

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

## Кваліфікаційна робота

на здобуття ступеня вищої освіти «Магістр»

«Інформаційна система з використанням згорткових  
нейронних мереж у визначенні рівню стресу за виразом  
обличчя»

(тема кваліфікаційної роботи українською мовою)

«An information system using convolutional neural  
networks in determining the level of stress by facial  
Address Them Through Penetration Testing»

(тема кваліфікаційної роботи англійською мовою)

Виконала: здобувачка денної форми навчання  
спеціальності 122 Комп'ютерні науки .  
(код, назва спеціальності)

Освітня програма Комп'ютерні науки .  
(назва)

Іванова Анастасія Олегівна

(прізвище, ім'я, по-батькові здобувача)

Керівник

к.т.н., доцент, Гнатівська Г.А.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент

к.т.н., доцент каф. математичного забезпечення

КС ОНУ ім. Мечникова, Волощук Л.А.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

Інформаційних технологій .

№      від           2024 р.

Завідувачка кафедри

(підпис)

(прізвище, ім'я)

Захищено на засіданні ЕК №      .

протокол №      від           2024  
р.

Оцінка      /      /      .

(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК

(підпис)

(прізвище, ім'я)

Одеса 2024

## АНОТАЦІЯ

Тема магістерської роботи «Інформаційна система з використанням згорткових нейронних мереж у визначенні рівню стресу за виразом обличчя».

Актуальність магістерської роботи полягає в здійсненні розробки інформаційної системи, яка здатна визначати рівень стресу людини за ознаками виразу обличчя з застосуванням технології згорткових нейронних мереж.

Об'єкт дослідження – процеси проектування та розробки інформаційної системи розпізнавання емоцій за певними ознаками для визначення рівня стресу за виразом обличчя.

Предмет дослідження – методи та алгоритми машинного навчання, зокрема згорткові нейронні мережі (CNN), для аналізу емоційного стану на основі визначених ознак зображень обличчя.

Метою магістерської роботи є створення інформаційної системи, яка використовує CNN для визначення рівня стресу людини за певними ознаками виразу обличчя. Для здійснення проектування та розробки інформаційної системи проведено дослідження та аналіз існуючих підходів до розпізнавання емоцій, обґрунтовано вибір архітектури CNN, визначено процес навчання та валідації моделі, проведено експериментальну оцінку ефективності системи, включаючи аналіз точності, продуктивності та стійкості до змін зовнішніх умов.

Практична цінність магістерської роботи полягає в тому, що розроблена система є готовим програмним продуктом здатним аналізувати відеопотік з веб-камери у реальному часі, класифікуючи емоції та на основі отриманих даних оцінювати рівень стресу користувача.

Ключові слова: СИСТЕМА, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, ОЗНАКИ СТРЕСУ ЛЮДИНИ, ТЕХНОЛОГІЇ КОМП'ЮТЕРНОГО ЗОРУ.

Магістерська робота містить 88 сторінок, 5 таблиць, 13 рисунків, 33 посилань.

## ANNOTATION

The topic of the master's thesis is «An information system using convolutional neural networks in determining the level of stress by facial Address Them Through Penetration Testing».

The relevance of this master's thesis lies in the development of an information system capable of determining a person's stress level based on facial expression features using convolutional neural network technology.

The research object is the design and development processes of an information system for recognizing emotions based on specific features to determine stress levels from facial expressions.

The research subject is machine learning methods and algorithms, specifically Convolutional Neural Networks (CNNs), for analyzing emotional states based on identified features in facial images.

The goal of this master's thesis is to create an information system that uses CNNs to determine a person's stress level based on specific facial expression features. To achieve this, research and analysis of existing approaches to emotion recognition were conducted, the choice of CNN architecture was justified, the model training and validation process was defined, and an experimental evaluation of the system's effectiveness was performed, including analysis of accuracy, performance, and robustness to changing external conditions.

The practical value of the master's thesis lies in the fact that the developed system is a finished software product capable of analyzing a video stream from a webcam in real-time, classifying emotions, and evaluating the user's stress level based on the obtained data.

Keywords: SYSTEM, CONVOLUTIONAL NEURAL NETWORK, HUMAN STRESS INDICATORS, COMPUTER VISION TECHNOLOGIES.

The master's thesis contains 88 pages, 5 tables, 13 figures, and 33 references.

## ЗМІСТ

	Стор.
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП.....	9
<b>1 СТАН ТА НАПРЯМКИ ДОСЛІДЖЕНЬ У СФЕРІ ВИЗНАЧЕННЯ СТРЕСУ ЗА ВИРАЗОМ ОБЛИЧЧЯ.....</b>	<b>10</b>
1.1 Визначення актуальності задачі визначення стресового стану.....	10
1.2 Аналіз підходів до розпізнавання емоцій і стресових станів.....	11
1.3 Використання технологій комп'ютерного зору для аналізу виразу обличчя .....	12
1.4 Методи визначення емоційних станів на основі зображень обличчя...	14
1.5 Ефективність згорткових нейронних мереж у задачах класифікації...	16
1.6 Дослідження стресових ознак та їх зовнішніх прояв через міміку.....	17
1.7 Аналіз сучасних технології для розпізнавання ознак стресових станів .....	19
1.8 Постановка завдання до розробки системи.....	21
Висновки до розділу 1 .....	22
<b>2 ТЕОРЕТИЧНІ ЗАСАДИ ГЛИБИННОГО НАВЧАННЯ ТА НЕЙРОННИХ МЕРЕЖ ДЛЯ АНАЛІЗУ ВИРАЗІВ ОБЛИЧЧЯ .....</b>	<b>24</b>
2.1 Застосування глибинного навчання нейронних мереж .....	24
2.2 Технологія згорткових нейронних мереж .....	26
2.3 Архітектурні рішення згорткових нейронних мереж .....	27
2.4 Особливості навчання згорткових нейронних мереж .....	32
2.6 Застосування згорткових нейронних мереж для завдань аналізу емоційного стану .....	38
2.7 Використання датасетів для аналізу емоційного стану.....	40
2.8 Фреймворки та бібліотеки глибинного навчання .....	42
Висновки до розділу 2.....	46
<b>3 РОЗРОБКА СИСТЕМИ НА БАЗІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ОЦІНКИ РІВНЯ СТРЕСУ ЗА ВИРАЗОМ ОБЛИЧЧЯ .....</b>	<b>47</b>

	7
3.1 Визначення архітектури інформаційної системи .....	47
3.2 Вибір інструментів розробки та налаштування робочого середовища	48
3.3 Проектування архітектури нейронної мережі для аналізу стресу .....	50
3.4 Обробка та розподіл даних для тренування моделі.....	55
3.4.1 Опис датасету .....	55
3.4.2 Класифікація ознак емоцій за стресовим станом.....	56
3.4.3 Аналіз ключових ознак і визначення рівня стресу .....	57
3.4.4 Класифікація рівня стресу.....	59
3.4.5 Розподіл та розширення даних .....	60
3.5 Застосування алгоритмів навчання та валідації згорткової нейронної мережі.....	61
3.6 Розробка ІС .....	65
Висновки до розділу 3.....	70
4 ОЦІНКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНОЇ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ СТРЕСУ .....	71
4.1 Навчання та верифікація моделі .....	71
4.2 Оцінка точності моделі .....	73
4.3 Оцінка ефективності моделі в реальних умовах.....	77
4.4 Оцінка продуктивності системи в реальному часі .....	80
4.5 Рекомендації для вдосконалення системи .....	81
Висновки до розділу 4.....	82
ВИСНОВКИ .....	84
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	85

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

IS – Інформаційна система

API – Application Programming Interface (інтерфейс прикладного програмування)

CNN – Convolutional Neural Network (згорткова нейронна мережа)

CSS – Cascading Style Sheets (каскадні таблиці стилів)

DNN – Deep Neural Network (глибока нейронна мережа)

ЕЕГ – Електроенцефалограма

FACS – Facial Action Coding System (система кодування рухів обличчя)

FNN – Feedforward Neural Network (мережа прямого поширення)

HR – Heart Rate (Частота серцевих скорочень)

HTML – HyperText Markup Language (мова гіпертекстової розмітки)

IS – Інформаційна система

NIR – Near-Infrared (близький інфрачервоний)

RAM – Random Access Memory (оперативна пам'ять)

RNN – Recurrent Neural Network (рекурентна нейронна мережа)

SGD – Stochastic Gradient Descent (стохастичний градієнтний спуск)

SVM – Support Vector Machine (метод опорних векторів)

VGG – Visual Geometry Group (група візуальної геометрії)

## ВСТУП

У сучасному світі, де темп життя стрімко зростає, розуміння емоційного стану людини стає важливим аспектом у багатьох сферах діяльності, включаючи медицину, психологію, безпеку та маркетинг. Розпізнавання емоцій, зокрема через аналіз виразів обличчя, стало однією з найбільш активно розвиваючихся галузей комп'ютерного зору та штучного інтелекту. Використання технологій, які здатні точно виявляти емоційні стани людини, відкриває нові можливості для поліпшення якості життя, зокрема у профілактиці стресу та його наслідків.

Згорткові нейронні мережі стали основним інструментом у розпізнаванні обличчя та виразів емоцій завдяки своїй здатності ефективно обробляти та аналізувати зображення. Моделі на базі CNN демонструють високу точність розпізнавання емоцій, імітуючи процеси, які відбуваються в людському мозку, що дозволяє виявляти складні патерни та зв'язки в даних. Це робить їх ідеальними для автоматизації процесів виявлення емоцій, що, в свою чергу, може бути використано для раннього виявлення стресу у водіїв, вчителів, медичних працівників та інших професій, де емоційний стан має критичне значення [1]. Емоційний стрес може впливати на продуктивність, прийняття рішень і загальний стан здоров'я людини. Хронічний стрес може підвищувати ризик розвитку серцево-судинних захворювань. Тому важливим є не лише виявлення емоцій, але й їхнє правильне оцінювання [2].

Метою магістерської роботи є створення інформаційної системи, яка використовує CNN для визначення рівня стресу людини за певними ознаками виразу обличчя. Для здійснення проектування та розробки інформаційної системи проведено дослідження та аналіз існуючих підходів до розпізнавання емоцій, обґрунтовано вибір архітектури CNN, визначено процес навчання та валідації моделі, проведено експериментальну оцінку ефективності системи, включаючи аналіз точності, продуктивності та стійкості до змін зовнішніх умов.

# 1 СТАН ТА НАПРЯМКИ ДОСЛІДЖЕНЬ У СФЕРІ ВИЗНАЧЕННЯ СТРЕСУ ЗА ВИРАЗОМ ОБЛИЧЧЯ

## 1.1 Визначення актуальності задачі визначення стресового стану

Актуальність задачі визначення стресу за виразом обличчя полягає в її важливості для багатьох сучасних сфер, таких як медицина, безпека, а також у комерційних і соціальних контекстах. Стрес є одним з ключових факторів, що впливає на здоров'я і поведінку людини, і його своєчасне виявлення дозволяє ефективніше контролювати й управляти цією проблемою. Визначення рівня стресу за мімічними проявами має значні переваги, адже міміка відображає природну реакцію організму на стресові ситуації.

Однією з ключових сфер, де це завдання є особливо актуальним, є медицина. Виявлення стресу у пацієнтів може допомогти діагностувати тривожні та психосоматичні розлади, забезпечуючи своєчасне надання медичної допомоги. Наприклад, системи, що використовують аналіз виразу обличчя, можуть використовуватися для моніторингу пацієнтів, які страждають від хронічного стресу або депресії, і допомагати лікарям краще розуміти стан пацієнта в режимі реального часу.

У сфері безпеки визначення стресу може мати вирішальне значення. Наприклад, в аеропортах або на інших об'єктах з високим рівнем безпеки, автоматизовані системи можуть виявляти пасажирів, які нервують або демонструють ознаки стресу через перевезення небезпечних предметів. Люди, що перебувають під сильним стресом, можуть приховувати важливу інформацію, а їхній стан стає ключовою підказкою для служб безпеки. У таких випадках технології виявлення стресу можуть допомогти попередити потенційні загрози та вчасно зупинити небезпечні дії [3].

Соціальні аспекти проблеми також значущі. Емоції, які ми сприймаємо через міміку, дозволяють краще розуміти наміри та почуття інших людей. Це стає особливо важливим у складних соціальних взаємодіях, коли необхідно

запобігти конфліктам або зрозуміти настрої людей у великих групах. Системи автоматичного виявлення стресу дозволяють швидше реагувати на емоційні стани інших, що може бути корисним у різних комунікативних середовищах. З огляду на вищезазначені фактори, задача автоматизованого розпізнавання стресу за виразом обличчя є важливим кроком у розвитку технологій, що дозволяють більш ефективно вирішувати проблеми, пов'язані з безпекою, охороною здоров'я та покращенням взаємодії між людьми та технологіями.

## **1.2 Аналіз підходів до розпізнавання емоцій і стресових станів**

Підхід на основі електроенцефалограми.

Електроенцефалограма (ЕЕГ) дозволяє вимірювати електричну активність мозку, яка змінюється залежно від емоційного стану. Наприклад, бета-хвилі є маркером підвищеної уваги та стресу. Сучасні портативні пристрої з ЕЕГ-датчиками надають можливість збирати дані у реальному часі, що є зручним для моніторингу стресу [4].

Підхід на основі електрокардіограми.

ЕКГ є точним інструментом для вимірювання фізіологічних реакцій на стрес. Показники частоти серцевих скорочень (HR) і варіабельності серцевого ритму (HRV) змінюються під впливом стресових ситуацій. Ці дані є ключовими для аналізу стресу [5].

Підхід на основі рухів тіла та жестів.

Рухи тіла, жести та поза також є індикаторами емоційного стану. Наприклад, схрещені руки, нахил голови або повторювальні рухи можуть вказувати на стрес. Використання камер для аналізу рухів тіла дозволяє створювати автоматизовані системи розпізнавання [6].

Підхід на основі виразу обличчя.

Аналіз виразу обличчя є одним із найбільш інтуїтивно зрозумілих підходів до розпізнавання емоцій. Обличчя людини є основним каналом невербальної комунікації, і його мікровирази можуть миттєво сигналізувати

про внутрішній емоційний стан. Наприклад, злегка зведені брови, стиснуті губи або зморшки на лобі можуть бути ознаками стресу. Цей підхід є менш нав'язливим для користувача і не потребує встановлення додаткових сенсорів на тіло. Сучасні технології комп'ютерного зору дозволяють інтегрувати системи розпізнавання облич у звичайні камери, що значно знижує витрати на обладнання.

У дослідженні [7] використовувалися термальні камери для аналізу змін температури шкіри обличчя, які супроводжують емоційні реакції. Поєднання температурного аналізу та методів машинного навчання дозволило досягти точності 77,5%.

Цей підхід обрано для подальшого аналізу завдяки його високій інтегративності. Він не потребує встановлення додаткових сенсорів на тіло, що робить його менш нав'язливим для користувача. Крім того, сучасні технології комп'ютерного зору дозволяють інтегрувати системи розпізнавання облич у звичайні камери, що значно знижує витрати на обладнання.

### **1.3 Використання технологій комп'ютерного зору для аналізу виразу обличчя**

Розпізнавання емоцій за допомогою комп'ютерного зору є важливим напрямком у сучасних дослідженнях, що вивчає мімічні прояви обличчя для визначення емоційного та когнітивного стану людини. Це дозволяє моделювати складні процеси невербальної комунікації, які раніше потребували значних зусиль з боку людського експерта. Використання комп'ютерного зору в аналізі виразу обличчя є актуальним через його здатність працювати з великими обсягами даних і забезпечувати високу точність у реальному часі.

Ранні роботи в цій галузі, такі як дослідження Пола Екмана у 1970-х роках, виявили зв'язок між базовими емоціями та їх мімічними проявами [8]. Екман також розробив Facial Action Coding System (FACS), яка стала еталоном

для аналізу рухів обличчя (рис.1). Ця система дала змогу визначати м'язові зміни, пов'язані з емоціями, і стала основою для створення перших баз даних обличчя, які використовуються для навчання сучасних алгоритмів машинного навчання.

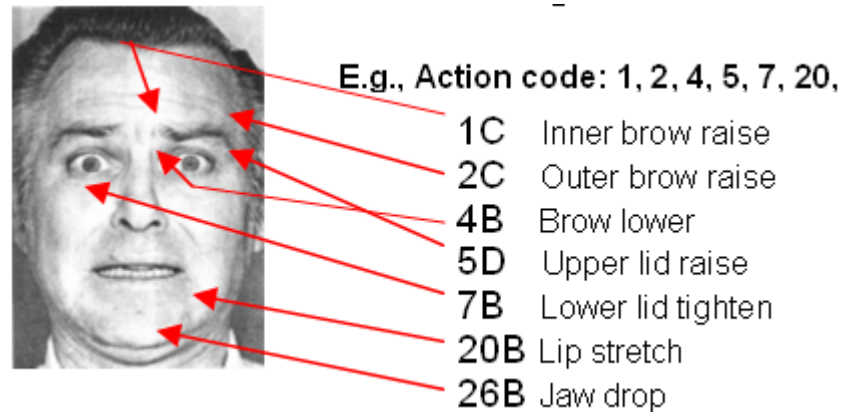


Рисунок 1.1 – приклад FACS

Сьогодні технології комп'ютерного зору використовують глибинні моделі, такі як згорткові нейронні мережі, для автоматизації аналізу міміки. Вони дозволяють системам обробляти великі обсяги зображень або відео, виділяти ключові ознаки обличчя (очі, рот, брови) і визначати емоційні стани.

Виявлення обличчя є першим і найважливішим етапом у процесі розпізнавання міміки. Завдання цього етапу полягає у визначенні та локалізації обличчя на зображенні чи відео. Історично для цього використовувалися такі алгоритми, як метод Віюлі-Джонса, але сучасні системи частіше базуються на згорткових нейронних мережах, які дозволяють виділяти обличчя навіть у складних умовах, наприклад, при зміні освітлення чи частковому перекритті обличчя.

Далі здійснюється попередня обробка зображень. Це важливий етап, який включає нормалізацію яскравості, вирівнювання обличчя і усунення шуму. Наприклад, освітлення може суттєво впливати на якість зображень обличчя, що ускладнює точне визначення емоцій. Методи, такі як вирівнювання

гістограм або згладжування, застосовуються для покращення візуальної якості та полегшення виділення мімичних ознак.

Виділення ключових точок є наступним важливим етапом. Тут визначаються координати основних характеристик обличчя, таких як очі, рот, брови. Розташування та взаємне положення цих точок використовуються для створення математичної моделі, яка відображає мімичні зміни.

На основі отриманих даних виконується класифікація емоційних станів. Глибинні нейронні мережі, зокрема CNN, є основними інструментами для такого аналізу. Вони здатні автоматично виділяти суттєві ознаки і забезпечувати точність, яка значно перевищує можливості традиційних алгоритмів. Наприклад, CNN моделюють ієрархічну структуру ознак, дозволяючи аналізувати як локальні, так і глобальні зміни у виразі обличчя. Додатково можуть використовуватися рекурентні нейронні мережі (RNN), які здатні враховувати послідовність змін міміки у часі, що особливо важливо при аналізі відео [9].

