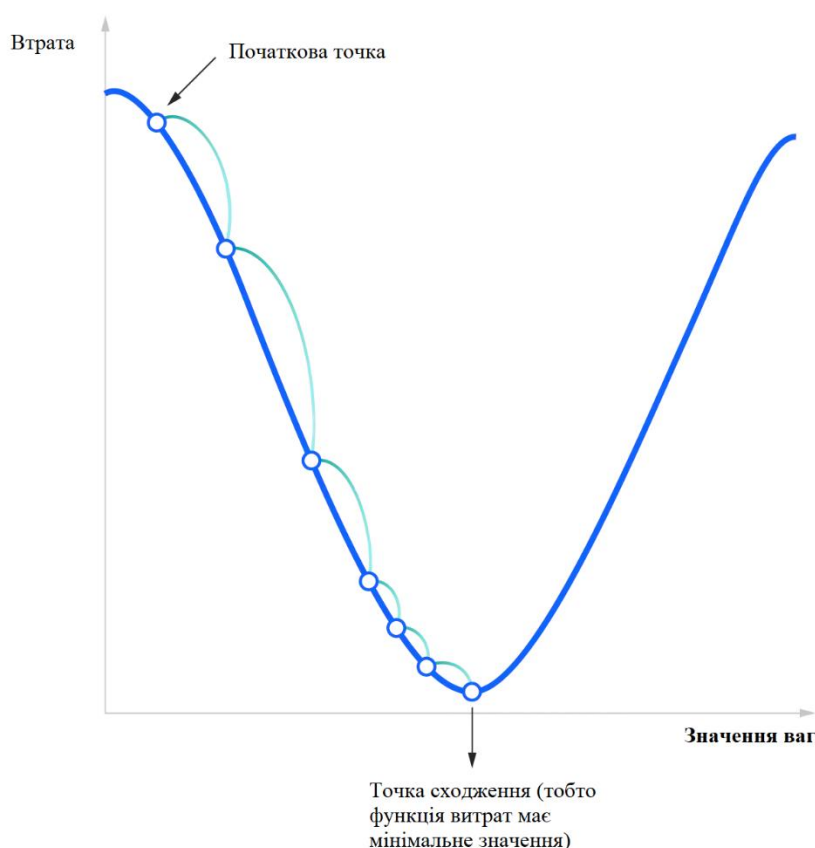


Рисунок 2.6 – Корегування помилки для тренування нейронної мережі

Більшість глибоких нейронних мереж мають прямий зв'язок, тобто вони працюють лише в одному напрямку, від входу до виходу. Однак ви можете навчити свою модель за допомогою зворотного поширення. Тобто рухатися в зворотному напрямку від виходу до входу. Зворотне розповсюдження дозволяє нам обчислити та відобразити помилку, пов'язану з кожним нейроном, дозволяючи нам відповідно коригувати та коригувати параметри моделі(й).

Типи нейронних мереж: нейронні мережі можна розділити на різні типи, які використовуються для різних цілей. Хоча це не вичерпний список типів, нижче наведено найпоширеніші типи нейронних мереж, з якими ви зіткнетеся у типових випадках використання:

Перцептрон — найстаріша нейронна мережа, створена Френком Розенблатом у 1958 році. Він складається з нейрона і є найпростішою формою нейронної мережі:

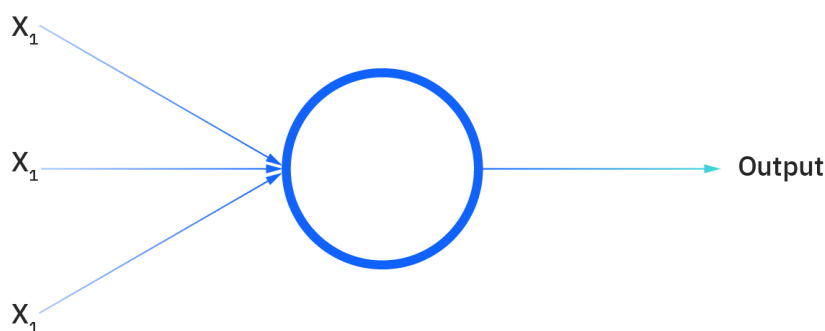


Рисунок 2.6 – Перцептрон

Нейронні мережі прямого зв'язку, або багатошарові перцептрони (MLP), є основною темою цього розділу. Вони складаються з вхідного шару, одного або кількох прихованих шарів і вихідного шару. Хоча ці нейронні мережі також зазвичай називають MLP, важливо відзначити, що насправді вони складаються з сигмовидних нейронів, а не з перцептронів, оскільки більшість завдань реального світу є нелінійними. Дані зазвичай вводяться в ці моделі для навчання, що є основою для комп'ютерного зору, обробки природної мови та інших нейронних мереж.

Згорткові нейронні мережі (CNN) схожі на мережі прямого зв'язку, але часто використовуються для розпізнавання зображень, розпізнавання образів і комп'ютерного зору. Ці мережі використовують принципи лінійної алгебри, особливо множення матриць і знаходження закономірностей у зображеннях.

Повторювані нейронні мережі (RNN) ідентифікуються за циклами зворотного зв'язку. Ці алгоритми навчання в основному використовуються, коли дані часових рядів використовуються для прогнозування майбутніх результатів, наприклад прогнозування ринку або прогнозування продажів.

Штучний інтелект пережив величезний ріст у подоланні розриву між людськими та машинними можливостями. Дослідники та любителі працюють над різними аспектами цієї галузі та досягають дивовижних результатів. Однією з багатьох областей є комп'ютерний зір.

Проблема в цій галузі полягає в тому, що машини бачать світ так само, як і люди, сприймають його подібним чином, а потім застосовують ці знання для багатьох завдань, таких як розпізнавання зображень і відео, аналіз і класифікація зображень і відтворення медіа. Його можна використовувати, системні рекомендації. Досягнення комп'ютерного зору завдяки глибокому навчанню були створені та вдосконалені з часом, головним чином на основі одного конкретного алгоритму: згорткових нейронних мереж.

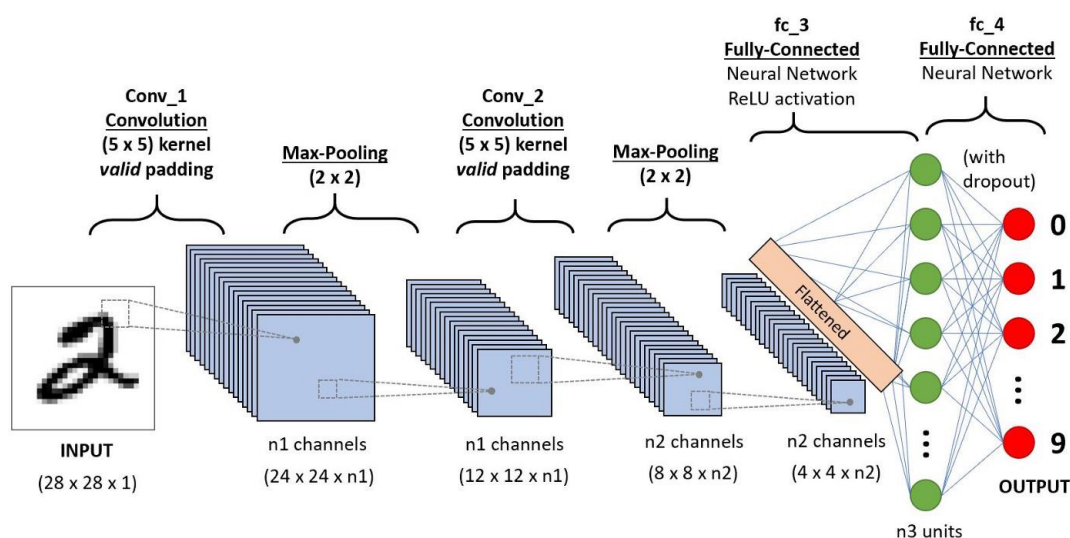


Рисунок 2.7 – Послідовність CNN для класифікації рукописних цифр

Згорткові нейронні мережі (ConvNet/CNN) — це алгоритми глибокого навчання, які можуть отримувати вхідне зображення, призначати важливість

(вивчені ваги та зміщення) різним аспектам/об'єктам зображення та відрізнити один від іншого. Попередня обробка, яку вимагає ConvNet, набагато нижча порівняно з іншими алгоритмами класифікації. У примітивних методах фільтри розробляються вручну, але після достатнього навчання ConvNet має можливість вивчати ці фільтри/характеристики.

