

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

# **Analiza ambianței culorilor în format digital**

propusă de

***Samson Andreea***

**Sesiunea:** *iulie, 2019*

**Coordonator științific**

**Lect. Dr. Anca Ignat**

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ

# **Analiza ambianței culorilor în format digital**

***Samson Andreea***

**Sesiunea:** *iulie, 2019*

**Coordonator științific**

***Lect. Dr. Anca Ignat***

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele \_\_\_\_\_

Data \_\_\_\_\_

Semnătura

**DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemnatul(a) .....

domiciliul în .....

născut(ă) la data de ....., identificat prin CNP ....., absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de ..... specializarea ..... promoția ....., declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul:

\_\_\_\_\_ el

aborată sub îndrumarea dl. / d-na \_\_\_\_\_, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi, .....

Semnătură student .....

## DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Analiza ambianței culorilor în format digital*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași,

Absolvent,  
Samson Andreea

---

# Cuprins

Cuprins	4
Introducere	5
Context	5
Motivație	5
Contribuții	7
Capitolul 1 - Ambianța culorilor în societatea de astăzi	8
Capitolul 2 - Culorile	9
Capitolul 3 - Culorile neutre	12
Capitolul 4 - Ambianța culorilor	14
Transformarea unei culori date într-o culoare „generică”	14
Transpunerea pozei inițiale în poză formată din culorile generice	14
Armonii bazate pe cercul culorilor	16
Complementaritate directă	16
Complementaritate împărțită	16
Analogie	16
Triada	17
Armonia generalizată	17
Concluzii și direcții viitoare	21
Bibliografie	22

# Introducere

## Context

În contextul societății de astăzi, îmbinarea armonioasă a culorilor este importantă în toate aspectele vieții. De la amenajarea spațiilor de birouri într-o manieră plăcută pentru a crea un sentiment de confort, pentru a încuraja creativitatea și pentru a nu distrage vizual, la încropirea unei ținute potrivite pentru situația în cauză, care nu deranjează ochiului prin îmbinări stridente.

Fie că ne dăm seama sau nu, armonia ce ne înconjoară vizual poate avea un efect imens asupra psihicului nostru, iar stăpânirea tehnicilor de asociere reprezintă un avantaj. Un exemplu foarte întâlnit este reprezentat de televiziune. Paletele de culori folosite în anumite scene setează tonul, amplificând sentimentul ce se dorește a se transmite. O scenă tristă capătă amploare prin utilizarea de culori sumbre, pe când culorile pastelate, ce denotă o bună dispoziție, vor fi asociate cu un fir narativ vesel.

Prezenta lucrare își propune să analizeze culorile și îmbinarea acestora, atât prin tehnici predefinite, cât și prin realizarea unui standard ce reflecta viziunea subiectivă.

## Motivație

Fiecare individ poate fi pus în situația de a îmbina culorile într-un mod armonios, proces ce se poate dovedi dificil, în funcție de cunoștințele și de instinctul personal.

Utilizarea corectă a culorilor în locuința proprie este importantă pentru a forma un cămin plăcut, ce nu stresează ochiului sau crează un sentiment de neliniște și nervozitate, agitație.

Pe plan profesional, situațiile variază de a pune la punct o ținută, până la alegerea fundalului unei prezentări sau a culorilor unei aplicații dezvoltate pentru a atrage tipul de client țintit. Alegerea îmbrăcăminte este importantă, pe de o parte, pentru alinierea cu standardul impus într-un anumit eveniment (interviu, petrecere, etc) și, pe de altă parte, pentru accentuarea și manifestarea personalității. De exemplu, în momentul susținerii unei prezentări sau apariției într-un public larg, se poate opta fie spre o ținută clasică, formală, din tonuri neutre, fie spre una mai excentrică, care să atragă atenția asupra culorilor deschise. Un exemplu concret din viața

reală este Regina Elisabeta. Aceasta optează întotdeauna spre nuanțe vii, curajoase, tocmai pentru a putea fi ușor depistată de către mulțime, chiar când se află într-un grup mai numeros.

Principalul motiv pentru care am ales drept tematică ambianța cromatică este determinat de cunoștințele minime în domeniu. Am dorit să aprofundez subiectul, să descopăr cum pot îmbina armonios culorile și să dezvolt o modalitate de a verifica, cu simplitate, dacă o poză este armonioasă din mai multe puncte de vedere. Așadar, am transpus câteva tehnici de verificare a armoniei în algoritmi ușor accesibili, ulterior creând și o metodă generală de a verifica proporția în care o poză este armonioasă, aceasta din urmă se bazându-se pe opinia personală asupra subiectului.

