Analiza şi descrierea perceptuală a artei vizuale româneşti

Raport cercetare 2015

Cuprins

În acest raport vom sumariza eforturile noastre în relizarea scopurilor propuse prin proiectul de față. Din punct de vedere al obiectivelor abordăm două probleme distincte: recunoașterea curentului artistic și recunoașterea clasei de compuși chimici dintr-un pigment colorat. Din punct de vedere tehnic ambele sunt tratate ca o problemă de clasificare/regresie. Astfel prezentarea inițială teoretică se axează

pe detalierea conceptelor folosite atât în partea de descriere a conținutului cât și în partea de clasificare, Mai departe vom continua cu o descriere a bazelor de date folosite, pentru a încheia cu rezultate obținute până la momentul depunerii raportului.

1. Descrierea conținutului și clasificare

1.1. Descriptori de conținut

1.1.1 Histograma de trăsături topografice

Histogramele de trăsături topografice, ca descriptor de imagine au fost introduse de Florea et al. [Florea14] pentru descrierea feței. Construcția descriptorului pornește de la dezvoltarea în serie Taylor imaginii văzută ca o funcţie bidimensională:

$$I(i+\Delta_i, j+\Delta_j) \approx I(i,j) + \nabla I \cdot [\Delta_i \ \Delta_j] + \frac{1}{2}[\Delta_i \ \Delta_j]\mathfrak{H}(i,j)\begin{bmatrix}\Delta_i \\ \Delta_j\end{bmatrix}$$

unde I(i,j) este imaginea cu un singur plan de culoare (e.g. cu niveluri de gri) în punctul (i,j), ∇I este gradientul direcțional al imaginii, iar $\mathfrak{H}(i,j)$ este matricea 2x2 Hessiană a imaginii în aceeași locație.

Pentru calculul eficient al derivatelor imaginii se folosește spațiul scalelor [Frangi98]. Procedura de construcție a spațiului scalelor ține cont de răspunsurile produse de aplicarea unui nucleu local variabil în funcție de scala σ, urmând ca scala caracteristică a unei regiuni să fie aleasă în maximul din acest spațiu. Lindeberg [Lindeberg13] a propus construirea spațiului scalelor utilizând un nucleu Gaussian G(i,j,σ) și derivatele sale. Metoda constă în alegerea scalei caracteristice a regiunii pentru care o funcție dată atinge extremul în spațiul scalelor.

Imaginea în spațiul scalelor este obținută prin filtrarea cu nuclee Gaussiene de varianțe crescătoare:

$$L(i,j,\sigma) = G(i,j,\sigma) * I(i,j)$$

, unde * reprezintă operația de convoluție, G(i,j,σ) este o Gaussiană simetrică la rotație cu un nucleu de varianță σ^2, care este denumit, în acest caz, parametrul de scală, conform ecuației:

$$G(i,j,\sigma) = \frac{1}{2\sigma^2} \, e^{-(i^2+j^2)/2\sigma^2}$$

Derivata imaginii este calculată ca fiind convoluția cu derivata nucleului Gaussian:

$$\frac{\partial}{\partial i} L(i,j,\sigma) = \sigma I(i,j) \cdot \frac{\partial}{\partial i} G(i,j,\sigma)$$

Matricea Hessiană este calculată astfel:

$$\mathfrak{H}(i,j) = \begin{pmatrix} L_{ii}(i,j,\sigma) & L_{ij}(i,j,\sigma) \\ L_{ji}(i,j,\sigma) & L_{jj}(i,j,\sigma) \end{pmatrix}$$

Considerând că toți pixelii unei regiuni conţin informaţie topografică importantă ce poate fi utilizată în histograme de orientare sau în histograme de magnitudine normalizate, s-a dezvoltat metoda HoT. Pentru o regiune de interes Ω, descriptorii HoT includ:

- Date de ordinul doi, obținute din matricea Hessiană:
  - Histograma de voturi nete a orientării curburii suprafeţei imaginii. Pentru fiecare pixel din Ω, se adaugă "1" la orientarea vârfului/văii extrase prin calcularea unghiului primului vector propriu al Hessianei, dacă a doua valoare proprie este mai mare decât un prag, $\lambda_2 > T_\lambda$.

$$H_1^H([\theta]) = \frac{1}{Z_1} \sum_{(i,j)\in\Omega} (\theta_\lambda(i,j) == [\theta]) \bullet (\lambda_2(i,j) > T_\lambda)$$

  - Histograma de votare ponderată a orientărilor vârfurilor: valorile ei se vor incrementa cu diferenţa în modul dintre valorile proprii ale Hessianei.

$$H_2^H([\theta]) = \frac{1}{Z_2} \sum_{(i,j)\in\Omega} (\theta_\lambda(i,j) == [\theta]) \bullet \left(\lambda_2(i,j) - \lambda_1(i,j)\right)$$

    Histogramele H_1^H și H_2^H creează, fiecare, un vector de lungime egală cu numărul de bini de orientare şi descriu tăria curburii suprafeței date de pixelii imaginii.
  - Histograma de gamă a celor mai mici valori proprii, având un interval predefinit (spre exemplu *[0, $M_{\lambda 2}$ = 30]*).

$$H_3^H(k) = \frac{1}{Z_3} \sum_{(i,j)\in\Omega} \left(\lambda_1(i,j) \in \left[(k-1)\frac{M_{\lambda_2}}{N_{bin}}; k\frac{M_{\lambda_2}}{N_{bin}}\right]\right)$$

  - Histograma de gamă a diferenţelor dintre valorile proprii având un interval predefinit (spre exemplu *[0, $M_{\lambda 12}$ = 50]*).

$$H_4^H(k) = \frac{1}{Z_4} \sum_{(i,j)\in\Omega} \left((\lambda_2(i,j) - \lambda_1(i,j)) \in \left[(k-1)\frac{M_{\lambda_2}}{N_{bin}}; k\frac{M_{\lambda_2}}{N_{bin}}\right]\right)$$

- Date de ordinul unu (extrase din gradient):
  - Histograma de orientare, H_1^G , introdusă anterior de Dalal și Triggs [Dalal05]; pentru fiecare pixel ce are un gradient mai mare de un anume prag T_G se va adaugă un vot;
  - Histograma de magnitudine a gradientului, H_2^G. Magnitudile au valorile între 0 şi o valoare maximă, spre exemplu 100.

1.1.2 Descrierea conținutului color. Spații de culoare

Spațiul de culoare de bază în care sunt în general reprezentate imaginile și care de altfel este și spațiul în care se face achiziția imaginilor fotografice este spațiul RGB (Red, Green, Blue - Roșu, Verde, Albastru). Dar în funcție de aplicație acest spațiu ar putea să nu fie suficient și din acest motiv au apărut și alte spații de culoare obținute prin combinații liniare sau neliniare ale planurilor inițiale RGB. În cele ce urmează vom face o prezentare foarte scurtă a spațiilor de culoare folosite în cele două aplicații.

