

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

_____ Сергій СТИРЕНКО
(підпис)

“ ___ ” _____ 20__ р

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Комп'ютерні системи та мережі»

спеціальності 123 «Комп'ютерна інженерія»

**на тему: «Система розпізнавання особи користувача за клавіатурним
почерком»**

Виконав:

студент IV курсу, групи Ю-61

_____ Терейковський Олег Ігоревич _____

(прізвище, ім'я, по батькові)

(підпис)

Керівник

_____ проф. д.т.н. Кулаков Юрій Олексійович _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Консультант

_____ н. контроль проф. д.т.н. Сімоненко В.П. _____

(назва розділу)

(вчені ступінь та звання, прізвище, ініціали)

(підпис)

Рецензент проф. каф. СПіСКС проф. д.т.н. Володимир ЗАЙЦЕВ _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМ. ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій СТИРЕНКО

(підпис)

“ ___ ” _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Терейковському Олегу Ігоревичу

1. Тема проєкту «Система розпізнавання особи користувача за клавіатурним почерком»

керівник проєкту Кулаков Юрій Олексійович, д.т.н., проф, затверджені наказом по університету від « 07 » травня _____ 2020р. № 1081-с _____

2. Термін здачі студентом закінченої роботи _____ 2020р.

3. Вихідні дані до проєкту технічне завдання, теоретичні дані.

4. Зміст пояснювальної записки: аналіз рішень в області розпізнавання особи за клавіатурним почерком, спосіб розробки системи розпізнавання особи користувача за клавіатурним почерком, розробка програмного забезпечення та експериментальні дослідження

5. Консультант роботи, з вказівкою розділів роботи, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Сімоненко В.П.		

6. Дата видачі завдання 01.09 .2019 року

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів дипломного проекту (роботи)	Строк виконання етапів проекту(роботи)	Примітки
1.	<i>Затвердження теми роботи</i>	<i>01.09.2020</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>15.12. 2020 - 15.03.2020</i>	
3.	<i>Розробка архітектури системи</i>	<i>15.03. 2020 - 25.03.2020</i>	
4.	<i>Розробка модулів системи</i>	<i>25.03 2020 - 5.04.2020</i>	
5.	<i>Програмна реалізація системи</i>	<i>5.04.2020-15.04.2020</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>15.04. 2020 - 20.05.2020</i>	
7.	<i>Захист програмного продукту</i>	<i>25.04.2020 – 24.05.2020</i>	
8.	<i>Передзахист</i>	<i>26.05.2020</i>	
9.	<i>Захист</i>	<i>20.06 2020</i>	

Студент

Олег ТЕРЕЙКОВСЬКИЙ

(підпис)

Керівник

Юрій КУЛАКОВ

(підпис)

Анотація

В бакалаврському дипломному проєкті розроблено систему розпізнавання особи користувача за клавіатурним почерком. В основу системи розпізнавання покладена згортова нейромережева модель типу Squeezenet, що пристосована до аналізу параметрів динаміки клавіатурного почерку.

Запропонована система може бути використана в контурі захисту комп'ютерної системи для прихованої аутентифікації оператора при введенні даних з клавіатури. При розробці програмного забезпечення системи використано мови програмування Python, C++, MATLAB.

Annotation

In the bachelor's diploma project the system of recognition of the user's personality by keyboard handwriting is developed. The recognition system is based on a convolutional neural network model of the Squeezenet type, which is adapted to the analysis of the parameters of the dynamics of keyboard handwriting.

The proposed system can be used in the security circuit of a computer system for covert authentication of the operator when entering data from the keyboard. Python, C++, MATLAB programming languages were used in the development of the system software.

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ІАЛЦ.467200.001 ВП	Відомість проєкту	1	
3	A4	ІАЛЦ.467200.002 ТЗ	Технічне завдання	4	
4	A4	ІАЛЦ.467200.003 ПЗ	Пояснювальна записка	61	
5	A4	ІАЛЦ.467200.004 Д1	Діаграма прецедентів	1	
6	A4	ІАЛЦ.467200.005 Д2	Діаграма класів	1	
7	A4	ІАЛЦ.467200.006 Д3	Схема алгоритму	1	
8	A4	ІАЛЦ.467200.007 Д4	Схема алгоритму	1	
9	A4	ІАЛЦ.467200.008 Д5	Схема структурна	1	
10	A4		Текст програми	6	

				<i>ІАЛЦ.045420.001 ВП</i>		
<i>Зм.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>	Відомість дипломного проєкту		
<i>Розробив</i>	<i>Терейковський</i>					
<i>Перевірів</i>	<i>Кулаков Ю.О.</i>					
<i>Н. Контр.</i>	<i>Сімоненко В.П.</i>					
<i>Затвердив</i>	<i>Стіренко С.Г.</i>					
				<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушіє</i>
					1	1
<i>НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ зр. 10-61</i>						

ТЕХНІЧНЕ ЗАВДАННЯ

**до дипломної роботи
освітньо-кваліфікаційного рівня бакалавр**

на тему: “ Система розпізнавання особи користувача за клавіатурним почерком ”

Київ – 2020 року

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ	2
3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1. Вимоги до розробленого продукту	3
5.2. Вимоги до програмного забезпечення.....	3
5.3. Вимоги до апаратної частини	3
6. ЕТАПИ РОЗРОБКИ	4

					<i>ІАЛЦ.045420.002 ТЗ</i>			
<i>Зм.</i>		<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		<i>Терейковський</i>			Система розпізнавання особи користувача за клавіатурним почерком Технічне завдання	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірів</i>		<i>Кулаков Ю.О</i>				1	4	
<i>Н. Контр.</i>		<i>Сімоненко В.П.</i>				<i>НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ гр. ІО-61</i>		
<i>Затвердив</i>		<i>Стіренко С.Г.</i>						

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Назва розробки: «Система розпізнавання особи користувача за клавіатурним почерком». Область застосування: захист комп'ютерних систем від несанкціонованого доступу.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи кваліфікаційно-освітнього рівня «бакалавр комп'ютерної інженерії», затверджене кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний інститут ім. Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даної розробки є підвищення рівня захисту комп'ютерних систем від несанкціонованого доступу. Призначення розробки полягає у створенні програмних засобів, що дозволяють розпізнати особу користувача на основі аналізу динаміки клавіатурного почерку.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є науково-технічна література з теорії і практики програмування, бакалаврські роботи інших студентів, публікації в Інтернеті з даних питань.

					<i>ІАЛЦ.045420.002 ТЗ</i>	Лист
						2
<i>Зм.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>		

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- Точність реєстрації параметрів клавіатурного почерку до 10^{-3} с.
- Точність розпізнавання особи користувача не менша ніж 70%.
- Малоресурсність програмного забезпечення, що реалізує модуль нейромережевого розпізнавання.

5.2. Вимоги до програмного забезпечення

- Операційна система MS Windows 10.
- Наявність на комп'ютері Python не нижче версії 3.7.
- Наявність на комп'ютері MATLAB не нижче версії R2018b.

5.3. Вимоги до апаратної частини

- Процесор Intel або AMD x86-64 з частотою 3,8-4,2 ГГц.
- Обсяг оперативної пам'яті 8 ГБ.
- Відеокарта з підтримкою OpenGL 3.3 та 1GB GPU.
- SSD з обсягом вільної пам'яті 8 ГБ.

					<i>ІАЛЦ.045420.002 ТЗ</i>	Лист
Зм.	Арк.	№ докум.	Підпис	Дата		3

ЕТАПИ РОЗРОБКИ

	Дата
Вивчення літератури	28.03.2020
Складання і узгодження технічного завдання	03.04.2020
Створення модулів системи, що розробляється	15.04.2020
Тестування окремих модулів системи	25.04.2020
Допрацювання, налагодження і виправлення помилок	01.05.2020
Оформлення документації дипломної роботи	15.05.2020

					<i>ІАЛЦ.045420.002 ТЗ</i>	Лист
Зм.	Арк.	№ докум.	Підпис	Дата		4

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломного проекту
на тему: «Система розпізнавання особи користувача за
клавiатурним почерком»

Київ – 2020 року

ЗМІСТ

ВСТУП	2
РОЗДІЛ 1. АНАЛІЗ РІШЕНЬ В ОБЛАСТІ РОЗПІЗНАВАННЯ ОСОБИ	
КОРИСТУВАЧА ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ.....	4
1.1. Характеристика задачі аналізу клавіатурного почерку	4
1.2. Аналіз методів розпізнавання особи користувача за клавіатурним почерком.	8
Висновки до розділу 1	15
РОЗДІЛ 2. СПОСІБ РОЗРОБКИ СИСТЕМИ РОЗПІЗНАВАННЯ ОСОБИ	
КОРИСТУВАЧА ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ.....	16
2.1 Проектування системи розпізнавання особи користувача	16
2.2 Розробка базової нейромережевої моделі	21
2.3 Нейромережева модель для розпізнавання особи користувача	27
Висновки до розділу 2	36
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА	
ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ	37
3.1. Опис функціональних можливостей програмного забезпечення	38
3.2. Опис інтерфейсу користувача	43
3.3. Експериментальні дослідження.....	51
Висновки до розділу 3	56
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	58
ДОДАТКИ	
Додаток 1. Копії графічних матеріалів	
Додаток 2. Лістинг програми	

					<i>ІАЛЦ.045420.003 ПЗ</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		<i>Терейковський О.І.</i>			Системи розпізнавання особи користувача за клавіатурним почерком Пояснювальна записка	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірив</i>		<i>Кулаков Ю. О.</i>				1	62	
<i>Реценз.</i>						<i>НТУУ «КПІ ім.Ігоря Сікорського» ФІОТ, ІО-61</i>		
<i>Н. Контр.</i>		<i>Сімоненко В. П.</i>						
<i>Затвердив</i>		<i>Стіренко С.Г.</i>						

ВСТУП

В теперішній час однією із найбільш актуальних загроз безпеці інформаційних систем як загального, так і спеціального призначення є загроза витоку даних, яка зростає пропорційно інтенсивності використання інформаційних технологій. При цьому якщо впливу зовнішніх загроз витоку даних можна протидіяти за допомогою апробованих апаратно-технічних засобів, котрі стосуються захисту каналів передачі даних, захисту зовнішніх веб-ресурсів організації, антивірусного захисту, то протидіяти внутрішнім загрозам набагато складніше.

Разом з тим статистика, що стосується розкритих порушень в області витоку даних, свідчить про причетність до цього так званих інсайдерів, тобто співробітників організацій, де використовуються вказані інформаційні системи. Таким чином, забезпечення ефективного контролю доступу співробітників до ресурсів інформаційної системи організації дозволить значно підвищити рівень безпеки конфіденційних даних.

Серед програмно-апаратних засобів контролю доступу перспективним напрямком є використання засобів, котрі базуються на аналізі динамічних біометричних характеристик людини. Перевагами таких засобів є можливість прихованої ідентифікації, невіддільність біометричної характеристики від власника і висока складність підробки. Тому за останні декілька років засоби аналізу параметрів клавіатурного почерку знаходять все більш широке застосування в інформаційних системах самого різного призначення. Крім означених, до переваг засобів аналізу параметрів клавіатурного почерку відносять можливість реєстрації параметрів клавіатурного почерку з використанням тільки стандартного периферійного обладнання, широке застосування в інформаційних системах парольних і технологічних даних у вигляді набору символів, що дозволяє провести ідентифікацію особи не тільки під час вводу парольних даних, але й в процесі взаємодії з системою. Крім того, доведено тісний зв'язок параметрів клавіатурного почерку з

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		2

фізіологічним і емоційним станом користувача, що вказує на потенційні можливості моніторингу емоційного і фізіологічного стану користувачів інформаційних систем. Це може бути використано, наприклад, для прихованого моніторингу операторів об'єктів критичної інфраструктури або для відслідковування настроїв корпоративних клієнтів.

В сучасних умовах засоби аналізу параметрів клавіатурного почерку в основному використовуються в контурах багатofакторної перевірки особи для біометричної аутентифікації користувача. Також відомі спроби впровадження подібних засобів в системи дистанційного навчання для ідентифікації слухачів та оцінки рівня сприйняття ними навчальних матеріалів. Однак, широкому застосуванню засобів аналізу параметрів клавіатурного почерку заважає ряд недоліків, пов'язаних з тим, що особу користувача необхідно розпізнати на основі аналізу зашумлених багатовимірних даних. Задача ускладнюється тим, що у випадку обробки текстового паролю аналізу підлягає достатньо обмежений обсяг даних, а у випадку обробки технологічної інформації виникає необхідність аналізу великих обсягів багатовимірних даних. Зазначимо, що на сьогодні одним із найбільш перспективних шляхів вирішення подібних задач є застосування нейромережових технологій.

Цим пояснюється актуальність завдання впровадження нейромережових технологій для удосконалення засобів аналізу параметрів клавіатурного почерку призначених для розпізнавання особи користувача.

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		3

РОЗДІЛ 1

АНАЛІЗ РІШЕНЬ В ОБЛАСТІ РОЗПІЗНАВАННЯ ОСОБИ КОРИСТУВАЧА ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ

1.1. Характеристика задачі аналізу клавіатурного почерку

В загальному випадку під поняттям клавіатурного почерку розуміють поведінкову біометричну характеристика, що складається з патернів ритму і динаміки, характерних для даного оператора при наборі тексту [1, 5]. Іншими словами клавіатурний почерк - це унікальний стиль введення символів, котрий описується динамікою вводу символів з клавіатури та наявністю помилок, характерних для кожного оператора.

Аутифікація/ідентифікація користувача комп'ютерної системи за клавіатурним почерком дозволяє підвищити захищеність системи, ускладнити можливість відмови від авторства та підвищити зручність для користувача за рахунок можливого використання спрощеного паролю. Людині буде досить запам'ятати зручну комбінацію, наприклад, довжина пароля може складатися від 4 до 8 символів. З огляду на особливості введення, загроза розкрадання пароля зловмисником стає менш пріоритетною, так як швидкість введення може значно відрізнятись. Крім того, різниця між процедурою ідентифікації та аутифікації полягає в тому, що при ідентифікації користувач повинен ввести загальновідому інформацію (ім'я користувача), а при аутифікації користувач вводить відому тільки йому секретну інформацію – пароль. Таким чином обидві процедури можуть бути зведені до розпізнавання особи користувача за його клавіатурним почерком, що являється ще однією перевагою відповідних систем [2, 3].

Системи аутифікації користувачів за клавіатурним почерком відносяться до біометричних систем аутифікації на основі поведінкових характеристик людини, що відповідно міжнародним стандартам в області захисту інформації повинні виконувати наступні функції:

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		4

- Отримати біометричний зразок безпосередньо від кінцевого користувача.

- Отримувати біометричні дані з отриманого зразка або ж виводити біометричні ознаки з біометричних даних у вигляді, придатному для зіставлення з одним або декількома контрольними шаблонами.

- Порівнювати біометричні ознаки з тими, що містяться в одному або декількох контрольних шаблонах.

- Визначати ступінь подібності відповідно до індексу або іншим показником або. Як варіант, ранжувати шаблони відповідно до рівня подібності.

- Повертати результат з додатком із зазначенням, чи була ідентифікація і/або аутентифікація виконана успішно.

- Зберігати біометричні дані і пов'язану з ними системну інформацію і управляти цими даними.

При описі такої системи використовується стандартизовані терміни:

- Біометричний зразок (biometric sample) - необроблені дані, котрі представляють собою біометричну характеристику кінцевого користувача, які були зареєстровані біометричної системою.

- Біометричні дані (biometric data) - інформація, отримана в результаті обробки біометричного зразка, що використовується для побудови біометричних, в тому числі і контрольних шаблонів.

- Біометричний профіль (biometric profile) - набори або комбінації біометричних даних, які використовуються для виконання певних біометричних функцій.

У випадку біометрії на основі клавіатурного почерку біометричні дані представляють собою список хронологічно впорядкованих подій, що містить наступну інформацію:

- Події, які генеруються при роботі з клавіатурою. Розрізняють два види подій: натискання клавіш і їх відпускання;

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		5

- Код використаної клавіші та введений символ;
- Часові характеристики, що відповідають часу тривалості подій.

Базовими параметрами, що характеризують клавіатурний почерк у випадку використання універсальної клавіатури, є параметри динаміки вводу до яких відносяться:

- Термін утримання клавіші (ТУК) - це період, протягом якого клавіша знаходиться в натиснутому стані;

- Термін накладення натискань клавіш це термін події, що відбувається тоді, коли одна клавіша ще не відпущена, а інша вже натискається. З підвищенням швидкості набору тексту збільшується число накладень;

- Пауза між натисканням - це термін часу, коли одна клавішу вже відпущена, а наступна ще не натиснута.

Вважається, що для реалізації аналізу необхідно, щоб вказані терміни були зареєстровані з точністю до однієї мілісекунди. Крім того, в деяких системах аналізу клавіатурного почерку використовується похідні вказаних термінів та додаткові параметри:

- Чистий розмір тексту без урахування видалених символів;
- Загальний термін вводу тексту, що розраховується від моменту натискання першої клавіші до моменту натискання останньої клавіші;
- Мінімальна пауза між натисканнями;
- Максимальна пауза між натисканнями;
- Середня пауза між натисканнями;
- Різкість натискання, що розраховується як середній час між натисканням і відпусканням клавіші;
- Кількість набраних символів в хвилину;
- Швидкість netto - чиста швидкість набору тексту, що розраховується для всіх невидалених символів тексту;
- Швидкість wrm, що розраховується як швидкість нетто поділена на 5;

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		6

- Швидкість бруто - швидкість набору з урахуванням віддалених символів, дозволяє оцінити втрати швидкості в зв'язку з неправильним введенням;

- Втрати від виправлень - показує в процентному співвідношенні, наскільки падає швидкість через часу, витраченого на вчинення помилок і їх виправлення;

- Ступінь аритмічності при наборі, що розраховується як середнє відхилення паузи між поточним натисненням клавіші від середнього значення;

- Число виправлень - число символів, віддалених за допомогою клавіші BackSpace;

- Розмір максимального фрагмента тексту, набраного без виправлень;

- Число перекриттів між клавішами при введенні тексту заданого обсягу.

Крім того, у випадку використання спеціалізованої клавіатури аналізу можуть підлягати показники, що являються похідними від значень тиску користувачем на клавіші.

Використавши описані особливості біометричних систем та визначений набір параметрів клавіатурного почерку сформовано, представлену на рис. 1.1, типову послідовність функціонування системи аутентифікації користувачів на основі клавіатурного почерку [4, 8]. Як і для більшості систем біометричної аутентифікації, що базуються на динамічних характеристиках людини системам аутентифікації за клавіатурним почерком притаманні два режими функціонування ідентифікації та навчання, що призначений для формування біометричних профілів користувачів на основі параметрів клавіатурного почерку. В режимі ідентифікації система проводить аналіз параметрів клавіатурного почерку на предмет відповідності зареєстрованого профілю користувача з відомими профілями. Саме труднощі аналізу параметрів клавіатурного почерку є основною перепорою при розробці сучасних систем аутентифікації за клавіатурним почерком [19, 22].

