

Szegedi Tudományegyetem
Informatikai Intézet

SZAKDOLGOZAT

Huri Ferenc Tamás

2023

**Szegedi Tudományegyetem
Informatikai Intézet**

**Offline aláírás-ellenőrzés a határpixelek-
ből származó iránykódok elemi kombinációinak
felhasználásával**

Szakdolgozat

Készítette:
Huri Ferenc Tamás
programtervező
informatikus szakos
hallgató

Témavezető:
Dr. Németh Gábor
adjunktus

**Szeged
2023**

Feladatkiírás

A témát kiíró oktató neve: Dr. Németh Gábor

A témát meghirdető tanszék: Képfeldolgozás és Számítógépes Grafika Tanszék

Típus: Szakdolgozat

Ki jelentkezhethet: 1 fő, Programtervező informatikus BSc, Mérnökinformatikus BSc, Gazdaságinformatikus BSc szakos hallgató

A feladat rövid leírása:

Az offline aláírás hitelesítési eljárások az aláírás képi jellemzőit veszik alapul. Számos jellemző mellett az aláírás kontúrjának vonalszegmensei, azok típusai és darabszáma is informatív lehet.

M. D. Ajij és szerzőtársai (2021) az aláírások kontúrját vonalszegmensekre bontották, majd azokat alakjuk szerint osztályokba sorolták. Az egymást követő vonalszegmensek, valamint az egyes vonalszegmens-típusok száma az aláírás kontúrján megfelelő jellemző lehet tanulmányuk szerint.

A jelentkező feladata az Ajij és szerzőtársai által javasolt eljárás implementálása tetszőleges programozási nyelven és annak kiértékelése.

Előismeret: nem szükséges

Szakirodalom: M. D. Ajij, S. Pratihari, S. R. Nayak, T. Hanne, D. S. Roy: Off-line signature verification using elementary combinations of directional codes from boundary pixels. Neural Computing and Applications, 2021

Tartalmi összefoglaló

- **A téma megnevezése:**

Offline aláírás-ellenőrzés a határpixelekkel származó iránykódok elemi kombinációinak felhasználásával

- **A megadott feladat megfogalmazása:**

Az offline aláírás-ellenőrzés célja a digitális aláírások automatizált és megbízható ellenőrzése. A rendszer a kézírás kvázi-egyenes szakaszainak elemzésével és jellemzővektorok kialakításával azonosítja az egyedi aláírásokat, lehetővé téve a megbízható és hatékony azonosítást. A módszer alapja a kontúr vonalszegmensekre történő felbontása, majd ezek kapcsolatának gépi tanulása, amely során a rendszer a hamis és a valódi aláírásokat rögzíti és azokat felhasználva hoz létre egy személyre szabott verifikációs eljárást.

- **A megoldási mód:**

A rendszer a kézírás kontúrjának kvázi-egyenes szakaszait határozza meg, és ezek alapján nyer ki jellemző vektorokat. Az egyedi vektorokat tanító adatok alapján validálja, lehetővé téve az aláírások automatizált és megbízható azonosítását. A gépi tanulás alkalmazása segít a rendszernek alkalmazkodni és optimalizálni az azonosítási folyamatot a felhasználó egyedi írásmódjához.

- **Alkalmazott eszközök, módszerek:**

Programozási nyelv: Python

Könyvtárak: Numpy, Pandas, Scikit-learn, Matplotlib, Concurrent, OpenCV

Módszerek: objektumorientált programozás, párhuzamos programozás, hiperparaméterek használata a gépi tanulás teljesítmény javításához.

Gépi tanulási modellek:

- SVM: LINEAR, RBF, POLY, SIGMOID

- Véletlen erdők (Random Forest Classifier)

- **Elért eredmények:**

- **Kulcsszavak:**

Aláírás-ellenőrzés, offline, svm, randomforest

Tartalomjegyzék

Feladatkiírás	4
Tartalmi összefoglaló	5
Tartalomjegyzék.....	6
BEVEZETÉS	8
1. ALAPFOGALMAK ÉS HÁTTÉR.....	9
1.1. Aláírások és aláírások verifikációja.....	9
1.1.1. Aláírások jelentősége	9
1.1.2. Offline és online aláírások közötti különbségek.....	9
1.2. Technológiai alapok	11
1.2.1. Digitális képfeldolgozás	11
1.2.2. Jellemzők kinyerése a képekből.....	12
1.2.3. Osztályozó algoritmusok alkalmazása.....	13
2. OFFLINE ALÁÍRÁSOK TULAJDONSÁGAI	14
2.1. Offline aláírások változatos tulajdonságai.....	14
2.1.1. Offline aláírások egyedisége	14
2.2. Kvázi-egyenes szakaszok	15
2.2.1. Az aláírások strukturált elemzése	15
2.2.2. Kvázi-egyenes szakaszok vizsgálata	16
3. ALGORITMUSOK MEGVALÓSÍTÁSA	18
3.1. Detect-Quasi-Straight-Segments (DQSS) algoritmus	18
3.1.1. Inicializáció	18
3.1.2. Szakaszok detektálása és kiterjesztése	19
3.2. Extend-Segment eljárás	20
3.2.1. Működési elv	20
3.2.2. Kód implementáció	21
3.3. Freeman lánc kód	23
3.3.1. Freeman lánc kód implementáció.....	24
3.4. Példák a kvázi-egyenes szakaszok kinyerésére.....	24
4. ALÁÍRÁS MINTA JELLEMZŐK.....	25
5. OSZTÁLYOZÓ ALGORITMUSOK.....	28
5.1. SVM (Support Vector Machine)	28

5.2. Véletlen erdők (Random forests)	30
6. EREDMÉNYEK KIÉRTÉKELÉSE	30
6.1. Metrikák.....	31
6.2. Aláírás konzisztencia.....	32
ÖSSZEFOGLALÁS.....	33
IMPLEMENTÁCIÓ	33
EREDMÉNYEK ÉS TAPASZTALATOK	34
TOVÁBBI FEJLESZTÉSI LEHETŐSÉGEK.....	34
<i>IRODALOMJEGYZÉK</i>	36
Nyilatkozat	37
Köszönetnyilvánítás	38

BEVEZETÉS

Az offline aláírás-ellenőrzés kulcsfontosságú területe a biometrikus azonosításnak, ahol a kézírásalapú azonosítás rendkívüli jelentőséggel bír mind a biztonsági, mind az automatizált azonosítási rendszerek terén. Az offline aláírásverifikáció az aláírás képét vizsgálva arra keresi a választ, hogy az aláírások valódiak-e, vagy hamisítottak. A fejlett digitális technológiák térnyerésével az offline aláírás-ellenőrzés különféle iparágakban, mint például a banki szektor vagy az okmányigazgatás, egyre inkább előtérbe kerül.

A kihívások között szerepel az offline aláírások változatosságának kezelése, az érintett területeken történő skálainvariáns azonosítás, és az aláírások részleteinek érzékeny kezelése. Jelen dolgozat célja, hogy implementálja egy adott, a dolgozat készítését motiváló offline aláírásverifikációs eljárást [1], kitérve az ott alkalmazott eljárások hatékonyságára, alkalmazási területeire és a jövőbeli fejlesztési irányokra.

Az offline aláírásverifikáció mélyebb megértéséhez az alapfogalmak és a technológiai háttér részletes vizsgálata szükséges. A digitális képfeldolgozás és gépi tanulás elveit felhasználva, az offline aláírásokat reprezentáló jellemzők és az azokra alkalmazott verifikációs módszerek részletes elemzése által kívánjuk árnyaltabban bemutatni ezt a komplex területet. A fejlesztett algoritmusok és kísérletek eredményeinek prezentálása során az olvasó betekintést nyerhet a gyakorlati alkalmazásokba és a módszerek hatékonyságába.

A továbbiakban a kutatás kiemelt területei között szerepel a skálainvariancia vizsgálata és a jellemzők szelekciójának fontossága. A kihívások feltárása és a jövőbeli irányok felvázolása hozzájárul a kutatás területének fejlődéséhez és az offline aláírásverifikáció hatékonyságának fokozásához. Ezen felül, a dolgozat tartalmazza a kísérletek során alkalmazott standard adathalmazok részletes bemutatását, amelyek révén az eredményeket értékeljük és azokat az iparban valós alkalmazásokra alkalmazhatóvá tesszük. Az olvasó így átfogó képet kap a kutatás háttéréről, módszertanáról és elérhető eredményeiről.

Az offline aláírás-ellenőrzés mélyreható vizsgálata és hatékony megoldásai nem csak az információbiztonság terén játszanak kulcsszerepet, hanem számos más területen is potenciális alkalmazási lehetőségeket rejt magában. Ezen érvek fényében, az algoritmus megvalósításának célja a jelenlegi állapot átfogó elemzése, új megközelítések bemutatása és a terület előremutató fejlesztésének támogatása.

1. Alapfogalmak és háttér

1.1. Aláírások és aláírások verifikációja

1.1.1. Aláírások jelentősége

Az aláírások hosszú ideje központi szerepet játszanak a személyazonosság hitelesítésében és a dokumentumok érvényességének igazolásában. Az emberi kézírás egyéni vonásai, stílusa és dinamikája alapján az aláírások olyan személy-specifikus azonosítók, melyek elengedhetetlenek a jogi és üzleti tranzakciók során. A hagyományos papíralapú környezetből való áttérés a digitális világba kiemeli az offline aláírásverifikáció fontosságát, amely kifejezetten a képi alapú aláírások vizsgálatát jelenti. Ez a folyamat a fizikai aláírások képét elemezi, és nem a digitálisan rögzített adatokat, amelyek inkább az online aláírásverifikáció területéhez tartoznak.