Spațiul RGB. Primul sistem propus pentru reprezentarea culorilor este sistemul RGB (Red, Green, Blue - Roșu, Verde, Albastru). În acest sistem funcțiile de potrivire a culorilor relativ la lungimea de undă a luminii (funcții de transfer) sunt asociate culorilor primare monocromatice de lungimi de undă de

700nm, 546.1nm și respectiv 435.8nm. Aceste valori au fost alese astfel încât valorile asociate luminii albe să fie egale. Totuși pentru a forma unele culori sunt necesare și valori negative. În reprezentarea culorilor în RGB se folosesc numai valori pozitive ceea ce înseamnă că există culori percepute de ochiul uman pe care acest spațiu de culoare nu le poate reproduce. Culorile fizic realizabile prin sumarea celor trei valori pozitive ale funcțiilor r(λ), g(λ) și b(λ) formează un cub numit cubul RGB reprezentat în figura 1.1. În colțurile cubului sunt culorile primare și cele secundare: roșu, verde, albastru, galben, turcoaz și mov, precum și alb și negru. Diagonala dintre colțurile alb-negru conține niveluri de gri.

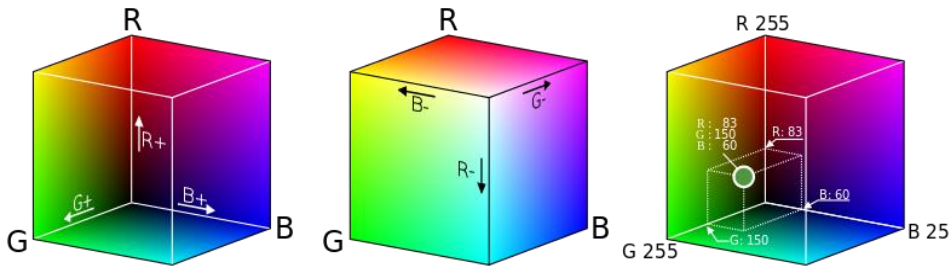


Figura 1.1 Cubul RGB: Gama culorilor reprezentabile prin sumarea unor valori pozitive în spațiul RGB (imagine preluată de la https://commons.wikimedia.org/wiki/File:RGB_color_cube.svg)

Pentru că este spațiul de culoare cel mai ușor de realizat din punct de vedere fizic, spațiul RGB este și cel mai folosit. Totuși el are unele neajunsuri și din acest motiv în aplicații specifice s-au construit și alte spații de culoare.

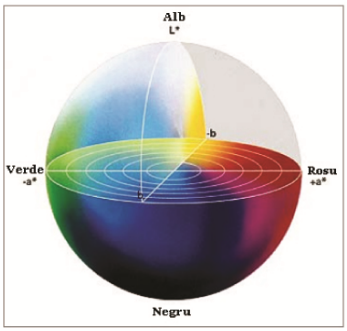
Spațiul XYZ Așa cum am discutat în spațiul RGB ar trebui să existe și valori negative pentru a putea reprezenta toate culorile perceptibile de către sistemul vizual uman. Din acest motiv s-a definit spațiul XYZ. Transformarea din spațiul RGB în spațiul XYZ este o transformare liniară obținută prin setul de ecuații:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Spațiul Lab Distanțele euclidiene între culori reprezentabile în spațiul RGB nu corespund distanțelor percepute de un observator uman dintre culorile respective. Acest lucru pornește de la faptul că în mecanismul percepției umane există neliniarități. Pentru a rezolva această problemă s-a plecat de la studiul neliniarității sistemului vizual uman prin elipsele McAdams. S-a introdus noțiunea de diferențe abia perceptibile între culori (culori între care un observator uman nu poate vedea diferența) și aceste culori se reprezintă grafic. Într-un spațiu perceptual uniform aceste culori trebuie să formeze niște cercuri. Plecând de la această observație s-a căutat modul în care se poate trece de la un spațiu perceptual neuniform cum este spațiul RGB la unul uniform. Astfel spațiul Lab este obținut printr-o transformare neliniară a spațiului XYZ.

Culorile reprezentabile cu ajutorul spațiului de culori Lab formează o sferă. Aceasta se poate vedea în figura 1.2. Axa L este asociată luminanței culorii, iar axele a și b sunt axele asociate crominanței. Cele trei axe replică percepția umană. Astfel axa de luminanță este neliniară, iar axele de crominanță sunt conform principiului culorilor opuse care spune ca o culoare nu poate fi simultan roșu și verde, respectiv galben și albastru.

Figura 1.2 Sfera Lab: Gama culorilor reprezentabile în spațiul Lab.



Transformările XYZ-Lab sunt:

$$L = 116 f\left(\frac{Y}{Y_n}\right) - 16 \quad a = 500 f\left(\frac{X}{X_n} - \frac{Y}{Y_n}\right) \quad b = 200 f\left(\frac{Y}{Y_n} - \frac{Z}{Z_n}\right)$$ Unde

$[X_n, Y_n, Z_n]$ este setul tristimul al iluminantului, iar funcția neliniară f este:

$$f = \begin{cases} t^{\frac{1}{3}}, 0.008856 \le t \le 1 \\ 7.787t + \frac{16}{116}, 0 \le t \le 0.008856 \end{cases}$$

Spațiul HSV. Alte spații de culoare extrem de utilizate sunt spațiile din familia HSV. Aceste spații sunt construite astfel încât să fie descrise cât mai aproapre de modul intuitiv, natural de descriere verbală a culorilor. Astfel fiecare axă are o semnificație specifică:

1. *Nuanța* (Hue – H): spune cu ce fel de culoare avem de-aface. S-a constatat experimental că există culori pe care oamenii le definesc în mod unic.
2. *Puritatea* (Saturation - S): spune cât de pură, cât de intensă este culoarea. Acest tip de descriere este dată de axa purității. O culoare cu puritate minimă devine nivel de gri.
3. *Luminozitatea* (Value - V): spune cât de luminoasă sau de întunecată este culoarea. Orice culoare este o combinație de culoare pură cu o cantitate mai mică sau mai mare de alb sau negru. Această cantitate de alb sau negru, dă luminozitatea culorii.

Pentru a trece din spațiul RGB într-un spațiu în care axele să aiba semnificațiile menționate este necesară o transformare neliniară. Ecuațiile de transformare nu sunt unanim acceptate, diferind în funcție de sursă dar păstrând conceptul. În lucrarea de față am folosit ecuațiile următoare:

$$V = \max(R, G, B)$$

$$S = \begin{cases} 0, V = 0 \\ \frac{V - \min(R, G, B)}{V}, V > 0 \end{cases}$$

$$H = \begin{cases} 0, S = 0 \\ \frac{1}{6}\left(\frac{G - B}{SV}\right), V = R \\ \frac{1}{6}\left(2 + \frac{B - R}{SV}\right), V = G \\ \frac{1}{6}\left(4 + \frac{R - G}{SV}\right), V = B \\ H + 360, H < 0 \end{cases}$$

Aceste ecuații corespund modelului Hexcone descris în Smith78 [Smith78].

1.2. Sisteme de clasificare

Pasul următor descrierii imaginii este de folosire a unui clasificator care să asocieze anumite valori particulare unei clase. Teste extinse am efectuat bazându-ne pe clasificator de tip random forest, cel mai apropiat vecin, mașină cu vectori suport și rețea convoluțională adâncă.

