

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: **«Система розпізнавання емоцій людини із
використанням алгоритмів машинного навчання»**

Виконав: студент групи Іт-62

Спеціальності 126 «Інформаційні
системи та технології»

(шифр і назва)

Владика Дем'ян Михайлович

(Прізвище та ініціали)

Керівник: к.т.н., в.о. доцента Падюка Р.І.

(Прізвище та ініціали)

Рецензент: _____

(Прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти Спеціальність 126
«Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

д.т.н., проф. А.М. Тригуба

«_____» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Владиці Д'ємяну Михайловичу

-
1. Тема роботи: «Розробка системи розпізнавання емоцій людини із використанням алгоритмів машинного навчання»

Керівник роботи: Падюка Роман Іванович, в.о. доцента

затверджені наказом по університету від 28 квітня 2023 року № 133/к-с.

2. Строк подання студентом роботи 10.01.2024 р.

3. Вихідні дані до роботи: набір даних для навчання мережі; алгоритми машинного навчання; методика дослідження моделей машинного навчання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Вступ.

1. Аналіз стану комп'ютерного зору та завдання кваліфікаційної роботи.

2. Особливості вирішення задач класифікації та вибір методів машинного навчання для розпізнавання емоцій людини

3. Результати розробки системи розпізнавання емоцій людини

4. Охорона праці та безпека у надзвичайних ситуаціях.

5. Визначення економічної ефективності запропонованої системи.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових слайдів): Мета проектування; практична цінність; дерево проблем та дерево цілей; проблеми пов'язані з визначенням емоцій; згорткова нейронна мережа; технології і інструменти які використовуються для реалізації системи; пракад роботи системи; висновки.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	<i>Падюка Р.І., в.о.доцента кафедри інформаційних технологій</i>		
4	<i>Городецький І.М., доцент кафедри управління проектами та безпеки виробництва</i>		

7. Дата видачі завдання завдання 01 травня 2023 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	При-мітка
1	<i>Написання першого розділу</i>	01.05.-30.05.23	
2	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	01.06.-30.06.23	
3.	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	01.09.-30.09.23	
4.	<i>Написання розділу «Охорона праці та безпека унадзвичайних ситуаціях»</i>	01.10.-30.10.23	
5.	<i>Оцінення ефективності запропонованої системи</i>	01.11.-30.11.23	
6.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	01.12.-30.12.23	
7.	<i>Завершення роботи в цілому</i>	01.01.-10.01.24	

Студент _____ Владика Д.М.
(підпис)

Керівник роботи _____ Падюка Р.І.
(підпис)

УДК 004.9 : 631.1

Розробка системи розпізнавання голосових даних із використанням алгоритмів машинного навчання.

Владика Д.М. Кафедра інформаційних технологій – Дубляни, ЛНУП, 2024. Кваліфікаційна робота: 80 с. текст. част., 21 рисунок, 7 таблиць, 13 арк.ілюстраційного матеріалу, 22 джерела і 1 додаток.

Кваліфікаційна робота присвячена проектуванню та розробці системи розпізнавання емоцій людини з використанням штучних нейронних мереж.

У вступній частині обгрунтовано актуальність дослідження, визначено мету, об'єкт, предмет, завдання, методи наукового дослідження кваліфікаційної роботи та описано теоретичну та практичну цінність роботи

В першому розділі описано об'єкт дослідження проектованої системи та здійснено постановку задачі на бакалаврську роботу. А також здійснено огляд літературних джерел та проаналізовано наявні підходи до вирішення проблеми.

У другому розділі проведено системний аналіз розроблюваної системи та побудовано дерева проблеми та цілей. Та описано алгоритми роботи системи у вигляді різних діаграм.

У третьому розділі описано основні моменти реалізації модулів та інтерфейсу системи, що реалізується. Також описується процес тестування та інструкції користувачам із встановлення та користування системою.

У четвертому розділі описано охорону праці та безпеку у надзвичайних ситуаціях.

У п'ятому розділі зроблено розрахунок вартості системи та проведено оцінку економічної ефективності системи.

У висновках підсумовані результати теоретичного та практичного дослідження кваліфікаційної роботи.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ СТАНУ ТЕОРІЇ ТА ПРАКТИКИ СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ТА ПОСТАНОВКА ЗАВДАННЯ.....	8
1.1 Основні поняття.....	8
1.2 Характеристика об'єкту проектування	8
1.3 Постановка задачі.....	9
1.4 Сфера застосування.....	10
1.5 Огляд принципів роботи нейронної мережі	14
1.6 Огляд існуючих нейронних мереж для задач розпізнавання.....	17
1.6.1 Згорткова нейронна мережа.....	18
Висновок до розділу	22
РОЗДІЛ 2. РЕЗУЛЬТАТИ ПРОЕКТУВАННЯ (МОДЕЛЮВАННЯ) ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ.....	23
2.1 Прогнозування проблем, що можуть виникнути	23
2.2 Побудова дерева проблем.....	23
2.3 Побудова дерева цілей.....	26
2.4 Концептуальна модель.....	30
2.5 Діаграма IDF0	31
2.5.1 Контекстна діаграма системи.....	31
2.5.2 Діаграма декомпозиції системи 1 рівня	32
2.5.3 Діаграма декомпозиції системи 2 рівня	33
2.6 Принципи роботи згорткові нейронних мереж.....	34
2.6.1 Вхідні та вихідні дані.....	34
2.6.2 Біологічні зв'язки.....	35
2.6.3 Перший шар - математична частина	35
2.6.4 Наступні шари мережі	37
Висновки до розділу.....	38

	4
РОЗДІЛ 3. РЕЗУЛЬТАТИ ВИРІШЕННЯ ЗАДАЧІ	39
3.1 Загальна структура програмного проекту	39
3.2 Опис використаних сторонніх бібліотек та модулів	40
3.3 Розробка та опис навчання нейронної мережі.....	41
3.4 Опис проблем і нестандартних ситуацій, які виникали під час розробки та заходів для їх вирішення.....	44
3.5 Інструкції програмісту	44
3.5.1 Навчання мережі	45
3.5.2 Вимоги до апаратного забезпечення	49
3.5.3 Тестування	50
Висновки до розділу.....	52
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ	53
4.1 Аналіз небезпечних і шкідливих виробничих чинників та розробка заходів щодо покращення умов праці	53
4.2. Розробка логічно-імітаційної моделі процесу виникнення травм під час монтажу інтелектуальної інформаційної системи	53
РОЗДІЛ 5. ВИЗНАЧЕННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ.....	60
5.1 Економічна характеристика проектного рішення.....	60
5.2 Інформаційне забезпечення та формування гіпотези щодо потреби розроблення товару	60
5.3 Бюджетування.....	61
Висновки до розділу.....	67
ВИСНОВКИ І ПРОПОЗИЦІЇ	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71
ДОДАТКИ.....	73
ДОДАТОК А.....	73

Перелік умовних позначень та термінів

Dataset – електронний набір даних.

НМ – нейронна мережа.

API - це набір готових класів, процедур, функцій, структур і констант, що надаються додатком (бібліотекою, сервісом) для використання в зовнішніх програмних продуктах.

Widget - контент-модуль.

TensorFlow — відкрита програмна бібліотека для машинного навчання цілій низці задач, розроблена компанією Google для задоволення її потреб у системах, здатних будувати та тренувати нейронні мережі

CNN – згорткова нейронна мережа

OpenCV - бібліотека комп'ютерного зору з відкритим кодом) — бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом.

ВСТУП

Повноцінне спілкування між людьми неможливе без прояву і аналізу емоцій. В сучасному світі, напрям розпізнавання емоцій людини є дуже цікавим не тільки для психологів, а й для багатьох розробників та маркетологів. Емоції людини, наприклад радість або сміх, можна трактувати як гарну оцінку для якогось фільму чи відеоролика. Саме визначення, які емоції відчуває чи показує людина, дуже важливе. Це дає змогу зрозуміти, чи сподобався людині товар, як вона на нього реагує тощо.

В останні роки розпізнавання емоцій стало особливо актуальним, коли з'явилися і розвинулися мобільні пристрої та комп'ютери, вчені та розробники дійшли згоди, що прояв людських емоцій – один із головних напрямків для більш гнучкого розвитку систем штучного інтелекту. Такі системи використовуються в багатьох сферах нашого життя, вони дозволяють підвищувати рівень безпеки, наприклад: в сучасних автомобілях існують системи розпізнавання людської втоми або сонливості, при виявленні яких, водію приходять сигнали, які не дають йому заснути. На основі міміки людей, багато психологів можуть робити висновки про те, чи приховує щось людина, саме такі моменти і цілі переслідують програмісти, при створенні автоматизованих систем.

Отже, при створенні сучасних людино-машинних систем є **актуальним** застосування методів автоматичного розпізнавання емоцій. Такі системи і методи з кожним роком розширюють свою область застосування.

Метою бакалаврської кваліфікаційної роботи є проектування та розробка системи розпізнавання емоцій людини з використанням штучних нейронних мереж.

Об'єктом проектування є процеси аналізу даних за допомогою штучних нейронних мереж.

Предметом проектування є методи та засоби розпізнавання емоцій людини з використанням штучних нейронних мереж.

Для досягнення моєї цілі, я перед собою поставив такі **завдання**:

- провести огляд літератури та визначити основні поняття;
- спрогнозувати проблеми, що можуть виникнути та розробити алгоритм роботи;
- обрати необхідні інструменти та мову програмування для розробки;
- реалізувати програмний продукт;
- провести тестування готового продукту.

Практична цінність. В першу чергу розробка буде цінна для наукових досліджень, пов'язаних з обробкою та аналізом даних для напряму комп'ютерного зору. З іншої сторони, буде корисна, як аналізатор емоцій на співбесідах, що допоможе скласти психологічну оцінку людини, оскільки емоційний стан людини напряму пов'язаний з її працездатністю і умінням працювати у колективі. Окрім цього, подібні системи можна вбудовувати в інтелектуальні системи автомобіля, щоб визначати чи особа за кермом в стані вести автомобіль.

РОЗДІЛ 1

АНАЛІЗ СТАНУ ТЕОРІЇ ТА ПРАКТИКИ СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Основні поняття

Комп'ютерний зір або Комп'ютерне бачення — теорія та технологія створення машин, які можуть проводити виявлення, стеження та визначення об'єктів[12].

Комп'ютерний зір – це наукова дисципліна, яка вивчає теорію та технологію створення штучних систем, які містять інформацію з зображенням. Відеодані можуть бути представлені в різних формах, наприклад , на фоні відео, зображення з різних камер або тривимірні дані медичного сканера.

Комп'ютерний зір, як технологічна дисципліна, передбачає використання теорії та моделі комп'ютерного зору для створення системи комп'ютерного зору. Прикладами таких систем є системи управління процесами, такі як промислові роботи та автономні транспортні засоби; відеоспостереження; організація інформації, така як індексація бази даних зображення; моделювання об'єктів або навколишнього середовища, такі як аналіз медичних зображень; і системи взаємодії, такі як пристрої, введені для системи людино-машинної взаємодії.

Комп'ютерний зір можна описати як доповнення, хоча завжди не протилежність, біологічного зору. Біологія вивчає очі як людини, так і різних тварин. У результаті цього дослідження створено моделі роботи таких систем у термінах фізіологічних процесів. З іншого боку, комп'ютерний зір вивчає та описує системи комп'ютерного зору, які забезпечуються апаратно або програмно. Дослідження показали, що взаємодія між дисциплінами біології та комп'ютерної науки була досить корисною для обох наукових галузей.

1.2 Характеристика об'єкту проектування

Об'єктом проектування даної роботи є процеси аналізу даних за допомогою штучних нейронних мереж, а саме розпізнавання емоцій людини.

Аналіз візуальної інформації є основним методом виявлення емоцій іншої людини. Отже, автоматизація цього процесу очевидно повинна базуватися на методах і засобах комп'ютерного зору.

Дослідження в галузі комп'ютерного зору охоплюють теорію та основні алгоритми аналізу зображення об'єктів і сцени [1]. Часто також використовуйте слово «машинний зір» або «технічний зір» замість слова «комп'ютерний зір». Тим не менш, останні ідеї належать до більш широкої науково-практичної сфери, яка охоплює всі етапи розробки системи, що базуються на обробці та аналізі відеоінформації.

Звичайно, комп'ютерний зір і його методи не можуть гарантувати 100% точного визначення емоцій. Це тому, що обличчя може бути неправильно розташовано до камери та аксесуари можуть заважати. Крім того, основною причиною є унікальність емоцій, оскільки кожна людина може сприймати або демонструвати певний вираз обличчя окремим чином..

1.3 Постановка задачі

Постановка задачі звучить наступним чином: «Розробити програмний засіб для детектування емоції людини із використанням алгоритмів машинного навчання». Система має працювати у будь-якій операційній системі з використання сучасних інформаційних технологій.

Проаналізувавши постановку задачі, можна виділити ключові питання, які необхідно розглянути:

1. Які інформаційні технології слід використати для розробки системи?
2. Які алгоритми слід використати для розробки поставленої задачі?
3. Чим ця система буде кращою за вже існуючі аналоги?
4. Як спроектувати графічний інтерфейс користувачів, щоб він був і приємним на вигляд, і дозволяв виконати необхідні операції за мінімальну кількість дій?
5. Чи окупиться розробка системи?

Ці питання було виділено з наступних причин:

1. Технології для розробки системи цікавлять, насамперед, програмістів. Вибір коректних мов та засобів для команди дозволить не лише завершити розробку згідно з поставленими термінами, а й, можливо, пришвидшити її процес. Додатково, коректний вибір технологій може як покращити функціонал (логіку), так і представлення (інтерфейс для користувачів) системи та зробити її більш гнучкою та привабливою на вигляд.

2. Вибір алгоритмів при розробці є дуже важливим. Неправильно підібраний для конкретної задачі алгоритм, як мінімум, погіршить точність результатів, а у гіршому випадку призведе до абсолютно хибних результатів. Таку систему користувачі не будуть використовувати.

3. Розгляд аналогів може на початковому етапі відповісти на питання щодо доцільності створення системи. Якщо на ринку вже представлено продукт, який має хорошу репутацію та широку функціональність, а наш продукт буде дещо гіршим або лише незначною мірою надасть додаткові несуттєві можливості, користувачі не перейдуть до використання нашої системи.

4. Приємний та легкий у використанні інтерфейс – важка задача, що стає важчою при зростанні складності системи. Якщо кольори та шрифти підібрані не зовсім вдало, але функціонал системи дуже багатий та працює коректно – користувачі ще зможуть з цим змиритись, але все одно шукатимуть аналоги з кращим зовнішнім виглядом. Якщо ж у системі все буде відображено «на купу», а для операцій знадобиться виконати багато зайвих кроків, це однозначно спонукає користувача відмовитись від такої системи.

5. На останнє питання слід відповісти після відповіді на п'ять попередніх. Проаналізувавши можливості нашої системи та можливості систем-аналогів на ринку, її зовнішній вигляд та працездатність, можна або впевнитись у доцільності її створення, або відкинути ідею її розробки. Додатково, слід провести економічну оцінку вартості розробки системи для визначення її ціни, оскільки цей фактор для замовника має велике значення в більшості випадків.

1.4 Сфера застосування

Потреба в технологіях, здатних оцінити потреби потенційних клієнтів і вибрати найкращі рішення, різко зростає в результаті швидкого зростання використання інтелектуальних технологій у суспільстві та розвитку галузі. Автоматична емоційна оцінка особливо важлива в таких сферах, як робототехніка, маркетинг, освіта та розваги. такі системи використовуються для досягнення різноманітних цілей:

- в робототехніці для розробки інтелектуальних спільних або сервісних роботів, які можуть взаємодіяти з людьми;

В даний час активно розвивається робототехніка. З самого початку роботи використовувалися у військовій промисловості, потім вони почали виконувати побутові домашні справи. Згідно з робототехнічними конференціями щороку, здається, що роботи скоро починають повністю замінювати людей. Робот-пилосос, оснащений засобами машинного зору, зараз є хорошим прикладом. В окремих робототехнічних системах машинний зір використовується в основному для визначення положень, аналізу навколишнього середовища, об'їзду перешкод, винесення конкретних предметів і так далі. Крім того, важливо пам'ятати, що реабілітаційна робототехніка стає все більш популярною, і для цього необхідно правильно взаємодіяти з людьми, оцінюючи їх емоційний, чисельний та емоційний стан. Щоб покращити соціальні функції роботи, можливо, найкращий спосіб — дати роботам справжні протоколи навчання. Таким чином, здатність побутових робіт в природному порядку розпізнавати емоції людини дозволяє підвищити рівень інтелектуалізації їх взаємодії., наприклад, для забезпечення правильного реагування на стан людини.

- в маркетингу: створювати спеціалізовані рекламні оголошення, засновані на емоційному стані потенційного клієнта;

Сет Годін, всесвітньо відомий підприємець і письменник з Америки, сказав: «Люди не купують товари і послуги. Вони купують магію, історію та зв'язок.

Здатність людини усвідомлювати почуття, досягати їх і створювати їх, допомагаючи думати, розуміти почуття та те, що вони позначають, контролюють

їх, підвищуючи свій емоційний та розумовий розвиток, є основою емоційного маркетингу. Емоційний маркетинг означає правильно розуміти стрес, впливати на неї, розуміти, чого хочуть і чого потребують інші, знати свої сильні та слабкі сторони, не піддаватися стресу та бути привабливим. Сьогодні рекламисти, маркетологи та брендисти звертають велику увагу на проблеми, які впливають на людей.

У маркетинговій сфері системи розпізнання емоцій і облич можуть бути використані з метою оперативного відстеження та реагування на різні проблеми в торгових центрах, супермаркетах та інших місцях продажів товарів і послуг, наприклад:

- визначення черг і їх оптимізація;
- підрахунок відвідувачів з класифікацією за віком, статтю, расою;
- оцінка якості обслуговування;
- аналіз поведінки персоналу;
- аналіз ефективності промоакцій;
- аналіз ефективності методики продажу товару в магазині (мерчендайзингу);
- визначення «гарячих» зон в магазинах;
- показ реклами в залежності від статі і віку;
- визначення оптимального положення рекламних місць.

Автоматичне виявлення емоцій клієнтів може значно підвищити ефективність вирішення різноманітних маркетингових завдань.

