

**Neumann János Informatikai Kar
Óbudai Egyetem**

Dr. Tadiy Vladimír László

Képfeldolgozási és alakfelismerési módszerek

Budapest, 2026.

ISBN 978-963-449-426-3

Kiadó: Óbudai Egyetem, Budapest, Bécsi út 96/B, 1034 Hungary

Tartalomjegyzék

1. Bevezető.....	1
2. A digitális ikerpár fogalma és létrehozása.....	2
2.1. A digitális ikerpár fogalma	2
2.2. A digitális ikerpár létrehozása.....	3
2.3. A digitális-fizikai rendszerek elemei.....	6
3. A digitális képfeldolgozás alapelvei.....	8
3.1. Bevezetés a digitális képfeldolgozásba.....	9
3.2. A kép matematikai és grafikus ábrázolása	14
3.3. A képfeldolgozó rendszerek.....	15
3.4. A 2D jelek és rendszerek.....	17
4. A digitális képek szűrése és szegmentálása	19
4.1. A képszűrés alapelvei.....	20
4.2. A képszegmentáció alapjai	26
5. Az alakfelismerési módszerek	46
5.1. A strukturált módszerek.....	50
5.2. Az elméleti, döntéshozáson alapuló módszerek	55
6. A fuzzy logika..	109
6.1. A fuzzy logika definíciói.....	112
6.2. A tagsági függvények	116
6.3. A fuzzy rendszerek.....	120
6.4. Az ANFIS..	140
6.5. A fuzzy klaszterizáció	144
7. Következtetés...	150
8. Irodalom.....	151

Köszönetnyilvánítás

Ezúton szeretném megköszönni Dr. Odry Péternek a tananyag elkészítésében nyújtott segítségét és támogatását, valamint a tudományos kutatómunkám során nyújtott mentorálását, motivációját és szakmai irányítást.

Végül, hálás köszönettel tartozom Családomnak és Szüleimnek.

Temerin, Vajdaság, 2026.

Dr. Tadity Vladimir László

Előszó

A tananyag elején röviden ismertetve lesz az ikertechnológia fogalma és a digitális ikerpárok létrehozása, valamint azok kapcsolata a képfeldolgozással és alakfelismeréssel mivel ezek az eljárások biztosítják a valós tárgyak pontos azonosítását és digitális leképezését.

Ezután következnek a digitális képfeldolgozás alapelvei, ahol röviden ismertetésre kerülnek a képfeldolgozási módszerek típusai, a digitális kép létrehozása annak alapvető tulajdonságaival. Be lesznek mutatva röviden a 2D jelek és rendszerek is.

Ezt követi a képszűrési és képszegmentálási rendszerek részletes leírása, a hozzájuk tartozó matematikai háttérrel és példákkal.

Végül következik az alakfelismerésről szóló fejezet, ahol be lesz mutatva a különböző alakfelismerési eljárások sorozata, a hozzájuk tartozó matematikai háttérrel és példákkal.

Budapest, 2026.

Szerző

1. Bevezető

A digitális ikerpár technológia koncepciója 2002 óta létezik és először az űrkutatás, valamint a gyártás és a termék életciklus-menedzsment területén használták. A digitális ikerpár lényegében kapcsolatot képez egy valós objektum és annak digitális ábrázolása között, folyamatosan felhasználva az érzékelők-szenzorok adatait. Minden adat egy fizikai objektumon, tárgyon elhelyezett érzékelőkből származik, az adatokat a virtuális objektum reprezentációjának létrehozására használják. A digitális ábrázolást később vizualizációra, modellezésre, elemzésre, szimulációra és további tervezésre használják. Ezek az adatok kihatnak a döntések visszacsatolási ciklusára és a munkafolyamatok változására, amelyek befolyásolják a valós objektum, vagy rendszer vezérlési folyamatait.

Manapság a digitális ikerpár technológia a robotikában is fontos szerepet kapott a különböző ipari robotok tervezésében, fejlesztésében és gyártásában.

A digitális ikerpár technológia fontos összetevője a digitális képfeldolgozás, melynek segítségével történik a valós termékek digitális úton való vizualizációja.

Ez a tananyag a digitális ikerpár koncepciójának témájában felhasználható képfeldolgozási és képelemzési eljárások áttekintésével és elemzésével foglalkozik.

A tananyagban a következő témák lesznek áttekintve:

1. a digitális ikerpár megoldásoknál az objektumok felismerésére (alakfelismerésre) és osztályozására használatos képfeldolgozási és képelemzési eljárási megoldások áttekintése,
2. a képfeldolgozás matematikai hátterének áttekintése,
3. a képelemzési alkalmazások szoftver megoldásainak rövid áttekintése.

A tananyag felépítése a következő: a tananyag elején a digitális ikerpár technológia fogalma és alapelvei lesznek ismertetve. Ezt követi majd a digitális képfeldolgozási eljárások részletes ismertetése a matematikai leírásokkal, majd végül röviden ismertetve lesznek a képfeldolgozási szoftver megoldások. A tananyag egy rövid következtetéssel és a felhasznált irodalommal fejeződik.

2. A digitális ikerpár fogalma és létrehozása

2.1. A digitális ikerpár fogalma

A digitális ikerpárok a fizikai termékek, ebben a konkrét esetben a robotok virtuális másolatai, amelyek valós időben modellezik a robot valós viselkedését és a lehetséges hibák, problémák felderítésére szolgálnak. Különböző elemzéseket végeznek, tesztelik az új funkciókat és szimulálják azokat a lehetséges forgatókönyveket, amelyeket drága, vagy akár lehetetlen megvalósítani egy fizikai terméken, roboton. A digitális környezetben található szenzorok és más technológiák segítségével a digitális ikerpárok megjelenítik az objektumok, tárgyak elemeit, összetevőit és dinamikáját- összekapcsolt hálózatokon keresztül, amely lehetővé teszi az objektum állapotának, környezetének, valamint az életciklus során történő változásainak nyomon követését. Az érzékelő szerepe, hogy helyes információt szerezzen az eszköz, robot és a környezet aktuális állapotáról, amelyet egy számítógép tárol. A fizikai adatforrások mellett a számítógépek digitális környezetben létrehozott adatokat is tárolnak, például tervezési specifikációkat, CAD (computer-aided design) modelleket, alkatrész- és anyaglistákat, számítási eredményeket, szimulációkat stb. Ezeknek az adatoknak az érzékelőkön keresztül gyűjtött adatokkal történő összevonása teljes képet nyújt a robotos rendszer működéséről és teljesítményéről, lehetővé téve a valós időben történő működést az alkalmazás és a karbantartás fázisában, de a robot jobb alkalmazkodását is a tényleges működési körülményekhez.

A digitális ikerpár bevezetése lényegesen megváltoztatja a robotok és más termékek felépítésének, szerkezetének és az általuk átértékelte folyamatoknak a termékek (robotok) élettartamára gyakorolt hatását. A terméktervezési szakaszban fontos figyelembe venni az összes szükséges fizikai elemet és a megfelelő számítógépes támogatást annak érdekében, hogy az elemeket funkcionális rendszerbe lehessen összekapcsolni. Mint ismeretes, a moduláris robotok esetében az egyik fő jellemző a

modularitás, amely magában foglalja annak szükségességét, hogy a rendszerek képesek legyenek számos különböző funkció egyidőbeni végrehajtására, ami növeli a rendszer összetettségét. Például egy robotcella szerkezet esetében lényeges az egyes elemek-alkatrészek közös hatása a robotcella fizikai kitartására és a betartandó tűréshatárookra, mint egységes szerkezetre. A robot állapotával kapcsolatos tényleges adatok kíséréssel valósabb betekintést lehet nyerni a tényleges terhelésbe és kopásba, amely lehetőséget nyújt a szükséges biztonsági tényezők csökkentésére. Így lehetőség lesz a működés finomabb beállítására és a gyártási költségek csökkentésére. Az életciklusuk alatt a robotok különféle műveleteket hajtanak végre, amelyek során különböző munkakörülményeken mennek keresztül, alkatrészeik sok változást tapasztalnak, pl. elkopnak, megsérülnek stb. A digitális ikerpárok segítségével kísérni lehet a robotok élettartama során végbemenő folyamatokat és paramétereket, amelyek lehetővé teszik a robot jobb megtervezését és fejlesztését.

A digitális ikerpár használatának végső célja az összköltségek csökkentése, az elemek hibás működésének időben történő észrevétele, amellyel optimalizálható a karbantartás és lehetőség nyílik új termékek és szolgáltatások kifejlesztésére.

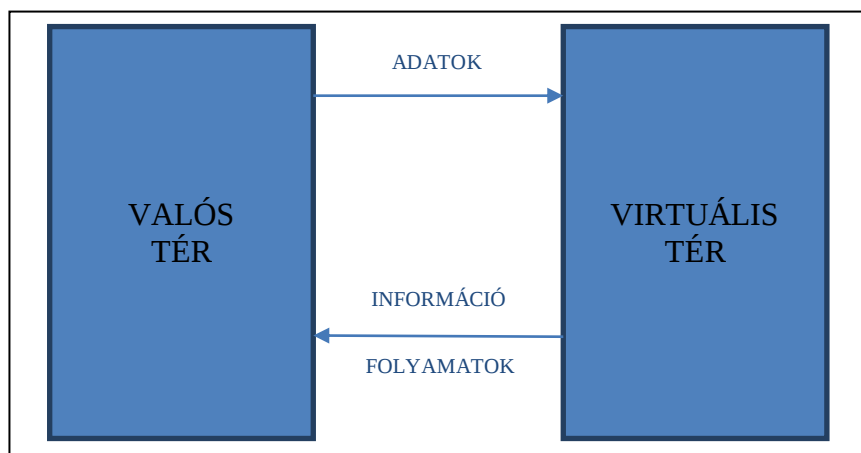
2.2. A digitális ikerpár létrehozása

Mint ahogyan említésre került a bevezetőben, a digitális ikerpár kezdetei egészen az űrkutatás elejére nyúlnak vissza, pontosabban az Apollo programig

A digitális ikerpár fogalmi modellje nagyban hasonlít, de egyben kibővíti az Apollo program „ikreit” a következő elvekkel:

1. digitalizálja az ikermodellt, és a digitális kifejezés segítségével azonos tartalmú és természetű virtuális terméket készít, mint a megjelenített eredeti termék,
2. bevezeti a virtuális teret és létrehozza a virtuális tér és a valós tér közötti összefüggést, hogy az adatok és információk kölcsönösen cserélhetők legyenek,

3. vizuálisan tükrözi a valóság integrálásának koncepcióját a képzeletbeli, imagináris modellel, valamint a valóságot is vezérli a képzeletbeli modellel,
4. a koncepció kiterjesztése és bővítése alkalmazható bármilyen termék megfelelő virtuális modelljének, digitális ikerpárjának létrehozására.



2.1. ábra: A digitális ikerpár elvi modellje

Azonban ez az elv és a modell 2003 körül nem keltett különösebb figyelmet a tudósok körében. Ennek fő okai a következők voltak:

1. korlátozott technikai eszközök álltak rendelkezésre, különösen nehéz volt megvalósítani a termelési adatok valós idejű begyűjtését, mert az akkori áramkörök és technikai megoldások nem voltak elég fejlettek,
2. a fizikai termékek digitális leírásához nem létezett megfelelő szoftveres és hardveres támogatás,
3. akkoriban még nehéz volt megvalósítani nagy mennyiségű adat valós idejű feldolgozását a rendelkezésre álló számítógépek teljesítményével és a mobil kommunikációs technológia sem volt eléggé fejlett ilyen komoly feladatok ellátására a valós idejű adatátvitelneheztségei mellett.

2011 után a digitális ikerpár új fejlesztési lehetőséget kínált. A digitális ikerpár alkalmazásának célja, az Amerikai Légierő Kutatólaboratóriuma által 2011-ben továbbfejlesztett javaslata szerint az, hogy megoldják a repülőgépek karbantartását és élettartamának előrejelzését az összetett szolgáltatási környezetben. A digitális ikerpárnak két rendkívüli hasznos tulajdonsága van:

1. magában foglalja az összes geometriai adatot,
2. tartalmazza az összes anyagi és a mikroszerkezeti adatokat.

A 2011 utáni években számos ország, intézet és cég a digitális ikerpárok fejlesztésével foglalkozott. Ennek eredményeként a digitális ikerpár a világ 10 legjobb intelligens gyártási technológiái közé lett sorolva.

Ebből látható, hogy a digitális ikerpárok rohamosan fejlődtek mind elméleti téren, úgy az alkalmazások terén is az elmúlt években. Ugyanakkor az alkalmazások lehetősége fokozatosan eltolódott a termékek tervezési szakaszától a termékek gyártási szakaszáig, valamint az üzemeltetési és szolgáltatási szakaszig is, amely nagy figyelmet keltett a tudósok és a vállalkozók körében. Ennek fő okai a következők:

1. a digitális technológia fejlődése és széleskörű alkalmazása lehetővé teszi a termék pontos fizikai leírását a termék életciklusának különböző szakaszaiban, többféle digitális módszer alkalmazásával
2. az információs és kommunikációs technológiák új generációinak gyors fejlődése és népszerűsítése megnövelte a termékek megbízhatóságát, lehetővé tette a dinamikus valós idejű adatgyűjtést és a lehetséges hibák megbízható előrejelzését-predikcióját.

2.3. A digitális-fizikai rendszerek elemei

A digitális ikrek számos, kumulatív, valós adatokon alapulnak, amelyeket különböző forrásokból gyűjtöttek össze, valós időben mérve. Ezek az adatok digitális formában hozzák létre a termék, vagy folyamat dinamikus modelljét, amely jobb betekintést nyújthat a rendszer teljesítményébe és ezáltal ösztönözheti a fizikai rendszeren végrehajtandó változtatásokat annak javítása érdekében. Az elmúlt években a számítástechnika gyors fejlődésének köszönhetően a számítógépes rendszerek teljesítménye hatékonyabb lett, miközben csökkentek a költségek és rövid idő alatt lehetővé vált összetett algoritmusok nagy mennyiségű adattal történő elvégzése, futtatása különböző elemzésekhez és szimulációkhoz.

Ha például figyelünk egy digitális-fizikai rendszert, amely a fizikai részből és a digitális ikerpárból áll, akkor itt megtalálhatóak azok a legfontosabb összetevők, amelyeket minden rendszernek tartalmaznia kell ahhoz, hogy elkészíthető legyen annak digitális másolata. A fizikai környezet részeként a terméket fel kell szerelni olyan érzékelőkkel, amelyek jeleket küldenek az eszköz és a környezet aktuális állapotáról, valamint olyan aktuátorral kell őket felszerelni, amellyel el lehet érni a kívánt reakciókat. Ami összeköti a fizikai és a digitális részt az nem más, mint az adat. Az érzékelők érzékelik és „kiolvassák” ezeket az adatokat a fizikai környezetből és egy számítógépre küldik azokat, ahol tárolják őket. Mint említettük a bevezetőben, a fizikai adatforrások mellett a számítógépek más digitális környezetben létrehozott adatokat is tárolnak. A tárolt adatokat át kell alakítani olyan formába, amely a különböző forrásokból származó adatokkal együtt használható. Csak ezután végezhetőek el a különféle adatelemzések, algoritmikus szimulációk és vizualizációk, amelyek a digitális ikrek segítségével tisztább képet alkotnak és jobb betekintést nyújtanak a rendszer működésébe és viselkedésébe. Az elvégzett elemzések eredményei jelzik azokat a lehetséges szabálytalanságokat vagy optimalizálási lehetőségeket, amelyek alapján korrigálni lehet a rendszer viselkedését aktuátorok segítségével.

A digitális ikerpárak létrehozása két lépésre bontható sorolható:

1. a digitális ikerpár életciklusára vonatkozó információk meghatározásának folyamata és a követelmények meghatározása

2. olyan technológia fejlesztése, amely lehetővé teszi az érzékelők által érzékelt adatok és más számítógépes adatok gyűjtését, tárolását és felhasználását úgy, hogy az integrálja a fizikai terméket és annak digitális másolatát.

Az első lépésben meg kell határozni, hogy mit kell elérni egy termék digitális ikerpárjának létrehozásával. A műveletek elvégzéséhez és a meglévő termékek hatékonyságának növeléséhez szükséges költségek és/vagy idő optimalizálását említik leggyakrabban, mint kívánt célt. Ebben a szakaszban az üzleti folyamatokat, az alkalmazást, az információkat, a fizikai összetevőket és az eljárásbanrésztvevő személyeket érintő folyamatok modellezésére szolgáló szabványos módszereket alkalmaznak, majd diagramokat készítenek az alkalmazásokkal, az adatkövetelményekkel és a szükséges érzékelők típusaival együtt.

A második lépés a rendszerarchitektúra kialakítására szolgál, amely lehetővé teszi az összes előre meghatározott folyamat, beleértve a fizikai összetevők kiválasztását és a sikeres gyűjtés, tárolás, átalakítás és adatelemzés technikai támogatását. A digitális ikerpár felépítését a következő hat lépés tükrözi legjobban: kreáció, kommunikáció, gyűjtés, elemzés, áttekintés, cselekvés. A kreáció egy olyan rendszer modellezésére vonatkozik, amely érzékelők segítségével képes adatokat gyűjteni a működési paraméterekről. Az érzékelők által gyűjtött adatok két csoportra oszthatók: a rendszer működését megjelenítő adatokra és a munkakörnyezetet leíró adatokra. A kommunikáció egy integrációs folyamat, azaz fizikai folyamatok és a digitális platformok összekapcsolása. A hálózati kommunikáció kulcsfontosságú tényező, amely lehetővé teszi a digitális ikrek létrehozását, amely egy olyan folyamatinterfészből áll, amelyen keresztül a jelek és az adatok továbbításra kerülnek a termékről a digitális platformra, valamint azon keresztül egy kommunikációs interfészre, amely lehetővé teszi az alkatrészek és platformok közötti kommunikációt, valamint az adatvédelmi biztonsági rendszerek alkalmazását. A következő lépés az adatok gyűjtése, tárolása és elemzésre való előkészítése. Az elemzés során az adatok feldolgozásra kerülnek és az eredmények vizuálisan jelennek meg, lehetővé téve a felhasználó számára, hogy jobban

érzékelje, átlássa az egész rendszert, így biztosítva a fejlődést. Ennek alapján történnek a döntések a további intézkedésekről, amelyeket fizikai formában, aktuátorok hajtanak végre.

Végül levonható a következtetés, hogy a digitális ikerpár technológia nemcsak az emberek elméleteit és ismereteit használhatja virtuális modellek létrehozásához, hanem a virtuális modellek szimulációs technológiáját is felhasználhatja az ismeretlen világ feltárására és előrejelzésére, valamint jobb utak megtalálására, a kreatív tevékenység folyamatos ösztönzésére.

3. A digitális képfeldolgozás alapelvei

A digitális képfeldolgozás a számítógépes képfeldolgozás módszereinek összességét foglalja magába. A feldolgozott képek eredete eltérő lehet. A kamerán kívül, ami a képfeldolgozó rendszer leggyakoribb bemenete, a bemeneti képek műholdakból, repülőgépekből, orvosi diagnosztikai eszközökből, telekommunikációs átviteli eszközökből, adathordozókból, radarból, szonárból és egyéb eszközökből származhatnak. A legtöbb esetben a képfeldolgozó rendszerbe bejövő kép a spektrum látható részéből származik, bár egyes esetekben ez lehet kétdimenziós (2D) jel az elektromágneses spektrum láthatatlan részéből is. Ezért tágabb értelemben a digitális képfeldolgozás kifejezést használják bármely 2D adat feldolgozására is.

3.1. Bevezetés a digitális képfeldolgozásba

A digitális képfeldolgozás egy multidiszciplináris tudomány, amely az optika, az elektronika, a matematika, a fényképezés és a számítástechnika különböző aspektusait foglalja magába.

A digitális képfeldolgozó rendszer kimeneti jele általában egy új kép, amely egy videomonitoron, vagy másmilyen kijelzőn jelenik meg. Ezenkívül a kimeneti adatok megjeleníthetők fénykép vagy film formájában, egyes alkalmazásokban pedig számszerű adatok formájában is.

A digitális képfeldolgozás az 1920-as évek elejére nyúlik vissza. Ugyanis a digitális képfeldolgozás első alkalmazása olyan képek kódolására vonatkozott, amelyeket kábelon továbbítottak az Atlanti-óceánon keresztül és rekonstruáltak a vételi oldalon. A digitális képfeldolgozás elemi technikáinak bevezetése jelentősen lerövidítette a képátadás folyamatát és javította a minőséget. Bár a képátviteli és rekonstrukciós technikák terén a későbbi időszakban jelentős előrelépés történt, a digitális képfeldolgozás csak a számítógépes berendezések és mikroszámítógépek fejlesztésével és az űrkutatás megkezdésével indult gyorsabb fejlődésnek.. Például 1964-ben először alkalmaztak digitális számítógépet a "Ranger 7" űrszondával a Holdról készített képek minőségének javítására. Ezt követően a digitális képfeldolgozás elkerülhetetlen részévé vált a különböző műholdak és űrkutatási szondák adatfeldolgozásának. Ezért a digitális képfeldolgozás éveken keresztül szinonimája volt a nagyon drága csúcstechnológiának és mint ilyen, a világon csak kevés kutató laboratórium számára állt rendelkezésre teljes mértékben.

Az elektronika és a számítástechnika, és különösen a mikroprocesszorok és az integrált digitális jelfeldolgozó processzorok gyors fejlődésével a digitális képfeldolgozás számos más területre behatolt, ahol korábban magas költségei miatt nem lehetett használni. Így például a digitális képfeldolgozást intenzíven használják az orvostudományban a röntgen- és ultrahangképek feldolgozására és javítására, a szervek háromdimenziós rekonstrukciójára a tomográfia és a mágneses rezonanciánál stb. A digitális képfeldolgozás további fontos alkalmazásai megtalálhatóak a fizikában, a csillagászatban, a biológiában, a kriminológiában, a kohászatban, a geológiában, sőt a régészetben és a műalkotások rekonstrukciójában

és restaurálásában is. Az elmúlt húsz évben a digitális képfeldolgozás alkalmazásra talált a filmekben, a televízióban, a zene gyártásában és reklámokban, a robotikában, az ipari termékellenőrzésben, az automatikus vezérlőrendszerekben stb.

Bár a képfeldolgozó rendszer bemeneti és kimeneti jelei nagyon sokfélék, mint azok alkalmazása is, számtalan képfeldolgozási algoritmus megegyezik annak ellenére, hogy nagyon eltérő területeken használják őket. Ezért a digitális képfeldolgozásban használt összes módszer csak hat területre osztható: képábrázolási módszerek, képjavító módszerek, kép-helyreállítási módszerek, képtömörítési módszerek, képelemzési módszerek és vetületi (projekciós) képrekonstrukciós módszerek. Ez a lista kibővíthető még az objektum felismerésével is, habár a szakirodalomban gyakran az objektum felismerés teljesen különálló tudományágnak számít az összes módszerrel, amit magába foglal.

3.1.1. Képábrázolási módszerek

A képábrázolás alapvető problémája a kép *egyik elemével* képviselt méret jellemzése, amelyet röviden *pixelnek* vagy *pel-nek* hívnak. A kép az eredetétől függően reprezentálhatja a kamera előtti jelenetben lévő tárgyak fényességét (ha a spektrum látható részét érzékeny kamera érzékeli), a szövetek abszorpciós jellemzőit (röntgen), a megjelenítendő tárgyak hőmérsékletét (infravörös kamerák), a szövetek reflexiós jellemzőit (ultrahang képek) stb.

A képábrázolás fontos problémája az a *hűség*, amellyel a kép leírja a megfigyelt fizikai jelenséget. Az ábrázolás hűsége függ a kamerában használt érzékelő minőségétől, de a térbeli és amplitúdóbeli diszkretizálás folyamatától is (azaz a mintavételezéstől és az A/D átalakítástól). A képábrázolás különféle képtranszformációs eljárásokkal is foglalkozik, amelyek a kép tartalmának jobb leírására, minőségének javítására és a tartalom tömörítésére szolgálnak.

A képábrázolás problémáihoz szorosan kapcsolódnak a képmodellezési eljárások is. A képet általában statisztikailag modellezzük első és másodrendű

statisztikai jellemzők felhasználásával. A kép stacionaritásának feltételezését gyakran használják, bár ez a feltételezés nem mindig teljesül.

A képelemzésnél sokkal magasabb szintű modellezést alkalmaznak, ahol a képet több objektumból állónak tekintik és a modell meghatározza a jelenetben lévő objektumok kapcsolatait.

Külön problémát jelent a képsorok ábrázolása és modellezése, ahol az idő változóként jelenik meg, vagyis a háromdimenziós (3D) tér függvénye.

3.1.2. Képjavító módszerek

Az alapvető *képjavító* módszereknél a fő cél a kép egyes jellemzőinek megváltoztatása, hogy azok alkalmasabbak legyenek kijelzőn történő megjelenítésre, vagy további elemzésre. A képjavítás elvileg nem javítja a kép információtartalmát, hanem megkönnyíti a meglévő információk felhasználását. Számos módszer létezik a kép javítására, de sok minden függ a kép elkészítésének körülményeitől, valamint a továbbfejlesztett kép alkalmazásától is.

Ezért sok képjavító eljárás interaktív, vagy adaptív. A képjavítás leggyakoribb módszerei a kontrasztnövelés, a zajcsökkentés, a képélesítés, a kép színezése stb. Bizonyos módszerekben a kép minden képpontját a többitől függetlenül, más esetekben a képpontokat a környezetüktől függően kezelik. A képjavító módszerekre jellemző, hogy általában nagyon egyszerűek, így valós időben is elvégezhetőek.

3.1.3. Kép-helyreállítási módszerek

A *kép helyreállítását, restaurációját* a képminőség javítására is használják, de olyan esetekben, amikor a kép romlása ismert, vagy amikor kialakítható a romlás modellje.

A helyreállítás jellemző példái a kamera optikai részének geometriai torzításainak korrekciójára, a statisztikai modell alapján ismert zaj eltávolítására, a

kamera mozgása vagy fókuszálása miatt csökkent képélesség korrekciójára vonatkoznak. A korábban említett képjavító módszerekkel ellentétben a helyreállítási módszerek sokkal összetettebbek és általában nem valós időben hajtják végre őket.

3.1.4. Képtömörítési módszerek

A kép hatalmas mennyiségű adatot tartalmaz. Például egy szabványos filmkép (régi fényképezőgépeknél), amelynek méretei 24mm×36mm, körülbelül 10^7 elemmel rendelkezik, ha a film felbontása 0,01mm. Egy szabványos 512×512 pixel felbontású digitális kép körülbelül 256000 elemet tartalmaz. Ha minden pixelt 8 bit képvisel, akkor 256 kB-ra van szükség egy ilyen kép ábrázolásához. Még nagyobb mennyiségű adat található a képsorokban. Az 1 másodpercig tartó TV képsor megközelítőleg 10 MB tárhelyet igényel.

Az ilyen nagy mennyiségű adatok továbbítása telekommunikációs csatornákon komoly problémát jelent. Hasonló problémák merülnek fel, ha kép- vagy videojelet kell tárolni egy memória adathordozón. Annak érdekében, hogy a képek átvitele és tárolása technikailag lehetséges és gazdaságilag kifizetődő legyen, bizonyos *tömörítési technikákat* kell alkalmazni, amelyek csökkentik a képábrázoláshoz szükséges bitek számát.

Tekintettel a kép adattömörítésének fontosságára és szükségességére, nagyszámú tömörítési módszert fejlesztettek ki. A tömörítési módszerek egy csoportja információvesztés nélkül csökkenti a képábrázolás bitjeinek számát. A tömörítési módszerek második csoportja viszont a tömörítést jelentéktelen mennyiségű információ elvesztésével hajtja végre. Természetesen, a második módszer csoport tömörítési foka sokkal magasabb. A képsorokban történő adattömörítéshez is számos módszert dolgoztak ki, amelyek nagyon nagy mértékű tömörítést tesznek lehetővé, így biztosítva a videojelek digitális formában történő tárolását és továbbítását.

3.1.5. Képelemzési módszerek

A *képelemzést* a képen lévő objektumok kvantitatív jellemzőinek és kölcsönös viszonyainak meghatározása céljából végzik. A képelemzés egyszerű példái: az automatizált gyártósorok nyomon követése az iparban, automatikus képelemzés a biomedicinában, képelemzés a biológiában stb. Az elemzés összetettebb példái közé tartozik az alakfelismerés a robotikában és más alakfelismerési alkalmazásokban, amit manapság számítógépes vagy mesterséges látás néven ismernek. Ezek az eljárások a modern technológiák alapját képezik és széleskörűen használják őket.

A képelemzés számos eljárásból áll, amelyek bizonyos mértékig utánozzák az emberek észlelésének folyamatát. Ez az eljárás csoport magában foglalja az élek kiemelésére vonatkozó eljárásokat, szegmentációs eljárásokat stb.

3.1.6. Vetületi képrekonstrukciós módszerek

A *vetületi, projekciós képrekonstrukció* a helyreállítási eljárások különleges esete, amely 2D (vagy többdimenziós) objektumokat rekonstruál, állít helyre több 1D (vagy kétdimenziós) vetületből. Legfontosabb alkalmazási területük az orvostechikai eszközök, pontosabban a képalkotásos diagnosztika (különböző típusú szkennerek), de még alkalmazást nyertek az anyagok roncsolásmentes tesztelésében, a csillagászatban stb.

3.1.7. Objektum felismerési, alakfelismerési módszerek

Az *objektum felismerés* két kategóriába sorolható: elméleti-döntéshozó és strukturált. Az első kategória olyan mintákon alapul, amelyek kvantitatív leírást használnak, mint a hossz, a terület, a textúrák stb. A második kategória a legjobban ábrázolt mintákon alapszik, mint szimbolikus információ, például a karakterláncokon és tulajdonságokon, valamint a köztük lévő kapcsolatokon. Az

objektumfelismerés központi témája a kiválasztott mintákból történő „tanulás” fogalma.

Manapság az alakfelismerés leghatékonyabb algoritmusai a mesterséges intelligencián alapulnak, mint pl. a neurális hálózatok, fuzzy logika, anfis, Deep Learning, konvolúciós neurális hálózatok stb.

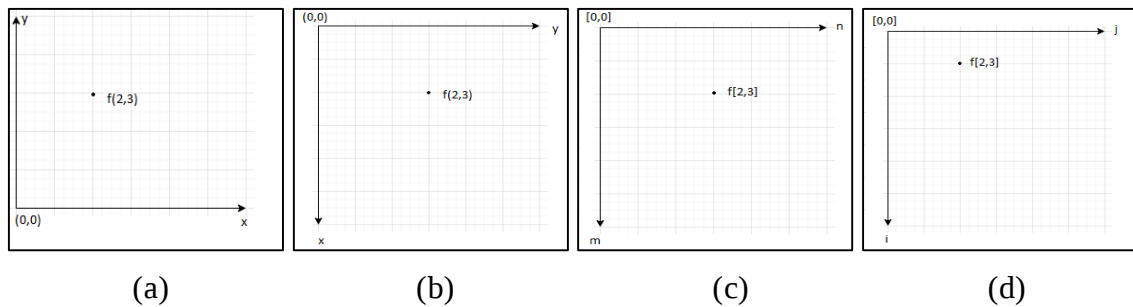
Az objektum felismerési módszereknek külön fejezet lesz szentelve a tananyagban.

3.2. A kép matematikai és grafikus ábrázolása

A következőkben a monokromatikus kép (fekete-fehér vagy szürke kép) kifejezés a folytonos $f(x,y)$ függvényt jelzi, amely 2D intenzitáseloszlást képvisel és ahol a x és az y a térbeli koordinátákat képviselik. Az $f(x,y)$ függvény értéke a szürke árnyalat szintjét jelenti az (x,y) pontban. A digitális monokromatikus kép egy olyan $f(x,y)$ kép, amelyben a térbeli koordinátákat és az intenzitást is diszkrétizálják. Ezért egy digitális képnek véges számú pixel értéke van és egy olyan mátrix által ábrázolható, amelynek a sor és oszlop indexei a képben figyelt pont helyzetét jelölik, a mátrixelem értéke pedig az adott pont szürke árnyalatát, szintjét mutatja ki.

A megszokott derékszögű Descartes koordináta-rendszerrel ellentétben, a televíziós képtől származó képkoordináta-rendszert sokkal gyakrabban használják a képfeldolgozásban. Könnyen észrevehető, hogy a kép koordináta-rendszere eltér a Descartes koordináta-rendszertől az origó helyzetét figyelve, amely most a bal felső sarokban van, valamint abban, hogy a koordinátatengelyek megváltoztatták szerepüket. Ezenkívül az ordináta, az y tengely lefelé irányul. A harmadik ábrán látható koordináta-rendszer a digitális kép ábrázolására szolgál. A digitális kép pixelkoordinátái egész számok, a koordináták kiinduló pontja pedig a $[0,0]$ pontnak felel meg. Mátrixos koordináta-rendszer, amely megfelel a szokásos mátrix jelölésnek (az első koordináta a sorait, a második pedig a mátrix oszlopait jelöli) a

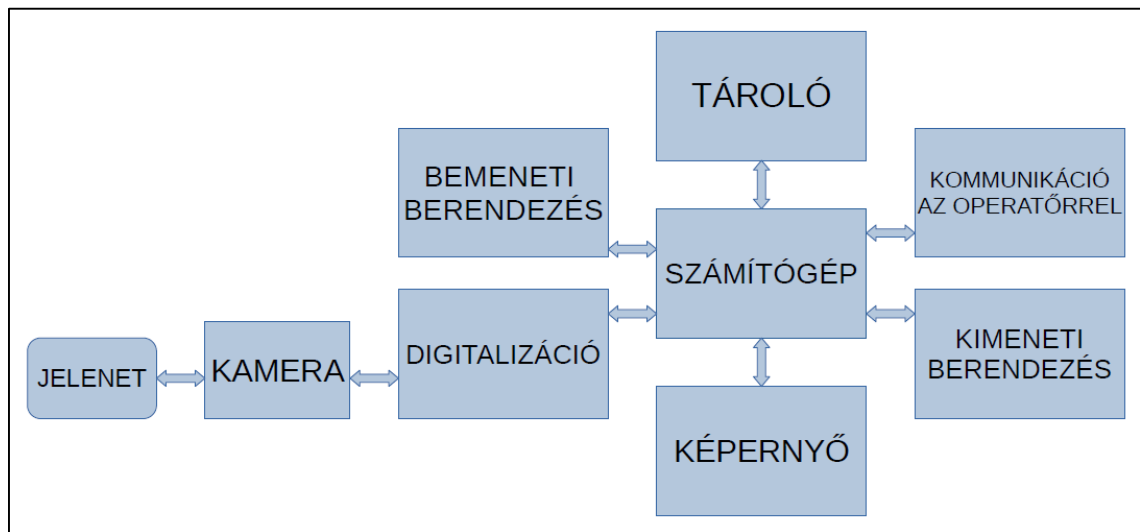
negyedik ábrán látható. Csak annyiban tér el az előzőtől, hogy a koordinátarendszer origójának az $[1,1]$ pont felel meg.



3.1. ábra: Derékszögű koordinátarendszerek: (a) derékszögű Descartes-féle koordinátarendszer, (b) folytonos-analóg képábrázolási koordinátarendszer, (c) digitális képábrázolási koordinátarendszer, (d) mátrix koordinátarendszer

3.3. A képfeldolgozó rendszerek

Habár a digitális képfeldolgozó rendszerek felépítése nagymértékben függ a rendeltetésüktől, a rendszer alapvető elemei az esetek többségében megegyeznek és a 3.2. ábrán láthatóak. A rendszer alapeleme a képet feldolgozó számítógép. A számítógép a rendszer más részeihez csatlakozik, amelyek közül a legfontosabbak: a memória, videókijelző, digitalizáló eszközzel ellátott kamera, beviteli eszközök, kimeneti eszközök és a felhasználóval való kommunikációhoz szükséges perifériás eszközök.



3.2. ábra: *A képfeldolgozó rendszer alapvető elemei*

A memóriát a digitális képek mátrix formájú állandó, vagy ideiglenes tárolására használják. A gyakorlatban leggyakrabban mágneses adathordozókat (lemezeket, hajlékonylemezeket és sokkal ritkábban szalagokat) használnak memóriaeszközként, ezeket követik az optikai lemezek (CD-ROM és digitális optikai lemezeket). Manapság az árcsökkenésnek köszönhetően akár SSD memóriákat is használnak.

Mivel a legtöbb digitális képfeldolgozó rendszer megköveteli a feldolgozott kép megjelenítését a rendszerüzemeltető számára, erre a célra egy speciális, nagy felbontású, nagyobb képernyős videokijelző használható. A számítógépes rendszernek rendelkeznie kell még egy kisebb, gyengébb jellemzőkkel rendelkező kijelzővel is, amelyen keresztül az operátor kommunikál a számítógéppel.

A digitalizáló eszközzel ellátott kamera alapvető eszköz a digitális képek számítógépbe történő beviteléhez. Meg kell jegyezni, hogy ma már vannak beépített A/D átalakítóval rendelkező kamerák, amelyek közvetlenül adnak ki digitális jelet. Mégis, még ma is a legtöbb rendszer továbbra is olyan kamerákat használ, amelyek analóg kimenettel rendelkeznek és kompatibilisek valamilyen televíziós szabvánnyal, míg a kép digitalizálási része a képfeldolgozó rendszerhez tartozik.

Más beviteli eszközök is használhatók alternatív módként a képek számítógépbe történő bevitelére. Ezek elsősorban különféle típusú szkennerek, analóg videorekorderek, digitális videorögzítők, nem szabványos kamerák, filmkép képátalakítók, orvostechikai eszközök érzékelői stb. Természetesen a kép bármilyen adathordozóról, valamint helyi számítógépes hálózaton, vagy az interneten keresztül is beírható számítógépbe.

A kimeneti eszközök nagyon sokoldalúak lehetnek. Ezek közül a legfontosabbak a képek nyomtatott másolatának elkészítésére szolgáló eszközök, például a különféle lézer, vagy tintasugaras nyomtatók (fekete-fehér vagy színes), a fényképek készítéséhez használt nyomtatók, analóg videorekorderek, digitális videorekorderek stb.

A számítógép felhasználóval való kommunikáció leggyakoribb eszközei a kijelző, a billentyűzet és az egér. A botkormányokat, grafikus táblákat és az ehhez hasonló eszközöket sokkal ritkábban használják.

3.4. A 2D jelek és rendszerek

Mivel a kép lényegében egy 2D függvény, a képen végzett számos műveletet a digitális jelfeldolgozás során használt matematikai műveletekkel lehet kifejezni. A képen végzett műveletek egy része a műveletek egyszerű általánosítását jelenti az 1D és a 2D tér között, míg egyes műveletek és jelek csak 2D térben végezhetők el.

A 2D diszkrét rendszer bemeneti és kimeneti jelének kapcsolata a legáltalánosabb esetben a következő összefüggéssel ábrázolható:

$$y[m,n] = R\{x[m,n]\} \quad (3.1)$$

Mint az 1D rendszerek esetében, az R műveletoperátor számára különféle korlátozások bevezetésével a rendszer matematikai leírásának jelentős egyszerűsítése érhető el. A legfontosabb korlátozások, amelyek bevezethetők, a

linearitás és a térbeli (időbeli) változatlanság. A linearitás tulajdonságát a reláció fejezi ki:

$$R\{\sum_{i=1}^K a_i x_i[m, n]\} = \sum_{i=1}^K a_i y_i[m, n] \quad (3.2)$$

amelyet gyakran a szuperpozíció elvének is neveznek. A térbeli változatlanság tulajdonságát a következő kifejezés fejezi ki:

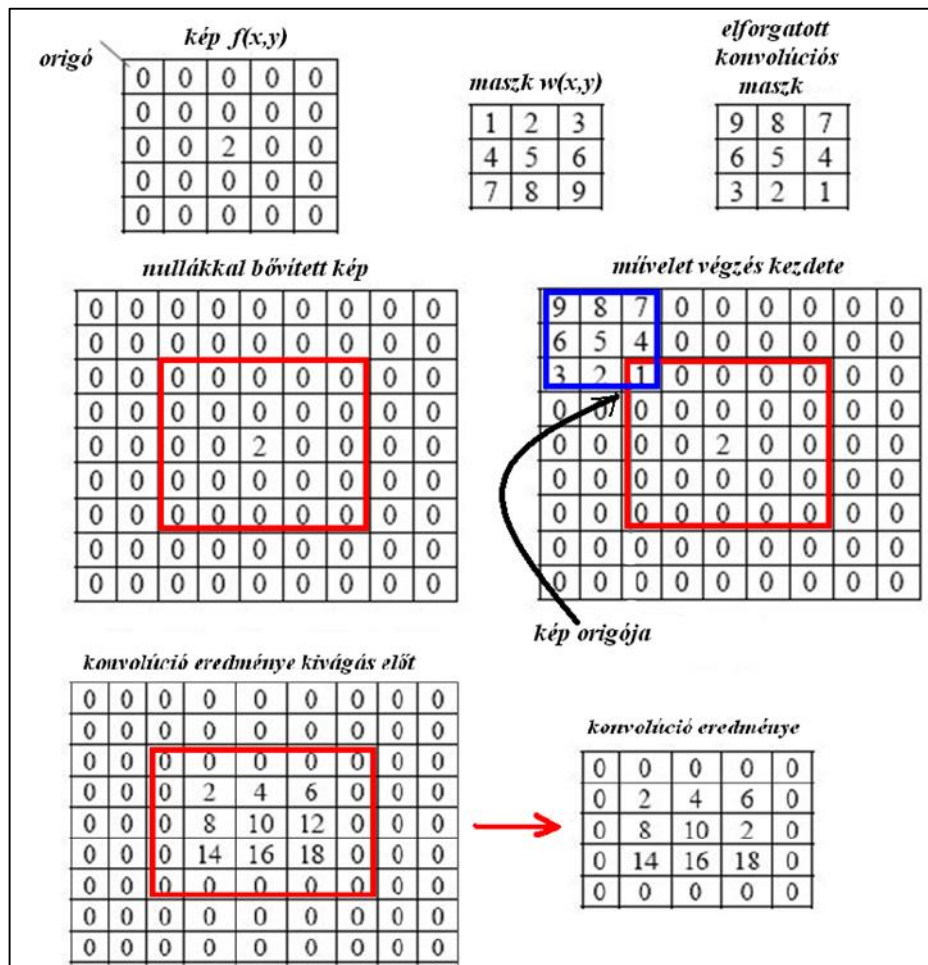
$$R\{x[m - m_1, n - n_1]\} = y[m - m_1, n - n_1] \quad (3.3)$$

Az egyik legfontosabb művelet a 2D jeleknél, ezáltal a képfeldolgozásban is a konvolúció. A konvolúció lényegében teljesen meghatározza a rendszer válaszát egy tetszőleges $x[m, n]$ gerjesztő jelre, ha ismert a rendszer impulzus válaszfüggvénye $h[m, n]$. Diszkrét jelek (képek) esetében a konvolúció a következő kifejezéssel fejezhető ki:

$$y(m, n) = \sum_{m'=-\infty}^{+\infty} \sum_{n'=-\infty}^{+\infty} x(m', n') h(m - m', n - n') \quad (3.3)$$

A konvolúció eredményeként kapott kép méretet a következő módon határozzuk meg. Ha a kép nagysága $M \times N$ pont, a konvolúciós maszk, ez esetben a válaszfüggvény nagysága pedig $K \times L$ pont, akkor az eredmény nagysága $(M+K-1) \times (N+L-1)$ képpont.

A konvolúciónak nagy szerepe van a képszűrésnél, amely a legalapvetőbb műveletnek számít a képfeldolgozási algoritmusokban. Pl. az alakfelismerési algoritmusoknál a szűrés és a képszegmentáció azok a műveletek, amelyek a bemeneti képet átalakítják, megformálják, előkészítik ahhoz, hogy lehetséges legyen elvégezni a megfelelő objektumok felismerését.



3.3. ábra: A 2D konvolúció példája

4. A digitális képek szűrése és szegmentálása

A digitális ikerpár technológia egyik alapköve a digitális képfeldolgozás és az objektumok felismerése a képek alapján. A képfeldolgozási technikáknak köszönhetően, ahová az objektumok felismerése is tartozik, hozzák létre a valós termékek virtuális, digitális iker párjait. A következőkben részletesen foglalkozunk a digitális képfeldolgozás alapvető műveleteivel, melyek legfőbb feladata a bemeneti képet előkészíteni és átalakítani olyan formába, hogy az objektum felismerési

algoritmus képes legyen felismerni a megfelelő objektumokat a képen.

4.1. A képszűrés alapelvei

Mint említve volt, a lineáris térbeli szűrés az egyik leggyakrabban alkalmazott művelet a digitális jelfeldolgozásban, digitális képfeldolgozásban és a híradástechnikában. A digitális jelfeldolgozásban 1D jelek szűrését végzik (pl. hang, beszéd, zene, orvosi jelek, stb), míg a digitális képfeldolgozásban 2D és 3D jelek szűrését végzik (pl. állóképek, mozgóképek, stb).