# Contribuții

Tematica lucrării de licență a fost aleasă împreună cu profesorul coordonator, crescând gradual de la concepte de baza ale procesării de imagine spre produsul final.

În timpul explorării pachetelor deja existente pentru prelucrarea imaginilor, am optat pentru folosirea Emgu CV, un framework dezvoltat peste .NET Framework, ce are la baza Open CV, cel mai cunoscut pachet de procesare de imagine. Cu toate acestea, cu cât învățam mai multe despre Emgu CV și îmi dădeam seama de ce am nevoie în elaborarea lucrării, am ajuns la concluzia că funcționalitățile oferite de către framework-ul .NET au fost suficiente pentru implementarea soluției.

Așadar, pe partea de server am ales framework-ul .NET Core 2.1, avantajos din punct de vedere al performanței pe care o oferă sistemului și al scalabilității, deci posibilitatea de a dezvolta soluția. System.Drawing este namespace-ul ce oferă funcționalitate pentru a lucra cu imaginile, fiind un substitut suficient al Emgu CV-ului.

Pe partea de client am utilizat cea mai nouă versiune de Angular, și anume Angular 8.0.0.

Am ales să lucrez cu ultimele tehnologii din dorința de a învăța și de a experimenta cu acestea, de a rămâne la curent cu noutățile într-o societate în care tehnologia se dezvoltă mult prea repede pentru a ține pasul și a reuși a fi în aliniament cu acestea.



# Capitolul 1 - Ambianța culorilor în societatea de astăzi

În momentul de față, problema ambianței culorilor poate fi extinsă spre aproape orice aspect al vieții, iar soluțiile prezintă numeroase avantaje.

Un oraș al cărui clădiri sunt în aliniament cromatic este mai prezentabil și oferă o impresie mai bună decât unul în care fiecare bloc are o culoare diferită. Prin stăpânirea noțiunilor cromatice se pot forma cartiere sau chiar orașe atractive, ce poate dezvolta turismul și îmbunătăți bugetul local.

În marea de aplicații software aflată pe piață, fiecare idee are măcar câțiva competitori. Ce oferă avantajul aplicației de succes este aspectul pe care îl oferă, gradul de atractivitate pentru public. Alături de ușurința de utilizare (“User Experience”), modul în care arată o platformă (“User Interface”) este factorul decisiv pentru a determina gradul de succes.

Într-o comunitate din ce în ce mai feroce, în care oamenii se luptă pentru a ajunge în top, masca, latura arătată publicului joacă un rol extrem de important și este formată, în primul rând, din îmbrăcămintea și nivelul de încredere pe care îl emanăm. Ținutele alese, dacă sunt armonioase, nu numai că exprimă o parte din personalitatea individului, cât și pot crește nivelul de siguranță de sine pe care îl simte.

Tematica licenței provine din dorința de a aprofunda subiectul culorilor, al armoniei cromatice, și de a ajunge la algoritmi care poate determina nivelul de ambianță dintr-o poză dată.

O tematică puternic dezbătută de-a lungul timpului, termenul de “ambianță” a culorilor este unul subiectiv. Viziunea proprie, filtrul personal bazat pe experiențe anterioare, pe asocieri culoare - sentiment, fac din acest subiect unul complex, care nu ar putea fi restrâns în parametri universali valabili. Din această cauză, am soluția propusă este împărțită în două:

- pe de o parte, avem tipuri de ambianță ce se bazează pe cercul culorilor (“color wheel”);
- pe de altă parte, avem un algoritm general, bazat pe propria viziune asupra culorilor ce sunt armonioase împreună într-un anumit procentaj.

## Capitolul 2 - Culorile

Alegerea culorilor s-a bazat pe cele 12 din roata culorilor. (Fig 1.)

