

עבודה 2 – מרחב צבעים

מגשים: נתן דוידוב 211685300

ניקולאי קרחמלוב 320717184

1. RGB & Grayscale

1.1

בסעיף זה קראנו את התמונה, המרנו אותה ל- double ונרמלנו אותה בין 0 ל-1

```
%% 1 RGB and Grayscale|  
%% 1.1  
img = imread('picasso.jpg');  
doubleImg = im2double(img);  
minImg = min(doubleImg(:));  
maxImg = max(doubleImg(:));  
normalizedImg = ((doubleImg-minImg)/(maxImg-minImg));
```

1.2

הצגת התמונה

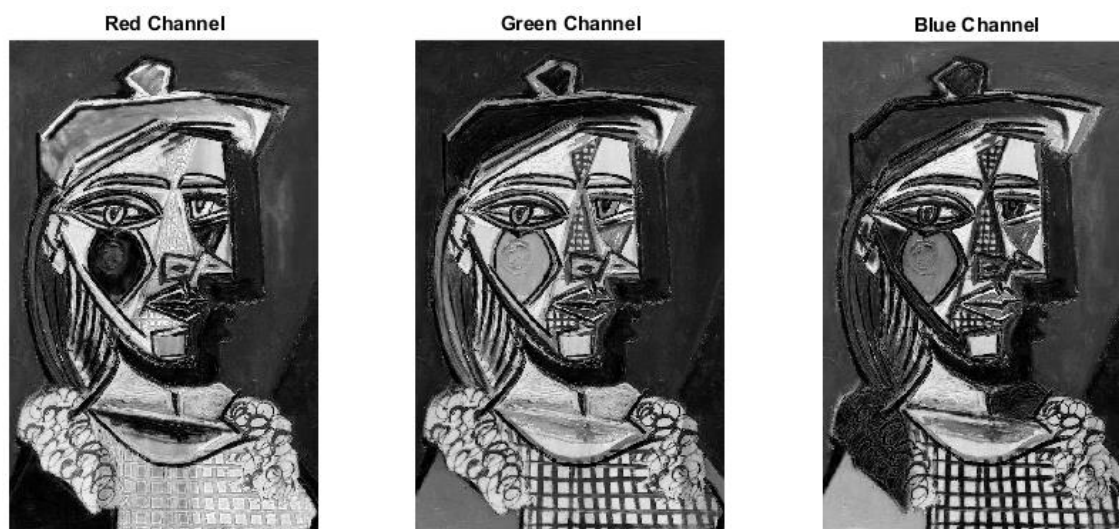
```
% 1.2  
figure();  
imshow(normalizedImg);
```



1.3

חילצנו את ערוצי האדום, הירוק והכחול והצגנו אותם:

```
% 1.3
redCh = normalizedImg(:,:,1);
greenCh = normalizedImg(:,:,2);
blueCh = normalizedImg(:,:,3);
figure();
subplot(1,3,1);
imshow(redCh);
title('Red Channel');
subplot(1,3,2);
imshow(greenCh);
title('Green Channel');
subplot(1,3,3);
imshow(blueCh);
title('Blue Channel');
```



1.4

כאן כתבנו את הפונקציה הבאה על מנת להמיר את התמונה ל-grayscale

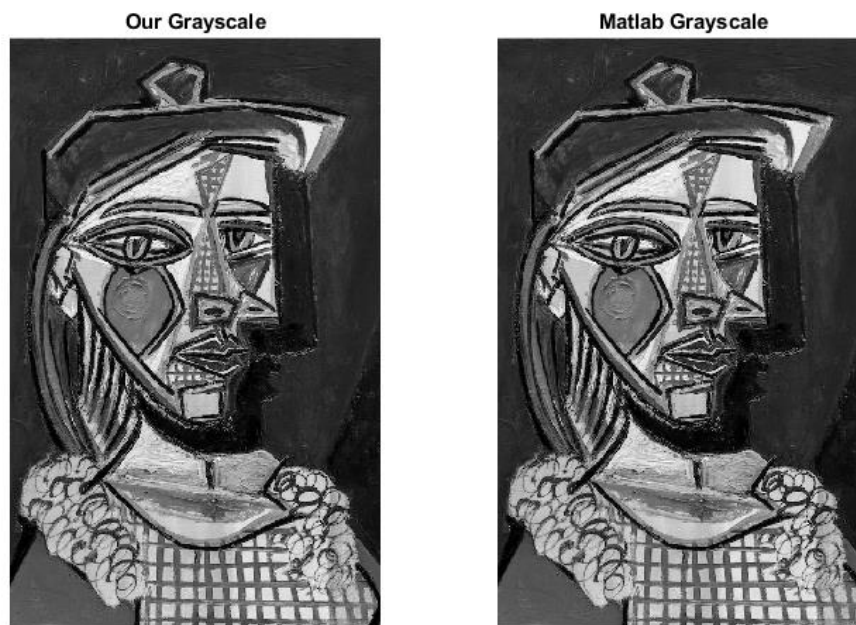
```
function grayScale = convert2gray(redCh, greenCh, blueCh)
    grayScale = 0.2989*redCh + 0.5870*greenCh + 0.1140*blueCh;
end
```

