SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

PRIMJENA RAZLIČITIH METODA STROJNOG UČENJA U PROBLEMU KLASIFIKACIJE SLIKARSKIH DJELA PREMA AUTORU

Autor: Eva Cetinić, mag. ing.

Predmet: Otkrivanje znanja u skupovima podataka

Zagreb, ožujak 2013.

Sadržaj

1. Uvod

Tehnološke mogućnosti u obradi i analizi slike iznimno se razvijaju u proteklome desetljeću. S ciljem da računala u procesima pretraživanja ili klasifikacije slika dostignu sposobnosti ljudskog vizualnog sustava, intenzivno se istražuju metode ekstrakcije i obrade informacija iz slike. Ključni koraci u ostvarenju programa za računalnu klasifikaciju slikarskih djela jesu ponajprije postupak preslikavanja informacija iz domene ljudskoga djelovanja koju karakterizira kompleksnost umjetničkog izražaja u domenu računalnog djelovanja koja funkcionira isključivo na načelu numeričkoga prikaza informacija, te zatim pravilna interpretacija i uporaba dobivenih numeričkih podataka. Računalna se klasifikacija pokušava ostvariti imitiranjem čovjekovog vizualnog iskustva korištenjem baze digitalno pohranjenih slika, pri čemu su karakteristike slikarskoga djela opisane numeričkim skupom značajki, dok su moždane funkcije učenja i pamćenja ostvarene upotrebom metoda za strojno učenje. U ovom radu naglasak je upravo na aspektu strojnog učenja, te su dani opisi korištenih metoda i usporedba rezultata klasifikacije slikarskih djela prema autoru dobivenih korištenjem različitih klasifikatora.

2. Baza slikarskih djela

Bazu podataka čini skup od ukupno 500 slika, odnosno skup od 20 različitih slikara zastupljenih s 25 slika odabranih iz ukupnog opusa pojedinog autora. Baza obuhvaća autore iz različitih razdoblja i umjetničkih pravaca. Prvi korak u izgradnji baze bio je prikupljanje slikarskih djela u digitalnom formatu korištenjem Interneta (najvažniji izvori: [1], [2] i [3]). U postupku prikupljanja slika važan kriterij pri odabiru slika bilo je postojanje zajedničkih karakteristika među slikarskim djelima pojedinog autora. Moguće je da djela iz ranije i djela iz kasnije faze rada nekog autora budu stilski izrazito različita ili da ne izražavaju karakterističan slikarov rukopis, stoga se pri kreiranju baze slika pojedinog autora težilo tome da slike budu stilski ujednačene i karakteristične za određenog autora. Prikupljene slike pohranjivane su u *.jpg* formatu.

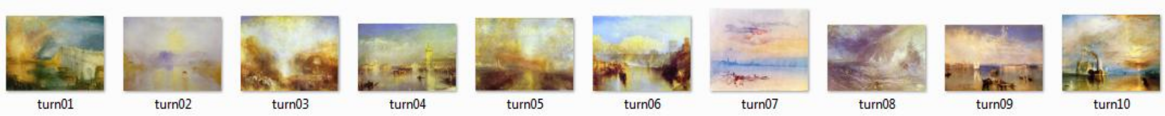
U nastavku je naveden popis odabranih slikara i umanjeni prikaz nekih od njihovih djela koja su korištena za klasifikaciju:

- Sandro Botticelli

- Michelangelo Merisi da Caravaggio

- Rembrandt van Rijn

- Joseph Mallord William Turner



- Claude Monet

- Vincent van Gogh

- John William Waterhouse

- Odilon Redon

- Henri Rousseau

- Paul Gauguin

- Gustav Klimt

- Alphonse Mucha

- Pablo Picasso

- Salvador Dali

- René Magritte

- Wassily Kandinsky

- Kazimir Malevich

- Jackson Pollock

- Roy Lichtenstein

- H. R. Giger

3. Značajke slike

Ključan korak u procesu klasifikacije jest preslikavanje informacija o karakteristikama slikarskog djela kao što su kombinacija boja, modulacija svjetlosti i sjene, mekoća rubova, jedinstveni slikarski rukopis sadržan u potezima kista, materičnosti namaza boje i karakterima linija u skup numeričkih veličina dobivenih primjenom raznih metoda za ekstrakciju značajki slike. U nastavku je dan pregled značajki slike koje su korištene u ovom radu.

Amplitudne značajke

Amplitudne značajke slike određuju parametri izračunati iz vrijednosti intenziteta piksela u na razini cijele slike. Prije izračunavanja tih vrijednosti slika je transformirana iz RGB modela prikaza u crno-bijeli način prikaza slika. Ovdje korištene amplitudne značajke predstavljaju neke od osnovnih statističkih informacije kao što su:

- srednja vrijednost
- varijanca
- standardna devijacija
- raspon varijacije intenziteta piksela

Značajke histograma prvog reda

Korištene su vrijednosti histograma prvog reda izračunate za crno-bijeli prikaz slike i za HSV (eng. *hue, saturation, value*) prikaz slike, te značajke dobivene iz tih histograma. Korištene značajke su:

- Histogram crno-bijele slike
- Histogrami matrice tona boje (H), zasićenja boje (S) i svjetline (V)
- srednje vrijednosti histograma
- varijance histograma
- energije histograma
- entropije histograma

Značajke teksture

Za ekstrakciju obilježja teksture korištene su značajke histograma drugog reda. Histogram drugog reda matrica je čiji elementi predstavljaju broj pojavljivanja određenog para intenziteta na slici u nekom međusobnom prostornom odnosu. Na temelju histograma drugog reda izračunate su sljedeće korištene značajke:

- kontrast
- korelacija
- energija
- homogenost

Detekcija rubova

Detekcija rubova jedna je od najvažnijih i najkorištenijih operacija u analizi slike. Rubovi su područja slike s velikim razlikama u intenzitetu točaka i predstavljaju granice objekata na slici. Budući da količina rubnih područja sadrži informaciju o razini kontinuiranosti ili detaljiziranosti slike, udio rubnih područja i broj regija u slici značajke su koje mogu biti korisne prilikom treniranja neuronske mreže za klasifikaciju slikarskih djela različitih autora.

Korištene značajke su:

- udio rubnih piksela dobiven korištenjem Robertsovog algoritma za detekciju rubova
- udio rubnih piksela dobiven korištenjem Sobelovog algoritma za detekciju rubova
- udio rubnih piksela dobiven korištenjem Cannyjevog algoritma za detekciju rubova
- broj različitih regija u slici

Frekvencijske značajke

Frekvencijske značajke čine onu grupaciju značajki koje su dobivene iz frekvencijske domene transformirane slike. Za dobivanje informacija o karakteristikama slike u frekvencijskoj domeni odabrane su dvije vrste frekvencijskih transformacija: diskretna Fourierova transformacija i diskretna valićna (eng. *wavelet*) transformacija.