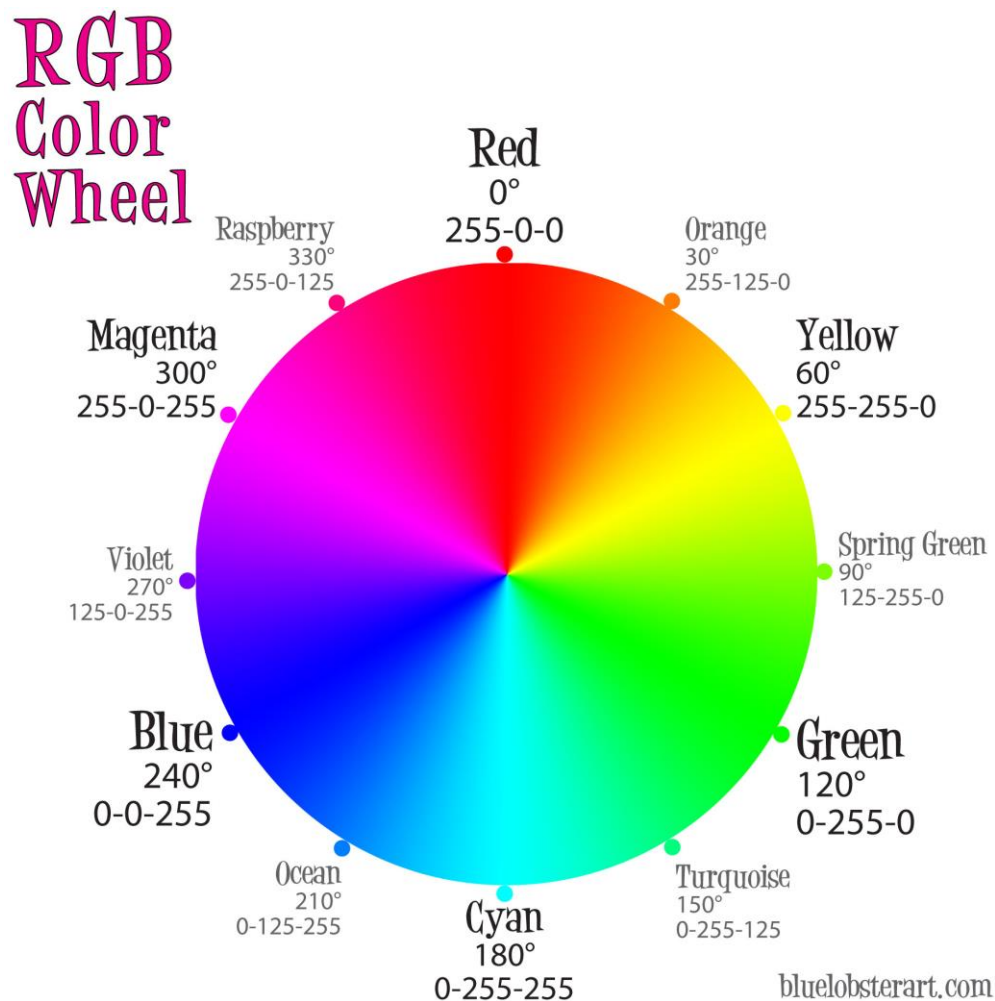


Fig 1. Roata culorilor

Cu toate că, din punct de vedere vizual, cele 12 culori sunt aceleași în mai toate reprezentările acestei roți, din punct de vedere RGB (Red, Green, Blue) exista foarte multe variate.

Prima încercare de a le reprezenta abordată de mine este cea bazată pe intervale. Astfel, soluția optimă a fost crearea unei entități numită "Colors" în care să îmi pot defini proprietățile necesare. Acestea sunt:

- Nume
- Index (ordinea fixă pe roata culorilor)
- Pragul inferior
- Pragul superior
- Culoarea ce va fi afișată în interfață (culori standard din System.Drawing)

- Lista procentajelor de armonie în raport cu celelalte culori (valoare folosită la calcularea armoniei generale, prezentate în capitolul 4)

Limita inferioară reprezintă, pentru fiecare culoare, valoarea absolută a acelei culori. Așadar, intervalul roșu este între  $\text{Rgb}(0, 0, 255)$  și  $\text{Rgb}(0, 124, 255)$ , portocaliu între  $\text{Rgb}(0, 125, 255)$  și  $\text{Rgb}(0, 254, 255)$  și așa mai departe, până când cercul este complet. Rgb este o entitate creată de mine, menită a substitui RGB-ul din Emgu CV. Singurele proprietăți sunt R (Red), G (Green) și B (Blue), de tip int.

În continuare, am transpus poza originală în una nouă formată doar din cele 12 culori (printr-un algoritm ce este prezentat în capitolul 4), rezultatul fiind prezentat în Fig 2.

