
Extragere de trasaturi

Anghel Mihail

Rotaru Andrei

Ștefănuță Ionel

Nicu Andrei

Gațu Vlad

Scop

Echipa noastră va prelucra imaginile primite de la echipa I și va extrage trăsături și caracteristici necesare. În final vom avea un clasicator de imagini pe baza de sentimente.

Algoritm LCH

- Histograma globala de culori pentru cele trei canale (Anghel Mihai & Stefanuca Ionel)
- Impartirea imaginilor (Rotaru Andrei)
- Luminozitate imagine (Nicu Andrei)

Despre clasificador

Clasificatorul a rezultat din compunerea a mai multi algoritmi pe care membrii echipei i-au analizat in detaliu.

Rezultatul final va fi media rezultatelor algoritmilor propusi de membrii echipei.

Algoritmi de clasificare & APIs

Propus de:

- **Anghel Mihail**
- **Stefanuca Ionel**

Algoritmii: Algoritmul Puzzle-Gaussian si Algoritmul 5Culori

Documentare: API + functionare

Prezentare Algoritm Puzzle-Gaussian

Datele de intrare si rezultatul algoritmului:

Input: O imagine

Output: Un array cu sentimente ce reies din acea imagine

Cum functioneaza Algoritmul?

Pas 1: Pe imagine se aplica trei neteziri Gaussiene

Pas 2: Imaginea este impartita in noua imagini (imagini de puzzle).

Pas 3: Pentru fiecare imagine de la **Pas 2** se va calcula culoarea cu cel mai bun range (calculul se face pe baza unei histograme de culori)

Pas 4: Pentru fiecare culoare se va asigna un sentiment corespunzator

Pas 5: Toate sentimentele sunt pastrate intr-un HashSet (astfel evitam problema dublurilor)

Pas 1

Netezirile Gaussiene se aplica pentru a scoate in evidenta si a uniformiza culorile predominante din imagine.



Imagine originala



Prima netezire



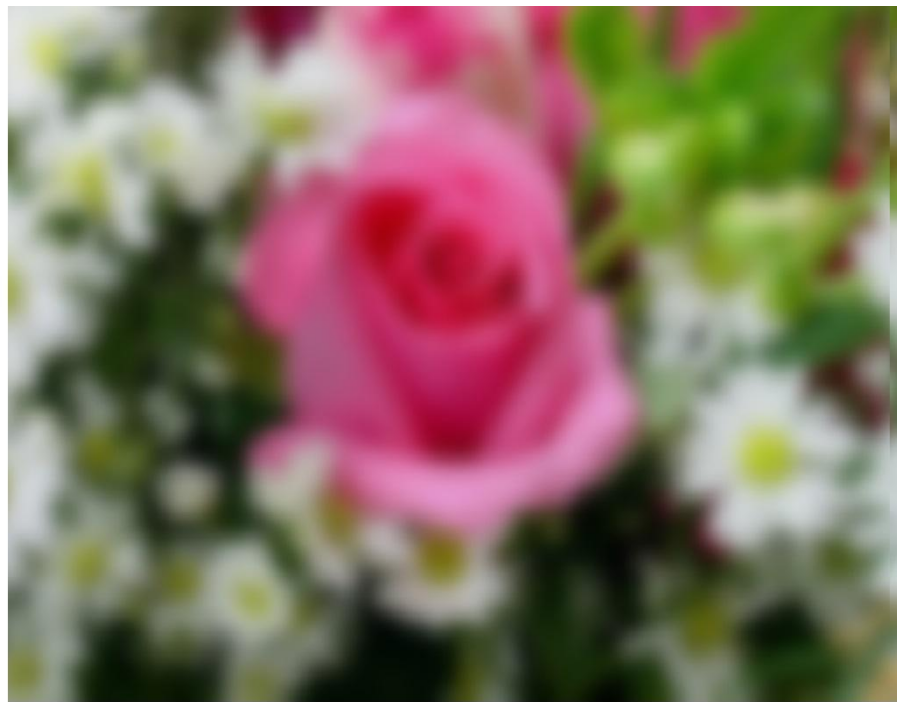
A doua netezire



A treia netezire



Imagine Originala



Imagine dupa procesare

Pas 2



img0



img1



img2



img3



img4



img5



img6



img7



img8

Pas 3 - intervalele maxime pentru exemplul nostru

img0 -> r[51-104] g[102-205] b[0-26]