1.2.1 Ansambluri de arbori de clasificare

Arborii de clasificare **Error! Reference source not found.** sunt instrumente uzuale pentru determinarea unei partiții într-un spațiu de dimensiuni conformă unui set oferit de etichete. Ideea lor de bază este de a diviza recursiv problema în două probleme, mai simple, fiecare rezolvabile de un model de complexitate redusă. Diviziunile sunt efectuate în nodurile interne ale arborelui, pe baza unor teste binare specifice problemei. Nodurile terminale conțin eticheta care aproximează rezultatul dorit. Cea mai folosită funcție obiectiv utilizată pentru a construi arbori este indexul Gini. În practică, procesul de construcție al arborelui este o problemă de învățare supervizată, bazat pe perechi dată-etichetă.

Testele binare de la nodurile interne ale unui arbore se bazează pe compararea unei dimensiuni specifice al descriptorului. Formal, testul în nodul N_k este:

$$\begin{cases} 0 & , R(i) \le t_{N_k} \\ 1 & , R(i) > t_{N_k} \end{cases}$$

unde t_{N_k} este pragul nodului și $R(i)$ este valoarea celei de a i dimensiuni a trăsăturii. Dimensiunea i folosită în test și valoarea pragului se determină în procesul de antrenare.

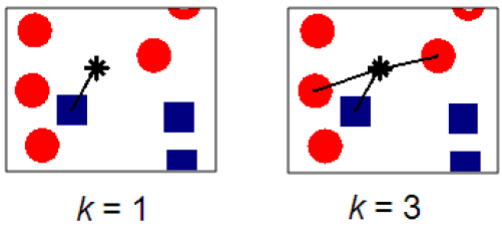
Este bine cunoscut faptul că un singur arbore va supraînvăța datele de antrenament. Pe de altă parte, un ansamblu de arbori poate obține rezultate impresionante. Cele mai comune soluții pentru a combina mai mulți arbori sunt „păduri aleatoare” [Breiman01] sau prin augumentarea gradientului [Friedman01]. Prima variantă presupune creșterea arborilor independent și simultan, iar valoarea prezisă de ansamblu este agregarea rezultatului arborilor individuali (prin majoritate). Varianta a doua presupune creșterea arborilor în mod secvențial. Fiecare arbore adăugat la ansamblul este învățat să reducă eroarea anteriorilor. Ambele metode au ca parametri adâncimea maximă a fiecărui copac, d, și numărul de arbori într-un ansamblu, T.

Unul dintre motivele din spatele succesului ansambluri de arbori este aleatorismul prezent în procesul de învățare. Dezordinea este injectată în construcția fiecărui arbore prin procesul de eșantionare de tip bootstrap și prin alegerea la întâmplare a caracteristicilor folosite drept candidați investigați pentru fiecare nod intern.

1.2.2 Cel mai apropiat vecin

Metoda de clasificare bazată pe regula „Cel mai apropiat vecin" (Nearest neighbor - NN) este una dintre cele mai populare metode de clasificare non-parametrice. Este foarte simplă, intuitivă și precisă și este utilizată într-o mare varietate de aplicații din lumea reală. Principul de bază este să se atribuie unui exemplu de clasificat eticheta celui mai apropiat vecin din setul de referință. „Apropiat" are sensul de distanța cea mai mică sau similaritatea cea mai mare.

Figura 1.3 Funcționarea algoritmului cel mai apropiat vecin (k = 1) și cei mai apropiați k = 3 vecini.



O variantă îmbunătățită constă în algoritmul "cei mai apropiați *k* vecini" (k-NN), care este exemplificat în figura 1.3 partea dreaptă. Și acest algoritm are o fază de antrenament și una de clasificare propriu-zisă. Exemplele de antrenament sunt vectori într-un spațiu de caracteristici multidimensionale, fiecare cu o etichetă de clasă. Faza de antrenare a algoritmului constă din stocare vectorilor de trăsături și a etichetelor de clasă ale probelor de antrenare. În faza de clasificare, se definește k drept o constantă iar un vector neetichetat este clasificat prin atribuirea etichetei care este cea mai frecventă între cele mai apropiate k exemple de antrenament.

De obicei, distanța *euclidiană* este folosită ca măsură de similaritate, însă aceasta se aplică numai la variabile cu valori continue. În cazuri cu varibile cu valori discrete (e.g. valori de etichete) pot fi folosite alte metrici, cum ar fi distanță Hamming. Adesea, precizia de clasificare a unui algoritm de tip k-NN poate fi îmbunătățită semnificativ dacă metrica de distanță este antrenată cu algoritmi de specialitate.

Un dezavantaj de bază al algoritmilor de tip *k*-NN este că clasele cu cele mai frecvente exemple au tendința de a domina clasificarea noului vector, deoarece acestea cel mai probabil se află în cei mai apropiați k vecinii datorită numărului lor mare. O modalitate de a depăși această problemă este să se pondereze clasificarea, luând în considerare distanța de la exemplul nou pentru fiecare dintre cei mai apropiați *k* vecinii ai săi.

1.2.3 Mașină cu vectori suport

O mașină cu vectori suport (SVM)[Cortes95] este un clasificator discriminativ definit formal de către un hiperplan de separație. Cu alte cuvinte, fiind date datele de antrenare (care conțin perechi exemplu-etichetă), algoritmul generează un hiperplan optim care clasifică noile exemple.

O alegere a hiperplanului de separație care trece prea aproape de punctele din setul de antrenament este nepotrivită deoarece clasificatorul rezultat va fi sensibil la zgomot și nu va generaliza corect. Prin urmare, scopul este de a găsi hiperplanul care trece pe cât posibil, cât mai departe de toate punctele. Funcționarea algoritmului SVM este bazată pe găsirea hiperplanului care maximizează distanța la exemplele de antrenare.

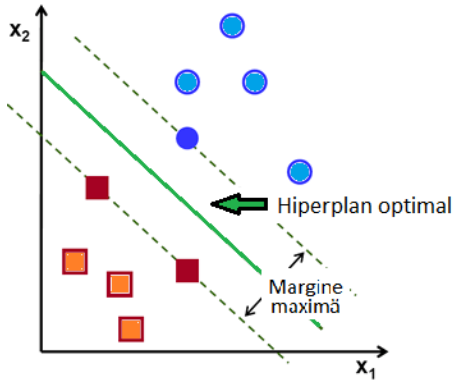


Figura 1.4 Hiperplanul optimal și marginea SVM-ului

Hiperplanul optim poate fi reprezentat într-un multitudine de moduri dar, convențional, cel ales este:

$$|\beta_0 + \beta^T x| = 1$$

unde *x* reprezintă exemplele din setul de antrenament cele mai apropiate de hiperplan (așa numiții vectori suport). Această reprezentare este denumită hiperplanul canonic. Distanța de la un vector suport *x* la hiperplanul canonic este:

$$d = \frac{|\beta_0 + \beta^T x|}{\|\beta\|} = \frac{1}{\|\beta\|}$$

Marginea maximală este la jumătatea distanței față de vectorii suport, deci este de două ori distanța. În final problema maximizării marginii M, este echivalentă cu problema minimizării unei funcționale L(β) cu constrângeri. Constrângerea impune ca hiperplanul să clasifice corect toate exemplele x_i. Formal:

$$L(\beta) = \frac{1}{2}\|\beta\|^2 \; s.t. \; y_i(\beta^T x_i + \beta_0), \forall i$$

unde y_i reprezintă etichetele exemplelor din setul de antrenament. Problema de minimizat este una tipică, rezolvabilă prin optimizare Lagrangiană. Totuși singura metodă cu convergență garantată este algorimul SMO [Platt98]

1.3. Sisteme mixte. Rețele convoluționale adânci

În ultima vreme s-a arătat că foarte multe competiții de clasificare au fost câștigate detașat de rețele neuronale (convoluționale) adânci. În continuare vom prezenta câteva din caracteristicile acestui tip de sisteme, care, tradițional, incorporează și pași pentru descrierea imaginilor de intrare.