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		7

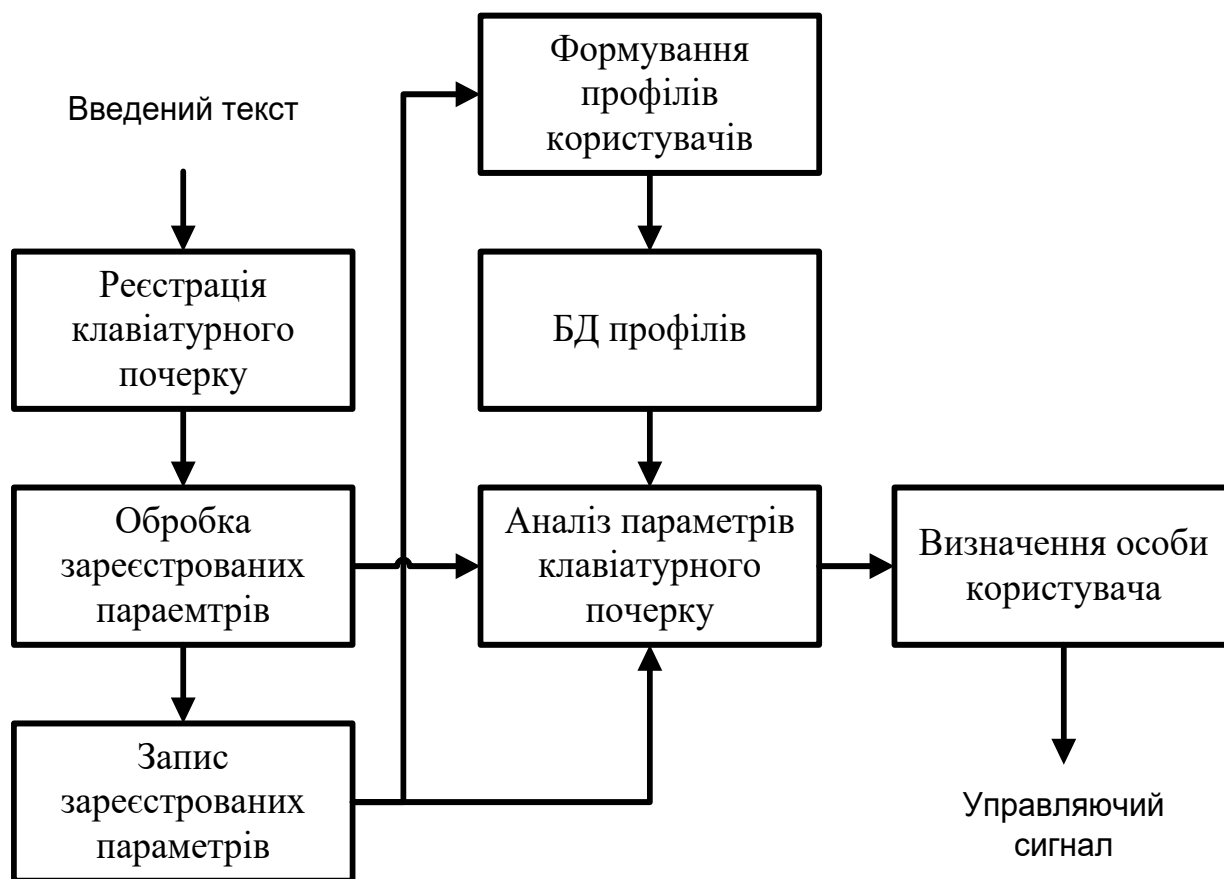


Рис. 1.1. Структура системи аутентифікації за клавіатурним почерком

1.2. Аналіз методів розпізнавання особи користувача за клавіатурним почерком.

В роботі [10] визначено, що завдання аутентифікації по клавіатурному почерку, по суті, зводиться до задачі розпізнавання клавіатурного почерку конкретного користувача за допомогою двох основних груп методів:

- Геометричні, що використовують різні міри близькості (міра Хеммінга, Евклідова міра та ін.);
- Методи на основі застосування штучних нейронних мереж.

Геометричні методи і методи, засновані на застосуванні ІНС, мають наступні недоліки:

- Відносно невисоку точність класифікації внаслідок грубої апроксимації областей рішення (геометричні методи).

- Довгий процес навчання, можливість виникнення паралічу навчання (методи на основі ІНС).

- Проблема переносу результатів навчання на більш широкий клас користувачів, клавiатурний почерк яких не був використаний при навчанні (методи на основі ІНС).

- Необхідність збору великої кількості даних для навчання.

- Збільшення довжини парольної фрази для підвищення точності розпізнавання. Декларується, що унікальні характеристики клавiатурного почерку можуть бути виявлені як при наборі вільного тексту, так і по набору ключової фрази.

Потрібно відзначити, що реалізація таких методів дозволяє не тільки ідентифікувати користувачів, але і визначити їх функціональний стан.

Такий підхід дозволяє:

- Контролювати фізичне самопочуття користувачів.

- Забезпечити простий і надійний метод ідентифікації.

- Спростити запам'ятовування користувачами складних паролів, оскільки в якості паролю може бути використано довільний текст.

Стаття [12] присвячена задачі зменшення розмірності вхідної вибірки системи розпізнавання клавiатурного почерку. Вказується, що актуальність цієї задачі спричинена великим обсягом характеристик клавiатурного почерку та великим обсягом символів, котрі вводяться користувачем в режимі прихованої ідентифікації. Пропонується зменшити розмірність вхідної вибірки за рахунок методів стиснення статистичних даних. Їх використання дозволяє: скоротити обчислювальні витрати при обробці даних; боротися з перенавчанням; стискати дані для більш ефективного зберігання інформації; візуалізувати дані; отримувати нові ознаки.

До способів вирішення даного завдання відноситься метод головних компонент - один з найбільш поширених класичних методів, що застосовуються для обробки великих обсягів статистичних даних. Ідея методу

					<i>ІАЛЦ.045420.003 ПЗ</i>	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

полягає в пошуку в вихідному просторі гіперплощини заданої розмірності з подальшою проекцією вибірки на дану гіперплощину. При цьому вибирається та гіперплощина, помилка проектування даних на яку є мінімальною в сенсі суми квадратів відхилень.

Також визначено, що використання методу головних компонент має ряд обмежень. По-перше, якщо вибірка даних утворює приховану поверхню, яка є нелінійною, метод головних компонент може призводити до неадекватних результатів. По-друге, існує можливість визначення прихованих компонент тільки з точністю до афінного перетворення. По-третє, існує складність у визначенні моменту зупинки процедури виділення факторів, так як в процесі послідовного виділення факторів вони включають в себе все менше і менше варіативних складових. Крім того, запропоновано застосовувати для вирішення задачі стиснення вхідної інформації використовувати факторний аналіз. Факторний аналіз використовується з метою скорочення даних, визначення структури взаємозв'язків між змінними, тобто класифікації змінних, відбору факторів, що визначають досліджувані результативні показники, визначення форми залежності між факторами і результативними показниками, котрі використовуються для моделювання. Однак використання факторного аналізу також має ряд обмежень:

- При використанні факторного аналізу виходять з того, що фактори змінюються незалежно один від одного.
- Факторний аналіз супроводжується втратою вихідної інформації про зв'язки між змінними, і ця втрата часто досить відчутна - від 30 до 50%.
- Цінність представляє рішення, коли групи змінних, які відповідають різним чинникам, незначно корелюють один з одним.
- Не всі дані можуть бути піддані факторному аналізу.

Крім того, зазначається, що використання статистичних методів не дозволяє в достатній мірі ефективно виявляти неявні залежності, приховані в отриманих даних. Тому зроблено висновок про те, що для вирішення даного

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		10

завдання перспективним є застосування методів штучного інтелекту. Це пояснюється тим, що їх найважливішою особливістю є здатність вирішувати слабо формалізовані завдання, домагаючись при цьому результатів, ефективність яких співвідноситься з рішеннями, які отримуються людиною-експертом. Зроблено припущення, що найбільш придатними для вирішення завдання виявлення прихованих характеристик клавiатурного почерку є імунні системи і нейронні мережі. Але найсерйозніша проблема використання клавiатурного почерку - це сильна залежність результату від психофізичного стану користувача. Наприклад, якщо людина погано себе почуває, вона, можливо, не пройде перевірку справжності, так як швидкість введення символів може значно змінитись. Також необхідно враховувати залежність вводу тексту від досвіду користувача і технічних характеристик клавiатури. Крім того, в роботі [20] визначено недоліки використання занадто довгих паролів. Це може негативно вплинути на результати розпізнавання, оскільки користувач замислюється над введенням, згадуючи комбінацію, що призводить до зміни значень часових інтервалів між натисненнями клавiш. Навпаки, рухи рук при введенні завченого короткого пароля управляються підсвідомим процесом мислення. Отже, час введення завченого короткого пароля буде майже завжди однаковим.

Аналіз джерел [1-4, 9-11] дозволяє стверджувати, що на сьогодні не існує загальноприйнятої методології формування множини вхідних даних модуля розпізнавання клавiатурного почерку. Однак спільною рисою більшості відомих підходів до розпізнавання клавiатурного почерку є використання в якості вхідних даних відфільтрованих значень часу між натисканням двох і трьох окремих клавiш. Також можна визначити залежність номенклатури вхідних даних від математичного забезпечення процесу розпізнавання, який зводиться до порівняння вхідної послідовності з еталоном клавiатурного почерку певної емоції оператора комп'ютерної системи. Через те, що параметри клавiатурного почерку корелюються як з особою конкретного

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		11

оператора, так і з його емоційним станом, завдання розпізнавання емоцій може вирішуватися спільно з завданням аутентифікації. Розрізняють два типи моделей розпізнавання: по заздалегідь визначеному фрагменту тексту і по фрагменту тексту довільного змісту. В обох випадках для визначення еталонів клавіатурного почерку оператору необхідно декілька разів ввести один або кілька фрагментів одного і того ж тексту. Більшість відомих еталонів представляють собою статистичні моделі параметрів клавіатурного почерку, що базуються наприклад на нормальному або бімодальному законах розподілу [13]. У випадку необхідності аналізу клавіатурного почерку на базі визначеного текстового фрагмента основу еталонів, як правило, становлять показники часу утримання клавіші і часу між утриманням клавіш, що стосуються послідовного порядку натискання клавіш. При розпізнаванні на основі фрагменту тексту довільного змісту еталони, в своїй більшості, формуються на основі статистичних показників часу утримання клавіші та часу між утриманням клавіш для окремих стійких послідовностей подій клавіатури, що відображають особливості динаміки набору тексту окремого оператора. Разом з тим, в роботах [8, 9, 13] на основі експериментальних досліджень доведена незадовільна адаптованість статистичних моделей до зміни емоційного стану оператора та складність їх формування у випадку аналізу фрагменту тексту довільного змісту. При цьому вказано, що як раз з точки зору доведеної ефективності аналізу багатовимірних даних, перспективним напрямком вдосконалення систем розпізнавання клавіатурного почерку є застосування нейромережових моделей. Це підтверджується даними робіт [14, 15, 17], в яких, разом з позитивними результатами застосування нейромережових моделей, відзначається їх обмеженість, пов'язана з труднощами формування номенклатури вхідних параметрів та визначенням архітектурних параметрів. Також відзначена застарілість використаних нейромережових моделей типу багат шарового персептрону, ймовірнісної нейронної мережі, карти Кохонена, мережі Хопфілда та мережі РБФ.

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		12

Аналіз сучасних науково-практичних робіт [5, 10-13], присвячених розробці нейромережевих засобів аналізу динаміки клавіатурного почерку дозволяє стверджувати, що основна тенденція розвитку зазначених засобів пов'язана з розробкою використовуваних в них нейромережевих моделей на базі згорткової нейронної мережі. Так, в статті [11] запропонований метод аутентифікації користувачів, що базується на аналізі клавіатурного почерку довільного тексту за допомогою згорткової нейронної мережі. Розроблено процедуру подання довільного тексту в вигляді квадратного багатоканального кольорового рисунку. Як джерело даних для формування вхідного поля застосовувалися параметри, отримані за допомогою виразів виду:

$$D=R_i - P_i, \quad (1.1)$$

$$F_{type1}=P_{i+1}-R_i, \quad (1.2)$$

$$F_{type2}=R_{i+1}-R_i, \quad (1.3)$$

$$F_{type3}=P_{i+1}-P_i, \quad (1.4)$$

$$F_{type4}=R_{i+1}-P_{i+1}, \quad (1.5)$$

де R_i - час відпускання i -ої клавіші,

P_i - час натискання i -ої клавіші,

i - номер клавіші, яка використовується.

Усереднені значення кожного із зазначених параметрів відповідають одному із каналів вхідного поля згорткової нейронної мережі, яке співвідноситься з розкладкою клавіатури, що використовується. В результаті проведених експериментів показано, що запропонований метод нейромережевої аутентифікації дозволяє досягти точності розпізнавання

користувачів на рівні 96,09%, що на 10% вище точності розпізнавання на базі методу опорних векторів. Також в роботі визначено необхідність дослідження перспективних видів згорткових нейронних мереж. Крім цього вказано, що запропонована процедура перетворення тексту до вигляду, придатного для нейромережевого аналізу, не враховує послідовності натискання клавіш. Схожі результати отримані також в роботах [10, 12], які в основному присвячені застосуванню згорткової нейронної мережі типу LeNet для підвищення надійності систем біометричної аутентифікації користувачів на основі аналізу клавіатурного почерку. При цьому в статтях [17, 18] запропонована процедура перетворення параметрів клавіатурного почерку в багатоканальне кольорове зображення, яка на відміну від [11] враховує послідовність натиснення клавіш при наборі тексту. В роботах [20-22] для аналізу клавіатурного почерку використані інші типи нейромережних моделей. Наприклад, в [22] розроблена глибока нейронна мережа з переднавчанням на базі автоенкодера, а в роботі [21] запропонована модель розпізнавання типу мережі LSTM. Однак глибокі нейронні мережі по типу багат шарового персептрона не дозволяють врахувати топологію даних, а побудова мережі LSTM пов'язана зі складністю формування навчальної вибірки. Показано, що розробка нейромережевої моделі на базі згорткової нейронної мережі викликає ряд труднощів, пов'язаних з вибором типу.

Слід зазначити, що характеристики різних типів згорткових нейронних мереж відрізняються досить сильно, оскільки адаптовані під різні умови застосування. Тому становить інтерес дослідження можливостей аналізу клавіатурного почерку з використанням сучасних типів згорткових нейронних мереж. Базуючись на результатах [21], визначено що при розробці нейромережевої системи розпізнавання доцільно використовувати згорткову нейронну мережу типу SqueezeNet. Відмінними рисами SqueezeNet є відносно невисока ресурсомісткість, висока швидкість і точність розпізнавання, що має вирішальне значення для систем біометричної аутентифікації.

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		14

Висновки до розділу 1

В результаті проведеного аналізу відомих рішень в області розпізнавання особи користувача за клавіатурним почерком визначено:

- Перспективність застосування засобів розпізнавання особи за клавіатурним почерком пояснюється можливістю підвищення надійності системи аутентифікації користувачів за рахунок реалізації прихованого моніторингу справжності особи користувача. Крім того, застосування таких засобів дозволить підвищити ергономічність системи аутентифікації за рахунок використання в якості парольних даних довільних текстів.

- Труднощі розробки засобів розпізнавання особи за клавіатурним почерком полягають у необхідності аналізу зашумлених багатовимірних даних, що співвідносяться з параметрами клавіатурного почерку. Зашумленість даних пояснюється в першу чергу залежністю значень вказаних параметрів від функціонального та психоемоційного стану користувача.

- Перспективним шляхом вдосконалення засобів розпізнавання особи за клавіатурним почерком є впровадження в них модулю аналізу параметрів динаміки клавіатурного почерку на основі згорткової нейронної мережі типу SqueezeNet.

					<i>ІАЛЦ.045420.003 ПЗ</i>	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

СПОСІБ РОЗРОБКИ СИСТЕМИ РОЗПІЗНАВАННЯ ОСОБИ КОРИСТУВАЧА ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ

2.1 Проектування системи розпізнавання особи користувача

Перший етап розробки системи розпізнавання особи користувача за клавіатурним почерком асоційовано з побудовою відповідних моделей даних такої системи, що дозволяють провести аналіз предметної області, сформулювати вимоги та здійснити проектування структури програмного забезпечення [1, 9]. Оскільки процес розробки програмного забезпечення не передбачав можливість розпаралелювання, то з позицій спрощення документування використано каскадну модель життєвого циклу. Для побудови моделей застосовано мову проектування UML (Unified Modeling Language). Вибір UML пояснюється її універсальністю, апробованістю та доступністю засобів автоматизації процесу проектування. Для розробки UML-діаграм використано сервіс Rational Rose, виробництва компанії IBM.

Побудовано наступні діаграми:

- Прецедентів (варіантів використання), що дозволяє представити систему розпізнавання у вигляді множини сутностей або акторів, що взаємодіють з системою шляхом так званих варіантів використання. При цьому актором (дійовою особою) називається будь-яка сутність, що взаємодіє з системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, яка може служити джерелом впливу на систему, що проектується. У свою чергу, варіант використання служить для опису сервісів, які система надає актору. Таким чином, кожен варіант використання визначає деяку множину дій, яка реалізується системою при взаємодії з актором. При цьому в діаграмі прецедентів не міститься інформації про спосіб реалізації взаємодії акторів з системою розпізнавання, що дозволяє підвищити рівень формалізації проектування.

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		16

- Пакетів, що в першу чергу призначена для організації елементів системи в окремі групи для спрощення структури і спрощення організації роботи з моделлю системи. В пакетах можуть групуватися практично будь-які елементи UML (в тому числі і самі пакети). Кожен пакет ідентифікується за рахунок унікального імені, що уточнює область видимості кожного елемента пакета. Елементи, що входять в один пакет, можуть взаємодіяти між собою без уточнення імен.

- Класів, котра визначає типи класів системи розпізнавання клавіатурного почерку та статичні зв'язки, що існують між цими класами. В загальному випадку діаграма класів представляє собою граф, вершинами якого є елементи типу "класифікатор", що пов'язані між собою різними типами структурних відносин. Слід зауважити, що діаграма класів може також містити інтерфейси, об'єкти і зв'язки між ними. Вважається, що діаграма класів повинна відображати статичну структурну модель системи, що проектується. Тому діаграму класів прийнято вважати графічним представленим таких структурних взаємозв'язків логічної моделі системи, що не залежать від часу. На діаграмі класів передбачено відображати основні атрибути класів та обмеження, які накладаються на зв'язки між класами.

- Компонентів, характеризує особливості фізичного представлення системи розпізнавання та забезпечує візуалізацію структури програмної системи. Також діаграма компонентів дозволяє за рахунок формування залежностей між програмними компонентами, спростити визначення архітектури системи. В ролі вказаних компонент можуть бути використані практично всі види представлення програмного коду. Пунктирні стрілки, що з'єднують модулі даної діаграми, показують відношення взаємозалежності, аналогічні тим, які мають місце при компіляції початкового програмного коду. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними.

Розробка архітектури системи була розпочата з побудови діаграми прецедентів (див. додаток 1). До складу цієї діаграми входять актори “Administrator”, “Monitor” та “User”.

Актор “Administrator” який представляє адміністратора системи та відповідає за налаштування параметрів системи розпізнавання особи користувача за клавіатурним почерком.

Актор “User” співвідноситься з користувачем комп’ютерної системи, котрий вводить текст з клавіатури та чия особа підлягає розпізнаванню.

“Monitor” – актор, що співвідноситься з блоком моніторингу користувачів комп’ютерної системи.

Розроблена діаграма прецедентів передбачає виконання наступних функціональних завдань:

- Register keyboard handwriting options – реалізація реєстрації параметрів клавіатурного почерку.
- Specify registration options – визначення параметрів реєстрації клавіатурного почерку.
- Save – збереження зареєстрованих параметрів клавіатурного почерку.
- Recognize user identity – розпізнавання особи користувача.
- Define architectural parameters CNN – визначення архітектурних параметрів згорткової нейронної мережі.
- Conduct training CNN – навчання згорткової нейронної мережі.
- Save CNN – збереження параметрів згорткової нейронної мережі.
- Analyze keyboard handwriting options with CNN – аналіз параметрів клавіатурного почерку за допомогою згорткової нейронної мережі.
- Signal – сигналізація про результати нейромережевого аналізу, що полягають в розпізнаній особі користувача.

Розроблена діаграма прецедентів передбачає, що між прецедентами використовуються відношення розширення (“extend”) та включення (“include”). Відношення розширення відзначає той факт, що один із

прецедентів може приєднувати до своєї поведінки деяку додаткову поведінку, характерну для іншого варіанту використання. Відношення включення між двома прецедентами вказує, що поведінка одного прецеденту включається як складовий компонент в послідовність поведінки іншого. На рис. 2.1 зображена діаграма пакетів, які складають систему розпізнавання.

