



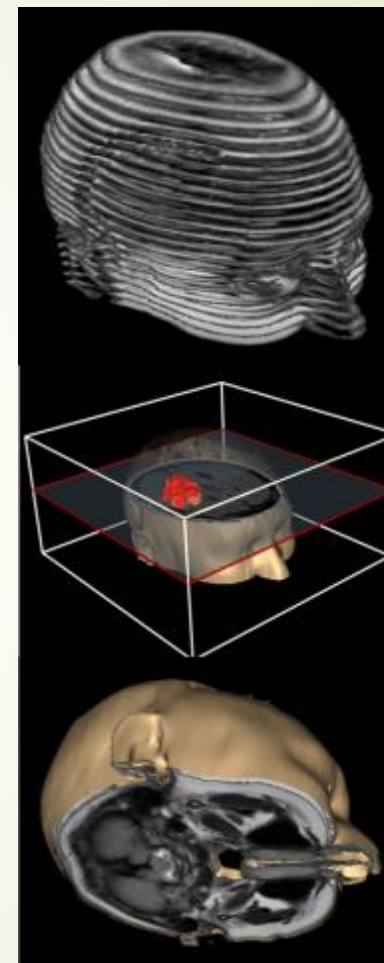
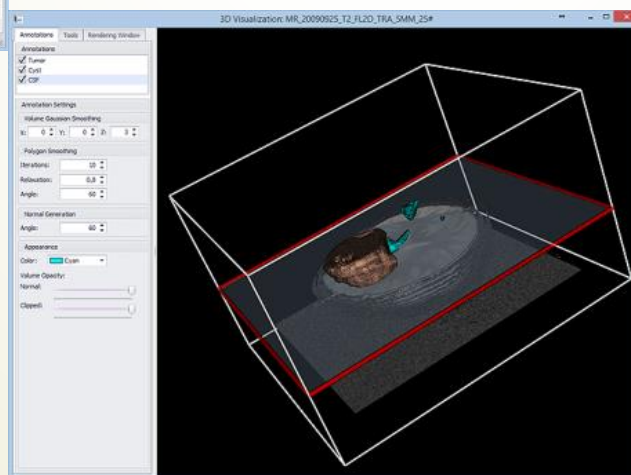
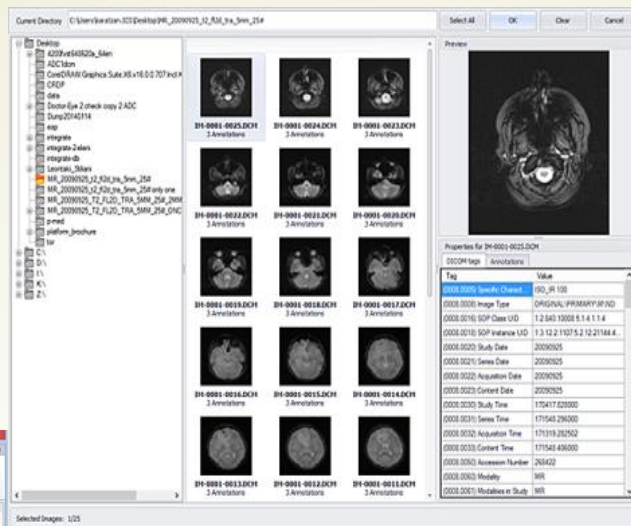
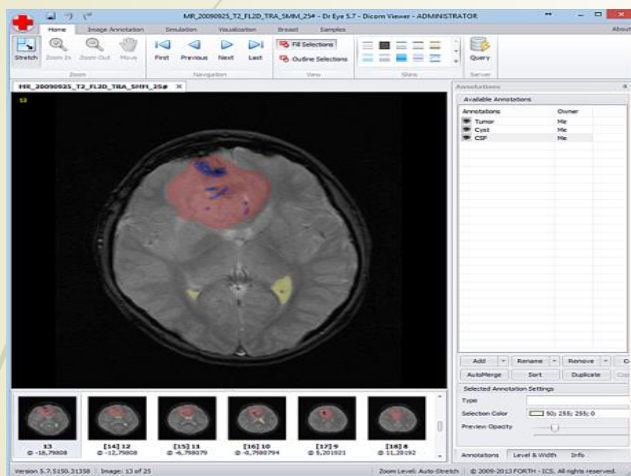
Εργαστήριο 1

Ψηφιακής Επεξεργασίας

Εικόνας

Κώστας Μαριάς

2



<http://biomodeling.ics.forth.gr/>

Η ψηφιακή εικόνα...