img5 ->r[51-104] g[102-205] b[0-2]

img1 ->r[102-205] g[102-205] b[0-50]

img6 ->r[51-104] g[102-205] b[0-26]

img2 ->r[51-104] g[102-205] b[0-25]

img7 ->r[51-104] g[102-205] b[0-26]

img3 -> r[51-104] g[102-205] b[0-26]

img8 ->r[51-104] g[102-205] b[0-26]

img4 ->r[102-205] g[102-205] b[0-50]

Pas 4

img0 -> sadness

img1 -> ecstasy

img2 -> terror

img3 -> sadness

img4 -> ecstasy

img5 -> terror

img6 -> sadness

img7 -> sadness

img8 -> sadness

Pas 5

```
HashSet hset = new HashSet<String>();  
  
    for (int i = 0; i < 9; i++){ hset.add(img.genereazaCuloare()); }
```

Emotiile din fiecare imagine vor fi adaugate intr-un HashSet, urmand sa fie procesate pe viitor.

Prezentare algoritm 5culori

Date de intrare: o imagine

Date de iesire: trei sentimente din acea imagine

Pasi algoritmi

Pas 1: Grupare sentimente dupa opt culori

Pas 2: Pentru fiecare culoare de la **Pasul 1**, am ales 5 culori RGB

Pas 3: Se determina culoare cea mai predominanta dintre culorile din gruparea facuta Pasul 1

Pas 4: Pentru “cea mai buna” culoare se returneaza sentimentele corespunzatoare

Pas1: Grupare sentimente

Am grupat cate 3 sentimente dupa o anumita culoare:

- albastru deschis: grief, pensiveness, sadness
- albastru inchis: amazement, distraction, surprise
- galben: ecstasy, joy, serenity
- portocaliu: anticipation, interest, vigilance
- roz: anger, annoyance, rage
- violet: boredom, disgust, loathing
- lime: acceptance, admiration, trust
- verde: apprehension, fear, terror

Pas 2 - alegere culori RGB (doar 4 imagini)

Albastru inchis:

Blue	00 00 FF	0 0 255
MediumBlue	00 00 CD	0 0 205
DarkBlue	00 00 8B	0 0 139
Navy	00 00 80	0 0 128
MidnightBlue	19 19 70	25 25 112

Albastru deschis:

LightBlue	AD D8 E6	173 216 230
SkyBlue	87 CE EB	135 206 235
LightSkyBlue	87 CE FA	135 206 250
DeepSkyBlue	00 BF FF	0 191 255
DodgerBlue	1E 90 FF	30 144 255

Roz:

Pink	FF C0 CB	255 192 203
LightPink	FF B6 C1	255 182 193
HotPink	FF 69 B4	255 105 180
DeepPink	FF 14 93	255 20 147
MediumVioletRed	C7 15 85	199 21 133

Verde:

GreenYellow	AD FF 2F	173 255 47
Chartreuse	7F FF 00	127 255 0
LawnGreen	7C FC 00	124 252 0
Green	00 80 00	0 128 0
DarkGreen	00 64 00	0 100 0

Pas 3 & 4

- Se parcurge imaginea pixel cu pixel
- Se verifica daca rgb-ul corespunde uneia dintre alegerile facute de noi la pasul 2
- Daca da, atunci se incrementeaza un vector corespunzator culorilor alese la pasul 1
- In final se returneaza sentimentele corespunzatoare celei mai bune culori

Explicatii si mai scurte

Avem 5 nuante de rgb pentru o anumita culoare, iar pentru acea culoare sunt asociate 3 sentimente.

Am ales astfel asocierea pentru ca asa grupeaza imaginile echipa 1.

Emotions APIs

Acest API este dezvoltat de studentii de la Oxford si poate fi folosit cu ajutorul internetului. La intrare primeste o imagine, iar la iesire vom avea un fisier de tipul JSON care contine un scor pentru fiecare emotie si emotia in sine.

Poate fi folosit pe mai multe limbaje de programare, cum ar fi: Java, Php, Curl, C#, JavaScript, ObjC, Python, Ruby etc.