A szűréselmélet alapját a Fourier transzformáció (a továbbiakban FT) képezi. Az idő, vagy a képtertományban pedig két művelet, a korreláció és konvolúció azok, amelyek képezik a szűréselmélet alapjait. A korreláció és a konvolúció között annyi a különbség, hogy a konvolúciónál a maszk 180° -kal el van fordítva a korrelációhoz képest. A képek szűrésnél leginkább a konvolúciót alkalmazzák. A digitális jelfeldolgozásban különösen kedvelt a konvolúció, mert léteznek gyors algoritmusok a kiszámítására és az utóbbi években kifejlesztették az úgynevezett konvolúciós digitális jelfeldolgozó (DSP) processzorokat is, amelyek alkalmasak a gyors adatfeldolgozására, így az összetett jelek gyors szűrésére is, ami a valós idejű alkalmazásokban igen lényeges szempont.

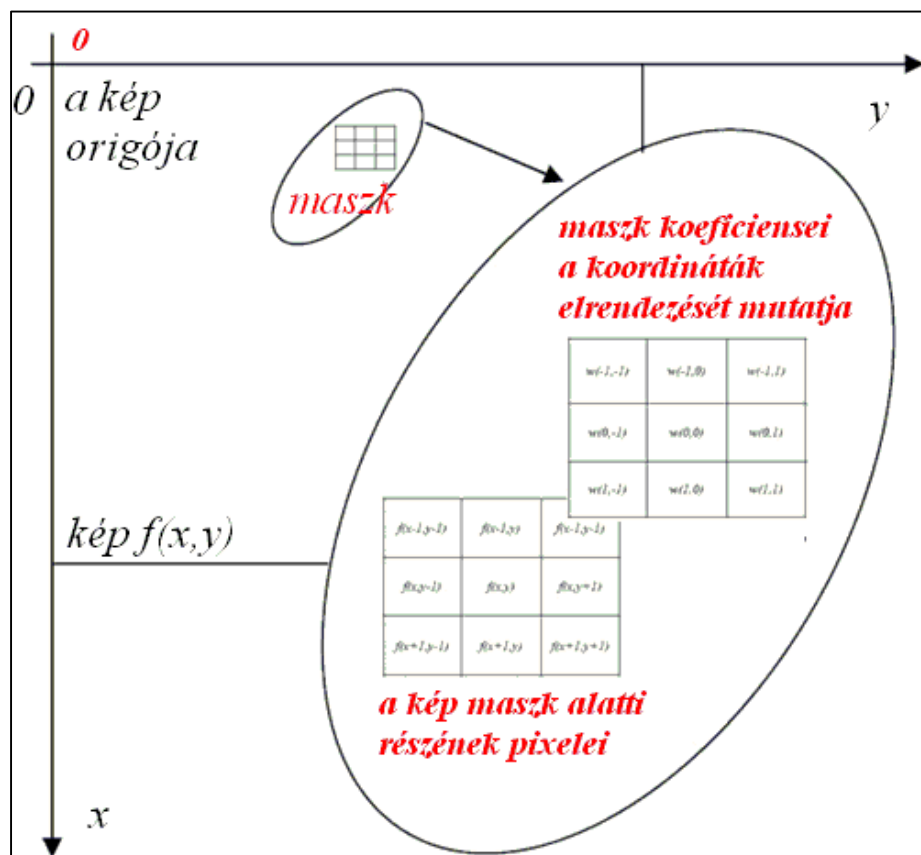
Gyakorlatban a térbeli szűrés művelete valójában a szomszédos pontokkal való műveletekből tevődik össze következőképpen:

1. a középpont meghatározása, (x,y) ,
2. a műveletek elvégzése azokkal a pixelekkal, amelyek az előre meghatározott szomszédságban helyezkednek el a középpont körül,
3. a kapott eredmény jelzi a kimenetet az “adott” pontban,
4. a művelet megismétlése a kép minden pontjában.

A maszk középpontjának mozgatása új szomszédságokat hoz létre, egyet a bemenő kép minden pixelére. Ezt a műveletet nevezik térbeli szűrésnek (*Spatial Filtering*).

Lineáris térbeli szűrésnél az eredményt a szűrő koeficiensek és a hozzátartozó kép-pixelek szorzatának az összegeként számolhatók, a szűrő maszkkal lefedett részen. Ha R -nek jelöljük a szűrés eredményét, akkor a szűrés eredménye:

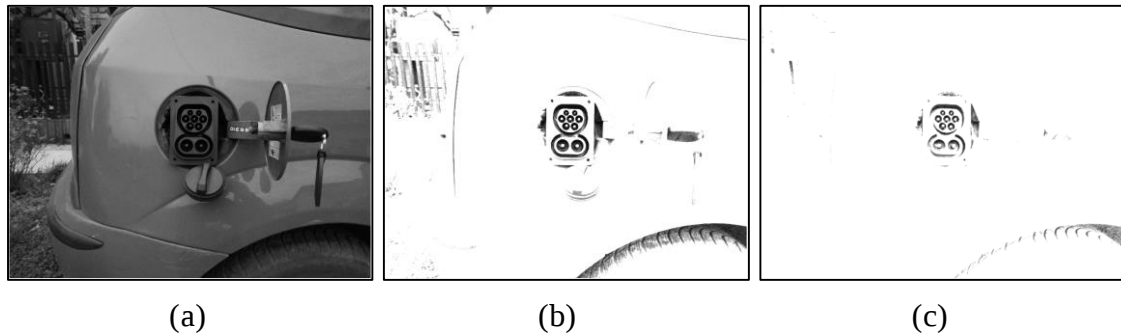
$$R = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \dots + w(0,0)f(x,y) + \dots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1) \quad (4.1)$$



4.1. ábra A térbeli szűrés mechanizmusa

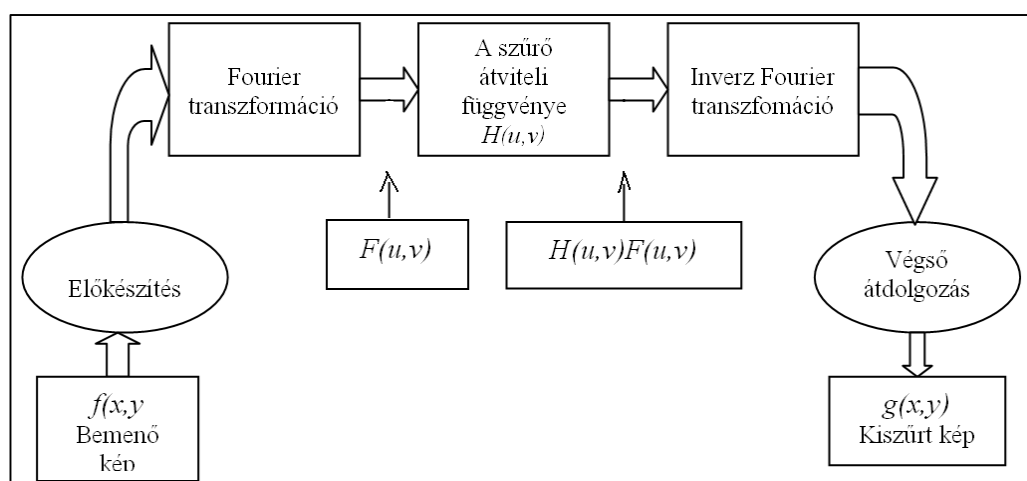
A következő ábrán szemléltetve lesz egy térbeli szűrés példája. Kétféle nagyságú szűrővel történt a szűrés: az első esetben 3x3 pixel felületű, a második esetben pedig 5x5 pixel felületű szűrővel. Mindkét szűrőmaszk csak egyeseket tartalmaz és az ilyen szűrők lényegében helyi átlagolást végeznek a képben. A legfontosabb észrevétel az, hogy a szűrő méretének növekedésével növekszik az

elmosódás is, a kép jobban veszt a tartalmából és a szűrt kép szürke árnyalatokban jelenik meg.



4.2. ábra A térbeli szűrés példája: (a) bemeneti kép; (b) az 3x3; (c) 5x5 szűrőkkel való szűrés eredményei

A frekvencia tartományban szűrés a Fourier Transzformáció (FT) alkalmazásán alapszik. A képtartománybeli szűréshez képest, a frekvencia tartománybeli szűrésnek számos előnye van és mindezek az előnyök jelentkeznek a szűrők tervezésében, képek restaurálásában, képek tömörítésénél, képek javításánál és más alkalmazásokban is. A jobb szűrési eredmények elérése céljából lehet a képtartománybeli és frekvenciatartománybeli szűrési eljárások kombinációját is alkalmazni. A szűrés frekvencia tartománybeli folyamatát a 4.3. ábra szemlélteti.



4.3. ábra A szűrés mechanizmusa frekvencia tartományban

A frekvenciatartománybeli szűrés lépései a következők:

1. a bemeneti képet be kell szorozni $(-1)^{x+y}$ -nal, hogy a transzformáció középpontos legyen
2. az $F(u,v)$ kiszámítása, vagyis az $f(x,y)$ diszkrét FT-nak (DFT) kiszámítása
3. az $F(u,v)$ megszorozása a szűrő $H(u,v)$ átviteli függvényével
4. a szorzat inverz DFT-nak (IDFT) kiszámítása
5. a kapott eredmény valós részének kiválasztása
6. a valós rész beszorzása $(-1)^{x+y}$ -nal, hogy megsemmisítse a kezdeti szorzás hatását.

Előkészítés alatt leginkább a kép átalakítását értik megfelelő formába (szürke árnyalatokba, bináris alakba, kép méretének átalakítása, stb.). A végső feldolgozás alatt legtöbbször a kapott eredmény átméretezését értik, jobb értelmezés és látás céljából. A folytatásban röviden be lesz mutatva a frekvenciabeli szűrés folyamata.

Az első lépésben a szorzásra azért van szükség, hogy a frekvencia tartományban a DFT eredménye középpontos legyen, vagyis:

$$\mathfrak{F}[f(x,y)(-1)^{x+y}] = F(u - M/2, v - N/2). \quad (4.2)$$

Ezzel a szorzással az $F(u,v)$ origójának koordinátái eltolódnak az $(M/2, N/2)$ koordinátákra a frekvencia tartományban, ahol az $M \times N$ méretű kép középpontja van.

A második lépésben ki kell számítani a bemeneti kép 2D DFT-ját a következő képlet szerint:

$$\mathfrak{F}[f(x,y)] = F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)} \quad (4.3)$$

ahol az $F(u,v)$ a $M \times N$ nagyságú $f(x,y)$ bemeneti kép 2D DFT-ját jelzi. Az u és v a frekvenciatartománybeli változókat jelölik (amelyeket sokszor f_1 és f_2 -vel is jelölnek), az x és y pedig a kép térbeli koordinátáit jelölik.

A harmadik lépésben a DFT-val kapott frekvenciabeli $F(u,v)$ képet össze kell szorozni a $H(u,v)$ átviteli függvénnyel:

$$G(u,v) = F(u,v)H(u,v) \quad (4.4)$$

ahol a $G(u,v)$ a kapott eredményt, kiszűrt képet, annak a spektrumát jelöli.

A negyedik lépésben ki kell számítani a $G(u,v)$ inverz DFT-ját a következő kifejezés szerint:

$$\mathfrak{T}^{-1}[G(u,v)] = g(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M + vy/N)} \quad (4.5)$$

ahol a $g(x,y)$ a térbeli tartományban kapott képet jelöli.

A következő lépésben a $g(x,y)$ kiszűrt képet szét kell bontani a reális és az imagináris részekre és ki kell emelni a valós részt.

$$g_f(x,y) = \text{Re}[g(x,y)] \quad (4.6)$$

ahol a $g_f(x,y)$ a $g(x,y)$ valós részét jelöli.

Végül, a meghatározott reális képet ismét be kell szorozni $(-1)^{x+y}$ -nal, hogy a végső kiszűrt kép középpontos pozicionálású legyen, vagyis a kép origója a $g_f(0,0)$ -ban legyen.

A konvolúció és a szorzás összefüggése Fourier transzformáció alkalmazása esetében a következő kifejezésekkel fejezhető ki:

$$f(x,y) * h(x,y) \leftrightarrow F(u,v)H(u,v) \quad (4.7)$$

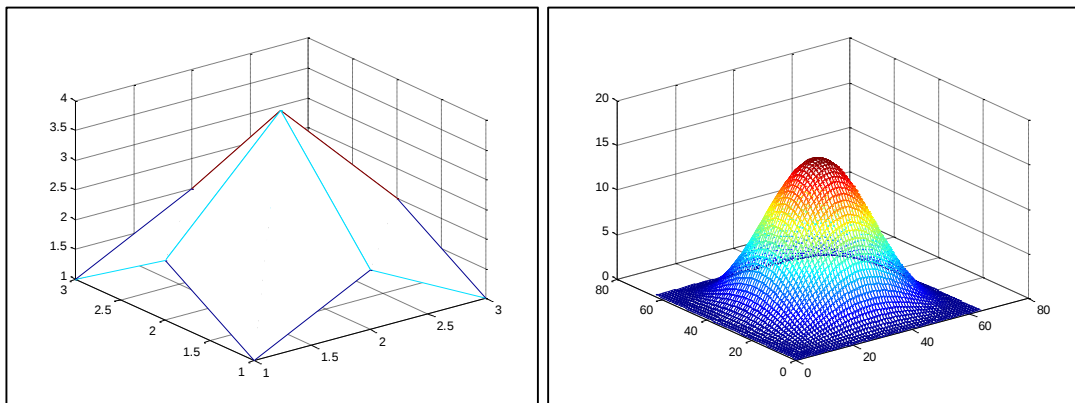
$$f(x,y)h(x,y) \leftrightarrow F(u,v) * H(u,v) \quad (4.8)$$

A következő példában egy aluláteresztő szűrő lesz bemutatva az alábbi szűrőmaszkkal:

1	2	1
2	4	2
1	2	1

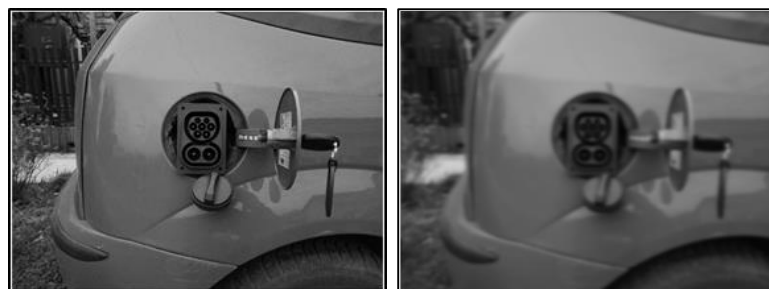
4.4. ábra Az aluláteresztő szűrő maszkja

Az adott szűrő térbeli és frekvenciabeli változatai a 4.5. ábrán láthatók:



4.5. ábra: Az aluláteresztő szűrő: (a) térbeli és (b) frekvenciabeli változatai

A 4.6. ábrán látható egy kép szűrésének eredménye aluláteresztő szűrővel frekvencia tartományban. Az aluláteresztő szűrő alapvető tulajdonsága, hogy „elmossa” a kép tartalmát és részben eltávolítja a zajokat, mint ahogy látni az ábrán.



(a)

(b)

4.6. ábra: (a) A bemeneti kép, (b) a frekvencia tartománybeli szűrés eredménye aluláteresztő szűrővel

Fontos megjegyezni, hogy a gyakorlatban rengeteg képszűrő létezik és igen széleskörű a felhasználásuk. Némely szűrőknek a zajcsökkentésben van szerepük, más szűrőknek a kontrasztjavításban van szerepük, viszont külön csoportot képeznek azok a szűrők, amelyeket a képek szegmentálásában alkalmaznak. Ide tartoznak az élkiemelő és textúra detektáló és kiemelő szűrők. Lényegében ezek a szűrők képezik a képszegmentálás és alakfelismerés alapját, mert a hatékony alakfelismerés alapja a megfelelő képszegmentáció.

4.2. A képszegmentáció alapjai

A következő fejezetben a képszegmentáció alapelvei és korszerű technikái kerülnek szemléltetésre röviden. Mint ismeretes, a képszegmentáció a képfeldolgozási algoritmusok legfontosabb művelete mert a megfelelő és minőséges szegmentálás nagyban befolyásolja a képfeldolgozási algoritmus végeredményét. Ez kiemelten vonatkozik az objektum felismerési algoritmusokra, mert a sikeres objektum felismerés előfeltétele a jó képszegmentáció. Az évek során rengeteg képszegmentációs eljárás alakult ki, ennek ellenére napjainkban sem nélkülözhetetlenek az alapvető szegmentációs eljárások a képfeldolgozási és alakfelismerési algoritmusokban.

A képszegmentálás a digitális képfeldolgozás egyik legigényesebb, legérdekesebb és legkutatottabb területe. A kifejlesztett képszegmentálási algoritmusok egyike sem alkalmazható egyetemesen minden képre és a különböző algoritmusok sem egyformán alkalmasak minden fajta kép szegmentálásának alkalmazására. A szegmentálás alapvető célja az adott kép régiókra, szegmensekre való osztása.

Ezt a problémát többféle megközelítéssel lehet megoldani, a leggyakoribbak:

- az intenzitás-folytonosság változásának meghatározása,
- küszöb alkalmazása,
- morfológiai műveletek alkalmazása stb.

Az intenzitás-folytonosság változásának három alapvető típusa van: a pont, a vonal és az élek. A detektálásukhoz leggyakrabban lineáris és nemlineáris szűrőket használnak, ahol a szűrést szűrőmaszkokkal, azaz szűrőkkel végzik 2D konvolúcióval. Tehát a kép minden egyes pixelére szükséges meghatározni a megfelelő szűrő választát.

Egy elszigetelt pont detekcióját a képben a következő szűrővel lehet elvégezni:

-1	-1	-1
-1	8	-1
-1	-1	-1

4.7. ábra: Szűrőmaszk a pont detektálásához

A vonalak detektálásához hasonló felépítésű szűrők alkalmazhatók azzal, hogy a szűrők megkülönböztethetők aszerint, hogy milyen fekvésű vonalakat emelnek ki (vízszintes, függőleges, ferde). A következő ábrán láthatók a szűrők példái:

-1	-1	-1
2	2	2
-1	-1	-1

-1	-1	2
-1	2	-1
2	-1	-1

-1	2	-1
-1	2	-1
-1	2	-1

2	-1	-1
-1	2	-1
-1	-1	2

4.8. ábra: Szűrőmaszkok a vonalak detektálásához

Az első szűrő a vízszintes vonalakat emeli ki, a második és a negyedik a ferde vonalakat, a harmadik szűrő pedig a függőleges vonlakat emeli ki. A vonalak

detektálása mindegyik esetben a kép és a megfelelő szűrőmaszk 2D konvolúciójával lesz meghatározva.

Az éldetektálás gradiens operátorok segítségével történik és ez képezi a képszegmentálás egyik legalapvetőbb, és gyakran nélkülözhetetlen műveletét valahol az algoritmuson belül. Erre a célra leginkább a Prewitt, Sobel és Canny féle éldetektorokat használják. A kiemelt, detektált élek összekapcsolásához gyakran további feldolgozásra van szükség. Ezek az eljárások sok esetben bonyolultak, és globális (az egész képen) és helyi (a kép egyes részein) eljárásokra oszthatóak fel. Az éldetektást az első és másodrendű deriváltak segítségével határozzák meg. Az első derivált, amelynek jelentősége van a képszegmentációban a gradiens, melynek a vektoros formája a következő kifejezéssel adott:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}. \quad (4.9)$$

Az így meghatározott vektor modulusa, amplitúdója a következőképpen fejezhető ki:

$$|\nabla f| = \text{mag}(\nabla f) = (G_x^2 + G_y^2)^{1/2}. \quad (4.10)$$

A gradiens vektor fő jellemzője, hogy az iránya az f maximális változásának az irányába mutat az (x,y) koordináta pontokban. A maximális változás szöge a következő kifejezéssel fejezhető ki:

$$\alpha(x, y) = \tan^{-1} \left(\frac{G_y}{G_x} \right). \quad (4.11)$$

A legfőbb problémának az mutatkozott, hogy hogyan lehet a G_x és G_y deriváltakat digitális formában meghatározni.

Később a deriváltak alkalmazásának kutatásával bebizonyosodott, hogy a másodrendű deriváltak hatékonyabban emelik ki az éleket, mint az elsőrendű deriváltak. A képfeldolgozásban a másodrendű deriváltak a Laplacian alkalmazásával határozhatók meg az alábbi kifejezések segítségével:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}. \quad (4.12)$$

Magát a Laplaciant használják az élek kiemelésére annak ellenére, hogy másodrendű deriváltaként nagyon érzékeny a zajokra, az amplitúdója kettős éleket eredményez és végül nem észlelhető vele az élek iránya. Azonban a Laplacian nagyon hatékony az élek detektálásánál más technikákkal kombinálva. Bár a kettős élek nem alkalmasak közvetlenül az élek kiemelésére, ez a másodrendű derivált hatékonyan felhasználható az élek lokalizálására.

Az éldetektálás alapötlete az, hogy meg kell találni azokat a helyeket a képben, ahol az intenzitások (fényárnyalatok) hirtelen megváltoznak a következő tulajdonságok felhasználásával:

1. meg kell határozni azokat a helyeket, ahol az intenzitás első deriváltja amplitúdó szerint nagyobb, mint egy bizonyos küszöb,
2. meg kell határozni azokat a helyeket, ahol az intenzitás második deriváltja nulla.

Ha a következő aproximációk be lesznek vezetve:

$$\nabla f \approx G_x^2 + G_y^2 \text{ vagy } \nabla f = |G_x| + |G_y| \quad (4.13)$$

és a maszk a következő formában adott,

z1	z2	z3
z4	z5	z6
z7	z8	z9

4.9. ábra: Az operátor maszk formája

abban az esetben a gradiens a következő kifejezéssel határozható meg:

$$g = \left[G_x^2 + G_y^2 \right]^{1/2} = \left\{ \left[(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \right]^2 + \left[(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \right]^2 \right\}^{1/2}. \quad (4.14)$$

Végül, az élek kiszámítása a képen konvolúcióval történik, ahol lényegében a képet kiszűrjük a gradiens operátorok maszkjaival vízszintes és függőleges irányokban. Ha az operátorokat h_1 és h_2 jelölik, és a kapott éldetektált képeket g_1 és g_2 jelölik mindkét irányban, a kifejezések a következőképpen definiáltak:

$$g_1(m, n) = \sum_i \sum_j h_1(i, j) u(m - i, n - j) \quad (4.15)$$

$$g_2(m, n) = \sum_j \sum_i h_2(i, j) u(m - i, n - j) \quad (4.17)$$

A kapott kép intenzitása a következő képlettel írható fel:

$$g(m, n) = \sqrt{g_1^2(m, n) + g_2^2(m, n)}. \quad (4.18)$$

Mindezeket a műveleteket nagyon egyszerűen meg lehet valósítani pl. a Matlab és OpenCV szoftvercsomagokban, mivel számos beépített funkció létezik ezekre a célokra. Az éldetektáláshoz több maszk (operátor) létezik és mindegyik rendelkezik speciális tulajdonságokkal. A Prewitt, a Sobel és a Canny operátorokat használják leggyakrabban. Ezek közül a Prewitt operátor az, mely leggyengébb minőséggel emeli

ki az éleket, a legjobban pedig a Canny operátor. A Sobel operátor minőségileg valahol az előző kettő között található, de a leggyakrabban alkalmazzák a gyakorlatban. Ennek fő oka az, hogy tökéletesen érzékeli a fő éleket és elnyomja a kevésbé fontos éleket, ezért alkalmas széleskörű alkalmazásra a gyakorlatban. A szakirodalomban szereplő operátorok közül néhányat az alábbi ábra mutat be:

-1	-1	-1
0	0	0
-1	-1	-1

-1	0	1
-1	0	1
-1	0	1

Prewitt

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Sobel

4.10. ábra: A Prewitt és Sobel operátorok maszk formái

A képszegmentálásának egyik leggyakoribb megközelítése a küszöb alapú szegmentálás, azaz a küszöbölés. Ennél a módszernél a szegmentálás a pixelintenzitás egy vagy több küszöbértékkel történő összehasonlításán és a pixelek két vagy több kategóriába sorolásán alapul. Például, a szegmentálás legegyszerűbb formája az, ahol a pixelintenzitást csak egy küszöbértékkel hasonlítják össze és ezt a műveletet a kép binarizálásának folyamatában használják, azaz a képalkotás csak két intenzitási szinttel rendelkezik. Gyakorlatban nagyon gyakran használják ezt az eljárást, ugyanis a bináris képek a legalkalmasabbak a különféle feldolgozásra, mert csak két intenzitás szintjük van, a fehér és fekete. Gyakran, az objektum felismerési algoritmus végrehajtásának megelőző művelete a binarizáció valamelyik küszöbölési technikával. A bináris képeken jóval könnyebb elvégezni a klasszifikációs eljárásokat, mint a többárnyalatú, vagy színes képeken.

Lényegében a küszöb egy függvény mellyel döntéshozást végeznek meghatározott tulajdosságok alapján.

Szürke skálás kép küszöb kiválasztásának egyik legegyszerűbb módja az, ha a kép képpontjainak világosság átlagához viszonyítva határozzuk a küszöböt, mert a háttértől világosabb, vagy sötétebb képpontok az előtér pontjai lesznek és fordítva. Ez az eljárás zajmentes, egyszínű háttérrel és egyszínű objektumokkal rendelkező képek esetében nagyon egyszerűen és jól valósítható meg, de a gyakorlatban, az esetek többségében nem léteznek ezek az ideális feltételek. Egy olyan küszöb meghatározó algoritmus definiálása a cél, amely egyszerű, nem igényel a képről plussz információkat és zajos képek esetében is jól működik.

A gyakorlatban számos szegmentációs küszöb létezik, a leggyakrabban alkalmazottak a globális, helyi (lokális) és az adaptív küszöbök. Az adaptív küszöbnek külön jelentősége van a gyenge minőségű, árnyékos, zajos képek szegmentációjánál, mert alkalmazásával binarizálás után nem vesznek el a lényeges információk a képből.

A küszöbölés részletes leírása a szakirodalomban található.



(a)



(b)



(c)

(d)

4.11. ábra: Az adaptív küszöbölés alkalmazásának példája: (a) bemeneti RGB kép, (b) szürke árnyalatú kép, (c) globális küszöbölés, (d) adaptív küszöbölés

A 4.11. ábrán látható a különbség a globális és adaptív küszöbölési műveletek között. A példában egy olyan bementi kép lett kiválasztva, amelyben az árnyék zavarokat okoz és behomályosítja a kép egy részét. A globális küszöb alkalmazásánál ezek az árnyékok gondot jelentenek, mert ugyanaz a küszöb van alkalmazva a teljes képre és binarizáció után a kép kevesebb részletet tartalmaz, azaz az árnyék hatása kifejezésre jut. Az adaptív küszöb esetében a küszöb értéke igazodik a képhez és ezért ebben az esetben a binarizáció után a kép több részletet tartalmaz.

4.2.1. A morfológia

Legtöbbször az objektumok azonosítása a képekben igen nehéz feladat. Az egyik leggyakoribb technikát, amelyet a keresett objektumok kiemelésére használnak morfológiai képfeldolgozásnak nevezik (a görög *morphe* szóból ered, aminek jelentése forma vagy alak). Ennél az eljárásnál a szürke képskálát szükséges átalakítani bináris képpé, amelyben minden pixel értéke korlátozva van vagy 0-ra vagy 1-re. Természetesen, a morfológiai műveletek kiterjeszthetők a szürke képskálára is ahol a szürke árnyalatok szintjein külön-külön elvégezhetők a

morfológiai műveletek. A morfológiai algoritmusok nagy része egyszerű logikai műveleteken, halmazműveleteken alapszik.

A morfológia a szegmentáció egyik legfontosabb és leghatékonyabb módszere, melynek segítségével a keresett objektum gyakran szinte tökéletesen elválasztható a háttértől. A morfológiának jelentős szerepe van az objektumok felismerésében ezért a továbbiakban az alapvető műveletek kerülnek bemutatásra

Mint korábban említve volt, a morfológia a halmazelméleten alapszik. A továbbiakban definiálva lesz néhány művelet a halmazelmélet segítségével. Legyenek az A és B halmazok a Z^2 térben az $a=(a_1, a_2)$ és $b=(b_1, b_2)$ elemekkel. Az A halmaz eltolását (transzlációját) $x=(x_1, x_2)$ -el $(A)_x$ -nek jelöljük és úgy definiáljuk, mint:

$$(A)_x = \{c | c = a + x, a \in A\} . \quad (4.19)$$

A B halmaz inverzét (refleksióját, tükrözését) a következő kifejezés definiálja:

$$\hat{B} = \{x | x = -b, b \in B\} . \quad (4.20)$$

Az A halmaz komplementjét a következő módon definiálják:

$$A^\circ = \{x | x \notin A\} . \quad (4.21)$$

Az A és B halmazok különbségét $A - B$ -vel jelölik és úgy definiálják, hogy:

$$A - B = \{x | x \in A, x \notin B\} = A \cap B^\circ . \quad (4.22)$$

Ezekre a halmazműveletkre szükség lesz az alapvető morfológiai műveletek meghatározásánál.

A morfológia két alapvető művelete a dilatáció és erózió. Ez a két művelet azért fontos, mert az összes többi morfológiai művelet ebből a két alaplóműveletből vezethető le. Ezeknek a műveletnek a kombinálása érdekes és fontos szerepet fog játszani a képek szegmentálásában, szűrésében és az alakfelismerésben is.

Az A és B halmazok dilatációját a Z^2 halmazban $A \oplus B$ -vel jelölik és következőképpen definiálják:

$$A \oplus B = \left\{ x \mid (\hat{B})_x \cap A \neq \emptyset \right\}. \quad (4.23)$$

Ez az egyenlet úgy magyarázható, hogy előbb meg kell határozni a B halmaz inverzét az origója körül, majd azt eltolni x távolsággal. Ez azt jelenti, hogy az A halmaz dilatációja a B halmazzal nem más, mint az összes x uniform (egyenletes) eltolásának a halmaza, ahol a $\hat{B}$ és A lefednek legalább egy nullától különböző elemet. Ennek alapján a következő kifejezést lehet meghatározni:

$$A \oplus B = \left\{ x \mid [(\hat{B})_x \cap A] \subseteq A \right\}. \quad (4.24)$$

A B halmazt ebben az esetben strukturált elemnek (a továbbiakban SE) nevezik. Ez nem az egyetlen definíciója a dilatációnak, de ennek a meghatározásnak a többi definícióhoz képest vannak előnyei is, ugyanis a B strukturált elemet úgy figyelik lényegében, mint egy konvolúciós maszkot és ez az operátor széleskörűen ismert. Bár a dilatáció a halmazelméleten, a konvolúció pedig aritmetikai műveleteken alapszik, a legalapvetőbb műveleti módszerek megegyeznek. Pl. a B maszk (SE) elfordítása az origó körül és annak szukcesszív eltolása megegyezik a képi műveletek tekintetében a képi konvolúció műveletével, azaz a maszk „csúsztatásával” az A halmaz (kép) felett.

A dilatáció legfontosabb következménye, hogy a művelet bővíti az eredeti képet, azaz bővíti a képben lévő tárgyak méretét. A bővítés mértéke kizárólag a SE nagyságának és alakjának kiválasztásától függ.

Az A és B halmazok erózióját a Z^2 halmazban $A \ominus B$ -vel jelölik és a következőképpen definiálják:

$$A \ominus B = \{x \mid (B)_x \subseteq A\}. \quad (4.25)$$

Ez azt jelenti, hogy az A halmaz eróziója B halmazzal megegyezik azon x pontok halmazával, amelyekben a B el van tolva x értékkel, de ugyanakkor az A halmazhoz is tartozik. Természetesen ez nem az egyetlen definíciója az eróziónak, de a dilatációnál említett alkalmassági okok miatt ez a kifejezés lett kiválasztva.

Az erózió műveletének legfontosabb következménye, hogy a folyamatlényegében szűkíti a tárgyakat a képben. Ez a szűkítés a SE alakjától és nagyságától függ.

A dilatáció és erózió egymással duális műveletek és ezt az összefüggést a halmazok komplemente és inverziója segítségével lehet szemléltetni:

$$(A \ominus B)^\circ = A^\circ \oplus \hat{B}. \quad (4.26)$$

A dilatációt és eróziót igen gyakran alkalmazzák a képfeldolgozásban. A dilatáció egyik legegyszerűbb alkalmazási területe a kisebb szakadások-lyukak áthidalása a képekben. Az erózió leggyakoribb alkalmazása a kép kisméretű, nem fontos részeinek eltörlésében nyilvánul meg. A dilatációnak és eróziónak (és a többi morfológiai műveletnek is) az egyik legnagyobb előnye, hogy közvetlenül a bináris képekre alkalmazhatóak, míg a lineáris szűrésnél szürke árnyalatokban jelennek meg az eredmények és csak ezután szükséges alkalmazni a különböző küszöbszámítási eljárásokat, hogy megfelelő és látható bináris kép legyen a végeredmény.

A gyakorlatban az eróziót és a dilatációt leggyakrabban együttesen alkalmazzák, vagy kombinálva egyéb halmazműveletekkel. Ilyen módon összetett morfológiai transzfotmációk hozhatóak létre, mint pl. a zárás és a nyitás. Mint

említve volt, a dilatáció kiszélesíti az objektumot, az erózió pedig szűkíti az objektumot. A nyitás alapvetően simává teszi a tárgyak kontúráit a képekben, javítja a keskeny repedéseket és eltörli a vékony kiemelkedéseket-kidudorodásokat. A tárgyak méreteinek tanulmányozására alkalmazzák. A zárás a kontúrák egyes részeit szintén elsimítja, de a nyitással ellentétben a keskeny repedéseket és a hosszú, vékony mélyedéseket tágítja, eltörli a kis lyukakat, kitölti a repedéseket a kontúrákban. A zárást a tárgyak közötti távolság elemzésénél alkalmazzák.

Az A halmaz nyitását a B SE-mel $A \circ B$ -vel jelölik és következésképpen definiálják:

$$A \circ B = (A \ominus B) \oplus B, \quad (4.26)$$

ami azt jelenti, hogy az A halmaz nyitása a B halmazzal nem más, mint az A halmaz eróziója a B SE-mel, aminek az eredményét dilatáció követi a B SE-mel.

Az A halmaz zárását a B SE-mel, $A \bullet B$ -vel jelöljük és következésképpen definiálják:

$$A \bullet B = (A \oplus B) \ominus B, \quad (4.27)$$

ami azt jelenti, hogy az A halmaz zárása a B halmazzal nem más, mint az A halmaz dilatációja a B SE-mel, aminek az eredményét erózió követi a B SE-mel.

A morfológiai műveletek leírásának végén megemlítjük a "bottom-hat" (BH) transzformációt. Ezt a műveletet olyan esetekben alkalmazzák, amikor a képen a textúrákat szükséges detektálni és kiemelni a későbbi alakfelismerés megsegítésének céljából. Magát a műveletet a zárás művelete és a bemeneti kép közötti különbségként definiálják:

$$h = (f \bullet b) - f \quad (4.28)$$

ahol az f a bemeneti kép, az $f \bullet b$ pedig az f kép zárása a kiválasztott b SE elemmel. Ez a morfológiai művelet az árnyék jelenlétében hatékonyan kiemeli a textúrákat és

képrészleteket, ezért érdemes olyan alkalmazásokban használni, ahol árnyékban készült rossz minőségű képeket szükséges feldolgozni. Egyébként ez a művelet a "top-hat" transzformáció megfelelője, amelyet a bemeneti kép és a nyitott bemeneti kép különbségével definiálnak. A "bottom-hat" transzformáció alkalmazásának példája az alábbi ábrán látható. Megfigyelhető, hogy ez a művelet nagyon jól kiemeli a képben a textúrákat, éleket és kereteket.



4.12. ábra: A „bottom-hat” transzformáció alkalmazása: (a) bemeneti kép, (b) „bottom-hat” transzformáció hatása

Természetesen még számos olyan morfológiai művelet létezik, amelyeket ebben a tananyagban nem részletezünk, de ezek részletes leírása és alkalmazásaiknak lehetősége megtalálható a szakirodalomban. Végül meg kell említeni ismét, hogy a morfológiai műveletek monokromatikus képeken is alkalmazhatók. A morfológiai műveletek szürkeárnyaltos képen történő alkalmazásának fő oka és előnye azokban az esetekben nyilvánul meg, amikor a tárgy, vagy a háttér fényereje kicsit változik, elsősorban árnyékok vagy a jelenet nem egyenletes megvilágítása miatt. Ilyen esetekben a kép szegmentálási küszöbének alkalmazása gyakran nagy hibákat okozhat és ezért célszerű a morfológia alkalmazása a hatékony szegmentálás céljából. A morfológia műveleteit

alkalmazzák szürkeárnyaltos képekben az orvosbiológiai képek elemzésében és az ipari folyamatok irányításában.

A későbbi fejezetekben részletesen is bemutatást nyer a morfológiai műveletek alkalmazása és hatékonysága, konkrét objektumfelismerési példákban.

4.2.2. A textúrák kiemelése

A kép textúrája valójában nem más, mint a képfeldolgozás során kiszámított metrikák összessége, amelynek célja a kép érzékelt textúráinak számszerűsítése, azaz detektálása. A kép textúrája információt nyújt a szín vagy az intenzitás térbeli elrendezéséről egy képben, vagy egy adott kép kiválasztott régiójában.

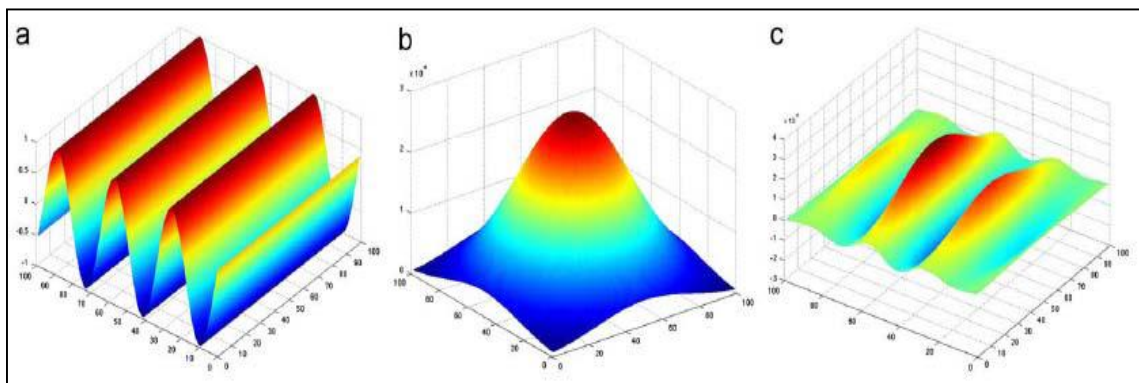
A textúrák elemzésénél számos megközelítés létezik, de ezek az elvek két fő csoportba sorolhatók: textúrák elemzése és a textúrák szintézise. A textúrák elemzésénél három fő módszer létezik: a strukturált, a statisztikai és a Fourier analízis. A textúrák szintézisének szintén három módszer létezik: az első a Markov téren alapszik, a második a gráf elméleten, a harmadik pedig a kép analógiákon alapszik. A szintézisnél a cél kis minta esetén generálni nagyobb, reális változatait a textúráknak. A gyakorlatban a textúrák elemzése széleskörűen alkalmazott módszer a képszegmentálásnál és alakfelismerésnél, ezért ez a téma kicsit bővebben lesz elemezve.

A strukturált elemzés azon az elven alapszik, hogy textúrákat olyan textúra elemek (texelek) összességének tekintik, amelyek valamilyen szabályos vagy ismétlődő mintában fordulnak elő. Ez a módszer hatékony abban az esetben amikor mesterséges textúrákat elemeznek, de problémák jelentkezhetnek olyan esetekben amikor a textúrákat valódi, zajos képekben kell elemezni és kiemelni. A statisztikai elemzésnél a textúrákat kizárólag szürkeárnyaltos képekből (vagy színekből) kiszámított statisztikai mérésekkel elemzik. Ez a módszer kevésbé intuitív, de minden képre alkalmazható és számítási szempontból is hatékony. Használható mint egy adott bemeneti textúra besorolására, mint a kép különböző texturált régiókra történő felosztására is. A Fourier analízis esetén az egyik gyakori felhasználási mód

szűrés alkalmazása frekvencia tartományban (mint ismeretes, a térbeli konvolúció spektrumbeli megfelelője az egyszerű szorzás). Megfelelő szűrő kiválasztásával hatékonyan detektálhatók és kiemelhetők a textúrák összetett képekből, mint pl. a 2D Gábor szűrő alkalmazása. Továbbá, a Fourier analízisnél lehetőség van elemezni a kép valós és imagináris részeit külön-külön, aminek gyakran igen komoly gyakorlati jelentősége lehet.

A továbbiakban röviden be lesz mutatva a Gábor szűrő alkalmazása a textúrák hatékony detektálására. A Gábor szűrő nagy előnye az egyszerű felépítése, alkalmazása és robusztussága, mert lényegében úgy alkalmazható, mint bármelyik szűrő, akár térbeli, akár spektrumbeli elemzéseknél.

A Gábor szűrő lényegében egy 2D szinuszjel és egy 2D Gauss-féle függvény szorzataként építhető meg.



4.13. ábra: A 2D Gábor szűrő: (a) szinuszjel, (b) Gauss-féle függvény, (c) Gábor szűrő

A 2D Gábor szűrő a következő kifejezéssel írható fel:

$$g(x, y, f, \Phi, \sigma) = Ae^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)} e^{j(2\pi f(x\cos\Phi + y\sin\Phi) + \varphi)} \quad (4.29)$$

ahol az első exponenciális tagot a Gauss-féle függvény képezi, a második exponenciális tag pedig a komplex szinuszjelet képezi. A Φ a függvény orientációját

mutatja meg, az f pedig térbeli frekvenciát jelöli és ez a két paraméter dominánsan kihat a szűrés eredményére. A frekvencia és a hullámhossz közötti összefüggés a következő kifejezéssel adott és pixelekből van kifejezve:

$$f = \frac{1}{\lambda} \quad (4.30)$$

A Gabor szűrővel a szűrés elvégezhető térbeli szűrés esetében konvolúcióval, frekvenciabeli szűrés esetében pedig szorzás műveletén keresztül.



4.14. ábra: A 2D Gabor szűrő alkalmazása textúrák detektálásában: (a) eredeti kép, (b) kiemelt textúrák

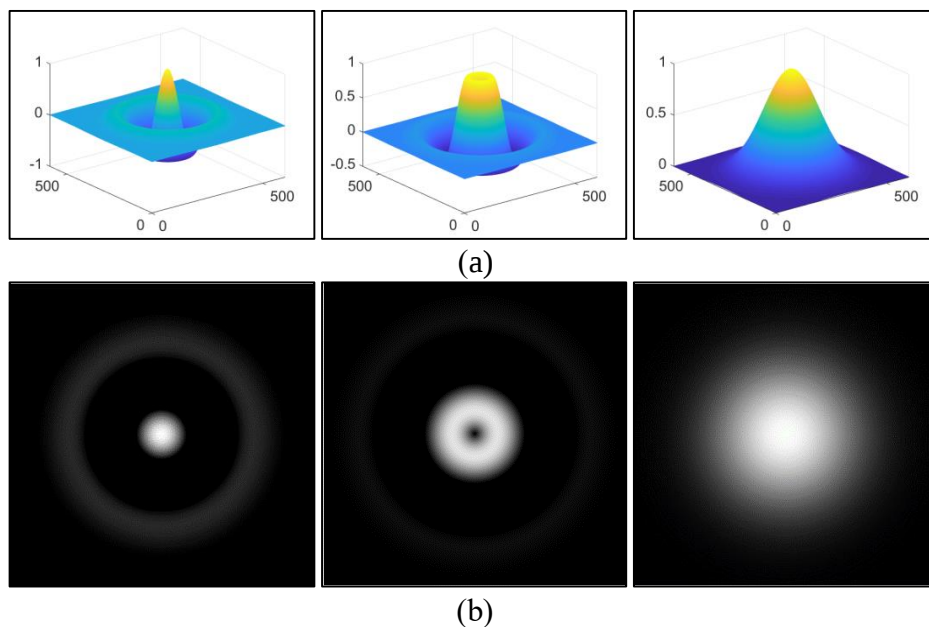
A 4.14. ábrán látható a példa, ahol a cél a rendszámtábla texturáinak kiemelése volt. Ezt az eredményt egy függőlegesen orientált Gabor szűrő eredményezte szűrés után. Mint megfigyelhető, a rendszámtábla textúrái láthatóan ki vannak emelve az összetett képből. Ez a kép igen alkalmas a későbbi feldolgozásra, amelynek célja a rendszámtábla felismerése lenne.