- ▶ Μπορούμε να σκεφτούμε μια εικόνα ως συνάρτηση f , από $\mathbb{R}^2 \rightarrow \mathbb{R}$:
 - ▶ $f(x, y)$ δίνει την ένταση στο (x, y)
 - ▶ Ρεαλιστικά , περιμένουμε την εικόνα για να οριστεί μόνο σε ένα ορθογώνιο , με έναν πεπερασμένο εύρος:
- ▶ Μια έγχρωμη εικόνα είναι μια διανυσματική συνάρτηση τριών εικόνων R, G, B

$$f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix}$$

Η ψηφιακή εικόνα...

- Χρησιμοποιούμε διακριτές τιμές στις εικόνες (π.χ. δειγματοληψία)
- Η εικόνα μπορεί τώρα να αναπαρασταθεί ως ένας πίνακας με ακέραιες τιμές
- Η Ένταση στο $i=3, j=2$ είναι $f(3, 2) = 157$

	0	1	2	3	4
0	156	159	158	155	158
1	160	154	157	158	157
2	156	159	158	155	158
3	160	154	157	158	
4	156	153	155	159	
5	155	155	155	157	
6	156	153	157	156	
7	159	159	156	158	



Η ψηφιακή εικόνα...

- Υπάρχουν 3 βασικά formats ψηφιακής εικόνας:

RGB



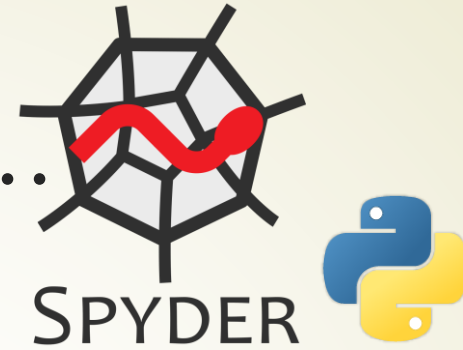
Grayscale



Binary



Εικόνες σε Python...



- ▶ Η βασική δομή δεδομένων στη Python είναι ο πίνακας, ένα ταξινομημένο σύνολο πραγματικών ή σύνθετων στοιχείων. Αυτό είναι φυσικά κατάλληλο για την απεικόνιση εικόνων με πραγματικές τιμές χρώματος ή έντασης.
- ▶ Η Python αποθηκεύει τις περισσότερες εικόνες ως δισδιάστατους Πίνακες, όπου κάθε στοιχείο της μήτρας αντιστοιχεί σε ένα μόνο εικονοστοιχείο στην απεικονιζόμενη εικόνα.
- ▶ (Το Pixel προέρχεται από το στοιχείο εικόνας και συνήθως υποδηλώνει μια μόνο κουκίδα σε μια οθόνη υπολογιστή.) Για παράδειγμα, μια εικόνα που αποτελείται από 200 σειρές και 300 στήλες από διαφορετικές έγχρωμες κουκίδες θα αποθηκεύεται σε Python ως πίνακας 200 x 300.

https://www.w3schools.com/python/python_arrays.asp

Python requirements?

- ▶ Η Python έχει παρόμοιο τρόπο λειτουργίας με την προγραμματιστική γλώσσα C. Προκειμένου να εκτελεστεί ο εκάστοτε κώδικας που έχουμε δημιουργήσει, χρειάζεται να εισάγουμε packages.