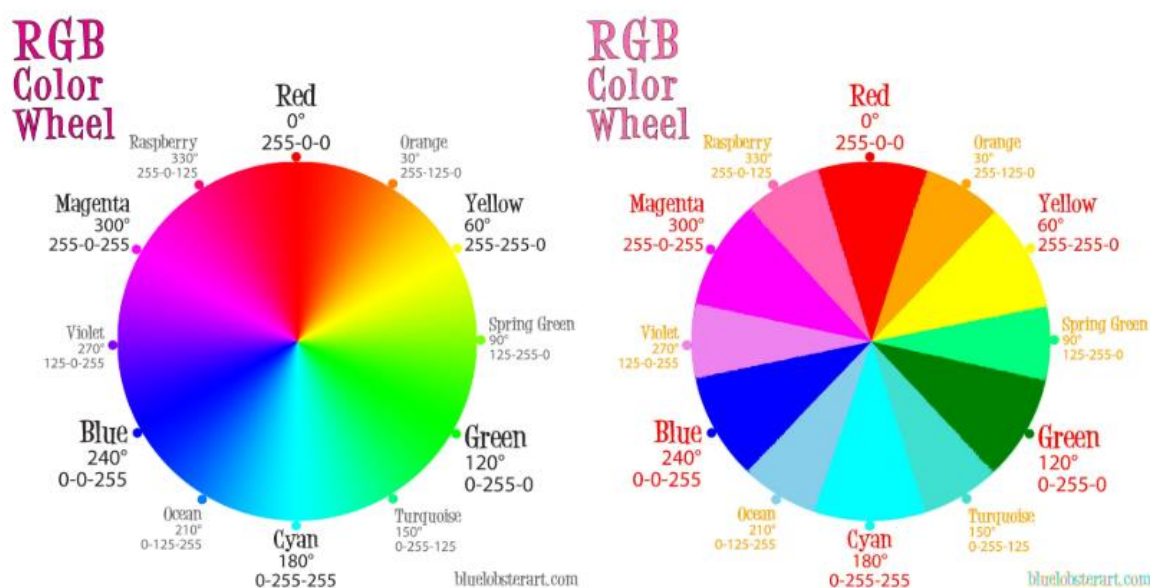


Fig 2. Roata culorilor în culorile absolute pe intervale

Se poate observa că intervalele nu sunt de dimensiune egală, însă complementarele sunt întotdeauna de aceeași dimensiune. După diverse încercări de modificare a intervalelor pentru a obține dimensiuni egale, am constatat că acest fapt nu este posibil din cauza multitudinilor de nuanțe pe care le poate avea o culoare.

După cum am menționat anterior, am renunțat la Emgu CV în favoarea namespace-ului System.Drawing de la .NET, care are definite o multitudine de culori, de toate nuanțele. În cea de-a doua variantă de definire a culorilor, am renunțat la intervale (cu prag inferior și superior), și am numit drept culoare de bază pentru fiecare proprietate corespondentul din

System.Drawing. Astfel, entitatea inițială Colors s-a modificat, proprietățile acesteia devenind următoarele:

- Nume
- Index
- B (valoarea Blue din RGB)
- G (valoarea Green din RGB)
- R (valoarea Red din RGB)
- Culoarea ce va fi afișată în interfață (culori standard din System.Drawing) - corespund definirii anterioare
- Lista procentajelor de armonie în raport cu celelalte culori (valoare folosită la calcularea armoniei generale, prezentate în capitolul 4)

În Fig 3. se poate observa transpunerea pozei pe baza noilor proprietăți. Se remarcă cu ușurință o diferență majoră între cele două interpretări, fapt ce subliniază subiectivitatea tematicii. O singură culoare ce diferă la două definiri ale spectrului cromatic poate duce la diferențe uriașe în rezultatul final.

Am ales să mă axez în continuare pe cea de-a doua interpretare a culorilor, în principal datorită încrederii acordate namespace-ului System.Drawing, menit a veni în ajutorul utilizatorilor.