Modelele computaționale de rețele neuronale au existat de mai mult de jumătate de secol, începând cu cel mai simplu model dezvoltat de McCulloch și Pitts în 1943 [McCulloch1943], împreună cu algoritmul de învățare dezvoltat de Hebb pentru un astfel de model [Hebb1949], bazat pe regula Delta (cuanta de ajustare a unei ponderi este direct proporțională cu produsul între intrare și ieșirea dorită).

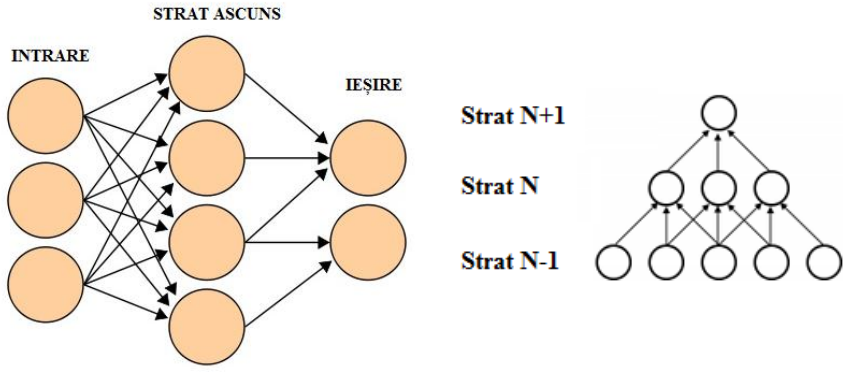
1.3.1 Rețele neuronale cu reacție pozitivă

Algoritmul de propagare inversă se realizează printr-o rețea neuronală multistrat, cu reacție pozitivă. O astfel de rețea poate fi definită de activarea neuronilor și de puterea conexiunilor dintre fiecare pereche de neuroni (un exemplu este în figura 1.5 a).

Deoarece rețeaua are o reacție pozitivă, straturile de neuroni sunt conectate între ele unidirecțional, aciclic, astfel încât activarea rețelelor are puncte clare de pornire și de oprire (un stat de intrare și un stat de ieșire). Staturile dintre aceste extreme sunt denumite straturi ascunse.

Fluxul de activare în aceste rețele este realizat printr-un proces de însumare ponderată. Fiecare neuron își trimite activarea curentă la orice alt neuron cu care este conectat. Această activare este înmulțită cu ponderea conexiunii dintre cei doi neuroni și trecută printr-o funcție de comprimare. Tradițional funcția de comprimare era de tip sigmoidal, dar în ultima perioadă s-a arătat că funcția de activare de tip ReLU (Rectified Linear Unit) [LeCunn2015] conduce la rezultate superioare, în special în ce privește timpul de antrenare, care este de câteva ori mai mic. Dacă procesul folosit ar fi pur liniar, atunci straturile suplimentare ar fi inutile, deoarece adăugând două combinații liniare rezultatul este tot o combinație liniară.

Presupunând o conexiune completă între straturi consecutive (fiecare neuron dintr-un strat este conectat la fiecare neuron din stratul următor), calculele se pot face prin înmulțirea vectorului de activări cu matricea ponderilor și apoi trecerea rezultatelor prin funcția neliniară (pasul de propagare directă).



(a)

Figura 1.5. (a) Rețea tipică cu reacție pozitivă folosită pentru propagarea inversă
(b) Reprezentare 1D a procesului de convoluție cu un filtru 1x3

Învățarea în aceste rețele se produce prin modificarea ponderilor conexiunilor, astfel încât să minimizeze o funcție de eroare, de obicei specificată ca diferență dintre vectorul de activare al stratului de ieșire și vectorul de activare dorită. Minimizarea erorii se realizează treptat prin algoritmul de propagare inversă. Acesta presupune calculul derivatei parțiale a erorii față de ultimul strat de ponderi și folosirea acestei informații pentru actualizarea ponderilor. Apoi derivatele parțiale pot fi calculate pentru penultimul strat de ponderi și procesul se repetă recursiv până se ajunge la actualizarea primului strat de ponderi.

Deși în teorie aceste rețele sunt un aproximator universal pentru orice fel de funcție, în practică există mai multe probleme cărora nu le pot face față. O primă problemă este descrierea și recunoașterea obiectelor prezentate vizual. Deoarece fiecare neuron dintr-un strat este conectat la fiecare neuron din stratul următor, numărul de ponderi crește foarte rapid cu dimensionalitatea datelor de intrare rezultând într-un proces de învățare foarte încet în cazul datelor vizuale de mari dimensiuni.

O problemă mai gravă a acestor rețele este lipsa de informație spațială. Deoarece fiecare pereche de neuroni dintre două straturi au propria lor pondere, învățarea pentru a recunoaște un obiect într-o

locație nu este transferabilă la același obiect prezent într-o altă locație; în acest caz sunt necesare noi ponderi și deci o nouă învățare.

Este deci necesară o arhitectură ce beneficiază de pe urma constrângerilor spațiale existente în intrările vizuale, și care, în același timp, reduce numărul de parametri implicați în antrenare. O astfel de arhitectură o au rețele neuronale convoluționale.

1.3.2 Rețele neuronale convoluționale

Soluția la problemele rețelelor anterioare legate de procesarea imaginilor, a fost inspirată din neurobiologie. LeCun și Bengio au încercat să modeleze organizarea neuronilor din cortexul vizual, care la momentul respectiv se știa că are hărți de câmpuri receptive locale, care scad în granularitate cu cât te muți mai în față în cortex.

Există mai multe teorii despre cum se definește precis un model convoluțional, dar toate diferitele implementări implică următorul proces:

- Convoluția a mai multor filtre mici pe imaginea de intrare,
- Subeșantionarea spațiului rezultat din convoluție,
- Repetarea pașilor 1 și 2 până râmane un număr suficient de trăsături de nivel înalt,
- Folosirea unei rețele neurale standard (cu reație pozitivă) pentru a rezolva problema, folosind trăsăturile rezultate anterior ca strat de intrare.

1.3.3. Structura de tip LeNet

Una dintre cele mai cunoscute rețele neuronale convoluționale este cea propusă de LeCun [LeCunn1998], LeNet. Ea cuprinde mai multe straturi specifice.

Convoluție în contextul acestei rețele, înseamnă să se aplice pe imagine un filtru reprezentat de un strat de ponderi, având dimensiuni tipic mici (3x3, 5x5, 7x7, etc), iar ieșirea contând într-o singură unitate. Deoarece acest filtru este aplicat în mod repetat la deplasamente consecutive în imagine, conectivitatea rezultantă arată ca o serie de câmpuri receptive suprapuse, ce se organizează într-o nouă matrice la ieșirea din filtru (sau în mai multe matrice, în general folosindu-se mai multe straturi de astfel de filtre – ilustrate în figura 1.5 b).