Архітектура ConvNet нагадує схему з'єднання нейронів у людському мозку та натхненна організацією зорової кори. Окремі нейрони реагують на стимули лише в обмеженій області свого поля зору, відомої як рецептивне поле. Набір таких полів перекривається, щоб охопити всю візуальну область.

Зображення — це просто матриця значень пікселів, чи не так? Тож чому б не згладити зображення (наприклад, матрицю зображення  $3 \times 3$  або вектор  $9 \times 1$ ) і передати його в багатошаровий перцептрон для цілей класифікації?

Для дуже простих бінарних зображень цей метод може мати середню точність у прогнозуванні класів, але при роботі зі складними зображеннями, які повсюдно залежать від пікселів, він має низьку точність.

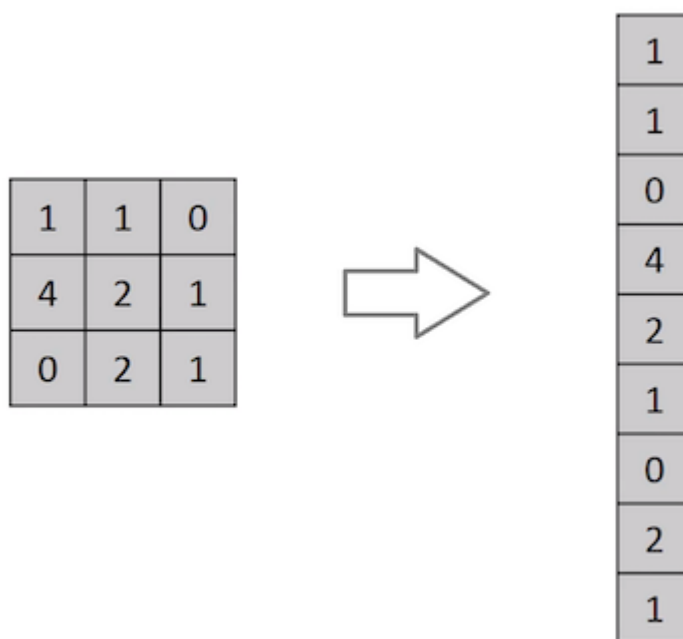


Рисунок 2.8 – Зведення матриці зображення  $3 \times 3$  у вектор  $9 \times 1$



CNN можуть успішно фіксувати просторові та часові залежності в зображеннях за допомогою відповідних фільтрів. Ця архітектура ідеальна для наборів даних зображень, оскільки вона зменшує кількість залучених параметрів і вагові коефіцієнти можна повторно використовувати. Іншими словами, мережі можна навчити краще розуміти складність зображень.

На зображенні нижче ми маємо зображення RGB, розділене на три кольорові площини: червону, зелену та синю. Існує багато таких кольорних просторів, у яких існують зображення, включаючи відтінки сірого, RGB, HSV і CMYK.

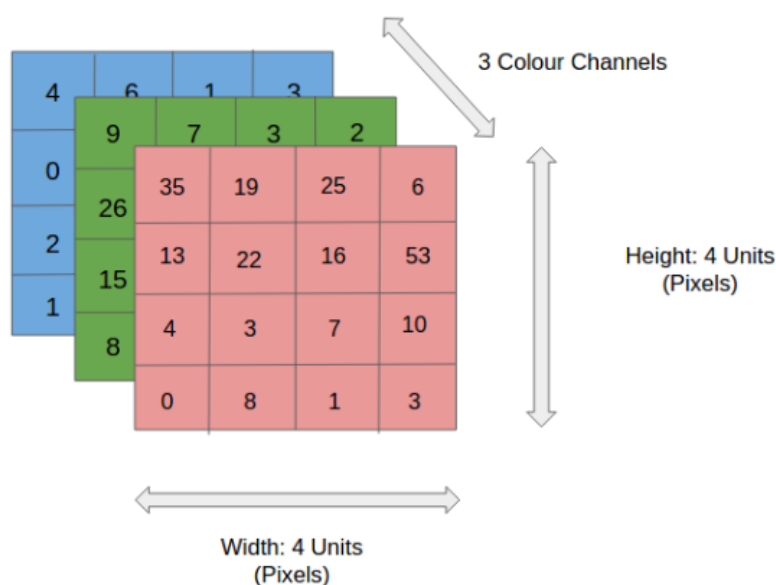


Рисунок 2.9 – RGB-зображення 4x4x3

Ви можете уявити, наскільки обчислювально інтенсивним стає, коли зображення досягає розміру, скажімо, 8K (7680x4320). Роль CNN полягає в тому, щоб перетворювати зображення у формат, який легко обробляти без втрати важливих функцій для отримання хороших прогнозів. Це важливо, коли вам потрібно розробити архітектуру, яка підходить не лише для вивчення функцій, але й може вмістити великі набори даних.

У наведеній нижче демонстрації зелена частина нагадує вхідне зображення І розміром 5x5x1. Елемент, який бере участь у виконанні операції згортки в першій

частині шару згортки, називається ядром/фільтром  $K$  і показаний жовтим кольором. Ми вибрали  $K$  як матрицю  $3 \times 3 \times 1$ .

Для дуже простих бінарних зображень цей метод може продемонструвати середню точність у прогнозуванні класів, але при роботі зі складними зображеннями, де співвідношення пікселів є значними, точність низька.

CNN вміло виявляють просторові та часові залежності в зображеннях за допомогою відповідних фільтрів. Ця архітектура ідеально підходить для завдань обробки зображень, оскільки вона використовує менше параметрів і може повторно використовувати ваги. Іншими словами, мережі можна навчити краще розуміти складність зображень.

## 2.2 Технології розробки системи

Для реалізації згорткової нейронної мережі будемо використовувати мову програмування Python та наступні моделі CNN : Faster R-CNN, YOLO, VGG16.

VGG16 — це архітектура згорткової нейронної мережі (CNN), яка була використана для перемоги в конкурсі ILSVR (Imagenet) у 2014 році. На сьогоднішній день вона вважається однією з найкращих архітектурних моделей бачення. Найбільш унікальною особливістю VGG16 є те, що замість великої кількості гіперпараметрів вони зосередилися на наявності шарів згортки фільтра  $3 \times 3$  із кроком 1 і завжди використовували той самий рівень заповнення та шару  $\text{maxpool } 2 \times 2$  фільтра кроку 2. Це відповідає такому розташуванню шарів згортки та максимального пулу послідовно по всій архітектурі. Зрештою, він має 2 FC (повністю зв'язані шари), за якими йде  $\text{softmax}$  для виведення. 16 у VGG16 означає, що він має 16 шарів, які мають ваги. Ця мережа є досить великою та має близько 138 мільйонів (приблизно) параметрів.

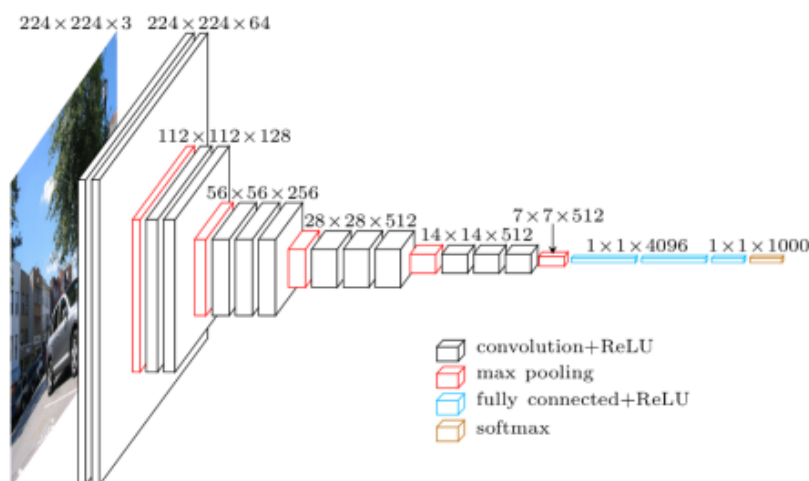


Рисунок 2.10 – Візуальний вигляд архітектури VGG16

### Основні характеристики VGG16:

- архітектура: мережа складається з 16 шарів, включаючи 13 згорткових шарів та 3 повнозв'язаних шари. Згорткові шари використовують невеликі фільтри розміром 3x3 з функцією активації ReLU, а після кожного згорткового шару слідує шар пулінгу для зменшення просторових розмірів;
- функція активації: усі згорткові шари використовують функцію активації ReLU (Rectified Linear Unit), яка допомагає уникнути проблеми з ванішінгом градієнту та прискорює збіжність;
- повнозв'язані шари: Останні три шари є повнозв'язаними, призначеними для класифікації. Зазвичай, вони використовують функцію активації softmax для отримання ймовірностей приналежності об'єкта до різних класів.
- розмір фільтрів: Використання невеликих фільтрів розміром 3x3 у всіх згорткових шарах є характерним для VGG16. Це дозволяє отримувати рецептивні поля більших глибин із меншою кількістю параметрів;
- архітектурна концепція: архітектура VGG16 визначається простотою та структурною єдністю. Така концепція робить її легкою для розуміння та адаптації;
- попереднє навчання: VGG16 зазвичай використовується з передньою навчаною моделлю на великому наборі даних, такому як ImageNet. Це дозволяє

моделі вивчати загальні особливості зображень, що полегшує її використання для подальших завдань класифікації чи визначення об'єктів.