1.5

כעת הצגנו את התמונת grayscale שלנו ליד התמונת grayscale שנוצרה ע"י הפונקציה 'rgb2gray' של מאטלב והשוונו אותם ע"י פקודת ssim שמשווה בין 2 תמונות

```
% 1.5
matlab_grayImg = rgb2gray(normalizedImg);
figure();
subplot(1,2,1);
imshow(grayImg);
title('Our Grayscale');
subplot(1,2,2);
imshow(matlab_grayImg);
title('Matlab Grayscale');

similarity = ssim(grayImg, matlab_grayImg)
```



2. Additive vs Subtractive Color space

2.1

מרחב הצבעים CMYK הוא מרחב צבעים חיסורי שמשתמשים בו בהדפסה צבעונית. ראשי התיבות הם: Cyan (ציאן), Magenta (מג'נטה), Yellow (צהוב) ו-Key, מפתח - בדרך כלל שחור. במודל הזה הצבעים נוצרים ע"י חיסור של אחוזים מסוימים של הצבעים הללו אחד מהשני.

ציאן – מייצג את הספקטרום הכחול-ירוק ומשמש לשליטה בכמות הכחול או הירוק בתמונה

מג'נטה - מייצג את ספקטרום הצהוב ומשמש לשליטה בכמות הצהוב בתמונה

מפתח – בדרך כלל שחור, משתמשים בו על מנת להוסיף פרטים ועומק ולהגדיל את



הניגודיות של התמונה.

2.2

בסעיף זה

%% 2 Additive vs Subtractive Color space

% 2.2

```
black = min(1-blueCh,min(1-redCh,1-greenCh));  
cyan = (1-redCh-black)./(1-black);  
magenta = (1-greenCh-black)./(1-black);  
yellow = (1-blueCh-black)./(1-black);
```

```
figure();  
subplot(2,2,1);  
imshow(black);  
title('Black')  
subplot(2,2,2);  
imshow(cyan);  
title('Cyan')  
subplot(2,2,3);  
imshow(magenta);  
title('Magenta')  
subplot(2,2,4);  
imshow(yellow);  
title('Yellow')
```

% 2.3

```
displayCYMK(cyan,yellow,magenta,black);
```

להלן הערוצים CYMK בנפרד.



3. HSV Color Space

3.1

מרחב הצבע HSV לוקח את מרחב ה-RGB הקובייתי וממפה אותה למרחב גלילי בוא הקורדינטות הן Hue, Saturation, Value. במרחב זה הזווית קובעת את הגוון של הצבע (Hue), הרדיוס את הסטורציה (Saturation), כלומר כמה הגוון חזק ודומיננטי, ואליו הגובה קובע את הבהירות של הצבע (Value/Brightness). המודל הזה תוכנן להיות יותר אינטואיטיבי עבור אנשים מאשר המודל האדיטיבי RGB. המיפוי נעשה באופן הבא:

ראשית מנרמלים את הערכים בערוצים ע"י חלוקה ב-255. לאחר מכן מחשבים את שלושת הביטויים הבאים:

$$C_{max} = \max(R', G', B')$$

$$C_{min} = \min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

לבסוף נחשב את הערכים של כל ערוץ לפי:

Hue calculation:

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \bmod 6 \right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , C_{max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

Value calculation:

$$V = C_{max}$$

3.2

בסעיף זה מימשנו פונקציה משלנו שממפה את המרחב הצבע RGB למרחב HSV לפי הנוסחאות המפורטות מעלה. להלן הקוד:

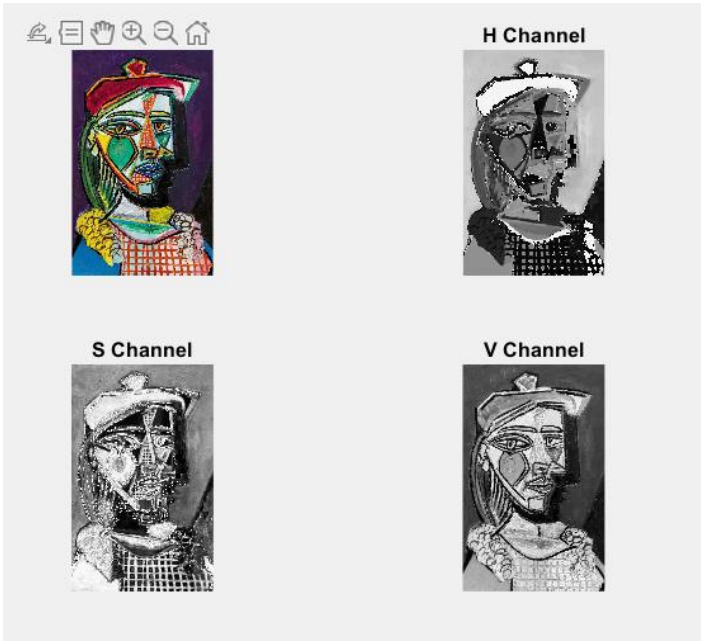

```

1 %Natan Davidov 211685300, Nikolai Krokhmal 320717184
2
3 function hsvImage = dip_rgb2hsv(image)
4     image = double(image);
5     [rows, cols, pixel] = size(image);
6     hsvImage = zeros(rows,cols,pixel);
7     for row = 1:rows
8         for col = 1:cols
9             hue = getHue(reshape(image(row,col,:),1,[]));
10            sat = getSat(reshape(image(row,col,:),1,[]));
11            val = getVal(reshape(image(row,col,:),1,[]));
12            hsvImage(row,col,1) = hue/360;
13            hsvImage(row,col,2) = sat;
14            hsvImage(row,col,3) = val;
15        end
16    end
17 end
18
19 function hue = getHue(pixel)
20     r = pixel(1)/255;
21     g = pixel(2)/255;
22     b = pixel(3)/255;
23     cMax = max([r,g,b]);
24     cMin = min([r,g,b]);
25     deltaC = cMax - cMin;
26     if deltaC == 0
27         hue = 0;
28     elseif cMax == r
29         hue = 60*mod(((g-b)/deltaC),6);
30     elseif cMax == g
31         hue = (2+((b-r)/deltaC))*60;
32     else
33         hue = (((r-g)/deltaC)+4)*60;
34     end
35 end
36
37 function sat = getSat(pixel)
38     r = pixel(1)/255;
39     g = pixel(2)/255;
40     b = pixel(3)/255;
41     cMax = max([r,g,b]);
42     cMin = min([r,g,b]);
43     deltaC = cMax - cMin;
44     if cMax == 0
45         sat = 0;
46     else
47         sat = deltaC/cMax;
48     end
49 end
50
51 function val = getVal(pixel)
52     r = pixel(1)/255;
53     g = pixel(2)/255;
54     b = pixel(3)/255;
55     val = max([r,g,b]);
56 end
57

```

3.3

בסעיף זה המרנו את התמונה `picasso.jpg` ל-`HSV` ואנחנו מראים כל אחד מהערוצים בנפרד:



ניתן בירור לראות שכפי שציפינו ערוץ V הוא פשוט תמונה greyscale של התמונה המקורית (כפי שראינו פעמים רבות בעבודה הראשונה). בנוסף ניתן לראות שערוץ S השחיר כמעט את הפנים שם לפי התמונה המקורית כמעט אין צבע כלל, ואילו איזורים בהם הצבעים יותר בוהקים וברורים בתמונה המקורית כמו החולצה, הכובע או הלחי הירוקה ניהיו בלבן.

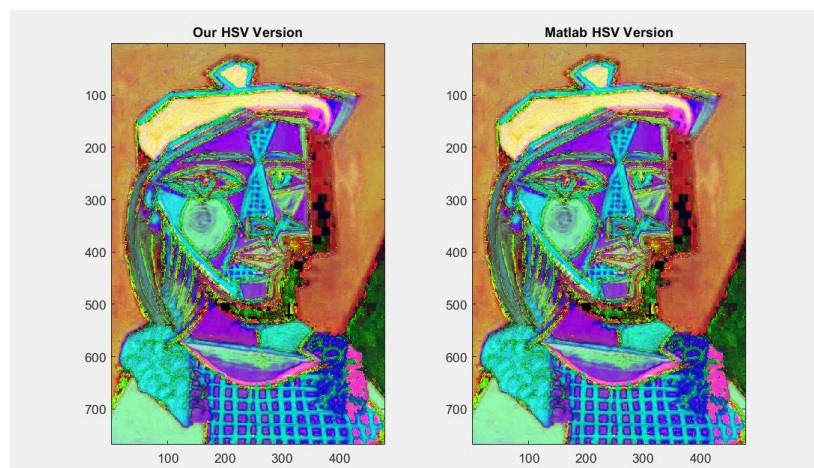
בערוץ H קשה לפרש את מה שקיבלנו ללא color map, אבל כן ניתן להבחין בקובע שמוצג לבן משום שצבעי אדום כהה הם איזור ה-360 מעלות לעומת אדום בהיר שהוא באיזור ה-10 מעלות (האף שחור).

3.4

כמו ששמנו לב בסעיף הקודם, ללא colormap נכון להציג את המיפוי החדש של הצבעים בערוץ H קשה מאוד להבין איזה צבע נמצא איפה. לכן במקרה כזה נאמר שה colormap ממש נחוש. מצד שני שימוש ב-hsv colormap שלא לצורך יכול לבלבל אותנו בהבנת הצבעים בתמונה, נניח את נשתמש בו לא במיפוי hsv.

3.5

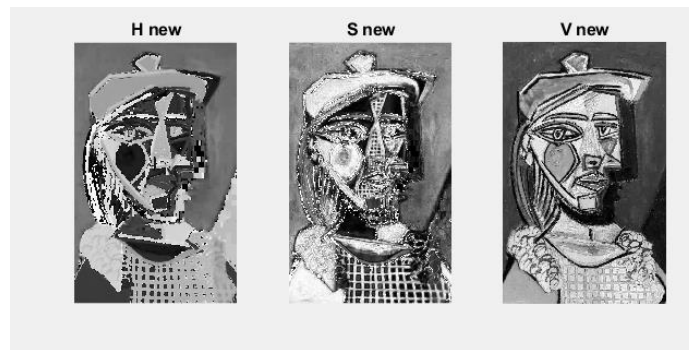
נציג השוואה של המימוש שלנו של hsv למימוש הקיים במאלטאב:



כפי שניתן לראות המימוש מוצלח.

