Képosztályozás emberi és gépi tanulás esetén

Papp Dávid
Távközlési és Médiainformatikai Tanszék,
Budapesti Műszaki és Gazdaságtudományi Egyetem,
Magyar Tudósok krt. 2., H-1117, Budapest, Magyarország
pd948@hszk.bme.hu / pappdavid27@gmail.com

A gépi tanulás és az emberi tanulás összehasonlítása izgalmas téma, melynél párhuzamba állítható a két folyamat. A gépi tanuláshoz nagy mennyiségű címkézett tartalomra van szükség, melynek előállítása költséges, míg az emberi tanulásnál elég néhány, vagy akár egyetlen mintakép egy séma elsajátításához. A két folyamat közti alapvető és egyben legnagyobb különbséget a priori információk eltérő mértéke képezi. Korunk egyik alapvető célja, hogy a gép által elérhető eredményeket maximalizáljuk, így rengeteg emberi erőforrás takarítható meg. A tanulási folyamatok eredményességét képi tartalmak osztályozásával mértem, melyet iteratívan végeztem az emberi és gépi esetek összehangolásához.

Kulcsszavak—emberi tanulás; gépi tanulás; képosztályozás

I. BEVEZETÉS

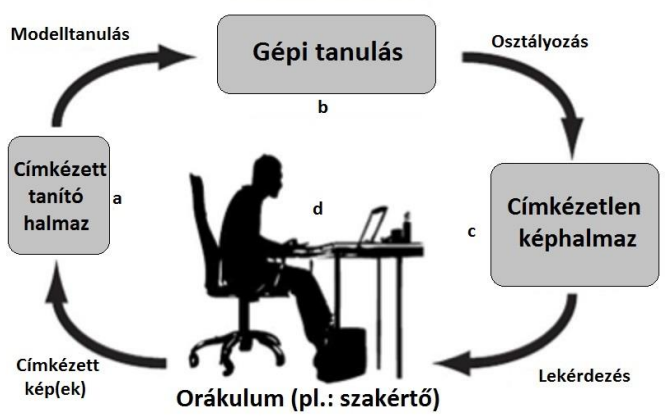
Manapság, a digitális világ növekedésének köszönhetően rengeteg fénykép található különböző adathordozókon, szerte a világban. Az interneten fellelhető képi tartalmak száma pedig megbecsülhetetlen, így korunk egyik alapvető problémája ennek a hatalmas adatmennyiségnek a rendszerezése. Ez rendkívül fontos számtalan alkalmazás számára, amelyek képekkel foglalkoznak, gondoljunk csak egy képkereső programra, vagy képnézegetőre. Ezen alkalmazások többféle módszert használnak a felhasználók segítése érdekében, hogy minél egyszerűbben találjanak számukra minél relevánsabb fényképeket, valamint több típusú részfeladatot kell megoldaniuk, mint például a képek osztályozása, rangsorolása, csoportosítása.

Vizuális információk alapján, emberi szemmel szinte tévesztés nélkül lehet képeket kategorizálni, viszont ez nagyon időigényes és (emberi) erőforrás igényes feladat. Ezzel szemben a számítógép kontextusában egy kép pixeleket, színeket, alakokat, textúrákat tartalmaz, melyekből meghatározni azt, ami egy ember számára egyértelmű egy képre ránézve, közel sem triviális feladat. Nem ritka, hogy több száz kategória közül kell eldöntenie egy képosztályozó algoritmusnak, hogy melyik az az egy vagy néhány, amelyikbe az adott kép beleillik. Alapvető probléma az is, hogy hasonló szemantikus információval bíró képek is lehetnek nagyon különbözőek.

Tehát a szemantikus képosztályozásnak két fontos és nehéz folyamata van, az első a képek olyan módú reprezentálása, amely lehetővé teszi azok összehasonlítását, a második pedig a képek osztályozása ezen kinyert információk alapján. A képosztályozási eljárások által elért eredmények általában gyengék, még bonyolult, nagy futási időt igénylő algoritmusok használata esetén is, mivel a fentebb említett okokból adódóan nagyon összetettek és rengeteg a hibalehetőség. Körülhatárolt területeken viszont jó eredménnyel alkalmazhatóak, mint például az orvosi diagnosztika, vagy a karakterfelismerés.