Загальною ідеєю VGG16 є використання глибокої архітектури для вивчення складних особливостей зображень та досягнення вражаючих результатів у завданнях класифікації зображень.

Для використання VGG16 для класифікації дорогоцінного каміння, спочатку збирається набір даних з різними видами каменів. Після цього завантажується передньо навчена модель VGG16, і виконується адаптація шляхом заміни останнього повнозв'язаного шару для відповідності кількості класів дорогоцінного каміння. Модель навчається на власному наборі даних, використовуючи передньо навчені ваги для витягування важливих ознак. Після завершення навчання модель може бути використана для класифікації нових зображень дорогоцінного каміння.

YOLO — це сучасний алгоритм виявлення об'єктів, який завдяки своїй потужності обробки став майже стандартним способом виявлення об'єктів у полі комп'ютерного зору. Раніше люди використовували такі методи, як розсувні вікна, RCNN, швидкий RCNN і більш швидкий RCNN для виявлення об'єктів.

Але в 2015 році було винайдено YOLO (You Only Look Once), і цей алгоритм та його наступники почали перевершувати всі інші.

Підхід YOLO значно відрізняється від двоступеневих детекторів, оскільки використовує одну нейронну мережу для всього зображення. Ця мережа сегментує зображення на регіони та прогнозує обмежувальні рамки та ймовірності для кожного регіону одночасно. Це призводить до збільшення швидкості під час процесу виявлення. Незважаючи на значне підвищення швидкості виявлення, у YOLO спостерігається зниження точності локалізації порівняно з двоступеневими детекторами, особливо при виявленні невеликих об'єктів. У наступних версіях YOLO цій проблемі приділено більше уваги.

Ця версія використовує найновіші розробки глибокого навчання та комп'ютерного зору, забезпечуючи виняткову швидкість і точність. Його ефективний дизайн відповідає широкому спектру додатків і може бути плавно

налаштований на різноманітні апаратні платформи, від периферійних пристроїв до хмарних API, завдяки його реалізації в простому у використанні пакеті Ultralytics Python.

YOLO — це найсучасніший (SOTA) алгоритм виявлення об'єктів, який настільки швидкий, що став одним із стандартних способів виявлення об'єктів у полі комп'ютерного зору. Раніше операції з ковзним вікном були найпоширенішими при виявленні об'єктів. Потім з'явилися вдосконалення та представлені швидші версії виявлення об'єктів, такі як CNN, R-CNN, Fast RCNN та багато інших.

У цій статті ми розглянемо вичерпний посібник, щоб зрозуміти кілька концепцій цієї дивовижної моделі виявлення об'єктів. Ми глибше зануримося в особливості та зрозуміємо вдосконалення YOLOv8, як безперерійно впровадити його в користувацький набір даних за допомогою платформи, наданої Paperspace, і спробуємо зрозуміти еволюцію YOLO та виклики й обмеження, пов'язані з попередніми версіями YOLO.

YOLOv8 (You Only Look Once version 8) є однією з версій алгоритму об'єктного виявлення, що визначає об'єкти на зображенні чи відео. Вона є наслідком постійного вдосконалення і розвитку алгоритму YOLO, що став популярним у галузі комп'ютерного зору.

Основні характеристики YOLOv8:

- швидкість та ефективність: однією з ключових переваг YOLO (You Only Look Once) є його швидкодія. Алгоритм аналізує зображення одноразово, визначаючи об'єкти та їх координати. Це робить його особливо ефективним для реального часу виявлення об'єктів у відеопотоці;

- архітектура мережі: YOLOv8 має модульну архітектуру, яка дозволяє використовувати різні варіанти архітектури залежно від конкретного завдання та обчислювальних можливостей. Це включає YOLOv8-S, YOLOv8-M, YOLOv8-L і YOLOv8-X, які відрізняються за розміром та складністю;

- вдосконалення точності: YOLOv8 внесла ряд вдосконалень в порівнянні з попередніми версіями YOLO. Це включає удосконалені методи витягування ознак та зменшення помилок у визначенні об'єктів;
- підтримка різних класів об'єктів: YOLOv8 може бути навчений на визначення об'єктів різних класів. Це робить його варіативним та придатним для використання в різноманітних задачах виявлення об'єктів;
- використання в різних областях: YOLOv8 застосовується у багатьох областях, таких як автономні автомобілі, системи відеоспостереження, визначення об'єктів для розробки ігор, медичні дослідження та багато інших;
- відкрите джерело: одна з важливих особливостей YOLOv8 полягає в тому, що він є відкритим програмним забезпеченням, що дозволяє розробникам і дослідникам вносити свої внески та вдосконалювати алгоритм.

Узагальнюючи, YOLOv8 представляє сучасний підхід до завдань виявлення об'єктів із підтримкою високої швидкодії та точності.

У 2014 році група дослідників з Каліфорнійського університету в Берклі розробила глибоку згорткову мережу під назвою R-CNN (скорочено від regional-based convolutional neural network), яка може виявляти 80 різних типів об'єктів на зображеннях. Порівняно із загальним конвеєром методів виявлення об'єктів, показаним на попередньому малюнку, основний внесок R-CNN полягає лише у вилученні функцій на основі згорткової нейронної мережі (CNN). Крім цього, все схоже на загальний конвеєр виявлення об'єктів. На наступному малюнку показано роботу моделі R-CNN.

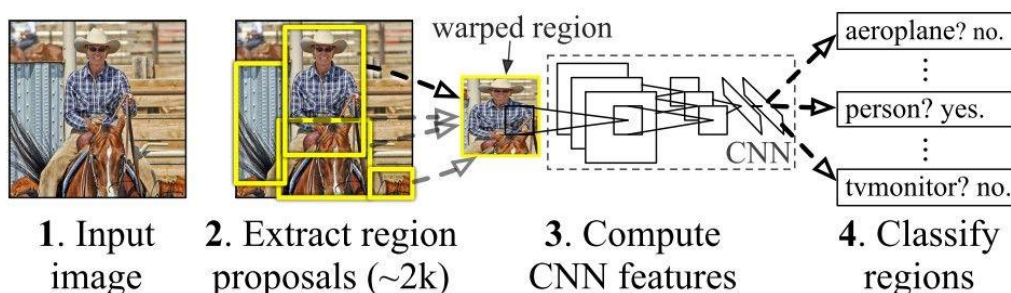


Рисунок 2.11 – Концепція мереж регіональних пропозицій (RPN)

Faster R-CNN (Faster Region-based Convolutional Neural Network) - це алгоритм глибокого навчання для виявлення та локалізації об'єктів на зображеннях. Він поєднує швидкодію та високу точність, що робить його ефективним в різних задачах комп'ютерного зору.

Faster R-CNN складається з двох основних компонентів:

- Region Proposal Network (RPN): RPN використовується для автоматичного створення пропозицій областей, де можуть знаходитися об'єкти. Він генерує багато прямокутних областей, які потенційно можуть містити об'єкти, і оцінює ймовірність того, що об'єкт присутній в цих областях.

- об'єктний виявлення: Після отримання пропозицій RPN, об'єктний виявлення використовується для класифікації областей та точної локалізації об'єктів у цих областях. Цей етап використовує Convolutional Neural Network (CNN) для ефективного витягування ознак та прогнозування класів та координат об'єктів.

Region Proposal Network (RPN) є ключовою складовою алгоритму Faster R-CNN (Faster Region-based Convolutional Neural Network), який використовується для автоматичного генерації пропозицій областей, де можуть розташовуватися об'єкти на зображенні. RPN допомагає визначити потенційні області і їхню відповідність об'єктам чи фону.