Fontos megjegyezni, hogy a Gabor szűrőnek létezik egy különleges formája is, a cirkuláris Gabor szűrő, amellyel körhöz hasonló alakú textúrákat, formákat lehet kiemelni a képből. Ez a szűrő eddig kevésbé kutatott, az utóbbi években került a kutatási témák előterébe. Ennek a szűrőnek az előnyei megegyeznek a klasszikus szűrőével.

A cirkuláris Gábor szűrőt a következő kifejezéssel lehet definiálni:

$$G(x, y, F, \sigma) = Ae^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)} e^{j2\pi F(\sqrt{x^2+y^2})} \quad (4.31)$$

ahol az F a központi frekvenciát jelöli és a gyakorlatban empirikusan határozzák meg. A kísérletek megmutatták, hogy a szűrő valós része hatékonyabban szűri ki a keresett textúrákat és ezért a gyakorlatban a szűrést legtöbbször csak a szűrő reális részével végzik.



4.15. ábra: A cirkuláris Gábor szűrő: (a) térbeli, (b) frekvenciabeli változatok

A 4.15. ábrán az első oszlopban a szűrő reális részei, a második oszlopban az imagináris részei és a harmadik oszlopban a modulusok láthatók. Észlelhető, hogy a szűrő reális része dominánsabban feje ki a körvonal hatását és ezért a szűrésekben a szűrő reális részét alkalmazzák gyakrabban.



(a)

(b)

4.16. ábra: A cirkuláris Gábor szűrő alkalmazása textúrák detektálásában: (a) eredeti kép, (b) kiemelt textúrák

A 4.16. ábrán látható a példa, ahol a cél a forgalmi jelzőtábla körvonal alakú textúráinak kiemelése volt. Mint megfigyelhető, a körvonal textúrái láthatóan ki vannak emelve az összetett képből annak ellenére, hogy a fényképezés pozíciója miatt a tábla alakja eltorzult elipiszis alakúvá. Ez a kép is alkalmas a későbbi feldolgozásra, amelynek célja a kör alakú alakzatok felismerése lenne.

Természetesen, még nagyszámú textúra elemzési módszer létezik, amelyek részletes leírása megtalálható a szakirodalomban.

4.2.3. Különleges szegmentálási eljárások

Ebben az alfejezetben röviden kerül megemlítésre néhány újabb, különleges képszegmentálási technika, a szegmentálási technikákkal foglalkozó fejezet kiegészítéseként. A szegmentálás problémájának megoldására általában kétféle megközelítés létezik: az ún. modell nélküli módszerek és a tudásalapú módszerek. A modell nélküli módszerek gyakran valamilyen klaszterezési eljárásan alapulnak, amelynek célja hasonló vagy azonos tulajdonságú pixelek csoportosítása valamilyen hasonlósági kritérium szerint. Másrészt, a tudásalapú módszerek azt feltételezik, hogy a megengedett megoldások száma korlátozott és olyan megoldást keresnek,

amely kompromisszumot jelent a megfigyelésen alapuló megoldások és egy adott térmodell alkalmazásával kifejezett megoldások között.

A nem paraméteres módszerek közé tartozik a „*mean-shift*“, amelyben a kimenet minőségét csak a kernelszélességet meghatározó paraméter megválasztása, a *Mumford-Shah* sorrend és annak variánsai, valamint a gráfokon alapuló módszerek határozzák meg. Mivel ezekben az algoritmusokban gyakorlatilag nincsenek feltételezések/információk az érdekelt tárgyak geometriai alakjairól, ezek az algoritmusok bizonyos kritériumok alapján elfogadható megoldásokkal állnak elő, amelyek tévesek lehetnek a kép különböző zavarai miatt, például a fényintenzitás változása, a zaj jelenléte stb.

Másrészt, a tanulás alapú módszereket a képzési szakaszban készített statisztikai modell vezérli. A legismertebbek között vannak az aktív alak modellek (*active shape models*) és az aktív megjelenés modellek (*active appearance models*), amelyekben az aktív alakot iteratív módon deformálják, hogy megfeleljen a képen látható objektum példájának. A formák annyira korlátozottak, hogy csak a képzési készletben látható módon térhetnek el a modelltől. Az alakmodelltől eltérően a megjelenési modell több információt merít a képből, például a textúrát a keresett tárgy mentén.

Ezekkel a módszerekkel ellentétben az aktív kontúrák módszere (ACM *active contoure method – snakes*), amely az ún. „*manifold enchanced*“ módszerek közé tartozik, csak a kívánt kontúra alakjának előzetes ismeretét igényli. Az ACM-ek célja a deformálható modell hozzáigazítása a képhez, energia-minimalizálás segítségével a cél függvényében. A tanulásalapú módszerek mindkét típusának megvannak a maga korlátai. Az alakok és jelenségek aktív modelljei robusztusak a zaj hatására, míg gyengébbek azokban az esetekben, ahol az alak eltér a statisztikai modelltől. Az aktív kontúrák gyakran kompromisszumot jelentenek az említett modell által korlátozott megközelítések és a nem paraméteres módszerek között.

A továbbiakban megemlítünk egy népszerű, tanulásalapú eljárást, amelyet gyakran alkalmaznak az orvosbiológiai képek feldolgozásában, az ún. atlasz/térkép alapú módszert (*atlas-based method*). Ennek a megközelítésnek az az alapgondolata,

hogy a szegmentációt az atlasz célképre történő rögzítésével érje el, az ún. atlaszillesztéssel. Az atlasz valójában egy referencia adatbázis, amely a tényleges anatómiai struktúrákról tartalmaz információt a megfelelő koordinátákon keresztül. A kezdeti kép szegmentálását úgy kapják, hogy referenciális alapként a képregisztrálás műveletet végzik el az atlaszon/térképen, amelyet nagyszámú anatómia megfigyelés alapján kapnak. Lényegében a szegmentálás kiindulópontjaként használt atlasz/térkép jó kiválasztása nagyon fontos lépés a további szegmentálási folyamat során. Az idő folyamán kifinomultabb, több atlaszon alapuló módszereket fejlesztettek ki az alakváltozatok rögzítésére, a keresett anatómiai formákon belül.

Végül röviden ismertetjük azokat a konvolúciós neurális hálózatokat, amelyeket ma használnak a képszegmentálásra a mély tanulási módszerek esetében. A konvolúciós neurális hálózatok különleges fajta neurális hálózatok, amelyek nagyon hatásosnak bizonyultak a képek tartalmának szegmentálására és felismerésére. Ugyanis a klasszikus neurális hálózatok nagy hiányossága a kapcsolatok, „idegszálak“ nagy száma, amelyek ahhoz szükségesek, hogy egy réteg összes neuronját összekapcsolják az előző réteg összes neuronjával, tehát a nagyszámú neuron hatalmas memória mennyiséget foglal el. A konvolúciós neurális hálózatok hasonlóak a klasszikus hálózatokhoz, de kevesebb, teljesen összekapcsolt rétegük van. Minden neuron összefoglalja a bemenő értékeket súlytényezők szerint skálázva és egy nemlineáris függvényt használ a kimeneti értékek kiszámításához. A kimeneti értékeknek a kívánt értékhez viszonyított eltérését a hibafüggvény méri, amelyet a súlyozási tényezők értékének beállításával lehet minimalizálni. A klasszikus neurális hálózatoktól eltérően a konvolúciós neurális hálózatok rétegeinek neuronjai 3D adatszerkezetekbe szerveződnek, az ún. szűrőkbe és mindegyik neuron lokálisan hat a kép csak egy részére, ami hozzájárul a kapcsolatok számának csökkenéséhez. Mivel a képek osztályozásakor fontos az invariancia elérése az elmozdulás vonatkozásában, a neuronoknak a teljes képre kell hatniuk. Ezt úgy érik el, hogy a képen lévő szűrőt mozgatják, azaz konvolúciót alkalmaznak, amely egy kimenet helyett több kimenetet ad a szűrő mindegyik térbeli helyzetének esetében.

Így, egy neuron összes kimenetének halmaza egy adatréteget képvisel. Egy konvolúciós neurális hálózat konvolúciós rétegekkel rendelkezik, amelyek mindegyike egy bemeneti adatsort alakít át egy adott képet reprezentáló kimeneti adatsorozattá. A legújabb, mélytanulós algoritmusok többsége konvolúciós neurális hálózatokat használ az összetett képek szegmentálására. A későbbiekben részletesen lesznek elemezve a mélytanulós módszerek az objektum felismerési módszerek elemzésénél.

5. Az alakfelismerési módszerek

Miután egy képet régiókra vagy azok határait szegmentálnak az előző fejezetben említett módszerek valamelyikével, az így kapott szegmentált képet általában további számítógépes feldolgozásra alkalmas formává szükséges alakítani. A szegmentálást követő lépés jellemzően a bizonyos tulajdonságok, jellemzők (*features*) kinyerése, amely magából a jellemzők detektálásából (*feature detection*) és a jellemzők leírásából (*feature description*) áll. A jellemzők detektálása a jellemzők keresésére vonatkozik egy képen belül, egy régióban a képen belül vagy valamely határvonalon a képen belül. A jellemzők leírása kvantitatív attribútumokat rendel a detektált jellemzőkhöz. Például detektálhatók a sarkok egy régió belül és ezeket a sarkokat leírhatjuk tájolásuk és elhelyezkedésük alapján és ez mindkettő kvantitatív tulajdonság. A jellemzők detektálását és leírását követően a következő lépés az objektumok leírása lenne.

Az alakfelismerés, illetve objektumfelismerés egy általános elnevezés olyan képfeldolgozási technikák esetében, amely leírja a számítógépes látási feladatok gyűjteményét, amelyek magukba foglalják a digitális képek tárgyainak, alakjainak, objektumainak azonosítását.

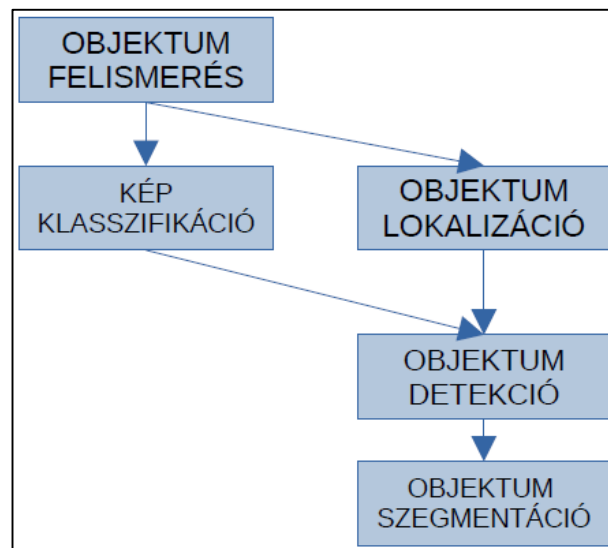
A kép klasszifikáció magában foglalja a képen lévő egy objektum klasszájának, osztályának előrejelzését, predikcióját. Az objektum lokalizáció egy vagy több objektum helyének azonosítására vonatkozik, egy képen belül. Az

objektumfelismerés egyesíti ezt a két feladatot és lokalizálja majd osztályozza az egy vagy a több objektumot a képben.

Az előbbieken felsorolt három számítógép látási feladat külön-külön is elemezhető és gyakran a kutatásokban csak az említett feladatok egy részét kutatják:

- a kép klasszifikáció feladata megjósolni a képen lévő objektum típusát vagy klasszáját.
 - Bemenet: egy kép egyetlen objektummal , például egy fénykép.
 - Kimenet: klassza címke (pl. egy vagy több egész szám, amely a klassza címkékhez van hozzárendelve).
- az objektum lokalizáció feladata megkeresni az objektumok jelenlétét a képen és behatárolni, detektálni azok lokációját a képben.
 - Bemenet: egy vagy több objektumot tartalmazó kép, például egy fénykép.
 - Kimenet: egy vagy több, a detektált objektumokat behatároló keret (*bounding box*) (pl. amelyet a pont, szélesség és magasság határoz meg).
- az objektum felismerés feladata meghatározni objektumok jelenlétét egy behatároló mezővel, valamint meghatározni a képen található objektumok típusait vagy klasszáit.
 - Bemenet: egy vagy több tárgyat tartalmazó kép, például fénykép.
 - Kimenet: egy vagy több behatároló keret (*bounding box*) (pl. amelyet a pont, szélesség és magasság határoz meg) és egy-egy klassza címke, amely minden behatároló kerethez tartozik.

A számítógépes látási feladatok további kiterjesztése a felismert objektumszegmentálás lehetne, ahol a felismert objektumokat tovább dolgozzák fel valamilyen célból.



5.1. ábra: A számítógépes látás objektum felismerési feladatainak ábrázolása

Az emberek rendelkeznek a legfejlettebb objektumfelismerő képességekkel az ismert biológiai világban. Ezzel szemben a jelenlegi objektum felismerő gépek és algoritmusok képességei elhalványulnak az ember által rutinszerűen elvégzett feladatokhoz képest, kezdve az összetett képek jelentésének értelmezésétől, az agyban tárolt ismeretek általánosításának képességéig. Azonban, az alakfelismerő algoritmusok fontos, néha döntő szerepet játszanak a mindennapi életben. El kell képzelni, milyen lenne a modern élet vonalkódokat olvasó, banki csekkeket feldolgozó, az előállított termékek minőségét ellenőrző, az ujjlenyomatokat olvasó, a leveleket válogató és a beszédet felismerő berendezések és algoritmusok nélkül.

Az objektum felismerési algoritmusoknak általánosan négy fő részük van:

1. érzékelés, képrögzítés,
2. képfeldolgozás, képszegmentálás,
3. jellemzők kiemelése, detektálása,
4. klasszifikáció, osztályozás.

A képfeldolgozás szempontjából az érzékelés nem más, mint a térbeli 2D, vagy 3D formátumú jelek, képek előállítása.

Az objektum felismerési módszerek két fő kategóriába sorolhatók:

- elméleti, döntéshozáson alapuló és
- strukturált.

Az első kategória olyan mintákon alapszik, amelyek olyan kvantitatív jellemzőket használnak, mint a hosszúság, a terület, a textúrák stb. A második kategória pedig a legjobban ábrázolt, kimutatott mintákon alapszik, mint szimbolikus információ, például a karakterláncokon (*string*) és tulajdonságaikon, valamint a közöttük fennálló kapcsolatokon. Az objektumfelismerés központi témája lényegében a kiválasztott mintákból történő „tanulás” fogalma. A továbbiakban részletezve lesz az objektum felismerés néhány módszere, mint pl.

- karakterláncok illesztése (*string matching*)
- alakzatok szám szerinti illesztése (*matching shape numbers*)
- a minták egyeztetése a minimális távolság becslésével
- korreláció
- optimális statisztikai klasszifikátorok
- SIFT algoritmus
- SURF algoritmus
- MSER algoritmus
- neurális hálózatok
- mélytanulós módszerek
- fuzzy logika alkalmazása.

A folytatásban röviden ismertette lesz néhány alapfogalom. A *minta* (*pattern*) a *deszkriptorok* (*descriptors*), azaz a leíró jellemzők elrendezését képezi. A *vonást* (*feature*), azaz valamilyen leírható tulajdonságot gyakran használják a deszkriptorok jelölésére. A *minta osztály*, *minta klassza* (*pattern class*) egy olyan minták családját, összességét jelzi, amelynek vannak közös tulajdonságaik. A minta osztályokat, klasszákat $\omega_1, \dots, \omega_W$ jelölik, ahol a W a klasszák számát jelzi. A mintafelismerés (*pattern recognition*) azon a technikán alapszik, amely alapján a

mintákat automatikusan hozzárendelik a saját osztályaikhoz, a lehető legkevesebb emberi beavatkozással.

A két fő mintaelrendezés a vektorok (a kvantitatív leírásnál) és a karakterláncok (string) / fa topológia (a strukturált leírásnál). A vektorokat $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ -vel jelölik és ezeknek a vektoroknak a méretei $n \times 1$:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Az $\mathbf{x}$ vektor komponenseinek tulajdonságai attól a megközelítéstől függenek, amivel a fizikai mintát leírni, jellemezni szeretnék. Egyes alkalmazásokban a minta tulajdonságainak leírását legjobb elvégezni valamilyen strukturált reláció segítségével.

5.1. A strukturált módszerek

A minta vektorok (*pattern vectors*) alkalmazása nem megfelelő olyan alkalmazásokban, amelyekben az objektumok a strukturált jellemzőkön keresztül vannak leírva, mint pl. a szimbólumokból alkotott karakterláncok, „sztringek”. Ennek ellenére, a képfeldolgozási algoritmusokban sokkal kevesebbet kerülnek felhasználásra, mint az elméleti módszerekben alkalmazott minta vektorok. A strukturált módszerek meghatározóak olyan objektumok felismerésénél, ahol az alak, azaz a forma a jelentős. Tehát ezeknél a módszereknél a strukturált összefüggések közötti kapcsolathelyeződik előtérbe.

5.1.1. Az alakzatok szám szerinti illesztése (*Matching Shape Numbers*)

Ez a módszer hasonlít a minimális távolság meghatározásán alapuló elvhez azzal, hogy ebben az esetben a régiók, alakzatok határainak (határvonalainak)

összehasonlítását kell elvégezni a számuk alapján. Ebben az esetben a *hasonlóság mértéke*, *foka* a két alakzat határai között meghatározott legnagyobb rend, amelyre a két alakzatszám egybeesik. Pl. legyenek az a és b két egymáshoz közeli alakzat számai, amelyek 4-irányú lánckódokkal vannak reprezentálva. E két alakzatnak a k hasonlósági foka akkor adott, ha érvényes:

$$\begin{aligned} s_j(a) &= s_j(b) & \text{ha } j &= 4, 6, 8, \dots, k \\ s_j(a) &\neq s_j(b) & \text{ha } j &= k + 2, k + 4, \dots \end{aligned} \quad (5.1)$$

ahol az s az alakzatszámát, az index pedig a rendet jelzik. A két alakzat közötti távolságot a hasonlósági fok reciprokjaként definiálják:

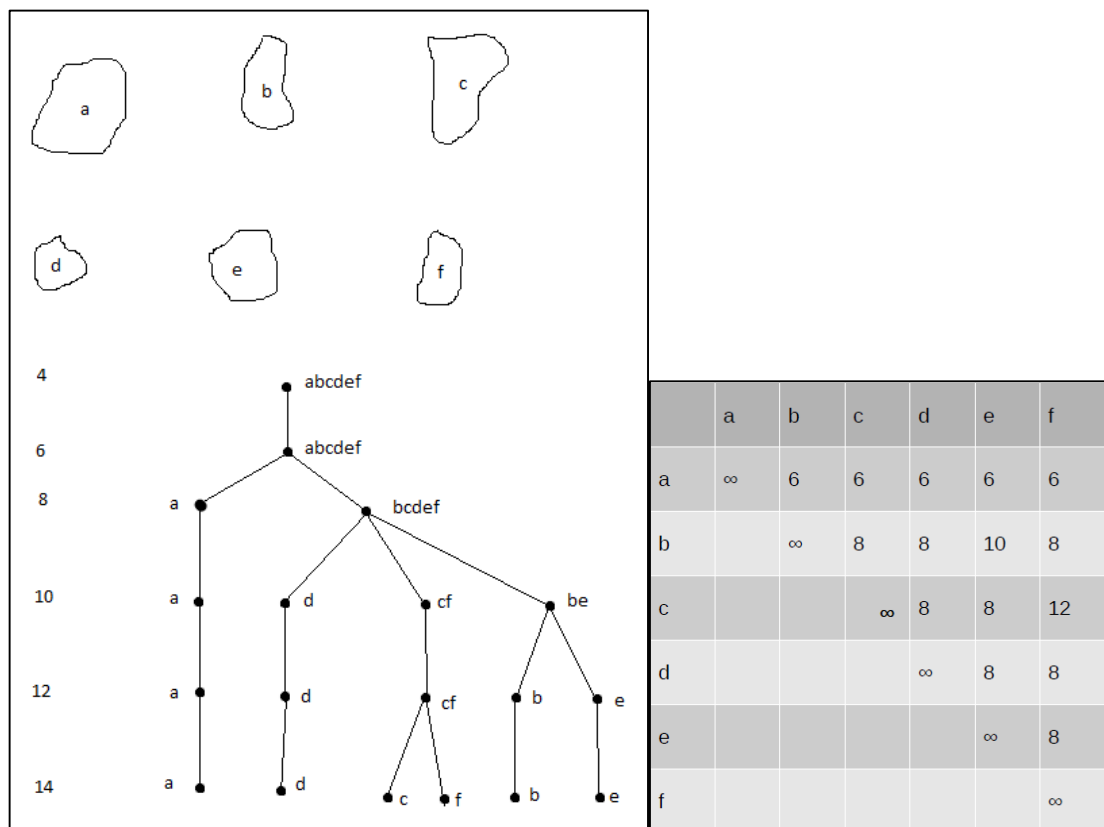
$$D(a, b) = \frac{1}{k} \quad (5.2)$$

Ez a távolság kielégíti a következő feltételeket:

$$\begin{aligned} D(a, b) &\geq 0 \\ D(a, b) &= 0 \quad \text{ha } a = b \\ D(a, c) &\leq \max[D(a, b), D(b, c)] \end{aligned} \quad (5.3)$$

Az összehasonlításokban egyaránt használható a k és a D is. Abban az esetben, ha a k hasonlósági fok van alkalmazva az összehasonlításra, a k nagyobb értéke arra utal, hogy az alakzatok, formák jobban hasonlítanak. Elvileg, ha a k értéke végtelen, az azt jelenti, hogy a két összehasonlított alakzat teljesen megegyezik. Abban az esetben, ha a D van használva az összehasonlításokban, minden ugyanúgy érvényes, mint a k esetében, csak fordítva.

A következő példában adott egy f alakzat, forma, a cél pedig meghatározni, hogy melyik alakzat hasonlít hozzá legjobban az a, b, c, d, e formák közül. Az 5.2. ábrán láthatók az adott alakzatok és a teljes illesztési folyamat.



5.2. ábra: Az alakzatok szám szerinti illesztése

Ez az illesztési feladat analóg azzal, hogy adott az 5 alakzat prototípusa és meg kell határozni, hogy melyikhez hasonlít legjobban az adott ismeretlen alakzat. A megoldás fa struktúrán keresztül van levezetve, ahol a „gyökér” (*root*) a legkisebb hasonlósági fokú alakzatnak felel meg, aminek az értéke ebben az esetben 4. Legyen feltételezve, hogy a 8-as értékig az összes alakzat egyenlő kivéve az a alakzatot, amelynek a hasonlósági foka végtelen. Haladva lefelé a fa struktúrán, a d forma hasonlósági foka 8 a többihez képest és így tovább. Az f és c alakzatok egyedülállóan megegyeznek, mert a hasonlósági fokuk nagyobb, mint bármelyik másik két alakzaté.

5.1.2. A karakterláncok illesztése (*String Matching*)

A karakterláncok, sztringek illesztését gyakrabban alkalmazzák az alakfelismerési alkalmazásokban. Ebben az esetben is szükséges használni valamilyen hasonlósági kritériumot, mint pl. a távolság alkalmazását.

Legyen a két régió a és b -vel jelölve, a határvonalai pedig legyenek kódolva a következő karakterláncokkal: $a_1, a_2, \dots, a_n$ és $b_1, b_2, \dots, b_n$ sorban. Legyen az α az illesztések, megegyezések száma az adott karakterláncok között, ahol az illesztés a k -dik pozíción történik abban az esetben ha érvényes a következő egyenlőség: $a_k = b_k$. Azoknak a szimbólumoknak a száma, amelyek nincsenek illesztve, azaz nem egyeznek meg, adott a következő kifejezéssel:

$$\beta = \max(|a|, |b|) - \alpha \quad (5.4)$$

ahol az abszolút jelben lejelölt argumentum $|arg|$ a sztring hosszát (a szimbólumok számát) jelöli az adott argumentumban. Bebizonyítható, hogy $\beta = 0$ abban az esetben amikor az a és b sztringek, azaz az a és b régiók megegyeznek.

Az a és b sztringek közötti hasonlóság mértékét a következő kifejezéssel lehet meghatározni:

$$R = \frac{\alpha}{\beta} = \frac{\alpha}{\max(|a|, |b|) - \alpha} \quad (5.5)$$

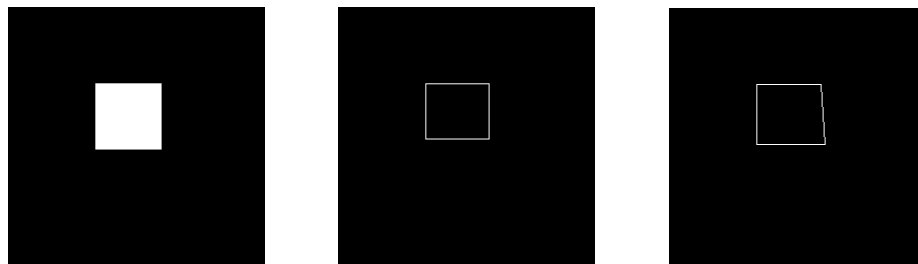
Az R értéke végtelen abban az esetben, ha az illesztés, megegyezés a két sztring között teljes, azaz tökéletes, és $R=0$ abban az esetben ha egyetlen szimbólum sem egyezik meg a két a és b sztringek szimbólumai között (ebben az esetben az $\alpha=0$).

Mivel az illesztést szimbólumonként kell elvégezni a karakterláncok között, ezért fontos meghatározni mindkét határvonal (sztring) esetében a kezdőpontokat, a

feldolgozási idő csökkentése miatt. Továbbá, mivel a karakterláncokban hasonló szimbólumok illesztését kell elvégezni, fontos regisztrálni, rögzíteni őket a pozícióktól függetlenül, hogy a módszernek értelme legyen. A regisztrálás egyik egyszerű módja az egyik sztring elmozgatása a másikhoz képest mindaddig, amíg nem lesz elérve az R maximális értéke. Ez és az ehhez hasonló módszerek a különféle sztringműveletek segítségével végezhetőek el.

A karakterláncok illesztése két rövid példán keresztül kerül bemutatásra. Mindkét példában az első kép az eredeti alakzatot mutatja be, amely segítségével lesz elvégezve a kísérlet. A második képen az eredeti alakzat kontúrája látható, ami a külső szögek kiszámításának segítségével lett meghatározva. A harmadik képen látható a második kép zajos, eltorzult változata. Az illesztési folyamat elején előbb kódolni kell a megfelelő sztringeket, azaz sztringbe kell alakítani a második és harmadik képen található alakzat határvonalait, amit az alakzatok belső szögeinek kiszámításával végeznek. Ezt később a szögek kvantizációja követi, amellyel egy olyan karakterlánc lesz előállítva, ami tartalmazza a számokat 1-8 között következőképpen: az 1-nek megfelel a $0^\circ \leq \theta \leq 45^\circ$ intervallum, a 2-nek megfelel a $45^\circ \leq \theta \leq 90^\circ$ intervallum és így tovább. A végén az egész számokból álló sztringet karakterekből álló sztringbe kell alakítani. Az eljárásban a sztringek összehasonlítása következik, azaz az illesztés folyamata. Ennek az eredménye egy valós szám (R), amely kifejezi a két sztring közötti illesztés mértékét.

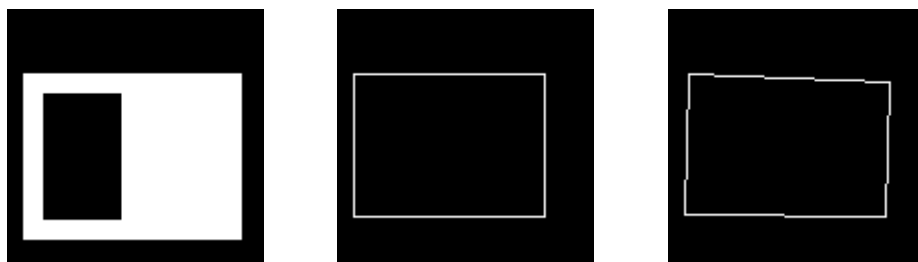
Az 5.3. ábrán látható példában az illesztés mértéke $R=3$. Ez elég jó eredménynek számít és az ábrán is látható, hogy a második és harmadik kép határvonalai emberi szemmel figyelve is jól megegyeznek, kivétel csak a jobboldali egy függőleges oldal.



(a) (b) (c)

5.3. ábra: Az alakzatok karakterláncos illesztése

Az 5.4. ábrán található példában a második és harmadik kép határvonalai nagyban eltérnek, amit a fokozottabb zajhozadással lehet elérni. Ebben az esetben is az illesztés folyamata ugyanaz, mint az előző példában. A példa lényege bemutatni azt az esetet, amikor az illesztés mértéke gyengébb. Ebben a példában az $R=1$, ami az előző példához képest várható eredmény volt, hiszen szemmel látható a nagyobb fokú határvonal eltérés.



(a) (b) (c)

5.4. ábra: Az alakzatok karakterláncos illesztése

5.2. Az elméleti, döntéshozáson alapuló módszerek

Ezek a módszerek döntéshozási függvényeken alapulnak. Legyen az $\mathbf{x} = [x_1, \dots, x_n]$ a minták n -dimenziós vektora. A W számú klasszák $\omega_1, \dots, \omega_W$, mintáinak esetében az objektumok elméleti döntéshozáson alapuló módszerrel történő felismerésében a fő probléma meghatározni a W számú $d_1(\mathbf{x}), \dots, d_W(\mathbf{x})$ döntéshozási

függvényt úgy, hogy érvényes legyen a tulajdonság, ha az $\mathbf{x}$ minta egy adott ω_i klasszához, osztályhoz tartozik, akkor a következő érvényes:

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad j = 1, 2, \dots, W: j \neq i. \quad (5.6)$$

A *döntés határát*, amely elválasztja az ω_i osztályt az ω_j osztálytól, az $\mathbf{x}$ értékei adják meg, amelyekre érvényes, hogy:

$$d_i(\mathbf{x}) = d_j(\mathbf{x}). \quad (5.7)$$

A döntéshozási függvények megkeresése olyan paraméterek becslését vonja maga után, amelyek leírják a számunkra érdekes klasszákat, osztályokat. A paraméterek becsléséhez használt mintákat *edzési*, *tanítási*, azaz *tréning (training) mintáknak* vagy *tréning halmazoknak* nevezik. A becsléshez nem használt mintahalmazokat egy adott felismerési megközelítés teljesítményének tesztelésére használják és *teszt-* vagy *független halmazoknak* nevezik őket.

A prototípus egyeztetés magába foglalja az ismeretlen minta és a prototípus készlet összehasonlítását, ahol az ismeretlen mintához hozzárendelik azt a prototípus osztályt, amely az ismeretlennel „hasznló” leginkább. Minden prototípus egyedi mintaosztályt képvisel, de osztályonként több prototípus is létezhet. Ennek a legegyszerűbb módja a minták egyeztetése a minimális távolság becslésével, azaz az Euklidészi távolság becslésével, egy adott ismeretlen vektor és a prototípus vektorok között. A módszer lényege meghatározni a legkisebb távolságot a döntéshozás céljából.

5.2.1. A minták egyeztetése a minimális távolság becslésével

Feltételezve legyen, hogy az ω_j minták minden osztályát, klasszáját az $\mathbf{m}_j$ átlagérték vektorával írják le. Így az egyes tréning vektor populációk átlagérték vektorait használják a vektorosztályok leírására:

$$m_j = \frac{1}{N_j} \sum_{x \in \omega_j} x_j = 1, \dots, W, \quad (5.8)$$

ahol az N_j a ω_j klasszatréning mintáinak a száma és az összegzését ω_j minták szerint végzik. A W a mintaklasszák számát jelöli. Az ismeretlen $\mathbf{x}$ mintavektor valamely klasszához való kapcsolódását, hozzátartozását úgy határozzák meg, hogy hozzárendelik a legközelebbi prototípus klasszájához. Ha az Euklideszi távolságot használják a közelség mértékeként, akkor a probléma megoldható egyszerűen a távolság kiszámításával:

$$D_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}_j\| \quad j = 1, \dots, W \quad (5.9)$$

Ezután az $\mathbf{x}$ vektort akkor rendelik az ω_i osztályhoz ha $D_i(\mathbf{x})$ a legkisebb távolság. Így a legkisebb távolság mutatja meg a legjobb összepárosítást, illesztést ebben a formulációban.

A minimális távolság becslési módszer akkor működik jól, ha az átlagértékek közötti távolság nagy, ha viszonyítják a véletlenszerű szóródást az egyes klasszák átlagértékeihez képest. Azonban, ha maga a bemeneti kép rossz minőségű, zajos stb. lenne és rajta a részletek nem lennének jól láthatók, ez a módszer nem lenne alkalmas a megfelelő klasszifikáció elvégzésére. Ez nem is meglepő, mert a módszer a legelső kifejlesztett módszerek közé tartozik és az alkalmazási területe is korlátozott. Leginkább akkor alkalmazzák, amikor kisszámú, jól kiemelhető alakot, objektumot kell felismerni, különösen ellenőrzött feltételek között.

Ennek a módszernek egyik alkalmazása pl. az Amerikai Bankok Egyesülete E-13B (The American Bankers Association E-13B fontcharacter set) karakter felismerő szabványának és alkalmazásának a kidolgozása volt, ahol a cél banki automata szkennerek által felismerni a specifikáció által előírt 13 karaktert. A leolvasás előtt a banki szkennerek a hivatalnok beszkenne a csekket és ezután beindul a leolvasás

folyamata, azaz a karakterek felismerésének művelete. Természetesen, ez a feladat egyszerű, mert szigorúan ellenőrzött feltételek között történik a karakterfelismerés és a rendszer nincs kitéve semmiféle lehetséges nehézségnek (zajok, árnyékolás stb.). Ha véletlenül sérülés lenne a csipkén, az már a szkennelés előtt észrevehető és magára a szkennelésre már nem kerülne sor.

5.2.2. Illesztés korrelációval

A korreláció az egyik alapvető művelet a képfeldolgozásban és igen egyszerűen elvégezhető. Lényegében a korreláció és a konvolúció közötti különbség csak egy előjelben van.

Egy adott $f(x,y)$ kép esetében a korreláció lényegében detektálja, megtalálja a kép azon helyeit, amelyek megfelelnek az adott $w(x,y)$ alképnek, azaz szűrőmaszknak. Általában az $w(x,y)$ mérete sokkal kisebb, mint az $f(x,y)$ kép mérete. Az egyik megközelítés a $w(x,y)$ térszűrőként történő kezelése, ami azt jelenti, hogy ki kell számítani a szorzatok összegeit az összes w az f képen belüli pozíciók esetében, hasonlóan, mint a térbeli szűrésnél, konvolúciónál. Ezután a $w(x,y)$ -nek az $f(x,y)$ képen belüli „legjobb megegyezéseit, átfedéseit” kell meghatározni, ami nem más, mint a kapott értékek maximális fényintenzitásai a kapott korrelációs képen. Abban az esetben, ha a $w(x,y)$ kis méretű, ez a térbeli megközelítés számításigényessé válik, ami gondot okozhat, főleg gyengébb hardver felépítés esetén, de ugyanúgy a valós időben működő rendszereknél is. Ezért a térbeli korreláció gyakorlati megvalósítása hardver-orientált megoldások esetében nagyon ritka.

$$c(x, y) = \sum_{s=-\infty}^{+\infty} \sum_{t=-\infty}^{+\infty} w(s, t) f(x + s, y + t) \quad (5.10)$$

Alternatív megközelítés a korreláció megvalósítása frekvenciatartományban, a korrelációs tétel felhasználásával. Legyen a $\circ$ korreláció műveletének a jele, a $*$ pedig jelölje a komplex konjugált számot. Ekkor a korrelációs tétel így írható fel:

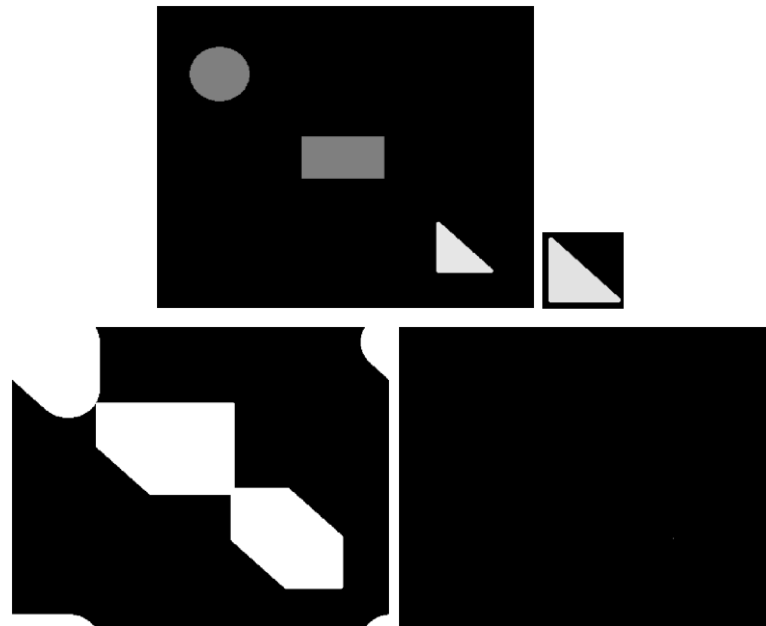
$$f(x, y) \circ w(x, y) \Leftrightarrow F(u, v) W * (u, v) \quad (5.11)$$

Mint ismeretes, a korreláció értéke azokban a pontokban a legnagyobb, ahol az f és a w egyenlők, vagy majdnem egyenlők. Tehát a (5.11) kifejezés meghatározza, hogy az f képhez hol illeszkedik legjobban a w maszk. Ennek az egyenletnek az a hátránya, hogy a korreláció eredménye érzékeny az f és w függvények amplitúdóinak változására, ingadozására. Annak érdekében, hogy az amplitúdóingadozás normalizálva legyen, a korrelációs együtthatót vesszük figyelembe az illesztésnél:

$$\gamma(x, y) = \frac{\sum_s \sum_t [w(s, t) - \bar{w}] [f(x+s, y+y) - \bar{f}_{xy}]}{\{\sum_s \sum_t [w(s, t) - \bar{w}]^2 \sum_s \sum_t [f(x+s, y+y) - \bar{f}_{xy}]^2\}^{\frac{1}{2}}} \quad (5.12)$$

ahol az összeadást az f és w közös felületein végzik, a $\bar{w}$ a maszk átlagértéke, az $\bar{f}_{xy}$ és az f átlagértéke a w maszkkal való átfedés helyén. A γ maximális értéke akkor következik be, ha a normalizált w maszk és a megfelelő normalizált felület, régió azonosak az f képen belül. Ez eredményezi a maximális korrelációt, azaz a lehető legjobb illesztést.

Az 5.5. ábrán látható a korrelációs illesztés példája. Az első képen látható egy bemeneti kép szürke árnyalatokban. A kép geometriai alakzatokat ábrázol. A második kép ábrázolja a korrelációs maszkot, ami lényegében a bemeneti kép kivágott kis részlete. Elvileg bármilyen maszk választható, akár bináris formájú maszk is. A harmadik képen látható a korreláció eredménye. Sajnos, a korreláció eredménye „szétmosódott” a szűrés effektus miatt. A legjobb illesztés meghatározása céljából a harmadik képre küszöbölést lehet alkalmazni, amivel a fényes részek közül kiválasztható a legfényesebb, azaz a legjobb illesztés helye. A negyedik képen látható a legjobb illesztés helye. Ezt a helyet a kisméretű fehér rész jelzi.



5.5. ábra: A korrelációs illesztés példája

5.2.3. Optimális statisztikai klasszifikátorok – Bayes klasszifikátor

A következőkben röviden a mintaklasszifikáció valószínűségi megközelítését említjük meg. Mint más tudományok esetében, a minták klasszifikációjánál is lehetőség nyílik a valószínűség elméletén keresztül megközelíteni a problémát, mert a valóságban a felismerni kívánt minták véletlenszerűen jelentkeznek. A módszer lényege, hogy lehetséges olyan klasszifikációs, osztályozásos megközelítést kidolgozni, amely abból a szempontból optimális, miszerint átlagosan a legkisebb hibavalószínűséget adja a klasszifikációnál.

Egy adott $\mathbf{x}$ minta valószínűsége, amely az ω_i klasszához (osztályhoz) tartozik a következő képpen van jelölve:

$$p(\omega_i/\mathbf{x}). \quad (5.13)$$

Ha a minta klasszifikátor eldönti, hogy az $\mathbf{x}$ minta az ω_j -ből származik, akkor az valójában az ω_i klasszából származik, veszteséget hoz létre, amit L_{ij} -vel jelölnek. Mivel az $\mathbf{x}$ minta bármely klasszához tartozhat a W figyelt klasszák közül, az átlagos jelentkező veszteség az $\mathbf{x}$ minta hozzárendelésekor az ω_j -hez a következő kifejezéssel írható fel:

$$r_j(\mathbf{x}) = \sum_{k=1}^W L_{kj} p(\omega_k/\mathbf{x}). \quad (5.14)$$

Ezt a kifejezést gyakran *feltételes átlagos kockázatnak*, vagy *veszteségnek a döntéshozásban* nevezik.

Alkalmazva a valószínűségszámítás alaptételét, hogy $p(A/B) = [p(A)p(B/A)]/p(B)$ az (5.14) kifejezés felírható, mint:

$$r_j(\mathbf{x}) = \frac{1}{p(\mathbf{x})} \sum_{k=1}^W L_{kj} p(\mathbf{x}/\omega_k) P(\omega_k) \quad (5.15)$$

ahol a $p(\mathbf{x}/\omega_k)$ az ω_k klasszák valószínűségi sűrűségfüggvénye, a $P(\omega_k)$ meg a ω_k klasszák jelentkezésének a valószínűsége. Mivel az $1/p(\mathbf{x})$ tag pozitív előjelű és közös mindegyik $r_j(\mathbf{x})$, $j=1, 2, \dots, W$, esetében, ezért elhanyagolható az előző kifejezésből anélkül, hogy kihatna az értékek sorrendjére a legkisebbtől a legnagyobbig:

$$r_j(\mathbf{x}) = \sum_{k=1}^W L_{kj} p(\mathbf{x}/\omega_k) P(\omega_k) \quad (5.16)$$

Így a klasszifikátornak W különböző klasszából van lehetősége kiválasztani az ismeretlen mintát. Abban az esetben, ha meglesz határozva az $r_1(\mathbf{x}), r_2(\mathbf{x}), \dots, r_W(\mathbf{x})$, minden egyes $\mathbf{x}$ minta esetében és a minta a legkisebb veszteségű klasszához lesz hozzárendelve, abban az esetben az átlagos teljes veszteség az össz döntéshez képest minimális lesz. Azt a klasszifikátort, amely minimizálja a teljes átlagos veszteséget

Bayes klasszifikátornak nevezik. Így a Bayes klaszifikátor az ismeretlen $\mathbf{x}$ mintát az ω_i klasszához rendeli akkor ha érvényes, hogy:

$$r_i(\mathbf{x}) < r_j(\mathbf{x}), j = 1, 2, \dots, W; i \neq j. \quad (5.17)$$

Tehát, az $\mathbf{x}$ akkor lesz hozzárendelve az ω_i klasszához, ha érvényes:

$$\sum_{k=1}^W L_{ki} p(\mathbf{x}/\omega_k) P(\omega_k) < \sum_{q=1}^W L_{qj} p(\mathbf{x}/\omega_q) P(\omega_q) \quad (5.18)$$

minden egyes j -re, ha $i \neq j$. A veszteséghez helyes döntés esetében nullát rendelnek hozzá, helytelen döntés esetében a veszteséghez pedig egy nem nulla értéket rendelnek, pl. 1. Ilyen feltételek alatt a veszteségfüggvény alakja a következő lesz:

$$L_{ij} = 1 - \delta_{ij} \quad (5.19)$$

ahol $\delta_{ij} = 1$, ha $i = j$ és $\delta_{ij} = 0$, ha $i \neq j$. Ha az (5.19) kifejezést visszahelyettesítjük az (5.15) kifejezésbe, a következő lesz az eredmény:

$$\begin{aligned} r_j(\mathbf{x}) &= \sum_{k=1}^W (1 - \delta_{kj}) p(\mathbf{x}/\omega_k) P(\omega_k) = \\ &= p(\mathbf{x}) - p(\mathbf{x}/\omega_j) P(\omega_j) \end{aligned} \quad (5.20)$$

Ekkor a Bayes klasszifikátor az $\mathbf{x}$ mintát hozzárendeli az ω_i klasszához, ha $i \neq j$:

$$p(\mathbf{x}) - p(\mathbf{x}/\omega_i) P(\omega_i) < p(\mathbf{x}) - p(\mathbf{x}/\omega_j) P(\omega_j), \quad (5.21)$$

vagy másképp felírva:

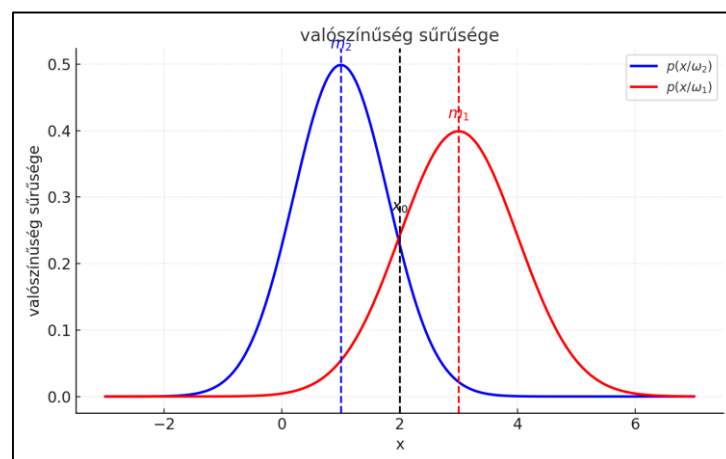
$$p(\mathbf{x}/\omega_i)P(\omega_i) > p(\mathbf{x}/\omega_j)P(\omega_j) \quad \text{ha } j = 1, 2, \dots, W; i \neq j. \quad (5.22)$$

Innen látható, hogy a Bayes klasszifikátor a 0-1 veszteségfüggvény esetében nem más, mint a következő döntéshozási függvény kiszámítása:

$$dj(\mathbf{x}) = p(\mathbf{x}/\omega_j)P(\omega_j) \quad j = 1, 2, \dots, W, \quad (5.23)$$

ahol a $p(\mathbf{x}/\omega_j)$ az ω_j klasszák valószínűségi sűrűségfüggvénye, a $P(\omega)$ meg az ω_j klasszák jelentkezésének a valószínűsége. Az előzőekhez hasonlóan a cél az összes W döntési függvény meghatározása, majd egy minta hozzárendelése ahhoz a klasszához, amelynek a döntési függvénye eléri a legmagasabb numerikus értéket.