Az aláírásoknak többféle szempontból is kiemelt jelentősége van. Először is, az aláírások szolgálnak azonosítóként a jogi dokumentumokon. A hiteles aláírás igazolja, hogy a dokumentum tartalma a megjelölt személy által jóváhagyott és elfogadott. Ezzel az aláírással kijelenthető, hogy a személy vállalja a dokumentumban leírt felelősséget vagy az abban foglaltak elfogadását.

Másodszor, az aláírások a pénzügyi tranzakciókban is alapvetőek. Bankszámlák, szerződések, vagy akár szállítólevelek esetében az aláírások segítik azonosítani és ellenőrizni az érintett feleket. Ezen keresztül hozzájárulnak a csalások és visszaélések megelőzéséhez, mivel csak a jogosult személyek képesek érvényes aláírást készíteni.

Mindezek a tényezők kiemelik az offline aláírásverifikáció kihívásait és szükségességét. A hagyományos, kézzel írt aláírások digitalizációja és verifikációja terén felmerülő technológiai fejlesztések egyre növekvő jelentőségűek a modern társadalom és gazdaság szempontjából. Az aláírásoknak tehát nem csak a fizikai dokumentumokon, hanem az elektronikus platformokon is kulcsszerepet kell játszaniuk az azonosítás és az érvényesség terén.

1.1.2. Offline és online aláírások közötti különbségek

Az aláírások terén az offline és online megközelítések közötti különbségek kulcsfontosságúak a modern aláírásverifikációs rendszerek kialakításában. Az offline

aláírások azokat a hagyományos papíralapú aláírásokat jelentik, melyeket kézzel írnak egy fizikai dokumentumon, például egy szerződésen vagy bankszámlakivonaton. Ezek az aláírások a papír és a tintagyűjtő alapú hagyományos írásos formában rögzülnek, és az emberi kéz írásbeli stílusának egyedi jegyeit hordozzák.

Az online aláírások viszont az elektronikus formában rögzített aláírásokat jelentik. Ezek az aláírások digitálisan generált, és különböző eszközökkel, például érintőképernyőkön vagy digitális aláíráspadokon készülnek. Az elektronikus aláírások olyan formátumot öltenek, amely digitális fájlban tárolható és könnyen továbbítasz elektronikus dokumentumokkal együtt.

Különbségek az offline és online aláírások között:

1. Fizikai manifesztáció vs. Digitális reprezentáció:

- **Offline:** Fizikai megnyilvánulások, mint például a tintafoltok a papíron, vagy a nyomok a felszínen.
- **Online:** Digitális reprezentáció, amely számok és karakterek formájában tárolódik digitális formátumban.

2. Kézírásbeli jelleg vs. Digitális eszközhasználat:

- **Offline:** Az offline aláírásokat kézírással készítik, és az írásbeli stílus személyre szabott és egyedi.
- **Online:** Az online aláírásokat digitális eszközök segítségével készítik, és a digitális ceruzák vagy érintőképernyők alkalmazásával jönnek létre.

3. Fizikai dokumentumok vs. Elektronikus dokumentumok:

- **Offline:** Az offline aláírások jellemzően papíralapú dokumentumokhoz kapcsolódnak.
- **Online:** Az online aláírások elektronikus dokumentumokkal vannak összekapcsolva, amelyek számítógépes rendszereken vagy az interneten keresztül továbbíthatók.

4. Azonosítás módja:

- **Offline:** Az offline aláírásokat az írásbeli stílus egyedi vonásai alapján azonosítják.
- **Online:** Az online aláírásokat gyakran két faktoros azonosítókkal, például jelszavakkal vagy biometrikus adatokkal ötvözik.

1.2. Technológiai alapok

1.2.1. Digitális képfeldolgozás

A digitális képfeldolgozás olyan kulcsfontosságú technológiai terület, amely számos alkalmazási területen, beleértve az aláírásverifikációt is, kiemelkedő fontosságú. A digitális képfeldolgozás fő célja a digitális képek elemzése, manipulációja és értelmezése. Az aláírások verifikációjában a digitális képfeldolgozás rendkívül hasznos, mivel lehetővé teszi az aláírások számos jellegzetességének kinyerését és elemzését. Az alábbiakban megvizsgáljuk a digitális képfeldolgozás néhány kulcsfontosságú aspektusát, különös tekintettel az aláírások verifikációjára.

Digitális képfeldolgozás alapfogalmai:

A digitális képfeldolgozás a képeket számítógépes rendszerekben értelmezi és manipulálja. A digitális képek matematikai reprezentációját bináris kódok alkotják, ahol minden képpont egy adott pixel értékét jelöli. Ez a matematikai megközelítés lehetővé teszi a képek számos jellemzőjének analízisét, beleértve a színeket, a textúrát, a kontrasztot és az alakot.

Gépi tanulás és a mesterséges intelligencia alkalmazása:

A digitális képfeldolgozásban alkalmazott gépi tanulás (Machine Learning) és mesterséges intelligencia (AI) technikák jelentős előrelépést hoztak az aláírásverifikációban. Az intelligens rendszerek képesek tanulni az egyedi aláírói stílusok változásairól és alkalmazkodni az új aláírásokhoz. A mélyneuronhálók tanuló algoritmusai kiválóan alkalmasak az összetett és magas dimenziós jellemzők felismerésére, amelyek az aláírásoknál gyakran előfordulnak.

Biztonság és adatvédelem:

A digitális képfeldolgozásnak kulcsszerepe van az aláírások verifikációjának biztonságában és adatvédelmében. Az aláírások digitális lenyomatának kinyerése és tárolása korszerű kriptográfiai technikákkal és biztonsági protokollokkal védi az érzékeny adatokat.

A digitális képfeldolgozás alkalmazása az aláírásverifikációban átfogó és hatékony megoldást kínál az egyedi aláírói stílusok azonosítására és a hamisítások elleni védelemre. Az ezen technológiákba beépített intelligencia és adaptivitás tovább növeli az aláírásverifikáció hatékonyságát és pontosságát a változó aláírói környezetekben.

1.2.2. Jellemzők kinyerése a képekből

A digitális aláírásverifikációban a jellemzők kinyerése a kulcsfontosságú lépés, amely lehetővé teszi az aláírói stílusok egyedi azonosítását és megkülönböztetését. A digitális képfeldolgozás alkalmazásakor számos technika segítségével különböző jellemzőket nyerhetünk ki aláírások képéből.

Aláírói stílusok kulcsjellemzői:

1. **Vonalak és ívek:** Az aláírásokban szereplő vonalak és ívek meghatározó jellemzők, amelyek segítségével az aláírói stílus könnyen felismerhetővé válik. A digitális képfeldolgozás során az algoritmusok képesek az egyes vonalak és ívek paramétereit kinyerni, például hosszúságukat, ívhosszúságukat és ívhajlításukat.
2. **Alakjellemzők:** Az aláírások alakjait és geometriai tulajdonságait, mint például a zárt alakokat, az ívek száma és a kontúrok alakja, további jellemzőket kínálnak. Ezeket a jellemzőket a digitális képfeldolgozási algoritmusok hatékonyan kinyerik és analizálják az azonosítási célokra.

Jellemzők kinyerése digitális képfeldolgozással:

1. **Kontúrkövetés:** A kontúrkövetés során az algoritmusok az aláírásokon található vonalakat és íveket követik. Ez lehetővé teszi a kontúr hosszának és formájának pontos meghatározását, amelyek alapján azonosító jellemzőket kinyerhetünk.
2. **Pixelanalízis:** A pixelanalízis segítségével az algoritmusok számolhatják a pixelek számát az egyes régiókban és osztályozhatják azokat az előre definiált kategóriákba.

Előnyök és kihívások:

A digitális aláírások jellemzőinek kinyerése számos előnyt és kihívást rejt. Az előnyök közé tartozik az aláírói stílusok egyedi és részletes azonosítása, valamint az aláírások

közötti különbségek pontosabb felismerése. Ugyanakkor a kihívások közé tartozik az adaptív algoritmusok szükségessége az egyedi írási stílusok változékonyságához, valamint a magas dimenziós jellemzők hatékony kezelése.

A digitális képfeldolgozás lehetőséget kínál az aláírói stílusok számos jellemzőjének hatékony kinyerésére és az aláírásverifikáció hatékonyságának javítására. Az innovatív algoritmusok és a gépi tanulás alkalmazása tovább növeli a rendszerek pontosságát és alkalmazkodóképességét az egyre változó aláírói környezetekben.

1.2.3. Osztályozó algoritmusok alkalmazása

A digitális aláírásverifikáció területén az osztályozó algoritmusok kardinális szerepet játszanak az aláírások érvényességének és az azonosító jellemzők kinyerésének folyamatában. Ezek az algoritmusok felelnek azért, hogy az aláírásokat két kategóriába sorolják: a valódi (genuine) és a hamis (forged) aláírásokba.

Osztályozó algoritmusok szerepe:

1. **Gépi tanulás alapú osztályozók:** A gépi tanulás alkalmazása lehetővé teszi olyan osztályozó algoritmusok létrehozását, amelyek sajátítják el az aláírói stílusok és jellemzők finom különbségeit. Az SVM (Support Vector Machine), döntési fák, és neurális hálók széles körben használt gépi tanulósos osztályozók ezen a területen.
2. **Statikus osztályozó módszerek:** A statikus osztályozó módszerek közé tartozik a k-NN (k-Nearest Neighbors) algoritmus, ahol az adott aláírást az ehhez legközelebbi k darab tanító adatpont alapján osztályozzák. Ez a módszer gyors és egyszerű, különösen kisebb adathalmazok esetén.
3. **Döntési fák és Random Forests:** A döntési fák és Random Forests olyan módszerek, amelyeknél a döntési fa struktúráját felhasználják az osztályozáshoz. A Random Forests több fa összeállításával tovább növeli az osztályozó rendszer megbízhatóságát és ellenállóképességét.