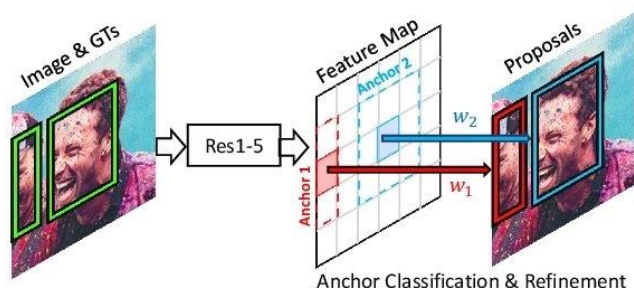


Рисунок 2.12 – Концепція мереж регіональних пропозицій (RPN)

## Висновки до розділу 2

У результаті проведеного дослідження розглянуто загальні методи реалізації комп'ютерного зору за для виконання завдання по розпізнаванню дорогоцінних каменів. Виявлено, що ці методи є дуже ефективними в різних завданнях обробки зображень, таких як класифікація, детекція об'єктів, семантична сегментація тощо.

Проведений огляд літератури показав, що моделі CNN мають великий потенціал у вирішенні різноманітних завдань відповідно до вимог сучасних досліджень у галузі комп'ютерного зору. Вони здатні автоматично визначати та розпізнавати об'єкти на зображеннях з вражаючою точністю, що робить їх незамінним інструментом у багатьох прикладних областях, включаючи медицину, автономні транспортні засоби, робототехніку та багато інших.

Крім того, у розділі також розглянуто основні принципи роботи згорткових нейронних мереж, їх архітектуру та навчання. Було показано, що використання великих обсягів даних для навчання таких моделей може значно покращити їхню продуктивність та точність у вирішенні завдань комп'ютерного зору.

Також було встановлено, що кращим варіантом буде створення системи на Python, яка реалізує комп'ютерний зір через згорткові нейронні мережі декількох моделей, а саме Faster R-CNN, YOLOv8 та VGG16.



## 3 ОПИС ВХІДНИХ ДАНИХ, СТВОРЕННЯ СИСТЕМИ ТА ТРЕНУВАННЯ МОДЕЛЕЙ

### 3.1 Опис вхідних даних

Для тренування моделей нам буде потрібно великий обсяг зображень дорогоцінних каменів різних типів. У відкритому доступі на Kaggle присутній великий датасет який містить понад 3200 зображень різних дорогоцінних каменів.

Цей набір даних складається з двох папок:

- train (~56 МБ): цей файл містить 87 вкладених папок і загалом ~2800 файлів. Кожна вкладена папка містить зображення .jpeg різних дорогоцінних каменів.

- test (~8 МБ): цей файл містить 87 вкладених папок і загалом ~400 файлів. Кожна вкладена папка містить зображення .jpeg різних дорогоцінних каменів

Зображення згруповані в 87 класів, які вже розділені на тренувальні та тестові дані. Усі зображення мають різні розміри та формат .jpeg. Приклад частини однієї з директорій датасету можна побачити на малюнку нижче:

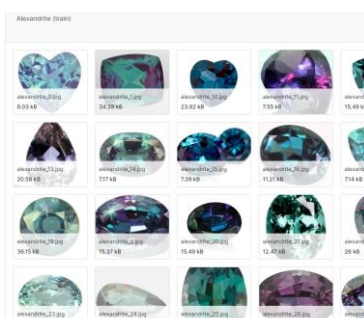


Рисунок 3.1 – Колекція зображень для тренування александриту

### 3.2 Завантаження та аугментація датасету

Після обрання датасету, підв'яжемо та перевіримо його. Для цього встановлюємо необхідні пакети та перевіряємо кількість файлів у кожному класі дорогоцінних каменів у папці /input/gemstones-images. Зображення вже поділено на

тренувальні (~2800 зображень) і тестові (~400 зображень) дані. Кожен клас у наборі поїздів містить 27 - 47 зображень, у тестовому наборі - 4 - 6 зображень.

Створимо список CLASSES, який містить назви 87 класів дорогоцінних каменів на основі імен папок та побудуємо графік розподілу даних;

```

1 import os
2 import matplotlib.pyplot as plt
3 import seaborn as sn
4
5 import cv2
6 from random import randint
7
8 import numpy as np

```

Рисунок 3.2 – Імпортування необхідних бібліотек

```

10 CLASSES, gems = [], [] # назва класів
11
12 for root, dirs, files in os.walk('/kaggle/input/gemstones-images'):
13     f = os.path.basename(root)
14     |
15     |
16     | if len(files) > 0:
17         gems.append(len(files))
18         if f not in CLASSES:
19             CLASSES.append(f) # add folder name
20
21     # розкоментуйте цей блок, якщо ви хочете виводити текст про кожну вкладену папку
22     # count_dirs = 0
23     # для f у каталогах: # кількість вкладених папок
24     # count_dirs += 1
25     # глибина = root.split(os.sep)
26     # print((len(depth) - 2) * '-'+>', '{}:\t {} folders, {} imgs'.format(os.path.basename(root)
27
28     gems_count = len(CLASSES) # 87 = к-сть класів
29     print('{} classes with {} images in total'.format(len(CLASSES), sum(gems)))
30
31     f, ax = plt.subplots(figsize=(15, 6))
32     if (gems[0]) < 10:
33         plt.bar(range(gems_count), gems[gems_count:], label='Train data')
34         plt.bar(range(gems_count), gems[0:gems_count], label='Test data')
35     else:
36         plt.bar(range(gems_count), gems[0:gems_count], label='Train data')
37         plt.bar(range(gems_count), gems[gems_count:], label='Test data')
38     ax.grid()
39     ax.legend(fontsize=12)

```

Рисунок 3.3 – Створення списку CLASSES та будовання графіку

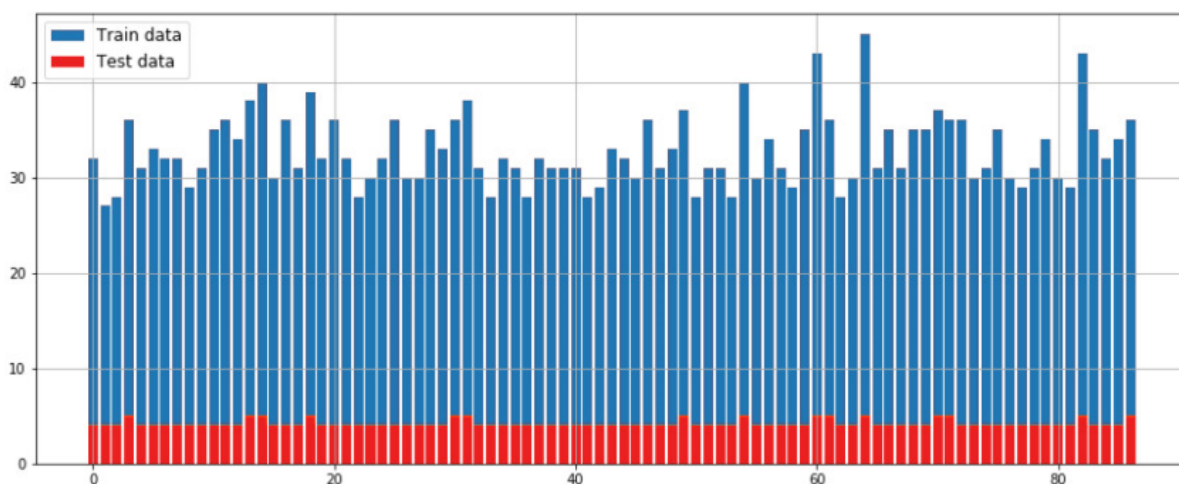


Рисунок 3.4 – Графік розподілу зображень у датасеті

Після перевірки датасету, потрібно провести аугментацію зображень. В нашому випадку змінюємо розмір зображення до `img_w`, `img_h` - ця опція буде використовуватися при кадруванні та як параметр нейронної мережі та надамо шлях до датасету навчання.

```
img_w, img_h = 220, 220 # width and height of image
train_dir = 'input/gemstones-images/train/'
```

Рисунок 3.5 – Зміна розміру зображення

Далі створюємо функцію зчитування зображень та класів, її функціонал буде виглядати наступним чином. Читання кожного зображення з диска за допомогою `cv2` і змінювання його розміру до `img_w*1.5`, `img_h*1.5`. Потім встановлення опції `cv2.COLOR_BGR2RGB`, оскільки `opencv` читає та відображає зображення як колірний формат BGR замість колірного формату RGB. Без цього параметра зображення відобразатимуться синього кольору, оскільки `matplotlib` використовує RGB для відображення зображення. Створення списку імен класів під час читання папок - Amethyst, Онух тощо. Конвертування в масив Numpy. Повернення котежу 2 елементів: зображення та відповідні мітки.

```

44 def read_imgs_lbls(_dir):
45     Images, Labels = [], []
46     for root, dirs, files in os.walk(_dir):
47         f = os.path.basename(root) # ім'я класу - Amethyst, Onyx, etc
48         for file in files:
49             Labels.append(f)
50             try:
51                 image = cv2.imread(root + '/' + file) # читання зображення (OpenCV)
52                 image = cv2.resize(image, (
53                     int(img_w * 1.5), int(img_h * 1.5))) # зміна розміру
54                 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # конвертація BGR на RGB
55                 Images.append(image)
56             except Exception as e:
57                 print(e)
58     Images = np.array(Images)
59     return (Images, Labels)

