МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

«Інтелектуальна технологія детектування стану трубопроводів з аугментацією даних в режимі екзамену»

Завідувач
випускаючої кафедри **Довбиш А.С.**

Керівник роботи **Коробов А.Г.**

Студента групи ІН.м-02 **Кириченка І.О.**

СУМИ 2021

Сумський державний університет

(назва вузу)

Факультет *ЕлІТ* Кафедра *Комп'ютерних наук*

Спеціальність *«122 -Комп'ютерні науки»*

Затверджую:

зав.кафедрою _____

_____

"_____" _____ 20___р.

ЗАВДАННЯ

НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Кириченко Ігорю Олексійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) *Інтелектуальна технологія детектування стану трубопроводів з аугментацією даних в режимі екзамену*

затверджую наказом по інституту від "____" _____ 20 ___ р. № _____

2. Термін здачі студентом закінченого проекту (роботи) _____

3. Вхідні данні до проекту (роботи) _____

_____

_____

_____

_____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми. Постановка задачі дослідження. 2) Аналіз існуючих рішень детектування 3) Вибір об'єктів дослідження. Аугментація даних. Інтелектуальна система в режимі екзамену 4) Розробка інформаційної технології детектування стану трубопроводів

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

_____

_____

_____

_____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту (роботи)	Термін виконання проекту (роботи)	Примітка
1.	*Аналіз проблеми. Постановка задачі дослідження*		
2.	*Аналіз існуючих рішень детектування*		
3.	*Вибір об'єктів дослідження. Аугментація даних. Інтелектуальна система в режимі екзамену*		
4.	*Розробка інформаційної технології детектування стану трубопроводів*		
5.	*Оформлення пояснювальної записки до дипломної роботи*		

Студент – дипломник _____

(підпис)

Керівник проекту _____

(підпис)

РЕФЕРАТ

Записка: 55 стор., 27 рис., 1 табл., 1 додаток, 20 джерел.

Об'єкт дослідження — процес аналізу трубопроводів та трубних систем на предмет зношення.

Мета роботи — Підвищення функціональної ефективності інтелектуальної системи розпізнавання технічного стану трубопроводів з використанням техніки аугментації данних.

Методи дослідження — метод аугментації зображень для нейронних мереж.

Результати — спроектовано та розроблено класифікатор детектування стану трубопроводів. При цьому задача оцінки стану труб була розв'язана за допомогою підходу аугментації зображень, а сама технологія працює в режимі екзамену. Розроблений алгоритм реалізовано у формі програмного забезпечення, створеного за допомогою інструментального програмного середовища Python 3.0.

ТЕХНОЛОГІЯ ДЕТЕКТУВАННЯ, ІНФОРМАЦІЙНИЙ КРИТЕРІЙ, ВИБІРКА ЗОБРАЖЕНЬ, МЕТОД ФУНКЦІОНАЛЬНО-СТАТИСТИЧНИХ ВИПРОБУВАНЬ, ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ, АУГМЕНТАЦІЯ ДАНИХ.

Зміст

ВСТУП

Системи розпізнавання зображень на даний момент знаходяться у стані розвитку. Щодня ми чуємо про нові системи, здатні розпізнавати обличча, механізми, техніку. Машинний зір, навчений тисячами прикладів зображень та корегуваннями операторів, має набагато більший відсоток коректного розпізнавання проблем, ніж рядовий робітник. А час, необхідний для навчання інформаційної системи в десятки разів менший у порівнянні з часом навчання робітника. Не кажучи вже про втрати підприємств, які з'являться при відсутності кваліфікованих експертів.

Однією із основних тенденцій сучасного розвитку науково-технічного прогресу всіх галузей соціальноекономічної сфери суспільства є інтелектуалізація творчої діяльності шляхом моделювання на базі ЕОМ притаманних людині когнітивних процесів при прийнятті рішень. Особливо актуальним для промисловості України є використання інтелектуальних технологій при проектуванні та виготовленні виробів, що повинні задовольняти вимоги європейських та світових стандартів. На сьогодні будь-який складний виріб не є конкуренто-спроможним, якщо він не містить інтелектуальної складової. І далі ця тенденція буде тільки посилюватися. Тому основні світові виробники сучасних автоматизованих систем керування (АСК) вже давно зрозуміли, що орієнтація на виготовлення високоточного технологічного обладнання без інтелектуальної складової не дозволяє ефективно здійснювати керування слабо формалізованими процесами за умов апріорної невизначеності та ресурсних обмежень. Наприклад, основною умовою використання такого обладнання є вимога жорсткого вхідного контролю сировини та матеріалів. Але для сировини та матеріалів природного походження, які широко використовуються в хімічній, металургійній, харчовій та інших галузях промисловості, виконання такої умови є ускладненим через невирішені технічні проблеми поточного контролю. Тому саме інтелектуалізація АСК складними

слабо формалізованими об’єктами та процесами дозволяє надати їм властивості адаптивності на основі самонавчання та розпізнавання образів.

Саме тому інформаційні системи є необхідністю науково-технічного прогресу, що поступово зменшує витрати та збільшує ефективності підприємств.

1.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Дослідження актуальності проблеми

Водопостачання України є відносно складним інженерним комплексом з річною потужністю 2×109 м3. Більшість будівель комплексу вийшли за розраховані терміни експлуатації і потребують ремонту. Існуюча система за час роботи зазнала серйозних технологічних, соціально-економічних, екологічних та інших змін, що вимагає першочергової модернізації сучасної системи водопостачання країни світового рівня. Загальна сума, необхідна для відновлення водопостачання, може скласти 14 мільярдів євро. Такий рівень вартості вимагає від експертів проведення комплексного аналізу проблеми та формулювання чіткої стратегії вирішення.

Найбільшою проблемою української системи водопостачання є її ККД 30%. Технічні умови загальної системи водопостачання, в тому числі водопровідної мережі, не є ідеальними, що негативно впливає на якість очищеної води та є причиною вторинного забруднення. Для відновлення ефективної роботи водопровідної мережі кошти, необхідні на відновлення всієї системи, склали майже 76%.

Виходячи із загальної довжини водопровідної мережі 180 000 кілометрів і відповідної довжини труби певного діаметра, загальний об’єм трубопроводу становить 14,8 млн. кубометрів, а середній діаметр труби – 324 мм. Враховуючи вищезгадане значення середнього діаметра українських водопроводів, вартість ремонту близько 70 тис. кілометрів трубопроводів становить близько 3 млрд євро.

Із погіршенням технічного стану водопровідних труб значно знизилася ефективність їх експлуатації, зросли необґрунтовані втрати та витоки води. Коефіцієнт втрат води міською трубопровідною мережею занадто високий, у порівнянні із Західною Європою він коливається від 0,4 до 3,0 м3/км/год і становить 0,1-0,4 м3/км/год.

В Україні та Центрально-Східній Європі проблема вторинного забруднення води у водопровідній мережі стає все більш актуальною. Зміни в системі управління економікою суттєво вплинули на зниження споживання води, знизивши тим самим ефективність очисних споруд, насосних станцій та водорозподілення. У системах з важливими геометричними розмірами зниження продуктивності водорозподілу призводить до збільшення тривалості перебування води в ній.

Однак у цьому випадку виникає проблема, як дізнатися, яку частину трубопроводу потрібно замінити? Звичайно, ви можете найняти та навчити команду інженерів для підготовки експертних висновків на основі стану будівлі. Але якщо ця команда буде непідготовлена, компанія постраждає. Адже людям повинні платити за навчання і виділяти кошти. Пам’ятайте, що стан трубопроводів та іншого обладнання буде тільки погіршуватися.

З іншого боку, участь інтелектуальних інформаційних систем може не тільки запобігти непотрібним витратам, пов’язаним зі створенням експертних груп, а й запобігти надмірному зносу об’єктів, передчасним виходам з ладу та компенсувати пов’язані з цим витрати. Використання інтелектуальних експертних систем вимагає участі єдиного оператора, який контролює результати обробки даних системи. Сам процес визначення стану займає в рази менше часу, а деякі визначення зростають в геометричній прогресії.

1.2. Аналіз систем та моделей ропізнавання

В даний час є багато завдань, які потребують визначення існування або класифікації об’єктів на зображенні. Здатність «розпізнавати» є ключовою ознакою живих істот, а комп’ютерні системи не володіють цією властивістю повністю.

Розглянемо загальні елементи моделі класифікації.

Клас - це група об'єктів із загальними атрибутами. Передбачається, що між об’єктами одного класу існує «подібність». Можна визначити будь-яку кількість класів, більшу за один. Для задачі ідентифікації кількість класів представлена числом S, і кожен клас має свій власний ідентифікатор класу.

Класифікація — це процес присвоєння об’єктам атрибутів класу на основі певних характеристик атрибутів об’єкта. Класифікатор – це пристрій, який приймає набір атрибутів об’єкта як вхідні дані та створює ім’я класу в результаті.

Перевірка - це процес порівняння екземпляра об'єкта з окремою моделлю об'єкта або описом класу.

Зображення — це назва регіону в просторі атрибутів, який представляє багато об’єктів чи явищ матеріального світу. Виняток становить кількісний опис конкретних ознак досліджуваного об'єкта чи явища.

Простір символів — це N-вимірний простір, визначений для даної задачі розпізнавання, де N — фіксована кількість символів, виміряна для будь-якого об’єкта. Вектор символьного простору x, що відповідає предмету задачі розпізнавання, є N-вимірним вектором, а його компоненти (x_1, x_2,..., x_N) є символьними значеннями об’єкта.

Іншими словами, розпізнавання образів можна визначити як віднесення основних даних до певного класу шляхом виділення ключових ознак або атрибутів, що описують дані із загальної кількості неважливих даних.