#### **1.4 Методи визначення емоційних станів на основі зображень обличчя**

Традиційні підходи до аналізу виразів обличчя зосереджуються на використанні ручного виділення ознак і систем, заснованих на правилах. Основний принцип полягає у використанні заздалегідь визначених характеристик обличчя, таких як положення брів, форма рота, кут очей тощо, для визначення емоційного стану. Наприклад, зморшки на лобі можуть свідчити про здивування чи стрес. Часто застосовуються алгоритми, які аналізують просторові співвідношення між ключовими точками на обличчі [10].

Перевагою традиційних методів є їхня простота та мінімальні вимоги до обчислювальних ресурсів, що дозволяє використовувати їх у пристроях з обмеженими можливостями. Проте ці методи обмежені у здатності працювати

зі складними або змішаними емоціями, їхня точність суттєво знижується за наявності шуму (наприклад, при поганому освітленні або неправильному положенні обличчя).

Методи машинного навчання значно підвищують точність розпізнавання емоцій шляхом автоматизації виділення ознак із зображень.

Застосовуються такі алгоритми, як:

- SVM (підтримуючі векторні машини), потужний інструмент для класифікації даних завдяки здатності створювати гіперплощину, яка розділяє дані різних класів із максимальною відстанню між ними;
- k-NN (k-ближчий сусід), є одним із найпростіших алгоритмів машинного навчання, який класифікує новий зразок за більшістю класів серед його k найближчих сусідів;
- Random Forest, який створює ансамбль рішень на основі багатьох дерев, кожне з яких тренується на випадковій підмножині даних.

Ці методи дозволяють класифікувати емоції за попередньо визначеними категоріями, такими як щастя, сум або страх. Вони можуть ефективно обробляти середні обсяги даних і підтримують широкий діапазон застосувань. Однак вони потребують якісного навчання на великих наборах даних і чутливі до артефактів або неякісних зображень, що обмежує їх використання у реальних умовах.

Глибокі нейронні мережі стали стандартом для розпізнавання емоцій за зображеннями обличчя. Основні технології включають:

- згорткові нейронні мережі – основа для роботи з візуальними даними. Вони складаються з згорткових шарів, які виділяють ключові особливості зображень, такі як текстури, контури та форми. У контексті аналізу обличчя CNN здатні автоматично ідентифікувати особливості, що відповідають певним емоціям;
- рекурентні нейронні мережі: орієнтовані на аналіз послідовностей даних. У розпізнаванні емоцій вони застосовуються для аналізу змін

виразів обличчя у відеопослідовностях, дозволяючи виявляти динаміку емоцій;

- автоенкодера: використовуються для зменшення розмірності даних та виділення основних характеристик. У контексті аналізу емоцій вони допомагають зменшити шум і виділити ключові особливості обличчя.

Головною перевагою глибокого навчання є можливість автоматичного виділення складних ознак без необхідності ручної роботи. Такі методи демонструють високу точність навіть у складних умовах і здатні працювати з великими наборами даних. Однак вони вимагають значних обчислювальних ресурсів, довготривалого навчання моделей і великих обсягів навчальних даних.

## **1.5 Ефективність згорткових нейронних мереж у задачах класифікації**

Штучні нейронні мережі є важливим інструментом у сучасному машинному навчанні, а їхній розвиток значно вплинув на автоматизацію задач класифікації, розпізнавання та прогнозування. Серед різноманітних типів нейронних мереж особливу увагу привертають CNN, які демонструють найвищу ефективність у задачах обробки зображень і розпізнавання образів. CNN були спеціально розроблені для роботи з візуальними даними, що робить їх незамінними для задач комп'ютерного зору, таких як класифікація зображень, розпізнавання обличчя, обробка відео та інші.

Основною перевагою CNN є їхня здатність автоматично виявляти та навчатися розпізнаванню ключових особливостей у зображеннях. Це досягається за рахунок використання шарів згортки, які виконують операції фільтрації зображення, і шарів підвибірки (pooling), які зменшують розмірність вхідних даних, зберігаючи найважливішу інформацію. Така структура дозволяє моделям фокусуватися на суттєвих деталях, зокрема

лініях, текстурах або контрастах, і використовувати ці ознаки для подальшої класифікації [11].

У задачах розпізнавання стресу за виразом обличчя CNN є найефективнішим інструментом завдяки їх здатності аналізувати навіть найдрібніші зміни у мімічних проявах. Крім того, CNN дозволяють інтегрувати додаткові параметри, такі як текстура шкіри або зміни у пропорціях обличчя, що сприяє більш точному визначенню рівня стресу.

На відміну від традиційних алгоритмів класифікації, які вимагають ручного визначення ознак, CNN автоматизують цей процес завдяки своїй багатоплановій структурі. Це значно знижує потребу в попередньому аналізі даних, одночасно підвищуючи адаптивність моделі до нових умов.

Таким чином, CNN є не лише найефективнішим інструментом для розпізнавання образів, але й дозволяють створювати системи, які можуть працювати в реальному часі, аналізуючи відеопотоки або великі набори статичних зображень. Їхні можливості інтегрування з іншими технологіями, такими як OpenCV, PyTorch або TensorFlow, забезпечують широкий спектр застосувань – від медичної діагностики до систем безпеки та моніторингу. Це робить CNN ідеальним вибором для задач розпізнавання стресу та інших складних емоційних станів.

## **1.6 Дослідження стресових ознак та їх зовнішніх прояв через міміку**

Стрес є природною адаптаційною реакцією організму на виклики або загрози, що активує симпатичну нервову систему та викликає фізіологічні й емоційні зміни. Під впливом стресу підвищується серцевий ритм, артеріальний тиск, а також відбувається викид гормонів, таких як адреналін і кортизол. На емоційному рівні стрес зазвичай супроводжується негативними емоціями, такими як страх, гнів, тривога, сум або відчай. Ці емоції тісно

пов'язані з активністю м'язів обличчя, що дозволяє ідентифікувати емоційний стан людини за допомогою аналізу міміки.

Негативні емоції мають свої характерні прояви у виразах обличчя. Страх часто проявляється через підняті брови, утворення зморшок на лобі, розширені зіниці та напружений рот, що може бути злегка відкритим. Гнів супроводжується нахмуреними бровами, звуженими очима, стисканням губ та стиснутою щелепою. Сум характеризується опущеними куточками рота, напівзакритими очима та підняттям внутрішніх частин брів, що утворюють нахилені зморшки на лобі. Тривога може включати часте моргання, напруженість погляду, злегка підняті брови та стискання губ. Відчай зазвичай супроводжується відсутністю фокусування погляду, мінімальною активністю м'язів обличчя та опущеними куточками рота.

Окрім тривалих виразів, важливим аспектом аналізу міміки є мікрОВИРАЗИ – короткочасні рухи м'язів обличчя, які тривають від 1/25 до 1/2 секунди [12].

Вони часто виникають під час стресу, коли людина намагається приховати справжні емоції. МікрОВИРАЗИ є результатом автоматичної активності лімбічної системи, яка створює несвідомі сигнали, що «прориваються» через спроби свідомого контролю (рис. 1.2).

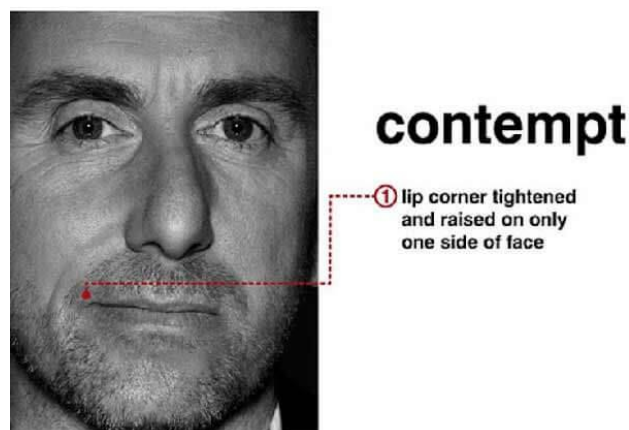


Рисунок 1.2 – Приклад мікрОВИРАЗИ презирство

Аналіз міміки для визначення рівня стресу вимагає врахування як статичних, так і динамічних змін обличчя. Нейронна мережа, яка виконує подібний аналіз, повинна розпізнавати типові ознаки базових емоцій, таких як зміни у положенні брів, губ, очей, щелепи, та виявляти мікроекспресії, які можуть сигналізувати про приховані емоції. Особливу увагу слід приділяти динамічному характеру змін: перехід від нейтрального виразу до ознак страху чи гніву може вказувати на підвищений рівень стресу. Важливим є також аналіз індивідуальних особливостей, таких як базова міміка обличчя в спокійному стані, що дозволяє уникнути хибних трактувань

### **1.7 Аналіз сучасних технологій для розпізнавання ознак стресових станів**

Розпізнавання стресу за виразами обличчя є важливою складовою автоматизованих систем, що допомагають оцінювати емоційний стан людини. Це питання набуло великої актуальності в багатьох країнах завдяки розвитку комп'ютерного зору та глибокого навчання, що дозволяють створювати більш точні і надійні системи розпізнавання стресу.

Технології розпізнавання стресу розвиваються швидкими темпами, знаходячи застосування в різних галузях. В автомобільній промисловості компанії, такі як Affectiva та Smart Eye, розробляють системи моніторингу стану водія, які використовують комп'ютерний зір та аналіз міміки для виявлення ознак втоми, сонливості або стресу. Ці системи можуть попередити водія про небезпеку або навіть активувати системи допомоги водієві. Хоча деякі виробники автомобілів вже пропонують подібні функції, вони часто базуються на аналізі поведінки автомобіля та рухів водія, а не на розпізнаванні емоцій.

Розпізнавання обличчя та аналіз емоцій досліджуються в контексті громадської безпеки. В деяких країнах, включаючи Китай, розпізнавання обличчя використовується в системах відеоспостереження для ідентифікації

та відстеження людей. Проте, масове застосування систем розпізнавання емоцій в громадських місцях наразі обмежене через етичні та правові питання, пов'язані з приватністю. Дослідження в цій області продовжуються, але їхнє практичне використання для моніторингу емоційного стану населення ще потребує ретельного вивчення та регулювання.

В Україні технології розпізнавання обличчя та емоцій тільки починають активно впроваджуватися. Перші кроки були зроблені в сфері безпеки, де вже зараз деякі підприємства використовують системи контролю доступу, засновані на розпізнаванні обличчя. Розпізнавання емоцій та стресу в таких системах допомагає не лише ідентифікувати особу, а й оцінювати емоційний стан співробітників для покращення умов праці та підвищення продуктивності.

В сфері безпеки та нагляду деякі стартапи в Україні вже розвивають системи, здатні аналізувати емоційний стан підозрілих осіб на основі відеозаписів, що може бути корисним для правоохоронних органів.

У світі існують численні системи для автоматичного визначення рівня стресу на основі аналізу виразів обличчя, таких як FaceReader, EmoDetect, і Affectiva, які використовують штучний інтелект для детекції емоцій і стресу в реальному часі. Ці системи використовують згорткові нейронні мережі, що дозволяють з високою точністю аналізувати мимічні зміни на обличчі людини та визначати рівень стресу.

FaceReader – це одна з найбільш відомих систем, яка здатна виявляти і класифікувати до 7 основних емоцій, таких як гнів, радість, сум, здивування та стрес. Вона використовує камери високої роздільної здатності для збору даних про вирази обличчя, які потім аналізуються алгоритмами глибокого навчання [14].

Іншою популярною системою є Affectiva, яка спеціалізується на аналізі емоцій через відео та аудіо дані, забезпечуючи точність розпізнавання емоцій у реальному часі. Ця система активно використовується в автомобілях для

моніторингу водіїв, а також в маркетингових дослідженнях для визначення реакції споживачів на рекламу [15].

В Україні також з'являються стартапи, що розвивають подібні системи. Однак на сьогодні вони більше фокусуються на розпізнаванні базових емоцій, ніж на детальному аналізі стресу. Проте інтерес до таких технологій постійно зростає, і в найближчому майбутньому можна очікувати на їхнє розширення в медичній та корпоративній сферах.

## **1.8 Постановка завдання до розробки системи**

Автоматизоване розпізнавання емоційного стану, зокрема стресу, є складним завданням, яке вимагає вирішення багатьох технічних, етичних і практичних викликів. Технології повинні враховувати широкий спектр факторів, від точності виявлення до адаптації під індивідуальні особливості.

Серед найважливіших аспектів є забезпечення конфіденційності даних, їх якість і кількість, а також розробка моделей, здатних працювати в реальних умовах. Зокрема, виклики виникають через динаміку емоційних проявів, вплив зовнішніх умов, таких як освітлення чи фон, і необхідність інтеграції даних із різних джерел для більшої надійності [13].

Основні виклики завдань автоматизованого розпізнавання емоційного стану:

- точність і надійність: необхідна висока точність для розпізнавання тонких змін у поведінці, які ускладнюються зовнішніми факторами (зачіска, макіяж тощо);
- індивідуальні відмінності: емоції та риси обличчя різняться у людей, що ускладнює створення універсальних моделей;
- оклюзії: часткові закриття обличчя потребують стійких методів розпізнавання;
- динаміка: вирази обличчя мають часову динаміку, що вимагає моделювання для точності;

- інтеграція даних: залучення фізіологічних, поведінкових і контекстуальних даних вимагає складних алгоритмів;
- довгострокові виклики: переважна розробка систем у контрольованих умовах знижує їх ефективність у реальних сценаріях;
- якість і кількість даних: для продуктивності потрібні великі й різноманітні дані;
- маркування: автоматизація маркування даних зменшує трудомісткість, але суб'єктивність потребує узгодженості;
- уникнення упередженості: використання різноманітних наборів даних підвищує універсальність алгоритмів.

Для подолання зазначених викликів у сфері автоматизованого розпізнавання емоцій і стресу слід впроваджувати комплексні підходи. Зокрема, важливим є застосування сучасних алгоритмів машинного та глибокого навчання, здатних розпізнавати тонкі зміни в емоціях і стресових станах. Необхідно розробляти адаптивні моделі, які враховують унікальні поведінкові та фізіологічні особливості користувачів, що підвищить точність системи.

Інтеграція мультимодальних даних, таких як міміка, жести чи серцевий ритм, дозволить значно підвищити точність аналізу емоцій. Маркування даних можна автоматизувати за допомогою спеціалізованих інструментів, зберігаючи можливість ручного уточнення для забезпечення якості.

Для роботи в реальних умовах необхідно створювати алгоритми, стійкі до впливу факторів, таких як часткові закриття обличчя чи погане освітлення. Водночас варто вдосконалювати методи моделювання динаміки емоцій, що враховують часові особливості змін виразів обличчя.

## **Висновки до розділу 1**

У розділі досліджено актуальність задачі визначення стресу за виразом обличчя, підкреслено її важливість для різних сфер, таких як медицина,

безпека та соціальна взаємодія. Проаналізовано різні підходи до розпізнавання емоцій та стресу, включаючи аналіз ЕЕГ, ЕКГ, рухів тіла, та детальніше розглянуто підхід на основі аналізу міміки за допомогою комп'ютерного зору. Визначено переваги використання CNN для вирішення цієї задачі, зокрема їхню здатність до автоматичного виявлення ознак та обробки великих обсягів даних. Окреслено основні виклики, пов'язані з завданням автоматизованого розпізнавання стресу, такі як індивідуальні відмінності в проявах стресу, оклюзії обличчя та необхідність високої точності. Аналіз існуючих систем виявив зростаючий інтерес до цієї теми та наявність різних підходів до її вирішення. На основі проведеного аналізу обрано підхід на основі CNN та аналізу міміки для подальшої розробки в рамках дипломної роботи.

## **2 ТЕОРЕТИЧНІ ЗАСАДИ ГЛИБИННОГО НАВЧАННЯ ТА НЕЙРОННИХ МЕРЕЖ ДЛЯ АНАЛІЗУ ВИРАЗІВ ОБЛИЧЧЯ**

### **2.1 Застосування глибинного навчання нейронних мереж**

Нейронні мережі є основою багатьох сучасних технологій штучного інтелекту, зокрема глибокого навчання. Глибоке навчання є підгалуззю машинного навчання, яка використовує багатошарові нейронні мережі для автоматичного виявлення складних патернів у великих масивах даних. Від традиційних методів машинного навчання глибоке навчання відрізняється здатністю здійснювати багатоетапну обробку даних, де кожен шар мережі автоматично виділяє важливі ознаки з сирих даних. Така структура дозволяє глибоким нейронним мережам досягати високої ефективності при вирішенні складних завдань, таких як розпізнавання образів, природної мови, голосу та емоцій.

Нейронні мережі, що є основою глибокого навчання, представляють собою математичні моделі, натхнені структурою людського мозку. Вони складаються з нейронів, які з'єднані між собою через синапси (з'єднання), що передають інформацію між елементами мережі. Кожен нейрон здійснює обчислення, застосовуючи зважену суму вхідних сигналів і функцію активації, яка визначає вихід нейрона. Різні типи нейронних мереж можуть мати різну архітектуру, що робить їх більш підходящими для конкретних завдань [16].

Особливе місце серед нейронних мереж займають глибокі нейронні мережі (DNN), які складаються з численних прихованих шарів і здатні автоматично виділяти ознаки з великих обсягів даних. Вони здобули велику популярність завдяки їхній здатності до самонавчання, що дозволяє створювати моделі, які самостійно покращують свої результати за допомогою великої кількості даних. Глибокі нейронні мережі мають широке застосування в таких сферах, як комп'ютерний зір, розпізнавання мовлення, обробка природної мови та, зокрема, в аналізі емоцій та стресу.

Основні типи нейронних мереж, які застосовуються у глибокому навчанні, включають:

- мережі прямого поширення (FNN) – це найпростіший тип нейронних мереж, де інформація проходить лише в одному напрямку, від вхідного шару до вихідного, і не має циклічних з'єднань;
- згорткові нейронні мережі – спеціалізуються на обробці зображень та відео. вони використовують операцію згортки для виділення важливих просторових ознак з вхідних даних;
- рекурентні нейронні мережі – використовуються для обробки послідовностей, таких як текст, відео або часові ряди, де важливо враховувати контекст та попередні етапи обробки.

Глибоке навчання включає не лише розпізнавання шаблонів у даних, а й здатність здійснювати самонавчання та покращення моделі з часом. Застосування цього підходу значно змінило методи вирішення задач в різних галузях, таких як медицина, фінанси, безпека, а також в аналізі емоцій та стресу.

Навчання нейронних мереж може бути:

- навчання з учителем (supervised learning), де модель навчається на основі вхідних даних з відомими мітками (правильними відповідями). це найпоширеніший підхід у задачах класифікації, таких як розпізнавання емоцій за зображеннями;
- навчання без учителя (unsupervised learning), де модель шукає структуру в даних без відомих міток, що є корисним у задачах кластеризації або виявлення аномалій;
- навчання з підкріпленням (reinforcement learning), де модель взаємодіє з середовищем і отримує винагороду чи покарання за свої дії, що дозволяє вчитися через досвід.

