**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА** **КОМП’ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ**

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В. Казмірчук

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ р.

На правах рукопису

УДК 004.056.5:510.22(043.3)

**МАГІСТЕРСЬКА АТЕСТАЦІЙНА РОБОТА**

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ**

**«МАГІСТР»**

**Тема**:Нейромережева модель розпізнавання підписів користувачів інформаційних систем

|  |  |
| --- | --- |
| **Автор:** | В.Я. Войтович |
| **Науковий керівник:** к.т.н., доц. | Л.О. Терейковська |
| **Нормоконтролер:** асист. |  |

**Київ 2020**

**Факультет:** Кібербезпеки, комп’ютерної та програмної інженерії

**Кафедра:** Компютеризованих систем захисту інформації

**Освітній ступінь:** Магістр

**Спеціальність:** 125 «Кібербезпека»

**Освітньо-професійна програма**: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В. Казмірчук

«\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на виконання магістерської атестаційної роботи**

**магістранта Войтовича Владислава Ярославовича**

1. Тема: *Нейромережева модель розпізнавання підписів користувачів інформаційних систем*

затверджена наказом ректора від «\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_ № \_\_\_\_\_/ст*.*

1. Термін виконання з \_\_.\_\_.20\_\_р. по \_\_.\_\_.20\_\_р*.*
2. Вихідні дані: проаналізувати існуючі моделі та методи аналізу і розпізнавання підписів користувачів інформаційних систем; на основі аналізу виділити вхідні і вихідні параметри, завдяки яким можливо провести порівняння існуючих моделей, виявлення їх переваг і недоліків; розробити нейромережеву модель, алгоритмічне та програмне забезпечення нейромережевої системи розпізнавання підписів.

4. Зміст пояснювальної записки: аналіз існуючих систем та методик розпізнавання підписів користувачів інформаційних систем; розробка нейромережевої моделі розпізнавання підписів користувачів; розробка програмного забезпечення запропонованої моделі, верифікація отриманих результатів.

**КАЛЕНДАРНИЙ ПЛАН**

**виконання магістерської роботи**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Етапи виконання магістерської роботи** | **Термін виконання етапів** | **Примітка** |
|  | Уточнення постановки задачі |  | *Виконано* |
|  | Аналіз літературних джерел |  | *Виконано* |
|  | Обґрунтування вибору рішення |  | *Виконано* |
|  | Збір інформації |  | *Виконано* |
|  | Дослідження сучасних систем і методик розпізнавання підписів користувачів інформаційних систем |  | *Виконано* |
|  | Розробка нейромережевої моделі та структури системи розпізнавання підписів користувачів інформаційних систем |  | *Виконано* |
|  | Розробка алгоритму та програмного забезпечення системи розпізнавання підписів користувачів |  | *Виконано* |
|  | Апробація роботи на науково-технічній конференції КМІТ-2019 |  | *Виконано* |
|  | Перевірка на антиплагіат |  | *Виконано* |
|  | Оформлення і друк пояснювальної записки |  | *Виконано* |
|  | Оформлення презентації |  | *Виконано* |
|  | Отримання рецензій від рецензента |  | *Виконано* |
|  | Захист в ЕК |  | *Виконано* |

**Р****ЕФЕРАТ**

Магістерська атестаційна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, додатків і має 86 сторінки основного тексту, 33 рисунка, 7 таблиць, 13 сторінок додатків. Список використаних джерел містить 32 найменування і займає 4 сторінки. Загальний обсяг роботи 135 сторінок.

Метою роботи є підвищення рівня захищеності інформаційних систем за рахунок застосування нейромережевої моделі розпізнавання аналізу підписів користувачів вказаних систем.

В роботі вирішено задачу побудови нейромережевої моделі розпізнавання підписів користувачів інформаційних систем.

На основі побудованої моделі розроблено структуру системи розпізнавання, а також відповідне алгоритмічне та програмне забезпечення.

Розроблена модель, система та програмне забезпечення відносяться до галузі інформаційної безпеки і можуть бути використані для підвищення рівня захищеності.

Можливі напрямки розвитку цієї роботи пов’язані із розширенням моделі і алгоритму програмного забезпечення відповідно до модернізації комп’ютерних технологій реєстрації параметрів рукописних підписів, наприклад до використання графічних планшетів, що дозволить підвищити точність розпізнавання користувачів.

Ключові слова: нейромережева модель, система захисту, інформаційна безпека, інформаційна система, нейронні мережі, процедура згортки, розпізнавання підписів, користувач.

ЗМІСТ

|  |  |
| --- | --- |
| ВСТУП………………………………………...………………………………. | 7 |
| РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ В ОБЛАСТІ РОЗПІЗНАВАННЯ ПІДПИСІВ КОРИСТУВАЧІВ ІНФОРМАЦІЙНИХ СИСТЕМ ……………………….………………………………………........... | 10 |
| 1.1 Проблематика розпізнавання підписів користувачів інформаційних систем..……………………………………………… | 10 |
| 1.2 Особливостi аналiзу бiометричних образiв у системах аутентифікації………………………………………............................ | 15 |
| 1.3 Аналіз нейромережевих засобів розпізнавання підписів користувачів………………………………………............................... | 20 |
| 1.4 Висновки до розділу ……………………………………………... | 35 |
| РОЗДІЛ 2. РОЗРОБКА НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ ТА СИСТЕМИ  РОЗПІЗНАВАННЯ РУКОПИСНОГО ПІДПИСУ..……............. | 36 |
| 2.1 Проектування системи розпізнавання підписів………………… | 36 |
| 2.2 Процедура визначення вхідних параметрів нейромережевої моделі………………………………….……......................................... | 40 |
| 2.3 Розробка базової нейромережевої моделі……………………… | 51 |
| 2.4 Адаптація нейромережевої моделі до розпізнавання підписів користувачів………………………………………............................... | 58 |
| 2.5. Висновки до розділу ..................................................................... | 65 |
| РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА  ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ  СИСТЕМИ ………………….…………………………….……… | 66 |
| 3.1 Опис інструментарію………………………………………........... | 66 |
| 3.2 Архітектура і основні характеристики………………………….. | 68 |
| 3.3 Експериментальні дослідження…………………………………. | 74 |
| 3.4. Висновки до розділу ..................................................................... | 80 |
| ВИСНОВКИ………………………………………........................................... | 81 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ…………………………………….. | 82 |
| ДОДАТОК А. Документи, що підтверджують впровадження результатів роботи…………………………………………………………………………. | 86 |
| ДОДАТОК Б. Фрагмент вихідного коду програмного забезпечення ........ |  |
| ДОДАТОК В. Слайди ………………………………...………………............ |  |

# ВСТУП

**Актуальність**. Оскільки біометричні технології аутентифікації, що використовуються на практиці, створено в рамках ймовірнісного підходу, то жодна з них не гарантує відсутність помилки. Тому підвищення достовірності ідентифікації особи є важливою задачею, актуальність розв’язання якої підсилюється необхідністю введення в Україні біометричних посвідчень особи, що є однією з умов її інтеграції в європейську спільноту. Одним з методів ідентифікації людини є розпізнавання її власноручного підпису. Підпис - один з класичних способів ідентифікації, який застосовується в юридичній практиці, банківській справі і торгівлі. Як правило, підпис не повторює звичайне написання букв і має додаткові елементи (розчерки, накладення букв і т. д.). Існують два способи ідентифікації людини за підписом: ідентифікація по динаміці підпису, що вводиться в електронний пристрій, і ідентифікація по зображенню підпису на документі. У першому випадку розглядається одновимірний сигнал, який формується електронним пером. У другому випадку, як правило, потрібно порівняти два зображення - еталонне і поточне. З цим завданням досить добре справляється людина, тому в даній роботі пропонується для автоматизації та оптимізації цього процесу використовувати нейронні мережі, що є одним із найбільш апробованих апаратів інтелектуальної обробки даних.

**Відомі підходи до вирішення поставленої задачі.** Для ідентифікації рукописного підпису найбільше використовуються методи, побудовані на аналізі траєкторії формування підпису. Але переважна більшість відомих методів ідентифікації підпису має модельний характер, оскільки вони не враховують перетин класів розпізнавання і довільні умови формування підпису, що робить необхідним на попередньому етапі ідентифікації проведення достатньо складної та ресурсоємної нормалізації образів. Разом з тим, сучасні досягнення в області застосування нейромережевих моделей та методів для аналізу текстової інформації дозволяють прогнозувати можливість їх ефективного застосування і для розпізнавання рукописних підписів користувачів. Цим пояснюється актуальність теми роботи, що пов’язана з розробкою нейромережевої системи розпізнавання підписів користувачів.

**Метою роботи** є підвищення ефективності засобів розпізнавання підписів користувачів, що використовуються в засобах біометричної аутентифікації.

Для досягнення поставленої мети вирішуються такі **задачі:**

* аналіз існуючих рішень в області розпізнавання підписів користувачів інформаційних систем;
* розробка нейромережевої моделі та системи розпізнавання рукописного підпису;
* розробка програмного забезпечення та експериментальне дослідження розробленої системи.

**Галузь застосування**. Розроблена модель та програмне забезпечення відносяться до галузі інформаційної безпеки і можуть бути використані для підвищення рівня захищеності інформаційних систем за рахунок використання сучаних нейромережевих рішень для аналізу рукописного підпису.

**Об'єктом дослідження** є процес розпізнавання підписів користувачів.

**Предметом дослідження** являються нейромережеві моделі та методи розпізнавання рукописних підписів користувачів.

**Методи дослідження** базуються на основі теорії нейронних мереж (для розробки моделі розпізнавання рукописних підписів), та об’єктно-орієнтованого програмування (для програмної реалізації розробленої системи).

**Новизна одержаних результатів полягає в наступному:**

* отримала подальший розвиток нейромережева модель, що за рахунок адаптації архітектурних параметрів дозволяє підвищити точність розпізнавання рукописних підписів.

**Практичне значення отриманих результатів:**

* розроблено інструмент для розпізнавання рукописних підписів користувачів інформаційних систем.
* розроблена методика проведення експерименту з використанням розробленого програмного засобу розпізнавання рукописних підписів користувачів інформаційних систем.

**Апробація.** Основні положення роботи доповідалися та обговорювалися на конференції:

* Комп’ютерне моделювання та інформаційні технології: матеріали першої науково-практичної конференції студентів, аспірантів та молодих вчених (Львів: кафедра інформаційних технологій НЛТУ України, 2019.)

# РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ В ОБЛАСТІ РОЗПІЗНАВАННЯ ПІДПИСІВ КОРИСТУВАЧІВ ІНФОРМАЦІЙНИХ СИСТЕМ

## 1.1. Проблематика розпізнавання підписів користувачів інформаційних систем

Принцип розпізнавання людини по рукописному пiдпису, в тому числi без використання електронних засобiв її обробки, застосовується не одне сторiччя. Цей спосiб розпізнавання особистостi побудований на тому, що автор придумує собi факсимiле, бажано, воно мало iстотнi вiдмiнностi за формою вiд класичного написання букв у виглядi додаткових елементiв (розчеркiв, повернень, накладення букв).

При розглядi даного типу розпізнавання необхiдно звернути увагу на наступнi незалежнi пiдходи: розпізнавання автора по статичному пiдпису еталон якого доступний системі аутентифікації; розпізнавання автора по динамiчному образу, що відтворюється автором пiдпису, під час входу в комп'ютерну систему при можливостi спостереження iндивiдуальних особливостей звичних для автора пiдсвiдомих рухiв.

«У першому випадку завдання розпізнавання особи користувача зводиться до порiвняння зображень. При цьому послідовність та технологія створення цих зображень зостаються поза увагою системи розпізнавання. У другому випадку відомими вважаються є данi про параметри коливання пера автора при вiдтвореннi їм пiдпису в тривимiрному просторi» [3]. Якщо користуватися тривимiрною системою координат (X, Y, Z), то данi про динамiку вiдтворення пiдпису отримують у виглядi двох функцiй часу коливань пера в площинi графiчного планшета X(t), Y(t) i ще однiєї функцiї - змiна тиску пера на графiчний планшет Z (t). Необхiдно враховувати, що рiшення першого пiдходу є важко реалiзованим для рiвня розвитку сучасних iнформацiйних технологiй i, як правило, автоматичнi банкiвськi системи iдентифiкацiї автографiв працюють iстотно гiрше досвiдченого касира. «Як правило, цi системи тiльки напiвавтоматичнi, вони полегшують роботу перевiряючого, даючи йому, вiдповiднi чисельнi характеристики близькостi фрагментiв пiдписи до оригiналу, але остаточне рiшення приймає людина» [5].

На сьогоднi, вiдомi технiчнi рiшення першого варiанту постановки задачi по iмовiрнiсним характеристикам iстотно гiрше, нiж статистика роботи досвiдчених людей (експертiв). Саме з цiєї причини фiрми виробники не дають статистичних даних про помилки першого i другого роду для iдентифiкацiї автора по статичному образу його пiдпису.

Другий пiдхiд навпаки дозволяє забезпечити набагато бiльшу точнiсть розпiзнавання при використаннi автоматизованих систем, що здiйснюють витяг чисельними параметроми вiдповiдностi. Дану особливiсть слiд враховувати при розглядi можливих варiантiв атак на систему, в тому числi, ситуацiї обвода, чужi пiдписи i зупинок для прийняття свiдомого рiшення пiд час пiдробки. Експерт не має подiбної динамiчної iнформацiї i вiдповiдно зростає ймовiрнiсть прийняття даного пiдпису, як дiйснго. За наявностю факту вiзуальної вiдповiдностi еталонного пiдпису. Практично всi сучаснi комерцiйнi систем iдентифiкацiї користувача по рукописному образу слiд роздiлити на одну, двох i трьох координатнстьi, вiдмiннiсть полягає в тому, що аналiз динамiки охоплює одну криву, пару кривих або повну множину кривих X (t), Y (t), Z ( t). Однокоординатнi системи можуть бути побудованi шляхом облiку будь-який з цих тимчасових функцiй, забезпечуючи ймовiрностi помилок першого i другого роду на рiвнi 0,1. Двох координатнi системи використовують будь-яку пару функцiй часу з трiйки X(t), Y(t), Z(t), i на сьогоднiшнiй день, дозволяють досягти рiвня ймовiрностi помилок близько 0,01. Найбiльш складнi системи використовують повну трiйку функцiй, забезпечуючи рiвень вiрогiдностi помилок першого/другого роду на рівні 0,003. Слiд зазначити, що деякi з систем бiометричної iдентифiкацiї за пiдписом використовують не самi проекцiї функцiй X(t), Y(t), Z(t) на осi координат, а їх першу або другу похiдну. Дана особливiсть обумовлена ​​тiльки типом використовуваного датчика, чутливого до похiдної, i практично не впливає на якiсть i обсяг вихiдної iнформацiї з нього одержуваної. «Штучне нарощування числа аналiзованих функцiй за рахунок вимiрювання їх самих i їх же похiдних недоцiльне, так як всi лiнiйнi перетворення первiсних дають сильно корельованi данi» [12]. Зокрема, кореляцiя бiометричних даних, отриманих з первiсної X(t), i бiометричних даних отриманих з її похiдною dX(t)/dt, наближаєьтся до одиницi. Необхiдно пояснити, що помилка першого роду або помилкова вiдмова справжньому користувачевi системи з ймовiрнiстю 0,01 - це цiлком прийнятна характеристика для бiльшостi областей застосування даного виду систем iдентифiкацiї. Iнше працювати з помилками другого роду або хибним пропуском зловмисника. Для ряду практичних застосувань ймовiрнiсть помилок другого роду 0.01 є неприйнятно великою величиною. Для зниження цiєї помилки вдаються до розпізнавання особи по динамiцi вiдтворення бiометричного пароля. В якості біометричного паролю зазвичай використовується якась пов'язана послiдовнiсть літер. Це може бути, наприклад, слово зi словника, або група слiв. Автором зберiгається в секретi як саме слово - пароль, так i iндивiдуальнi особливостi його написання. В цьому випадку для парольного слова, що мiстить вiд 5 до 10 букв, похибка другого роду зменшується в 104 ... 106 разiв. Бiльшого зниження помилки другого роду можна домогтися, використовуючи в якостi бiометричного пароля слова, якi не мають сенсу, але цей шлях пов'язаний з додатковими труднощами запам'ятовування слiв з випадкового поєднання букв. У свою чергу це викликає виникнення уразливостi запису користувачем складного пароля на будь-який небезпечний носiй, наприклад, аркуш паперу, що безумовно знижує ефективнiсть даного виду захисту лише до рiвня забезпечуваного iндивiдуальнiстю динамiки пiдпису.

Зазвичай динамiчнi параметри, що враховуються при розпізнаванні особи користувача, отримують шляхом обчислення деяких лiнiйних функцiоналiв при повнiй реалiзацiї пiдпису або при реалізації деякого його фрагменту. Відповідно, глобальними динамiчнi параметри називаються параметри, обчисленi по повнiй реалiзацiї кривих Y(t), X(t), Z(t). Локальними динамiчними параметрами називаються аналогiчнi параметри, але обчисленi на деяких вiдрiзках повних кривих Y(t), X(t), Z (t). Це можуть бути, наприклад, окремі букву парольного слова. При обчисленнi вказаних лiнiйних функцiоналiв зручно використовувати ортогональнi перетворення Фур'є, Уолша, Хаара. Вибiр вказаних ортогональних перетворень обумовлений тим, що вони дають данi з меншою внутрiшньою кореляцiєю в порiвняннi з iншими не ортогональними функцiоналами. Для визначеностi достатньо часто використовують функцiонали Фур'є. В цьому випадку глобальнi динамiчнi параметри будуть мати сенс коефiцiєнтiв ряду Фур'є при синусах i косинусах. За звичай враховується кiнцеве число коефiцiєнтiв ряду Фур'є. Лiнiйнiсть використовуваних ортогональних функцiоналiв дозволяє вкрай просто здiйснювати операцiю масштабування сигналiв по осях X, Y, Z.