Un avantaj important al acestei structuri este dat de faptul că, deși modelul păstrează un număr mare de conexiuni între stratul de intrare și statul de ieșire al filtrării, ponderile sunt corelate. Din acest motiv, în procesul de propagare inversă este necesară ajustarea unui număr foarte redus de parametri, corespunzători unei singure instanțe a filtrului.

Un al doilea avantaj al filtrării se datorează posibilității de a aplica acest mecanism oricărei structuri spațiale, nu doar o imagine de intrare. Astfel, la ieșirea filtrării inițiale se pot adăuga straturi suplimentare de filtre. Totuși, deoarece dimensionalitatea după aplicarea unui strat de filtrare este egală cu dimensionalitatea intrării (ignorând marginile), adăugarea de filtre suplimentare nu ar îmbunătăți invarianța la translație, ci doar s-ar efectua o analiză pixel la pixel asupra unor trăsături din ce în ce mai abstracte. Pentru a rezolva această problemă s-a introdus un nou tip de strat, stratul de subeșantionare.

Subeșantionarea (pooling) se referă la reducerea dimensiunii generale a unui semnal. Tipic se implementează ca maximul dintr-o vecinătate spațială predefinită. În domeniul filtrării 2D, subeșantionarea ajută în plus la creșterea invarianței la poziția spațială.

Metoda de subeșantionarea specifică arhitecturii LeNet este punerea în comun a maximului ("max-pooling"). Această tehnică presupune divizarea matricei de ieșire a convoluției într-o grilă de dreptunghiuri care nu se suprapun și selectarea valorii maxime din fiecare dreptunghi pentru a obține o matrice redusă (cu cât dreptunghiurile sunt mai mari, cu atât se reduce mai mult dimensionalitatea). Aplicând astfel de straturi de subeșantionare între straturile de convoluție se poate îmbunătăți abstractizarea trăsăturilor, concomitent cu creșterea invarianței spațiale.

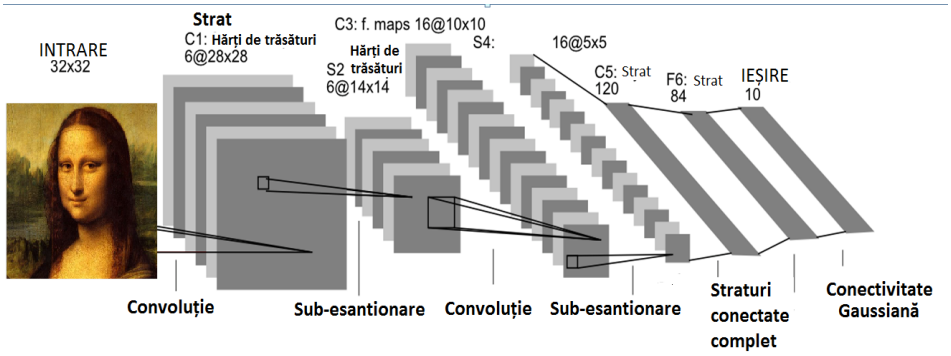


Figura 1.6. Arhitectura LeNet standard. Notații: *Cx* (strat convoluțional), *Sx* (strat de subeșantionare), *Fx* (strat complet conectate), *x* = poziția în lanțul de straturi. Imagine preluată după [LeCun1998]

Deși structura generală a acestui model este stabilită, există mulți parametri variabili ce pot fi adaptați la diferite aplicații, precum numărul de filtre de convoluție utilizate, dimensiunea filtrelor și dimensiunea subeșantionării.

După cum se poate observa în Figura 1.6, arhitectura LeNet conține multe straturi, fiecare având propriile ponderi ce trebuiesc antrenate. Intrarea este o imagine de 32x32 pixeli și, pentru a accelera antrenarea, este nomalizată astfel încât să aibă media 0 și varianța 1.

2. Recunoașterea curentului artistic

2.1. Rezultate relevante din literatură

În domeniul analizei tablourilor **recunoaşterea curentului artistic** este o subtemă dificilă, chiar şi pentru specialişti datorită variaţiei existente într-un curent. Dacă soluţiile iniţiale se bazau pe caracteristici simple, primare (cum ar fi nivelul mediu de luminanţă sau culoare) [Gunsel05], în ultima perioadă s-a efectuat o tranziţie către caracteristici elaborate cum ar fi cele din lucrările lui Shamir et al. [Shamir10], Arora şi ElGammal [Arora12], Karayev et al. [Karayev13] sau Condorovici et al. [Condorovici13], [Condorovici14] sau Saleh et al. [Saleh15].

Recent, odată cu renașterea rețelelor neurale adânci se constată o multitudine de încercări care dezvoltă această direcție pentru investigarea tablourilor artistice. În acest sens notăm soluțiile propuse de către Gatys et al. [Gatys15] care încearcă să recreeze pe baza rețelelor neuronale artificial percepția umană a stilului artistic, de către Lu et al. [Lu15] care folosesc rețele neuronale adânci pentru a trata unitar problema descrierii de trăsături și a clasificării în vederea estimării calității estetice a unei lucrări.

Puțin înainte, Bar et al. [Bar14] concluzionează că pentru clasificarea corectă a stilului unui tablou, în contextul unei baze foarte mari de date, extractori similari cu stratul de filtrare și cel de pooling (sub-eșantionare) dintr-o rețea convoluțională conduc la cele mai bune performanțe.

În concluzie, în ultima perioadă capătă accent și interes metodele bazate pe rețele convoluționale adânci. Din acest motiv în continuare vom descrie principale caracteristici tehnice ale acestora, urmând ca mai departe să le utilizăm pentru inferarea curentului artistic.

2.2. Baze de date

Problematica. În problema recunoașterii curentului artistic, un aspect critic este baza de date. Din punct de vedere istoric și în această direcție s-a început cu baze de date mici și cu rezultate nerealist de mari. În ultima perioadă s-a făcut tranziția la baze de date mai mari, în care generalizarea rezultatelor nu mai este limitată. Astfel baze de date mai mari sunt folosite în [Condorovici14] (aproximativ 4000 tablouri) sau [Khan14] (4266 tablouri). Chiar şi aceaste baze de date sunt inferioare volumului de lucrări relevante la nivel mondial întrucât, de exemplu, doar Luvrul are peste 7500 de picturi. O notă discordantă face lucrarea lui Bar et al. [Bar14] ce testează un sistem de recunoaştere a curentului pe o scară foarte mare, cu 40724 imagini din Wikiart (www.wikiart.org), dar baza lor de date nu este diponibilă.

În concluzie, pentru a obține rezultate relevante este necesar să folosim și noi o bază de date de dimensiuni mai mari.

2.2.2. Baza de date proprie

Baza de date de tablouri internaționale. Una dintre cele mai mari provocări în evaluarea metodelor de clasificare automată a tablourilor este baza de date standard [Cornelis11]. Soluția tipică folosită este de colectare a unei colecții de picturi de pe internet. În acest sens am pornit de la baza de date folosită anterior de noi [Condorovici14] și am completat-o în special pentru curentele de artă modernă.