3.6

בסעיף זה שינינו את הסדר של ערוצי הצבע במרחב RGB ואז המרנו ל-HSV. כן אנחנו מציגים את שלושת הערוצים החדשים:

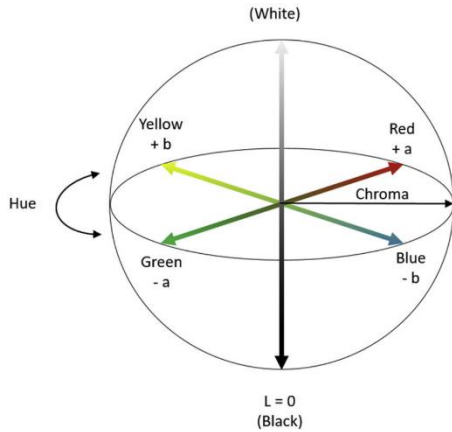


ניתן לראות שהערוץ היחיד שהשתנה זהו ערוץ ה-H, שכן שיחלוף ערוצי ה-RGB משנה את גווני הצבע אבל לא את מידת הבהירות שלו או כמה הוא בוהק. זאת משום ששחלוף הערוצים שינה כמה מכל צבע יש, אבל לא כמה האם יש או אין צבע כלל (מה שמשפיע על ערוץ S) ולא את הבהירות הכללית (מה שמשפיע על ערוץ V).

4. L*a*b Color Space

4.1

מרחב הצבע Lab הוא מרחב צבע שפותח ע"י הנציבות הבינ"ל לתאורה במטרה לקרב את התפיסה האנושית של צבעים. מרחב צבע זה תופס את כל מכלול הצבעים שהעין האנושית יכולה לראות באור יום. זהו מרחב תלת מיימדי (L,a,b) כך ש:



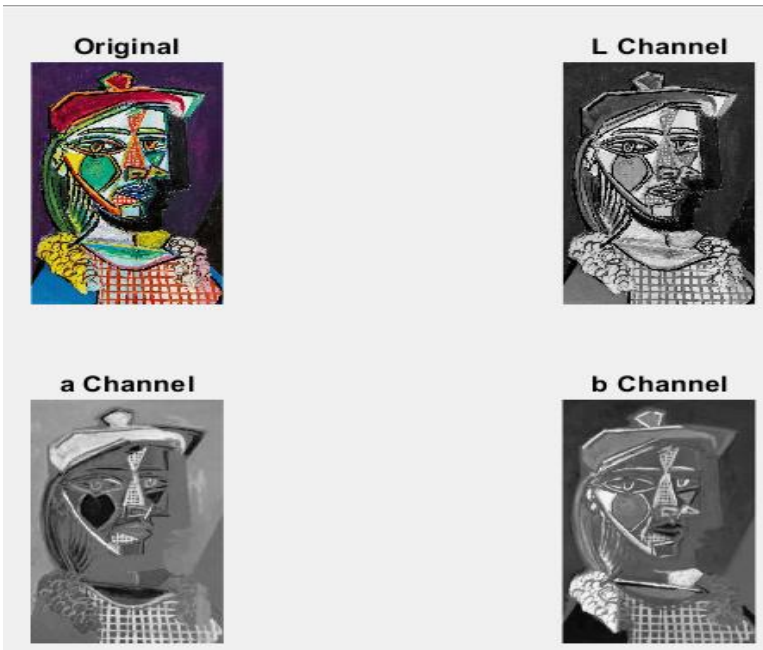
L – בהירות הצבע. מיימד זה יכול לקבל ערכים בין 0 כאשר 0 מייצג שחור 100 מייצג לבן.

a – מנעד של אדום עד ירוק – מיימד זה קובע כמה הצבע של העצם הוא ירוק או אדום, כאשר ערכים חיוביים מיוחסים לצבע האדום וערכים שליליים לצבע הירוק.

b – מנעד של כחול על צהוב – מיימד זה קובע כמה הצבע של העצם הוא צהוב או כחול, כאשר ערכים חיוביים מיוחסים לצבע הצהוב וערכים שליליים מיוחסים לבע הכחול.