Для приведення даних до масштабу першого введеного пiдпису досить помножити значення обчислених лiнiйних функцiоналiв на отриманi з рiвнянь масштабнi коефiцiєнти. Слiд мати на увазi, що проблема приведення до єдиного масштабу даних досить важлива i має коректне рiшення тiльки в класi лiнiйних функцiональних перетворень. При використаннi нелiнiйних функцiоналiв ця проблема може не мати коректного рiшення.

Локальнi динамiчнi параметри обчислюються так само, як i глобальнi, з тiєю лише рiзницею, що перiод iнтегрування береться рiвним деякому спостерiгаєму локальному вiдрiзку часу. Зокрема локальнi динамiчнi параметри можуть бути обчисленi на 7 iнтервалах торкання пера графiчного планшета, мiж 8 вiдривом пера (8 синхронними зверненнями в нуль функцiй Y(t), X(t)). Крiм вiдривiву пера вiд планшета для дроблення пiдписи на фрагменти можуть використовуватися перетин фрагментiв траєкторiї пiдпису, точки змiни напрямку руху, кутовi точки літер, що мають розрив похiдної. Зазвичай число видiлених локальних дiлянок пiдпису в першому наближеннi виявляється пропорцiйним числу літер в рукописному словi (коефiцiєнт пропорцiйностi знаходиться в межах вiд 1 до 2,4 в залежностi вiд використаних алгоритмiв фрагментування). На кожній з локальних дiлянок обчислюється порiвняно невелике число параметрiв (вiд 2 до 6).

Слiд пiдкреслити, що спiввiдношення мiж глобальними i локальними динамiчними параметрами має бути розумно збалансовано. Глобальнi динамiчнi параметри описують схожiсть пiдпису на зразок в цiлому, а локальнi динамiчнi параметри дозволяють оцiнювати схожiсть вiдтворення на еталон кожної з літер, кожного фрагменту пiдпису. При машинальному пiдсвiдомому вiдтвореннi слова з певною частотою з'являються новi додатковi фрагменти пiдпису. При цьому так само можуть зникати (зливатися) ранiше роздiленi сусiднi фрагменти. В результатi подрібнення пiдпису на фрагменти виявляється неоднозначним, а при навчаннi фiксується декiлька варiантiв пiдпису характерних для одного автора. З ситуацiї неоднозначної розмiтки пiдпису можливі два виходи. Перший вихiд пов'язаний з формуванням декiлькох незалежних еталонiв. При цьому для формування кожного еталона потрiбно декiлька пiдписiв, що ускладнює процес навчання системи.

Другий шлях пов'язаний з приведенням всiх варiантiв до деякого усередненого варiанту. Бiометричний еталон для цього класу систем формується на етапi навчання i повинен вiдображати як стабiльну, так i нестабiльну складовi бiометричних даних розпізнавання особи.

Бiометричний зразок пiдпису (парольного слова) може формуватися декiлькома способами залежно вiд прийнятого в системi правила прийняття рішення. Зокрема, якщо рiшення в системi приймається нейронною мережею, то роль бiометричного зразка гратимуть структура штучної нейронної мережi, вид i параметри функцiй актьивації. У разi, коли рішення примається за рахунок використання детермiнованого вирiшального правила, роль бiометричного зразка можуть грати вектор значень математичних очiкувань вимiрюваних динамiчних параметрiв i вектор значень дисперсiй зафіксованих динамічних параметрів. Однiєю з основних проблем систем iдентифiкацiї цього класу є те, що їх параметри iстотно залежать вiд психологiчного стану людей i стабiльностi їх почерку. Для людей зi стабiльним рукописним почерком цi системи працюють значно краще, нiж для середнього людини, але, в той же час, системи цього класу можуть бути скомпроментовані людьми з нестабiльним почерком. Крiм того, введення бiометричної авторизацiї та iдентифiкацiї, сильно обмежують свободу дiй користувачiв в корпоративних мережах, перший час можуть спостерiгатися спроби навмисної компроментації бiометричних систем через штучну симуляцiю нестабiльного почерку.

## 1.2 Особливостi аналiзу бiометричних образiв у системах аутентифiкацiї

Для того щоб скористатися бiометричним захистом, необхiдно вмiти перетворювати багатовимiрний континуум бiометричних даних образу «Свiй» в код криптографiчного ключа доступу. Для цього потрiбно вiдсканувати заявлений бiометричний образ i обчислити контрольованi бiометричнi параметри [6]. Далi слiд пред'явити бiометричнi параметри на вхiд перетворювача бiометрiя-код, як це показано на рис. 1.1. Сам перетворювач бiометрiї в код може бути виконаний рiзними технологiями, наприклад, можуть бути використанi так званi «нечiткi екстрактори» i [11]. Шляхом створення «нечiтких екстракторiв» йдуть в основному англомовнi дослiдники.

База образiв «Чужі»

Образ «Свій»

Сканування даних

Розрахунок параметрiв

Перетворення бiометрiя-код

N=512

n=256

Рис. 1.1 Перетворення бiометричних параметрiв в код ключа

В багатьох країнах світу перетворювачi бiометрiя-код будуються з використанням штучних нейронних мереж [12]. Необхiдно заздалегiдь вибрати структуру зв'язкiв штучних нейронiв мережi, а потiм навчити її [13] так, щоб приклади бiометричного образу «Свiй» давали на виходi перетворювача код особистого криптографiчного ключа громадянина України, Росiї або Казахстану, а для прикладiв образiв «Чужий» на виходi перетворювача виходила випадкова кодова комбiнацiя. Незалежно вiд того, яка технологiя використовується, перетворювач бiометрiя-код завжди має бiльше вхiдних бiометричних даних, нiж число його виходiв (нiж довжина вихiдного криптографiчного ключа). Так, на рис. 1.1 перетворювач має N = 512 вхiдних бiометричних параметрiв i тiльки n = 256 вихiдних розрядiв криптографiчного ключа. Умова N=n завжди виконується через низьку iнформативнiсть бiометричних даних. Одного бiометричного параметра, як правило, не вистачає для отримання одного бiта криптографiчного ключа. Обов'язково необхiдно використовувати надлишкове число бiометричних параметрiв для того, щоб виправляти помилки кодування в «нечiтких екстракторах» або «збагачувати» вхiднi бiометричнi данi нейронною мережею перетворювача. Ще однiєю важливою причиною труднощiв бiометричної аутентифiкацiї є висока розмiрнiсть завдання (доводиться враховувати 512 i бiльше «поганих» бiометричних параметрiв). Саме з цiєї причини не вдається скористатися апаратом класичної лiнiйної алгебри та багатовимiрної статистики. Положення ускладнюється тим, що перетворювачi бiометрiя-код доводиться навчати (налаштовувати) на малому числi прикладiв в навчальнiй вибiрцi.

Так, при навчаннi бiометричних засобiв аутентифiкацiї користувачi готовi пред'явити вiд 10 до 20 прикладiв бiометричного образу «Свiй». Однак, якщо їх попросити пред'явити 100 або 200 прикладiв, то ця робота сприймається користувачами негативно. На сьогоднi користувачi не готовi докладати значних зусиль для навчання своєї бiометричної програмної роботи (перетворювача бiометрiя-код). Останнє означає, що 512-мiрнi розподiлу континумів параметрiв образу «Свiй» ми змушенi представляти всього 20 прикладами по кожному з параметрiв. Ми можемо спробувати уявити 512-мiрний об'єм природно-континуальної форми подання даного способу «Свiй» через внутрiшнiй обсяг графа 20 прикладiв «Свiй», однак це буде дуже i дуже слабким наближенням. Користуючись всього 20 прикладами, неможливе точне обчислення математичного сподiвання бiометричних параметрiв E(vi), їх середньоквадратичне вiдхилення  i коефiцiєнти кореляцiї мiж параметрами ri, j. На навчальнiй вибiрцi з 20 прикладiв вiдносна помилка в оцiнцi математичного очiкування може бути вiд 0 до25%, вiдносна помилка середнє вiдхилення може становити вiд 0 до 50%, вiдносна помилка оцiнки коефiцiєнтiв кореляцiї може бути вiд 0 до 100%. При настiльки неточних даних неможливо побудувати багатовимiрну аналiтичну модель розподiлу даних образу «Свiй».

Необхiдно вiдзначити, що унiкальнiсть вiдкритих рукописних образiв не велика в порiвняннi з вiдбитками пальцiв або райдужної оболонкою ока. Так, якщо зловмисниковi вiдомий рукописний пароль з 5 букв i його накреслення, то вiн входить в систему з ймовiрнiсть 0.01. Для пiдбору динамiки вiдтворення рукописного способу досить близько 100 спроб. Однак кожне нове слово, яке вiдрiзняється хоча б одним знаком ми повиннi розглядати, як абсолютно нове слово. Динамiка рукописного написання букв iстотно змiнюється в залежностi вiд попереднiх знакiв.

Сучаснi бiометричнi системи захисту повиннi вмiти обчислювати якiсть всiх контрольованих бiометричних параметрiв i вмiти прогнозувати свою стiйкiсть до злому простим перебором можливих поєднань бiометричних параметрiв. Сьогоднi виробниками декларуються тiльки середньостатистичнi показники стiйкостi бiометричної системи, що явно недостатньо. Реальнi характеристики бiометричних систем при роботi з конкретними користувачами можуть вiдрiзнятися вiд середньостатистичних на десятки порядкiв. Тому виникає необхiднiсть вiдiйти вiд абстрактного поняття середньостатистичний користувач i перейти до оцiнки стiйкостi, а вiдповiдно iнформативного способу конкретного користувача системи.

На рис. 1.2 наведенi процентнi спiввiдношення людей в кожному класi i стiйкiсть класiв по вiдношенню до атак пiдбору за умов вiдомого рукописного слова i невiдомого рукописного слова.

Найбiльш стабiльний клас користувачiв при збереженнi в таємницi бiометричного способу має ймовiрнiсть помилки другого роду на рiвнi 10-33. Найстабiльнiший сьомий клас користувачiв взагалi не може бути однозначно пiзнаний системою. Середньостатистичний користувач має можливiсть помилки другого роду на рiвнi 10-9. У пiдсумку можна зробити висновок про те, що люди з унiкальним i стабiльним почерком мають ймовiрнiсть помилки другого роду на 24 порядки менше, нiж середньостатистичний користувач. Таким чином, стiйкiсть системи багато в чому визначається iндивiдуальними характеристиками самого користувача.

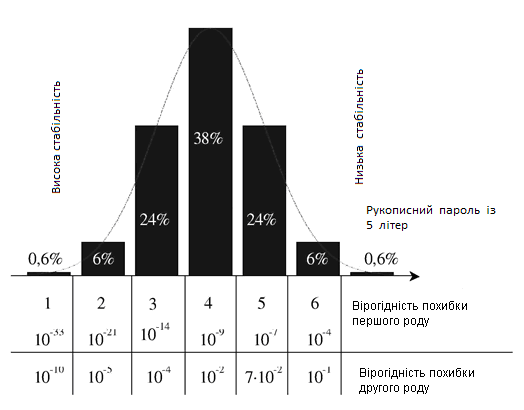


Рис. 1.2 Стiйкiсть класiв стабiльностi рукописного введення по вiдношенню до атак пiдбору

Помилка у визначеннi класу може привести до завищення або заниження стiйкостi системи на кiлька порядкiв. Необхiдно використовувати спецiальнi нейромережевi механiзми для коректного i достовiрного визначення класу користувача по його реальним бiометричними параметрами. Для систем заснованих на пред'явленнi вiдбиткiв пальцiв групи користувачiв по стабiльностi не видiляються на увазi статичностi образу. Атаки на етапах введення образу i процедури криптографiчний аутентифiкацiя по ключу (паролю) не розглядаються, так як успiшнiсть першої не може бути описана математично, а друга ставитися до областi криптографiї та еквiвалентної атаки на систему з дуже довгим ключем. Оцiнка iнформативностi за допомогою реалiзацiї атаки пiдбору бiлим шумом. Даний вид атаки заснований на прагненнi зловмисника здiйснити повний перебiр i подальшу установку параметрiв системи, якi прийме система, визнавши його легiтимним користувачем. Атаки даного типу повиннi бути менш бажаними, нiж заснованi на кореляцiйному принципi, так iстотно зростає потужнiсть безлiчi можливих комбiнацiй параметрiв, адже вони вибираються не згiдно з яким-небудь законом розподiлу, а на пiдставi рiшення генератора псевдовипадкових чисел. Практичнi дослiдження показали, що данi атаки представляють найбiльшу загрозу для систем високонадiйнiй бiометричної аутентифiкацiї, так як, не вимагають знання про особливостi користувачiв системи, математичного обґрунтування вибору кореляцiйних матриць. Даний тип атак доцiльно застосовувати лише на етапi перетворення вектора бiопараметров в ключ, в тому числi не тiльки подачею на вхiд нейромережi, а й на окремi її дiлянки.

Стiйкiсть класiв стабiльностi рукописного введення по вiдношенню до атак пiдбору випадковим сигналом представлено на рис. 1.3. До переваг даного пiдходу можна вiднести незалежнiсть вiд використовуваного бiометричного способу лише вiд конфiгурацiї нейромережi.

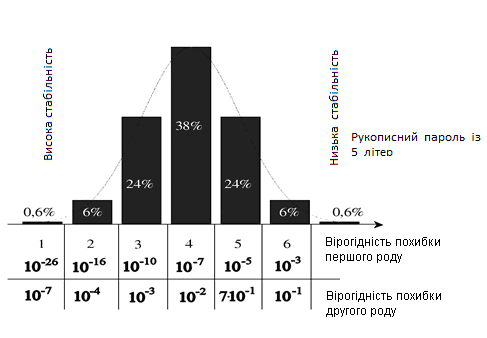


Рис. 1.3 Стiйкiсть класiв стабiльностi рукописного введення по вiдношенню до атак пiдбору випадковим сигналом

У пiдсумку можна сформулювати висновок про те, що оцiнити iнформативнiсть бiометричних образiв можливо на пiдставi показникiв точності нейромережевої системи аутентифiкацiї.

## 1.3 Аналіз нейромережевих підходів до розпізнавання підписів користувачів

В роботах [1-4] відзначається, що один із найпоширеніших напрямків розвитку нейромережевих технологій пов'язаний з біометрією - технологією, що вимірює і аналізує різні фізичні характеристики людини. Вона пропонує автоматизовані заходи нейромережевої верифікації (нейромережевого порівняння зразка з біометричними шаблоном) і нейромережевої ідентифікації (нейромережевого порівняння зразка зі багатьма з бази даних) особи, засновані на фізіологічних (наприклад, відбитки пальців, райдужна оболонка ока, сітківка ока, форма обличчя) і поведінкових (наприклад, мова, почерк, хода людини і т.д.) принципах. При цьому нейромережева верифікація рукописного підпису знаходиться серед найбільш поширених методів, що використовуються наприклад, при проведенні фінансових і комерційних транзакцій, перевірці документів і контролі фізичного доступу.

В якості джерела інформації для аналізу нейромережевих засобів розпізнавання підписів використано роботи [4-12] в яких наведено опис застосування нейронних мереж з метою аутентифікації користувачів на основі аналізу їх рукописного підпису. Так в роботі [6] зазначено, що найбільш найбільш природний і давно використовуваний метод перевірки особи - це перевірка її рукописного підпису. Також зазначається, що по відношенню до інших методів біометричної аутентифікації верифікація рукописного підпису є більш достатньо поширеним методом перевірки особи. Розпізнавання підписів пропонується розглядати як задачу класифікації зображень. Процес класифікації в цьоиу випадку полягеє у введенні в систему розпізнавання вхідного зображення і виведення нею назви класу (ймовірності класу), що найкраще описують зображення. Також факт того, що рукописний підпис широко використовується як засіб перевірки особи, підтверджує необхідність використання відповідної системи автоматичної перевірки. Відзначається, що автономні системи ідентифікації працюють з відсканованим зображенням підпису, що вказує на можливість його розпізнавання за рахунок нейромережевих засобів комп'ютерного зору. Ця можливість базується на тому, що нейронні мережі більш як півстоліття є фундаментальною частиною комп'ютеризованих задач розпізнавання образів, і можуть використовуватися в дуже широкому діапазоні проблемних областей. Підхід до вирішення завдання за допомогою нейронних мереж має такі переваги, уніфікація процедури визначення і класифікації ознак, а також гнучкість пошуку оптимальних рішень. В роботі [1] стверджується, що найбільш відповідною основою для розробки автоматичної системи класифікації зображень є згорткова нейронна мережа типу LeNet, структура якої показана на рис. 1.4.

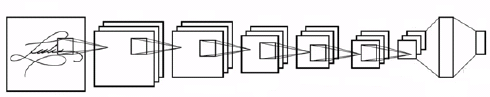


Рис. 1.4 Структура згорткової нейронної мережі типу LeNet

Дана мережа складається з двох подібних ділянок, які містять по два шари згортки і по одному шару підвибірки. Ділянки призначені для виділення основних ознак рукописного підпису. В кінці мережі розташований класифікатор, який складається з одного повнозв'язну шару, що містить 512 нейронів, і вихідного шару, який містить 10 нейронів. На вхід мережі надходять кольорові зображення розміром 32 на 32 пікселів. Використовується три канали кольору: червоний, зелений і синій. В першому шарі згортки міститься 32 карти ознак. При цьому розмір ядра згортки 3 на 3. Тобто кожен нейрон шару згортки підключений до квадратної ділянки вхідного зображення розміром 3 на 3. Наступний шар згортки має таку ж архітектуру. В цьому шарі також містяться 32 карти ознак з ядром згортки 3 на 3. Після цього йде шар підвибірки, на якому виконується зменшення розмірності зображення. Зміна розмірності виконується для кожної карти ознак окремо, тому тут також використовується 32 карти ознак. Розмір поля підвибірки 2 на 2. Після шару підвибірки починається новий каскад шарів згортки. На третьому і на четвертому шарі згортки використовується карти ознак розміром 3 на 3, а на другому шарі підвибірки, яка йде після цих шарів згортки, також відбувається зменшення розмірності в квадраті 2 на 2. Після цього дані перетворюються з двовимірного формату в одновимірний і передаються на повнозв'язні шари нейронів, які здійснюються остаточну класифікацію.

