

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,

освітньо-професійної програми «Інформатика»

на тему: «Інтелектуальна система аналізу волейбольних матчів»

здобувача групи ІН - 01 Нікітенко Катерини Сергіївни

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Катерина НІКІТЕНКО

(підпис)

Керівник, кандидат технічних наук

доцент кафедри комп'ютерних наук

Ігор ШЕЛЕХОВ

(підпис)

Суми – 2024

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерні науки, освітньо-професійної програми «Інформатика»
здобувача групи ІН-01 Новікова Катерина Дмитрівна

1. Тема роботи: «Інтелектуальна система аналізу волейбольних матчів»
затверджую наказом по СумДУ від _____
2. Термін здачі здобувачем кваліфікаційної роботи _____
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз предметної області, постановка й формування завдань дослідження.
2) Огляд технологій, що використовуються для розробки інтелектуальних систем. *3) Розробка та тестування інтелектуальної системи.* *4) Аналіз отриманих результатів.*
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 « ____ » _____ р.

Завдання прийняв до виконання _____

(підпис)

Керівник _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз предметної області, постановка й формування завдань дослідження</i>	06-08.05.2024	
2	<i>Огляд технологій, що використовуються для розробки інтелектуальних систем</i>	09-11.05.2024	
3	<i>Розробка та тестування інтелектуальної системи.</i>	12-14.05.2024	
4	<i>Аналіз отриманих результатів</i>	15-17.05.2024	
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>	18-20.05.2024	

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

АНОТАЦІЯ

Записка: 43 сторінок, 22 рисунки, 1 додаток, 1 таблиця, 23 використаних джерел.

Обґрунтування актуальності теми роботи – Актуальність задачі інтелектуальної обробки відео волейбольних матчів обумовлена такими факторами як підвищення рівня аналізу гри, покращення тренувального процесу, справедливість і точність суддівства, розширення можливостей трансляцій, підвищення ефективності гравців і розробки нових стратегій, комерційні можливості.

Об'єкт дослідження — процес навчання інтелектуальної системи розпізнавання образів і сцен волейбольного матчу.

Предмет дослідження — методи і моделі технології проектування здатної до навчання системи аналізу волейбольних матчів.

Мета роботи — розробка інформаційного і програмного забезпечення інтелектуальної системи аналізу волейбольних матчів.

Методи дослідження — методи розпізнавання образів, методи розпізнавання сцен, методи машинного навчання, методи інформаційно-екстремальної інтелектуальної технології

Результати — розроблено інформаційне та програмне забезпечення здатної до навчання системи аналізу волейбольних матчів. При цьому вирішено задачі з формування вхідного математичного опису системи, розробки моделей функціонування системи в режимі навчання і екзамену, оцінки функціональної ефективності параметрів системи, розробки та програмної реалізації алгоритмів навчання та екзамену. Оцінки працездатності системи виконувалася на задачі детектування волейбольного м'яча.

МАШИННЕ НАВЧАННЯ, РОЗПІЗНАВАННЯ ОБРАЗІВ, КЛАСИФІКАТОР,
КРИТЕРІЙ ФУНКЦІОНАЛЬНОЇ ЕФЕКТИВНОСТІ

ЗМІСТ

ВСТУП.....	5
1 АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	6
1.1 Баскетбольний трекер: відстеження гравців за допомогою відеоаналітики ...	6
1.2 Автоматична система аналізу гри в американський футбол на основі глибокого навчання.....	9
1.3 Виявлення м'яча в тенісних іграх за допомогою згорткових нейронних мереж	13
1.4 Постановка задачі.....	16
2 ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ ..	17
2.1 Основні визначення.....	17
2.2 Математична модель інформаційно-екстремального навчання	19
2.3 Алгоритм інформаційно-екстремального навчання	23
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	26
3.1 Опис вхідних даних.....	26
3.2 Опис програмної реалізації.....	26
3.3 Тестування інтелектуальної системи.....	28
ВИСНОВКИ	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	35
ДОДАТОК.....	37

ВСТУП

Спортивні матчі завжди привертали увагу широкої аудиторії на різних рівнях участі, від гравців і тренерів до загальної громадськості. У сучасному світі спортивні матчі стали одним із найпопулярніших та найбільш відомих способів розваги та розвитку фізичної активності. Завдяки телевізійним трансляціям та інтернет-платформам, спортивні події тепер доступні для перегляду на всій планеті, привертаючи увагу мільйонів глядачів з усього світу. [1-5]

Результати матчів та управлінські рішення в спорті зазвичай є предметом багатьох дебатів і обговорень, що означає, що як ентузіасти, так і інсайдери цікавляться багатьма застосунками, такими як тактичний аналіз, визначення основних моментів або, в більш загальному сенсі, статистичний аналіз.[5-10]

Актуальність аналітичних систем аналізу записів спортивних матчів прослідковується в різних видах спорту. Хоча види спорту достатньо різноманітні, всі вони мають загальну необхідність в ідентифікації активних об'єктів, таких як м'яч чи гравці. У футболі, волейболі, баскетболі та хокеї на льоду системи відстеження та ідентифікації гравців дозволяють тренерам та аналітикам аналізувати дії гравців на полі, відслідковувати активність кожного з них під час гри, розробляти стратегії для досягнення перемоги розробляючи індивідуальні підходи до тренувань.[11-13]

Не менш важливим напрямком аналізу відеозаписів є розпізнавання об'єктів гри, таких як м'яч у тенісі. Системи розпізнавання м'яча дозволяють зацікавленим сторонам аналізувати траєкторію м'яча під час гри та реагувати на нього відповідно.[14-16]

Звідси витікає, що комп'ютерний зір та машинне навчання необхідні для забезпечення можливості автоматизованої чи напівавтоматизованої обробки великих об'ємів даних в спорті.

1 АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Баскетбольний трекер: відстеження гравців за допомогою відеоаналітики

Розглянемо проект, що був розроблений повністю на MATLAB, де для досягнення кінцевої мети двовимірного зображення з позиціонуванням гравців використовувався п'ятикроковий алгоритм, блок схема якого зображена на рисунку 1.1.

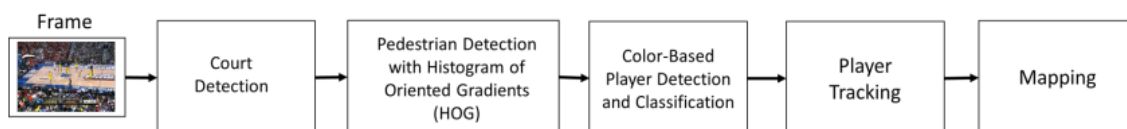


Рисунок 1.1 - Блок-схема алгоритму виявлення та відстеження гравців.

А. Виявлення майданчика

Кадри відео, що отримувались з YouTube, спочатку були перетворені з RGB в HSV (відтінок, насиченість та значення) колірну модель, а потім - в двійкове зображення. Пізніше використовується перетворення Хафа, щоб знайти прямі лінії (Hough Lines) зображення.

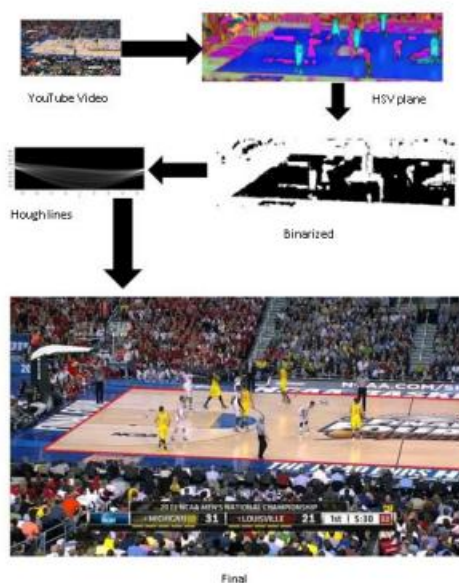


Рисунок 1.2 - Схема роботи, виконаної для виявлення країв системи.

В. Виявлення гравців за гістограмами орієнтованих градієнтів (HOG)

HOG в основному будує гістограми орієнтацій градієнтів у локалізованих частинах зображення, які можуть бути використані для ідентифікації об'єктів на зображенні. Хоча складно встановити точні характеристики цих гістограм для виявлення певного об'єкта, класифікатори машинного навчання, такі як метод опорних векторів (SVM), можуть бути використані для ідентифікації бажаного об'єкта на зображенні на основі набору навчальних даних.

Для цього проекту використовувався детектор HOG з OpenCV. Це було головним чином обумовлено тим, що OpenCV вже має стандартний набір даних для виявлення пішоходів, а також тим, що обчислення ознак HOG та SVM вже ефективно реалізовані.

Рисунок 1.3 показує приклад виявлення пішоходів баскетбольних гравців за допомогою описів HOG і класифікатора SVM. Вставка на цьому рисунку показує описи HOG одного з гравців, а також інверсію, яка відповідає відновленому зображенню за описами HOG. Це зображення показує, що описи HOG несуть значну інформацію про виявлення. У вибіркового кадру всі гравці були виявлені, але були деякі помилкові виявлення, і двоє гравців були виявлені в одній коробці, оскільки вони були занадто близько один до одного.



Рисунок 1.3 - Приклад виявлення гравців у баскетбольній грі за допомогою детектора пішоходів OpenCV з HOG.

С. Кольорова класифікація

Колірний детектор також відіграє важливу роль як резервний для НОГ детектора. Під час гри деякі гравці можуть стати частково прихованими іншими гравцями, або просто не бути виявленими НОГ детектором в певному кадрі. У цих різних сценаріях колірний детектор буде викликаний, щоб знайти відсутнього гравця, виконавши виявлення в невеликому оточенні відповідного боксу з попереднього кадру. Це працює досить добре, оскільки використовується відео з частотою кадрів 24 кадри в секунду, тому очікується, що гравець буде в околицях того місця, де він був у попередньому кадрі

Д. Відстеження

Після виявлення наступна мета - встановлення позиції гравців кадр за кадром для розуміння загальної гри. Таким чином, алгоритм відстеження відслідковував рухи гравців. Цей алгоритм використовував інформацію з послідовних кадрів за допомогою НОГ детектора та колірного класифікатора згідно з наступними сценаріями: у сценарії 1 гравця виявлено в послідовних кадрах, що забезпечує високу перекриття боксів і оновлення його позиції. У сценарії 2, коли гравець не виявлено НОГ, але може бути знайдений за допомогою колірного виявлення, система використовує попередню позицію для пошуку у сусідній області. Сценарії 3 та 4 включають додавання та видалення гравців, де втрата зв'язку з попереднім кадром вирішується за допомогою мінімального відстаневого корелятора.