Приклади класифікаційних завдань:

- Розпізнавання символів;

- Розпізнавання мови;
- Медичний діагноз
- Прогноз погоди;
- Ідентифікація
- Упорядкувати файли тощо.

Вихідним матеріалом зазвичай є зображення з камери. Проблема може бути виражена як отримання вектора атрибутів для кожного класу на зображенні, що розглядається. Цей процес можна розглядати як процес кодування, який приймає значення для кожної функції з простору опису кожного класу.

Якщо розглядати два види об’єктів: дорослі та діти. Ви можете вибрати зріст і вагу як символи. З малюнка видно, що ці два класи утворюють дві незалежні множини, що можна пояснити виділеними символами. Однак не завжди можна вибрати правильно виміряні параметри як ознаки категорії. Наприклад, вибрані налаштування не підходять для створення неперевершених футбольних та баскетбольних курсів.

Друге завдання розпізнавання — витягти ознаки або атрибути з вихідного зображення. Це завдання можна класифікувати як попередню обробку. За роллю розпізнавання мовлення можна розрізняти дзвінкі та глухі символи. Атрибут має бути описом конкретного класу і має бути відкритим атрибутом цього класу. Атрибут, що характеризує відмінність між ними, є міжкласовим атрибутом. Символи, спільні для всіх категорій, не несуть корисної інформації і не є впізнаваними символами. Відбір компетенцій є одним із найважливіших завдань, пов’язаних із розробкою системи розпізнавання.

Після визначення характеристик необхідно визначити найкращу процедуру визначення класифікації. Розглянемо систему розпізнавання моделі, яка використовується для розпізнавання різних категорій M, позначених m_1, m_2,..., m3. Тоді ми можемо припустити, що простір зображень складається з областей M, і кожна область має точку, що відповідає класу зображень. Тоді задачу

ідентифікації можна розглядати як створення межі категорії поділу M на основі отриманого вектора вимірювання.

Попередня обробка зображення, вилучення символів, вирішення проблеми отримання та класифікації оптимального рішення, як правило, вимагає оцінки кількох параметрів. Це призводить до завдання оцінки параметрів. Зрозуміло також, що у виводі функції може використовуватися додаткова інформація на основі природи класу.

Порівняння об'єктів може базуватися на їх представленні у вигляді векторів вимірювання. Дані вимірювання мають бути виражені в дійсних числах. Тоді подібність вектора опису двох об’єктів можна описати евклідовою відстанню, що зображена на рис.1.1.

$$\|x_1 - x_2\| = \sqrt{\sum_{i=1,d} (x_1[i] - x_2[i])^2}$$

Рисунок 1.1 – формула Евклідової відстані

Де d — розмір вектора шансів.

Існує 3 групи методів розпізнавання образів:

- Порівняння з моделлю. Ця група включає класифікацію за найближчим середовищем і класифікацію за відстанню до найближчого сусіда. Метод ідентифікації структури також може посилатися на групу контролю моделі.
- Статистичні методи. Як випливає з назви, статистичний метод полягає у використанні певної статистичної інформації для вирішення проблеми ідентифікації. Цей метод визначає, чи належить об’єкт до певного класу за ймовірністю. У деяких випадках, якщо властивість об'єкта отримує відповідне значення, необхідно визначити подальшу ймовірність належності об'єкта до певного класу. Прикладом є метод правила Байєса.

- Нейронні мережі. Методи ідентифікації окремих категорій. Різниця полягає в здатності вчитися.

Далі ми вивчимо різні методи для різних груп.

1.1.1. Найближча середня класифікація

Він використовується для класифікації невідомих об’єктів класичним способом моделі розпізнавання, яка виражається як вектор основного символу. Систему розпізнавання на основі функцій можна спроектувати багатьма способами. В результаті вивчення цих векторів передбачення в реальному часі або система прогнозування може бути заснована на будь-якій моделі.

Простий алгоритм класифікації полягає у використанні вектора очікування класу (середнього) для групування опорних даних класу. Алгоритм зображено на рис.1.2.

$$\bar{x}_i = \frac{1}{n_i} \sum_{j=1,n_i} x_{i,j}|$$

Рис.1.2– простий алгоритм класифікації

Де x(i, j) — j-й опорний символ i-го типу, а n_j — номер i-го типу опорного вектора.

Якщо невідомий об’єкт ближче до математичного вектора очікування класу і, ніж до математичних векторів очікування інших класів, то невідомий об’єкт належить до і-го класу. Цей метод підходить для задачі, що точки кожного класу дуже компактні і віддалені від точок інших класів.

Рисунок 1.3 – схема розміщення класів розпізнавання

Наприклад, якщо клас має складнішу структуру, виникають труднощі, як показано на малюнку. При цьому друга категорія поділяється на дві області, що не перекриваються, що важко описати середнім значенням. Крім того, занадто витягнута модель 3-го класу має велике значення координат x_2, яке ближче до середнього значення 1-го класу, ніж модель 3-го класу. Цю схему зображено на рис.1.4:

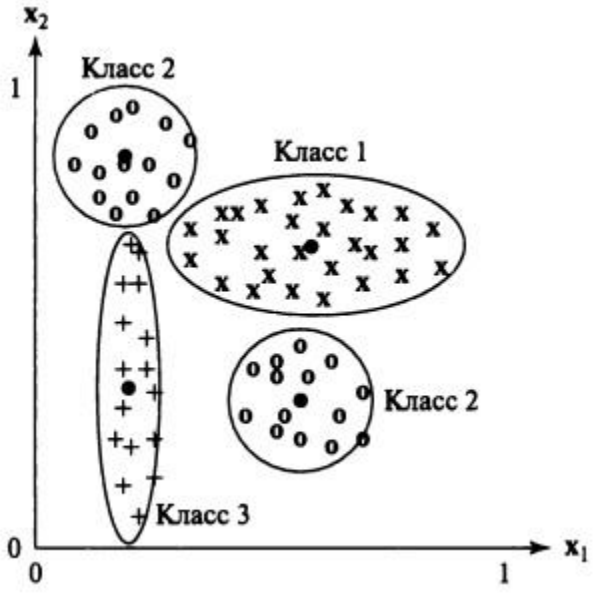


Рисунок 1.4 *– розташування класів більш складним варіантом*

У деяких випадках описану проблему можна вирішити, змінивши розрахунок відстані.

Розглянемо опис «розподілу» класу value-σ_i, для кожного координатного напрямку i. Стандартне відхилення дорівнює квадратному кореню дисперсії. Масштабована евклідова відстань між вектором x і вектором математичного очікування x_c дорівнює (1.2.1.4):

$$\|x - x_c\| = \sqrt{\sum_{i=1,d} \left(\frac{x[i] - x_c[i]}{\sigma_i} \right)^2}$$

Рисунок 1.5 - Масштабована евклідова відстань між вектором x і вектором очікування

Ця формула відстані зменшує кількість помилок класифікації, але насправді більшість проблем не можуть бути представлені таким простим класом.

1.2.2. Сортувати за відстанню до найближчого сусіда

Інший метод класифікації полягає в тому, щоб присвоїти невідомий вектор знаків x класу, вектор якого найближче до однієї моделі. Це правило називається правилом найближчого сусіда. Навіть якщо клас має складну структуру або відповідність класу, класифікація найближчого сусіда може бути більш ефективною.

Для цього методу не потрібно робити припущення щодо моделі розподілу різницевого вектора в просторі. Алгоритм використовує лише інформацію про відомі еталонні моделі. Метод рішення заснований на обчисленні відстані x до кожного зразка в базі даних і знаходженні мінімальної відстані. Переваги цього методу очевидні:

Ви можете в будь-який момент додати нові шаблони до бази даних;

Структури даних дерева та сітки можуть зменшити кількість обчислених відстаней.

Крім того, якщо шукати основу для k замість сусідів, рішення буде кращим. Тоді для k> 1 це дає найкращий приклад розподілу вектора в d-вимірному просторі. Однак ефективне використання k залежить від того, чи є достатня кількість у кожній області простору. Якщо класів більше двох, то прийняти правильне рішення складніше.

1.3. Аугментація даних

Аугментація даних – це методика створення додаткових навчальних датасетів із наявних вхідних даних. Для досягнення хороших результатів глибокі мережі повинні навчатися дуже великому обсязі даних. Отже, якщо початковий навчальний набір містить обмежену кількість зображень, необхідно виконати аугментацію, щоб покращити результати моделі.

Існує багато варіантів аугментації. Найпопулярнішими є такі: відображення по горизонталі (horizontal flip), випадкове кадрування (random crop) та зміна кольору (color jitter). Можна застосовувати різні комбінації, наприклад одночасно виконувати поворот і випадкове масштабування. Крім того, можна варіювати величину насиченості (saturation) та значення (value) всіх пікселів (компоненти S та V колірного простору HSV). Зокрема, можна звести ці компоненти в ступінь з інтервалу від 0,25 до 4, помножити їх на коефіцієнт від 0,7 до 1,4, або додати до них величину з інтервалу від –0,1 до 0,1. Також можна додати величину інтервалу від –0,1 до 0,1 до величини тону (hue) всіх пікселів (компонента H колірного простору HSV). Аналогічні перетворення можна застосувати до фрагментів зображень.

У 2012 році при навчанні знаменитої мережі Alex-Net був використаний спеціальний варіант методу головних компонентів (principal component analysis, PCA). Суть підходу полягає у зміні інтенсивності RGB-каналів навчальних зображень. Спочатку RGB-значення пікселів всіх навчальних зображень застосовується PCA. Потім в межах кожного зображення до значення RGB кожного пікселя ($I_{xy} = [I_{xy}^R, I_{xy}^G, I_{xy}^B]^T$)додається наступна величина:

$[p_1, p_2, p_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$,

де pi і λi є відповідно i-й власний вектор і i-е власне значення коваріаційної матриці RGB-значень пікселів розмірності 3×3, а αi є випадковою величиною з нормального розподілу з математичним очікуванням 0 і середньоквадратичним відхиленням 0,1. Зверніть увагу, кожне значення αi генерується один раз для всіх

пікселів цього навчального зображення. Коли модель знову зустріне це навчальне зображення, вона згенерує інше випадкове значення αi для аугментації даних.