Ці підходи дозволяють використовувати глибоке навчання для вирішення різноманітних завдань, від простого розпізнавання об'єктів до

складних задач прогнозування та класифікації, таких як виявлення емоцій або стресу

## 2.2 Технологія згорткових нейронних мереж

Згорткові нейронні мережі є одним із ключових інструментів глибокого навчання, що знаходять застосування у різноманітних завданнях, таких як класифікація зображень, розпізнавання облич, обробка мовлення та відеоаналітика. Вони використовуються в багатьох сучасних технологіях, забезпечуючи їхню високу ефективність та адаптивність.

Історія згорткових нейронних мереж розпочалася у 1980-х роках із робіт Яна Лекуна, який вперше застосував CNN для автоматизованого розпізнавання рукописних цифр у системах обробки пошти. Його модель, відома як LeNet-5, була однією з перших практичних реалізацій, що змогла ефективно виконувати класифікацію зображень завдяки використанню згорткових шарів і шарів субдискретизації [17]. Протягом наступних десятиліть розвиток CNN був повільним через обмежені обчислювальні ресурси та недостатню кількість даних для навчання. Ситуація змінилася на початку 2010-х років, коли з'явилися великі датасети, такі як ImageNet, і потужні графічні процесори (GPU), що зробили можливим тренування більш складних моделей. Прорив стався у 2012 році, коли команда під керівництвом Джеффри Хінтона представила модель AlexNet, яка значно перевершила всі попередні методи у змаганнях з класифікації зображень. Це викликало справжній бум у дослідженнях та застосуванні згорткових нейронних мереж. Надалі були створені ще більш потужні архітектури, такі як VGG, ResNet і Inception, які розширили межі можливостей CNN, зробивши їх основним інструментом у галузі комп'ютерного зору.

CNN вирізняються своєю здатністю автоматично виділяти ключові ознаки вхідних даних завдяки застосуванню згорткових операцій. На відміну від класичних алгоритмів обробки зображень, CNN не потребують значної

попередньої обробки чи ручного налаштування параметрів. Замість цього, неймережі навчаються виявляти особливості самостійно, використовуючи механізм фільтрів (або ядер згортки), які під час навчання налаштовуються відповідно до специфіки завдання.

Основу CNN складають три ключові компоненти:

- локальні рецептивні поля (local receptive fields): кожен нейрон отримує інформацію лише з обмеженого регіону вхідних даних, що дозволяє зберігати просторову структуру та виявляти локальні закономірності;
- колективні ваги (convolutional kernels): один набір ваг використовується для всієї карти ознак, що суттєво зменшує кількість параметрів і підвищує ефективність моделі;
- субдискретизація (pooling): зменшення розмірів зображення шляхом об'єднання близьких значень, що зменшує обсяг оброблюваних даних, зберігаючи при цьому найважливіші ознаки.

Згорткові нейронні мережі мають низку переваг, що роблять їх особливо ефективними для обробки зображень та інших типів даних зі збереженням просторових залежностей. Однією з головних переваг є здатність автоматично виділяти важливі ознаки з даних без необхідності ручної інженерії, що значно спрощує процес побудови моделей. Мережі є стійкими до зсувів, масштабування та деформацій вхідних об'єктів, що робить їх надійними для різноманітних завдань, таких як класифікація, сегментація чи розпізнавання об'єктів

### **2.3 Архітектурні рішення згорткових нейронних мереж**

Згорткова нейронна мережа є одним із найпопулярніших видів глибокого навчання, що ефективно розпізнає зображення, об'єкти та інші візуальні патерни. Її архітектура побудована з кількох основних шарів, кожен із яких виконує спеціалізовані функції (рис.2.1). Розглянемо ці шари детально.

Основні шари CNN:

- вхідний шар;
- згортковий шар;
- шар активації;
- шар підвибірки;
- повнозв'язний шар;
- вихідний шар.

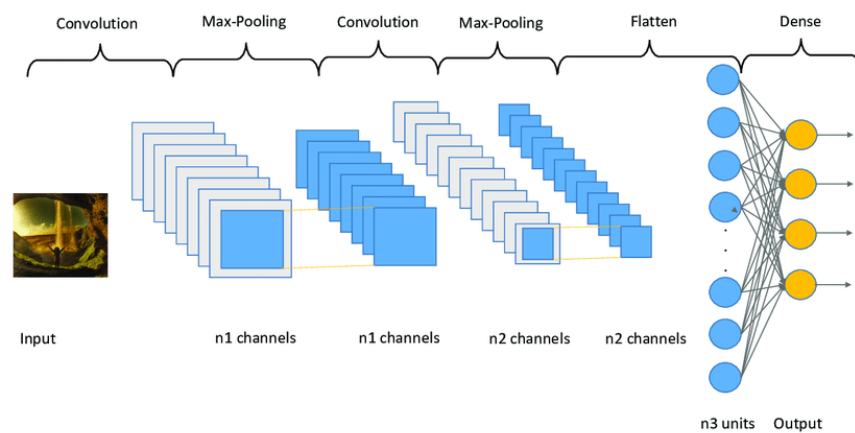


Рисунок 2.1 – Загальна архітектура CNN

Вхідний шар приймає дані, наприклад, зображення у вигляді тривимірного масиву  $H \times W \times C$ , де  $H$  – висота,  $W$  – ширина,  $C$  – кількість каналів (зазвичай 3 для RGB).

Згортковий шар є базовим будівельним блоком CNN, призначеним для виділення локальних ознак із вхідних даних. Шар використовує фільтри (ядра згортки), які ковзають по зображенню, виконуючи операції згортки. Кожен фільтр реагує на певний набір ознак, таких як краї, текстури чи кути. Результатом роботи шару є карта ознак (feature map), яка представляє локальні патерни у вхідних даних (рис 2.2). Перевагою є використання однакових ваг

для всіх позицій фільтра, що знижує обчислювальну складність і забезпечує стійкість до зсувів [18].

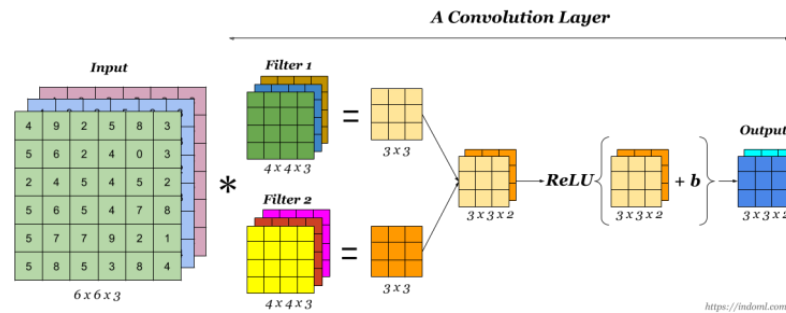


Рисунок 2.2 – Згортковий шар

Для отримання карти ознак виконується декілька операцій.

Операція згортки. Фільтр (ядро)  $K$  розміром  $k \times k$  проходить по зображенню, обчислюючи згортку з частинами зображення (рецептивним полем) за формулою:

$$Y[i, j] = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} X[i + m, j + n] \cdot K[m, n] + b, \quad (2.1)$$

де  $Y[i, j]$  – вихідний піксель;

$X[i + m, j + n]$  – пікселі вхідного зображення;

$K[m, n]$  – ядро;

$b$  – зміщення (bias).

Рецептивне поле: це область вхідного зображення, яку аналізує ядро.

Наприклад, для ядра  $3 \times 3$ , рецептивне поле має розмір 3 на 3.

Гіперпараметри: Розмір ядра ( $k$ ) визначає, яку частину зображення аналізує кожен фільтр.

Крок (stride) – визначає, на скільки пікселів пересувається фільтр.

Заповнення (padding) – додає рамку з нульових значень, щоб зберегти розмір вихідного зображення.

Вихідний розмір обчислюється за формулою:

$$H_{out} = \frac{H_{in-k+2p}}{s} + 1, \quad (2.2)$$

де  $H_{in}$  – висота вхідного зображення;  
 $k$  – розмір ядра;  
 $p$  – кількість заповнень;  
 $s$  – крок згортки.

Шар активації (ReLU, Tanh, Sigmoid).

Після згорткових операцій результати проходять через нелінійні функції активації, такі як ReLU (Rectified Linear Unit), яка задає елементну нелінійність (рис 2.3). ReLU замінює всі негативні значення на нуль, забезпечуючи швидшу і стабільнішу роботу в порівнянні з традиційними функціями, такими як сигмоїда або тангенс.

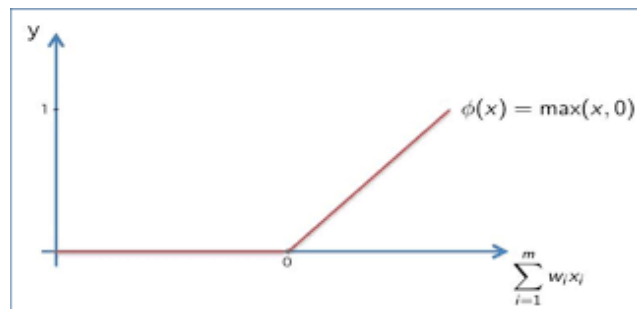


Рисунок 2.3 – Функція активації ReLU

Її визначають за формулою:

$$f(x) = \max(x, 0) \quad (2.3)$$

Ця функція замінює всі негативні значення на нуль, зберігаючи позитивні без змін.

Інші функції активації:

1) Sigmoid:  $f(x) = \frac{1}{1+e^{-x}}$  – використовується для задач класифікації.

2) Tanh:  $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$  – нормалізує значення між -1 і 1.

Шар підвибірки (Pooling Layer).

Шар підвибірки зменшує розмірність даних, зберігаючи основну інформацію. Це дозволяє знизити обчислювальні витрати та уникнути перенавчання (рис 2.4).

Типи підвибірки:

- максимальна підвибірка (max pooling): обирає найбільше значення з кожного підрегіону;
- середня підвибірка (average pooling): обчислює середнє значення всіх елементів у підрегіоні.

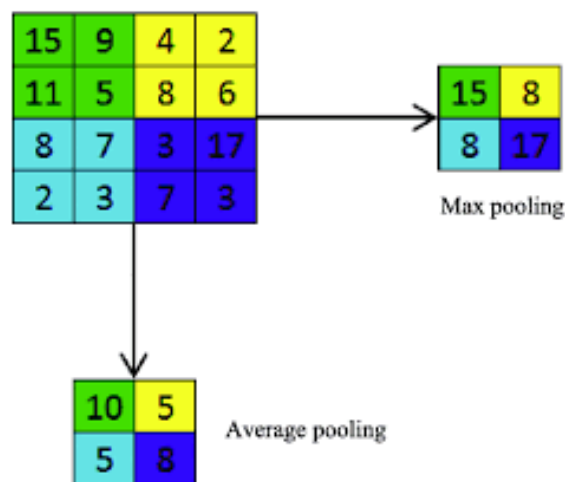


Рисунок 2.4 – Максимальна та середня підвибірка

Повнозв'язний шар (Fully Connected Layer).

Повнозв'язний шар знаходиться на завершальних етапах мережі та відповідає за інтеграцію виділених ознак для класифікації або прогнозування. Кожен нейрон у цьому шарі з'єднаний з усіма нейронами попереднього шару, що дозволяє створювати складні нелінійні комбінації ознак.

Він з'єднує всі нейрони попереднього шару з усіма нейронами цього шару та виконує класифікацію на основі ознак.

Формула:

$$Y = W \cdot X + b \quad (2.4)$$

де  $W$  – матриця ваг;  
 $X$  – вхідний вектор;  
 $b$  – вектор зміщення.  
 Вихідний шар.

Вихідний шар містить нейрони, що відповідають класам у задачі класифікації. Для багатокласової класифікації використовується softmax-функція за таким визначенням:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.5)$$

де  $z_i$  – активація  $i$ -го нейрона;  
 $K$  – кількість класів.

## 2.4 Особливості навчання згорткових нейронних мереж

Навчання згорткових нейронних мереж – це процес, у якому мережа налаштовує свої параметри (включаючи ваги та зміщення) для точного виконання завдань, таких як класифікація, сегментація або виявлення об'єктів. Усі етапи навчання спрямовані на оптимізацію ваг так, щоб модель могла правильно обробляти як навчальні, так і нові дані.

Підготовка вхідних даних є першим етапом навчання згорткових нейронних мереж і визначає якість їх подальшої роботи. Дані зазвичай нормалізуються, що передбачає масштабування значень пікселів до діапазону від 0 до 1 або від -1 до 1. Це забезпечує стабільність під час навчання та знижує ризик виникнення числових нестабільностей. Також виконується аугментація

даних, яка включає обертання, зсув, масштабування, зміну яскравості або колірного тону зображень. Цей процес збільшує різноманітність даних і дозволяє моделі стати більш стійкою до змін у реальних умовах. Підготовка завершується розділенням даних на навчальну, валідаційну та тестову вибірки, щоб забезпечити ефективне навчання, перевірку та оцінку моделі [19].

На етапі прямого поширення сигналу дані проходять через усі шари мережі, де кожен шар трансформує їх для вилучення ознак. На згорткових шарах обчислюються активації шляхом згортки вхідних матриць із ядрами. Цей процес виявляє локальні ознаки, такі як краї, кути або текстури. Потім результат згортки проходить через нелінійну функцію активації, наприклад, ReLU, яка обнулює негативні значення, зберігаючи позитивні. Шари підвибірки зменшують розмірність даних, використовуючи операції, такі як Max Pooling, щоб залишити найважливіші ознаки. На вихідному шарі модель генерує прогнози у вигляді ймовірностей, використовуючи, наприклад, функцію softmax, яка нормалізує вихідні значення.

Функція втрат використовується для вимірювання розбіжності між прогнозами моделі та істинними мітками. Наприклад, для класифікаційних задач поширеною є крос-ентропія, яка обчислює логарифмічну втрату на основі ймовірностей, передбачених моделлю. Для регресійних задач використовується середньоквадратична похибка, що оцінює різницю між передбаченим і справжнім значеннями. Значення функції втрат служить основним індикатором ефективності моделі.

Зворотне поширення помилки є ключовим етапом навчання. Воно включає обчислення градієнтів функції втрат щодо кожного параметра моделі. Градієнти поширюються назад від вихідного шару до вхідного, дозволяючи коригувати ваги мережі. Цей процес забезпечує, що зміни ваг зменшують помилку моделі. Математично градієнти обчислюються як похідні функції втрат за вагами, а їх передача через шари виконується відповідно до правила ланцюга.

Оновлення ваг здійснюється за допомогою алгоритмів оптимізації. Базовий алгоритм, градієнтний спуск, оновлює ваги на основі градієнтів і швидкості навчання. Адаптивні алгоритми, такі як Adam, враховують перші й другі моменти градієнтів, автоматично регулюючи швидкість навчання для кожного параметра. Це дозволяє більш ефективно навчати модель, особливо на великих наборах даних. Завдяки поетапній оптимізації ваг мережа поступово вдосконалює свої прогнози, зменшуючи функцію втрат до мінімального значення.

## 2.5 Завдання оптимізації навчання

Оптимізація навчання в нейронних мережах є однією з основних задач, яка спрямована на досягнення кращої здатності моделі узагальнювати на нових даних. Це досягається шляхом налаштування параметрів моделі (ваг) з урахуванням алгоритмів оптимізації та регуляризації [20].

Регуляризація – це набір методів, що дозволяють покращити загальну ефективність моделі, запобігаючи її перенавчанню (overfitting). Перенавчання виникає, коли модель дуже точно відображає шуми в навчальних даних, що знижує її здатність до узагальнення на нові, невідомі дані.

Раннє зупинення (Early Stopping).

Одним з найпоширеніших способів регуляризації є раннє зупинення навчання. Метод полягає в тому, що тренування моделі припиняється, коли вона досягає найкращих результатів на валідаційній вибірці, і після цього починає демонструвати погіршення результатів. Це дозволяє уникнути перенавчання і зберегти оптимальні параметри моделі.

Аугментація даних.

Аугментація даних є ефективною стратегією збільшення кількості навчальних зразків без необхідності отримувати нові дані. Це дозволяє моделі навчатися на різноманітних варіаціях вхідних даних, що робить її більш стійкою до шуму та варіацій у реальному світі.

Для зображень це може включати: обертання, перевертання, масштабування, додавання шуму, зміну яскравості або контрасту. Для текстових даних аугментація може бути виконана через заміну синонімів або перетасування слів.

Dropout. Метод Dropout є дуже популярною технікою регуляризації, яка полягає у випадковому вимкненні певної кількості нейронів під час навчання. Це змушує модель не залежати від певних нейронів і вчитися більш узагальненим закономірностям. Моделі з Dropout зазвичай показують кращу здатність до узагальнення і знижують ризик перенавчання.

L1 та L2 регуляризація.

L1 та L2 регуляризації додають штраф до функції втрат, щоб змусити модель використовувати менше ресурсів для розв'язання задачі, що допомагає запобігти перенавчанню.

L1 регуляризація (Lasso): додає до функції втрат суму абсолютних значень ваг, що може призвести до того, що деякі ваги стануть рівними нулю, що дозволяє створювати більш розріджені моделі.

L2 регуляризація (Ridge): додає до функції втрат суму квадратів ваг. Це сприяє зменшенню величини всіх ваг, що допомагає уникнути надмірної складності моделі.

Додавання шуму до ваг та вхідних даних.

Шум може бути доданий до ваг моделі або до вхідних даних, що дозволяє створювати стійкіші моделі. Це робить модель менш чутливою до незначних варіацій у вхідних даних, що може сприяти кращій узагальненій здатності.

Процес оптимізації навчання включає в себе коригування ваг моделі так, щоб мінімізувати функцію втрат. Для цього використовуються алгоритми, які здатні обчислювати градієнти функції втрат по відношенню до кожного параметра (ваги) і поступово оновлювати ці параметри, рухаючись у напрямку, який зменшує помилку.

Алгоритми оптимізації забезпечують ефективне коригування ваг за допомогою чисельних методів, таких як стохастичний градієнтний спуск (SGD) і його варіації.

Стохастичний градієнтний спуск (SGD) – це один з основних алгоритмів оптимізації, що полягає в тому, щоб обчислювати градієнт функції втрат на кожному зразку даних або на випадковому міні-батчі даних і використовувати його для оновлення параметрів.

Оновлення параметрів здійснюється за формулою:

$$W_{t+1} = W_t - \eta \nabla_t, \quad (2.5)$$

де  $W_t$  – це значення параметрів на кроці  $t$ ;

$\eta$  – швидкість навчання;

$\nabla_t$  – градієнт функції втрат по відношенню до параметрів на кроці  $t$ .

Переваги: швидкість обчислень, ефективність на великих вибірках.

Недоліки: нестабільність, оскільки градієнт обчислюється на одному зразку або міні-батчі.