Е. Картографування

Останнім етапом було пов'язано з проекцією місцезнаходження кожного гравця на верхню перспективу майданчика. Знаючи розміри майданчика, можна знайти матрицю гомографії розміром 3×3 , яка обчислюється за допомогою афінного перетворення. Потім позиція кожного гравця множиться на матрицю гомографії, яка проектує їх на модельний майданчик, як показано на рис. 1.4.



Рисунок 1.4 - Порівняння між виявленими гравцями та відображеним зображенням на майданчику. Як видно, позиція гравців відповідає положенню на моделі майданчика зверху.

Загальний підхід, який базується на використанні відеоаналітики та машинного навчання, демонструє можливість автоматичного виявлення та відстеження гравців у реальному часі. Отримані результати дозволяють зробити висновок про високу ефективність алгоритму, яка демонструється через точність виявлення та відстеження гравців навіть у складних сценаріях гри.

1.2 Автоматична система аналізу гри в американський футбол на основі глибокого навчання

Загальна схема роботи включає три підсистеми розпізнавання образів: виявлення гравців, розпізнавання номерів футболок та ідентифікація годинника. Для виявлення гравців використовується "Детектор трансформатора", що дозволяє ефективно розпізнавати гравців у складних умовах, коли їхні образи перекриваються. Друга підсистема обробляє виявлені пропозиції гравців і визначає їхні номери футболок. Розпізнавання цифр здійснюється за допомогою RetinaNet. Третя підсистема витягує інформацію про годинник гри для індексації ходу гри. З огляду на перемотки та тайм-аути відео, потрібно точно визначити час гри на кожному кадрі. Після завершення всіх підсистем інформація про номери футболок пов'язується з базою даних, індексованою за міткою годинника гри.

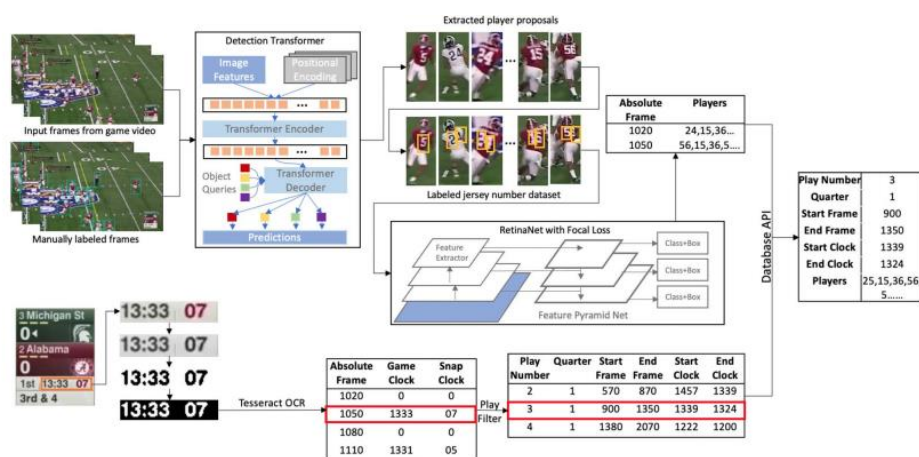


Рисунок 1.5 - Схема роботи

Система виявлення гравців.

Для забезпечення стійкості до рухливих кадрів розширюємо навчальний набір за допомогою аугментації даних та застосовуємо гаусівський фільтр до частини зображень. Також використаємо модель DETR для розв'язання проблеми виявлення гравців у перенаселених умовах.

Модель виявлення гравців здатна ідентифікувати обмежувальні прямокутники, що перекриваються розглядаючи їх просторові відносини. Порівняно з традиційними методами виявлення об'єктів, такими як Faster R-CNN, DETR користується перевагами біпартитного відношення та механізму уваги трансформера кодування-декодування, і тому він відмінно справляється з завданням виявлення в умовах перенаселеного місця футбольної гри.

Реалізація DETR ґрунтується на ResNet50. Функціональність включає вилучення ознак з зображення, додавання позиційного кодування і подачу на вхід до трансформера. Для декодера трансформера використовуються навчальні об'єктні запити, які проходять крізь кожен шар декодера, а також модуль перехресної уваги. У декодері також використовується модуль зворотного поширення для передбачення класу, координат та розмірів обмежувального прямокутника.

Підсистема розпізнавання номерів футболок має проблему нерівномірного розподілу даних. Існує два способи вирішення цієї проблеми. По-перше, замість розпізнавання двоцифрових чисел фокусуємося на одноцифровому

розпізнаванні, що значно зменшує потребу в навчальних даних. По-друге, для розпізнавання цифр використовуємо RetinaNet з фокусовою втратою, щоб зменшити проблему нерівноваги в наборі даних.



Рисунок 1.6 – Приклад пропозицій гравців; приклад цифри на футболці.

Підсистема годинника гри витягує годинник гри з відео та визначає окремі вікна відтворення. Для виявлення годинника гри використовується оптичне розпізнавання символів Tesseract (OCR). Кожен кадр вирізається, обробляється та аналізується для визначення інформації про годинник. Результати виводяться з абсолютним номером кадру. Програма фільтрування вікон відтворення визначає діапазон кадрів для кожного відтворення та виводить необхідну інформацію.

Система баз даних синхронізує результати з підсистем годинника гри та ідентифікації гравців і записує їх у зрозумілий формат. Додатково до вхідних даних з двох підсистем, використовується колір футболок команд та склад команди для категоризації гравців домашньої та гостьової команд. Через фільтрацію за кольором різних характеристик колірної гістограми визначаються команди. Для цього виділяється найрепрезентативніша область футболок гравців і аналізується інтенсивність кожного кольору. Це допомагає відрізнити гравців різних команд. Номери футболок перетворюються на імена гравців за допомогою складу команди.

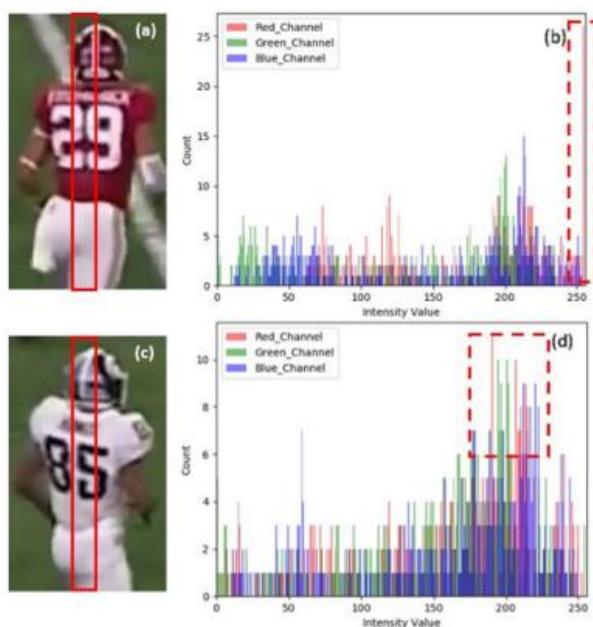


Рисунок 1.7 – Демонстрація кольорового фільтрування: (а) гравець у червоній футболці, (б) гістограма центральної області червоної футболки, (в) гравець у білій футболці; (г) гістограма центральної області білої футболки.

Play number	Quarter	Start time	End time	Home	Away	Participating players of Home team
1	1	15:00	14:54	Alabama	Michigan State	3: Calvin Ridley or Bradley Sylve; 5: Ronnie Clark or Cyrus Jones; 15: Ronnie Harrison; 19: Reggie Ragland; 24: Geno Smith; 25: Dillon Lee; 26: Marlon Humphrey; 32: Rashaan Evans; 42: Keith Holcombe; 46: Michael Nysewander; 52: Andre Sims; 99: Adam Griffith
2	1	13:53	13:24	Alabama	Michigan State	4: Daylon Charlot or Eddie Jackson; 5: Ronnie Clark or Cyrus Jones; 10: Reuben Foster; 15: Ronnie Harrison; 19: Reggie Ragland; 24: Geno Smith; 26: Marlon Humphrey; 29: Minkah Fitzpatrick; 86: A'Shawn Robinson; 93: Jonathan Allen
3	1	12:41	12:00	Alabama	Michigan State	2: Tony Brown or Derrick Henry; 3: Calvin Ridley or Bradley Sylve; 13: ArDarius Stewart; 14: Jacob Coker; 70: Ryan Kelly; 74: Cam Rombinson; 76: Dominick Jackson; 84: Hale Hentges; 88: O.J. Howard
4	1	11:37	11:18	Alabama	Michigan State	2: Tony Brown or Derrick Henry ArDarius Stewart; 14: Jacob Coker; 15: Ronnie Harrison; 50: Alphonse Taylor; 70: Ryan Kelly; 71: Ross Pierschbacher; 74: Cam Rombinson; 88: O.J. Howard

Рисунок 1.8 - Частина вихідного журналу гри, включаючи інформацію про виявлення гравців, номер гри, квартал, час початку/завершення, домашню/гостьову команду.

Дана автоматизована система аналізу гри в американський футбол, заснована на глибокому навчанні перш за все надає повне рішення з виявлення та ідентифікації гравців у грі в американський футбол, а також експертні знання з управління багатокористувацькою футбольною грою. По-друге, в запропонованому наборі даних використовуються виключно мітки обмежуючих рамок, без будь-яких інших вимог до маркування, таких як мітки людського тіла. Таким чином, система може бути більш загальною та вимагати менше зусиль.

1.3 Виявлення м'яча в тенісних іграх за допомогою згорткових нейронних мереж

Методологія, запропонована в цій роботі, використовує глибокий навчальний класифікатор для вирішення питання, чи можна певний фрагмент зображення позначити як М'яч чи Немає М'яча, а саме, згорткову нейронну мережу.