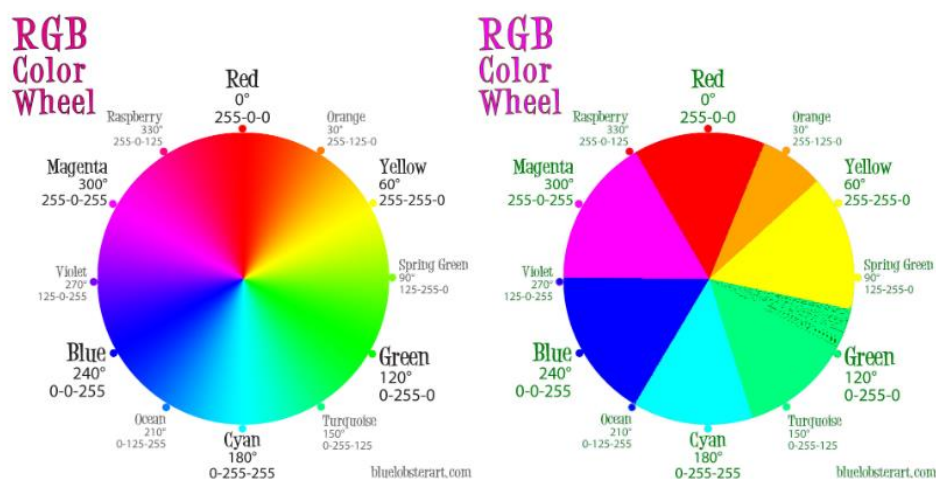


Fig 3. Roata culorilor în culorile absolute din System.Drawing

## Capitolul 3 - Culorile neutre

Una dintre dificultățile identificate armoniei dintre culori o reprezintă culorile neutre. Cele de bază sunt, bineînțeles, alb și negru însă, în combinație cu celelalte culori, pot rezulta nuanțe atât de apropiate de alb sau de negru, încât încadrarea lor la spectrul culorilor de pe roată nu ar fi potrivită.

Analizând mai multe variante de a le trata, am găsit valorile RGB ale câtorva de bază. Observând codurile decimale (RGB), am putut crea o metodă generală prin care se poate verifica dacă o culoare dată este neutră sau nu. Bineînțeles, există o marjă de eroare de eroare pe care mi-o asum, întrucât nuanțele se schimbă și la diferență de o decimală.

Pentru capătul negru al spectrului am ales un interval de 10 decimale pentru fiecare din cele 3 numere. Astfel, dacă atât valoarea din B, cât și cele din G și R sunt mai mici decât 10, spunem despre culoarea în cauză că este o nuanță de negru.

Pentru capătul alb al spectrului, intervalul utilizat este mai mare. Am considerat că, dacă B, G și R sunt în intervalul închis [215, 255], culoarea în cauză este o nuanță de alb.

În Fig 4. și Fig 5. se pot observa tabele folosite drept inspirație, preluate de pe site-ul Rapid Tables.

Black color codes chart



Color	HTML / CSS Color Name	Hex Code #RRGGBB	Decimal Code (R,G,B)
	black	#000000	rgb(0,0,0)
	dimgray / dimgray	#696969	rgb(105,105,105)
	gray / grey	#808080	rgb(128,128,128)
	darkgray / darkgrey	#A9A9A9	rgb(169,169,169)
	silver	#C0C0C0	rgb(192,192,192)

Fig 4. Nuanțele de negru

### White color codes chart

Color	HTML / CSS Color Name	Hex Code #RRGGBB	Decimal Code (R,G,B)
	white	#FFFFFF	rgb(255,255,255)
	snow	#FFFAFA	rgb(255,250,250)
	honeydew	#F0FFF0	rgb(240,255,240)
	mintcream	#F5FFFA	rgb(245,255,250)
	azure	#F0FFFF	rgb(240,255,255)
	aliceblue	#F0F8FF	rgb(240,248,255)
	ghostwhite	#F8F8FF	rgb(248,248,255)
	whitesmoke	#F5F5F5	rgb(245,245,245)
	seashell	#FFF5EE	rgb(255,245,238)
	beige	#F5F5DC	rgb(245,245,220)
	oldlace	#FDF5E6	rgb(253,245,230)
	floralwhite	#FFFAF0	rgb(255,250,240)
	ivory	#FFFFF0	rgb(255,255,240)
	antiquewhite	#FAEBD7	rgb(250,235,215)
	linen	#FAF0E6	rgb(250,240,230)
	lavenderblush	#FFF0F5	rgb(255,240,245)
	mistyrose	#FFE4E1	rgb(255,228,225)
	navajowhite	#FFDEAD	rgb(255,222,173)

Fig 5. Nuanțele de alb

După cum se poate observa, cea mai mare marjă de eroare asumată este la capătul cromatic negru, întrucât culorile Dimgray, Gray, Darkgrey și Silver nu vor fi considerate neutre în cadrul algoritmului meu, ci vor fi rotunjite la cea mai apropiată culoare din cele 12 prezente pe roată.