- в освіті: використовується для поліпшення процесів навчання, передачі знань і методологій сприйняття;

У будь-якому навчальному середовищі, будь то в класі чи онлайн, емоції студентів дуже важливі. Розпізнавання емоцій є місцем для розуміння настрою учня або студента в середовищі навчання. Кінцева мета цієї роботи — отримати ступінь розуміння залучення учнів до навчання, ефективності та корисності впровадження системи. Поступова оцінка емоцій учнів може допомогти у

покращенні навчального досвіду та оновленні змісту навчання. Такий емоційний аналіз може допомогти у зміні стратегії навчання. Можливість постійного відеоспостереження є основним недоліком системи розпізнавання емоцій.

- безпека людей;

За висновком експертів з безпеки, яскраві кадри чоловіків є одними з найжахливіших фотографій, зроблених 11 вересня 2001 року, коли терористична атака зруйнувала комплекс ВТЦ. Жахливо, тому саме в цей момент було б можна вжити заходів, щоб запобігти катастрофі. Якби існувала технологія, здатна застосувати Атту та Аль-Омарі як результатів захоплених тероризмом, безпеку аеропорту можна було б забезпечити і врятувати тисячі життя. Майже через двадцять років безпека залишається однією з окремих сфер розпізнавання людей.

Для ідентифікації емоцій особливо важливо точно застосувати стан людини в ситуаціях, коли його життя під загрозою. Системи розпізнавання втомі людини, які є частиною сучасних автомобілів, можуть служити прикладом. Подібні системи можуть у багатьох випадках запобігти аваріям, які виявляються через необережність водія, сонливість або погане самопочуття. Аналіз базується на результатах обробки зображення особи людини з відеокамери.

Забезпечення безпеки людей за допомогою автоматизованої системи охорони є ще однією сферою застосування методів автоматичного розпізнавання емоцій. У сучасних охоронних системах часто присутні інструменти для запису та аналізу відеоданих. Але в таких системах рішення, як правило, приймають оператори. Це може призвести до того, що реагування на різні ситуації, пов'язані з поведінкою людей на охоронюваних територіях, буде менш точним і менш ефективним. Підвищення ступеня автоматизації процедур, що забезпечують аналіз зображення, може призвести до підвищення ефективності охоронних систем. Це дозволяє дійти до висновку, що створення інтелектуальних детекторів для аналізу позаштатних ситуацій є виробництвом. Іншими проблемами, які необхідні для вирішення конкретних охоронних систем, є виявлення та підтримка людей, які демонструють емоції, характерні для порушень правопорядку, психічно хворих і т. д. [3].

Підводячи підсумок, слід зазначити, що розпізнавання обличчя та емоцій також має свої негативні сторони чи інші ризики. Конфіденційність є першим ризиком. погоджуються на те, що за ними слідкують камери, технологія розпізнавання обличчя може бути використана для повного спостереження за шкірною людиною. __ У той же час багато провідних компаній працюють над створенням технологій, які розпізнають емоції та обробляють ці дані, навіть якщо людина про це навіть не здогадується. Це може призвести до низки проблем з конфіденційністю. Другий можливі ризики викликають проблеми безпеки даних, яка зараз є надзвичайно актуальною.

Більшість компаній, які мають обмеження (або намір використовувати) таку технологію, повинні обов'язково встановлювати значення попередніх знаків та повідомлень, щоб зменшити виявлення людей про ці небезпеки.

1.5 Огляд принципів роботи нейронної мережі

Штучні нейронні мережі[17] (ANN) складають невід'ємну частину процесу глибокого навчання. Вони надихаються неврологічною структурою мозку людини. Згідно з AILabPage, ANN - це «складний комп'ютерний код, написаний кількістю простих, взаємопов'язаних елементів обробки, який натхненний біологічною структурою мозку людини для імітації моделей роботи та обробки даних мозку (Інформація)».

Поглиблене навчання фокусується на п'яти основних нейронних мережах, включаючи:

- Багат шаровий перцептрон
- Мережу радіальних основ
- Нейронні мережі, що повторюються
- Генеративні змагальні мережі
- Свертові нейронні мережі

У нейронній мережі потік інформації відбувається двома шляхами -

- Мережі прямої передачі: у цій моделі сигнали рухаються лише в одному напрямку, до вихідного шару. Мережі Feedforward мають вхідний рівень і один вихідний рівень з нульовим або декількома прихованими шарами. Вони широко використовуються при розпізнаванні зразків.

- Мережі зворотного зв'язку: У цій моделі рекурентні або інтерактивні мережі використовують свій внутрішній стан (пам'ять) для обробки послідовності входів. У них сигнали можуть рухатися в обох напрямках через петлі (прихований шар / шари) в мережі. Вони зазвичай використовуються в часових рядах та послідовних завданнях.

Нейрон - основна одиниця нейронної мережі. Вони отримують вхідні дані із зовнішнього джерела чи інших вузлів. Кожен вузол пов'язаний з іншим вузлом із наступного шару, і кожне таке з'єднання має певну вагу. Ваги присвоюються нейрону на основі його відносної важливості щодо інших вхідних даних.

Коли всі значення вузлів із жовтого шару множать (разом з їх вагою) та підсумовують, це генерує значення для першого прихованого шару. На основі підсумованого значення, синій шар має заздалегідь визначену функцію «активації», яка визначає, чи буде цей вузол «активованим» і наскільки «активним» він буде.

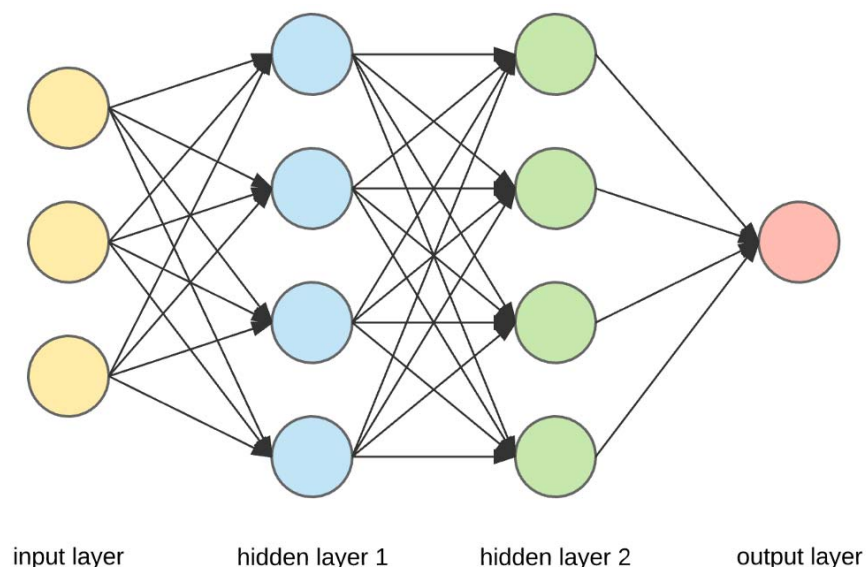


Рисунок 1.1. - Базова структура нейромережі

Шар або шари, приховані між вхідним і вихідним шаром, відомий як прихований шар. Його називають прихованим шаром, оскільки він завжди прихований від зовнішнього світу. Основне обчислення нейронної мережі відбувається в прихованих шарах. Отже, прихований рівень бере всі вхідні дані з вхідного рівня і виконує необхідні обчислення для отримання результату. Потім цей результат пересилається на вихідний рівень, щоб користувач міг переглянути результат обчислення.

У нейронній мережі процес навчання зніціюється поділом даних на три різні набори:

- Набір навчальних даних - Цей набір даних дозволяє нейронній мережі розуміти ваги між вузлами.
- Набір даних перевірки - Цей набір даних використовується для точної настройки продуктивності нейронної мережі.
- Тестовий набір даних - Цей набір даних використовується для визначення точності та межі помилок нейронної мережі.

Як тільки дані сегментовані на ці три частини, до них застосовуються алгоритми нейронної мережі для навчання нейронної мережі. Процедура, що використовується для полегшення тренувального процесу в нейронній мережі, відома як оптимізація, а використовуваний алгоритм називається оптимізатором. Існують різні типи алгоритмів оптимізації, кожен із яких має свої унікальні характеристики та такі аспекти, як вимоги до пам'яті, числова точність та швидкість обробки.

Перш ніж заглибитися в обговорення різних алгоритмів нейронної мережі, давайте спочатку розберемося з проблемою навчання.

Ми представляємо навчальну проблему з точки зору мінімізації індексу втрат(f). Тут " f " - це функція, яка вимірює продуктивність нейронної мережі на даному наборі даних. Як правило, індекс збитків складається з терміну помилки та терміну регуляризації. Хоча термін помилки оцінює, як нейронна мережа відповідає набору даних, термін регуляризації допомагає запобігти проблемі переобладнання, контролюючи ефективну складність нейронної мережі.

Функція втрат $[f(w)]$ залежить від адаптаційних параметрів - ваг та упереджень - нейронної мережі, які можна згрупувати в єдиний n -мірний вектор ваги (w) .

Ось наочне зображення функції втрат:

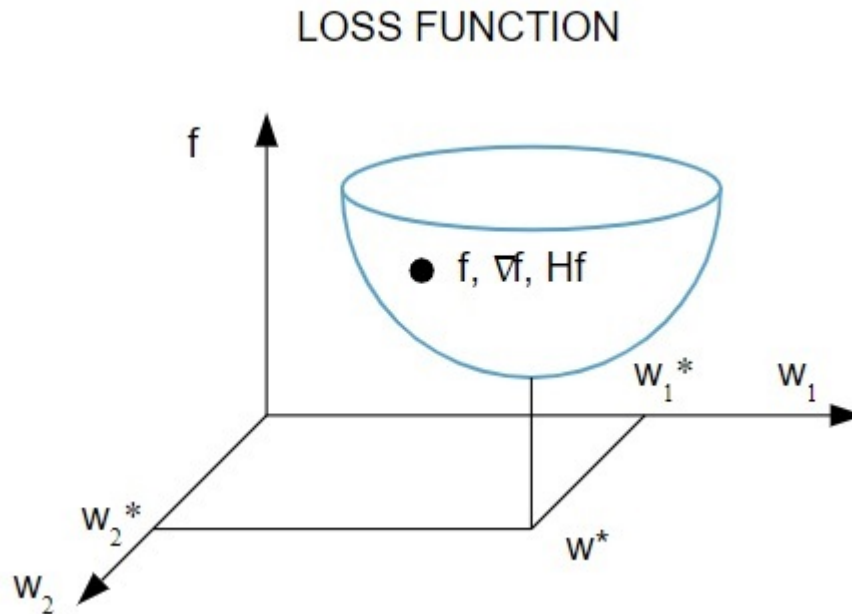


Рисунок 1.2. - Функція втрат

Згідно з цією діаграмою, мінімум функції втрат відбувається в точці (w^*) . У будь-який момент ви можете обчислити першу та другу похідні функції втрат. Перші похідні згруповані у градієнтному векторі, а його компоненти зображені як:

$$\nabla_i f(w) = \frac{\partial f}{\partial \omega_i} \quad \text{тут } i = 1, \dots, n.$$

Другі похідні від функції збитків згруповані в матриці Гесса так:

$$H_{i,j} f(w) = \frac{\partial^2 f}{\partial \omega_i \partial \omega_j}, \quad \text{тут } i, j = 0, 1, \dots$$

1.6 Огляд існуючих нейронних мереж для задач розпізнавання

В наш час вже є декілька технологій для вирішення цієї проблеми:

1. Гістограма орієнтованих градієнтів (HOG)
2. Конволюційна нейронна мережа (CNN)

3. Faster R-CNN, Mask R-CNN

Для даної роботи, найкращим типом нейромережі є згорткові нейромережі, оскільки вони найкраще справляються з задачею розпізнавання зображень та відеопотоків.

1.6.1 Згорткова нейронна мережа

У згортковій нейронній мережі[20] (CNN) використовується варіація багатошарових перцептронів. CNN складається з одного або багатьох згорткових шарів. Ці шари можуть бути повністю об'єднані або повністю взаємопов'язані.

Згортковий рівень використовує згорткову операцію на вході перед передачею результату наступного шару. Ця згортка дозволяє мережі бути досить глибою, але з меншими параметрами .

Ця здатність робить згорткові нейронні мережі надзвичайно ефективними в обробці природної мови, системах рекомендацій і розпізнавання зображень і відео.

Крім того, згорткові нейронні мережі чудово підходять для виявлення парафрази та семантичного синтаксичного аналізу. Обробка сигналів і класифікація зображень є двома областями, які також виконують їх.

Крім того, CNN також використовують у сільському господарстві для аналізу та розпізнавання зображень. У цих випадках погодні особливості, отримані із супутників, таких як LSAT, використовують для прогнозування зростання та врожаю на конкретній території. Ось зображення того, як виглядає згорткова нейронна мережа.

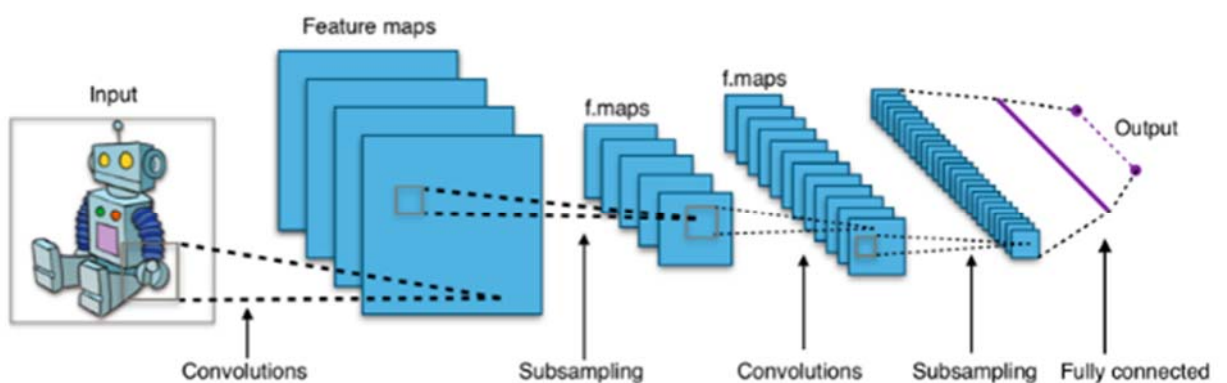


Рисунок 1.3. - Приклад згорткової нейромережі

Метакласифікації зображення CNN обробляє вхідні зображення та класифікує їх у різні категорії, наприклад тарілки, миски, чи глечики. Комп'ютери сприймають вхідне зображення як масив пікселів на основі його роздільної здатності. Він побачить $h \times w \times d$ (h = висота, w = ширина, d = розмір) відповідно до роздільної здатності зображення. Наприклад, зображення масиву матриці зображення сірого розміру 4 на 4 на 1 і зображення масиву матриці зображення RGB розміром 6 на 6 на 3 (3 відносяться до значення RGB). Для технічного навчання та тестування моделей CNN використовується функція Softmax, об'єднані, повністю зв'язані шари (FC) і кілька згорткових шарів із фільтрами (Kernels) з імовірними значеннями від 0 до 1.

Згорткові нейронні мережі чудово працюють із входами зображення, мови чи звукового сигналу, що відрізняє їх від інших нейронних мереж. Вони складаються з трьох основних типів шарів, а саме:

- Згортковий шар
- Об'єднуючий шар
- Повністю з'єднаний шар

Згортковий шар - це перший шар згорткової мережі. У той час як за згортковими шарами можуть виступати додаткові згорткові шари або шари, що об'єднують з'єднаний шар є кінцевим шаром. З кожним шаром CNN складнішає, виявляючи більші частини зображення. Раніше шари розпізнавали прості функції, такі як кольори та краї. Коли зібрані дані зображення передаються по шарах CNN, вони з кожним кроком розпізнають більші елементи або форми об'єкта, поки об'єкт не буде повністю розпізнаний.

В основі CNN є те що більшість обчислень проводиться в згортковому шарі. Для цих необхідних наступних компонентів: карта об'єктів, фільтр і вхідні дані. Припустимо, що вхідний сигнал це кольорове зображення, з матриці пікселів у трьох вимірах. Виходить, що вихідні дані будуть мати: ширину, глибину та висоту, які відповідають вимірам RGB на зображенні.

Детектор ознак — це масив ваг у двох вимірах, який представляє собою всю картинку. на те, що їхні розміри можуть відрізнятися, розмір фільтра, як

правило, стає розміром 3 на 3. Цей розмір також визначається розміром рецептивного поля . Після області це зображення обробляється фільтром, а точний добуток додатково між вхідними пікселями та фільтром. Після цього точковий компонент подається у вихідний масив . Після цього фільтра зміщується , після повторення цієї процедури поки ядро не відкриває все зображення . Карта функцій, карта активації або згорнутий елемент — це терміни, які використовуються для опису остаточного результату з набору точкових продуктів із входу та фільтра.

Input Image	Filter	Output Array																																											
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>9</td><td>4</td><td>1</td><td>2</td><td>2</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>4</td></tr> <tr><td>1</td><td>2</td><td>1</td><td>0</td><td>6</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>2</td><td>8</td></tr> <tr><td>9</td><td>6</td><td>7</td><td>4</td><td>6</td></tr> </table>	9	4	1	2	2	1	1	1	0	4	1	2	1	0	6	1	0	0	2	8	9	6	7	4	6	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>2</td><td>1</td></tr> <tr><td>4</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> </table>	0	2	1	4	1	0	1	0	1	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>16</td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table>	16								
9	4	1	2	2																																									
1	1	1	0	4																																									
1	2	1	0	6																																									
1	0	0	2	8																																									
9	6	7	4	6																																									
0	2	1																																											
4	1	0																																											
1	0	1																																											
16																																													
	$\text{Output}[0][0] = (9*0) + (4*2) + (1*1) + (1*4) + (1*1) + (1*0) + (1*1) + (2*0) + (1*1)$ $= 0 + 8 + 1 + 4 + 1 + 0 + 1 + 0 + 1$ $= 16$																																												

Рисунок 1.4. - Приклад розкладень зображення

Карти об'єктів не можна підключати будь-які значення пікселя на вхідному зображенні. Потрібно тільки підключитися до рецептивного поля фільтра. Згорткові (та об'єднані) шари зазвичай називають «частково пов'язаними», останній вихідний масив не потребує звичайного зіставлення з кожним вхідним значенням. Тим не менше, цю функцію також можна назвати локальною підключеністю.