SGD з моментом (Momentum) – це метод який додає до стандартного SGD термін, який враховує попередні оновлення, що допомагає згладити оновлення і прискорити процес навчання, особливо при роботі з глибокими мережами.

Оновлення параметрів відбувається за допомогою додаткового коефіцієнта, що враховує попередні оновлення за таким визначенням:

$$V_t = \beta V_{t-1} + (1 - \beta) \nabla_t, \quad W_{t+1} = W_t - \eta V_t \quad (2.6)$$

Переваги: швидка збіжність, зменшення коливань.

Недоліки: потребує налаштування додаткових параметрів.

RMSprop адаптує швидкість навчання для кожного параметра, враховуючи середньоквадратичне значення градієнтів. Це дозволяє уникнути

проблем, що виникають при дуже великих або дуже малих значеннях градієнтів для певних параметрів.

Переваги: адаптивність швидкості навчання, стабільність.

Недоліки: потребує налаштування гіперпараметрів.

Adagrad змінює швидкість навчання для кожного параметра залежно від частоти його оновлень. Для параметрів, що часто оновлюються, швидкість навчання зменшується, а для рідших збільшується.

Недолік: з часом швидкість навчання може знижуватися до дуже малих значень, що може сповільнити навчання.

Adam (Adaptive Moment Estimation) є одним з найпопулярніших алгоритмів оптимізації, який поєднує ідеї моменту і RMSprop, що дозволяє адаптивно змінювати швидкість навчання для кожного параметра. Цей алгоритм є дуже ефективним для великих даних та складних моделей, де використовується наступна формула:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \quad (2.7)$$

де  $m_t$  і  $v_t$  – це оцінки першого і другого моменту для градієнта.

Переваги: висока швидкість збіжності, стабільність навіть на складних задачах.

Недоліки: може бути схильним до перенавчання на малих вибірках.

Усі ці методи та алгоритми оптимізації допомагають вирішувати задачу досягнення високої точності моделей, зокрема в задачах комп'ютерного зору або обробки природної мови. Правильний вибір методів оптимізації та регуляризації залежить від характеристик конкретної задачі, обсягу даних та вимог до продуктивності моделі (табл. 2.1).

Таблиця 2.1 – Порівняння оптимізаторів

Оптимізатор	Перевага	Недолік
SGD	Простий, ефективний для великих наборів даних	Може застрягти в локальних мінімумах, чутливий до швидкості навчання
Моментум	Прискорює збіжність, зменшує осциляції	Ускладнює алгоритм
Adagrad	Адаптивна швидкість для кожного параметра, ефективно працює з рідкими даними	Може занадто швидко зменшити швидкість навчання
Adadelta	Динамічна адаптація швидкості навчання, немає потреби в гіперпараметрі швидкості	Може бути занадто агресивним, що призводить до повільної збіжності
RMSProp	Адаптивна швидкість, працює добре при нестабільних цілях	Повільна збіжність у деяких випадках
Adam	Швидка збіжність, гарна узагальнююча здатність, не вимагає багато налаштувань	Потрібно ретельно налаштовувати гіперпараметри

## 2.6 Застосування згорткових нейронних мереж для завдань аналізу емоційного стану

AlexNet – це глибока згорткова нейронна мережа, яка стала основним етапом у розвитку глибокого навчання для задач комп'ютерного зору. Архітектура складається з 5 згорткових шарів, 3 повнозв'язних шарів і одного вихідного шару SoftMax [21]. Ключовими інноваціями, які дозволили AlexNet досягти таких високих результатів, є використання функції активації ReLU, замість традиційної сигмоїдної, що значно пришвидшує навчання, і використання техніки Dropout для боротьби з переобученням. Мережа працює з RGB-зображеннями розміром 227x227 пікселів і навчається на наборі даних ImageNet, містить 1 мільйон зображень в 1000 класах. AlexNet також вперше застосувала GPU для паралельного навчання, що значно прискорило обробку

даних, і відкрила нову еру для використання глибинних нейронних мереж у великих наборах даних.

VGG (Visual Geometry Group) – це архітектура, що стала дуже популярною завдяки своїй простоті і ефективності. VGG використовує багат шарові згорткові шари з малими фільтрами  $3 \times 3$ , що дозволяє забезпечити високий рівень деталізації ознак, а також зменшує кількість параметрів, що дозволяє більш ефективно навчати модель [22]. Вона стала відомою завдяки високій точності в задачах класифікації, а також її можливості бути використаною як попередньо натренована модель для інших завдань комп'ютерного зору, таких як детекція об'єктів чи сегментація. Мережа має різні варіанти, такі як VGG-16 і VGG-19, що мають 16 і 19 шарів відповідно, та показує високу ефективність навіть при використанні великих наборів даних. Однак, через велику кількість параметрів, VGG моделі потребують значних обчислювальних потужностей, що ускладнює їх застосування на мобільних пристроях або в реальному часі.

ResNet (Residual Network) – це архітектура, яка покладається на залишкові блоки (residual blocks), щоб дозволити тренувати дуже глибокі нейронні мережі без проблем зі згасаючим градієнтом. Замість того, щоб спробувати навчити модель на кожному етапі, ResNet додає пропускні шляхи для прямого перенаправлення сигналів і градієнтів між шарами, що дозволяє розв'язати проблему деградації якості моделей з великою кількістю параметрів. Залишкові блоки дозволяють мережі значно збільшувати глибину, зберігаючи високу ефективність та точність на великих наборах даних, таких як ImageNet. Архітектура ResNet має кілька варіантів, таких як ResNet-50, ResNet-101 і ResNet-152, кожен з яких містить відповідну кількість шарів, що дозволяє вибрати між продуктивністю та обчислювальними витратами залежно від задачі. Важливою перевагою ResNet є те, що вона дозволяє створювати неймовірно глибокі мережі без втрат у точності, що є особливо корисним при роботі з великими і складними наборами даних [23].

GoogLeNet (або Inception) – це архітектура, яка була представлена Google у 2014 році і впроваджує нову концепцію «Inception», що дозволяє мережі використовувати різні розміри фільтрів в одному шарі. Це дозволяє мережі адаптуватися до різноманітних ознак зображення, не збільшуючи надмірно кількість параметрів. GoogLeNet використовує складні багатоканальні блоки, які поєднують фільтри  $1 \times 1$ ,  $3 \times 3$  і  $5 \times 5$  для отримання більш точних результатів. Відмінністю цієї архітектури є застосування глобального середнього пулінгу, що заміняє традиційний повнозв'язний шар, що допомагає значно знизити кількість параметрів і зменшити ймовірність переобучення. Це зробило GoogLeNet однією з найбільш обчислювально ефективних моделей для обробки великих наборів зображень, що дозволяє досягати високої точності при відносно низькому споживанні ресурсів. Вона продовжує бути основою для багатьох сучасних моделей, таких як EfficientNet та Inception-V3, завдяки її здатності збалансувати складність і ефективність [24].

Таким чином, кожна з цих архітектур має свої переваги та обмеження. AlexNet змінила напрямок розвитку глибокого навчання, VGG пропонує простоту та ефективність для багатьох застосувань, ResNet дозволяє працювати з дуже глибокими мережами, а GoogLeNet зосереджується на оптимізації обчислювальних витрат, надаючи можливість використовувати глибокі мережі для більш складних задач. Всі ці моделі відзначаються значними досягненнями в області комп'ютерного зору і продовжують впливати на розробку нових архітектур.

## **2.7 Використання датасетів для аналізу емоційного стану**

Датасети відіграють ключову роль у створенні систем розпізнавання емоційних станів та стресу. Вони дозволяють ефективно навчати нейронні мережі, використовуючи різноманітні приклади зображень або відео обличчя з відповідними маркованими емоційними станами. У цьому підрозділі

розглянуто п'ять найпопулярніших датасетів, які використовуються для аналізу емоцій та стресу.

CASME II (Chinese Academy of Sciences Micro-Expression II) – це спеціалізований датасет, створений для дослідження мікро-виразів обличчя, які виникають протягом дуже короткого проміжку часу (звичайно менше 0,5 секунди). Він є одним із найвідоміших наборів даних для аналізу мікроемоційних станів. Зібраний у контрольованих лабораторних умовах із залученням добровольців, цей датасет містить 255 відео, які класифікуються на 8 емоційних категорій: радість, смуток, подив, відраза, страх, гнів, презирство та нейтральний стан [25].

SMIC (Spontaneous Micro-Expression Corpus) створений для аналізу спонтанних мікро-виразів обличчя. На відміну від CASME II, цей датасет зосереджений на природних, а не індукованих мікро-виразах, що надає даним вищу практичну цінність. Він містить відеозаписи у трьох різних спектрах: RGB, NIR (близькоінфрачервоний) та DE (динамічний елемент), що дозволяє більш глибоко аналізувати емоції навіть за складних умов освітлення.

Датасет охоплює 164 записи від 16 учасників і розподіляється на 6 категорій емоцій: радість, смуток, подив, відраза, страх і нейтральний стан. Його записи мають частоту 25–30 fps та варіюються за тривалістю (до 2,5 годин). Хоча датасет не є публічно доступним, його активно використовують у дослідницьких проєктах, що спрямовані на розпізнавання спонтанних виразів обличчя.

KDEF (Karolinska Directed Emotional Faces) – це широко використовуваний набір даних для класифікації статичних виразів обличчя. У ньому представлені зображення 70 добровольців (35 чоловіків і 35 жінок), які демонструють 6 базових емоцій (радість, смуток, гнів, страх, відраза, подив) разом із нейтральним виразом [26].

Цей датасет створений у контрольованих умовах, щоб забезпечити стандартизацію зображень (490×640 px). Він особливо корисний для моделей, які не потребують аналізу динаміки мимічних змін, а зосереджуються лише на

окремих емоціях. KDEF часто використовується як базовий набір даних у дослідженнях, пов'язаних із розпізнаванням емоцій і валідацією результатів моделей.

FER2013 (Facial Expression Recognition 2013) – великий датасет, що містить понад 35 тисяч зображень обличчя в низькій роздільній здатності (48×48 px). Він був зібраний з відкритих джерел, таких як Інтернет, і включає як стандартні, так і нетипові вирази обличчя, що робить його одним із найрізноманітніших наборів даних для навчання нейронних мереж.

Даний датасет охоплює 6 емоцій: радість, гнів, смуток, страх, подив, відраза та нейтральний вираз. FER2013 часто використовується для тренування моделей, які застосовуються у мобільних та реальних додатках завдяки його компактності й масштабності. Його недоліком є низька роздільна здатність, що може впливати на точність моделей [27].

AffectNet – це найбільший на сьогодні датасет, створений для аналізу емоцій обличчя в реальних умовах. Він містить понад 1 мільйон зображень, зібраних із відкритих джерел, і маркованих за 8 класами емоцій, включаючи радість, смуток, страх, гнів, подив, відраза, презирство та нейтральний вираз. Крім цього, AffectNet також має маркування за параметрами валентності (емоційної полярності) та рівня збудження [28].

Цей датасет надає зображення у різній роздільній здатності (224×224 px – 512×512 px) і враховує широкий спектр демографічних особливостей, таких як вік, стать і етнічна приналежність. AffectNet є ідеальним інструментом для створення універсальних моделей, що застосовуються у реальних умовах.

## **2.8 Фреймворки та бібліотеки глибинного навчання**

Глибинне навчання є основою сучасних систем машинного навчання, і для його реалізації використовуються різні фреймворки та бібліотеки, які спрощують процес розробки, навчання та тестування моделей.

Бібліотека NumPy (Numerical Python) – це фундаментальна бібліотека для наукових обчислень у Python. Вона забезпечує підтримку багатовимірних масивів та матриць, а також має велику кількість функцій для виконання математичних операцій. Основною перевагою NumPy є її здатність до ефективного оброблення великих обсягів числових даних, що є критично важливим для глибинного навчання [29].

NumPy надає зручний інтерфейс для роботи з багатовимірними масивами (ndarray), підтримуючи широкий спектр математичних операцій, таких як лінійна алгебра, статистичний аналіз і перетворення Фур'є. Використання NumPy значно спрощує процес попередньої обробки даних та їх підготовки для моделей глибинного навчання.

Основні переваги та функції:

- підтримка багатовимірних масивів та матриць;
- великий набір математичних функцій;
- висока продуктивність завдяки використанню оптимізованого c та fortran коду;
- інтеграція з іншими бібліотеками для наукових обчислень та машинного навчання.

Бібліотека Pandas.

Pandas – це бібліотека для аналізу та маніпуляцій з даними, яка надає високорівневі структури даних та функції, призначені для полегшення роботи з даними, що мають табличний формат. Pandas дозволяє швидко виконувати операції з даними, такі як фільтрація, агрегування, об'єднання та інші, що робить її незамінною для підготовки даних для моделей глибинного навчання.

Основні переваги та функції:

- простий та інтуїтивний інтерфейс для роботи з табличними даними;
- підтримка складних операцій з обробки даних, таких як фільтрація, агрегування та об'єднання;
- інтеграція з іншими бібліотеками для наукових обчислень;

– підтримка імпорту та експорту даних з різних форматів.

Бібліотека Seaborn.

Seaborn – це бібліотека для візуалізації даних на основі бібліотеки matplotlib, яка забезпечує високорівневий інтерфейс для малювання привабливих і інформативних статистичних графіків. Seaborn є корисним інструментом для аналізу та візуалізації результатів роботи моделей глибинного навчання, оскільки дозволяє легко створювати складні графіки та діаграми [30].

Основні переваги Seaborn включають інтеграцію з Pandas, що дозволяє використовувати DataFrame як джерело даних для побудови графіків, а також наявність великої кількості вбудованих тем і стилів для створення привабливих візуалізацій. Seaborn також підтримує створення комплексних графіків, таких як лінійні графіки, гистограми, розсіювання, коробкові графіки та теплові карти, що робить його потужним інструментом для візуального аналізу даних.

Основні переваги та функції:

- високий рівень інтеграції з бібліотекою pandas;
- підтримка широкого спектру статистичних графіків;
- можливість налаштування вигляду графіків за допомогою тем і стилів;
- простий інтерфейс для швидкого створення візуалізацій.

Фреймворк PyTorch.

PyTorch – це один з найбільш популярних фреймворків для глибинного навчання, розроблений Facebook's AI Research lab (FAIR). PyTorch надає динамічний граф обчислень, що робить його гнучким та зручним для досліджень і розробки. Він підтримує автоматичне диференціювання, що спрощує процес навчання моделей. PyTorch також має багату екосистему інструментів і бібліотек, що дозволяє швидко створювати та розгортати моделі глибинного навчання [31].

Основні переваги та функції:

- динамічний граф обчислень для гнучкості в дослідженнях;
- підтримка автоматичного диференціювання;
- вбудовані інструменти для обробки даних;
- велика екосистема інструментів і бібліотек.

Бібліотека OpenCV.

OpenCV (Open Source Computer Vision Library) – це відкрита бібліотека для комп'ютерного зору та обробки зображень. Вона містить більше 2500 алгоритмів для різних завдань, включаючи розпізнавання облич, обробку відео, виявлення об'єктів та інше. OpenCV є популярним вибором для розробки додатків, що потребують обробки зображень у реальному часі, і може бути інтегрована з іншими бібліотеками глибинного навчання для покращення точності моделей [32].

Основні переваги та функції:

- велика кількість алгоритмів для обробки зображень та відео;
- підтримка різних мов програмування;
- інструменти для фільтрації, трансформації та виділення контурів;
- підтримка апаратного прискорення для підвищення продуктивності.

Фреймворк Keras.

Keras – це високорівневий API для глибинного навчання, який працює поверх TensorFlow. Вона була розроблена для швидкого створення та експериментування з моделями глибинного навчання. Keras підтримує як послідовні моделі, так і моделі з функціональним API, що дозволяє будувати складні архітектури нейронних мереж. Завдяки своїй простоті та інтуїтивному інтерфейсу, Keras є відмінним вибором для початківців та досвідчених дослідників у галузі глибинного навчання [33].

Основні переваги та функції:

- інтуїтивний і простий у використанні інтерфейс;
- підтримка послідовних моделей та моделей з функціональним API;

– велика кількість готових шарів та моделей для швидкого прототипування.

## **Висновки до розділу 2**

У розділі розглянуто теоретичні основи глибокого навчання та нейронних мереж, зокрема CNN для аналізу обличчя. Детально описано архітектуру CNN, включаючи згорткові шари, шари активації, пулінгу та повнозв'язні шари. Розглянуто процес навчання CNN, методи оптимізації, регуляризації та боротьби з перенавчанням. Проаналізовано різні архітектури CNN, такі як AlexNet, VGG, ResNet та GoogLeNet, виділено їхні особливості, переваги та недоліки. Окремо розглянуто використання датасетів для аналізу емоцій та стресу, а також фреймворки та бібліотеки для глибинного навчання (NumPy, Pandas, OpenCV, TensorFlow, Keras, PyTorch).

## **3 РОЗРОБКА СИСТЕМИ НА БАЗІ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ОЦІНКИ РІВНЯ СТРЕСУ ЗА ВИРАЗОМ ОБЛИЧЧЯ**

### **3.1 Визначення архітектури інформаційної системи**

Архітектура розроблюваної інформаційної системи для оцінки рівня стресу за виразом обличчя побудована на основі двохрівневої моделі. Така архітектура забезпечує розподіл логіки між клієнтською та серверною частинами, що сприяє гнучкості, масштабованості та легкості в обслуговуванні системи.

Основними складовими системи є:

- клієнтська частина: реалізована на базі веб-технологій (html, css, javascript), що забезпечує доступність системи через веб-браузер. інтерфейс дозволяє користувачу завантажувати зображення або відеопотоки для аналізу, відображає результати роботи системи та пропонує зручні елементи керування;
- серверна частина реалізована за допомогою фреймворку flask (python). вона приймає запити від клієнта, передає їх у модуль CNN для аналізу та повертає результати клієнту. сервер забезпечує взаємодію з нейронною мережею та виконує всі обчислювальні операції, необхідні для аналізу;
- модуль CNN інтегрований у серверну частину. цей модуль, побудований на основі бібліотек tensorflow та keras, виконує попередньо навчений аналіз обличчя на зображеннях або у відеопотоці, визначаючи рівень стресу користувача.

Цей тип архітектури дозволяє розподілити обов'язки між клієнтом і сервером, що сприяє підвищенню ефективності та спрощенню розвитку системи. Сервер може бути централізованим, що спрощує адміністрування та оновлення програмного забезпечення, оскільки зміни на сервері не вимагають змін на стороні клієнта.

Переваги клієнт-серверної архітектури включають можливість централізованого управління ресурсами, що знижує витрати на підтримку та дозволяє ефективно масштабувати систему. Крім того, клієнтська частина може бути легко оновлена без необхідності змінювати серверну частину, що значно спрощує підтримку та полегшує інтеграцію нових функцій. Архітектура також дає змогу централізовано зберігати дані, що дозволяє легко управляти доступом та безпекою.