Az emberi és gépi tanulás összehasonlításához egy konkrét mérési tervet dolgoztam ki, valamint több különböző képhalmazt állítottam össze. Az emberi tanulás eredményének méréséhez egy böngészőben futó alkalmazást használtam, melyben a felhasználó személyek osztályozni tudták a képeket. Egy képhalmaz osztályozásának kezdetén, minden egyes kategóriából megadtam egy mintaképet, majd véletlenszerűen következtek a képhalmaz további képei, melyeket a megfelelő osztályban kellett elhelyeznie a felhasználónak. Minden egyes kép osztályozása után megadtam annak valós címkéjét, így kiderült, hogy helyes volt-e a döntés. A gépi tanulás mérését hasonlóan oldottam meg, így valóban összehasonlíthatók az eredmények. A folyamat az 1. ábrán tekinthető meg. Minden iterációban egy új, ismeretlen képet osztályozok, majd kiegészítem azzal a tanító halmazt, és újratanítom a rendszert a következő kép osztályozása előtt. Ezzel azt érem el, hogy a tanítórendszer is folyamatosan bővíti tudását, akár csak a felhasználó, miközben osztályozza a képeket. A lépéseket addig folytatom, míg minden képet fel nem címkéz az orákulum. Minél „később” osztályozok egy képet, annál több tanító képből tanul a rendszer. Ezért fontos kérdés, hogy a lépéseknél mikor melyik képet válasszam ki tesztelésre, majd címkézésre, és itt a hangsúly a címkézésen van. Megkülönböztetünk *passzív* és *aktív tanulási* módszereket, melyek a kiválasztás sorrendjének felállítási módjában különböznek.

Passzív tanulás esetén egy véletlenszerű, statikus sorrendet határozunk meg, míg aktív tanulás során a tanulórendszernek lehetősége van a tanuló adathalmaz egyes mintáit dinamikusan kiválasztani [1]. Az emberi és gépi tanulást passzív esetben hasonlítottam össze, hiszen az emberi aktív tanulás hasonlóan precíz mérése nehezen oldható meg. A képhalmazok mellé meghatároztam azt a statikus sorrendet, mely szerint a címkézetlen képek tesztelése történik, mind emberi, mind gépi esetben, így ezen a ponton is megegyezik a két folyamat. A következő fejezetben a képek matematikai reprezentálásának módját ismertetem, majd az osztályozás folyamatát tárgyalom. Végül az emberi és gépi tanulás összehasonlításának kísérletét és annak eredményeit mutatom be.



1. ábra: A gépi tanulás mérése: címkézett tanító halmaz (a), címkézetlen képhalmaz (c, teszthalmaz), gépi tanuló algoritmus (b), *orákulum* (d)

II. KÉPEK REPREZENTÁLÁSA

A. *Vizuális kódszavak*

Az eljárásom alapját egy általános, hasonló feladatokban gyakran használt módszer képezi, ami nem más, mint a szózsák modell (*Bag of Words*) képeken alkalmazott változata [2, 3]. Az elképzelés lényege, hogy egy képet a rajta szereplő vizuális elemek, más néven *vizuális kódszavak* összességével reprezentálunk, és ezek térbeli elhelyezkedésétől eltekintünk. Tehát egy képet csupán a vizuális kódszavak eloszlásával jellemzünk, és ebből próbálunk meg következtetni a szemantikus tartalmára. A vizuális kódszavak meghatározásához úgynevezett *alacsony szintű leíró*kra (röviden: leíró) van szükségünk, melyek a kép egy adott pontjának környezetében előforduló lokális tulajdonságokat foglalják magukban. Ezek lehetnek színek, textúrák, alakok és még sok más. Az algoritmus első feladata tehát, hogy meghatározza azokat a pontokat, melyek érdemesek leírók számítására, más néven a *kulcspontok*at.

A kulcspontok meghatározásához a Harris-Laplace detektort használtam, melynek alapja a Harris detektor, vagy más néven kombinált sarok és él detektálás [4, 5]. Az eljárás lényege, hogy egy képen azonosítja az éleket, ahol a kép pontjainak intenzitásértékeiben nagy változás következik be. Két él találkozásánál (sarokpontnál) pedig az intenzitásváltozás, azaz a derivált, két irányban is nagy abszolút értékű lesz. Ennek meghatározására egy csúszó ablakot vizsgál. A Harris detektort K. Mikolajczyk és C. Schmid fejlesztette tovább úgy, hogy az invariáns legyen a skálázásra is, ezt nevezik Harris-Laplace detektornak [6].