Dezavantaje: Se pot trimite cel mult 20 de imagini pe minut si maxim 10.000 pe luna.

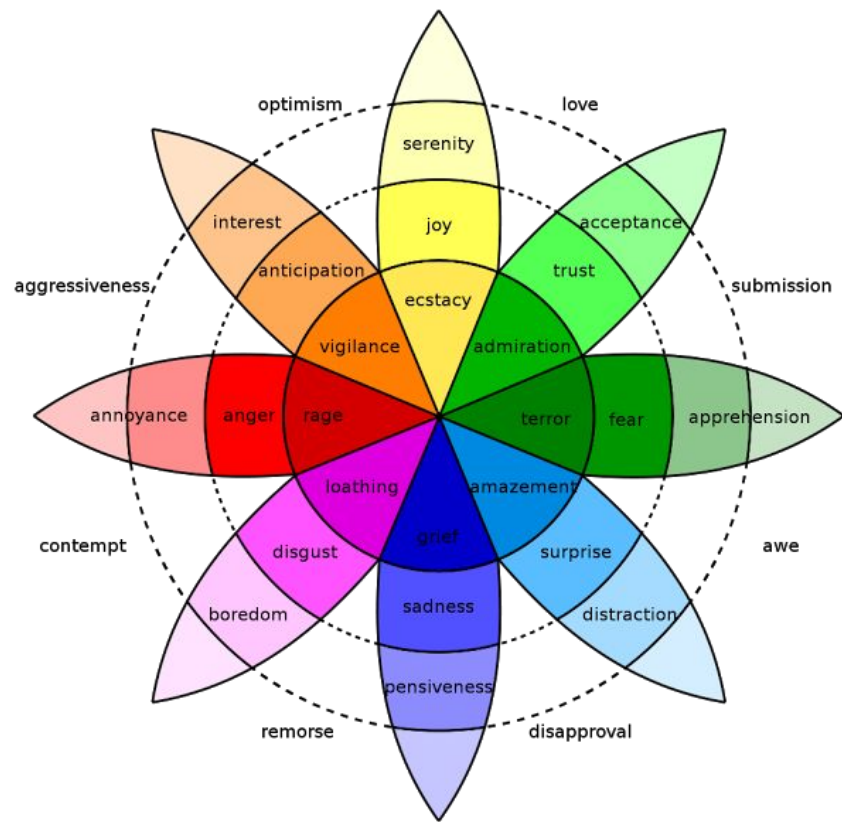
Se poate testa la adresa: <https://www.projectoxford.ai/demo/Emotion>

Algoritmi de clasificare în funcție de distribuția culorilor

Propus de Nicu Andrei

În cele care urmează vom face o scurtă trecere în revistă a câtorva algoritmi axați doar pe distribuția culorilor în imagini. Algoritmii vor primi ca input o imagine (nu este necesară antrenarea în prealabil a algoritmilor) și în funcție de scorul obținut vor forța o decizie și vor returna 3, dintre cele 24, cele mai predominante sentimente.

Imaginea alăturată exprimă modalitatea prin care se poate face delimitarea fizică a sentimentelor, în funcție de culori, pe baza conceptului de puritate a sentimentelor (fiecare sentiment este cel mai puternic exprimat în funcție de o singură culoare predominantă).



În urma exploatării acestui delimitator am putut să facem anumite observații bazate pe manipularea pixelilor din fiecare imagine. De asemenea am putut să concepem o serie de algoritmi ce produc o serie de rezultate destul de apropiate față de sentimentul exercitat de fiecare imagine.

Pentru a putea face o exprimare cu o acuratețe mai ridicată, fiecare imagine va fi redimensionată la un standard de 900x600 pixeli.

1. Clasificare bazată pe range

Acest tip de algoritm își propune clasificarea fiecărui pixel în funcție de valorile parametrilor RGB. Fiecărui sentiment îi este atribuit un range al parametrilor RGB și în funcție de valorile fiecărui pixel se determină care sunt cele 3 cele mai predominante sentimente. Intervalele utilizate se regăsesc în funcția `sortCategory` prezentată în cod.