## Capitolul 4 - Ambianța culorilor

### Transformarea unei culori date într-o culoare „generică”

Pentru transpunerea culorilor din poză în cele 12 culori definite anterior, am ales să folosesc norma euclidiană (Fig 6.).

```
public static Colors GetGenericColorOfGivenColor(Color givenColor)
{
    var smallestDifference = 1000.00;
    var mostUsedColor = Red;

    foreach (var color in GetColors())
    {
        var subtraction = Math.Abs(color.R - givenColor.R) +
                          Math.Abs(color.G - givenColor.G) +
                          Math.Abs(color.B - givenColor.B);
        if (subtraction < smallestDifference)
        {
            smallestDifference = subtraction;
            mostUsedColor = color;
        }
    }

    return mostUsedColor;
}
```

Fig 6. Transformarea unei culori date în culoare generică

Astfel, cele două variabile pentru reținerea distanței minime și a culorii corespunzătoare sunt inițializate cu valori la întâmplare și extreme (1000 pentru diferența minimă). Apoi, parcurg lista cu cele 12 culori și, pentru fiecare în parte, fac diferențele în modul dintre valorile R, G și B ale culorii generice și celei primită ca parametru, pe care apoi le adun. Diferența cea mai mică dintre cele 12 reprezintă distanța minimă dintre culoarea primită și cele generice.

### Transpunerea pozei inițiale în poză formată din culorile generice

Pentru a combina transpunerea fiecărui pixel în culoare generică cu aflarea numărului de apariții a fiecărei din cele 12 culori, mi-am creat o clasă numită “ColorDistributionResult”. Acesta are următoarele proprietăți:

- Distribution - un dicționar în care îmi păstrez culoarea generică alături de numărul de apariții
- Imaginea transpusă (de tip Bitmap)

Metoda care returnează un obiect de tipul `ColorDistributionResult` se găsește în Fig 7. Aceasta parcurge imaginea inițială, primită drept parametru, pixel cu pixel. Un prim pas foarte important pe care îl face este verificarea dacă este culoare neutră sau nu. Dacă este neutră, pixelul este lăsat nemodificat. Dacă nu este neutră, se caută culoarea generică cea mai apropiată, se incrementează numărul de apariții pentru respectiva în dicționar, și se modifică efectiv pixelul în imagine.

```
private ColorDistributionResult GetColorsDistribution(Bitmap image)
{
    var dictionary = Colors.GetColors().ToDictionary(x => x, x => 0);
    for (var i = 0; i < image.Size.Width; i++)
    {
        for (var j = 0; j < image.Size.Height; j++)
        {
            var pixelColor = image.GetPixel(i, j);
            if (!Colors.CheckColorIsNotNeutral(pixelColor))
            {
                var genericColor =
Colors.GetGenericColorOfGivenColor(pixelColor);
                dictionary[genericColor]++;
                image.SetPixel(i, j, genericColor.Color);
            }
        }
    }

    return new ColorDistributionResult() { Distribution = dictionary,
Image = image};
}
```

Fig 7. Transpunerea pozei în culori generice și obținerea distribuției culorilor

Având dicționarul drept rezultat, a fost foarte ușor de manipulat în continuare datele colectate. La fiecare tip de armonie abordat mi-am putut ordona cu ușurință lista culori în funcție de numărul de apariție și prelua, după nevoie, primele 2, 3 sau 4 culori cele mai des întâlnite în imagine.

Principalul avantaj al utilizării unei astfel de metode este performanța. Pozele pot fi, bineînțeles, de dimensiuni foarte mari și, dacă parcurgerea pixel cu pixel s-ar efectua de multiple ori, timpul necesar procesării imaginii ar crește substanțial.



## Armonii bazate pe cercul culorilor

Prima categorie de armonii de culori abordată este cea bazată pe cercul culorilor. Având cercul celor 12 culori, diverse teorii s-au format de-a lungul timpului asupra combinațiilor posibile, care sunt armonioase și care nu. Figurile prezentate în continuare sunt preluate de pe site-ul Sensational Color.

### Complementaritate directă

Complementaritatea directă face referire la culorile opuse de pe cercul cromatic care, datorită contrastului puternic, formează un efect vizual vibrant. Datorită faptului că în entitatea Colors mi-am definit indexul de pe cerc, definirea metodei care verifică dacă două culori sunt complementare s-a rezumat la verificarea egalității resturilor împărțirii la 6.