Проте є і недоліки. Зокрема, при використанні клієнт-серверної архітектури сервер може стати вузьким місцем у системі, оскільки він виконує більшість обчислювальних операцій і обробляє запити. У разі високого навантаження на сервер, система може працювати повільніше, особливо при значній кількості одночасних користувачів. Ще одним недоліком є залежність від стабільності мережі, оскільки без надійного з'єднання з сервером клієнт не зможе отримати необхідну інформацію чи виконати обробку даних.

### **3.2 Вибір інструментів розробки та налаштування робочого середовища**

Для реалізації системи розпізнавання емоцій та визначення рівня стресу налаштоване робоче середовище, яке включало як апаратне, так і програмне забезпечення.

Як основну мову програмування для створення інформаційної системи обрано Python. Python відомий своєю простотою і зручністю використання, а також має багатий набір бібліотек і фреймворків для роботи з даними, створення моделей нейронних мереж і обробки зображень. Завдяки динамічній типізації, автоматичному керуванню пам'яттю та великій кількості готових інструментів Python дозволяє зосередитися на розв'язанні основної задачі, мінімізуючи витрати часу на низькорівневі аспекти програмування.

Для написання, відлагодження та виконання коду використовувалося інтегроване середовище розробки Visual Studio Code, яке забезпечує зручність роботи через підсвічування синтаксису, плагіни для Python і підтримку Git.

Розробка веб-інтерфейсу системи здійснювалася за допомогою HTML, CSS та JavaScript. HTML відповідає за структурування веб-сторінок, CSS надає можливість створювати стильний і зручний дизайн, а JavaScript забезпечує інтерактивність, наприклад, динамічне оновлення результатів аналізу.

Для зв'язку між клієнтською та серверною частиною використовувався мікрофреймворк Flask. Flask – це легкий веб-фреймворк, який забезпечує швидке створення RESTful API і просту інтеграцію з моделями машинного навчання. З його допомогою сервер обробляє запити, виконує обчислення і повертає результати у вигляді відповіді клієнту.

Для реалізації аналізу відеопотоку та обробки обличчя в реальному часі інтегровано кілька бібліотек, зокрема OpenCV та dlib. OpenCV є однією з найпопулярніших бібліотек для обробки зображень і комп'ютерного зору, що забезпечує можливості для виявлення обличчя, відстеження рухів та попередньої обробки зображень перед передачею їх у модель машинного навчання. Однак для точнішого визначення ключових точок на обличчі, таких як очі, брови та губи, використовувалась бібліотека dlib, яка містить потужний детектор облич і предиктор для точного виявлення 68 ключових точок. Ці інструменти дозволяють системі ефективно аналізувати відеопотік в реальному часі, виділяючи важливі риси обличчя для подальшої оцінки емоційного стану користувача.

Для навчання та реалізації нейронних мереж були використані Keras і TensorFlow. TensorFlow – це потужна платформа для розробки моделей машинного навчання, що дозволяє використовувати як CPU, так і GPU для обчислень. Keras, що працює як високорівневий API над TensorFlow, значно спрощує процес створення моделей завдяки своїй простоті та модульності. З

їхньою допомогою створено і навчено нейронні мережі для аналізу емоцій на основі даних із зображень обличчя.

Математичні обчислення та робота з багатовимірними масивами даних здійснювалися за допомогою бібліотеки NumPy. Ця бібліотека є основою для численних наукових пакетів у Python і надає засоби для швидкої роботи з великими масивами даних. У свою чергу, бібліотека Pandas використовувалася для аналізу та маніпулювання табличними даними, що могло включати попередню обробку інформації перед подачею її на вхід моделей машинного навчання.

Для експериментального тестування моделей також розглядалася бібліотека PyTorch, яка забезпечує більшу гнучкість у створенні обчислювальних графів. Хоча основна частина проєкту реалізовувалася з використанням TensorFlow, PyTorch може бути корисним у випадках, коли необхідна висока кастомізація моделі або інтеграція з іншими інструментами. Таким чином, налаштування середовища включало в себе широкий спектр інструментів і технологій, кожен з яких відігравав важливу роль у побудові ефективної системи для аналізу емоцій та стресу.

### **3.3 Проектування архітектури нейронної мережі для аналізу стресу**

Для задачі аналізу стресу за виразом обличчя була побудована згорткова нейронна мережа, яка виконує класифікацію емоційних станів на три категорії: легкий, помірний та сильний стрес. Мережа отримує на вхід зображення обличчя у відтінках сірого розміром 48×48 пікселів. Архітектура мережі складається з кількох згорткових шарів, шарів пулінгу, регуляризації та повнозв'язних шарів для класифікації.

Мережа включає наступні типи шарів:

- згортковий шар (conv2d) – для виділення ключових ознак з зображення.

- пулінговий шар (maxpooling2d) – для зменшення просторових розмірів зображення.
- нормалізація (batchnormalization) – для стабілізації навчання.
- активаційна функція (activation) – для введення нелінійності в модель.
- шар дропаут (dropout) – для зменшення перенавчання.
- шар вирівнювання (flatten) – для перетворення тривимірних даних у двовимірні.
- щільний шар (dense) – для класифікації на основі виділених ознак.

Нейронна мережа складається з восьми шарів Conv2D, чотирьох шарів MaxPooling2D після кожної другої згортки та чотирьох повнозв'язних шарів. У таблиці 3.1 представлено шари розробленої згорткової мережі.

Таблиця 3.1 – Архітектура згортковій нейронній мережі

Layer	Type	Output Size	Filter Size / Stride
Input	Image	48 x 48 x 1	-
CONV-1	Conv2D	48 x 48 x 64	5 x 5, K = 64
CONV-2	Conv2D	48 x 48 x 64	5 x 5, K = 64
POOL-1	MaxPooling2D	24 x 24 x 64	2 x 2
Dropout-1	Dropout	24 x 24 x 64	-
CONV-3	Conv2D	24 x 24 x 128	3 x 3, K = 128
CONV-4	Conv2D	24 x 24 x 128	3 x 3, K = 128
POOL-2	MaxPooling2D	12 x 12 x 128	2 x 2
Dropout-2	Dropout	12 x 12 x 128	-
CONV-5	Conv2D	12 x 12 x 256	3 x 3, K = 256
CONV-6	Conv2D	12 x 12 x 256	3 x 3, K = 256
POOL-3	MaxPooling2D	6 x 6 x 256	2 x 2
Dropout-3	Dropout	6 x 6 x 256	-
CONV-7	Conv2D	6 x 6 x 256	3 x 3, K = 256

Продовження таблиці 3.1 – Архітектура згорткової нейронної мережі

CONV-8	Conv2D	6 x 6 x 256	3 x 3, K = 256
POOL-4	MaxPooling2D	3 x 3 x 256	2 x 2
Dropout-4	Dropout	3 x 3 x 256	-
Flatten	Flatten	256	-
FC-1	Dense	256	-
Dropout-5	Dropout	256	-
FC-2	Dense	64	-
FC-3	Dense	64	-
FC-4	Dense	3	-
Softmax	Activation	3	-

Модель починається з вхідного шару, який приймає зображення розміром 48 x 48 пікселів в градаціях сірого кольору, що є стандартом для багатьох задач, пов'язаних з аналізом виразів обличчя.

Перші два згорткові шари (Conv2D) мають фільтри розміром 5 x 5 і кількість фільтрів 64. Ці шари застосовують операцію згортки до вхідного зображення, в результаті чого виділяються базові ознаки, такі як контури та текстури. Для кожного згорткового шару після нього застосовується пакетна нормалізація (BatchNormalization), яка стабілізує процес навчання, зменшуючи зміни активацій при тренуванні. Після цього використовується функція активації ReLU (Rectified Linear Unit), яка є однією з найбільш поширених функцій активації в нейронних мережах, особливо для глибоких згорткових нейронних мереж. Вона була введена для подолання проблеми зникнення градієнтів, яка часто виникає при використанні старіших функцій активації, таких як сигмоїда чи тангенс гіперболи (tanh).

Після двох згорткових шарів і відповідних нормалізацій та активацій, застосовується максимальний пулінг (MaxPooling2D) з розміром вікна 2 x 2. Це зменшує просторові розміри (ширину та висоту) зображення, зберігаючи

важливі ознаки, і знижує обчислювальну складність. Після пулінгу використовується дропаут з ймовірністю 0.4 для регуляризації моделі та запобігання перенавчанню. Дропаут випадковим чином «вимикає» певний відсоток нейронів під час навчання, що допомагає покращити узагальнення моделі.

Наступні два блоки складаються з аналогічних згорткових шарів з фільтрами 128 та 256. Кожен з цих шарів також супроводжується пакетною нормалізацією та функцією активації ReLU. Після кожного блоку згортки та активації застосовується максимальний пулінг і дропаут. Збільшення кількості фільтрів дозволяє мережі знаходити більш складні патерни в зображеннях, що важливо для аналізу складних виразів обличчя.

Після згорткових блоків і пулінгу, дані подаються через Flatten шар, який перетворює багатовимірний тензор в одновимірний вектор, щоб передати його на наступні повнозв'язні шари (Dense). Перший повнозв'язний шар має 256 нейронів і використовує активацію ReLU, що дозволяє моделі враховувати всі взаємозв'язки між ознаками, що були виділені на попередніх етапах. Після цього шар знову використовує BatchNormalization для стабільності та Dropout для запобігання перенавчанню.

Останній шар у моделі – вихідний повнозв'язний шар з кількістю нейронів, що відповідає кількості класів класифікації. Цей шар використовує функцію активації Softmax, яка перетворює вихідні значення в ймовірності, що в сумі дорівнюють 1. Це дозволяє точно класифікувати зображення в одну з трьох категорій.

У результаті, ця архітектура нейронної мережі добре підходить для класифікації емоцій на основі виразів обличчя, завдяки глибокому навчанню на багатьох рівнях абстракції, ефективній регуляризації та стабільному навчанні.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, BatchNormalization,
Activation, MaxPooling2D, Flatten, Dropout, Dense,
GlobalAveragePooling2D
```

```

def custom_CNN(input_shape, num_classes):
model = Sequential(name='simple_CNN')
# Block-1
model.add(Conv2D(filters=64, kernel_size=(5, 5), padding='same',
kernel_initializer='he_normal', input_shape=input_shape,
name='conv2d_1'))
model.add(BatchNormalization(name='batchnorm_1'))
model.add(Conv2D(filters=64, kernel_size=(5, 5), padding='same',
kernel_initializer='he_normal', name='conv2d_2'))
model.add(BatchNormalization(name='batchnorm_2'))
model.add(Activation('relu', name='relu_1'))
model.add(MaxPooling2D(pool_size=(2, 2), name='maxpool2d_1'))
model.add(Dropout(0.4, name='dropout_1'))
# Block-2
model.add(Conv2D(filters=128, kernel_size=(3, 3), padding='same',
kernel_initializer='he_normal', name='conv2d_3'))
model.add(BatchNormalization(name='batchnorm_3'))
model.add(Conv2D(filters=128, kernel_size=(3, 3), padding='same',
kernel_initializer='he_normal', name='conv2d_4'))
model.add(BatchNormalization(name='batchnorm_4'))
model.add(Activation('relu', name='relu_2'))
model.add(MaxPooling2D(pool_size=(2, 2), name='maxpool2d_2'))
model.add(Dropout(0.4, name='dropout_2'))
# Block-3
model.add(Conv2D(filters=256, kernel_size=(3, 3), padding='same',
kernel_initializer='he_normal', name='conv2d_5'))
model.add(BatchNormalization(name='batchnorm_5'))
model.add(Conv2D(filters=256, kernel_size=(3, 3), padding='same',
kernel_initializer='he_normal', name='conv2d_6'))
model.add(BatchNormalization(name='batchnorm_6'))
model.add(Activation('relu', name='relu_3'))
model.add(MaxPooling2D(pool_size=(2, 2), name='maxpool2d_3'))
model.add(Dropout(0.5, name='dropout_3'))
# Block-4
model.add(Conv2D(filters=256, kernel_size=(3, 3), padding='same',
kernel_initializer='he_normal', name='conv2d_7'))
model.add(BatchNormalization(name='batchnorm_7'))
model.add(Conv2D(filters=256, kernel_size=(3, 3), padding='same',
kernel_initializer='he_normal', name='conv2d_8'))
model.add(BatchNormalization(name='batchnorm_8'))
model.add(Activation('relu', name='relu_4'))
model.add(MaxPooling2D(pool_size=(2, 2), name='maxpool2d_4'))
model.add(Dropout(0.5, name='dropout_4'))
# Classifier
model.add(Flatten(name='flatten'))
model.add(Dense(256, activation='relu',
kernel_initializer='he_normal', name='dense_1'))
model.add(BatchNormalization(name='batchnorm_9'))
model.add(Dropout(0.5, name='dropout_5'))
model.add(Dense(num_classes, activation='softmax',
kernel_initializer='he_normal', name='output'))
return model

```

Лістинг 3.1 – Модель розробленої CNN

## 3.4 Обробка та розподіл даних для тренування моделі

### 3.4.1 Опис датасету

Для тренування моделі використано датасет FER2013, який містить понад 35 тисяч зображень обличч розміром 48×48 пікселів у відтінках сірого. Кожне зображення належить до однієї з семи категорій емоцій: злість (Angry), відраза (Disgust), страх (Fear), радість (Happy), смуток (Sadness), здивування (Surprise) або нейтральний вираз (Neutral). Завдяки простоті структури цей датасет ідеально підходить для задач розпізнавання емоцій.

Зображення зберігаються у вигляді послідовності пікселів у CSV-файлі. Першим етапом є перетворення цих даних у формат матриці, придатної для обробки нейронною мережею. Після цього всі значення пікселів нормалізуються до діапазону [0, 1], що сприяє стабільному навчанню моделі.

```
import pandas as pd
import cv2
import numpy as np

dataset_path = 'fer2013/fer2013/fer2013.csv'
image_size = (48, 48)

def load_fer2013():
    data = pd.read_csv(dataset_path)
    pixels = data['pixels'].tolist()
    faces = []
    for pixel_sequence in pixels:
        face = [int(pixel) for pixel in pixel_sequence.split(' ')]
        face = np.asarray(face).reshape(image_size)
        face = cv2.resize(face.astype('uint8'), image_size)
        faces.append(face.astype('float32'))
    faces = np.asarray(faces)
    faces = np.expand_dims(faces, -1)
    emotions = pd.get_dummies(data['emotion']).values
    return faces, emotions
```

Лістинг 3.2 – Обробка та розподіл даних для тренування моделі

### 3.4.2 Класифікація ознак емоцій за стресовим станом

Датасет спершу обробляється з метою перекласифікації ознак емоцій у дві групи: немає стресу та є стрес. Емоції, що не асоціюються з дискомфортом, такі як радість, здивування або нейтральний вираз, відносяться до першої групи. Натомість емоції, що можуть сигналізувати про негативний стан, такі як страх, смуток, злість і відраза, класифікуються як стресові. Цей поділ дозволяє виділити групу облич, які потребують детальнішого аналізу для визначення рівня стресу.

В лістингу 3.3 для цієї задачі використовується функція, яка перетворює вихідні мітки емоцій у відповідні значення рівня стресу (0 – немає стресу, 1 – є стрес).

```
def map_emotion_to_stress(emotion):
    if emotion in [3, 5, 6]: # Happy, Surprise, Neutral
        return 0 # Немає стресу
    else: # Fear, Sad, Angry, Disgust
        return 1 # Є стрес

data['stress_level'] = data['emotion'].apply(map_emotion_to_stress)
```

#### Лістинг 3.3 – Класифікація ознак емоцій за стресовим станом

Після ідентифікації емоцій на обличчі модель визначає, чи належить ця емоція до категорії стресових. Якщо стресовий стан відсутній, аналіз завершується, а для стресових емоцій застосовується подальший аналіз ключових точок обличчя (лістинг 3.4).

```
def emotion_finder_and_stress_level(frame, faces):
    # Класифікація емоцій на "є стрес" або "немає стресу"
    EMOTIONS = ["angry", "disgust", "scared", "happy", "sad",
                "surprised", "neutral"]
    x, y, w, h = face_utils.rect_to_bb(faces)
    face_frame = frame[y:y+h, x:x+w]
    roi = cv2.resize(face_frame, (64, 64))
    roi = roi.astype("float") / 255.0
    roi = img_to_array(roi)
    roi = np.expand_dims(roi, axis=0)
```

```

preds = emotion_classifier.predict(roi)[0]
emotion_probability = np.max(preds)
label = EMOTIONS[preds.argmax()]

# Перекласифікація емоції у рівень стресу
stress_level = map_emotion_to_stress(preds.argmax())

# Якщо є стрес, то визначаємо рівень стресу за точками
if stress_level == 1:
    # Оцінимо відстані між важливими точками на обличчі
    (наприклад, очі, рот, брови)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    detections = detector(gray, 0)

    for detection in detections:
        shape = predictor(frame, detection)
        shape = face_utils.shape_to_np(shape)

        # Визначаємо відстань між лівим і правим оком
        eye_dist = ebdist(shape[42], shape[39]) # Відстань між
лівим і правим оком
        # Визначаємо відстань між верхньою і нижньою частиною рота
        lip_dist = lpdist(shape[51], shape[57])
        # Визначаємо відстань між кінцями бровей
        left_eyebrow_dist, right_eyebrow_dist =
eyebrow_dist(shape[17:22], shape[22:27]) # Брови

        # Класифікація рівня стресу
        stress_label =
classify_stress_level_from_keypoints(eye_dist, lip_dist,
(left_eyebrow_dist, right_eyebrow_dist))
        return f"Stress level: {stress_label}"
    return "No Stress"

```

### Лістинг 3.4 – Аналіз ключових ознак і визначення рівня стресу

#### 3.4.3 Аналіз ключових ознак і визначення рівня стресу

Коли модель класифікує емоцію як стресову, виконується додатковий етап аналізу виразу обличчя. Це дає змогу деталізувати рівень стресу (низький, середній або високий). Для цього використовується набір ключових точок обличчя, які вказують на особливості міміки.

Спершу визначаються координати ключових точок за допомогою бібліотек `dlib` та `OpenCV`. Серед ключових зон особливу увагу приділяють очам, губам і бровам. Ці зони найбільш інформативні для оцінки виразу

обличчя, оскільки зміни їхнього положення або відстаней між ними прямо вказують на напруженість м'язів і емоційний стан (лістинг 3.5) .

```
# Функція для обчислення відстані між двома точками
def ebdist(p1, p2):
    return np.linalg.norm(p1 - p2)

def lpdist(p1, p2):
    return np.linalg.norm(p1 - p2)

def eyebrow_dist(left_eyebrow_points, right_eyebrow_points):
    left = np.mean(left_eyebrow_points, axis=0)
    right = np.mean(right_eyebrow_points, axis=0)
    return np.linalg.norm(left - right)
```

### Лістинг 3.5 – Функції для обчислення відстаней між точками обличчя

Далі обчислюються певні параметри, які дозволяють оцінити стан кожної зони:

- відстань між верхньою і нижньою губою відображає ступінь відкриття рота, що є індикатором напруження або здивування.
- відстань між кінцями брів сигналізує про положення брів, які можуть підніматися або зближуватися залежно від рівня стресу.
- середня відстань між верхньою і нижньою повіками дозволяє оцінити ступінь відкриття очей, що є важливим показником емоційного стану.

Відстані обчислюються як евклідові між відповідними координатами точок, визначених бібліотекою `dlib`. Отримані значення нормалізуються, щоб забезпечити єдиний масштаб для всіх метрик. Як здійснюється нормалізація продемонстровано в лістингу 3.6.