Osztályozó algoritmusok képességei:

1. **Skálainvariancia:** Az osztályozó algoritmusok képesek skálainvariáns jellemzők felismerésére, így az aláírások különböző méretei esetén is hatékonyan működnek. Ez

lehetővé teszi a rendszerek alkalmazását változatos eszközökkel, például tabletek vagy érintőképernyők használatakor.

2. **Szöveges jellemzők:** Az osztályozók nemcsak vizuális, hanem szöveges jellemzőkkel is dolgozhatnak, például az aláírások időbeli adattartamával és sebességével. Ez további információkat szolgáltat az aláírói stílusokról.

Osztályozó algoritmusok kihívásai:

1. **Overfitting és regularizáció:** Az osztályozó algoritmusoknak ellenállniuk kell az overfitting veszélyének, amikor a modell túlságosan alkalmazkodik a tanítóadatokhoz. Az ilyen kihívások elleni védelem érdekében a regularizációs technikákat alkalmazzák.
2. **Adatok kevéssége:** A gépi tanulási algoritmusok nagy mennyiségű tanító adatot igényelnek a hatékony működéshez. Az aláírások esetében nehéz lehet elegendő mennyiségű változatos adatot gyűjteni minden egyes felhasználótól.

2. Offline aláírások tulajdonságai

Ebben a fejezetben részletesen megvizsgáljuk az offline aláírások jellemzőit és tulajdonságait, amelyek alapvető fontosságúak a digitális aláírásverifikáció során.

2.1. Offline aláírások változatos tulajdonságai

2.1.1. Offline aláírások egyedisége

Az offline aláírások sokféle jellemzőt magukban foglalnak, amelyek segítenek az azonosításban és verifikációban. Az egyik alapvető tulajdonság a **vonások és ívhosszak**. Ezek a digitálisan rögzített aláírásokban találhatók, és fontos információkat hordoznak az író egyedi stílusáról.

- **Vonások:** Az aláírások általában vonásokból állnak, amelyek az írás közbeni folytonos mozdulatokat reprezentálják. A vonások hossza, iránya és ívének karakterisztikái mind olyan jellemzők, amelyek alapján azonosítható az író.
- **Ívhosszak:** Az írás során alkalmazott ívhosszak is fontosak. Az írás sebessége és az ívek hossza egyaránt változhat az egyéni írói stílustól függően.

Ezen kívül a következők is fontosak:

- **Nyomásérzékenység:** Az offline aláírásokon a nyomás erőssége is megfigyelhető egyrészt a vonal tónusából, másrészt az elfolyt tinta nyomából, vagyis a vonalvastagságból. Az írás közbeni nyomás változása további információkat szolgáltat az azonosításhoz.

A digitális képfeldolgozás és az algoritmusok alkalmazása lehetővé teszi ezeknek a tulajdonságoknak az elemzését és kinyerését a verifikáció során. A következő részekben részletesen kifejtjük ezeket az offline aláírások jellemzőit és mutatóit, amelyek alapján hatékonyan végezhető el az aláírások verifikációja.

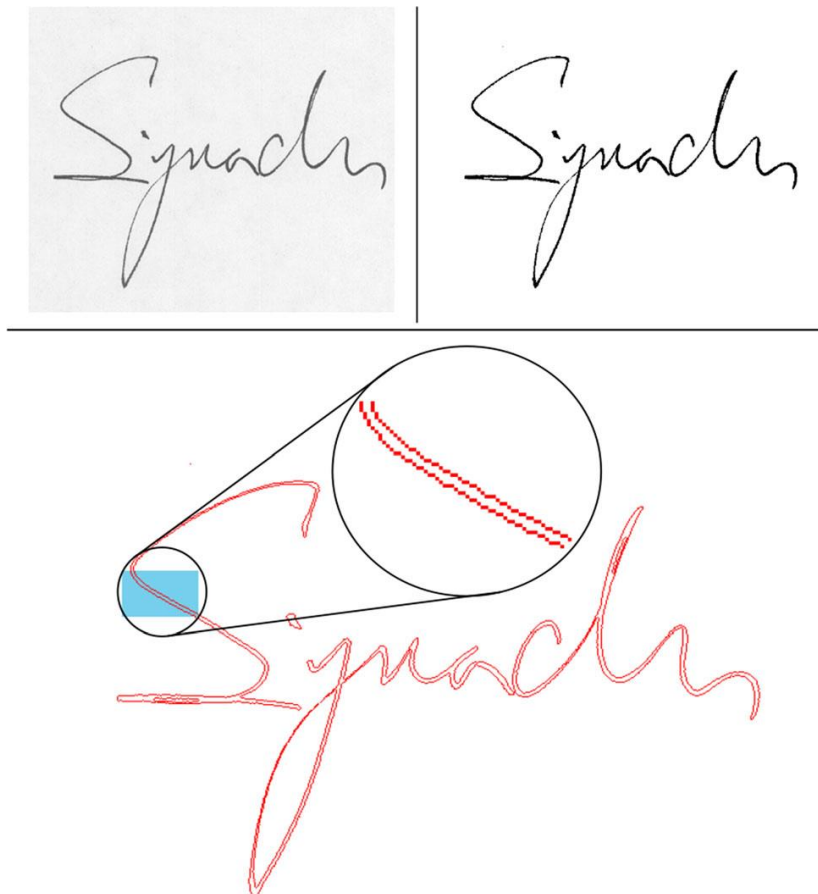
2.2. Kvázi-egyenes szakaszok

2.2.1. Az aláírások strukturált elemzése

Az offline aláírások strukturált elemzése során a kvázi-egyenes szakaszok kiemelt jelentőséggel bírnak [1]. Ezek a szakaszok olyan összefüggő pixelhalmazok, amelyek közelítően egyenes vonalat alkotnak az aláírás képén. Az aláírások strukturált elemzése a következő fontos területekre összpontosít:

1. **Kvázi-egyenesek hossza és iránya:** Az egyes kvázi-egyenes szakaszok hossza és iránya információval szolgál az író írásmódjáról. Az írás egyedi stílusa megjelenik az egyenes vonalak hosszában és irányában, így hozzájárulva a verifikációs folyamathoz.
2. **Szomszédos kvázi-egyenesek kapcsolata:** Az egyes kvázi-egyenes szakaszok kölcsönhatása és kapcsolata a szomszédos szakaszokkal fontos részleteket rejthet el. Az egymáshoz közel elhelyezkedő egyenesek vagy azok kölcsönhatása különféle írásstílusjegyeket fedhet fel.
3. **Rugalmas variabilitás a szegmensek között:** Az aláírások strukturált elemzése során a rugalmasság és variabilitás a szegmensek között is vizsgálat alá kerül. Az egyedi vonások nem csak az egyes szakaszokban, hanem azok kapcsolatában és átmenetében is megnyilvánulnak.
4. **Szakaszok geometriai elrendezése:** Az egyes szakaszok geometriai elrendezése, azok térbeli elhelyezkedése és összefüggése további információval szolgál az író egyedi stílusáról. A különböző geometriai minták és elrendezések szignifikánsak lehetnek a verifikáció szempontjából.

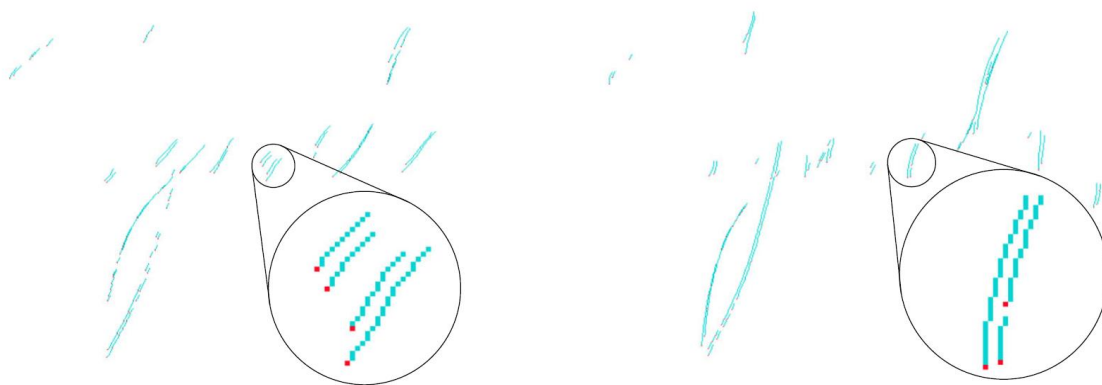
A kvázi-egyenesek strukturált elemzése lehetővé teszi az aláírások egyedi jellemzőinek kiemelését és azok hatékony felhasználását az azonosítási és verifikációs folyamatokban. Az ilyen részletes vizsgálatok segítik a rendszerek fejlesztését, amelyek a digitális aláírásokat használják az azonosítási és hitelesítési célok eléréséhez.



2.1. ábra: Egy aláírás szkennelt képe (balra fent), a szegmentál íráskép (jobbra fent), a kontúr egy kvázi-egyenes szakasza (lent) [1]

2.2.2. Kvázi-egyenes szakaszok vizsgálata

A kvázi-egyenes szakaszok elemzése kulcsfontosságú tényező az offline aláírások vizsgálatában. Ezek a szakaszok nem csupán vonalak halmazai, hanem azok hossza és alakja alapján is tovább bonthatók, létrehozva ezzel olyan részletezett jellemzőket, amelyek értékes információkat nyújtanak az aláírások azonosításához és verifikációjához.