Érdekes az az eset, amikor a sűrűségfüggvény formája a Gauss-függvény. Az N -dimenziós Gauss valószínűségi sűrűségfüggvény a következő ábrán látható:



5.6. ábra: A valószínűség sűrűsége

és a következő kifejezéssel írható fel:

$$p(\mathbf{x}/\omega_j) = \frac{1}{(2\pi)^{n/2} |C_j|^{1/2}} e^{-1/2[(\mathbf{x}-\mathbf{m}_j)^T C_j^{-1} (\mathbf{x}-\mathbf{m}_j)]}. \quad (5.24)$$

A C_j a kovariánciák mátrixa, az $\mathbf{m}_j$ pedig a minták populációjának átlagvektora az ω_j klasszából.

Mivel a logaritmus monoton növekvő függvény, a legnagyobb $dj(\mathbf{x})$ meghatározása a minták klasszifikációjához nem más, mint a legnagyobb $\ln[dj(\mathbf{x})]$ logaritmus meghatározása és akkor a döntéshozó függvény következő formája alkalmazható:

$$dj(\mathbf{x}) = \ln[p(\mathbf{x}/\omega_j)P(\omega_j)] = \ln p(\mathbf{x}/\omega_j) + \ln P(\omega_j), \quad (5.25)$$

ahol a logaritmus értéke biztosan valós szám lesz mivel a $p(\mathbf{x}/\omega_j)$ és $P(\omega_j)$ nem negatív értékűek. Az (5.25) kifejezést alkalmazva a Gauss függvényre a következő kifejezés írható fel:

$$dj(\mathbf{x}) = \ln P(\omega_j) - \frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |C_j| - \frac{1}{2} [(\mathbf{x} - \mathbf{m}_j)^T C_j^{-1} (\mathbf{x} - \mathbf{m}_j)] \quad (5.26)$$

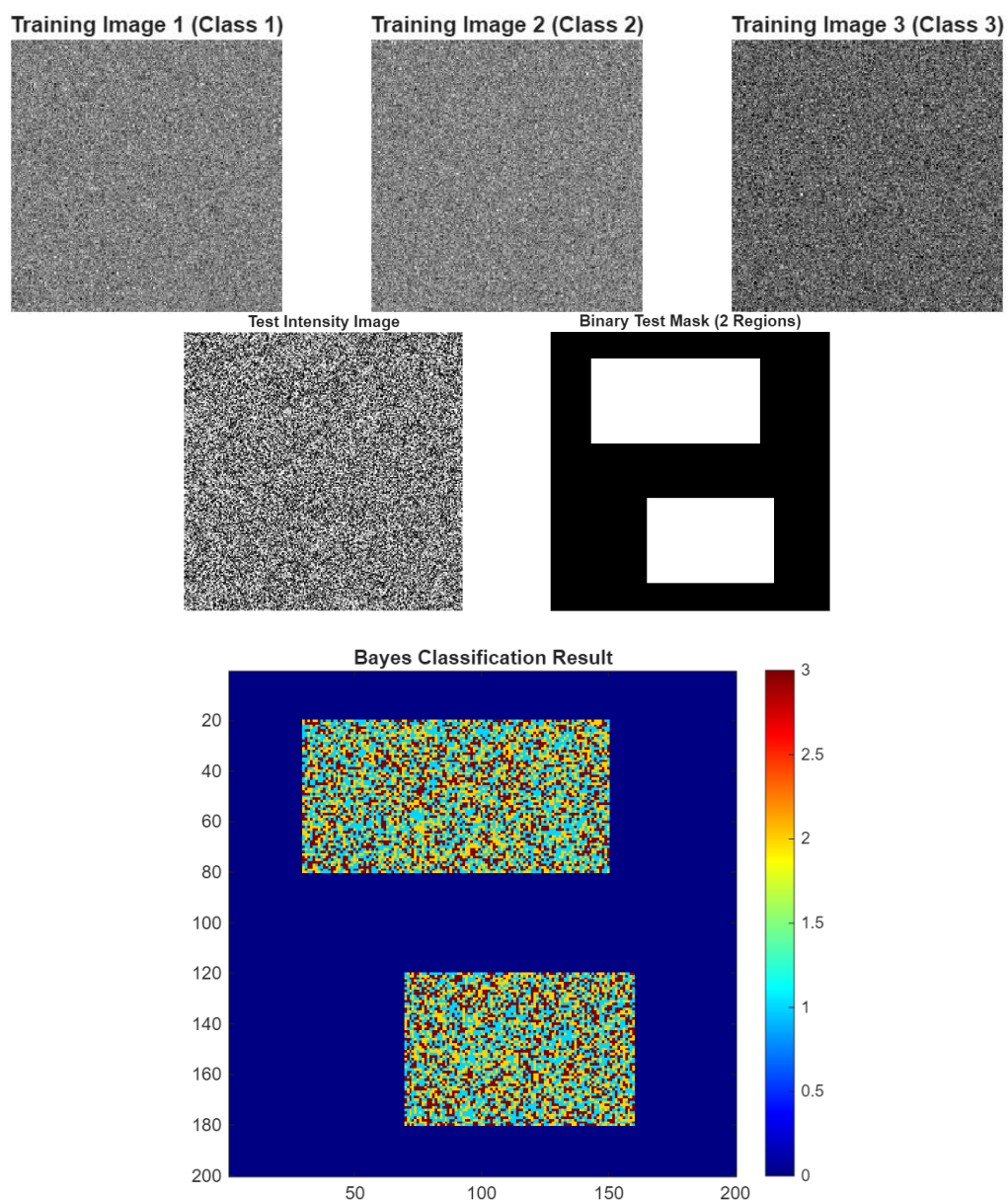
Az $(n/2)\ln 2\pi$ pozitív állandó minden egyes klassza esetében és ezért elhanyagolható is a magyarázatokban és ekkor a kifejezés alakja a következő lesz:

$$dj(\mathbf{x}) = \ln P(\omega_j) - \frac{1}{2} \ln |C_j| - \frac{1}{2} [(\mathbf{x} - \mathbf{m}_j)^T C_j^{-1} (\mathbf{x} - \mathbf{m}_j)], \quad j = 1, 2, \dots, W \quad (5.27)$$

A sarkos zárójelben lévő tagot Mahalanobis távolságnak nevezik.

A Bayes klasszifikációt leggyakrabban a régiók, felületek klasszifikálására használják a multispektrális képfeldolgozásban, mint pl. a térképeknél.

A Bayes klasszifikáció alkalmazása és hatása egy mesterségesen generált képen lesz bemutatva. Az első sorban a három kép valójában ugyanazt a képet ábrázolja, de különböző egymás utáni felvételeken. Teljesen véletlenszerűen lett kiválasztva két régió a bináris maszkon belül, amik a kép egy-egy területét fedik le. Ez látható a második sorban. A harmadik sorban a Bayes klasszifikáció eredménye látható.



5.7. ábra: Klasszifikáció példája Bayes klasszifikátorral

A harmadik sorban a téglalapokon belül a három különböző szín a három bemeneti képben található minták számát jelöli.

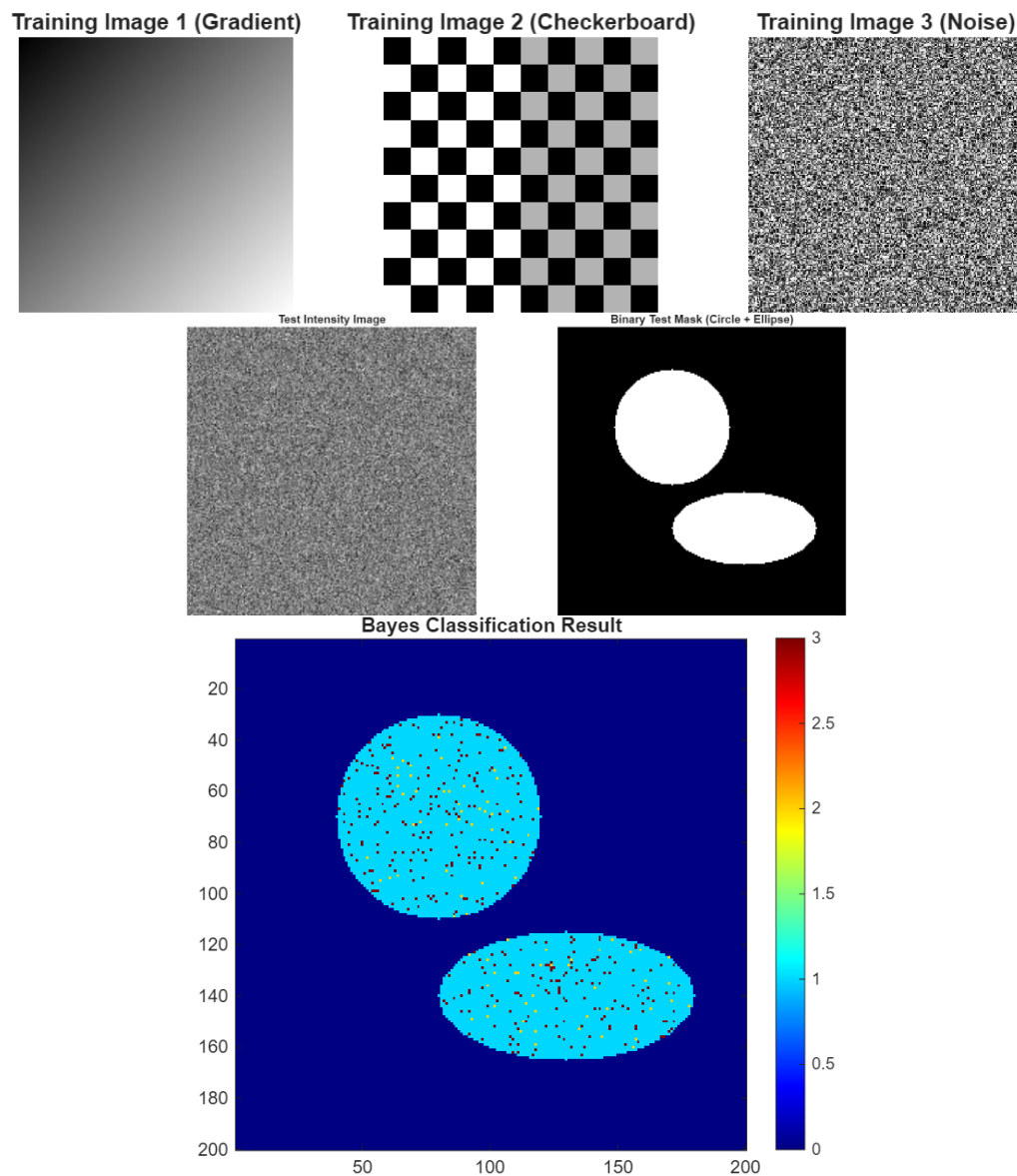
A következő táblázatban látható a minták száma, valamint a talált minták aránya. A klasszák (osztályok) száma lényegében a két kiválasztott területet jelzi.

Klasszifikáció Bayes klasszifikátorral				
Klasszák (osztályok) száma	Minták száma	Talált minták aránya az 1. képben	Talált minták aránya az 2. képben	Talált minták aránya az 3. képben
1 (nagyobb téglalap régió)	7381	39.19	29.59	31.28
2 (kisebb téglalap régió)	5551	38.07	29.45	32.48

5.1. táblázat: *A klasszifikáció eredménye Bayes klasszifikátorral*

A bemutatott példában a bemeneti képek mindegyikében 40000 minta/pixel található.

A következő példában egy olyan eset lesz bemutatva, ahol a bemeneti képek teljesen eltérők lesznek, míg a kiválasztott régiók kör és ellipszis alakúak. A példában lévő képek ismét mesterségesen lettek generálva, ismét 40000 minta található bennük. Az első sorban a három bemeneti kép található. Ismét teljesen véletlenszerűen lett kiválasztva két régió a bináris maszkon belül, amik a kép egy-egy területét fedik le. Ez látható a második sorban. A harmadik sorban a Bayes klasszifikáció eredménye látható, ismét három különböző színnel van jelölve a három bemeneti képben található minták.



5.8. ábra: Klasszifikáció példája Bayes klasszifikátorral

A következő táblázatban ismét látható a minták száma, valamint a talált minták aránya. A klasszák (osztályok) száma ismét a két kiválasztott területet jelzi.

Klasszifikáció Bayes klasszifikátorral				
Klasszák (osztályok) száma	Minták száma	Talált minták aránya az 1. képben	Talált minták aránya az 2. képben	Talált minták aránya az 3. képben
1 (nagyobb téglalap régió)	5025	92.02	0.86	4.12
2 (kisebb téglalap régió)	3915	95.35	1.00	3.65

5.2. táblázat: *A klasszifikáció eredménye Bayes klasszifikátorral*

5.2.4. A SIFT algoritmus

A SIFT (Scale Invariant Feature Transform) algoritmus olyan jellemzők detektálásán alapszik, amelyek invariánsak az elforgatásra és méretváltozásra. A SIFT algoritmus viszonylag ellenálló az affin-féle traszformációkra (torzítás 3D-ben), a fényváltozásokra és a tárgyak árnyékolására. A SIFT képes felismerni a képek jellemző tulajdonságait, amelyek az imént említett átalakítások után is megőrződnek. A kapott jellemzők ismert pozícióval, mérettel, orientációval és leíró vektorral rendelkeznek és tovább felhasználhatók a képek összehasonlításának, regisztrációjának meghatározására. Az algoritmus jellemzője, hogy nagyon alacsony az esélye a megfelelő elemek-jellemzők hibás párosításának-illesztésének, valamint az adott képen belül talált jellemzők nagy adatbázissal is illeszthetők. Ez a megközelítés nagyszámú olyan tulajdonságot generál, amelyek sűrűn lefedik a kép mindegyik fényárnyalatát és pozícióit a képen belül. Pl. egy 500x500 pixeles kép kb. 2000 stabil jellemzővel rendelkezik (ez a szám a kép tartalmától és a kiválasztott paraméterektől függ). A jellemzők mennyisége különösen fontos az objektumfelismerés szempontjából. Ugyanis egy összetett felületen elhelyezkedő kis tárgyfelismeréséhez legalább három jellemzőnek pontosan meg kell egyeznie.

Röviden leírva, a kép illesztése és felismerése érdekében a SIFT jellemzőket először kivonják a referenciaképekből, majd egy adatbázisban tárolják. Az új kép megegyezik az alapképpel oly módon, hogy mindegyik jellemzőjét egyenként hasonlítják össze az adatbázisban lévő adatokkal és a jelöltet (felismerni kívánt képet) a közös jellemzők összehasonlításával határozzák meg a leíró vektorai (jellemzők vektorai) Euklideszi távolsága alapján.

A kulcspontok jellemzői nagyon jól felismerhetők, ami nagy valószínűséggel lehetővé teszi az egyes jellemzők megfelelő párjának megtalálását a nagy adatbázisban. Azonban, túlságosan összetett kép esetében sok jellemző nem találja a saját megfelelő párját az adatbázisban, ami sok hamis illesztést eredményez a helyes illesztések mellett.

Az alapalgoritmus néhány alapvető lépésre bontható. Az algoritmus elején meghatározzák a méretarányra invariáns szélsőértékeket a Gauss szűrővel elmosott képek különbségeiből, ami lényegében a Laplace-féle éldetektor aproximációja. Ilyen módon a szélsőértékek nagyszámú halmaza lesz az eredmény, amelyből el szükséges távolítani mindazokat a jelölteket amelyek nem stabilak az alacsony kontrasztus miatt, de másrészt nagyon érzékenyek a zajokra, és azokat a jelölteket amelyeknél gyengén lettek lokalizálva az élek. Amikor meghatározásra került a jellegzetes jellemzők halmaza, következik az egy vagy több orientáció hozzárendelése mindegyik jellemzőhöz a halmazból. Az orientációt a gradiens kiszámításával határozzák meg, a figyelt jellemző környezetében, de közben biztosított az invariancia az elforgatással szemben. Az algoritmus végén mindegyik jellemzőhöz hozzárendelnek egy leíró vektort, mely biztosítja az invarianciát részben a fényváltozásokra, valamint a kamera helyzetváltozásához képest is. A leírt folyamat négy lépésbe sorolható.

1. Az első lépésben meg kell határozni azokat a képeket, amelyekben keresve lesznek a potenciális jellemzők. Elsőszámú feladatként a bemeneti kép megszűrése „elmosása” szerepel egy Gauss szűrővel, melynek standard deviációja 0.5. Ezzel eltávolításra kerül az *aliasing* (szomszédos képelemek átfedése), amely a digitalizáció folyamatában jelentkezett. A következő lépés a bemeneti kép felbontásának megnövelése lineáris interpolációval. A gyakorlatban bebizonyosodott, hogy elegendő a bemeneti képet egyszer növelni,

2 faktorról. A megnövelt képet ismét Gauss szűrővel kell szűrni, hogy az interpoláció mellékhatásai elnyomódjonak. A Gauss szűrő nem visz be új, hamis struktúrákat a képbe, tehát gond nélkül alkalmazható. Az I kép és a G Gauss szűrő L konvolúcióját a következő kifejezéssel lehet felírni:

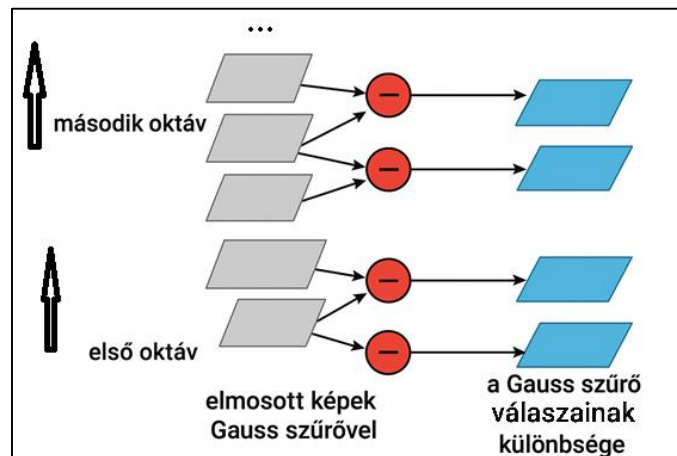
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (5.28)$$

ahol a Gauss szűrő a következő kifejezéssel van meghatározva:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (5.29)$$

Minél nagyobb a σ deviáció értéke, annál nagyobb az elmosódás a képben.

2. A második lépésben a σ értéke egy k faktorról növekszik, egy L kép lesz az eredménye amit a következő szintben kell felhasználni. A képméretetek terében a szomszédos szintek képei a k faktorról különböznek. Az összes egyenlő méretű kép az össz képméret szintjein egy *oktávot* képez. Amikor befejeződik az első oktáv képeinek szűrése minden egyes méret esetében, akkor a kép felbontását 2 faktorról csökkentik. A kép felbontásának csökkentését úgy végzik, hogy minden második képelemet kivonnak a képsorokból és képoszlopokból. A lecsökkentett felbontású kép képezi a következő oktáv alapképét, amelyet tovább szűrnek Gauss szűrővel minden méret esetében. A leírt folyamat ismétlődik minden oktáv esetében. Az oktávok és méretszintek száma a bemeneti kép nagyságától függ. A szakirodalomban javasolt az 5 méret 4 oktávval való csökkentése, amely elegendő a jellemzők sikeres detektálásához.
3. Az L szűrt képek alapján, amelyek piramist képeznek, egy új képhalmaz jön létre, amit Gaussian különbségeknek (*Difference of Gaussians – DoG*) neveznek.



5.9. ábra: A Gauss-féle piramis

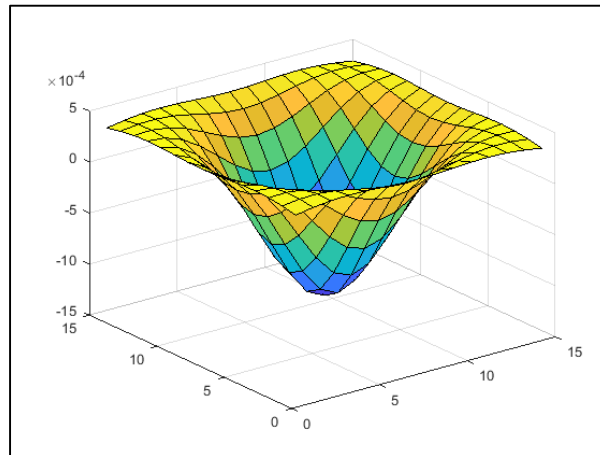
A D kép DoG képkülönbségét a Gauss-féle piramis mindegyik oktávan belül számítják a szomszédos képek kivonásával. A számuk megegyezik mindegyik oktávon belül és mindig egyel kevesebb, mint a méretek szintjei. A D kép a következő kifejezéssel írható fel:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (5.30)$$

azaz:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (5.31)$$

4. A szélsőértékek meghatározásához a méretek terében, amik egyben a legstabilabb jellemző jelöltek is a Gauss szűrés után a Laplace-féle éldetektort (*Laplacian of Gaussian – LoG*) kell használni. A következő képen látható a Laplace-féle éldetektor:



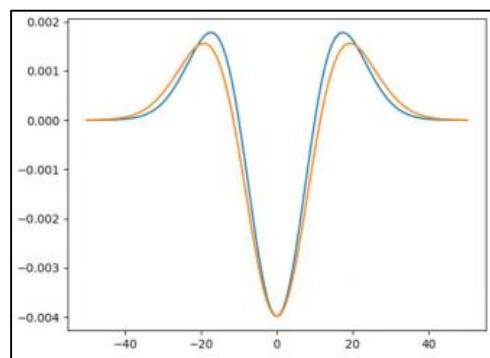
5.10. ábra: A Laplace-féle éldetektor

A LoG a szélsőérték-választ az illesztett méret környezetében határozza meg.

Az LoG a következő kifejezéssel írható fel:

$$\nabla^2 G = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} = (x^2 + y^2 - 2\sigma^2) e^{-(x^2 + y^2)/2\sigma^2} \quad (5.32)$$

Amint látható az (5.28) kifejezésből, a Laplacian esetében ki szükséges számítani a második szintű parciális deriváltakat, ami kihatással lehet a SIFT algoritmus lefutási sebességére. Ezért azokat a DoG képeket kell használni, amelyek a LoG megfelelő aproximációi, mert így az algoritmus sebessége jelentősen nagyobb lesz. Ekkor a parciális deriválás csak egy műveletre fog lecsökkeni, a két szűrt kép kivonására.



5.11. ábra: A LoG aproximációja DoG segítségével, a Laplacian kék színnel van jelölve, a DoG piros színnel van jelölve

Ahhoz, hogy bizonyítható legyen, hogy a *DoG* az *LoG* közeli aproximációja, amely normalizálva van a méretre, ki lehet használni a következő hődiffúziós egyenletet, amely a σ szerint van paraméterezve:

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \quad (5.33)$$

Mivel a $\frac{\partial G}{\partial \sigma}$ approximálható differenciál segítségével, akkor a következő egyenletet lehet felírni:

$$\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (5.34)$$

amiből következik, hogy:

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G. \quad (5.35)$$

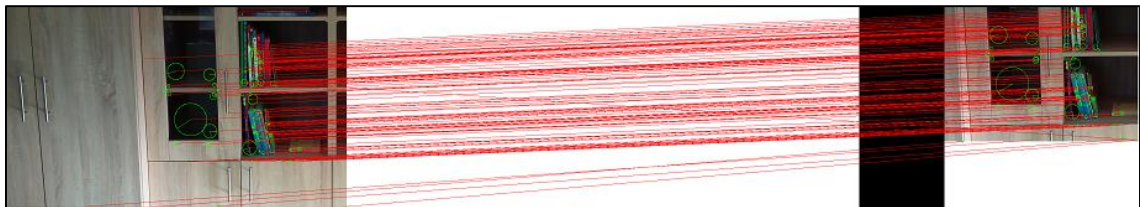
A $(k-1)$ tag az (5.31) kifejezésben állandó mindegyik méret esetében és ezért a minimum/maximum értékei ugyanazon a lokáción maradnak. A helytelen aproximáció nullába tart, ahogyan a k közeledik egyhez. A kutatások megmutatták, hogy pl. a nagyobb különbségek a méretek között nem hatnak ki a szélsőértékek detekciójának stabilitására és azok lokációjára sem.

A következő példában a SIFT algoritmus működése lesz bemutatva. A példa célja megtalálni a közös jellemzőket a két bemeneti képben. A bemeneti képek egy szekrény kisebb részét ábrázolják, két különböző módon és méretben. Meg kell jegyezni, hogy a két képben találhatók megegyező részek is, ami jól észlelhető az 5.12. ábrán.



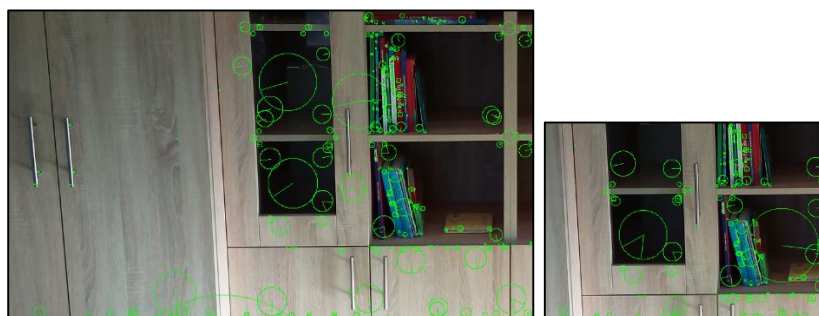
5.12. ábra: A SIFT algoritmus két bemeneti képe

A SIFT algoritmus futtatása után a két képben meg lettek határozva a megegyező pontok, ami szépen látható az 5.13. ábrán.



5.13. ábra: A SIFT algoritmus eredménye: a megegyező pontok a két bemeneti képen

A következő ábrán látható az összes detektált kulcspont mindkét bemeneti képben. Ezen kulcspontok alapján lesznek meghatározva az illesztett, közös pontok mindkét bemeneti képen. Megfigyelhető, hogy a bal oldali kép bal oldalában nincs egy detektált pont sem, ami teljesen helyes eredmény, mert a jobb oldali bemeneti képen nem látható a szekrény bal oldali része, tehát a SIFT algoritmusnak nincs is mit detektálnia.



5.14. ábra: A SIFT algoritmus eredménye: a detektált kulcspontok a két bemeneti képen

Az 5.15. ábrán láthatóak az illesztett, azaz a közös detektált pontok mindkét bemeneti képen. Az így detektált közös jellemzők később kihasználhatóak az objektum felismerési algoritmusokban.



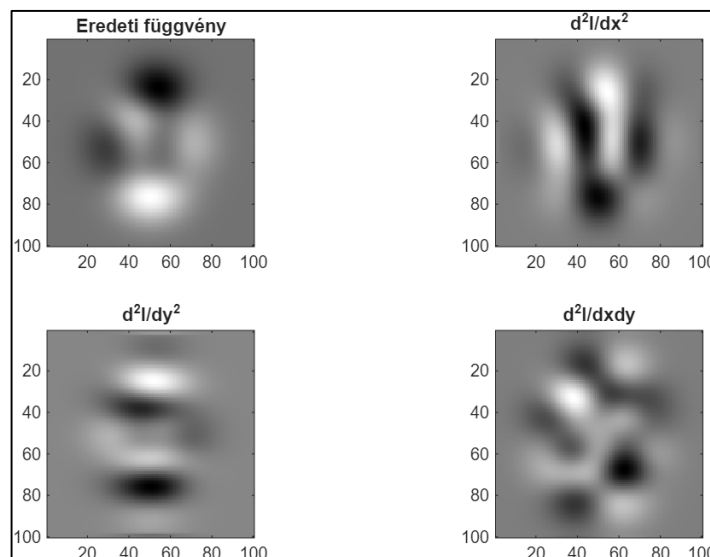
5.15. ábra: A SIFT algoritmus végeredménye: az illesztett, közös pontok a két bemeneti képen

5.2.5. A SURF algoritmus

A SURF (*Speed Up Robust Features*) algoritmus 2008-ban jelent meg és a SIFT algoritmus hátrányainak áthidalása, a hibásan detektált jellemzők csökkentése miatt vezették be. Később magának a SURF technikának is jelentkeztek különböző továbbfejlesztett változatai. Annak érdekében, hogy megfelelő teljesítménye és ésszerű számítási sebessége legyen az algoritmusnak, egyensúlyt kell teremteni az adott követelmények között, ami a fő célja a SURF algoritmus fejlesztésének. A SURF detektor fókusza a skálázás és a rotációs invariancia, mivel az ilyen típusú jellemzők, leírók jó kompromisszumot kínálnak az alapvető követelmények között. Továbbá a SURF kifejezetten robusztus az illuminancia és a különféle zajok ellen is.

A SURF algoritmus alapja a multiskaláris tér elméletén és a Hessian mátrixon alapszik és használja azok alapvető aproximációit. A SURF lényegében létrehoz egy stack-féle adattípust és eredményeként azonos méretű képeket tartalmaz. Az integrált képek használata miatt a SURF algoritmus szűri a stack-et alkalmazva a téglalap-szűrő (*box filter*) aproximációját a második rendű Gauss-féle

részleges deriváltak felhasználásával. Az integrált képek megközelítőleg állandó idő alatt teszik lehetővé a téglalap-szűrők kiszámítását. Az 5.16. ábrán a második rendű Gauss-féle részleges deriváltak az x és y irányokban láthatók.



5.16. ábra: A másodrendű Gauss-féle részleges deriváltak az x és y irányokban

A SURF előnye, hogy a konvolúció ennél az eljárásnál könnyen meghatározható az integrált képek segítségével és akár párhuzamosan, több skálán is kiszámítható. Az integrált képek lényegében egy különleges adatstruktúrát képeznek, amit a téglalap alakú mátrixban lévő értékek összegének hatékony kiszámításához használnak. Ezeket a mátrixokat legelőször Frank Crow magyarázta meg 1984-ben, majd a képfeldolgozásban és a számítógépes látásban Lewis népszerűsítette 1995-ben, végül nagy jelentősége volt a Viola-Jones-féle objektum felismerési algoritmusban.

A $T = (x, y)$ pozícióban integrált kép a T pont által határolt négyzeten belül elhelyezkedő összes pixel intenzitásának és a kép kiinduló pontjának az összegét jelenti.

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (5.36)$$

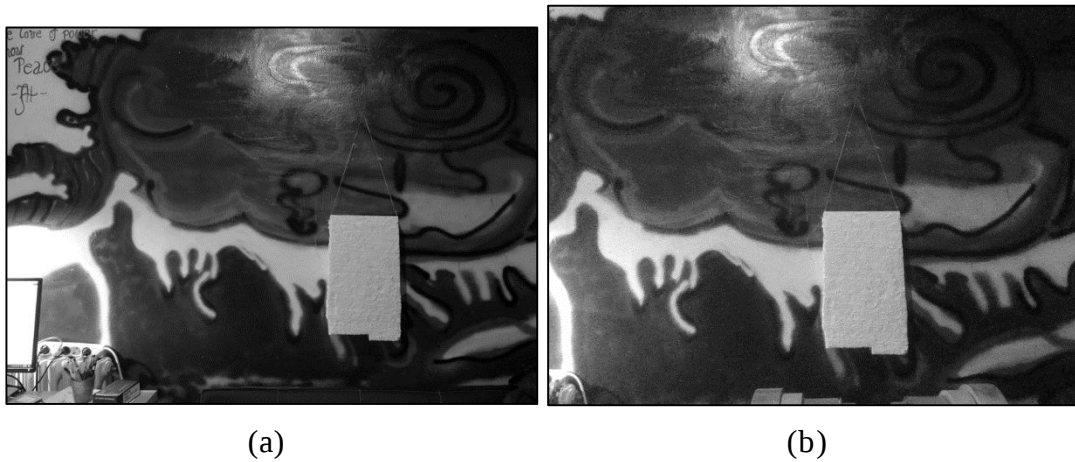
Az integrális kép kiszámítása után könnyen meghatározható az intenzitások összege is bármelyik négyzet alakú terület esetében. Ezután az említett Hessian-mátrix aproximációjára alkalmazzák a téglalap-szűrőt, ez aproximálja a Gauss-féle másodrendű deriváltat, mely nagyon gyorsan kiszámítható az integrált kép segítségével.

Mivel a kulcspontokat különböző skálakon kell észlelni, ezeket a kulcspontokat különböző skálán lévő képekben is össze kell tudni hasonlítani. Az integrált képek alkalmazása miatt a SURF algoritmussal nem szükséges minden egyes skálát külön szűrni, “elmosni” egy-egy Gauss-szűrővel, mint a SIFT algoritmusnál, hanem közvetlenül alkalmazható egy téglalap-szűrő.

A skála kezdőrétege egy 9x9 pixel méretű szűrő, amely megfelel a Gauss-féle deviáció 1.2 értékű approximációjának. A skála tér oktávokra van osztva és az egyik oktáv a kezdeti kép, a szűrőkkel történő konvolúciójával kapott szűrőválaszok sorozatát jelenti, ahol a méretek növekednek.

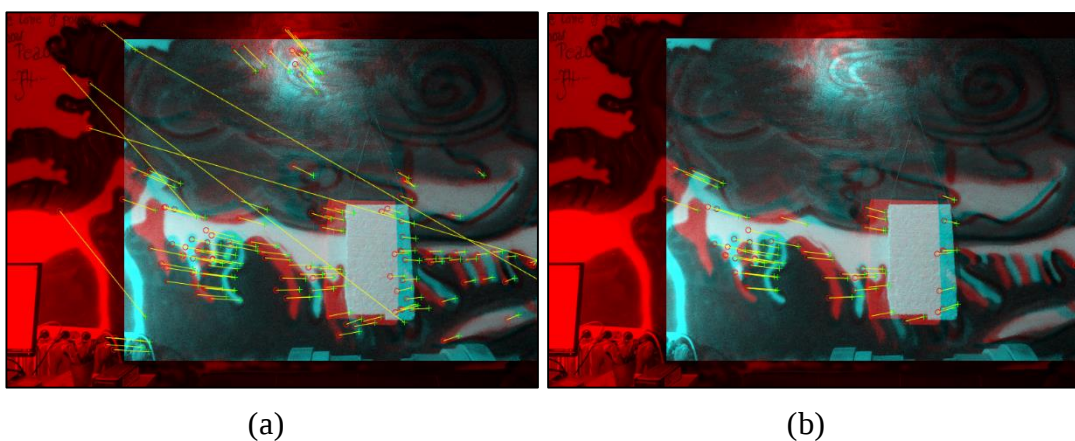
A SURF jellemző leírja az intenzitás eloszlását a kulcspontok szomszédságain keresztül. A jellemzők az elsőrendű Haar-féle hullámok vízszintes és függőleges irányú válaszai alapján készülnek egy integrált kép felhasználásával, amely hozzájárul az algoritmus sebességéhez. Az első lépés a reprodukálható rotáció rögzítése az érdekes pont körüli kör alakú régióból nyert információk alapján. Ezt követően egy négyzet alakú régiót építenek, amely igazodik a kiválasztott orientációhoz és kivonják belőle a SURF leíró, jellemzőt.

A következő példában a SURF algoritmus alkalmazása kerül bemutatásra egy képsorozaton. A példa lényege illeszteni a gyenge minőségű kamerával rögzített képet a nagyminőségű kamerával rögzített képhez, későbbi képfeldolgozás céljából. A két kép hasonló pozíciókból van rögzítve, különböző távolságról, de különböző készülékekkel. A megvilágítás minősége szerény, a helyiség árnyékolt. A fal szándékosan tarka a kísérlet nehezítésének céljából, a végső cél pedig egymáshoz illeszteni a téglalap alakú tárgyat a falon, a két bemeneti kép alapján.



5.17. ábra: A két bemeneti kép: (a) a minőséges kamerával rögzített nagyfelbontású kép, (b) a gyenge minőségű kamerával rögzített kisfelbontású kép

Az 5.17. ábrán látható a két bemeneti kép. Megfigyelhető, hogy a képek méretei és méretarányai nem egyeznek meg. Továbbá észrevehető, hogy a kisfelbontású kép világosabb és erősebb fényvisszaverődés van jelen benne. Az illesztés célja felismerni a gyenge minőségű képben a téglalap alakú tárgyat, a megfelelő jellemzők meghatározásával SURF algoritmuson keresztül, majd illeszteni azt a jó minőségű képhez, azaz megtalálni benne a téglalapot a jellemzők alapján.



5.18. ábra: A két bemeneti kép: (a) a feltételezett jellemzők, (b) a megegyező, illesztett közös jellemzők

Az 5.18. ábrán látható a jellemzők meghatározásának példája. Az első ábrán látható az összes megtalált, lehetséges jellemző, míg a második képen a közös, illesztett jellemzők vannak bejelölve. A második képen lényegében a 100%-ban megegyező jellemzők találhatók. Az apró piros köröcskék jelölik a nagyfelbontású minőséges képben talált jellemzőket, a zöld színű apró pluszok pedig jelölik a kisfelbontású képben talált jellemzőket. A vonalak amelyek összekötik a talált jellemzőket lényegében jelölik azok megegyezését, összepárosítását. A második képen szépen megfigyelhetők a közös, teljesen megegyező jellemzők, amelyek leírják ugyanazt a tárgyat mindkét képben, ez esetben a téglalap éleit és a háttérben látható alakzatok kontúráit. A második kép a RANSAC (*Random Sample Consensus*) iteratív algoritmus egyik alfajtájával, a MSAC (*M-estimator Sample Consensus*) algoritmussal lett meghatározva. Az algoritmus célja kiszűrni, eliminálni a kívülálló, távolálló (*outlier*) jellemzőket a belső, közeli (*inlier*) jellemzőktől.



5.19. ábra: Az összeillesztett kép

Az 5.19. ábra mutatja az összeillesztett képet. A fekete keret a két kép különböző méretéből ered. Amint látni, a végső képben jól láthatóak és behatárolhatóak a különböző tárgyak és alakzatok.

5.2.6. A MSER algoritmus

A MSER (*Maximally Stable Extremal Region*) algoritmus kiinduló lépése valamilyen képszegmentációs eljárás. A régiók, amelyeket a MSER algoritmus referencia pontoknak tekint lényegében azok a régiók, amelyek a legstabilabbaknak számítanak.

Az algoritmus a kép összes pixelének intenzitás szerinti rendezésével, szétválasztásával indul. A pixelsorozatot olyan küszöb segítségével választják ki, amely változtatja az értékeit a kép legkisebb intenzitásától a legnagyobb intenzitásig. Ebben a lépésben szétválasztják a pixeleket, amelyek egy adott intenzitás küszöbtől nagyobb értékűek azoktól, amelyek az adott küszöbtől kisebb intenzitás értékűek. Ily módon olyan régiók alakulnak ki a képben, amelyek a küszöb növelésével egyesülnek egymással, míg végül egyetlen régióvá alakulnak. Ebben a folyamatban minden régióhoz egyedi azonosítót, jellemzőt rendelnek. Amikor a régiók összeolvadnak, egy kisebb terület veszi át a nagyobb régió azonosítóját, jellemzőjét.

A régió stabilitását úgy definiálják, mint a régió felületének állandóságát a küszöb változás függvényében. Ha az R a figyelt régió, az $R \pm \Delta$ pedig azok a régiók amelyeket a küszöb növelésével (csökkentésével) határoztak meg Δ értékkel, akkor felírható a következő ρ metrika:

$$\rho(R, \Delta) = \frac{(R+\Delta) - (R-\Delta)}{R} \quad (5.37)$$

ahol az $|R|$ a régió területét jelöli. Ha a $\rho(R, \Delta)$ kisebb a $\rho(R+\Delta, \Delta)$ és $\rho(R-\Delta, \Delta)$ értékektől, akkor az R régió maximálisan stabilnak bizonyul, azaz referencia elemnek lesz nyilvánítva.

A MSER algoritmus hatékonysága és egyetemessége abban áll, hogy egyforma jól detektálja a foltszerű alakzatokat (*blob*) és a szabálytalan, amorf alakzatokat is. Ezenkívül a MSER algoritmus nagyon hatékony mechanizmus az alakzatok jellemzőinek leírására, mert megjegyzi a régió össz pixelét és ezzel bármilyen összehasonlítási folyamatot megkönnyít. A jellemzők leírásához leginkább a másodrendű felületi nyomatékokat használják.

A következő példában a MSER algoritmus alkalmazása lesz bemutatva röviden. Az 5.20. ábrán látható az eredeti színeskép és annak szürkeárnyaltos változata. A képen előtérben van egy forgalmi jelzőtábla, a háttérben pedig láthatók fák, fal, épületrész stb. Ebben az összetett tartalmú képen szükséges hatékonyan detektálni a táblához tartozó területeket, azaz a jellemzőket.



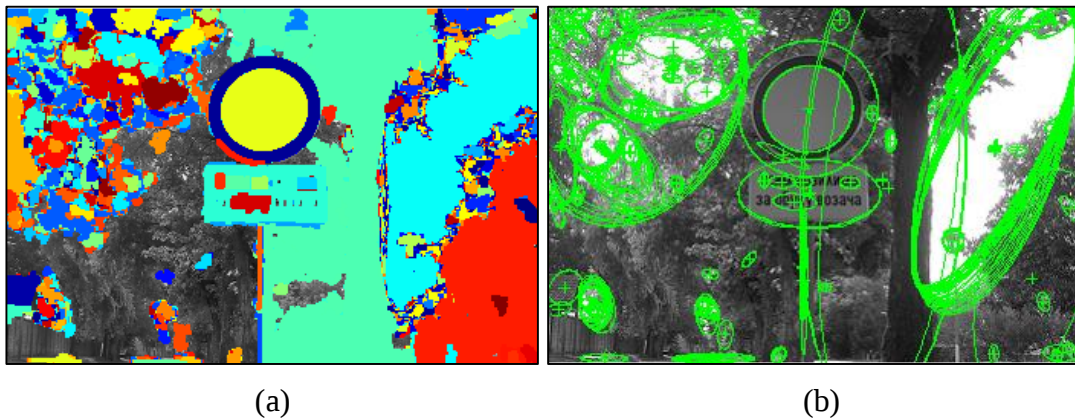
(a)

(b)

5.20. ábra: A bemeneti kép: (a) színesben, (b) szürke árnyalatokban

Az 5.21. ábrán látható a MSER algoritmus alkalmazásának eredménye. Az első képen láthatóak a detektált területek. Megfigyelhető, hogy a táblához tartozó területek nagyon jól vannak detektálva. A tábla kör alakú része sárga és sötétebb kékkel van jelölve, a tábla téglalap alakú része világos kékkel van jelölve, a területen belül pedig piros, sárga, kék stb. színekkel vannak jelölve a felirathoz tartozó részek.

A második képen a detektált területek ellipszissel vannak bejelölve, ahol a + az ellipszis középpontját jelöli. Szépen megfigyelhető, hogy a tábla környékén kisszámú, jól elhatárolt ellipszis látható, míg a háttérhez tartozó területekben számtalan ellipszis jelöli az egyéb detektált területeket. Ez a két kép később hatékonyan felhasználható az objektumfelismerési algoritmusokban.