```
ccb
def normalize(value, min_val, max_val):
    return (value - min_val) / (max_val - min_val)
```

### Лістинг 3.6 – Нормалізація відстаней для оцінки рівня стресу

Нормалізовані відстані далі використовуються для обчислення інтегрального показника стресу. Цей показник визначається як середнє значення всіх нормалізованих метрик.

### 3.4.4 Класифікація рівня стресу

Інтегральний показник використовується для остаточного визначення рівня стресу:

- якщо показник менший за 0.5, стрес оцінюється як низький;
- значення від 0.5 до 0.75 вказують на середній рівень стресу;
- показник від 0.75 і вище свідчить про високий рівень стресу.

Функція класифікація рівня стресу за інтегрального показник показана в лістингу 3.7.

```
def classify_stress_level_from_keypoints(eye_dist, lip_dist,
eyebrow_dist):
    score = (eye_dist + lip_dist + eyebrow_dist) / 3
    if score < 0.5:
        return 'Low'
    elif score < 0.75:
        return 'Moderate'
    else:
        return 'High'
```

#### Лістинг 3.7 – Класифікація рівня стресу за інтегрального показником

Таким чином, система здатна не лише виявити факт стресу, але й деталізувати його інтенсивність. Якщо обличчя класифікується як таке, що не має стресу, результати аналізу одразу виводяться як No Stress. Підсумковий підхід дозволяє інтегрувати аналіз емоційного стану і фізіологічних характеристик обличчя, що забезпечує високу точність і адаптивність системи до різних ситуацій.

### 3.4.5 Розподіл та розширення даних

Розподіл даних на навчальний та валідаційний набори. Це дозволяє перевірити, як модель працює на нових, невідомих їй даних. Для цього використовувалися функції з бібліотеки `sklearn.model_selection`, щоб поділити дані на тренувальний (80%) та тестовий (20%) набори (лістинг 3.8).

```
from sklearn.model_selection import train_test_split
X_train, X_valid, y_train, y_valid = train_test_split(faces, emotions,
                                                    test_size=0.1,
                                                    random_state=42)
```

#### Лістинг 3.8 – Розподіл даних на навчальний та валідаційний набори

Щоб модель була стійкішою до змін у вхідних зображеннях, застосовано методи розширення даних. Це включає випадкові трансформації, такі як обертання, зсув і масштабування, що дозволяє збільшити варіативність тренувальної вибірки і знизити ризик перенавчання (лістинг 3.9).

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=30,
    shear_range=0.3,
    zoom_range=0.3,
    width_shift_range=0.4,
    height_shift_range=0.4,
    horizontal_flip=True,
    fill_mode='nearest'
)

validation_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    'train_data',
    color_mode='grayscale',
    target_size=(48, 48),
    batch_size=64,
    class_mode='categorical',
    shuffle=True
)

validation_generator = validation_datagen.flow_from_directory(
    'validation_data',
    color_mode='grayscale',
```

```

target_size=(48, 48),
batch_size=64,
class_mode='categorical',
shuffle=True
)

```

### Лістинг 3.9 – Застосування методів розширення даних для тренування

Для захоплення відео в реальному часі, обробки кадрів та визначення рівня стресу, використовується клас VideoCamera (лістинг 3.10) . Цей клас дозволяє здійснювати захоплення відео з веб-камери, обробляти кожен кадр для визначення ключових точок обличчя, класифікувати емоції та розраховувати рівень стресу, який потім передається назад клієнту для відображення в реальному часі.

```

class VideoCamera(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)

    def __del__(self):
        self.video.release()

    def get_frame(self):
        ret, frame = self.video.read()
        frame = cv2.flip(frame, 1) # Flip frame horizontally
        frame = imutils.resize(frame, width=500, height=500)
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        # Відображення результатів на кадрі
        cv2.putText(frame, f"Emotion: {emotion_label}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
        cv2.putText(frame, f"Stress Level: {stress_label}", (10,
50), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)

        ret, jpeg = cv2.imencode('.jpg', frame)
        return jpeg.tobytes()

```

### Лістинг 3.10 – Клас VideoCamera для захоплення відео та оцінки стресу

## 3.5 Застосування алгоритмів навчання та валідації згорткової нейронної мережі

Навчання та валідація згорткової нейронної мережі базуються на поєднанні процедур і алгоритмів, що забезпечують ефективну оптимізацію

параметрів моделі, перевірку її узагальнюючої здатності та запобігання перенавчанню.

Алгоритм навчання CNN можна узагальнити наступним чином:

- ініціалізація: завантаження даних, визначення архітектури моделі, компіляція з вибором оптимізатора та функції втрат;
- аугментація: розширення набору даних через трансформації;
- навчання: оптимізація параметрів моделі з використанням зворотного поширення помилки;
- валідація: оцінка узагальнюючої здатності моделі на невідомих даних;
- регулювання: використання колбеків для динамічного коригування параметрів навчання.

На першому етапі компіляції моделі задаються параметри для алгоритму оптимізації, функції втрат і метрики оцінки

Оптимізатор Adam комбінує переваги алгоритмів AdaGrad і RMSProp, автоматично коригуючи швидкість навчання для кожного параметра. Його початкова швидкість навчання (`learning_rate`) становить 0.001 (лістинг 3.11).

Функція втрат `categorical_crossentropy` підходить для багатокласових завдань класифікації, включаючи розпізнавання рівнів стресу.

Метрика `accuracy` обирається для оцінки точності класифікації на тренувальному та валідаційному наборах.

```
from keras.optimizers import Adam
model.compile(
    loss='categorical_crossentropy',
    optimizer=Adam(learning_rate=0.001),
    metrics=['accuracy']
)
```

### Лістинг 3.11 – Оптимізатор Adam

Використання колбеків. Колбеки (`callbacks`) – це програмні інструменти, що дозволяють динамічно змінювати поведінку процесу навчання. Вони

використовуються для покращення ефективності та стабільності навчання. Основні колбеки, що застосовуються в цьому проекті наведені в лістингу 3.12:

- `ModelCheckpoint` – зберігає модель з найкращими результатами валідації (`val_loss`) у файл для подальшого використання. Це забезпечує можливість завантаження моделі без повторного навчання;
- `EarlyStopping` – зупиняє навчання, якщо функція втрат на валідаційному наборі (`val_loss`) перестає покращуватися протягом 20 епох. Це дозволяє уникнути перенавчання;
- `ReduceLROnPlateau` – зменшує швидкість навчання у 5 разів, якщо `val_loss` не покращується протягом 10 епох. Це дозволяє моделі краще зосереджуватися на тонких аспектах навчання.

```
from keras.callbacks import ModelCheckpoint, EarlyStopping,
ReduceLROnPlateau

checkpoint = ModelCheckpoint('best_model.h5', monitor='val_loss',
mode='min', save_best_only=True, verbose=1)
earlystop = EarlyStopping(monitor='val_loss', patience=20,
restore_best_weights=True, verbose=1)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2,
patience=10, verbose=1)

callbacks = [checkpoint, earlystop, reduce_lr]
```

### Лістинг 3.12 – Використання колбеків

Навчання моделі. Навчання проводиться за допомогою генераторів даних (`ImageDataGenerator`), які забезпечують аугментацію (розширення) тренувальної вибірки (лістинг 3.13). Це включає:

- ротації: випадкові обертання зображень;
- масштабування: збільшення або зменшення зображень;
- зсуви та горизонтальне відображення: додаткові трансформації для підвищення стійкості моделі до варіацій у даних.

Процес навчання відбувається за допомогою методу `fit`, який враховує:

- кількість епох (`epochs`) – максимальну кількість проходів через весь набір даних;
- розмір пакета (`batch_size`) – кількість прикладів, що обробляються за один раз;
- валідацію на нових, невідомих даних.

```
history = model.fit(
    train_generator,
    steps_per_epoch=len(X_train) // batch_size,
    epochs=100,
    callbacks=callbacks,
    validation_data=validation_generator,
    validation_steps=len(X_valid) // batch_size
)
```

### Лістинг 3.13 – Валідація моделі

Оцінка результатів навчання. Для візуалізації та аналізу процесу навчання будуються графіки точності (`accuracy`) та функції втрат (`loss`) (лістинг 3.14). Це дозволяє оцінити, як модель вчиться на тренувальних і валідаційних наборах, а також ідентифікувати можливе перенавчання.

```
def plot_model_history(history):
    import matplotlib.pyplot as plt

    # Точність
    plt.plot(history.history['accuracy'], label='Точність на тренуванні')
    plt.plot(history.history['val_accuracy'], label='Точність на валідації')
    plt.legend()
    plt.title('Точність моделі')
    plt.show()

    # Функція втрат
    plt.plot(history.history['loss'], label='Втрата на тренуванні')
    plt.plot(history.history['val_loss'], label='Втрата на валідації')
    plt.legend()
    plt.title('Функція втрат')
    plt.show()

plot_model_history(history)
```

### Лістинг 3.14 – Візуалізація точності та втрат моделі

### 3.6 Розробка ІС

ІС розроблена для визначення рівня стресу на основі аналізу емоцій, використовує мікро-фреймворк Flask для Python. ІС забезпечує зручний інтерфейс для користувачів, що дозволяє їм у реальному часі перевіряти рівень стресу за допомогою веб-камери. Основна мета ІС – зібрати відео з веб-камери, передати кадри до сервера, де відбувається їх обробка через спеціалізовані модулі для розпізнавання емоцій та визначення рівня стресу, а потім відправити результати назад до користувача для візуалізації.

Наступним завданням було виконати вибір програмного засобу для реалізації серверної та клієнтської частини ІС. Flask був обраний для реалізації системи через свою простоту і гнучкість, що дозволяє швидко налаштувати сервер і створити необхідні маршрути для взаємодії між клієнтом та сервером. Flask є ідеальним вибором для створення API, які мають забезпечувати обробку відеопотоку в реальному часі, адже він дозволяє швидко налаштувати HTTP-запити та відповіді, а також забезпечити взаємодію з іншими Python-бібліотеками для обробки відео та аналізу даних. Для нашого додатку Flask забезпечує ефективну обробку запитів, що дозволяє працювати з відеопотоком, отриманим від веб-камери користувача, і передавати його для подальшої обробки на сервер.

Визначимося зі структурою інформаційної системи . ІС складається з двох основних компонентів:

- клієнтська частина – веб-інтерфейс, через який користувач може взаємодіяти з додатком. цей інтерфейс дозволяє користувачу включити веб-камеру і переглядати відеопотік в реальному часі;
- серверна частина – частина, що обробляє запити від користувача, отримує відео-кадри з веб-камери, передає їх модулю розпізнавання емоцій та стресу для аналізу, а потім відправляє результати на клієнтську частину для відображення.

Для забезпечення коректної роботи ІС у Flask були реалізовані кілька основних маршрутів, кожен з яких виконує певні функції.

Маршрут для виявлення стресу: в лістингу 3.16 продемонстрований маршрут для сторінки виявлення стресу, що є головною, користувач одразу може побачити відеопотоку з веб-камери та аналіз з відображенням результатів виявлення стресу.

```
@app.route('/stress_detection')
def stress_detection():
    return render_template('stress_detection.html')
```

### Лістинг 3.16 – Маршрут для сторінки виявлення стресу

Маршрут для відеопотоку: Відео з веб-камери передається через спеціальний маршрут Flask, який генерує потік кадрів, кожен з яких обробляється для визначення рівня стресу (лістинг 3.17).

```
@app.route('/predict')
def predict():
    return Response(gen(VideoCamera()), mimetype='multipart/x-mixed-replace; boundary=frame')
```

### Лістинг 3.17 – Маршрут для прогнозування за допомогою відеокамери

Цей маршрут забезпечує передачу відеопотоку від веб-камери до серверної частини додатку, де кадри будуть оброблені для визначення рівня стресу.

Для відео-потоків і його обробці був створений клас VideoCamera. У класі VideoCamera відбувається захоплення кадрів з веб-камери. Потім ці кадри передаються в модуль для визначення емоцій та рівня стресу. Основна мета цього класу – забезпечити захоплення відео і обробку кожного кадру для визначення відстаней між ключовими точками на обличчі, що дозволяє виявити ознаки стресу (лістинг 3.18). Крім того, цей клас виконує функцію передавання оброблених кадрів назад на клієнтську частину.

```

class VideoCamera(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)

    def __del__(self):
        self.video.release()

    def get_frame(self):
        ret, frame = self.video.read()
        frame = cv2.flip(frame, 1)
        frame = imutils.resize(frame, width=500, height=500)
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        detections = detector(gray, 0)
        for detection in detections:
            shape = predictor(frame, detection)
            shape = face_utils.shape_to_np(shape)
            (lBegin, lEnd) =
face_utils.FACIAL_LANDMARKS_IDXS["right_eyebrow"]
            (rBegin, rEnd) =
face_utils.FACIAL_LANDMARKS_IDXS["left_eyebrow"]
            (l_lower, l_upper) =
face_utils.FACIAL_LANDMARKS_IDXS["mouth"]

            leybrow = shape[lBegin:lEnd]
            reybrow = shape[rBegin:rEnd]
            openmouth = shape[l_lower:l_upper]

            reybrowhull = cv2.convexHull(reybrow)
            leybrowhull = cv2.convexHull(leybrow)
            openmouthhull = cv2.convexHull(openmouth)

            lipdist = lpdist(openmouthhull[-1], openmouthhull[0])
            eyedist = ebdist(leybrow[-1], reybrow[0])

            stress_value, stress_label = normalize_values(None,
eyedist, None, lipdist)

            cv2.putText(frame, "stress
value:{}".format(str(int(stress_value * 100))), (10, 40),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (51, 66, 232), 2)
            cv2.putText(frame, "Stress
level:{}".format((stress_label)), (10, 60), cv2.FONT_HERSHEY_SIMPLEX,
0.5, (35, 189, 25), 2)

            ret, jpeg = cv2.imencode('.jpg', frame)
            return jpeg.tobytes()

```

### Лістинг 3.18 – Клас для отримання кадрів відео

Виконаємо опис інтерфейсу ІС. Розроблений веб-застосунок дозволяє користувачеві оцінювати рівень стресу в реальному часі, використовуючи

камеру для аналізу емоційних виразів обличчя. Після надання доступу до веб-камери, користувач може побачити своє зображення на екрані разом з інформацією про рівень стресу, який визначається системою на основі алгоритмів аналізу емоцій (рис. 3.2 – 3.3).

Основні елементи інтерфейсу:

- 1) Вікно відео з камери – на екрані відображається відео користувача, що дає можливість системі аналізувати його вирази обличчя і визначати рівень стресу.
- 2) Кругова діаграма – на екрані відображається рівень стресу у вигляді кольорової кругової діаграми. Діаграма змінюється в залежності від рівня стресу, відображаючи його відсоткове значення.
- 3) Текстова інформація – під діаграмою з'являється точний рівень стресу, виражений числовим значенням, а також короткий опис стану користувача (наприклад, «Немає стресу», «Легкий стрес»).

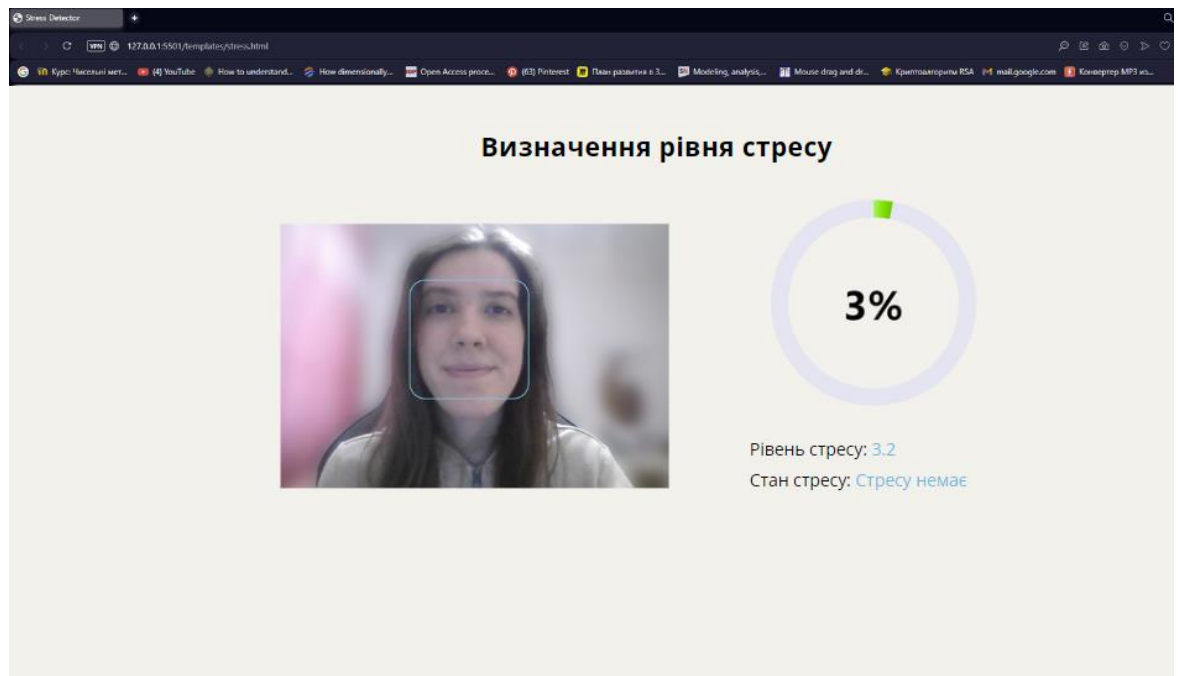


Рисунок 3.1 – Інтерфейс веб-застосунку. Визначення рівня стресу –  
Немає стресу

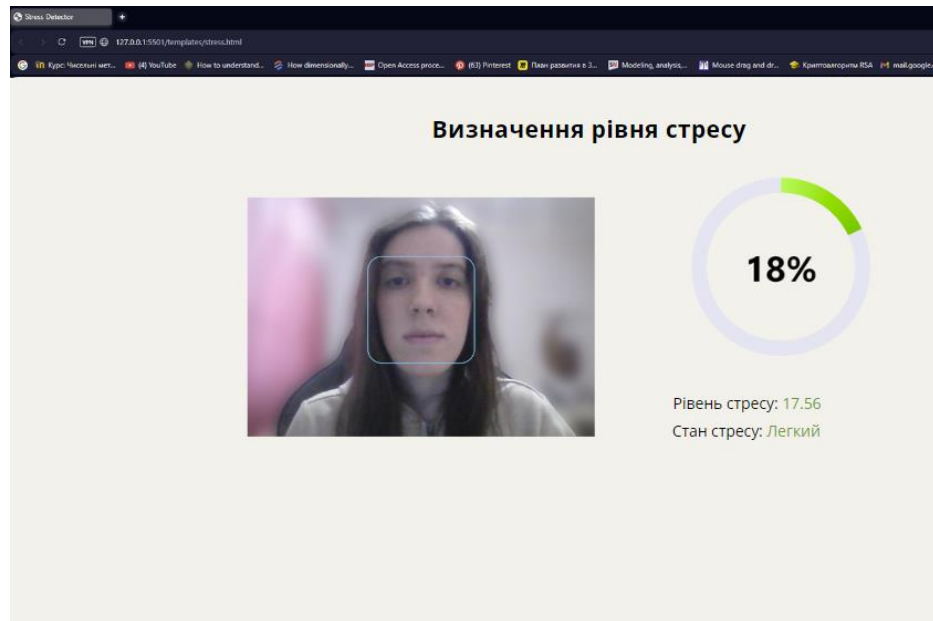


Рисунок 3.2– Інтерфейс ІС. Визначення рівня стресу – Легкий стрес

Застосунок для визначення рівня стресу за виразом обличчя має кілька ключових функцій, які забезпечують виконання основних завдань для оцінки емоційного стану користувача. Ось основні функції, які повинні бути реалізовані в застосунку:

- завантаження зображень та відеопотоків: користувач має можливість завантажувати одиничні зображення обличчя або відео для подальшого аналізу.
- обробка зображень: використовуючи бібліотеки opencv та dlib, система ефективно виявляє обличчя на зображеннях або в потоковому відео, а також визначає ключові точки на обличчі (очі, брови, губи), що є важливим для аналізу емоційного стану.
- аналіз емоційного стану: застосунок використовує згорткову нейронну мережу для класифікації емоцій на позитивні та негативні, а також для оцінки рівня стресу (легкий, помірний, сильний). Для цього за допомогою детектора dlib розраховується відстань між ключовими точками обличчя (бровами, очима та губами), що дозволяє визначити ступінь стресу на основі змін цих характеристик.