У згорткових мережах використовується підхід навчання з учителем. Тобто для побудови згорткової нейронної мережі необхідно мати розмічений набір даних. В випадку аутентифікації користувача використовуються рукописні підписи осіб. В роботі [11] кожен користувач розписався ручкою на листку вісім разів. Після цього, рукописні підписи були переведені в зображення розміром 32 на 32 пікселів. Зображення мали растровий формат png. Приклади зображень підписів наведені на рис. 1.5.



Рис. 1.5 Приклади зображень підписів користувачів

Перед навчанням зображення рукописних підписів розподілили по двом наборам: набір для навчання (зображення і класів) і набір для тестування (зображення і мітки класів). Набір для навчання складався з 60 зображень, а набір тестування складався з 20 зображень. Мітки класів містили ідентифікатори користувачів, рукописні підписи яких представлено на зображенні. Описана нейромережева модель була реалізована за допомогою відкритої бібліотеки Keras. Навчання мережі на описаному наборі растрових зображень виконувалося протягом 25 епох. Термін навчання тривав приблизно 8 хвилин. Точність розпізнавання навчальних даних склала приблизно 95%. Після того, як навчання завершилося, точність роботи мережі перевірили на тестових даних, які не були представлені у навчальних прикладах. Точність розпізнавання тестових даних склала приблизно 91%. В роботі сформовано висновок про те, що результати експериментів свідчать про високу ефективність побудованої нейромережевої моделі, а відповідно і про перспективність застосування засобів аутентифікації користувачів на основі аналізу їх рукописних підписів.

Робота [12] присвячена оцінці перспектив використання нейронних мереж для ідентиіації рукописних підписів. Зазначається, що системи верифікації підпису націлені на ідентифікацію особистості людини, базуючись на розпізнаванні його рукописного підпису. Незважаючи на багато досягнень в цій галузі за останні кілька десятиліть, створення систем, які можуть розрізняти справжні підписи і фальшивки, як і раніше є важким завданням, про що свідчать високі коефіцієнти помилок, отримані при тестуванні даних систем на великих наборах даних. Для того, щоб успішно вирішити дану проблему, пропонується використовувати штучні нейромережі. Штучна нейронна мережа - математична модель, а також її програмна або апаратна реалізація, побудована за принципом організації та функціонування біологічних нейронних мереж - мереж нервових клітин живого організму. Це поняття виникло при вивченні процесів, що протікають в мозку, і при спробі змоделювати ці процеси. Першою такою спробою були нейронні мережі У. Маккалока і У. Піттса. Після розробки алгоритмів навчання, одержані моделі стали використовувати в практичних цілях: в задачах прогнозування, для розпізнавання образів, в задачах управління та ін. При розпізнаванні образів в якості самих образів можуть виступати різні за своєю природою об'єкти: символи тексту, зображення, зразки звуків і т.д. При навчанні мережі пропонуються різні зразки образів із зазначенням того, до якого класу вони відносяться. Зразок, як правило, представляється як вектор значень ознак. При цьому сукупність усіх ознак повинна однозначно визначати клас, до якого належить зразок. У разі якщо ознак недостатньо, мережа може неправильно співвіднести один і той же зразок з декількома класами. Після закінчення навчання мережі їй можна пред'являти невідомі раніше образи і отримувати відповідь про належність до визначеного класу. З іншої точки зору штучна нейронна мережа представляє собою систему з’єднаних і взаємодіючих між собою простих процесорів (штучних нейронів). Функціонуання кожного із процесорів нейронної мережі полягає в тому, що він періодично отримує сигнали від інших процесорів, оброблює їх та періодично посилає власний сигнал іншим процесорам. Доведено, що з’єднані в досить велику мережу з керованою взаємодією, такі окремо прості процесори разом здатні виконувати досить складні завдання. Характерним етапом побудови нейромережевої моделі є її навчання. Технічно навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. В процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення. Це означає, що в разі успішного навчання мережа зможе повернути вірний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних і/або «зашумлених», частково викривлених даних. Системи верифікації підписів працюють із зображеннями даних підписів і це є завданням розпізнавання образів. З такими завданнями добре справляються згорткові нейронні мережі. Свою назву архітектура мережі отримала через наявність операції згортки, суть якої в тому, що кожен фрагмент зображення поелементно множиться на матрицю (ядро) згортки, а результат підсумовується і записується в аналогічну позицію вихідного зображення. Таким чином, використовуючи згорткові нейронні мережі, можна в 80-90 відсотках випадків домогтися вірної ідентифікації підпису. При цьому головним мінусом даного підходу є те, що для навчання нейромережі необхідний досить великий набір зображень з підписом людини, чия особистість буде ідентифікуватися. Дані системи можуть бути успішно застосовані в різних сферах. Наприклад, в університеті для перевірки того, що викладач дійсно сам поставив свій підпис в заліковці або відомості, а не студент підробив його підпис або для перевірки того, що студент сам писав певну перевірочну роботу. Для цього необхідно зібрати чималу кількість відсканованих підписів викладачів і студентів, потім за допомогою них навчити нейромережу і зберегти навчену модель. Надалі при перевірці справжності підпису потрібно буде тільки отримати його зображення і за допомогою навченої моделі нейромережі провести ідентифікацію.

Задекларованою метою роботи [5] стала розробка алгоритму розпізнавання рукописних підписів на зображеннях, що забезпечує високий відсоток точного розпізнавання. Наведено обґрунтування розробки. Вказано, що системи розпізнавання зображень знаходить широке застосування в різних областях: це може бути класифікація об'єктів, розпізнавання осіб, символів, маркування об'єктів, автомобільних номерних знаків, відбитків пальців і т.д. На сьогоднішній день дані технології застосовуються в стільникових телефонах, планшетних комп'ютерах, системах контролю доступу до секретних об'єктах, оперативному пошуку в картотеці і т.д. Однак досі при вирішенні задач в цій області виникає ряд складних наукових і технічних проблем. У зв'язку з цим можна припустити, що в найближчому майбутньому будуть актуальні завдання ідентифікації та розпізнавання рукописних підписів, тому що підпис кожної людини може бути важливим ідентифікатором особистості. Визначено, що в даний час не існує програмного забезпечення, яке вирішує це завдання. Рукописні підписи мають досить складну структуру і дрібну деталізацію, все це представляє велику складність для вирішення даного завдання математичними методами і вимагає великих обчислювальних витрат. При аналізі методів розпізнавання образів на зображеннях було вирішено використовувати штучні нейронні мережі, тому що вони можуть виконувати роль класифікатора, добре моделює складну функцію розподілу рукописних підписів на зображеннях, тим самим збільшуючи точність розпізнавання порівняно з іншими методами. Також одне із завдань роботи [5] полягало у порівнянні ефективності застосування для розпізнавання підписів згорткової нейронної мережі з класичним багатошаровим персептроном. Для цього вказані нейромережеві моделі були програмно реалізовані, що надало змогу експериментально порівняти результати їх роботи. Розроблені нейронні мережі навчалися і тестувалися на одних і тих же базах даних зображень.

Зроблено припущення про доцільність розпізнавання підписів за допомгою згорткових нейронних мерж, архітектура яких адаптована для роботи з двовимірними поверхнями і забезпечує високу ступінь інваріантності до просторових спотворень вхідних даних, таких як зміна масштабу, зміщення, поворот, зміна ракурсу і т.д. Вхідний шар являє собою площину, розміри якого відповідають розміру вхідного зображення. Приховані шари складаються з набору площин і призначені для виділення характерних ознак з вхідного зображення. Кожен нейрон шару пов'язаний з областю нейронів попереднього шару (локальні рецепторні поля), причому всі нейрони в площині мають однакові ваги. Це забезпечує знаходження характерних ознак в будь-якому місці зображення і зменшує загальне число вагових коефіцієнтів. З кожним наступним шаром кількість площин в шарах збільшується, а їх розміри зменшуються. На кожному наступному рівні виділяються все більш абстрактні ознаки класифікуються об'єктів. Останні шари складаються з простих сигмоїдальних нейронів, які проводять фінальну класифікацію ознак, виділених в попередньому шарі. Проведено дослідження різних конфігурацій згорткової нейронної мережі по такими параметрами: число шарів, число площин в шарах, число зв'язків між шарами, функції активації нейронів. В ході експериментальних досліджень було встановлено, що для вирішення поставленого завдання ефективніше використовувати згорткову нейронну мережу без шарів підвибірки. Це твердження базувалось на тому, що застосування шарів підвибірки може призвести до втрати дрібних характерних ознак рукописних підписів. Найкраща якість класифікації було досягнута при використанні мережі з конфігурацією, що представлена на рис. 1.6. На рис. 1.6 цифрою 1 позначно вхідний шар, цифрами 2, 3 – згорткові шари, а цифрами 4 та 5 шари звичайних нейронів. Розмір вхідного шару - 40 на 80 нейронів, що відповідає розміру розпізнаються підписів. Другий шар є згортковим і складається з 8 площин розміром 20 на 40 нейронів. Третій шар також є згортковим і складається з 16 площин розміром 12 на 22 нейронів. Четвертий шар також є згортковим і складається з 48 площин розміром 6 на 10 нейронів. П'ятий шар складається з 100 звичайних сигмоїдальних нейронів. Шостий шар є вихідним і складається з 15 нейронів, що відповідають 15 типам рукописних підписів. Розмір локального рецептивного поля дорівнює чотирьом пикселям.



Рис. 1.6. Архітектура згорткової нейронної мережі, пристосованої до розпізнавання рукописних підписів.

У табл. 1.1 представлено параметри ​​конфігурації описаної згорткової нейронної мережі.

*Таблиця 1.1*

Конфігурація згорткової нейронної мережі

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер шару | 1 | 2 | 3 | 4 | 5 | 6 |
| Кількість шаів | 1 | 8 | 16 | 48 | 100 | 15 |
| Розмір шару | 40х80 | 20х40 | 12х22 | 6х10 | 1 | 1 |
| Розмір ядра згортки | 4х4 | 4х4 | 4х4 | 4х4 | 1 | 100 |

Згорткові шари здійснюють фільтрацію зображень прямокутної маскою з загальних синаптичних коефіцієнтів нейронів. Результат фільтрації після нелінійного перетворення функцією активації відображається на виходи нейронів. Процес функціонування нейрона шару згортки задається виразом:

, (1.1)

де  - вихідний сигнал (*i,j*)-го нейрону *k*-ої карти ознак;

 - зсув нейронів *k*-ої карти ознак;

*К* - розмір рецептивної області нейрону (розмір ядра згортки);

 - коефіцієнт (*s,t*)-го синаптичного зв’язку нейрону *k*-ої карти ознак;

*x* - вихід нейрону попереднього шару.

Поступово згорткова нейронна мережа навчається виділяти ключові характеристики рукописних підписів, що надходять на вхід у вигляді зображення.

Була протестована робота двох видів активаційних функцій: сигмоїдальної і гіперболічного тангенсу. Кращі результати показала функція гіперболічний тангенс виду:

. (1.2)

В роботі [9] виявлено, що до основних переваг цієї функції відносяться: симетричність функції забезпечує високу збіжність, функція має неперервну першу похідну, функція має просту похідну, що може бути обчислена через її значення. Інтервал значень цієї функції знаходиться в межах від -1 до 1, тому значення вихідних сигналів вихідних нейронів мережі також знаходяться в даному інтервалі.

Для порівняння ефективності двох архітектур була розроблена класична нейронна мережа типу багатошарового персептрону. В процесі розробки класичної нейронної мережі було проведено дослідження різних конфігурацій мережі за наступними параметрами: число шарів, число нейронів в шарах, число зв'язків між нейронами. Найкраща точність класифікації була досягнута при використанні мережі з конфігурацією, представленої на рис. 1.7.

На рис. 1.7 цифрою 1 позначено вхідний шар нейронів, цифрою 2 – схований шар, а цифрою 3 – вихідний шар. Так як розмір зображень рукописних підписів становить 40 на 80 пікселів, то відповідно вхідний шар складається з 3200 (40 × 80) нейронів. Другий шар складається з 30 нейронів. Вихідний шар складається з 15 нейронів, що відповідають 15 типам рукописних підписів.

Структура зв’язків між нейронними шарами є повнозв'язною. В якості функції активації використано гіперболічний тангенс, тому що він показав кращі результати в порівнянні з сигмоїдальною функцією активації.

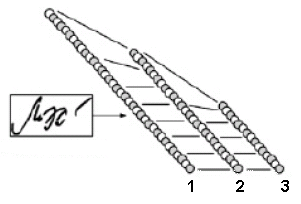


Рис. 1.7 Структура двохшарового персептрону, призначеного для розпізнавання рукописних підписів

Параметри конфігурації побудованого двохшарового персептрону наведені в табл. 1.2.

*Таблиця 1.2*

Параметри конфігурації двохшарового персептрону, призначеного для розпізнавання рукописних підписів

|  |  |  |  |
| --- | --- | --- | --- |
| Номер шару | 1 | 2 | 3 |
| Кількість нейронів | 3200 | 30 | 15 |
| Кількість синаптичних зв’язків | 96000 | 96450 | 450 |

Програмна система розроблялася і тестувалася на процесорі Intel (R) Core (TM) 2 Quad CPU 3.00 GHz. Процес реалізації програмного забезпечення проходив в середовищі розробки Visual Studio 2010 року на мові С ++. Для навчання побудованих нейронних мереж була створена база даних з 2000 зображень рукописних підписів, що належать різним людям.

Для тестування мереж також була створена окрема база даних з 2000 зображень рукописних підписів. Навчальна та тестова бази даних не містять однакових зображень підписів, тобто кожне зображення підпису є унікальним. Дані, які подаються на вхід нейронних мереж, повинні бути однаковими з точки зору формату і розміру. Тому для вхідних зображень передбачена процедура попередньої обробки відповідно до наступного алгоритму:

1. Застосування фільтра, що згладжує. Фільтр представляє собою одиничну матрицю 3 на 3 пікселі та використовується для очистки зображення від шумових завад.

2. Переведення зображення в чорно-білий формат із зазначенням мінімального порога яскравості.

3. Центрування рукописного підпису.

4. Зміна розміру отриманого зображення. Розмір вхідних зображень приводиться до фіксованого значення 40 на 80 пікселів.

Навчання нейронних мереж проводилося трьома різними алгоритмами: Backpropagation, Quickprop, Левенберга-Марквардта. Для згорткової нейронної мережі найкращу якість навчання було досягнуто при використанні алгоритму навчання Левенберга-Марквардта. Однак недоліком застосування даного алгоритму став довгий час навчання, що в кілька разів перевищує час навчання при застосуванні інших алгоритмів. Число невірно класифікованих зображень тестової вибірки дорівнює 32. Час розпізнавання рукописних підписів згортковою нейронною мережею склав 21 мс. При використанні класичної нейронної мережі алгоритм Левенберга - Марквардта також показав кращі результати в порівнянні з іншими алгоритмами, при найбільшому терміні навчання. Кількість невірно класифікованих зображень тестової вибірки становить 83 зображення. Термін розпізнавання рукописних підписів класичною нейронною мережею склав 40 мс. Результати навчання та тестування згорткової нейронної мережі наведені в табл. 1.3, а результати навчання та тестування двохшарового персептрону наведені в табл. 1.4.

*Таблиця 1.3*

Результати навчання та тестування згорткової нейронної мережі

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Алгоритм навчання** | **Похибка при навчанні** | **Похибка при тестуванні** | **Кількість епох навчання** | **Термін навчання**  **(години)** |
| Backpropagation | 26,2 % | 31,8 % | 49 | 5,1 |
| Quickprop | 3,4 % | 3,2 % | 15 | 4,8 |
| Левенберга-  Марквардта | 0,8 % | 1,2 % | 24 | 17,7 |

*Таблиця 1.4*

Результати навчання та тестування двохшарового персептрону

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Алгоритм навчання** | **Похибка при навчанні** | **Похибка при тестуванні** | **Кількість епох навчання** | **Термін навчання**  **(години)** |
| Backpropagation | 38,27 % | 40,14 % | 61 | 4,5 |
| Quickprop | 8,58 % | 10,2 % | 19 | 4,1 |
| Левенберга-  Марквардта | 6,2 % | 6,8 % | 27 | 12,3 |

Як свідчать дані табл. 1.3 і 1.4 при однакових алгоритмах навчання і однакових навчальних вибірках згорткова нейронна мережа показує кращі результати, ніж двохшаровий персептрон. Тому можна зробити висновок, що застосування класичних нейромережевих архітектур в даній задачі неефективно за такими причинами:

* Зображення мають велику розмірність, внаслідок чого збільшується розмір нейронної мережі, час і обчислювальна складність процесу навчання.
* Ігнорується топологія вхідних даних. Компоненти вхідного шару можуть бути представлені в будь-якому порядку. Однак зображення мають строгу двовимірну структуру, в якій існує залежність між просторово сусідніми пікселями.
* Рукописні підписи кожної людини можуть бути написані з спотвореннями, а класичні нейронні мережі досить чутливі до спотворень вхідного сигналу.

Тому в роботі [4] сформовано висновок про доцільність розпізнавання рукописних підписів за допомогою згорткової нейронної мережі з використанням в якості функції активації гіперболічного тангенсу та методу навчання Левенберга-Марквардта, що забезпечує точність розпізнавання на рівні 98 %.

В роботі [7] обґрунтовано нейромережевих підхід до верифікації рукописного підпису. Зазначається, що існує «два підходи до верифікації підпису, що розрізняються за способом отримання даних: статичний (офлайн) метод, що полягає в обробці та аналізі зображення підпису; динамічний (онлайн) метод, заснований на зчитуванні підпису в режимі реального часу і виділення таких характеристик, як швидкість написання, кут нахилу та сила натиску на перо» [7]. Онлайн підхід є більш точним, тому що використовуються і динамічні характеристики підписи, але офлайн метод також широко поширений, наприклад, в областях, де людина не присутня фізично під час процесу верифікації або ж коли немає підключення до мережі або графічного планшета (перевірка банківського чека може бути виконана лише офлайн). Зазначається перспективність застосування другого підходу, як більш універсального і доступного в застосуванні.