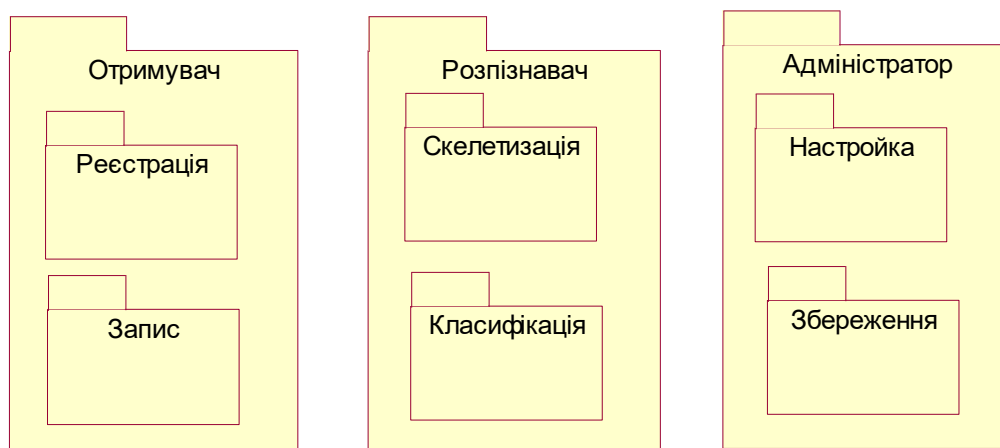


Рис. 2.1. Діаграма пакетів

Як показано на рис. 2.1 до складу системи розпізнавання входять пакети:

- “Отримувач” – пакет, який відповідає за отримання параметрів клавіатурного почерку користувача, створення матриць вказаних параметрів і побудову відповідних зображень. Передбачено, що зображення повинні бути представлені у форматі RGB. У складі пакету “Отримувач” є підпакети: “Реєстрація” – для фіксації параметрів клавіатурного почерку, “Запис” – для збереження цих параметрів до бази даних.

- “Розпізнавач” – пакет для розпізнавання особи користувача на основі аналізу параметрів клавіатурного почерку. До складу даного пакету входять підпакети “Скелетизація” для скелетизації параметрів клавіатурного почерку та “Класифікація”, який відповідає за розпізнавання особи.

- “Адміністратор” – пакет для адміністрування системи. Даний пакет складається з підпакетів “Настройка” та “Збереження”.

Підпакет “Настройка” відповідає за надання адміністратору можливості налаштувати відповідні параметри скелетизації вхідних даних і їх обробку.

Підпакет “Збереження” призначений для збереження результатів розпізнавання особи користувача.

На наступному кроці розробки архітектури системи розпізнавання побудовано, показано на рис. 2.2, діаграму компонентів даної системи.

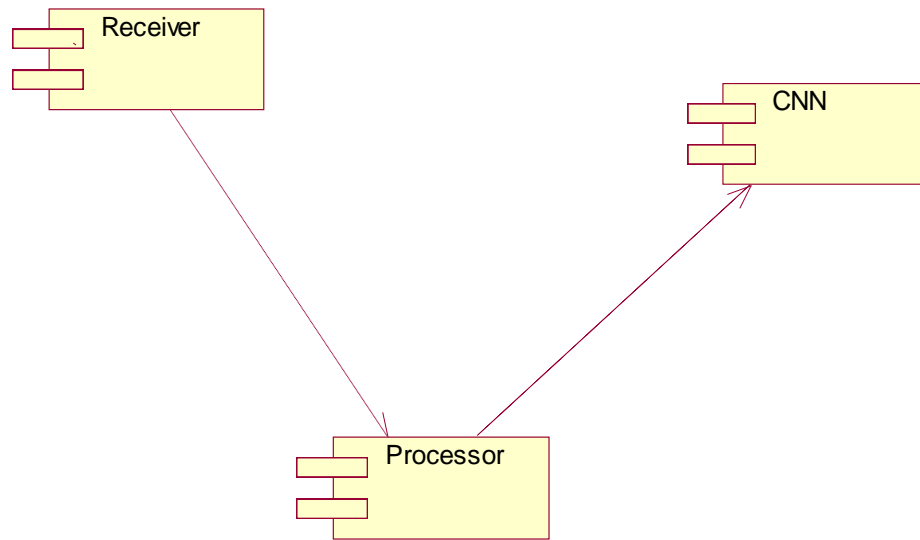


Рис. 2.2. Діаграма компонентів

Передбачено, що система повинна складатись з трьох модулів Receiver, Processor та CNN. Вказані модулі представляють інтерфейс отримання параметрів клавіатурного почерку користувача, обробку отриманих параметрів клавіатурного почерку та розпізнавання особи користувача. Відповідно результатам аналізу літературних робіт в області аналізу клавіатурного почерку, для розпізнавання особи передбачено використовувати згорткову нейронну мережу.

На заключному етапі розробки архітектури системи розпізнавання було побудовано діаграму класів (див. додаток 1). До її складу входять класи KeyboardProgram, FormImagePy, Pr2_FormImages, Pr2 та Pr3, що призначені для реєстрації і обробки параметрів клавіатурного почерку, навчання

нейронної мережі та її застосування для розпізнавання особи користувача. Розробка вказаних діаграм дозволила перейти до наступного етапу проектування, що співвідноситься з розробкою математичного забезпечення системи розпізнавання особи користувача за клавіатурним почерком, котре базується на використанні нейромережових рішень.

2.2 Розробка базової нейромережової моделі

Базуючись на результатах [6, 16] в якості базової нейромережової моделі використано згорткову нейронну мережу. Така модель є розвитком класичного типу багатошарової нейромережової моделі з прямим розповсюдженням сигналу (багатошарового персептрону), структура показана на рис. 2.3.

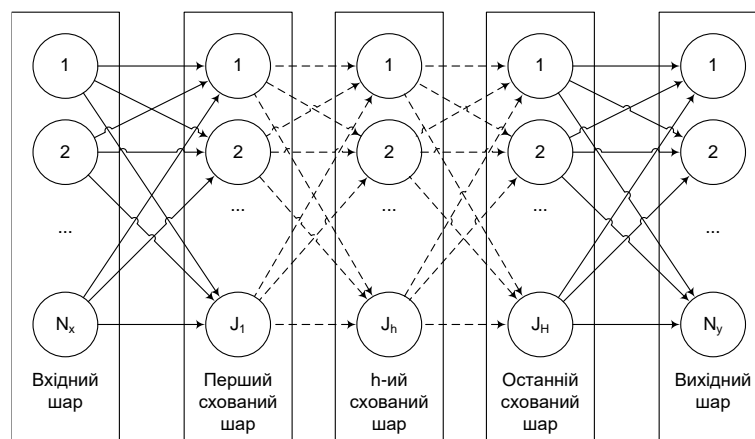


Рис. 2.3. Структура багатошарового персептрону

До складу показаного на рис. 2.3 багатошарового персептрону входять: вхідний шар, сховані шари та вихідний шар. Кількість схованих шарів нейронів дорівнює H . Кількість нейронів у вхідному шарі дорівнює N_x , кількість нейронів у кожному із схованих нейронних шарів дорівнює J , а кількість нейронів у вихідному шарі дорівнює N_y . В загальному випадку $N_x \neq J \neq N_y$. Як правило, $N_x > N_y$. Особливістю багатошарового персептрону є повнозв'язна структура зв'язків між нейронами сусідніх шарів. Кількість вагових коефіцієнтів у багатошаровому персептроні визначається так:

$$K_w = (N_x + N_y + (H - 1)) \times J, \quad (2.1)$$

де K_w – кількість вагових коефіцієнтів.

Зазначимо, що саме кількість вагових коефіцієнтів в основному визначає ресурсоемність нейромережевої моделі. У вхідних нейронах багатошарового перцептронів використовується лінійна функція активації виду:

$$y_i = x_i, \quad (2.2)$$

де x_i – вхідний сигнал для i -го нейрону вхідного шару;

y_i – вихідний сигнал для i -го нейрону вхідного шару.

У схованих та вихідних нейронах як правило використовуються функції активації типу сигмоїд, гіперболічний тангенс та ReLU (rectified linear unit), що визначаються за допомогою виразів (2.3-2.5) відповідно.

$$y_j = 1/(1 + e^{-ax_{j,s}}), \quad (2.3)$$

$$y_j = d(e^{ax_{j,s}} - e^{-ax_{j,s}})/(e^{ax_{j,s}} + e^{-ax_{j,s}}), \quad (2.4)$$

$$y_j = \max(0, x_{j,s}), \quad (2.5)$$

де $x_{j,s}$ - сумарний вхідний сигнал для j -го нейрону;

y_j – вихідний сигнал для j -го нейрону;

a – параметр швидкості навчання;

d – фіксований коефіцієнти.

Сумарний вхідний сигнал для j -го нейрону розраховується так:

$$x_{j,s} = \sum_{i=0}^I w_{j,i} x_{j,i}, \quad (2.6)$$

де $w_{j,i}$ – ваговий коефіцієнт для i -го вхідного зв'язку j -го нейрону;

$x_{j,i}$ – i -ий вхідний сигнал j -го нейрону;

I – кількість вхідних зв'язків для j -го нейрону.

Використання функції активації ReLU пояснюється простотою розрахунку її значення, кращим поширенням градієнту в порівнянні з функціями активації типу сигмоїд та гіперболічний тангенс. В сучасних версіях багат шарового перцептрон у нейронах вихідного шару може використовуватись функція активації типу softmax:

$$y_j = e^{x_{j,s}} / Q(x_{j,s}), \quad (2.7)$$

$$Q(x_{j,s}) = \sum_{j=1}^J e^{x_{j,s}}, \quad (2.8)$$

де J - кількість нейронів у вихідному шарі.

Використання функції softmax пояснюється можливістю інтерпретації вихідного сигналу у вигляді ймовірності. В протилежному випадку для цього необхідно застосувати окремий блок, що не входить до складу моделі.

Загальна кількість схованих нейронів визначається так:

$$N_H = \eta \sqrt{\frac{P}{J}}, \quad (2.9)$$

де P – кількість навчальних прикладів;

$\eta = [1..10]$ – коефіцієнт пропорційності.

Кількість нейронів у кожному схованому шарі розраховується так:

$$N_h = N_H / H. \quad (2.10)$$

Мінімальна достатня кількість навчальних прикладів визначається так:

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		23

$$P_{min} = 10 \times J \times N_x . \quad (2.11)$$

При мінімально допустимій кількості навчальних прикладів загальна кількість схованих нейронів розраховується так:

$$N_{Hmin} = \eta \sqrt{P_{min}/J} . \quad (2.12)$$

Кількість нейронів у кожному схованому шарі при мінімально допустимій кількості навчальних прикладів розраховується так:

$$N_{hmin} = N_{Hmin}/H . \quad (2.13)$$

На відміну від багат шарового перцептронну згорткова нейронна мережа адаптована до розпізнавання образів, що можуть бути представлені у вигляді квадратних зображень. Суть адаптації полягає в переході від самого об'єкту, який розпізнається, до його абстракції, виділення певних ознак і їх дослідження. Відповідні зміни отримала структура НММ та математичне забезпечення. Типова структура згорткової нейронної мережі, призначена для розпізнавання зображень показана на рис. 2.4 [6, 18].

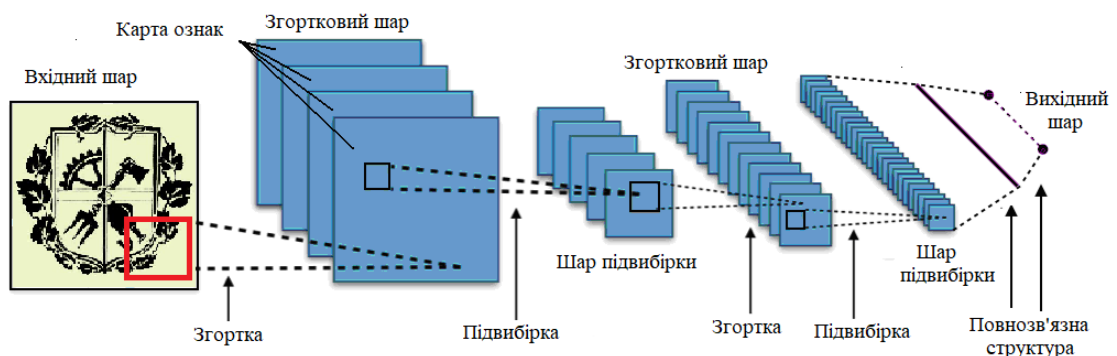


Рис. 2.4 Структура згорткової нейронної мережі типу LeNet-5

Вхідні параметри такої нейромережевої моделі відповідають окремим

пікселям. Кількість вихідних нейронів дорівнює кількості розпізнаваних образів, а кількість прихованих нейронів підбирається експериментальним шляхом. По суті згорткові мережі представляють багатошаровий перцептрон, що складається з згорткових шарів, шарів підвибірки і повнозв'язних шарів. Шари підвибірки також називають шарами субдискретизації або пулінгу. Додатково можуть бути використані шари інших типів. Сумарний вхідний сигнал нейрона згорткового шару розраховується так:

$$x_k^{(i,j)} = b_k + \sum_{s=1}^t \sum_{t=1}^K w_{k,s,t} x^{((i-1)+s,(j+t))}, \quad (2.14)$$

де $x_k^{(i,j)}$ - вхідний сигнал (i, j) -го нейрона k-ої карти ознак;

b_k - зміщення нейронів k-ої карти ознак;

K - розмір рецептивної області нейрона (розмір ядра згортки);

$w_{k,s,t}$ - ваговий коефіцієнт (s, t)-ої зв'язку нейрона k-ої карти ознак;

x - вихід нейрона попереднього шару.

Вихідний сигнал нейрона карти ознак розраховується шляхом підстановки вхідного сигналу в функцію активації:

$$y = f(x). \quad (2.15)$$

Досить часто в якості функції активації використовується гіперболічний тангенс, що заданий виразом виду (2.4). Шари субдискретизації, які слідують після згорткових, зменшують розмірність карти ознак. Вихідний сигнал нейрона шару субдискретизації розраховується так:

$$y_k^{(i,j)} = b_k + 0,25w_k \sum_{s=1}^2 \sum_{t=1}^2 x^{((i-1)+s,(j+t))}. \quad (2.16)$$

Вираз (2.16) отриманий за умов використання механізму пулінгу при

реалізації масштабування карти ознак. Після всіх операцій згортки і субдискретизації матриця, що описує вихідне зображення вироджується в вектор, котрий відповідає нейронам останнього схованого шару нейронів.

На основі результатів теоретичних робіт, присвячених згортковим нейронним мережам [6, 17-19] визначено, що основними структурними параметрами даного типу нейромережевої моделі є:

- Кількість вхідних нейронів - L_{in} .
- Кількість вихідних нейронів - L_{out} .
- Кількість нейронів в повнозв'язному шарі - L_f .
- Кількість згорткових шарів - K_{ls} .
- Кількість карт ознак в кожному згортковому шарі - $K_{h,k}, k \in [1, K_{ls}]$.
- Кількість шарів підвибірки (субдискретизації) - K_{ld} .
- Розмір ядра згортки для k-го згорткового шару $(b \times b)_k, k \in [1, K_{ls}]$.
- Зміщення рецептивного поля при виконанні кожної k-ої процедури згортки $d_k, k \in [1, K_{ls}]$. Прийнято, що $d_k = d = 1$.
- Розмір карти ознак для k-го згорткового шару - $(a \times a)_k, k \in [1, K_{ls}]$:

$$a_k = a_{k-1} - b_k + 1 . \quad (2.17)$$

- Структура зв'язків між сусідніми шарами згортки/підвибірки. Цю структуру можна представити у вигляді матриці:

$$Q_{i,i+1} = \left\| \begin{array}{ccc} q_{(i,1),(i+1,1)} & \cdots & q_{(i,1),(i+1,j)} \\ \cdots & \cdots & \cdots \\ q_{(i,G),(i+1,1)} & \cdots & q_{(i,G),(i+1,J)} \end{array} \right\|, \quad (2.18)$$

де G - кількість карт в i -му шарі;

J - кількість карт в $(i + 1)$ -му шарі;

$Q_{i,i+1}$ - матриця, компоненти якої визначають наявність зв'язків між i -им і $(i + 1)$ -им прихованими шарами;

$q_{(i,g),(i+1,j)}$ - компонент, який вказує на наявність/відсутність зв'язку між g-ою картою і-го шару і j-ою картою (i + 1) -го шару.

Якщо зв'язок між g-ою картою і-го шару і j-ою картою (i + 1)-го шару є, то $q_{(i,g),(i+1,j)} = 1$. У випадку відсутності зв'язку між g-ою картою і-го шару і j-ою картою (i + 1)-го шару вираз $q_{(i,g),(i+1,j)} = 0$.

З урахуванням необхідності мінімізації помилки розпізнавання модель оптимізації структурних параметрів згорткової нейронної мережі можна записати за допомогою виразу:

$$\Delta \left(L_{in}, L_{ls}, L_{out}, K_{h,k}, b_k, K_{ls}, |Q_{i,i+1}|_{K_{ls}} \right) \rightarrow \min, \quad (2.19)$$

де Δ - помилка розпізнавання;

$|Q_{i,i+1}|_{K_{ls}}$ - вектор, що складається з матриць які визначають зв'язки між сусідніми прихованими шарами нейронів.

Для виконання завдання дипломної роботи базова модель згорткової нейронної мережі типу LeNet-5 є малоприсадоною через великий обсяг пам'яті необхідний для її збереження, недостатньо високу точність розпізнавання та тривалий період навчання мережі.

2.3 Нейромережева модель для розпізнавання особи користувача

Squeezenet – це згорткова нейронна мережа, що є розвитком класичної згорткової мережі типу AlexNet, що в свою чергу являється вдосконаленим варіантом LeNet [7, 19]. Дана нейронна мережа проектувалась з метою підвищення обчислювальної ефективності розпізнавання зображень, що входять до складу БД ImageNet, розміром 227 на 227. Основним нововведенням в Squeezenet є використання шарів Squeeze, в яких ядро згортки має розмірність 1 на 1. В той же час класичні шари згортки також

використовуються в цій мережі. Структура мережі, візуалізована за допомогою вбудованої в програмний комплекс MATLAB R2018b функції plot, показана на рис. 2.5.

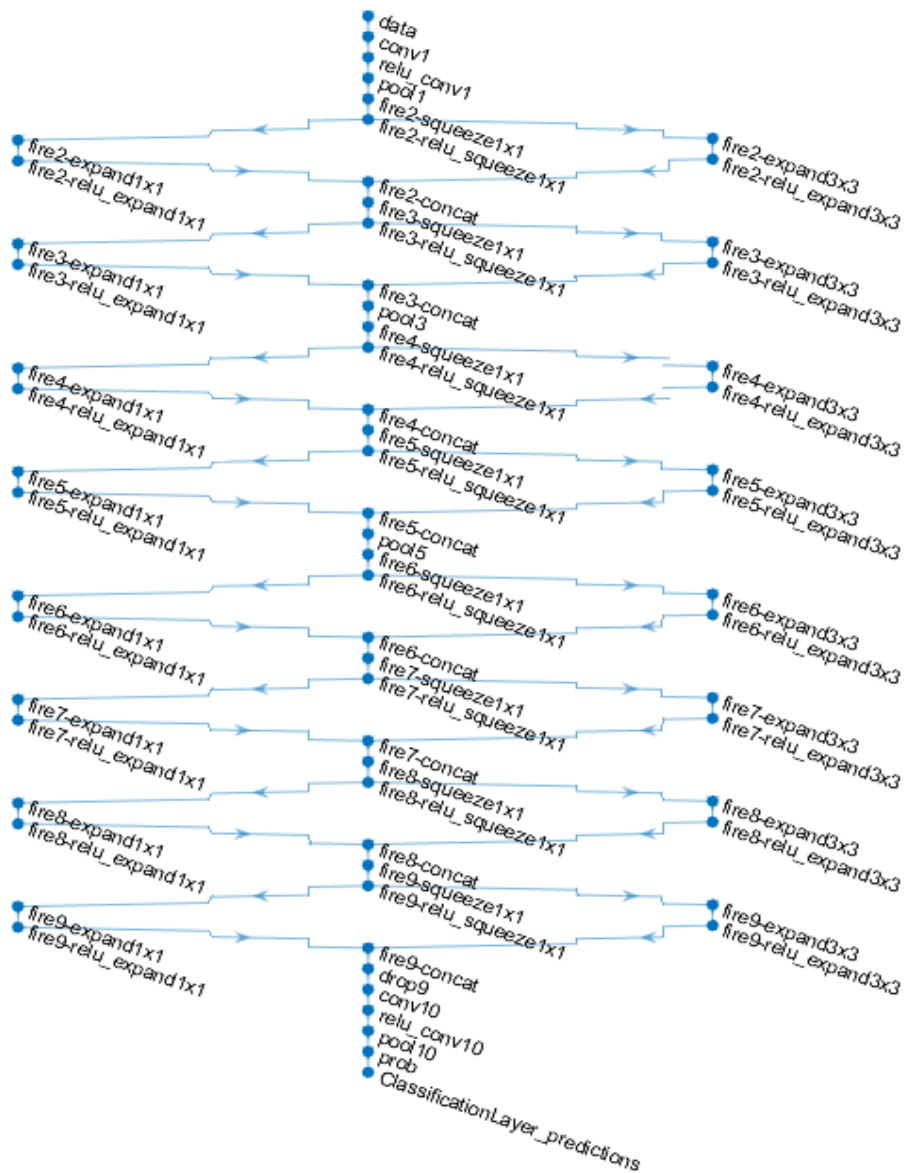


Рис. 2.5. Структурна схема нейромережевої моделі типу Squeezenet

Основними перевагами мережі Squeezenet є набагато менший розмір і в 50 разів менша кількість параметрів в порівнянні з AlexNet. Зазначимо, що обсяг пам'яті, котрий потрібен для збереження вагових коефіцієнтів мережі Squeezenet, навченої на прикладах із загальновідомої БД ImageNet, становить

менше 0,5 Мб. При цьому програмна реалізація навченої мережі Squeezenet знаходиться у вільному доступі, а програмний комплекс MATLAB дозволяє проводити її редагування. Крім того використання бібліотеки CUDA дозволяють розпаралелити процес функціонування Squeezenet за умови застосування графічних процесорів сімейства NVIDIA. В наслідок цього нейромережевий засіб, створений на основі мережі Squeezenet, характеризується порівняно невеликим обсягом споживаної пам'яті, високою швидкістю розпізнавання, доступністю переднавченої моделі, можливістю реалізації за допомогою апробованих засобів, а також достатньою точністю розпізнавання. До недоліків Squeezenet можна віднести дещо меншу точність розпізнавання в порівнянні з найбільш сучасними типами згорткових нейронних мереж (GoogleNet, VGG).