În total a rezultat o bază de date care conține 6119 de picturi aparținând la opt mișcări sau curente artistice diferite, de la 826 de autori cunoscuți plus aproape 1000 de opere cu autori necunoscuți. Structura bazei de date este prezentată în tabelul 2.1. Cele opt mișcări de artă studiate sunt: renascentist, baroc, rococo, romantism, impresionism, iconoclastic, arta greaca veche și cubism. Genurile au fost alese astfel încât să includă cazuri tipice de clase foarte separabile (genuri care sunt foarte diferite și ușor de a discerne, cum ar fi cubism versus renascentism) și clase mixte (genuri care sunt foarte similare, greu de separat, cum ar fi baroc și renascentist). Imaginile au fost adunate din diverse surse de internet, condițiile de achiziție și rezoluțiile de imagine sau de calitate putând varia în limite extreme, de la achiziție profesionist controlată, la capturarea imaginii de către amatori. Pentru a evita problemele de normalizare legate de dimensiunea imaginii, toate imaginile au fost scalate la rezoluție de 0,3 megapixeli.

Tabelul 2.1. Structura bazei de tablouri colectate

Curent	Structură baza de date	
	[Condorovici14]	Actuală
Baroc	731	1000
Cubism	575	1000

Renascentist	485	1000
Iconoclastic	625	625
Impresionism	789	1000
Arta veche greacă	332	332
Rococo	292	869
Romantism	290	900

2.2.2. Baza de date Painting 91

Baza de date Painting – 91 a fost introdusă în [Khan14] și cuprinde lucrările digitizate a 91 de autori. Lucrările acelora dintre ei care sunt catalogați clar într-un curent artistic au format un subset cu etichete de curent artistic. Pentu acest scop creatorii bazei de date au construit un set de antrenament și unul de test astfel încât să fie relativ ușor soluțiilor noi să se compare cu variantele de bază oferite în lucrarea introductivă.

2.3. Rezultate obținute

Baza de date proprie. Pe această bază de date am experimentat următoarele variante

- Retea convoluțională adâncă de tip LeNet. În cadrul experimentului am încercat două variante: imaginile eu fost aduse la dimensiunea standard a rețelei (32x32) și dintr-o pictură a rezultat o singură dată de intrare sau am adus imaginile la dimensiune 96x96 si au rezultat 3x3 = 9 date dintr-o pictură. În urma antrenării rețelei rezultatele obținute au fost de 46% clasificare corectă pentru cazul imaginilor de 32x32 și respectiv de 48% în cel de-al doilea caz. Rezultatele sunt inferioare metodei din [Condorovici14] unde acuratețea obținută a fost de 68%. Problema mare este dată de faptul că nu există suficiente exemple de antrenare. O altă concluzie rezultantă este că varianta holistică funcționează mai precis.
- Descriptor HoT (calculat la trei scale: σ=1, σ=2.5, σ=5) și clasificator random forest sau mașină cu vectori suport. În acest caz, am urmat procedura „10-fold cross validation”.
- Descriptor color obținut prin concatenarea histogramelor calculate independent pe planul de L, a și respectiv b.

Rezultatele sunt în tabelul 2.2 de mai jos. Pentru varianta cea mai bună (HoT+SVM) am experimentat cu mai multe alternative ale descriptorului. Acestea sunt prezentate în tabelul 2.3.

Tabelul 2.2. Acuratețea recunoașterii curentului artistic pe baza de date proprie cu tablouri internaționale

Descriptor	**HoT**		**Histogramă Lab**		CNN
Clasificator	RF	SVM	RF	SVM	
Rată de recunoaștere corectă	52.1%	54.5%	34.4%	28.9%	48.0%

Tabelul 2.3. Acuratețea recunoașterii curentului artistic pe baza de date proprie cu tablouri internaționale folosind diferite variante ale descriptorului HoT și clasificator de tip SVM.

Descriptor	HoT - σ=1	HoT - σ=2.5	HoT - σ=5	HoT – (σ=2.5+σ=5)	HoT – (σ=1 + σ=2.5 + σ=5)
Recunoaștere	49.24	54.50	55.47	58.48	57.35

Baza de date Paintings -91.

Am implementat aceleași variante ca și în cazul bazei de date proprii, cu excepția soluției bazate pe rețele convoluționale adânci, care anterior s-a dovedit, cel puțin pentru moment neperformantă.

Rezultatele obținute sunt în tabelul 2.4. Pentru comparație se prezintă și câteva din rezultatele raportate în [Khan14], marcate cu roșu

Tabelul 2.4. Acuratețea recunoașterii curentului artistic pe baza de date Paintings-91 folosind diferite variante ale descriptorului HoT și clasificator de tip SVM.
Cu roșu sunt marcate rezultatele raportate de creatorii bazei de date

Tip descriptor	**Structură**					**Color**		
Descriptor	**HoT**		**LBP**	**PHOG**	**SIFT**	**Histogramă Lab**		**Color names**
Clasificator	RF	SVM	SVM	SVM	SVM	RF	SVM	SVM
Rată de recunoaștere corectă	45.92	48.25	42.2	29.5	53.2	31.3	28.9	33.3

3. Recunoașterea automată a pigmenților

Problema recunoașterii compușilor chimici dintr-un pigment colorat este relevantă în domeniul restaurării tablourilor, unde este imperativ necesară, înainte de restaurare, identificarea compușilor chimici, astfel încât la refacerea culorilor să nu degradeze iremediabil tabloul. Pentru aceasta am început cu construcția unui sistem care să se specializeze în identificarea unui pigment fiind dată o imagine RGB –IR. Sistemul construit are două componente: alegerea spațiului de culoare, respectiv sistemul de învățare automată folosit.

3.1.Baza de date de pigmenți

Această bază de date ne-a fost oferită prin amabilitatea Muzeului Național de Istorie al României și este alcătuită din mostre de pigmenți Kremer pe gumă arabică fotografiați cu cameră microscopică în domeniul vizibil (senzori sensibili la roșu, verde, albastru) respectiv infraroșu (IR). Pigmenții Kremer folosiți sunt descriși în calogul dedicat [Kremer 2015]. Baza de date *este proprietatea Muzeului Național de Istorie* al României.

Pregatirea bazei de date. Baza de date disponibilă nu a fost etichetată digital la începutul cercetării, ceea ce a dus la necesitatea dezvoltării unei soluții capabile de a rezolva această problemă.

Dată fiind structura fixă a imaginilor din baza de date, pentru etichetarea acestora s-a utilizat regiunea din imagine ce conținea numele etichetei fiecărei imagini și s-a implementat o soluție simplistă de identificare a etichetei (pentru exemplificare a se vedea figura 3.1).

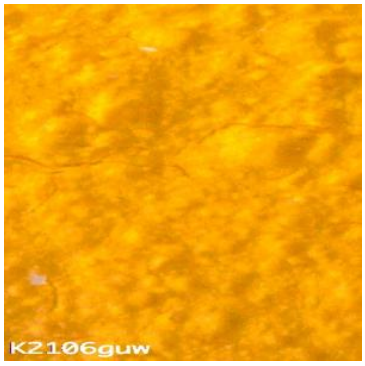


Figura 3.1. Exemplu de mostră de pigment. Pigmentul utilizat în această imagine, K2106 este din categoriu pigmenți bazați pe cadmiu, cu specificția „Galben Cadmiu Nr.9, opac, întunecat".

Dat fiind că soluția întrebuințată pentru situația de față este necesară doar în acest context, s-a implementat o variantă de bază pentru un algoritm de recunoaștere a caracterelor. Caracterele etichetelor imaginilor din baza de date constituie un subset bine determinat al mulțimii caracterelor posibile. Astfel, s-a putut creea un model pentru fiecare caracter posibil, iar ulterior fiecare caracter din regiunea de interes a fost comparat cu toate modelele creeate. Modelul cel mai apropiat de caracterul curent este declarat câștigător.