5.21. ábra: (a) Detektált régiók a bemeneti szürke árnyalatú képben, (b) a detektált régiók megjelölése ellipszisekkel

5.2.6. A neurális hálózatok

A neurális hálózatok az információ-feldolgozó rendszerek új generációját képviselik, amelyek megmutatják a tanulás, a memorizálás és az általánosítás jellemzőit a képzésükben, tanításukban résztvevő adatok alapján. Kimutatták, hogy nagyon hatékonyak olyan feladatokban, mint az illesztés és az alakfelismerés, a függvények aproximációja, az optimalizálás, a vektorkvantálás és az adatok klaszterezése, míg a hagyományos számítógépek architektúrájuk miatt hatástalanok, különösen az alakfelismerés terén. Másrészt a hagyományos számítógépek gyorsabbak a számítási algoritmusokban és pontosabbak a számtani műveletekben.

Ennek a koncepciónak az ötlete elemi, nemlineáris számítási elemek (ún. neuronok) felhasználása, amelyek valamilyen módon összekapcsolt hálózatokként működnek, mint az agyban összekapcsolt neuronok. Az ilyen modelleket neurális

hálózatoknak nevezik. Ezeket a hálózatokat a döntési függvény együtthatóinak adaptív megszerzésének mechanizmusaként használják, mint a minták képzési, tanítási halmazainak egymást követő prezentációi.

A neurális hálózatok iránti érdeklődés az 1940-es évek elejéig nyúlik vissza egészen a napjainkig. Manapság a neurális hálózatokat a számítástechnika és az elektrotechnika minden ágában használják és ezek különösen fontosak a beszédtechnológiák jelfeldolgozásában és az objektumok felismerésekor a képfeldolgozásban.

A neurális hálózatok nagyszámú, sűrűn összekapcsolt adatfeldolgozó elemből, csomópontokból (*nodes*) állnak, amelyek általában párhuzamosan, egyidejűleg működnek és valamilyen reguláris architektúra szerint vannak rendezve, kiépítve.

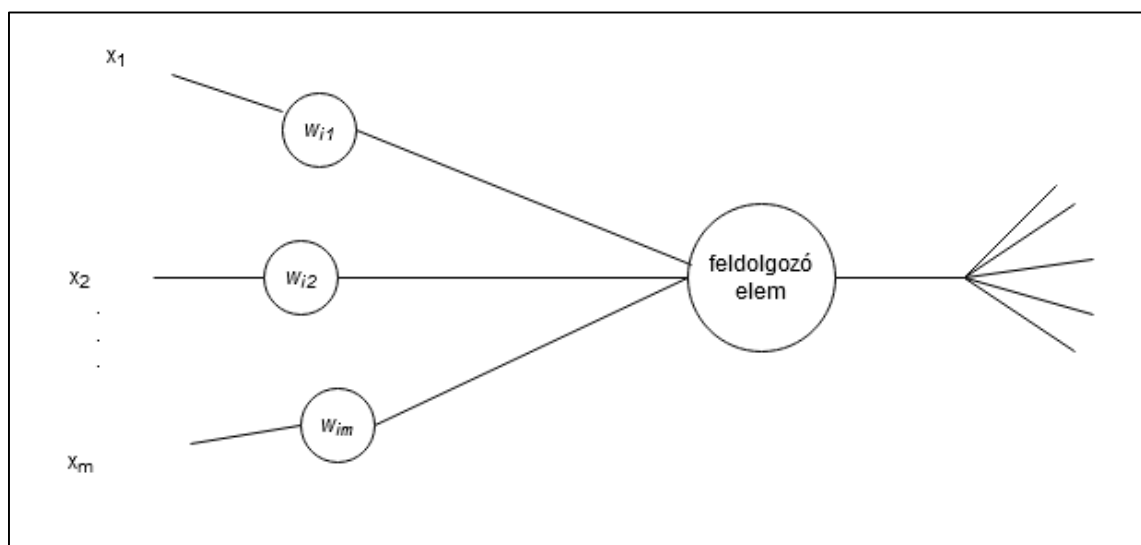
A neurális hálózatok fejlődését az motiválta, hogy valahogyan modellezni szerették volna a biológiai idegsejtek viselkedését az agyban. Ezért a „mesterséges” neurális hálózat (*Artificial Neural Network - ANN*) attribútumát nagyon gyakran hozzáadják a neurális hálózatokhoz, mert a csomópontokat, vagy feldolgozó elemeket gyakran mesterséges neuronoknak, vagy csak neuronoknak nevezik. Példaként az emberi agy megközelítőleg 10^{11} különböző típusú idegsejtből áll.

A biológiai rendszerekben a tanulás az idegsejtek axonjait és dendritjeit összekötő szinaptikus kapcsolatok szabályozásával történik. A mesterséges hálózatokban a tanulást példákkal érik el, pl. edzés az input-output adatok pontos halmazával, amelyek az algoritmust ismétléssel képezik, edzik ki, a kapcsolatok (szinapszisok) súlyozási együtthatóinak beállításával. Ezek a kapcsolatok tárolják az adott probléma megoldásához szükséges „tudást”. A legalapvetőbb formájában a perceptron megtanul egy lineáris döntési függvényt, amelyet két, lineárisan elkülöníthető képzési halmazra osztanak.

A neurális hálózatok lehetővé teszik egy adott szoftver evolúciós fejlődését, azaz a hálózat utólagosan adaptálható a valós és új feltételekhez. A neurális hálózatok képesek megváltoztatni szerkezetüket és működésüket, ellentétben a klasszikus algoritmusokkal, amelyek nem rendelkeznek ekkora rugalmassággal. A

neurális hálózat használatának egyik legklasszikusabb és legrégebbi példája a karakterfelismerő alkalmazásokban van. Az ilyen típusú karakterfelismerő rendszerek számos alkalmazás alapjai különböző területeken, amelyeket a mindennapi életben használunk, például: postahivatalok, bankok, biztonsági rendszerek stb.

Az 5.22. ábra egy McCulloch és Pitts által 1943-ban javasolt és MP neuronnak nevezett biológiai idegsejt egyszerű matematikai modelljét mutatja be.



5.22. ábra: Az MP neuron

Ebben a modellben az i -edik feldolgozóelem kiszámítja a bemenet súlyösszegét és kimenetként $y_i = 1$ vagy $y_i = 0$ jelet generál attól függően, hogy a súlyösszeg nagyobb vagy kisebb, mint valamilyen előre definiált θ_i küszöb:

$$y_i(t + 1) = a(\sum_{j=1}^m w_{ij}x_j(t) - \theta_i) \quad (5.38)$$

ahol az $a(f)$ aktivációs függvény egy egységnyi step-függvény (lépcsős függvény):

$$a(f) = \begin{cases} 1 & \text{ha } f \geq 0 \\ 0 & \text{ha } f < 0 \end{cases} \quad (5.39)$$

A w_{ij} súlyok a j neuron (forrás) és az i neuron (cél) közötti szinapszis, vagy kapcsolat erősségét jelentik. A pozitív súly a gerjesztő (izgató) szinapszist, a negatív a gátló (tiltó) szinapszist jelenti. Ha a $w_{ij} = 0$, nincs kapcsolat a két neuron között. A feldolgozóelem működését leíró egyenletben feltételezik, hogy van egy átviteli késleltetés attól a pillanattól kezdve, hogy a jel a bemenetre érkezik és eljut a kimeneti jel generálásáig, ezeket a pillanatok t_i és $t + 1$ -el jelölik. Bár egyszerű, a neuronok MP modellje hihetetlen számítási potenciállal rendelkezik. Az alapvető logikai műveleteket NOT, OR és AND is elvégezheti, ha a feldolgozási elemek küszöbértékeit megfelelően választják ki. Ez azt jelenti, hogy kellően nagyszámú ilyen neuron megfelelő kapcsolataival az univerzális számítógépek funkciója teljes mértékben megvalósulhat.

Az eddig leírtaknak megfelelően a neurális hálózat meghatározható, definiálható párhuzamos elosztott struktúraként az adatok (információk) feldolgozásához, amelyek a következő tulajdonságokkal rendelkeznek:

- 1) Ez egy matematikai modell, amelyet a biológiai idegrendszer inspirált;
- 2) Nagyszámú, egymással szorosan összefüggő feldolgozási elemből áll;
- 3) A tudást a csomópontok közötti súlyok vagy kapcsolatok tartalmazzák;
- 4) A feldolgozó elem dinamikusan reagál a bemeneti ingerre és a válasz teljesen függ a környezetében található helyi információktól;
- 5) A neurális hálózat képes egy tanulási készlet alapján tanulni, emlékezni és általánosítani;
- 6) A neurális hálózat minden részének közös viselkedése képviseli a számítási teljesítményt, ez a tulajdonság egyedül a magányos csomópontokban nem szerepel.

5.2.6.1. Az alapvető neurális hálózati modellek és a tanítási szabályok

A neurális hálózati modellt három tulajdonság határozza meg. Az első az elemek vagy csomópontok feldolgozásának modellje, a második a csomópontok és szerkezetük közötti kapcsolat, a harmadik fontos tulajdonság pedig, amely meghatározza a hálózat jellegét, az az edzésszabály (tanítási szabály), amely az egyes kapcsolatok súlyát állítja be.

Feldolgozási elemek

Az MP neuronok működése kiterjeszthető a feldolgozási elem általános modelljére. A csomópont által feldolgozott információ be- és kimeneti jelre osztható. Valójában az i -edik csomópontban a bemeneti és kimeneti jel nagyon fontos kapcsolata az f_i integrációs függvény, amely a megfigyelt csomóponthoz kapcsolódik és arról szól, hogy a bemeneti jelek hogyan képeznek közös képviselőt (reprezentet). Az MP neuronok példájában ez az x_j bemeneti jelek lineáris függvénye, a következő kifejezéssel adott:

$$f_i = \sum_{j=1}^m w_{ij} x_j - \Theta_i \quad (5.40)$$

ahol a Θ_i az i -dik csomópont küszöbe. Az ettől összetettebb integrációs függvények a következők:

- másodfokú függvény:

$$f_i = \sum_{j=1}^m w_{ij} x_j^2 - \Theta_i \quad (5.41)$$

- szférikus függvény:

$$f_i = \rho^{-2} \sum_{j=1}^m (x_j - w_{ij})^2 - \Theta_i \quad (5.42)$$

- polinomiális függvény:

$$f_i = \sum_{j=1}^m \sum_{k=1}^m w_{ijk} x_j x_k + x_j^{\alpha_j} - \Theta_i . \quad (5.43)$$

A feldolgozó elemek következő tevékenysége az f_i bemeneti jelek összesített ábrázolásának átadása a csomópont kimenetére valamilyen kimeneti jel formájában. Ez az úgynevezett aktivációs függvény, vagy az átviteli függvény $a(f)$ feladata. Az aktivációs függvények néhány általános formája adott a következőkben:

- step-függvény:

$$a(f) = \begin{cases} 1 & \text{ha } f \geq 0 \\ 0 & \text{ha } f < 0 \end{cases} \quad (5.44)$$

- hard-limiter függvény:

$$(f) = \operatorname{sgn}(f) = \begin{cases} 1 & \text{ha } f \geq 0 \\ -1 & \text{ha } f < 0 \end{cases} \quad (5.45)$$

- rámpa-függvény:

$$a(f) = \begin{cases} 1 & \text{ha } f > 1 \\ f & \text{ha } 0 \leq f \leq 1 \\ 0 & \text{ha } f < 0 \end{cases} \quad (5.46)$$

- unipoláris Sigmoid függvény:

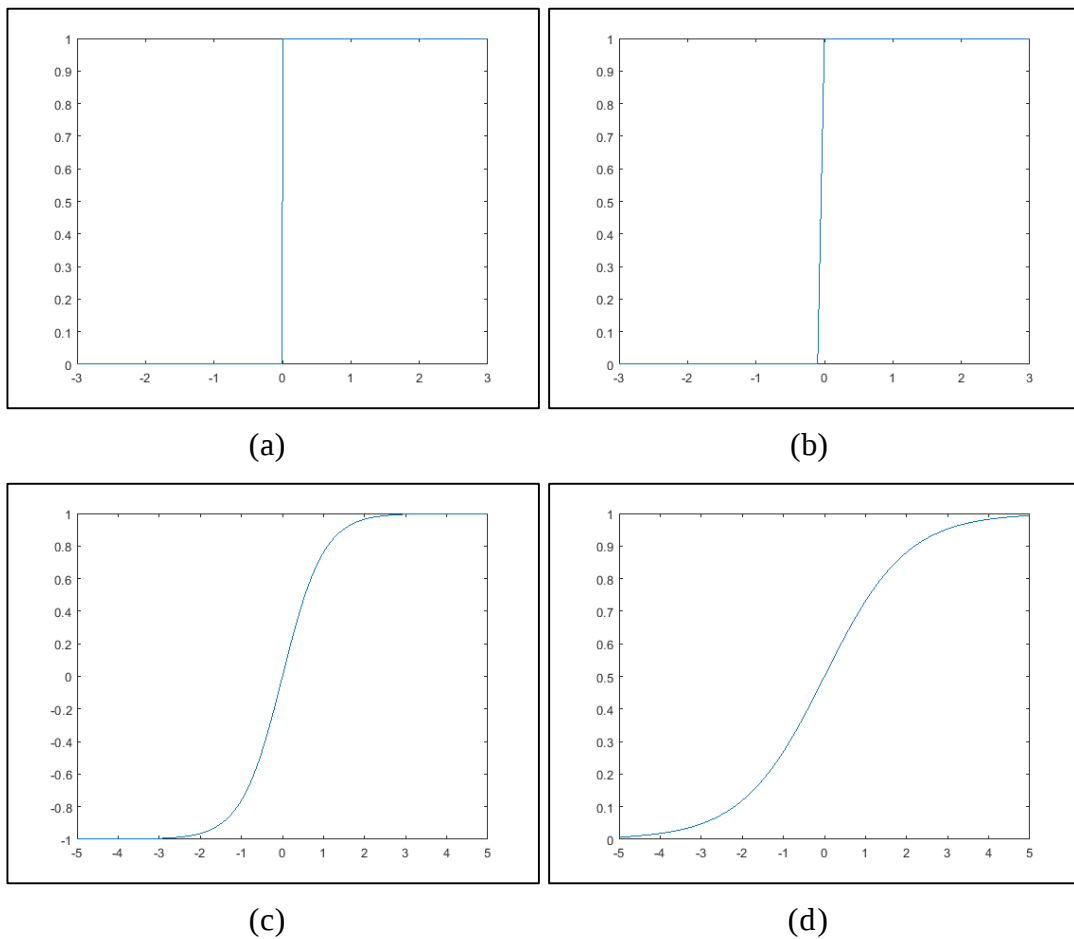
$$a(f) = \frac{1}{1 + \exp(-\lambda f)} \quad (5.47)$$

- bipoláris Sigmoid függvény:

$$a(f) = \frac{2}{1 + \exp(-\lambda f)} - 1 \quad (5.48)$$

ahol a $\lambda > 0$ és ez a paraméter határozza meg a Sigmoid függvény meredekségét. Ennek a paraméternek a nagyon kicsi értéke esetén a Sigmoid függvény általában lineáris függvényhez közelít, míg a paraméter nagyon nagy értékénél ez a függvény nagyon hasonlít a hard-limit függvényhez.

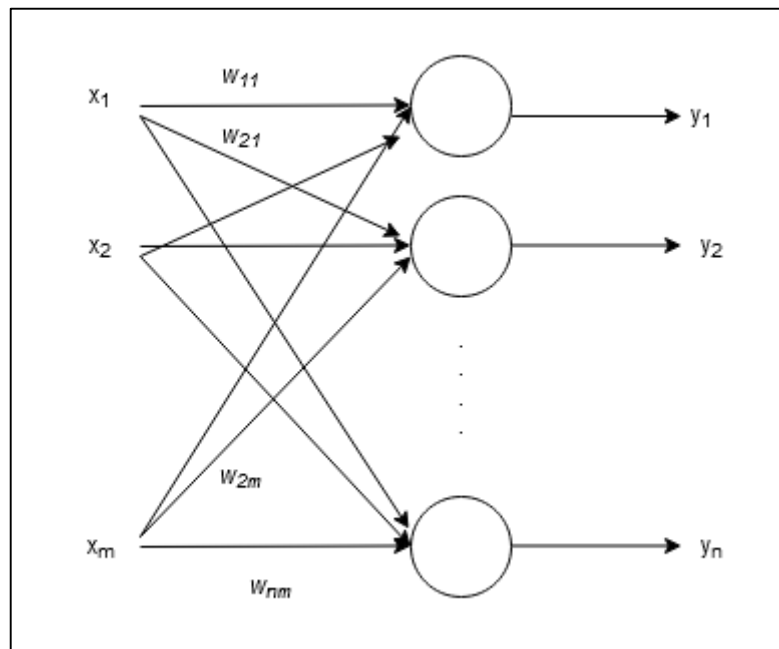
A bemeneti jelek nemlineáris integrációs funkciói lehetővé teszik az összetett partíciós térelemeket, úgynevezett *döntési határokat* (az alakfelismeréssel analóg módon és valójában azt a kifejezést jelölik, hogy két osztály között rendkívül nemlineáris elválasztási határok alakíthatók ki). Azonban minél összetettebb az integrációs funkció, annál bonyolultabbnak kell lennie az ideghálózat csomópontjai közötti interakciónak is, ami gyakran növeli a hálózat számítási terhelését.



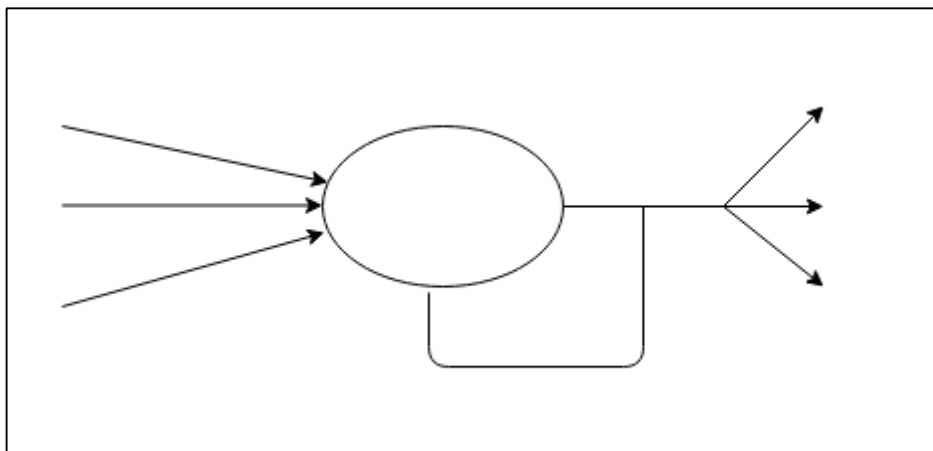
5.23. ábra: Az aktivációs függvények példai: (a) step-függvény, (b) hard-limiter függvény, (c) bipoláris Sigmoid függvény, (d) unipoláris Sigmoid függvény

A csomópontok közötti kapcsolatok

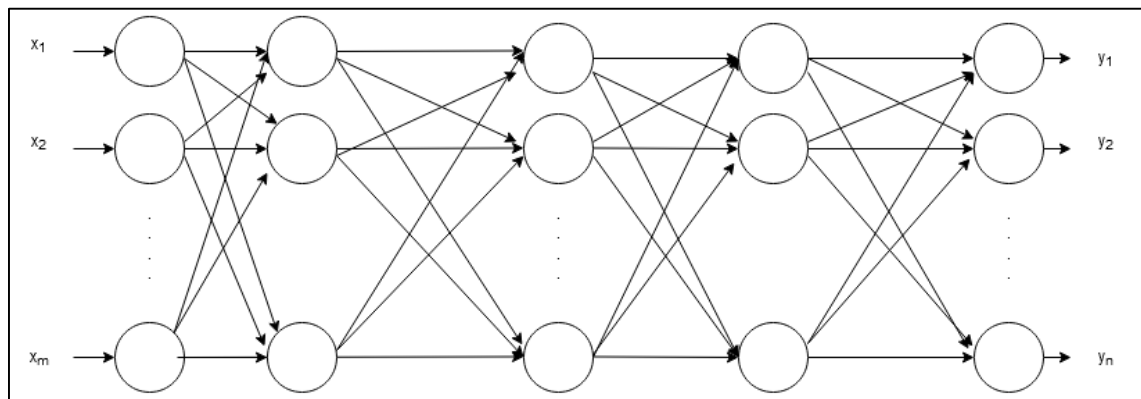
A neurális hálózat nagyszámú csomópontból áll, amelyek sűrűn kapcsolódnak más csomópontokhoz, valamint önmagukhoz is, ahol megengedett a jel csomópontokon keresztül történő terjedésének késleltetése is. A következő ábrákon a neurális hálózatok öt alapstruktúrája, azaz a csomópontok közötti kapcsolatok öt alapvető ún. geometriája lesz mutatva.



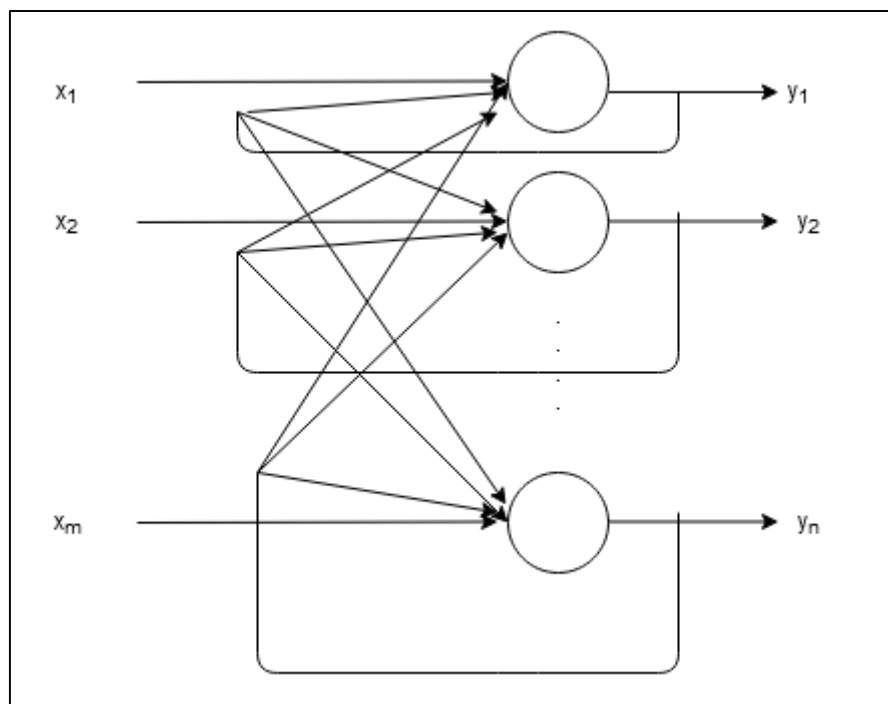
5.24. ábra: Egyrétegű feed-forward neurális hálózat



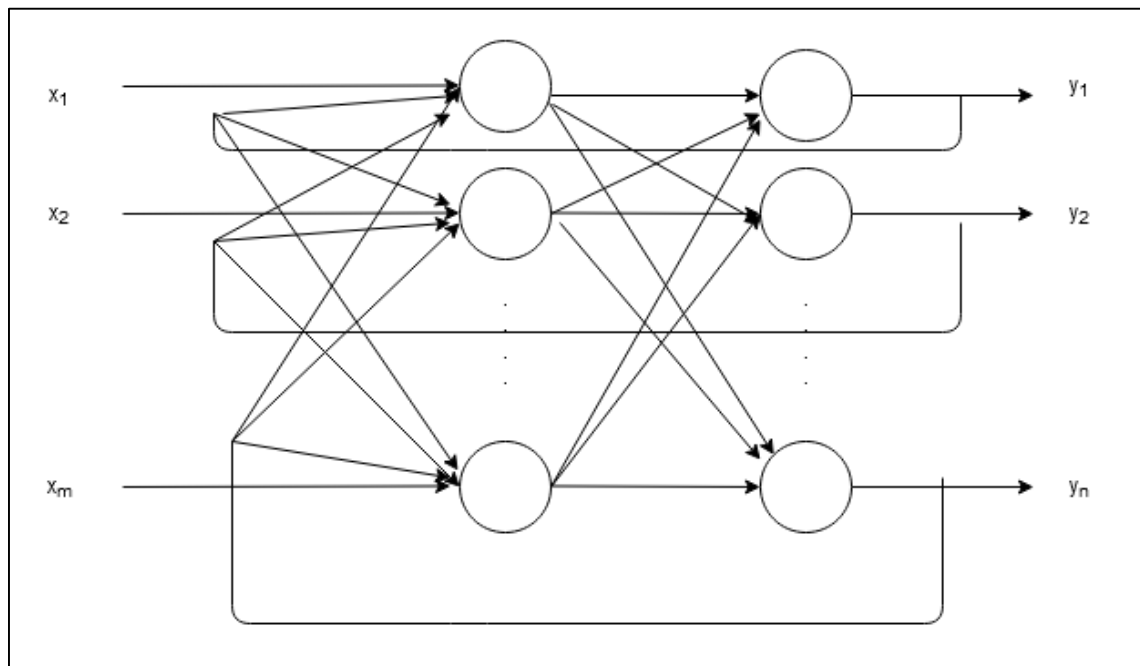
5.25. ábra: Visszacsatolással ellátott csomópont



5.26. ábra: Többrétegű feed-forward neurális hálózat



5.27. ábra: Egyrétegű rekurens neurális hálózat



5.28. ábra: Többrétegű rekurens neurális hálózat

Először is ezekből a hálózati struktúrákból látható, hogy mindig több csomópont rendeződik rétegenként, ahol mindegyik csomópontba több különböző súlyú bemeneti jel lép be, majd több kimeneti jel hagyja el a réteget, minden csomópontból egy. A hálózat rétegeinek száma lehet egy, vagy több, megkülönböztetve ezzel az egyrétegű hálózatokat a többrétegű hálózatoktól. Látható továbbá, hogy a többrétegű hálózatokban az első réteg valójában a bemeneti réteget képviseli és nincs más feladata, mint a bemeneti jelek többszörözése annyiszor, amennyit a következő réteg igényel. A hálózati kimeneteket a kimeneti réteg generálja. A bemenet és a kimenet közötti csomópontok minden rétegét rejtett rétegnek nevezzük. A neurális hálózatokban a rejtett rétegek száma lehet nulla, vagy attól több. Azt mondják, hogy egy neurális hálózat teljesen össze van kötve (csatlakoztatva van), ha az előző mindegyik csomópontja csatlakozik a következő réteg minden csomópontjához. Megfigyelhető továbbá, hogy az említett struktúrákban kétféle hálózatot különböztetnek meg. Az első típusú hálózatok az ún. *előremenő* hálózatok (*feedforward*), amelyekben a csomópontokból való kilépés

nem tér vissza bemenetként az azonos, vagy az előző réteg csomópontjaihoz. Az előremenő hálózatoktól eltérően a *visszacsatoló* hálózatoknak (*feedback*) megvan az a tulajdonsága, hogy a csomópontokból való kilépés bemenetként visszatér ugyanazon, vagy az előző réteg csomópontjaihoz. A visszacsatolást, amelyben a csomópontok kimenetei visszatérnek a csomópontok bemenetéhez ugyanabból a rétegből, *laterális visszacsatolásnak* nevezzük. A visszacsatolt neurális hálózatokat, amelyek zárt hurokkal rendelkeznek, *visszatérő* (*rekurens*) neurális hálózatoknak nevezik.

A neurális hálózatok tanítási (edzési) szabályai

A neurális hálózat specifikációjának harmadik nagyon fontos eleme a tanítási, edzési szabály. Általánosságban elmondható, hogy a neurális hálózatok tanításának két különböző módja van: az első a parametrikus tanítás, amely a neurális hálózat súlyainak beállításához kapcsolódik, a második pedig a strukturális tanítás, amely magába foglalja a hálózati szerkezet, a rétegek számának és a bennük lévő csomópontok számának megváltoztatását. Ez a kétféle képzés egyszerre, vagy külön is alkalmazható.

Először szemléltetve lesz a neurális hálózatok paraméteres tanításának, edzésének fogalma. Tegyük fel, hogy n csomópontunk van a hálózatban és mindegyikhez hozzátartozik m kapcsolat, azaz m súly együtthető. Ezután a súly mátrix, vagy másképp a kapcsolatok mátrixa a következőképpen határozható meg:

$$W = \begin{bmatrix} w_1^T \\ \vdots \\ w_n^T \end{bmatrix} = \begin{bmatrix} w_{11} & \cdots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nm} \end{bmatrix} \quad (5.49)$$

ahol a $w_i = [w_{i1}, \dots, w_{im}]^T$; $i=1, 2, \dots, n$ az i -dik csomópont súlyvektora és a w_{ij} a kapcsolat súlya a j -dik csomópontig. Feltételezzük, hogy a W mátrix olyan elemeket tartalmaz, amelyek változtathatók a neurális hálózatban. Ekkor az össz létező W mátrixok halmaza meghatározza az össz lehetséges információs feldolgozó

konfigurációt az adott neurális hálózat esetében. Ez azt jelenti, hogy olyan W mátrixot kell kiválasztani, amely viselkedésében, (adott bemeneti jelhez megfelelő kimeneti jelet generál) eléri azt, amit elvárunk tőle. Attól függően, hogy mik az elvárások egy neurális hálózattól, a tanítási szabályokat három csoportba lehet sorolni:

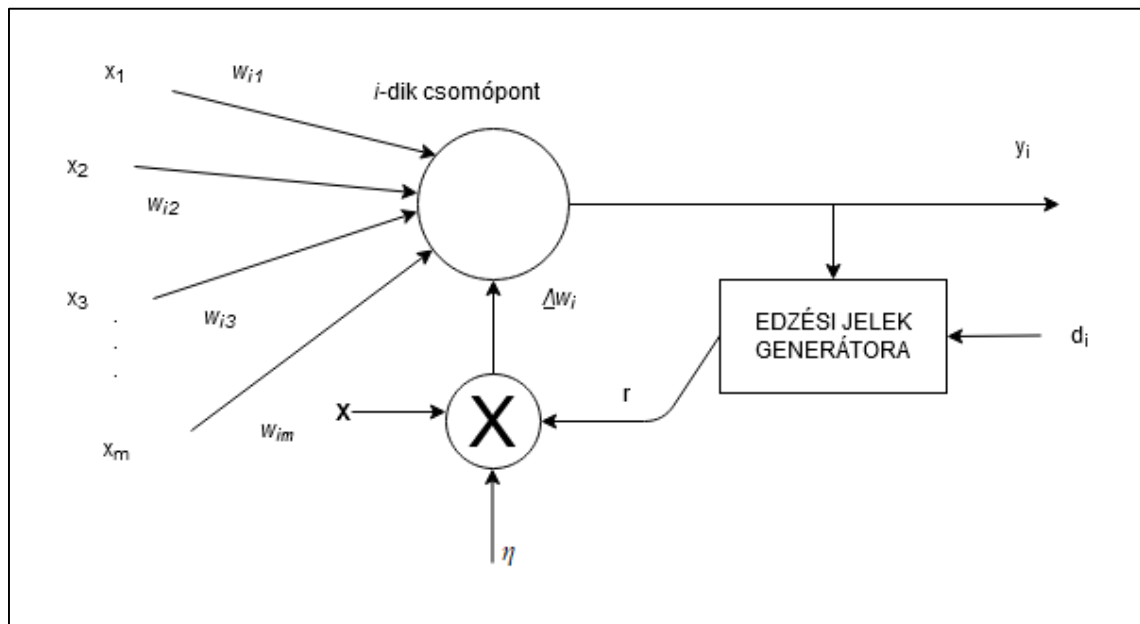
- felügyelt tanítás,
- nem felügyelt tanítás,
- úgynevezett *reinforcement* („megerősítő”) tanítás.

Ezek közül az első az úgynevezett „tanító tanítás” és ez a fajta tanítás analóg egy klasszifikációs eljárással amely rendelkezik tanítási halmazzal -készlettel, ahol mindkét klasszából vannak minták és ahol a klasszifikátort úgy tervezték, hogy egyértelműen elkülönítse a különböző klasszákhöz tartozó formákat.

A nem felügyelt, vagy a „tanító” nélküli tanítás az alakzatok felismerésében alkalmazott klaszterizációval analóg eljárás, ahol a minták teljes halmazából nem tudni előre, hogy melyik minta melyik osztályba tartozik.

A *reinforcement* tanítás az úgynevezett kritikai tanítás, ahol nincs teljes információ arról, hogy mi a jó és mi a rossz reakciója a neurális hálózatnak és csak néhány indikátor van arról, hogy mi is a helyes viselkedése a neurális hálózatnak.

A neurális hálózat tanítási rendszerének általános felépítését a következő ábra mutatja be:



5.29. ábra: A neurális hálózat tanítási rendszerének általános felépítése

Ebben az esetben az x_j ; $j=1,2,\dots,m$ jel lehet egy generált jel az előző rétegből, vagy egy külső bemeneti jel is. Meg kell figyelni, hogy az adott sémában a Θ küszöb be lett vezetve, mint súly együttható abban az ágba, amely az állandó -1 értékű bemeneti jeltől indul. A d_i jel a kívánt kimeneti jel és csak a felügyelt tanítás esetében (vagy néha a *reinforcement* tanításnál) elérhető, míg a nem felügyelt tanításnál a neurális hálózat súlyának megváltoztatása csak a bemeneti és kimeneti jelek alapján történik.

A súlyozási együtthatók beállításának (hálózati edzés) általános alakja abból áll, hogy a t időpontban bekövetkező súlyváltozás arányos az redzésjellel és az $x(t)$ bemeneti jellel, azaz:

$$\Delta w_i(t) = \eta r x(t) \quad (5.50)$$

ahol az η egy pozitív állandó, amit *tanítási állandónak* neveznek és a tanítás fokát határozza meg. Az r jel a tanítási (edzési) jel, ami általában egy függvény, amely függ a w_i , x -től és ha lehetőség van a kívánt d_i kimenettől is:

$$r = f_r(w_i, x, d_i). \quad (5.51)$$

Tehát, a súlyozási tényező értéke a $(t+1)$ időpillanatban felírható a következő kifejezéssel:

$$w_i(t+1) = w_i(t) + \eta f_r(w_i(t), x(t), d_i(t))x(t) \quad (5.52)$$

Az (5.52) kifejezés gyakran a következő formában van felírva:

$$w_i^{t+1} = w_i^t + \eta f_r(w_i^t, x^t, d_i^t)x^t. \quad (5.53)$$

Az (5.53) reláció a tanítási törvény (szabály) a súlyozási együtthatók beállítására diszkrét (időbeli) esetben, míg a folytonos neurális hálózatok esetében a tanítási törvény a megfelelő differenciális egyenlettel fejezhető ki:

$$\frac{dw_i(t)}{dt} = \eta r x(t). \quad (5.54)$$

Tehát a tanítás (edzés) törvényei elsősorban abban különböznek, hogy milyen választási lehetőséget kell választani az r edzésjel számára. A neurális hálózat elméletben számos jellegzetes tanítási törvény létezik. Fontos megemlíteni Hebb tanítási törvényét, amely az aktuális kimeneti jel értékét veszi fel az edzésjel értékeként és fontos megemlíteni a visszafelé propagációs hiba (*backpropagationerror*) módszerét.

A Hebb tanítási törvénye 1949-ben felvetett hipotézisen alapszik, amely a következőket állítja: „Ha az A neuron és a B neuron közötti axiális kapcsolat a B neuron elkerülhetetlen reakcióját okozza és ha ez a helyzet sokszor megismétlődik, vagy permanens, akkor ennek a kapcsolatnak a hatékonysága a B neuron

reakciójával idővel erősödni fog”. Más szavakkal, az agy szinaptikus kapcsolatai a preszinaptikus és a posztzinaptikus idegsejtek közötti korrelációval arányosan erősödnek. Kvantitatív értelemben az r edzésjelnek meg kell egyeznie a kimeneti jel értékével:

$$r = a(w_i^T x) = y_i \quad (5.55)$$

azaz a súly együttható változása a következő formában írható fel:

$$\Delta w_{ij} = \eta a(w_i^T x) x_j = \eta y_i x_j \quad \text{ha } i = 1, \dots, n ; j = 1, \dots, m . \quad (5.56)$$

5.2.6.2. A neurális hálózatok alkalmazásának példája

A folytatásban be lesz mutatva a neurális hálózatok alkalmazása egy példán, ahol a cél felismerni a detektált rendszám táblából a betűket és számokat, tehát a karaktereket. A példában egy *feed-forward backpropagation* típusú neurális hálózat került alkalmazásra.

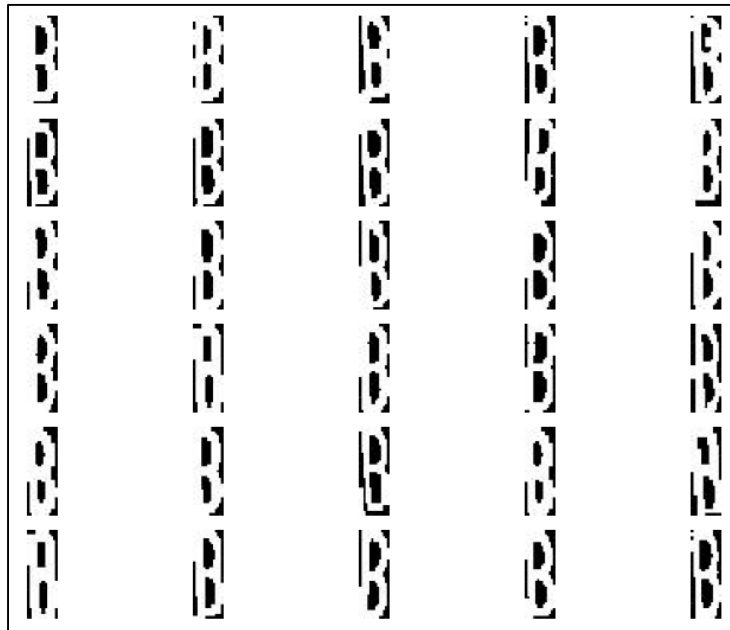
A neurális hálózat tanítási folyamatának négy alapvető lépése van:

- a tanítási adatok kialakítása,
- a neurális hálózat architektúrájának a kiválasztása,
- a neurális hálózat edzése,
- a hálózat válaszána a szimulációja az új bemeneti adatokra.

A tanítási adatok kialakítása

Először a tanítási készlet definiálása szükséges. A rendszám tábla leginkább két betűből és 4 és 8 közötti számokból áll. Ebben a példában a tanítási halmaz a „0”-tól „9”-ig terjedő számokra és a „B”, „G”, „N” és „S” betűkre lett szűkítve (más

karakterek, azaz más városok rendszámtábláinak összegyűjtésének lehetetlensége miatt). Az alábbi ábra a "B" betű képzési halmazát mutatja be.



5.30. ábra: A neurális hálózat tanítási halmazának egy része

Az edzőkészlet kialakítása a következőképpen lett végrehajtva:

- Először minden karaktert külön-külön, amelyek méretei 20x10 pixel, 200x1 pixel méretű mátrixoszloppá, vektorrá kell alakítani. A kialakítandó mátrix méretei attól függnnek, hogy hány minta alapján szükséges a hálózatot betanítani. A mátrixoszlopok száma függ a tanítási képek számától.
- Ebben az esetben 14 (10 szám + 4 betű) tanítási mátrixot kell kialakítani. A 14-es szám a neurális hálózati kimeneteinek számáról szól, minden karakterhez egy kimenet tartozik.
- A végén történik a végleges edzőkészlet kialakítása. 200xN méretű mátrix létrehozva (ahol $N = 14 * n$, "n" az azonos karakterű minták száma).

	1	2	...	n	n+1	n+2	...	2n	...	N
0	a "0" mintáinak oszlopa				az "1" mintáinak oszlopa				...	az "S" mintáinak oszlopa
1									...	
2									...	
3									...	
.									...	
.									...	
.									...	
.									...	
.									...	
.									...	
200									...	

5.31. ábra: A tanítási minták mátrixa

Amikor a tanítási adatkészlet létre lett hozva, a következő lépés kialakítani a célok (*target*) halmazát:

- Minden karakterhez külön-külön kell létrehozni egy célmátrixot. Mivel a rendszerünknek 14 kimenete van, a karakterenkénti célmátrixnak 14 sort és "n" oszlopot kell tartalmaznia (karakterenként annyi képünk van a neurális hálózat betanításához).
- Az első karakter esetében (esetünkben „0”) az első sor a $14 \times n$ méretű célmátrixban csak egyeseket tartalmaz, míg a mátrix többi eleme pedig nulla. A második karakterhez ("1") szintén célmátrix lett képezve, azzal a feltétellel, hogy most már csak a második sor mátrixelemei egyesek, a többi pedig nulla. Ily módon mind a 14 karakterből célok, ún. *targeteket* lettek alkotva.
- Végül folytatódik a végső célkészlet kialakítása. Ez egy $14 \times N$ méretű mátrix, ahol az első "n" (1-n) oszlop a "0" szám célpontja, a második "n" ($n + 1 - 2n$) oszlop az "1", a harmadik "n" ($2n + 1 - 3n$) oszlop száma "2" stb.

	1	2	...	n	n+1	...	2n		...	N		
0	1	1	...	1	0	0	...	0	0	...	0	
1	0	0	...	0	1	1	...	1	0	0	...	0
2	0	0	...	0	0	0	...	0	0	0	...	0
3	0	0	...	0	0	0	...	0	0	0	...	0
4	0	0	...	0	0	0	...	0	0	0	...	0
5	0	0	...	0	0	0	...	0	0	0	...	0
6	0	0	...	0	0	0	...	0	0	0	...	0
7	0	0	...	0	0	0	...	0	0	0	...	0
8	0	0	...	0	0	0	...	0	0	0	...	0
9	0	0	...	0	0	0	...	0	0	0	...	0
"B"	0	0	...	0	0	0	...	0	0	0	...	0
"G"	0	0	...	0	0	0	...	0	0	0	...	0
"N"	0	0	...	0	0	0	...	0	0	0	...	0
"S"	0	0	...	0	0	0	...	1	1	...	1	

5.32. ábra: A célok mátrixa

A mellékelt táblázatból látható, hogy minden adatszlop-vektor (minden karakterhez) hozzárendelésre kerül a céloszlop-vektorhoz.

A neurális hálózat kiválasztása

A következő lépés a neurális hálózat kiválasztása több lehetséges különböző architektúrából. A szakirodalom szerint a karakterfelismerés terén a legjobb eredményt adó neurális hálózat a *feed-forward backpropagation* neurális hálózat.

A backpropagáció a nemlineáris átviteli funkcióval rendelkező többrétegű hálózatok tanítására vonatkozó "Widrow-Hoff" szabály általánosításával lett létrehozva. Ezt a tanítási szabályt a következőképpen írják le:

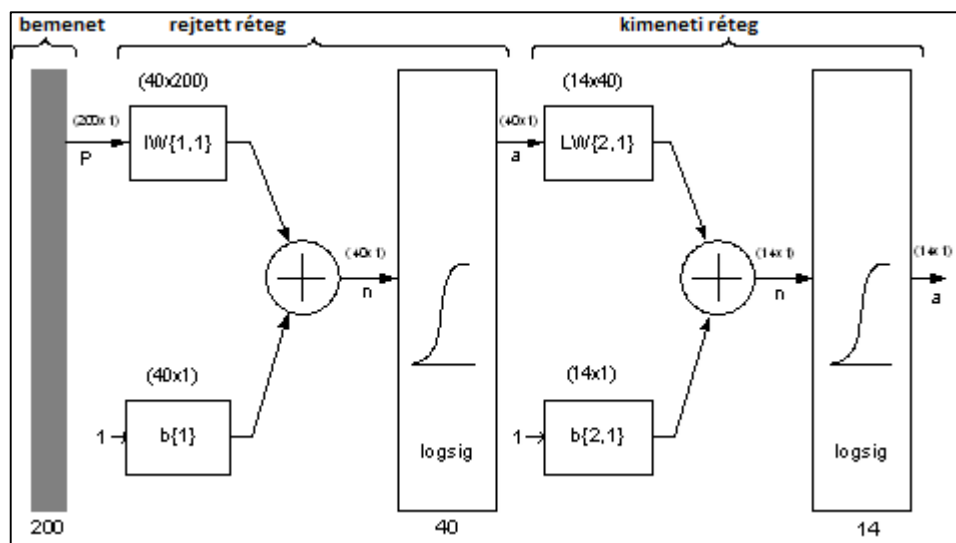
$$W(k+1) = W(k) + 2\alpha e(k)pT(k) \quad (5.57)$$

$$b(k+1) = b(k) + 2\alpha e(k) \quad (5.58)$$

ahol az e (*error*) hiba és a b felbecsülés (*bias*) vektorok, míg az α a tanítási fokot (szintet) fejezi ki.