Автори публікації стверджують, що завдяки цьому підходу, модель отримує можливість ідентифікувати об'єкти незалежно від яскравості та кольору освітлення. У рамках змагання ImageNet 2012 цей метод аугментації дозволив зменшити помилку більш ніж на 1%.

Використання методів аугментації даних показало себе добре завдання класифікації зображень. Попри це, досліджень впливу аугментації на точність моделей розпізнавання був. Враховуючи ресурси на розмітку зображень, для завдання розпізнавання об'єктів аугментація може бути кориснішою.

Дослідники з'ясували, що методи аугментації даних для класифікації зображень можуть бути корисні завдання розпізнавання. Але такі методи дають обмежений приріст у точності моделі. Тому фокусом дослідження було визначити, як вивчені методи аугментації покращують узагальнюючу здатність моделі. У цьому модифікується лише тренувальний набір даних, а тестовий залишається незмінним. Запропонований метод на маленьких датасетах постає як спосіб регуляризації і не дозволяє перевчитися на малопредставлених класах.

Дослідники спочатку склали список можливих модифікацій, що проводяться над зображеннями перед навчанням моделі з розпізнавання. Усього функцій вийшло 22. Ці функції були реалізовані на Tensorflow.

Виділені функції поділяються на такі типи:

- операції з кольором (підкрутити колірні канали, збільшення контрасту чи яскравості);
- геометричні операції (повернути зображення, відобразити зображення тощо);
- операції з об'єктами (повернути об'єкт, відобразити об'єкт тощо)

Дослідники використовують комбінацію з RNN та методу навчання з підкріпленням для пошуку оптимальної модифікації зображення.

Експерименти на датасеті COCO показують, що оптимізована аугментація даних збільшує точність більш ніж на 2.3 пункти в порівнянні з state-of-the-art. Як метрика - mAP (mean Average Precision). Важливо, що вивчена аугментація COCO працює для будь-якого іншого датасета з розпізнавання об'єктів. Наприклад, аугментація COCO покращує базову модель для PASCAL-VOC на 2.7 mAP. Результати показують, що вивчена аугментація працює краще, ніж методи регулювання для розпізнавання об'єктів.

2. ОГЛЯД АРХІТЕКТУРИ ТА ВИБІР МЕТОДІВ РЕАЛІЗАЦІЇ

2.1. Постановка завдання

Враховуючи всі переваги та недоліки вищезгаданих систем, необхідно звернути увагу на головну відмінність розробленої системи від запропонованих.

Розширення даних перевірки також підвищує точність системи. Завдяки великій кількості вибірок вхідних даних система може краще виявляти помилки відповідно до заданих стандартів. Тестовий режим дозволяє перевірити якість розпізнавання системи.

Виходячи з вищесказаного, необхідно спроектувати інтелектуальну систему для визначення стану трубопроводу на основі отриманого зображення.

Залежно від вхідних даних така система отримує масив форм труб з різною структурою та станами з явними чи прихованими дефектами стану. На зображені показано внутрішню частину трубопроводу, забруднення, стан стінок труби, стан стику тощо. Обробка зображення виконується шляхом аугментації даних.

Також слід зазначити, що система повинна працювати в тестовому режимі. При роботі ІС у тестовому режимі, тобто в режимі прямого запуску, необхідно визначити, що тестова реалізація зображення розпізнавання належить до однієї з категорій розпізнавання за ключовими правилами, сформульованими на етапі навчання з поданого алфавіту.

2.2.Класифікатор зображень

Розпізнавання зображення відноситься до завдання введення зображення в нейронну мережу та присвоєння будь-якої мітки для цього зображення. Мітка, яку виводить мережа, відповідатиме заздалегідь певному класу. Може бути присвоєно як кілька класів, і лише один. Якщо існує лише один клас, зазвичай застосовується термін «розпізнавання», тоді як завдання розпізнавання кількох класів часто називається «класифікацією».

Підмножина класифікацій зображень є вже визначенням об'єктів, коли певні екземпляри об'єктів ідентифікуються як такі, що належать до певного класу, наприклад, тварини, автомобілі або люди.

Яскравим прикладом такої класифікації є рішення найпоширенішої капчі — ReCaptcha v2 від Google, де з набору картинок необхідно вибрати ті, що належать до вказаного в описі класу.

Щоб виконати розпізнавання або класифікацію зображень, нейронна мережа повинна витягти ознаки. Ознаки - це елементи даних, які становлять максимальний інтерес і які будуть передаватися по нейромережі. У конкретному випадку розпізнавання зображень такими ознаками є групи пікселів, такі як лінії та точки, які мережа аналізуватиме на наявність патерну.

Розпізнавання ознак (або вилучення ознак) - це процес отримання відповідних ознак з вхідного зображення, щоб їх можна було проаналізувати. Багато зображень містять анотації або метадані, які допомагають нейромережі знаходити відповідні ознаки. Процес розпізнавання признаків зображено на рис.2.1:

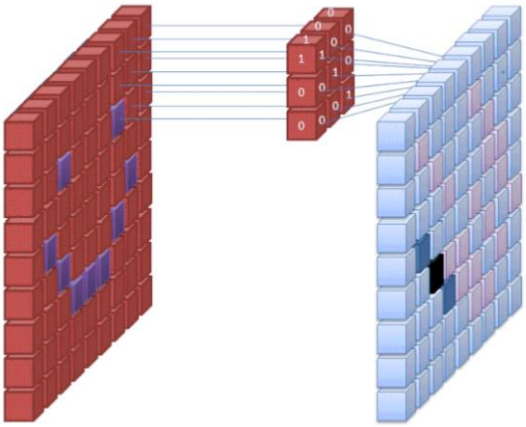


Рисунок 2.1 – розпізнавання признаків за допомогою фільтрів

Перший шар нейронної мережі приймає всі пікселі у зображенні. Після того, як усі дані введені в мережу, до зображення застосовуються різні фільтри, що формують розуміння різних частин зображення. Це витяг ознак, що створює «карти ознак».

Цей процес отримання ознак із зображення виконується за допомогою «згорткового шару», і згортка просто формує уявлення частини зображення. Саме з цієї концепції згортки ми отримуємо термін "Згорткова нейронна мережа" (Convolutional Neural Network, CNN) - тип нейронної мережі, що найчастіше використовується в класифікації та розпізнаванні зображень.

Створення карт ознак можна уявити як процес піднесення ліхтарика до зображення у темній кімнаті. Коли ви ковзаєте променем на картинці, ви дізнаєтеся про особливості зображення. Фільтр це те, що мережа використовує для формування уявлення про зображення, і в цій метафорі світло від ліхтарика є фільтром.

Ширина променя ліхтарика визначає розмір фрагмента зображення, який ви переглядаєте за один раз, і нейронні мережі мають аналогічний параметр - розмір фільтра. Розмір фільтра впливає на те, скільки пікселів перевіряється за один раз. Загальний розмір фільтра, використовуваного CNN, дорівнює 3, і він охоплює як висоту, так і ширину, тому фільтр перевіряє область пікселів 3 х 3. Цей процес зображено на рис.2.2:

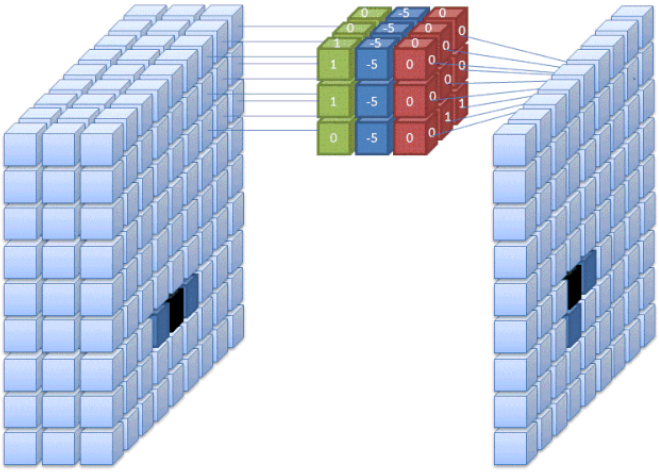


Рисунок 2.2 – згорткова мережа зі слоями признаків

У той час як розмір фільтра покриває висоту та ширину фільтра, глибина фільтра також має бути вказана.

Але як 2D зображення може мати глибину? Справа в тому, що цифрові зображення відображаються у вигляді висоти, ширини і деякого значення RGB, яке визначає колір пікселя, тому «глибина», що відстежується, - це кількість колірних каналів, які має зображення. Зображення в градаціях сірого (не кольорові) мають лише 1 кольоровий канал, у той час як кольорові зображення мають глибину 3 канали.

Все це означає, що для фільтра розміром 3, застосованого до повнокольорового зображення, підсумкові розміри цього фільтра будуть 3 x 3 x 3. Для кожного пікселя, охоплюваного цим фільтром, мережа множить значення фільтра на значення самих пікселів, щоб отримати числове уявлення цього пікселя . Потім цей процес виконується для всього зображення, щоб отримати повне уявлення. Фільтр переміщається по решті зображення відповідно до параметра, званого «крок», який визначає, на скільки пікселів повинен бути переміщений фільтр після того, як він обчислить значення у своїй поточній позиції. Нормальний розмір кроку для CNN - 2.