На рисунку 1.9 представлено діаграму CNN у вигляді послідовних шарів, починаючи зі шару вхідного зображення і закінчуючи шаром класу виходу. Мережа призначена для роботи з $r \times s$ RGB-зображеннями як вхідними даними. Згорткові нейронні мережі можуть зберігати просторові зв'язки на оброблених зображеннях, знаходячи велику кількість невеликих фільтрів, що імітують систему людського зору, за допомогою лінійних та нелінійних операцій. Лінійність представлена згортковим шаром, призначеним для ідентифікації карт ознак, визначених як 64 ядра розміром 5×5 , за якими слідує пакетна нормалізація. Нелінійність вводиться за допомогою функції ReLU, що підтримує класифікатор у роботі з нелінійно роздільними класами. Шар максимального зведення 2×2 , також називається шаром пониження розміру, поступово зменшує розмірність даних.

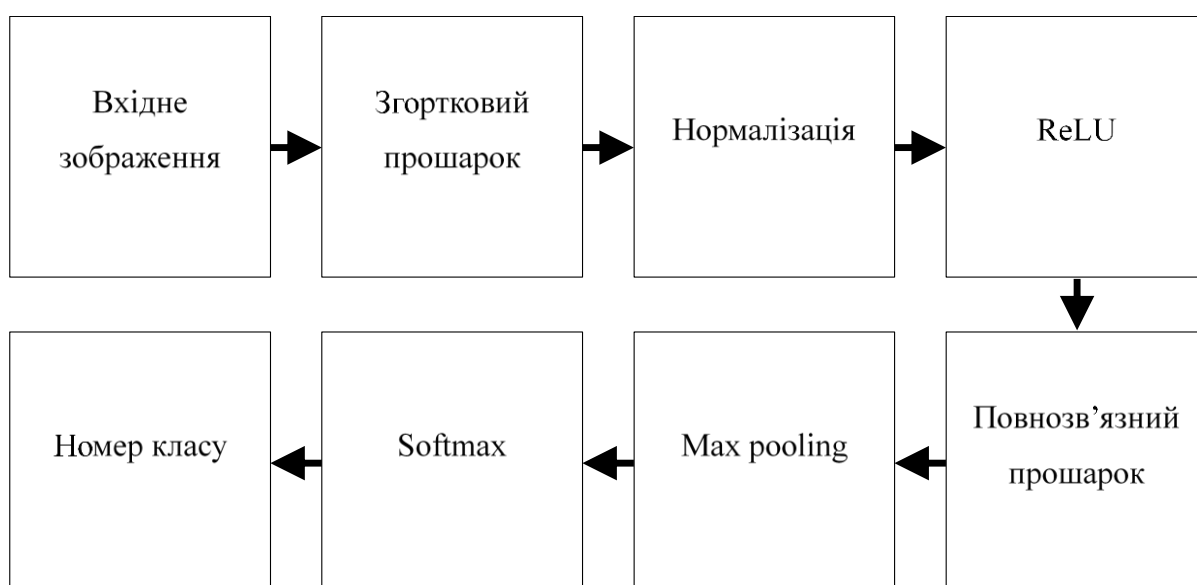


Рисунок 1.9 - Діаграма запропонованої глибокої навчальної мережі, в якій кожен блок представляє різний шар

Це означає, що глибокий навчальний конвеєр може бути повторений лише обмежену кількість разів (у даній реалізації 4). Параметри фільтрів, а також загальна кількість шарів, були емпірично визначені, щоб знайти компроміс між обчислювальною складністю та точністю. Архітектура CNN завершується шаром повністю з'єднаних 2 виходами, за яким слідує Softmax і, нарешті, шар виходу класифікації. Варто зазначити щодо розміру прямокутного фрагмента, який подається на вхід CNN, що його слід належним чином налаштувати, щоб врахувати всі можливі способи вигляду м'яча на сцені: великого або малого через перспективу, чіткого або розмитого через його швидкість. На прикладі рисунка 1.10 показано рухливий м'яч.



Рисунок 1.10 - Приклад кадру з збільшеною деталлю участка $r \times s$, що містить рухливий м'яч.

Навчання CNN проводилося за допомогою алгоритму стохастичного градієнтного спуску з моментом, із початковим коефіцієнтом навчання 0,05 та з поступовим зменшенням на кожній епохі. Спочатку набір даних розділявся на набори для навчання, валідації та тестування. Під час навчання параметри CNN оновлювалися на основі набору для навчання, а точність оцінювалася на наборі для валідації. Коефіцієнт зменшення зменшував швидкість навчання на кожній епохі для прискорення збігання та запобігання перенавчанню.

Було зібрано $N = 116385$ RGB-зображень розміром $r = 50 \times s = 50$, взятих з RAW-відео, знятих за допомогою камери AVT Prosilica GT 1920C на реальному майданчику. Зображення були отримані під час тренувань з тенісу та товариського матчу на ґрунтовому корті, та вручну класифіковані як "М'яч" або "Немає м'яча". З серед N зображень 27600 фрагментів були позначені як М'яч, а 88785 - як Немає м'яча. Приклади зображень, що належать до цих двох класів, показані на рисунку 1.11, де в першому рядку показані зразки м'яча, а в другому - фрагменти без м'яча. Перше спостереження стосується деяких прикладів зображень м'яча, вибраних і показаних у першому рядку: м'яч видно в різних умовах, наприклад, один, за сіткою, у руці гравця або розмитий через швидкий рух. Крім того, немає обмежень стосовно положення м'яча щодо фрагмента зображення, що означає, що його присутність можна спостерігати як у кутах фрагмента, так і в його центрі. Слід зазначити, що вибірки Ball і No Ball є незбалансованими за потужністю, оскільки для кожного зображення природно менше прикладів Ball порівняно з іншим класом.

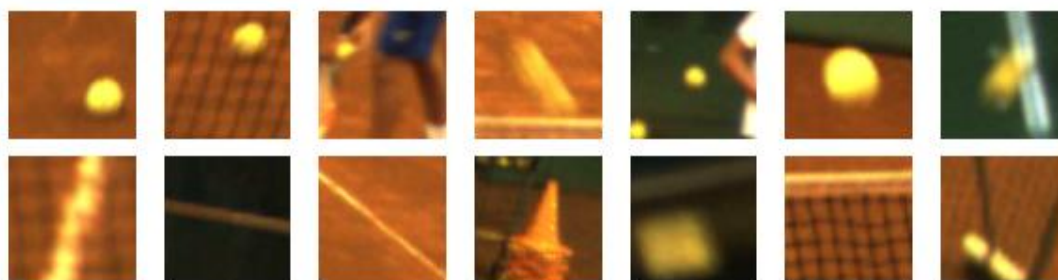


Рисунок 1.11 - Приклади зображень з м'ячем (перший ряд) та без м'яча (другий ряд), вилучених із набору даних.

Експерименти на реальних даних показали, що значення точності класифікатора досягає 98,77%, що свідчить про реалістичність такого підходу.

Штучний інтелект, зокрема машинне та глибоке навчання, надає аналітичним системам здатність аналізувати великі обсяги даних та виявляти складні зв'язки між ними. Це дозволяє точно визначати положення гравців на

полі, розпізнавати їх за номерами на футболках та відстежувати рухи м'ячів. Без цих технологій автоматизація аналізу спортивних подій була б значно складнішою та менш ефективною.

1.4 Постановка задачі

За результатами проведено огляду можна зробити висновок про важливість і актуальність задачі інтелектуального аналізу спортивних змагань, в тому числі волейбольних матчів. Задача детектування та класифікації окремих образів та сцен матчу – є однією з базових при розробці системи інтелектуального аналізу матчів. Для розв'язання такої задачі необхідно виконати ряд завдань:

- 1) Сформулювати математичний опис інтелектуальної системи,
- 2) Обрати математичну модель навчання розпізнаванню образів,
- 3) Обрати критерій оцінки ефективності параметрів навчання системи,
- 4) Розробити алгоритм навчання розпізнаванню образів та програмної його реалізувати
- 5) Перевірити працездатність розробленої системи на практичній задачі детектування та класифікації окремих образів та сцен матчу.

2 ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНА ІНТЕЛЕКТУАЛЬНА ТЕХНОЛОГІЯ

2.1 Основні визначення

Образ, сцена або клас X_k^0 – це відображення особливостей, ознак або характеристик k -тої групи подібних об'єктів або реалізацій класу, де $k = \overline{1, M}$.

Множина $\{X_k^0\}$ або алфавіт класів розпізнавання має задану потужність з M класів.

Простір ознак розпізнавання Ω визначається множиною ознак розпізнавання заданої потужності з N ознак. Клас розпізнавання є частиною такого простору $X_k^0 \subset \Omega$.

Навчальна вибірка $\{Y_{k,i}^{(j)} | k = \overline{1, M}, i = \overline{1, N}, j = \overline{1, n}\}$ складається з n – реалізацій кожного класу розпізнавання.

Інформаційно-екстремальна інтелектуальна технологія (ІЕІ-технологія) в процесі навчання для кожного класу формує вирішальне правило - його контейнер K_k^0 . Для підвищення оперативності цього процесу простір ознак розпізнавання Ω спрощується до свого бінарного варіанту Ω_B , де реалізації ознак мають значення або 0, або 1. Змінюється і метрика простору, а саме евклідова відстань замінюється на кодову, яка обчислюється як кількість ознак двох бінарних реалізацій, значення яких не співпадають. [17-18]

$$C(x^{(a)}, x^{(b)}) = \sum_{i=1}^N x_i^{(a)} \oplus x_i^{(b)} \quad (2.1)$$

де $x^{(a)}, x^{(b)}$ – довільні бінеаризовані N -вимірні вектори, таблиця істинності для бінарної операції \oplus подано в табл.2.1

Таблиця 2.1 - Таблиця істинності для бінарної операції \oplus

$x_i^{(a)}$	0	0	1	1
$x_i^{(b)}$	0	1	0	1
$x_i^{(a)} \oplus x_i^{(b)}$	0	1	1	1

Така метрика (2.1) зменшує кількість можливих значень відстаней між двома бінарними реалізаціями до $N+1$. При цьому можливі відстані приймають значення від 0, коли бінарні реалізації повністю співпадають, до N – коли в них відрізняється кожна ознака.