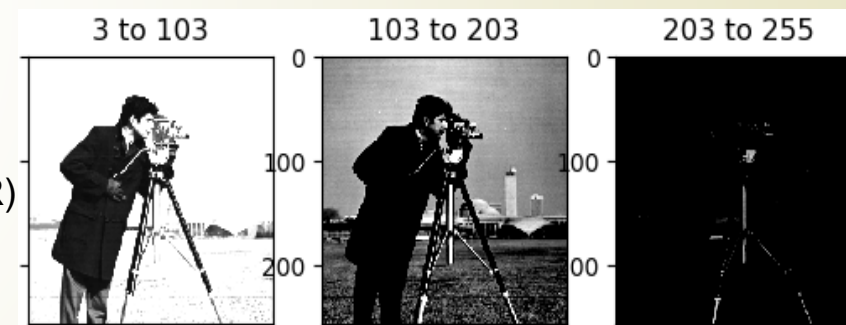


Display Εικόνων

- Για να εμφανιστεί μια εικόνα στην Python χρησιμοποιούμε τη συνάρτηση **matplotlib.pyplot.imshow**, με την σύνταξη :
 - `import matplotlib.pyplot`
 - `matplotlib.pyplot.imshow(f,cmap="String")`
- *f είναι ένας πίνακας εικόνας και cmap="String" είναι ο αριθμός των επιπέδων έντασης (intensity levels) για το display. Αν παραλειφθεί το cmap το default είναι RGB 256 levels .*
- Λόγω της μεγάλης έκτασης της εντολής αυτής και πολλών άλλων είναι καλο να μετανομάζουμε κάθε package με συντομογραφία.
 - `import matplotlib.pyplot as plt`
 - `plt.imshow(f,cmap="String")`

Με την σύνταξη:

- `plt.imshow(f,cmap = "gray",vmin=NUMBER, vmax=NUMBER)`
- Όλες οι τιμές < από το low, εμφανίζονται με μαύρο
- Όλες οι τιμές > από το high, εμφανίζονται με λευκό
- Οι ενδιάμεσες τιμές με αποχρώσεις του γκρι (8bit->256 αποχρώσεις)
- Με αυτό τον τρόπο μπορούμε να απεικονίσουμε εικόνες με χαμηλό εύρος τιμών (dynamic range) ή με θετικές και αρνητικές τιμές πίνακα.



Read and write image data, get information about contents of image files

- ▶ Οι βασικές εντολές της Python για να διαβάσουμε/αποθηκεύσουμε εικόνες είναι:
- ▶ import matplotlib.image as mpimg (import package)
- ▶ mpimg.imread (Read image from graphics file)
- ▶ mpimg.imsave (Write image to graphics file) ή plt.savefig(Write figure to graphics file)

Διαβάζοντας εικόνες από διαφορετικά format

- Ας χρησιμοποιήσουμε την συνάρτηση `mpimg.imread`. Με το παρακάτω παράδειγμα διαβάζουμε μια εικόνα **truecolor** στο workspace του Python και την αναθέτουμε στην μεταβλητή **RGB**.
- `RGB = mpimg.imread('football.jpg')`
- `plt.figure(1)`
- `plt.imshow(RGB)`

Εάν η μορφή αρχείου εικόνας χρησιμοποιεί εικονοστοιχεία 8-bit, το `mpimg.imread` αποθηκεύει τα δεδομένα στον χώρο εργασίας ως πίνακα `uint8`. Για μορφές αρχείων που υποστηρίζουν δεδομένα 16 bit, όπως PNG και TIFF, το `imread` δημιουργεί ένα πίνακα `uint16`.