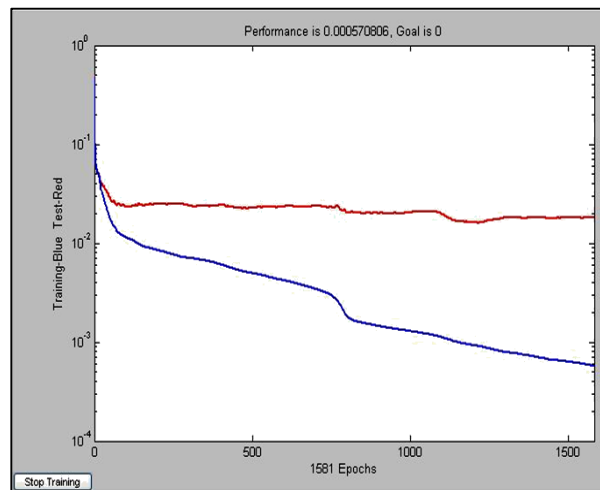
A bemeneti vektorok és a hozzájuk tartozó célvektor került felhasználásra a hálózat tanítására és a megfelelő bemeneti vektorok összekapcsolására, a korábban definiált

megfelelő kimeneti vektorral. A standard backpropációs algoritmus egy gradiensen csökkenő algoritmus. A megfelelően betanított hálózatnak alkalmas választ kell generálnia azokra az adatokra is, amelyekkel nem tanították be. A 30 mintával betanított és 20 tesztmintával tesztelt feed-forward backpropagációs hálózat architektúráját a következő ábra ábrázolja:



5.33. ábra: A példában alkalmazott neurális hálózat modellje

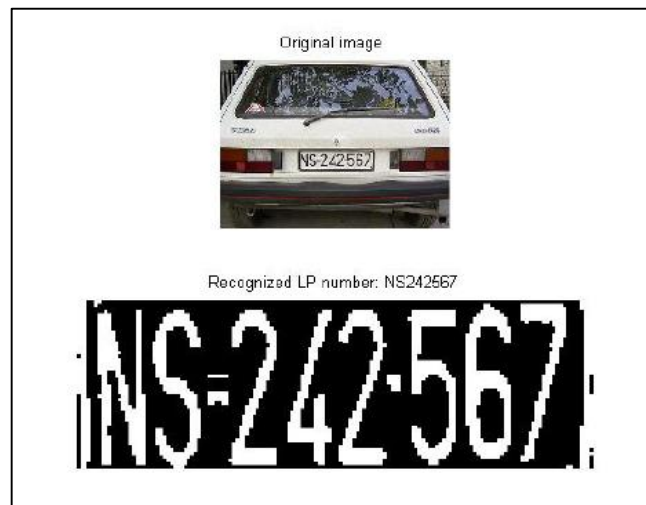
Miután elkészültek az adatmátrixok az edzéshez, a teszteléshez és a célokhoz, majd be lett állítva az összes szükséges paraméter, megindul a neurális hálózat tanítása. A tanítási folyamat a következő grafikonon látható:



5.34. ábra: A tanítási folyamat grafikonja

Két csökkenő görbe észlelhető a grafikonon. A kék görbe az edzés folyamatát mutatja, a piros pedig a tesztelést. Egy ilyen rendszer elért teljesítménye 0,000571, 1581 iteráció esetén (*epochs*). A tanítási folyamatot bármikor leállíthatjuk, ha elégedettek vagyunk a neurális hálózat elért teljesítményével. A képzés befejezése után a neurális hálózat reakciója, válasza új bemeneti adatokkal tesztelhető. Minden kimenet sorrendben egy adott karakternek felel meg, amelyet az edzésadatok kialakításakor definiáltunk.

Némely esetekben egy ilyen hálózat nem képes megkülönböztetni a „B” betűt a „8” számtól, aminek oka leginkább a zajos, deformált bemeneti adat. Természetesen hasonló probléma merülhet fel másmilyen karakter esetében is. Továbbá fontos megemlíteni, hogy a karakterméretek megnövelésével és a minták számának növelésével a neurális hálózat hatékonysága is növekedne, de ugyanúgy a feldolgozási idő is növekedne, ezért esetenként szükség lehet hatékonyabb hardverre is.



5.35. ábra: A neurális hálózat alkalmazásának eredménye

Az 5.35. ábrán látható a neurális hálózat alkalmazása a rendszámtáblák karaktereinek felismerésében. A jármű és a detektált rendszámtábla között láthatók a felismert rendszámtábla karakterei.

Mint látható, a neurális hálózat igen hatékony módja lehet az objektumfelismerési alkalmazásoknak abban az esetben, ha maga a hálózat kiépítése és betanítása megfelelően van elvégezve. Sokszor a legnagyobb problémát a tanítási adatok definiálása, begyűjtése, kialakítása okozza, mert a jó betanításhoz leginkább nagyszámú adatra van szükség, amit sok esetben nem egyszerű összegyűjteni.

Mindezek ellenére, a neurális hálózatok a mesterséges intelligencia kiemelkedő módszerei közé tartoznak és igen népszerűek a kutatók és alkalmazásfejlesztők körében.

5.2.7. A mélytanulós módszerek – mélytanulású neurális hálózatok

A mélytanulás egy gépi tanulási technika, amely a neurális hálózatokon alapul. A mélytanulás a neuronok különböző rétegeivel fokozatosan vonja ki a jellemzőket a bemeneti adatokból, így például a képfeldolgozás során a neurális hálózat alsó rétegei képesek azonosítani a kép széleit, míg a magasabb rétegek az emberi arc széleit. A számítógépes látás, az emberi beszédfelismerés, a bioinformatikai, autonóm vezetési és

más alkalmazások a mély neurális hálózatok (*deep neural networks*), a mély hiedelem hálózatok (*deep belief network*) és a konvolúciós neurális hálózatok (*convolutional neural networks-CNN*) mélyreható tanulási architektúráit alkalmazzák.

A legtöbb mélytanulási modell a konvolúciós neurális hálózatokon alapszik. A „mély” szó a mély neurális hálózat nevében a rétegek számára utal, amelyeken keresztül a bemeneti adatok áthaladnak és átalakulnak. A mélytanulás során minden szint megtanulja a bemenő adatokat összetettebbé és absztraktabbá alakítani.

A mélytanulási modell neurális hálózatokra épül, amelyek bemeneti és kimeneti rétegből és több különböző rejtett rétegből állnak. A neurális hálózat minden rétege számos idegsejtet, neuront tartalmaz, amelyek hozzárendelt súlyokat használnak a bemeneti adatok átalakításához és a kimenet meghatározásához. A neurális hálózat fő célja, hogy ne a felhasználónak kelljen beállítani azokat a paramétereket amelyeket a modell megkap. Ehelyett a neurális hálózat önállóan be tudja állítani ezeket és képes őket az igényekhez alakítani az edzés, tanulás során. .

Tehát a mélytanulás a gépi tanulási algoritmusok egyik formája, amely:

- a többrétegű nemlineáris processzoregységeket használ a funkciók kibontására és átalakítására. Minden következő réteg bemenetként veszi az előző réteg kimeneti elemeit.
- felügyelt és/vagy felügyelet nélküli módon tud tanulni.
- megtanulják az absztrakció különböző fokainak megfelelő reprezentációs szinteket.
- valamilyen formában csökkenő gradiensű, backpropagációs edzésű algoritmust alkalmaznak a hibák csökkentésére.

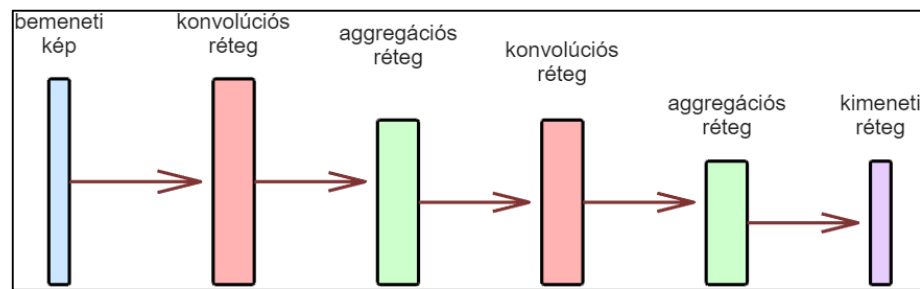
A mély programozásban használt rétegek magukba foglalják a mesterséges neurális hálózat rejtett rétegeit és az állítási képletek sokaságát. Ezenkívül rétegesen szervezett rejtett változókat is tartalmazhatnak a mélyen generált modellekben.

A folytatásban röviden be lesznek mutatva a mély konvolúciós neurális hálózatok, amelyeket leggyakrabban alkalmaznak az alakfelismerében és a képfeldolgozásban.

5.2.7.1. A mély konvolúciós neurális hálózatok

A konvolúciós neurális hálózat a konvolúcióról kapta a nevét, amely a kép- és jelfeldolgozás során széleskörben használt művelet. Ezekben a szakterületekben konvolúciós szűrőket alkalmaznak, amelyeket a képek élesítésére és elmosására, valamint az élek detektálására használnak. A CNN szempontjából fontos az éldetektáló algoritmusok alkalmazása, amelyek a képen lévő objektumokat, tárgyakat érzékelik. Ahol a klasszikus neurális hálózatok megmutatják hátrányaikat, előtérbe kerülnek a CNN hálózatok előnyei. Például, ha képeket használnának bemeneti adatként, akkor a 200×200 pixel méretű képet a bemeneti rétegben 40000 neuronná kellene átalakítani, ami hatalmas szám és gyakorlatilag lehetetlen lenne a teljesen összekapcsolt neurális hálózatokat betanítani. Színes képek esetében az alapszíneket (RGB) képviselő 3 csatornát is figyelembe kellene venni, ami 120000 neuront adna a bemeneti rétegben. A CNN hálózatok ötlete az, hogy több réteget alkalmazzanak a bemeneti adatok lényeges tulajdonságainak feltárására. Ennek az elképzelésnek megfelelően konvolúciós szűrőket alkalmaznak a képre, a hasznos tulajdonságok kinyerésére és térképeik (*feature maps*) létrehozására. Szűrők alkalmazásával a bemeneti képben a jelentős jellemzők észlelése mellett a kép felbontása is csökken. A CNN hálózatok hátránya a jelentős hardvererőforrások iránti igény, valamint az a tény, hogy a korábbi modelleket nagyszámú mintával (képpel) tanították be. A legújabb eredmények ennek ellenére mégis arra utalnak, hogy viszonylag kis számú mintával is nagy teljesítmény érhető el.

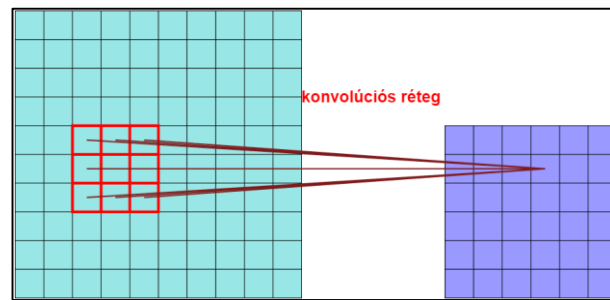
Az eddigiek alapján a konvolúciós hálózatok fontossága abban rejlik, hogy nem igényelnek emberi részvételt a jel releváns tulajdonságainak meghatározásában (ami valószínűleg nem is a legjobb eredményt hozná), de a megoldandó problémától függően megfelelő szűrők megtanulásával meghatározzák, hogy mely tulajdonságok fontosak. Ez a fajta rugalmasság azonban optimalizálási kihívásokkal, valamint nagy mennyiségű adat felhasználással jár.



5.36. ábra: A konvolúciós neurális hálózat modellje

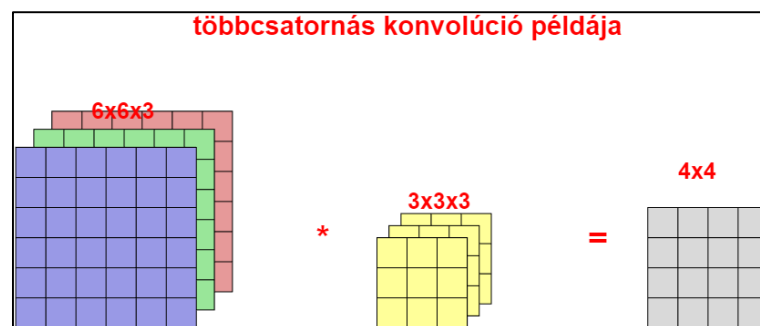
A CNN hálózatok gyakorlatilag mély neurális hálózatok. Ezt az 5.36. ábrán lehet megtekinteni. A konvolúciós hálózat szokásos felépítése kétféle réteg váltakozását foglalja magába, a konvolúciós rétegek és az aggregációs rétegek felváltva. A konvolúciós réteg kimeneteit egy nemlineáris aktiválási függvény transzformálja. Az utóbbi rétegek kimeneteit általában egy teljesen összekapcsolt hálózat köti össze, amely megtanulja az előző rétegek által létrehozott attribútumokat, jellemzőket.

A konvolúciós réteg szerepe az új attribútumok felépítésében van. Minden egyes konvolúciós réteg rendelkezik egy sorozat paraméterezett szűrővel, amelyek elvégzik a konvolúciós feladatokat. Például ha egy szűrő függőleges vonalakat detektál, akkor természetes, hogy legyen egy olyan szűrő is, amely a vízszintes vonalakat párhuzamosan, ugyanabban az előző rétegben (vagy bemenetben) detektálja. Ugyanis a szűrő paraméterei a konvolúciós hálózat paramétereit képviselik és a hálózat betanításának folyamatában tanulják meg azokat. A konvolúciós szűrést olyan egységek végzik, amelyek sorozatba (1D jel), vagy mátrixba (2D jel) vannak szervezve és osztják a paraméterek értékeit. Általában az előző réteg szomszédos egységeinek értékeit vesszik alapul. Így, szem előtt tartva, hogy egy rétegben minden egységnek ugyanazok az együttartói, a konvolúciós réteg hatása úgy értelmezhető, mint az előző réteg konvolúciója az egységgel, amelyet az aktuális rétegparaméterek határoznak meg, vagy egyszerűbben kifejezve, mintha a szűrő „át volna húzva” a képen és az értékek ki lennének számítva minden egyes helyzetben.



5.37. ábra: A konvolúciós réteg

Az 5.37. ábrán minden piros körrel jelölt egység bemenetként az előző réteg kimeneteit veszi figyelembe, amelyek megfelelnek a piros négyzet egységeinek. Ugyanazon konvolúciós réteg minden egységének ugyanazok a paraméter értékei vannak és ugyanazt a műveletet hajtják végre az előző rétegben.



5.38. ábra: A többcsatornás konvolúciós szűrő hatása

Az eddig leírtakban nem volt tárgyalva az, hogy a szűrő csak az előző réteg egy csatornájára, vagy több csatornájára hat. A gyakorlatban gyakran többcsatornás konvolúciós műveletek valósulnak meg, amelyek lehetővé teszik a szűrő számára, hogy az előző réteg több csatornájára is egyszerre kihasson.

Az aggregációs réteg összesíti az előző rétegből nyert információkat, általában az előző réteg szomszédos egységeinek néhány egyszerű aggregációs függvényének kiszámításával, például a maximum, vagy az átlag. A konvolúció egy csatornája egy aggregációs csatornának felel meg. Az információk ilyenfajta elhanyagolása gyakran nem károsítja az elérendő célt. Például, ha egy szem, fül, száj és orr található a képen, a

pontos helyzetre vonatkozó információk valószínűleg nem relevánsak annak eldöntésében, hogy egy arc szerepel-e a képen. Ugyanis a valóságban rendkívül kicsi az esélye annak, hogy a fenti elemeket tartalmazó kép nem az arcot ábrázolja. Ha azonban olyan hálózatot kell létrehozni, amelyben az objektumok helyzete a képen fontos, akkor nem ajánlott az aggregációt használni. Az aggregáció szerepe lényegében a számítási műveletek számának csökkentése a magasabb rétegekben, valamint a paraméterek számának csökkentése (amelyek hatalmasak lehetnek) egy teljesen összekapcsolt hálózatban, amely követi a konvolúció és az aggregáció rétegeit. Mindez az optimalizálás számítási összetettségnek és a modell rugalmasságának csökkenését eredményezi.

A CNN hálózat következő rétege az aktivációs függvény rétege. Az aktiválási függvény helyes megválasztása jelentősen javíthatja a CNN teljesítményét bizonyos feladatokban. Az egyik leggyakrabban használt aktivációs függvény az ún. ReLU (*Rectified Linear Unit*), így a "ReLU réteg" kifejezés gyakran megtalálható az irodalomban is, amely arra a rétegre utal, amelyben a ReLU aktiválási függvényt alkalmazzák.

A következő réteg a teljesen összekapcsolt réteg, neve jelzi annak konfigurációját is: ebben a rétegben minden neuron az előző réteg összes kimenetéhez kapcsolódik. Jellemző, hogy a teljesen összekapcsolt réteget utoljára használják a CNN hálózat kiépítésében a struktúra végén.

A gyakorlatban és a kutatásban számos, már kialakított CNN hálózat van alkalmazásban: *AlexNet* (2012), *ZF Net* (2013), *VGG Net* és *GoogLeNet* (2014), *ResNet* (2015) és *WinnigEnsembles*(2016). Mindezeket a CNN hálózatokat a kutatók és fejlesztők alkalmazzák a kutatásokban és fejlesztésekben. A leggyakrabban alkalmazott hálózatok közé tartozik az *AlexNet*, amelyet a Matlab programcsomag CNN hálózatokkal kapcsolatos fejlesztési környezete is használ.

A CNN hálózatokat ma széleskörben használják és a jelentőségük nagyon nagy. Kimutatták, hogy az újabb, nagyon mély CNN hálózatokban történő osztályozás

pontossága nagyon magas. Ennek a ténynek az egyik következménye a mély CNN hálózatok alkalmazása az orvostudományban, de egyenlőre csak kutatás és kísérleti vizsgálatok szintjén. A tumorszövetek CNN hálózatokkal történő felismerésének ereje és hatékonysága megközelítőleg ugyanazon a szinten van, mint a szakértőé. A kutatók intenzíven dolgoznak olyan problémák megoldásán, mint például a hazugságok észlelése a válaszadók arcából. Számos nagyvállalat jelentős összegeket fektet be a CNN hálózatok kutatásába. A mély architektúrák hasznosnak bizonyultak bizonyos problémák megoldása során is, például a számítógépes játékok autonóm játéka, vagy az emberek tevékenységének felismerése a videofelvételek alapján. A mélytanulás teljes fejlődését nagyon erős számítógépek fejlesztése kíséri, teljesítményük napról napra növekszik, ami döntő fontosságú a kutatók jövőbeni munkája szempontjából.

6. A fuzzy logika

Ez a fejezet a fuzzy logika elméletét és alkalmazását írja le a képfeldolgozás és objektumfelismerés szemszögéből. Röviden be lesz mutatva a fuzzy logika elméleti háttere, majd később példa szemlélteti alkalmazását és hatékonyságát is. A példában részletesebben lesz látható a matematikai morfológia alkalmazása is az alakfelismerésben.

A fuzzy logika a mesterséges intelligencia egyik legalapvetőbb elméleti és gyakorlati „eszközei” közé tartozik. A legelső fuzzy típusú vezérlőt E. Mamdani fejlesztette ki laboratóriumban, a gőzkazánok vezérlésére. Az első nyilvános fuzzy logikán alapuló alkalmazásokat Japánban fejlesztették ki, a legelső jelentős alkalmazás Sendai város metróközlekedésében jelent meg, a metró járatok vezérlésében. Később a fuzzy logika alkalmazása legtöbbször az irányítástechnikai alkalmazásokban terjedt el, majd az utóbbi években egyre nagyobb szerepet kap a jelfeldolgozásban és képfeldolgozásban. Ugyanis a fuzzy döntéshozás nagyban hasonlít az emberi gondolkodáshoz és ezt egyre hatékonyabban lehet kihasználni a hangfelismerésben, orvostudományban és képfeldolgozásban, az objektumok

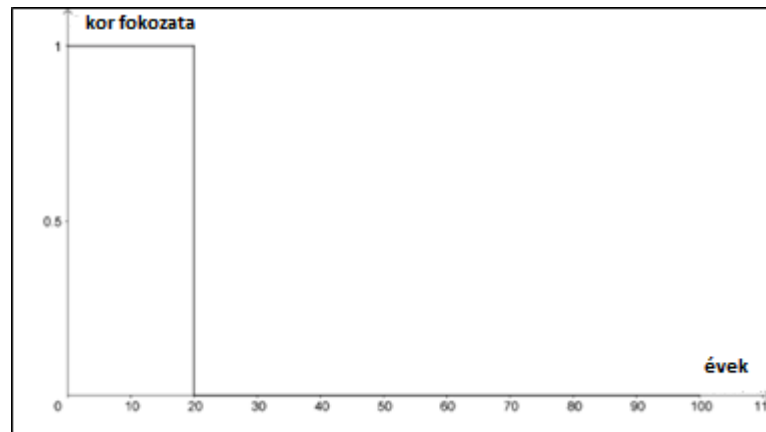
felismerésében és detektálásában. A fuzzy rendszerek létrehozása egyszerű és gyors folyamat, alkalmazásuk pedig bizonyítottan nagy hatékonyságot eredményez. Lényegében a legnagyobb gondot a fuzzy rendszerek alkalmazásában a nagy számítási idő okozhatja olyan alkalmazások esetében, ahol nagy adatokkal kell dolgozni, mint pl. a képfeldolgozás és alakfelismerés. Manapság az elektronika és számítástechnika fejlődésének köszönhetően ez a feldolgozási terheltség sem nagy nehézség, mert a számítógépek egyre gyorsabbak és olcsóbbak.

A fuzzy halmazelmélet több, mint négy évtizeddel ezelőtt lett létrehozva L. A. Zadeh (1965) által. Ez az elmélet képezi a *bizonytalanság modellezésének formálisan leírt matematikai módját a lingvisztikában*. A klasszikus halmazelméletben egy adott elem vagy benne van, vagy nincs benne egy definiált halmazban. Másszóval, az elem halmazhozvaló hozzátartozása határozottan definiált. A fuzzy halmaz ebben az értelemben a klasszikus halmaz általánosítását képezi, mivel a halmazhozvaló hozzátartozást, tagságot jellemezni lehet egy számmal a $[0,1]$ intervallumból. Tehát a fuzzy halmaz *tagsági függvénye* (*membership function*) minden egyes elemet leképez az univerzum halmazból a $[0,1]$ valós számok halmazába. Az univerzum egy előre megadott nagy halmaz amely magába foglalja az össz objektumot (részhalmazokat), amelyek érdekelve vannak. Az egyik legnagyobb különbség a klasszikus és a fuzzy halmazok között az, hogy a klasszikus halmazok egyedi tagsági függvénnyel rendelkeznek, míg a fuzzy halmazok végtelen sokféle tagsági függvénnyel írhatók le, határozhatók meg (a fuzzy kifejezés homályosat, nem egyértelműt, zavarosat, elmosódottat jelent).

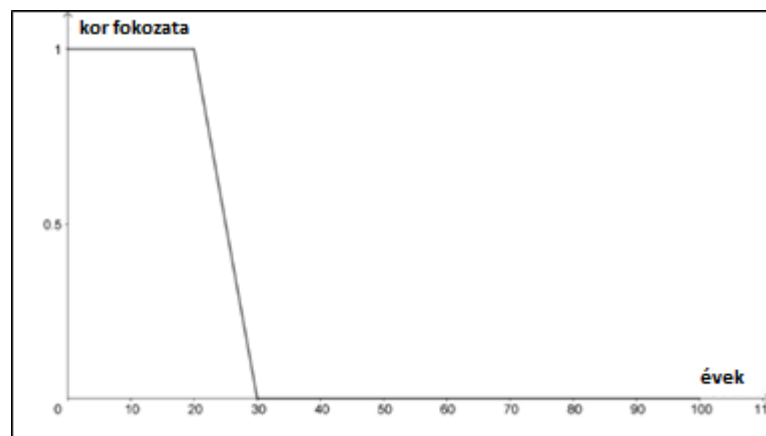
A fuzzy logika egyik legfontosabb jellemzője, hogy képes kifejezni az emberi gondolkodás bizonytalanságát és szubjektivitását. Ezért a fuzzy logika alkalmazásához azokban az esetekben, problémákban folyamodnak, amelyekre jellemző a matematikai modell hiánya, vagy túlságosan bonyolult ahhoz, hogy valós időben történjen az algoritmus működése. Így a fuzzy logika számos műszaki területen találta meg az alkalmazását, mint pl.: az irányításelmélet, alakfelismerés, kvantitatív elemzés, szakértői diagnosztikai rendszerek, tervezés és előrejelzés, információs rendszerek, digitális képfeldolgozás. A fuzzy logika alkalmazásának

ötlete éppen a pontatlan, tisztázatlan fogalmakkal való tervezésre és munkavégzésre vonatkozik, amely a következő példán lesz röviden bemutatva.

Példa: Ha figyeljük a 6.1. ábrán az első képet, akkor a klasszikus halmazelmélet alapján a 20 év alatti személy fiatalnak van tekintve, de az a személy aki betöltötte a 20. életévét, akár pár pillanattal később már nem számít fiatalnak. Épp ez az észrevétel és a gondolkodás a klasszikus halmazok legnagyobb problémája és hátránya, ezért gyakran korlátozott az ilyen elmélet alkalmazása a gyakorlatban. Ugyanis az adott példában nyilvánvaló, hogy rugalmasabb módon szükséges definiálni mit jelent „fiatalnak” lenni. Ez esetben a probléma könnyen megoldható a fuzzy halmazelmélet segítségével, amelynek tagsági függvénye látható a 6.1. ábra második képén. Itt szépen megfigyelhető a folyamatos átmenet a „fiatal” és „idős” személyek meghatározása között. Ez az elmélet lehetővé teszi, hogy létezzen ún. korbeli fokozat, azaz létezzen egy adott személy tagsági fokozata a fiatalok, vagy idősök kategóriájához. Az ilyen jellegű tagsági fokozás leírható részintervallumok segítségével is, mint pl. $[0,20]$, $(20,24]$, $(24,26]$, $[26,30]$ és (>30) . Az efajta leírás sokkal közelebb áll az emberi gondolkosáshoz, érveléshez, ami lényegében a fuzzy logika célja. A következőkben a fuzzy logika matematikai leírása lesz bemutatva.



(a)



(b)

6.1. ábra: (a) Klasszikus halmaz tagsági függvénye, (b) Fuzzy halmaz tagsági függvénye

6.1. A fuzzy logika definíciói

Legyen a Z halmaz olyan elemek halmaza, ahol a generikus elem z -vel van jelölve és $z \in Z$. Az $A \subset Z$ fuzzy halmaz jellemezve van a $\mu_A(z)$ saját tagsági függvényével, amely a Z halmaz mindegyik eleméhez hozzárendel egy valós számot a $[0,1]$ intervallumból. A $\mu_A(z)$ értéke a z pontban képviseli a z tagsági fokát az A

halmazban és fordítva. Amikor a $\mu_A(z)$ értéke közelebb van az egyeshez, annál nagyobb a z tagsági fokozata az A halmazhoz, ha pedig a $\mu_A(z)$ értéke közelebb van a nullához, a tagsági (hozzátartozási) fokozat annál kisebb. A „hozzátartozás” elve ez esetben nem egyezik meg azzal az elvvel, amit alkalmaznak a klasszikus halmazelméletben. A klasszikus halmazoknál az elem vagy benne van, vagy nincs az adott halmazban. A fuzzy halmazoknál minden elem, amelyre érvényes, hogy $\mu_A(z)=1$ teljesen hozzátartozik, teljesen tagja a fuzzy halmaznak, míg azok az elemek, amelyek esetében a $\mu_A(z)=0$ nem tartoznak, nem tagjai a fuzzy halmaznak. A Z halmaz többi elemei, amelyekre érvényes, hogy $0 < \mu_A(z) < 1$ pedig részben tartoznak, részben tagjai a fuzzy halmaznak. Ezek alapján a fuzzy halmaz felírható rendezett párok segítségével, amelynek tagjai a z értéke és a hozzá tartozó tagsági függvény értéke az adott z pontban. A rendezett pár a következő módon írható fel:

$$A = \{z, \mu_A(z) \mid z \in Z\}. \quad (6.1)$$

Abban az esetben, ha a változók folytonosak, az A halmaznak a (6.1) kifejezésből végtelen számú értéke lehet. Ha a z változó értékei diszkrét és véges számú elemmel bírnak, akkor az A halmazt sima felsorolással lehet felírni. Ha a 6.1. ábrán az éveket simán csak egész számokkal fejeznénk ki, akkor felírható a következő A halmaz:

$$A = \{(1,1), (2,1), (3,1), \dots, (20,1), (21,0.9), (22,0.8), \dots, (25,0.5), (24,0.4), \dots, (29,0.1)\} \quad (6.2)$$

ahol pl. a $(22,0.8)$ azt fejezi ki, hogy a 22 éves személy A halmazhoz való hozzátartozásának, tagságának értéke 0.8. A 6.1. ábra második grafikonja teljes egészében jellemzi az A fuzzy halmazt. Ha pl. a $\mu_A(z)$ -nak csak két értéke lenne, pl. a 0 és 1, akkor a tagsági függvény leegyszerűsödne és teljesen megegyezne a klasszikus halmaz tagsági függvényével.

A folytatásban a fuzzy elmélethez tartozó alapvető definíciók kerülnek bemutatásra, amelyek a klasszikus halmazok bővítményei, de némely esetekben akár meg is egyeznek a klasszikus halmazelmélet definícióival is.

Üreshalmaz: A fuzzy halmaz üres akkor és csak akkor, ha a tagsági függvénye identikusan egyenlő nullával a Z halmazban.

Halmazok egyenlősége: Két fuzzy halmaz egyenlő akkor és csak akkor, ha érvényes, hogy:

$$\mu_A(z) = \mu_B(z), (\forall z \in Z) \quad (6.3)$$

és az egyenlőséget $A = B$ jelölik.

Halmaz komplemente: Az A fuzzy halmaz komplemente (NOT) $\bar{A}$ -val, vagy $\text{NOT}(A)$ van jelölve és a következőképpen definiálják:

$$\mu_{\bar{A}}(z) = 1 - \mu_A(z), (\forall z \in Z). \quad (6.4)$$

Részhalmaz: Az A fuzzy halmaz a B fuzzy halmaz részhalmaza akkor és csak akkor, ha érvényes, hogy:

$$\mu_A(z) \leq \mu_B(z), (\forall z \in Z). \quad (6.5)$$

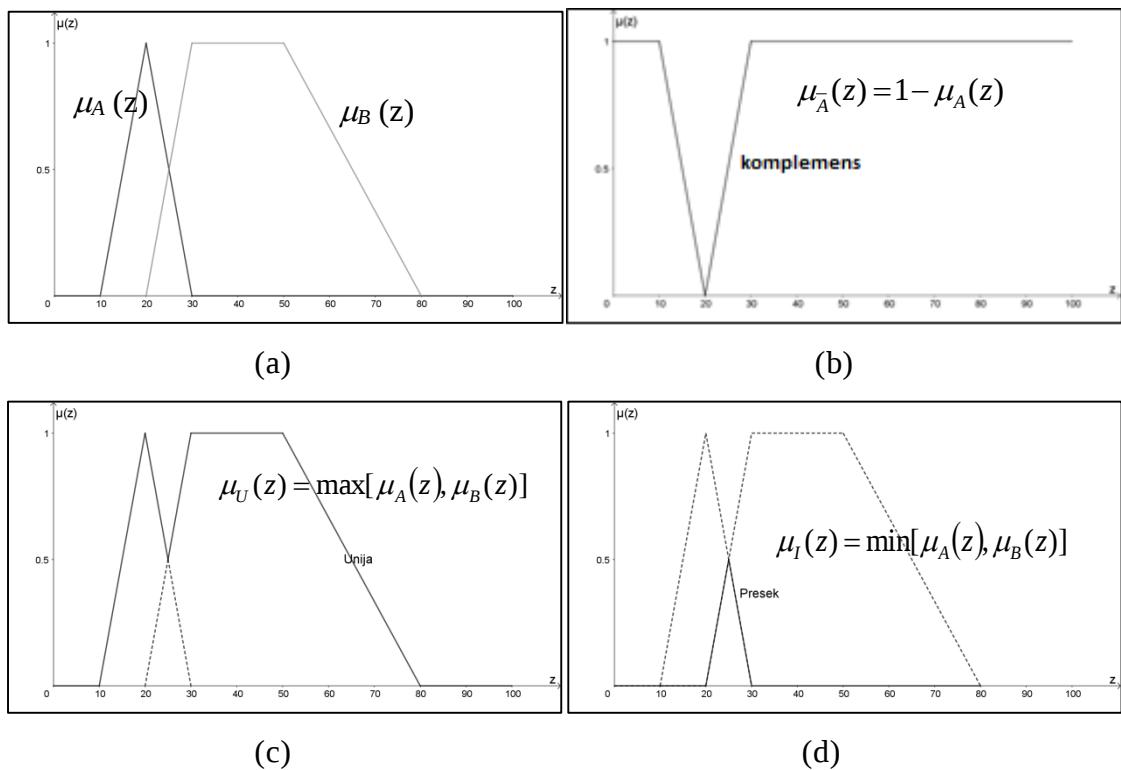
Únió: Két A és B fuzzy halmaz únióját (OR) $A \cup B$, vagy A OR B -nek jelölik és U fuzzy halmaznak jelölik, a tagsági függvénye következőképpen definiált:

$$\mu_U(z) = \max[\mu_A(z), \mu_B(z)], (\forall z \in Z). \quad (6.6)$$

Metszet: Két A és B fuzzy halmaz únióját (AND) $A \cap B$, vagy A AND B -nek és I fuzzy halmaznak jelölik, a tagsági függvénye következőképpen definiált:

$$\mu_I(z) = \min[\mu_A(z), \mu_B(z)], (\forall z \in Z). \quad (6.7)$$

A 6.2. ábrán láthatók a fent bemutatott definíciók. Az első képen látható két fuzzy halmaz, a másodikon az A halmaz komplemente, a harmadik képen látható a két halmaz úniója és a negyedik képen látható a két halmaz metszete.



6.2. ábra: (a) Két fuzzy halmaz, (b) komplement, (c) únió, (d) metszet

6.2. A tagsági függvények

Ez az alfejezet bemutatja a leggyakrabban használt tagsági függvényeket, amelyeket a szakirodalom jellegzetes, karakterisztikus függvényeknek is nevez.

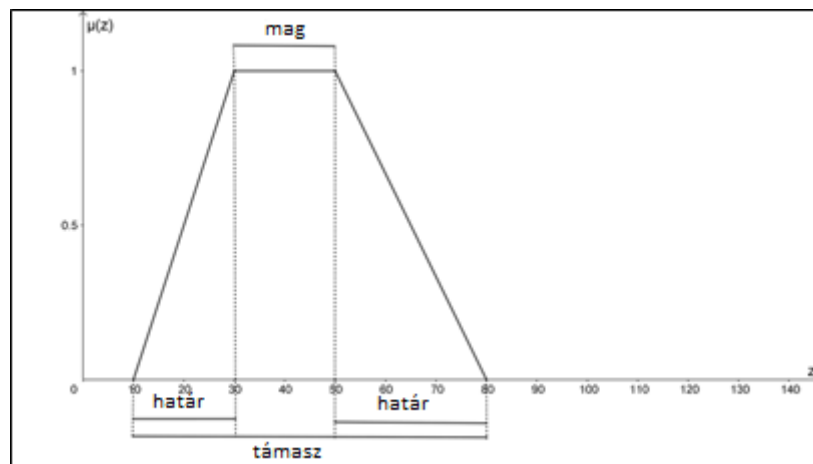
A fuzzy logika bizonytalanságát és határozatlanságát tagsági függvények segítségével írják le. Segítségükkel teljesen leírható a halmaz minden eleme. Ezeket a függvényeket gyakran grafikusán ábrázolják a fuzzy halmazok leírásánál és az ilyen ábrázolás többféle formát (alakot), grafikont is magában foglalhat, ami nem jellemző a klasszikus halmazok leírásában. Természetesen vannak korlátozások és szabályok is az alkalmazott tagsági függvények felhasználásban. Ebben a fejezetben a tagsági függvények alakjai és tulajdonságai lesznek tárgyalva. A tagsági függvények alakját, formáját három fogalom segítségével definiálják:

1. mag
2. támasz (hordozó)
3. határ.

A mag az univerzum halmaz azon részét képezi, amely elemeinek a tagsági függvénye $\mu_A(z)=1$ értékű. Tehát ezek az elemek teljes egészében az A halmazhoz tartoznak.

A támasz az univerzum halmaz részhalmazát képezi, amely elemeire érvényes, hogy $\mu_A(z)>0$. Ezek az elemek csak részben, parciálisan tartoznak az A halmazhoz.

A határ az univerzum halmaz részhalmazát képezi, amely elemei részben, de nem teljes egészében tartoznak az A fuzzy halmazhoz. Ezek az elemek a támasz részhalmazai, de diszjunkcióban vannak a maggal, azaz érvényes rájuk, hogy $0 < \mu_A(z) < 1$. A 6.3. ábrán láthatóak az említett tulajdonságok.



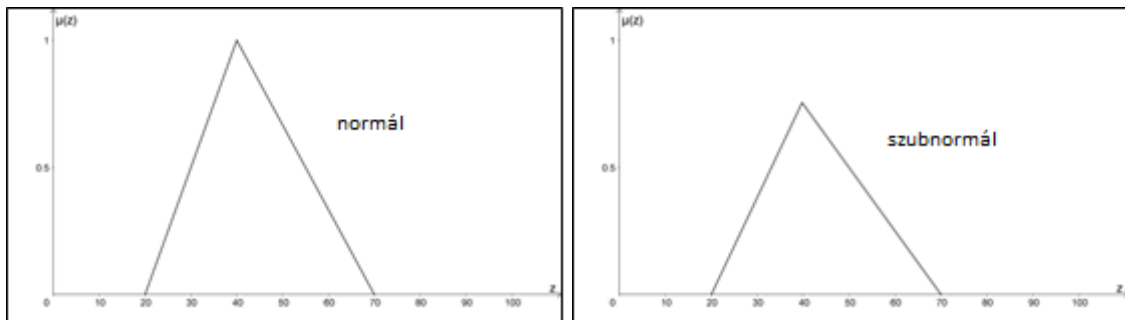
6.3. ábra: A tagsági függvények tulajdonságai

Maga a tagsági függvény alakja lehet szimmetrikus és aszimmetrikus. A tagsági függvények alapján a fuzzy halmazokat a következő módon lehet besorolni:

- *Normál fuzzy halmaz:* Ha az A fuzzy halmazban van legalább egy elem, amelynél a tagsági függvény értéke megegyezik egyel, akkor az a fuzzy halmaz *normál fuzzy halmaz*.
- *Szubnormál fuzzy halmaz:* Ha az A fuzzy halmazban egyetlen elem tagsági függvényének értéke sem egyenlő egyel, akkor az a fuzzy halmaz *szubnormál fuzzy halmaz*.
- *Konvex fuzzy halmaz:* Az A fuzzy halmaz abban az esetben konvex, ha bármely $\lambda \in [0,1]$ érvényes, hogy:

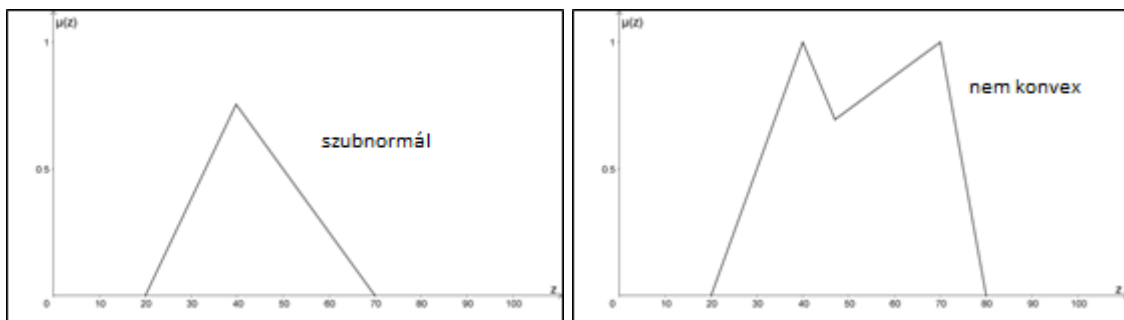
$$\mu_A(\lambda z_1 + (1 - \lambda)z_2) \geq \min(\mu_A(z_1), \mu_A(z_2)). \quad (6.8)$$

A 6.4. ábrán láthatók az előbb felsorolt fuzzy halmazok fajtáinak grafikonjai.



(a)

(b)

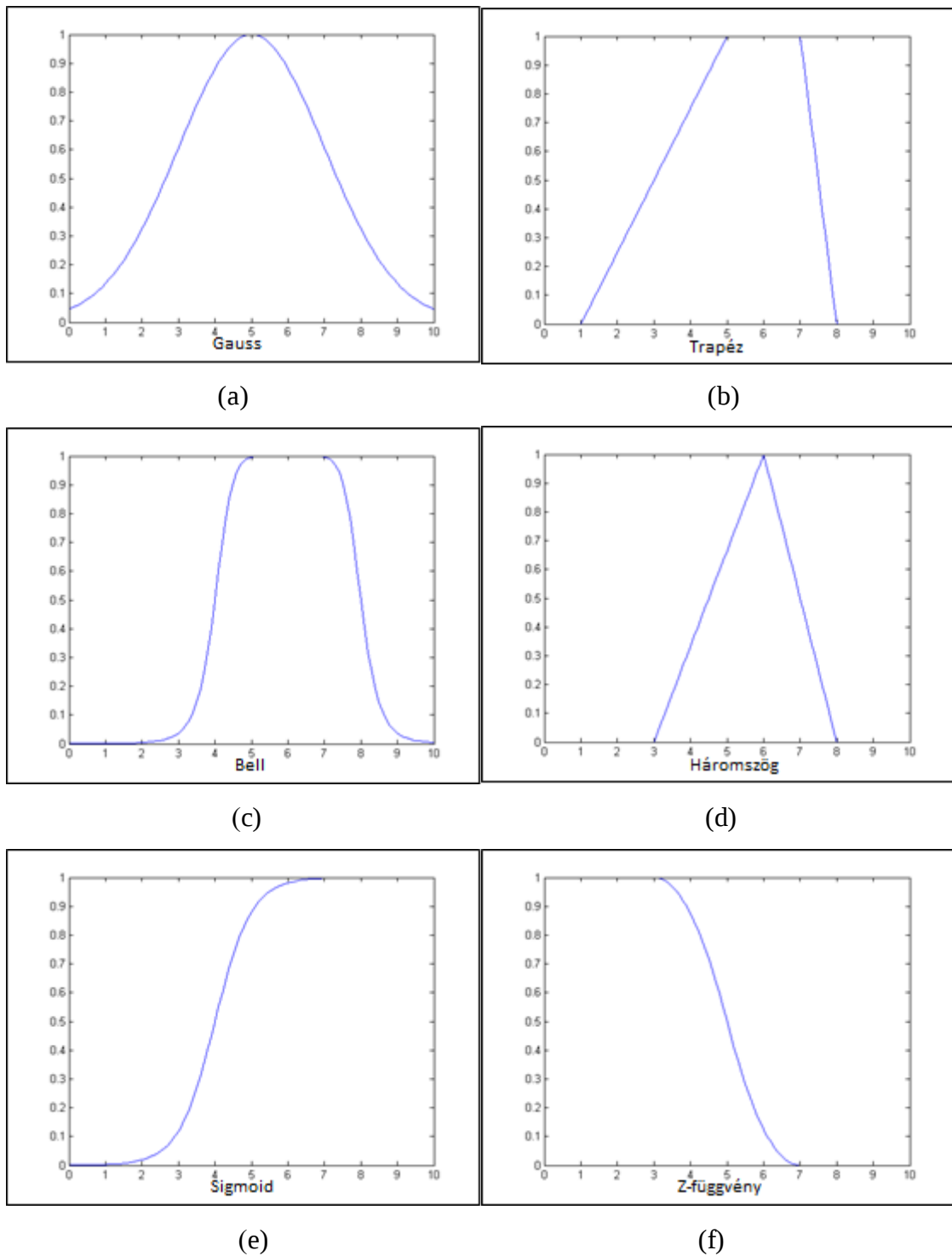
6.4. ábra: (a) Normál és (b) szubnormál fuzzy halmaz

(a)

(b)

6.5. ábra: (a) Konvex és (b) nem konvex fuzzy halmaz

A következő ábrán láthatók a leggyakrabban alkalmazott tagsági függvények. A első képen látható a Gauss-féle függvény, a másodikon a trapéz alakú függvény, azt követik a Bell-féle és háromszög függvények és a végén található a Sigmoidis és Z-féle függvények.