Сутність обох підходів однакова. вона включає в себе збір даних, їх попередню обробку, визначення ознак, прийняття рішення і оцінку його ефективності. Також зазначається, що при розробці методів верифікації рукописних підписів основна увага концентрується на мінімізації помилок при розпізнаванні зразка рукописного підпису. Як і в загальному випадку при нейромережеві верифікації підпису дані помилки поділяють на два види:

- 1-го роду - помилкова відмова в прийнятті підпису, коли система відмовляє в доступі зареєстрованій особі (далі - FRR, False Rejection Rate);

- 2-го роду - помилкове прийняття підпису (далі - FAR, False Acceptance Rate – система сприймає підроблений підпис в якості справжнього).

Обидва типи помилок пов'язані між собою і залежать від прийнятого граничного значення для визначення справжності підпису. Таким чином, при збільшенні його значення, FRR зменшується, а FAR збільшується, і навпаки.

Для реалізації статичного методу верифікації використовують різні підходи: наприклад, штучні нейронні мережі, сховані марковские моделі, різні види метрик, метод опорних векторів, обчислення локальних екстремумів або обчислення матриці відстані [14]. В даний час перевагу все частіше віддається нейромережевому підходу по ряду причин [2]. По перше, за останні півстоліття розвиток нейромережевих технологій зробило великий стрибок. По-друге, застосування даного все більш розвивається підходу допомагає верифікувати підпис точніше, ніж при використанні інших. Це обумовлюється тим, що нейронні мережі ефективно будують нелінійні залежності, які точніше описують дані, вони більш стійкі до шумів у вхідних даних і більш адаптовані до їх змін. Як видно з наведеного в табл. 1.5 порівняння, застосування штучних нейронних мереж (5, 6 рядки табл. 1.5) дозволяє досягти меншого відсотка помилок першого і другого роду, ніж при використанні інших підходів.

*Таблиця 1.5*

Порівняння FRR та FAR для різних підходів

|  |  |  |
| --- | --- | --- |
| **Підхід до верифікації** | **FRR** | **FRR** |
| Сховані маркіські процеси | 0,2 % | 32,6 % |
| Машина опорних векторів | 20,06 % | 18,53 % |
| Евклідова метрика | 21,7 % | 19,2 % |
| Вейвлет-перетворення | 16,2 % | 16,1 % |
| Мережа радіальних базисних функцій | 7 % | 5 % |
| Багатошаровий персептрон | 2,9 % | 7,4 % |

Також в роботі [24] наведено типовий алгоритм нейромережевої офлайн верифікації рукописного підпису. Наведений алгоритм передбачає необхідність поділу процесу нейромережевої верифікації підпису на дві стадії: навчання та тестування нейромережевої моделі. Навчання нейромережевої моделі включає в себе вилучення зображення підпису з навчальної вибірки, первинну обробку зображення, яка полягає в конвертації зображення в бінарне, обрізки, нормалізації розміру, скелетизації, визначення інформативних ознак, навчання мережі на оброблених зображеннях. Тестування нейромережевої моделі включає в себе такі кроки: визначення підпису, справжність якого необхідно перевірити; по аналогії із стадією навчання обробку зображення підпису, визначення множини ознак, подачу на вхід навченої нейронної мережі множини ознак, отримання результату. Відзначається значний вплив на ефективність розпізнавання вибір типу нейронної мережі. Стверджується, що в даній області в основному використовуються багатошарові штучні нейронні мережі з прямим поширенням сигналу, що навчаються за допомогою алгоритму зворотньго поширення помилки, а також мережі радіальних базисних функцій [9]. Також в роботі [24] наведено опис та результати експериментальних досліджень, пов’язаних з розробкою нейромережевої системи розпізнавання підписів. В якості прототипу нейронної мережі був обраний багатошаровий персептрон, структура якого показана на рис. 1.8. У схованих шарах в якості функції активації використано гіперболічний тангенс. У вихідному шарі використано порогову функцію активації. Таким чином, значення на виході персептрону відповідає 0 в разі, якщо підпис визнається підробленим, і дорівнює 1, якщо підпис визнається справжньої.

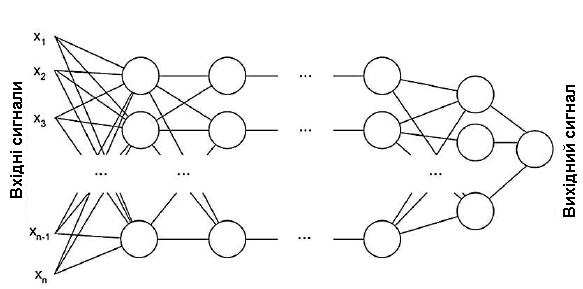


Рис. 1.1 Структура багатошарового персептрону для розпізнавання підпису

Для реалізації системи розпізнавання було використано мову програмування Python та бібліотеки Skimage, OpenCV - для обробки зображень, NumPy і SciPy - для розрахунків, Keras і TensorFlow -для реалізації нейронної мережі. Для формування навчальної та тестової вибірки виокристана база даних підписів для ICFHR (International Conference on Frontiers in Handwriting Recognition) 2010 Signature Verification Competition. Тестова вибірка складалась із 19 справжніх і 15 підроблених підписів. До складу навчальної вибірки включено 60 справжніх і 70 підроблених підписів. Результати експериментів наступні: FRR=3.3 %, а FAR=5 %. Наведені показники на думку авторів [24] свідчить про перспективність застосування нейромережевих засобів розпізнавання підписів користувачів.

## 1.4 Висновки до розділу

В результаті проведеного дослідження можна сформулювати такі вимоги до нейромережевої системи розпізнавання рукописних підписів користувачів інформаційних систем:

* слід застосувати адаптовану до поставленої задачі розпізнавання підписів згорткову нейрону мережу, в якій в якості вхідних параметрів можуть бути використані пікселі відсканованого чорно-білого зображення рукописного підпису або/та характерні точки рукописного підпису;
* програмний додаток повинен відповідати сучасним вимогам до надійності, швидкості та ресурсоємності;
* програмний додаток повинен бути орієнтований на використання в комп’ютерних системах, що функціонують під управлінням сучасних операційних систем.

Крім того, проведений аналіз задачі розробки комп’ютерної системи розпізнавання підписів користувачів дозволяє виділити такі етапи її вирішення: побудова архітектури системи розпізнавання підписів, розробка математичного забезпечення комп’ютерної системи, розробка інформаційного забезпечення, розробка програмного забезпечення та верифікація отриманих результатів.

**2. РОЗРОБКА НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ ТА СИСТЕМИ РОЗПІЗНАВАННЯ РУКОПИСНОГО ПІДПИСУ**

## 2.1 Проектування системи розпізнавання підписів користувачів комп’ютерних систем

Застосувавши результати проведеного аналізу, було побудовано показані на рис. 2.1–2.4 моделі даних системи розпізнавання рукописного пілпису. Для цього використано мову моделювання UML. Так, на рис. 2.1 показана діаграма пакетів.



Рис. 2.1. Діаграма пакетів

На вказаній діаграмі відображено пакети та підпакети, з яких складається система виявлення:

* «Сканер» – пакет, що відповідає за сканування та реєстрацію в системі розпізнавання підпису користувача комп’ютерної системи. До складу пакету «Сканер» входять підпакети «Сенсор» та «Реєстратор», що відповідають за введення в систему нового підпису та його збереження.
* «Класифікатор» – пакет, що відповідає за класифікацію підпису користувача комп’ютерної системи. До його складу входять підпакети «Фільтратор» та «Розпізнавач», що відповідають за фільтрацію відсканованого зображення підпису та його класифікацію.
* «Модератор» – пакет, що пов’язаний з управлінням системою. Цей пакет складається із двох підпакетів «Адміністратор» та «Сигналізатор». Підпакет «Адміністратор» призначений для можливості настроїти параметри вводу та фільтрації підпису користувача. Підпакет «Сигналізатор» забезпечує сигналізацію про результати розіпзавання.

На рис. 2.2 показана діаграма компонентів системи розпізнавання, що складається із трьох модулів «Detector», «Filter» та «ANM». Вказані модулі визначають інтерфейс отримання зображення підпису користувача комп’ютерної системи, його фільтрацію класифікацію за допомогою штучної нейронної мережі.

## 

Рис. 2.2. Діаграма компонентів

Діаграма прецедентів показана на рис. 2.3. При побудові цієї діаграми враховано, що до її складу входить 3 актори: «Детектор», «Класифікатор», «Модератор». Актори «Детектор» та «Класифікатор» призначені для вводу в систему зразка підпису користувача, його фільтрації, розпізнавання та сигналізації про результати розпізнавання. Актор «Модератор» є діючою особою, що взаємодіє з системою розпізнавання. Мета прецендентів використання відповідає їх назвам.



Рис. 2.3. Діаграма прецедентів

Діаграма класів, що використовується для відображення структур класів, які складають архітектуру системи, показана на рис. 2.4.

## 

Рис. 2.4. Діаграма класів

При побудові цієї діаграми враховано, що до її складу входить 4 класи: «Moderator», «Detector», «Filter» та «ANN». Класи пов’язані між собою відношенням залежності, тобто зміна специфікації одного елемента може вплинути на інший, який його використовує.

**2.2. Процедура визначення вхідних параметрів нейромережевої моделі**

Розглянемо базові положення в області визначення характерних точок підпису, що можуть бути використані в якості вхідних параметрів нейромережевої моделі. Хоча дослідження в галузі застосування ЕОМ до криміналістичної експертизи почерку проводяться давно, але розвиток комп'ютерних технологій зробив можливим проведення аналізу динаміки нарисних рухів, що виникають при виконанні рукописного підпису.

Спеціальні графічні планшети дозволяють відстежувати траєкторію руху пера, а також кут нахилу ручки та тиск, що чиниться на перо. В англомовній літературі цей підхід носить назву «on-line» методу розпізнавання, на відміну від аналізу відсканованих зображень або «off-line» методу розпізнавання. Саме можливість вимірювання динаміки робить привабливим використання рукописного підпису як біометричного параметра, зручного для застосування в широкому спектрі інформаційних систем. Зазначимо, що такий підхід відноситься до категорії функціональних методів розпізнавання і базується на процедурі еластичного зіставлення методом динамічного програмування. Під динамічною або «on-line» кривою розуміється параметрически задана крива Γ: {x(t), y(t), t∈[0, T]}. Інформація, що міститься в даній часовій послідовності, відображає динаміку м'язових рухів руки і тому може бути використана як біометрична характеристика людини. В даний час загальноприйнятою вважається така схема побудови систем доступу, в яких користувач вказує свій ідентифікатор, який підтверджується потім рукописним підписом. Даний підхід дозволяє користувачеві надавати системі тільки свой підпис, за яким програма знаходить еталон (проводить ідентифікацію), а потім перевіряє справжність пред'явленого підпису (здійснює верифікацію). Для вирішення завдання пошуку еталона пропонується описувати динамічну криву набором екстремальних точок. Динамічні криві можуть бути параметризовані різними способами. Оскільки переважна прикладних систем орієнтовані на вирішення задачі верифікації, то найбільш поширеною є часова параметризація. Однак крім часової параметризації, можлива параметризація за довжиною траєкторії:

. (2.1)

Це означає сталість швидкості переміщення уздовж траєкторії. Також допустима афінна параметризація за довжиною траєкторії, яка задовольняє умові виду:

. (2.2)

Це означає сталість площі паралелограма, побудованого на векторах  та , де параметр s знаходиться з умови:

. (2.3)

Зазначимо, що динамічні криві містять інформацію як про динаміку руху пера, так і про часову послідовність точок, що складають траєкторію. Оскільки в даному підході на етапі ідентифікації (пошуку еталона) цікавить тільки форма кривої, то далі під терміном динамічної кривої розуміється параметризована за довжиною крива

, (2.4)

де параметр  знаходиться з умови .

Відзначимо, що швидкість написання рукописних кривих, тобто іншими словами «динаміка» руху руки, тісно пов'язана з формою траєкторії. Дійсно, при виконанні заокруглень швидкість пера нижче, ніж при виконанні прямолінійних рухів. Також темп виконання елементів підпису варіюється від одного виконання до іншого, що іноді сильно ускладнює перевірку справжності. Виходячи з подібних міркувань, В. Налва в своїй роботі [4] відмовився від параметризації за часом, і розглядає тільки форму траєкторії, що призводить до відмови розгляду таких характеристик як, наприклад, тиск на перо або нахил пишучої ручки. Тому в силу специфічності постановки завдання і наведених доводів в даній роботі не розглядається часова параметризація, тим більше що, як показано в роботі [2], вона не має явних переваг перед іншими видами параметризації, а іноді і програє їм. Розглянемо зіставлення траєкторій на основі екстремальних точок. Розпізнавання динамічних кривих дуже близько до тематики розпізнавання злитної людської мови. Дані, одержані за графічного планшета, схожі на звуковий сигнал. Як і фонеми при розпізнаванні мовлення, об'єкти розпізнавання рукописного тексту (букви і символи) чітко задані. Форма рукописних знаків залежить від положення в слові, що також схоже з варіативністю фонем при проголошенні слів в залежності від їх оточення. Усе це дозволило адаптувати розроблені методи з області розпізнавання мови до розпізнавання рукописного тексту [5, 6]. Проте, пряме застосування цих методів до розпізнавання злитного тексту пов'язане з певними труднощами, пов'язаними із знаходженням діакритичних знаків [7]. У той же час слова в письмовій мові часто вдається досить легко відокремлювати одне від іншого, на відміну від звуків у злитній мові. Основними методами розпізнавання рукописного підпису є: статистичний підхід (заснований на апараті схованих марковських моделей), підхід з використанням нейронних мереж, а також зіставлення динамічних кривих методом динамічної трансформації часової шкали (dynamic time warping). Залежно від принципів зіставлення елементів траєкторій в літературі зустрічаються різні варіації останнього методу:

* Symmetric Dynamic Time Warping.
* Continuous Dynamic Time Warping.
* Piecewise Aggregate Approximation Dynamic Time Warping.
* Iterative Deepening Dynamic Time Warping.

У більшості робіт при порівнянні задіються всі точки траєкторії. Наприклад, в роботі Ф. Грісса [12] був застосований метод порівняння траєкторій, апробований раніше в роботах С. Коннелла при розпізнаванні рукописних символів [13], що проілюстровано на рис. 2.5. Проте, як було відзначено в роботі [14], метод DTW володіє двома основними недоліками: високою обчислювальної вартістю і приведенням підробленого підпису у відповідність з еталоном. Для усунення зазначених недоліків в роботі [14] був запропонований метод порівняння підписів на основі пошуку відповідності екстремальних точок (extreme points warping, EPW).

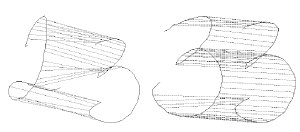


Рис. 2.5 Приклад зіставлення двох рукописних цифр методом dynamic time warping

Вихідний сигнал можна представити як низку «піків» і «впадин», показаних на рис. 2.6. Оскільки при такому підході відповідність необхідно знаходити тільки між точками відповідного типу, то процес порівняння істотно прискорюється. На основі зразків підписів 25 користувачів Ф.Хао і К.Чан зробили емпіричне узагальнення, що помилки такого зіставлення бувають трьох типів: через пропуск сегмента на початку або в кінці одного з підписів, а також через так звані «зубчики» - випадкову пару «пік-впадина» з невеликим перепадом по висоті.

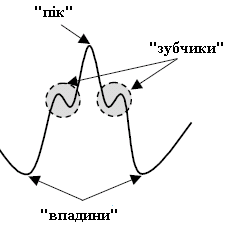


Рис. 2.6 Вид нестійких екстремальних точок

Для виключення з розгляду подібних точок вводиться евристичне правило: точка позначається як екстремальна, якщо виконані одночасно дві умови:

, (2.5)

де - перепад висот між послідовними «впадиною» і «піком»;

 - перепад висот між «піком» і подальшою «впадиною»;

 - порогове значення (рівне одному пікселю).

Отримані екстремальні точки зіставлялися методом динамічного програмування [13, 14]. На рис.2.7 показана схема прямого ходу, коли шлях з точки *(i, j)* може бути продовжений в ту з трьох точок *(i + 1, j + 1), (i + 1, j + 3),* *(i + 3, j + 1),* яка дає мінімальний штраф за невідповідність.

Прямий прохід дозволяє знайти міру відмінності двох кривих, при цьому послідовність точок *(i, j)* задає шлях, кожна точка якого позначає локально-оптимальну відповідність між *i*-тою точкою еталона (точки еталона пронумеровані по осі абсцис) та *j*-тою точкою зразка (точки зразка пронумеровані по осі ординат).



Рис. 2.7 Реалізація extreme points warping, EPW

Правила можливого продовження шляху випливають з факту чергування екстремумів, при цьому вибирається таке продовження, яке дає мінімальний штраф за невідповідність. Після закінчення розрахунку в правому верхньому куті таблиці буде міститися міра відмінності двох кривих. У разі необхідності зворотний прохід по вже встановленому шляху дозволяє знайти оптимальну відповідність двох кривих. Аналогічний спосіб зіставлення кривих був раніше запропонований в роботі [15]. Однак неможливість врахувати деякі варіативні особливості почерків, а також нагромадження помилок в ході послідовного зіставлення хорд, утворених відповідними екстремальними точками, вимагало переглянути процедуру відбору і зіставлення екстремальних точок.