Перетворення початкового ознакового простору на бінарний виконується шляхом застосування процедури накладання контрольних допусків (2.2)

$$X_{k,i}^{(j)} = \begin{cases} 0, \text{ якщо } X_{k,i}^{(j)} > VDK_i \\ 1, \text{ якщо } NDK_i \leq X_{k,i}^{(j)} \leq VDK_i \\ 0, \text{ якщо } X_{k,i}^{(j)} < NDK_i \end{cases} \quad (2.2)$$

де VDK_i – верхній контрольний допуск i -тої ознаки розпізнавання, NDK_i – нижній контрольний допуск i -тої ознаки розпізнавання.

Визначення поля допусків здійснюється на формулою (2.3)

$$\begin{cases} VDK_i = \frac{1}{n} \sum_{j=1}^n Y_{k^*,i}^{(j)} + \delta_i \\ NDK_i = \frac{1}{n} \sum_{j=1}^n Y_{k^*,i}^{(j)} - \delta_i \end{cases} \quad (2.3)$$

де δ_i – ширина поля допусків, k^* - номер базового класу.

Контейнери K_k^0 визначаються на етапі навчання у вигляді N -вимірних гіперсфер шляхом оптимізації їх геометричних параметрів: N -вимірного центру $x_k^* = \langle x_{k,1}^*, \dots, x_{k,i}^*, \dots, x_{k,N}^* \rangle$ (2.4) та радіусу d_k^* (кодова відстань) (2.5).

$$x_k^* = \langle x_{k,1}^*, \dots, x_{k,i}^*, \dots, x_{k,N}^* \rangle = \begin{cases} 1, \text{ якщо } \frac{1}{n} \sum_{j=1}^n X_{k,i}^{(j)} \geq 0,5 \\ 0, \text{ якщо } \frac{1}{n} \sum_{j=1}^n X_{k,i}^{(j)} < 0,5 \end{cases} \quad (2.4)$$

$$d_k^* = \arg \max_{\{d\}} E_k^{(d)} \quad (2.5)$$

де $d = \overline{0, N}$ – множина потенційних радіусів контейнерів в N -вимірному бінараризованому просторі, $E_k^{(d)}$ – значення критерію функціональної ефективності геометричних параметрів контейнеру K_k^0 .

При цьому на значення геометричних параметрів контейнерів накладаються додаткові обмеження (2.6), що дозволяють контейнерам перетинатися без поглинання центрів один одного.

$$\left\{ \begin{array}{l} (\forall K_k^0 \in \Omega_B) [K_k^0 \neq \emptyset] \\ (\exists K_k^0 \in \Omega_B) (\exists K_l^0 \in \Omega_B) [k \neq l \rightarrow K_k^0 \cap K_l^0 \neq \emptyset]; \\ (\forall K_k^0 \in \Omega_B) (\forall K_l^0 \in \Omega_B) [k \neq l \rightarrow d_k^* < C(x_k^*, x_l^*) > d_l^*]; \\ \bigcup_{k=1..m} K_k^0 \subseteq \Omega_B; \quad k \neq l; k, l = \overline{1, M} \end{array} \right. \quad (2.6)$$

2.2 Математична модель інформаційно-екстремального навчання

Подамо математичну модель машинного навчання у вигляді графу, вершини якого відображають множини вхідних, вихідних та проміжних даних, що формуються або змінюються в процесі навчання, а дуги – операторам, що регламентують відповідні перетворення [19]. В математичній моделі розглянемо два класи розпізнавання: X_1^0 – на зображенні, що аналізується присутній об'єкт спортивних змагань або відображено певну ситуацію, яка склалася під час спортивного змагання; та X_0^0 на зображенні, що аналізується такі об'єкти або ситуації відсутні. (рис. 2.1)

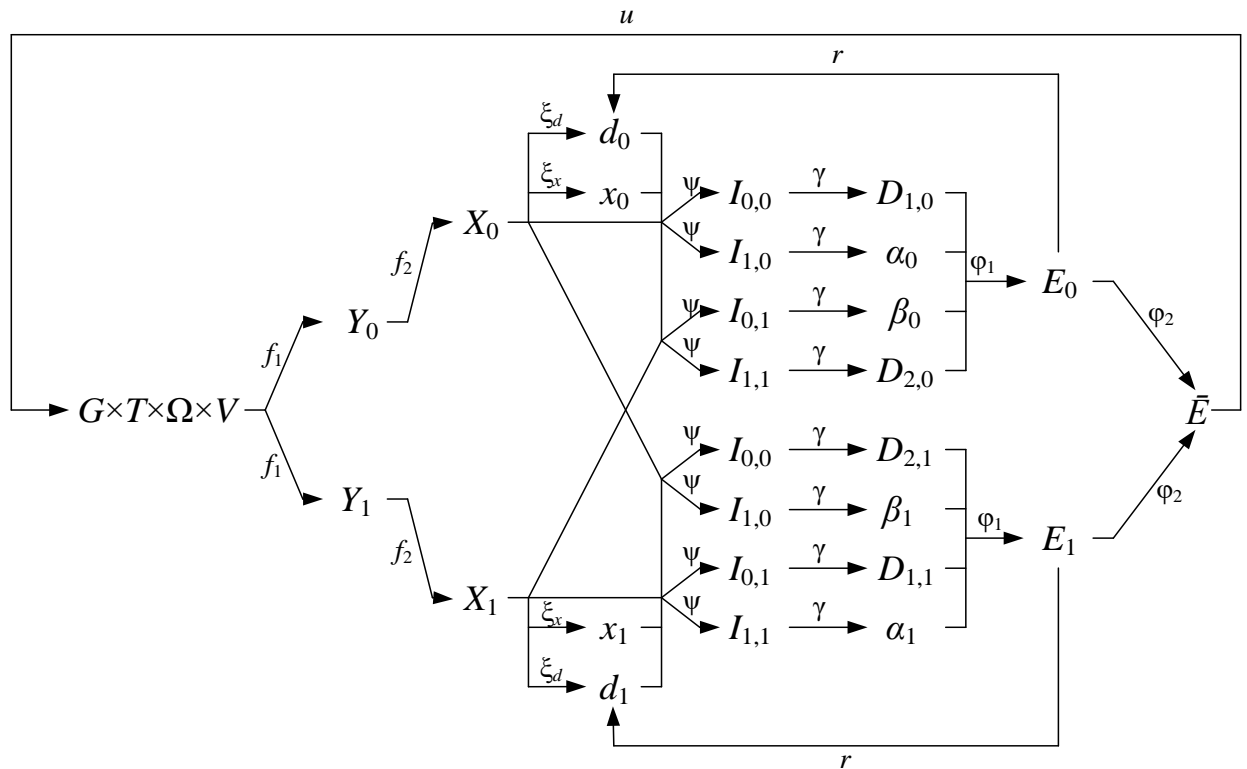


Рисунок 2.1 – Категорійна модель машинного навчання з оптимізацією геометричних параметрів контейнерів

У категорійній моделі (рис. 2.5) прийнято такі позначення:

T – множина моментів аналізу зображень та сцен спортивного змагання;

G – простір входних факторів, які впливають на процес аналізу;

Ω – простір ознак;

V – множина вирішальних правил;

Y_1 – вибіркова множина, що сформована для класу X_1^0 – на зображенні, що аналізується присутній об'єкт спортивних змагань або відображено певну ситуацію, яка склалася під час спортивного змагання;

Y_0 – вибіркова множина, що сформована для класу X_0^0 на зображенні, що аналізується такі об'єкти або ситуації відсутні;

X_1 – бінараризована вибіркова множина класу X_1^0

X_0 – бінараризована вибіркова множина класу X_0^0

d_1, x_1 – геометричні параметри контейнеру K_1^0 ,

d_0, x_0 – геометричні параметри контейнеру K_0^0 ,

$I_{k,l}$ – гіпотези щодо належності реалізацій класу X_k^0 до контейнеру K_l^0 .

$D_{1,k}$ – перша достовірність вирішального правила класу X_k^0 ,

$D_{2,k}$ – друга достовірність вирішального правила класу X_k^0 ,

α_k – помилка першого роду вирішального правила класу X_k^0 ,

β_k – помилка другого роду вирішального правила класу X_k^0 ,

E_k – значення критерію функціональної ефективності геометричних параметрів контейнеру K_k^0 ,

\bar{E} – усереднене значення критерію функціональної ефективності.

f_1 – оператор формування навчальної матриці,

f_2 – оператор бінеаризації навчальної матриці,

ξ_x – оператор формування центру контейнеру,

ξ_d – оператор формування радіусу контейнеру,

ψ – оператор формування гіпотез щодо належності реалізацій класів до контейнерів,

γ – оператор перевірки гіпотез і визначення точнісних характеристик,

φ_1 – оператор обчислення значення критерію функціональної ефективності,

φ_2 – оператор обчислення усередненого значення критерію функціональної ефективності,

r – оператор оптимізації радіусу контейнеру,

u – оператор керування процесом машинного навчання.

Оптимізація геометричних параметрів контейнерів подана у вигляді циклічної послідовності операторів (2.2).

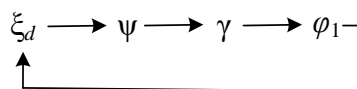


Рисунок 2.2 – Контур оптимізації геометричних параметрів контейнерів

Для оптимізації інших параметрів системи в базову модель (рис. 2.1) необхідно додати відповідні замкнені контури, що включають в себе оптимізація геометричних параметрів контейнерів. На рис. 2.3 подано модифіковану категорійну модель машинного навчання з оптимізацією параметрів бінеаризації навчальної матриці.