6.6. ábra: A tagsági függvények: (a) Gauss, (b) trapéz, (c) Bell, (d) háromszög, (e) Sigmoid és (f) Z-függvény

6.3. A fuzzy rendszerek

Mint ismeretes, a fuzzy logika képes kifejezni az emberi gondolkodás bizonytalanságát és szubjektivitását. Ezen okok miatt a fuzzy rendszerek (*Fuzzy Inference System – FIS*) számos alkalmazásban megtalálták a helyüket, amelyek esetében nem lehetséges teljesen pontosan definiálni a megoldandó problémát.

A fuzzy rendszer tervezésében és kialakításában kulcsfontosságú szerepe van a fuzzy szabályoknak, amelyek kifejezik a bemeneti változók aktuális értékeinek következményeit, kimeneteit. Ezeket a szabályokat HA-AKKOR („*IF-THEN rules*“) szabályoknak nevezik és a következő módon alakítják ki őket:

IF <feltétel> THEN <következmény>.

Ezeket a szabályokat leginkább az angol elnevezéssel jelölik és a szabályok célja reprezentálni egy adott problémát a fuzzy változókon keresztül. Ezek a szabályok lényegében az össz adat összegét jelképezik a problémáról amit meg kell oldani. A fuzzy változók lényegében lingvisztikus-nyelvi változók. A szabályok leírnak egy fogalmat, amelyet a mindennapi nyelvezetben használnak egy olyan probléma leírására, amelynek „homályos“, vagy pontatlan, azaz *fuzzy* értéke van (pl. a fejezet elején leírt példában az ember életkora olyan fuzzy változó, amelynek a lehetséges értékei a "fiatal", vagy az "idős"). Az össz szabály párhuzamosan hajtódik végre, a sorrend nem számít. A kialakított szabályok listáját szabály bázisnak nevezik. Megfigyelhető, hogy ezek a szabályok nem lehetnek egészen pontosak. Ennek oka, hogy a változók korlátozott halmazból vesznek nyelvi értékeket, ezért korlátozott szabályrendszerünk van. Ezek a szabályok csak absztrakt elképzelések arról, hogyan kellene egy folyamatot kezelni, vagy megoldani egy adott problémát. A cél nem a pontosság, hanem a megoldandó probléma szempontjából releváns tényezők bemutatása.

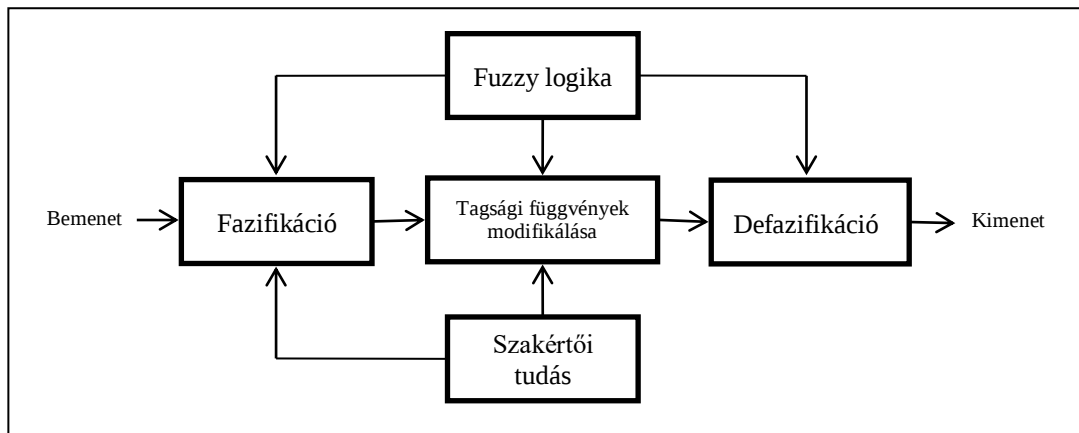
Miután az összes elnevezés, azaz változó és tulajdonság, amely leírja az adott változókat definiálva lettek, megkezdhető a szabályokat értelmező rendszer kialakítása.

A fuzzy rendszer fuzzy szabályokon való tervezésének folyamata röviden leírható a következő lépésekkel:

1. *A bemeneti adatok fazifikációja (Fuzzyfi inputs):* Minden skaláris bemenethez meg kell találni a megfelelő fuzzy értéket azáltal, hogy az adott bemenetet a $[0,1]$ intervallumba kell letérképezni az egyes szabályok megfelelő tagsági függvényével. A fazifikáció lényegében csak a klasszikus halmazokban található „merev” (*crisp*) mennyiségek egyfajta „átírását” jelenti a fuzzy logika alkalmazható formájába. Ezt az eljárást gyakran kódolásnak is nevezik. A műveletet a tagsági függvények teszik lehetővé, amelyek a bemeneti értéket a kijelentés igazságának mértékében térképezik, vetítik le.
2. *A fuzzy logika műveleteinek alkalmazása (Fuzzy logical operations):* Ebben a lépésben össze kell vonni a kimeneti adatokat az összes feldolgozott bemenet fazifikálása után (a szabály IF része, amely leírja a feltételeket), hogy egyedi érték legyen kapható eredményként. Ebben a lépésben a *max*, vagy *min* műveletek kell alkalmazni attól függően, hogy a szabályban lévő részek logikai összeadással (OR) vagy szorzással (AND) vannak összekapcsolva. A feltételek száma (a szabály IF része) és a kapcsolathoz használt logikai operátorok típusa eltérő lehet szabálytól szabályig.
3. *Implikáció alkalmazása (Implication):* Az egyedi kimenetek minden feltétel után (IF rész után), azaz a fazifikációk egy adott szabály következményének (kimenetének) kialakítására szolgálnak. Az implikációhoz logikai szorzást (AND, *min*) vagy összeadást (OR, *max*) használunk. Ez a művelet gyakorlatilag „levágja” a megfelelő tagsági függvényt a feltétel által meghatározott értékekben. Itt megkapjuk az egyes szabályok kimeneti függvényét, következményét, amely megmutatja, hogy a szabály elején kezdődő feltétel mennyiben igaz, azaz mennyire teljesül.

4. *Aggregáció alkalmazása a 3. lépésben kapott fuzzy halmazokra (Aggregation method):* Ebben a lépésben az összes szabály fuzzy halmazainak következményeit és kimeneteit a megfelelő logikai művelettel, az összeadással (OR, max), vagy a szorzással (AND, min) kombinálják. Az összes szabály kimenetének egyesítése kompakt matematikai ábrázolást eredményez a teljes tudásbázisról, amely alapvetően egy felület keresztmetszete.
5. *Defazifikáció (Defuzzification):* Valójában az ellentéte a fazifikáció folyamatának, ezért gyakran dekódolásnak is nevezik. Ez egy olyan folyamat, amelynek át kell alakítania az aggregáció eredményét, amely alapvetően egy felület keresztmetszete, egy olyan függvényre, jellel amely értelmezhető az adott folyamat (feladat) számára. Ebben a lépésben a fuzzy kimenetet (jelet, függvényt) ismét „merev”, hagyományos skaláris alakba alakítják. A kimenetnek egyedi értékkel kell rendelkeznie, amelyet általában egy valós szám képvisel. Erre azért van szükség, mert minden folyamat csak skaláris értékeket ért meg és fogad el bemenetként. A defazifikációnál a leggyakrabban alkalmazott módszerek a következők: a felület közepének meghatározása (gravitáció), az összeg közepének meghatározása, a legnagyobb felület közepének meghatározása stb.

A 6.7. ábrán látható a fuzzy rendszer tervezésének és kialakításának a folyamata.



6.7. ábra: A fuzzy rendszer kialakításának a folyamata

Manapság két típusú fuzzy rendszert alkalmaznak legtöbbször a kutatásokban és gyakorlatban: a Mamdani és Sugeno fuzzy rendszereket.

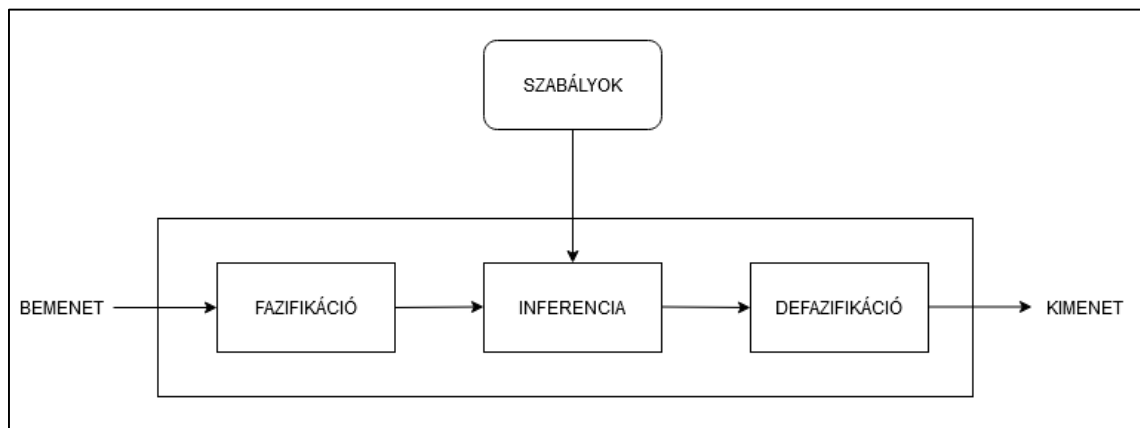
A Mamdani fuzzy inferenciát először egy vezérlő rendszer kialakításához vezették be, ahol az emberi tapasztalatok alapján szintetizálták a lingvisztikai szabályozási szabályok. A Mamdani rendszerben minden egyes szabály kimenete fuzzy halmaz. Mivel a Mamdani rendszereknek intuitívabb és könnyebben érthető szabály bázisuk van, ezért jól alkalmazhatóak olyan szakértői rendszerek alkalmazásaiban, ahol a szabályokat humán szakértői ismeretek alapján hozzák létre, mint például az orvosi diagnosztikában, képfeldolgozásban, alakfelismerésben, jelfeldolgozásban...

Lényegében a Mamdani fuzzy rendszerek kialakításának lépései megegyeznek a fejezet első részében leírt lépésekkel:

1. a bemenetek fazifikációja
2. logikai műveletek alkalmazása, bemeneti aggregáció
3. következtetés levezetése Mamdani implikációval
4. kimeneti aggregáció, végső következtetés létrehozása összeadással
5. defazifikáció.

Meg kell jegyezni, hogy a Mamdani rendszereknél az implikáció mindig logikai szorzással (min), a kimeneti aggregáció pedig logikai összeadással (max) történik.

Tehát ebben az esetben minden szabály kimenete egy fuzzy halmaz, amely a kimeneti tagsági függvényből és a fuzzy inferencia rendszer implikációs módszeréből származik. Ezeket a kimeneti fuzzy halmazokat egyetlen fuzzy halmazzá kell egyesíteni a fuzzy inferencia rendszer összesítési, összeadási módszerével. Ezután a végső „merev“ (*crisp*) kimeneti érték kiszámítása a megfelelő defazifikációs módszerek egyikével lesz meghatározva.



6.8. ábra: A Mamdani fuzzy rendszer

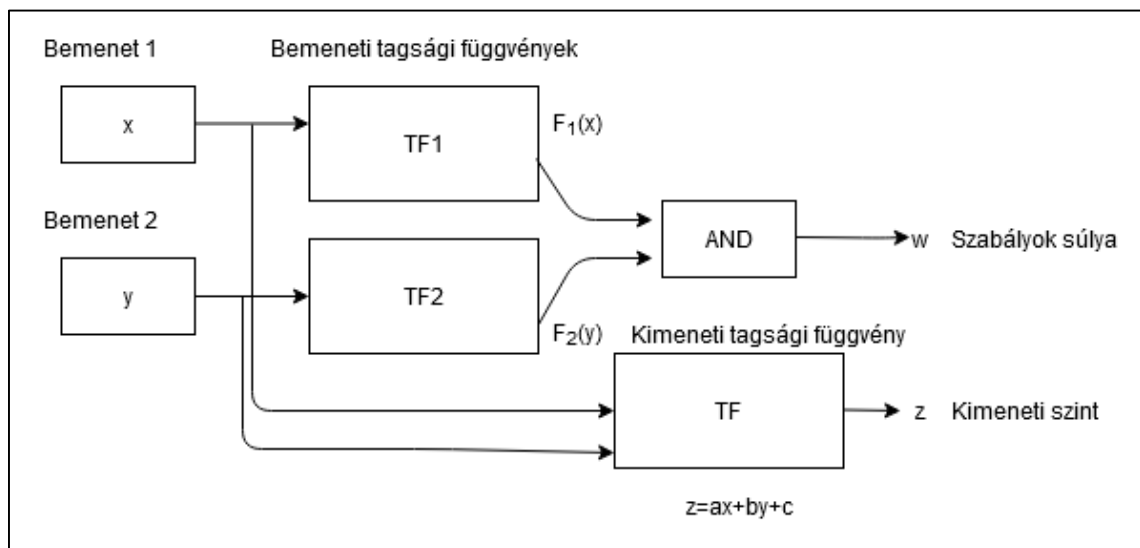
A Mamdani típusú rendszerek esetében a feldolgozás gyakran túlságosan terhelő a számítási időt figyelve. Sajnos az ilyen rendszerek nem alkalmasak olyan rendszerekben való alkalmazásra, ahol fontos szempont a valós idejű működés. Ezért a valós idejű alkalmazásokban, mint pl. a különféle irányítástechnikai alkalmazások más típusú fuzzy inferencia rendszert kell használni, a Sugeno rendszert.

A Sugeno fuzzy inferencia rendszer egyetlenes (*singleton*) kimeneti tagsági függvényeket használ, amelyek konstansak, vagy a bemeneti értékek lineáris függvényei. A Sugeno rendszer defazifikációs folyamata számítási szempontból hatékonyabb, mint egy Mamdani rendszeré, mivel néhány adatpont súlyozott átlagát,

vagy súlyozott összegét használja, nem pedig a kétdimenziós terület centroidját számítja ki.

A Sugeno rendszer jóval egyszerűbb felépítésű, mint a Mamdani rendszer. A Sugeno típusú következtetés abban különbözik a Mamdani rendszerétől, hogy nem léteznek benne a fazifikációs és defazifikációs részek, ami azt jelenti, hogy a Sugeno fuzzy inferencia bemenetére nem vezethető fuzzy halmaz, a kimenetnek pedig crisp értéknek kell lenni. Ebből következik, hogy a Sugeno rendszer esetében a szabályok struktúrája különbözik a Mamdani-féle IF-THEN szabályoktól. A lényeges különbség a szabály THEN részében van, ahol a fuzzy javaslatok helyett lineáris függvény található, amelynek kimenete a bemeneti mennyiségektől függ. Magának a Sugeno rendszernek a sebessége a lineáris függvények alakjától függ, azaz milyen gyorsan lehet az adott függvényekkel számolni.

A Sugeno rendszer minden szabálya az alábbi ábra szerint működik, amely két bemeneti rendszert mutat be az x és y bemeneti értékekkel.



6.9. ábra: A Sugeno fuzzy rendszer

Minden szabály két értéket generál:

- z_i – ez a szabály kimeneti szintje, amely vagy állandó érték, vagy a bemeneti értékek lineáris függvénye

$$z_i = a_i x + b_i y + c_i \quad (6.9)$$

Ebben az esetben az x és y az 1. és a 2. bemenet értéke, az a_i , b_i és c_i pedig állandó együtthatók. Nulladik rendű Sugeno rendszer esetén a z_i állandó, azaz $a = b = 0$.

- w_i – ez a szabály előzményéből származtatott „szabályerő”

$$w_i = \text{AndMethod}(F_1(x), F_2(y)) \quad (6.10)$$

Ebben az esetben az F_1 és F_2 az 1. és 2. bemenetek tagsági függvényeit jelölik.

Minden szabály kimenete a súlyozott kimeneti szint, amely nem más, mint a w_i és a z_i szorzata. Az elsőrendű Sugeno rendszerek (az a és b nem nulla) vizualizálásának legegyszerűbb módja, hogy minden szabályra úgy kell gondolni amint meghatározzák a mozgó *singleton*ok helyét, azaz az egyes kimeneti jelugrások (*spike*) lineárisan mozoghatnak a kimeneti térben, a bemeneti értékektől függően. A súlyozott szabály ezután meghatározza a *singleton*, egyetlen jelugrás méretét.

A rendszer végső kimenete az összes szabály kimenetének súlyozott átlaga:

$$\text{Final Output} = \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i} \quad (6.11)$$

ahol az N a szabályok számát jelöli.

Mivel az összes szabály lineárisan függ a bemeneti változóktól, a Sugeno rendszer ideális több lineáris vezérlő interpolációs felügyeletének ellátására, amelyeket egy dinamikus nemlineáris rendszer különböző működési körülményeire kell alkalmazni. Hasonlóképpen, a Sugeno rendszer alkalmas nemlineáris rendszerek modellezésére, több lineáris modell közötti interpolációval.

Fuzzy rendszer	Előnyök
Mamdani	• intuitív • jól illik a humán bemenetekhez • jobban interpretálható a szabályok bázisa • széleskörűen elfogadott
Sugeno	• adatfeldolgozási és számítási szempontból hatékony • jól alkalmazható a lineáris technikáknál, mint pl. a PID vezérlőnél • jól alkalmazható az optimizációs és adaptív technikáknál • jól alkalmazható a matematikai elemzésekben • biztosított a kimeneti felület folytonossága

6.1. Táblázat: A Mamdani és Sugeno rendszerek előnyeinek összehasonlítása

A fuzzy logikával kapcsolatos fejezet végén egy példa segítségével lesz szemléltetve a fuzzy logika hatékonysága az objektumok felismerésében. A példa lényege detektálni és kiemelni a rendszám táblát a jármű és környezetének összetett képéből. A példában használt kép szerény minőségű kamerával lett rögzítve, ezérta kép minősége igen gyenge. A képben található különböző zajok és zavarok. A példa alapja egy fazifikált 2D Gábor szűrő, amelynek a szögei és hullámhosszai lettek fazifikálva abból a célból, hogy már maga a szűrő jól kiemelje a rendszám táblát és annak közvetlen, szűk környékét. Mint ismeretes, a Gábor szűrő igen hatékonyan emeli ki a képben a keresett texturákat abban az esetben, ha a szűrő paraméterei megfelelően vannak beállítva. Ebben a példában a paraméterek beállítása fuzzy logika alkalmazásával lett elvégezve. A fuzzy rendszer Mamdani típusú. A fuzzy Gábor szűrő alkalmazása után, a rendszám tábla bináris maszkja

morfológiai műveletek sorozatával lett létrehozva, majd végül a végső bináris maszk segítségével megtörtént a rendszámtábla detektálása és kiemelése.

A példában lényegében egy szűrőbank lett létrehozva, mert egyidőben több paraméter segítségével történik a szűrés. Ebben a konkrét esetben empirikusan be lett bizonyítva (de a szakirodalom is ezt támasztja alá), hogy a szögek és a hullámhosszak hatnak ki dominánsan a szűrési eredményre. A járműveket gyakran oldalról és különböző távolságról fényképezik, aminek következtében ingadozik a rendszámtábla dőlési szöge és mérete is. Természetesen lehetetlen az összes lehetséges szöget és hullámhosszat figyelembe venni, de egy meghatározott intervallumon belül figyelve az említett paramétereket lehetséges létrehozni hatékony fuzzy rendszert. A fuzzy logika alkalmazásával a szűrő hatékonysága fokozódott a hagyományosan meghatározott szűrőparaméterek alkalmazásához képest. A szűrő összes paramétere nem lett fazifikálva, mert a túlságos fazifikáció másfelől megnövelné a feldolgozási időt, ami ez esetben nagy kihatással lehet a teljes rendszerre, mert digitális képfeldolgozásról van szó, a képek pedig nagyméretű adatoknak számítanak feldolgozás szempontjából.

A fazifikáció folyamatában bemeneti tagsági függvényként a Bell-féle függvény lett kiválasztva az orientációk (szögek) és a hullámhosszak esetében is. Ennek oka a Bell-féle függvény alakja, mert a középponti értékektől a függvény mindkét irányban egyenletesen csökken és így megfelelően magába foglalja az érdekelt értékeket, mindkét paraméter esetében. A központi értékek lényegében azok az értékek, ahol a tagsági függvények értéke pontosan egy. A szögek esetében ezt az összefüggést a következő kifejezéssel lehet leírni:

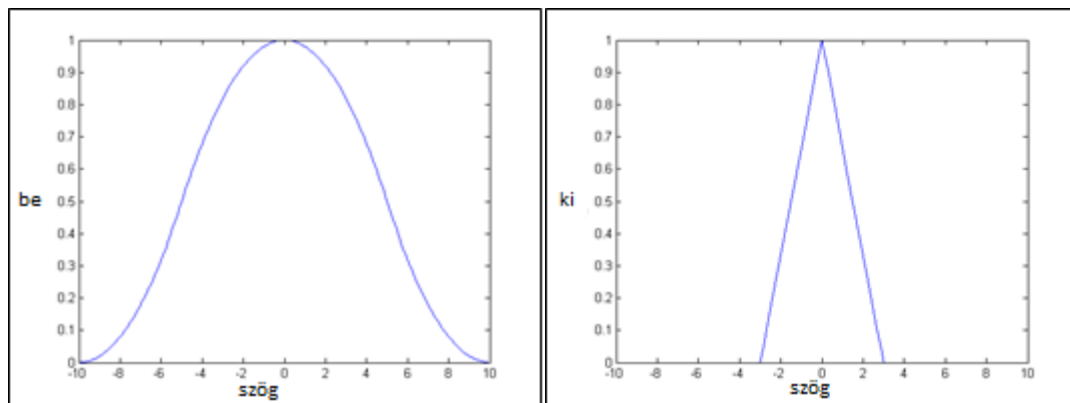
$$\mu(z) = \begin{cases} S(z, \phi_1, \phi_2) & z < \phi_2 \\ S(2\phi_2 - z, \phi_1, \phi_2) & \phi_2 \leq z \end{cases} \quad (6.12)$$

Természetesen a hullámhosszak esetében is hasonlóképpen nézne ki a kifejezés, csak a ϕ helyett λ szerepelne a képletben. Mindkét esetben a fazifikálni kívánt paraméterek egy adott intervallumon belül helyezkednek el.

A kimeneteken a háromszög-féle tagsági függvény lett alkalmazva mindkét paraméter fazifikálásánál. A kimeneten szűkíteni szükséges az intervallumokat a központi értékek körül, a jobb szűrés érdekében. Ebben az esetben is a központi érték az, ahol a tagsági függvény értéke egy, amivel lényegében a fuzzy halmazhoz való hozzátartozás mértéke van leírva (az egy esetében az adott változó teljesen az adott fuzzy halmazhoz tartozik). Ezt az összefüggést a háromszög típusú függvény esetében a következő kifejezés írja le:

$$\phi_{crisp} = \frac{\phi_i + \phi_f}{2} \quad (6.13)$$

ahol a $\mu(\phi_{crisp}) = 1$ a 0° értékű szög esetében, ϕ_i, ϕ_f a kimeneti szögintervallumok határai. Az elv teljesen megegyezik a hullámhosszak fazifikálása esetében is.

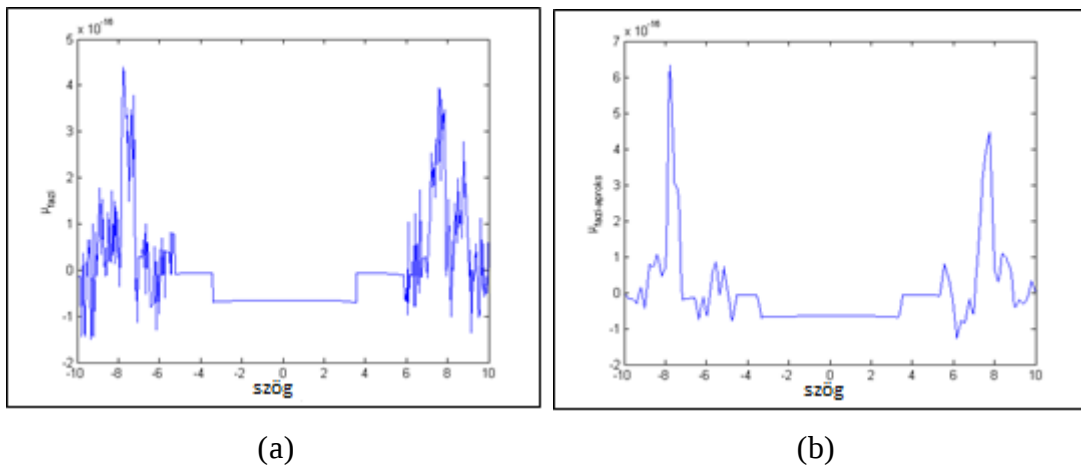


(a)

(b)

6.10. ábra: A tagsági függvények a szögek fazifikálása esetében: (a) bemeneti és (b) kimeneti

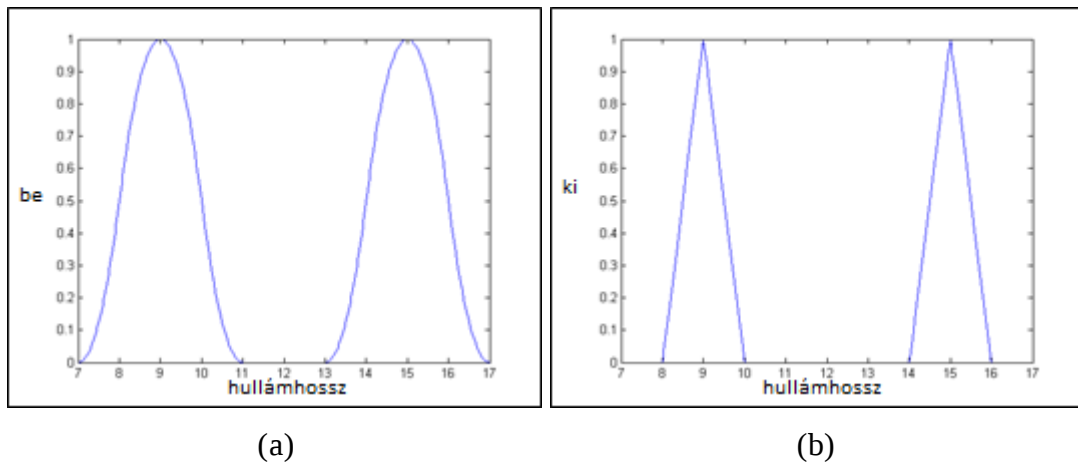
A 6.10. ábrán láthatóak a tagsági függvények a szögek fazifikálása esetében.. Az abcisszán a szögek vannak jelölve, míg az ordinátán a tagsági függvények értékei. Szépen láthatóak az adott paraméter fazifikáció közben figyelt értékei.



6.11. ábra: A fuzzy rendszer szögek esetében: (a) eredeti és (b) aproximáció után

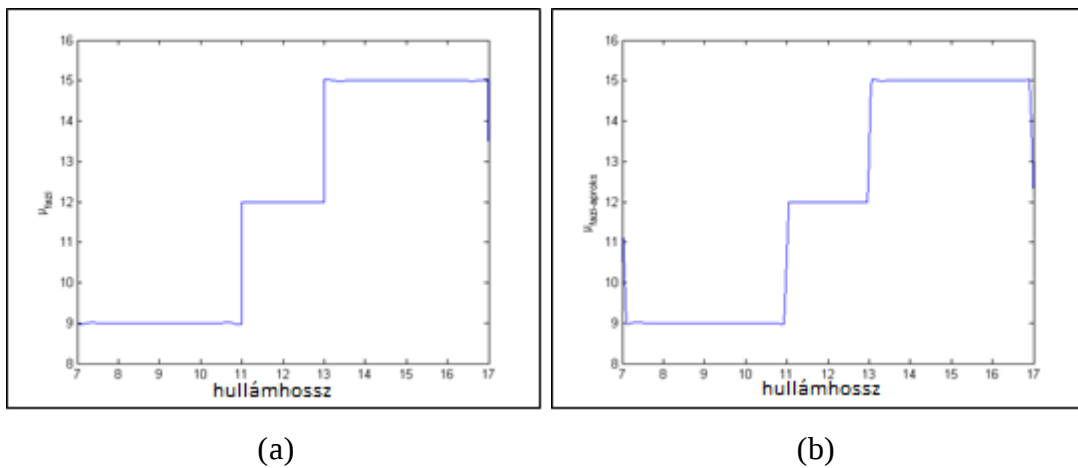
A 6.11. ábrán láthatók a szögek esetében a fuzzy rendszerek. A második ábrán látott rendszer az első ábrán látott rendszer aproximációjával lett meghatározva egy *Look-Up* táblázat segítségével. Ennek célja az eredeti rendszer „kisimítása” a feldolgozási idő és számítási összetettség csökkentésének szemszögéből. Ez az aproximáció abszolút nem hat ki a szűrés minőségére, de csökken a feldolgozási idő.

A következő két ábrán a hullámhosszak fazifikációjának menete lesz bemutatva. Magában a módszerben nincs semmi különbség, a lényeges eltérés abban van, hogy a hullámhosszak esetében 2 intervallum lesz figyelve a bemeneten és a kimeneten is.



6.12. ábra: A tagsági függvények a hullámhosszak fazifikálása esetében: (a) bemeneti és (b) kimeneti

Ebben az esetben az abcisszán a hullámhosszak vannak jelölve, míg az ordinátán a tagsági függvények értékei. Szépen láthatók az adott paraméter fazifikáció közben figyelt értékei, de most két intervallumban.

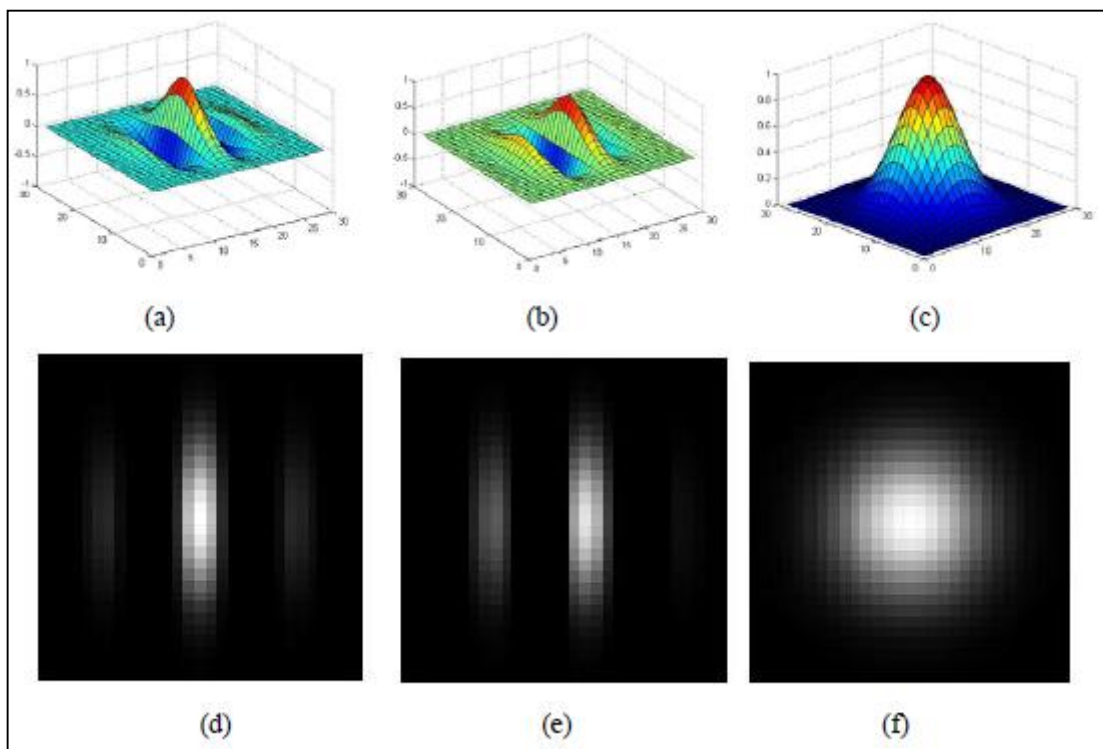


6.13. ábra: A fuzzy rendszer hullámhosszak esetében: (a) eredeti és (b) aproximáció után

A 6.13. ábrán láthatók a hullámhosszak esetében a fuzzy rendszerek. A második ábrán látott rendszer az első ábrán látott rendszer aproximációjával lett

meghatározva *Look-Up* táblázat segítségével ebben az esetben is. Itt észrevehető, hogy nincs nagy különbség a két fuzzy rendszer között. Ennek oka, hogy az eredeti fuzzy rendszerben is „lassan“ változnak az értékek, nem olyan „gyorsan“, mint a szögek esetében.

A 6.14. ábrán látható egy fazifikált Gábor szűrő egy adott pontban. A fazifikációval ehhez hasonló szűrők sorozata lett előállítva és gyakorlatilag egy fuzzy szűrőbank keletkezett a fazifikáció után, amivel majd szűrve lesz a bemeneti kép. Az első három kép a szűrő térbeli alakját mutatja, a második három kép pedig a spektrumot. Az első oszlop a szűrő valós részeit, a második oszlop a szűrő imagináris részeit mutatja, míg a harmadik oszlopban látható a szűrő amplitúdója.



6.14. ábra: *A fuzzy 2D Gábor szűrő egy adott pontban*

A képen látható apró négyzetek a hullámhosszak kicsi értékeinek a következményei.

Példa következik, amely bemutatja a fazifikáció gyakorlati alkalmazását az objektumok felismerésében. A 6.15. ábrán az első képen látható egy jármű fekete-fehér árnyalatokba alakított képe. A második képen látható a fuzzy 2D Gábor szűrő szűrésének az eredménye. Mint megfigyelhető, a rendszámtábla környéke szinte tökéletesen ki van emelve.

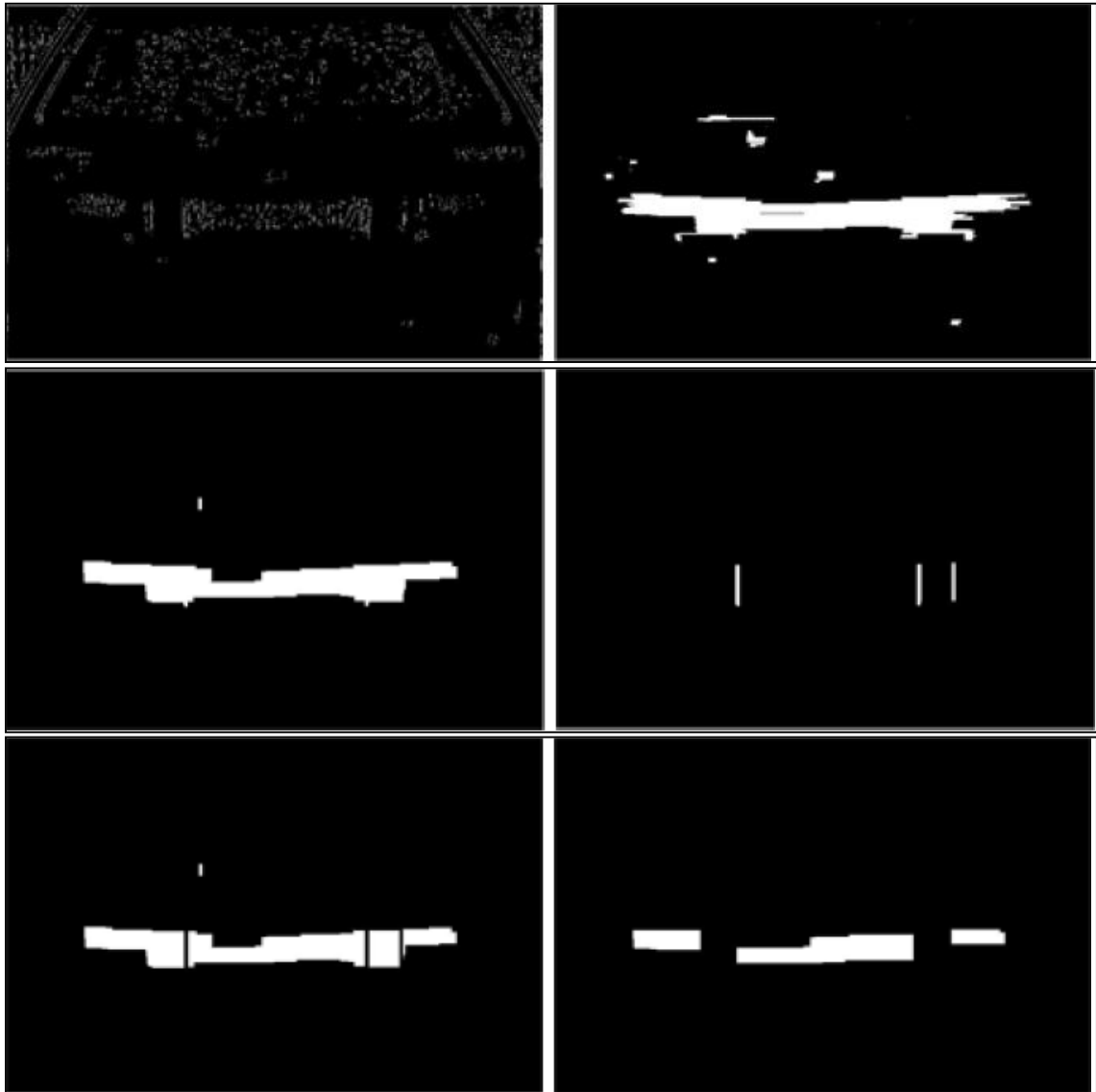


(a)

(b)

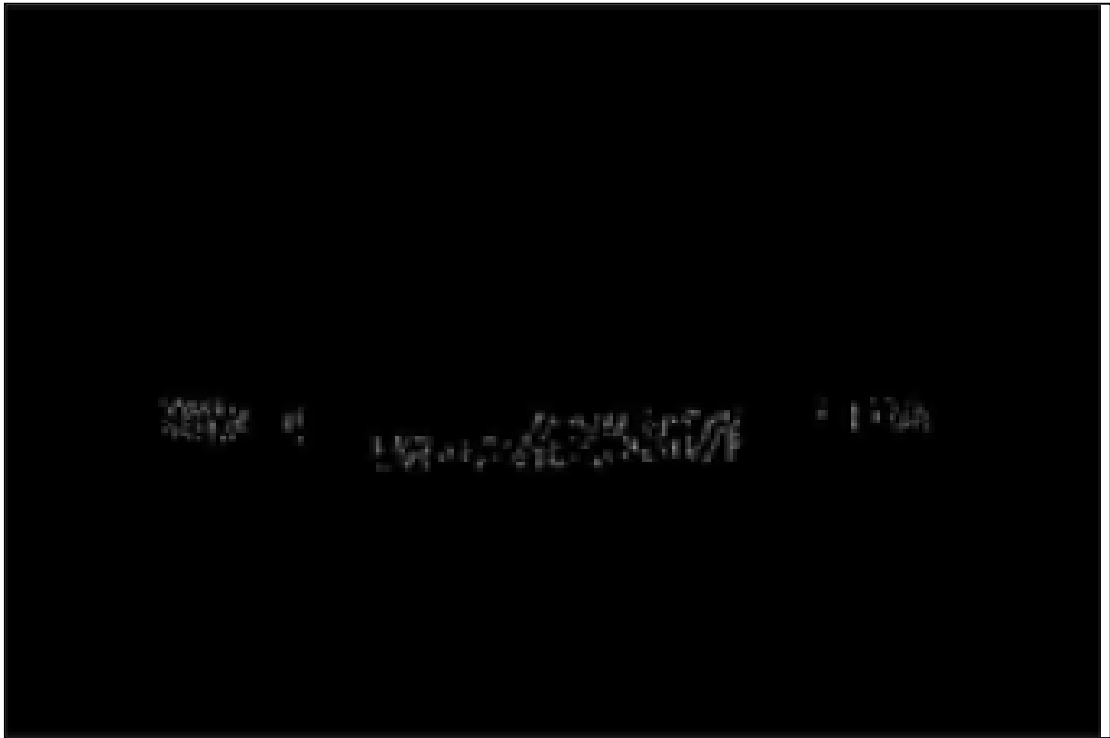
6.15. ábra: (a) Bemeneti kép és (b) fuzzy 2D Gábor szűrő hatása

A szűrés után következik az éldetektálás függőleges Sobel operátor segítségével, majd azt követi a morfológiai műveletek sorozata, amelyek célja létrehozni a rendszámtábla bináris maszkját, ez fogja detektálni rendszámtáblát. A második képen zárás művelet volt alkalmazva vízszintes SE elemmel, azt követi két egymás utáni nyitás művelet, az ötödik képen egy kivonás művelete látható, amit követ egy végső nyitás művelet.



6.16. ábra: *A Sobel éldetektor alkalmazása és a morfológiai műveletek sorozata*

A következő képen a végső nyitás eredménye binárisan lett összeszorozva a Sobel operátor által kapott bináris képpel. Ennek a műveletnek a célja kiemelni a rendszámtábla szűkebb környékében az összes vertikális elemet. A rendszámtábla karakterekből (betűkből és számokból) áll, amelyek többnyire vertikális elemeket tartalmaznak és így már a kezdetekben ki vannak szűrve a felesleges képkomponensek. A szorzás eredménye a 6.17. ábrán látható.



6.17. ábra: *A rendszámtábla környékében található vertikális képkomponensek*

A rendszámtáblához tartozó képkomponens csoport a morfológiai összekötött-komponensek elemzésével lett meghatározva és a felesleges, nem a rendszámtáblához tartozó képkomponensek ez alapján el lesznek távolítva a 6.16. ábrán található bináris képből, így lesz meghatározva a végső bináris maszk. Ez a bináris maszk végül össze lesz szorozva binárisan a bemeneti kép bináris alakjával a rendszámtábla kiemelése céljából.



6.18. ábra: (a) Bináris maszk és (b) kiemelt rendszámtábla bináris szorzással

Mint látható a 6.18. (b) ábrán, a rendszámtábla tartalma nem látható teljesen, ez nyilván a bemeneti kép hibáiból és a szerény minőségű felszerelés alkalmazásából ered.

A végén a bináris maszk és a kiemelt rendszámtábla kereteinek morfológiai kivizsgálása után ki lesz alakítva egy négyszögkeret, amely segítségével a rendszámtábla be lesz jelölve a bemeneti képben, majd e keret segítségével ki lesz vágva a tábla az eredeti színes képből.



6.19. ábra: (a) Megjelölt rendszámtábla és a (b) kiemelt rendszámtábla

A 6.19. ábra első képén a rendszámtábla egy piros színű kerettel van megjelölve, majd ugyanez a keret alapján a rendszámtábla ki is van emelve a képből.

Mint látható, a rendszám tábla sikeresen lett detektálva az összetett, gyenge minőségű képben a fuzzy módszer alkalmazásával.

A következő rövid példában be lesz mutatva a fuzzy módszer hatékonysága a sima, nem fuzzy Gábor szűrőbankkal szemben. Erre a célra egy olyan kép lett kiválasztva, ahol a jármű igen ferde szög alatt lett fényképezve.