Розглянемо класичні підходи до виділення екстремальних точок рукописного підпису. В тій чи іншій мірі багато підходів до верифікації рукописного підпису ґрунтуються на процедурі знаходження характерних або екстремальних точок. В одній з перших робіт [16] точки вибиралися вручну, після чого їх координати вводилися в ЕОМ для обробки. Застосування сенсорних панелей і графічних планшетів дозволяє автоматизувати дану процедуру. Але які точки кривої є найбільш інформативними? Як правило, вибір падає на точки із значенням кривизни більше деякого порогового значення. Обчислення кривизни по формулі диференціальної геометрії вимагає існування другої похідної за часом, що накладає обмеження на гладкість досліджуваної траєкторії. Тому на практиці кривизну визначають за швидкістю пера (для цього потрібно тільки перша похідна) як корелятивної величини, при цьому в якості порогового значення вибирається значення, яке становить 15-18% від середньої швидкості написання, що проілюстровано графіками показаними на рис.2.7. Або замінюють межу, визначену за допомогою виразу (2.6), на кінцеву різницю

, (2.6)

де - зміна кута нахилу дотичної на протязі ділянки траєкторії від  до .



Рис. 2.7 Залежності кривизни траєкторії і значення модуля швидкості від часу

Кривизна траєкторії в *i*-тій точці сімейства точок *Pi*−2, *Pi*−1, *Pi* , *Pi*+1, *Pi*+2 визначається за кутом між векторами та . Як варіант, допустимо замінити функцію кривизни її дискретним аналогом у вигляді часової послідовності зміни кутів. Але якщо в разі встановлення справжності підпису досить порівняти дві часові послідовності точок, то в разі розпізнавання підпису на основі форми та структури кривих стає необхідним проведення ретельного відбору найбільш інформативних точок.

Для подальшого опису будемо використовувати наступні визначення екстремальних точок.

Визначення 1. Вертикальним екстремумом називається точка , для якої .

Визначення 2. Горизонтальним екстремумом називається точка , для якої .

При цьому кожен вертикальний екстремум визначає локальний максимум або мінімум траєкторії. Під екстремальними точками по кривизні будемо розуміти точки, локальна кривизна траєкторії в яких більше порогового значення. На основі даних [26] можливо стверджувати, що вище описані методи знаходження екстремальних точок не завжди оптимальні, а стосовно даного підходу і малоефективні. Дійсно, дуже часто вертикальні і горизонтальні екстремуми знаходяться поблизу від екстремальних точок по кривизні, що робить необхідним залучення процедури фільтрації. В той же час, прийняття неправильного порогового значення для даного конкретного підпису може привести до пропуску важливих точок, що не мають великої кривизни. На рис. 2.8 наводиться приклад написання двох букв «Я», що відрізняються формою траєкторії. Через пропуск точки, що характеризує опуклість на спадаючій ділянці, перша буква може бути помилково класифікована як буква «е». Розроблений спосіб знаходження екстремальних

точок дозволяє відстежувати зміни траєкторії зазначеного типу, що особливо важливо при розпізнаванні рукописних букв і слів, наприклад, при графічному введенні тексту на кишенькових комп'ютерах.



Рис. 2.8 Приклади написання букв «Я»

На рис. 2.9 показане «скелетне» уявлення тих самих букв, отримане шляхом з'єднання екстремальних точок. З рисунку видно, що вся необхідна інформація про структуру букв була збережена і передана набагато меншою кількістю точок. Як було встановлено в роботі [15] зіставлення всіх точок сегментів по евклідовій мірі з подальшим підсумовування всіх локальних штрафів може призводити до помилок розпізнавання, коли рукописне слово «Лена» зіставляється зі словом «Лера», тому що штраф за пропуск зайвого елемента виявляється менше підсумкового штрафу за невідповідність двох варіантів слова «Лена».



Рис. 2.9 Приклади знаходження екстремальних точок

На рис. 2.10 приведена послідовність кроків, які виконуються для отримання набору екстремальних точок. Перші дві процедури відносяться до етапу перед обробки, коли вихідна траєкторія згладжується і проводиться повторна вибірка точок методом інтерполяції для отримання еквідистантного представлення кривої. Оскільки внаслідок шумів в оцифрування, а також через тремтіння пера точки розташовуються нерівно уздовж траєкторії, то бажано робити згладжування кривої. Для цієї мети був використаний досить метод згладжування за координатами сусідніх точок. У порівнянні з більш поширеним методом згладжування «по Гаусу» він має меншу обчислювальну вартістю при практично тих же результатах. Усереднення проводилося за двома сусідніми точками, окремо для кожної з координат, за формулами:

. (2.7)

Для позначення попередньо знайдених екстремальних точок далі по тексту будемо використовувати термін «критичні точки», тому що потім частина з цих точок буде відфільтрована. Схема процедури пошуку критичних точок показана на рис.2.10.



Рис. 2.10 Схема процедури пошуку критичних точок

Зазначимо, що на відміну від загальноприйнятого алгоритму пошуку в даному випадку з розгляду не виключались критичні точки з малим перепадом по висоті. Причиною цього є те, що послідовність таких точок може утворювати так зване «плато» показане на рис. 2.11.

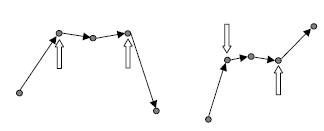


Рис. 2.11 Горизонтальне «плато»

Аналогічна ситуація можлива при знаходженні горизонтальних критичних точок. Тому на кроці знаходження критичних точок виділялися наступні типи точок: кінцеві точки сегментів кривої, вертикальні і горизонтальні екстремуми, граничні точки вертикального плато, а також точки перегину. При цьому для знаходження вертикальних екстремумів застосовувалося евристичне правило: точка P*і* є екстремальною, якщо виконанується умова:

. (2.8)

Якщо для деякої *i-*тої точки наведений вираз дорівнює нулю, а при цьому ординати трьох послідовних точок не збігаються, то така точка додавалася в набір як гранична точка плато. Аналогічно знаходилися і горизонтальні екстремуми (критичні точки).

Оскільки особлива увага приділялася швидкості та ефективності алгоритму, точки перегину знаходилися не з рівності нулю другої похідної, а шляхом аналізу знака векторного множення

. (2.9)

В нашому випадку вираз (2.9) можна замінити на спрощений координатний запис для плоского випадку:

. (2.10)

Таким чином, після виконання даного кроку можна отримати набір критичних точок, який є надлишковим. Залежно від параметрів програми кількість отриманих точок в 2-2,5 рази перевищує кількість остаточно знайдених екстремальних точок. Тому наступним кроком є видалення зайвих точок, а також додавання нових, що більш точно характеризують форму траєкторії. Зокрема, всі точки перегину підлягають видаленню, тому що розташовуються, як правило, в середині «прямих» ділянок, що знижує їх інформаційну значимість. Замість цього в набір додаються екстремальні точки, що умовно звуться екстремумами по кривизні, хоча саме значення кривизни в цих точках може бути і не дуже велике.

**2.3.** **Розро****бка базової нейромережевої моделі**

Найбiльш природний i давно використовуваний метод перевiрки особистостi - це перевiрка рукописного пiдпису. Метод верифiкацiї рукописного пiдпису бiльш поширений i менш нав'язливий, нiж iншi методи бiометричної аутентифiкацiї. Той факт, що рукописний пiдпис широко використовується як засiб особистої перевiрки, пiдкреслює необхiднiсть автоматичної системи перевiрки. В якості вхідних параметрів такої моделі можливо використовувати глобальні та/або локальні ознаки рукописного підпису. Також можливо використовувати параметри окремих пікселів зображення рукописного підпису.

Зазначимо, що до глобальних ознак підпису відносять: центр мас, висота підпису в пікселях після нормалізації, відношення висоти до ширини, область зображення (це кількість чорних пікселів на зображенні), вертикальний розподіл, горизонтальний поділ, максимум вертикальної проекції, максимум горизонтальної проекції, кількість крайових точок, кількість точок перетину. Локальні ознаки підпису можливо визначити шляхом поділу зображення на окремічастини і обчислення щільності пікселів для кожної з них. Як правило при обробці зображення на основі глобальних і локальних ознак вхідний вектор ознак зображення складається більше ніж з 150 компонент.

Автономнi системи iдентифiкацiї працюють з вiдсканованим зображенням пiдпису, а значить його можна розпiзнати, використовуючи комп'ютерний зiр. Нейроннi мережi є фундаментальною частиною комп'ютеризованих задач розпiзнавання образiв вже понад пiвстолiття, i продовжують використовуватися в дуже широкому дiапазонi проблемних областей [16]. Пiдхiд до вирiшення завдання за допомогою нейронних мереж пропонує кiлька переваг, таких як унiфiкованi пiдходи до вилучення i класифiкацiї ознак i гнучкi процедури пошуку ефективних, помiрно нелiнiйних рiшень. Коли даний пiдхiд використовується в динамiчнiй або автономної перевiрки пiдпису, вiн також показує розумну продуктивнiсть.

Розпiзнавання пiдписiв можна розглядати як задачу класифiкацiї зображень [17]. Класифiкацiя в такому випадку є введення вхiдного зображення i виведення класу або ймовiрностi класiв, якi найкраще описують зображення. Найбiльш вiдповiдною основою для розробки автоматичної системи класифiкацiї зображень є згорнута нейронна мережа [18].

Безпосередньо формуючи постановку задачi розробити алгоритм автоматичної перевiрки рукописного пiдпису за допомогою згорнутої нейронної мережi, виходячи з даних, отриманих вiд осiб, чиї пiдписи повиннi бути автентифiковано системою, варто зазначити, що алгоритм повинен по набору вхiдних даних визначити, чи є власноручний пiдпис справжньої або пiдробленої, коли фiзична особа вимагає пiдтвердження особи. Пiдхiд до автоматичної перевiрки пiдпису повинен здiйснюватися на основi iнтелектуальних алгоритмiв, а саме штучних нейронних мереж.

Вiдповiдно до сучасної методологiї застосування нейронних мереж в областi захисту iнформацiї побудова ефективної нейромережевої моделi оцiнки iнформативностi образу пiдпису роздiляється на два основнi блоки:

1. Визначення найбiльш ефективного виду моделi.

2. Адаптацiя параметрiв моделi ефективного виду до умов поставленого завдання.

Результати [20] свiдчать про те, що завдання визначення найбiльш ефективного виду НММ можливо вирiшити на пiдставi пiдходу - найбiльш ефективним є той вид НММ, характеристики якого бiльш повно вiдповiдають значущим умовами завдання визначення iнформативностi бiометричних образiв.

Також встановлено, що безлiч значимих умов дiлиться на категорiї, що характеризують навчальнi данi, обмеження процесу навчання, обчислювальнi потужностi, вихiдну iнформацiю, технiчну реалiзацiю i сферу застосування НММ.

Показано, що вiдповiднiсть виду НММ k-му умовi можна визначити за допомогою k-го параметра ефективностi - цей параметр дорiвнює -1:

* якщо умова не виконується, дорiвнює 1;
* якщо умова виконується i дорiвнює 0;
* якщо умова виконується частково.

Значимiсть k-го умови пропонується враховувати за допомогою вагового коефiцiєнта. Математичної iнтерпретацiєю такого пiдходу є вираз:

 (2.11)

де  – iнтегральний критерiй ефективностi i-го виду НММ;

*аi* – *i*-й вид НММ;

*А* – безлiч допустимих видiв НММ;

*I* – кiлькiсть допустимих видiв НММ;

У свою чергу розрахунок iнтегрального критерiю ефективностi для i-го виду НММ можливо реалiзувати так:

 (2.12)

 – значення k-го параметра ефективностi i-го виду НММ;

*К* – кiлькiсть параметрiв ефективностi;

*rk* – ваговий коефiцiєнт k-го параметра ефективностi.

Згiдно з результатами, отриманими в [11], в сучасних апробованих системах бiометричної аутентифiкацiї застосовуються НММ з прямим розповсюдженням сигналу, архiтектура яких подiбна до класичного БШП. На пiдставi даних [22] I власного практичного досвiду в першому наближеннi компоненти А можливо визначити так:

 (2.13)

де БШП - багатошаровий персептрон,

ДШП - двошаровий персептрон,

ГНМА - глибока нейронна мережа з предобученiем,

ГНМР - глибока нейронна мережа в прихованих нейронах якої використовується функцiя активацiї типу ReLU,

ЗНМ – згорткова нейронна мережа.

Вiдзначимо, що функцiя ReLU (випрямлена лiнiйна функцiя) визначається так:

 (2.14)

де  – вихiдний сигнал прихованого нейрона;

– сумарний вхiдний сигнал прихованого нейрона, який визначається виразом (2.15);

– функцiя визначення максимуму вiд переданих їй аргументiв.

 (2.15)

де – кiлькiсть вхiдних сигналiв нейрона прихованого шару,

– -iй вхiдний сигнал (),

– ваговий коефiцiєнт -го вхiдного сигналу.

Також в базовому випадку прийнято, що величини вагових коефiцiєнтiв для всiх параметрiв однаковi i рiвнi 1.

Це дозволило за допомогою виразу (2.12) розрахувати значення iнтегрального критерiю ефективностi для всiх видiв НММ, якi входять до складу безлiчi *А*. Отриманi величини представленi в табл. 2.1.

*Таблиця 2.1*

Величини iнтегрального критерiю ефективностi для апробованих видiв нейромережевих моделей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Величина iнтегрального критерiю** | **Вид нейромережової моделi** | | | | |
| **ДШП** | **ЗНМ** | **ГНМA** | **ГНМР** | **БШП** |
|  | 1 | 13 | 10 | 6 | 2 |

Пiдставивши данi табл. 2.1 в вираз (2.10) визначено, що найбiльш ефективним видом НММ є ЗНМ.

Типова структура згорткової нейронної мережi, призначеної для розпiзнавання зображень, показана на рис. 2.12. Вхiднi параметри такої моделi вiдповiдають окремим пiкселям. Тому кiлькiсть вхiдних параметрiв дорiвнює розмiру зображення.

Кiлькiсть вихiдних нейронiв дорiвнює кiлькостi розпiзнаваних образiв, а кiлькiсть прихованих нейронiв пiдбирається експериментальним шляхом з позицiй максимальної точностi розпiзнавання.

Дана мережа складається з двох повторюваних дiлянок, якi мiстять по два шари згорнутi i по одному шару пiдвибiрки. Данi дiлянки призначенi для видiлення основних ознак рукописного пiдпису. В кiнцi мережi розташований класифiкатор, який складається з одного повно зв'язаного шару, який мiстить 512 нейронiв, i вихiдного шару, який мiстить 10 нейронiв.

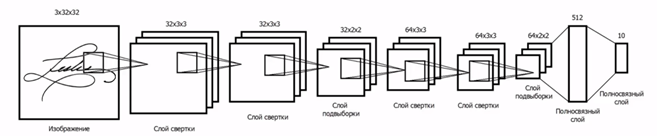


Рис. 2.12 Архiтектура згорткової нейронної мережi

На вхiд мережi надходять зображення розмiром 32 на 32 пiкселiв в трьох каналах: червоний, зелений i синiй. На першому шарi згортки використовується 32 карти ознак розмiру 3 на 3. Тобто кожен нейрон згорнутого шару пiдключений до квадратному дiлянцi зображення розмiром 3 на 3.

Наступний згорнутий шар має таку ж архiтектуру: 32 карти ознак c ядром згорнутi 3 на 3. Пiсля цього йде шар пiдвибiрки, на якому виконується зменшення розмiрностi зображення.

Змiна розмiрностi виконується для кожної карти ознак окремо, тому тут використовується також 32 карти ознак.

Розмiр поля пiдвибiрки 2 на 2. Пiсля шару пiдвибiрки починається новий каскад згорнутих шарiв.

На третьому i на четвертому шарi згортки використовується 64 карти ознак розмiром 3 на 3, а на другому шарi пiдвибiрки, яка йде пiсля цих згорнутих шарiв, також вiдбувається зменшення розмiрностi в квадратi 2 на 2. Пiсля цього данi перетворюються з двовимiрного формату в одновимiрний i передаються на повно зв'язаний шар, на якому вже виконується класифiкацiя.

Сумарний вхiдний сигнал деякого нейрона першого згорнутого шару розраховується так:

 (2.16)

де  - вхiдний сигнал (*i,j*)-го нейрона першої карти ознак,

 - змiщення нейронiв першої карти ознак,

*S* - розмiр рецептивної областi нейрона (розмiр ядра згортки),

 - ваговий коефiцiєнт (*s,t*)-го синаптичного зв'язку нейрона першої карти ознак,

*x* - вихiд нейрона попереднього шару.

Сумарний вхiдний сигнал довiльного нейрона наступних шарiв згортки розраховується за допомгою наступного виразу:

 (2.17)

де  - вхiдний сигнал (*i,j*)-го нейрона *k*-ой карти признакiв,

 - змiщення нейронiв *k*-ой карти признакiв,

 - ваговий коефiцiєнт (*s,t*)-ой синаптичного зв'язку нейрона k-ой карти ознак,

*x* - вихiд нейрона попереднього шару.

*f* – функцiя активацiї.

Шари субдiскретiзацiї, що слідують пiсля шарів згортки, зменшують розмiрнiсть карти ознак та забезпечують iнварiантнiсть до масштабу. Вони мають один настроюється параметр - коефiцiєнт Субдiскретiзацiя, що визначає, у скiльки разiв буде зменшено зображення, вiдповiдне попереднiй картi ознак. Вихiдний сигнал нейрона шару Субдiскретiзацiя визначається виразом:

 . (2.18)

Сумарний вхiдний сигнал нейронiв повнозв'язну шару розраховується за допомогою формул (2.14), а вихiдний сигнал в залежностi вiд використовуваної функцiї активацiї.

В нейронах вихiдного шару ЗНМ, як правило, використовується функцiя активацiї типу SOFTMAX, що дозволяє iнтерпретувати вихiдний сигнал мережi у виглядi ймовiрностi. У цьому випадку вихiдний сигнал нейронної розраховується так:

 (2.19)

де – кiлькiсть нейронiв в повнозв'язну шарi,

– сумарний вхiдний сигнал вхiдний сигнал нейронiв вихiдного шару, який визначається за допомогою виразу.