Korištene značajke su:

- energija transformacijskih koeficijenata dobivenih DFT-om
- faktor učinkovitosti DFT-a
- udio energije aproksimacijskih koeficijenata dobivenih valićnom transformacijom
- udio energije koeficijenata detalja dobivenih valićnom transformacijom

4. Metode strojnog učenja

4.1. Neuronska mreža

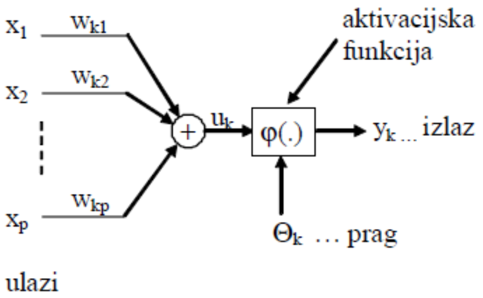
Umjetna neuronska mreža (eng. *Artificial Neural Network - ANN*) je skup međusobno povezanih jednostavnih procesnih elemenata (jedinica, čvorova) čija se funkcionalnost temelji na biološkom neuronu i koji služe distribuiranoj paralelnoj obradi podataka. U širem smislu umjetnu neuronsku mrežu moguće je opisati kao sustav koji se temelji na oponašanju rada ljudskog mozga, ponajviše u nastojanju simuliranja postupka učenja i obrade podataka. Neuronske mreže prikladne su za rješavanje problema kod kojih postoji složena veza ulaza i izlaza kao što su klasifikacija i predviđanja. Sposobnost mreže za rješavanje takvih problema posljedica je stvaranja veze među procesnim elementima, tzv. neuronima, koja se postiže kroz proces adaptacije ili učenjem iz skupa primjera. S obzirom na arhitekturu mreže, način učenja i aktivacijsku funkciju razlikujemo više vrsta neuronskih mreža. U ovom radu je za kvalifikaciju slikarskih djela, na temelju prethodno opisanih značajki, korištena neuronska mreža poznata kao višeslojni perceptron, odnosno višeslojna neuronska mreža bez povratnih veza (eng. *multilayer feed-forward network, multilayer perceptron – MLP*) čije su karakteristike detaljnije opisane u nastavku.

Višeslojni perceptron

Perceptron je najjednostavnija neuronska mreža koja se sastoji od jednog neurona i kao takva predstavlja osnovnu građevnu jedinicu višeslojnih neuronskih mreža.

Model neurona

Neuron je osnovni procesni element neuronske mreže, zamišljen kao matematička funkcija koja imitira primitivan model biološkog neurona. Model neurona prikazan je na slici 1.



$$u_k = \sum_{j=1}^{p} w_{kj} x_j$$

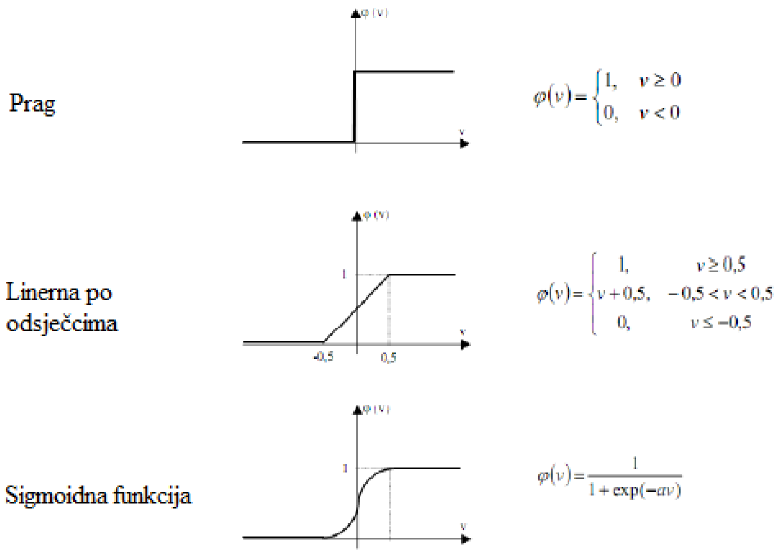
$$y_k = \varphi(u_k - \Theta_k)$$

Slika 1. Model neurona

Elementi modela neurona su:

- skup sinapsi, tj. ulaza ($x_1 ... x_p$) od kojih svaki ima svoju težinu ($w_{k1} ... w_{kp}$) (signal x_j na ulazu j neurona k ima težinu w_{kj}),
- sumator za zbrajanje otežanih ulaza, tj. računanje linearne kombinacije ulaza,
- aktivacijska funkcija koja ograničava izlaz neurona na interval [0,1].

Aktivacijska funkcija može biti linearna ili nelinearna. Neke od najčešćih aktivacijskih funkcija prikazane su na slici 2.



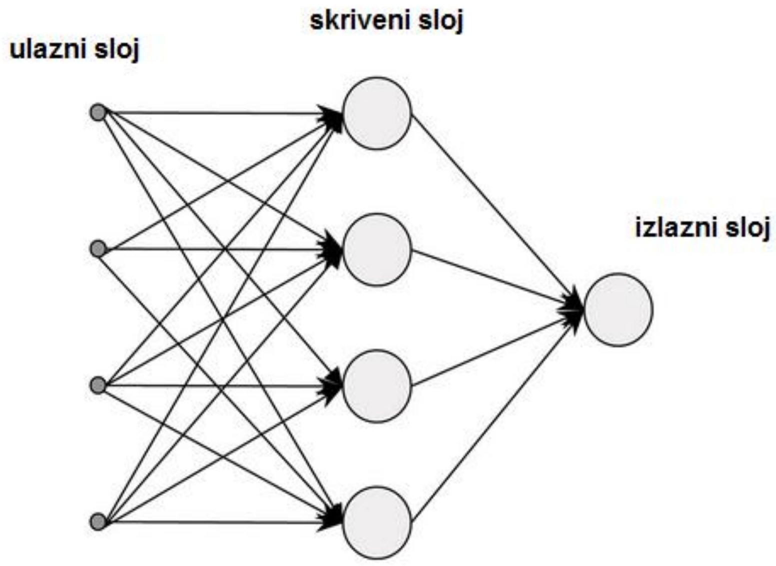
Slika 2. Različite aktivacijske funkcije

Višeslojni perceptron karakterizira nelinearna aktivacijska funkcija, najčešće sigmoidna funkcija.