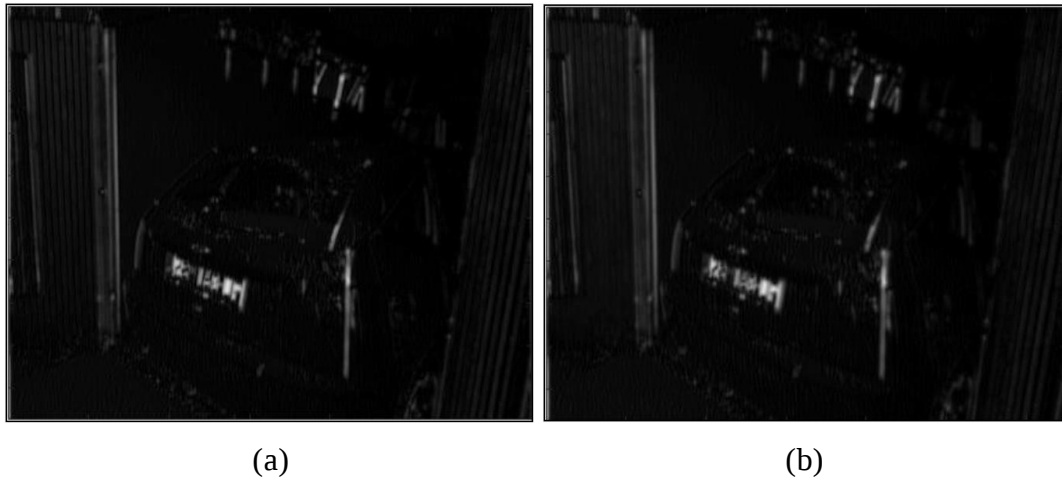


(a)

(b)

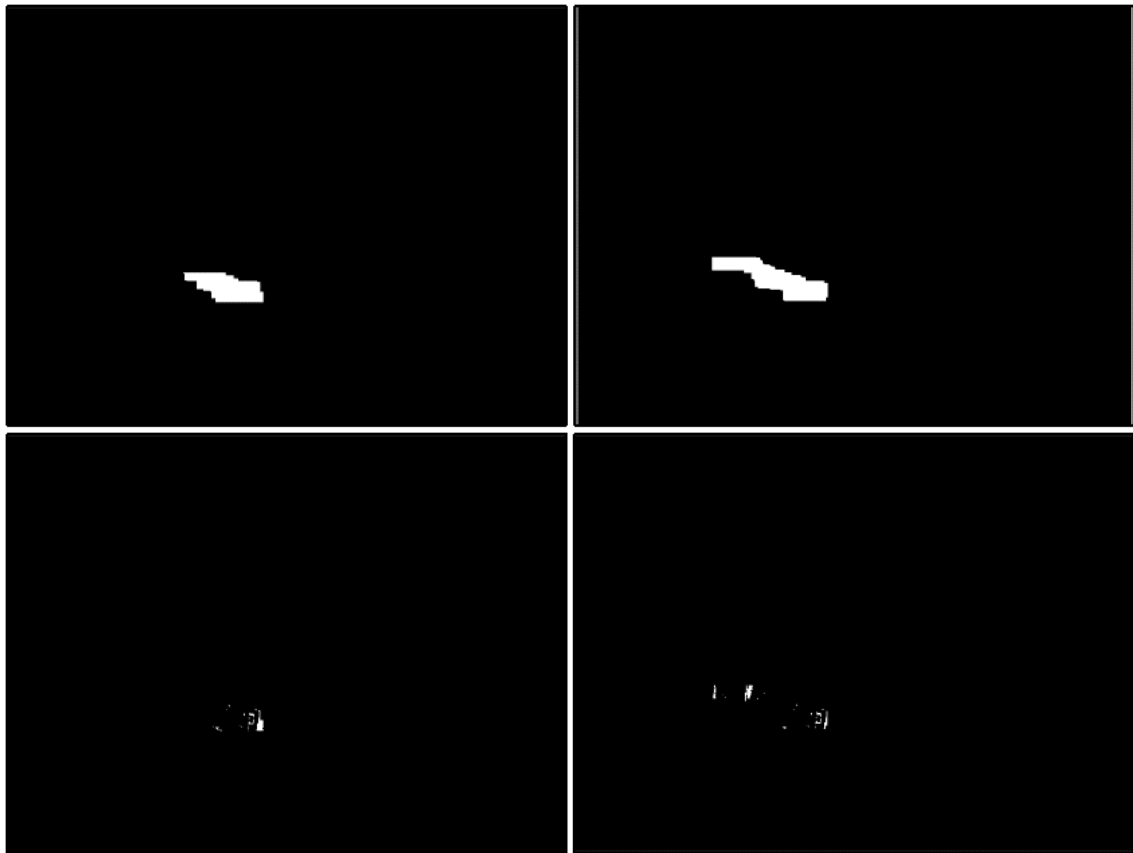
6.20. ábra: A bemeneti kép: (a) színesben és (b) szürke árnyalatokban

A 6.21. ábrán láthatók a Gábor szűrőbank és a fuzzy Gábor szűrő szűréseinek eredményei. Megfigyelhető, hogy a szűrőbank esetében a rendszám tábla környékében kicsit gyengébb a szűrőválasz.



6.21. ábra: Szűrés eredményei: (a) Gábor szűrőbank és (b) fuzzy Gábor szűrő

A következő ábrán a bináris maszkok láthatók mindkét szűrésváltozat után és a kiemelt rendszám tábla környezetek. Bal oldalon található a Gábor szűrőbank példája, jobb oldalról pedig a fuzzy Gábor szűrés példája. Észrevehető, hogy a fuzzy Gábor szűrő esetében a bináris maszk mérete szélesebb, ami azt jelenti, hogy a rendszám tábla nagyobb eséllyel lesz hatékonyan detektálva. Az egyszerű szűrőbank esetében a bináris maszk egy része le lett vágva, aminek következménye lehet a hiányosan detektált rendszám tábla. Ez esetben sem látható a bináris rendszám tábla tartalom, a rossz minőségű bemeneti kép és gyenge minőségű fényképezőgép miatt.



6.21. ábra: *A bináris maszkok és a bináris rendszámtáblák tartalmai*

Végül, a 6.22. ábrán láthatók a detektált és a binarás maszkok alapján kiemelt rendszámtáblák. Megfigyelhető, hogy az egyszerű Gábor szűrő esetében a rendszámtábla hiányos, míg a fuzzy Gábor szűrő esetében a rendszámtábla teljes tartalma ki van emelve. Ennek oka, hogy a fuzzy Gábor szűrő az algoritmus elején sokkal jobban érzékeli a rendszámtáblát és annak környékét a fazifikált paramétereknek köszönhetően, hiszen a fuzzy logika sokkal rugalmasabban illeszkedik az adott problémához és jobban érzékeli a túlságosan elferdült rendszámtáblát.



(a)

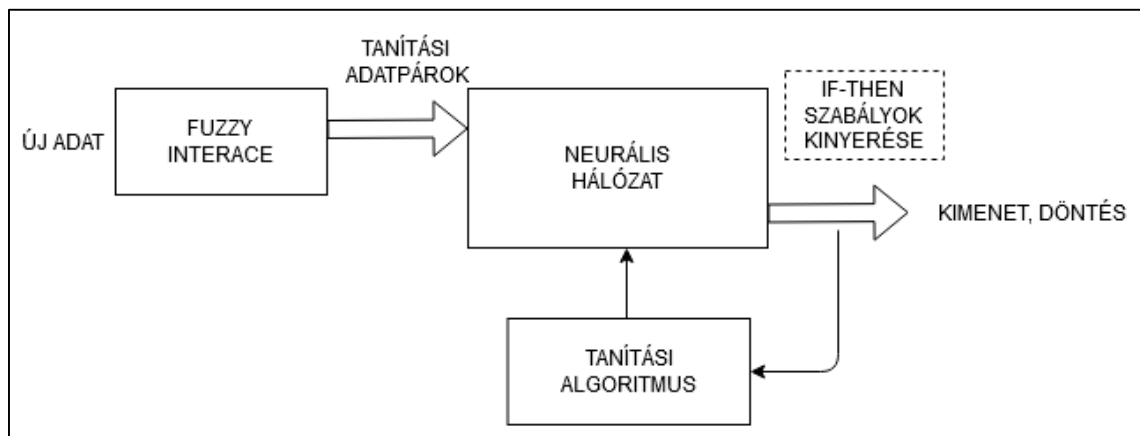
(b)

6.22. ábra: A detektált rendszám táblák: (a) egyszerű szűrőbank és (b) fuzzy Gábor szűrő esetében

6.4. Az ANFIS

Az adaptív neuro-fuzzy inferenciás rendszerek (ANFIS) a hibrid intelligens rendszerek modern osztályát képviselik. Ezeket a rendszereket a fuzzy paraméterekkel meghatározott mesterséges neurális hálózatokként lehet legjobban jellemezni és leírni. A mesterséges intelligencia két különböző koncepciójának ötvöztetésével a homogén felépítésű hibrid rendszerekben, a fuzzy logika és a mesterséges neurális hálózatok előnyeit kívánják kihasználni az ANFIS rendszerekben. Az ilyen rendszereket egyre inkább alkalmazzák a mindennapi összetett problémák megoldására. A fuzzy modellek idegháló formájában történő megjelenítésének képességét leggyakrabban a fuzzy modell paramétereinek automatikus meghatározásánál használják a bemeneti-kimeneti adatbázis alapján. Az ANFIS rendszerek megalkótójaként Roger Jang-ot tartják számon.

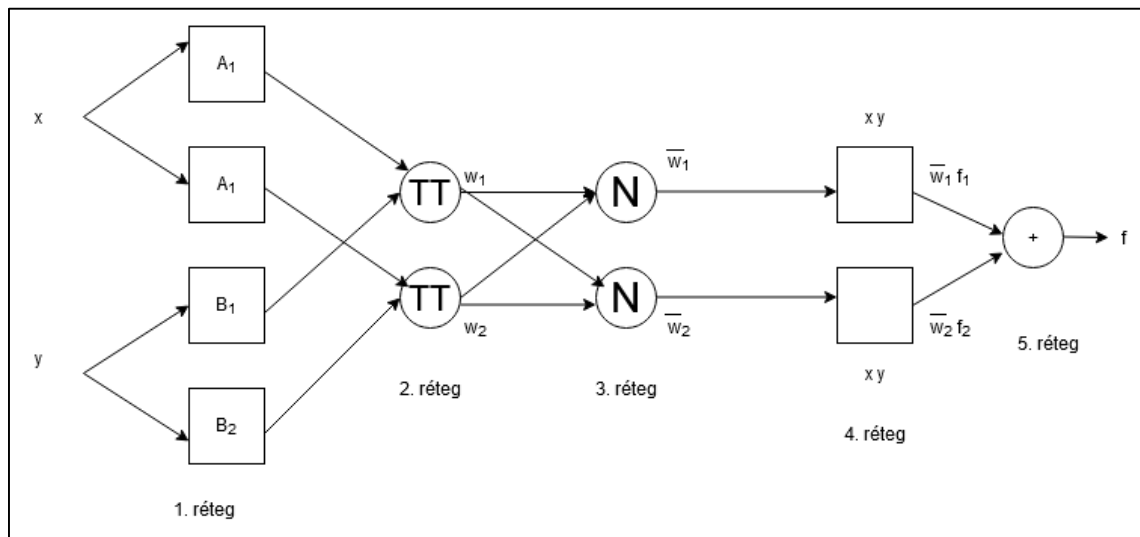
Az ANFIS alapgondolata a fuzzy modellezésen és az adott adatsoron alapuló tanulási módszereken alapszik. A neuro-fuzzy intelligens rendszerek felépítése hasonló a neurális hálózatok felépítéséhez. A bemenet-kimenet adatkészlet alapján megfelelő fuzzy rendszer jön létre, ahol ki lesznek számítva a tagsági függvények megfelelő paraméterei. A megfelelő tagsági függvényekre jellemző paraméterek a hálózati tanulási folyamat révén meg fognak változni az ANFIS neurális hálózat részében. Ezeknek a paramétereknek a kiszámítása általában a hibagradiens vektoron alapszik, a fuzzy rendszer bemeneti halmazból kimeneti halmazba való leképezésénél.



6.23. ábra: Az ANFIS rendszer struktúrája

A fuzzy rendszer tagsági függvényeinek paraméterei a visszafelé-propagációs algoritmussal (*backpropagation algorithm*), vagy a *backpropagation* algoritmus és a legkisebb négyzetek módszer kombinációjával állíthatók be. Ez a beállítás lehetővé teszi a fuzzy rendszerek számára, hogy a bemenet-kimenet adatbázis alapján tanuljanak. Ez a módszer nagyjában hasonlít a neurális hálózatoknál alkalmazott tanulási módszerhez.

Fontos megjegyezni, hogy az ANFIS rendszerek kizárólag Sugeno fuzzy rendszereket támogatnak és ezért csak a Sugeno típusú rendszereket lehet felhasználni az ANFIS rendszerek tervezésében és kialakításában. Ez a korlátozás a Sugeno rendszerek tulajdonságaiból, előnyeiből ered, ahol az említett rendszerek legfőbb tulajdonsága a hatékony adatfeldolgozás és a jóval rövidebb számolási idő, a Mamdani rendszerekkel ellentétben. Az ANFIS rendszereknél ez külön kifejezésre jut, mert el kell végezni a neurális hálózat tanítását is, ami plusz terhelő szempont a számítási idő és adatfeldolgozás szempontjából.



6.24. ábra: Az két szabályos elsőrendű Sugeno fuzzy rendszer megfelelő ANFIS struktúrája

Ebben az esetben a Sugeno fuzzy rendszer a következő szabályokkal írható le:

- 1. szabály: Ha az x az A_1 és az y az B_1 , akkor $f_1 = p_1x + q_1y + r_1$
- 2. szabály: Ha az x az A_2 és az y az B_2 , akkor $f_2 = p_2x + q_2y + r_2$

ahol a Sugeno fuzzy inferencia rendszer kimenete a következő kifejezéssel lesz meghatározva:

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} = \bar{w}_1 f_1 + \bar{w}_2 f_2 \quad (6.13)$$

ahol a w változók a normalizáció súlyozási értékeit képezik. A gyakorlatban a kapott ANFIS architektúra tovább alakítható különféle módon, pl. a 3. és 4. szint összevonható és az így kapott architektúra csak 4 szintből állna. Természetesen más átalakítások is elvégezhetők, de ez nem ennek a tananyagnak a témája és a szakirodalomban, amely szűken ezzel a témával foglalkozik megtalálhatók a leírások.

Annak ellenére, hogy a számítástechnika rohamosan fejlődik, az ANFIS rendszereket többnyire az irányítástechnikában és az 1D jelfeldolgozásban használják (hangfeldolgozás, 1D biomedicinai jelek feldolgozása stb.). A képfeldolgozásban az ANFIS-t ritkábban alkalmazzák, mert a képek igen nagy adatok és probléma a számítási idő és a feldolgozási összetettség. Az ANFIS alkalmazásának tipikus példája a képfeldolgozásban a karakterfelismerés. Ebben az esetben a karaktereket kisméretű képmátrixokkal mutatják ki, amelyeket később egydimenziós vektorokba alakítanak a tanítási folyamat céljából. Ezeket a kisméretű képmátrixokat az ANFIS rendszer képes belátható időn belül feldolgozni, míg a legtöbb esetben jelentkező nagyméretű képet az ANFIS rendszer nehezen tudja kivitelezni, vagy nagyon hosszú ideig tartana az adatfeldolgozás.

Az adaptív neuro-fuzzy rendszerek modellezésének folyamata a következő lépésekből áll:

1. A bemeneti-kimeneti adathalmaz meghatározása a neuro-fuzzy következtetési (inferencia) rendszer tanításához, kiképzéséhez megfelelő, adaptált formában.
2. Feltételezve lesz a modell felépítése a paraméterekkel együtt, amely a szabályok alapján feltérképezi a kimeneti tagsági függvényekhez tartozó bemeneti tagsági függvényeket.
3. A modell tanítási adatok alapján fog tanulni. A tagsági függvények paraméterei a kiválasztott hibakritériumnak megfelelően módosulnak tanítás közben úgy, hogy a modell eredményei validálhatóak (érvényesek) legyenek.

Általános esetben ez a modellezési módszer akkor megfelelő, ha a tanítási adatok (az edzésadatok) teljes mértékben reprezentálják mindazokat a tulajdonságokat, amelyekkel a neuro-fuzzy inferencia rendszernek rendelkeznie kell. A gyakorlatban ez nem mindig van így. Bizonyos esetekben a tanításhoz használt adatok kisebb-nagyobb mértékben mérési hibákat tartalmaznak, így az adott adatok nem reprezentálják teljes mértékben a modell összes jellemzőjét. Ezért a modellt tesztadatokkal kell ellenőrizni. Kétféle módon lehet tesztelni a neuro-fuzzy

modelleket. A modell ellenőrzésének első módja, amikor a tanításhoz nem használt adatokat adják meg bemenetként. Ez az eljárás megmutatja, hogy a modell pontosan megjósolja-e a kimeneti értékek halmazát. A modellek tesztelésének egyik problémája az adathalmaz megválasztása, hogy azok az adatok kellően különbözzenek azoktól az adatoktól, amelyeket nem használtak fel a modell tanítása során. A modell ellenőrzésének másik módja pedig egy matematikai eljárás. Ha tesztadatokat használnak tesztadathalmazként, akkor a minimális lehető hibával kell a kimenetet meghatározni.

6.5. A fuzzy klaszterizáció

A klaszterizáció a felügyelet nélküli gépi tanulás egyik formája. Ebben az esetben az adatokból rendelkezésre állnak azok a példányok, klaszterek adatai, amelyeket valamilyen módon csoportosítani kell. Ami fontos, nincs előre ismert adat a kívánt/helyes csoportról a klaszterekhez.

A klaszterizáció lényegében az adatok, adatképzőminták csoportosítása olyan módon, hogy minden példány jobban hasonlít a saját klasztere példányaihoz, mint más klaszterek példányaihoz. Az adatképzőminták hasonlóságát valamilyen mérték alapján kell meghatározni:

- hasonlósággal, mint pl. a Kosinus-hasonlóság,
- a két adatképzőminta közötti távolsággal, mint pl. az Euklidészi távolság.

A klasszifikációtól eltérően, a klaszterizációnál nem létezik „pontos” eredmény és ezért az algoritmus sikerességének felmérése jóval nehezebb, mint a klasszifikáció esetében. A megoldás helyessége és megfelelése nagyban függ az alkalmazási területtől, mert egyetlen eredmény sikeressége teljesen másképp értelmezhető két külön esetben. Szakértői véleményezés szükséges a helyes megoldás kiválasztásához.

Legyen a fuzzy klaszterizáció problémája felállítva a következőképpen:
Figyelve legyen egy véges számú elemekkel létező $X = [x_1, x_2, \dots, x_n]$ halmaz, ahol

a halmaz elemei pontokat képeznek a p -dimenziós Euklidi térben, $\Re^p$. A probléma abban van, hogy meg kell határozni az adott elemek kolekciójának partícióját a c fuzzy halmazokban valamilyen kritérium szerint, ahol a c az előre megadott klaszterek számát jelöli. A kritérium leginkább egy *objektív függvény*, amely a klaszterizációhoz hozzárendel valamilyen performancia indexet. A klaszterizáció végeredménye kifejezhető egy ún. U partíció mátrixszal:

$$U = [u_{ij}]_{i=1,\dots,c; j=1,\dots,n} \quad (6.15)$$

ahol az u_{ij} egy szám a $[0,1]$ intervallumból és kifejezi az x_j elem hozzátartozási fokát az i -dik klaszterhez. Azonban két lényeges korlátozás létezik az U mátrix koeficiensei esetében. Az első, hogy az x_j elem minden klasszához való hozzátartozása 1 kell legyen, azaz:

$$\sum_{i=1}^c u_{ij} = 1 \quad \text{ha } j = 1, 2, \dots, n \quad (6.16)$$

A másik lényeges dolog, hogy minden klaszter nullától több elemet, de n -től kevesebb elemet kell, hogy tartalmazzon, azaz:

$$0 < \sum_{j=1}^n u_{ij} < n \quad \text{ha } i = 1, 2, \dots, c \quad (6.17)$$

Tovább, az objektív függvény a következő kifejezéssel írható fel:

$$J(u_{ij}, v_k) = \sum_{i=1}^c \sum_{j=1}^n \sum_{k=1}^c g[w(x_j), u_{ij}] d(x_j, v_k) \quad (6.18)$$

ahol a $w(x_j)$ az előre meghatározott súlyozást jelöli, amit hozzáadnak az x_j elemhez és a $d(x_j, v_k)$ jelzi az eltérés, megkülönböztetés fokát az x_j elem és a v_k elem között, amit lényegében a k -adik klaszter középpontjának lehet tekinteni.

A megkülönböztetés foka az a mérték, amelynek teljesítenie kell a következő két axiómát:

$$\begin{aligned} d(x_j, v_k) &\geq 0 \\ d(x_j, v_k) &= d(v_k, x_j) \end{aligned} \quad (6.18)$$

ahol ez az elv gyengébb, szabadabb, mint a távolság mértéke. Ezzel az alappal a fuzzy klaszterizáció precízen megfogalmazható, mint egy optimalizációs probléma ahol optimalizálni kell a J függvényt a felsorolt korlátozások mellett. Az egyik általánosan elfogadott fuzzy klaszterizációs módszer a *fuzzy c-mean* klaszterizációs módszer, amelyet Bezdek alakított ki 1981-ben.

Ebben az esetben a performancia index a következő kifejezéssel van megadva:

$$J(u_{ij}, v_k) = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m \|x_j - v_k\|^2 \quad \text{ha } m > 1 \quad (6.19)$$

ahol az m jelöli az ún. exponenciális, nehézségi tényezőt, amely kihat az U mátrix fazifikációjának fokára. Az említett kritérium minimalizációjának céljából előbb el kell végezni a differenciálást a $v_j, j=1, \dots, c$ elemek szerint, azután pedig az U mátrix elemei szerint is. Ennek eredménye a következő két egyenlet:

$$v_i = \frac{1}{\sum_{j=1}^n (u_{ij})^m} \sum_{j=1}^n (u_{ij})^m x_j \quad \text{ha } i = 1, 2, \dots, c \quad (6.20)$$

$$u_{ij} = \frac{(1/\|x_j - v_i\|^2)^{1/(m-1)}}{\sum_{k=1}^c (1/\|x_j - v_k\|^2)^{1/(m-1)}} \quad \text{ha } i = 1, \dots, c; j = 1, \dots, n \quad (6.21)$$

A felsorolt egyenletrendszer nem oldható meg analitikusan, de a megoldásért hozzá lehet állni iteratív módszerrel a kritérium függvény minimumának meghatározásához. Ez az iteratív algoritmus formalizálható a következő lépésekben:

1. Ki kell választani a c klaszterek számát ($2 \leq c \leq n$) és az m exponenciális nehézségi faktort ($1 < m < \infty$). Ki kell választani az iniciális partíció mátrixot $U^{(0)}$ és az ε terminális kritériumot. Az i iterációs számlálót nullára kell helyezni.
2. Ki kell számolni a klaszterek $v^{(l)}_i$, $j=1, \dots, c$ középpontjait a jelenlegi $U^{(l)}$ partíció mátrix alapján alkalmazva a (6.20) kifejezést.
3. Ki kell számolni az új $U^{(l+1)}$ partíció mátrixot alkalmazva a klaszter középpontokat az előző lépésből a (6.21) kifejezés alapján.
4. Ki kell számítani a $\Delta = \|U^{(l+1)} - U^{(l)}\| = \max_{i,j} |u^{(l+1)}_{ij} - u^{(l)}_{ij}|$ mennyiséget. Ha $\Delta > \varepsilon$, az i számláló megnövekszik és az algoritmus visszaugrik a 2. lépésbe, ellenkező esetben az algoritmus befejeződik.

A megadott algoritmus minimalizálja a beállított függvényt oly módon, hogy az a helyi minimumok egyikében végződjön, attól függően, hogy az iteratív eljárás honnan indul. Ismét hangsúlyozni kell, hogy az m paraméter megválasztása befolyásolja az egész algoritmus fazifikációját. Ha $m \rightarrow \infty$, akkor a fazifikációban, az U partíció mátrixának minden eleme az $1/c$ érték felé tart, ami azt jelenti, hogy minden elemet azonos hitelességgel rendel hozzá az egyes klasszákhöz. A gyakorlatban leggyakrabban ennek a paraméternek az értékei a $[15, 30]$ intervallumból származnak. Egy másik fontos kérdés a klaszterek számának megválasztása. Leggyakrabban ez az érték nem ismert eleve és gyakran emiatt a klaszterezés után kiszámítják az úgynevezett *érvényességi, validációs indexeket (mutatókat)*, amelyek némi információt nyújtanak arról, hogy a klaszterek számára vonatkozó feltételezés helyes volt-e, vagy sem. Ezek a mutatók heurisztikus módon vannak meghatározva és az egyik példa rá az ún. *partíció entrópiája*. Mindezeknek a heurisztikus

szabályoknak az érvényessége kísérletekkel bizonyított annak ellenére, hogy a tulajdonságaik elméleti úton nincsenek megerősítve.

A következő példában be lesz mutatva a klaszterizáció alkalmazása a digitális képfeldolgozásban. A példa lényege a bemeneti képet klaszterekbe, csoportokba sorolni, pontosabban négy klaszterbe. A képfeldolgozásban a klaszterizációs alkalmazásokban az eredeti bemeneti RGB képet gyakran átkonvertálják $L^*a^*b^*$ színrendszerbe, mert az alkalmazások megmutatták, hogy ebben a színrendszerben a klaszterizáció, azaz a színpixelek csoportosítása hatékony.

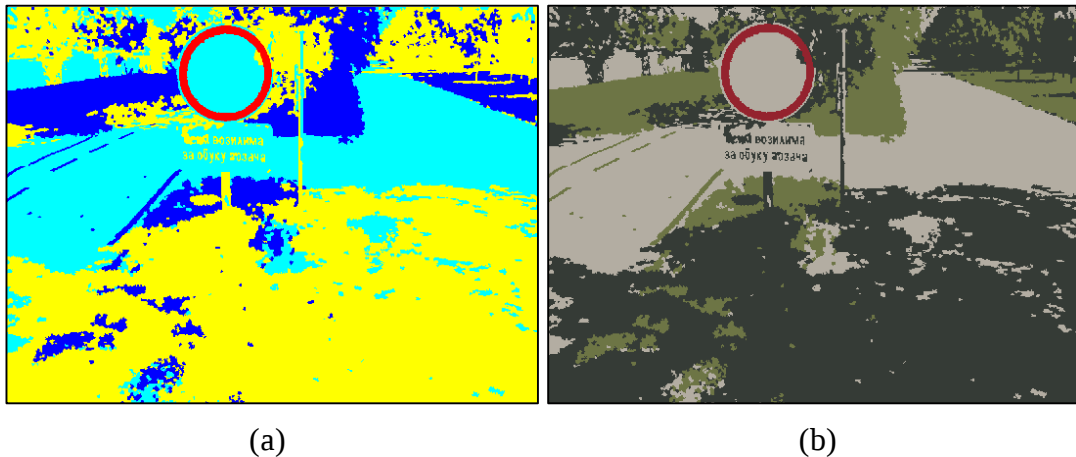
Az $L^*a^*b^*$ színrendszer lényegében egy speciális színmód, ahol a világosság (*Lightness*), valamint a zöld-bíbor különbség és a kék-sárga különbség értékei vannak leírva. Ezt a színrendszert a színes televíziós képátvitelhez alakították ki. A színes televíziózás kezdeti időszakában fontos volt, hogy a kisugárzott színes adást a fekete-fehér készülékek is tudják fogni fekete-fehér árnyalatokban. Ezért az egyik csatorna a "világosságjel" és ez a jel egy önmagában is használható fekete-fehér képet hordoz. A szín-információt a másik két csatorna adja hozzá a képhez, azaz így alakul ki a színes kép. Az "a" betűvel jelölt csatorna a telített zöld és bíbor komplementer színpár közötti színeket tartalmazza. A "b" csatorna pedig a telített sárga és kék színek közötti értékeket hordozza. A számítástechnikában a $L^*a^*b^*$ mód a Kodak Photo CD képformátummal jelent meg és eredetileg a televízión való visszajátszásra fejlesztették ki. Ennek a színkezelésnek számos gyakorlati előnye van (mint pl. a klaszterizációban való alkalmazás), ezért fontos ismerni.

A bemeneti kép $L^*a^*b^*$ színrendszerbe lett alakítva az első lépésben, majd ezután következett a klaszterizációs algoritmus futtatása.



6.24. ábra: A klaszterizáció alkalmazása: (a) bemeneti színes kép, (b) a bemeneti kép $L^*a^*b^*$ színrendszerben

A klaszterizációs algoritmus úgy lett beállítva, hogy 4 klaszterbe legyenek sorolva az $L^*a^*b^*$ színrendszer pixelei a bemeneti képben. Ennek lényege, a bemeneti képet felosztani-szegmentálni a végén 4 különböző árnyalatba a későbbi alakfelismerési, vagy bármilyen más célból. Természetesen a klaszterek száma másként is beállítható, de ebben az esetben a cél a klaszterizáció bemutatása. A 6.25. (a) ábrán látható a klaszterizáció eredménye. Az algoritmus felosztotta a bemeneti képet az $L^*a^*b^*$ színek alapján, majd mindegyik klaszterhez automatikusan egy színt rendelt hozzá (4 klaszter $\rightarrow$ 4 szín, ez esetben a világos kék, kék, sárga és piros színek). Észrevehető, hogy a régiók az algoritmusba bemenő $L^*a^*b^*$ kép régióinak pixelértékei alapján lettek csoportosítva és jól elkülöníthető a körvonal (piros), mert annak színe az eredeti és az $L^*a^*b^*$ képekben is jól elkülöníthető. Továbbá, a sötét árnyalatú részek (sárga szín) is jól el vannak különítve a kép tartalmában. Az aszfaltos és a füves területek is jól láthatók a klaszterizáció eredményében (a két kék szín). Végül, a klaszterizáció eredménye rá lett tükrözve, vetítve az eredeti bemeneti RGB képre és az alapján az eredeti képben a klaszterek alapján megtörtént a szegmentáció. Ennek eredménye látható a 6.25. (b) ábrán. Később, egy így szegmentált kép tovább feldolgozható más meghatározott célból is.



6.25. ábra: A klaszterizáció alkalmazása: (a) a kép klaszterizációja színek szerint, (b) a klaszterizáció eredményének tükrözése a bemeneti képre

Mint látható az előző példában, a fuzzy klaszterizációval bármely kép hatékonyan szegmentálható megfelelő régiókba, ahol a klaszterizáció eredménye a későbbiekben hatékonyan felhasználható különböző képfeldolgozási és alakfelismerési célok és eredmények elérésére.

7. Következtetés

Ebben a tankönyvben a képfeldolgozási, képelemzési és az objektum felismerési módszerek kerültek áttekintésre. A könyv célja bemutatni a leggyakrabban alkalmazott eljárásokat, amelyeket a különféle kutatásokban alkalmaznak. Külön hangsúly van fektetve az objektumok felismerésére, mert ezek az eljárások igen fontosak a képfeldolgozás, robotika, ipar és ikertechnológia szempontjából.

8. Irodalom

- [1] Ulrich Dahmen, Juergen Rossmann, „Experimentable Digital Twins for a Modeling and Simulation-based Engineering Approach“, DOI: 10.1109/SysEng.2018.8544383
- [2] Aidan Fuller, Zhong Fan, Charles Day, Chris Barlow, „Digital Twin: Enabling Technologies, Challenges and Open Research“, DOI: 10.1109/ACCESS.2020.2998358, IEEE Access
- [3] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, Ben Hicks, „Characterising the Digital Twin: A systematic literature review“, CIRP Journal of Manufacturing Science and Technology 29 (2020) 36–52
- [4] K. Eric Harper, Dr. Somayeh Malakuti, Dr. Christopher Ganz, „Digital Twin Architecture and Standards“, IIC Journal of Innovation, 2019.
- [5] Barbara Rita Barricelli, Elena Casiraghi, Daniela Fogli, „A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications“, DOI: 10.1109/ACCESS.2019.2953499
- [6] „Digital twins and simulations“, ABB REVIEW, 2019.
- [7] Andrzej Burghardt, Dariusz Szybicki, Piotr Gierlak, Krzysztof Kurc, Paulina Pietruś, Rafał Cygan, „Programming of Industrial Robots Using Virtual Reality and Digital Twins“, Appl. Sci. 2020, 10, 486; doi:10.3390/app10020486
- [8] Calin Bojea, Annie Guerrieroa, Sylvain Kubickia, Yacine Rezguib, „Towards a semantic Construction Digital Twin: Directions for future research“, Automation in Construction, 114 (2020), 103179
- [9] Aaron Parrott, Dr. Lane Warshaw, „Industry 4.0 and the digital twin“, Deloitte, University Press
- [10] Zongyan Wang, „Digital Twin Technology“, <http://dx.doi.org/10.5772/intechopen.80974>

- [11] Dr Shoumen Palit Austin Datta, „Emergence of Digital Twins“, <https://dspace.mit.edu/handle/1721.1/106496>
- [12] Paul Gardner, „Digital Twins: State-of-The-Art Future Directions for Modelling and Simulation in Engineering Dynamics Applications“, ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg · March 2020
- [13] Fei Tao, He Zhang, Ang Liu, A. Y. C. Nee, „Digital Twin in Industry: State-of-the-Art“, IEEE Transactions on Industrial Informatics, vol. 15, no. 4, april 2019.
- [14] Miodrag Popović, *Digitalna obrada slike*, Akademska misao, 2006.
- [15] Rafael C. Gonzales, Richard E. Woods, *Digital Image Processing*, 3rd Edition, New Jersey: Pearson Prentice Hall, 2008.
- [16] Rafael C. Gonzales, Richard E. Woods, *Digital Image Processing*, 4th Edition, Global Edition, New Jersey: Pearson Prentice Hall, 2018.
- [17] Rafael C. Gonzales, Richard E. Woods, Steven L. Eddins, *Digital Image Processing Using Matlab*, 2nd Edition, Gatesmark, LLC. 2009.
- [18] Odry Péter, *Hang- és képfeldolgozás*, Szabadkai Műszaki Szakfőiskola, Szabadka, 2008.
- [19] Javier R. Movellan “Tutorial on Gabor Filters”, 2006, Kolmogorov Project, Machine Perception Laboratory, San Diego
- [20] Vladimir Tadic, Miodrag Popovic, Peter Odry, “Fuzzified Gabor filter for license plate detection”, *Engineering Applications of Artificial Intelligence*, 48 (2016) 40-58, Elsevier
- [21] L. A. Zadeh, "Fuzzy Sets", *Information and Control*, vol. 8, no. 3, pp. 338-353, 1965.
- [22] Juh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani, “Neuro-Fuzzy and Soft Computing, Prentice Hall, New Jersey, 1997.
- [23] Nebojša Perić, *Neke primene teorije fazi skupova i fazi logike u procesiranju slika*, Master rad, Univerzitet u Beogradu, Matematički fakultet, 2014.

- [24] Vladimir Tadić, “Fazifikacija Gaborovog filtra i njena primena u detekciji registarskih tablica”, University of Novi Sad, FTN Novi Sad, Serbia, Doktorska disertacija - Doctoral Thesis, 2018.
- [25] Vladimir Tadić, “Segmentacija slike”, University of Novi Sad, FTN Novi Sad, Serbia, Diplomski rad-Graduate Thesis, 2004.
- [26] Vladimir Tadić, “Detekcija i očitavanje registarskih tablica s posebnim osvrtom na slike snimljene klasičnim digitalnim fotoaparatom”, University of Novi Sad, FTN Novi Sad, Serbia, Magistarska teza-Postgraduate Thesis, 2008.
- [27] Vladimir Tadić, Željen Trpovski, Peter Odry, “Filtriranje u predprocesiranju slike u detekciji i prepoznavanju registarskih tablica”, DOGS 2008 Proceedings, okt 2-3 2008, Subotica, Serbia
- [28] Vladimir Tadić, Željen Trpovski, Peter Odry, “License Plate Detection using Gabor Filter Banks and Texture Analysis”, Proc. 9th IEEE International Symposium on Intelligent Systems and Informatics, Subotica, pp. 381-386, sep 2011., Serbia
- [29] Vladimir Tadić, Zoltan Kiraly, Željen Trpovski, Tatjana Lončar Turukalo, Peter Odry, “Evolucija Gaborovog filtra u aplikaciji detekcije registarskih tablica”, ETRAN 2018, Palić
- [30] “Evolution of Gabor Filter in Application of License Plate Detection”, Proceedings of Papers – 5th International Conference on Electrical, Electronic and Computing Engineering, IcETRAN 2018, Palić, Serbia
- [31] Vladimir Tadić, Zoltan Kiraly, Željen Trpovski, Tatjana Lončar Turukalo, Peter Odry, “Fuzzified Gabor Filter for Text Recognition in Images”, Proc. 16th IEEE International Symposium on Intelligent Systems and Informatics, Subotica, sep 2018., Serbia
- [32] Vladimir Tadic, Zoltan Kiraly, Peter Odry, Zeljen Trpovski, Tatjana Loncar-Turukalo, “Comparison of Gabor Filter Bank and Fuzzified Gabor Filter for License Plate Detection”, Acta Polytechnica Hungarica, Budapest, August 17, 2019, DOI: 10.12700/APH.17.1.2020.1.4

- [33] Vladimir Tadic, Akos Odry, Istvan Kecskes, Ervin Burkus, Zoltan Kiraly, Peter Odry, "Application of Intel RealSense Cameras for Depth Image Generation in Robotics", Corfu - AICT '19, 10th International Conference on Applied Informatics and Computing Theory, 2019, Greece
- [34] Vladimir Tadic, Akos Odry, Istvan Kecskes, Ervin Burkus, Zoltan Kiraly, Peter Odry, "Application of Intel RealSense Cameras for Depth Image Generation in Robotics", WSEAS Transactions on Computers, E-ISSN: 2224-2872, Volume 18, 2019
- [35] Vladimir Tadic, Zoltan Kiraly, Peter Odry, Zeljen Trpovski, Tatjana Loncar Turukalo, "A Robust Text Detection Method in Low-Quality Images Using Fuzzy Gabor Filter", Book of Abstracts - Fifth International Conference TAKTONS – Novi Sad, Serbia, 6-9.11.2019., ISBN 978-86-6022-223-9. COBISS.SR-ID 331459079
- [36] Vladimir Tadić, Ervin Burkus, Akos Odry, Istvan Kecskes, Zoltan Kiraly, Peter Odry, "Effects of the Post-processing on Depth Value Accuracy of the Images Captured by RealSense Cameras", Contemporary Engineering Sciences, Vol. 13, 2020, no. 1, 149 – 156, HIKARI Ltd, doi.org/10.12988/ces.2020.91454
- [37] Vladimir Tadic, Akos Odry, Attila Toth, Zoltan Vizvari, Peter Odry, "Fuzzified Circular Gabor Filter for Circular and Near-Circular Object Detection", IEEE Access, DOI: 10.1109/ACCESS.2020.2995553
- [38] Vladimir Tadic, Akos Odry, Ervin Burkus, Istvan Kecskes, Zoltan Kiraly, Peter Odry, "Edge-Preserving Filtering and Fuzzy Image Enhancement in Depth Images captured by RealSense Cameras in Robotic Applications", Advances in Electrical and Computer Engineering – AECE, no. 3, pp. 83 – 92, 2020, doi: 10.4316/AECE.2020.03010
- [39] E. Burkus, Á. Odry, I. Kecskés, V. Tadić and P. Odry, "A Novel Leg Design for the Szabad(ka) III Robot," 2020 IEEE 18th International

- Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 2020, pp. 000095-000100, doi: 10.1109/SISY50555.2020.9217077.
- [40] Tadić Vladimir, “2D Gábor-szűrő fuzzifikálása és alkalmazása a járművek rendszámtábláinak felismerésében”, Az Informatika Korszerű Technikái Konferencia 2020: Jövőformáló tudomány programfüzet és absztraktkötet Dunaújváros, 2020. november 9-10, Konferencia helye, ideje:: Dunaújváros, Magyarország 2020.11.09. - 2020.11.10., Dunaújváros: DUE Press, pp 29-30 (2020), Nyelv: Magyar | Befoglaló link(ek): ISBN: 9786156142030
- [41] Bratislav Milovanović, Jugoslav Joković, Nikola Dojčinović, Nikola Božinović, “Primena algoritama za detekciju referentnih elemenata objekata u video sekvenci”, XXIX Simpozijum o novim tehnologijama u poštanskom i telekomunikacionom saobraćaju – PosTel 2011, Beograd, 06. i 07. decembar 2011.
- [42] Bo Xiang, “Knowledge-based image segmentation using sparse shape priors and high-order MRFs”, Signal and Image processing, Ecole Centrale Paris, 2013. English.
- [43] Dorin Comaniciu and Peter Meer, “Mean shift: A robust approach toward feature space analysis”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pages 603–619, 2002. 7
- [44] David Mumford and Jayant Shah, “Optimal approximations by piecewise smooth functions and associated variational problems”, Communications on pure and applied mathematics, vol. 42, no. 5, pages 577–685, 1989. 7, 12, 19
- [45] Tony F Chan and Luminita A Vese, “Active contours without edges”, IEEE Transactions on Image Processing, vol. 10, no. 2, pages 266–277, 2001. 7, 12, 19

- [46] Jianbo Shi and Jitendra Malik, “Normalized cuts and image segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pages 888–905, 2000. 7, 18
- [47] Yuri Boykov and Gareth Funka-Lea, “Graph cuts and efficient ND image segmentation”, *International Journal of Computer Vision*, vol. 70, no. 2, pages 109–131, 2006. 7, 8, 21, 28, 29, 106
- [48] Timothy F Cootes, Christopher J Taylor, David H Cooper, Jim Graham et al., “Active shape models - their training and application”, *Computer Vision and Image Understanding*, vol. 61, no. 1, pages 38–59, 1995. 7, 13, 38, 69, 97, 125
- [49] Timothy F. Cootes, Gareth J. Edwards and Christopher J. Taylor, “Active appearance models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pages 681–685, 2001. 7, 15, 38
- [50] Lawrence H Staib and James S Duncan, “Model-based deformable surface finding for medical images”, *IEEE Transactions on Medical Imaging*, vol. 15, no. 5, pages 720–731, 1996. 8, 38, 39
- [51] Torsten Rohlfing, Robert Brandt, Randolph Menzel, Daniel B Russakoff and Calvin R Maurer Jr., “Quo vadis, atlas-based segmentation”, In *Handbook of Biomedical Image Analysis*, pages 435–486. Springer, 2005. 8
- [52] Željko Đurović, *Lecture Notes on Object Recognition*, ETF, Belgrade, Serbia
- [53] A. Garcia-Garcia, S. Orts-Escolano, S.O. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, “A Review on Deep Learning Techniques Applied to Semantic Segmentation”, arXiv:1704.06857v1
- [54] Hyeonwoo Noh, Seunghoon Hong, Bohyung Han, “Learning Deconvolution Network for Semantic Segmentation”, arXiv:1505.04366
- [55] Dan Cireşan, Ueli Meier, Jonathan Masci, Jürgen Schmidhuber, “Multi-column deep neural network for traffic sign classification”, *Neural Networks* 32 (2012) 333–338

- [56] Marko Zadavec, “Text Recognition with Convolutional Neural Networks”, Master Thesis, Međimurje University, Croatia, 2017
- [57] Kevin W. Bowyer, Karen Hollingsworth, Patrick J. Flynn, “Image understanding for iris biometrics: A survey”, *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 281–307, 2008.
- [58] João M. C. Sousa, Member, IEEE, João M. Gil, João R. Caldas Pinto, “Word Indexing of Ancient Documents Using Fuzzy Classification”, *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 5, pp. 852-862, 2007.
- [59] Jinhai Cai, Zhi-Qiang Liu, “Off-Line Unconstrained Handwritten Word Recognition”, *Computer Vision and Machine Intelligence*, vol. 14, no.3, pp. 259-280, 2000.
- [60] Feng Yue, Wangmeng Zuo, David Zhang, Kuanquan Wang, “Orientation selection using modified FCM for competitive code-based palmprint recognition”, *Pattern Recognition* vol. 42, no.11, 2841 – 2849, 2009.
- [61] Annamaria R. Varkonyi-Koczy, “Fuzzy logic supported corner detection”, *Journal of Intelligent & Fuzzy Systems* vol. 19, no. 1, pp. 41–50, 2008.
- [62] Xinbo Gao, Rong Fu, Xuelong Li, Dacheng Tao, Beichen Zhang, Huigen Yang, “Aurora image segmentation by combining patch and texture thresholding”, *Computer Vision and Image Understanding*, vol. 115, no. 3, pp. 390–402, 2011.
- [63] R.S.Choras, “Fuzzy Approach for Image Retrieval”, *Institute of Telecommunications, Pattern Recognition and Image Analysis*, vol. 17, no. 2, pp. 259-267, 2007.
- [64] Mike Nachtgael, Dietrich Van der Weken, Etienne E. Kerre, Wilfried Philips (Editors), *Soft Computing in Image Processing*, Berlin, Heidelberg, Springer-Verlag, 2007.
- [65] Zheru Chi, Hong Yan, Tuan Pham, *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*, Singapore, World Scientific, 1996.

- [66] James C. Bezdek, James Keller, Raghu Krisnapuram, Nikhil Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, New York, Springer Science, 2005.
- [67] Jainguo Zhang, Tieniu Tan, Li Ma, “Invariant Texture Segmentation Via Circular Gabor Filters”, *Object recognition supported by user interaction for service robots*, 1051-4651/02, IEEE, 2002, DOI: 10.1109/ICPR.2002.1048450
- [68] Zhenfeng Zhu, Ming Tang, Hanqing Lu, “A New Robust Circular Gabor based Object Matching by Using Weighted Hausdorff Distance”, *Pattern Recognition Letters* 25 (2004) 515–523, Elsevier
- [69] Qingbo Yin, Jong Nam Kim, “Rotation Invariant Texture Classification using Circular Gabor Filter Banks”, Y. Shi et al. (Eds.): *ICCS 2007, Part III*, LNCS 4489, pp. 149–152, 2007., Springer-Verlag Berlin Heidelberg 2007
- [70] Sai-Cong Xu, Nong-Liang Sun, Mao-Yong Cao, „Wall-Pasted Cell Segmentation Based on Circular Gabor Filter Bank“, *Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition*, Beijing, China, 2-4 Nov. 2007, 1-4244-1066-5/07/\$25.00 ©2007 IEEE
- [71] Tang L, Xie WX, Huang JJ, “Automatic Multilevel Image Segmentation Based on Fuzzy Reasoning” *Image Analysis & Stereology*, 23:23-31, DOI: 10.5566/ias.v23.p23-31, 2004.