Algoritmul pentru extragerea etichetelor corespunzătoare fiecărei etichete este prezentat mai jos:

- Generare model caractere etichete
    I. Selectare regiune de interes corespunzătoare etichetelor
    II. Binarizare imagine
    III. Etichetare imagine
    IV. Creeare model pentru fiecare eticheta din imagine
- Identificare etichetă
    I. Selectare regiune de interes corespunzătoare etichetelor
    II. Binarizare imagine
    III. Etichetare imagine
    IV. Pentru fiecare eticheta:
        a. Calculare distanță euclidiană față de fiecare model disponibil
        b. Identificare minim distanță

Odată aplicat algoritmul prezentat anterior se obține eticheta digitală pentru fiecare din imaginile disponibile în baza de date.

Compoziția bazei de date. Baza de date conține 270 de mostre. Kremer, în materialul descriptiv, unde se prezintă compoziția lor, permite gruparea acestora în clase generice. În acest fel am format 15 clase, iar datele propriu zise au fost obținute prin considerarea de eșantione distincte (eșantioane care, ținând cont de faptul ca sunt preluate din mostre neomogene diferă ca și compoziție) din mostrele oferite.

3.2. Implementare și rezultate

Pentru recunoșterea pigmentului am implementat două variante de soluții. Una primară bazată pe o descriere cu medie și varianță și una directă, în care i-am furnizat unui clasificator direct valorile din imaginea RGB. În continuare vom descrie cele două variante.

Varianta 1. Această variantă conține doi pași:

1. *Extragerea trăsăturilor*. Pentru fiecare din imaginile din baza de date s-a extras un set de trăsături descriptive, necesare în procesul de antrenare/clasificare. Astfel, fiecare din imaginile din baza de date a fost împărțită în blocuri de 40x40 pixeli suprapuse parțial (fiecare al 500-lea pixel a fost considerat ca centru al regiunii curente). Fiecare regiune de 40x40 pixeli a fost transformată în spațiul HSV, extrăgându-se ca trăsături media și varianța pe fiecare din planele H, S și V. Acest proces a dus la un set de 6 trăsături pentru fiecare pixel ales ca fiind reprezentativ.
2. *Clasificare.* Odată extrase trăsăturile pentru imaginile din baza de date s-a putut trece la clasificarea propriu-zisă. Astfel, într-o primă instanță s-au împărțit trăsăturile din baza de date în 90% trăsături de antrenare și 10% trăsături de test. Trăsăturile de antrenare au fost folosite pentru antrenarea a diverși clasificatori implementați în biblioteca Weka (Rețea Bayesiană, Perceptron Multistrat, Mașină cu Vectori Suport, LogitBoost,). În urma testării cele mai bune rezultate au fost produse de un clasificator de tip Mașină cu Vectori Suport și s-a obținut 86% rată de clasificare corectă pe baza de date de **antrenare**. În urma clasificării pe baza de date de testare rata de clasificare corectă a scăzut către 57% (minim) – 88% recunoaștere corectă pentru o clasă (pentru o medie de 72%). Evident rezultatele obținute nu sunt mulțumitoare și am trecut la a doua variantă.

Varianta 2. În această a doua variantă am creat o bază de date de peste 7000 eșantione, unde pentru fiecare clasă am considerat o divizare de tipul 10-părți cu validare încrucișată (10-fold cross validation). Cei doi pași tipici sunt:

1. *Extragerea trăsăturilor*. Fiecare din imaginile din baza de date a fost împărțită în blocuri de MxM pixeli ne-suprapuse. Pentru varianta în care am folosit o rețea convoluțională adâncă blocurile au rămas astfel (MxM în RGB). Pentru varianta în care ne-am bazat pe Random Forest din blocurile M x M am utilizat medierea pentru reducerea dimensiunii la blocuri mult mai mici 4x4, iar datele de tip 16 (4x4) x 3 (RGB) au fost utilizate, în varianta de bază (RGB) sau transformate în HSV, respectiv Lab, drept intrare în ansamblul de arbori aleatori.
2. *Clasificare.* Clasificarea propriu-zisă a fost efectuată fie cu o rețea convoluțională adâncă, cu un clasificator de tipul cel mai apropiat vecin, fie cu ansamblul de arbori. Rezultatele finale pot fi urmărite în tabelul 3.1.

Tabelul 3.1 . Rezultate acuratețe în recunoașterea pigmenților

Dimensiune zonă investigată	**32x32**	**32x32**	**32x32**	**32x32**
Clasificator	**Deep - CNN**			
Augumentare	Nici una	Transpunere+rotație	Full=Transp+rot+zgomot	Full+Rețea redusă
Acuratețe	88.2%	91.7 %	92.9%	92.5%
Dimensiune	**4x4**	**4x4**	**4x4**	**3x3**

zonă investigată				
Clasificator	**Random Forest**			
Spațiu de culoare	**RGB**	**HSV**	**Lab**	**Lab**
Acuratețe	89.11 %	90.86%	90.57%	90.94%
Dimensiune zonă investigată	**4x4**	**4x4**	**4x4**	**4x4**
Clasificator	**1-NN**	**1-NN**	**1-NN**	**5-NN**
Spațiu de culoare	**RGB**	**HSV**	**Lab**	**Lab**
Acuratețe	84.11 %	65.7%	86.45%	82.94%

Discuții și Observații:

- Prima observație legată de rezultatele obținute este că această abordare produce rezultate net superioare comparativ cu varianta 1. Abordarea problemei ca o problemă de clasificare supervizată direct pe datele achiziționate evită introducerea diferitelor perturbații.
- A doua observație se referă la performanța rețelei convoluționale: aceasta a necesitat augumentarea datelor pentru a putea produce rezultate comparabile sau superioare altor sisteme de clasificare. Parte din augmentarea datelor a fost să introducem variabilitate în rapot cu structură sau zgomot pentru a evita supraînvățarea. O observație laterală este că totuși efortul computațional este net superior anasamblurilor de arbori, ceea ce nu o recomandă drept soluție rapidă.
- Legat de performanța ansamblului de arbori, după cum se poate observa în figura 3.2, există mici variații în funcție de parametri sistemului: numărul de arbori, respectiv numărul de dimensiuni alese aleator la un nod. Totuși pentru Lab neliniaritatea spațiului convine mai mult rețelei, performanța fiind mai stabilă. Comparativ, pentru HSV există mai multă susceptibilitate la parametri ansamblului. În mod evident spațiul RGB nu permite rezultate atât de bune, iar ansamblul de arbori este incapabil să compenseze neliniaritatea.