Кінцевим результатом усіх цих розрахунків є карта ознак. Цей процес зазвичай виконується з кількома фільтрами, які допомагають зберегти складність зображення.

Після того, як карта ознак зображення була створена, значення, що представляють зображення передаються через функцію активації або шар активації. Функція активації набуває цих значень, які завдяки згортковому шару знаходяться в лінійній формі (тобто просто список чисел) і збільшує їх нелінійність, оскільки самі зображення є нелінійними.

Типовою функцією активації, що використовується для досягнення цієї мети, є випрямлена лінійна одиниця (ReLU), хоча є деякі інші функції активації, які також іноді використовуються (ви можете прочитати про них тут).

Після активації дані відправляються через шар, що об'єднує. Об'єднання "спрощує" зображення: бере інформацію, яка представляє зображення, і стискає її. Процес об'єднання в пул робить мережу гнучкішою і здатною краще розпізнавати об'єкти та зображення на основі відповідних функцій.

Коли ми дивимося на зображення, нас зазвичай хвилює не вся інформація (наприклад, що на задньому плані зображення), а лише ознаки, які нас цікавлять — люди, тварини тощо.

Аналогічно, що об'єднує шар CNN позбавиться від непотрібних частин зображення, залишивши тільки ті частини, які він вважає релевантними, в залежності від заданого розміру об'єднуючого шару.

Оскільки мережа повинна приймати рішення щодо найбільш важливих частин зображення, розрахунок йде на те, що вона вивчить тільки ті частини зображення, які дійсно є суть об'єкта, що розглядається. Це допомагає запобігти “перенавченню” — коли мережа надто добре вивчає всі аспекти навчального прикладу і вже не може узагальнювати нові дані, оскільки враховує нерелевантні відмінності.

Існують різні способи поєднання значень, але найчастіше використовується максимальне об'єднання. Максимальне об'єднання має на увазі взяття максимального значення серед пікселів у межах одного фільтра (у межах одного фрагмента зображення). Це відсіює 3/4 інформації за умови використання фільтра розміром 2 x 2.

Максимальні значення пікселів використовуються для врахування можливих спотворень зображення, а кількість параметрів (розмір зображення) зменшено, щоб контролювати перенавчання. Існують інші принципи об'єднання, такі як середнє чи сумарне об'єднання, але вони використовуються не так часто, оскільки максимальне об'єднання дає велику точність.

Останні шари нашої CNN – щільно пов'язані шари – вимагають, щоб дані були представлені у формі вектора для подальшої обробки. З цієї причини дані необхідно «звести докупи». Для цього значення стискаються у довгий вектор або стовпець послідовно впорядкованих чисел.

У той час як розмір фільтра покриває висоту та ширину фільтра, глибина фільтра також має бути вказана.

Кінцеві шари CNN є щільно пов'язані шари або штучну нейронну мережу (Artificial neural networks (ANN))(рис.2.3). Основною функцією ANN є аналіз вхідних ознак та об'єднання їх у різні атрибути, які допоможуть у класифікації. Ці шари утворюють набори нейронів, які представляють різні частини об'єкта, що розглядається, а набір нейронів може являти собою, наприклад, висячі вуха собаки або почервоніння яблука. Коли достатня кількість цих нейронів активується у відповідь вхідне зображення, воно буде класифіковано як об'єкт.

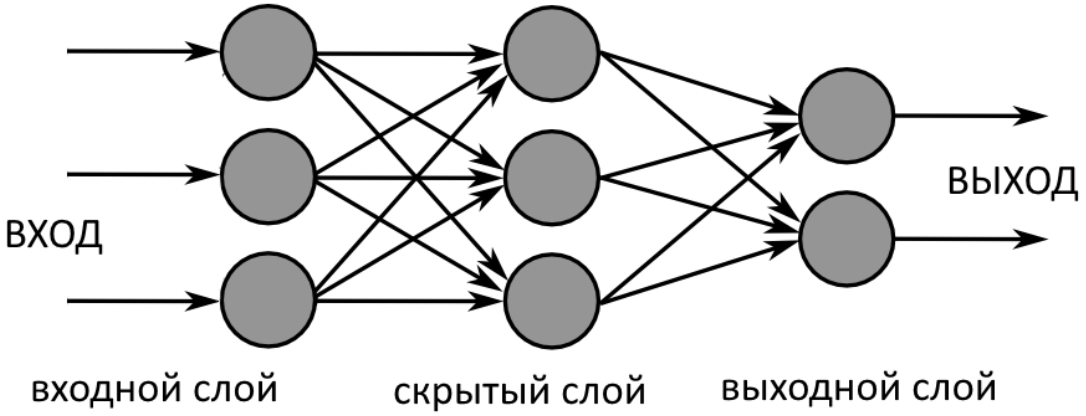


Рисунок 2.3 – схема шарів CNN

Помилка або різниця між розрахованими значеннями та очікуваним значенням у навчальному наборі розраховується за допомогою ANN. Потім мережа піддається методу зворотного розповсюдження помилки, де розраховується вплив даного нейрона на нейрон у наступному шарі, а потім його вплив (вага) коригується. Це зроблено для оптимізації продуктивності моделі. Цей процес повторюється знову і знову: так мережа навчається на даних та вивчає зв'язки між вхідними ознаками та вихідними класами.

Нейрони середніх пов'язаних шарах виводитимуть двійкові значення, які стосуються можливим класам. Якщо у вас є чотири різні класи (скажімо, собака, машина, будинок і людина), нейрон матиме значення "1" для класу, який, як він вважає, представляє зображення, і значення "0" для інших класів.

Кінцевий повністю пов'язаний шар, отримавши вихідні дані попереднього шару, надає можливість кожному з класів у межах одиниці (у сукупності). Якщо категорії «собака» надано значення 0,75 — це означає 75% ймовірність того, що зображення є собакою.

2.3.Особливості ІС в режимі екзамену

Для розуміння, в деяких джерелах літератури режимом екзамену називають також режим висновків (inference mode), а режим навчання – режимом передбачення (prediction mode).

Висновок і передбачення — два терміни, які часто плутають, частково, можливо, тому, що вони не виключають один одного. Обидва містять фрагменти "Про що мені говорять дані?". Насправді, в результаті передбачень виникає багато запитань на висновок: наприклад, ви можете передбачити, як вхідні змінні X, Y і Z впливають на вихідну змінну B. Тоді ви можете зробити висновок, наскільки важливі (чи не важливі) окремі вхідні змінні є. Незважаючи на ці типи зв’язків, ці два терміни відрізняються кількома ключовими ознаками. Рисунок 2.4 окреслює основні відмінності:

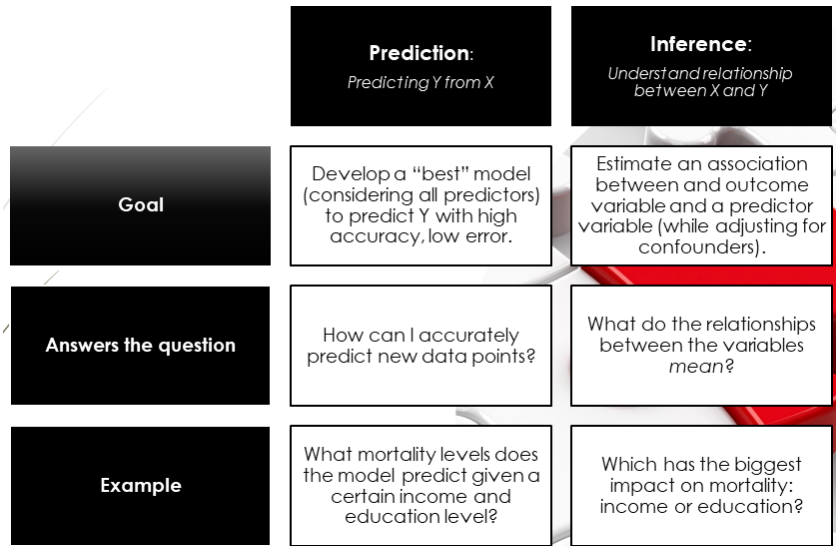


Рисунок 2.4 - основні відмінності режимів навчання та екзамену

Прогноз — це оцінка значення змінної відповіді на основі провісників або вхідних змінних. Передбачення можна назвати як відображення набору значень предиктора до їх відповідного результату.

Висновок – це розуміння зв’язку між реакцією та провісниками. Йдеться про розуміння наступного:

- Які предиктори пов’язані зі змінною відповіді?
- Як змінюється значення змінної відповіді за величиною по відношенню до змінних провісників?
- Як змінюється значення змінних відповіді за знаком (позитивним чи негативним) по відношенню до змінних провісників?
- Чи можна описати зв’язок за допомогою лінійного рівняння, чи зв’язок між предиктором і змінною відповіді є більш складним?

Висновок можна далі класифікувати як наступне:

Однофакторний висновок (зв'язок між одним предиктором і відповіддю) - цей тип висновку допомагає нам зрозуміти, як змінюється значення однієї змінної предиктора щодо зміни іншого предиктора або набору предикторів.

Двовимірний висновок (зв’язок між відповіддю та двома змінними-провісниками) - цей тип висновку допомагає зрозуміти, як змінюється значення змінної відповіді щодо зміни двох змінних-провісників.

Багатоваріантний висновок (зв’язок між відповіддю та більш ніж двома змінними предикторів) - цей тип висновку допомагає зрозуміти, як змінюється значення змінної відповіді щодо зміни кількох предикторів.

У той час як передбачення використовується для оцінки значення змінної відповіді, умовивод використовується для розуміння зв’язку між предикторами та змінними відповіді.

Коли метою є передбачення, можна вибрати використання нелінійних методів моделей машинного навчання. Ці нелінійні методи не піддаються інтерпретації.