Arhitektura neuronske mreže

Kod umjetne neuronske mreže neuroni su međusobno organizirani u slojeve. Postoje tri vrste slojeva: ulazni, skriveni i izlazni sloj. Svaka mreža ima točno jedan ulazni i jedan izlazni sloj, dok broj skrivenih slojeva može varirati od nula do bilo kojeg broja. Ulazni sloj prima informacije iz okoline, izlazni sloj prikazuje rezultat mreže, a skriveni sloj obrađuje informacije i šalje ih u neurone izlaznog sloja. Svaki neuron jednog sloja može biti povezan sa svakim neuronom istog ili drugog sloja. Veza između neurona može biti jednosmjerna ili povratna. Broj neurona u sloju, broj neurona i način povezivanja neurona određuje arhitekturu mreže. S obzirom na arhitekturu mreže razlikujemo: mreže bez povratne veze koje mogu biti jednoslojne ili višeslojne, mreže s povratnom mrežom te ljestvičaste mreže.

Višeslojni perceptron je višeslojna je neuronska mreža bez povratne veze koja osim ulaznog i izlaznog sloja može imati jedan ili više skriveni sloj.



Slika 3. Arhitektura neuronske mreže

Učenje neuronske mreže

Rad umjetne neuronske mreže odvija su u dvije faze: najprije se odvija faza učenja ili treniranja neuronske mreže, a zatim slijedi faza testiranja. Učenje mreže je proces u kojem se mijenjaju iznosi težina ($w_{kj}(n)$) u mreži, a odvija se kao reakcija na podatke izvana koji su predstavljeni u ulaznom sloju, a u nekim mrežama i u izlaznom sloju. Postoje različiti algoritmi učenja mreža koji određuju način izračunavanja težina u mreži te različite paradigme učenja određuju odnos neuronske mreže prema okolini.

Osnovne paradigme učenja neuronske mrežu su:

- *učenje pod nadzorom* - postoji znanje o okolini u obliku parova ulaz-izlaz; vrši se ponavljanje procesa učenja dok mreža ne nauči imitirati odnos ulaz-izlaz;
- *učenje podrškom* – ulazno-izlazno preslikavanje kroz proces pokušaja i pogreške u kojem se maksimizira indeks mjere kvalitete učenja;

- *učenje bez nadzora* – optimizacija parametara s obzirom na mjeru znanja koju mreža mora zadovoljavati.

Višeslojni perceptron neuronska je mreža koja u postupku učenja koristi algoritam s povratnom propagacijom pogreške, tzv. BP algoritam (engl. *back-propagation*), koji spada u skupinu algoritama koji se temelje na učenju pod nadzorom. Osnovna ideja BP algoritma jest smanjivanje razlike između izlaza koje daje neuronska mreža i izlaza zadanim u primjerima za učenje, odnosno minimizacija funkcije pogreške. Za ocjenu kvalitete neuronske mreže koristi se prosječna kvadratna pogreška za sve uzorke, a cilj učenja je odrediti težine tako da se funkcija pogreške minimizira za što se koristi deterministički algoritam gradijentnog opadanja (engl. *gradient descent*)

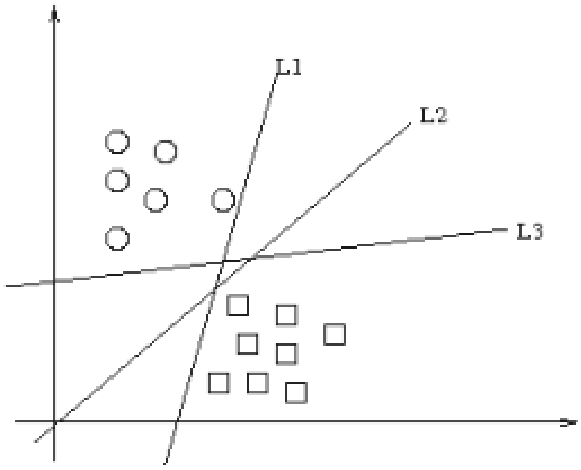
Tok podataka kroz mrežu moguće je opisati u nekoliko koraka:

- od ulaznog sloja prema skrivenom sloju: ulazni sloj učitava podatke iz ulaznog vektora i šalje ih u skriveni sloj;
- u skrivenom sloju: neuroni u skrivenom sloju primaju „otežane" ulazne podatke i prenose ih u izlazni sloj koristeći prijenosnu funkciju;
- kako podaci putuju mrežom računaju se sumirani ulazi i izlazi za svaku jedinicu obrade;
- u izlaznom sloju: za svaki neuron se računa skalirana, lokalna greška koja se upotrebljava u određivanju povećanja ili smanjenja težina;
- propagiranje unatrag od izlaznog sloja do skrivenih slojeva: skalirana, lokalna greška, te povećanje ili smanjenje težina računa se za svaki sloj unatrag i težine se sukladno tome podešavaju.

U izvedbi BP algoritma učenje se odvija uzastopnim prezentiranjem parova za treniranje. Jedna prezentacija svih parova za treniranje naziva se epoha. Učenje se odvija epohu za epohom dok se srednja kvadratna pogreška dovoljno ne smanji i težine i pragovi mreže ne stabiliziraju

4.2. Metoda potpornih vektora

Metoda potpornih vektora (eng. *Support Vector Machine – SVM*) poznata je i kao metoda maksimalno granične hiperravnine jer njezin cilj nalaženje hiperravnine koja razdvaja hiperprostor primjera u dva poluprostora koji odgovaraju dvjema klasama podataka tako da je udaljenost te hiperravnine od najbližih točaka podataka maksimalna.



Slika 4. Razdvajanje podataka u dvije klase

Hiperravnina određena je vektorom *w* i jednadžbom:

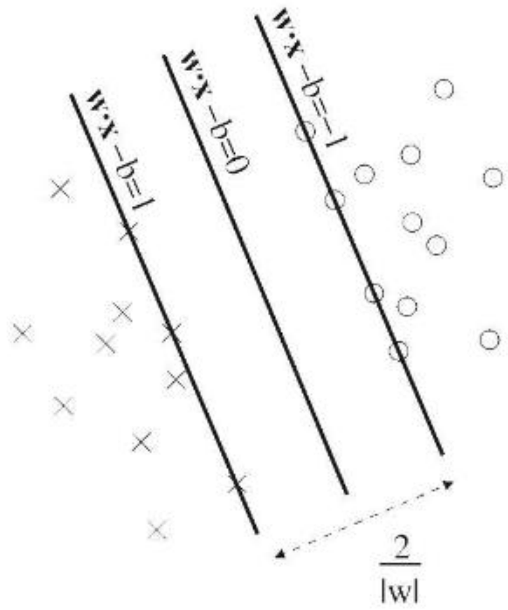
$$w \cdot x - b = 0$$

Podaci su linearno separabilni ako postoje dvije hiperravnine paralelne zadanoj (u različitim poluprostorima) između kojih nema točaka iz skupa primjera, tj. $|w \cdot x_i - b| \geq 1$. Primjeri koji leže na tim paralelnim hiperravninama nazivaju se potporni vektori (eng. *support vector*).