```

Рисунок 3.6 – Функція зчитування зображень та класів

Наступною функцією буде функція, яка перетворює мітки рядків на список чисел за допомогою списку CLASSES. Індекс представлятиме мітку класу, наприклад. Рубін = 0, Аметист = 24 і т.д.

```

62 def get_class_index(Labels):
63     for i, n in enumerate(Labels):
64         for j, k in enumerate(CLASSES):
65             if n == k:
66                 Labels[i] = j
67     Labels = np.array(Labels)
68     return Labels

```

Рисунок 3.7 – Функція перетворення міток на числа

Далі створюємо два масиви Train\_Img, Train\_Lbls, які містять зображення та відповідні назви класів дорогоцінних каменів відповідно. Перетворюємо Train\_Lbls із рядками на список із відповідними номерами. Надрукуємо розміри обох масивів numpy: Train\_Img, який зберігає зображення, є 4-вимірним: кількість зображень x ширина зображення x висота зображення x канал зображення.

```

70 Train_Imgs, Train_Lbls = read_imgs_lbls(train_dir)
71 Train_Lbls = get_class_index(Train_Lbls)
72 print('Shape of train images: {}'.format(Train_Imgs.sh
73 print('Shape of train labels: {}'.format(Train_Lbls.sh

```

Рисунок 3.8 – Створення масивів які містять зображення та назви

Далі виводимо зображення та номер класу з використанням matplotlib і random, виведемо 16 (4x4) випадкових зображень із набору та їхні мітки (як рядок і як ціле число).

```

dim = 4 можна змінити проєрп

fig, ax = plt.subplots(dim, dim)
fig.subplots_adjust(0, 0, 2, 2)
for i in range(0, dim):
    for j in range(0, dim):
        rnd_number = randint(0, len(Train_Imgs))
        cl = Train_Lbls[rnd_number]
        ax[i, j].imshow(Train_Imgs[rnd_number])
        ax[i, j].set_title(CLASSES[cl]+' : ' + str(cl))
        ax[i, j].axis('off')

```

Рисунок 3.9 – Виведення випадкових зображень

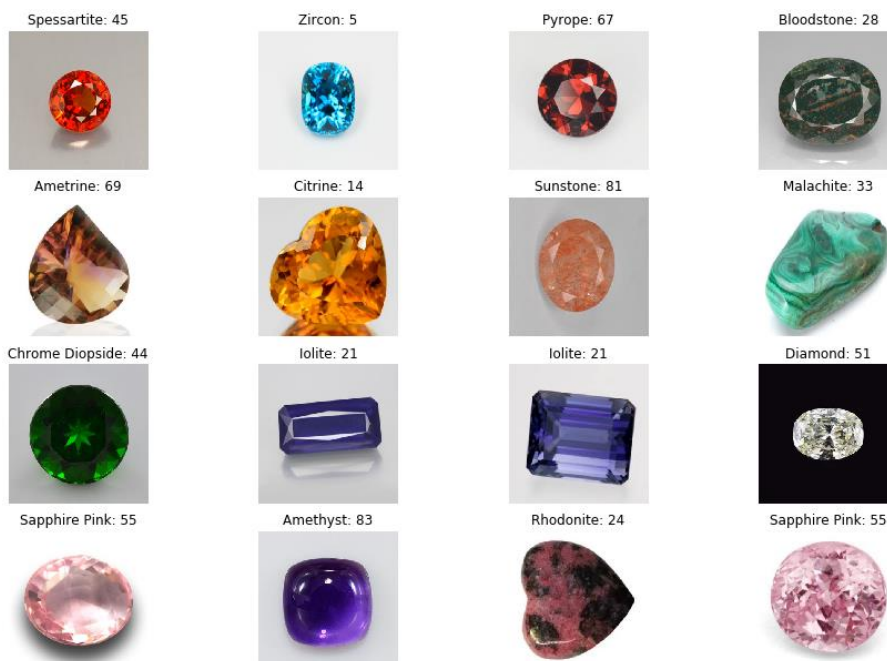


Рисунок 3.10 – Масив 4x4 випадкових зображень

Наступним кроком буде обрізання зображень по лініям контуру каменів з використанням алгоритму Canny задля видалення непотрібних пікселів. Canny — це популярний алгоритм виявлення країв, який виявляє краї об'єктів, присутніх на зображенні.

Використовуючи cv2.Canny, знайдемо масив, що представляє рамку, яка є краями, якими буде обрізано оригінальне зображення. Функція `edge_and_cut(img)` отримує одне зображення та повертає обрізане зображення (`new_img`) розміром `img_w`, `img_h`. Іноді алгоритм Canny не може виявити краї (наприклад, коли об'єкт має майже такий самий колір, як фон), тому краю масиву будуть мати нульове значення. У цьому випадку використовуйте оригінальне зображення.

```

88 def edge_and_cut(img):
89     try:
90         edges = cv2.Canny(img, img_w, img_h)
91
92         if (np.count_nonzero(edges) > edges.size / 10000):
93             pts = np.argwhere(edges > 0)
94             y1, x1 = pts.min(axis=0)
95             y2, x2 = pts.max(axis=0)
96
97             new_img = img[y1:y2, x1:x2] # обрізання
98             new_img = cv2.resize(new_img, (img_w, img_h)) # конвертація у минулий
99         else:
100            new_img = cv2.resize(img, (img_w, img_h))
101
102     except Exception as e:
103         print(e)
104         new_img = cv2.resize(img, (img_w, img_h))
105
106     return new_img

```

Рисунок 3.11 – Функція обрізання країв зображення

Після створення функції, випробуємо її на наших зображеннях та виведемо результат. Функція `show_cropped` є свого роду дублікатом `edge_and_cut()`: вона показує однакові випадкові приклади роботи алгоритму Canny: оригінальне зображення, краї Canny, зображення з обмежувальною рамкою, обрізане зображення для кращого розуміння того, як працює алгоритм Canny.

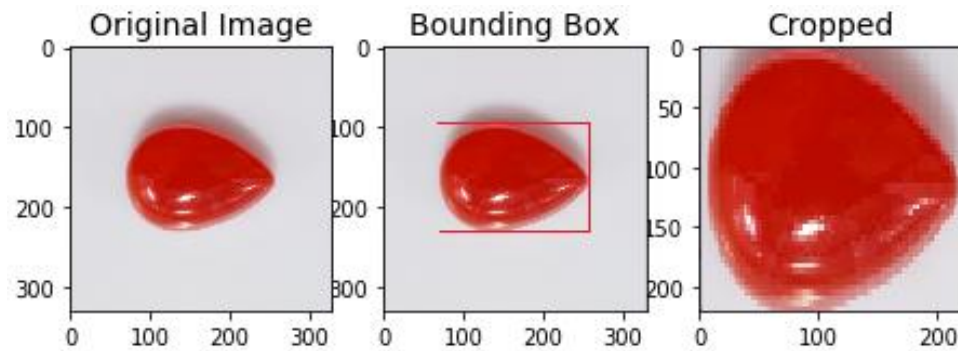


Рисунок 3.12 – Тестування алгоритму на зображеннях

Провівши тестування алгоритму, можна замінювати наші зображення перетвореними. Створимо функцію, яка викликає `edge_and_cut` і замінює масив `Train_Imgs` numpy масивом обрізаних зображень. Не забуваємо, що зображення, які не можна обрізати, будуть замінені оригіналами.

```

108  def crop_images(Imgs):
109      CroppedImages = np.ndarray(shape=(len(Imgs), img_w, img_h, 3), dtype
110
111      ind = 0
112      for im in Imgs:
113          x = edge_and_cut(im)
114          CroppedImages[ind] = x
115          ind += 1
116
117      return CroppedImages

```

Рисунок 3.13 – Функція замінення оригіналних зображень на обрізані

Останнім кроком перед створенням моделей поділимо датасет на тренувальний та перевірочний. Використаємо `sklearn` щоб розділити `Train_Imgs`, `Train_Lbls` на набори тренування (80%) і перевірки (20%).

```

122  from sklearn.model_selection import train_test_split
123  X_train, X_val, y_train, y_val = train_test_split(Train_Imgs, Train_Lbls, shuff
124  print('Shape of X_train: {}, y_train: {}'.format(X_train.shape, y_train.shape)
125  print('Shape of X_val: {}, y_val: {}'.format(X_val.shape, y_val.shape))

```

Рисунок 3.14 – Розподілення датасету на тренувальний та перевірочний

### 3.3 Створення моделі на архітектурі VGG16

VGG16 – одна з найзнаменитіших моделей, відправлених на змагання ILSVRC-2014. Вона є покращеною версією AlexNet, в якій замінені великі фільтри (розміру 11 і 5 в першому і другому шарі верстата, відповідно) на кілька фільтрів розміру 3x3, наступних один за одним.