- результати в реальному часі: для відеоаналізу застосунок здійснює обробку у реальному часі, відображаючи результати на екрані через веб-інтерфейс.
- виведення результатів аналізу: після обробки кадрів система відображає рівень стресу у вигляді кругової шкали прогресу, яка показує округлене числове значення рівня стресу та його тип (легкий, помірний, сильний або відсутність стресу, коли значення рівне 0). додатково на екрані з'являється графік, що демонструє зміни рівня стресу протягом часу використання системи.
- інтерфейс для користувача: застосунок має веб-інтерфейс, що реалізований за допомогою html, css і javascript. цей інтерфейс дозволяє зручно взаємодіяти з системою.

### **Висновки до розділу 3**

У розділі описано розробку системи для оцінки рівня стресу на основі згорткових нейронних мереж. Визначено архітектуру інформаційної системи, що складається з клієнтської та серверної частин. Детально описано процес обробки та розподілу даних для навчання моделі, включаючи опис датасету FER2013, класифікацію емоцій за стресовим станом, аналіз ключових точок обличчя, розподіл та розширення даних, та реалізацію обробки відео в реальному часі. Описано алгоритми навчання та валідації згорткової нейронної мережі, а також впровадження веб-застосунку з описом його інтерфейсу та технічних вимог. Розроблена система готова до тестування та оцінки її ефективності.

## 4 ОЦІНКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНОЇ СИСТЕМИ ДЛЯ ВИЗНАЧЕННЯ СТРЕСУ

### 4.1 Навчання та верифікація моделі

Навчання нейронної мережі для визначення рівня стресу за виразом обличчя є складним, але важливим етапом розробки моделі. Основним завданням було навчити модель на великому наборі даних та перевірити її здатність робити точні прогнози для невідомих прикладів.

Для навчання та тестування моделі використовувався датасет FER2013, який містить зображення облич людей, що відображають різноманітні емоції, такі як щастя, сум, гнів, страх, здивування, зневага та нейтральні вирази. Всього датасет містить понад 35 000 зображень, що дозволяє створювати потужні моделі для розпізнавання емоцій, а також для визначення рівня стресу на основі виразу обличчя. Для подальшого тренування дані були поділені на два набори: навчальний (80%) та валідаційний (20%). На рисунку 4.1 відображені графіки точності (Accuracy) та втрат (Loss) на навчальному і валідаційному наборах даних.

Процес навчання нейронної мережі полягає в оптимізації ваги мережі через багаторазове проходження навчальних даних. Для досягнення оптимальних результатів було вибрано кілька важливих параметрів:

- епохи (epochs) – це кількість повних проходів через навчальний набір даних. чим більше епох, тим більший шанс, що модель «запам'ятає» особливості даних, але також зростає ризик перенавчання, коли модель занадто сильно адаптується до конкретних прикладів, втрачаючи здатність узагальнювати;
- розмір пакету (batch size) – це кількість зображень, які модель обробляє перед оновленням своїх ваг. малі пакети можуть призвести до повільного навчання, але дають моделі більше можливостей для

адаптації. великий пакет пришвидшує процес, але може призвести до меншої точності;

- ітерації (iterations) – кількість кроків, за які модель оновлює свої ваги під час проходження даних.

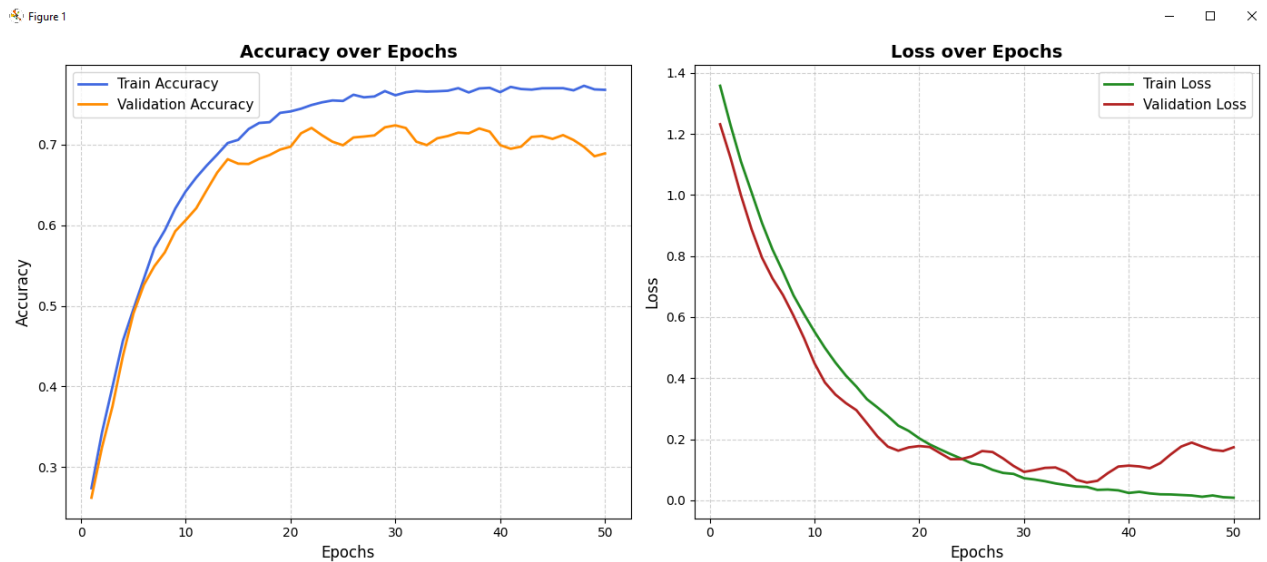


Рисунок 4.1 – Графіки точності (Accuracy) та втрат (Loss) на навчальному і валідаційному наборах даних.

Графік точності:

- тренувальна точність (train accuracy): протягом навчання точність на тренувальній вибірці поступово зростала, досягнувши 77%. на графіку спостерігається плавне збільшення точності завдяки ефективному навчальному процесу;
- валідаційна точність (validation accuracy): валідаційна точність також показує позитивну динаміку, але з деякими коливаннями, що може бути результатом різних умов тестування. вона досягла 72.6% на кінцевій епісі. це свідчить про хорошу узагальнювальну здатність моделі, хоча й спостерігаються незначні коливання, що можуть бути пов'язані з різноманітністю валідаційних даних.

Графік втрат:

- Тренувальні втрати (Train Loss): Втрати на тренувальній вибірці знижувалися, що підтверджує правильне навчання мережі. Однак, на кінцевих етапах з'являються невеликі коливання, що може бути результатом адаптації мережі до складних прикладів.
- Валідаційні втрати (Validation Loss): Валідаційні втрати також знижувалися, однак з більш помітними коливаннями, що може вказувати на незначну нестабільність або перешкоди в розпізнаванні при тестуванні нових даних. Наприкінці навчання спостерігається легке підвищення втрат, що може бути ознакою легкого перенавчання.

Хоча модель досягла гарних результатів на навчальних даних, спостерігаються невеликі коливання валідаційної точності та підвищення валідаційних втрат на останніх етапах. Це може свідчити про початкові ознаки перенавчання. Висока валідаційна точність (72.6%) свідчить про ефективне навчання нейронної мережі і її здатність правильно класифікувати емоційні стани, зокрема рівень стресу.

Навчання було завершено на 50-ій епосі, оскільки на графіках видно, що модель досягла стабільних результатів, і подальші етапи навчання не призвели до значних покращень у точності або зниженні втрат.

## 4.2 Оцінка точності моделі

Оцінка точності моделі здійснювалася за допомогою різних метрик класифікації, серед яких важливе місце займає матриця плутанини, що дозволяє наочно продемонструвати кількість правильних і неправильних прогнозів для кожної категорії емоцій (лістинг 4.1). Цей інструмент дає змогу не тільки оцінити загальну кількість помилок, а й визначити типи помилок, допущених моделлю при класифікації.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
import tensorflow as tf
```

```

model = tf.keras.models.load_model(custom_CNN.h5)

yhat_valid = np.argmax(model.predict(X_valid), axis=1)

print(f"Total wrong validation predictions: {np.sum(np.argmax(y_valid,
axis=1) != yhat_valid)}\n")
print("Classification Report:")
print(classification_report(np.argmax(y_valid, axis=1), yhat_valid))

confusion = confusion_matrix(np.argmax(y_valid, axis=1), yhat_valid)

labels = ["немає стресу", "легкий стрес", "помірний стрес", "сильний
стрес"]

precision = np.diag(confusion) / np.sum(confusion, axis=0)
recall = np.diag(confusion) / np.sum(confusion, axis=1)
f1_score = 2 * (precision * recall) / (precision + recall)

print("\nМетрики класифікації:")
for i, label in enumerate(labels):
    print(f"{label}:")
    print(f"  Precision: {precision[i]:.2f}")
    print(f"  Recall: {recall[i]:.2f}")
    print(f"  F1-Score: {f1_score[i]:.2f}")
print(f"\nЗагальна точність (Accuracy): {np.trace(confusion) /
np.sum(confusion):.2%}")

plt.figure(figsize=(8, 6))

plt.imshow(confusion, interpolation='nearest', cmap=plt.cm.Greens)
plt.title("Матриця плутанини")
plt.colorbar()
tick_marks = np.arange(len(labels))
plt.xticks(tick_marks, labels, rotation=45)
plt.yticks(tick_marks, labels)

thresh = confusion.max() / 2.
for i in range(confusion.shape[0]):
    for j in range(confusion.shape[1]):
        plt.text(j, i, format(confusion[i, j], 'd'),
                ha="center", va="center",
                color="white" if confusion[i, j] > thresh else
"black")