Також варто вiдзначити, що пiсля всiх операцiй згортки i субдiскретiзацiї матриця, що описує вихiдне зображення, вироджується в вектор, вiдповiдний нейронам останнього прихованого шару.

Постановка завдання навчання i тестування згорткової нейронної мережi зводиться до того, що у згорткових мережах використовується пiдхiд навчання з учителем [19]. Тобто необхiдно мати розмiчений набiр даних.

## 2.4 Адаптація нейромережевої моделі до розпізнавання підписів користувачів

У загальному випадку методологiя адаптацiї НММ до умов поставленого завдання класифiкацiї передбачає синтез алгоритмiв настройки безлiчi керованих параметрiв. Структура цiєї методологiї представлена на рис. 2.13.

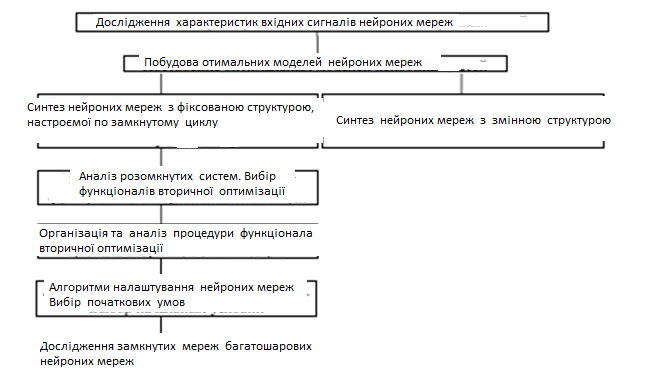


Рис. 2.13 Структура адаптацiї нейромережевої моделi до умов завдання

Специфiчнi вимоги прикладних задач визначали деякi особливостi структур нейромережевої моделi, є об'єктом управлiння з допомогою специфiчних алгоритмiв настройки, представлених в [13]:

- Континуум числа класiв, коли вказiвка вчителя системи формується в виглядi не перериваного значення функцiї в деякому дiапазонi змiни;

- Континуум рiшень багатошарової нейронної мережi, що формується вибором континуальної функцiї активацiї нейрона останнього шару;

- Континуум числа ознак, що формується переходом в просторi ознак вiд уявлення вихiдного сигналу у виглядi N-мiрного вектора дiйсних чисел до речової функцiї в деякому дiапазонi змiни аргументу. Континуум числа ознак, як наслiдок, вимагає специфiчної програмної i апаратної реалiзацiї нейронної мережi.

- Континуум числа нейронiв в шарi. Реалiзацiя багатошарових нейронних мереж з континуумом класiв i рiшень проводиться вибором вiдповiдних видiв функцiй активацiї нейронiв останнього шару. Варiант континууму ознак вхiдного простору був реалiзований в задачi розпiзнавання перiодичних сигналiв без перетворення їх з допомогою аналого-цифрового перетворювача на входi системи, i реалiзацiєю аналого-цифровий багатошарової нейронної мережi [25].

При цьому основою зазначеної методологiї є особливостi НММ як логiчного базису алгоритмiв розв'язання складних задач. До таких особливостей вiдносяться:

- iнварiантнiсть методiв синтезу нейронних мереж вiд розмiрностi простору ознак;

- можливiсть вибору структури нейронних мереж в значному дiапазонi параметрiв в залежностi вiд складностi та специфiки розв'язуваної задачi з метою досягнення необхiдної якостi рiшення;

- адекватнiсть поточним i перспективним технологiям мiкроелектронiки;

- вiдмовостiйкiсть в сенсi його неточного, але не катастрофiчного змiни якостi виконання завдання в залежностi вiд числа що вийшли з ладу елементiв.

Вiдзначимо, що останнi двi особливостi НММ актуальнi тiльки в разi апаратної реалiзацiї нейромережевої системи розпiзнавання.

Iмовiрнiсна модель свiту, взята за основу при побудовi алгоритмiв адаптацiї в багатошарових нейронних мережах, дозволила формувати критерiй первинної оптимiзацiї в розглянутих системах у виглядi вимог мiнiмуму середньої функцiї ризику i його модифiкацiй [5]:

- максимум апостерiорної ймовiрностi;

- мiнiмум середньої функцiї ризику;

- мiнiмум середньої функцiї ризику за умови рiвностi умовних функцiй ризику для рiзних класiв;

- мiнiмум середньої функцiї ризику за умови заданого значення умовної функцiї ризику для одного з класiв;

- iншi критерiї первинної оптимiзацiї, що випливають з вимог конкретної практичної задачi.

Базуючись на показаних типових структурах ЗНМ, з урахуванням даних теоретичних робiт [12, 15], визначено, що основними структурними параметрами даного типу НММ є:

- Кiлькiсть вхiдних нейронiв -.

- Кiлькiсть вихiдних нейронiв -.

- Кiлькiсть нейронiв в повнозв'язну шарi -.

- Кiлькiсть згортальних шарiв -.

- Кiлькiсть карт ознак в кожному згортковому шарi - .

- Кiлькiсть шарiв пiдвибiрки (субдiскретiзацiї) - .

- Коефiцiєнт масштабування для кожного шару пiдвибiрки (субдiскретiзацiї) - .

- Розмiр ядра згортки для кожного k-го згорткового шару .

- Зсув рецептивного поля при виконаннi кожної k-ой процедури згортки . Принято, що.

- Розмiр карти ознак для кожного k-го згорткового шару - . При цьому:

 (2.20)

Структура зв'язкiв мiж сусiднiми шарами згортки. Цю структуру можна представити у виглядi матрицi:

 (2.21)

де - матриця, компоненти якої визначають наявнiсть зв'язкiв мiж i-им i (i + 1) -им схованими шарами,

- компонент, який вказує на наявнiсть / вiдсутнiсть зв'язку мiж g-ой картою i-го шару i j-ой картою (i + 1) -го шару,

G - кiлькiсть карт в i-му шарi,

J - кiлькiсть карт в (i + 1) -му шарi.

При цьому , якщо зв'язок мiж g-ой картою i-го шару i j-ой картою (i + 1) -го шару є, , якщо зв'язку мiж g-ой картою i-го шару i j-ой картою (i + 1) -го шару немає.

Вiдповiдно до загальноприйнятої концепцiєю iєрархiчного розпiзнавання образiв згорнутої нейронною мережею багаторазове використання згортальних шарiв вiдповiдає iєрархiчним розпiзнаванню значущих ознак, а використання шарiв пiдвибiрки адаптує процес розпiзнавання до можливої змiни масштабу зазначених ознак.

З урахуванням необхiдностi мiнiмiзацiї помилки розпiзнавання модель оптимiзацiї структурних параметрiв СНС можна записати за допомогою виразу:

 (2.22)

де - помилка розпiзнавання,

 – вектор, що складається з матриць , якi визначають зв'язки мiж сусiднiми прихованими шарами нейронiв.

Вiдсутнiсть у формулi (2.12) iнших структурних параметрiв згорнутої нейронної мережi пояснюється тим, що вони є похiдними вiд використовуваних компонентiв цього виразу. Iнтеграцiя загальнозастосованої методологiї побудови нейромережевих систем розпiзнавання пiдпису [12], сформованих принципiв оптимiзацiї з певними обмеженнями на процес введення i попередньої обробки зображень в системi бiометричної аутентифiкацiї дозволила запропонувати метод оптимiзацiї параметрiв ЗНМ. Реалiзацiя запропонованого методу передбачає виконання 7 етапiв. Укрупнення блок-схема алгоритму розробленого методу представлена на рис. 2.14.

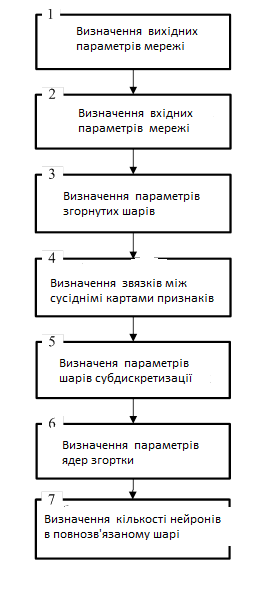


Рис. 2.14 Блок-схема алгоритму адаптацiї параметрiв

згорткової нейронної мережi

Етап 1. Визначити безлiч вихiдних параметрiв мережi. Для цього необхiдно встановити вiдповiднiсть мiж безлiччю розпiзнаються користувачiв i вихiдними нейронами мережi. Таким чином, кiлькiсть вихiдних нейронiв вiдповiдає потужностi безлiчi розпiзнаються користувачiв.

Етап 2. Визначити безлiч вхiдних параметрiв мережi. Для цього слiд розрахувати розмiри зображень, що подаються для розпiзнавання, а також встановити вiдповiднiсть мiж окремими елементами зображень i вхiдними нейронами. Також необхiдно врахувати розмiр зображень, вiдповiдних бiометричних образiв користувачiв в доступних базах даних, якi застосовуються для навчання мережi. Якщо розмiр зображень доступних навчальних вибiрок вiдрiзняється вiд розпiзнаються, то їх слiд масштабувати.

Етап 3. Базуючись на першому i другому принципах, використовуючи геометричнi характеристики еталонiв розпiзнаються зображень, методом експертного оцiнювання визначити кiлькiсть згортальних шарiв i кiлькiсть карт ознак в кожному згорнутої шарi.

Етап 4. Базуючись на третьому принципi методом експертного оцiнювання визначити структуру зв'язкiв мiж сусiднiми картами ознак.

Етап 5. Виходячи з вимог точностi i ресурсоємностi розпiзнавання методом експертного оцiнювання визначити наявнiсть i параметри шарiв Субдiскретiзацiя.

Етап 6. Базуючись на четвертому принципi, з урахуванням необхiдностi згортки зображення до вектору ознак визначити розмiр ядра i кроку перемiщення згортки для кожного рiвня розпiзнавання. Також на цьому етапi розраховуються розмiри карт ознак для кожного шару згортки.

Етап 7. Базуючись на методi визначення кiлькостi схованих нейронiв в БШП [26], визначити кiлькiсть нейронiв в повнозв'язному шарi.

У загальному випадку з урахуванням (3.21) перетворення iнформацiї, що реалiзовується запропонованим методом, можна припустити за допомогою таких висловiв:

, (2.23)

, (2.24)

. (2.25)

де - множина зображень, що характеризують легiтимних користувачiв;

 - кiлькiсть розпiзнаваних користувачiв;

- множина типiв бiометричних образiв, якi характеризуються геометричними параметрами;

 - множина доступних обчислювальних ресурсiв;

 - допустима точність розпізнавання;

 - бiометричний образ, вiдповiдний вiдповiдний рукописному почерку;

*t* – тип розпiзнається бiометричного способу ();

- множина доступних баз даних, якi можливо використовувати для формування навчальної вибiрки;

 - доступна база даних, яку можливо використовувати для формування навчальної вибiрки для t-го типу образiв пiдпису;

 - кортеж параметрiв ЗНМ.

## 2.5 Висновки до розділу

Другий розділ присвячено вирішенню науково-прикладної задачі розробки нейромережевої моделі розпізнавання рукописних підписів користувачів інформаційних систем. В результаті проведеного дослідження:

- Отримала подальший розвиток архітектура нейромережевої системи розпізнавання підписів користувачів, що за рахунок адаптації структурних параметрів до умов задачі розпізнавання підписів забезпечує можливість ефективного застосування сучасних типів нейронних мереж.

- Отримала подальший розвиток процедура визначення вхідних параметрів нейромережевої моделі, що за рахунок визначення множини інформативних параметрів рукописного підпису забезпечує можливість достатньої точності розпізнавання при мінімізації ресурсоємності нейромережевої моделі.

- Отримала подальший розвиток нейромережева модель розпізнавання підписів, яка за рахунок теоретично обгрунтованого вибору параметрів нейронної мережі, забезпечує можливість ефективного розпізнавання.

**3. РОЗРО****БКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ СИСТЕМИ**

## 3.1 Опис інструментарію

Для реалізації згорткової нейронної мережі використано мову програмування Python. Python - динамічна, строго типізована, об'єктно-орієнтована багаторівнева мова програмування, призначена для швидкого (вивчення, використання і розуміння) і забезпечення чіткого і однакового синтаксису. Python строго типізований. Це означає, що якщо a = "5" (рядок, значення якої дорівнює "5"), залишиться рядком і ніколи не буде примушувати до числа, якщо це вимагає контекст. Кожне перетворення типів в python має виконуватися явно. Це відрізняється від, наприклад, Perl або Javascript, де у вас слабкий введення тексту, і можете писати такі речі, як "hello" + 5, щоб отримати "hello5". Python орієнтований на об'єкти, з успадкуванням на основі класів. Все це об'єкт (включаючи класи, функції, модулі і т.д.) В тому сенсі, що їх можна передавати як аргументи, мати методи і атрибути і т.д. Python багатоцільовий: він не спеціалізується на конкретної мети користувачів (наприклад, R для статистики або PHP для веб-програмування). Він розширено через модулі та бібліотеки, які дуже легко підключаються до мови програмування C. Python забезпечує правильний відступ коду, роблячи відступ в синтаксисі. В Python немає контрольних фігурних дужок. Блоки коду ідентифікуються рівнем відступів. Незважаючи на те, що багато програмісти не звикли до цього, це дуже цінно, оскільки він дає дуже однорідний стиль і дає код, який візуально приємний для читання. Код компілюється в байтовий код, а потім виконується на віртуальній машині. Це означає, що попередньо скомпільований код переноситься між платформами. Python це динамічно надрукована мова програмування. Це означає, що в програмі не потрібно оголошувати тип для імені змінної, а потім привласнювати щось з цього типу. Замість цього в програмі є імена змінних, і програміст прив'язує їх до об'єктів, тип яких залишається з самою сутністю. З одного боку, це робить виконання програми достатньо непередбачуваним. З іншого боку, дуже легко домогтися деяких результатів, які статичні типізовані мови дуже ускладнюють. Від традиційних машинних Python відрізняє цілий ряд характеристик. До числа його особливостей можна віднести:

- Повністю автоматичне управління пам'яттю. Ця функція дозволяє програмістам уникнути хвилювань з приводу необхідності розподіляти або звільняти пам'ять.

- Виконання операцій здійснюється в більш високому рівні абстракцій почасти через архітектури мови, почасти завдяки розширеній бібліотеці кодів, що поставляється разом з Python.

- Масив може включати об'єкти різних типів.

- Програма на Python легко об'єднується з написаними на С і С++ модулями, що дозволяє істотно збільшити швидкість програм. Завдяки цим особливостям розгортка додатків може виконуватися дуже швидко.

Основним недоліком Python є не повна контрольованість мови, оскільки відбувається часткова трансляція до внутрішньої формі коду байта, виконаного інтерпретатором. Python підтримує кілька парадигм програмування: структурний, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване. У мові прісутствет динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень і зручні високорівневі структури даних. Програмний код на Python організовується у функції та класи, які можуть об'єднуватися в модулі, а вони в свою чергу можуть бути об'єднані в пакети.

Власне згорткова нейронна мережа реалізована за рахунок використання бібліотеки Tensorflow. TensorFlow - це вільно поширювана програмна бібліотека для додатків з області машинного навчання. TensorFlow можна застосовувати для створення нейронних мереж і реалізації інших алгоритмів глибокого навчання. Ця бібліотека, випущена Google в листопаді 2015 року, від початку була написана на Python. Для навчання та оцінки створюваних моделей в ній застосовуються обчислення на CPU або GPU. Спочатку ця бібліотека створювалася для роботи на високопродуктивних серверах з використанням ресурсовитратності GPU. Останні оновлення дозволили оптимізувати цю бібліотеку і використовувати в середовищах з більш обмеженими ресурсами - наприклад, на мобільних пристроях і в веб-браузерах. Робота з TensorFlow будується навколо побудови і виконання графа обчислень. Граф обчислень - це конструкція, яка описує те, яким чином будуть проводитися обчислення. У класичному імперативний програмуванні ми пишемо код, який виконується за рядком. В TensorFlow звичний імперативний підхід до програмування необхідний тільки для якихось допоміжних цілей. Основа TensorFlow - це створення структури, яка задає порядок обчислень. Програми природним чином структуруються на дві частини - складання графа обчислень і виконання обчислень в створених структурах.

В TensorFlow граф складається з плейсхолдеров, змінних і операцій. З цих елементів можна зібрати граф, в якому будуть обчислюватися тензори. Тензори - багатовимірні масиви, вони служать «паливом» для графа. Тензором може бути як окреме число, вектор ознак з розв'язуваної задачі або зображення, так і цілий батч описів об'єктів або масив із зображень. Замість одного об'єкта ми можемо передати в граф масив об'єктів і для нього буде вирахувано масив відповідей. Робота TensorFlow з тензорами схожа на те, як обробляє масиви numpy, у функціях якого можна вказати вісь масиву, щодо якої буде виконуватися обчислення. Як і багато інших сучасних системи для наукових обчислень і машинного навчання, TensorFlow має добре документований API для Python, де тензори представлені у вигляді знайомих масивів ndarray бібліотеки NumPy. TensorFlow виконує обчислення за допомогою оптимізованого C++, а також підтримує нативний API для C і C++.

## 3.2 Архітектура та основні характеристики

Архiтектура запропонованої нейромережевої системи розпізнавання рукописних підписів користувачів інформаційних систем розроблена з позицiй iнтеграцiї рiшень розробленого методу адаптацiї структурних параметрiв згорткової нейронної мережi до умов завдання оцiнки iнформативностi бiометричних образiв з рiшеннями, застосовуваними в вiдомих нейромережевих системах бiометричної аутентифiкацiї. Запропонована структура системи оцiнки iнформативностi образiв пiдпису складається з 5 основних пiдсистем.