Коли метою є висновок, можна вибрати для моделювання використання лінійних методів (наприклад, лінійної регресії, регресії Лассо тощо). Лінійні моделі дозволяють робити відносно прості висновки, які можна інтерпретувати, але можуть не давати таких точних прогнозів, як деякі інші підходи.

На рисунку 2.5 представлені різні алгоритми машинного навчання (лінійні та нелінійні) у порівнянні з необхідністю інтерпретації та гнучкості. У разі високої гнучкості метою є передбачення, але інтерпретація стає низькою. У разі високої інтерпретації метою є умовивод. Однак точність роботи моделі страждає.

Рисунок 2.5 – залежність алгоритмів навчання від гнучкості та інтерпретації залежностей системи

Прогнозування та висновок машинного навчання – це два різні аспекти машинного навчання. Прогноз – це здатність точно передбачити змінну відповіді, тоді як умовивод має справу з розумінням зв'язку між змінними-провісником і змінними відповіді. Різницю в моделях прогнозування та висновків можна побачити на таких прикладах, як прогнозування успіху маркетингової кампанії або розуміння того, як медіа впливають на продажі на рекламні кампанії тощо. Ви можете використовувати лінійну регресію для моделі висновку, тоді як нелінійні методи найкраще працюють, коли ваша мета є прогнозом.

2.4.Вибір підходу аугментації даних для навчання класифікатора

Проблема з невеликими наборами даних полягає в тому, що моделі, що навчаються з ними недостатньо ефективно узагальнюють дані для навчання і тестовий набір зображень. Отже, ці моделі страждають від проблеми надмірного прилягання. Отже потрібно обрати метод аугментації, що найефективніше працюватиме в наших умовах.

Розглянемо два різних підходи до аугментації даних. Перший підхід — генерувати доповнені дані раніше навчання класифікатора. Наприклад, ми будемо застосовувати GAN і основні перетворення для створення більшого набору даних. Всі зображення подаються в мережу під час навчання та під час тестування, для перевірки використовуються лише оригінальні зображення.

Другий підхід намагається навчитися аугментації за допомогою попередньої нейронної мережі. Під час навчання ця нейронна мережа займає у двох випадкових зображеннях із навчального набору та виходів а єдине «зображення», щоб це зображення відповідало стилю або у контексті із заданим зображенням із навчального набору. Це вихід, який представляє створене доповнене зображення мережею, подається у другу класифікуючу мережу разом із вихідними даними навчання. Втрата навчання є потім розповсюджується назад, щоб тренувати збільшуючі шари мережі, а також класифікаційні рівні мережі. Під час тестування запускаються зображення з набору перевірки або тестування тільки через мережу класифікації. Мотивація полягає у навчанні моделі для визначення найкращих доповнень заданий набір даних. Решта цього розділу буде розглянута детальна інформація про трюки збільшення даних, які ми спробували.

2.4.1. Традиційні трансформації

Традиційні перетворення полягають у використанні комбінації афінних перетворень для маніпулювання навчальними даними. Для кожного вхідного зображення ми створюємо «дублікат» зображення що зміщується,

збільшується/зменшується, повертається, перевертається, спотворено або затінені відтінком. І зображення, і дублікат подаються нейронну мережу. Для набору даних розміром N ми створюємо набір даних Розмір 2N. Приклади традиційних перетворень зображено на рис.2.6:

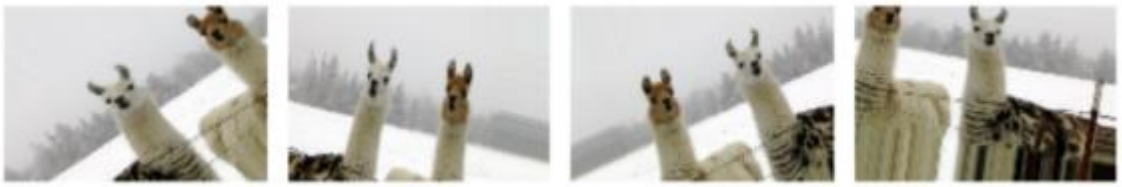


Рисунок 2.6 – приклади традиційних перетворень зображень

2.4.2. Генеративні змагальні мережі

Generative Adversarial Nets (GAN) — це потужна техніка для створення нових зображень без нагляду для навчання. Вони також виявилися надзвичайно ефективними у багатьох завданнях генерації даних, наприклад, у створенні нових абзаців. GAN використовувалися для передачі стилю, наприклад для перенесення зображень з одного налаштування в інше (CycleGAN). Крім того, GAN були ефективними навіть із відносно невеликими наборами даних, виконуючи методи навчання передачі.

Крім того, вони показали, що вони надзвичайно добре збільшують набори даних, наприклад, збільшують роздільну здатність вхідних зображень.

Для кожного вхідного зображення ми вибираємо зображення стилю з підмножини 6 різних стилів: Сезанн, Поліпшення, Моне, Укійо, Ван Гог і Зима. Створюється стильове перетворення вихідного зображення. Оригінальний і стильний образ є годують, щоб тренувати мережу. Приклад перетворень зображено на рис. 2.7:

Рисунок 2.7 – перетворення за технікою GAN

2.4.3. Навчання аугментації

На етапі навчання мережа складається з двох частин. Мережа аугментації отримує два зображення з той самий клас, що і вхідне зображення, і повертає шар такого ж розміру, як одне зображення. Цей шар розглядається як «доповнене» зображення. Доповнене зображення, а також вихідне вхідне зображення потім передаються до другої мережі, класифікаційна мережа. Втрата класифікації в кінці мережі є перехресною втратою ентропії на сигмоїдах бали занять. Додаткова втрата обчислюється в кінці мережі розширення, щоб регулювати, наскільки доповнене зображення має бути схожим на вхідне зображення.

2.4.4. Порівняння підходів аугментації та вибір оптимального

Щоб перевірити ефективність різних аугментацій, скористаємося порівнянням, проведеним науковцями Стенфордського університету. Вони провели 10 експериментів над даними image-net. Результати експериментів наведено в таблицях нижче. Усі експерименти проводяться протягом 40 епох зі швидкістю навчання 0,0001 за допомогою оптимізації Адама. Найвища точність випробування у всі епохи повідомляється як найкраща оцінка. Отримавши доповнені зображення, ми годуємо їх в нейронну мережу, яка виконує класифікацію. Ми називаємо цю нейронну мережу SmallNet, оскільки вона має лише 3 згорткові шари в парі з пакетною нормалізацією та максимальним пулом в 2 повністю з’єднаних шари. Виходом є матриця оцінок для ваг для кожного класу. Шари мережі детально описано нижче, хоча конкретна мережа не дуже важлива. Будь-яка мережа, яка може надійно передбачити класів достатнья для дослідження. Отже, цю сітку можна замінити на VGG16 з тонким налаштуванням на повністю підключений і останній шари згортки, щоб забезпечити достатню підготовку.

Щоб визначити втрату, нам потрібна комбінація a втрат класифікації, Lc, і втрата збільшення, La. Контент втрати — це середня квадратична помилка між доповненим зображенням A та цільовим зображенням T, де D — довжина зображення A і T. (Формула 2.1):

$$L_a^{content} = \frac{1}{D^2} \sum_{i,j} (A_{ij} - T_{ij})$$

Формула 2.1 – Похибка втрати контенту

Втрата стилів – це втрата вмісту на матриці Грама доповнене зображення A та цільове зображення T. Матриця Грама карти ознак F визначено нижче. Ми застосовуємо матрицю Грама до необроблених зображень Тоді втрати визначаються нижче (Формула 2.2):де C — кількість каналів

$$L_a^{style} = \frac{1}{C^2} \sum_{i,j} (G_{ij}^A - G_{ij}^T)$$

Формула 2.2 – Формула втрати вмісту

, де C — кількість каналів.

Моделі навчаються з коефіцієнтом навчання 0,0001 с Оптимізацією ADAM. Для навчання використовувався Tensorflow на GPU моделі, хоча вони досить швидко тренуються на ЦП (близько 30 секунд на епоху).

Результати експериментів представлені в таблицях нижче. Зафіксовано найкращу точність перевірки за 40 епох. У таблиці 1 показані всі експерименти з даними собак і золотих рибок. У таблиці 2 наведено результати експериментів на зображення собак та кішок. Відповідно, чим вищий коефіцієнт, тим більша точність розпізнавання зображення(Рис.2.8):

Dogs vs Goldfish	
Augmentation	Val. Acc.
None	0.855
Traditional	0.890
GANs	0.865
Neural + No Loss	**0.915**
Neural + Content Loss	0.900
Neural + Style	0.890
Control	0.840

Dogs vs Cat	
Augmentation	Val. Acc.
None	0.705
Traditional	**0.775**
GANs	0.720
Neural + No Loss	0.765
Neural + Content Loss	0.770
Neural + Style	0.740
Control	0.710

Рисунок 2.8 – таблиці порівнянь підходів аугментації

З результатів дослідження можна зробити висновок, що для нашої технології найбільш оптимальною є аугментація стандартними методами, так як вона дозволить максимізувати вибірку тренувальних та тестових даних, а також потребує найменше ресурсів для роботи з нею.

3. РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ ТА ПРИКЛАД ОБЧИСЛЕННЯ

3.1.Проектування системи класифікатора зображень

На рисунку 3.1 зображено діаграму потоків даних системи детектування стану трубопроводів:

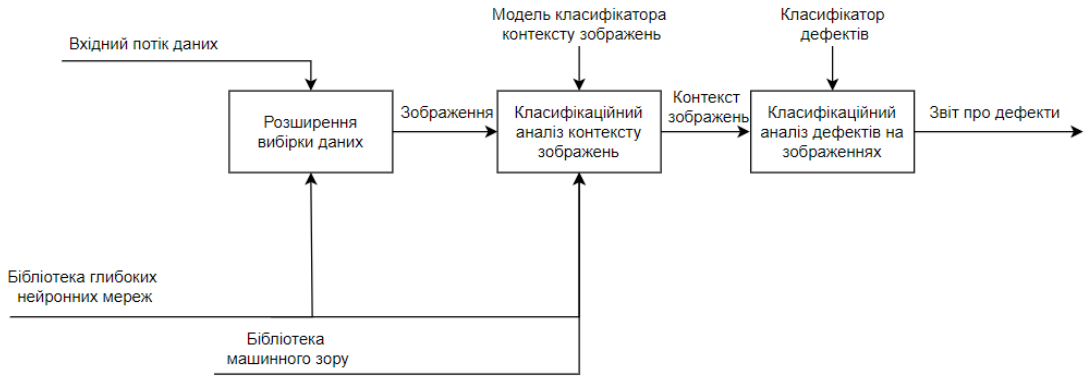


Рисунок 3.1 - діаграма потоків даних системи детектування стану трубопроводів

На вхід система приймає потік даних, що являє собою масив зображень трубопроводів. В залежності від етапів функціонування технології, дані можуть бути як тренувальними, тестовими, так і валідаційними, на яких перевірятиметься точність та якість отриманої системи. Далі модель класифікатору контексту зображень аналізує вхідні дані на предмет зображення труб та трубопроводів.

Далі потік даних обробляється бібліотекою нейронних мереж для розширення вибірки. Розширена вибірка зображень обробляється моделю класифікатора контексту зображень на аналіз контексту зображень.

Після виявлення класифікатором контексту зображень переходимо вже на блок класифікаійного аналізу дефектів, де система розпізнаватиме дефекти по зонам інтересу. Таким чином класифікатор дефектів формуватиме вихідний масив дефектів у формі звіту.

3.2.Формування вибірки зображень

Набір даних Sewer-ML містить 1,3 мільйона зображень із 75 618 відео, зібраних від трьох датських комунальних компаній за дев’ять років. Усі відео були анотовані ліцензованими інспекторами каналізації відповідно до данського стандарту перевірки каналізації Fotomanualen. Вибірка складається з 17 анотованих класів дефектів. Повний перелік матеріалів зображено на рисунку 3.1:

Type	Training	Validation	Test	Total
Normal	552,820	68,681	69,221	690,722
Defective	487,309	61,365	60,805	609,479
Total	1,040,129	130,046	130,026	1,300,201

Рисунок 1.1 - кількість даних, розбитих за класифікаціями

Анотації зображень були вилучені на основі набору евристичних правил, що дають змогу методам класифікації з кількома мітками, і розділені на окремі розділи навчання, перевірки та тестування. Набір даних складається з майже 50-50% зображень без дефектів та з дефектами принаймні одного класу дефектів. Приклади зображень з датасету наведено на рисунку 3.2:

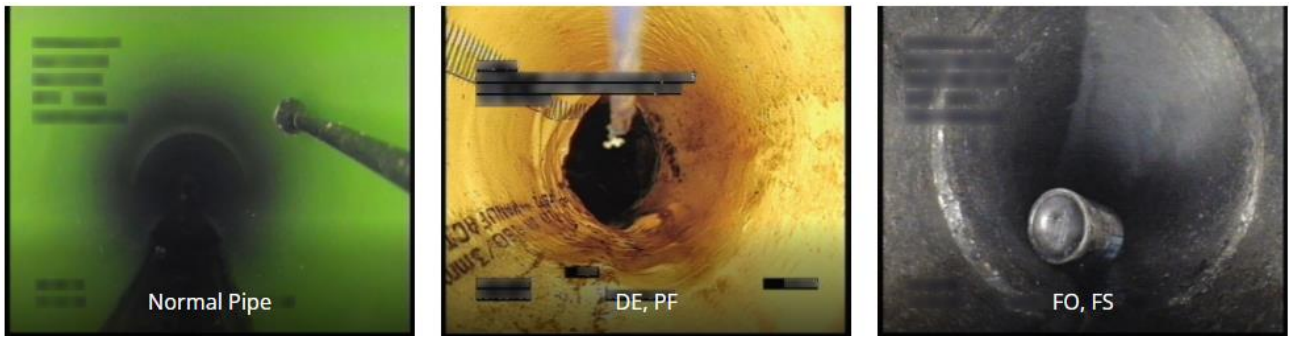


Рисунок 3.2 – приклад зображень з датасету

Для визначення признаків зношеності труб обрано наступні критерії:

- Наявність деформації труби;
- Наявність розломів;
- Наявність засмічення в трубі;
- Деформація стику труб;

Приклади дефектів трубопроводів зображено на рис. 3.3:

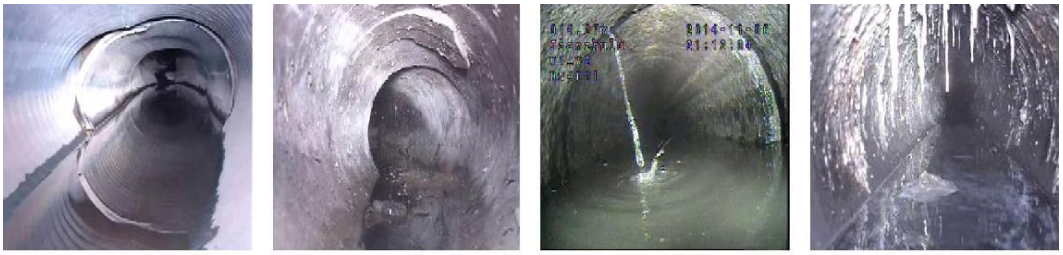


Рисунок 3.3 – приклади дефектів трубопроводу

3.3.Програмна реалізація класифікатора

В ході реалізації класифікатора було використано готові модулі та бібліотеки для машинного навчання. Повний перелік наведено в таблиці 3.1

Таблиця 3.1 – Опис використаних бібліотек та фреймворків

Назва	Опис
OpenCV	OpenCV — це комп'ютерна бібліотека з відкритим вихідним кодом, ліцензована BSD. Містить інтерфейси Python, C++ та Java з високою обчислювальною ефективністю. Існують методи обробки зображень та відео
NumPy	Бібліотека розширення Python. Додає підтримку масивів, багатовимірних масивів і колекцій. Містить масив математичних функцій.
TensorFlow	Безкоштовна бібліотека з відкритим кодом для машинного навчання. Її можна використовувати для різноманітних завдань, але особлива увага приділяється вивченню глибинних нейронних мереж. Алгоритм бібліотеки може працювати на різних платформах (CPU, GPU, TPU). TensorFlow підтримує C++, python і мови програмування, розроблені командою Google Brain від Google AI. Повністю підтримує java-скрипт, java та Google Colab..

TensorBoard	Набір інструментів для візуалізації та експериментів з машинним навчанням. Він має такі характеристики: -Візуалізація показників навчання моделі, таких як точність і втрати; -Візуалізація шару моделі; -Друк малюнків; -Створення гістограм ваг, переміщень та інших тензорів;
MobileNetV2	MobileNet — це клас CNN з відкритим кодом Google. MobileNet використовує згортки, які можна розділити по глибині. Це значно зменшує кількість параметрів у порівнянні з мережею зі звичайними згортками з однаковою глибиною в сітках. Це призводить до легких глибоких нейронних мереж.

Імплементація була розроблена в середовищі Google Colab.

Colab Notebook — це безкоштовне інтерактивне хмарне середовище кодування від Google. Він дозволяє працювати з даними в режимі онлайн. Це середовище підходить для:

- Роботи з Big Data;
- Аналітики даних;
- Обробки даних;
- Роботи зі штучним інтелектом;

В основі «Colaboratory» лежить блокнот Jupyter для роботи на Python, тільки з базою на Google Drive, а не на комп'ютері. Функціонал так само підтримує текст, формули, зображення, HTML-розмітку тощо. Тобто ви можете програмувати на Python і не завантажувати купу бібліотек, не перевантажувати машину і не переживати, що місце на жорсткому диску ось-ось закінчиться. Єдина умова – мати обліковий запис Google.

Класифікатор побудований з використанням наступних модулів:

- Багатошарова нейронна мережа, що складається з 4 повноцінних шара та 4 нейронів на виході, що дорівнює по 1 на кожен клас розпізнавання(рис 3.4):

```
model = tf.keras.models.Sequential([
    # Note the input shape is the desired size of the image 150x150 with 3 bytes color
    # This is the first convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    # The second convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The third convolution
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The fourth convolution
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # Flatten the results to feed into a DNN
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.5),
    # 512 neuron hidden layer
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(4, activation='softmax')
])
```

Рисунок 3.4 – реалізація багатошарової нейронної мережі

- Компілятор нейронної мережі (рис. 3.5):

```
model.summary()

model.compile(loss = 'categorical_crossentropy', optimizer='rmsprop', metrics=['accuracy'])
```

Рисунок 3.5 – компілятор нейронної мережі

- Функція навчання нейронної мережі (рис 3.6):

```
history = model.fit(train_generator, epochs=25, steps_per_epoch=20, validation_data = validation_generator, verbose = 1, validation_steps=3)

model.save("rps.h5")
```

Рисунок 3.6 – функція навчання нейронної мережі

3.4.Реалізація підходу аугментації

Для аугментації даних використані такі методи перетворення оригінальних вхідних зображень: оберти по вертикалі та горизонталі, невеликі повороти та здвиги, розмиття, накладання областей гаусового шуму випадкового розміру, шумів типу «сіль та перець». Програмно аугментацію реалізовано за допомогою класу ImageDataGenerator.