Нейронна мережа Squeezenet складається із 18 шарів нейронів. Вказані нейронні шари розділяються на такі типи: вхідний, згортковий, пулінгу, модуль fire, класифікації. В базовому варіанті вхідний шар має вхідний розмір зображень 227 на 227, що відповідає розміру і кількості каналів кольору зображень в базі даних ImageNet. Основним завданням вхідного шару є прийом даних із зовнішнього середовища в нейронну мережу. Нейрони даного шару мають лінійну функцію активації виду (2.2). Згортковий шар виконує роль згортки зображення і виявлення певних ознак на ньому. Математична модель функціонування нейронів в згортковому шарі визначається виразами (2.3-2.8, 2.15). Шар пулінгу призначений для зменшення кількості ознак, знайдених за допомогою шарів згортки. Розрахунок вихідного сигналу нейронів шару пулінгу здійснюється за допомогою виразу (2.16). Модуль fire має такий вигляд: згортковий squeeze шар (який має фільтри розмірності 1 на 1) і шар, в якому половина фільтрів є класичними згортковими, а інша половина є squeeze шарами. Шар класифікації має функцію активації виду (2.2). Цей шар призначений для сигналізації про результат розпізнавання.

Особливістю, як класичних, так і найбільш сучасних варіантів

					<i>ІАЛЦ.045420.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

згорткової нейронної мережі є необхідність представлення вхідної інформації у вигляді квадратного зображення. У базовому варіанті використовується чорно-біле зображення. Більш складні варіанти передбачають використання трьохвимірною сірого і кольорового зображення. Зазначена особливість накладає істотне обмеження на використання згорткової нейронної мережі - можливість аналізу клавіатурного на текстових фрагментах з фіксованою кількістю символів. При цьому загальна постановка задачі розпізнавання особи користувача передбачає необхідність аналізу як заздалегідь визначеного фрагмента тексту, так і фрагмента тексту довільного змісту. Перший випадок можна співвіднести з моніторингом особи користувача при введенні їм паролівних даних. Другий випадок співвідноситься з поточним моніторингом підтвердження особи користувача при введенні їм з клавіатури текстової інформації. Відповідно до даних [17-19], такий моніторинг можливий за рахунок аналізу клавіатурного почерку при введенні тексту фіксованої довжини. Таким чином, наведене обмеження не робить негативного впливу на функціональність мережі. У першому випадку кількість символів в паролівних даних і так є фіксованою величиною. У другому випадку обсяг фрагмента тексту обмежується заздалегідь визначеною кількістю символів.

Розглянемо принцип кодування параметрів клавіатурного почерку в чорно-біле квадратне зображення, на якому базується запропонована процедура кодування. Пропонується вісь ординат співвіднести з розкладкою клавіатури - ASCII-кодами клавіш або символами, що відповідають клавішам. Ось абсцис пропонується співвіднести з введенням текстом. Таким чином, одній окремій точці зображення відповідатиме окремий символ введеного тексту. По осі абсцис координата закодованого символу відповідає позиції (номеру) даного символу в тексті. Координата по осі ординат відповідає позиції символу на клавіатурі/ASCII-коду попереднього символу в тексті. При цьому вважається, що перший символ на клавіатурі відповідає символу пробілу. У випадку, коли кількість символів тексту буде більше, ніж кількість

лініями. Кожна залита чорним кольором точка зображення відповідає Закодованому значенню символу тексту. Наприклад, символу «Т» відповідає чорна точка зображення, що знаходиться на перетині вертикалі, проведеної з позиції №3 («Т») по осі абсцис, і горизонталі, проведеної з позиції №22 («U») осі ординат. Це пояснюється тим, що в слові «AUTOMOBILE» буква «Т» слід за буквою «U». У числовому вигляді даний рисунок являє собою квадратну матрицю. Залиті чорним кольором точки малюнка відповідають елементам матриці рівним 1, а світлі точки - елементам матриці, рівним 0.

Базуючись на результатах [17-19], визначено доцільність використання показників ТУК, ТМК, відношення ТУК до ТМК (ВТТ), динаміки терміну утримання клавіші (ДТУК) та динаміки терміну між послідовним натисненням двох клавіш (ДТМК). Розрахунок вказаних показників реалізується так:

$$y_r(i) = t_d(i) - t_u(i), \quad (2.20)$$

$$y_b(i, i - 1) = t_u(i) - t_d(i - 1), \quad (2.21)$$

$$g(i, i - 1) = y_b(i, i + 1)/y_r(i), \quad (2.22)$$

$$w_r(i, i - 1) = (y_r(i) - y_r(i - 1))/y_r(i), \quad (2.23)$$

$$w_b(i, i - 1) = (y_b(i) - y_b(i - 1))/y_b(i), \quad (2.24)$$

де y_r - ТУК;

t_d - час натискання клавіші;

t_u - час відпускання клавіші;

y_b - ТМК;

w_r - ДТУК;

w_b - ДТМК;

g – ВТТ;

i - номер натискання клавіші при введенні тексту.

Зауважимо складність використання t_d і t_u , яка пояснюється необхідністю їх реєстрації з точністю до мілісекунди. Для цього розроблена Windows-орієнтована програма KeyboardProgram, що дозволяє реєструвати значення t_d і t_u з помилкою, яка дорівнює тривалості 50 тактів процесора.

Використання даної програми дозволило встановити, що значення ТУК і ТМК не перевищують 500 мс. ТУК і ТМК, що перевищує цю величину, повинні бути відфільтровані. Слід відзначити значну залежність ТУК і ТМК від типу клавіатури. Так, за даними [19], на клавіатурі з коротким ходом клавіш (клавіатура ноутбука) в середньому $y_r = 100$ мс, $y_d = 150$ мс, а для клавіатури з довгим ходом клавіш (стандартна клавіатура) $y_r = 150$ мс, $y_d = 200$ мс. Тому нормалізація ТУК і ТМК до середніх значень може спричинити за собою помилки, пов'язані з характеристиками використовуваної клавіатури. Таким чином, на вхід в згорткову нейронну мережу доцільно подавати абсолютні відфільтровані значення параметрів клавіатурного почерку. Введений символ в першому наближенні пропонується представити у вигляді відповідного ASCII-коду.

Кодування введеного тексту в вигляді чорно-білого зображення не дозволяє реалізувати аналіз клавіатурного почерку, яке передбачає аналіз, як мінімум одного з основних параметрів клавіатурного почерку. Тому запропонована процедура кодування передбачає подання введеного тексту в вигляді кольорового квадратного зображення з багатоканальним растром. У базовому випадку процедура передбачає два растрових канали. Кожна точка такого зображення повинна характеризувати введений символ і один з параметрів клавіатурного почерку, який відноситься до цього символу. В якості ілюстрації результатів описаної процедури кодування на рис. 2.7 фрагментарно представлено відображення тексту «HEY THERE», закодованого з використанням ТУК. На відміну від рис. 2.6, кожна точка

зображення, що відповідає закодованому значенню символу тексту, характеризується двома цифрами, записаними у відповідній клітинці. Перша цифра - це ASCII-код введеного символу, а друга - ТУК. Наприклад, символу «Y» відповідає точка зображення, що знаходиться на перетині третього стовпчика з шостим рядком. У відповідній клітинці представлені значення 121 (ASCII-код) і 51 (ТУК).

Z	27												
Y	26			32;51									
X	25												
W	24												
V	23												
U	22												
T	21					104;37							
S	20												
R	19								101;36				
Q	18												
P	17												
O	16												
N	15												
M	14												
L	13												
K	12												
J	11												
I	10												
H	9		101;55					101;44					
G	8												
F	7												
E	6			121;51					114;64				
D	5												
C	4												
B	3												
A	2												
	1	104;49				116;52							
		1	2	3	4	5	6	7	8	9	10	11	12
		H	E	Y		T	H	E	R	E			

Рис. 2.7. Відображення тексту «HEY THERE», закодованого з використанням параметра ТУК у вигляді зображення з двоканальним растром

Варіант процедури кодування введеного тексту в вигляді кольорового квадратного зображення з багатоканальним растром є розвитком базового випадку в напрямку додавання каналів, які відповідають параметрам клавіатурного почерку. Так на рис. 2.8 фрагментарно показано відображення тексту «SECOND», закодованого з використанням ТУК і ТМК. Як і в попередніх випадках кожна точка зображення відповідає одному закодованому значенню символу тексту, однак характеризується трьома

цифрами, записаними у відповідній клітинці. Перша цифра - це ASCII-код введеного символу, друга – ТУК, третя - ТМК. Наприклад, символу «Е» відповідає точка зображення, що знаходиться на перетині другого стовпчика з двадцятим рядком. У відповідній клітинці представлені значення 101 (ASCII-код), 38,3 (ТУК), 78,5 (ТМК). Результат кодування в числовому вигляді представляє собою багатовимірну матрицю, глибина якої дорівнює кількості параметрів, які використовуються при аналізі клавіатурного почерку.

z	27						
y	26						
x	25						
w	24						
v	23						
u	22						
t	21						
s	20	101; 38.3; 78.5					
r	19						
q	18						
p	17						
o	16				110; 51.4; 23.8		
n	15					100; 41.7; 27.6	
m	14						
l	13						
k	12						
j	11						
i	10						
h	9						
g	8						
f	7						
e	6		99; 62; 22				
d	5						
c	4			111; 48.6; 18.9			
b	3						
a	2						
1	1	115; 42.7; 20.4					
		1	2	3	4	5	6
		s	e	c	o	n	d

Рис. 2.8. Відображення тексту «SECOND», закодованого з використанням параметрів ТУК і ТМК у вигляді зображення з трьохканальним растром

На основі результатів [17-19] прийнято, що ефективність розробленої нейромережевої моделі доречно оцінювати за допомогою значень параметрів точності розпізнавання (*Accuracy*) та втрат (*Loss*):

$$Accuracy = \frac{N_{right}}{N} \times 100\%, \quad (2.25)$$

$$Loss = \frac{1}{N} \sum_{t=1}^N e^T(t, Q)W(\theta)e(t, Q), \quad (2.26)$$

де N_{right} - кількість правильно розпізнаних прикладів;

N – загальна кількість прикладів;

$e(t, Q)$ – вектор помилки в момент t розмірністю n_y на 1 з параметром Q ;

n_y – кількість виходів нейронної мережі;

$W(Q)$ – матриця вагових коефіцієнтів вхідних зв'язків нейронів.

Висновки до розділу 2

Другий розділ присвячено створенню способу розробки системи розпізнавання особи користувача за клавіатурним почерком. В результаті проведених досліджень:

- За допомогою мови проектування UML, розроблені моделі даних системи розпізнавання особи користувача за клавіатурним почерком. Побудовані UML-діаграми прецедентів, пакетів, компонентів та класів, що забезпечило деталізований опис та документування системи розпізнавання з орієнтацією на її наступну реалізацію у вигляді програмного забезпечення. Таким чином, реалізовано перші два етапи життєвого циклу каскадної моделі програмного забезпечення, що полягають у формулюванні вимог та у проектуванні структури програмного забезпечення.

- Розроблено базову нейромережеву модель розпізнавання особи користувача за клавіатурним почерком, що дозволило визначити конструктивні параметри, необхідні для адаптації моделі до умов поставленої задачі.

- Розроблена малоресурсна нейромережева модель типу Squeezenet, адаптована до задачі розпізнавання особи користувача за клавіатурним почерком. Особливістю розробленої моделі став модуль представлення параметрів клавіатурного почерку у вигляді квадратного зображення з трьома каналами кольору.

					<i>ІАЛЦ.045420.003 ПЗ</i>	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

Для програмної реалізації системи розпізнавання особи користувача за клавіатурним почерком обрано мови програмування C++, Python та MATLAB. Вибір трьох мов програмування пояснюється тим, що програмне забезпечення системи розпізнавання повинно вирішувати три специфічні та достатньо різнопланові задачі:

- Реєстрації параметрів клавіатурного почерку.
- Представлення параметрів клавіатурного почерку у вигляді багатовимірної матриці.
- Побудову нейромережевої моделі типу SqueezeNet, що свою чергу передбачає реалізацію архітектури нейромережевої моделі, її навчання та застосування для розпізнавання особи користувача.

Відповідно наведеним задачам використання мов програмування пояснюється так:

- C++ забезпечує зручний та апробований доступ до API-функцій операційної (ОС), що дозволило реалізувати реєстрацію параметрів клавіатурного почерку з точністю до 50 тактів процесора, що вважається достатнім для розпізнавання особи користувача [19].
- Python забезпечує доступ до розвиненого інструментарію для роботи з багатовимірними масивами даних великого обсягу.
- MATLAB забезпечує можливість використання переднавченої моделі SqueezeNet, що значно спрощує реалізацію моделі та зменшує вимоги до формування баз даних параметрів клавіатурного почерку.

Програмне забезпечення на C++ створене в середовищі розробки Visual Studio, на Python - в середовищі розробки PyCharm, а на MATLAB - в одноіменному середовищі розробки.

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		37

Рекомендовані системні вимоги до апаратного забезпечення системи розпізнавання особи користувача: процесор Intel або AMD x86-64 з частотою 3,8-4,2 ГГц, обсяг оперативної пам'яті 8 ГБ, відеокарта з підтримкою OpenGL 3.3 та 1GB GPU, SSD з обсягом вільної пам'яті 8 ГБ. Інші характеристики не мають вагомого впливу для коректної роботи з програмою.

3.1. Опис функціональних можливостей програмного забезпечення

Базуючись на результатах проведених досліджень визначено, що програмне забезпечення системи розпізнавання особи користувача за клавіатурним почерком повинно надавати наступні сервіси:

- Визначення множини параметрів, що характеризують клавіатурний почерк.
- Переключення між режимами зняття параметрів клавіатурного почерку.
- Зняття параметрів клавіатурного почерку, їх запис і збереження у БД.
- Перетворення кожного із векторів зареєстрованих параметрів клавіатурного почерку у відповідну матрицю та збереження цієї матриці у БД.
- Формування із матриць відповідних зображень та їх запис і збереження у БД.
- Створення нейронної мережі, редагування параметрів її навчання, проведення процедури навчання та збереження параметрів навченої мережі.
- Аналіз параметрів клавіатурного почерку за допомогою навченої нейронної мережі та сигналізація про результати розпізнавання.

Відповідно побудованих UML-діаграм з використанням об'єктно-орієнтованого підходу визначено доцільність розділення програмного забезпечення на три окремі класи:

- Клас *KeyboardProgram* призначений для реєстрації параметрів клавіатурного почерку та збереження зареєстрованих даних.

- Клас *FormImagePy* призначений для обробки параметрів клавіатурного почерку та їх подання у вигляді матриць.
- Клас *Pr2_FormImages* призначений для формування зображення із матриці параметрів клавіатурного почерку.
- Клас *Pr2* призначений для навчання нейронної мережі.
- Клас *Pr3* призначений для застосування навченої нейронної мережі для розпізнавання особи користувача.

Наведемо коротку характеристику функціональності кожного із означених класів.

Клас *KeyboardProgram* забезпечує інструментарій для зняття параметрів клавіатурного почерку. До атрибутів класу *KeyboardProgram* відносяться: *file*, *keys*, *keysAscii*, *timePressed*, *timeBetweenPressed*, *timeStart*, *timeFinished*, *StartTime*. *EndTime*. Методами класу *KeyboardProgram* являються: *SetConsoleCursorPosition*, *GetNumberOfConsoleInputEvents*, *ReadConsoleInput*, *bKeyDown*, *dwControlKeyState*. Призначення атрибутів класу *KeyboardProgram*:

- *file* – змінна, за допомогою якої відбувається запис зареєстрованій параметрів клавіатурного почерку користувача у БД.
- *keys* – масив, в якому зберігаються символи, що відповідають натисненим клавішам.
- *keysAscii* – масив, в якому зберігаються ASCII-код кожної із натиснутих користувачем клавіш.
- *timePressed* – масив, в якому зберігається час затиснення кожної клавіші в мілісекундах.
- *timeBetweenPressed* – масив, в якому зберігається час між натисканнями попередньої і наступної клавіші.
- *timeStart* – масив, в якому зберігається час натискання на клавішу.
- *timeFinished* – масив, в якому зберігається час кінця затискання кожної клавіші.

- *StartTime* – змінна для зберігання початку натискання клавіші.
- *EndTime* – змінна для зберігання кінця натискання клавіші.

Призначення методів класу *KeyboardProgram*:

- *SetConsoleCursorPosition* – для встановлення каретки вводу.
- *GetNumberOfConsoleInputEvents* – для перевірки, чи є незчитані записи в буфер записів консолі.
- *ReadConsoleInput* – для зчитування символу, що відповідає натиснутій клавіші.
- *bKeyDown* – для фіксації часу затискання клавіші.
- *dwControlKeyState* – для перевірки стану затиснення клавіші.

Клас *FormImagePy* забезпечує інструментарій для попередньої обробки параметрів клавіатурного почерку для їх представлення у вигляді матриць. До атрибутів класу *FormImage* відносяться: *sizeWork*, *numChannel*, *workList*, *file*.

Методи класу *FormImage*: *open*, *close*, *glob*, *process*. Призначення атрибутів:

- *sizeWork* – містить в собі розмірність каналу зображення.
- *numChannel* - містить в собі кількість каналів зображення.
- *workList* – для масиву, який являє собою матрицю параметрів клавіатурного почерку.
- *file* – змінна для зберігання імені файлу із параметрами клавіатурного почерку користувача.

Призначення методів:

- *open* – для відкриття файлу для зчитування параметрів клавіатурного почерку.
- *close* - для закриття файлу.
- *glob* – для ітераційного доступу до кожного файлу із заданого шляху до директорії, яка містить файли із параметрами клавіатурного почерку користувача.
- *process* – метод для обробки зчитаних параметрів клавіатурного почерку, що забезпечує формування множини вхідних нейронів мережі. Блок-

схема алгоритму даного методу показана в додатку 1.

Клас *Pr2_FormImages* забезпечує інструментарій для представлення матриць параметрів клавіатурного почерку у вигляді кольорового зображення, що придатне для подачі в згорткову нейронну мережу типу *SqueezeNet*. До атрибутів класу *Pr2_FormImages* відносяться: *workList*, *file*. Методи класу *Pr2_FormImages*: *open*, *close*, *image*. Призначення атрибутів:

- *workList* – для масиву, який являє собою матрицю параметрів клавіатурного почерку.