A második főbb lépés, hogy minden kulcspontot SIFT (Scale Invariant Feature Transform) leíróval jellemzek, melyet David G. Lowe fejlesztett ki [7]. Ez tulajdonképpen egy 128 dimenziós vektor, amely egy pont lokális környezetében lévő gradiensek orientációjából számolható. Minden kép minden kulcspontjában kiszámítom ezeket a leírókat.

Az egymáshoz hasonló leírók jelentenek egy vizuális kódszót. Ezek meghatározásához klaszterezem az összes képből kinyert SIFT leírót a GMM (Gaussian Mixture Model) módszer segítségével [8]. A GMM egy generatív módszer az alacsony leírók klaszterezésére, tehát az egyes klaszterekhez egy-egy valószínűségi modellt rendel. A pontok valószínűségi sűrűségfüggvényét K darab Gauss-féle függvény súlyozott összegével írja le:

$$p(X \mid \lambda) = \sum_{j=1}^{K} \omega_j g(X \mid \mu_j, \sigma_j) \qquad (1)$$

Itt az $X=\{x_1,x_2,...,x_N\}$ jelöli az M dimenziós adatvektorokat (SIFT leírók), λ a keresendő paramétert, $g(X|\mu_j\sigma_j)$ a Gauss-féle függvényeket (ahol μ_j a várható értéket, σ_j a szórást jelöli), valamint ω_j pedig a súlyozó együtthatókat. A GMM λ paraméterét ML (Maximum Likelihood) becsléssel határozzuk meg. Az x_i adatvektorok közt függetlenséget feltételezve a következőt írhatjuk fel:

$$p(X \mid \lambda) = \prod_{i=1}^{N} p(x_i \mid \lambda) \qquad (2)$$

Ez a kifejezés nem-lineáris a λ paraméterre nézve, tehát a direkt maximalizálása nem lehetséges. Erre egy speciális iteratív módszert szokás használni, az EM (Expectation Maximization) algoritmust [9, 10].

A GMM eredményeként megkapom a vizuális kódszavakat (az egyes Gauss függvények), melyekre tekinthetünk úgy, mint a képhalmaz tömör reprezentálására. A célom az, hogy minden képet jellemezni tudjak a tartalma alapján, méghozzá olyan módon, hogy azok összehasonlítása, megkülönböztetése, osztályozása lehetővé váljon. Így mindenképpen szükségem van egy olyan magasabb szintű képi leíró megalkotására, amely nem csak egy kulcspontot jellemez, hanem egy teljes képet. Erre a feladatra a Fisher-vektor használom.

B. *Fisher-vektor*

A Fisher-vektor a képfeldolgozási témakörök közül a képosztályozásban a legelterjedtebb így az én algoritmusomba tökéletesen beleillik [11, 12]. Ez egy rendkívül komplex leíró, amely arra használható, hogy egy teljes képet jellemezzünk vele egyetlen vektor formájában. Kiszámításához szükség van a SIFT leírókra, illetve a vizuális kódszavakra, tehát a GMM-re. Valójában azt fogja ez megmondani, hogy egy kép milyen eloszlásban tartalmaz vizuális elemeket, ahol ezek a vizuális elemek a képhalmaz egészéből kerülnek ki.

A GMM tárgyalásánál bevezetett jelöléseknek megfelelően legyen $p(X|\lambda)$ a valószínűségi sűrűségfüggvény ($\lambda=\{\omega_j,\mu_j,\sigma_j|j=1...K\}$), legyenek $X=\{x_1,x_2,...,x_N\}$ az M dimenziós adatvektorok, melyek itt nem az összes SIFT leírót jelentik, csak azokat, melyek egy adott képre vonatkoznak. A sűrűségfüggvény gradiense, azaz a logaritmusának deriváltja megadja, hogyan írja le legjobban a modell az adatvektorokat, tehát a képet, így tulajdonképpen ez a mennyiség a Fisher-vektor:

$$\nabla_\lambda \log p(X \mid \lambda) \qquad (3)$$

Ebből következik, hogy a Fisher-vektor összesen $K(2M+1)-1$ dimenziós (-1, mivel egy súlyozó együttható kiszámítható a többi ismeretében). Az én implementációmban $K=256$, mivel ennyi Gauss-függvényt definiálok $M=128$, mivel a SIFT leírók 128 dimenziósak. Ez azt jelenti, hogy az én esetemben egy Fisher-vektor 65791 dimenziós. A Fisher-vektorok tehát egységes és egyértelmű reprezentációi a képeknek és ezeket a vektorokat fogom felhasználni az osztályozáshoz.