ImageDataGenerator створює пакети тензорних даних зображень із змінами у реальному часі. Дані будуть зациклені пакетами. Ми використовуємо метод flow_from_directory, який є генератором, який визначає шлях до батьківського каталогу, що містить різні класи даних зображень, і генерує пакети зображень для передачі в ImageDataGenerator. Це очікувана структура каталогів методу flow_from_directory (рис. 3.5).

```
TRAINING_DIR = "/tmp/rps/"
training_datagen = ImageDataGenerator(
      rescale = 1./255,
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest')

VALIDATION_DIR = "/tmp/rps-test-set/"
validation_datagen = ImageDataGenerator(rescale = 1./255)
```

Рисунок 3.5 – програмна реалізація аугментації даних

Приклади використаних маніпуляцій із зображеннями для розширення набору можна побачити на рис. 3.6:

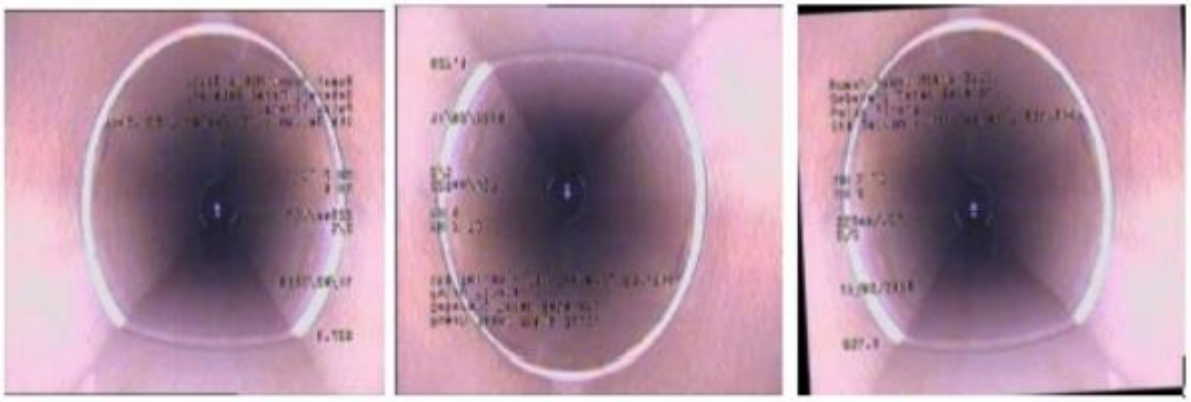
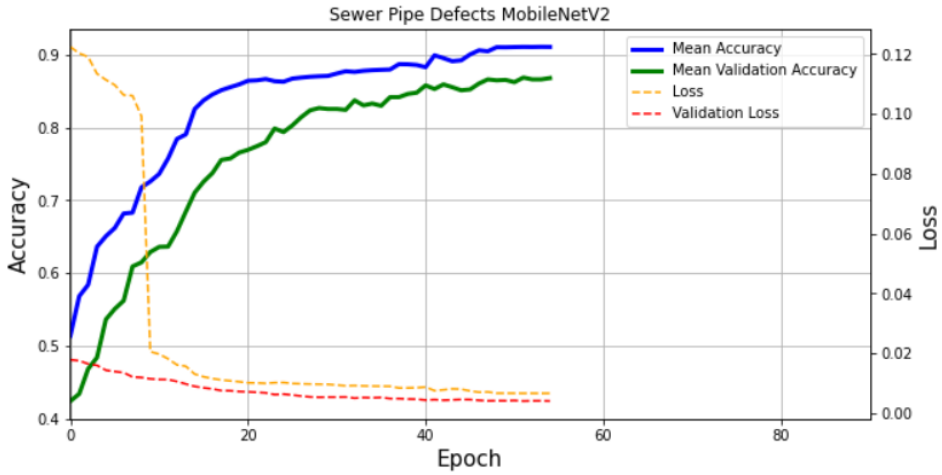


Рисунок 3.6— приклад аугментації зображень

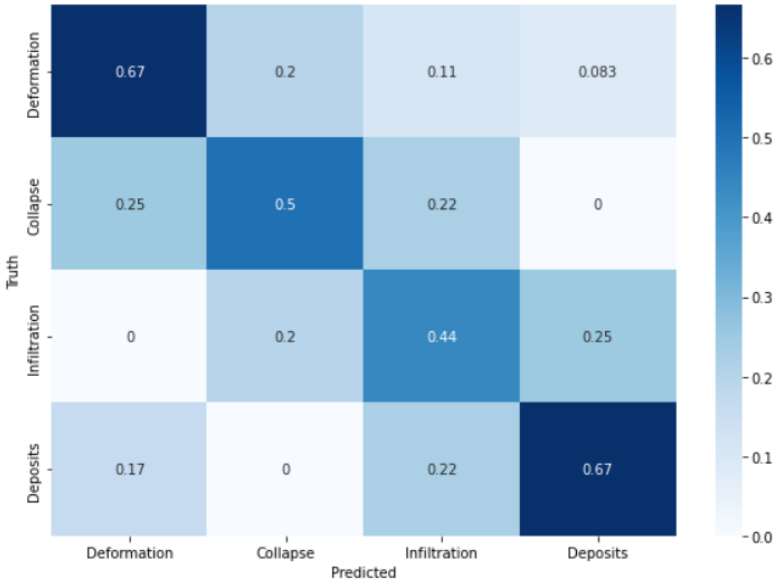
3.5. Результати роботи класифікатора

Навчання проводилося з використанням навчального набору зображень. Всього було проведено більше 60 генерацій функціонування системи. На графіку 3.1 зображено залежність точності від кількості прогонів генерацій нейронної мережі, а також залежності коефіцієнта втрат при класифікації від кількості прогонів. Синім та зеленим кольором зображено графіки точностей тестової та валідаційної вибірок відповідно. А оранжевим та червоним – графіки втрат при класифікації тестової та валідаційної вибірок:



Графік 3.1 – графік залежностей точності класифікатора та коефіцієнта втрат від кількості генерацій системи

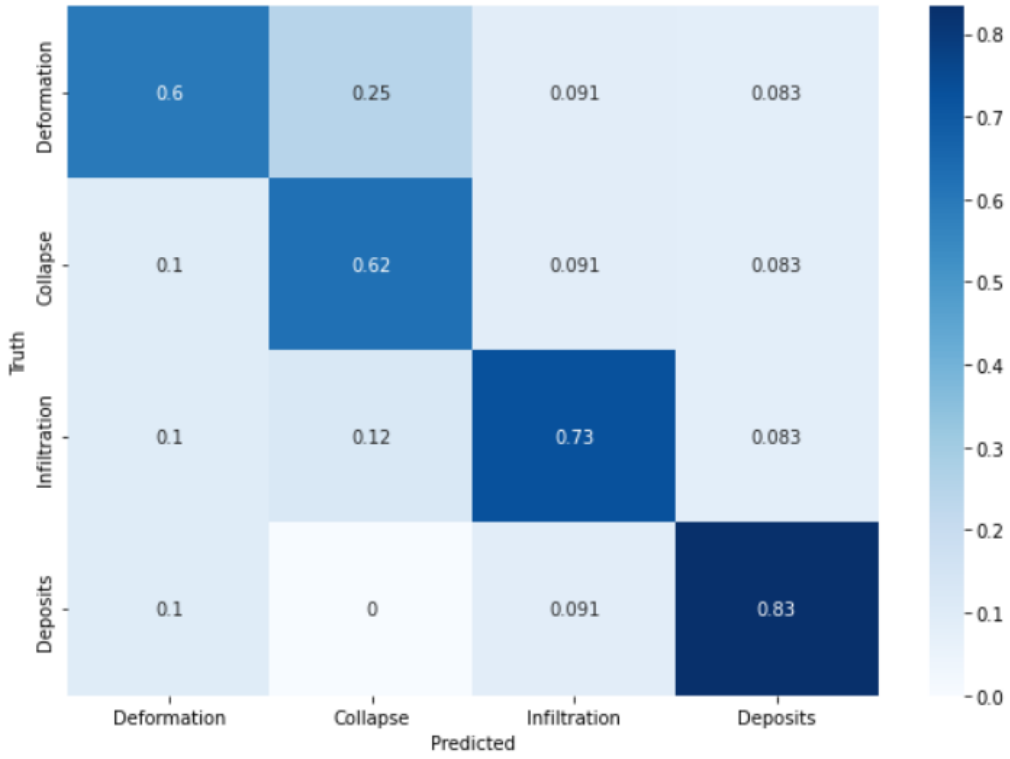
На графіку 3.2 зображено матрицю похибок класифікації класів розпізнавання при прогонах валідаційної вибірки без використання аугментації даних:



Графік 3.2 – матриця похибок класифікації до використання аугментації

На графіку простежується велика похибка класифікації, що сягає іноді 0,25 коефіцієнту розпізнавання.

На графіку 3.3 зображено матрицю похибок класифікації класів розпізнавання при прогонах валідаційної вибірки вже після застосування методу аугментації даних:



Результати, зображені на графіку, доводять, що метод аугментації справді збільшує точність розпізнавання, адже коефіцієнт похибки в максимумі знизився до 0,12. При цьому коефіцієнти вірних класифікацій класів збільшились.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи магістра було проаналізовано сучасні методи детектування станів трубопроводів каналізаційної та водопровідної систем. Підтверджено важливість та актуальність автоматизації оцінки функціонального стану трубопроводів. Аналіз існуючих методів детектування показав, що на даний момент залучення інтелектуальних систем у галузі оцінки стану інфраструктури знаходиться на початковому рівні. При цьому кількість наукових підходів до описаної проблеми зростає.

Було описано підхід аугментації даних при навчанні інтелектуальних систем, проведено аналіз готових рішень аугментації та підбір найбільш ефективних до даної задачі. Проведено аналіз роботи інтелектуальних та експертних систем, наведено чітке формулювання та опис ІС, що навчається та ІС в режимі екзамену, наведені основні відмінності.