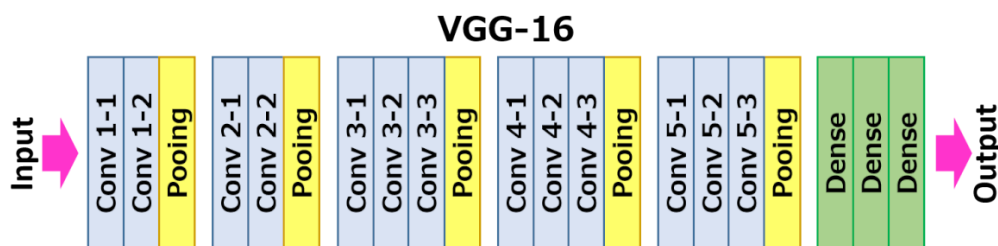


Рисунок 3.15 – Візуальний вигляд архітектури VGG-16

Параметри для шарів:

- розмір фільтра в згортковому шарі (filters): кількість фільтрів повинна залежати від складності набору даних та глибини нейронної мережі. Зазвичай починають з налаштування 32, 64, 128 для трьох шарів;
- розмір ядра: він визначає розмір фільтра, який є невеликим вікном пікселів одночасно (наприклад, 3x3), яке буде переміщуватися до того, як буде проскановано всю зображення. Якщо зображення менше 128x128, рекомендується працювати з меншими фільтрами розміром 1x1;
- ширина і висота зображень: ці розміри вже надані. 2D згорткові шари приймають тривимірний вхід, зазвичай зображення з трьома кольоровими каналами;
- максимальне згортання: це застосування рухомого вікна по 2D вхідному просторі, де максимальне значення в цьому вікні є вихідним, часто виконується з вікном розміром 2x2.



```

6 filters = 32      # the dimensionality of the output space
7 kernel_size = 3  # length of the 2D convolution window
8 max_pool = 2     # size of the max pooling windows

```

Рисунок 3.16 – Розподілення датасету на тренувальний та перевірючий

Параметри для підгонки моделі:

Епоха описує кількість разів, коли алгоритм бачить ВЕСЬ набір даних. Кожного разу, коли алгоритм бачив всі вибірки в наборі даних, завершилася одна епоха.

Оскільки одна епоха занадто велика, щоб подати її в пам'ять одночасно, розбиваємо її на кілька менших партій. Розмір партії завжди є множителем 2.

Ітерації на епоху = кількість проходів, кожен прохід використовує партію з кількістю прикладів, рівною розміру партії.

Таким чином, якщо у нас є ~2200 (80%) прикладів для навчання, і розмір партії становить 32, тоді для завершення 1 епохи знадобиться ~70 ітерацій.

```

10 EPOCHS = 70      # while testing you
11 batch_size = 32 # number of training
12 iter_per_epoch = len(X_train) // batch_size # each sample will
13 val_per_epoch = len(X_val) // batch_size   # each sample will

```

Рисунок 3.17 – Параметри моделі

Додаємо 5 блоків:

- Conv2D з вказаними гіперпараметрами: Conv2D(kernel\_size, (filters, filters), input\_shape=(img\_w, img\_h, 3)) з функцією активації для кожного шару у вигляді Rectified Linear Unit (ReLU): Activation('relu');

- шар MaxPooling2D для зменшення просторового розміру вхідних ознак; 2D вхідний простір: MaxPooling2D (pool\_size=(max\_pool, max\_pool));

- виконаємо те ж саме, збільшуючи розмір ядра: 32 -> 64 -> 128 -> 128 -> 128;

- вирівнюємо вхід: перетворити багатовимірний вектор у одновимірний вектор: Flatten();
- додаємо dropout шар, який випадковим чином встановлює певну частину свого входу в 0 і допомагає зменшити перенавчання: Dropout(0.5);
- додаємо повністю з'єднаний шар з 512 вузлами та функцією активації relu: Dense(512), Activation('relu');
- додаємо останній повністю з'єднаний шар, який вказує кількість класів дорогоцінних каменів: 87. Функція активації softmax виводить вектор, який представляє розподіл ймовірностей списку можливих результатів: Dense(87, activation='softmax').

```

Model: "sequential_1"
-----
Layer (type)                Output Shape              Param #
-----
conv2d_1 (Conv2D)           (None, 220, 220, 32)     896
-----
max_pooling2d_1 (MaxPooling2 (None, 110, 110, 32)     0
-----
conv2d_2 (Conv2D)           (None, 110, 110, 64)     18496
-----
max_pooling2d_2 (MaxPooling2 (None, 55, 55, 64)     0
-----
conv2d_3 (Conv2D)           (None, 55, 55, 128)     73856
-----
max_pooling2d_3 (MaxPooling2 (None, 27, 27, 128)     0
-----
conv2d_4 (Conv2D)           (None, 27, 27, 128)     147584
-----
average_pooling2d_1 (Average (None, 13, 13, 128)     0
-----
conv2d_5 (Conv2D)           (None, 13, 13, 128)     147584
-----
max_pooling2d_4 (MaxPooling2 (None, 6, 6, 128)     0
-----
flatten_1 (Flatten)         (None, 4608)             0
-----
dropout_1 (Dropout)         (None, 4608)             0
-----
dense_1 (Dense)             (None, 512)              2359808
-----
dense_2 (Dense)             (None, 87)                44631
-----
Total params: 2,792,855
Trainable params: 2,792,855
Non-trainable params: 0
-----

```

Рисунок 3.18 – Висновок по моделі

Для початку модель буде складена з використанням бібліотеки Keras. Послідовно будуть додані різні типи шарів: згорткові, пулінгу та повністю з'єднані, з активацією ReLU для кожного з них. Також буде використаний Dropout шар для запобігання перенавчанню.

Після складання моделі, вона буде скомпільована за допомогою оптимізатора Adam, який є узагальненням алгоритму стохастичного градієнтного спуску (SGD). В якості функції втрати обрано `sparse_categorical_crossentropy`, оскільки ми робимо багатокласову класифікацію.

```
model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=
```

Рисунок 3.19 – Компіляція моделі

Image augmentation - це процес створення додаткових тренувальних даних на основі існуючих зображень, таких як зміщення, обертання, відображення та збільшення. З огляду на обмежену кількість зразків для кожного класу, до тренувального генератора даних слід додати клас `ImageDataGenerator` з параметрами аугментації. Клас `ImageDataGenerator` з бібліотеки Keras використовується для створення додаткових зображень кожного класу дорогоцінних каменів у пам'яті.

```
37 model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
38
39 from keras.preprocessing.image import ImageDataGenerator
40
41 train_datagen = ImageDataGenerator( # this is the augmentation configuration
42     rotation_range=25,
43     zoom_range=0.1,
44     width_shift_range=0.1,
45     height_shift_range=0.1,
46     shear_range=0.2,
47     horizontal_flip=True
48 )
49
50 val_datagen = ImageDataGenerator()
```

Рисунок 3.20 – Клас `ImageDataGenerator`

### Оригінальне зображення + приклади роботи ImageDataGenerator:

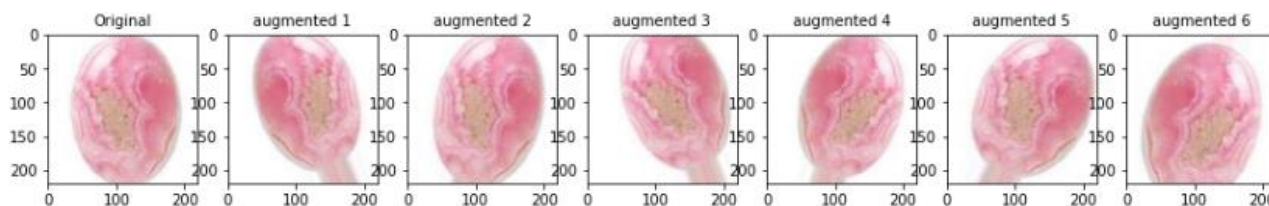


Рисунок 3.21 – Приклад роботи класу

Створимо два ітератори масиву numpy `train_gen` і `val_gen` і заповнимо їх додатковими зображеннями:

```
train_gen = train_datagen.flow(X_train, y_train, batch_size=batch_size)
val_gen = val_datagen.flow(X_val, y_val, batch_size=batch_size)
```

Рисунок 3.22 – Масиви train і val

Далі можна буде побачити, що параметр `val_loss` збільшується, це переобладнання. Це трапляється, коли наша модель занадто добре пояснює навчальні дані, а не збирає шаблони, які можуть допомогти узагальнити невидимі дані.

```
m = model.fit_generator(
    train_gen,
    steps_per_epoch= iter_per_epoch,
    epochs=EPOCHS,
    validation_data = val_gen,
    validation_steps = val_per_epoch,
    verbose = 1 # Verbosity mode. 0 = silent, 1 = progress bar,
)
```

Рисунок 3.23 – Масиви train і val

На рисунку нижче, будуть показані епохи, час тренування епохи, `acc`, `val_acc`, `loss`, `val_loss`.