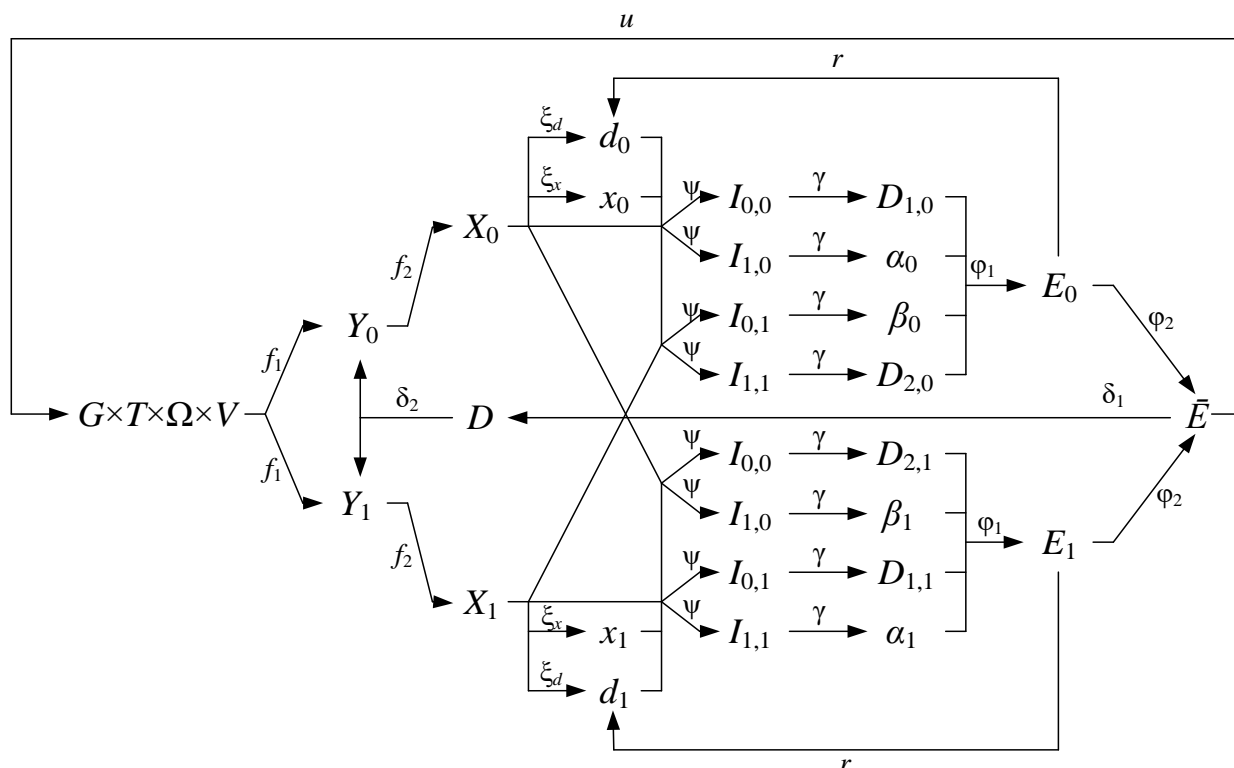


Рисунок 2.3 – Категорійна модель машинного навчання з оптимізацією параметрів бінеаризації навчальної матриці

де множина D складається з варіантів системи контрольних допусків, що формуються для різних значень δ_i – ширини поля допусків, k^* - номеру базового класу.

Додані оператори δ_1 та δ_2 керують процесами створення системи допусків, перевірки її функціональної ефективності та формують зовнішній контур операторів (рис. 2.4).

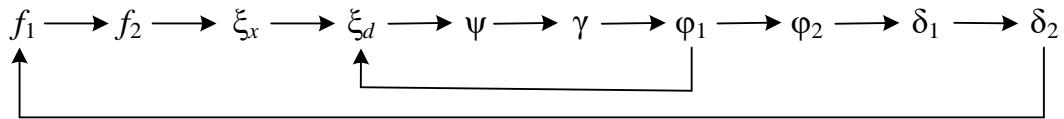


Рисунок 2.4 – Контур оптимізації параметрів бінеаризації навчальної матриці

Таким чином, запропоновані математичні моделі в явному вигляді описують перетворення вхідних даних на вирішальні правила в ході машинного навчання системи аналізу спортивних змагань і здатні легко модифікуватися шляхом додавання нових контурів оптимізації функціональних параметрів системи.

Центральним елементом математичних моделей машинного навчання є оцінка функціональної ефективності системи. В цій роботі використовувалася модифікований інформаційний критерій Кульбака (2.4)

$$\begin{aligned}
 & E_k \left(D_{1,k}(d_k), D_{2,k}(d_k), \alpha_k(d_k), \beta_k(d_k) \right) = \\
 & = 0,5 \log_2 \left(\frac{D_{1,k}(d_k) + D_{2,k}(d_k) + \frac{1}{n+1}}{\alpha_k(d_k) + \beta_k(d_k) + \frac{1}{n+1}} \right) * \\
 & * \left[(D_{1,k}(d_k) + D_{2,k}(d_k)) - (\alpha_k(d_k) + \beta_k(d_k)) \right]
 \end{aligned} \tag{2.7}$$

2.3 Алгоритм інформаційно-екстремального навчання

Алгоритми машинного навчання подамо за допомогою псевдокоду:

algorithm Начання з оптимізацією геометричних параметрів контейнерів **is**

input:

система контрольних допусків $\{\delta_i | i = \overline{1, N}\}$,

навчальна вибірка $\{Y_{k,i}^{(j)} | k = \overline{1, M}, i = \overline{1, N}, j = \overline{1, n}\}$

output:

геометричні параметри контейнерів $K_k^0 = \{x_k^*, d_k^* | k = \overline{1, M}\}$,

максимальні значення КФЕ $\{E_{max,k} | k = \overline{1, M}\}$

точносні характеристики $\{D_{1,k}, D_{2,k}, \alpha_k, \beta_k | k = \overline{1, M}\}$

for each класу $k = \overline{1, M}$ **do**
 for each ознаки $i = \overline{1, N}$ **do**
 for each реалізації $y_{k,i}^{(j)}$ **in** $\{Y_{k,i}^{(j)}\}$ **do**
 if $y_{k,i}^{(j)}$ **in** $\delta_{K,i}$ **then** $x_{k,i}^{(j)} \leftarrow 1$ **else** $x_{k,i}^{(j)} \leftarrow 0$
 if $\frac{\sum_{j=1}^n x_{k,i}^{(j)}}{n} \geq 0.5$ **then** $x_{k,i} \leftarrow 1$ **else** $x_{k,i} \leftarrow 0$
for each класу $k = \overline{1, M}$ **do**
 for each класу $l = \overline{1, M}$ **do**
 $d_{k,l} \leftarrow d(x_k \oplus x_l)$
 $d_c^{(k)} \leftarrow \max_{l \neq k} d_{k,l}; k_c^{(k)} \leftarrow \arg \max_{l \neq k} d_{k,l}$
 for each реалізації $x_k^{(j)}$ **in** $\{X_k^{(j)}\}$ **do**
 $s_1^{(j)} \leftarrow d(x_k \oplus x_k^{(j)})$
 for each реалізації $x_{k_c^{(k)}}^{(j)}$ **in** $\{X_{k_c^{(k)}}^{(j)}\}$ **do**
 $s_2^{(j)} \leftarrow d(x_k \oplus x_{k_c^{(k)}}^{(j)})$
 for each радіусу $d = \overline{0, N}$ **do**
 $K_1 \leftarrow \sum_{j=1}^n (s_1^{(j)} \leq d); K_2 \leftarrow \sum_{j=1}^n (s_2^{(j)} \leq d)$
 $E_d \leftarrow 0.5 \log_2 \left(\frac{K_1 + n - K_2 + \frac{1}{n+1}}{n - K_1 + K_2 + \frac{1}{n+1}} \right) \left(\frac{K_1 + n - K_2 - (n - K_1 + K_2)}{n} \right)$
 $D_{1,d} \leftarrow \frac{K_1}{n}; D_{2,d} \leftarrow 1 - \frac{K_2}{n}; \alpha_d \leftarrow 1 - \frac{K_1}{n}; \beta_d \leftarrow \frac{K_2}{n}$
 $E_{max,k} \leftarrow \max_{\substack{d < d_c^{(k)} \\ D_{1,d} \geq 0.5 \text{ and } D_{2,d} \geq 0.5}} E_d; d_k^* \leftarrow \arg \max_{\substack{d < d_c^{(k)} \\ D_{1,d} \geq 0.5 \text{ and } D_{2,d} \geq 0.5}} E_d$
 $D_{1,k} \leftarrow D_{1,d_k^*}; D_{2,k} \leftarrow D_{2,d_k^*}; \alpha_k \leftarrow \alpha_{d_k^*}; \beta_k \leftarrow \beta_{d_k^*}$
return $K_k^0 = \{x_k^*, d_k^* | k = \overline{1, M}\}, \{E_{max,k} | k = \overline{1, M}\} \{D_{1,k}, D_{2,k}, \alpha_k, \beta_k | k = \overline{1, M}\}$

algorithm навчання з оптимізацією параметрів бінеаризації навчальної матриці **is**

input:

навчальна вибірка $\{Y_{k,i}^{(j)} | k = \overline{1, M}, i = \overline{1, N}, j = \overline{1, n}\}$

система нормованих допусків $\{\delta_{H,i} | i = \overline{1, N}\}$,

output:

система контрольних допусків $\{\delta_{K,i}^* | i = \overline{1, N}\}$,

геометричні параметри контейнерів $K_k^0 = \{x_k^*, d_k^* | k = \overline{1, M}\}$,

максимальні значення КФЕ $\{E_{max,k} | k = \overline{1, M}\}$

точносні характеристики $\{D_{1,k}, D_{2,k}, \alpha_k, \beta_k | k = \overline{1, M}\}$

for each δ_K **in** δ_H **do**

$\{K_k^0, E_{max,k}, D_{1,k}, D_{2,k}, \alpha_k, \beta_k\} \leftarrow$ навчання з оптимізацією

геометричних параметрів контейнерів $(Y_{k,i}^{(j)}, \delta_K)$

$$\bar{E}_{max} \leftarrow \max_{\delta_K} \frac{\sum_{k=1}^M E_{max,k'}}{M}; \delta_K^* \leftarrow \arg \max_{\delta_K} \frac{\sum_{k=1}^M E_{max,k'}}{M}$$

$\forall k (D_{1,k} \geq 0.5 \text{ and } D_{2,k} \geq 0.5)$ $\forall k (D_{1,k} \text{ and } D_{2,k} \geq 0.5)$

$\{K_k^0, E_{max,k}, D_{1,k}, D_{2,k}, \alpha_k, \beta_k\} \leftarrow$ навчання з оптимізацією геометричних

параметрів контейнерів $(Y_{k,i}^{(j)}, \delta_K^*)$

return $\{\delta_{K,i}^* | i = \overline{1, N}\}, K_k^0 = \{x_k^*, d_k^* | k = \overline{1, M}\}, \{E_{max,k} | k = \overline{1, M}\}, \{D_{1,k}, D_{2,k}, \alpha_k, \beta_k | k = \overline{1, M}\}$

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

3.1 Опис вхідних даних

Для формування вхідних даних були використані фото спортивних змагань, які проводяться в Волейбольному центрі Сумського державного університету.