Potporni vektori mogu se opisati jednadžbama:

$$w \cdot x - b = 1$$

$$w \cdot x - b = -1$$



Slika 5. Hiperravnina i potporni vektori

Ako pretpostavimo skup primjera podataka oblika $\{(x_1 c_1), \dots, (x_n c_n)\}$ gdje su $c_i \in \{-1, 1\}$, zadatak se svodi na maksimizaciju margine, tj. udaljenosti među potpornim vektorima $\frac{2}{\|w\|}$, odnosno na minimizaciju $\|w\|$. Također, u svrhu sprječavanja pojave podataka (točaka) unutar margine dodaje se novi uvjet za svaki x_i:

$$w \cdot x_i - b \geq 1 \ \text{ za } x_i \text{ iz prve klase}$$

$$w \cdot x_i - b \leq 1 \ \text{ za } x_i \text{ iz druge klase}$$

Jednadžbe () i () mogu se zapisati kao:

$$c_i(w \cdot x_i - b) \geq 1$$

Maksimizacija margine ovisi o minimizaciji $\|\boldsymbol{w}\|$, budući da je to operacija modula koja u sebi sadrži korjenovanje koje je računski zahtjevno, vrši se supstitucija $\|\boldsymbol{w}\| \to \frac{1}{2}\|\boldsymbol{w}\|^2$ koja neće promijeniti konačni rezultat. Za pronalazak minimuma od $\frac{1}{2}\|\boldsymbol{w}\|^2$ uvode se Lagrangeovi multiplikatori α. Metoda Langrangeovih operatora pruža strategiju za pronalazak lokalnih maksimuma i minimuma neke funkcije. Uvođenjem Lagrangeovih multiplikatora, prethodno navedeni uvjeti svode se na:

$$\min_{w,b} \max_{\alpha \geq 0} \left\{ \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i [c_i(w \cdot x_i - b - 1)] \right\}$$

Za određivanje Langrangeovih multiplikatora pravilo klasifikacije zapisuje se u dualnom obliku i otkriva da maksimum margine ovisi o vektorima potpore. Uz činjenicu da je $\|\boldsymbol{w}\|^2 = w \cdot w$ i supstituciju $\boldsymbol{w} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \alpha_i \, c_i x_i$ dobiva se sljedeći izraz:

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j c_i \, c_j x_i, x_j = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j c_i \, c_j K(x_i, x_j)$$

gdje se s $K(x_i, x_j)$ označava kernel funkcija.

Izvorni problem klasifikacije može se definirati u nekom konačnom dimenzionalnom prostoru, no često se događa da klase koje se žele klasificirati nisu linearno separabilne u tom prostoru. Navedeni problem rješava se preslikavanjem originalnog ulaznog prostora u prostor više dimenzionalnosti. Izvorište takvom rješenju jest Coverov teorem koji kaže da prelaskom u višedimenzionalni prostor raste vjerojatnost linearne separabilnosti. Osnovna ideja svodi se na nelinearno preslikavanje ulaznog prostora u novi prostor značajki više dimenzionalnosti u kojem se zatim konstruira optimalna ravnina razdvajanja.

Nelinearni klasifikatori koriste se onda kada klase nisu linearno separabilne i nije moguće između njih povući linearni klasifikator. Kreiranje nelinearnih klasifikatora vrši se korištenjem kernel funkcija u svrhu maksimiziranja margine razdvajanja. Algoritam pronalaska nelinearnih klasifikatora isti je kao i kod linearnih klasifikatora, osim što se u tom slučaju svaki skalarni produkt zamijeni nelinearnom kernel funkcijom.

Jednadžba koja opisuje ravninu razdvajanja tada se opisuje kao:

$$\sum_{j=1}^{m} w_j \varphi_j(x) - \; b = 0$$

Vidljivo je da razlika u odnosu na jednadžbu koja opisuje linearni klasifikator korištenje $\varphi(x)$ umjesto x, što predstavlja vektor x u novom prostoru značajki. Postupak određivanja nelinearnog klasifikatora identičan je onomu za određevinje linearnog klasifikatora uz suptituciju $x \to \varphi(x)$.

Prethodno spomenuta kernel funkcija u slučaju linearnog klasifikatora predstavlja skalarni produkt vektora x_i, x_j, a slučaju nelinearnog klasifikatora kernel funkcija izravno je povezana s tranformacijom φ:

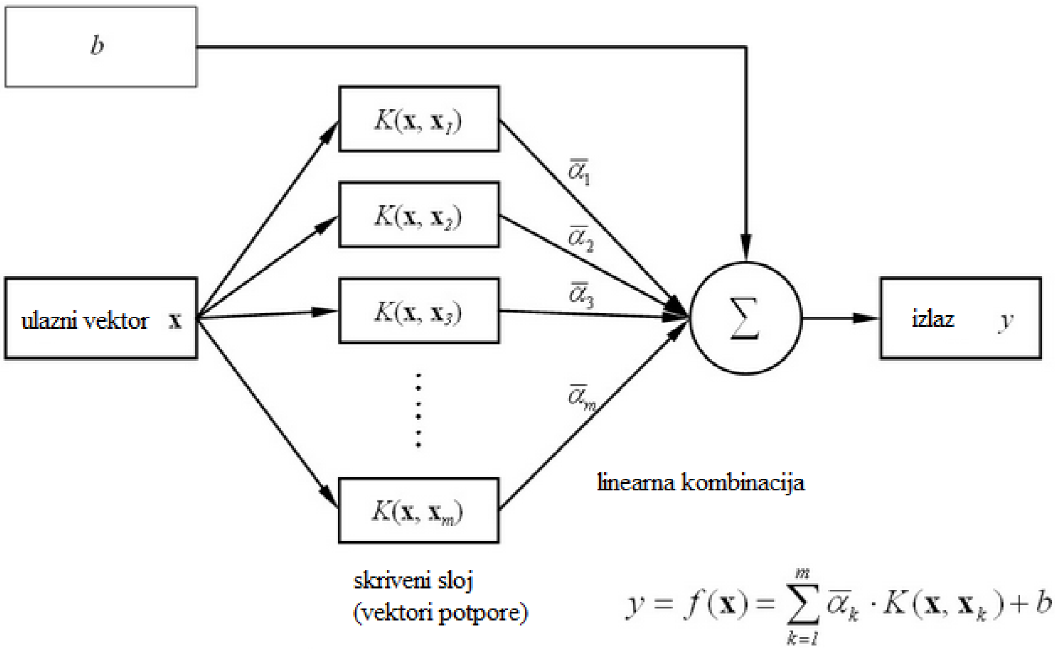
$$K\left(x_i, x_j\right) = \varphi(x_i)\varphi(x_j)$$