4.2 כעת נציג פירוק של התמונה Picasso.jpg לערוצים Lab:

כפי שניתן לראות ערוץ L נותן את הבהירות של התמונה, באופן דומה ל grey scale מהעבודה הקודמת. בנוסף נבחין כי הרקע של התמונה מכיל הרבה כחול ולכן בערוץ b הרגע מקבל ערכים שליליים ונראה כהה. הכובע אדום ועל הלחי יש חלק ירוק, שהם בערוץ a נראים לבן ושחור בהתאמה כפי שציפינו. להלן הקוד:



%4.

```
image = double(imread("picasso.jpg")); %% change when merge
minImg = min(image(:));
maxImg = max(image(:));
normalizedImg = ((image-minImg)/(maxImg-minImg));
labImage = rgb2lab(normalizedImg);
lChannel = labImage(:,:,1);
aChannel = labImage(:,:,2);
bChannel = labImage(:,:,3);
subplot(2,2,1), imshowLab(lChannel, aChannel, bChannel), title('Original');
subplot(2,2,2), imshow(lChannel, []), title('L Channel');
subplot(2,2,3), imshow(aChannel, []), title('a Channel');
subplot(2,2,4), imshow(bChannel, []), title('b Channel');
```

5. Compare Color Spaces

בסעיף זה נדרשנו לבחור מניפולציה אחד על ערוצי הצבע של של ארבעת מרחבי הצבע בהם עסקה העבודה ולהשוות בין השפעות המניפולציה על ערוצי הצבע השונים. בחרנו במניפולציה של נגטיב, כלומר לכל ערוץ צבע להחסיר את הערך הקיים מהערך המקסימלי שהערוץ יכול לקבל.

מניפולציות על RGB:

קטע הקוד שעושה את המניפולציה לערוצי הצבע:

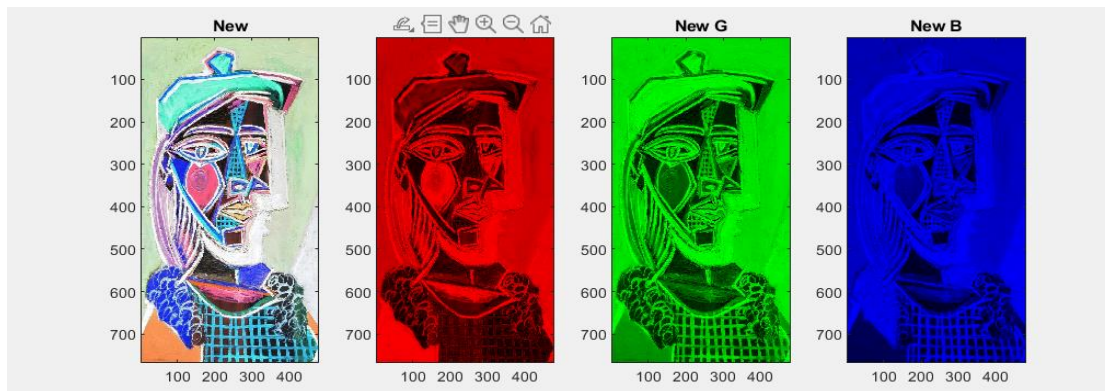
```
%% 5. Compare Color Space

% RGB channel manipulation

img5 = imread('picasso.jpg');
doubleImg5 = im2double(img5);
minImg5 = min(doubleImg5(:));
maxImg5 = max(doubleImg5(:));
normalizedImg5 = ((doubleImg5-minImg5)/(maxImg5-minImg5));

redCh5 = normalizedImg5(:,1);
greenCh5 = normalizedImg5(:,2);
blueCh5 = normalizedImg5(:,3);
oneMat = ones(size(zeroMat));
redCh5 = oneMat - redCh5;
greenCh5 = oneMat - greenCh5;
blueCh5 = oneMat - blueCh5;
zeroMat = zeros(size(redCh5));
newRGB = cat(3,redCh5,greenCh5,blueCh5);
redCh = cat(3,redCh,zeroMat,zeroMat);
greenCh = cat(3,zeroMat,greenCh,zeroMat);
blueCh = cat(3,zeroMat,zeroMat,blueCh);
redCh5 = cat(3,redCh5,zeroMat,zeroMat);
greenCh5 = cat(3,zeroMat,greenCh5,zeroMat);
blueCh5 = cat(3,zeroMat,zeroMat,blueCh5);
figure()
subplot(1,4,1), imagesc(newRGB), title('New');
subplot(1,4,2), imagesc(redCh5), title('New R');
subplot(1,4,3), imagesc(greenCh5), title('New G');
subplot(1,4,4), imagesc(blueCh5), title('New B');
```

להלן התוצאות:



ניתן לראות בבירור כי פיקסלים עם ערוצים בערכים גבוהים קיבלו לאחר המניפולציה ערכים נמוכים ולהיפך. ניתן לראות זאת בבירור בפנים שבתמונה המקורית היו ברובן לבנות וכאן הן שחורות. בנוסף נשים לב לכך שבתמונת הנגטיב יש המון כחול – זאת משום שבמתמונה המקורית כנראה הפיקסלים היו עם ערכים גבוהים של אדום וירוק ונמוכים של כחול וראינו שם בעצם את הצבע הצהוב. באופן דומה הלחי שבמקור הייתה ירוקה כהה (כנראה שילוב של ירוק עם כחול) הפכה לאדמדמה סגולה מה שמרמז על כך שרמת הכחול בתמונה המקורית הייתה בינונית עד גבוהה.