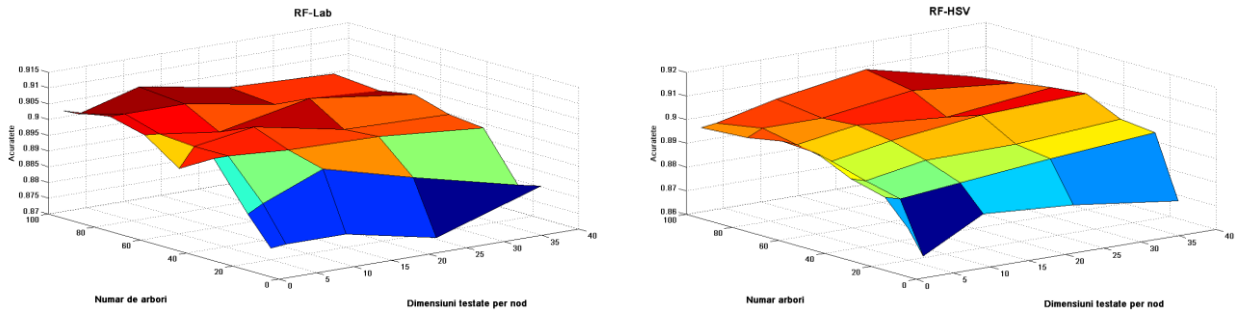


Figura 3.2. Variația performanței de recunoaștere a clasei pigmentului în funcție de numărul de arbori din ansamblu, respectiv numărul de dimensiuni alese aleator per nod. Rezultatele sunt afișate pentru cazul HSV, respectiv RGB.

- Performanța sistemului bazat pe cel mai apropiat vecin arată gruparea pigmenților în spațiul culorilor. În primul rând deși intuitiv, spațiul HSV nu este perceptual, iar distanța euclidiană în interiorul lui nu produce rezultate spectaculoase. Alegerea drept spațiu de culoare a RGB sau Lab, în special, este net mai bună. Apoi performanța este relativ apropiată de a altor clasificatori, ceea ce

înseamnă că gruparea naturală este bună, iar clasificatorii nu au multe zone neliniare pe care să le învețe.

- În condițiile în care performanța rețelei adânci este mai bună (ea folosind și zone mai mari de imagine), o întrebare la care trebuie răspuns este dacă structura spațială (granularitatea) este distinctă pentru diferite tipuri de pigmenți. Pentru a investiga acest fapt am descris imaginile cu descriptorul HoT. Acesta folosește imagini cu niveluri de gri (informația de culoare a fost pierdută), și se bazează pe calculul unor derivate, deci nivelul mediu nu contează.
  Performanța de recunoaștere corectă a pigmenților, în acest caz, este de **32.59%** (semnificativ mai mare decât 6.67% - prin asociere aleatoare). Acest test arată că, la acest nivel de vizibilitate (imaginile sunt achiziționate cu camere semi-microscopice) structura **granulară** este **distinctă** pentru tipurile diferite de pigment.
- Folosirea informației de **infraroșu**. Aceasta permite creșterea performanței. Pentru sistemul bazat pe rețea convoluțională adâncă perfomanța a crescut de la 92.5% la 96.2%. Din nefericire această creștere de performanță nu a fost vizibilă și pentru clasificatorul de tip Random Forest unde perfromanța s-a limitat la 88.68%.

4. Bibliografie

[Arora14] R. S. Arora, , A. Elgammal, "Towards automated classification of fine–art painting style: a comparative study", in: Proc. of ICPR, 2012, pp. 3541–3544

[Bar14] Y. Bar, N. Levy, L. Wolf, "Classification of Artistic Styles using Binarized Features Derived from a Deep Neural Network" In Proc of ECCV workshop on VISART, 2014

[Breiman84] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, "Classiffication and Regression Trees", Chapman and Hall, 1984.

[Breiman01] L. Breiman „Random Forests" Machine Learning, 45(1), pp.3-32, 2001

[Condorovici15] R. Condorovici, C. Florea, C. Vertan "Automatically Classifying Paintings with Perceptual Inspired Descriptors", în J. of Visual Communication and Image Representation, Vol. 26 , pp. 222–230 2015

[Cornelis11] Cornelis, B., Dooms, A., Cornelis, J., Leen, F., Schelkens, P.: Digital painting analysis, at the cross section of engineering, mathematics and culture. In: Proceedings of European Signal Processing Conference. (2011) 1254–1259

[Cortes95]Cortes, C.; Vapnik, V. (1995). "Support-vector networks". Machine Learning 20 (3): 273

[Florea14] C. Florea, L. Florea, C. Vertan, Learning pain from emotion: Transferred HoT data representation for pain intensity estimation, in: European Conf. on Computer Vision: workshop on Assitive Computer Vision and Robotics, Vol. 8927 LNCS, 2014, pp. 778–790.

[Frangi98] A. Frangi, W. Niessen, K. Vincken, M. Viergever, Multiscale vessel enhancement filtering, in: Medical Image Computing 26 and Computer Assisted Intervention, 1998, pp. 130–137.

[Friedman01] Friedman, J. H., „Greedy function approximation: A gradient boosting machine". Annals of Statistics 29, 1189-1232, 2001

[Gatys15] L. Gatys, A. Ecker, M. Bethge, „A Neural Algorithm of Artistic Style", CoRR, abs/1508.06576, 2015, http://arxiv.org/abs/1508.06576.

[Gunsel05] B. Gunsel, S. Sariel, O. Icoglu, "Content-based access to art paintings", in: Proc. of ICIP, 2005, pp. 558–561

[Hebb49] Hebb, Donald „The Organization of Behavior". New York: Wiley (1949).

[Karayev13] Karayev, S., Hertzmann, A., Winnemoeller, H., Agarwala, A., Darrell, T.: “Recognizing image style”. arXiv preprint arXiv:1311.3715 (2013)

[Khan14] F. Khan, S. Beigpour, J. van de Weijer, M. Felsberg, „Painting-91: A Large Scale Database for Computational Painting Categorization”, Machine Vision and Application (MVAP), 25(6):1385-1397, 2014

[Kremer15] "Kremer - pigmente. Product Catalog". Catalog de pigmenti disponibil on-line la adresa " http://kremerpigments.com/download/krp_katalog_US_140806_web.pdf ", accesat noiembrie 2015

[LeCunn98] LeCun, Y.; Bottou, L. ; Bengio, Y. ; Haffner, P. "Gradient-based learning applied to document recognition."Proceedings of the IEEE 86.11 (1998): 2278-2324.

[LeCunn2015] LeCunn Y. LeCunn, Y. Bengio, G. Hinton (2015) Deep learning Nature 521, 436–444

[Lindeberg13] T. Lindeberg, Image matching using generalized scale-space interest points, Scale Space and Variational Methods in Computer Vision Volume 7893 LNCS, pp 355-367, 2013.

[Lu15] X. Lu, Z. Lin, H. Jin, J. Yang, J.Z. Wang, „Rating Image Aesthetics Using Deep Learning”, IEEE Transactions on Multimedia, 17(11), pp 2021 – 2034, 2015.

[McCulloch1943] McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics 5 (4): 115–133.

[Platt98] Platt J, C. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, raport tehnic 1998.

[Saleh15] B. Saleh, A. Elgammal, ”Large-scale Classification of Fine-Art Paintings: Learning The Right Metric on The Right Feature”, CoRR, abs/1505.00855, 2015, http://arxiv.org/abs/1505.00855.

[Shamir10] L. Shamir, T. Macura, N. Orlov, D. M. Eckley, I. G. Goldberg, “Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art”, *ACM Trans. on Applied Perception* 7 (2) (2010) 1–17.

[Smith78] A. R. Smith “Color gamut transform pairs”, Proceedings of SIGGRAPH, pp 12-19, 1978