Postoji određena sloboda u izboru kernel funkcije, no nužno je da odabrana funkcija zadovoljava određena svojstva (Merceov teorem). Korištenjem kernelovih funkcija omogućava se da algoritam pronađe ravninu razdvajanja s maksimalnom marginom u transformiranom prostoru značajki. Neke of najkorištenijih kernel funkcija su:

- Polinomni homogeni kernel: $K(x_i, x_j) = (x_i x_j)^d$
- Polinomni nehomogeni kernel: $K(x_i, x_j) = (x_i x_j + 1)^d$
- Gaussov radijalni kernel: $K(x_i, x_j) = e^{\left(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2\right)}$
- Hiperbolični kernel: $K(x_i, x_j) = \tanh(\kappa x_i \cdot x_j + c)$

Arhitektura SVM klasifikatora

Arhitektura SVM klasifikatora prikazana je na slici:



Slika 6. Arhitektura SVM klasifikatora

4.3. Naivni Bayesov klasifikator

Bayesov teorem

Bayesov teorem omogućuje odabir najvjerojatnije hipoteze iz skupa hipoteza H na osnovu skupa za učenje D, uz utjecaj predodređenih vjerojatnosti svake od ponuđenih hipoteza u skupu H. Za preciznije objašnjenje teorema, potrebno je definirati vjerojatnosti:

- P(h) – početna vjerojatnost hipoteze h. Ova vjerojatnost omogućuje prikazivanje početnog znanja o vjerojatnostima različitih hipoteza. Ukoliko ne postoji takvo znanje, može se svim hipotezama pridijeliti jednaka početna vjerojatnost.
- P(D) – početna vjerojatnost pojavljivanja instance D. Vjerojatnost izražava koja je vjerojatnost pojavljivanja D bez obzira na to koja je hipoteza ispravna.
- P(D|h) – uvjetna vjerojatnost pojavljivanja D uz uvjet ispravnosti hipoteze h.
- P(h|D) – uvjetna vjerojatnost ispravnosti hipoteze h nakon pojavljivanja instance D. Ova vjerojatnost omogućuje procjenu ispravnosti hipoteza nakon promatranja pojave novih instanci D.

Bayesov teorem omogućava izračunavanje P(h|D) preko:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Često je potrebno pronaći najvjerojatniju hipotezu h iz H uz uvjet pojavljivanja D. Takvu hipoteza naziva se *maximum a posteriori* (*MAP*) hipoteza. MAP hipotezu izračunavamo primjenjujući Bayesov teorem na svaku hipotezi h iz skupa H i zatim odabirući najvjerojatniju:

$$h_{MAP} = \arg max_{h \in H} P(h|D) = \arg max_{h \in H} \frac{P(D|h)P(h)}{P(D)} = \arg max_{h \in H} P(D|h)P(h)$$

U zadnjem koraku isključena je P(D) jer ta vjerojatnost predstavlja konstantu neovisnu o hipotezi h. Daljnje pojednostavljenje slijedi ukoliko pretpostavimo da su sve hipoteze iz skupa H jednako vjerojatne, tada je moguće zanemariti utjecaj P(h) te procjenjujemo samo na osnovi P(D|h). Hipotezu koja maksimizira P(D|h) nazivamo *maximum likelihood* (*ML*) hipoteza:

$$h_{ML} = \arg max_{h \in H} P(D|h)$$

Bayesov teorem ima široku upotrebu u klasifikacijskim postupcima. U ovom radu je korišten i detaljnije objašnjen Naivni Bayesov klasifikator.

Naivni Bayesov klasifikator

Ukoliko problem klasifikacije predstavimo kao problem pronalaženja najvjerojatnije klasifikacije v_{MAP} tada se v_{MAP} može računat prema:

$$v_{MAP} = \arg max_{v_j \in V} P(v_j | a_1, a_2, .. a_n)$$

U navedenom izrazu v_{MAP} predstavlja najvjerojatniji element konačnog skupa V svih mogućih klasifikacija ulazne instance. Svaka instanca prikazana je kao skup vrijednosti atributa $\{a_1, a_2, .. a_n\}$. Poznati je skup trening instanci definiran također istim skupom atributa. Prethodni izraz može se pisati i kao:

$$v_{MAP} = \arg max_{v_j \in V} P\left(v_j \middle| a_1, a_2, .. a_n\right) = \arg max_{v_j \in V} \frac{P\left(a_1, a_2, .. a_n \middle| v_j\right) P(v_j)}{P(a_1, a_2, .. a_n)}$$

$$= \arg max_{v_j \in V} P\left(a_1, a_2, .. a_n \middle| v_j\right) P(v_j)$$

Vrijednost izraza računa se na osnovu podataka za treniranje. Vjerojatnost $P(v_j)$ lako se izračunava jednostavnim prebrojavanjem pojavljivanja tražene rezultantne klasifikacije. Problem izračuna izraza $P\left(a_1, a_2, .. a_n \middle| v_j\right)$ proizlazi iz međusobne zavisnosti vrijednosti atributa $a_1, a_2, .. a_n$ tako da je broj mogućih $P\left(a_1, a_2, .. a_n \middle| v_j\right)$ izraza jednak broju svih mogućih n-torki $\{a_1, a_2, .. a_n\}$ pomnoženih s brojem svih mogućih klasifikacija. Stoga bi svaki primjer morao biti dio ulaznog skupa mnogo puta kako bi se pouzdano mogla ocijeniti tražena vjerojatnost.