Зверніть увагу, що спільний доступ до параметрів — цей процес, за допомогою якого ваги в детекторі функції залишаються фіксованими під час переміщення по зображенню. Через процеси зворотного розмноження та градієнтного спуску значення ваги та інших параметрів коригуються під час тренувань. Тим не менш, перед початком навчання нейронної мережі необхідно встановити три гіперпараметри, які впливають на обсяг виходу. До них належать:

1. На глибині сигналу велика кількість фільтрів . Наприклад, три окремі фільтри створюють три різні функції карти , що створюють три глибини.

2. Stride - це кількість пікселів, яке ядро переміщує над вхідною матрицею. Більше кроків дають менший результат, хоча вони мають менше значення від двох або більше.

3. Коли фільтри не використовують вхідне зображення, традиційно використовують нульове заповнення. Це робить результат більшим або однаковим за розміром, встановлюючи на нуль усі елементи, що виходять за межі вхідної матриці.

Дійсне заповнення, також відомо , як відбувається заповнення, є одним із трьох типів прокладок. У цьому випадку, якщо розміри не співпадають, остання звина відпадає.

Те саме заповнення: це заповнення гарантує, що розміри вихідного шару та вхідного шару однакові.

Повне заповнення: Цей тип заповнення збільшує розмір виводу, додаючи нулі до межі вводу.

Як я вже згадував, перший шар згортки може бути наступним за іншим шаром. Припустимо, що ми намагаємося застосувати, чи зображення велосипеда. Ви можете уявити велосипед як набір різних деталей. Він складається з рами, керма, колісу, педалі та інших компонентів. В нейронній мережі кожна частина велосипеда представляє зображення нижнього рівня, а накладання їх створює кінцевий візерунок вищого рівня, що і означає ієрархію у CNN. Зменшення вибору, також відомо як об'єднання шарів, досягає зменшення розміру за рахунок зменшення кількості параметрів, які приймаються. Операція об'єднання фільтра проводиться по всьому вводу, як і операція згорткового шару, але цей фільтр не має ваги. Натомість ядро заповнює вихідний масив за допомогою функції агрегування до значення у рецептивному полі. Двома основними типами об'єднання є максимальне об'єднання. Коли фільтр рухається на вході, він вибирає піксель, який має найбільше значення, і відправляє його у

вихідний масив. Крім того, цей метод використовують частіше, ніж середнє об'єднання.

Середнє об'єднання: коли фільтр рухається по входу, він отримує середнє значення сприйнятливого поля та надсилає його у вихідний масив.

На те, що шар об'єднання містить велику кількість інформації, це також дає CNN багато переваг. Вони зменшують складність, ефективність і ризик переобладнання.

Назва повнозв'язаного шару відповідає назві. Як зазначалося раніше, у частково з'єднаних шарах значення пікселів вхідного зображення не пов'язане одночасно з вихідним шаром. Тим не менш, у повному підключеному шарі кожен вузол вихідного шару підключений одночасно до вузла попереднього шару.

Завдання класифікації виконується на основі ознак, виділених через попередні шари та їх різних фільтрів. У той час як згорткові та об'єднані шари обов'язково вибирають функції ReLu, шари FC визначають ймовірність від 0 до 1, використовуючи створену функцію активації softmax.

Висновок до розділу

У цьому розділі було проведено загальну характеристику об'єкту - використання нейронних мереж у сучасному житті з метою вирішення однієї з задач комп'ютерного зору, а саме визначення емоцій людини. Було визначено призначення обраного програмного продукту та описано основні функціональні вимоги, його загальні проблеми та методи їх вирішення.

Було проведено огляд поставленої задачі. Розглянуто основні поняття, сфери застосування нейромереж, зокрема використання у робототехніці, бізнесі, навчанню та системах охорони. Окрім цього, розглянуто принципи роботи та навчання штучної нейронної мережі і задачі які вона може виконувати. Для моєї задачі, найкраще підходить згорткова нейронна мережа, оскільки вона максимально сумісна з навчанням пов'язаних з зображенням та розпізнаванні образів в них.

РОЗДІЛ 2

РЕЗУЛЬТАТИ ПРОЕКТУВАННЯ (МОДЕЛЮВАННЯ) ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ

2.1 Прогнозування проблем, що можуть виникнути

Процес збору та інтерпретації інформації, визначення проблем і розкладання системи на її частини відомий як системний аналіз. Основною метою системного аналізу є вивчення системи або її компонентів з метою встановлення цілей системи, а також забезпечення ефективності роботи кожної частини відповідно до її призначення. Системний аналіз визначає, що система повинна робити, а також вивчає альтернативні рішення проблеми [4,9].

Після вивчення та аналізу предметної області ми повинні виділити низку проблем, які мають різний рівень значення для нашої системи. інакше кажучи, елементів системи розпізнавання образів. Ми повинні провести системний аналіз, об'єктом якого є проблеми нашого проекту, щоб визначити численні проблеми, які виникають. Таким чином, ми створюємо вимоги до проекту та створюємо дерева цілей і проблем, які будуть орієнтирами для вибору технологій і їхнього впровадження.

Згідно предметної області визначаємо наступні вимоги до нашої системи:

- Простота у використанні
- Якісна обробка відеопотоку
- Швидкодія системи
- Надійна апаратна частина
- Доступність та поширеність системи для користувачів

2.2 Побудова дерева проблем

Згідно поставлених задач ми можемо передбачити низку проблем, що очікують нас при проектуванні та реалізації системи.

Для того, щоб відобразити всі проблеми, потрібно побудувати дерево проблем, що буде охоплювати якомога більшу кількість поставлених задач, або ж, форс-мажорних ситуацій.

У наступному дереві проблем виділено 2 основні області проблем, рис. 2.1.:

- Алгоритмічно-технологічні
- Апаратні



Рисунок 2.1. - Дерево проблем

Надалі переглянемо усі проблеми більш детально.

Першою областю проблем було виділено алгоритмічно-технологічні, так як робота над даним продуктом починається з визначення технологічного плану проекту в якому буде чітко описано як ми підходимо до вирішення задачі.

Проблема алгоритмічно-технологічна №1: Розпізнавання обличчя

Різні алгоритми перевіряють риси брів, очей, носа, роту та інших рис обличчя. Такі фактори, як відстані між елементами рис, іноді роблять сканування складним. Наприклад, алгоритм може зіткнутися з проблемою з

розпізнаванням занадто розставлених очей або занадто піднятих брів, що призведе до неправильної ідентифікації.

- Розмір частин обличчя. Наприклад, очі та вузькі губи також можуть призвести до проблем з розпізнаванням.

- Тон шкіри. Через недостатню кількість прикладів для навчання деякі алгоритми можуть не розпізнати риси обличчя через колір шкіри людини. Щоб уникнути такої проблеми, необхідно мати широку базу даних з людьми різних рас.

Проблема алгоритмічно-технологічна №2: Реалізація розпізнавання емоцій людини

Більшість алгоритмів навчаються на базах даних, які містять пікові зображення емоцій людини, або найбільш виражені риси. 2.2. Це збільшує ймовірність неправильного розпізнавання при аналізі неповних емоцій.

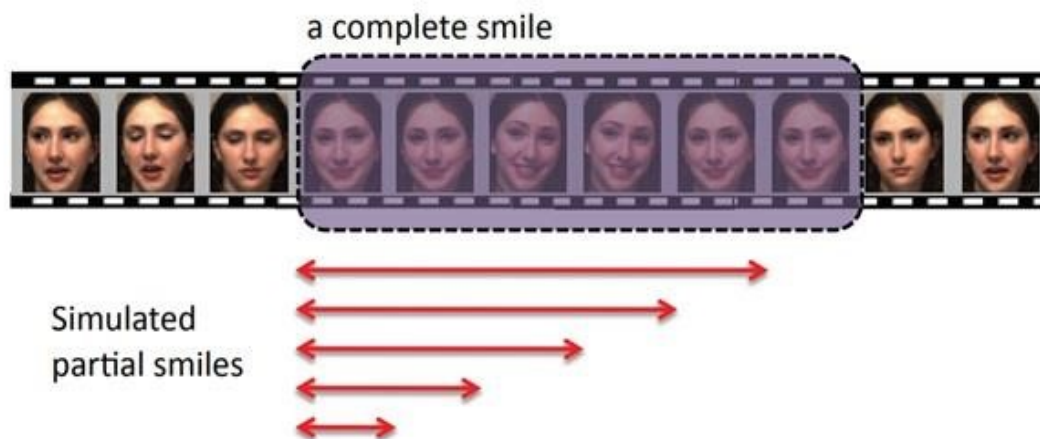


Рисунок 2.2. - Приклади етапів неповної посмішки

Покращення роботи нейронних мереж може вирішити ці проблеми. Завдяки цьому можна мінімізувати зміну, що відбувається при порівнянні пікової та неповної форм однієї та тієї ж емоції.

Проблема алгоритмічно-технологічна №3: Алгоритм системи навчання

Оскільки розпізнавання посмішки це завдання, яке потребує навчання системи, нам потрібно знайти ефективний та швидкий спосіб вивести системи на найвищий рівень продуктивності у співвідношенні до витрат часу на це

навчання. Саме тому ефективний алгоритм навчання системи може значно знизити час розробки системи і підвищити ефективність її роботи.

Проблема алгоритмічно-технологічна №4: Вибір мови програмування

В залежності від вибору мови програмування та апаратної платформи залежить реалізація нашої системи. Тому нам потрібно обрати мову, яка буде максимально незалежна від платформи та буде мати максимальну швидкодію.

Надалі варто перейти до апаратних проблем продукту, адже саме апаратна частина визначає наші обмеження та додає певні проблеми.

Апаратна проблема №1: Вибір обчислювального пристрою

В залежності від вибору обчислювального пристрою залежить те, як буде працювати наша система. Вирішення даної проблеми дозволить нам зробити компактну систему з максимальною швидкодією.

Апаратна проблема №2: Камера

В залежності від вибору камери залежить якість зображення, а від якості зображень залежить якість розпізнавання об'єктів на даних зображеннях. Тому вибір якісної та невеликої камери є важливим аспектом для успіху даного проекту.

Апаратна проблема №3: Відмовостійкість

В залежності від реалізації відмовостійкості буде залежати наскільки наш проект є автономним та захищеним від будь яких несприятливих подій зовнішнього середовища та користувацьких помилок. Тому реалізація відмовостійкості є ще одним важливим аспектом даного проекту.

2.3 Побудова дерева цілей

Після визначення всіх можливих потенційних проблем системи розпізнавання емоцій ми маємо можливість перейти до побудови дерева цілей системи.

Нам потрібно визначити максимальну можливу кількість цілей для реалізації, щоб продовжити системний аналіз. Також потрібно знайдені цілі

згрупувати за конкретними критеріями, аби не створювати пріоритизацію на кожну ціль.

Побудуємо дерево цілей(рис. 3.3.) нашої системи:

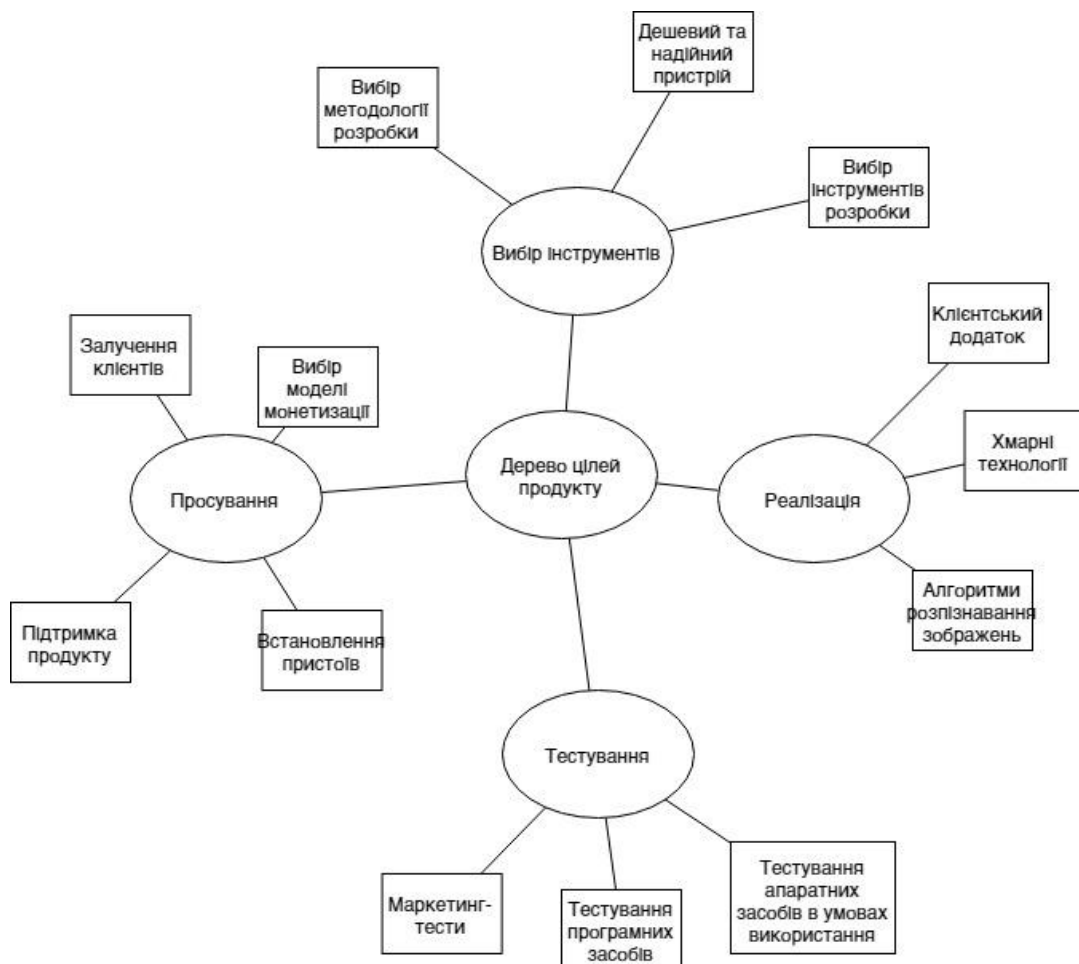


Рисунок 2.3. - Дерево цілей продукту

Дерево на рис.2.3. складається з чотирьох основних гілок, кожна з яких містить окремі категорії цілей. Вони розподілені на різні етапи, оскільки це спрощує виконання цих цілей у часовому розрізі та організовує більш послідовну роботу в процесі їх досягнення, а також дозволяє більш ефективно створити процес розробки.

Виділено наступні етапи:

- Вибір інструментів
- Реалізація
- Тестування

- Просування

Цілі вибору інструментів: Вибір методології розробки

Вибір методології розробки є одним з найбільш важливих етапів, оскільки він задає порядок роботи на протязі всього процесу розробки та підтримки нашого фінального продукту. Таким чином потрібно правильно ідентифікувати найбільш корисні для даного випадку методології

Цілі вибору інструментів: Вибір дешевого та надійного пристрою

Щоб проект був успішним, коли вибрані основні напрямки його розвитку, потрібно обрати найбільш підходящі технології та методи для досягнення високої швидкодії системи, в той же час з її низькою ціною.

Цілі вибору інструментів: Вибір програмних інструментів розробки.

Враховуючи велику кількість перспективних цілей, потрібно ідентифікувати найкращий набір інструментів для створення зручного та мінімального середовища розробки

Цілі етапу реалізації продукту: Розпізнавання зображень

Нам потрібно ідентифікувати оптимальну технологію для досягнення максимальної швидкодії програмної частини нашого додатку.

Цілі етапу реалізації продукту: Клієнтський додаток

Нам потрібно розробити єдиний і чіткий формат даних, для того щоб в майбутньому було просто створити клієнтські додатки та легко підключати їх як кінцеві точки виходу інформації.

Цілі етапу реалізації продукту: Хмарні технології

Зберігання та передача даних є дуже відповідальною та ресурсоємною частиною. Нам потрібно відштовхуватись від існуючих хмарних технологій та їх можливостей для створення відмовостійкого продукту який зможе похвалитись високою доступністю. Для того щоб перевірити відмовостійкість цієї системи нам знадобиться хаос тестування. Для цього і потрібен етап тестування.

Цілі етапу тестування: тестування програмних засобів

Тестування програмних засобів має відбуватись не тільки на стандартних відтренованих датасетах, а також на нових для перевірки їх коректної роботи.

Ще нам потрібно відтестувати програмні засоби та їх взаємодії у випадку їх міграції (у випадках поганої віртуалізації).

Цілі етапу тестування: тестування апаратних засобів

Тестування апаратних засобів, або в нашому випадку тестування хмари, дозволить перевірити та знайти нюанси які можуть викликати будь-які збої в програмному забезпеченні.

Цілі етапу тестування: маркетинг-тести

Так як ця частина комплексної роботи має під собою не тільки опис самого продукту, але і його висвітлення концептуальних деталей, то варто проаналізувати та поставити цілі щодо шляху, що проходить наша система включно до отримання її користувачем. Саме завдяки реалізації вірної маркетингової стратегії користувач дізнається про продукт та буде мати бажання використовувати його. Але специфіка сфери реклами та маркетингу має під собою величезний обсяг роботи щодо аналізу аудиторії та ніші ринку, перевірки декількох маркетингових стратегій, та постійному генеруванні нових ідей для залучення клієнтів. Відповідно маркетингові тести є обов'язковими, щоб зрозуміти, які важелі впливу варто використовувати на потенційних користувачів, та як саме припіднести сильні сторони продукту.

Цілі етапу просування: Встановлення пристроїв

Власне нас цікавить, як важко буде створити інфраструктуру, яка є важливою частиною роботи. Для того щоб ця система могла в подальшому адаптуватись бізнесом, нам потрібно отримати максимально простий процес створення системи з 0 на рівні хмари.

Цілі етапу просування: Підтримка продукту

Так як багато проблем та недоліків системи все одно будуть впливати і після виходу продукту на ринок, то варто підлаштовуватись під зростаючі потреби клієнтів, а також поступово усувати слабкі сторони системи. Також із звиканням клієнтів до продукту варто задуматись над тим, щоб постійно підтримувати цікавість до нього новими функціями та особливостями. Це може стосуватись як, в першу чергу, програмної частини, так і апаратної.