III. OSZTÁLYOZÁSI FOLYAMAT

Ebben a fejezetben a képosztályozás folyamatát fogom bemutatni, mely során passzívan tanul az algoritmus. Jelölje T a teljes képhalmazt, amelyet osztályozni szeretnék, és jelölje S azt a kiindulási képhalmazt, mely címkéi a tanulórendszer számára már ismertek. A célunk, hogy T minden elemére adjunk egy jóslatot a kategóriájára vonatkozóan. Ehhez dinamikusan fogom kezelni mind a T, mind pedig az S halmazt. Kezdetben a tanulóhalmazunk S, és a teszthalmazunk T. Lépésenként T-ből kiválasztunk és osztályozunk, majd ki is törlünk egy képet és ezzel a képpel bővítjük az S halmazt, így a rendszer tudása egyre növekszik, és várhatóan pontosabb becslést tud majd adni a következő kép(ek)re. Fontos megjegyezni, hogy ilyenkor a képpel együtt annak valódi osztálycímkéjét is átadom a tanulórendszernek. Passzív tanulás esetén, az S bővítése nem szabályozott, hiszen egy véletlenszerűen generált, statikus sorrend alapján kerülnek át az egykori tesztképek a tanulóhalmazba. Ahogy már korábban említettem, ez a sorrend egységes az emberi és gépi tanulás esetén. Nézzük, hogyan történik egy kép osztályozása.

A képek reprezentálásával megkaptam a képek magas szintű leíróit (Fisher-vektorok) $\varphi_1,\varphi_2,\ldots,\varphi_N$, valamint azok valós osztálycímkéi, melyeket az angol elnevezésnek megfelelően *ground truth*-nak nevezünk. Egy teszt képet a magas szintű leírója alapján osztályozok úgy, hogy minden c kategóriához meghatározok egy $f(\varphi)$ értéket, amely azt fogja megadni, hogy az adott kategória mekkora bizonyossággal jellemző a képre. Ezt úgy teszem meg, hogy létrehozok c darab egymástól független bináris osztályozót, melyekkel külön-külön elvégzem a tanítást és az osztályozást. Minden tanításnál a kiválasztok egy osztályt, és abba tartozó képek lesznek a pozitív minták, az összes többi pedig negatív minta. Ezt *one-against-all* módszernek nevezzük, és egyszerűsége ellenére rendkívül jól használható. Bináris osztályozóként az SVM (*Support Vector Machine*) egyik változatát használom amelyet C-SVC osztályozónak neveznek [13, 14]. A módszer lényege, hogy a végtelen sok szeparáló hipersík közül a maximális margójút találja meg.

IV. TESZTELÉS ÉS EREDMÉNYEK

A. Felhasznált képhalmazok

A teszteléshez öt különböző képhalmazt használtam, melyek mindegyike manuálisan kigyűjtött és címkézett képekből áll. Mivel célom az emberi és gépi tanulás összehasonlítása, ezért viszonylag kisméretűek a teszthalmazok, hogy az emberi tanulás kiértékelésében részt vevő személyek számára ne legyen túl megterhelő a rengeteg kép osztályozása. Az 1. táblázatban összefoglaltam ezek méretét, a definiált kategóriák számát, illetve a tesztelő személyek számát.

1. táblázat: Teszteléshez használt képhalmazok adatai

	Méret	**Kategóriák száma**	**Tesztelő személyek száma**
Képhalmaz 1	100	7	12
Képhalmaz 2	100	8	9
Képhalmaz 3	116	6	12
Képhalmaz 4	105	5	6
Képhalmaz 5	106	6	7

A kiindulási címkézett halmazhoz minden osztályból választottam egy-egy képet, ez alapján kezdték mind a tesztelő személyek és az algoritmus a tanulást. Az osztályozáshoz megadtam a megfelelő méretű szekvenciákat, melyek a címkézetlen képek érkezési sorrendjét határozzák meg, ez szintén egységes volt a webes felület és a képosztályozó rendszer esetén.