- *file* – змінна для зберігання імені директорії в якій знаходяться файли із матрицями в яких представлені параметрами клавіатурного почерку користувача.

Призначення методів:

- *open* – для відкриття файлу для зчитування параметрів клавіатурного почерку.

- *close* - для закриття файлу.

- *image* – метод для перетворення матриці параметрів клавіатурного почерку в зображення.

Клас *Pr2* забезпечує інструментарій для побудови та навчання згорткової нейронної мережі типу *SqueezeNet* для аналізу клавіатурного почерку. До атрибутів класу *Pr2* відносяться: *net*, *I*, *options* та *lgraph*. Методи класу *Pr2*: *layerGraph*, *convolution2dLayer*, *reluLayer*, *averagePooling2dLayer*, *softmaxLayer*, *classificationLayer*, *replaceLayer*, *trainingOptions*, *trainNetwork*.

Призначення атрибутів:

- *net* - представляє із себе об'єкт, котрий містить в собі завантажену модель стандартної нейронної мережі Squeezenet із програмного комплексу MATLAB.

- *I* - містить в собі зображення, сформоване із матриці параметрів клавіатурного почерку.

- *options* – масив, що містить у собі необхідні параметри для навчання

нейронної мережі.

- *lgraph* – об’єкт, який містить в собі параметри архітектури побудованої нейронної мережі.

Призначення методів:

- *layerGraph* – функція, яка присвоює об’єкту *lgraph* параметри архітектури завантаженої мережі.

- *convolution2dLayer* - для формування шару згортки шар розмірністю 1 на 1, що являється Squeeze шаром.

- *reluLayer* - для формування шару, який застосовує функцію ReLU для кожного із входів.

- *averagePooling2dLayer* - для формування шару пулінгу.

- *softmaxLayer* - для формування шару, який застосовує функцію softmax для кожного із входів.

- *classificationLayer* - для формування шару класифікації.

- *replaceLayer* – функція, за допомогою якої створені шари додаються до архітектури завантаженої мережі.

- *trainingOptions* – функція, за допомогою якої задаються параметри для навчання нейронної мережі, які записуються в масив *options*.

- *trainNetwork* - запускається процес навчання нейронної мережі, використовуючи параметри, задані функцією *trainingOptions* в масиві *options*. Означений процес навчання реалізується відповідно алгоритму, блок-схема якого показана на плакаті, наведеному в додатку 1.

Клас *Pr3* забезпечує інструментарій для застосування згорткової нейронної мережі типу *SqueezeNet* для аналізу клавіатурного почерку. До атрибутів класу *Pr3* відносяться: *net*, *I*. Методи класу *SqueezeNetPr*: *setOptions*, *classify*. Призначення атрибутів:

- *net* - представляє із себе об’єкт, котрий містить в собі модель навченої нейронної мережі *netTransfer*.

- *I* - містить в собі зображення, сформоване із матриці параметрів клавіатурного почерку.

Призначення методів:

- *setOptions* - визначає параметри розпізнавання.
- *classify* - за допомогою даної функції навчена нейронна мережа розпізнає особу користувача.

Блок-схема алгоритму функціонування мережі SqueezeNet показана на плакаті, наведеному в додатку 1.

3.2. Опис інтерфейсу користувача

Побудова інтерфейсу користувача проведена з позицій відповідності вимогам простоти та зручності використання розробленого програмного забезпечення запропонованої системи розпізнавання особи користувача за клавіатурним почерком.

Запуск програмного додатку здійснюється за допомогою файлу Main.py. У відповідь на запуск відкривається вікно головної форми програмного додатку, показане на рис. 3.1.

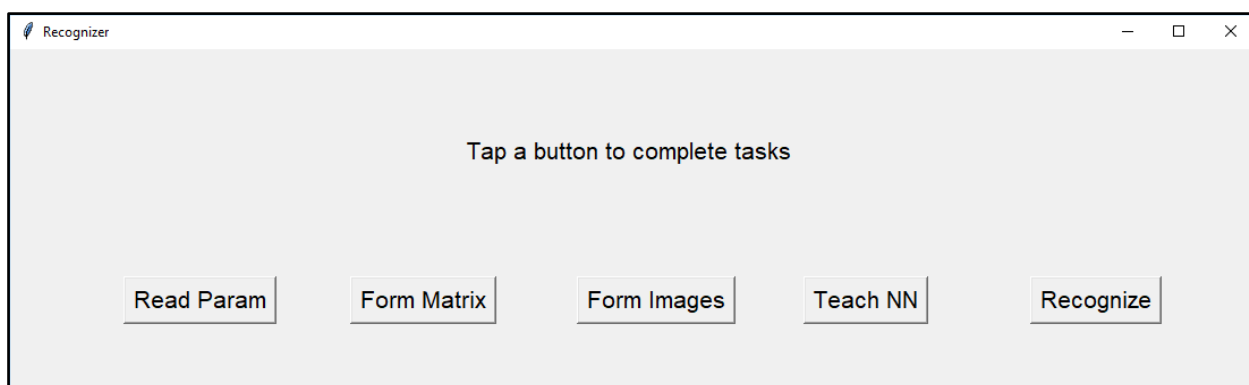


Рис. 3.1. Головне вікно програмного додатку Recognizer

На головній формі розташовані 5 кнопок: «Read Param», «Form Matrix», «Form Images», «Teach NN» та «Recognize». Функціонал використання вказаних кнопок:

- Кнопка «Read Param» - запуск модулю KeyboardProgram, що зчитує та зберігає параметри клавіатурного почерку.

- Кнопка «Form Matrix» - запуск модулю FormImagePy, що призначений для формування матриць параметрів клавіатурного почерку та їх збереження.

- Кнопка «Form Images» - запуск модулю Pr2_FormImages, що формує зображення із матриці параметрів клавіатурного почерку.

- Кнопка «Teach NN» - запуск модулю Pr2, що призначений для навчання нейронної мережі.

- Кнопка «Recognize» - запуск модулю Pr3, функціональність якого стосується застосування навченої нейронної мережі для розпізнавання особи.

Розглянемо функціонал та використання модулів, що спрацьовують у відповідь на натиснення вказаних кнопок.

Після натиснення кнопки «Read Param» активується модуль KeyboardProgram, що призводить до відкриття відповідного вікна, показаного на рис. 3.2. Передбачено, що користувач повинен ввести певний текст. При цьому починається реєстрація параметрів клавіатурного почерку, що подалі відображаються у цьому ж вікні. До основних параметрів, що будуть відображені відносяться загальний термін роботи програми KeyboardProgram, пауза між натисненням даної і попередньої клавіші, введений символ.



Рис. 3.2. Початок роботи програмного модулю KeyboardProgram

Для прикладу на рис. 3.3. показано параметри клавіатурного почерку, що стосуються натиснення клавіші, яка відповідає букві “d”. Як показано на даному рисунку загальний термін роботи програми складає 27,913 с, пауза між натисненням даної і попередньої клавіші – 1,568 с. Для завершення вводу тексту користувачеві слід натиснути клавішу ENTER.

```

E:\Working_project\KeyboardProgram\Release\KeyboardProgram.exe
Key Events : 11key loop counter: 1568
27913      _Or in miliseconds: 156.8
Here is 1loop counter:: 17307m
d

```

Рис. 3.3. Параметри натиснення клавiші букви “d”

У відповідь у вікні програми відображаються всі зареєстровані параметри клавіатурного почерку. Для прикладу на рис. 3.4. показано параметри клавіатурного почерку, що стосуються тексту “hello world”. Зареєстровані параметри також записуються в текстовий файл.

```

Microsoft Visual Studio Debug Console
Key Events : 12key loop counter: 8688
key loop counter in miliseconds: 86.88
Exiting...oop counter:: 43537m
There r all keys:
h ,ascii value: 104 , time: 116.7 , time between pressing: 0 , key down: 2200 , Key up: 2304
e ,ascii value: 101 , time: 109.6 , time between pressing: 18.4 , key down: 2890 , Key up: 3026
l ,ascii value: 108 , time: 86.4 , time between pressing: 15.1 , key down: 3858 , Key up: 4007
l ,ascii value: 108 , time: 86.9 , time between pressing: 15.4 , key down: 4634 , Key up: 4802
o ,ascii value: 111 , time: 186.7 , time between pressing: 17.7 , key down: 6717 , Key up: 6830
, ,ascii value: 32 , time: 85.3 , time between pressing: 16.2 , key down: 8430 , key up: 8641
w ,ascii value: 119 , time: 86.4 , time between pressing: 15.4 , key down: 9707 , Key up: 9896
o ,ascii value: 111 , time: 184.3 , time between pressing: 25.6 , key down: 10567 , Key up: 10682
r ,ascii value: 114 , time: 86.4 , time between pressing: 14.7 , key down: 11339 , Key up: 11501
l ,ascii value: 108 , time: 85.9 , time between pressing: 17.6 , key down: 12001 , Key up: 12133
d ,ascii value: 100 , time: 156.8 , time between pressing: 27.9 , key down: 17197 , Key up: 17307
, ,ascii value: 13 , time: 86.8 , time between pressing: 15.3 , key down: 43432 , Key up: 43537

E:\Working_project\KeyboardProgram\Release\KeyboardProgram.exe (process 304) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```

Рис. 3.4. Зареєстровані параметри клавіатурного почерку

При натисненні у головному вікні програми (рис. 3.1.) кнопки «Form Matrix» активується програмний модуль FormImagePy, що призначений для представлення зареєстрованих параметрів клавіатурного почерку у вигляді багатовимірної матриці. Вікно модулю FormImagePy показано на рис. 3.5.

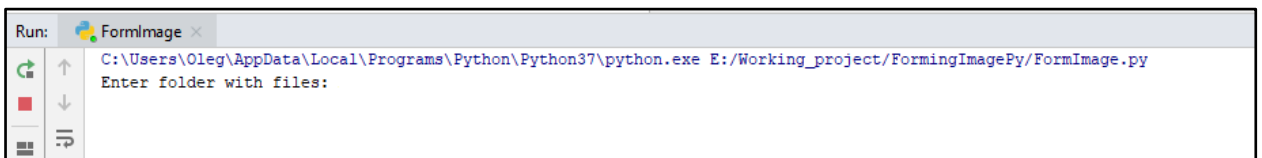


Рис. 3.5. Вікно програмного модулю FormImage

У верхній частині вікна відображається службова інформація, що

стосується програми FormImagePy. Подалі після тексту «Enter folder with files» користувач повинен ввести ім'я директорії в якій зберігаються текстові файли з записаними параметрами клавіатурного почерку. Для прикладу на рис. 3.6. показано, що користувач ввів ім'я директорії «Me». Після вводу імені директорії користувач повинен натиснути клавішу ENTER.

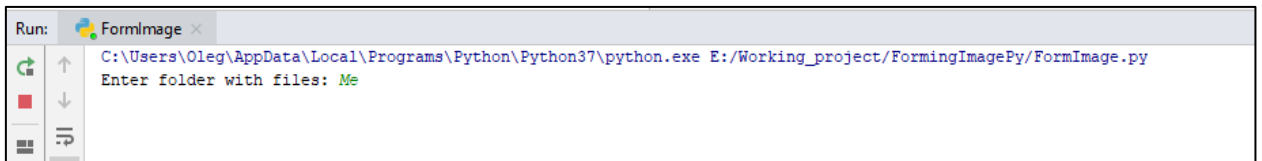


Рис. 3.6. Ввід шляху до директорії параметрів користувача Me

Результатом роботи модулю FormImagePy є матриці параметрів клавіатурного почерку. Вказані матриці відображаються у вікні програми та призначені для подальшого використання модулем Pr2_FormImages. Для прикладу на рис. 3.7. фрагментарно показано відображення декількох матриць клавіатурного почерку при вводі тексту «i like milk».

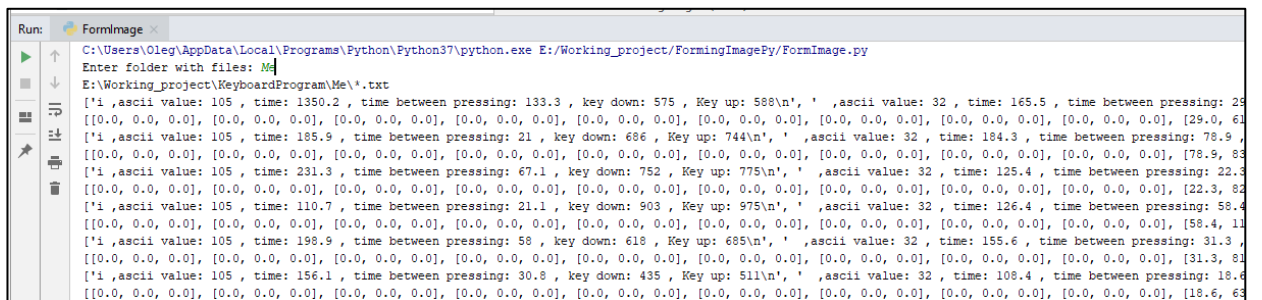


Рис. 3.7. Приклад матриць параметрів клавіатурного почерку

При натисненні у головному вікні програми (рис. 3.1.) на кнопку «Form Images» запускається програмний модуль Pr2_FormImages, першочергове вікно якого показане на рис. 3.8. Модуль запускається в режимі очікування вводу директорії в якій містяться файли з збереженими матрицями параметрів клавіатурного почерку.

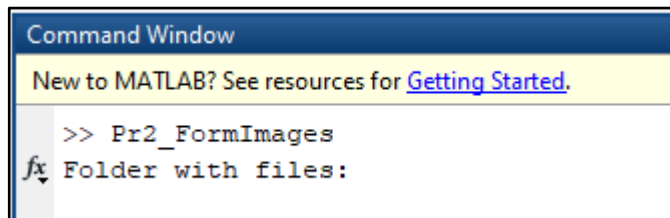


Рис. 3.8. Першочергове вікно модулю Pr2_FormImages

Після вводу директорії користувач натискає клавішу ENTER, що призводить до запуску процесу обробки матриць параметрів клавіатурного почерку і їх перетворення у трьох-канальні зображення. На рис. 3.9. відображається приклад вводу директорії користувача 'Oleg'.

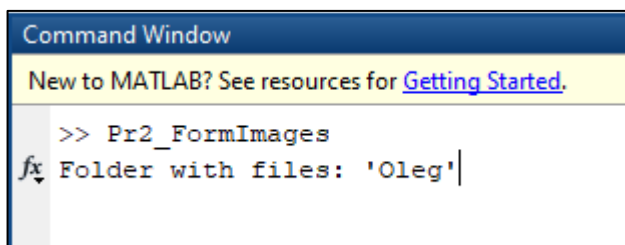


Рис. 3.9. Ввід директорії користувача Oleg

Після натискання клавіші ENTER програмний модуль Pr2_FormImages перетворює матриці параметрів клавіатурного почерку користувача, ім'я якого було введено в поле вікна програмного модулю, в трьох-канальні зображення у форматі bmp. Дані зображення збережені в однойменній директорії програмного модулю. На рис. 3.10. показано зображення параметрів клавіатурного почерку користувача 'Oleg'.

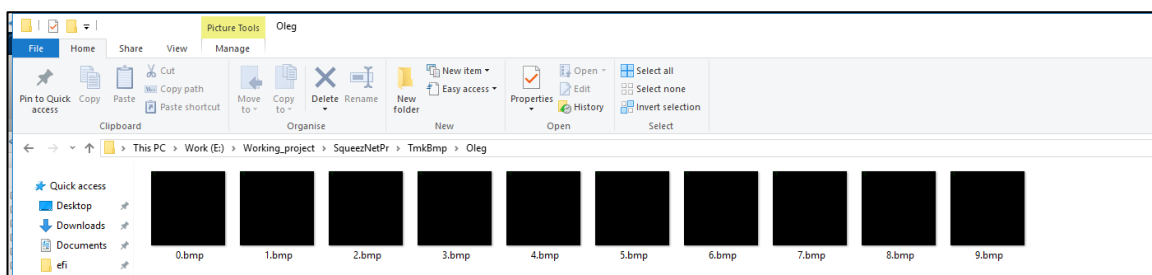


Рис. 3.10. Вміст директорії зі створеними зображеннями

При натисканні кнопки “Teach NN” в головному вікні програми (рис. 3.1) запускається програмний модуль Pr2. В вікні даного програмного модулю (рис. 3.11.) оператор системи вказує директорію із навчальними прикладами і натискає клавішу ENTER. Навчальні приклади представляють із себе директорії, які містять трьох-канальні зображення. Імена директорій відповідають іменам користувачів, на параметрах клавіатурного почерку яких навчається нейронна мережа.

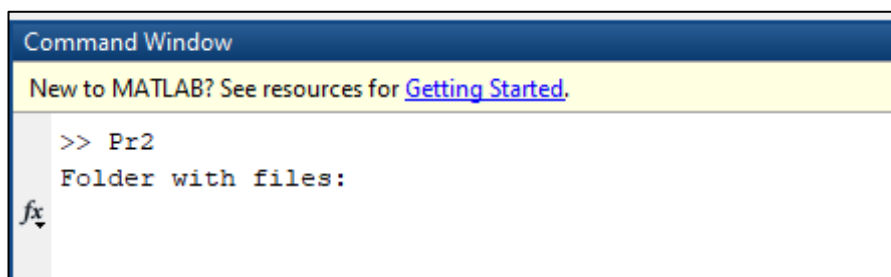


Рис. 3.11. Початкове вікно програмного модулю Pr2

Для прикладу на рис. 3.12. показано, що оператор системи вказав директорію ‘TmkBmp’, яка містить в собі навчальні приклади для навчання нейронної мережі.

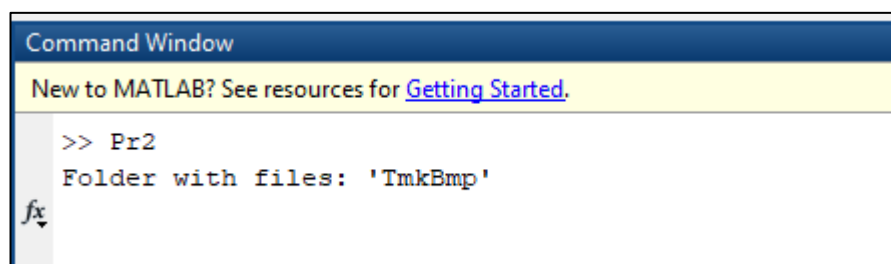


Рис. 3.12. Запуск програмного модулю Pr2

Після натискання клавіші ENTER програма виводить деяку інформацію про модель нейронної мережі (рис. 3.13), водночас починається процес навчання нейронної мережі, який також візуалізується за допомогою даного програмного модулю.

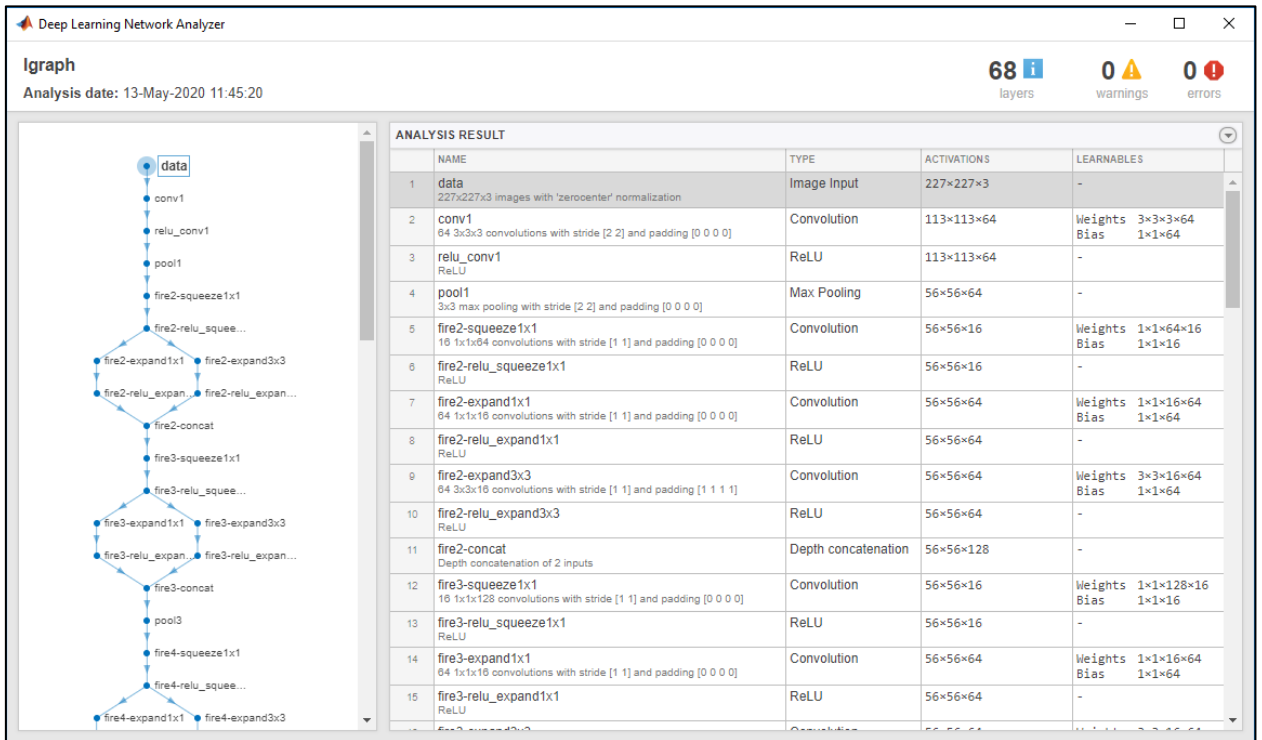


Рис. 3.13. Інформація про нейронну мережу

У вікні на рис. 3.14. показано два графіки: залежність точності розпізнавання та втрат при розпізнаванні на тренувальній і валідаційній вибірках від кількості навчальних ітерацій навчання.

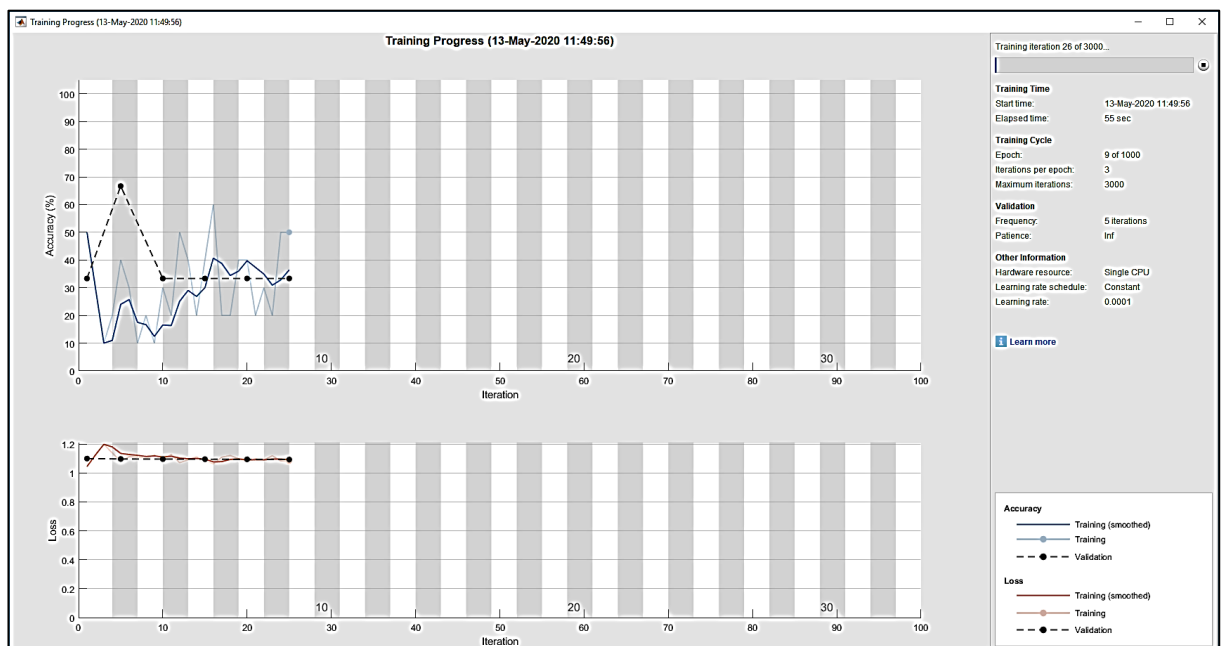
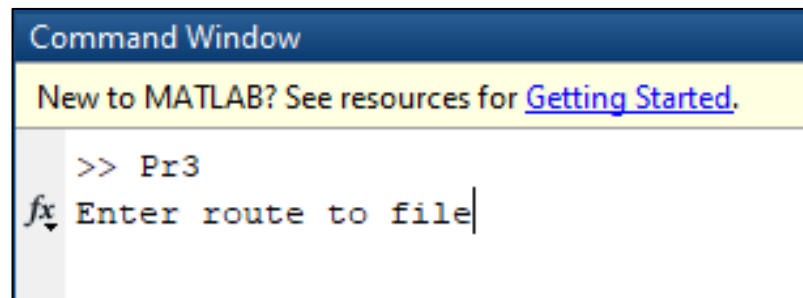


Рис. 3.14. Візуалізація навчання нейронної мережі

Навчена нейронна мережа зберігається у директорії проекту.

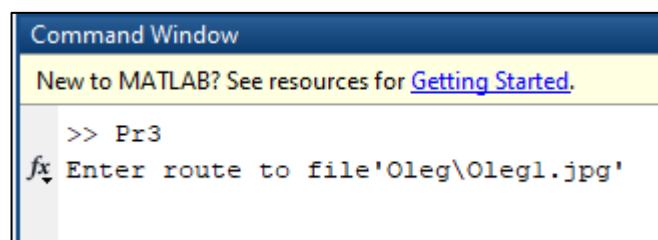
Після натиснення кнопки Recognize в основному вікні програми (рис. 3.1) запускається програмний модуль Pr3 (рис. 3.15), який розпізнає особу користувача за клавіатурним почерком за допомогою нейронної мережі, яка була навчена за рахунок застосування програмного модулю Pr2.