Цілі етапу просування: Монетизація.

Якщо ми знаємо основні портрети цільової аудиторії, а також їх основні та другорядні проблеми, відповідно ми маємо обрати максимально вигідну модель монетизації проекту. Також сьогодні існує багато додаткових джерел збільшення прибутку, такі як розміщення реклами і т.д. При швидкому зростанні масштабів, продукт сам перетвориться на авторитетний майданчик для залучення коштів зі сторони у вигляді інвестицій, або плати за розміщення чи співпрацю із продуктом.

2.4 Концептуальна модель

Концептуальна модель – це модель, яка представлена багатьма поняттями та зв'язками між ними. Ці зв'язки визначають смислову структуру певної, визначеної предметної області, чи конкретно визначеного об'єкта. Взаємопов'язані поняття в даній моделі, описують дану область чи об'єкт, враховуючи усі необхідні характеристики, класифікацію понять та закони протікання процесів.

Розглядають два основні підходи моделювання концептуальної моделі:

- Семантичні моделі
- Об'єктні моделі

Семантичні моделі найбільше уваги приділяють структурі даних, в той час як об'єктні моделі – поведінці об'єктів даних.

На рис. 2.4. зображена концептуальна модель розробки..

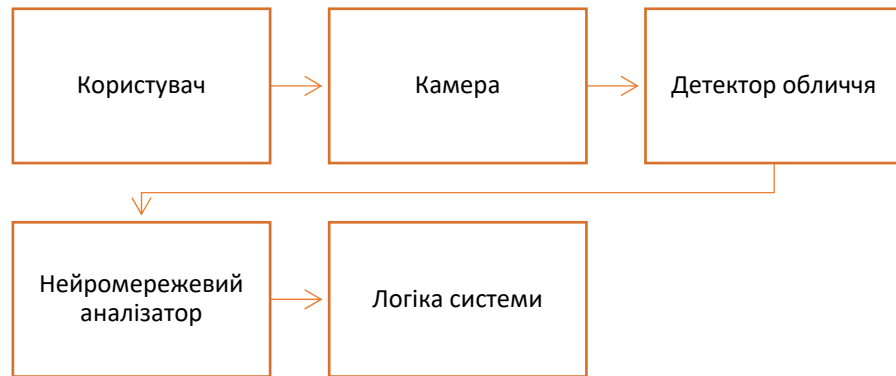


Рисунок 2.4. - Концептуальна модель

Спершу користувач використовуючи камеру знімає себе або навколишнє середовище. Камера передає на засіб збору та опрацювання даних зображення та визначає чи у кадрі є обличчя. Якщо детектор визначає обличчя, дані передаються на непромерений детектор, який визначає емоцію і надсилає у модуль логіки системи, а та у свою чергу виконує певну логіку.

2.5 Діаграма IDF0

Найбільш зручно використовувати мову IDEF0 для модифікації бізнес-процесів, де система представлена як сукупність взаємопов'язаних робіт або функцій. Такий тип орієнтації на функції є фундаментальним, оскільки функції системи досліджуються незалежно від об'єктів, з якими вона працює. Це дозволяє краще змоделювати взаємодію та логіку організації.

У IDEF0 проектування системи починається з контекстної діаграми, найбільш загального рівня опису системи. Ця діаграма містить визначення мети моделювання, суб'єкта моделювання та точки зору на модель.

2.5.1 Контекстна діаграма системи

Контекстна діаграма є вершиною деревоподібної структури діаграм і являє собою найбільш загальний опис системи і її взаємодії із зовнішнім середовищем.

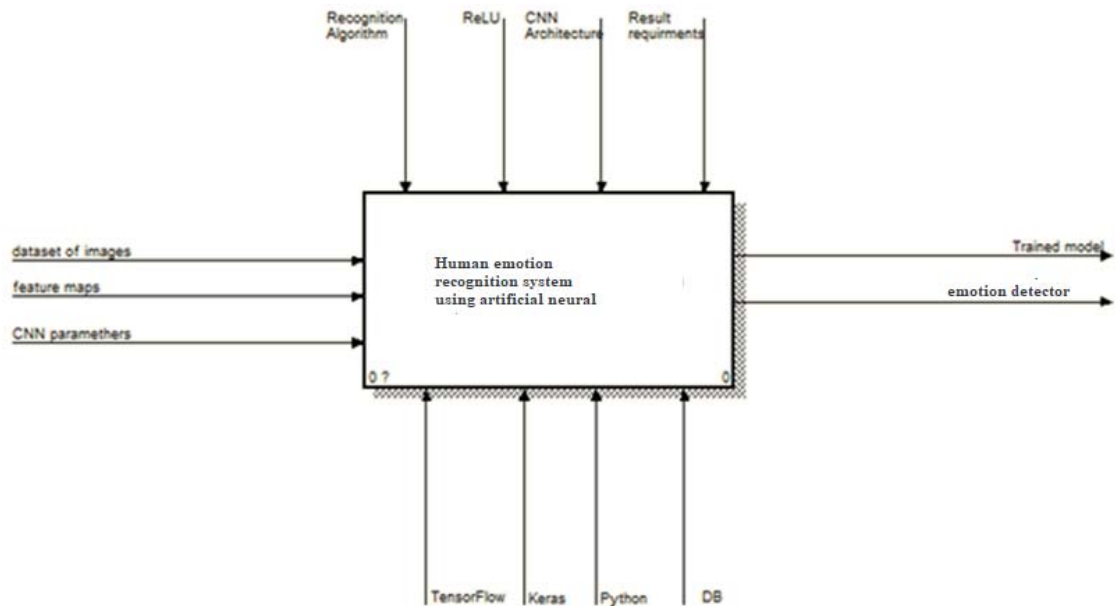


Рис. 2.5. Контекстна діаграма

Із рисунку 2.5 видно, що інформаційна система буде працювати за допомогою таких засобів як TensorFlow[4,11], Keras, Python та базою даних (DB). Дана інформаційна система буде працювати із набором визначений зображень (тестовий набір), картою особливостей та прийматиме на вхід параметри самої нейронної мережі. Керуватися дана ІС буде за допомогою певних алгоритмів розпізнавання зображень, математичними моделями, самою архітектурою нейронної мережі та вимогати до результату. Виходом даної інформаційної системи буде навчена модель, котра матиме змогу розпізнавати та класифікувати емоцію людини на фото чи на відео потоці, наприклад з веб камери.

2.5.2 Діаграма декомпозиції системи 1 рівня

Після опису системи в цілому проводиться розбивка її на великі фрагменти. Цей процес називається функціональною декомпозицією, а діаграми, які описують кожен фрагмент і взаємодію фрагментів, називаються діаграмами декомпозиції.

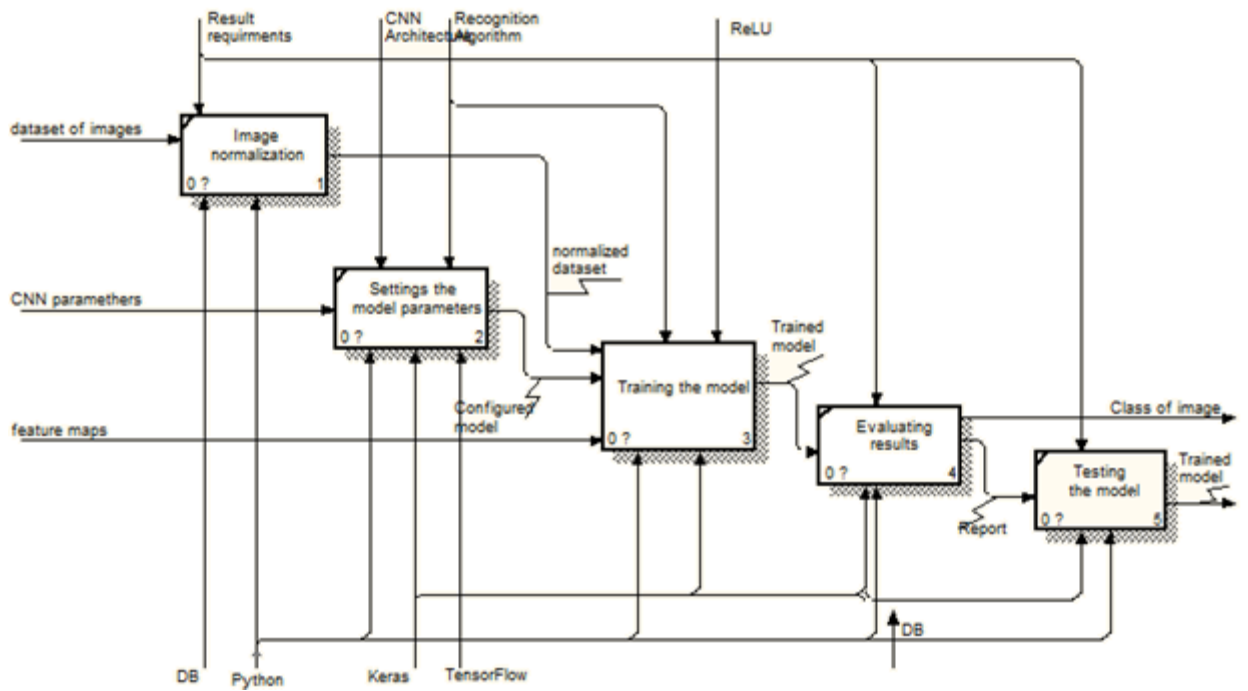


Рисунок 2.6. - Контекстна діаграма 1 рівня

На рисунку 2.6 видно, що для початку роботи необхідно нормалізувати зображення (привести їх до однакового виду), налаштувати параметри нейронної мережі. Після цього можна почати тренування мережі та оцінку результатів. Коли результати будуть задовільними, можна проводити кінцеве тестування моделі та почати використовувати її.

2.5.3 Діаграма декомпозиції системи 2 рівня

Після декомпозиції контекстної діаграми проводиться декомпозиція кожного великого фрагмента системи на більш дрібні і так далі, до досягнення потрібного рівня подробности опису (рис. 2.7).

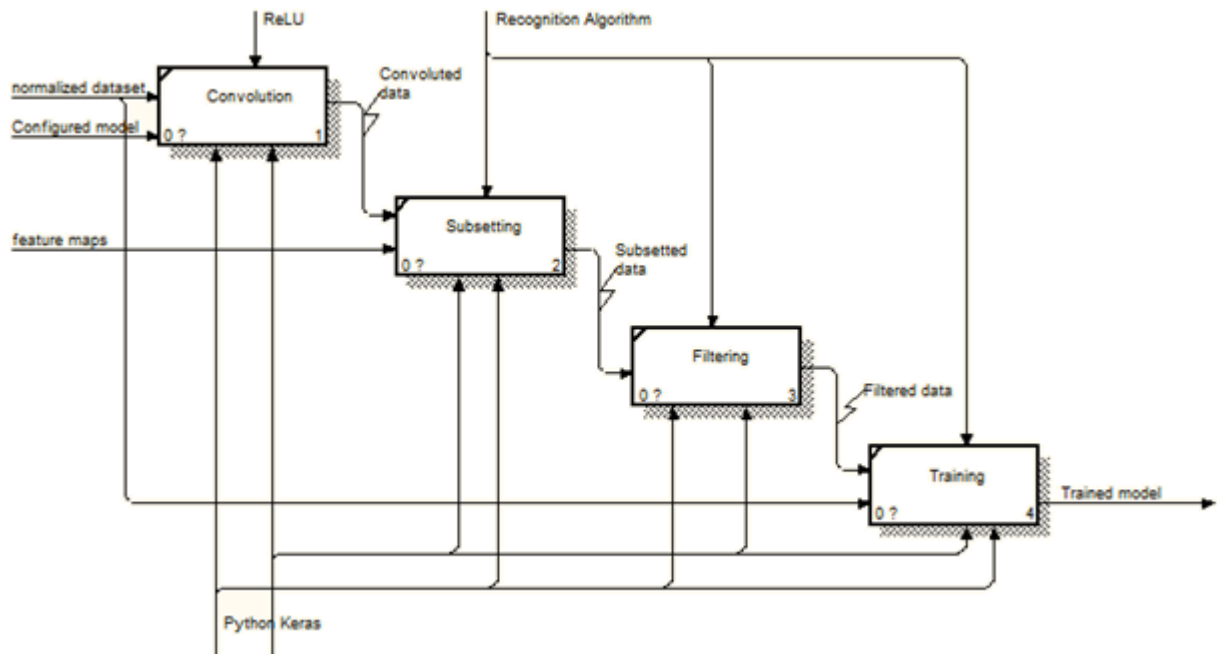


Рисунок 2.7. - Контекстна діаграма 2 рівня

Навчання моделі складатиметься із декількох етапів. Кожному етапу відповідає певний шар нейронів у кінцевій моделі нейромережі. Нормалізовані зображення поступають на шар згортки. Опісля за допомогою мапи особливостей відбудеться розбивання даних на підмножини із подальшою фільтрацією. Відфільтровані дані будуть виступати у ролі тренувального набору, за допомогою якого буде отримана натренована модель.

2.6 Принципи роботи згорткові нейронних мереж

2.6.1 Вхідні та вихідні дані

Згорткова нейронна мережа на вхід отримує масив пікселів. Залежно від розмірів зображення (його розширення, роздільної здатності), розмір масиву може бути розміром 640x480x3 (зображення розміром 640x480 пікселів, де кожен піксель містить 3 значення відповідно до компоненти RGB). Кожному із цих чисел буде присвоєне значення від 0 до 255, котре описує насиченість пікселя у даній точці. Ці числа являються безглуздими для людини, проте це єдиний спосіб передати комп'ютеру зображення. На основі такої тривимірної матриці, комп'ютер може вивести ймовірності класу емоції (наприклад 85% посмішка, 45% злість, 5% нетральна емоція і так далі).

Після того, як визначені вхідні та вихідні дані необхідно думати, як наблизитися до вирішення поставленої задачі. Нам необхідно, щоб комп'ютер мав змогу розпізнавати усі зображення та розпізнавати унікальні особливості для кожного класу зображень. Тобто такі особливості, котрі роблять посмішку посмішкою, а плач— сумом. У людей такі процеси відбуваються на рівні підсвідомості. Коли людина дивиться на зображення людини яка щиро сміється, вона може віднести її до радісної людини, якщо на зображенні видно зуби, оці змружені, щоки «роздуті», тощо. Аналогічно, комп'ютер зданий виконувати класифікацію зображень за допомогою пошуку особливостей базового рівня — меж та викривлень, а потім на основі абстрактних концепцій через спеціальні групи згорткових шарів.

2.6.2 Біологічні зв'язки

Згорткова нейронна мережа являється прототипом зорової кори мозку людини. Зорова кора має певні ділянки клітин, котрі реагують на певні ділянки поля зору. Ця ідея була розглянута за допомогою експерименту у 1962 році, в якому було показано, що окремі нервові клітини реагували лише при візуальному сприйнятті кордонів певної орієнтації. Тобто, деякі нейрони реагували у відповідь на вертикальну межу, деякі на горизонтальні межі або діагональні. Автори експерименту з'ясували, що ці нейрони зосереджені у вигляді сітки і разом формують візуальне сприйняття людини. Така ідея, як спеціалізовані компоненти, котрі вирішують лише певну категорію завдань — пошук специфічних характеристик і використовується у згорткових нейронних мережах.

2.6.3 Перший шар - математична частина

Перший шар в згорткових нейронних мережах являється згортковим. Якщо вхідне зображення - це матриця $640 \times 480 \times 3$ із значеннями компонент RGB у певній точці, то згортковий шар є „ліхтариком“, котрий світить на невелику частину зображення. Допустимо, що ліхтарик рухається по всіх областях вхідного зображення, при чому його світло має розмір лише 5×5 пікселів. У

контексті машинного навчання цей ліхтарик є фільтром (інколи його називають нейроном або ядром згортки), а ділянки котрі він освітлює називаються рецептивних полем (полем сприйняття).

Фільтр - це матриця (таку матрицю ще називають матрицею ваг або матрицею параметрів). Необхідно зазначити, що глибина (кількість компонент RGB) у фільтра повинна бути рівною глибині вхідного зображення. Тоді розміри фільтра будуть $5 \times 5 \times 3$. Результатом згортки такого фільтра буде поелементне множення значень фільтра на відповідний піксель у вхідному зображенні.

Усі добутки сумуються і в результаті виходить одне число. Це число характеризує значення фільтра у одній точці. Тепер необхідно повторити цей процес для кожної позиції у вхідному зображенні. Це досягається тим, що „ліхтарик“ зміщується на одиницю вправо і фільтр перераховується. Після закінчення рядка, фільтр опускається на 1 піксель вниз та знову проходиться по зображенню від одного краю до іншого. Кожна позиція вхідного зображення дає в результаті певне число (рис. 2.8).

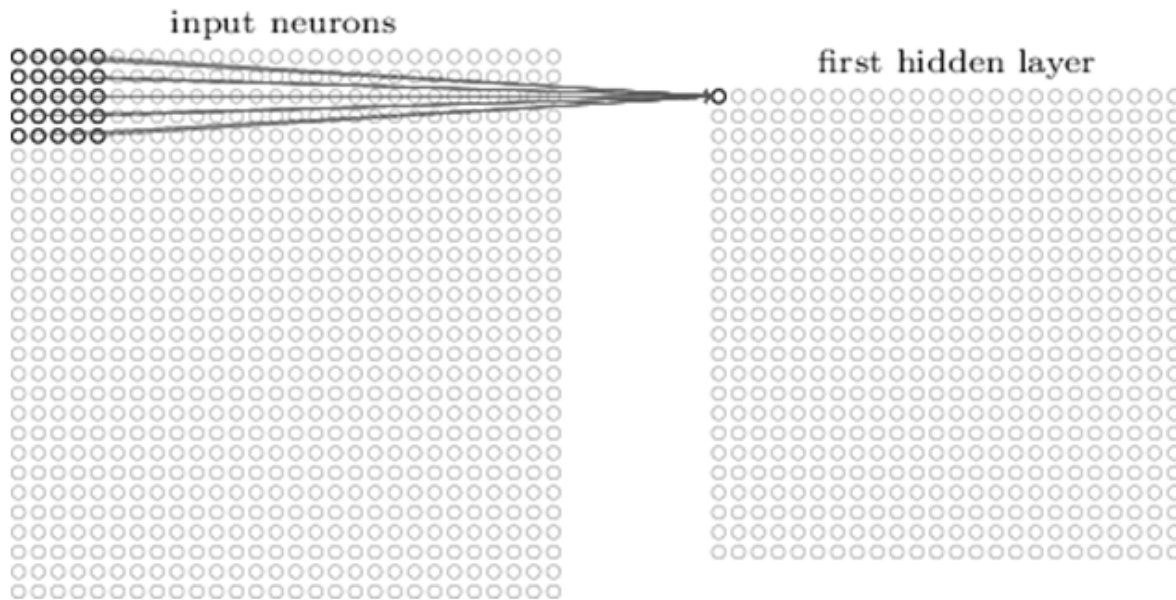


Рисунок 2.8. - Приклад згортки зображення у нейронних мережах

Після проходження фільтра по всіх позиціях виходить матриця, розміром $635 \times 475 \times 3$, котра називається функцією активації або картою ознак. Матриця $635 \times 475 \times 3$ виходить тому, що є лише певна кількість різних позиції, які можуть пройти через фільтр 5×5 зображення 640×480 .