B. Kiértékelés menete

Az osztályozások eredményeit a pontosság (*accuracy*) szempontjából értékeltem ki, melyhez szükség volt a tesztelt képek során született döntések eredményeire, hogy sikeres volt-e a kategóriába sorolás, vagy sem. A tesztképekhez tartozó szekvencia mellé egy 0/1 döntési eredményt rendeltem, így minden képről eltároltam, hogy annak osztályozását az adott tesztelő entitás (mely lehet gépi vagy emberi) miként végezte. Ez által, a pontosság meghatározása a következő képlet alapján történik:

$$accuracy = \frac{döntési\ vektor\ 1-eseinek\ száma}{szekvencia\ mérete} \quad (4)$$

A folyamatosan bővülő tanulóhalmaz egyre több információt ad a tesztelőnek, így elméletben, a fennmaradó címkézetlen képeket nagyobb pontossággal képes osztályozni, mint a korábbiakat. Ennek mérése érdekében egy további mutatót is kiszámítok, melyhez egy csúszó ablakot mozgatok a szekvencia elejétől a végéig, és mindig az *ablak alatti* szekvencia által kijelölt képek osztályozási *pontosság*át határozom meg:

$$accuracy_w = \frac{ablak\ alatti\ 1-esek\ száma}{ablak\ mérete} \quad (5)$$

Ahogy az 1. táblázatban látszik, minden képhalmazt több személy tesztelt, így a kapott pontossági értékeket átlagoltam. A következő alfejezetben a kapott eredményeket mutatom be, és hasonlítom össze. A kapott átlagos ablak alatti pontosságokat vonal diagramok segítségével ábrázolom, a tanított képek számának függvényében (tehát a szekvencián előre haladva). Az ablak méretét 10-re választottam, így a kiértékelés elején, a 10. beérkezett kép után kapom meg az első pontossági értéket, és az ábrázolás onnantól kezdődik az utolsó beérkező képig

C. Eredmények I

Az első képhalmaz különböző film poszterekből állt, a definiált kategóriák pedig az egyes filmek műfajai voltak (pl.: animációs, akció, vígjáték, horror, stb.). Ez azt jelenti, hogy az egyes osztályokba tartozó képek nem kizárólag vizuális hasonlóság alapján kapcsolódnak (lásd 2. ábra). Éppen ezért a képosztályozó algoritmusnak nehéz dolga volt ezzel a teszthalmazzal, hiszen az egyedül a tartalmi egyezéseket veszi figyelembe, míg az emberi agy könnyedén megtalálja a mögöttes jelentésből fakadó kapcsolatot is.

Ahogy a 2. ábrán látszik, az egyes kategóriánként megjelenített képek közt nem lehet egyértelmű vizuális kapcsolatot meghatározni. A gépi tanulás eredménye a 4. ábra (a) diagramján látható, az emberi tanulással kapott eredmények pedig a (b) diagramon tekinthetők meg.

2. ábra: Példa képek az első képhalmazból, (a)-(g)-ig oszloponként azonos osztályú képekkel

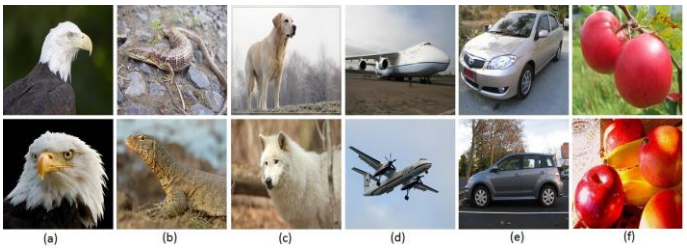
Jól látszik, hogy a gépi tanulás, sőt, az emberi tanulás esetén is a tanított képek számának növekedésével a pontosság is növekszik (kevés kivétellel). Az algoritmus a szekvencia elején érkező képeket nagy hibával osztályozta, viszont 70 kép után javul valamennyit a pontosság, viszont utána vissza is esik. Ezért azt állapítottam meg, hogy ezeket a kategóriákat nem sikerült eredményesen megtanulnia a gépnek. Az emberi tanulás láthatóan az első képtől kezdve jobban teljesít, viszont ez várható is volt, hiszen a gépnek nincs priori ismerete, emberek számára viszont sok segítséget nyújthatnak tapasztalati ismeretek, például ebben az esetben, ha a tesztelő látta már azt az adott filmet, melyet a poszter ábrázol. Ezért előfordult a 0,9-es ablak alatti pontosság is, míg gépi tanulás esetén ez a legjobb esetben 0,7 volt.