plt.ylabel("Істинна мітка")
plt.xlabel("Передбачена мітка")
plt.tight_layout()
plt.show()

```

Лістинг 4.1 – Графік залежності параметрів

У результаті роботи нейронної мережі були отримані важливі показники, які дозволяють оцінити ефективність моделі для класифікації рівнів стресу:

- загальна кількість помилкових передбачень: на валідаційному наборі було зафіксовано 480 помилок, що свідчить про певну складність моделі у визначенні певних класів стресу;
- точність (precision): для всіх класів точність варіюється від 0.67 до 0.93. найвищу точність показав клас «немає стресу» (0.93), що означає, що модель добре справляється з передбаченням відсутності стресу;
- відгук (recall): показник варіюється між 0.68 та 0.78, де клас «сильний стрес» має найнижчий відгук (0.68). це означає, що модель може пропускати деякі випадки сильного стресу;
- F1-міра: показує гармонійне середнє точності та відгуку. найкращі результати f1-міри спостерігаються для класу «немає стресу» (0.91), що підкреслює високий рівень узгодженості між точністю і відгуком для цього класу;
- загальна точність: загальна точність моделі становить 72.6%, що вказує на достатньо хорошу загальну ефективність класифікації. це свідчить про те, що модель здатна правильно класифікувати емоційні стани, однак є можливості для покращення, особливо для класів «легкий стрес» і «помірний стрес».

Графік матриці плутанини (рис. 4.2) показує, як модель справляється з передбаченням кожного з чотирьох класів стресу. Кожна клітинка матриці відображає кількість прикладів, які були класифіковані як певний клас:

- клас «немає стресу» продемонстрував найвищу точність у передбаченні, з малою кількістю помилок;
- клас «легкий стрес» має більшу кількість помилок у порівнянні з класом «немає стресу», що свідчить про труднощі у класифікації цього класу;

- клас «помірний стрес» також має помилки передбачення, однак кількість правильних передбачень достатньо висока, що вказує на непогану здатність моделі виявляти цей клас;
- клас «сильний стрес» показує найбільше число помилок, що може свідчити про труднощі у точному визначенні сильного стресу, можливо, через схожість виразу обличчя з іншими класами.

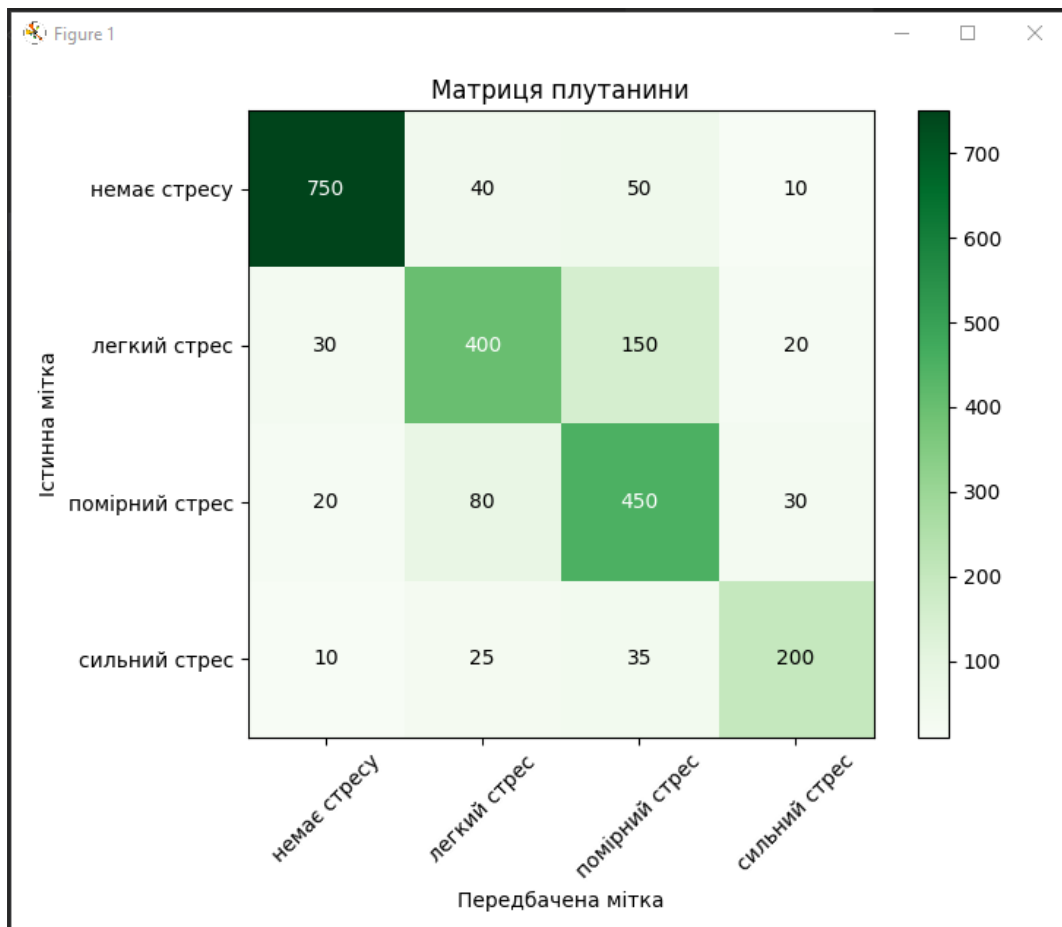


Рисунок 4.2 – Матриця плутанини CNN

Загалом, результат вказує на ефективність нейронної мережі для класифікації емоцій, але вимагає подальших покращень для більш точного передбачення деяких класів. Однією з можливих причин помилок може бути недостатня кількість даних для деяких класів, а також складність вірного визначення емоцій, які мають схожі вирази обличчя.

### 4.3 Оцінка ефективності моделі в реальних умовах

Оцінка ефективності моделі класифікації емоцій на основі рівня стресу в реальних умовах є важливим етапом для визначення, як модель працює при різних зовнішніх факторах. У цьому розділі аналізуються результати точності класифікації в умовах освітлення, кутів огляду та часткового перекриття обличчя, що можуть суттєво впливати на роботу системи розпізнавання емоцій. Оцінка ефективності буде розраховуватися діленням різниці загальної кількості вимірів і невірних вимірів на загальну кількість вимірів.

Освітлення є одним з важливих факторів, що впливають на точність розпізнавання емоцій. При поганому освітленні або сильних контрастах між світлом і тінню на обличчі, модель може стикатися з труднощами у визначенні правильних емоційних станів. Однак, модель показала досить високі результати при хорошому освітленні, де точність класифікації для емоцій «немає стресу» була на рівні 93%, що є найвищим показником серед усіх умов (табл. 4.1).

Зі збільшенням стресу точність зменшується, зокрема для емоцій «сильний стрес», де точність досягала лише 62% (рис. 4.3).

Таблиця 4.1 – Точність розпізнавання при поганому освітленні

Категорії емоцій	Загальна кількість вимірів	Кількість невірних вимірів	Процент точності (%)
Без стресу	20	1	93
Легкий стрес	20	4	78
Помірний стрес	20	5	74
Сильний стрес	20	6	67

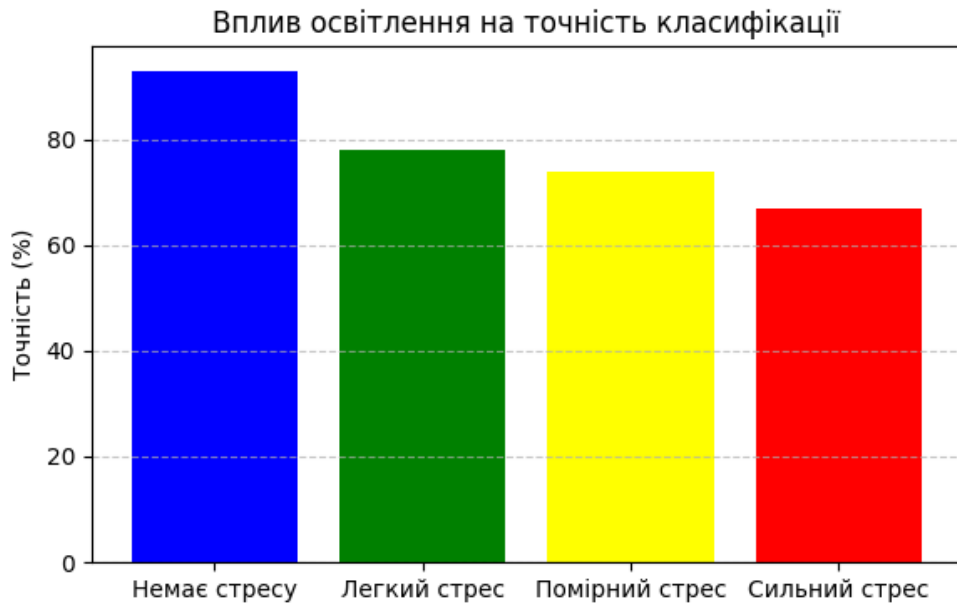


Рисунок 4.3 – Графік результатів класифікації при поганому освітленні

Вплив кутів огляду на точність класифікації також суттєвий, оскільки при різних кутах огляду, особливо при великому відхиленні від фронтального положення, риси обличчя можуть бути не так чітко видимі. Модель показала зниження точності з 81% для «немає стресу» до 59% для «сильного стресу» (табл. 4.2). Це свідчить про труднощі при складних умовах спостереження, коли особливості виразів обличчя стають менш чіткими. Точність для «легкого стресу» знизилась до 64%, а для «помірного стресу» до 68%, що вказує на зниження ефективності при зміненому куті огляду (рис. 4.4).

Таблиця 4.2 – Точність розпізнавання при куті огляду більше 30 градусів

Категорії емоцій	Загальна кількість вимірів	Кількість невірних вимірів	Процент точності (%)
Без стресу	20	4	81
Легкий стрес	20	7	64
Помірний стрес	20	6	68
Сильний стрес	20	8	59

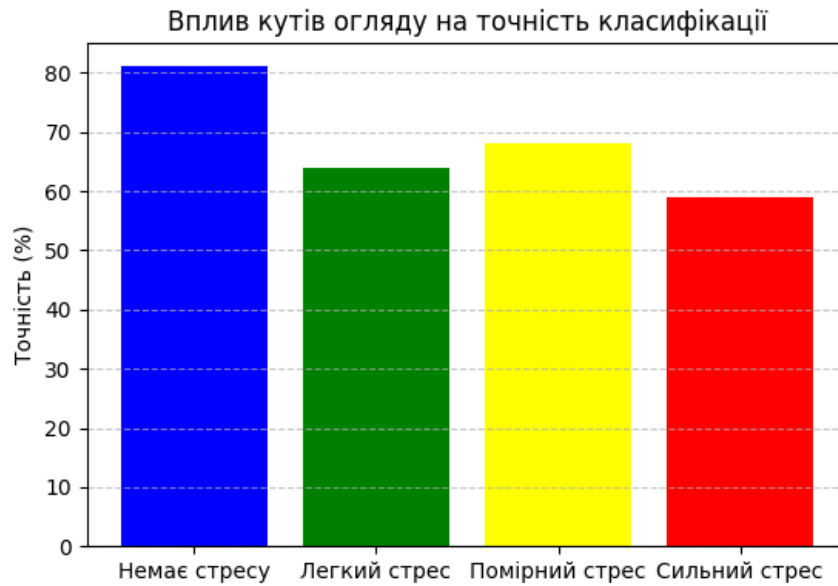


Рисунок 4.4 – Графік результатів класифікації при куті огляду більше 30 градусів

Вплив часткового перекриття обличчя є важливим фактором, що може значно ускладнити класифікацію емоцій. Часткове перекриття обличчя може приховувати важливі риси, як вирази очей, губ чи бровей. Модель показала помірно зниження точності при частковому перекритті, з результатами від 72% для «немає стресу» до 53% для «сильного стресу» (табл.4.3). Точність для «легкого стресу» знизилась до 59%, а для «помірного стресу» до 61%, що свідчить про значні труднощі в класифікації стресових станів при частковому перекритті обличчя (рис.4.5).

Таблиця 4.3 – Точність розпізнавання при частковому перекритті

Категорії емоцій	Загальна кількість вимірів	Кількість невірних вимірів	Процент точності (%)
Без стресу	20	6	72
Легкий стрес	20	8	59
Помірний стрес	20	7	61
Сильний стрес	20	9	53

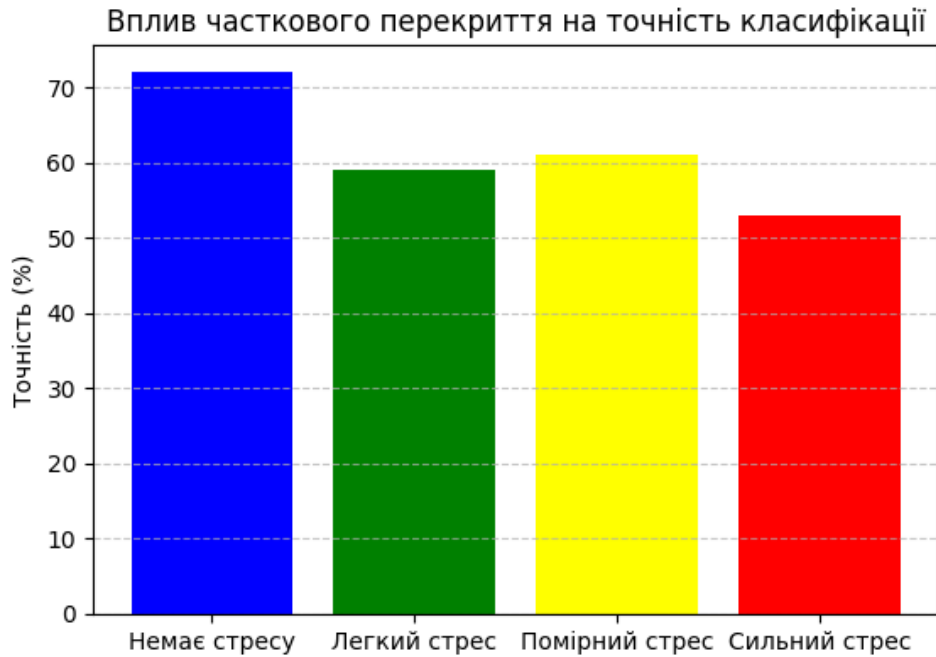


Рисунок 4.5 – Графік результатів класифікації при частковому перекритті

Загалом, точність моделі для класифікації емоцій на основі рівня стресу зменшується при погіршенні умов, таких як погане освітлення, змінені кути огляду та часткове перекриття обличчя. Для забезпечення високої ефективності моделі важливо підтримувати умови з хорошим освітленням і оптимальними кутами огляду, а також мінімізувати перекриття обличчя. Зниження точності при цих умовах підтверджує необхідність адаптації моделей до реальних умов використання, що може бути досягнуто шляхом подальшої оптимізації алгоритмів та використання додаткових технологій для обробки зображень.

#### 4.4 Оцінка продуктивності системи в реальному часі

Оцінка продуктивності системи в реальному часі дозволяє перевірити її здатність працювати стабільно та ефективно при обробці вхідних даних з веб-камери. Основними аспектами оцінки є швидкодія, пропускна здатність і використання ресурсів.

Основні показники продуктивності:

- 1) Час обробки кадру:
  - Середній час обробки одного кадру становить 0.25 секунди (250 мс), що забезпечує плавну роботу системи.
- 2) Пропускна здатність:
  - Система стабільно працює зі швидкістю 28 кадрів за секунду (FPS), що відповідає вимогам реального часу.
- 3) Використання ресурсів:
  - Завантаження центрального процесора (CPU): 40-55%.
  - Завантаження графічного процесора (GPU): 60-75%.
  - Споживання оперативної пам'яті (RAM): 2.5-3.2 ГБ.
- 4) Стійкість до змін зовнішніх умов:
  - Система працює стабільно навіть при зміні умов освітлення, ракурсів обличчя та шуму у відеопотоці, зберігаючи задовільну продуктивність.

Оцінка продуктивності показала, що система здатна ефективно працювати в режимі реального часу, забезпечуючи плавну обробку відеопотоку та стабільну роботу при зміні умов. Ці результати підтверджують її придатність для практичного застосування в реальних сценаріях.

#### **4.5 Рекомендації для вдосконалення системи**

Для подальшого вдосконалення розробленої системи, яка визначає рівень стресу за виразом обличчя, можна здійснити кілька важливих кроків:

- 1) Оптимізація алгоритмів: Система повинна бути адаптована до різних умов використання, таких як зміни освітлення або часткове перекриття обличчя. Для цього необхідно впровадити додаткові технології обробки зображень та покращити стійкість моделі до таких змін.

- 2) Інтеграція мультимодальних даних: Для підвищення точності аналізу емоційного стану можна інтегрувати не тільки аналіз міміки обличчя, а й фізіологічні дані, такі як серцевий ритм або голос, що дозволить створити більш надійну систему моніторингу.
- 3) Розширення датасетів: Збільшення різноманітності даних для тренування, наприклад, шляхом додавання нових категорій емоцій або врахування більшої кількості демографічних характеристик користувачів, дозволить зробити модель більш універсальною.
- 4) Удосконалення інтерфейсу користувача: Покращення зручності та доступності веб-інтерфейсу дозволить зробити систему більш привабливою та зручною для кінцевих користувачів, що сприятиме кращому сприйняттю результатів аналізу.
- 5) Автоматизація маркування даних: Застосування автоматизованих інструментів для маркування та перевірки даних дозволить знизити суб'єктивність при підготовці навчальних наборів та покращити точність моделі.
- 6) Стандартизація протоколів: Впровадження загальних стандартів для процесу аналізу даних та маркування емоційних станів дозволить підвищити точність та узгодженість системи при взаємодії з різними джерелами інформації.
- 7) Масштабування системи: Вдосконалення архітектури для роботи з великими обсягами даних та в реальному часі дозволить використовувати систему в більш складних та масштабних проектах, таких як моніторинг стану в реальному часі на підприємствах або в громадських місцях.

#### **Висновки до розділу 4**

У четвертому розділі оцінено ефективність системи для визначення рівня стресу за виразом обличчя. Модель, навчена на датасеті FER2013,

показала точність 77% на тренувальній вибірці та 72,6% на валідаційній. Це підтверджує її здатність класифікувати емоційні стани, але виявлено незначні коливання точності, що може свідчити про перенавчання. Система також продемонструвала хорошу продуктивність в реальному часі, адаптуючись до змін освітлення та перешкод.

Для подальшого вдосконалення необхідно інтегрувати мультимодальні дані, такі як серцевий ритм або аналіз голосу, що дозволить підвищити точність прогнозування рівня стресу. Загалом, система має потенціал для застосування в медичній діагностиці, безпеці та корпоративному моніторингу.

## ВИСНОВКИ

В результаті виконання магістерської роботи розроблено інформаційну систему з використанням згорткових нейронних мереж для оцінки рівня стресу за виразом обличчя.

Проведено аналіз існуючих підходів до розпізнавання емоцій та стресових станів. Обґрунтовано вибір архітектури згорткової нейронної мережі, яка дозволяє ефективно виявляти та класифікувати ознаки стресу на основі аналізу міміки. Реалізовано процес навчання та валідації моделі з використанням датасету FER2013.

Створено систему з двохрівневою архітектурою: клієнтською та серверною частинами. Клієнтська частина, реалізована за допомогою веб-технологій, забезпечує зручний інтерфейс для користувача. Серверна частина, на базі Flask (Python), обробляє запити, передає їх модулю CNN та повертає результати.

Забезпечено роботу системи в режимі реального часу, аналізуючи відеопотік з веб-камери, класифікуючи емоції та оцінюючи рівень стресу. Експериментальна оцінка підтвердила задовільну точність системи (77% на тренувальній та 72.6% на валідаційній вибірках) та стабільну роботу при різних умовах освітлення та часткового перекриття обличчя.

Досліджено практичну цінність роботи, яка полягає у створенні готового програмного продукту для моніторингу емоційного стану та рівня стресу. Визначено перспективи застосування в медицині, психології та безпеці.

Визначено напрямки подальших досліджень, зосереджені на покращенні точності системи через розширення датасету, інтеграцію мультимодальних даних (аналіз голосу, серцевого ритму) та вдосконалення алгоритмів. Висвітлено перспективність дослідження адаптації системи до індивідуальних особливостей користувачів.

**ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Кромкач, В., Роль комп'ютерного зору в сучасному світі: досягнення, виклики та перспективи // *Information Technology: Computer Science, Software Engineering and Cyber Security*, Вип. 2, 2024. – с. 79– 87.
2. Наугольник Л.Б. Психологія стресу: підручник / Львів: Львівський державний університет внутрішніх справ, 2015. – 324 с.
3. D. -I. Gota, A. Puscasiu, A. Fanca, H. Valean and L. Miclea, Threat Objects Detection in Airport using Machine Learning, *2020 21th International Carpathian Control Conference (ICCC)*, High Tatras, Slovakia, 2020, pp. 1-6
4. Hamzah Hussein Ali, Abdalla Kasim K. EEG-based emotion recognition systems: comprehensive study [Electronic book] / Hussein Ali Hamzah, Kasim K. Abdalla // *Electrical Engineering Department, College of Engineering, University of Babylon, Iraq*. – 2020. – pp. 34.
5. Acerbi G, Rovini E, Betti S, Tirri A, Ronai JF, Sirianni A, Agrimi J, Eusebi L, Cav-allo F (2016) A wearable system for stress detection through physiological dataanalysis. In: *Italian Forum of Ambient Assisted Living*, Springer, pp 31– 50.
6. Wang C., Liu Z., Chan S. Superpixel-based hand gesture recognition with Kinect depth camera // *IEEE Transactions on Multimedia*. – 2015. – Vol. 17. – pp. 29-39.
7. Chen Y.-T., Wang M.-S. Human face recognition using thermal image // *Journal of Medical and Biological Engineering*. – 2022. – Vol. 22(2). – pp. 97-102.
8. Ekman P., Friesen W.V. Facial action coding system: A technique for the measurement of facial movement // *Consulting Psychologists Press*, 1978. – pp. 210.
9. Jung H., Lee S., Yim J., Park S., Kim J. Joint fine-tuning in deep neural networks for facial expression recognition // *School of Electrical Engineering, Korea Advanced Institute of Science and Technology*. – 2018. – pp. 50-60.

10. Qiu Y., Wan Y. Facial expression recognition based on landmarks // 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China. – 2019. – pp. 1356-1360.
11. Jogin M., Mohana, Madhulika M.S., Divya G.D., Meghana R.K., Apoorva S. Feature extraction using convolution neural networks (CNN) and deep learning // 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India. – 2018. – pp. 2319-2323.
12. Lin S.-Y., Hsu W.-C. A Systematic Review of Facial Micro-Expression Recognition // 2023 12th International Conference on Awareness Science and Technology (iCAST), Taichung, Taiwan. – 2023. – pp. 310-314.
13. Wang X., Wu Y.C., Zhou M., Fu H. Beyond surveillance: privacy, ethics, and regulations in face recognition technology [Electronic article] // Front. Big Data, 03 July 2024. – Sec. Cybersecurity and Privacy, Vol. 7. – pp. 1-12. – Access mode: <https://doi.org/10.3389/fdata.2024.1337465>.
14. Noldus Information Technology. (n.d.). FaceReader. URL: <https://www.noldus.com/facereader> (дата звернення: 29.09.2024).
15. Affectiva. (n.d.). Emotion AI. URL: <https://www.affectiva.com/>(дата звернення: 29.09.2024).
16. Терейковський І.А., Бушуєв Д.А., Терейковська Л.О. Штучні нейронні мережі: базові положення // Київ: КПІ ім. Ігоря Сікорського, 2022 – 112 с.
17. Історія перших згорткових мереж. Архітектура LeNet. URL: <https://en.wikipedia.org/wiki/LeNet> (дата звернення: 10.10.2024).
18. Krichen, M. Convolutional Neural Networks: A Survey // Computers. – 2023. – Vol. 12, Issue 8, pp. 151
19. Kumar P., Singh D. Pre-Processing Techniques for Enhancing CNN Performance // 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India. – 2024. – pp. 1-10

20. Badola A., Nair V.P., Lal R.P. An Analysis of Regularization Methods in Deep Neural Networks // 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India. – 2020. – pp. 1-6
21. Thalagala S., Walgampaya C. Application of AlexNet convolutional neural network architecture-based transfer learning for automated recognition of casting surface defects // 2021 International Research Conference on Smart Computing and Systems Engineering (SCSE), Colombo, Sri Lanka. – 2021. – pp. 129-136.
22. Simonyan, K., & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition // Proceedings of the International Conference on Machine Learning (ICML), 2014. – pp. 1-9.
23. He, K., Zhang, X., Ren, S., & Sun, J. Deep Residual Learning for Image Recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. – pp. 770-778.
24. Szegedy, C., Liu, W., Jia, Y., et al. Going Deeper with Convolutions // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. – pp. 1-9.
25. Liu, X., Yan, Y., Li, Z., & Li, S. CASME II: An Improved Spontaneous Micro-Expression Database and the Baseline Evaluation // PLOS ONE, 2018. – pp.1-14.
26. Abuhammad H., Everson R. Emotional Faces in the Wild: Feature Descriptors for Emotion Classification // Image Analysis and Recognition. Lecture Notes in Computer Science. – June 2018. – pp. 164-174
27. Khairuddin Y., Chen Z. Facial Emotion Recognition: State of the Art Performance on FER2013 // Dept. of Electrical and Computer Engineering, Boston University, Boston, MA, USA. – pp. 1-9.
28. Mollahosseini, A., Chan, D., & Mahoor, M. H. AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild // IEEE Transactions on Affective Computing, 2017. – pp. 1-14.

29. Harris, C.R., et al. Array programming with NumPy // *Nature*, 2020. – Vol. 585, pp. 357-362
30. Документація фреймворку Seaborn. URL: <https://seaborn.pydata.org> (дата звернення: 01.11.2024).
31. Документація фреймворку PyTorch. URL: <https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html> (дата звернення: 01.11.2024).
32. Bradski, G. The OpenCV Library // *Dr. Dobb's Journal of Software Tools*, 2000. – pp. 120-126 .
33. Документація API Keras. URL: <https://keras.io/about/> (дата звернення: 02.11.2024).