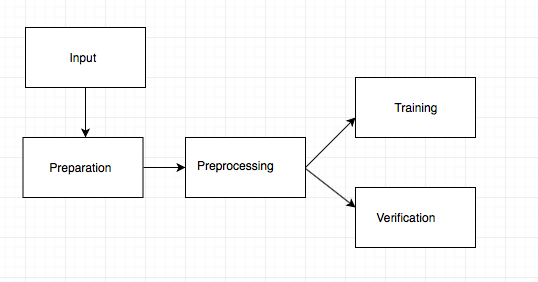


Рис. 3.1 Архiтектура розробленої системи

Призначення пiдсистем:

* Input – пiдсистема, яка подає на вхiд зображення пiдпису;
* Preparation - пiдсистема первинної обробки зображення;
* Preprocessing – пiдготовка зображення;
* Verification - пiдсистема розпiзнавання пiдпису;
* Training - пiдсистема додавання нового пiдпису в базу та тренування моделi.

Пiдсистеми “Preparation” та “Preprocessing” включають в себе первинну обробку зображень, яка подiляється на декiлька етапiв. Метою цих етапiв є пiдготовка пiдписiв до вилучення об'єктiв.

Обов’язковим кроком є бiнаризацiя зображення. Вона дозволяє виявити фон та переднiй план зображення, а також прийти до двох кольорiв та розмiтити пiкселi. Для вирiшення даної проблеми використовувала пакет, мови програмування R, *imager*. Вiн дозволяє задати пороговiсть бiнаризацiї числовими значеннями, або квантилями. Приклад бінаризованих зображень наведено на рис. 3.2.



Рис. 3.2 Приклади бінаризованих зображень рукописних підписів розміром 40х80 пікселів

Наступним кроком є скелитизацiя зображення, а саме аналiз площi iнформативних точок, що дозволяє також позбутися шумiв. Iснує декiлька методiв:  
 - Шаблона скелетизацiя;

- Хвильовий метод;

- Алгоритм Зонга-Суня;

- Алгоритм Щепiна.

В роботi використовували перший метод, шаблонної скелетизацiї. Вiн базується на отриманi шаблонiв для подальшого видалення зайвих пiкселiв. Завжди центральний чорного кольору. Процедура iтерується до тих пiр, поки не залишиться пiкселiв для видалення. На рис. 3.3 представлений вигляд шаблонiв.

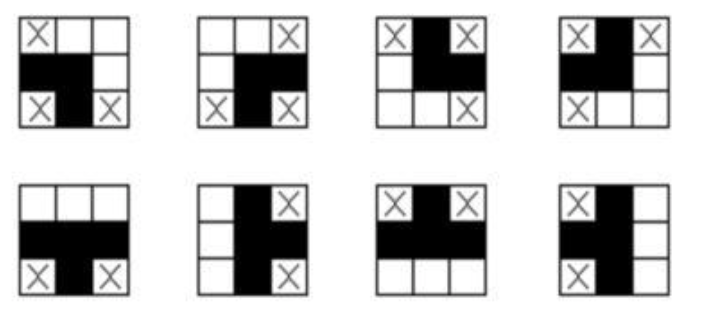


Рис. 3.3 Шаблони скелетизацiї

На рис. 3.3 клiтинки з крестиками позначенi неважливi пiкселi.

Основнi етапи скелетизацiї:

1. Для зберiгання координат створюємо пустий стек.

2. Зберiгаємо будь-яку точку скелета.

3. До тих пiр поки стек не навповнений, виконуємо наступнi кроки.

4. Обираємо точку зi стека.

5. Генерується ребра з обраної точкизображення до тих пiр, поки не вiдбудеться вiдгалуження або кiнець.

6. У випадку досягнення кiнцевої точки або нового ребра, заносимо пройденний шлях.

7. У випадку розгалуження заносимо послiдовнiсть ребер. Та повертаємося до пункту 3.

Для вирiшення даної задачi використовували пакет мови програмування Python, skimage.morfology. Також використовується змiна медiальної осi для обчислення ширини об'єктiв на першому планi. Метод medial\_axis вiддає вiддалене перетворення, крiм медиальної осi. Тому можливо обчислити вiдстань до фону для всiх точок медиальної осi з цiєю функцiєю. Це дає оцiнку локальної ширини об'єктiв.

Медiальна вiсь об'єкта - це сукупнiсть усiх точок, що мають бiльше нiж одну найближчу точку на межi об'єкта. Його часто називають топологiчним скелетом, тому що це 1-пiксельний широкий скелет об'єкта, з тим же сполученням, що й вихiдний об'єкт. Дiла використовуємо функцiю thin. Морфологiчне розщеплення, що реалiзується у данiй функцiї, працює за тим же принципом, що i скелетiнгу: видаляють пiкселi з кордонiв на кожнiй iтерацiї, доки жодна з них не може бути вилучена, не змiнюючи зв'язнiсть. Рiзнi правила видалення можуть прискорити скелетонiзацiю i привести до рiзних кiнцевих скелетiв.

Функцiя thin також приймає аргумент ключового слова max\_iter, щоб обмежити iнтенсивнiсть, таким чином, створювати вiдносно товстий скелет.

Пiдсистема “Preprocessing” включає в себе вторинну обробку зображень, змiна ромiру вхiдних зображень.

Особливостi вилучення є ключем до розробки системи. Ми використовуємо набiр iз п'яти глобальних функцiй, на якi не може впливати тимчасовий зсув.

Цi особливостi мають геометричнi характеристики на основi форми та розмiрiв зображення пiдпису. Рiзнi особливостi форми, якi ми використовуємо, є: кiлькiсть пiкселiв, горизонтальнi та вертикальнi центроїди пiдпису, спiввiдношення вiдстанi мiж краями елiпса та довжиною основної осi, мiра плоскостi розподiлу, а також мiра асиметрiї розподiлу. Пiд час процесу навчання вихiд нейронної мережi порiвнюється з цiльовим значенням, а корекцiя ваги мережi через алгоритм навчання виконується таким чином, щоб мiнiмiзувати функцiю помилки мiж двома значеннями.

В данiй роботi використали помилку MSE, яка намагається мiнiмiзувати середню похибку мiж виходом мережi та цiльовим значенням. Система перевiрена на точнiсть та ефективнiсть в базi даних близько 100 пiдписiв вiд 3 користувачiв, що мiстять як справжнiх, так i квалiфiкованих пiдроблених аналогiв зразкiв пiдписiв. Наша база даних складається з пiдписiв, виконаних з рiзними ручками з рiзними кольорами.

Всi зразки нашої бази даних були попередньо обробленi, а глобальнi функцiї були вилученi.

Пiсля вилучення функцiй, тестування виконується, i результат вiдображається, а пороговi значення було зроблено на 90% у дослiдженнi, яке нижче вiдсотка на 90%, пiдпис вважається пiдробленим.

Наступна пiдсистема “Training”. Включає в себе саме побудову навчання та тестування нейронної мережi. Розмiр ядра згорткової нейронної мережi визначає кiлькiсть ознак, якi будуть об'єднанi для отримання нової ознаки на виходi. В роботi задали 5 \* 5 = 25 ознак на входi i 3 \* 3 = 9 ознак на виходi. Для стандартного шару ми мали б масову матрицю 25 \* 9 = 225 параметрiв, а кожна вихiдна ознака була б сумою суми всiх ознак на входi. Кожна ознака на виходi отримується аналiзом не кожної ознаки на входi, а лише одного вхiдного, що знаходиться в "приблизно в тому ж мiсцi".

Cхема загального алгоритму роботи програмного комплексу розпізнавання рукописних підписів представлена на рис.3.4.

Створена комп’ютерна система розпізнавання рукописних підписів передбачає:

- Налаштування параметрів згорткової нейронної мережі.

- Навчання згорткової нейронної мережі.

- Введення файлу з текстовими даними для навчання нейромережевої моделі.

- Введення графічного файлу, в якому буде реалізовано розпізнавання рукописних підписів.

- Попередню обробку графічного файлу.

- Нейромереже розпізнавання рукописних підписів.

- Запис розпізнаних даних в текстовий файл.

Розроблене програмне забезпечення призначене для самостійного використання, однак після незначної модифікації може бути інтегроване в програмні комплекси, призначені для розпізнавання рукописних підписіву.



Рис. 3.4. Cхема загального алгоритму роботи програмного комплексу

Інтерфейс розробленої програми частково показано на рис. 3.5 та рис. 3.6.

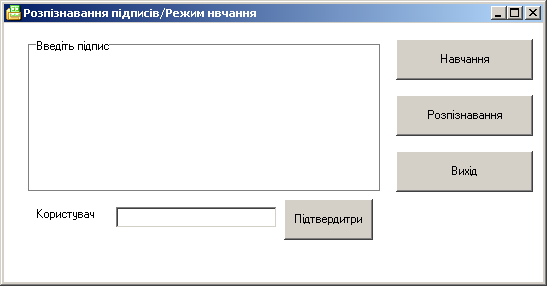


Рис. 3.5 Головне вікно програми

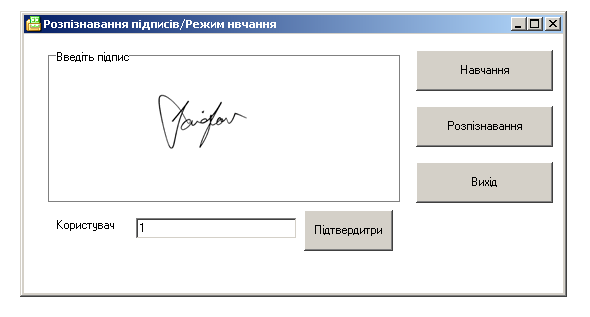


Рис. 3.6 Приклад введення параметрів рукописного підпису користувача

## 3.3 Експериментальні дослідження

Для верифікації отриманих результатів проведено ряд комп'ютерних експериментів. Експеримент проводився таким чином:

1. Вводились пiдписи, збереженi в папцi train, яка містить копiї зображень з аутентичними пiдписами.

2. Проведена обробка зображень та видiлення ознак пiдписiв.

3. Проведено навчання нейронної мережi.

4. Проведено тестування навченої нейронної мережі.

Для формування навчальної вибірки використано рукописнi пiдписи десяти користувачів. Кожен користувач розписався ручкою на листку вiсiм разiв. Пiсля цього, рукописнi пiдписи були переведенi в зображення розмiром 32 на 32 пiкселiв. Зображення мали растровий формат png.

Перед навчанням зображення рукописних пiдписiв розподiлили по двом наборам:

- набiр для навчання (зображення i мiтки класiв),

- набiр для тестування (зображення i мiтки класiв).

Набiр для навчання складався з 200 зображень, а набiр для тестування складався з 40 зображень.

Мiтки класiв мiстили правильнi вiдповiдi, тобто чий рукописний пiдпис представлено на зображеннi.

Для верифікації отриманих результатів проведені комп'ютерні експерименти, в яких розраховувалася точність розпізнавання тестових прикладів згортковою нейронною мережею при різних значеннях кількості епох навчання () і кількості карт ознак в 1-му і 2-му шарах згортки (). Досліджений діапазон зміни зазначених параметрів , , .

Експерименти проводились із використанням універсального персонального комп'ютера на основі процесора Intel (R) Core (TM) 2 Quad CPU Q6600 @ 2.40GHz з оперативною пам'яттю об'ємом 3,7 ГБ, жорстким диском обсягу 1 ТБ.

Результати експериментів показані на рис. 3.7-3.11.

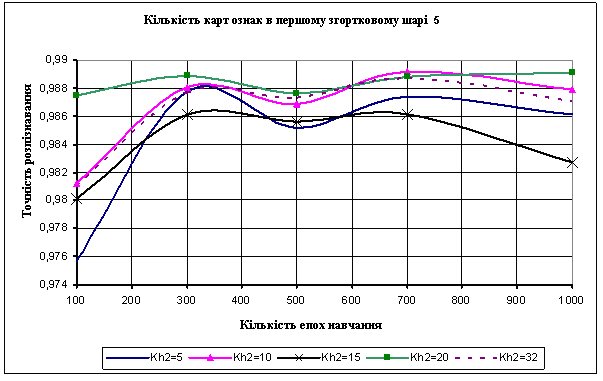


Рис. 3.7 Графіки залежності точності розпізнавання від кількості епох навчання для різних значень  при 

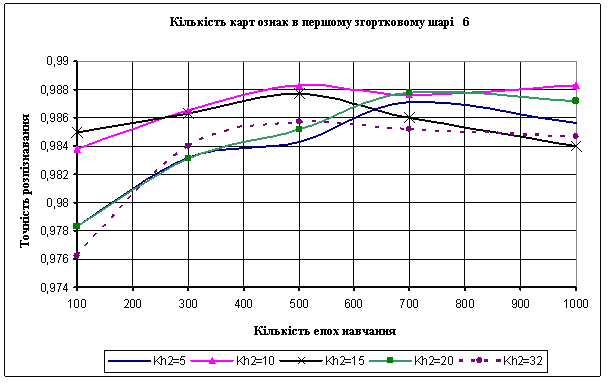


Рис. 3.8 Графіки залежності точності розпізнавання від кількості епох навчання для різних значень при 

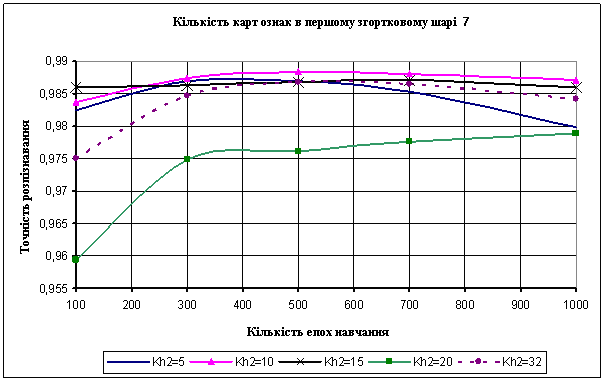


Рис. 3.9 Графіки залежності точності розпізнавання від кількості епох навчання для різних значень при 

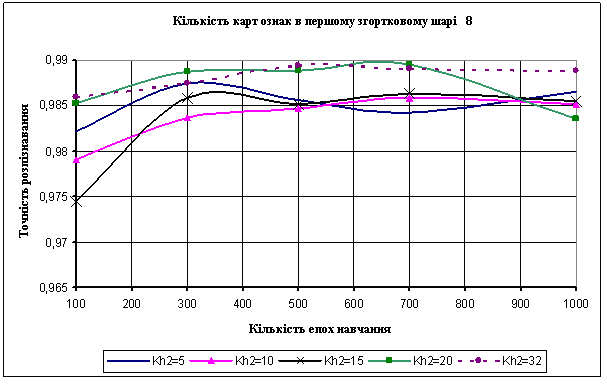


Рис. 3.10 Графіки залежності точності розпізнавання від кількості епох навчання для різних значень при 

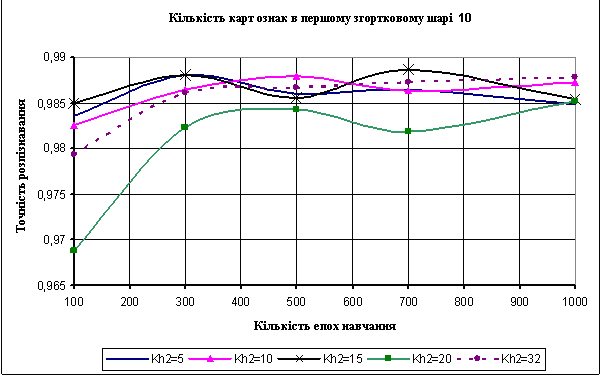


Рис. 3.11 Графіки залежності точності розпізнавання від кількості епох навчання для різних значень при 

Аналіз графіків, показаних на рис. 3.7-3.11, дозволяє стверджувати, що у вивченому діапазоні величин  та:

- Точність розпізнавання змінюється в межах , що є достатньо високою величиною для систем розпізнавання рукописних підписів.

- Збільшення кількості епох навчання понад 100 епох не призводить до гарантованого збільшення точності розпізнавання.

Тому з урахуванням необхідності мінімізації обчислювальних ресурсів найбільш ефективною буде згорткова нейронна мережа, параметри якої забезпечать максимальну точність розпізнавання при 100 епохах навчання. У нашому випадку це згорткова нейронна мережа, у якої, , а .

Підтвердженням цього факту є табл. 3.1, в якій представлені максимальні значення точності розпізнавання для кожної з можливих комбінацій при 100 епохах навчання.

*Таблиця 3.1*

Максимальна точність розпізнавання для різних комбінацій 

|  |  |  |
| --- | --- | --- |
| **Кількість карт ознак в 1-му згортковому шарі** | **Кількість карт ознак в 2-му згортковому шарі** | **Точність розпізнавання**  **рукописних підписів** |
| 5 | 20 | 0,9822 |
| 6 | 15 | 0,985 |
| 7 | 15 | 0,9859 |
| 8 | 32 | 0,986 |
| 10 | 15 | 0,9849 |

Відзначимо, що експериментальні значення  і  відповідають кількостям карт ознак, отриманим при використанні описаних механізмів побудови згорткової нейронної мережі до умов розпізнавання рукописних підписів. Таким чином, результати комп'ютерних експериментів підтверджують перспективність запропонованих рішень.

## 3.4 Висновки до розділу

Третій розділ присвячено вирішенню науково-прикладної задачі верифікації нейромережевої моделі розпізнавання рукописних підписів користувачів інформаційних систем. В результаті проведеного дослідження:

- Розроблене інструментальне забезпечення для реалізації нейромережевої системи розпізнавання рукописних підписів.

- Проведені експериментальні дослідження, спрямовані на верификацію запропонованих в роботі рішень. В результаті проведених експериментів показано, що точність розпізнавання рукописних підписів (≈0,98) відповідає точності сучасних систем аналогічного призначення і підтверджує ефективність запропонованих в роботі рішень.

# Висновки

Результатом виконаної роботи є вирішення задачі побудови нейромережевої моделі розпізнавання рукописних підписів користувачів інформаційних систем.