```
Command Window
New to MATLAB? See resources for Getting Started.
>> Pr3
fx Enter route to file|
```

Рис. 3.15. Початкове вікно програмного модулю Pr3

Для запуску процесу розпізнавання оператор системи повинен вказати шлях до файлу, в якому зберігаються параметри клавіатурного почерку. Файл представляє із себе трьох-кольорове зображення матриці параметрів клавіатурного почерку. На рис. 3.16. показано, що оператор системи вказав шлях до файлу Oleg1.jpg.



```
Command Window
New to MATLAB? See resources for Getting Started.
>> Pr3
fx Enter route to file'Oleg\Oleg1.jpg'
```

Рис. 3.16. Введення шляху до файлу зображення

Після визначення шляху оператор системи повинен натиснути клавішу ENTER, що призводить до застосування нейронної мережі для розпізнавання особи користувача. Результат розпізнавання зображення Oleg1.jpg показано на рис. 3.17. Як показано на рис. 3.17 ім'я розпізнаного користувача – Oleg.

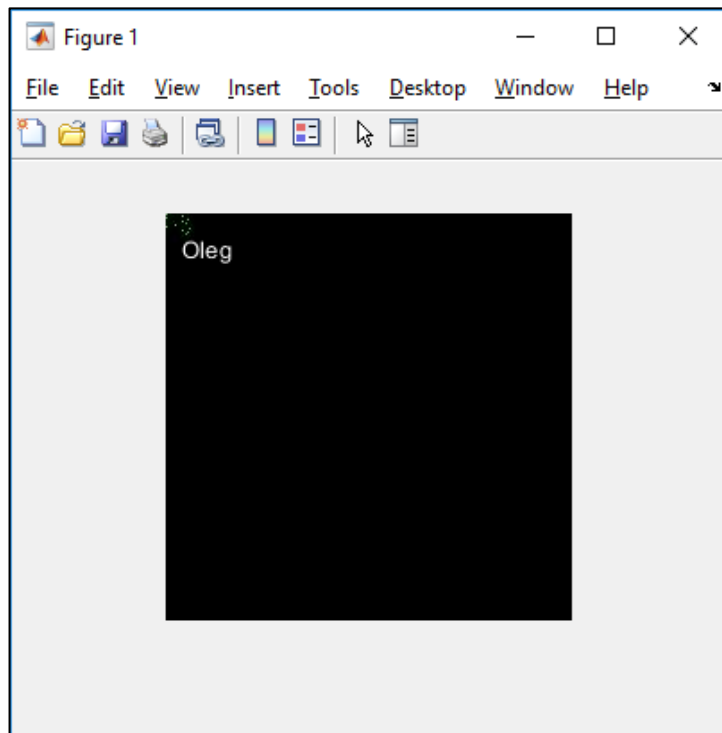


Рис. 3.17. Результат розпізнавання особи користувача

Після закриття вікна з результатом розпізнавання особи користувача програма завершує своє виконання.

3.3. Експериментальні дослідження

Проведено ряд комп'ютерних експериментів спрямованих на визначення ефективності розробленої нейромережевої моделі при використанні для аналізу різних комбінацій параметрів клавіатурного почерку.

Для оцінки ефективності використані параметри точності (*Accuracy*) та втрат (*Loss*), що розраховані за допомогою виразів (2.25, 2.26).

Базуючись на результатах [17-19] прийняті наступні умови застосування SqueezeNet:

- Кількість користувачів, особи яких мають бути розпізнані - 10.
- Текст, який підлягає аналізу, може складатися з малих літер англійського алфавіту і пробілу (всього 27 символів).

- Клавіатурний почерк аналізується при наборі текстів, довжина яких становить 12 символів. Для реєстрації параметрів введення розроблена Windows-орієнтована програма KeyboardProgram, що дозволяє реєструвати символ, відповідний натиснутій клавіш, а також значення часу натискання і відпускання клавіші з помилкою, яка дорівнює тривалості 50 тактів процесора. Попередня обробка зареєстрованих параметрів реалізується за допомогою розробленої програми FormImagePy.

- Обсяг тренувальної вибірки склав 1000 прикладів, по 100 прикладів для кожного користувача. Обсяг валідаційної вибірки склав 50 прикладів, по 5 прикладів для кожного користувача.

Зазначені умови застосування SqueezeNet визначені з позицій оціночного характеру експериментальних досліджень, спрощення формування навчальної вибірки і можливості коректного подання вхідного поля нейронної мережі.

В результаті першої серії експериментів окремо для кожного з параметрів клавіатурного почерку (ТУК, ТМК, ДТУК, ДТМК, ВТТ) побудовані залежності значень показників ефективності від кількості епох навчання. Так, на рис. 3.18 і рис. 3.19 показані графіки цих залежностей на навчальних і валідаційних даних для ТУК і ТМК.

З метою підвищення наочності отриманих результатів для тренувальної вибірки розраховувалися згладжені значення показника точності (Accuracy_smoothed) і показника втрат (Loss_smoothed), графіки яких також показані на рис. 3.18 і рис. 3.19.

Відповідно до рекомендацій [17-19], процес навчання тривав 1000 епох. Однак в експериментах визначено, що показники ефективності стабілізуються після 400-500 епох навчання, що продемонстровано на графіках рис. 3.18 - 3.19. Тому для зменшення часу досліджень частина експериментів обмежилася 500 епохами.

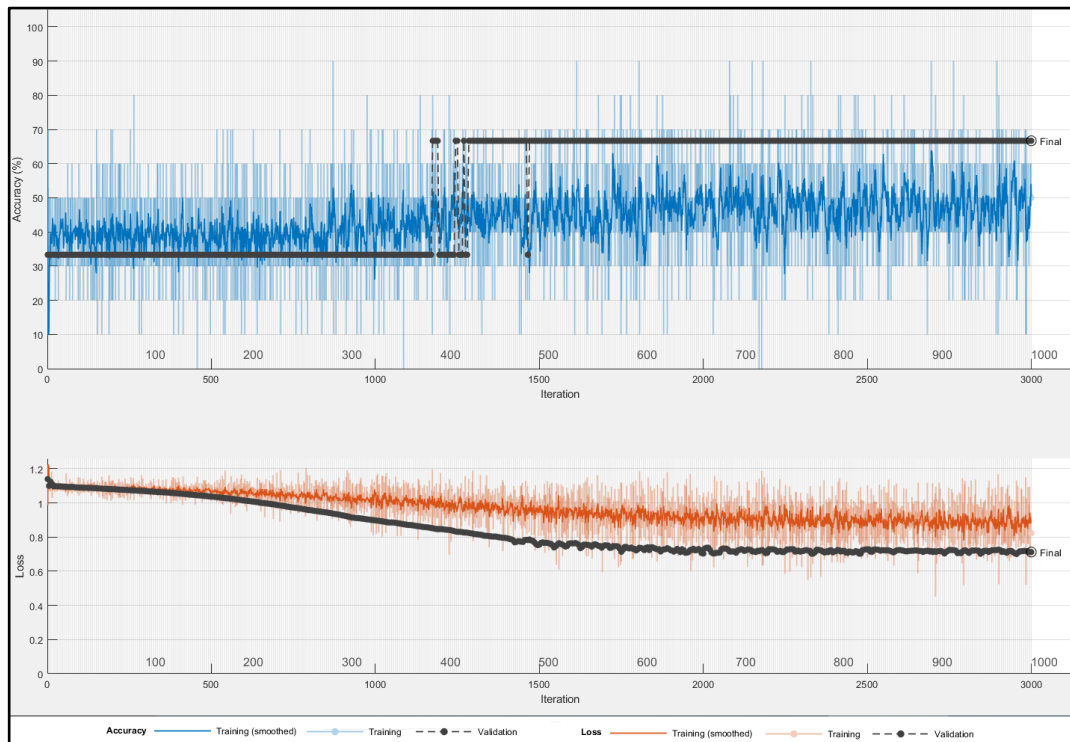


Рис. 3.18. Графік залежності точності розпізнавання і втрат від тривалості навчання при використанні ТУК

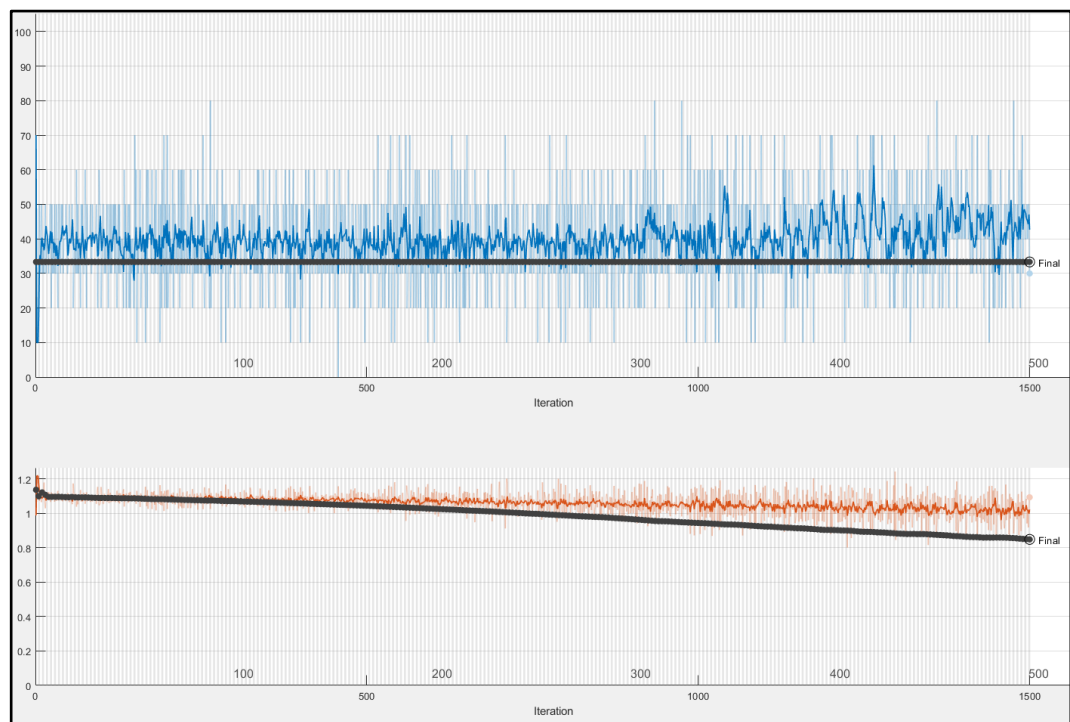


Рис. 3.19. Графік залежності точності розпізнавання і втрат від тривалості навчання при використанні ТМК

Зм.	Арк.	№ докум.	Підпис	Дата

Оскільки вхідне поле SqueezeNet може бути представлено у вигляді рисунка з багатоканальним растром, то в другій серії експериментів досліджувалися різні варіанти одночасного використання комбінацій декількох параметрів клавіатурного почерку. Так, на рис. 3.20 та рис. 3.21 показані графіки показників ефективності для комбінації ТУК і ДТУК та для комбінації ТМК і ДТУК при 500 епохах навчання. Основні результати проведених експериментів зведені в табл. 3.1, де представлені показники ефективності SqueezeNet моделі при використанні для аналізу клавіатурного почерку різних параметрів.

Як показує аналіз даних табл. 3.1, при формуванні вхідного поля SqueezeNet на базі одного параметра клавіатурного почерку досягти найбільш високої точності розпізнавання дозволяє застосування параметру ТУК. При цьому комбінація даного параметра з іншими не призводить до яких-небудь істотних позитивних змін.

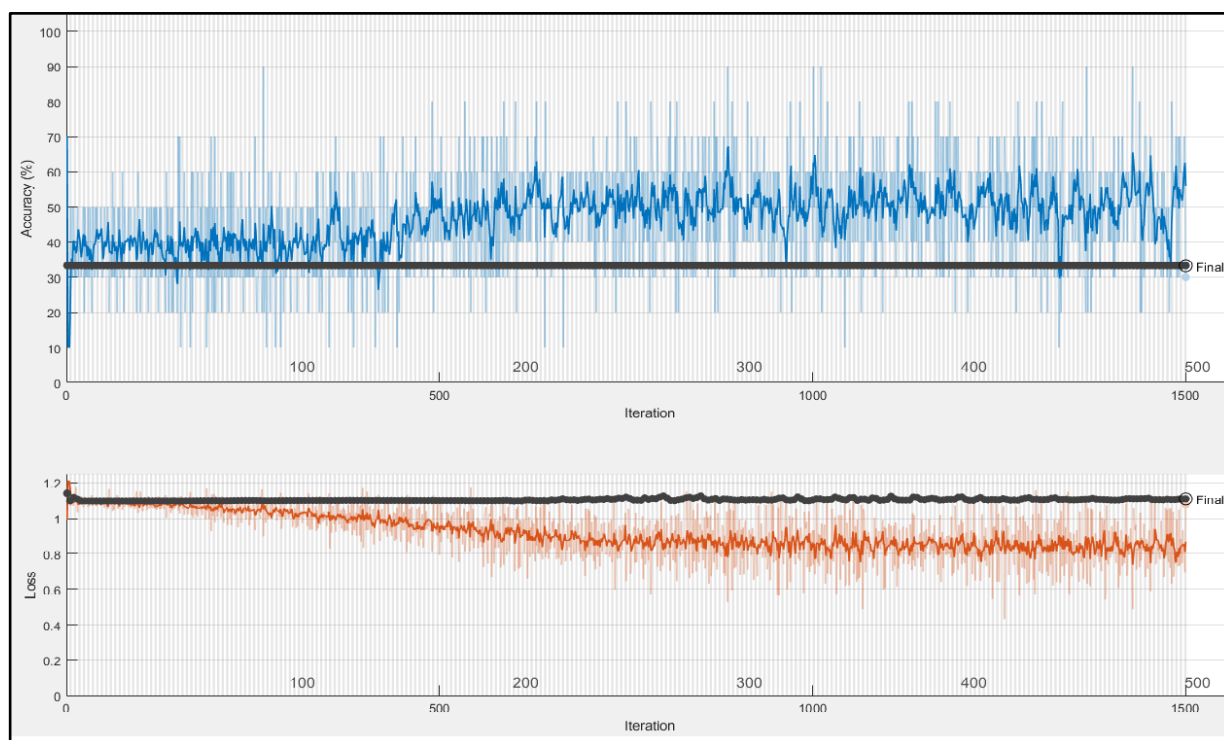


Рис. 3.20. Графік залежності точності розпізнавання і втрат від тривалості навчання при використанні ТУК і ДТУК

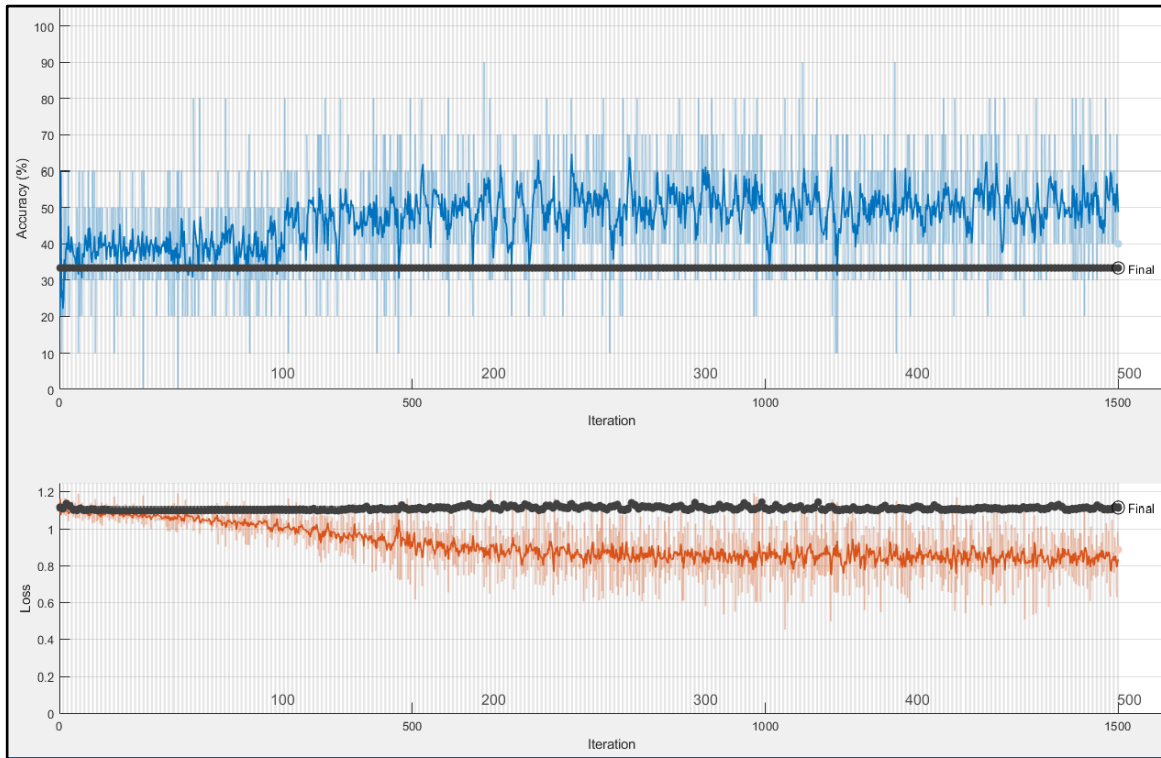


Рис. 3.21. Графік залежності точності розпізнавання і втрат від тривалості навчання при використанні ТМК і ДТУК

Таблиця 3.1

Значення показників ефективності

Використані параметри	Accuracy на тренувальній вибірці (%)	Accuracy на валідаційній вибірці (%)	Loss на тренувальній вибірці	Loss на валідаційній вибірці
ТУК	71	67	0,76	0,76
ТМК	70	33	1,01	1,17
ТМК, ДТУК	40	33	0,78	1,18
ДТМК	50	32	0,83	0,84
ВТТ	67	33	0,95	4,5
ТУК, ДТУК	68	53	0,67	1,16
ТУК, ДТМК	71	66	0,8	0,8
ТМК, ДТУК	70	34	1	1,17
ТУК, ТМК, ДТУК	70	35	0,68	1,17

Слід зазначити, що отриманий результат відповідає висновкам роботи [20], в якій клавіатурний почерк аналізувався за допомогою досить простої

ймовірнісної нейронної мережі і дещо суперечить висновкам [13], де клавiатурний почерк після досить складної попередньої обробки аналізувався за допомогою бімодальною статистичної моделі. Це вказує на необхідність проведення подальших досліджень в напрямку удосконалення процедур попередньої обробки параметрів клавiатурного почерку з метою збільшення їх інформативності. Крім того, представляє інтерес розробка методу визначення архітектурних параметрів нейромережевої моделі, призначеної для аналізу клавiатурного почерку.

Висновки до розділу 3

Третій розділ присвячено розробці програмного забезпечення системи розпізнавання особи користувача за клавiатурним почерком та експериментальним дослідженням спрямованим на верифікацію цієї системи.

В результаті проведених досліджень:

- Розроблено програмне забезпечення, що реалізує запропонований спосіб розробки системи розпізнавання особи користувача за клавiатурним почерком. Програмне забезпечення дозволяє реєструвати параметри динаміки клавiатурного почерку, оброблювати зареєстровані параметри для забезпечення можливості їх нейромережевого аналізу та реалізувати розпізнавання за допомогою згорткової нейронної мережі SqueezeNet. Наведено детальний опис функціональних можливостей та інтерфейсу програмного забезпечення.

- Проведені експериментальні дослідження, результати яких вказують не те, що запропонована система дозволяє досягти точності розпізнавання особи користувача на рівні 67%, що відповідає кращим рішенням в даній області.

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		56

ВИСНОВКИ

Роботу присвячено вирішенню задачі розробки системи розпізнавання особи користувача за клавіатурним почерком. Проведені дослідження дозволяють зробити наступні висновки:

- Визначено, що важливим напрямком підвищення захисту інформації від витоків є впровадження засобів розпізнавання особи користувача за клавіатурним почерком. Перспективність вказаних засобів пояснюється можливістю реалізації прихованого моніторингу справжності особи користувача та можливістю підвищити ергономічність системи аутентифікації за рахунок використання в якості паролю довільних текстів.

- Показана доцільність вдосконалення засобів розпізнавання особи за клавіатурним почерком шляхом впровадження в них модулю аналізу параметрів динаміки клавіатурного почерку на основі згорткової нейронної мережі типу SqueezeNet.