Рисунок 3.1 – Фото спортивного змагання за яким формувалася навчальна матриця системи

Навчальна матриця, що формувалася за фото (рис.3.1) виділяла один з невеликих об'єктів спортивного змагання і формувала класифікаційних контейнер для нього і ще один для його оточення. В цій роботі таким об'єктом був волейбольний м'яч, зображення якого мало розмірність 20 на 20 пікселів. Кожний піксель стовпця зображення розглядався як реалізація однієї ознаки. Таким чином, навчальна матриця складалася з 2 класів, 20 ознак і 20 реалізацій.

3.2 Опис програмної реалізації

Для програмної реалізації було обрано мову Java та середовище Eclipse IDE for Java Developers [20-23]. Основні змінні і їх опис наведено в табл. 3.1:

Таблиця 3.1 – Основні змінні

Назва	Тип	Опис
m	ціле	Кількість класів
N	ціле	Кількість ознак
n	ціле	Кількість реалізацій
img	Об'єкт класу ImageIO	Об'єкт для роботи з файлами
y	Тривимірний масив цілих	Навчальна матриця
x	Тривимірний масив цілих	Бінеаризована навчальна матриця
ev	Двовимірний масив цілих	Центри контейнерів класів
d_opt	Масив цілих	Радіуси контейнерів класів
E_max	Масив дійсних	Значення КФЕ
d1_opt	Масив дійсних	Значення першої достовірності
beta_opt	Масив дійсних	Значення помилки другого роду
avg_y	Масив дійсних	Середина поля допусків
delta	Дійсне	Ширина поля допусків
xc	Масив цілих	Найближчі класи
dc	Масив цілих	Кодова відстань між центрами класами
SK1	Масив цілих	Кодова відстань між центром класу і реалізаціями класу
SK2	Масив цілих	Кодова відстань між центром класу і реалізаціями найближчого класу

Програмний код наведено в додатку.

3.3 Тестування інтелектуальної системи

На першому етапі перевірки працездатності системи було задіяно алгоритм навчання з оптимізацією параметрів бінарної матриці за паралельним алгоритмом, коли ширина поля допусків була однаковою для всіх ознак. При цьому зміна ширини поля також виконувалася одночасно для всіх ознак. Базовим класом вважався клас X_1^0 – на зображенні, що аналізується присутній об'єкт спортивних змагань або відображено певну ситуацію, яка склалася під час спортивного змагання. Об'єктом, для якого відтворювався класифікатор був волейбольний м'яч.

Результати такої оптимізації наведено на рис. 3.2.

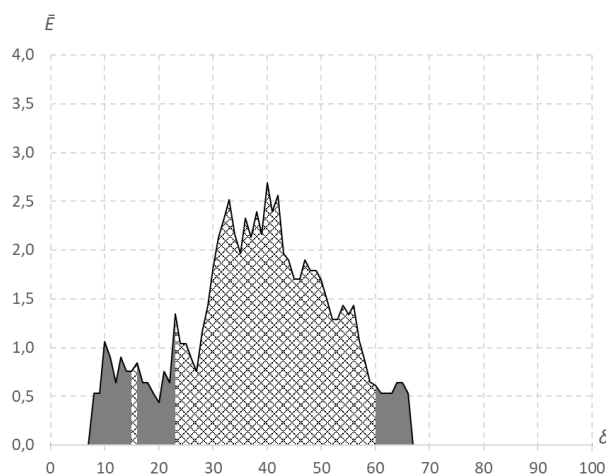


Рисунок 3.2 – Графік залежності усередненого значення критерію функціональної ефективності від ширини системи допусків і базовим класом

$$X_1^0$$

Аналіз рис. 3.2 показує, що найбільше усереднене значення критерію функціональної ефективності 2,69277 відповідає ширині системи допусків 40. Результати оптимізації геометричних параметрів контейнерів класів X_1^0 – на зображенні, що аналізується присутній об'єкт спортивних змагань або відображено певну ситуацію, яка склалася під час спортивного змагання та X_0^0 на зображенні, що аналізується такі об'єкти або ситуації відсутні наведено на рис.

3.3

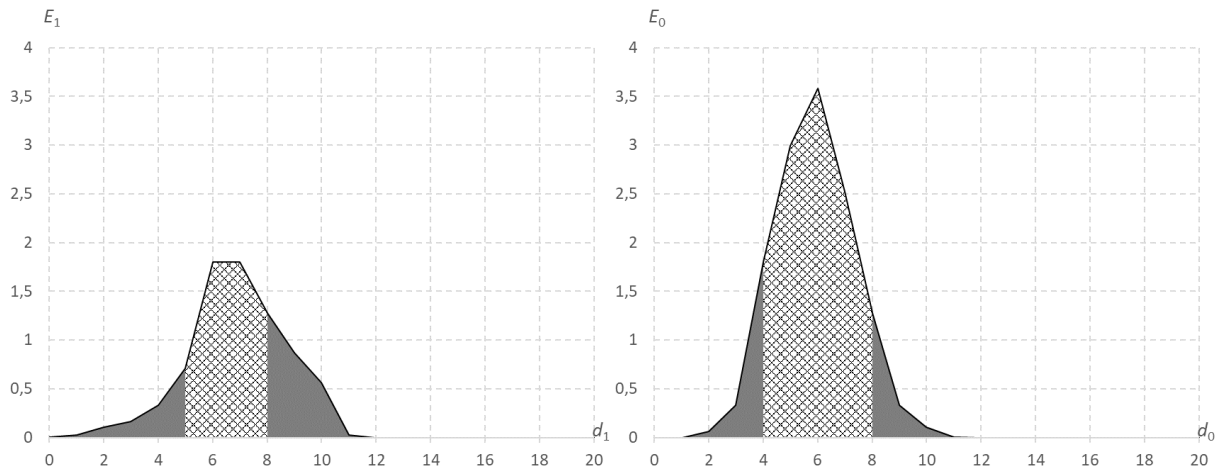


Рисунок 3.3 – Графік залежності значення критерію функціональної ефективності від радіусів (в кодових одиницях) контейнерів

Аналіз рис. 3.3 показує, що найбільше значення критерію функціональної ефективності для контейнера класу X_1^0 1,80157 відповідає радіусу 6 при міжцентровій кодовій відстані 9, а найбільше значення критерію функціональної ефективності для контейнера класу X_0^0 3,58396 відповідає радіусу 6 при міжцентровій кодовій відстані 9. Аналіз точнісних характеристик класу X_1^0 - $D_{1,1}(6) = 0,75$, $\beta_1(6) = 0,00$ та класу X_0^0 - $D_{1,0}(6) = 0,95$, $\beta_0(6) = 0,00$ підтверджує, те що класифікатор не є безпомилковим.

Далі було задіяно алгоритм навчання з оптимізацією параметрів бінарної навігації навчальною матриці за послідовним алгоритмом, коли ширина поля допусків була різною для всіх ознак. При цьому зміна ширини поля виконувалася аналогічно до паралельного алгоритму, але для кожної ознаки по черзі. Базовим класом залишився клас X_1^0 – на зображенні, що аналізується присутній об'єкт спортивних змагань або відображено певну ситуацію, яка склалася під час спортивного змагання. Об'єкт, для якого відтворювався класифікатор, також не змінювався.

Результати такої оптимізації наведено на рис. 3.4.

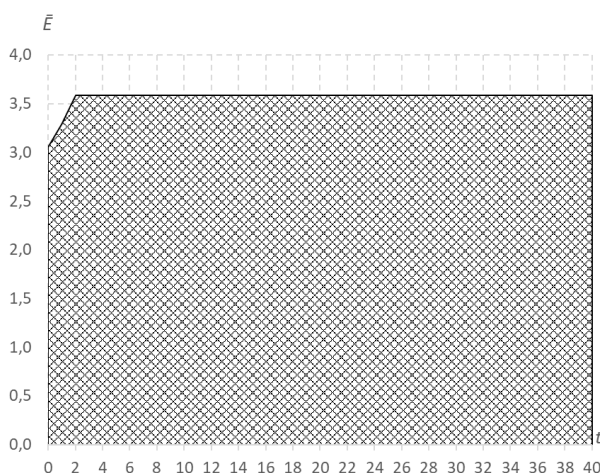


Рисунок 3.4 – Графік залежності усередненого значення критерію функціональної ефективності від кроку послідовної оптимізації і базовим класом X_1^0

Аналіз рис. 3.4 показує, що найбільше усереднене значення критерію функціональної ефективності 3.58396 було отримано на сороковому кроці оптимізації, тобто для кожної ознаки допуски оптимізувалися двічі. Результати оптимізації геометричних параметрів контейнерів класів X_1^0 – на зображенні, що аналізується присутній об’єкт спортивних змагань або відображено певну ситуацію, яка склалася під час спортивного змагання та X_0^0 на зображенні, що аналізується такі об’єкти або ситуації відсутні наведено на рис. 3.5