Naivni Bayesov klasifikator uvodi pojednostavljenje pretpostavljanjem međusobne nezavistnosti vrijednosti atributa u n-torkama $\{a_1, a_2, .. a_n\}$ tako da vrijedi izraz:

$$P\left(a_1, a_2, .. a_n \middle| v_j\right) = \prod_i P(a_i | P\left(a_1, a_2, .. a_n \middle| v_j\right)$$

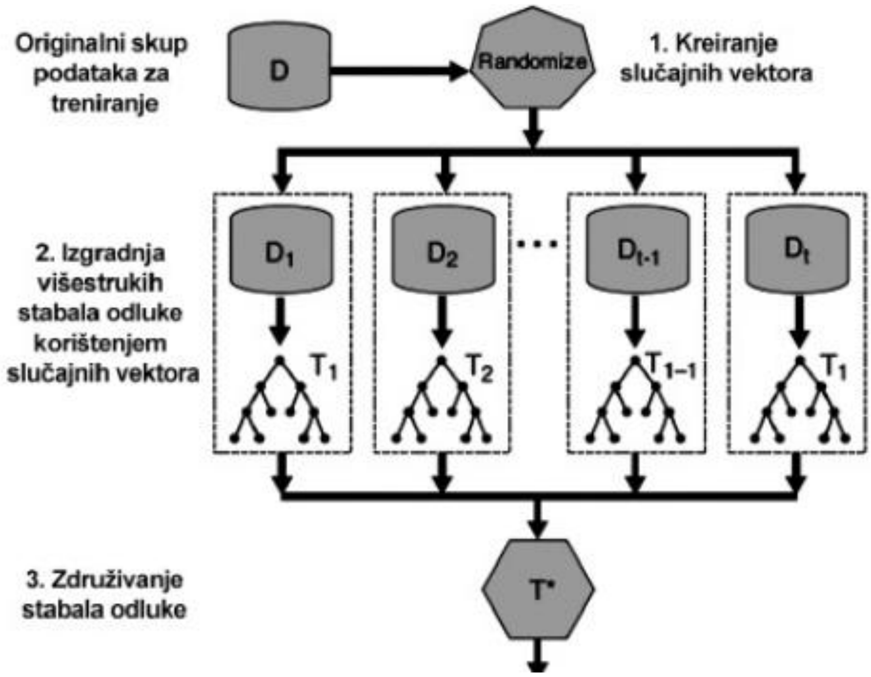
Izraz za klasifikaciju Naivnim Bayesovim klasifikatorom sada glasi:

$$v_{NBj} = \arg max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

Broj različitih vjerojatnosti $P(a_i|v_j)$ koje treba izračunati iz podataka za treniranje u ovom slučaju jednak je broju različitih vrijednosti atributra pomnoženog s brojem različitih mogućih klasifikacija što ispada mnogo manji broj izračuna nego što je potrebno za dobivanje vjerojatnosti $P(a_1, a_2, .. a_n|v_j)$. Pretpostavljeni uvjet nezavisnosti relativno je strog i može predstavljati problem u realnim situacijama. Međutim, u praktičnoj upotrebi Naivni Bayesov klasifikator pokazao se korisnim zbog jednostavnosti implementacije i zadovoljavajućih rezultata.

4.4. Metoda slučajne šume

Slučajna šuma (eng. *Random Forest*) je općenit naziv za skupinu metoda koje se koriste kolekcijom stablastih klasifikatora $\{h(\boldsymbol{x}, \theta_k), k = 1, \dots\}$ pri čemu je $\{\theta_k\}$ skup nezavisnih slučajnih vektora jednake distribucije, a $\boldsymbol{x}$ ulazni vektorski uzorak. Svako stablo daje svoj glas za ulazni uzorak, a šuma određuje klasu uzorka na temelju većine glasova.



Slika 7. Princip rada slučajnih šuma

Prilikom treniranja algoritam slučajne šume stvara velik broj stabala, a kup za treniranje pojedinog stabla dobiva se izdvajanjem *N* uzoraka iz početnog skupa za treniranje iste veličine slučajnim odabirom s ponavljanjem. Tako odabran skup za treniranje iste je veličine kao i originalni skup, te se neki od primjera mogu pojaviti više puta, a neki nijednom. Skupovi tako odabranih primjera nalaze se u korijenima stabala u šumi. Ostali, neodabrani uzorci čine oko trećinu početnog skupa koje interni estimatori koriste za praćenje i procjenu nepristrane pogreške klasifikacije, generalizacijske pogreške, snage i korelacije pojedinih stabala, kao i procjenu važnosti pojedinih atributa ulaznih instanci.

Osim formiranja skupa za treniranje, za svako se stablo šume slučajnim odabirom bira *m* od ukupno *M* atributa te se koriste oni koji omogućuju najbolje grananje. Vrijednost *m* određuje se unaprijed, najčešće se uzima iz intervala [1,*M*] i konstantna je za cijelu šumu. Atributi najpogodniji za grananje nalaze se na temelju mjera nečistoće. Atribut s najvećom vrijednošću indeksa nečistoće je onaj na temelju kojeg se vrši grananje.

Točnost slučajnih šuma

Uz dan skup klasifikatora $h_1(x)$, $h_2(x), \ldots, h_K(x)$, te skup podatka za treniranje nasumce odabranom iz distribucije slučajnih vektora X i Y, definicija funkcije margine je:

$$mg(X,Y) = av_k I(h_k(X) = Y) - \max_{y=Y} av_k I(h_k(X) = j)$$

pri čemu je $I(\cdot)$ indikatorska funkcija. Margina je mjera koja odražava koliko prosječan broj glasova za ispravnu klasu nadmašuje prosječan broj glasova za bilo koju drugu klasu. Veća margina znači veću pouzdanost klasifikacije.

Greška generalizacije:

$$PE^* = P_{X,Y}(mg(X,Y) < 0)$$

pri čemu ineksi X,Y upućuju na vjerojatost definiranu prostorom X,Y. Kako se broj stabala povećava, za gotovo sve uzorke θ, PE^* konvergira prema:

$$P_{X,Y}(P_\theta(h(X,Y) = Y) - \max_{y=Y} P_\theta(h(X,\theta) = j) < 0)$$

Iz navedenog izraza vidljivo je da ppovećanje broja stabala ne dovodi do pretreniranosti već do granične vrijednosti generalizacijske pogreške koja se definira kao:

$$PE^* \leq \frac{\overline{\rho}(1 - s^2)}{s^2}$$

pri čemu je $\overline{\rho}$ srednja vrijednost korelacije definirana s:

$$\overline{\rho} = \frac{E_{\theta,\theta'}(\rho(\theta,\theta')sd(\theta)sd(\theta')}{E_{\theta,\theta'}(sd(\theta)sd(\theta')}$$

a s predstavlja snagu skupa klasifikatora $\{h(\boldsymbol{X},\theta)\}$

5. Implementacija

Za učitavanje, predobradu, ekstrakciju značajki korišten je programski alat *MATLAB*. Dobivene značajke slika pohranjene su u matricu dimenzija 500x107, s obzirom da je broj korištenih slika 500, a ukupni broj svih značajki dobivenih iz jedne slike iznosi 107. Nakon izračunavanja matrice značajki, elementi matrice značajki su standardizirani, tj. od svake značajke oduzeta je srednja vrijednost te značajke, te je zatim podijeljena standardnom devijacijom.