Avantaje: Algoritmul are o înclinație ridicată de a face o clasificare cu o acuratețe mai mare de 50% în momentul în care imaginea prezintă foarte multe elemente.

Dezavantaje: Algoritmul tinde să facă clasificări greșite datorită faptului că unii pixeli nu pot fi clasificați (nu își regăsesc valorile în range-ul prestabilit)

2. Clasificare bazată pe aproximare

Pentru a rezolva problemele primului algoritm, am încercat să facem o aproximare a fiecărui pixel, folosind proprietățile distanței euclidiene. Astfel fiecare pixel este rotunjit la cel mai apropiat sentiment (în acest caz fiecare sentiment este exprimat doar de o singură valoare unică). După parcurgerea întregii imagini se realizează extragerea primelor 3 cele mai predominante sentimente.

Avantaje: Rezolvă problemele legate de neclasificarea unor pixeli. Crește semnificativ rata clasificărilor corecte în imaginile cu puține detalii.

Dezavantaje: Timpul necesar parcurgerii este crescut. Scade acuratețea la imaginile detaliate.

3. Clasificare bazată pe centrul de greutate al histogramelor

Această metodă propune realizarea unei clasificări realizând un cluster individual la nivelul fiecărei histograme (roșu, verde și albastru) folosind cele mai predominante n valori din cadrul fiecărei histograme.

n este stabilit folosind o limită de acuratețe, limită ce se stabilește ignorând toate valorile rar întâlnite în histogramă. De exemplu dacă $n = 1000$ vom ignora toate nuanțele care se regăsesc cu o valoare mai mică de 1000.

Menționăm că fiecare histogramă este compusă din descompunerea fiecărui pixel în manieră RGB. De exemplu dacă obținem pentru un pixel valorile R:100, G: 150 și B:50 atunci roșu va ce roșu[100]++, verde[150]++ și albastru[50]++.

După stabilirea parametrelor n vom realiza centrul de greutate al clusterului obținut prin cele n puncte și vom obține o valoare din partea fiecărei histograme. Aceste 3 puncte vor fi clasificate sub formă de culoare și folosind clasificarea numărul 2 vom putea determina sentimentul predominant.

Avantaje: Această metodă permite o acuratețe sporită, folosește o distribuție uniformă a pixelilor.

Dezavantaje: Putem extrage un singur sentiment, parametrul n nu poate fi static (există imagini cu o precizie mai mare folosind o valoare mai mică pentru n și imagini cu o precizie mai mare dacă n are o valoare mai ridicată)

4. Clasificare bazată pe centrul de greutate al luminozității

Acest tip de clasificare funcționează în manieră similară cu tipul de clasificare numărul 3 cu diferența că utilizează doar histograma generată de luminozitate.

Luminozitatea se obține prin netezirea parametrilor RGB. Fiecare pixel este raportat la luminozitate și se realizează o histogramă a acestor valori. Prin determinarea centrului de greutate (într-o manieră identică cu clasificatorul numărul 3) se obține o valoare care este aproximată prin metoda numărul 2.

Avantaje: Destul de surprinzător este faptul că acest tip de algoritm are rezultate relativ diferite față de primul în cazurile în care imaginea prezintă diversități mai mari sau mai mici de elemente.

Dezavantaje: Dezavantajele sunt aceleași cu ale clasificatorului numărul 3

5. Clasificare bazată pe distribuția totală a centrului de greutate

Această metodă utilizează toate cele 4 histograme obținute (luminozitate, roșu, verde și albastru) pentru a obține un centru de greutate mai netezit față de variantele de la punctele 3 și 4.

Metodele de determinare a centrului de greutate rămân neschimbate cu singura diferență că vom utiliza o pondere mai mare asupra histogramelor RGB pentru o acuratețe mai mare.

Avantaje: Deși a fost utilizată o pondere mai mare asupra histogramelor RGB, centrul de greutate este mult mai apropiat de cel de la punctul 4 față de cel de la punctul 3.

Dezavantaje: Identice cu cele de la algoritmi 3 și 4

6. Clasificare bazată pe distribuția în masă a culorilor

Acest tip de clasificator este o îmbinare a tuturor clasificatorilor prezentați mai sus și reușește să reabiliteze problemele celorlalte metode.