2.2. ábra: Kvázi-egyenes szakaszok kinyerése irány szerint [1]

A kvázi-egyenes szakaszok (**Quasi-straight Line - QSL**) olyan vonalak, amelyek közelítően egyenesek, de lehetnek kisebb görbületekkel vagy inhomogenitásokkal. Ezek a szakaszok az aláírásokban gyakran megjelennek, és a következőképpen alkalmazhatók a verifikációs folyamat során:

1. **Jellemzők kinyerése a QSL szegmensekből:** Az egyes QSL szegmensek részletes jellemzőinek kinyerése, például hossz, irány és görbület, segít a rendszernek azonosítani az író egyedi írásmódját. Az ilyen részletes jellemzők hozzájárulnak az azonosítási pontossághoz.
2. **Átfedés és kapcsolatok vizsgálata:** Az átfedések és a szakaszok közötti kapcsolatok elemzése szintén fontos. Az egymáshoz közel elhelyezkedő vagy átfedő QSL szegmensek különféle írási stílusjegyeket tárhatnak fel, kiegészítve az azonosítás pontosságát.
3. **Invariáns jellemzők meghatározása:** A QSL szegmensek segítenek az invariáns jellemzők meghatározásában. Ezek azok a jellemzők, amelyek kevésbé érzékenyek az aláírás méretváltozásaira vagy elforgatásaira, és ezáltal a rendszer skálainvariáns és elforgatásinvariáns tulajdonságokkal rendelkezik.
4. **Hibatűrő verifikáció:** A QSL szegmensek részletes elemzése lehetővé teszi a hibatűrőbb verifikációt. Az író egyediségét tükröző részletek fokozottabb figyelmet kapnak, csökkentve ezzel a hamis pozitív vagy negatív eredmények kockázatát.
5. **Kombinált jellemzők értékelése:** Az egyedi QSL szegmensek jellemzőit kombinálhatjuk más aláírásjellemzőkkel, például alakzatokkal vagy irányokkal, maximalizálva ezzel az aláírások komplex jellemzőinek kihasználását.

A QSL szegmensek hatékony alkalmazása az offline aláírások tulajdonságainak elemzésében javítja a rendszerek teljesítményét, és elősegíti a pontosabb azonosítást és verifikációt az aláírások területén.

3. Algoritmusok megvalósítása

A Quasi-Straight-Segments (QSS) algoritmusok implementálása kulcsfontosságú a kézírás-alapú azonosításban. Ebben a fejezetben részletesen bemutatom egy, a Python [2] programozási nyelvben megvalósított **EdgeProcessor** osztályt, mely tartalmazza a kvázi-egyenes szakaszok kinyeréséhez és elemzéséhez szükséges algoritmusok megvalósításait.

3.1. *Detect-Quasi-Straight-Segments (DQSS) algoritmus*

A DQSS algoritmus a kvázi-egyenes szakaszok azonosítását végzi el, és a kép területét részzszakaszokra osztja a további feldolgozás érdekében. Az algoritmus megvalósítása a következőképpen történik:

3.1.1. Inicializáció

Az algoritmus kezdetben előkészíti a szükséges változókat, például a **Q** mátrixot, ahol tároljuk a kvázi-egyenes szakaszokat. A folyamat során megadható továbbá a kiterjesztési küszöbérték (**L**), amely meghatározza, hogy egy szakasznak legalább hány pontból kell állnia.

```

def DQSS(self, n: int, s: int, l=4):

#####
#####

    # Az i-edik osztályban lévő él szegmenseinek száma
    ni = 0

    # Az E élhalmazban lévő, az i-edik osztályba tartozó
    él

    # pixeleinek száma
    pi = 0

#####
#####

    Q = [] # A kvázi-egyenes szakaszokat tároló mátrix
    qi = [] # Az éppen feldolgozott szakaszt tároló lista
    Visited = [] # A már feldolgozott pixeleket tároló
    lista
    p1, p2 = None, None # Az éppen vizsgált pixelek

    [p1,p2]=self.get_two_unvisited_sibling_pixels_by_n(p1,n,
    Visited)

    if p2 is None:
        return [[], 0, 0]

    qi = [p1, p2]
    Visited = [p1, p2]

```

A fenti kódrészlet mutatja a kezdeti szakaszpárok kiválasztását, amelyek a kvázi-egyenesek keresésének kiindulópontját képezik.

3.1.2. Szakaszok detektálása és kiterjesztése

Miután inicializáltuk a változókat és kiválasztottuk az első szakaszpárokat, elkezdődik a kvázi-egyenesek detektálása és kiterjesztése. Az algoritmus folyamatosan bővíti a szakaszokat mindkét irányba, figyelembe véve a szomszédos pixeleket.

```

while True:
    self.extend_segment(p2, n, s, qi, Visited)
    self.extend_segment(p1, (n+4) % 8, ((s+4) % 8 if s != EPSILON else EPSILON),
                        qi, Visited)
    if len(qi) >= l:
        Q.append(qi.copy()) # A megfelelő hosszúságú szakaszokat tároljuk
        qi = [] # Az aktuális szakaszt töröljük
    [p1, p2] = self.get_two_unvisited_sibling_pixels_by_n(p2, n, Visited)
    if p2 is None:
        return [Q, len(Q), sum(map(len, Q))]
    qi = [p1, p2]
    Visited.append(p1)
    Visited.append(p2)

```

Ebben a szakaszban a kódban látható, hogy az algoritmus folyamatosan vizsgálja a szomszédos pixeleket és bővíti a szakaszokat, amíg el nem éri a megadott hosszt (l). Az érvényes szakaszokat a Q mátrixban tároljuk, amely az algoritmusból visszatérés után elérhető lesz a további elemzésekhez.

3.2. Extend-Segment eljárás

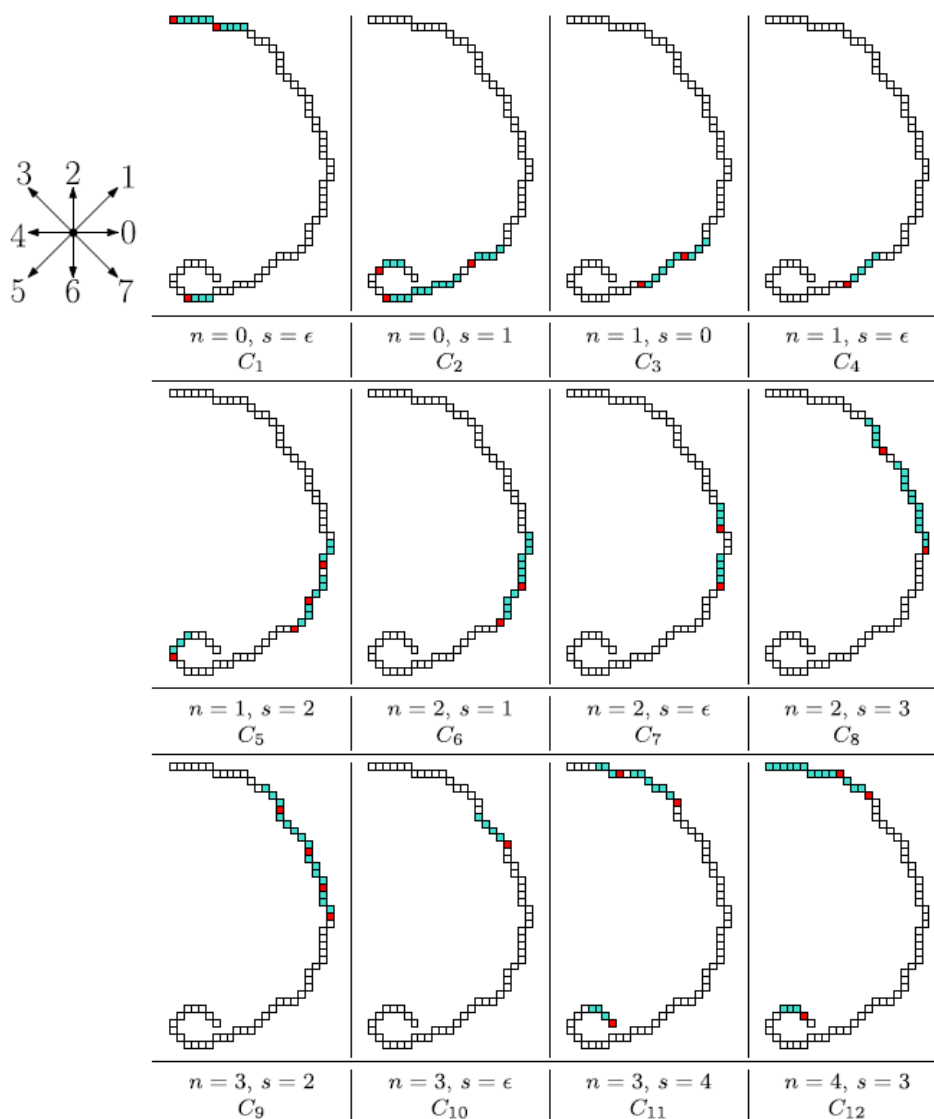
Az **Extend-Segment** eljárás az **EdgeProcessor** osztályban része a kvázi-egyenes szakaszok detektálását végző **DQSS** algoritmusnak. A feladat az aktuális szakasz további kiterjesztése az előzőleg meghatározott irányban.

3.2.1. Működési elv

Az eljárás a következő paraméterekkel rendelkezik:

- p : az aktuális pixel koordinátái
- n : a Freeman lánc kód iránya, amerre az új pixel keresése történik
- s : a szükséges második irány, amelyet csak egy második irány esetén veszünk figyelembe
- qi : a jelenlegi szakasz pontjainak listája
- **Visited**: a már feldolgozott pixelek listája

Az eljárás fő célja az, hogy a megadott irányban találjon egy új pixel-párt, és hozzáfűzze azt a jelenlegi szakaszhoz (qi). Az eljárás mindaddig fut, amíg az irányba található egy új pixel, és ezzel a szakasz bővíthet.