Також було спроектовано саму технологію детектування станів трубопроводів із залученням датасетів, що знаходяться у публічному доступі. Розробка програмного забезпечення велась із використанням мови програмування Python та вбудованих в неї бібліотек роботи зі штучним інтелектом та нейронними мережами. При роботі системи в режимі екзамену отримані вихідні дані свідчать про підвищену точність системи при залучені аугментованих даних.

Беручи до уваги вищеописані критерії, проект технології та програмну реалізацію, можна рекомендувати дану технологію для вдосконалення та втілення у вигляді інтелектуальних систем для оцінки стану інфраструктурних об’єктів, таких як трубопроводи, газопроводи, електромережі, канали високошвидкісного інтернетного кабелю та інші.

СПИСОК ЛІТЕРАТУРИ

1. Схема системи ОДК для ППУ. Оперативний дистанційний контроль трубопроводів ППУ - ефективний засіб контролю або марне додаток? Стандарт організації нп «ассоціаціація пптіпі» [Електронний ресурс]. URL: https://mrc-naves.ru/uk/teplosnabzhenie/shema-sistemy-odk-dlya-ppu-operativnyi-distancionnyi-kontrol-truboprovodov-ppu---effektivnoe-sredst/

2. ИТ-ТРАНЗИТ | Діагностика та оцінка технічного стану лінійних об’єктів - ИТ-ТРАНЗИТ [Електронний ресурс]. URL: https://it-transit.com/uk/produkti/rozrakhunkovo-analitychni-systemy/diagnostika-i-otsinka-tehnichnogo-stanu-linejnih-obyektiv

3. Система оперативного дистанційного контролю ізоляції. Система ОДК для труб ППУ як інструмент технічного обслуговування теплотраси. В. Замикання сигнального проводу на трубу [Електронний ресурс]. URL: https://sadovod-baza.ru/uk/landscape-design/sistema-operativnogo-distancionnogo-kontrolya-izolyacii-sistema-odk-dlya/.

4. ТЕХНІЧНИЙ СТАН СИСТЕМ ЦЕНТРАЛІЗОВАНОГО ВОДОПОСТАЧАННЯ ТА ВОДОВІДВЕДЕННЯ – УКРАЇНСЬКИЙ ЦЕНТР ВОДНО-ЕКОЛОГІЧНИХ ПРОБЛЕМ [Електронний ресурс]. URL: https://cleanwater.org.ua/tehnichnyj-stan-system-tsentralizovanoho-vodopostachannya-ta-vodovidvedennya/

5. S Nitish H. G. K. A. S. I. S. R. Dropout: a simple way to prevent neural networks from overfitting // J Mach Learn Res. 2014. № 1 (15). С. 1929–1958.

6. David G. Distinctive image features from scale-invariant keypoints // Int J Comput Vis. 2004. (2004). С. 91–110.

7. Data Augmentation | How to use Deep Learning when you have Limited Data [Електронний ресурс]. URL: https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/

8. Hung Y. T., Aziz H. A., Ramli M. H. Combined sewer overflow treatment // Handbook of Environment and Waste Management: Air and Water Pollution Control. 2012. С. 383–404.

9. Kim K. The applicability of LID facilities as an adaptation strategy of urban CSOs management for climate change // Water Supply. 2021.

10. Yu Y. Cluster analysis for characterization of rainfalls and CSO behaviours in an urban drainage area of Tokyo // Water Science and Technology. 2013. № 3 (68). С. 544–551.

11. McLellan S. L., Roguet A. The unexpected habitat in sewer pipes for the propagation of microbial communities and their imprint on urban waters // Current Opinion in Biotechnology. 2019. (57). С. 34–41.

12. Gong Y. Effectiveness analysis of systematic combined sewer overflow control schemes in the sponge city pilot area of Beijing // International Journal of Environmental Research and Public Health. 2019. № 9 (16).

13. Ye X. Diagnosis of sewer pipe defects on image recognition of multi-features and support vector machine in a southern Chinese city // Frontiers of Environmental Science and Engineering. 2019. № 2 (13).

14. Li K., He F., Chen X. Real-time object tracking via compressive feature selection // Frontiers of Computer Science. 2016. № 4 (10). С. 689–701.

15. Cao L. Advancing the incremental fusion of robotic sensory features using online multi-kernel extreme learning machine // Frontiers of Computer Science. 2017. № 2 (11). С. 276–289.

16. Shayeganpour S. Evaluating pixel-based vs. object-based image analysis approaches for lithological discrimination using VNIR data of WorldView-3 // Frontiers of Earth Science. 2021. № 1 (15). С. 38–53.

17. Zhang X. Multi-model ensemble deep learning method for intelligent fault diagnosis with high-dimensional samples // Frontiers of Mechanical Engineering. 2021. № 2 (16). С. 340–352.

18. Rahaman M. A. Bangla language modeling algorithm for automatic recognition of hand-sign-spelled Bangla sign language // Frontiers of Computer Science. 2020. № 3 (14).

19. Ding Y., Lee W. S., Li M. Feature extraction of hyperspectral images for detecting immature green citrus fruit // Frontiers of Agricultural Science and Engineering. 2018. № 4 (5). С. 475–484.

20. Jin Q. Effective removal of Cd 2+ and Pb 2+ pollutants from wastewater by dielectrophoresis-assisted adsorption // Frontiers of Environmental Science and Engineering. 2019. № 2 (13).

ДОДАТОК А

```
!wget --no-check-certificate \
    https://storage.googleapis.com/laurencemoroney-
blog.appspot.com/rps.zip \
    -O /tmp/rps.zip

!wget --no-check-certificate \
    https://storage.googleapis.com/laurencemoroney-blog.appspot.com/rps-
test-set.zip \
    -O /tmp/rps-test-set.zip



import os
import zipfile

local_zip = '/tmp/rps.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/tmp/')
zip_ref.close()

local_zip = '/tmp/rps-test-set.zip'
zip_ref = zipfile.ZipFile(local_zip, 'r')
zip_ref.extractall('/tmp/')
zip_ref.close()



deformation_dir = os.path.join('/tmp/rps/deformation')
collapse_dir = os.path.join('/tmp/rps/collapse')
infiltration_dir = os.path.join('/tmp/rps/infiltration')
deposits_dir = os.path.join('/tmp/rps/deposits')

print('total training deformation images:', len(os.listdir(deformation_dir))
)
print('total training collapse images:', len(os.listdir(collapse_dir)))
print('total training infiltration images:', len(os.listdir(infiltration_dir
 )))
print('total training deposits images:', len(os.listdir(deposits_dir)))
deformation_files = os.listdir(deformation_dir)
print(deformation_files[:10])
collapse_files = os.listdir(collapse_dir)
print(collapse_files[:10])
infiltration_files = os.listdir(infiltration_dir)
print(infiltration_files[:10])
scissors_files = os.listdir(deposits_dir)
print(deposits_files[:10])
```

```
%matplotlib inline

import matplotlib.pyplot as plt
import matplotlib.image as mpimg

pic_index = 2

next_deformation = [os.path.join(deformation_dir, fname)
                for fname in deformation_files[pic_index-2:pic_index]]
next_collapse = [os.path.join(collapse_dir, fname)
                for fname in collapse_files[pic_index-2:pic_index]]
next_infiltration = [os.path.join(infiltration_dir, fname)
                for fname in infiltration_files[pic_index-2:pic_index]]
next_deposits = [os.path.join(deposits_dir, fname)
                for fname in deposits_files[pic_index-2:pic_index]]


for i, img_path in enumerate(next_rock+next_paper+next_scissors):
  #print(img_path)
  img = mpimg.imread(img_path)
  plt.imshow(img)
  plt.axis('Off')
  plt.show()



import tensorflow as tf
import keras_preprocessing
from keras_preprocessing import image
from keras_preprocessing.image import ImageDataGenerator

TRAINING_DIR = "/tmp/rps/"
training_datagen = ImageDataGenerator(
      rescale = 1./255,
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest')

VALIDATION_DIR = "/tmp/rps-test-set/"
validation_datagen = ImageDataGenerator(rescale = 1./255)
```

```
train_generator = training_datagen.flow_from_directory(
  TRAINING_DIR,
  target_size=(150,150),
  class_mode='categorical',
  batch_size=126
)

validation_generator = validation_datagen.flow_from_directory(
  VALIDATION_DIR,
  target_size=(150,150),
  class_mode='categorical',
  batch_size=126
)

model = tf.keras.models.Sequential([
    # Note the input shape is the desired size of the image 150x150 with 3 b
ytes color
    # This is the first convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu', input_shape=(150, 1
50, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    # The second convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The third convolution
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The fourth convolution
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # Flatten the results to feed into a DNN
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.5),
    # 512 neuron hidden layer
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax')
])


model.summary()

model.compile(loss = 'categorical_crossentropy', optimizer='rmsprop', metric
s=['accuracy'])

history = model.fit(train_generator, epochs=25, steps_per_epoch=20, validati
on_data = validation_generator, verbose = 1, validation_steps=4)
```

```
model.save("rps.h5")


import matplotlib.pyplot as plt
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'r', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.title('Training and validation accuracy')
plt.legend(loc=0)
plt.figure()


plt.show()


import numpy as np
from google.colab import files
from keras.preprocessing import image

uploaded = files.upload()

for fn in uploaded.keys():

  # predicting images
  path = fn
  img = image.load_img(path, target_size=(150, 150))
  x = image.img_to_array(img)
  x = np.expand_dims(x, axis=0)

  images = np.vstack([x])
  classes = model.predict(images, batch_size=10)
  print(fn)
  print(classes)
```