Za klasifikaciju slika upotrebom opisanih metoda klasifikacije korišten je programski alat *Weka*. Kako bi bilo moguće podatke, tj. matricu značajki slika, dobivene u *MATLAB*-u učitati u programski alat *Weka,* koji se temelji na *Java* programskog jeziku, bilo je potrebno provesti transformaciju podataka iz *mat*. u *arff*. datoteku. Za to su korištene funkcije *matlab2weka*, *wekaPathCheck* i *saveArff* koje dostupne na službenim Internet stranicama *MATLAB*-a [4].

Nakon transformacije datoteka, uspješnog učitani podataci korišteni su za treniranje modela četiriju različitih klasifikatora sa sljedećim karakteristikama:

- Višeslojni perceptron
  - 1 ulazni sloj s (broj značajki) neurona
  - 1 izlazni sloj s (broj klasa) neurona
  - 1 skriveni sloj s ((broj značajki + broj klasa) / 2) neurona
- SMO – algoritam sekvencijalne minimalne optimizacije SVM-a
  - Polinomna kernel funkcija
- Naivni Bayesov klasifikator
- Slučajne šume
  - 1000 generiranih stabala
  - m = ($\log_2$(broj značajki) + 1) slučajno odabranih značajki

Modeli klasifikatora izgrađeni su korištenjem podataka nad kojima je primijenjena 10-struka međuvalidacija. Skup podataka od 500 slika, slučajno je podijeljen na 10 podskupova od kojih svaki sadržava 1/10 ukupnih ulaznih podataka. Za treniranje klasifikatora koriste se 9 podskupa, a testiranje se provodi s preostalim jednim podskupom. Taj postupak se ponavlja deset puta, svaki put s drugim podskupom za testiranje. Pogreška je srednja vrijednost svih deset izračuna.

6. Rezultati

Mjere uspješnosti

Za procjenu uspješnosti klasifikacije korištene su sljedeće mjere:

- Postotak točno klasificiranih slika
- Postotak netočno klasificiranih slika
- Kappa statistika - mjera uspješnosti promatranog klasifikatora prema idealnom uz korekciju slučajnog izbora
- Preciznost: $\frac{TP}{TP+FP}$
- Odziv: $\frac{TP}{TP+FN}$

  gdje su:
  - TP (eng. *true positives*) – primjeri klase *c* klasificirani u klasu *c*
  - FP (eng. *false positives*) - primjeri koji nisu dio klase *c* ali su klasificirani u klasu *c*
  - FN (eng. *false negatives*) – primjeri klase *c* koji nisu klasificirani u klasu *c*

- F1 mjera: $F_1 = 2 \cdot \frac{(preciznost) \cdot (odziv)}{(preciznost) + (odziv)}$

- Konfuzijska matrica

Usporedba rezultata klasifikacije

Klasifikator / Mjera uspješnosti	MPL	SMO	Naivni Bayes	Slučajne šume
točno klasificirane slike	74.4 %	73.4 %	64.6 %	73.8 %
netočno klasificirane slike	25.6 %	26.6 %	35.4 %	26.2 %
kappa statistika	0.7305	0.72	0.6274	0.7263
preciznost	0.742	0.742	0.65	0.737
odziv	0.744	0.734	0.646	0.738
F1 mjera	0.742	0.733	0.642	0.735

Tablica 1. Vrijednosti mjera uspješnosti za korištene klasifikatore

Podaci u gornjoj tablici pokazuju da najbolje rezultate u problemu klasifikacije slikarskih djela prema autoru s obzirom na navedeni skup značajki slike, daje višeslojni perceptron. Neznatno lošije i vrlo slične rezultate daju metoda slučajnih šuma i metoda potpornih vektora (SMO), dok se najlošiji rezultati klasifikacije postižu korištenjem Naivnog Bayesovog klasifikatora.

Konfuzijske matrice

Za bolje razumijevanje i interpretaciju dobivenih rezulatata klasifikacije slikarskih djela prema autoru, naročito s obzirom na sličnosti i različitosti djela pojedinih autora, u nastavku su prikazane konfuzijske matrice dobivenih modela klasifikatora.

- **Višeslojni perceptron**

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
14	0	1	0	0	0	1	1	0	0	2	0	2	1	0	2	0	0	0	1	a = Botticelli
0	23	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	b = Caravaggio
0	2	22	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	c = Rembrandt
0	0	0	19	1	0	2	1	0	0	0	0	0	0	2	0	0	0	0	0	d = Turner
0	0	0	1	16	1	1	4	0	0	0	0	0	0	1	1	0	0	0	0	e = Monet
0	0	0	1	0	21	0	0	1	0	1	0	0	0	0	0	0	0	0	1	f = VanGogh
2	0	0	1	1	0	18	1	0	0	1	0	0	0	0	0	0	0	0	1	g = Waterhouse
0	0	0	2	1	0	0	18	1	2	0	0	1	0	0	0	0	0	0	0	h = Redon
1	1	0	0	0	0	1	0	17	0	1	0	0	2	1	0	0	0	0	1	i = Rousseau
0	0	1	0	2	0	0	1	0	19	0	0	0	2	0	0	0	0	0	0	j = Gauguin
1	0	0	0	2	4	0	0	2	0	14	0	0	0	0	1	0	1	0	0	k = Klimt
0	0	0	0	0	0	0	1	0	0	0	23	0	1	0	0	0	0	0	0	l = Mucha
2	0	0	0	0	0	0	0	0	0	1	0	14	2	2	4	0	0	0	0	m = Picasso
1	1	0	0	0	0	0	1	1	1	0	0	1	17	0	2	0	0	0	0	n = Dali
0	0	0	2	1	0	0	1	0	0	0	0	1	0	19	1	0	0	0	0	o = Magritte
3	0	0	1	0	0	0	0	0	1	2	1	5	1	0	8	2	0	1	0	p = Kandinsky
0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	22	0	0	0	q = Malevich
0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	22	1	0	r = Pollock
0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	23	0	s = Lichtenstein
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	23	t = Giger