Припустимо, використовується два 5 x 5 x 3 фільтра замість одного. Тоді вихідним значенням буде 635 x 475 x 2.

2.6.4 Наступні шари мережі

Станом на сьогоднішній день, у традиційній архітектурі ЗНМ використовуються й інші шари, які перемешуюються із згортковими. Класична архітектура ЗНМ буде виглядати так:

Input -> Conv -> ReLU -> Conv -> ReLU -> Pool -> ReLU -> Conv -> ReLU -> Pool -> Fully Connected

Рисунок 2.9. - Схема архітектури нейронної мережі

Не зважаючи на те, що останній шар знаходиться в кінці, він являється одним із найважливіших - ми перейдемо до нього пізніше. Підсумовуючи вищесказане, фільтри першого згорткового шару вміють виявляти властивості базового рівня, такі як межі і криві. Як можна собі уявити, щоб припустити який тип емоції зображений на зображенні, нам необхідна мережа, котра здатна розпізнавати властивості вищого рівня, наприклад очі, губи, ніс, вуха тощо.

Коли зображення проходить крізь один згортковий шар, вихід першого шару стає вхідним значенням 2-го шару. На першому рівні, вхідними були тільки дані оригінального зображення. Але уже на другому шарі, вхідними значеннями для нього є одна або кілька мапа властивостей - результат обробки зображення попереднім шаром. Кожен набір вхідних даних описує місця, де на оригінальному зображенні зустрічаються певні базові ознаки.

Якщо застосувати набір фільтрів поверх попереднього (пропускати картинку через другий згортковий шар), то на виході будуть фільтри властивостей вищого рівня. Типами цих властивостей можуть бути півкільця (комбінація прямого контуру із вигином) або квадрати (поєднання декількох прямих ребер). Чим більше згорткових шарів проходить зображення і чим глибше воно рухається в мережу, тим складніші характеристики виводяться в

мапах активації. В кінці мережі можуть бути фільтри, котрі реагують на рукописний текст, лапи тварин, тощо.

Якщо рухатися вглиб мережі, фільтри працюють з дедалі більшим полем сприйняття, а значить, вони взмозі обробляти інформацію з більшої площі початкового зображення (простими словами, вони краще пристосовуються до обробки більшої області піксельного простору).

Висновки до розділу

У даному розділі було описано проблеми, які можуть виникнути під час створення даної системи. Проаналізувавши проблему було побудовано дерево цілей. Окрім цього, здійснено аналіз та вибір методів для вирішення поставленої задачі і наведено концептуальну модель, яка показує концепт дії моєї задумки.

Було проведено аналіз функціонування системи. Було побудовано діаграму IDEF0, декомпозицію першого та другого рівня для загального розуміння рішення та описано принцип дії штучної нейронної мережі, а саме згорткового мережі на простих прикладах, щоб показати її простоту та ефективність у розпізнаванні образів

РОЗДІЛ 3

РЕЗУЛЬТАТИ ВИРІШЕННЯ ЗАДАЧІ

3.1 Загальна структура програмного проекту

Програмний проект складається із виконавчого файлу на мові програмування Python. Мова Python була вибрана тому, що вона проста в освоєнні, зручна для роботи із великими даними та досить популярна серед науковців, котрі працюють із штучними нейронними мережами. Більшість популярних бібліотек для роботи із ШНМ націлені саме на Python. Ще одною його перевагою є те, що ця мова є інтерпретованою, тобто нема етапу компіляції. Це дозволяє вносити зміни у вихідний код програми і миттєво отримати результати. Це допомагає експериментувати із вихідним кодом та не втрачати час на компіляцію усього проекту.

Також, проект включає в себе мінімальний набір графічного контенту, за допомогою котрого можна здійснити навчання штучної нейронної мережі. Він поділений на дві частини — train та validation. Для навчання використовується набір train, для перевірки результатів набір validation. Кожен із наборів містить 3 класи зображень, які ШНМ буде розрізняти.

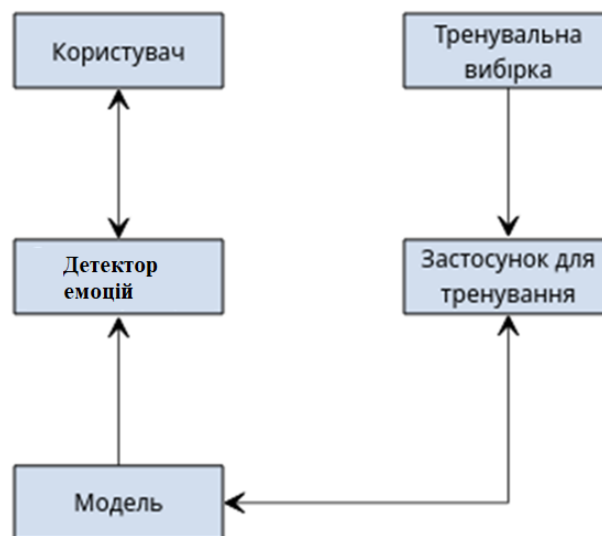


Рисунок 3.1. - Схема компонентів системи

Діаграма компонентів – це UML-діаграма, котра описує фізичне представлення системи. За її допомогою можна визначити архітектуру системи, котра розробляється та встановити залежності між компонентами.

Із рис. 3.1 видно, що у нас є такі компоненти як

- Компонент тренування моделі

Цей компонент відповідає за тренування моделі, валідацію та її збереження. Він взаємодіє із тренувальною вибіркою, оскільки на її основі відбувається тренування. Артефактом роботи даного компоненту є готова модель штучної нейронної мережі, котра готова до використання.

- Модель штучної нейронної мережі

Цей компонент являє собою набір певних моделей, котрі можуть класифікувати зображення за їхніми ознаками.

- Компонент класифікації

Із цим компонентом напряду взаємодіє користувач. За його допомогою завантажуються відповідні моделі, котрі вибрав користувач та проводиться безпосередня класифікація зображення.

3.2 Опис використаних сторонніх бібліотек та модулів

Для підвищення якості кінцевого програмного продукту було прийнято рішення використати готові напрацювання для роботи із штучними нейронними мережами. А саме такі бібліотеки як TensorFlow, Keras та NumPy.

TensorFlow — бібліотека для роботи із штучними нейронними мережами, котра розробляється компанією Google. Станом на сьогоднішній день є однією із найкращих бібліотек для роботи із ШНМ та аналізом даних. Ця бібліотека написана мовою C++, проте є підтримка таких мов як Java, C, та Python. Також, ця бібліотека є у вільному доступі та має відкритий код.

Це дозволяє розвивати бібліотеку разом із її користувачами, що позитивно впливає на зручність її користування. Також деякі досвідчені користувачі можуть вносити покращення, оптимізувати її, а також вводити нові методи та способи роботи із даними.

Keras — це обгортка над декількома бібліотеками для роботи із ШНМ. Перевагою Keras є те, що модель ШНМ описується абстракцією, котра не залежить від кінцевої бібліотеки, котра буде використана (бек-енд). Це дозволяє описати модель на високому рівні абстракції, швидко прототипувати моделі та вносити в них зміни.

Ця бібліотека була вибрана тому, що вона є простою у використанні. Не зважаючи на свою простоту, вона є досить потужним інструментом для роботи із штучними нейронними мережами.

Під час виконання програми буде використано відповідний бекенд для реалізації, створення та навчання моделі ШНМ. Опис моделі за допомогою абстракції дає можливість використовувати різні кінцеві бібліотеки та порівнювати їхню продуктивність.

Недоліком Keras є те, що абстрактний опис моделі може негативно вплинути на продуктивність ужитку. Keras не завжди може оптимально перетворити абстракцію у придатний для кінцевої бібліотеки код.

Проте це нівелюється тим, що кінцеві бібліотеки можна легко замінювати іншими. Станом на сьогодні, Keras підтримує такі кінцеві бібліотеки як TensorFlow та Theano.

NumPy — це бібліотека для мови програмування Python, котра дозволяє працювати із великими, багатовимірними масивами даних та матрицями. Також вона містить велику кількість високорівневих математичних алгоритмів та функцій для роботи із матрицями.

Використання NumPy у Python дає функціональні можливості схожі до MATLAB, проте відкриті та із відкритим вихідним кодом. Більше того, обидва продукти базуються на бібліотеці LAPACK, котра призначена для вирішення основних проблем лінійної алгебри.

3.3 Розробка та опис навчання нейронної мережі

Даний ужиток буде консольним, тобто графічного інтерфейсу не буде. Для успішного навчання нейронної мережі, нам потрібно буде приклади різних

емоцій, багато прикладів, у вигляді фотографій. Ми розмістимо їх на диску у відповідній директорії, `train_samples`, приклади фото на рис. 3.2.

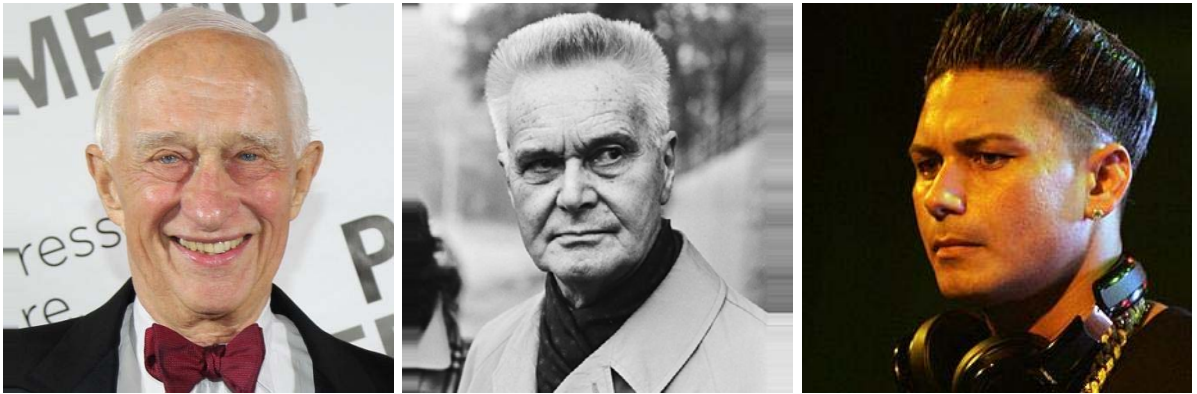


Рисунок 3.2. - приклади зображень для навчання нейронної мережі.

Зображення були взяті з Вікіпедії. В інтернеті є зібрані паки зображень для різних нейронної мереж, я взяв для навчання статі, відібрав по емоціях та заніс у відповідні директорії.

Наступним кроком, зображення потрібно промасштабувати до одного розміру, а саме 48x48. У такому випадку кількість входів на нейрону мережу повинно бути 6912. Але чи є така потреба. Колір зображення не сильно грає роль, тому можна зображення усереднити і зробити у відтінках сірого, як на рис.5.2. третє зображення.

Щоб кожен раз не брати зображення, декодувати його і не брати його пікселі, збережемо його у вигляді csv файлу, це в певній мірі таблиця у текстовому форматі, де поля розбити знаком «;» або якимсь іншим. В нашому файлі буде 2 поля, це емоція, і масив байтів зображення у градації сірого. Це пришвидшить роботу і час потрачений на навчання мережі зменшиться.

Перед початком навчання мережі, в коді потрібно було вказати, кількість входів, виходів, кількість епох (тобто ітерацій для навчання).

```
# parameters
batch_size = 32
num_epochs = 10000
input_shape = (48, 48, 1)
validation_split = .2
verbose = 1
num_classes = 7
```

```
patience = 50
base_path = 'models/'
```

Завантажимо дата сет для навчання мережі:

```
loading dataset
faces, emotions = load_dataset ()
faces = preprocess_input(faces)
num_samples, num_classes = emotions.shape
xtrain, xtest, ytrain, ytest = train_test_split(faces, emotions, test_size=0.2, shuffle=True)
model.fit_generator(data_generator.flow(xtrain, ytrain,
                                       batch_size),
                   steps_per_epoch=len(xtrain) / batch_size,
                   epochs=num_epochs, verbose=1, callbacks=callbacks,
                   validation_data=(xtest, ytest))
```

Функція `load_dataset()` зчитує з файлу всі дані, створить з них масиви та підв'яже до них асоціації у вигляді емоцій.

```
def load_dataset():
    data = pd.read_csv(dataset_path)
    pixels = data['pixels'].tolist()
    width, height = 48, 48
    faces = []
    for pixel_sequence in pixels:
        face = [int(pixel) for pixel in pixel_sequence.split(' ')]
        face = np.asarray(face).reshape(width, height)
        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']).as_matrix()
    return faces, emotions
```

Функція `fit_generator` запустить процес навчання. У свою чергу будуть викликані `callback` функції, які будуть визначати та перевіряти похибку навчання, щоб мережа не була перенавчена, логувати дії, для аналізу та зберігати результати епох, для наступних ітерацій навчання.

```

log_file_path = base_path + '_emotion_training.log'
csv_logger = CSVLogger(log_file_path, append=False)
early_stop = EarlyStopping('val_loss', patience=patience)
reduce_lr = ReduceLROnPlateau('val_loss', factor=0.1,
                               patience=int(patience / 4), verbose=1)
trained_models_path = base_path + '_mini_XCEPTION'
model_names = trained_models_path + '{epoch:02d}-{val_acc:.2f}.hdf5'
model_checkpoint = ModelCheckpoint(model_names, 'val_loss', verbose=1,
                                   save_best_only=True)
callbacks = [model_checkpoint, csv_logger, early_stop, reduce_lr]

```

3.4 Опис проблем і нестандартних ситуацій, які виникали під час розробки та заходів для їх вирішення

Під час розробки програмного продукту виникали певні проблеми. Першою проблемою, котра виникла — неможливість запуску ужитку із необхідними бібліотеками. Проблема полягала у тому, що у системі було встановлено дві версії інтерпретатора мови Python. Оскільки всі бібліотеки орієнтувалися на версію 2.6, а стандартною для системи являлася 3.7, програми із використанням даних бібліотек не запускалися. Вирішенням проблеми стала заміна стандартної версії інтерпретатора шляхом створення символічного посилання на коректну версію.

Ще однією проблемою стало те, що нейронна мережа показувала погані результати. Натренована модель мала досить високу точність у 85%, проте як показали результати тестування, нейронна мережа всі зображення розпізнавала як перший клас. Цей результат пояснюється тим, що значна частина вибірки складалася із зображень першого класу, і нейронна мережа не могла коректно розпізнати інші класи.

Після нормалізації вибірки (кількість зображень у навчальній вибірці у кожному класі стала приблизно однаковою) результати роботи покращилися, але все одно були незадовільними. Точність досягала 65%, проте модель уже могла розрізняти зображення та класифікувати їх.

3.5 Інструкції програмісту

3.5.1 Навчання мережі

Для навчання мережі потрібно перейти у кореневу директорію проекту та виконати команду :

```
$ python train_emotion_classifier.py
```

Запуститься процес навчання мережі, рис. 3.3.

```

*****
Total params: 58,423
Trainable params: 56,951
Non-trainable params: 1,472
-----
/Users/fedir-pylutskiy/Desktop/emotion-recognition/train_emotion_classifier.py:60: UserWarning: 'Model.fit_generator' is deprecated and will be removed in a future version. Please use 'model.fit_generator' instead.
  model.fit_generator(data_generator.flow(xtrain, ytrain,
Epoch 1/10
898/897 [*****] - ETA: 0s - loss: 1.7592 - accuracy: 0.3372
897/897 [*****] - 112s 123ms/step - loss: 1.7592 - accuracy: 0.3372 - val_loss: 1.6075 - val_accuracy: 0.4259 - lr: 0.0010
Epoch 2/10
224/897 [*****] - ETA: 1:19 - loss: 1.5475 - accuracy: 0.4274

```

Рисунок 3.3. - Процес навчання нейронної мережі.

Процес навчання є довготривалим. Для пришвидшення процесу навчання, можна застосувати потужності процесорів відеокарт (GPU), але на жаль у мене їх немає.