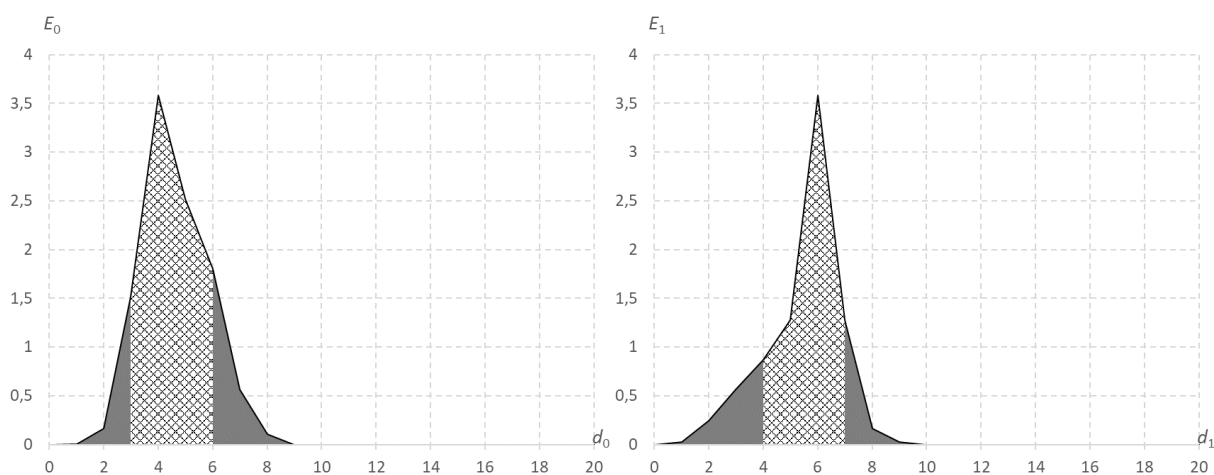


Рисунок 3.5 – Графік залежності значення критерію функціональної ефективності від радіусів (в кодівих одиницях) контейнерів

Аналіз рис. 3.5 показує, що найбільше значення критерію функціональної ефективності для контейнера класу X_1^0 ,58396 відповідає радіусу 6 при міжцентровій кодовій відстані 9, а найбільше значення критерію функціональної ефективності для контейнера класу X_0^0 3,58396 відповідає радіусу 4 при міжцентровій кодовій відстані 9. Аналіз точнісних характеристик класу X_1^0 - $D_{1,1}(6) = 0,95$, $\beta_1(6) = 0,00$ та класу X_0^0 - $D_{1,0}(6) = 0,95$, $\beta_0(6) = 0,00$ підтверджує, те що класифікатор не є безпомилковим, але помилка не перевищує 5%.

Змінемо базовий клас на X_0^0 – на зображенні, що аналізується відсутній об’єкт спортивних змагань або не відображено певну ситуацію, яка склалася під час спортивного змагання. Об’єктом, для якого відтворювався класифікатор, залишимо волейбольний м’яч.

Результати такої оптимізації параметрів процедури бінарноїзації наведено на рис. 3.6.

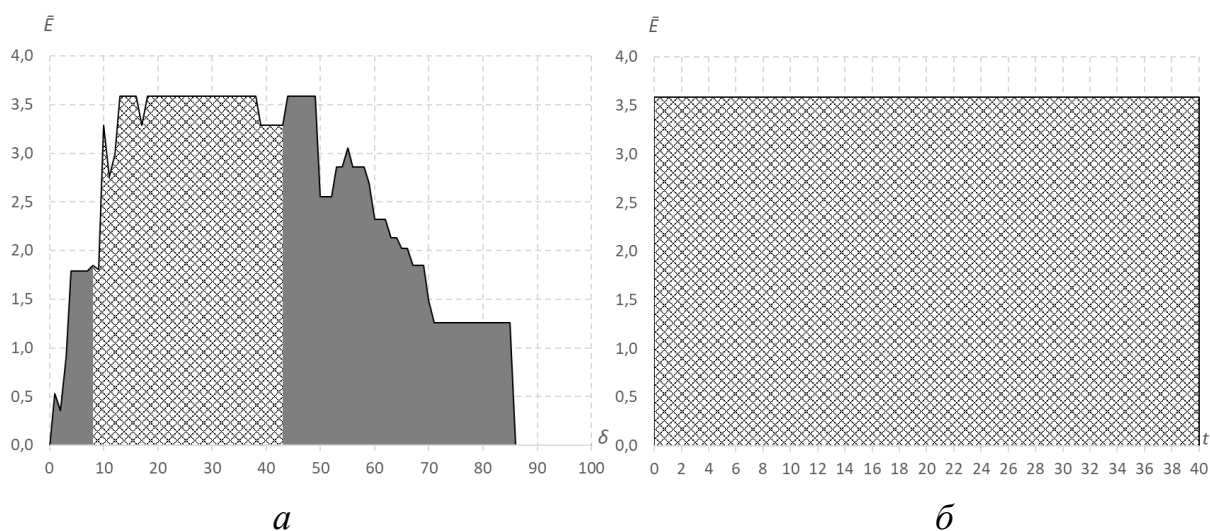


Рисунок 3.6 – Графік залежності усередненого значення критерію функціональної ефективності при оптимізації параметрів процедури бінарноїзації і базовим класом X_0^0 : *a* – паралельний алгоритм, *б* – послідовний

Аналіз рис. 3.6 показує, що найбільше усереднене значення критерію функціональної ефективності 3,58396 відповідає ширині системи допусків 13-16, 18-38 і не змінюється при застосуванні послідовної оптимізації. Результати оптимізації геометричних параметрів контейнерів класів X_1^0 – на зображенні, що аналізується присутній об’єкт спортивних змагань або відображено певну

ситуацію, яка склалася під час спортивного змагання та X_0^0 на зображенні, що аналізується такі об'єкти або ситуації відсутні наведено на рис. 3.7

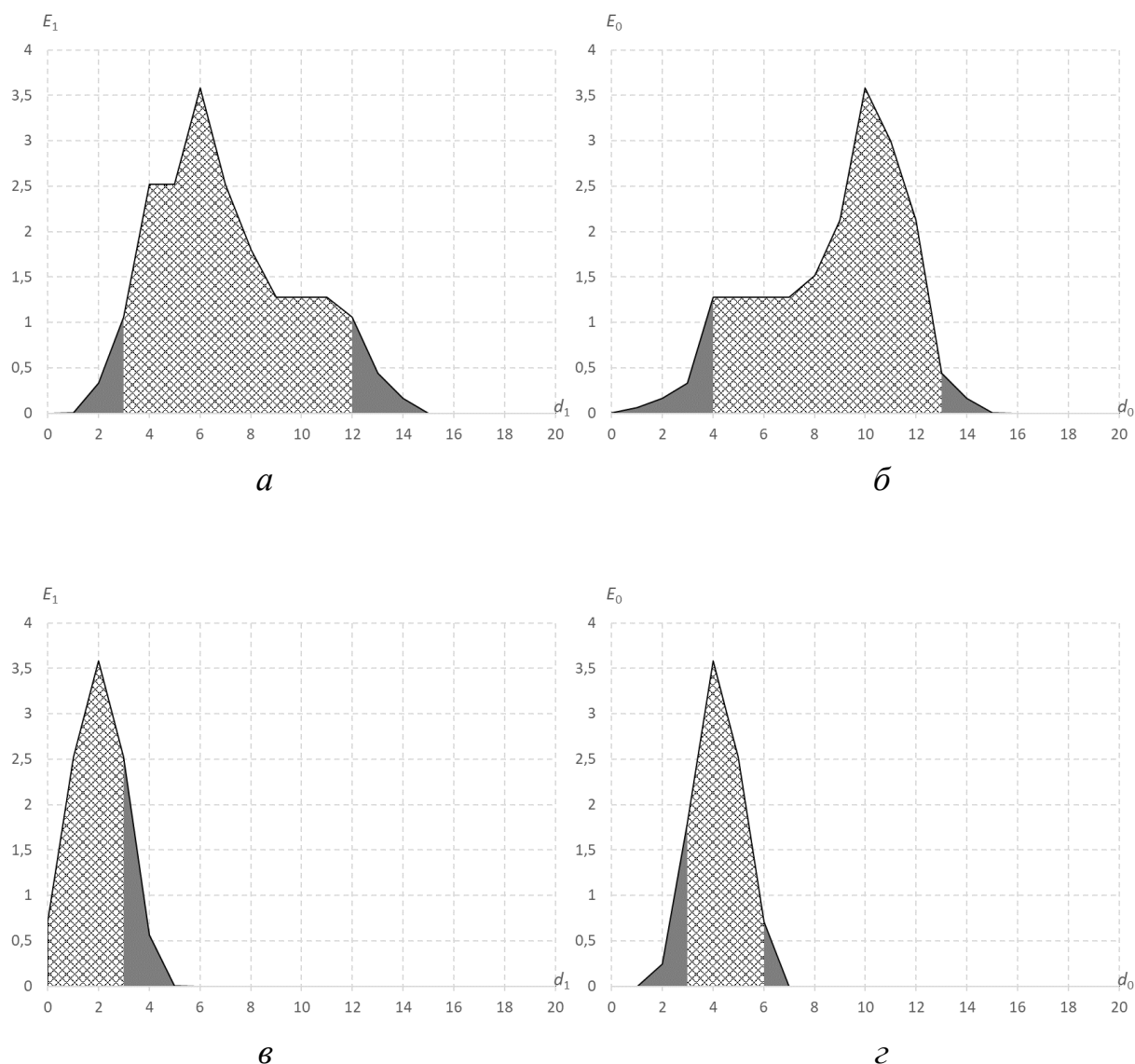


Рисунок 3.7 – Графік залежності значення критерію функціональної ефективності від радіусів (в кодових одиницях) контейнерів

Аналіз рис. 3.7а,б показує, що найбільше значення критерію функціональної ефективності для контейнера класу X_1^0 3,58396 відповідає радіусу 6 при міжцентровій кодовій відстані 15, а найбільше значення критерію функціональної ефективності для контейнера класу X_0^0 3,58396 відповідає радіусу 6 при міжцентровій кодовій відстані 10. Аналіз точнісних характеристик класу X_1^0 - $D_{1,1}(6)= 1$, $\beta_1(6)= 0,05$ та класу X_0^0 - $D_{1,0}(10)= 0,95$, $\beta_0(10)= 0,00$ підтверджує, те що класифікатор не є безпомилковим.

Аналіз рис. 3.7 в,г показує, що найбільше значення критерію функціональної ефективності для контейнера класу X_1^0 та X_0^0 залишились незмінними, При цьому відбулась зміна міжцентрової відстані до 7 кодових одиниць, а радіуси контейнерів зменшились до 2 та 4 кодових одиниць відповідно $X_1^0 - D_{1,1}(2) = 1, \beta_1(2) = 0,05$ та класу $X_0^0 - D_{1,0}(4) = 0,95, \beta_0(4) = 0,00$ підтверджує, те що класифікатор не є безпомилковим, але помилка не перевищує 5%.