Αποθηκεύοντας εικόνες από Python workspace σε graphics file

- ▶ Για να εξάγετε δεδομένα εικόνας από τον χώρο εργασίας Python σε ένα αρχείο γραφικών σε μία από τις υποστηριζόμενες μορφές αρχείων γραφικών, χρησιμοποιήστε τη συνάρτηση εγγραφής `mpimg.imsave`.
- ▶ Αυτό το παράδειγμα φορτώνει την ευρετηριωμένη εικόνα (`img`) από ένα `path` και στην συνέχεια εξάγει την εικόνα ως αρχείο bitmap (BMP).

```
1. img = mpimg.imread('cameraman.tif')  
2. mpimg.imsave('my_cameraman2.bmp',img,cmap = "gray")
```

Classed of images

Βασικοί τύποι αποθήκευσης πινάκων

Storage Class of Image	Storage Class of Image
bool	This format supports 1-bit images, imwrite creates a 1-bit image file.
int	Default integer type (same as C long; normally either int64 or int32)
int 16-32	Integer (-32768 to 32767)-(-2147483648 to 2147483647)
uint8	If the output image file format supports unsigned 8-bit images, imwrite creates an unsigned 8-bit image file.(0 to 255)
uint16	This format supports unsigned 16-bit images (PNG or TIFF), imwrite creates an unsigned 16-bit image file (0 to 65535)