У Вибірці приймають участь 56951 зображення. Кількість епох було встановлено 64. Процес навчання зайняв майже 2 години. Результат 65%, рис. 3.4.

```

Epoch 52/64
898/897 [*****] - ETA: 0s - loss: 0.9405 - accuracy: 0.6532
897/897 [*****] - 114s 127ms/step - loss: 0.9405 - accuracy: 0.6532 - val_loss: 1.0345 - val_accuracy: 0.6156 - lr: 0.0010
Epoch 53/64
898/897 [*****] - ETA: 0s - loss: 0.9422 - accuracy: 0.6470
897/897 [*****] - 118s 122ms/step - loss: 0.9422 - accuracy: 0.6470 - val_loss: 1.0186 - val_accuracy: 0.6271 - lr: 0.0010
Epoch 54/64
898/897 [*****] - ETA: 0s - loss: 0.9349 - accuracy: 0.6511
897/897 [*****] - 109s 122ms/step - loss: 0.9349 - accuracy: 0.6511 - val_loss: 1.0604 - val_accuracy: 0.6077 - lr: 0.0010
Epoch 55/64
898/897 [*****] - ETA: 0s - loss: 0.9368 - accuracy: 0.6524
897/897 [*****] - 117s 130ms/step - loss: 0.9368 - accuracy: 0.6524 - val_loss: 1.0276 - val_accuracy: 0.6262 - lr: 0.0010
Epoch 56/64
898/897 [*****] - ETA: 0s - loss: 0.9338 - accuracy: 0.6530
897/897 [*****] - 128s 143ms/step - loss: 0.9338 - accuracy: 0.6530 - val_loss: 1.0489 - val_accuracy: 0.6144 - lr: 0.0010
Epoch 57/64
898/897 [*****] - ETA: 0s - loss: 0.9313 - accuracy: 0.6542
897/897 [*****] - 121s 135ms/step - loss: 0.9313 - accuracy: 0.6542 - val_loss: 1.0337 - val_accuracy: 0.6162 - lr: 0.0010
Epoch 58/64
898/897 [*****] - ETA: 0s - loss: 0.9262 - accuracy: 0.6575
897/897 [*****] - 130s 145ms/step - loss: 0.9262 - accuracy: 0.6575 - val_loss: 1.0838 - val_accuracy: 0.6039 - lr: 0.0010
Epoch 59/64
898/897 [*****] - ETA: 0s - loss: 0.9297 - accuracy: 0.6539
897/897 [*****] - 120s 133ms/step - loss: 0.9297 - accuracy: 0.6539 - val_loss: 1.0801 - val_accuracy: 0.6046 - lr: 0.0010
Epoch 60/64
898/897 [*****] - ETA: 0s - loss: 0.9201 - accuracy: 0.6593
897/897 [*****] - 113s 126ms/step - loss: 0.9201 - accuracy: 0.6593 - val_loss: 1.0201 - val_accuracy: 0.6283 - lr: 0.0010
Epoch 61/64
898/897 [*****] - ETA: 0s - loss: 0.9209 - accuracy: 0.6573
897/897 [*****] - 112s 125ms/step - loss: 0.9209 - accuracy: 0.6573 - val_loss: 1.0375 - val_accuracy: 0.6220 - lr: 0.0010
Epoch 62/64
898/897 [*****] - ETA: 0s - loss: 0.9179 - accuracy: 0.6604
897/897 [*****] - 111s 124ms/step - loss: 0.9179 - accuracy: 0.6604 - val_loss: 1.0459 - val_accuracy: 0.6195 - lr: 0.0010
Epoch 63/64
898/897 [*****] - ETA: 0s - loss: 0.9175 - accuracy: 0.6593
897/897 [*****] - 96s 107ms/step - loss: 0.9175 - accuracy: 0.6593 - val_loss: 1.0419 - val_accuracy: 0.6176 - lr: 0.0010
Epoch 64/64
898/897 [*****] - ETA: 0s - loss: 0.9211 - accuracy: 0.6572
897/897 [*****] - 95s 106ms/step - loss: 0.9211 - accuracy: 0.6572 - val_loss: 1.0332 - val_accuracy: 0.6239 - lr: 0.0010

```

Рисунок 3.4. - Закінчений процес навчання нейронної мережі.

На виході ми дістанемо файл (`_mini_XCEPTION.102-0.66.hdf5`), в якому спеціальним чином будуть вказані ваги всіх нейронів в системі. В подальшому для демонстрації системи ми будемо використовувати саме цей файл.

Для тестування системи детектування емоцій людини, необхідно запустити наступну команду:

```
$ python face_detect.py
```

У успішному випадку на екрані появиться вікно з детектором емоцій. Зверну увагу, що для успішної роботи ужитку потрібна веб-камера. На рис. 3.5. Демонстрація результатів.

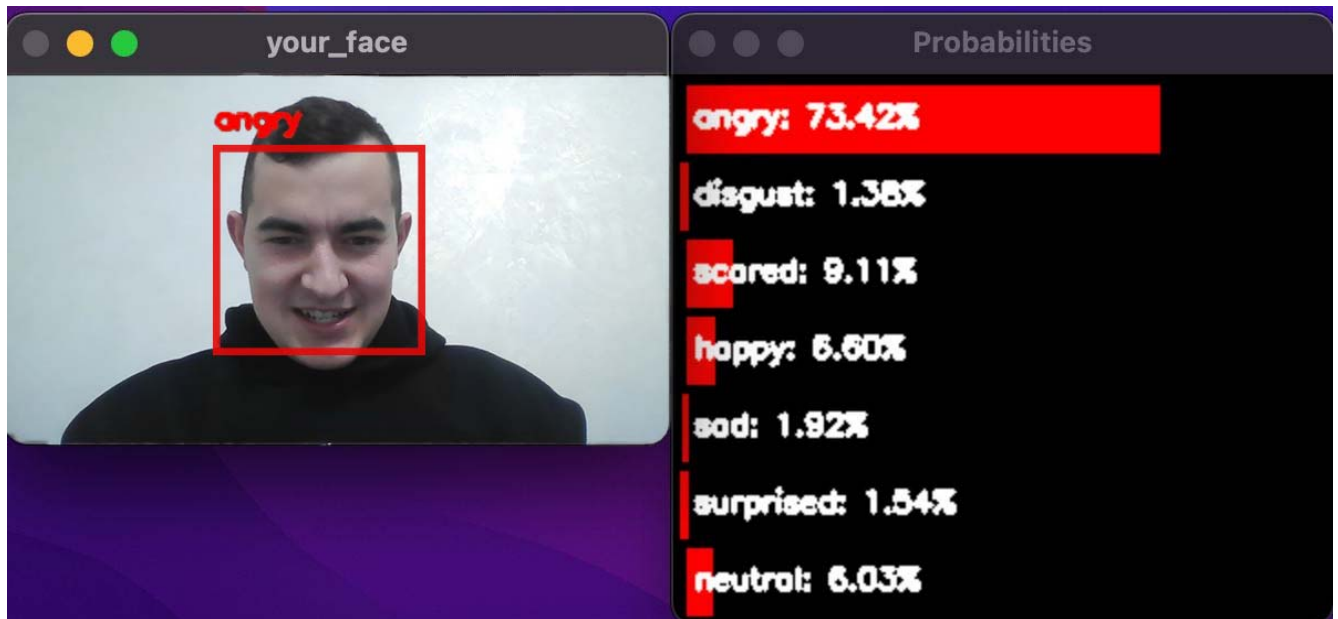


Рисунок 3.5. - детектування емоції «гнів».

Як бачимо система не ідеально визначає емоцію, тому гнів прийшлося відігравати майстерно.

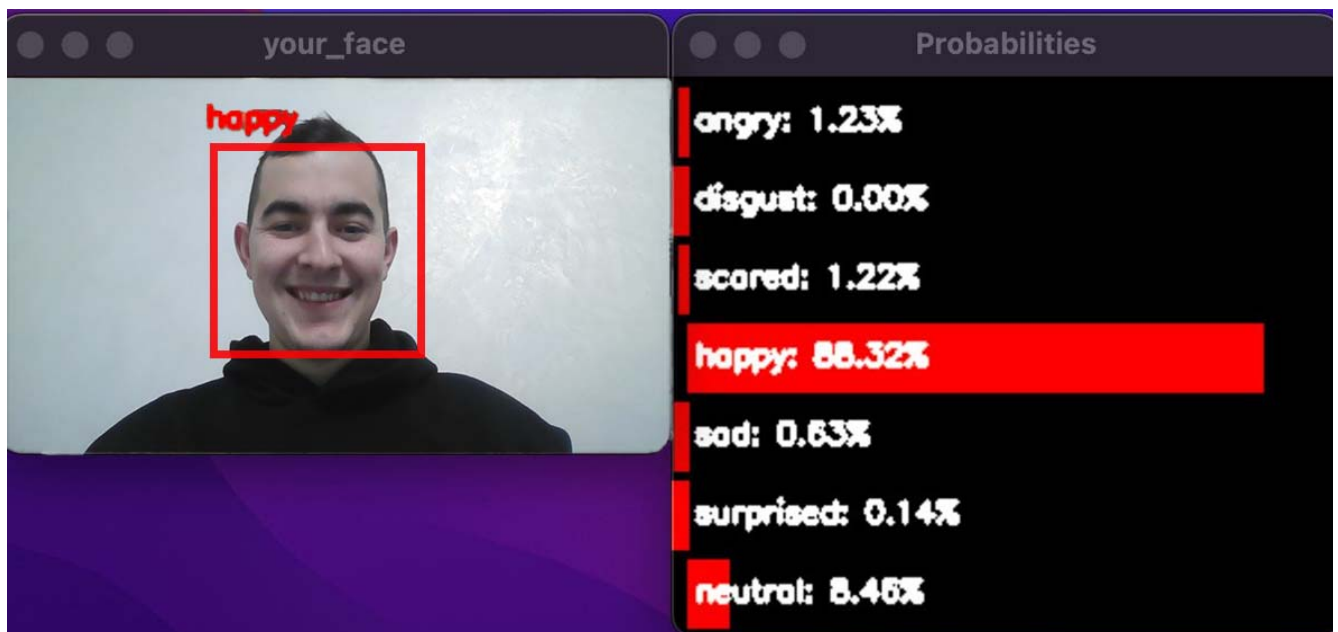


Рисунок 3.6. - детектування емоції «радість».

Така ж ситуація з іншими емоціями. Система поки що не може дати 100% гарантію. Ймовірніше дані на яких було здійснено навчання, дуже емоційно виражені, тому якщо не виражено посміхатись система буде вважати що у людини нейтральна емоція.

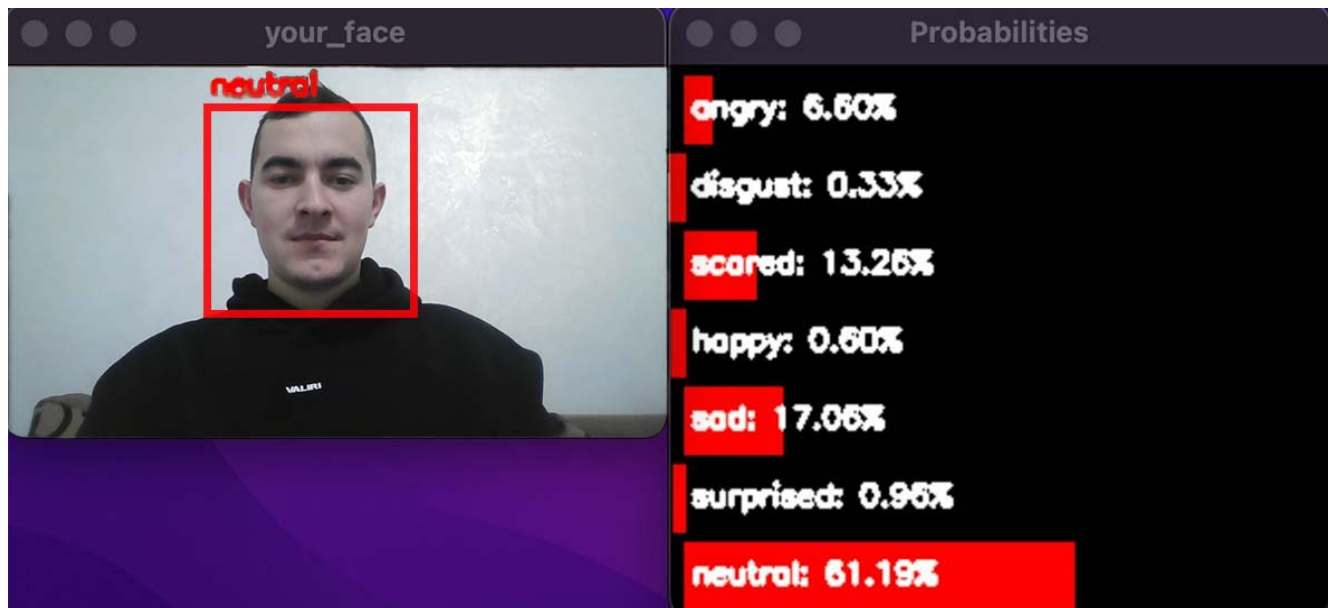


Рисунок 3.7. - детектування емоції «нетральна».

Так як вибірка, на якій ми навчали нейронну мережу зібрана з лиць зірок або великих людей, деякі емоції дуже «натягнуті».

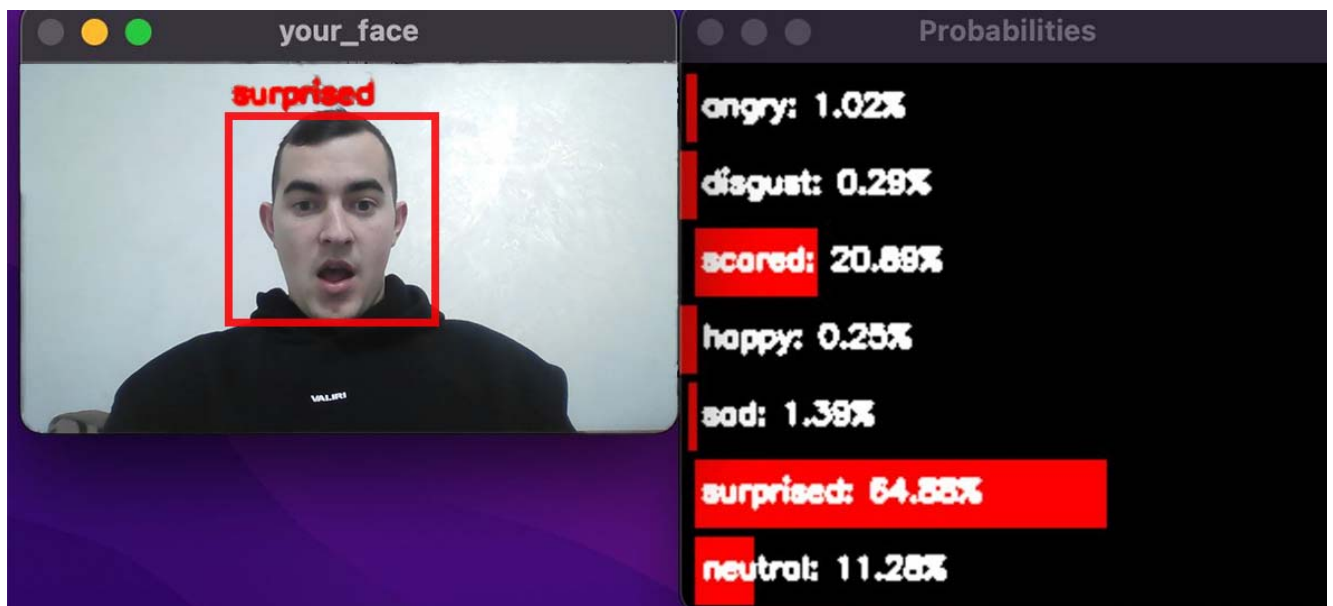


Рисунок 3.8. - детектування емоції «здивування».

Як бачимо нейронна мережа виконує поставлену задачу успішно. Деякі емоції даються дуже складно але якщо дуже постаратись то все реально.

3.5.2 Вимоги до апаратного забезпечення

Мінімальні апаратні характеристики:

- Центральний процесор – IntelCorei5 5 покоління і вище або AMD Ryzen 3 і вище.
- Об'єм оперативної пам'яті – не менше 4 Гігабайтів.
- Операційна система – ОС Linux, версії ядра 4.14 і вище, ОС Windows 7, 8, 10.
- Графічний процесор – опціонально, але його наявність прискорить виконання програм.
- Інтерпретатор мови Python 2.7

Перш ніж встановлювати бібліотеку Keras, необхідно встановити одну із кінцевих бібліотек для роботи із нейронними мережами, наприклад TensorFlow, Theano або CNTK Toolkit.

Також можна встановити опціональні залежності, а саме:

- cuDNN — рекомендовано для навчання нейронної мережі із застосуванням графічного прискорювача.
- HDF5 — для можливості збереження натренованих моделей на диск для подальшої її обробки та використання.
- GraphViz та PyDot - для візуалізації внутрішніх графів моделі.

Для успішного встановлення TensorFlow необхідно дотриматися наступних вимог.

Для Windows:

- x86 або x86_64 настільний комп'ютер або ноутбук.
- Версія Windows 7 і новіші.
- Для Linux:
- 64-бітний дистрибутив із ядром версії 3.14 та вище.
- Об'єм оперативної пам'яті не менше 2 Гб.

Для macOS:

- macOS 10.12.6 (Sierra) або новішої версії.

Якщо буде необхідна підтримка навчання нейронної мережі із застосуванням графічного прискорювача, необхідно встановити наступні залежності:

- CUDA Toolkit
- Драйвери nVIDIA із підтримкою обчислень CUDA
- cuDNN
- Графічний прискорювач із підтримкою технології обчислень CUDA

3.0 або вище.

3.5.3 Тестування

Важливе місце у розробці програмного продукту займає його тестування. Дуже часто цей процес ігнорується, тому як і розроблений продукт так і компанія можуть постраждати.

Тестування має багато переваг, і одне з найважливіших - економічність. Проведення тестів у вашому проекті може заощадити гроші в довгостроковій перспективі. Розробка програмного забезпечення передбачає багато кроків, і якщо помилки знайдені в попередніх кроках, їх виправити коштує набагато менше. Ось чому важливо провести тести якнайшвидше. Отримати тестувальників або контролерів якості, які мають технічну підготовку та досвід для програмного проекту - це як інвестиція, що принесе користь з точки зору бюджету.

По суті, тестування допомагає заощадити час і гроші в довгостроковій перспективі, оскільки проблеми вирішуються до появи великих проблем. Витрати на технічне обслуговування також нижчі, і, якщо продукт працює на 100% як слід, жодних експоненційних збитків для вашого бізнесу та продукту не буде з точки зору витрат.

Важливість тестування полягає і у тому, що воно гарантує, надійність продукту та задовольняє потреби користувача.

Важливо переконатися, що продукт не повинен спричиняти збоїв, оскільки це може коштувати дуже дорого у майбутньому або на пізніх стадіях розробки.

Правильне тестування забезпечує виявлення помилок та проблем на початку життєвого циклу продукту чи програми.

Якщо дефекти, пов'язані з вимогами або дизайном, виявляються пізно в життєвому циклі, виправити їх може бути дуже дорого, оскільки це може зажадати перепроєктування, повторної реалізації та повторної перевірки програми.

Користувачі не схильні використовувати програмне забезпечення з помилками. Вони можуть не прийняти програмне забезпечення, якщо вони не задоволені стабільністю програми.