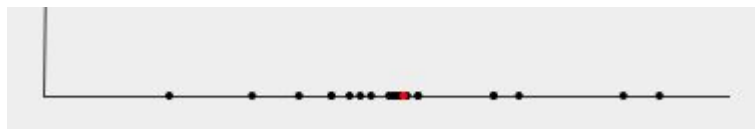
Algoritmul transformă toate cele 24 de sentimente în puncte ale spectrului alb-negru transformând planul bidimensional în unidimensional. Apoi calculează distanța de la fiecare culoare la fiecare centru de greutate al imaginii (centru obținut prin metoda numărul 3) și calculează cele mai apropiate 3 puncte de acest centru din spectrul alb-negru al celor 24 de sentimente.

Avantaje: Metoda are o foarte mare precizie indiferent de tipul imaginii, returnează 3 sentimente

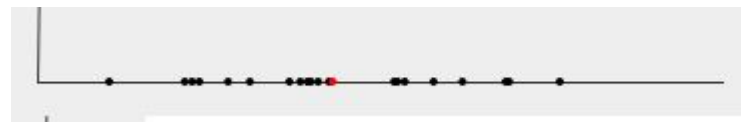
Dezavantaje: Metoda tinde să clasifice imaginile fie fiind foarte vesele (joy), fie fiind la polul opus (terror sau rage). Imaginile cuprinse între aceste două intervale tind să aibă un grad ridicat de amestecare a unor sentimente.

În cele care urmează o să luăm în considerare distribuția câtorva imagini predominante pe un anumit sentiment utilizând acest algoritm.

Pe o marjă de 20 de imagini (adnotate în funcție de un sentiment predominant) au fost înregistrate următoarele 4 rezultate predominante (punctul roșu exprimă centrul de greutate al distribuției fiecărei imagini) pentru algoritmul numărul 6:



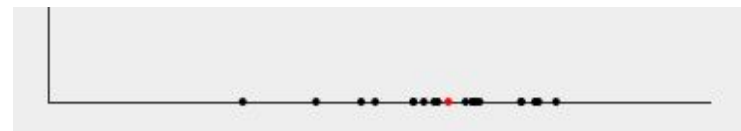
(rage)



(admiration)



(terror)

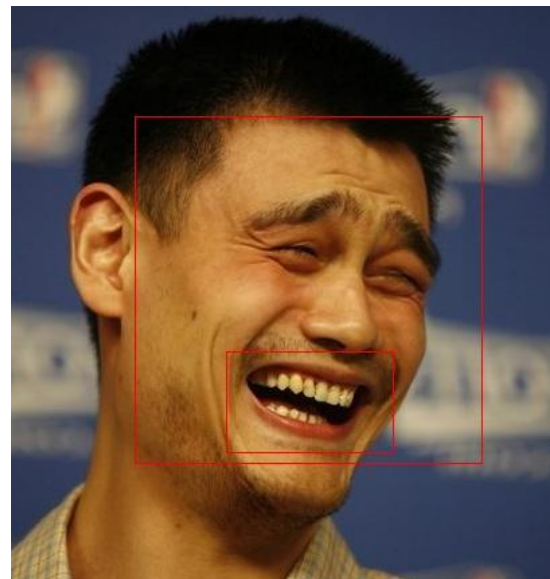
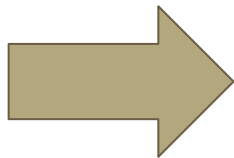


(joy)

Formula utilizată pentru centrul de greutate necesar metodelor 3,4,5 și 6 este dată de formula $x_g = [\text{Sumă}(i=0 \text{ până la } i=255)(K * x_i)] / n$, $y_g = [\text{Sumă}(i=0 \text{ până la } i=255)(K * y_i)] / n$, unde x_g , y_g sunt coordonatele centrului de greutate, K reprezintă o funcție cu valorile 1 dacă $y_i > \text{limita stabilită descrisă la metoda 4}$ și 0 în caz contrar și n reprezintă numărul de ori pentru care funcția K a returnat 1. Acest centru de greutate se aplică fiecărei histograme conform metodelor descrierii metodelor.