Δημιουργώντας έναν πίνακα-εικόνα σε

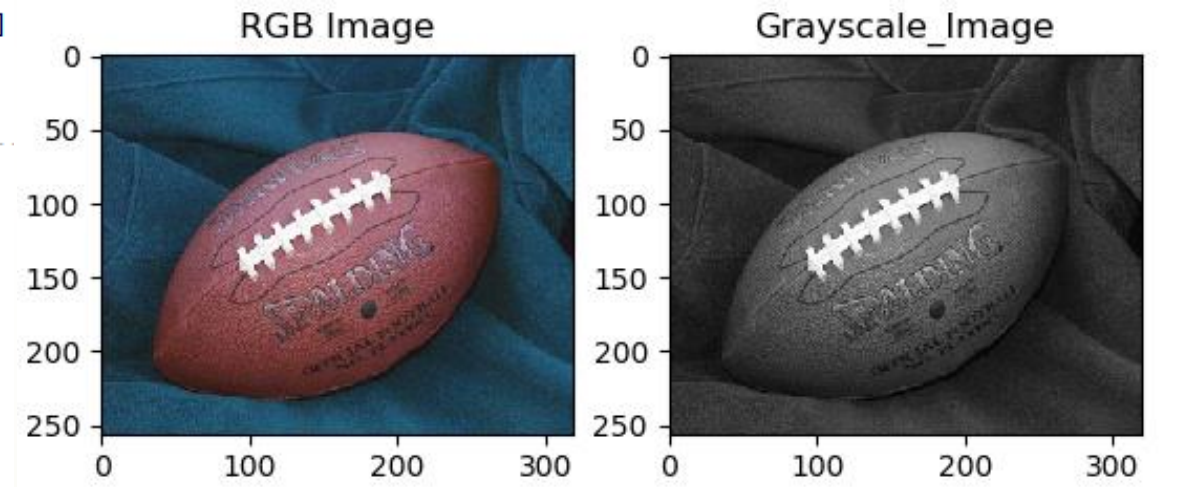
13

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 A=[[1, 2, 3, 4],[5, 6, 7, 8],[ 9, 10, 11, 12],[13, 14, 15, 16]]
5 A=np.asarray(A, dtype=np.int32)
6
7 plt.figure(1)
8
9 plt.imshow(A,cmap="gray")
10 plt.suptitle('Grayscale')
11
12 B=A+10
13 C=A-10
14
15 Image_A=np.hstack((A,B,C))
16 Image_B=np.hstack((C,B,A))
17 Image_C=np.hstack((B,A,C))
18
19 Image=np.vstack((Image_A,Image_B,Image_C))
20
21 np.min(A)
22 np.min(Image)
23 np.max(Image)
24
25 Image2=Image
26 Image2 = Image2.clip(min=0)
27 Image2=np.uint8(Image2)
28
29 plt.figure(2)
30
31 plt.imshow(Image,cmap="gray")
32 plt.suptitle('First-float32 image')
33
34 plt.figure(3)
35 plt.imshow(Image2,cmap="gray")
36 plt.suptitle('Second-uint8 image')
37
```


Grayscale image conversion

Για να μπορέσουμε να μετατρέψουμε μια RGB εικόνα σε grayscale, πρέπει να ακολουθήσουμε την εξής διαδικασία:

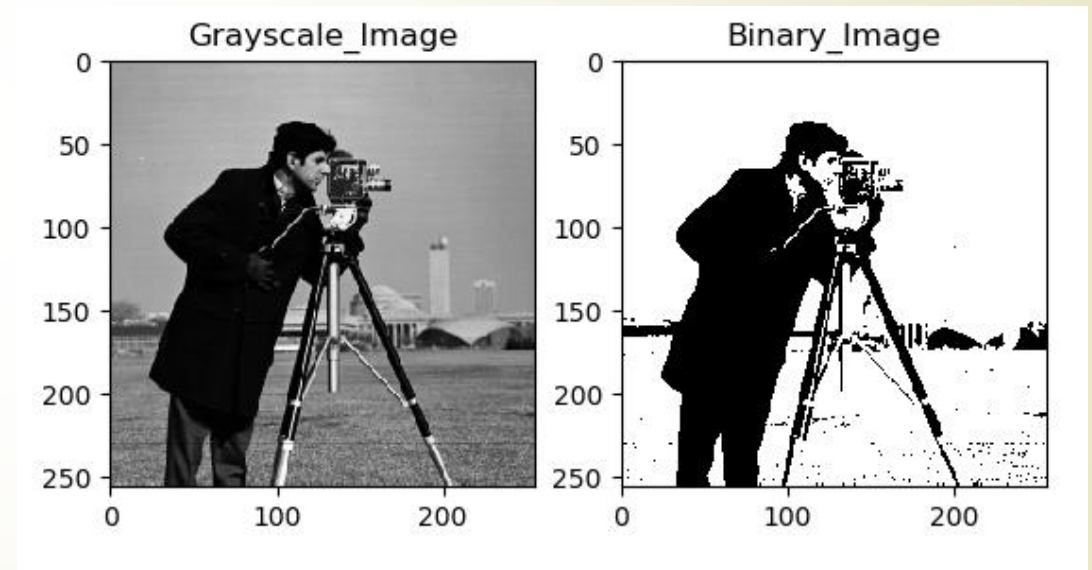
```
1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimg
3 import numpy as np
4 #-----
5 def rgb2gray(rgb):
6     r, g, b = rgb[:, :, 0], rgb[:, :, 1], rgb[:, :, 2]
7     gray = 0.2989 * r + 0.5870 * g + 0.1140 * b
8
9     return gray
10 #-----
11
12 RGB_img = mpimg.imread('football.jpg')
13
14 Grayscale_image=rgb2gray(RGB_img)
15 Grayscale_image=np.around(Grayscale_image)
16
17 plt.figure(1)
18
19 subplot_1=plt.subplot(1,2,1)
20 plt.imshow(RGB_img)
21 subplot_1.set_title('RGB Image')
22
23 subplot_1=plt.subplot(1,2,2)
24 plt.imshow(Grayscale_image,cmap="gray")
25 subplot_1.set_title('Grayscale_Image')
26 |
```



Binary image conversion

Για να μπορέσουμε να μετατρέψουμε μια grayscale εικόνα σε binary, πρέπει να ακολουθήσουμε την εξής διαδικασία:

```
1 import matplotlib.image as mpimg
2 from skimage.filters import threshold_otsu
3 import matplotlib.pyplot as plt
4
5 Grayscale_image = mpimg.imread('cameraman.tif')
6
7 thresh = threshold_otsu(Grayscale_image)
8 Binary_image = Grayscale_image > thresh
9 Binary_image = Binary_image.astype(int)
10
11 plt.figure(1)
12
13 subplot_1 = plt.subplot(1, 2, 1)
14 plt.imshow(Grayscale_image, cmap="gray")
15 subplot_1.set_title('Grayscale_Image')
16
17 subplot_2 = plt.subplot(1, 2, 2)
18 plt.imshow(Binary_image, cmap = plt.cm.gray)
19 subplot_2.set_title('Binary_Image')
20
```

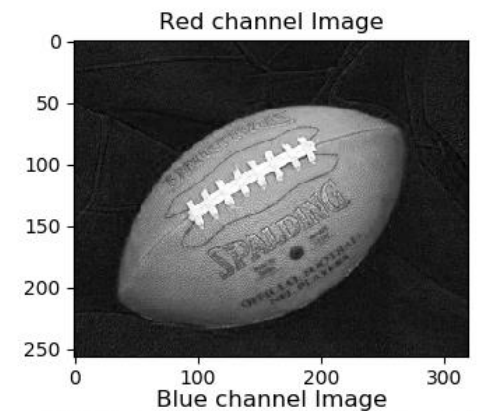
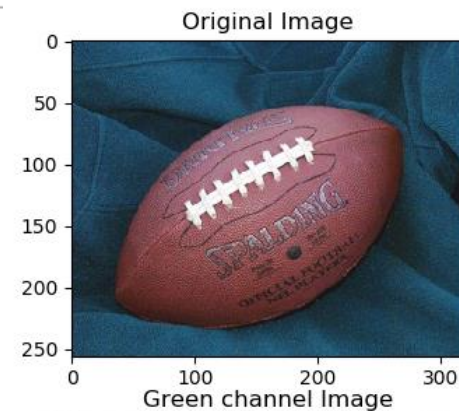


Images in Python

- ▶ Ορισμένες εικόνες, όπως οι εικόνες `truecolor`, απαιτούν μια τρισδιάστατη διάταξη πινάκων, όπου ο πρώτος πίνακας αντιπροσωπεύει τις εντάσεις κόκκινων εικονοστοιχείων, ο δεύτερος τις εντάσεις των πράσινων εικονοστοιχείων ο τρίτος αντιπροσωπεύει τις εντάσεις των μπλε εικονοστοιχείων.
- ▶ Αυτή η συνθήκη κάνει την εργασία με εικόνες σε Python παρόμοια με την εργασία με οποιοδήποτε άλλο τύπο δεδομένων `matrix`, και κάνει την πλήρη ισχύ της Python διαθέσιμη για εφαρμογές επεξεργασίας εικόνας.

Χρωματικά μοντέλα – Εξάσκηση Python DIP