У процесі виконання роботи отримані наступні результати:

* 1. Показано, що перспективним шляхом підвищення ефективності розпізнавання підписів користувачів комп’ютерних систем є впровадження в них сучасних рішень терії нейронних мереж. Також визначено доцільність застосування в системі розпізнавання згорткової нейронної мережі.
  2. Розроблена структура та математичне забезпечення згорткової нейронної мережі. Застосовані рішення які дозволили пристосувати згорткову нейронну мережу до поставленої задачі розпізнавання.
  3. Розроблена архітектура нейромережевої системи розпізнавання, що адаптована до застосування розробленого математичного забезпечення.
  4. Розроблене програмне забезпечення системи розпізнавання, що базується на використанні створеного математичного та інформаційного забезпечення. Це дає можливість підвищити рівень захищеності інформаційних систем.
  5. В результаті проведених комп’ютерних експериментів показано доцільність застосування розробленої нейромережевої моделі для розпізнавання рукописних підписів користувачів. Розроблене програмне забезпечення системи розпізнавання рекомендується використовувати в засобах біометричної аутентифікації користувачів інформаційних систем загального призначення.

**СПИСОК ВИ****КОРИСТАНИХ ДЖЕРЕЛ**

1. Гонзалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. – 1072с.
2. Демин А.А. Адаптивная обработка каллиграфической информации, представленной в виде рукописных символов: дис. … канд.. техн. наук: 05.13.01 / Демин А.А. - Москва, 2014. – 182 с.
3. Колядин Д. В. Анализ динамических кривых применительно к задаче верификации рукописной подписи // Математические методы распознавания образов (ММРО-11), 2003, с. 330-332.
4. Колядин Д. В., Савин А.А. О проблеме верификации подписи в системах контроля доступа // Обработка информации и моделирование. – М.: МФТИ, 2002, с.81-89.
5. Корченко А. Нейросетевые модели, методы и средства оценки параметров безопасности Интернет-ориентированных информационных систем: монографія / А. Корченко, И Терейковский, Н. Карпинский, С. Тынымбаев. К. : ТОВ «Наш Формат». 2016. – 275 с.
6. Ланцман Р.М. Кибернетика и криминалистическая экспертиза почерка. – М.: Наука, 1968, 94 с.
7. Міхайленко В. М. Нейромережеві моделі та методи розпізнавання фонем в голосовому сигналі в системі дистанційного навчання : [Монографія] / В. М. Міхайленко, Л. О. Терейковська, І. А . Терейковський., Б. Б. Ахметов. – К. : ЦП «Компринтр», 2017.– 252 с.
8. Попова Е.С. Проектирование сверточных нейронных сетей для распознования рукописных символов / Е.С. Попова, Ю.А. Болотова // Сб. нучн. тр. XIV Международной научно-практической конференции студентов аспирантов и молодых учёных «Молодёжь и современные информационные технологии». С. 84-85.
9. Прохоров В. Г. Использование сверточных сетей для распознавания рукописных символов / В. Г. Прохоров // Прикладне програмне забезпечення. – 2008. - №2-3. – С. 674.
10. Руденко О.Г. Штучні нейронні мережі. Навч. посіб. / О.Г. Руденко, Є.В. Бодянський. – Харків: ТОВ "Компанія СМІТ", 2006. – 404 с.
11. Солдатова О.П., Гаршин А.А. Применение сверточной нейронной сети для распознавания рукописных цифр / О. П. Солдатова, А.А. Гаршин // Компьютерная оптика. – 2010. - №2. – С.252-258.
12. Терейковський І. Нейронні мережі в засобах захисту комп’ютерної інформації / І. Терейковський.  К. : ПоліграфКонсалтинг.  2007. – 209 с.
13. Умяров Н.Х., Федяев О.И. Параметрическая модель свёрточной нейронной сети // VI международная научно-техническая конференция студентов, аспирантов и молодых научных работников «Информатика и компьютерные технологии»: Т. 2 – Донецк, ДонНТУ, 2010. – 292с.
14. Хаустов П. А. Алгоритмы распознавания рукописных символов на основе построения структурных моделей / П.А. Хаустов // Компьютерная оптика. – 2017. - том 4 1, №1. – С. 67-78.
15. Хуршудов А. А. Разработка системы распознавания визуальных образов в потоке данных: дис. … канд.. техн. наук: 05.13.01 / Хуршудов А.А. - Краснодар, 2015. – 130 с.
16. Fedotov D. V. Optimisation of convolutional neural network structure with self-configuring evolutionary algorithm in one identification problem / D. V. Fedotov, E. A. Popov, V. A. Okhorzin // Vestnik SibGAU. - Vol. 16, No. 4. - P. 857–862.
17. Esme B. Facial feature extraction using genetic algorithms/ B. Esme, B. Sankur, B. Anarim // 8-th European Signal Processing Conference, Trieste, - 1996. - P. 1511-1514.
18. LeCun, Y. Efficient BackProp in Neural Networks: Tricks of the trade / Y. LeCun, L. Bottou, G. Orr, K. Muller – Springer, 1998. – 44 p.
19. R. Plamondon, G. Lorette. Automatic signature verification and writer identification – the state of the art. // Pattern Recognition, 1989, vol. 22, no. 2, pp. 107 –131.
20. M. E. Munich, P.Perona. Visual Signature Verification using Affine Arc-length // Conference on Computer Vision and Pattern Recognition CVPR, 1999, pp. 2180-2186.
21. V. S. Nalwa. Automatic On-Line Signature Verification. // Proceedings of the IEEE, 1997, vol. 85, no. 2, pp.215-239.
22. T. Starner, J. Makhoul, R. Schwartz, G. Chou. On-line cursive handwriting recognition using speech recognition methods // IEEE Conference on Acoustics, Speech, and Signal Processing, 1994, vol. 5, pp. 125-128.
23. L. Yang, B.K. Widjaja, R. Prasad. Application of Hidden Markov Models for Signature Verification. // Pattern recognition, 1995, vol. 28, no.2, pp. 161-170.
24. G. Seni, J. Seybold. Diacritical processing for unconstrained on-line handwriting recognition using forward search // International Journal on Document Analysis and Recognition, 1999, no. 1, pp. 24-29.
25. T. Hastie, E. Kishon. A Model for Signature Verification. // IEEE International Conference on Decision Aiding for Complex Systems, 1991, pp. 191-196.
26. M.E. Munich, P.Perona. Continuous Dynamic Time Warping for translation-invariant curve alignment with applications to signature verification // California Institute of Technology International Conference on Computer Vision, 1999.
27. E. Keogh, M. Pazzani. Scaling up Dynamic Time Warping for Datamining Applications. // The 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000.
28. S.Chu, E.Keogh, D.Hart, M.Pazzani. Iterative Deepening Dynamic Time Warping for Time Series // IEEE International Conference on Data Mining, Maebashi City, Japan, 2002.
29. A. K. Jain, F. D. Griess, S. D. Connell. On-line Signature Verification. // Pattern Recognition, vol. 35, no. 12, pp. 2963–2972, 2002.
30. 13. S.D. Connell, A.K. Jain. Template-based Online Character Recognition. // Pattern Recognition, 2001, vol. 34, no. 1, pp.1-14.
31. F. Hao, C. W. Chan. Online Signature Verification Using a New Extreme Points Warping Technique // Pattern Recognition Letters, 2003, vol. 24, no. 16, pp. 2943-2951.
32. X. Li, M. Parizeau, R. Plamondon. Detection of Extreme Points of On-line Handwritten Scripts // Pattern Recognition Progress in Handwriting Recognition, A. C.Downton (ed.), S. Impedovo (ed.), 1997.

# Додаток А

# Додаток Б

Файл **Network.py**

import numpy as np

import random

class NeuralNetwork():

def \_\_init\_\_(self, sizes):

# sizes is an array with the number of units in each layer

# [2,3,1] means w neurons of input, 3 in the hidden layer and 1 as output

self.num\_layers = len(sizes)

self.sizes = sizes

# the syntax [1:] gets all elements of sizes array beginning at index 1 (second position)

# np,random.randn(rows, cols) retuns a matrix of random elements

# np.random.randn(2,1) =>

# array([[ 0.68265325],

# [-0.52939261]])

# biases will have one vector per layer

self.biases = [np.random.randn(y,1) for y in sizes[1:]]

#zip returns a tuple in which x is the element of the first array and y the element of the second

#sizes[:-1] returns all the elements till the second to last

#sizes[1:] returns all the elements from the second and on]

# [2,3,1] means:

# \* matrix of 3 rows and 2 columns -- will be multiplied by the inputs

# \* matrix of 1 row and 3 columns -- will multiply the hidden layer and produce the output

self.weights = [np.random.randn(y,x) for x,y in zip(sizes[:-1],sizes[1:])]

def feedforward(self, a):

for b,w in zip(self.biases, self.weights):

a = sigmoid(np.dot(w, a) + b)

return a

def separate\_batches(self, training\_data, batch\_size):

random.shuffle(training\_data)

n = len(training\_data)

# extracts chunks of data from the training set

# the xrange function will return indices starting with 0 untill n, with a step size o batch\_size

# batches, then, will have several chunks of the main set, each defined by the batch\_size\_variable

return [training\_data[i:i + batch\_size] for i in range(0, n, batch\_size)]

def update\_batches(self, batches, alpha):

for batch in batches:

nabla\_b = [np.zeros(b.shape) for b in self.biases]

nabla\_w = [np.zeros(w.shape) for w in self.weights]

m = len(batch)

# x is a array of length 901

# y is a single value indicating the digit represented by the 901 elements

for x, y in batch:

delta\_b, delta\_w = self.backpropagation(x, y)

nabla\_b = [nb + dnb for nb, dnb in zip(nabla\_b, delta\_b)]

nabla\_w = [nw + dnw for nw, dnw in zip(nabla\_w, delta\_w)]

self.weights = [w - (alpha / m) \* nw for w, nw in zip(self.weights, nabla\_w)]

self.biases = [b - (alpha / m) \* nb for b, nb in zip(self.biases, nabla\_b)]

def backpropagation(self, x, y):

nabla\_b = [np.zeros(b.shape) for b in self.biases]

nabla\_w = [np.zeros(w.shape) for w in self.weights]

activation = x

activations = [x]

zs = []

for b, w in zip(self.biases, self.weights):

# layer-bound b and w

z = np.dot(w, activation)+b

zs.append(z)

activation = sigmoid(z)

activations.append(activation)

# backward pass

delta = self.cost\_derivative(activations[-1], y) \* \

sigmoid\_prime(zs[-1])

nabla\_b[-1] = delta

nabla\_w[-1] = np.dot(delta, activations[-2].transpose())

for l in range(2, self.num\_layers):

z = zs[-l]

sp = sigmoid\_prime(z)

delta = np.dot(self.weights[-l+1].transpose(), delta) \* sp

nabla\_b[-l] = delta

nabla\_w[-l] = np.dot(delta, activations[-l-1].transpose())

return (nabla\_b, nabla\_w)

def sgd(self, training\_data, epochs, batch\_size, alpha, test\_data):

n\_test = len(test\_data)

for epoch in range(epochs):

batches = self.separate\_batches(training\_data, batch\_size)

self.update\_batches(batches, alpha)

print("Epoch {0}: {1} / {2}".format(epoch, self.evaluate(test\_data), n\_test))

def evaluate(self, test\_data):

#r = [self.feedforward(x) for (x, y) in test\_data]

#for a in r:

# print("{0}, {1}".format(format(a[0][0], 'f'), format(a[1][0], 'f')))

test\_results = [(np.argmax(self.feedforward(x)), y)

for (x, y) in test\_data]

return sum(int(x == y) for (x, y) in test\_results)

def cost\_derivative(self, output\_activations, y):

return output\_activations - y

def sigmoid(z):

return 1.0 / (1.0 + np.exp(-z))

def sigmoid\_prime(z):

return sigmoid(z) \* (1-sigmoid(z))

import tensorflow as tf

from tensorflow.contrib.learn.python.learn.datasets.mnist import read\_data\_sets

import itertools

import os

import filters

# TODO make them command line arguments

configurations = [[5], [5, 10, 15, 20, 32]]

DATA\_DIR = './data/'

if tf.gfile.Exists(DATA\_DIR):

tf.gfile.MakeDirs(DATA\_DIR)

LOG\_DIR = './tf\_logs/'

PERFORM\_TRAINING = True

mnist = read\_data\_sets(DATA\_DIR, one\_hot=True, reshape=True)

def convolutional\_layer(inputs, size\_in, size\_out, filter\_size, filter\_init\_values=None, name='conv'):

with tf.name\_scope(name):

if filter\_init\_values is None:

w = tf.Variable(tf.truncated\_normal([filter\_size[0], filter\_size[1], size\_in, size\_out], stddev=0.1), name="kernel")

else:

w = tf.Variable(filter\_init\_values, name="kernel", trainable=False)

b = tf.Variable(tf.constant(0.1, shape=[size\_out]), name="B")

conv = tf.nn.conv2d(inputs, w, strides=[1, 1, 1, 1], padding="SAME")

act = tf.nn.relu(conv + b)

return tf.nn.max\_pool(act, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding="SAME")

def train(data, configuration, configuration\_string, max\_epochs=1000):

tf.reset\_default\_graph()

# --- specify input data

inputs = tf.placeholder(tf.float32, [None, 28 \* 28], name='x')

inputs\_reshaped = tf.reshape(inputs, [-1, 28, 28, 1])

labels = tf.placeholder(tf.float32, [None, 10], name='labels')

'''

conv1 = convolutional\_layer(inputs=inputs\_reshaped, size\_in=1, size\_out=configuration[0],

filter\_size=[5, 5], filter\_init\_values=filters.initial\_values, name='conv1')

conv2 = convolutional\_layer(inputs=conv1, size\_in=configuration[0], size\_out=configuration[1],

filter\_size=[5, 5], name='conv2')

'''

filter\_initializer = tf.constant\_initializer(filters.initial\_values)

# --- specify layers of network

conv1 = tf.layers.conv2d(inputs=inputs\_reshaped, filters=configuration[0], kernel\_size=[5, 5], padding="same",

activation=tf.nn.relu, name='conv1')

pool1 = tf.layers.max\_pooling2d(inputs=conv1, pool\_size=[2, 2], strides=2, name='pool1')

conv2 = tf.layers.conv2d(inputs=pool1, filters=configuration[1], kernel\_size=[5, 5], padding="same",

activation=tf.nn.relu, name='conv2')

pool2 = tf.layers.max\_pooling2d(inputs=conv2, pool\_size=[2, 2], strides=2, name='pool2')

flattened = tf.reshape(pool2, [-1, 7 \* 7 \* configuration[1]])

dense = tf.layers.dense(inputs=flattened, units=1024, activation=tf.nn.relu, name='fc')

logits = tf.layers.dense(inputs=dense, units=10, name='output')

with tf.name\_scope("train"):

cross\_entropy = tf.nn.softmax\_cross\_entropy\_with\_logits(labels=labels, logits=logits)

train\_step = tf.train.AdamOptimizer(0.015).minimize(cross\_entropy)

with tf.name\_scope("accuracy"):

correct\_prediction = tf.equal(tf.argmax(logits, 1), tf.argmax(labels, 1))

accuracy = tf.reduce\_mean(tf.cast(correct\_prediction, tf.float32))

tf.summary.scalar("accuracy", accuracy)

summary = tf.summary.merge\_all()

sess = tf.Session()

sess.run(tf.global\_variables\_initializer())

saver = tf.train.Saver()

dir\_name = os.path.join(LOG\_DIR, configuration\_string)

model\_filename = dir\_name + '/latest\_model/model.ckpt'

if PERFORM\_TRAINING:

if tf.gfile.Exists(dir\_name):

tf.gfile.DeleteRecursively(dir\_name)

file\_writer = tf.summary.FileWriter(dir\_name, sess.graph)

for i in range(max\_epochs):

batch\_xs, batch\_ys = data.train.next\_batch(1000) # random batch of data

sess.run(train\_step, feed\_dict={inputs: batch\_xs, labels: batch\_ys})

if i % 5 == 0:

batch\_xs\_val, batch\_ys\_val = data.test.next\_batch(1000)

summ = sess.run(summary, feed\_dict={inputs: batch\_xs\_val, labels: batch\_ys\_val})

file\_writer.add\_summary(summ, i)

elif i % 100 == 99:

run\_options = tf.RunOptions(trace\_level=tf.RunOptions.FULL\_TRACE)

run\_metadata = tf.RunMetadata()

summ, \_ = sess.run([summary, train\_step], feed\_dict={inputs: batch\_xs, labels: batch\_ys},

options=run\_options, run\_metadata=run\_metadata)

file\_writer.add\_run\_metadata(run\_metadata, 'step%03d' % i)

file\_writer.add\_summary(summ, i)

saver.save(sess, model\_filename)

else:

if tf.gfile.Exists(model\_filename + '.data-00000-of-00001'):

saver.restore(sess, model\_filename)

batch\_xs\_val, batch\_ys\_val = data.test.next\_batch(1000)

accuracy = sess.run(accuracy, feed\_dict={inputs: batch\_xs\_val, labels: batch\_ys\_val})

print('CNN with configuration: "' + configuration\_string + '" has accuracy: %.4f' % accuracy)

else:

print('Error: model file is not found')

var = [v for v in tf.trainable\_variables() if v.name == "conv1/kernel:0"]

if var:

filters.save\_as\_image(sess.run(var), [5 \* configuration[0], 5],

'conv1\_kernels', configuration\_string)

var = [v for v in tf.trainable\_variables() if v.name == "conv2/kernel:0"]

if var:

filters.save\_as\_image(sess.run(var), [5 \* configuration[0] \* configuration[1], 5],

'conv2\_kernels', configuration\_string)

def conf\_to\_string(confs):

strings = []

for configuration in confs:

string = ''

string += 'layer1=' + str(configuration[0]) + '\_\_'

string += 'layer2=' + str(configuration[1])

strings.append(string)

return strings

if \_\_name\_\_ == '\_\_main\_\_':

all\_configurations = list(itertools.product(\*configurations)) # get all possible combinations

for conf, conf\_string in zip(all\_configurations, conf\_to\_string(all\_configurations)):

train(mnist, conf, conf\_string)

print('Run "tensorboard --logdir=' + LOG\_DIR + '" to see results of training')

# Додаток В