- **SMO**

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
17	0	1	0	0	1	1	1	0	1	0	0	2	0	0	1	0	0	0	0	a = Botticelli
0	20	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	b = Caravaggio
0	6	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = Rembrandt
0	0	0	17	4	0	2	1	0	0	0	0	0	1	0	0	0	0	0	0	d = Turner
0	0	0	3	17	0	0	3	0	0	2	0	0	0	0	0	0	0	0	0	e = Monet
0	0	0	0	1	18	0	0	3	0	2	0	0	0	0	1	0	0	0	0	f = VanGogh
1	1	0	0	0	0	20	0	1	0	1	0	0	0	0	0	0	0	0	1	g = Waterhouse
0	0	0	1	0	0	1	21	0	2	0	0	0	0	0	0	0	0	0	0	h = Redon
3	0	0	0	0	2	1	0	17	0	0	0	0	0	1	0	0	0	0	1	i = Rousseau
0	0	0	0	0	0	0	3	0	20	0	0	0	1	0	1	0	0	0	0	j = Gauguin
2	0	0	0	5	2	0	0	2	0	12	1	0	0	0	0	0	1	0	0	k = Klimt
0	0	0	0	0	1	0	0	0	0	1	23	0	0	0	0	0	0	0	0	l = Mucha
1	0	0	0	0	0	0	2	0	3	0	0	15	1	1	1	0	0	0	1	m = Picasso
2	0	0	1	0	0	0	3	0	2	0	0	0	14	0	3	0	0	0	0	n = Dali
0	1	0	1	1	0	0	1	0	0	0	0	0	0	20	1	0	0	0	0	o = Magritte
4	0	0	1	0	1	0	2	0	1	2	0	5	1	0	8	0	0	0	0	p = Kandinsky
0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	22	0	0	0	q = Malevich
0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	21	1	1	r = Pollock
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	24	0	s = Lichtenstein
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	t = Giger

- **Naivni Bayes**

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
12	0	1	0	0	1	6	0	1	1	0	0	1	2	0	0	0	0	0	0	a = Botticelli
0	16	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	b = Caravaggio
0	4	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	c = Rembrandt
0	0	0	17	2	0	2	1	0	0	0	1	0	0	2	0	0	0	0	0	d = Turner
0	0	0	3	12	1	0	2	0	1	3	0	0	1	2	0	0	0	0	0	e = Monet
0	0	0	0	2	10	1	0	3	0	4	3	0	1	0	0	0	1	0	0	f = VanGogh
2	0	0	0	0	0	18	1	1	0	1	0	0	0	0	0	0	0	0	2	g = Waterhouse
0	0	0	1	0	0	0	19	0	2	1	0	0	2	0	0	0	0	0	0	h = Redon
2	0	0	0	0	1	2	0	17	0	1	0	0	0	0	0	0	0	0	2	i = Rousseau
0	0	0	0	0	0	1	4	0	16	0	0	0	1	0	2	0	0	0	1	j = Gauguin
0	0	0	0	4	4	1	0	2	0	10	2	0	0	0	0	0	2	0	0	k = Klimt
0	0	0	0	1	0	0	0	0	0	1	22	0	1	0	0	0	0	0	0	l = Mucha
4	0	0	0	0	0	0	1	1	2	1	0	8	1	1	4	0	0	0	2	m = Picasso
0	0	0	0	0	0	0	2	0	0	0	0	2	17	0	4	0	0	0	0	n = Dali
0	0	0	1	0	0	0	3	0	0	0	0	1	3	16	1	0	0	0	0	o = Magritte
1	0	1	0	0	2	0	1	0	1	1	2	2	2	0	9	0	0	2	1	p = Kandinsky
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	20	0	0	0	q = Malevich
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	22	1	1	r = Pollock
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	23	0	s = Lichtenstein
1	0	0	0	0	0	1	0	2	0	0	0	1	0	0	0	0	0	0	20	t = Giger

- **Slučajne šume**

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
10	0	0	0	0	1	4	1	2	2	1	0	3	0	1	0	0	0	0	0	a = Botticelli
0	23	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	b = Caravaggio
0	3	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = Rembrandt
0	0	0	20	0	0	2	1	0	0	0	0	0	0	1	0	1	0	0	0	d = Turner
0	0	0	3	17	0	0	2	0	0	3	0	0	0	0	0	0	0	0	0	e = Monet
0	0	0	0	1	14	1	0	1	0	3	1	0	0	0	1	0	3	0	0	f = VanGogh
2	0	0	0	0	1	18	1	1	0	0	0	0	0	0	0	0	0	0	2	g = Waterhouse
0	0	1	1	0	0	0	20	0	2	0	0	0	1	0	0	0	0	0	0	h = Redon
2	0	0	0	0	1	1	0	16	0	1	0	1	0	0	0	0	0	0	3	i = Rousseau
0	1	0	0	1	0	0	1	0	18	1	0	1	0	1	1	0	0	0	0	j = Gauguin
1	0	0	0	2	2	1	0	1	1	12	3	0	0	0	1	0	1	0	0	k = Klimt
0	0	0	0	0	0	0	0	0	0	1	23	0	0	0	1	0	0	0	0	l = Mucha
2	0	0	0	0	0	0	0	0	2	1	0	14	0	1	2	1	0	0	2	m = Picasso
2	0	0	0	0	0	0	1	0	0	0	0	1	19	0	2	0	0	0	0	n = Dali
0	0	0	0	0	0	0	2	0	0	0	0	0	1	20	1	1	0	0	0	o = Magritte
1	0	0	0	1	1	0	1	0	2	2	1	3	1	0	10	1	0	0	1	p = Kandinsky
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	q = Malevich
0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	21	0	2	r = Pollock
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	s = Lichtenstein
1	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	21	t = Giger

7. Literatura

[1] *Olga's Gallery*, 1999.-2011., http://www.abcgallery.com/

[2] Nicolas Pioch, *WebMuseum*, 14.10.2002, http://www.ibiblio.org/wm/paint/

[3] *Wikipaintings – Visual Arst Encyclopedia*, www.wikipaintings.org/

[4] *MATLAB® CENTRAL*- http://www.mathworks.com/matlabcentral/

[5] *WEKA The University of Waikato* - http://www.mathworks.com/matlabcentral/

[6] I. Baskin, A. Varnek *Tutorial on Classification*

[7] *Otkrivanje znanja dubinskom analizom podataka - Priručnik za istraživače i studente*, Institut Ruđer Bošković

[8] Otkrivanje znanja u skupovima podataka – nastavni materijal: http://www.zemris.fer.hr/predmeti/kdisc/ref1.html

[9] J. Zujovic, L. Gandy, S. Friedman *Using Neural Networks to Classify Paintings by Genre*, Northwestern University

[10] M. Čuljak, K. Jež, B. Mikuš, S. Hadjić *Classification of art paintings by genre*, MIPRO 2011, May 23-27, 2011, Opatija, Croatia

[11] Thomas Lombardi, Sung-Hyuk Cha, Charles Tappert, *A Graphical User Interface for a Fine-Art Painting Image Retrieval System*

[12] Thomas Lombardi, *The Classification of Style in Fine-Art Painting*, Pace University

[13] Shiyu Luo, *Oil Painting Classification*, (2010.)

[14] Oguz Icoglu, Bilge Gunsel, Sanem Sariel, *Classification and indexing of paintings based on art movements*, Multimedia Signal Processing and Pattern Recognition Lab. Electrical-Electronics Eng. Faculty. Istanbul Technical University