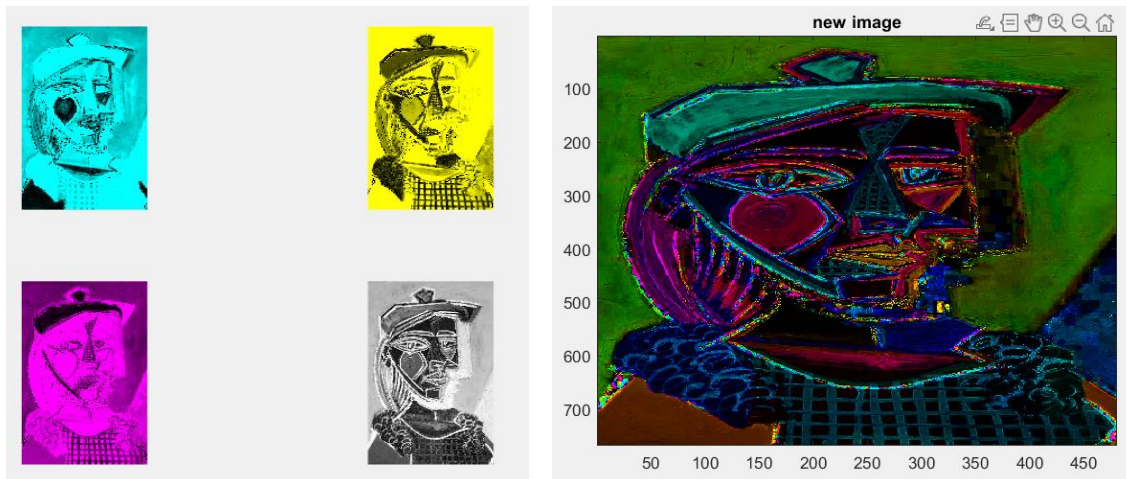
מניפולציות על CYMK:

הקוד שמבצע את המניפולציה על מרחב הצבע הנל:

```
%% CYMK channel manipulation

black5 = onesMat - black;
cyan5 = onesMat - cyan;
magenta5 = onesMat - magenta;
yellow5 = onesMat - yellow;
displayCYMK(cyan5,yellow5,magenta5,black5);
reconR = (onesMat - cyan5).*(onesMat - black5);
reconG = (onesMat - magenta5).*(onesMat - black5);
reconB = (onesMat - yellow5).*(onesMat - black5);
reconRGB = cat(3,reconR,reconG,reconB);
figure()
imagesc(reconRGB), title('new image');
```

להלן התוצאות בערוצים הנפרדים ולצידם התמונה ממורת חזרה לRGB:



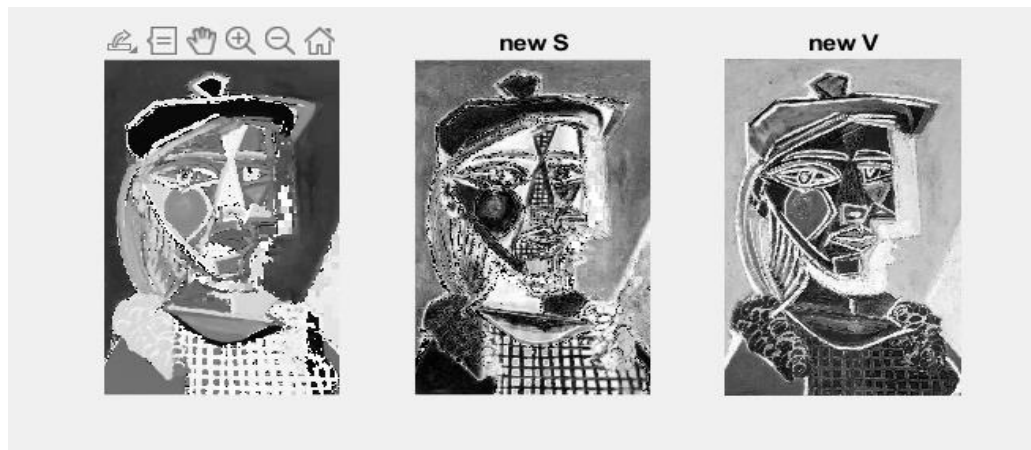
כאן ניתן לראות את האפקט של הנגיב על הערוצים בדיוק כמו במניפולציה על RGB, הכי ברור נראית תמונת ה-K. מצד שני בשחזור RGB ניתן לראות תמונה שונה ממש. כאמור מרחב CYMK הוא מרחב צבעים חיסורי, שמטרתו להיות מודפס על דף לבן ולהחסיר ממנו את הצבעים בערוצי RGB כדי לקבל את הבצע הרצוי. ע"י מניפולציית הנגיב אנחנו בעצם בכל ערוץ, הופכים ערכים גבוהים לנמוכים ולהיפך. מכאן הציפייה היא שבכל מקום בוא אנחנו "מחסירים קצת" כדי לקבל צבעים בהירים, לאחר המניפולציה "נחסיר המון" לקבלת צבעים כהים. בתמונה המקורית הפנים של הדמות כמעט לבנות וכאן הן שחורות לחלוטין כצפוי. פיקסלים שבהם היו ערכים גבוהים של ערוץ אחד, נניח הכובע האדום, כדי לקבל את הצבע יש צורך להחסיר המון ירוק וכחול (שנעשה ע"י ערכי Y גבוהים מ-Mi גבוהים כפי שניתן לראות בסעיף 2). לעומת זאת בנגיב נחסיר המון אדום ולכן הכובע התקבל בגוון ירק כחלחל טוריקיז שכזה שיצא יחסית בהיר מאחר והכוסע המקורי כהה.