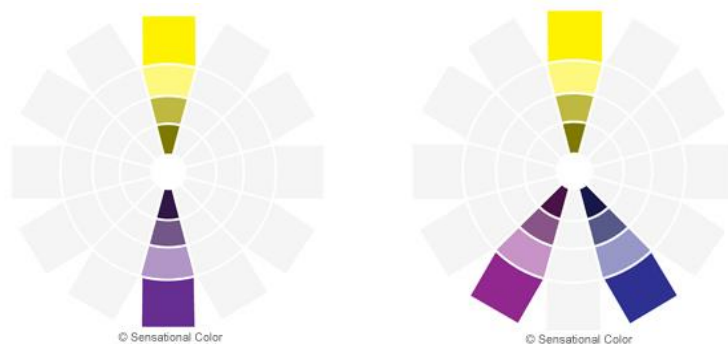


Fig 8. Complementaritate directă și complementaritate împărțită

### Complementaritate împărțită

Complementaritatea împărțită are loc atunci când două dintre culori sunt vecinii complementarei celeilalte culori. Algoritmul se bazează, din nou, pe indexul culorilor primite drept parametru, și verifică dacă diferența dintre cea de-a doua, respectiv a treia culoare (în ordinea indecșilor) este de 5, respectiv 7.

### Analogie

Analogia se bazează pe trei sau mai multe culori situate una lângă alta pe roata culorilor. Primind cele trei culori ca parametru, le-am ordonat după index și am verificat dacă aceștia sunt consecutivi.



Fig 9. Armonia bazată pe analogie

## Triada

Triada este formată din trei culori aflate la distanțe egale pe cercul cromatic. Pentru a fi verificată, compar resturile împărțirii indecșilor corespunzători la 4. Dacă acestea sunt egale, armonia are loc.



Fig 10. Armonia bazată pe triadă

## Armonia generalizată

Algoritmul propus în prezenta lucrare pentru verificarea armoniei generale a unei fotografii se bazează pe scoruri de ambianță.

Primul pas în realizarea acesteia a fost crearea unei clase numite “HarmonyMatrix” în care îmi salvez, pentru fiecare din cele 12 culori definite în clasa Colors, câte o listă de scoruri, procentaje de ambianță în raport cu celelalte culori. În Fig 11. se poate observa un exemplu de astfel de listă definită pentru culoarea roșie.

```

public static List<int> RedHarmonyPercentage = new List<int>()
{
    100, // red
    60, // orange
    30, // yellow
    50, // spring green
    20, // green
    60, // turquoise
    75, // cyan
    50, // ocean
    20, // blue
    50, // violet
    30, // magenta
    75 // raspberry
};

```

Fig 11. Lista procentajelor de ambianță ale culorii roșii

Ordinea procentajelor din listă nu este una aleatorie, ci bazată pe ordinea culorilor definite anterior. Astfel, accesând primul element din lista fiecărei culori, vom ști că este procentul de ambianță față de culoarea roșie.

```

public static float GetAverageImageScore(Bitmap image)
{
    var pictureScore = 0;

    for (var i = 1; i < image.Size.Width - 1; i++)
    {
        for (var j = 1; j < image.Size.Height - 1; j++)
        {
            var sum = 0;

            var pixelColor = image.GetPixel(i, j);
            if (!Colors.CheckColorIsNotNeutral(pixelColor))
            {
                var color = Colors.GetColors().First(x => x.R ==
pixelColor.R && x.B == pixelColor.B && x.G == pixelColor.G);

                sum = AddPercentage(color, sum, image, i - 1, j - 1);
                sum = AddPercentage(color, sum, image, i - 1, j);
                sum = AddPercentage(color, sum, image, i - 1, j + 1);
                sum = AddPercentage(color, sum, image, i, j - 1);
                sum = AddPercentage(color, sum, image, i, j + 1);
                sum = AddPercentage(color, sum, image, i + 1, j - 1);
                sum = AddPercentage(color, sum, image, i + 1, j);
                sum = AddPercentage(color, sum, image, i + 1, j + 1);
                sum = sum / 8;
                pictureScore += sum;
            }
        }
    }

    return (pictureScore / ((image.Size.Width - 2) * (image.Size.Height -
1)));
}

```

Fig 12. Algoritm de determinare al scorului de ambianță per imagine

În Fig 12. este prezentat algoritmul de obținere al scorului de ambianță al unei imagini. Practic, se parcurge imaginea pixel cu pixel și, pentru fiecare în parte se verifică, în primul rând, dacă este culoare neutră sau nu. Dacă nu este, înaintăm cu procesarea.