3.1. ábra: Kvázi-egyenes szakaszok irány szerinti osztályozása [1]

3.2.2. Kód implementáció

Az eljárás a Freeman lánc kód irányába (n) és a második irányba (s) történő kiterjesztés szabályainak megfelelően működik. Az **EPSILON** érték meghatározza, hogy van-e második irány vagy sem. Ha az s értéke **EPSILON**, akkor csak az n irányba történik a kiterjesztés.

Az eljárás lényegét az adott irányba mutató új pixelek megtalálása és hozzáfűzése jelenti, ezzel kiterjesztve a jelenlegi szakaszt. Az eljárás a kiterjesztendő pixelt mindig hozzáadja a **qi** listához, és frissíti a **Visited** listát, hogy a már feldolgozott pixelek között szerepeljen.

```

def extend_segment(self, p: tuple, n: int, s: int, qi: list, Visited: list):
    if s == EPSILON:
        while True:
            p_n = self.get_unvisited_neighbour_by_dir(p, n, Visited)
            if p_n is None:
                return
            qi.append(p_n)
            Visited.append(p_n)
            p = p_n
    else:
        d = n
        while True:
            p_n = self.get_unvisited_neighbour_by_dir(p, n, Visited)
            p_s = None if s == EPSILON else
self.get_unvisited_neighbour_by_dir(p,s,Visited)
            if p_n is None and p_s is None:
                return
            if d == n:
                if p_n is not None:
                    qi.append(p_n)
                    Visited.append(p_n)
                    p = p_n
                    d = n
                    continue
                elif p_s is not None:
                    qi.append(p_s)
                    Visited.append(p_s)
                    p = p_s
                    d = s
                    continue
            else:
                return

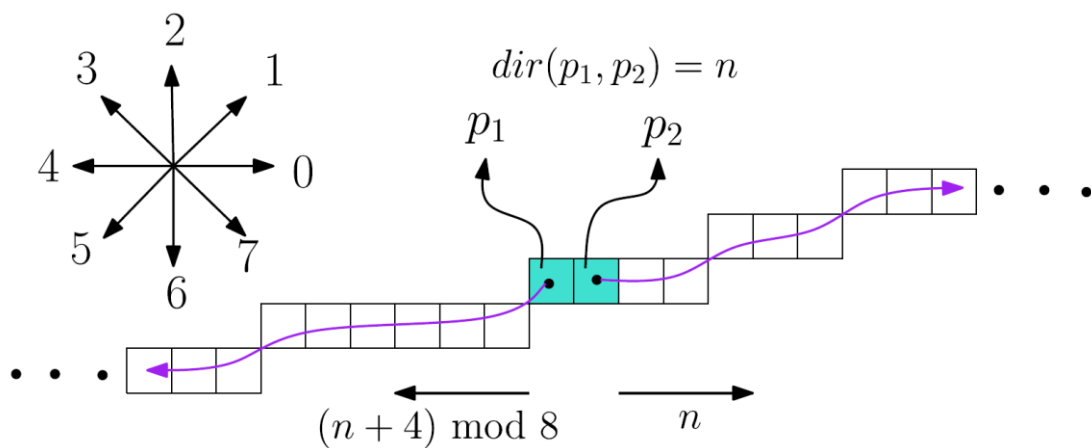
```

```

elif d == s:
    if p_n is not None:
        qi.append(p_n)
        Visited.append(p_n)
        p = p_n
        d = n
        continue
    else:
        return
else:
    return

```

3.3. Freeman lánc kód



3.2. ábra: Szegmens kiterjesztése adott pixelpártól (p_1, p_2) n és $(n + 4) \bmod 8$ irányba [1]

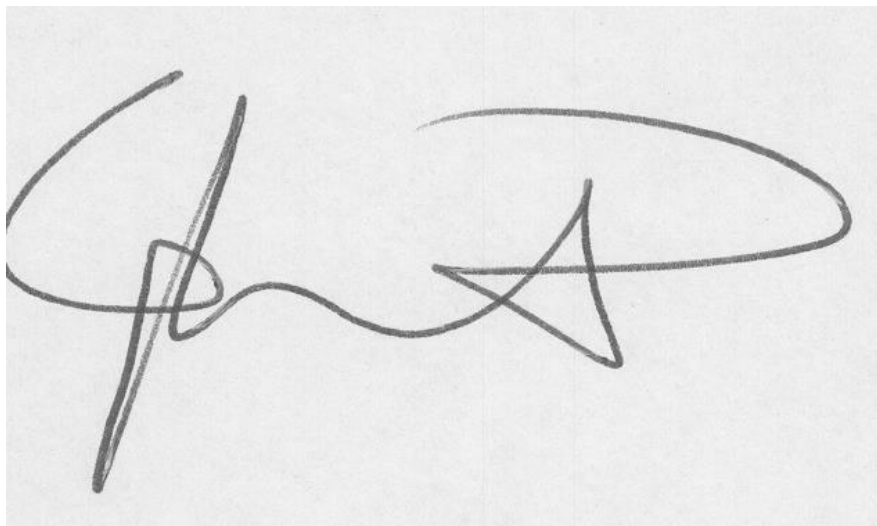
A Freeman lánc kód egy több irányú kódrendszer, melyet először 1961-ben publikált a nevéhez fűződő invarianciákra való tekintettel. A kvázi-egyenes szakaszok (DQSS) algoritmus és az azt támogató **EdgeProcessor** osztály implementációja során használt Freeman lánc kódok a képek éleknek való ábrázolására szolgálnak.

3.3.1. Freeman lánc kód implementáció

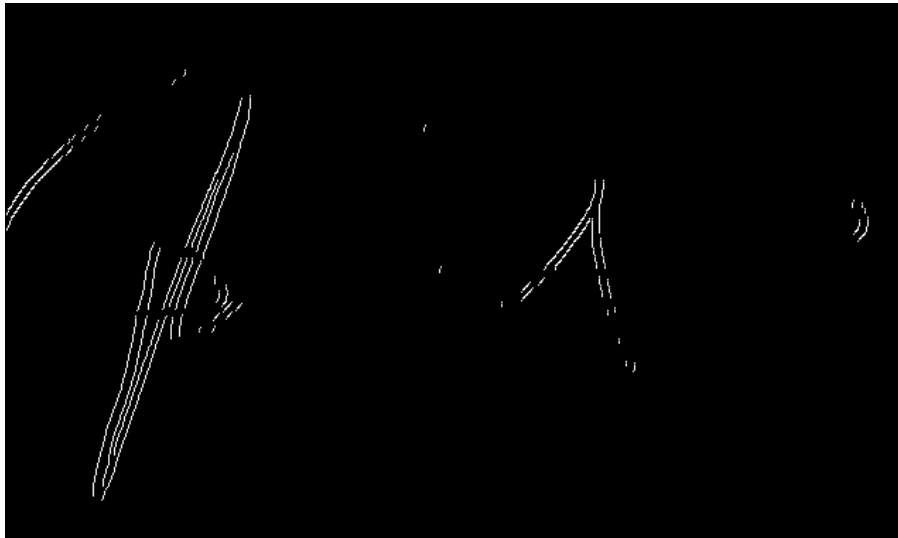
```
class EdgeProcessor:
    ...
    freeman_chain_vector_direction = {(0, 1): 0, (-1, 1): 1, (-1, 0): 2,
    (-1, -1): 3, (0, -1): 4, (1, -1): 5, (1, 0): 6, (1, 1): 7}
    freeman_chain_direction_vector = dict((v, k) for k,
    v in freeman_chain_vector_direction.items())
    def dir(self, p1: tuple, p2: tuple):
        dy: int = p2[0] - p1[0]
        dx: int = p2[1] - p1[1]
        if (dy, dx) in self.freeman_chain_vector_direction:
            return self.freeman_chain_vector_direction[(dy, dx)]
        return None
    ...
```

A `dir` metódus a Freeman lánc kód irányát határozza meg két pixel között. Az irányok meghatározása egyszerű és hatékony, kiválasztva a megfelelő irányt a Freeman lánc kód leírásához.

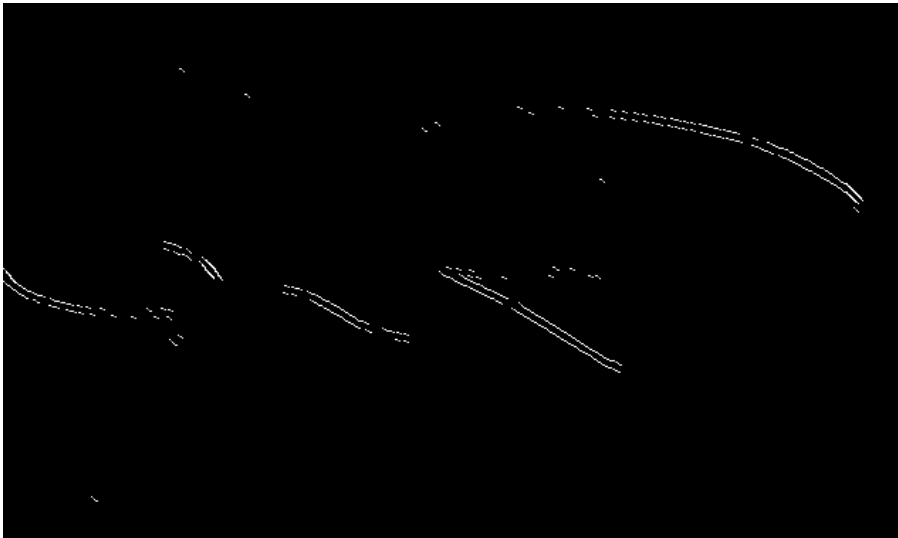
3.4. Példák a kvázi-egyenes szakaszok kinyerésére



3.3. ábra: egy kiválasztott felhasználó eredeti aláírásának képe



3.4. ábra: C6 osztályú szegmensek (küszöbérték: 4)



3.5. ábra: C11 osztályú szegmensek (küszöbérték: 4)

4. Aláírás minta jellemzők

A kvázi-egyenesekből származó jellemzők, mint például a szegmensek száma, a pixeltömeg és az átlagos élhossz, fontosak a vonalak struktúrájának és eloszlásának megértéséhez. Az ezekből származó adatok segítik a rendszert abban, hogy finom részleteket vegyen észre az aláírás minta vonalainak dinamikájáról és struktúrájáról.