Кафедра інтелектуальних інформаційних систем  
Система розпізнавання дорогоцінних каменів за допомогою комп'ютерного зору

```

Epoch 1/70
71/71 [-----] - 26s 360ms/step - loss: 5.2989 - accuracy: 0.0364 - val_loss: 3.8485 - val_accuracy: 0.0423
Epoch 2/70
71/71 [-----] - 25s 353ms/step - loss: 3.4145 - accuracy: 0.1159 - val_loss: 2.7371 - val_accuracy: 0.1889
Epoch 3/70
71/71 [-----] - 23s 322ms/step - loss: 2.7974 - accuracy: 0.2167 - val_loss: 2.6321 - val_accuracy: 0.2333
Epoch 4/70
71/71 [-----] - 23s 319ms/step - loss: 2.5033 - accuracy: 0.2664 - val_loss: 2.5391 - val_accuracy: 0.2926
Epoch 5/70
71/71 [-----] - 24s 340ms/step - loss: 2.2983 - accuracy: 0.3091 - val_loss: 2.6524 - val_accuracy: 0.3130
Epoch 6/70
71/71 [-----] - 23s 324ms/step - loss: 2.1509 - accuracy: 0.3508 - val_loss: 2.4501 - val_accuracy: 0.3574
Epoch 7/70
71/71 [-----] - 25s 354ms/step - loss: 2.0295 - accuracy: 0.3702 - val_loss: 1.6725 - val_accuracy: 0.5907
Epoch 8/70
71/71 [-----] - 24s 336ms/step - loss: 2.0215 - accuracy: 0.3925 - val_loss: 2.0731 - val_accuracy: 0.4759
Epoch 9/70
71/71 [-----] - 23s 324ms/step - loss: 1.8226 - accuracy: 0.4196 - val_loss: 2.1565 - val_accuracy: 0.4333
Epoch 10/70
71/71 [-----] - 24s 338ms/step - loss: 1.7442 - accuracy: 0.4347 - val_loss: 1.9034 - val_accuracy: 0.4352
Epoch 11/70
71/71 [-----] - 23s 322ms/step - loss: 1.6321 - accuracy: 0.4689 - val_loss: 1.8731 - val_accuracy: 0.4278
Epoch 12/70
71/71 [-----] - 23s 329ms/step - loss: 1.6472 - accuracy: 0.4751 - val_loss: 1.5584 - val_accuracy: 0.4463
Epoch 13/70
71/71 [-----] - 23s 322ms/step - loss: 1.6327 - accuracy: 0.4707 - val_loss: 1.6522 - val_accuracy: 0.4352
Epoch 14/70
71/71 [-----] - 22s 315ms/step - loss: 1.5628 - accuracy: 0.4956 - val_loss: 1.7782 - val_accuracy: 0.4889
Epoch 15/70
71/71 [-----] - 23s 330ms/step - loss: 1.4012 - accuracy: 0.5475 - val_loss: 2.1117 - val_accuracy: 0.4852
Epoch 16/70
71/71 [-----] - 22s 304ms/step - loss: 1.4849 - accuracy: 0.5121 - val_loss: 2.1573 - val_accuracy: 0.4778
Epoch 17/70
71/71 [-----] - 22s 306ms/step - loss: 1.5307 - accuracy: 0.5164 - val_loss: 1.4521 - val_accuracy: 0.5037
Epoch 18/70
71/71 [-----] - 23s 320ms/step - loss: 1.3594 - accuracy: 0.5613 - val_loss: 1.6437 - val_accuracy: 0.5663
Epoch 19/70
71/71 [-----] - 22s 309ms/step - loss: 1.2936 - accuracy: 0.5861 - val_loss: 2.0103 - val_accuracy: 0.5533

```

Рисунок 3.24 – Процес тренування моделі

Після тренування перевіримо точність. Для цього побудуємо графіки точності(train та val) до епохи та loss(train та val) до епохи.

```

205 fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(15,5))
206 axs[0].plot(m.history['accuracy'])
207 axs[0].plot(m.history['val_accuracy'])
208 axs[0].set_title('Model accuracy')
209 axs[0].legend(['Train', 'Val'], loc='upper left')
210
211 axs[1].plot(m.history['loss'])
212 axs[1].plot(m.history['val_loss'])
213 axs[1].set_title('Model loss')
214 axs[1].legend(['Train', 'Val'], loc='upper left')
215
216 for ax in axs.flat:
217     ax.set(xlabel='Epoch')

```

Рисунок 3.25 – Побудова графіків

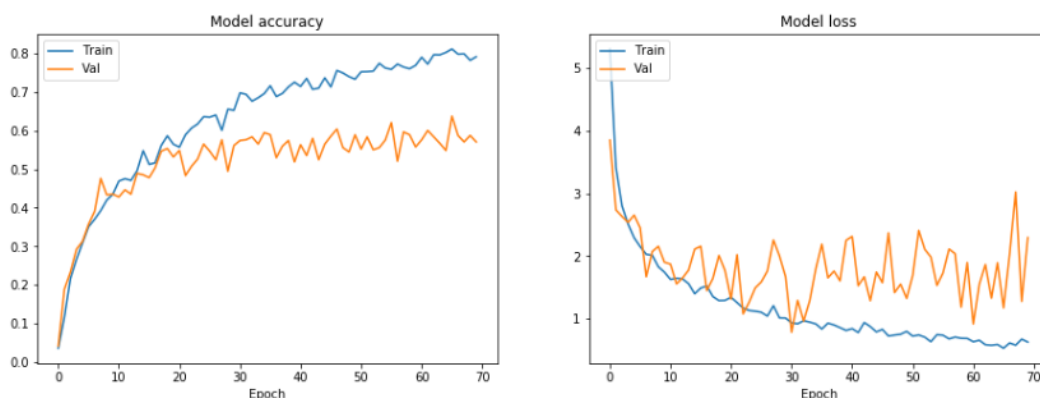


Рисунок 3.26 – Графіки точності та loss до епохи

Після тренування моделі за допомогою бібліотек у відкритому доступі були таким же чином реалізовані моделі YOLOv8 та Faster R-CNN. Після отриманих результатів була зроблена порівняльна таблиця для демонстрації ефективності різних моделей.

Таблиця 3.1 – Порівняння метрик моделей

Алгоритм	Точність	Точність val	loss	loss val
VGG16	0,7656	0,5412	0,6743	2,36
YOLOv8	0,8772	0,6194	0,4723	1,653
Faster R-CNN	0,8409	0,5904	0,6316	2,295

Не судіть погану модель. Просто подивіться на альмандин, червоний гранат, гессоніт, піроп і родоліт. Чи можете ви їх відрізнити?

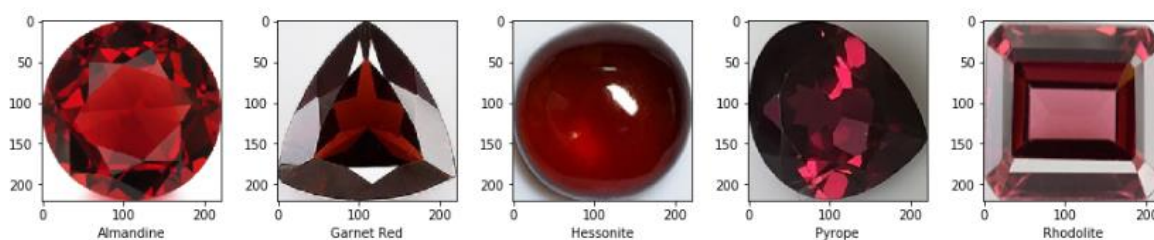


Рисунок 3.27 – Приклад майже однакових за виглядом каменів

Після того як створені моделі були натреновані, протестуємо їх на зображеннях які не були у датасеті. Ці зображення також були аугментовані по прикладу як робили з тестовими зображеннями. Для тестування було обрано модель з найкращим результатом (YOLOv8) та 25 зображень з відкритого доступу. Результат роботи тестування на нових зображеннях можна побачити на малюнку нижче.