Якщо продуктова організація або стартап має лише один продукт, низька якість програмного забезпечення може призвести до неприйняття продукту, і це може призвести до втрат, з яких бізнес може не відновитись.

Існує безліч типів тестування, нижче наведені одні з найпопулярніших:

Різні типи тестування програмного забезпечення

Нижче наведено перелік деяких поширених типів тестування програмного забезпечення:

До функціональних типів тестування належать:

- Тестування одиниць
- Інтеграційне тестування
- Тестування системи
- Тест на розум
- Тестування інтерфейсу
- Регресійне тестування
- Бета / тестування прийняття

Нефункціональні типи тестування включають:

- Тестування продуктивності
- Тестування навантаження
- Тест на стрес
- Тестування безпеки

Висновки до розділу

У даному розділі наведено загальну структуру програмного проекту. Було розроблено структурну схему, описано сторонні бібліотеки для реалізації ідеї, здійснено опис алгоритму згорткової мережі та описано проблеми які супроводжувались з розробкою ужитку для детектування емоцій людини.

Було описано детальну інструкцію користуванням, а саме поетапно як потрібно здійснити навчання нейронної мережі та як її застосовувати на практиці. Для успішного використання ужитку для визначення емоції людини, потрібно веб камеру і звісно комп'ютер з встановленим програмним забезпеченням.

Було визначено та вказано усі вимоги до апаратного забезпечення, для того щоб ефективно користуватися даним продуктом.

Також було визначено, що тестування є невід'ємною частиною розробки програмного продукту, адже воно дозволяє переконатися у коректній та надійній роботі системи. Було визначено основні та найпопулярніші типи тестування, що допоможуть швидко і чітко зрозуміти недоліки та несправності продукту.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Аналіз небезпечних і шкідливих виробничих чинників та розробка заходів щодо покращення умов праці

Потенційно шкідливим чинником кабінету особи, яка приймає управлінські рішення, вважається небезпека враження людини електричним струмом. Важливим, але менш ймовірним чинником являється пожежна небезпека під час аварійної ситуації. Хімічні та біологічні джерела практично не мають впливу.

Перелік небезпечних та шкідливих виробничих чинників наведено у таблиці 4.1.

Таблиця 4.1. Небезпечні та шкідливі виробничі чинники

Фізичні	Електронезбезпека, пожежа, шум, мікроклімат
Хімічні	Відсутні
Біологічні	Відсутні
Психофізіологічні	Відсутні

В приміщенні кабінету особи, яка приймає управлінські рішення, присутні небезпечні чинники, та за умов дотримання заходів безпеки, вони не є критичним.

4.2. Розробка логічно-імітаційної моделі процесу виникнення травм під час монтажу інтелектуальної інформаційної системи

Завжди можна знайти подію, з якої починається небезпечний процес ще до виникнення небезпечних наслідків, проаналізувавши кожен з логічних моделей процесів формування та потенційних травмонебезпечних та аварійних ситуацій.

У процесі оцінки рівня безпеки на робочих місцях, машинах, виробничих процесах чи окремих виробництвах необхідно знайти конкретний критерій рівня безпеки. Такий показник показує ймовірність аварії або травми залежно від явища.

Для оцінки рівня небезпеки цього об'єкта можна використовувати метод обчислення ймовірності виникнення будь-якого випадкового явища, який використовується в закордонній практиці. Його основним принципом є визначення виробничих небезпек, потенційних аварій і травм на основі обстеження робочої зони чи окремої машини. У процесі оцінки цих факторів визначаються події, які можуть бути вирішальними для створення логічно-імітаційної моделі травми. Наступним кроком є створення моделі, відомої як «дерева відмов і помилок оператора». Правильний вибір головної події має вирішальне значення.

Головну подію або як її ще називають «травма», модель якої нам необхідно створити, вибирають з оцінки вибраного об'єкта, виробництва чи окремого обладнання і змісту найбільш небезпечного явища, яке може виникнути за певних умов виробництва.

Розпочинаємо побудову моделі після вибору головного явища або події. Використовуючи оператори «і» та «або», ми можемо створити набір відомих ситуацій, які можуть призвести до події, вибраної як головна.

Після визначення відповідних травмонебезпечних ситуацій та їх кількості проводимо логічний аналіз із використанням операторів «і», «або» та інших, щоб визначити додаткові події, пов'язані з кожною з цих ситуацій. Поки не будуть знайдені всі основні події, які визначають межі моделі, процес побудови моделі не завершиться.

Слід мати на увазі, що кожна випадкова подія, до якої входять базові події, може формуватися та виникати за допомогою відповідних операторів, коли до неї входять два, три або більше базових подій.

Модель повинна бути математично оброблена, щоб визначити ймовірність кожної випадкової події, яка увійшла до неї, починаючи з базових і закінчуючи головною.

Ми використовуємо дані виробництва для визначення ймовірності основних подій. Основна подія, наприклад, «стан контролю з охорони праці».

Для визначення ймовірності ми повинні визначити, наскільки (у відсотках) від ідеального рівня здійснюється відповідний контроль на об'єкті. Ймовірність буде 0,5, якщо рівень контролю становить 50% або 30%. Якщо контролю немає, ймовірність «не здійснення контролю» становить 1, а якщо контроль ідеальний, то ймовірність дорівнює 0.

Після обчислення ймовірності всіх подій, розміщених у ромбах, і базових подій, починаючи з лівої нижньої гілки «дерева», ми позначаємо всі випадкові події, включені до моделі, номерами.

Таким чином, можна припустити, що певна модель готова до математичних обчислень ймовірностей випадкових подій у логічно-імітаційній моделі.

Таким чином, ми складемо список основних подій, необхідних для створення логіко-імітаційної моделі процесу, формування та виникнення аварій і травм під час створення інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств. Вони будуть основою для цієї моделі. Ми присвоюємо певне значення ймовірності виникнення кожному пункту списку. Нижче представлено повний список:

- | | |
|--|---------------|
| 1. Стан контролю з охорони праці | $P_1 = 0,2;$ |
| 2. Несерйозне відношення до проходження ТО інструменту | $P_2 = 0,1;$ |
| 3. Відсутність комплектуючих установки..... | $P_3 = 0,2;$ |
| 4. Невисока міцність | $P_4 = 0,03;$ |

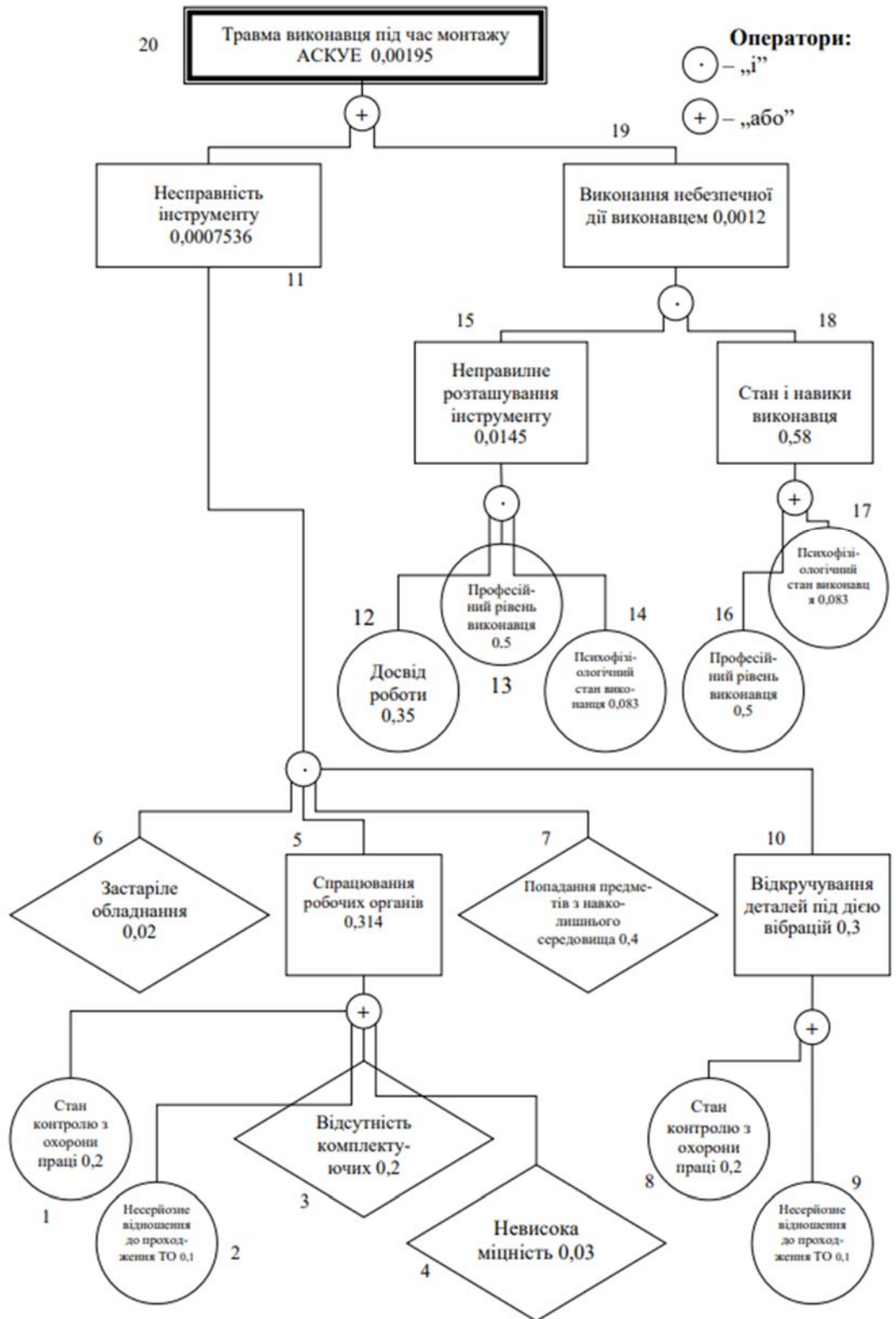


Рисунок 4.1. - Логіко-імітаційна модель процесу формування та виникнення аварії та травми під час монтажу системи розпізнавання емоцій людини

1. Використання застарілого обладнання..... $P_6 = 0,02$;
2. Попадання сторонніх предметів $P_7 = 0,4$;
3. Досвід роботи виконавця $P_{12} = 0,35$.
4. Професійний рівень виконавця $P_{13} = 0,5$;
5. Психофізіологічний стан виконавця..... $P_{14} = 0,083$;

На основі даного списку будуємо матрицю логічних взаємозв'язків між окремими пунктами, графічне представлення якої зображено на рис. 4.1.

Розрахуємо ймовірності виникнення подій, що входять у дану логіко-імітаційну модель процесу монтажу інтелектуальної інформаційної системи оцінювання платоспроможності сільськогосподарських підприємств (на прикладі ймовірності отримання травми виконавця).

Ймовірність виникнення події P_5 визначаємо наступним чином:

$$P_5 = 0,2 + 0,1 + 0,2 + 0,003 - 0,2 \cdot 0,1 - 0,2 \cdot 0,03 - 0,2 \cdot 0,03 - 0,1 \cdot 0,2 - 0,1 \cdot 0,03 - 0,2 \cdot 0,03 + 0,2 \cdot 0,1 \cdot 0,2 + 0,1 \cdot 0,2 \cdot 0,03 + 0,2 \cdot 0,1 \cdot 0,2 + 0,2 \cdot 0,1 \cdot 0,03 - 0,2 \cdot 0,1 \cdot 0,2 \cdot 0,03 = 0,314$$

Ймовірність виникнення події P_{10} визначаємо так:

$$P_{10} = 0,2 + 0,1 = 0,3.$$

Ймовірність виникнення події P_{11} визначаємо:

$$P_{11} = 0,02 \cdot 0,314 \cdot 0,4 \cdot 0,3 = 0,00075.$$

Ймовірність виникнення події P_{15} визначаємо наступним чином:

$$P_{15} = 0,35 \cdot 0,5 \cdot 0,083 = 0,0145.$$

Ймовірність події P_{18} :

$$P_{18} = 0,5 + 0,083 = 0,58.$$

Ймовірність події P_{19} :

$$P_{19} = 0,0145 \cdot 0,083 = 0,0012.$$

Зважаючи на те, що аварія можлива лише за умови встановлення автоматизованої системи управління енергоспоживанням людиною, ймовірність травми рівна ймовірності аварії.

Логіко-імітаційні моделі аварій і травм зменшують ймовірність аварій і травм. Якщо необхідно оцінити рівень небезпеки будь-якого робочого місця, слід ретельно вивчити та побудувати логічні моделі потенційних небезпечних ситуацій, які охоплюють стан обладнання та самого робочого місця, а також поведінку працівників, щоб обчислити ймовірність виникнення травми.

Після аналізу результатів моделювання ймовірність виникнення травми можна звести до дуже малої величини – достатньо зменшити вплив ймовірностей вихідних факторів, які до неї призводять.

4.3. Розробка заходів щодо безпеки у надзвичайних ситуаціях

Науково-технічний прогрес радикально змінив світ, породивши нові загрози для цивілізації. У житті сучасної людини все більше місце займають турботи, пов'язані з подоланням різних кризових явищ, що виникають в процесі розвитку земної цивілізації. В Україні, як і в усьому світі, в останні роки спостерігається зростання числа військових дій, катастроф природного та техногенного характеру. Це обумовлено, перш за все, прогресуючою урбанізацією територій, збільшенням щільності населення Землі, і, як наслідок, збільшенням антропогенного навантаження на навколишнє середовище. Захист природних систем і населення від надзвичайних ситуацій різного характеру сформувалася в останні роки як нагальна і об'єктивна потреба суспільства і держави.

Заходи щодо захисту цивільного населення плануються проводяться по населених пунктах де розміщені підприємства і охоплюють населення навколишніх сіл. Водночас характер та зміст захисних засобів встановлюються від ступеня загрози, місцевих умов з урахуванням важливості виробництва для безпеки населення і інших економічних і соціальних чинників.

Основні заходи щодо захисту населення плануються та здійснюються

завчасно і мають випереджувальний характер, це стосується насамперед підготовки, підтримання у постійній готовності індивідуальних та колективних засобів захисту, їх накопичення, а також підготовки до проведення евакуації населення із зон підвищеного ризику.

Також раз в три роки проводяться навчання по підготовці близьких до військових дій, що в разі небезпеки могло би не дістати людину зненацька. Керівництво докладає максимум зусиль, щоб працівники підприємств були хоча би мінімально захищені в разі будь-якої небезпеки пов'язаної з тими чи іншими обставинами.

РОЗДІЛ 5

ВИЗНАЧЕННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ.

5.1 Економічна характеристика проектного рішення

Метою бакалаврської кваліфікаційної роботи є проектування та розробка системи розпізнавання емоцій людини з використанням штучних нейронних мереж.

В першу чергу розробка буде цінна для наукових досліджень, пов'язаних з обробкою та аналізом даних для напряму комп'ютерного зору. З іншої сторони, буде корисна, як аналізатор емоцій на співбесідах, що допоможе скласти психологічну оцінку людини, оскільки емоційний стан людини напряму пов'язаний з її працездатністю і умінням працювати у колективі. Окрім цього, подібні системи можна вбудовувати в інтелектуальні системи автомобіля, щоб визначати чи особа за кермом в стані вести автомобіль.

5.2 Інформаційне забезпечення та формування гіпотези щодо потреби розроблення товару

В останні роки розпізнавання емоцій стало особливо актуальним, коли з'явилися і розвинулися мобільні пристрої та комп'ютери, вчені та розробники дійшли згоди, що прояв людських емоцій – один із головних напрямків для більш гнучкого розвитку систем штучного інтелекту. Такі системи використовуються в багатьох сферах нашого життя, вони дозволяють підвищувати рівень безпеки, наприклад: в сучасних автомобілях існують системи розпізнавання людської втоми або сонливості, при виявленні яких, водію приходять сигнали, які не дають йому заснути.

В умовах швидкого зростання використання інтелектуальних технологій в суспільстві та розвитку цієї галузі, потреба в технологіях, здатних оцінити потреби потенційного клієнта і вибрати найбільш оптимальне для них рішення, різко зростає. Автоматична оцінка емоцій особливо важлива в таких областях, як робототехніка, маркетинг, освіта і індустрія розваг.

Застосування таких систем використовується для досягнення різних цілей:

- в робототехніці для розробки інтелектуальних спільних або сервісних роботів, які можуть взаємодіяти з людьми;
- в маркетингу: для створення спеціалізованих рекламних оголошень, заснованих на емоційному стані потенційного клієнта;
- в освіті: використовується для поліпшення процесів навчання, передачі знань і методологій сприйняття;
- а також в сфері охорони/безпеки людей.

5.3 Бюджетування

Бюджетування є комплексно обґрунтованою системою розрахунку витрат, пов'язаних з виготовленням та реалізацією продукту, яка дає можливість здійснити аналіз витрат та розробити заходи щодо підвищення рентабельності виробництва. На даному етапі необхідно визначити собівартість продукту, який розробляється та економічно обґрунтувати доцільність вибору однієї із стратегій. Бюджет витрат на матеріали та комплектуючі наведено в таблиці 5.1.

Таблиця 5.1 - Бюджет витрат на матеріали та комплектуючі виробів

Назва матеріалів та комплектуючих	Марка, тип, модель	Фактична кількість, шт.	Ціна за одиницю, грн.	Разом, грн.
Оренда компютера	cpu i7, 10GBRam, nvidia rtx 3050, ssd -128	1	7000	7000
Папір	A4 80 г/м Maestro Standart	1	150	150
Флешка	Transcend 600GB	2	1700	3400
Разом:	-	4	8850	10550

Для розробки проекту також потрібні людські ресурси. Проте, щоб реалізувати цей проект не потрібно наймати велику команду. Для реалізації

проекту стануть в нагоді архітектор системи, програмісти та тестери. В таблиці 5.2 наведено бюджет витрат на оплату праці.

Таблиця 5.2 - Бюджет витрат на оплату праці

Посада, спеціальність	Кількіст ь працівників, осіб	Ча с роботи, дні	Денна заробітна плата працівників, грн.	Сума витрат на оплату праці, грн.
<i>Основна заробітна плата</i>				
Архітектор	1	3	2100	6 300
Програміст	2	21	2200	92 400
Тестер	2	7	2100	29 400
Разом:	-	-	-	128 100
<i>Додаткова заробітна плата</i>				
Архітектор	0	0	0	0
Програміст	0	0	0	0
Тестер	0	0	0	0
Разом:	-	-	-	0

У таблиці 5.3 представлено бюджет загальновиробничих витрат.