Class (C_i)	n_i	p_i/P	p_i/n_i	c_p/P	m_{C_i}	r_{ij}/P	l_{R_j}
C_1	36	0.10	9.92	0.06	4	0.07	–
C_2	45	0.16	12.60	0.05	4	0.05	–
C_3	41	0.16	14.63	0.06	1	0.05	R_6
C_4	32	0.08	8.78	0.06	2	0.02	–
C_5	85	0.28	12.09	0.16	4	0.10	–
C_6	75	0.41	19.81	0.13	4	0.15	R_2, R_3, R_4, R_5
C_7	69	0.14	7.62	0.08	3	0.05	–
C_8	53	0.17	11.45	0.03	3	0.06	–
C_9	24	0.06	9.75	0.02	1	0.02	–
C_{10}	11	0.02	7.55	0.02	1	0.01	–
C_{11}	22	0.09	15.18	0.04	1	0.06	R_1
C_{12}	48	0.15	11.52	0.08	4	0.08	–

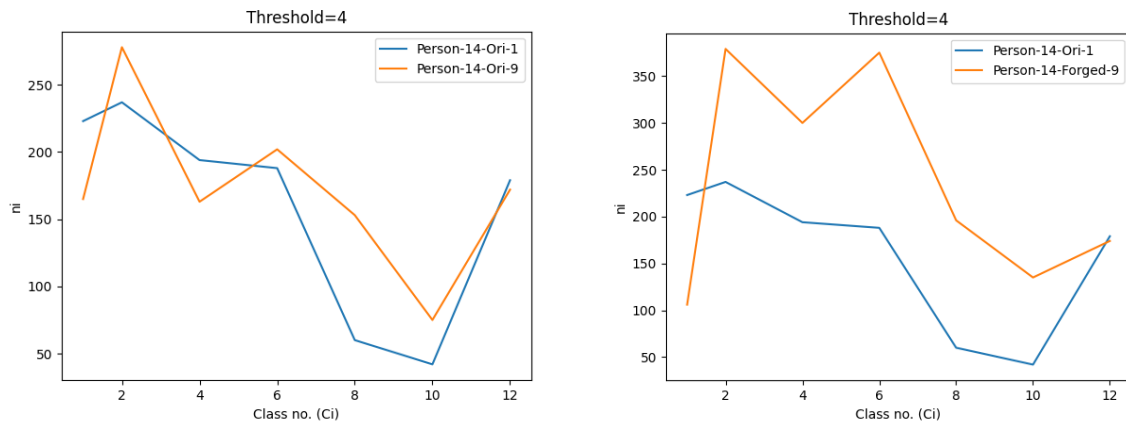
4.1. ábra: Adott aláírás mintából kinyert jellemzők halmaza

A közös pixelek száma a szomszédos osztályok közötti simaság mértékét mutatja. Ez a jellemző különösen érzékeny lehet a kifinomult hamisításokra, ahol az elkövetők igyekeznek a aláírásminta vonalait minél kevésbé torzítani. A közös pixelek számának hányadosa a teljes élpixelek számához (c_p/P) kulcsfontosságú az éles határok és az elmosódott területek megkülönböztetésében.

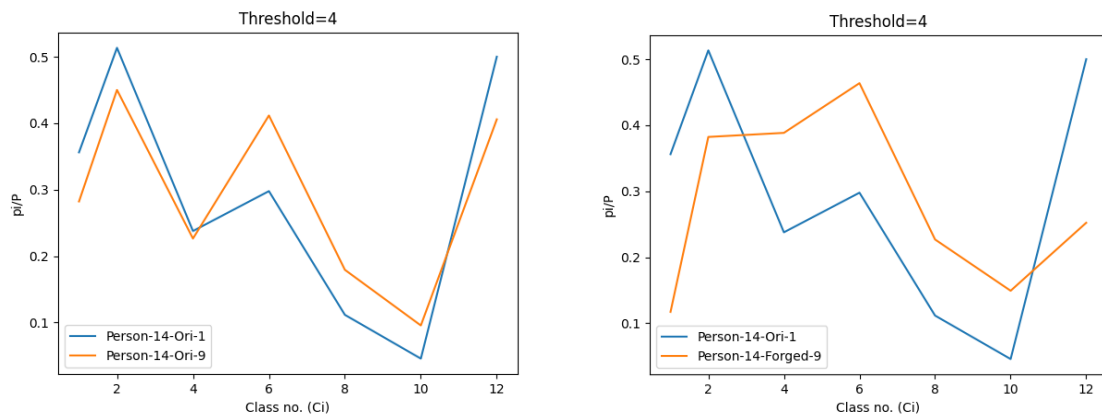
Jellemzők:

1. n_i : a képen található kvázi-egyenest darabok száma.
2. p_i/P : az él pixeleinek száma a teljes él pixeleinek számához viszonyítva.
3. p_i/n_i : az él pixeleinek számának az átlagos szakasz hosszához viszonyítva.
4. c_p/P : az azonos pixelek sűrűsége a szomszédos osztályok között.
5. m_{C_i} : az az osztály, amely a legnagyobb hozzájárulást nyújt egy adott régióban.
6. r_{ij}/P : az egyes osztályokhoz tartozó pixel-sűrűség a különböző régiókban.
7. l_{R_j} : az osztály, amely a legnagyobb hozzájárulást nyújtja egy adott régióban.

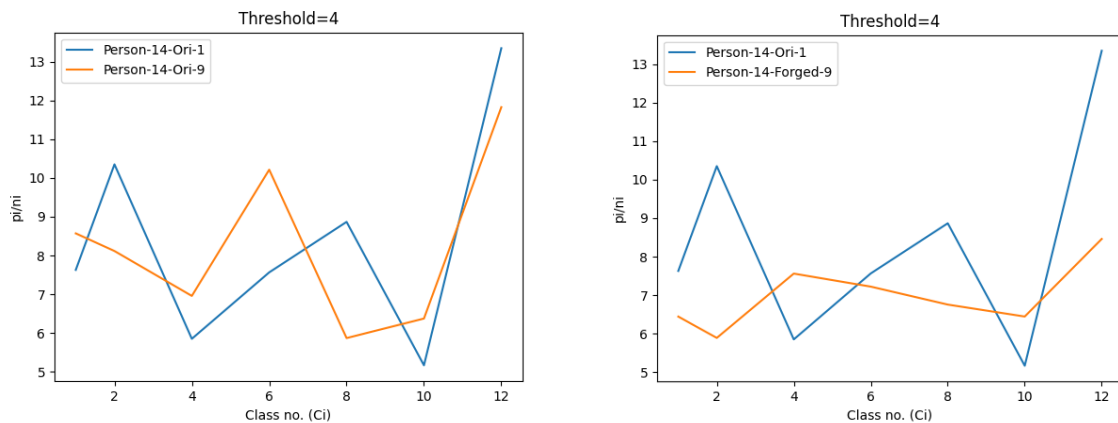
A régió-specifikus jellemzők, mint például a régióvezető és a pixelsűrűség, további információkat nyújtanak arról, hogy az adott osztályok milyen mértékben jelennek meg az egyes területeken. Ez segíti a rendszert a aláírásminta területén belüli változások pontosabb értelmezésében és az egyes régiók jellemzőinek azonosításában.



4.2. ábra: Aláírásminták összehasonlítása az ni jellemző szerint: az első képen két valódi, a másodikokon egy valódi és egy hamis aláírásminta látható



4.3. ábra: pi/P jellemző szerinti összehasonlítás



4.4. ábra: pi/ni jellemző szerinti összehasonlítás

5. Osztályozó algoritmusok

A megfelelő osztályozó algoritmus kiválasztása kulcsfontosságú, mivel az algoritmus hatékonyan kell kezelnie a komplex aláírásjellemzőket és a nem-lineáris összefüggéseket az adathalmazban. Különböző algoritmusok eltérő módon közelítik meg az adatokat, és az alkalmazott modell jelentősen befolyásolja a verifikációs teljesítményt. A helyesen kiválasztott osztályozó hozzájárulhat a pontos, megbízható és zajmentes aláírásverifikációhoz.

5.1. SVM (*Support Vector Machine*)

Az aláírási verifikációhoz a megfelelő kernel kiválasztása kiemelten fontos, mivel a kernel az SVM osztályozó hatékonyságát és pontosságát jelentősen befolyásolja. A kernel a bemeneti adatokat egy magasabb dimenziós térbe képes leképezni, ahol azok könnyebben elkülöníthetők és osztályozhatók. A jó kernel kiválasztása növeli az osztályozó rendszer képességeit az aláírások valósághű és pontos kategorizálásában.

A LINEÁR kernel használata kifejezetten azért volt előnyös az aláírásverifikációs probléma esetén, mert a LINEÁR kernel az alaphoz alacsony dimenziós térben is hatékonyan elvégzi az osztályozást. Az aláírások esetében az olyan tulajdonságok, mint a vonalak, élek és alakzatok, gyakran jól elkülöníthetők egy lineáris térben is. A LINEÁR kernel alkalmazása lehetőséget teremt arra, hogy a SVM egyszerűen és hatékonyan kezelje az ilyen típusú adatokat.