D. Eredmények II

A harmadik képhalmaz a leghasonlóbb a képosztályozási feladatoknál megszokott teszthalmazhoz. Az itt definiált osztályokban (sas, kutyafélék, hüllők, gyümölcsök, szárazföldi négykerekű járművek, repülőgépek) jelentős vizuális hasonlóság van jelen (lásd 3. ábra), így gépi tanulással hatékonyan kategorizálhatóak a képek. A 3. ábrán minden osztályból 2-2 képet jelenítettem, hogy szemléltessem a különbséget az első és harmadik teszthalmaz között. Jól látható, hogy a 2. ábra kategóriáival ellentétben most tényleg tartalmi hasonlóság adja a kategorizálás alapját, és tulajdonképpen az osztályozó rendszert ilyen képhalmaz tesztelésére készítettem fel, ami az eredményeken is látszik.

A kategóriák meghatározásnál annyi nehezítést alkalmaztam, hogy néhány esetben több különböző objektum is beletartozik ugyanabba az osztályba, tehát úgynevezett *superclass*-okat definiáltam. Például a gyümölcsök kategóriát a banán, narancs, alma, körte, barack, málna és áfonya alkotja, illetve a kutyafélék osztályt a farkas, róka és kutya objektumok alkotják, valamint ugyanez igaz a szárazföldi négykerekű járművek esetén is. Látni fogjuk, hogy ilyen esetben a felhasználók szinte tévesztés nélkül képesek megmondani az egyes képek valódi osztályát. Ez nem meglepő, hiszen például egy rókát egy narancstól előzetes tanítási folyamat nélkül is 100%-os biztonsággal és pontossággal vagyunk képesek megkülönböztetni. A 4. ábrán láthatóak a gépi (c), valamint az emberi (d) tanulás eredményei.

A gépi tanulás eredményeiből is jól látható, hogy ez egy olyan teszthalmaz, melyre az algoritmus fel van készítve, így a korábban jósolt javulás a pontosságban egyértelműen észrevehető, a tanított képek számának növekedésével.



3. ábra: Példa képek a harmadik képhalmazból, (a)-(f)-ig oszloponként azonos osztályú képekkel

Az első néhány képet hibásan osztályozza a rendszer, viszont nagyjából 30 tanító kép után stabilan 0,7-0,9 közé áll be az ablak alatti átlagos pontosság. A 4. ábra (c) diagramján látható hogy szinte végig 100%-os a pontosság, az elején volt egy, illetve később néhány tévesztés.

E. Eredmények összefoglalása

A tesztek jól jellemzik az emberi és gépi tanulás közti különbségeket. A legfontosabb különbség, hogy az emberek által birtokolt előzetes információ egy jó alapot nyújt az osztályok közti hasonlóság gyors, akár azonnali felismeréséhez. Igazán jól összemérni a kettőt akkor lehetne, ha olyan képeket tudnánk előállítani, melyek a tesztelő személyek semmilyen eddig tapasztalatához, ismereteihez nem köthetőek, így számukra is elengedhetetlen lenne az előzetes tanulás, ahogy a gépi tanulás esetén láttuk. Erre használhatnánk például absztrakt, gép által véletlen generált képeket.

A csúszó ablak segítségével jól mérhetővé vált az, hogy a pontosság hogyan változik az osztályozás során (főleg gépi esetben), viszont a teljes teszthalmazon számított pontossági érték (lásd 4. egyenlet) meghatározása nem képes a javulás megmutatására. Ennek oka, hogy a teljes döntési vektort használjuk a kiszámításhoz, figyelmen kívül hagyva azt, hogy a döntési vektorban (elemszámában) előre haladva az osztályozáshoz felhasznált tanulóanyag mérete növekszik. Ennek ellenére ezeket is kiértékeltem, és a 2. táblázatban összefoglaltam, ahol W, G és E rendre az ablak alatti, gépi és emberi pontosságtípusokat jelölik. Például $\max(\text{accuracy}_{W,G})$ az 5. egyenlet alapján számolt ablak alatti pontosságok közül a maximálist jelenti, az adott teszthalmazra, gépi tanulás esetén, míg átlag()-al a 4. egyenletben felírt képlet alapján kapott pontosságot jelölöm.