```
1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimg
3
4 #-----main program-----
5
6 img = mpimg.imread('football.jpg')
7
8 r=img[:, :, 0]
9
10 g=img[:, :, 1]
11
12 b=img[:, :, 2]
13
14 fig1=plt.figure(1)
15
16 subplot1=fig1.add_subplot(2,2,1)
17 plt.imshow(img,cmap="gray")
18 subplot1.set_title('Original Image')
19
20 subplot1=fig1.add_subplot(2,2,2)
21 plt.imshow(r,cmap="gray")
22 subplot1.set_title('Red channel Image')
23
24 subplot1=fig1.add_subplot(2,2,3)
25 plt.imshow(g,cmap="gray")
26 subplot1.set_title('Green channel Image')
27
28 subplot1=fig1.add_subplot(2,2,4)
29 plt.imshow(b,cmap="gray")
30 subplot1.set_title('Blue channel Image')
31
```

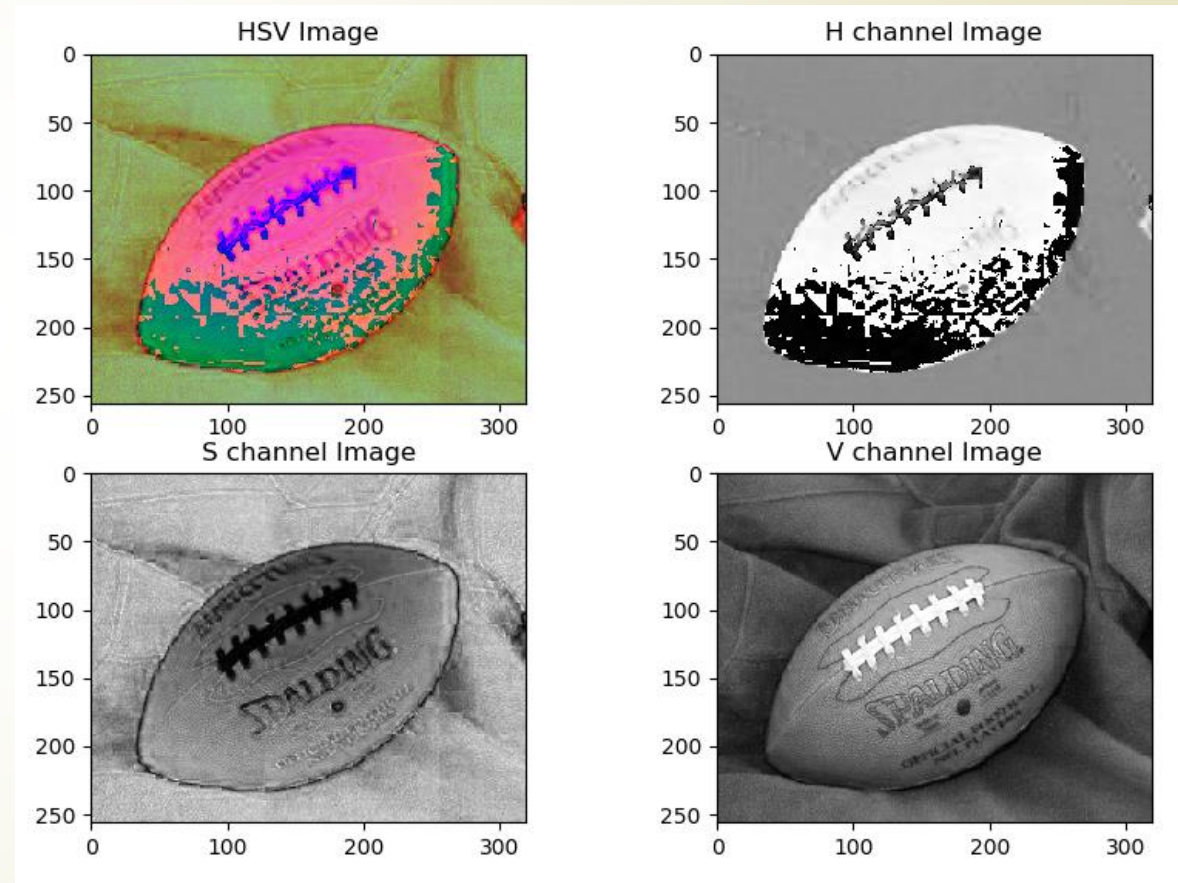


Χρωματικά μοντέλα – Εξάσκηση Python DIP

```

1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimg
3 from skimage.color import rgb2hsv
4
5 img = mpimg.imread('football.jpg',format=str)
6
7 img = rgb2hsv(img)
8
9 h=img[:, :,0]
10
11 s=img[:, :,1]
12
13 v=img[:, :,2]
14
15 fig1=plt.figure(1)
16 subplot1=fig1.add_subplot(2,2,1)
17 plt.imshow(img)
18 subplot1.set_title('HSV Image')
19
20 subplot1=fig1.add_subplot(2,2,2)
21 plt.imshow(h,cmap='gray')
22 subplot1.set_title('H channel Image')
23
24 subplot1=fig1.add_subplot(2,2,3)
25 plt.imshow(s,cmap='gray')
26 subplot1.set_title('S channel Image')
27
28 subplot1=fig1.add_subplot(2,2,4)
29 plt.imshow(v,cmap='gray')
30 subplot1.set_title('V channel Image')
31
32
33

```



Lab 1 Overview

► Create a script that will read the football.png and show a 2x2 subplot that includes:

1. The original RGB image.
 2. The HSV conversion of it.
 3. The grayscale conversion of it.
 4. The binary conversion of it.
- Then, save that FIGURE as png format.