- Розроблено спосіб проектування системи розпізнавання особи користувача за клавіатурним почерком, що базується на процедурі представлення параметрів клавіатурного почерку у вигляді придатному для аналізу згортковою нейронною мережею та нейромережевій моделі типу SqueezeNet пристосованій до аналізу параметрів клавіатурного почерку.

- Розроблено програмне забезпечення, що реалізує запропонований спосіб розробки системи розпізнавання особи користувача за клавіатурним почерком. Програмне забезпечення дозволяє реєструвати параметри динаміки клавіатурного почерку, оброблювати зареєстровані параметри для забезпечення можливості їх нейромережевого аналізу та реалізувати розпізнавання за допомогою згорткової нейронної мережі SqueezeNet.

- Проведені експериментальні дослідження результати яких вказують не те, що запропонована система дозволяє досягти точності розпізнавання особи користувача за клавіатурним почерком на рівні 67%, що відповідає кращим рішенням в даній області.

					<i>ІАЛЦ.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		57

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Alghamdi S., Elrefaei L. Dynamic user verification using touch keystroke based on medians vector proximity. In Computational Intelligence, Communication Systems and Networks (CICSyN), 2015 7th International Conference on, pp. 121–126. IEEE, 2015.
2. Ali, M. L., Thakur, K., & Tappert, C. (2015). User authentication and identification using neural network. I-manager's Journal on Pattern Recognition, 2(2), pp. 28-39.
3. Abisado M., Gerardo B., Fajardo A. Towards keystroke analysis using neural network for multi - factor authentication of learner recognition in on - line examination. In Manila International Conference on Trends in Engineering and Technology (2017), pp. 71–74.
4. Bo C., Zhang L., Jung T., Han J., Li X.-Y., Wang Y.. Continuous user identification via touch and movement behavioral biometrics. In 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC), pp. 1–8. IEEE, 2014
5. Hayreddin Ç., Shambhu U. Sensitivity analysis in keystroke dynamics using convolutional neural networks 2017 IEEE Workshop on Information Forensics and Security (WIFS) 4-7 Dec. 2017, pp. 1-6.
6. Hu, Z., Tereykovskiy, I., Zorin, Y., Tereykovska, L., Zhibek, A. Optimization of convolutional neural network structure for biometric authentication by face geometry // Advances in Intelligent Systems and Computing. 2018. Vol. 754, pp 567-577.
7. Iandola F.N., Han S., W. Moskewicz M.W. SqueezeNet: AlexNetlevel accuracy with 50x fewer parameters and <0.5MB model size. arXiv:1602.07360v4 [cs.CV], 2016. 13 p. Available at: <https://arxiv.org/pdf/1602.07360.pdf>.
8. Kanimozhi, M., Puvirajasingam, K., & Avitha, M. S. (2014). Survey on keystroke dynamics for a better biometric authentication system. International

Journal of Emerging Technologies and Engineering (IJETE), 1(9), pp. 116-139.

9. Kevin C. Using keystroke analytics to improve pass–fail classifiers, Journal of Learning Analytics, Vol. 4, No. 2, pp. 189–211, (2017).

10. Kun L., Jiafen L., Ping T., Qing L. Keystroke Biometrics for Freely Typed Text Based on CNN model. Twenty-Second Pacific Asia Conference on Information Systems, Japan 2018, pp. 2278-2324.

11. Liu, M., Guan, J. User keystroke authentication based on convolutional neural network, Communications in Computer and Information Science 2019, 971, pp. 157-168.

12. Lin, C.-H., Liu, J.-C., Lee, K.-Y. On neural networks for biometric authentication based on keystroke dynamics. Sensors and Materials 2018 30(3), pp. 385-396.

13. Mohammadreza H., Mehrbaksh N., Othman I., Ali Z., Sarminah S. Authentication systems: A Literature Review and Classification Telematics and Informatics, Vol. 35, Issue 5, pp. 1491–1511, (2018).

14. Saket M., Soumyajit G., Vikram P. Deep Secure: A Fast and Simple Neural Network based approach for User Authentication and Identification via Keystroke Dynamics Conference: IWAISE, International Joint Conference on Artificial Intelligence (IJCAI) - 2017At: Melbourne, Australia pp. 34-40.

15. Shankar S., Robertson D., Ioannou Y., Criminisi A., Cipolla R. Refining Architectures of Deep Convolutional Neural Networks. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, 2016, pp. 2212-2220.

16. Tereykovska L., Tereykovskiy I., Ayt Khozhaeva E., Tynymbayev S., Imanbayev A. Encoding of neural network model exit signal, that is devoted for distinction of graphical images in biometric authenticate systems // News of the national academy of sciences of the republic of kazakhstan series of geology and technical sciences. Volume 6, Number 426 (2017), pp. 217 – 224.

17. Tereikovskiy I., Tereikovska L., Korystin O., Mussiraliyeva S.,

					<i>IAJLIT.045420.003 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		59

Sambetbayeva A. User Keystroke Authentication and Recognition of Emotions Based on Convolutional Neural Network. In: Hu Z., Petoukhov S., He M. Advances in Artificial Systems for Medicine and Education III. AIMEE 2019. Advances in Intelligent Systems and Computing, vol 1126, pp 283-292.

18. Toliupa S., Kulakov Y., Tereikovskiy I., Tereikovskiy O., Nakonechnyi V., Keyboard Dynamic Analysis by Alexnet Type Neural Network, 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2020 – Proceedings. pp. 416-420.

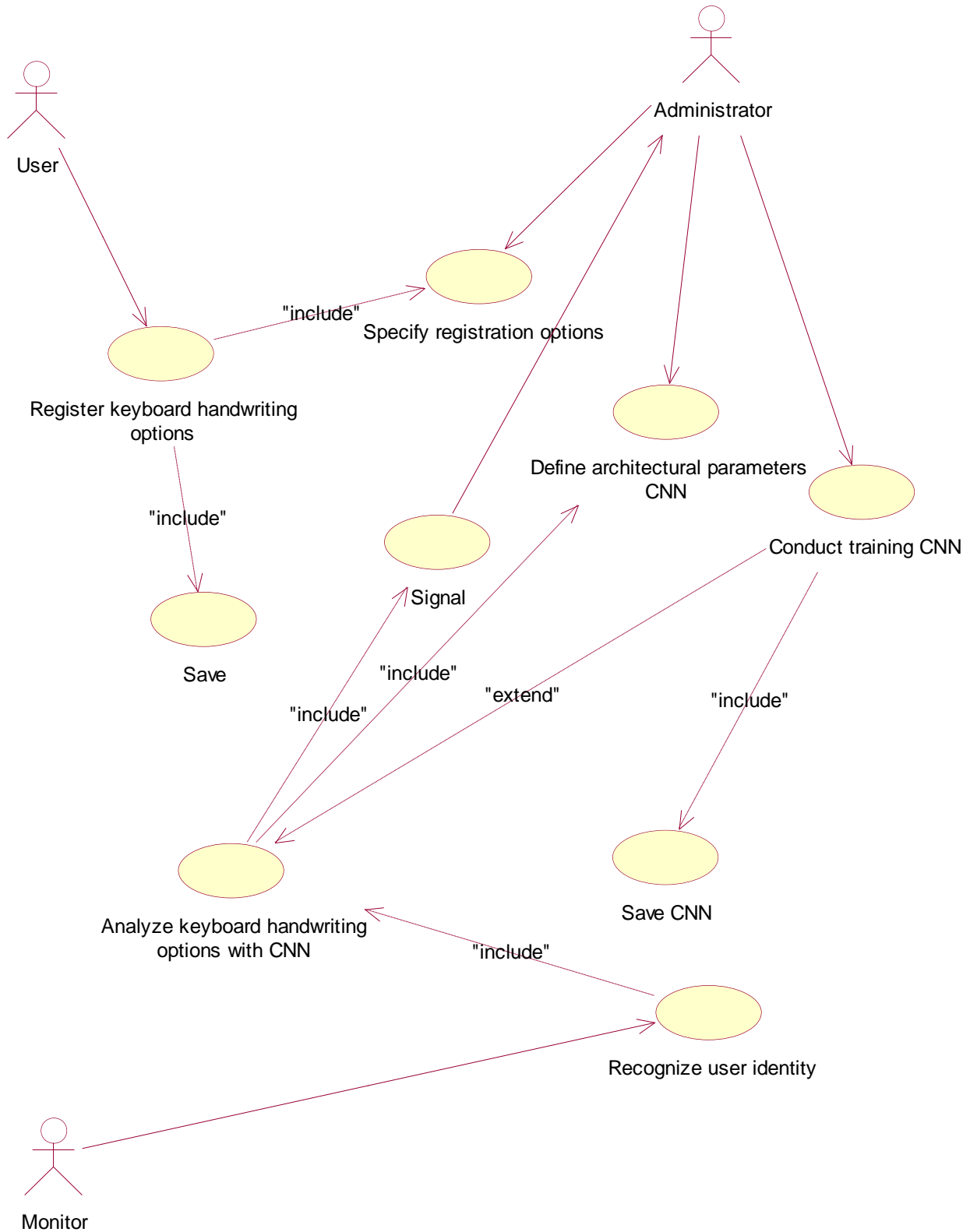
19. Toliupa S., Tereikovska L., Korystin O., Chernyshev D., Tereikovskiy I. Low-Resource Convolution Neural Network for Keyboard Recognition of the User. 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine, 2019, pp. 222-226.

20. Vysotska O., Davydenko A. (2020) Keystroke Pattern Authentication of Computer Systems Users as One of the Steps of Multifactor Authentication. In: Hu Z., Petoukhov S., Dychka I., He M. Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing, vol 938. pp. 784-796.

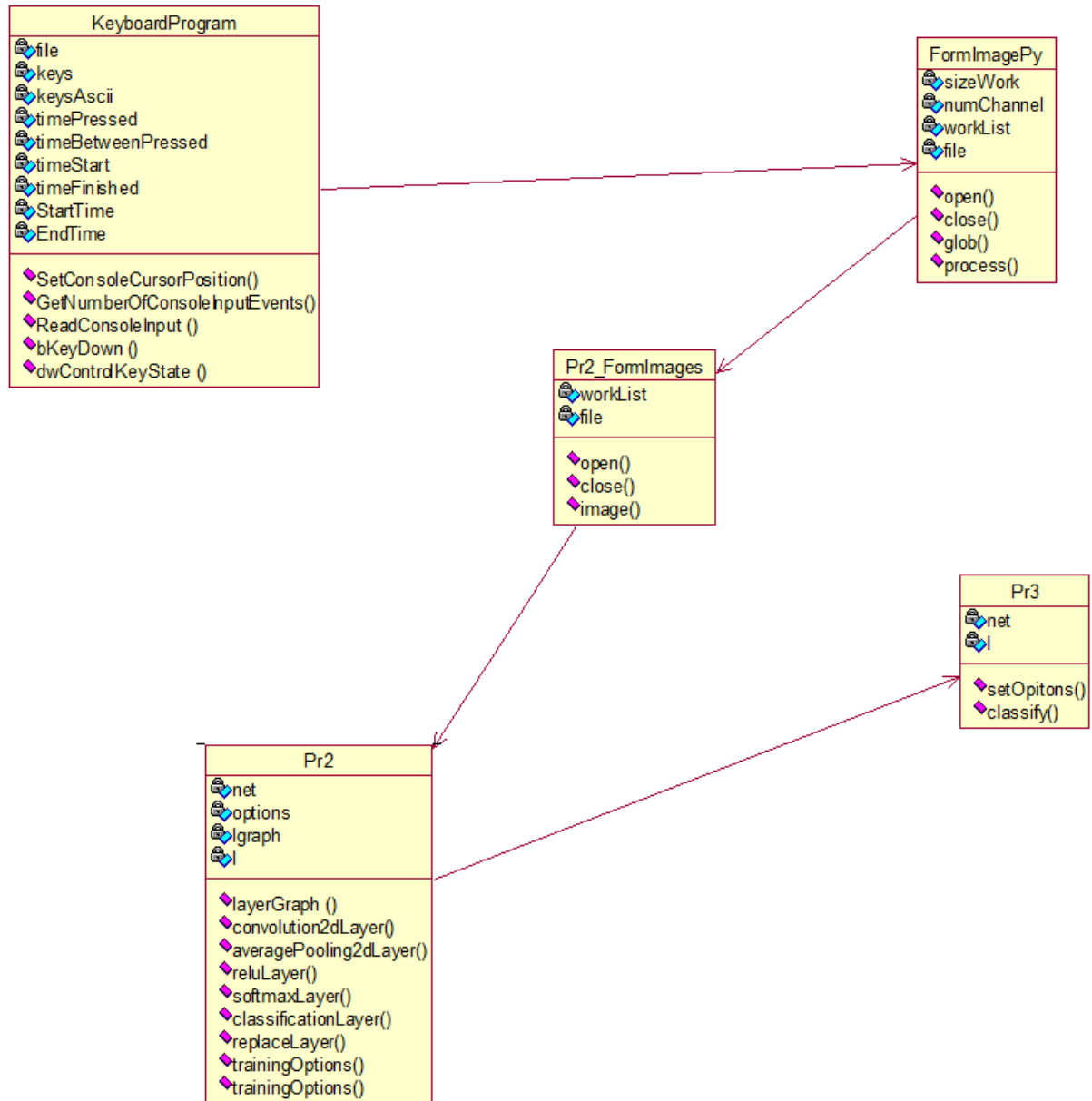
21. Xiaofeng, L., Shengfei, Z., Shengwei, Y. Continuous authentication by free-text keystroke based on CNN plus RNN Procedia Computer Science 147, pp. 314-318. 2019.

22. Yunbin D., Yu Z. Keystroke Dynamics Advances for Mobile Devices Using Deep Neural Network GCSR Vol. 2, pp. 59-70, 2015.

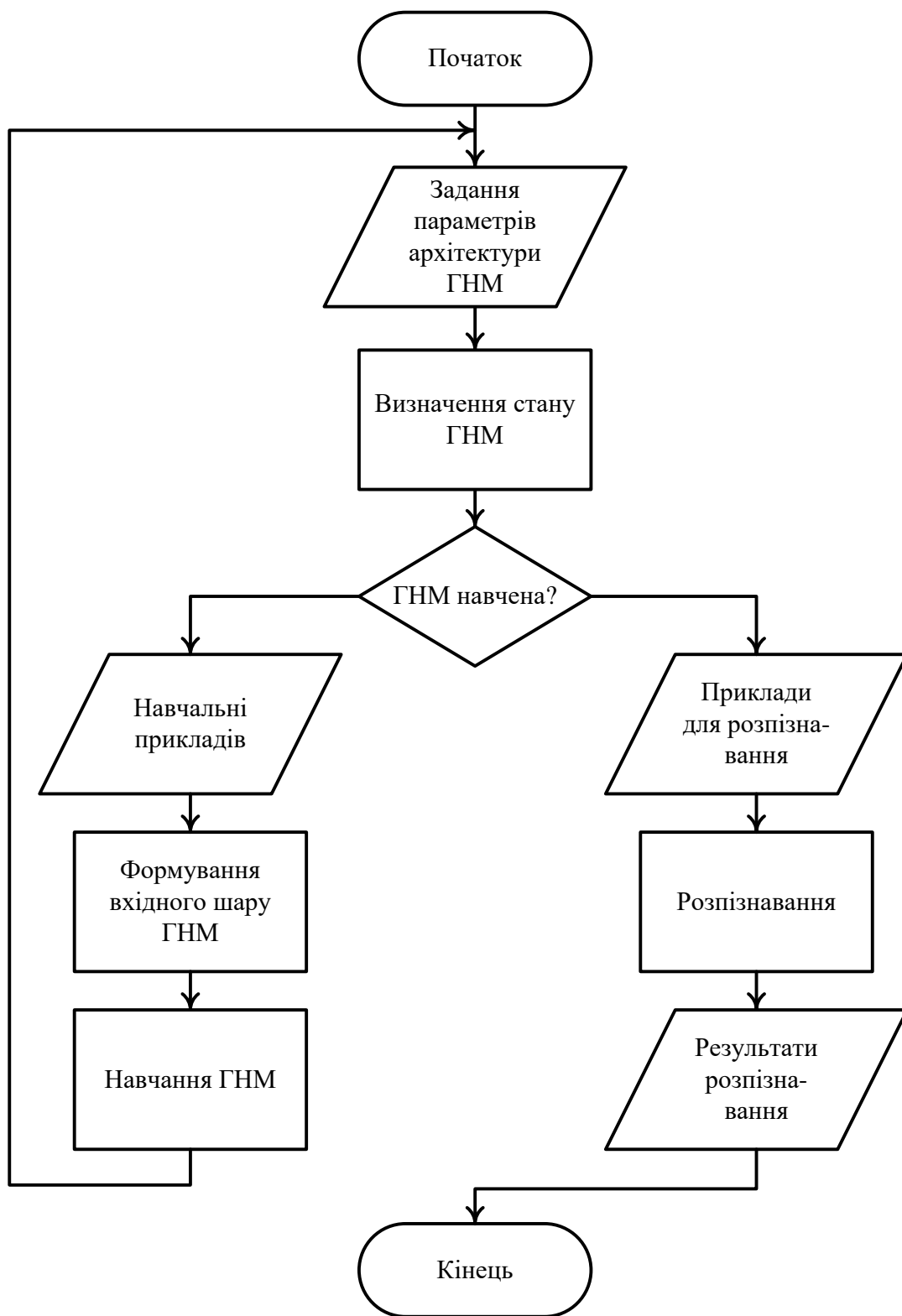
Додаток 1. Копії графічних матеріалів



					ІАЛЦ.045420.004 Д1		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Терейковський О.І.			Літ.	Аркуш	Аркушів
Перевірив		Кулаков Ю. О.				1	1
Реценз.					НТУУ «КПІ ім.Ігоря Сікорського» ФІОТ, ІО-61		
Н. Контр.		Сімоненко В. П.					
Затвердив		Стіренко С.Г.					
					Системи розпізнавання особи користувача за клавіатурним почерком Діаграма прецедентів		



					ІАЛЦ.045420.005 Д2			
Зм.	Арк.	№ докум.	Підпис	Дата	Системи розпізнавання особи користувача за клавіатурним почерком Діаграма класів			
Розробив		Терейковський О.І.						
Перевірив		Кулаков Ю. О.						
Реценз.								
Н. Контр.		Сімоненко В. П.						
Затвердив		Стіренко С.Г.			Лім.	Аркуш	Аркушів	
							1	1
						НТУУ «КПІ ім.Ігоря Сікорського» ФІОТ. ІО-61		

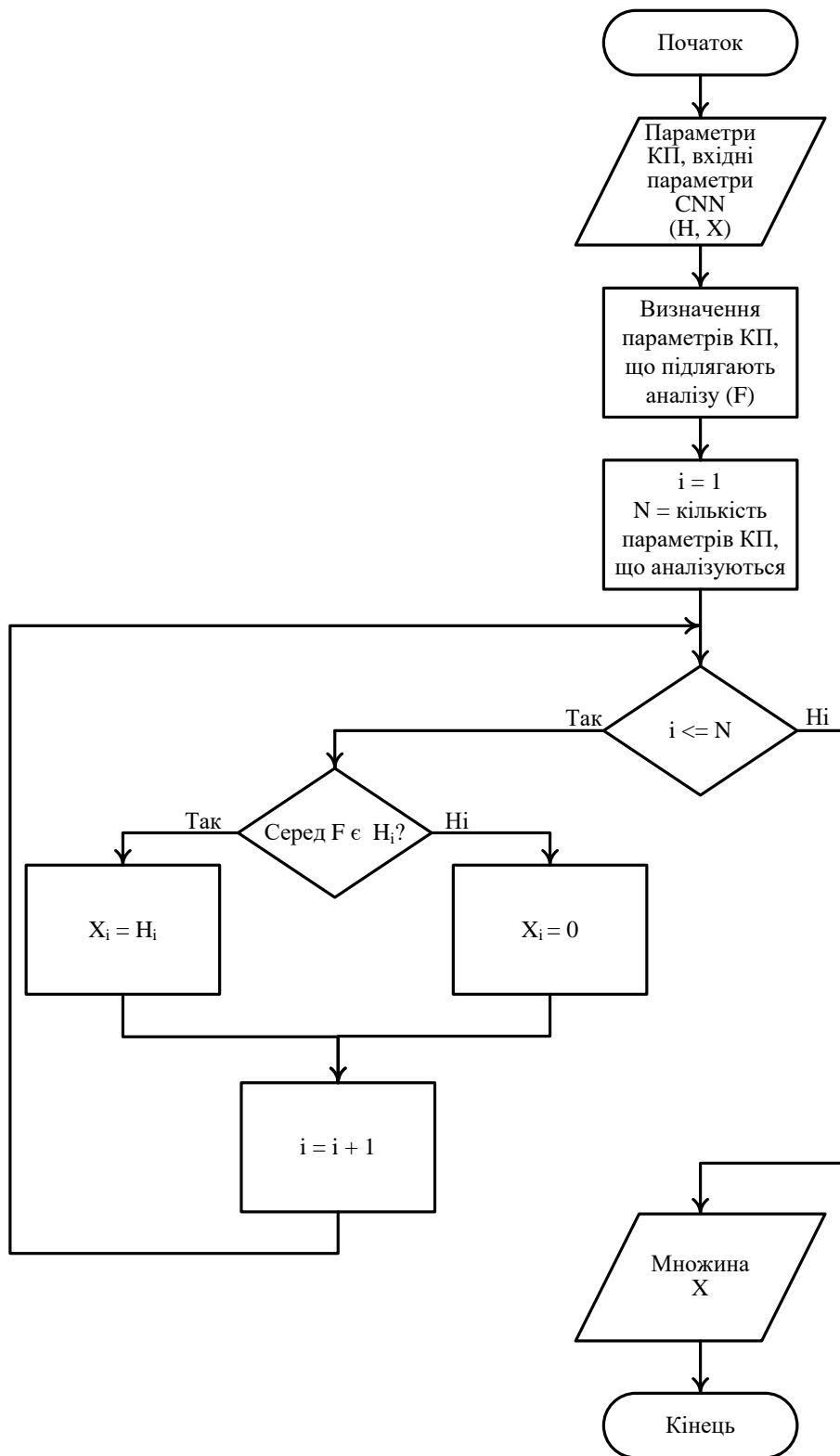


ІАЛЦ.045420.006 ДЗ

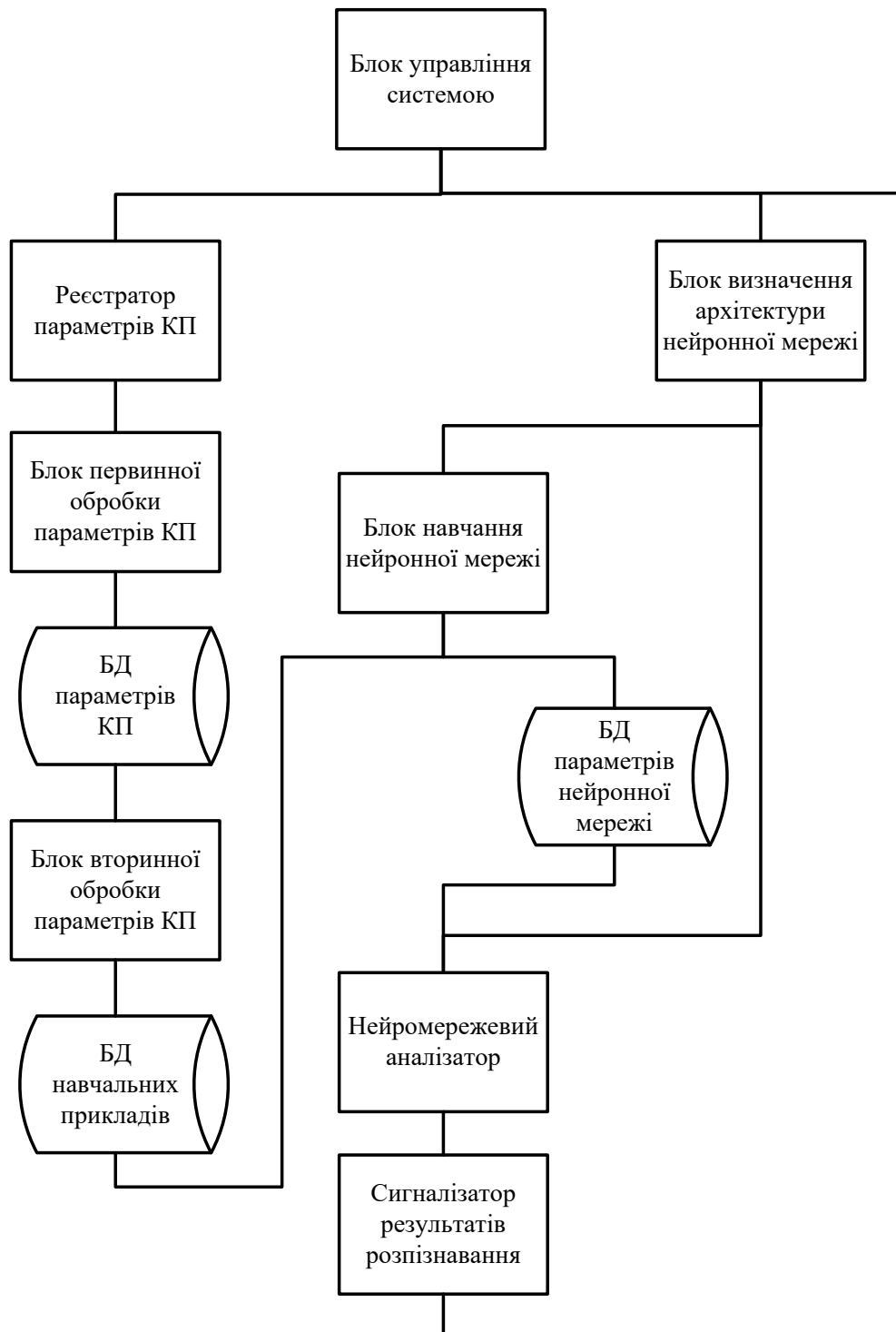
Зм.	Арк.	№ докум.	Підпис	Дата
Розробив		Терейковський О.І.		
Перевірив		Кулаков Ю. О.		
Реценз.				
Н. Контр.		Сімоненко В. П.		
Затвердив		Стіренко С.Г.		

Формування множини вхідних параметрів мережі SqueezeNet
Схема алгоритму

Літ.	Аркуш	Аркушів
	1	1
НТУУ «КПІ ім.Ігоря Сікорського» ФІОТ, ІО-61		



					<i>ІАЛЦ.045420.007 Д4</i>		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Терейковський О.І.			Літ.	Аркуш	Аркушів
Перевірив		Кулаков Ю. О.				1	1
Реценз.					НТУУ «КПІ ім.Ігоря Сікорського» ФІОТ, ІО-61		
Н. Контр.		Сімоненко В. П.					
Затвердив		Стіренко С.Г.					
Навчання нейронної мережі Схема алгоритму							



ІАЛЦ.045420.008 Д5

Зм.	Арк.	№ докум.	Підпис	Дата
Розробив		Терейковський О.І.		
Перевірив		Кулаков Ю. О.		
Реценз.				
Н. Контр.		Сімоненко В. П.		
Затвердив		Стіренко С.Г.		

Системи розпізнавання особи користувача за клавіатурним почерком
Схема структурна

Літ.	Аркуш	Аркушів
	1	1
НТУУ «КПІ ім.Ігоря Сікорського» ФІОТ, ІО-61		

KeyboardProgram