Algoritmul se aplică de fiecare dată pe o imagine deja transpusă în culorile generice (tip Colors), însă metoda “GetPixel()” returnează un obiect de tip Color, din System.Drawing. Întrucât valorile RGB sunt aceleași (ne amintim că am optat pentru definirea culorilor pe baza celor din System.Drawing), putem converti pixelul de la Color la Colors, căutând în lista de culori generice elementul cu aceleași valori RGB.

În continuare, pentru fiecare din cei 8 vecini ai pixelului, repetăm cei doi pași de mai sus, după care adunăm într-o variabilă primită ca parametru procentul de armonie în raport cu pixelul inițial. (Fig 13.)

```
public static int AddPercentage(Colors color, int sum, Bitmap image, int i, int j)
{
    var neighbour = image.GetPixel(i, j);
    if (!Colors.CheckColorIsNotNeutral(neighbour))
    {
        var neighbourColor = Colors.GetColors().First(x => x.R ==
neighbour.R && x.B == neighbour.B && x.G == neighbour.G);
        sum += color.PercentageList.ElementAt(neighbourColor.Index - 1);
    }

    return sum;
}
```

Fig 13. Metoda de adunare a procentajului de armonie a unui pixel în raport cu un pixel vecin

După parcurgerea tuturor celor 8 vecini, se face media de armonie al pixelului curent, care se adaugă la scorul de armonie al întregii imagini.

Aceiași pași sunt repetați pentru toți pixelii din imagine, după care se calculează media de armonie generală pentru poza în cauză, prin împărțirea sumei la numărul total de pixeli.

Având acum modalitatea de calculare al scorului general, ne putem defini de la ce procentaj în sus putem declara o poză armonioasă. Personal, am ales procentul de 75%, în principal datorită tematicii subiective. Având în vedere că lucrarea în cauză tratează spectrul cromatic numai în funcție de valorile RGB, și nu ține cont de nuanțe și saturații, gradul de generalizare crește suficient de mult, unele rezultate ale algoritmului de transformare a culorilor fiind interpretabile.

În Fig 14. este un exemplu de poză procesată, al cărei procentaj de armonie medie este de 83%., iar în Fig 15., procentajul este de 96%. Se poate observa că prima poză transpusă este cât de cât apropiată de culorile reale, având predominant culori deschise, iar pixelii de culoare neutră sunt tot în extrema albă a spectrului. În cea de-a doua imagine predomină nuanțele

închise, pixelii a căror valori nu se încadrează în algoritmul prezentat la Capitolul 3 fiind transpuși la culoarea cu RGB-ul cel mai apropiat, în cazul de față roz. Aici apare marja de eroare asumată ce determină algoritmul de generare al procentului mediu de ambianță să crească. Dacă respectivii pixeli ar fi fost încadrați la culori neutre, valorile adăugate la metoda din Fig 13. ar fi fost 0, determinând procentul mediu pe întreaga poză să scadă.



Fig 14. Imagine procesată cu procentaj de armonie de 83%



Fig 15. Imagine procesată cu procentaj de armonie de 96%

## Concluzii și direcții viitoare

În prezenta lucrare am analizat o tematică care, deși suficient de subiectivă, poate fi redusă la algoritmi predefiniți, cu asumarea marjei de eroare corespunzătoare. Consider că modalitatea în care a fost dezvoltată aplicația are beneficii multiple, precum ușurința de întreținere și adaptare la viziunea personală și posibilitatea de a vedea imaginea transpusă, pe care se bazează calculele ulterioare.

În viitor, aceasta s-ar putea păstra, pe de o parte, cu tipurile de ambianță prezentate și dezvoltate prin introducerea factorilor de saturație și nuanțare. În acest fel, s-ar măări rata de corectitudine al rezultatului final. Pe de altă parte, aplicația s-ar putea specializa pe analiza pozelor dintr-un anumit domeniu. Să luăm, de exemplu, design-ul interior. Având o bază de poze din domeniu, se poate face o analiză pe formele din imagini și determinarea obiectelor corespunzătoare. Respectivul pot determina, apoi, un factor decisiv în calcularea procentului de ambianță.

# Bibliografie

<https://www.rapidtables.com/web/color/white-color.html>

<https://www.rapidtables.com/web/color/black-color.html>

<https://www.diyphotography.net/50-movie-color-palettes-show-effectively-use-color-film/>

<http://www.tigercolor.com/color-lab/color-theory/color-harmonies.htm>

<http://www.sensationalcolor.com/understanding-color/theory/color-relationships-creating-color-harmony-1849#.XRPPDOgzaUI>