מניפולציות על ערוצי HSV:

הקוד שאיתו עשינו את המניפולציה:

```
159 %% HSV channel manipulation
160
161 newH = onesMat - hChannel;
162 newS = onesMat - sChannel;
163 newV = onesMat - vChannel;
164 figure();
165 imshowHSV(newH,newS,newV);
166 subplot(2,3,1), imagesc(hChannel), title('original h');
167 subplot(2,3,2), imagesc(sChannel), title('original s');
168 subplot(2,3,3), imagesc(vChannel), title('original v');
169 subplot(2,3,4), imagesc(newH), title('new h');
170 subplot(2,3,5), imagesc(newS), title('new S');
171 subplot(2,3,6), imagesc(newV), title('new V');
172 img123 = cat(3,newH,newS,newV);
173 img22 = hsv2rgb(img123);
174 figure()
175 imagesc(img22);
176
```

להן תוצאות המניפולציה:



כאשר התמונה בRGB נראית כך:



כאן נשים לב שצבעים בהירים יהפכו לכהים וההיפך
כתוצאה מהפעולה על ערוץ V. בנוסף פיקסלים
שבהם היו ערכים גבוהים של צבע מסויים והיו
saturated ניתן לראות שכעת הם ניראית דהויים
לחלוטין, כמו לדוגמא הלחי שהייתה ירוקה או הכובע.

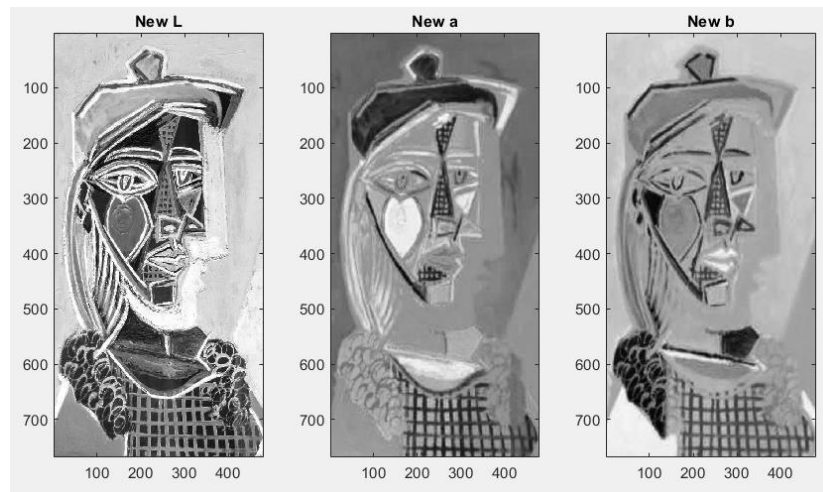
פחות ניתן לשים לב לכך, אבל הפעולה על ערוץ H
שינתה באופן קיצוני את כל הגוונים פרט לגוונים של
אדום שהם נמצאים שני קצוות הספקטרום ושם
השינוי היחיד הוא מגוונים בהירים של אדום כתום
לגוונים יותר כהים של אדום סגול. אמנם בגלל
הסטורציה הנמוכה ממש קשה להבחין בתופעה זו.

מניפולציה על מרחב LAB:

הקוד האחראי על המניפולציה:

```
177 %% L*a*b channel manipulation
178
179 newL = 100*oneMat-1Channel;
180 newA = -1*aChannel;
181 newB = -1*bChannel;
182 figure()
183 colormap("gray");
184 subplot(1,3,1), imagesc(newL), title('New L');
185 subplot(1,3,2), imagesc(newA), title('New a');
186 subplot(1,3,3), imagesc(newB), title('New b');
187 figure();
188 imshowLab(newL,newA,newB);
189
```


להלן תוצאות המניפולציה:

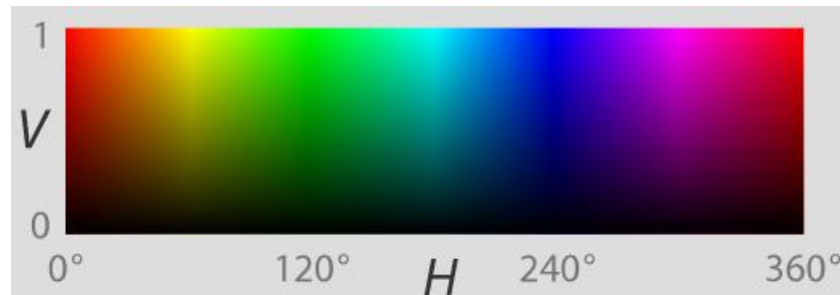


התמונה המלאה תראה כך:



ניתן לראות שקיבלנו בדיוק את תמונת הנגטיב כמו הערוץ RGB! כפי שנאמר בחלק 4, במחרב זה 3 צירים, L המגדיר את הבהירות, a המגדיר את מידת הירוק-אדום של הפיקסל ו-b המגדיר את מידת הכחול-צהוב של הפיקסל. כאן המניפולציה שלנו מעט שונה מבשאר הערוצים, מאחר וערוצים a ו-b מקבלים ערכים בין 100 ל-100- כי להפך אותם פשוט החלפנו להם סימן. לכן כל פיקסל החליף בין מידת האדום והירוק שלו ובין מידת הכחול והצהוב שלו. במודל זה בעצם אנחנו כביכול מגדירים את הצבעים אדום וירוק וכחול בתור צבעים נגדיים. אנחנו עושים זאת עי כך ששמים את הירוק והאדום על אותו הציר ואת הכחול למול הצהוב שלפי מודל RGB הוא בדיוק השילוב בין ירוק ואדום! לכן כאשר עושים נגטיב בצורה כזאת אנחנו בעצם לכל פיקסל בפועל הופכים את מידות הירוק אדום וכחול שלו, ולכן מקבלים בדיוק את הנגטיב של RGB.

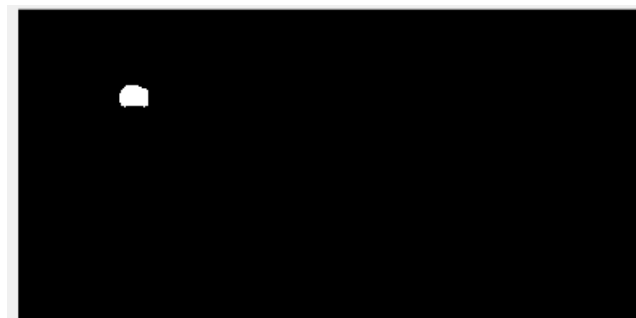
ראשית החלטנו לעבור למרחב HSV מכיוון שיש שם ערוץ אחד שמיועד לגוון צבע (H) ולכן יותר פשוט לקחת משם תחום שיתאים לנו, במקרה של הפקק נרצה צבע כחול שהוא בין 220 ל-240



מכיוון שנרמלנו את המרחב בין 0 ל-1 אז ניקח את התחום בין $\frac{220}{360} = 0.6111$ ל- $\frac{240}{360} = 0.6666$ ומכיוון שהכחול כהה ניקח רק את האזורים שבהם V הוא בין 0.1 ל-0.3 וזה מה שנקבל:



קיבלנו גם רעשים לא רצויים ולכן נסנן אותם עם median filter שראינו בתרגיל קודם שהוא טוב לסינון רעשים נקודתיים כאלה. להלן התוצאה שנקבל אחרי הפילטר:



לאחר מכן נמצע את ערכי ה-x וה-y של כל הנקודות הלבנות וניצור את העיגול במקום המתאים.



להלן הקוד

```

cap1 = imread("cap1.png");
cap1hsv = dip_rgb2hsv(cap1);
cap2 = imread("cap2.png");
cap2hsv = dip_rgb2hsv(cap2);
cap3 = imread("cap3.png");
cap3hsv = dip_rgb2hsv(cap3);

cap1hsv = cap1hsv(:,:,1)>0.6111 & cap1hsv(:,:,1)<0.6666 &...
    cap1hsv(:,:,3)<0.3 & cap1hsv(:,:,3)>0.1;
cap2hsv = cap2hsv(:,:,1)>0.6111 & cap2hsv(:,:,1)<0.6666 &...
    cap2hsv(:,:,3)<0.3 & cap2hsv(:,:,3)>0.1;
cap3hsv = cap3hsv(:,:,1)>0.6111 & cap3hsv(:,:,1)<0.6666 &...
    cap3hsv(:,:,3)<0.3 & cap3hsv(:,:,3)>0.1;

figure();
subplot(1,3,1);
imshow(cap1hsv);
subplot(1,3,2);
imshow(cap2hsv);
subplot(1,3,3);
imshow(cap3hsv);

cap1hsv = medfilt2(cap1hsv,[15,15]);
cap2hsv = medfilt2(cap2hsv,[15,15]);
cap3hsv = medfilt2(cap3hsv,[15,15]);

figure();
subplot(1,3,1);
imshow(cap1hsv);
subplot(1,3,2);
imshow(cap2hsv);
subplot(1,3,3);
imshow(cap3hsv);

[row1, col1] = find(cap1hsv);
x1_center = sum(col1)/length(col1);
y1_center = sum(row1)/length(row1);
cap1 = insertShape(cap1,"circle",[x_center y_center 30],LineWidth=5);

[row2, col2] = find(cap2hsv);
x2_center = sum(col2)/length(col2);
y2_center = sum(row2)/length(row2);
cap2 = insertShape(cap2,"circle",[x2_center y2_center 30],LineWidth=5);

[row3, col3] = find(cap3hsv);
x3_center = sum(col3)/length(col3);
y3_center = sum(row3)/length(row3);
cap3 = insertShape(cap3,"circle",[x3_center y3_center 30],LineWidth=5);

figure();
subplot(1,3,1);
imshow(cap1);
subplot(1,3,2);
imshow(cap2);
subplot(1,3,3);
imshow(cap3);

```