```
#include <windows.h>
#include <iostream>
#include <fstream>
#include <sstream>
#include <stdio.h>
#include <string>
#include <vector>
using namespace std;
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    HANDLE hIn;
    HANDLE hOut;
    COORD KeyWhere;
    COORD LoopWhere;
    COORD EndWhere;
    bool Continue = TRUE;
    DWORD EventCount = 0;
    int LoopCount = 0;
    int keyLoopCount = 0;
    int KeyEvents = 0;
    INPUT_RECORD InRec;
    DWORD NumRead;
    unsigned char HoldKey;
    LARGE_INTEGER ClockFrequency;
```

```
QueryPerformanceFrequency(&ClockFrequency);
LARGE_INTEGER NoActionStart;
LARGE_INTEGER NoActionFinish;
LARGE_INTEGER NoActionDelta;
LARGE_INTEGER NoActionDeltaWhole;
vector<int>keys;
vector<int>keysAscii;
vector<double> timePressed;
vector<double> timeBetweenPressed;
vector<double> timeStart;
vector <double> timeFinished;
int PressingStart = 0;
int PressingFinish = 0;
LARGE_INTEGER StartTime;
LARGE_INTEGER EndTime;
LARGE_INTEGER delta;
LARGE_INTEGER deltaWhole;
deltaWhole.QuadPart = 0;
NoActionDeltaWhole.QuadPart = 0;
ifstream counterOfFile("Counter.txt");
string numberCount = "";
counterOfFile >> numberCount;
if (numberCount == "")
{
    numberCount = "0";
}
counterOfFile.close();
string nameOfFile = "Result" + numberCount + ".txt";
ofstream file(nameOfFile);
```

```

hIn = GetStdHandle(STD_INPUT_HANDLE);
hOut = GetStdHandle(STD_OUTPUT_HANDLE);
cout << "Key Events  : " << flush; // SHOW no. of ENTERY " Key Events
"

KeyWhere.X = 15;
KeyWhere.Y = 0;
LoopWhere.X = 0;
LoopWhere.Y = 1;
EndWhere.X = 0;
EndWhere.Y = 2;

////////////////////////////////// FIRST LOOP ////////////////////////////////////////////
while (Continue)
{
    SetConsoleCursorPosition(hOut,
        LoopWhere);
    //LoopCount++;
    cout << LoopCount++ << "      " << flush;
    //Sleep(100); // To slow it down!!
    cout << EventCount << endl;
    GetNumberOfConsoleInputEvents(hIn, &EventCount);
    //EventCount "DWORD" always start with "SAME NUMBER"
////////////////////////////////// SECOND LOOP ////////////////////////////////////////////
    while (EventCount > 0)
    {
        QueryPerformanceCounter(&NoActionStart);
        ReadConsoleInput(hIn, &InRec, 1, &NumRead); //why?
        if (InRec.EventType == KEY_EVENT) // if event BY key
        {
            QueryPerformanceCounter(&NoActionFinish);

```

```

        NoActionDelta.QuadPart = NoActionFinish.QuadPart -
NoActionStart.QuadPart;
        NoActionDeltaWhole.QuadPart +=
NoActionDelta.QuadPart;
        double NoActiondeltaSeconds =
((double)(NoActionDeltaWhole.QuadPart)) / 10;
        if (InRec.Event.KeyEvent.bKeyDown)//
        {
            QueryPerformanceCounter(&StartTime);
            PressingStart = LoopCount;
            cout << "I am here" << endl;
            HoldKey =
InRec.Event.KeyEvent.uChar.AsciiChar;
            keys.push_back(HoldKey);
            cout << char(HoldKey) << endl;
            //cout << HoldKey << endl;
            QueryPerformanceCounter(&EndTime);
            delta.QuadPart = EndTime.QuadPart -
StartTime.QuadPart;
            deltaWhole.QuadPart += delta.QuadPart;
        }
        else
        {
            cout << "Don't Play with me " << flush << " here i
am" << endl;
            ++KeyEvents;
            //QueryPerformanceCounter(&PressingFinish);
            PressingFinish = LoopCount;
            SetConsoleCursorPosition(hOut,

```



```

        KeyWhere);
        cout << KeyEvents << flush;// SHOW no. of
ENTERY "keyEvents"
        cout << "key loop counter: " <<
deltaWhole.QuadPart << endl;
        double deltaSeconds =
((double)(deltaWhole.QuadPart)) / 10;//(float)(ClockFrequency.QuadPart);
        cout <<"key loop counter in miliseconds: "<<
deltaSeconds << endl;
        file << HoldKey << " ,ascii value: " <<
int(HoldKey) << " , time: " << deltaSeconds << " , time between pressing: " <<
NoAactiondeltaSeconds << " , key down: "<< PressingStart << " , Key up: "<<
PressingFinish << endl ;
        keysAscii.push_back(int(HoldKey));
        timePressed.push_back(deltaSeconds);
        timeBetweenPressed.push_back(NoAactiondeltaSeconds);
        timeStart.push_back(PressingStart);
        timeFinished.push_back(PressingFinish);
        cout <<"Here is lloop couter:: " << LoopCount <<
endl;
        deltaWhole.QuadPart = 0;
        NoActionDeltaWhole.QuadPart = 0;
        if (InRec.Event.KeyEvent.dwControlKeyState &
LEFT_ALT_PRESSED || (int)HoldKey == 13) //why?
        {
                cout << "I am in the hill" << endl;
                cout << "NO ONO " << endl;
                SetConsoleCursorPosition(hOut,
EndWhere);

```

```

        cout << "Exiting..." << endl;
        Continue = FALSE;
    }
} //END OF THE "else"
} //END OF THE FIRST "if-condition"
GetNumberOfConsoleInputEvents(hIn, &EventCount);
} // END OF THE SECOND LOOP
} //END OF THE FIRST LOOP
ofstream currFile(nameOfFile, ios::app);
currFile << "All ascii code of pushed buttons: " << endl;
    for (size_t i = 0; i < keys.size(); i++)
    {
        currFile << char(keys[i]) << " : " << keys[i] << endl;
    }
currFile.close();
ifstream counterOfFileNames("Counter.txt");
string numberCounter = "";
counterOfFileNames >> numberCounter;
if (numberCounter == "")
{
    numberCounter = "0";
}
int numIntCounter = stoi(numberCounter);
counterOfFileNames.close();
ofstream counterOfFileNamesOut("Counter.txt");
++numIntCounter;
stringstream ss;
ss << numIntCounter;
string str = ss.str();

```

```
counterOfFileNamesOut << str;
counterOfFileNamesOut.close();
timeBetweenPressed[0] = 0;
cout << "There r all keys: " << endl;
for (size_t i = 0; i < keys.size(); i++)
{
    cout << char(keys[i]) << " ,ascii value: " << keysAscii[i] << " , time: "
<< timePressed[i] << " , time between pressing: " << timeBetweenPressed[i] << " ,
key down: " << timeStart[i] << " , Key up: " << timeFinished[i] << endl;
}
file.close();
return 0;
}
```