În concluzie (doar prin analiza culorilor în manieră de clasificare) se poate spune cu certitudine că doar sentimentele puternice pot fi extrase cu precizie crescută utilizând aceste 6 metode.

Detectarea persoanelor care zambesc utilizand OpenCV



Pentru a face diferenta dintre un zambet si o stare neutra trebuie antrenat un clasificator care sa ne spuna daca o persoana zambeste sau nu.

Pentru aceasta am folosit SVM (suport vector machine) din OpenCV.

Zambet



!Zambet



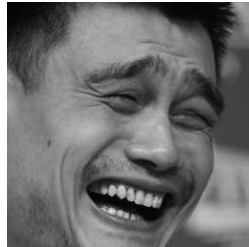
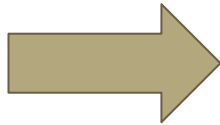
Pasii pentru detectarea zambetului

Pas1: Se incarca o imagine aplicand Greyscale.

Pas2: Se detecteaza toate fetele dintr-o imagine.

Pas3: Pentru fiecare fata se detecteaza si izoleaza gura.

Pas4: Se antreneaza SVM-ul si se compara cu datele de intrare.



Algoritm de clasificare in functie de culori

Propus de Rotaru Andrei

Prezentarea Generala

Intrare: una sau mai multe imagini

Iesire: Cele mai predominante 3 sentimente din fiecare imagine, determinate pe baza culorilor.

Mod de functionare

1. Se creaza histograma globala pentru imagine si se proceseaza fiecare pixel in parte.

2. Pixelul este convertit din format RGB in HSV(Hue Saturation Value).

HSV permite clasificarea mai usoara si precisa a pixelilor in culori.

Clasificare se face pe intervale, in final nu va ramane nici un pixel neclasificat.

In Java, fiecare din cele 3 proprietati(Hue, Saturation, Value) iau valori intre 0 si 1.

3. Pixelii care tind spre alb, negru sau gri sunt numarati separat intr-un vector. Aceste culori nu exprima un sentiment in sine, si vor fi folosite la sfarsitul clasificarii pixelilor in functie de celelalte culori prezente.

Pentru culorile foarte deschise, care tind spre alb am luat intervalul: $val \geq 0.95 \ \&\& \ sat \leq 0.15$ $val \geq 0.90 \ \&\& \ sat < 0.10$	Pentru
culorile foarte inchise, care tind spre negru am luat intervalul: $val \leq 0.20$ $sat < 0.10 \ \&\& \ val < 0.25$	Pentru nuantele de
gri inchis am luat intervalul: $sat < 0.10 \ \&\& \ val \leq 50$	
Pentru nuantele de gri deschis am luat intervalul: $sat < 0.10 \ \&\& \ val > 50$	

4. Restul pixelilor sunt clasificați în funcție de Hue astfel:

Rosu: $\text{hue} \geq 0.95 \mid \mid \text{hue} \leq 0.06$, Portocaliu: $\text{hue} > 0.06 \mid \mid \text{hue} \leq 0.14$,

Galben: $\text{hue} > 0.14 \mid \mid \text{hue} \leq 0.2$, Verde: $\text{hue} > 0.2 \mid \mid \text{hue} \leq 0.41$,

Bleu: $\text{hue} > 0.41 \mid \mid \text{hue} \leq 0.57$, Albastru: $\text{hue} > 0.57 \mid \mid \text{hue} \leq 0.72$,

Roz/mov: $\text{hue} > 0.72 \mid \mid \text{hue} < 0.95$;

Fiecare culoare , în funcție și de celelalte două atribute este atribuită unor sentimente. De exemplu, culoare albastru este atribuită sentimentelor grief și fear când are valoarea ≤ 40 (când este foarte închis), este atribuită la sadness și apprehensions când are valoare cuprinsă între 40 și 55 și este atribuită la trust, admiration, pensiveness și vigilance când în rest.

5. In final se iau in considerare culorile negru, alb si gri in proportie cu restul imaginii.

Unele sentimente ca terror sau rage nu primesc puncte daca culorile negru si gri inchis nu predomina, iar altele ca fear, anger sau sadness trebuie sa fie in proportie de cel putin 25% intunecate. Alte sentimente necesita culori deschise intr-o anumita proportie.