Az optimális küszöbérték (például $l = 4$) kiválasztása is kulcsfontosságú a jellemzők kinyerésében, mivel ez befolyásolja, hogy milyen kis szakaszok vesznek részt a jellemző kivonásában. Az optimális küszöbérték segíthet kizárni a zajt és csak a jelentős részeket tartani meg a jellemzővektorban.

```
#####
Linear SVM Classifier:

Legjobb parameterek: {'C': 0.0003000000000000003, 'kernel': 'linear'}
Eredmeny : 97.5%
Legjobb hiperparameterek melletti pontossaga:
precision    recall  f1-score   support

   0         1.00      0.50      0.67         2
   1         0.80      1.00      0.89         4

 accuracy          0.83         6
 macro avg          0.90      0.75      0.78         6
weighted avg          0.87      0.83      0.81         6

Pontossag: 0.8333333333333334
Tevesztesi matrix:
[[1 1]
 [0 4]]
Helyesen hamisnak címkézett aláírások száma(TN): 1
Tévesen valódinak címkézett aláírások száma(FP): 1
Tévesen hamisnak címkézett aláírások száma(FN): 0
Helyesen valódinak címkézett aláírások száma(TP): 4
Futasi ido: 0:00:12.663484
#####

Poly SVM Classifier:

Legjobb parameterek: {'C': 0.3562, 'kernel': 'poly'}
Eredmeny : 95.41666666666666%
Legjobb hiperparameterek melletti pontossaga:
precision    recall  f1-score   support

   0         0.50      1.00      0.67         2
   1         1.00      0.50      0.67         4

 accuracy          0.67         6
 macro avg          0.75      0.75      0.67         6
weighted avg          0.83      0.67      0.67         6

Pontossag: 0.6666666666666666
Tevesztesi matrix:
[[2 0]
 [2 2]]
Helyesen hamisnak címkézett aláírások száma(TN): 2
Tévesen valódinak címkézett aláírások száma(FP): 0
Tévesen hamisnak címkézett aláírások száma(FN): 2
Helyesen valódinak címkézett aláírások száma(TP): 2
Futasi ido: 0:00:13.148358
#####

Rbf SVM Classifier:

Legjobb parameterek: {'C': 0.0001, 'gamma': 0.060000000000000005, 'kernel': 'rbf'}
Eredmeny : 50.83333333333333%
Legjobb hiperparameterek melletti pontossaga:
precision    recall  f1-score   support

   0         1.00      1.00      1.00         3
   1         1.00      1.00      1.00         3

 accuracy          1.00         6
 macro avg          1.00      1.00      1.00         6
weighted avg          1.00      1.00      1.00         6

Pontossag: 1.0
Tevesztesi matrix:
[[3 0]
 [0 3]]
Helyesen hamisnak címkézett aláírások száma(TN): 3
Tévesen valódinak címkézett aláírások száma(FP): 0
Tévesen hamisnak címkézett aláírások száma(FN): 0
Helyesen valódinak címkézett aláírások száma(TP): 3
Futasi ido: 0:04:29.312328
#####
```

```
#####
Sigmoid SVM Classifier:

Legjobb parameterek: {'C': 0.9966, 'kernel': 'sigmoid'}
Eredmeny : 64.58333333333334%
Legjobb hiperparameterek melletti pontossaga:
      precision    recall  f1-score   support

         0         0.67        1.00        0.80         2
         1         1.00        0.75        0.86         4

 accuracy          0.83
 macro avg          0.83        0.88        0.83         6
weighted avg          0.89        0.83        0.84         6

Pontossag: 0.8333333333333334
Tevesztesi matrix:
[[2 0]
 [1 3]]
Helyesen hamisnak címkézett aláírások száma(TN): 2
Tévesen valódinak címkézett aláírások száma(FP): 0
Tévesen hamisnak címkézett aláírások száma(FN): 1
Helyesen valódinak címkézett aláírások száma(TP): 3
Futasi ido: 0:00:13.321760
#####
```

5.1. ábra: SVM kernelek összehasonlítása (küszöbérték=4): LINEAR, POLY, RBF, SIGMOID

5.2. Véletlen erdők (Random forests)

A véletlen erdők osztályozó hatékony az offline verifikációban, mert rendkívül sokféle döntési fát hoz létre és azokat kombinálja. Ennek eredményeként képes kezelni a nagy dimenziójú és komplex aláírásjellemzőket, amelyek gyakran jelen vannak az ilyen típusú feladatokban. Ezen túlmenően, a véletlen erdők adaptív és erős az overfitting ellen, így a modell jól teljesít a változatos és heterogén adathalmazokon is, amelyek a valóságos aláírások különbözőségeit tükrözhetik. Emellett a random forests nem igényel túl sok beállítást, és elég ellenálló a zajhoz, amely előnyös lehet a gyakorlatban, ahol az aláírások stílusának és minőségének változhatósága gyakori. Ezen tulajdonságok révén a random forests egy erős és megbízható alternatíva lehet az SVM-mel együtt az offline aláírásverifikációban.

6. Eredmények kiértékelése

A vizsgálatok során a CEDAR [9] adatkészletet alkalmaztuk. Ez az adathalmaz sokszínű és reprezentatív, ami lehetővé teszi az algoritmusok általános teljesítményének értékelését. A CEDAR (Center of Excellence for Document Analysis and Recognition) [9] egy kiváló minőségű offline aláírás-adathalmaz, amely 55 különböző személy 24 valódi és hamisított aláírásával rendelkezik. Minden személyhez 16 teszt tartozik, amelyeket 10-szeres keresztvalidáció során értékeltünk ki. Az adatkészlet sokfélesége és a hamisítások

változatossága ideálissá teszi a kísérletek végrehajtását és a rendszer általános teljesítményének vizsgálatát. Ezen adatkészlet eredményei alapján értékeljük a kidolgozott algoritmusok teljesítményét a különböző körülmények között. A teszteredmények részletes elemzése és értékelése a továbbiakban kerül bemutatásra a fejezet folytatásában.

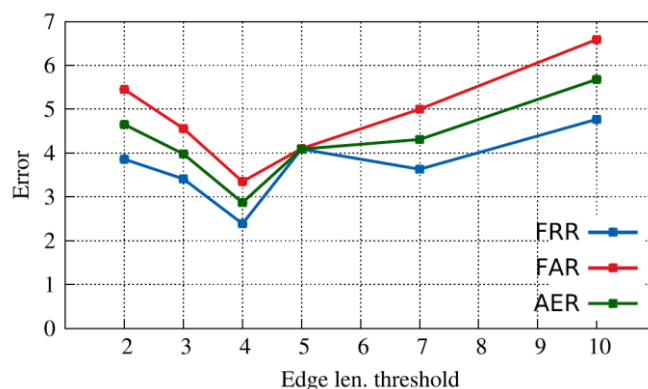
6.1. Metrikák

A kiértékelés során bevezettünk különböző metrikákat, melyek a rendszer teljesítményének értékelésére szolgálnak. Ezek a metrikák segítenek megérteni és értékelni, hogy a kialakított aláírásverifikációs rendszer milyen hatékonysággal képes megkülönböztetni a valódi és a hamis aláírásokat.

- **TN (True Negative):** Azoknak az eseteknek a száma, amikor a rendszer helyesen ismeri fel a hamis aláírást és helyesen kategorizálja azt, mint hamis.
- **FP (False Positive):** Azoknak az eseteknek a száma, amikor a rendszer tévesen azonosítja a hamis aláírást, mint valós.
- **FN (False Negative):** Azoknak az eseteknek a száma, amikor a rendszer tévesen kategorizálja a valós aláírást hamisnak.
- **TP (True Positive):** Azoknak az eseteknek a száma, amikor a rendszer helyesen ismeri fel a valós aláírást és helyesen kategorizálja azt, mint valós.

Ezen értékekből származtatott metrikák:

- **FRR (False Rejection Rate):** A valódi aláírások olyan aránya, amelyeket tévesen visszautasítottak az összes valódi aláíráshoz képest. Kiszámítása: $FN / (FN + TP)$.
- **FAR (False Acceptance Rate):** A hamis aláírások olyan aránya, amelyeket tévesen elfogadtak az összes hamis aláíráshoz képest. Kiszámítása: $FP / (FP + TN)$.
- **AER (Average Error Rate):** Az átlagos hibaráta, ami a FRR és a FAR súlyozott átlaga. Kiszámítása: $(FRR + FAR) / 2$.
- **EER (Equal Error Rate):** Az a pont a ROC görbén, ahol a FRR és a FAR értékek egyenlőek. Az EER az optimális pont, ahol a rendszer azonosan kezeli a hamis pozitív és hamis negatív eredményeket.



6.1. ábra: Hibaráta (FRR, FAR és AER) értékek különböző élhossz küszöbértékek (l) szerint [1]

6.2. Aláírás konzisztencia

Az FRR, FAR és AER értékek összehasonlítása az egyes személyek között fontos információkat szolgáltat a rendszer teljesítményéről és a felhasználók aláírásának konzisztenciájáról.