Рисунок 3.28 – Результат роботи застосунку

### Висновки до розділу 3

Здійснено розробку системи на мові Python, яка реалізує комп'ютерний зір через згорткові нейронні мережі таких моделей як Faster R-CNN, YOLOv8 та VGG-16. Розроблена з нуля модель VGG16 та зроблено порівняльну таблицю моделей. Система спроможна розпізнавати дорогоцінні камені з точністю 61% що перевершує людський показник у 58%.

## ВИСНОВКИ

У кваліфікаційній роботі магістра досліджено важливість та методи розпізнавання дорогоцінних каменів за допомогою сучасних технологій комп'ютерного зору. Аналіз предметної області виявив, що ідентифікація дорогоцінних каменів вимагає глибокого розуміння їхніх властивостей та характеристик. Використання машинного навчання та комп'ютерного зору відкриває нові можливості для точного та швидкого розпізнавання каменів, що може значно спростити та автоматизувати процес їх оцінки та класифікації.

Проаналізовано методи та підходи, що використовуються для розв'язання задачі ідентифікації, включаючи згорткові нейронні мережі та різні алгоритми машинного навчання. Розроблені моделі демонструють здатність ефективно розпізнавати та класифікувати дорогоцінні камені на основі візуальних даних, надаючи високу точність розпізнавання.

Створена система має змогу розпізнавати за допомогою комп'ютерного зору 83 види дорогоцінних каменів з точністю у 61%, що є вище чим показник опізнавання каменів геммологами професіоналами. Автоматизація процесу класифікації сприятиме підвищенню продуктивності та зниженню ймовірності помилок, що особливо важливо для забезпечення якості та конкурентоспроможності на сучасному ринку.

Використання комп'ютерного зору та машинного навчання дозволяє значно підвищити точність та швидкість аналізу, спрощуючи та автоматизуючи трудомісткі процеси, що традиційно вимагали високої кваліфікації та значних часових витрат. Такий підхід відкриває широкі перспективи для подальшого розвитку галузі, сприяючи впровадженню інновацій та покращенню якості геммологічного аналізу.



## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Craig L., Awati R. What is a convolutional neural network (CNN)?. *Enterprise AI*. URL: <https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network> (дата звернення: 01.02.2024).
2. Measurement of gem colour using a computer vision system: a case study with jadeite-jade. *MDPI*. URL: <https://www.mdpi.com/2075-163X/11/8/791> (date of access: 01.02.2024).
3. Multiclass amber gemstones classification with various segmentation and committee strategies. *IEEE Xplore*. URL: <https://ieeexplore.ieee.org/document/6662694> (date of access: 01.02.2024).
4. Rimkus K. Automatic amber gemstones identification by color and shape visual properties. *Academia.edu* - *Share research*. URL: [https://www.academia.edu/11955549/Automatic\\_amber\\_gemstones\\_identification\\_by\\_color\\_and\\_shape\\_visual\\_properties](https://www.academia.edu/11955549/Automatic_amber_gemstones_identification_by_color_and_shape_visual_properties) (date of access: 01.02.2024).
5. What are convolutional neural networks? | IBM. *IBM in Deutschland, Österreich und der Schweiz*. URL: <https://www.ibm.com/topics/convolutional-neural-networks> (дата звернення: 19.02.2024).
6. What Is a Convolutional Neural Network? | 3 things you need to know. *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink*. URL: <https://www.mathworks.com/discovery/convolutional-neural-network.html> (дата звернення: 01.02.2024).
7. Convolutional neural network. URL: [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network) (дата звернення: 01.02.2024).
8. Turker MBC – веб-сайт. URL: <https://www.researchgate.net/scientific-contributions/Mustafa-Turker-2120549925> (дата звернення: 01.02.2024).
9. U-Net Architecture. URL: <https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/> (дата звернення: 01.02.2024).

10. Seg-Net Architecture. URL: <https://arxiv.org/abs/1511.00561> (дата звернення: 01.02.2024).
11. FC-DenseNet Architecture. URL: [https://www.researchgate.net/figure/FC-DenseNet-architecture\\_fig1\\_337761543](https://www.researchgate.net/figure/FC-DenseNet-architecture_fig1_337761543) (дата звернення: 01.02.2024).
12. List of gemstones: precious and semi-precious stones - gem society. *International Gem Society*. URL: <https://www.gemsociety.org/gemstone-encyclopedia/> (дата звернення: 01.02.2024).
13. Gemstones: facts, photos, and information for over 100 gems. *Geology and Earth Science News, Articles, Photos, Maps and More*. URL: <https://geology.com/gemstones/> (дата звернення: 01.02.2024).
14. The Editors of Encyclopaedia Britannica. Gemstone | definition, history, types, & facts. *Encyclopedia Britannica*. URL: <https://www.britannica.com/science/gemstone> (дата звернення: 01.02.2024)
15. Caruana R. Multitask learning - machine learning. SpringerLink. URL: <https://link.springer.com/article/10.1023/A:1007379606734> (дата звернення: 12.02.2024).
16. Collobert R., Weston J. A unified architecture for natural language processing Proceedings of the 25th international conference on Machine learning. ACM Other conferences. URL: <https://dl.acm.org/doi/10.1145/1390156.1390177> (дата звернення: 12.02.2024).
17. Mikolov T. Recurrent neural network based language model | bibsonomy. BibSonomy. URL: <https://www.bibsonomy.org/bibtex/245d1279b03856348523ebefda4cac10> (дата звернення: 12.02.2024).
18. Sepp H., Jürgen S. Long Short-Term Memory. MIT Press. URL: <https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory?redirectedFrom=fulltext> (дата звернення: 13.02.2024).

19. Domsch S. Dialogue in video games. Academia.edu - Share research. URL: [https://www.academia.edu/37187222/Dialogue\\_in\\_video\\_games](https://www.academia.edu/37187222/Dialogue_in_video_games) (дата звернення: 13.02.2024).

20. Neuro-Evolution of Augmenting Topologies Algorithm. medium.com. URL: <https://aitoolzai.medium.com/neuro-evolution-of-augmenting-topologies-algorithm-f303fa349cc3> (дата звернення: 06.02.2024).

21. Collobert R., Weston J. A unified architecture for natural language processing Proceedings of the 25th international conference on Machine learning. ACM Other conferences. URL: <https://dl.acm.org/doi/10.1145/1390156.1390177> (дата звернення: 12.02.2024).

22. Mikolov T. Recurrent neural network based language model | bibsonomy. BibSonomy. URL: <https://www.bibsonomy.org/bibtex/245d1279b03856348523ebefda4cac10> (дата звернення: 12.02.2024).

23. Sepp H., Jürgen S. Long Short-Term Memory. MIT Press. URL: <https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory?redirectedFrom=fulltext> (дата звернення: 13.02.2024).

24. Gemstones. *American Scientist*. URL: <https://www.americanscientist.org/article/gemstones> (дата звернення: 01.02.2024).

25. URL: <https://www.igi.org/consumer-education/gemstone-articles/> (дата звернення: 01.02.2024).

26. Gemstone articles. *Gemstones*. URL: <https://www.gemstones.com/articles> (дата звернення: 01.02.2024).

27. The Editors of Encyclopaedia Britannica. Gemstone | definition, history, types, & facts. *Encyclopedia Britannica*. URL: <https://www.britannica.com/science/gemstone> (дата звернення: 01.02.2024)

28. Caruana R. Multitask learning - machine learning. SpringerLink. URL: <https://link.springer.com/article/10.1023/A:1007379606734> (дата звернення: 12.02.2024).

29. Gemstones with health benefits - international gem society. *International Gem Society*. URL: <https://www.gemsociety.org/article/gemstones-health-benefits/> (дата звернення: 01.02.2024).

30. The classification, identification and characteristics of gemstones - A collection of historical articles on precious and semi-precious stones by read books ltd. - ebook | everand. *Everand*. URL: <https://www.everand.com/book/323161739/The-Classification-Identification-and-Characteristics-of-Gemstones-A-Collection-of-Historical-Articles-on-Precious-and-Semi-Precious-Stones> (дата звернення: 03.02.2024).