V. KONKLÚZIÓ

A kísérlet fő célja az emberi és gépi tanulás összehasonlítása volt, melyhez egy mérési tervet dolgoztam ki. Az emberi tanulást egy böngészőben futó alkalmazás segítségével mértem, míg a gépi tanuláshoz elkészítettem egy iteratív tanulóalgoritmus,

2. táblázat: Teszthalmazonkénti pontosság értékek összesítése

	Teszt 1	**Teszt 2**	**Teszt 3**	**Teszt 4**	**Teszt 5**
$\max(\text{accuracy}_{W,G})$	0,7	0,6	0,9	0,7	0,9
$\text{átlag}(\text{accuracy}_{G})$	0,269	0,217	0,664	0,290	0,420
$\max(\text{accuracy}_{W,E})$	0,9	0,989	1	0,983	1
$\text{átlag}(\text{accuracy}_{E})$	0,716	0,859	0,998	0,901	0,991

melyre azért volt szükség, hogy a lehető legpontosabban mintázzam a tesztelő személyek által végzett osztályozási folyamatot. Az eredményeket kiértékeltem, és egyező beállítások mellett gépi tanulást használva is leosztályoztam összesen 5 különböző teszthalmazt, majd ezek eredményeinek kiértékelése következett. Az eredményeket összehasonlítottam osztályozási pontosság szempontjából, illetve egy csúszó ablak segítségével a tényleges tanulási folyamat „gyorsaságát" is lemértem (főként a gépi tanulásnál látszott ez).

Az emberek előnnyel indulnak a géppel szemben, hiszen rengeteg előzetes információ áll rendelkezésünkre, melyektől képtelenség elvonatkoztatni, és úgy tekinteni egy új osztályozási feladatra, mintha a tanító képeken kívül semmilyen tudással nem rendelkeznénk. Ezért a gépi tanulást fejleszteni kell, hogy minél jobban meg tudja közelíteni az emberi képességeket. A jövőben összetett aktív tanulási stratégiákat fogok kidolgozni és megvalósítani, hogy minél kevesebb tanító mintával minél pontosabb osztályozást tudjak elérni. Az elkészült aktív tanuló rendszerrel az eddig bemutatott képhalmazokat újratesztelem és az új tapasztalatokkal, eredményekkel egészítem ki a kiértékelést. A passzív és aktív tanulás szélesebb körű összeméréséhez további teszthalmazokat fogok gyűjteni, melyek nagyobb méretűek (itt már nem lesz emberi tanulás).

REFERENCIA

[1] Burr Settles, Active Learning Literature Survey, Computer Siences Technical Report 1648, University of Wisconsin – Madison, 2009.

[2] L. Fei-Fei, R. Fergus, and A. Torralba: *Recognizing and Learning Object Categories*, Part1 – Single object classes: *Bag of Words models, Part-based models, and Discriminative models*, 2009, pp. 2-16.

[3] S. Lazebnik, A. Torralba, L. Fei-Fei, D. Lowe, C. Szurka: *Bag-of-Words models*, Lecture 9, 2012, pp. 1-32.

[4] C. Harris, M. Stephens: *A combined corner and edge detector*. In C. J. Taylor, editors, Proceedings of the Alvey Vision Conference, pages 23.1-23.6. Alvey Vision Club, September 1988. doi:10.5244/C.2.23.

[5] Kató Zoltán: *Sarokpontok detektálása*, Képfeldolgozás és Számítógépes Grafika tanszék SZTE. http://www.inf.u-szeged.hu/~kato/teaching/DigitalisKepfeldolgozasTG/07-CornerDetection.pdf (letöltés dátuma: 2014.10.07.)

[6] Mikolajczyk, K., Schmid, C.: *Scale & affine invariant interest point detectors*, International Journal on Computer Vision 60(1), 2004, pp. 63-86.

[7] Lowe, D. G.: *„Distinctive Image Features from Scale-Invariant Keypoints"*, International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.

[8] Reynolds, D. A.: *Gaussian Mixture Models*, Encyclopedia of Biometric Recognition, Springer, Journal Article, February 2008. http://www.ll.mit.edu/mission/communications/ist/publications/0802_Reynolds_Biometrics-GMM.pdf (letöltés dátuma: 2014.10.07.)

[9] Carlo Tomasi: *Estimating Gaussian Mixture Densities with EM – A Tutorial*, Duke University. http://www.cs.duke.edu/courses/spring04/cps196.1/handouts/EM/tomasiEM.pdf (letöltés dátuma: 2014.10.07.)