Az alábbiakban bemutatunk néhány lehetséges következtetést:

- **Egyének közötti különbségek:**

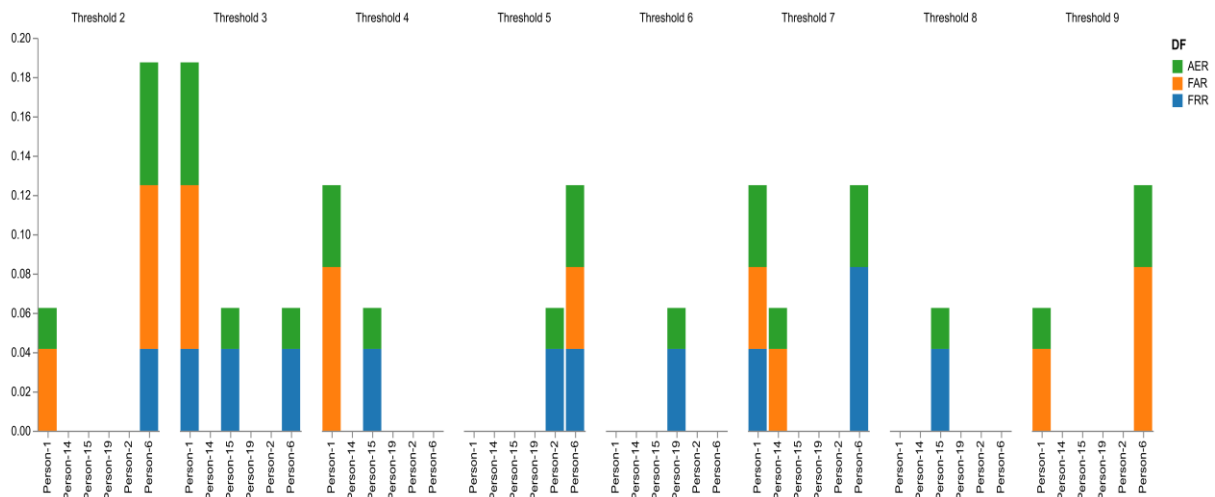
Ha az FRR, FAR és AER értékek jelentős különbségeket mutatnak a személyek között, akkor az azt jelezheti, hogy bizonyos személyek aláírásai könnyebben hamisíthatók vagy nehezebben azonosíthatók. Például, ha egy személy esetében magas az FRR, akkor az azt jelenti, hogy az illető valódi aláírásait a rendszer gyakran tévesen hamisnak minősíti, ami jelezheti, hogy az ő aláírásai nagyon változatosak. Ha viszont magas a FAR, az azt jelenti, hogy az illető hamisított aláírásait a rendszer gyakran tévesen valódinak minősíti, ami jelezheti, hogy az ő aláírásai könnyen hamisíthatók.

- **Rendszer megbízhatósága:**

Az összes személy AER értékeinek átlagolása általános képet adhat a rendszer megbízhatóságáról. Ha az átlagos AER érték alacsony, akkor a rendszer összességében jól teljesít. Magas átlagos AER érték esetén viszont a rendszer pontosságának javítása szükséges lehet.

- **Személyspecifikus finomhangolás szükségessége:**

Ha egyes személyek esetében kiugróan magas FRR vagy FAR értékeket tapasztalunk, akkor érdemes lehet a rendszer finomhangolását elvégezni az adott személyek aláírásaihoz igazítva. Például, az adott személyek aláírásainak speciális jellemzőit figyelembe vevő új modelleket vagy külön paraméterbeállításokat alkalmazni.



6.1. ábra: Különböző személyek aláírásának hibaráta (FRR, FAR és AER) a konzisztencia felderítéséhez

Összefoglalás

Az alkalmazás célja az offline aláírás verifikáció hatékonyságának növelése volt. Egy olyan rendszer fejlesztése, amely képes pontosan megkülönböztetni a valódi aláírásokat a hamisítottaktól, minimális hibaarányokkal. A rendszer több támogatott osztályozási eljárással (SVM) és egyéb (véletlen erdő (RandomForest)) algoritmusokkal végzett verifikáció hatékonyságát vizsgálta.

Az implementáció során először adatokat gyűjtöttünk és előkészítettük a CEDAR [9] adatbázisokból. Ez az adatbázis tartalmazott több személy valódi és hamisított aláírását. Majd ezeket az aláírásokat több jellemző alapján osztályoztuk az aláírások későbbi elemzéséhez. Ezután osztályozási eljárások (SVM) és véletlen erdő (RandomForest) algoritmusokkal modelleket tanítottunk. Az SVM esetében a LINEAR kernel bizonyult a legjobbnak, és a paraméterek optimalizálásához keresztvalidációt alkalmaztunk.

Implementáció

Az alkalmazás implementálása Python 3 [2] nyelven történt, amely egy rendkívül sokoldalú és népszerű nyelv a gépi tanulási és adatfeldolgozási feladatokhoz.

Ezenkívül a projekt során több Python [2] könyvtárat használtam, amelyek több hasznos eszközt biztosítottak több rész probléma megvalósításához:

1. Az aláírások adatainak gyűjtése és előkészítése a Numpy [3] és Pandas [4] könyvtárak segítségével történt. Az adatok megfelelő formátumba hozása, tisztítása és normalizálása volt az első lépés.
2. Az aláírásokból különböző jellemzők (pl. n_i , p_i/P , p_i/n_i , cp/P , mCi) kinyerése és előkészítése a gépi tanulási modellek számára. Ezenkívül az OpenCV [8] könyvtár segítségével az aláírások képfeldolgozása és az élek detektálása is megtörtént.
3. A Scikit-learn [5] könyvtár segítségével különböző gépi tanulási modelleket (SVM, RandomForest) betanítottam a rendelkezésre álló adatok alapján. A modellek betanításához különböző paramétereket és keresztvalidációs technikákat alkalmaztam.
4. A betanított modellek teljesítményének kiértékelése különböző metrikák (FRR, FAR, AER) segítségével történt. A Matplotlib [6] könyvtár segítségével vizualizáltam az eredményeket, hogy könnyebben átláthatóak legyenek.
5. A Concurrent [7] könyvtár segítségével optimalizáltam a folyamatokat, hogy a nagy mennyiségű adat feldolgozása gyorsabb és hatékonyabb legyen.

Eredmények és tapasztalatok

Az alkalmazás sikeresen implementálva lett, és a betanított modellek helyes eredményeket mutattak az aláírások verifikációjában. Az SVM és a RandomForest algoritmusok is hatékonyan működtek, és a különböző metrikák alapján jól teljesítettek. A fejlesztési folyamat során szerzett tapasztalatok alapján az alkalmazás továbbfejlesztése is lehetőséget biztosít a még pontosabb és megbízhatóbb aláírás verifikáció elérésére.

További fejlesztési lehetőségek

Az ügyfél új aláírásokat adhat meg, amelyeket a rendszernek figyelembe kell vennie az elemzése során. Ehhez szükség van egy automatizált folyamatra, amely képes az új adatok integrálására és a modellek újra tanítására úgy, hogy a lehető legkevesebb erőforrást igényelje. Ezenkívül biztosítani kell, hogy a rendszer képes legyen az új frissített modelleket elmenteni és könnyen visszaállítani, ha szükséges. Az újra tanítási folyamat során fontos

figyelembe venni a skálázhatóságot is, hogy a rendszer növekedésével is hatékonyan működjön.

A másik fejlesztési lehetőség az időbélyegek rögzítése az aláírásokhoz. Az időbélyeg hozzátartozása lehetővé tenné a rendszer számára, hogy csak adott időintervallum alatt készült aláírásokat használja fel a modell tanulási folyamatában. Ez biztosítaná, hogy a modell mindig naprakész maradjon, és a legfrissebb adatok alapján működjön, ami különösen fontos lehet a dinamikusan változó aláírások esetében. Az időbélyegek kezelése és figyelembevétele a tanulási folyamat során lehetővé teszi a rendszer számára, hogy a régebbi, esetleg már nem releváns adatokat kizárja, ezáltal javítva a modell pontosságát és relevanciáját. Az időbélyeg kezeléséhez szükség van egy olyan adatbázisra, amely képes tárolni és kezelni az aláírásokhoz kapcsolódó időinformációkat, valamint egy algoritmusra, amely képes dinamikusan kiválasztani a megfelelő tanulási adatokat az időbélyegek alapján.

Irodalomjegyzék

- [1] M. D. Ajj, S. Pratihar, S. R. Nayak, T. Hanne, D. S. Roy: Off-line signature verification using elementary combinations of directional codes from boundary pixels. Neural Computing and Applications, 2021
- [2] Python: high-level, general-purpose programming language. <https://www.python.org>. Version: 3.8.3
- [3] NumPy: an open source project that enables numerical computing with Python. <https://numpy.org>. Version: 1.21.2
- [4] pandas: an open source data analysis and manipulation tool. <https://pandas.pydata.org>. Version: 1.5.3
- [5] scikit-learn: a free software machine learning library for the Python programming language. <https://scikit-learn.org>. Version: 1.2.1
- [6] Matplotlib: Visualization with Python. Version: 3.8.2
- [7] concurrent.futures: a built-in module provides a high-level interface for asynchronously executing callables. <https://docs.python.org/3/library/concurrent.futures.html>
- [8] OpenCV: a library of programming functions mainly for real-time computer vision. <https://opencv.org/> Version: 4.8.1.78
- [9] CEDAR (Center of Excellence for Document Analysis and Recognition) Dataset. <http://www.cedar.buffalo.edu/NIJ/publications.html>. Last accessed: 2023-02-12


Nyilatkozat

Alulírott Huri Ferenc Tamás szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Képfeldolgozás és Számítógépes Grafika Tanszékén készítettem, programtervező informatikus BSc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat / diplomamunkámat a Szegedi Tudományegyetem Diplomamunka Repozitóriumban tárolja.

Dátum: 2023. 12. 16.



Aláírás

Köszönetnyilvánítás

Szeretném kifejezni mély hálámat mindazoknak, akik segítettek és támogattak a szakdolgozatom elkészítése során.

Először is, szeretném megköszönni konzulensemnek, dr. Németh Gábor irányítását, szakértelmét és türelmét. Köszönettel tartozom Önnek az értékes tanácsokért, melyek segítették a kutatásom fejlődését, és hozzájárultak a dolgozatom minőségéhez.

Hálás köszönetemet fejezem ki családomnak és barátaimnak is, akik támogattak és motiváltak az egész folyamat során. Az őszinte biztatásuk és megértésük nélkül ez a munka nem lett volna ugyanolyan.

Köszönet mindenkinek, aki hozzájárult ahhoz, hogy ez a szakdolgozat valósággá válhasson.

Tisztelettel,

Huri Ferenc Tamás