Таблиця 5.3 - бюджет загальновиробничих витрат

Статті витрат	Сума, грн.
<i>Змінні загальновиробничі витрати, у т.ч.:</i>	
- заробітна плата допоміжного персоналу;	9780
- витрати на МШП;	670
- витрати на електроенергію та технологічні цілі;	5000
- витрати на ремонт;	-
- інші змінні витрати;	-
Разом змінних витрат:	15450

<i>Постійні загальновиробничі витрати, у т.ч.:</i>	
- заробітна плата допоміжного персоналу;	-
- комунальні послуги;	2000
- витрати на оренду;	4580
- витрати на ремонт;	700
- інші постійні витрати;	-
Разом постійних витрат:	7280
<i>Разом загальновиробничих витрат:</i>	22730

До адміністративних витрат відносяться загальногосподарські витрати, спрямовані на обслуговування та управління підприємством. Витрати на збут включають витрати, пов'язані з реалізацією (збутом) продукції (товарів, робіт, послуг). Бюджет адміністративних витрат та витрат на збут наведено в таблиці 5.4.

Витрати на виробництво продукції у вартісному виразі формують її виробничу собівартість. Цей показник є одним з найважливіших економічних показників господарської діяльності підприємства, у якому дістають відображення зростання продуктивності праці, економія ресурсів.

Таблиця 5.4 - Бюджет адміністративних витрат та витрат на збут

Статті витрат	Сума, грн.
1	2
<i>Адміністративні витрати, у т.ч.:</i>	
- заробітна плата адміністративного персоналу;	7000
- витрати на МШП;	600
- витрати на відрядження;	-

- витрати на ремонт;	-
- витрати на паливно-мастильні матеріали;	-
- витрати на сплату податків і зборів;	1540
- знос адміністративного обладнання;	700
Разом адміністративних витрат:	9840
<i>Витрати на збут, у т.ч.:</i>	
- заробітна плата менеджерів зі збуту;	16000
- витрати на гарантійний ремонт;	2300
- витрати на відрядження;	-
- витрати на гарантійне обслуговування;	6100
- витрати на налагодження і експлуатацію;	3400
- витрати на паливо-мастильні матеріали;	5000
- витрати на рекламу;	7500
Разом витрат на збут:	40300

Для того, щоб можна було зручно переглянути та проаналізувати усі статті витрат, варто скласти таблицю зведеного кошторису витрат на розробку. Зведені витрати представлено в таблиці 5.5.

Таблиця 5.5 - Зведений кошторис витрат на розробку проектного рішення

Статті витрат	Одиниці виміру	Фактична кількість, шт.	Ціна одиниці, грн.	Разом, грн.
Сировина і матеріали		4	263 7,5	10550
Купівельні напівфабрикати та комплектуючі вироби				-
Зворотні відходи (вираховуються)				-
Паливо та електроенергія на технологічні цілі				-
Основна заробітна плата				128 100
Додаткова заробітна плата				-
Відрахування на соціальне страхування				28182
Витрати на утримання й експлуатацію устаткування				-
Загальновиробничі витрати, у т.ч.:				-
- змінні;				15450

- постійні;				7280
<i>Разом виробничих витрат:</i>				22730
Адміністративні витрати				9840
Витрати на збут				40300
Інші операційні витрати				-
<i>Разом виробничих і операційних витрат:</i>				72870

Для визначення фінансових результатів, необхідно розрахувати вартість (ціну) продукту (проектного рішення), який розробляється. Ціна визначається на основі суми виробничих і операційних витрат з врахуванням рентабельності виробництва 76%.

$$Ц = СБ * Р$$

де Ц – ціна одинці продукту, грн.

СБ – собівартість продукту, грн.

Р – рентабельність виробництва, %

$$Ц = 72870 * 0,76 = 55\,381,20 \text{ грн}$$

У таблиці 5.6 наведено бюджет фінансових результатів.

Таблиця 5.6 -Бюджет фінансових результатів

Показники	Сума, грн.
1	2
Дохід від реалізації продукції (обсяг 8 шт)	443049,60

Податок на додану вартість (20%)	88609,92
Чистий дохід від реалізації продукції	354439,68
Собівартість реалізованої	72870,00
Валовий прибуток	281569,68
Операційні витрати:	
- адміністративні витрати:	9840,00
- витрати на збут;	40300,00
- інші операційні витрати;	-
Фінансовий результат від операційної діяльності	231429,68
Податок на прибуток (18%)	41657,34
Чистий прибуток (збиток)	189772,34

Висновки до розділу

Отже, під час дослідження було розраховано економічні складові проектування та розробки системи розпізнавання емоцій людини з використанням штучних нейронних мереж.

На основі аналізу ринку оцінено фактори внутрішнього та зовнішнього середовища, а також їх вплив на формування потреби у програмному рішенні. За результатами експертної оцінки можна зробити висновок, що серед факторів зовнішнього середовища позитивний вплив мають високий рівень науково-технічного прогресу та споживачі (сформована потреба у продукті).

На основі аналізу стратегічних альтернатив було обрано стратегію розвитку існуючого продукту та стратегію розвитку ринку. Оскільки на ринку

вже існують подібні системи та дану систему можна буде застосовувати у нових сферах діяльності.

Проведено аналіз витрат, пов'язаних з реалізацією проекту і з'ясовано, що за рівня рентабельності 76% ціна на розроблюване ПЗ становитиме 55 381,20грн.

Величина чистого прибутку 189772,34 грн свідчить про доцільність розробки такого продукту.

ВИСНОВКИ І ПРОПОЗИЦІЇ

Під час розробки даної кваліфікаційної роботи я здобув усі необхідні навички, що потрібні для проектування та розробки системи розпізнавання емоцій людини з використанням штучних нейронних мереж.

Було проведено загальну характеристику об'єкту - використання нейронних мереж у сучасному житті з метою вирішення однієї з задач комп'ютерного зору, а саме визначення емоцій людини. Визначено призначення обраного програмного продукту та описано основні функціональні вимоги, його загальні проблеми та методи їх вирішення.

А також проведено огляд поставленої задачі. Розглянуто основні поняття, сфери застосування нейромереж, зокрема використання у робототехніці, бізнесі, навчанню та системах охорони. Окрім цього, розглянуто принципи роботи та навчання штучної нейронної мережі і задачі які вона може виконувати. Побудовано дерево проблема, та дерево цілей, обрано мову та технології для вирішення задумки.

Побудовано функціональні діаграми, IDEF0 та детально розглянуто принцип роботи згорткової мережі на простих прикладах, щоб показати її простоту та ефективність у розпізнаванні образів.

Розроблено загальну структуру програмного проекту. Було розроблено структурну схему, описано сторонні бібліотеки для реалізації ідеї, здійснено опис алгоритму згорткової мережі та описано проблеми які супроводжувались з розробкою ужитку для детектування емоцій людини.

Було описано детальну інструкцію користуванням, а саме поетапно як потрібно здійснити навчання нейронної мережі та як її застосовувати на практиці. Для успішного використання ужитку для визначення емоції людини, потрібно веб камеру і звісно комп'ютер з встановленим програмним забезпеченням.

Було визначено та вказано усі вимоги до апаратного забезпечення, для того щоб ефективно користуватися даним продуктом.

Також було визначено, що тестування є невід'ємною частиною розробки програмного продукту, адже воно дозволяє переконатися у коректній та надійній роботі системи. Було визначено основні та найпопулярніші типи тестування, що допоможуть швидко і чітко зрозуміти недоліки та несправності продукту.

В результаті виконання даної кваліфікаційної роботи було розроблено систему розпізнавання емоцій людини з використанням штучних нейронних мереж.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Катренко А. В. Системний аналіз об'єктів та процесів комп'ютеризації. Учб. посіб. / А. В. Катренко. – Львів, 2018.
2. Губанов В.А. і др. Введення в системний аналіз: Навчальний посібник /Под ред. Л.А. Петросяна. - Л.: вид-во ЛГУ.
3. Rezaul Karim. TensorFlow: Powerful Predictive Analytics with TensorFlow / Md. Rezaul Karim. – 2018р.
4. Python Cookbook, 2019. – 667 с. – (3). – (ISBN: 978-1-443-34037-7).
5. Секіне С. Названі сутності: розпізнавання, класифікація та використання / С. Секіне, Е. Ранчход // Видавництво Джона Бенджамінса – 2019.
6. Wenjie Luo, Alexander G Schwing, and Raquel Urtasun, Efficient deep learning for stereo matching, 2016 – 248 с.
7. Ravi Garg, Vijay Kumar B. G, and Ian D. Reid, Unsupervised CNN for single view depth estimation: Geometry to the rescue, 2016 – 173 с.
8. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A. and Bry, A., End-to-End Learning of Geometry and Context for Deep Stereo Regression, 2017 – 238 с
9. Pinard, C. and Chevalley, L. and Manzanera, A. and Filliat, D., End-toend depth from motion with stabilized monocular videos, 2017 – 74 с.
10. Nazar B. LaMetric — Information At a Glance / Bilous Nazar. – 2014. – Режим доступу до ресурсу: <https://lametric.com/about>.
11. Python eats away at R: Top Software for Analytics, Data Science, Machine Learning in 2018: Trends and Analysis // KDnuggets. – 2018. – Режим доступу до ресурсу: <https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html>
12. TensorFlow: Large-scale machine learning on heterogeneous systems // TensorFlow.org. – 2015. – Режим доступу до ресурсу: <http://download.tensorflow.org/paper/whitepaper2015.pdf>

13. TensorFlow, Google's Open Source AI, Points to a Fast-Changing Hardware World / Wired. – 2017. – Режим доступу до ресурсу: <https://www.wired.com/2015/11/googles-open-source-ai-tensorflow-signals-fast-changing-hardware-world/>
14. Michael D. Announcing Michael Droettboom as the lead matplotlib developer / Droettboom Michael // matplotlib.org. – 2018. – Режим доступу до ресурсу: <http://matplotlib.1069221.n5.nabble.com/ANN-Michael-Droettboom-matplotlib-lead-developer-td5037.html>.
15. Вашук В. Розумний дім нарешті може стати вашим домом / Віталій Вашук. – 2017. – Режим доступу до ресурсу: https://www.globallogic.com/ua/gl_news/the-smart-home-finally-comes-home/.
16. Machine Learning Crash Course – Режим доступу до ресурсу: <https://developers.google.com/machine-learning/crash-course/>.
17. Python IDE for Professional Developers. – 2016. – Режим доступу до ресурсу: <https://www.jetbrains.com/pycharm/>.
18. Моделі нейронних мереж. – Режим доступу: <https://studme.com.ua/1246122010028/neural/models.htm>
19. Gibbs S. Google launches video doorbell with facial recognition in UK / Samuel Gibbs // the Guardian. – 2018. – Режим доступу до ресурсу: <https://www.theguardian.com/technology/2018/may/31/nest-hello-google-launches-facial-recognition-data-doorbell-uk-privacy-concerns-amazon-ring>.
20. O'Hear S. LaMetric Is A Smart And Hackable Ticker Display / Steve O'Hear // TechCrunch. – 2014. – Режим доступу до ресурсу: <https://techcrunch.com/2014/06/17/lametric/>.
21. Convolutional Neural Networks (LeNet) // DeepLearning 0.1. LISA Lab. – 2013. – Режим доступу до ресурсу: <http://deeplearning.net/tutorial/lenet.html>.
22. Штучні нейронні мережі: що це таке? // futurum.today. – 2017. – Режим доступу до ресурсу: <https://futurum.today/shtuchni-neironni-merezhi-shcho-tse-take/>.

ДОДАТКИ

ДОДАТОК А

Модуль навчання нейронної мережі.

"""

Description: Train emotion classification model

"""

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

```
from keras.callbacks import ReduceLROnPlateau
```

```
from keras.preprocessing.image import ImageDataGenerator
```

```
from load_and_process import load_fer2013
```

```
from load_and_process import preprocess_input
```

```
from models.cnn import mini_XCEPTION
```

```
from sklearn.model_selection import train_test_split
```

```
# parameters
```

```
batch_size = 32
```

```
num_epochs = 10000
```

```
input_shape = (48, 48, 1)
```

```
validation_split = .2
```

```
verbose = 1
```

```
num_classes = 7
```

```
patience = 50
```

```
base_path = 'models/'
```

```
# data generator
```

```
data_generator = ImageDataGenerator(
```

```
    featurewise_center=False,
```

```
    featurewise_std_normalization=False,
```

```
    rotation_range=10,
```

```
    width_shift_range=0.1,
```

```
    height_shift_range=0.1,
```

```
    zoom_range=.1,
```

```

horizontal_flip=True)

# model parameters/compilation
model = mini_XCEPTION(input_shape, num_classes)
model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['accuracy'])
model.summary()

# callbacks
log_file_path = base_path + '_emotion_training.log'
csv_logger = CSVLogger(log_file_path, append=False)
early_stop = EarlyStopping('val_loss', patience=patience)
reduce_lr = ReduceLRonPlateau('val_loss', factor=0.1,
                              patience=int(patience / 4), verbose=1)
trained_models_path = base_path + '_mini_XCEPTION'
model_names = trained_models_path + '.{epoch:02d}-{val_acc:.2f}.hdf5'
model_checkpoint = ModelCheckpoint(model_names, 'val_loss', verbose=1,
                                   save_best_only=True)
callbacks = [model_checkpoint, csv_logger, early_stop, reduce_lr]

# loading dataset
faces, emotions = load_dataset ()
faces = preprocess_input(faces)
num_samples, num_classes = emotions.shape
xtrain, xtest, ytrain, ytest = train_test_split(faces, emotions, test_size=0.2, shuffle=True)
model.fit_generator(data_generator.flow(xtrain, ytrain,
                                       batch_size),
                  steps_per_epoch=len(xtrain) / batch_size,
                  epochs=num_epochs, verbose=1, callbacks=callbacks,
                  validation_data=(xtest, ytest))

import pandas as pd
import cv2
import numpy as np

```

```
dataset_path = 'fer2013/fer2013/fer2013.csv'
```

```
image_size = (48, 48)
```

```
def load_dataset():
```

```
    data = pd.read_csv(dataset_path)
```

```
    pixels = data['pixels'].tolist()
```

```
    width, height = 48, 48
```

```
    faces = []
```

```
    for pixel_sequence in pixels:
```

```
        face = [int(pixel) for pixel in pixel_sequence.split(' ')]
```

```
        face = np.asarray(face).reshape(width, height)
```

```
        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']).as_matrix()
```

```
    return faces, emotions
```

```
def preprocess_input(x, v2=True):
```

```
    x = x.astype('float32')
```

```
    x = x / 255.0
```

```
    if v2:
```

```
        x = x - 0.5
```

```
        x = x * 2.0
```

```
    return x
```

Модуль детектування емоцій:

```
import cv2
```

```
detection_model_path = 'haarcascade_frontalface_default.xml'
```

```
faceCascade = cv2.CascadeClassifier(detection_model_path)
```

```
video_capture = cv2.VideoCapture(0)
```

```
while True:
```

```
    # Capture frame-by-frame
```

```

ret, frames = video_capture.read()
gray = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.1,
    minNeighbors=5,
    minSize=(30, 30),
    flags=cv2.CASCADE_SCALE_IMAGE
)
# Draw a rectangle around the faces
for (x, y, w, h) in faces:
    cv2.rectangle(frames, (x, y), (x+w, y+h), (0, 255, 0), 2)
# Display the resulting frame
cv2.imshow('Video', frames)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
video_capture.release()
cv2.destroyAllWindows()
from keras.utils.image_utils import img_to_array
import imutils
import cv2
from keras.models import load_model
import numpy as np

# parameters for loading data and images
detection_model_path = 'haarcascade_files/haarcascade_frontalface_default.xml'
emotion_model_path = 'models/_mini_XCEPTION.102-0.66.hdf5'

# hyper-parameters for bounding boxes shape
# loading models
face_detection = cv2.CascadeClassifier(detection_model_path)
emotion_classifier = load_model(emotion_model_path, compile=False)
EMOTIONS = ["angry", "disgust", "scared", "happy", "sad", "surprised",
            "neutral"]

```

```

#feelings_faces = []
#for index, emotion in enumerate(EMOTIONS):
    # feelings_faces.append(cv2.imread('emojis/' + emotion + '.png', -1))

# starting video streaming
cv2.namedWindow('your_face')
camera = cv2.VideoCapture(0)
while True:
    frame = camera.read()[1]
    #reading the frame
    frame = imutils.resize(frame,width=300)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces =
face_detection.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags=cv2.
CASCADE_SCALE_IMAGE)

    canvas = np.zeros((250, 300, 3), dtype="uint8")
    frameClone = frame.copy()
    if len(faces) > 0:
        faces = sorted(faces, reverse=True,
            key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]
        (fX, fY, fW, fH) = faces
            # Extract the ROI of the face from the grayscale image, resize it to a fixed
28x28 pixels, and then prepare
            # the ROI for classification via the CNN
            roi = gray[fY:fY + fH, fX:fX + fW]
            roi = cv2.resize(roi, (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()]
else: continue

for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)):
    # construct the label text
    text = "{}: {:.2f}%".format(emotion, prob * 100)

    # draw the label + probability bar on the canvas
    # emoji_face = feelings_faces[np.argmax(preds)]

    w = int(prob * 300)
    cv2.rectangle(canvas, (7, (i * 35) + 5),
                  (w, (i * 35) + 35), (0, 0, 255), -1)
    cv2.putText(canvas, text, (10, (i * 35) + 23),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45,
                (255, 255, 255), 2)
    cv2.putText(frameClone, label, (fX, fY - 10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
    cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY + fH),
                  (0, 0, 255), 2)
# for c in range(0, 3):
#     frame[200:320, 10:130, c] = emoji_face[:, :, c] * \
#     (emoji_face[:, :, 3] / 255.0) + frame[200:320,
#     10:130, c] * (1.0 - emoji_face[:, :, 3] / 255.0)

cv2.imshow('your_face', frameClone)
cv2.imshow("Probabilities", canvas)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

camera.release()
cv2.destroyAllWindows()

```