Таким чином, отримані значення доводять функціональну ефективність розробленої системи для невеликих об'єктів спортивних змагань типу м'яч. Надалі планується виконати аналогічні дослідження для більших об'єктів, наприклад, гравців, а також аналіз сцен з ними.

ВИСНОВКИ

В кваліфікаційній роботі було розв'язано задачу детектування та класифікації окремих образів та сцен матча – є однією з базових при розробці системи інтелектуального аналізу матчів. Для розв'язання такої задачі було виконано такі завдання:

- 1) Сформовано математичний опис інтелектуальної системи,
- 2) Обрано математичну модель інформаційно-екстремального машинного навчання розпізнаванню образів,
- 3) Обрано інформаційних критерій оцінки ефективності параметрів навчання системи,
- 4) Розроблено алгоритми інформаційно-екстремального машинного навчання розпізнаванню образів та виконано їх програмну реалізацію.
- 5) Працездатність розробленої системи на практичній задачі детектування волейбольного м'яча.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Anderson C., Sally D. The Numbers Game: Why Everything You Know About Soccer Is Wrong. - Penguin Books, 2013. – 400 p.
2. TIFO The Athletic How To Watch Football: 52 Rules for Understanding the Beautiful Game, On and Off the Pitch. - Particular Books, 2023.- 224 p.
3. Tippett J. The Expected Goals Philosophy: A Game-Changing Way of Analysing Football.- Independently published, 2019. – 220 p.
4. Lewis M. Moneyball: The Art of Winning an Unfair Game. - W. W. Norton & Company, 2004. - 336 p.
5. Winston W. L. Mathletics: How Gamblers, Managers, and Sports Enthusiasts Use Mathematics in Baseball, Basketball, and Football. - Princeton University Press, 2021. – 392 p.
6. O'Hanlon R. Net Gains: Inside the Beautiful Game's Analytics Revolution. - Harry N. Abrams, 2022. – 288 p.
7. Kuper S., Szymanski S. Soccernomics. – HarperSport, 2018. – 576 p.
8. Moskowitz T. J., Wertheim L. J. Scorecasting: The Hidden Influences Behind How Sports Are Played and Games Are Won. –Three Rivers Press, 2012. – 288 p.
9. Biermann C. Football Hackers: The Science and Art of a Data Revolution.- Blink Publishing, 2019. – 304 p.
10. Alexander D. Outside the Box: A Statistical Journey through the History of Football. – Century, 2017. – 352 p.
11. Memmert D. Computer Science in Sport: Modeling, Simulation, Data Analysis and Visualization of Sports-Related Data. – Springer, 2024.
12. Sanders G. J., Peacock C. A. Data Analysis in Sports Science: Utilizing SPSS and ChatGPT. – Independently published, 2023. – 2032 p.
13. O'Donoghue P., Holmes L. Data Analysis in Sport (Routledge Studies in Sports Performance Analysis). – Routledge, 2014. -

14. Memmert D. Match Analysis: How to Use Data in Professional Sport. – Routledge, 2021.
15. Rabiю Muazu Musa, Anwar P.P. Abdul Majeed, et al. Machine Learning in Team Sports: Performance Analysis and Talent Identification in Beach Soccer & Sepak-takraw (SpringerBriefs in Applied Sciences and Technology). – Springer, 2020.
16. Rabiю Muazu Musa, Anwar P.P. Abdul Majeed, et al. Data Mining and Machine Learning in High-Performance Sport: Performance Analysis of On-field and Video Assistant Referees in European Soccer Leagues. – Springer, 2020.
17. Довбиш А.С., Москаленко В.В. Вступ до інформаційного аналізу і синтезу інфокомунікаційних систем: навч. посіб. - Суми: СумДУ, 2016. - 226 с.
18. Довбиш А. С., Васильєв А. В., Любчак В. О. Інтелектуальні інформаційні технології в електронному навчанні / А. В. Васильєв, А. С. Довбиш, В. О. Любчак. – Суми: Сумський державний університет, 2013. – 177 с.
19. Прилепа Д. В. Інформаційно-екстремальна інтелектуальна технологія діагностування емоційно-психічного стану людини [Електронний ресурс] : дис. ... канд. техн. наук : спец. 05.13.06 : галузь знань 12 / Дмитро Вікторович Прилепа ; наук. керівник Шелехов І. В. ; Сум. держ. ун-т ; Нац. техн. ун-т "Харків. політехн. ін-т". – Харків, 2023. – 188 с. - <https://repository.kpi.kharkov.ua/items/6d53bd36-af3d-41dc-83f8-bca813bcac3d>
20. Bloch J. Effective Java. - Addison-Wesley Professional, 2017. – 412 p.
21. Goetz B. Java Concurrency in Practice. - Addison-Wesley Professional, 2006. – 424 p.
22. Martin R.C. Clean Code: A Handbook of Agile Software Craftsmanship .- Prentice Hall, 2008. - 464 p.
23. Oaks S. Java Performance: The Definitive Guide. - O'Reilly, 2014. – 426 p.

ДОДАТОК

```
package hello;
import java.util.Scanner;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;

public class hello {

    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Input a number of classes: ");
        int m = in.nextInt();
        in.close();
        BufferedImage img = null;
        int N=0, n=0;
        int[][][] y = null, x=null;
        int[][] ev=null;
        double[] avg_y=null;
        double[] E_max = new double[m];
        int[] d_opt = new int[m];
        double[] d1_opt = new double[m];
        double[] betta_opt = new double[m];

        try
        {
            for (int delta=0; delta<250; delta++)
```

```

{

for (int k=0; k<m; k++)
{
    //System.out.printf("\n");
    //y from bmp.files
    //String filename =
"class"+Integer.toString(k+1)+".txt.bmp";
    String filename = Integer.toString(k+1)+".bmp";
    //System.out.println(filename);

    img = ImageIO.read(new File(filename));

    n = img.getHeight();
    N = img.getWidth();
    int rgb, red;

    if (k==0) {y = new int[m][N][n];
    avg_y = new double[N];}
    for (int i = 0; i<N; i++)
    {
        //System.out.printf("\n");
        if (k==0) avg_y[i]=0;
        for (int j = 0; j<n; j++)
            {
                rgb = img.getRGB(i, j);
                red = (rgb >> 16 ) & 0x000000FF;
                y[k][i][j] = red;
            }
        }
    }
}

```

```

//System.out.print(y[k][i][j]);
System.out.print(" ");

if (k==0) avg_y[i]+= red;
//green = (rgb >> 8 ) & 0x000000FF;
//blue = (rgb) & 0x000000FF;
    }
if (k==1) avg_y[i]=(0.5+avg_y[i])/N;
}
//System.out.printf("\n\n");
//for (int i = 0; i<N; i++) {System.out.print(avg_y[i]);
System.out.print(" ");}
//System.out.printf("\n");

//bm+ev

if (k==0)
{
    x = new int[m][N][n];
    ev = new int[m][N];
}

for (int i = 0; i<N; i++)
{
    int avg_x = 0;
    ev[k][i]=0;
    //System.out.printf("\n");
    for (int j = 0; j<n; j++)
    {

```

```

x[k][i][j]=((y[k][i][j]<avg_y[i]+delta)&(y[k][i][j]>avg_y[i]-delta))?1:0;
//      System.out.print(x[k][i][j]); System.out.print("
");

      avg_x+=x[k][i][j];
    }
    if (avg_x>=(n/2.0)) {ev[k][i]=1;};
  }
  //System.out.printf("\n\n");
  //for (int i = 0; i<N; i++) {System.out.print(ev[k][i]);
System.out.print(" ");}
  //System.out.printf("\n");
}

int[] xc = new int[m];
int[] dc = new int[m];

for (int k=0; k<m; k++)
{
  dc[k]=N+1;
  for (int c = 0; c<m; c++)
  {
    if (c!=k)
    {
      int t = 0;
      for (int i = 0; i<N; i++)
      {
        t += Math.abs(ev[k][i]-ev[c][i]);

```



```

        }
        //System.out.printf("%d ", t);
        if (t<=dc[k]) {xc[k]=c; dc[k]=t;}
    }
}
//System.out.printf("\n Dc: %d Xc: %d\n", dc[k], xc[k]);
int[] SK1 = new int[n];
int[] SK2 = new int[n];
for (int j=0; j<n; j++)
{
    SK1[j]=0;
    SK2[j]=0;
    for (int i=0; i<N; i++)
    {
        SK1[j]+=Math.abs(ev[k][i]-x[k][i][j]);
        SK2[j]+=Math.abs(ev[k][i]-x[xc[k]][i][j]);
    }
    //System.out.printf("\n %d %d", SK1[j], SK2[j]);
}
E_max[k]=0;
d_opt[k]=-1;
for (int d=0; d<=N; d++)
{
    double d1=0;
    double betta=0;
    for (int j=0; j<n; j++)
    {
        if(SK1[j]<=d) {d1+=1;}
    }
}

```

```

        if(SK2[j]<=d) {beta+=1;}
        //System.out.printf("\n %d %d %f", j, SK1[j],
d1);
    }
    d1=d1/n;
    beta=beta/n;
    double E=0.5*((d1+1-beta)-(1-
d1+beta))*Math.log((d1+1-beta+1.0/(n+1))/(1-d1+beta+1.0/(n+1)))/Math.log(2);
    //System.out.printf("\n %d %f %f %f", d, E, d1, beta);

    if (d_opt[k]>=0)
    {
        if ((d1>=0.5)&(beta<0.5)&(d<dc[k]))
        {
            if (E>=E_max[k])
            {
                E_max[k]=E;          d_opt[k]=d;
d1_opt[k]=d1; beta_opt[k]=beta;
            }
        }
    }
    else
    {
        if (E>=E_max[k])
        {
            E_max[k]=E;          d1_opt[k]=d1;
beta_opt[k]=beta;
            if ((d1>=0.5)&(beta<0.5)&(d<dc[k]))
            {

```

```
        d_opt[k]=d;
    }
}
}
}
    System.out.printf("\n delta= %3d Class: %2d Emax=%f
dopt=%3d D1=%3.2f betta=%3.2f Xc=%2d dc=%3d", delta, k, E_max[k], d_opt[k],
d1_opt[k], betta_opt[k], xc[k], dc[k]);
}
}
} catch (IOException e) {e.printStackTrace();}
}
}
```