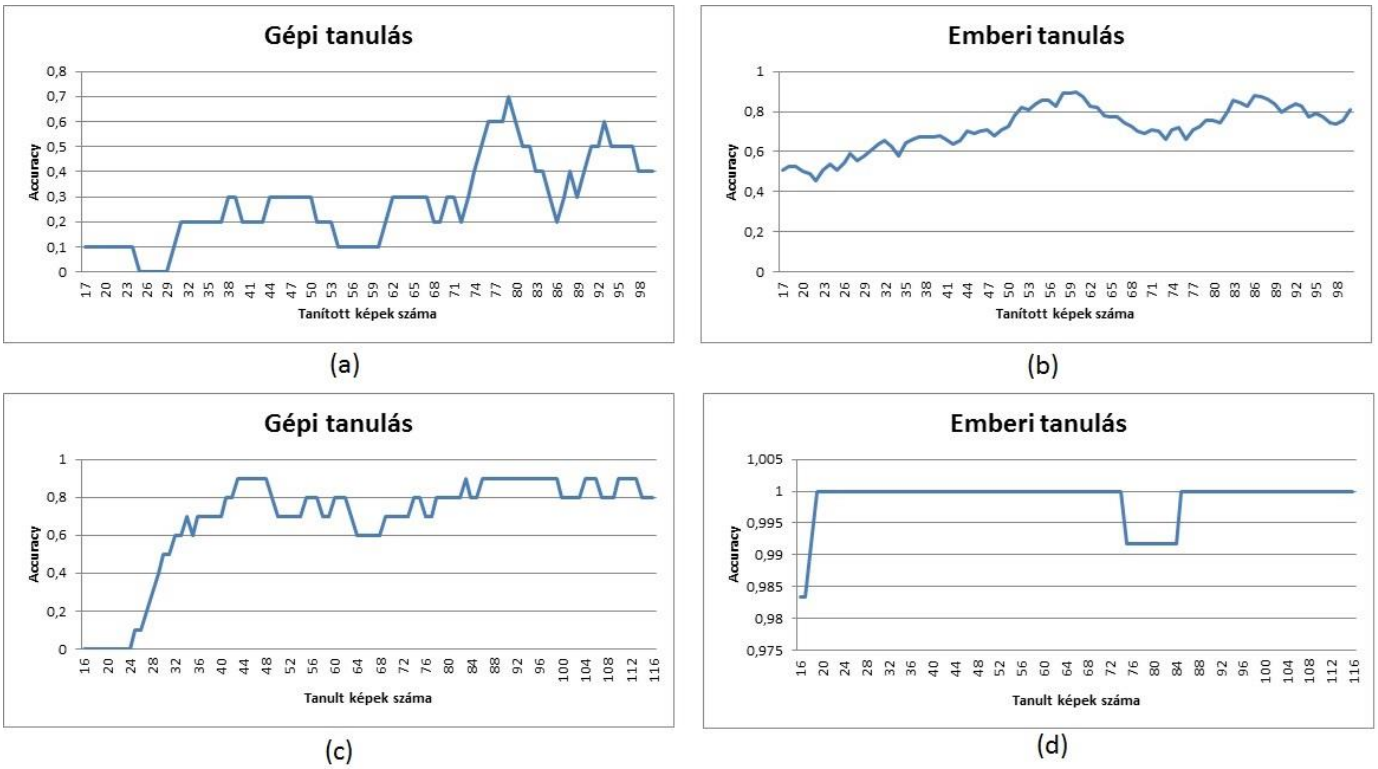
[10] Dempster, A., Laird, N., Rubin, D.: *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society 39(1), 1977, pp. 1–38.

[11] Jaakkola TS, Haussler D.: *Exploiting generative models in discriminative classifiers*. Advances in Neural Information Processing Systems (NIPS), Vol. 11, 1998, pp. 487-493.

[12] Florent Perronnin, Chris Dance: *Fisher kernel on visual vocabularies for image categorization*, CVPR, Computer Vision and Pattern Recognition, 2007.

[13] Boser, B. - Guyon, I. - Vapnik, V.: *A Training Algorithm for Optimal Margin Classifier*, Proc. of the 5th Annual ACM Workshop on Computational Learning Theory, 1992, pp. 144-152.

[14] Corinna Cortes, Vladimir Vapnik: *Support-vector networks*, Machine Learning, Volume 20, Number 3, 1995, pp. 273-297.



4. ábra: Az első képhalmaz tesztelése során kapott átlagos ablak alatti pontosságok a tanított képek számának függvényében; (a): gépi tanulás esetén, (b): emberi tanulás esetén. A harmadik képhalmaz tesztelése során kapott átlagos ablak alatti pontosságok a tanított képek számának függvényében; (c): gépi tanulás esetén, (d): emberi tanulás esetén