

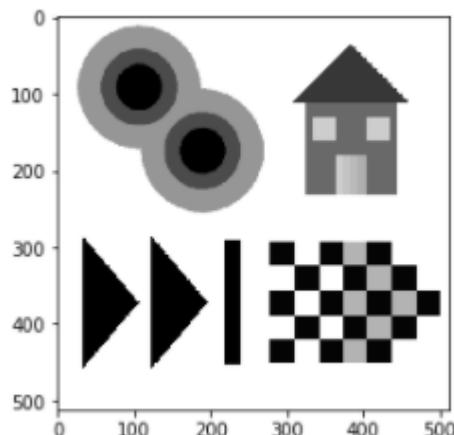
Στράτη Θεώνη Μαρία 03118186
Τσαρμποπούλου Μαργαρίτα Ελένη 03118848

ΜΑΘΗΜΑ: ΟΡΑΣΗ ΥΠΟΛΟΓΙΣΤΩΝ
Εξάμηνο: 8ο, Σχολή Ηλεκτρολόγων Μηχ. & Μηχ. Υπολογιστών ΕΜΠ
ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1
Εντοπισμός Σημείων Ενδιαφέροντος και Εξαγωγή Χαρακτηριστικών σε Εικόνες

Μέρος 1: Ανίχνευση Ακμών σε Γκρίζες Εικόνες

1.1. Δημιουργία Εικόνων Εισόδου

1.1.1) Διαβάζουμε την εικόνα με την cv2.imread και θέτουμε ως flag το cv2.IMREAD_GRAYSCALE, ώστε να πάρουμε ασπρόμαυρη εικόνα. Την προβάλλουμε μέσω της plt.imshow().

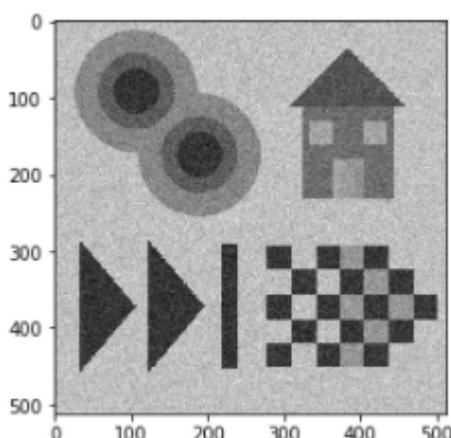


1.1.2) Φτιάχνουμε μία συνάρτηση psnr_function, η οποία δέχεται ως παράμετρο την τιμή που δώσαμε στο PSNR. Λύνουμε ως προς την τυπική απόκλιση:

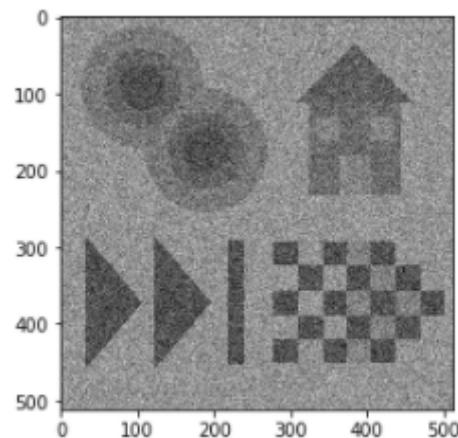
$$PSNR = 20 * \log_{10} \frac{I_{max}-I_{min}}{\sigma_n} \Rightarrow \sigma_n = \frac{I_{max}-I_{min}}{10^{\frac{PSNR}{20}}}$$

Για PSNR=20dB προέκυψε ση=0.1, ενώ για PSNR=10dB προέκυψε ση=0.31622776601683794

Για PSNR=20dB:



Για PSNR=10dB:



Η συνάρτηση επιστρέφει την τιμή της τυπικής απόκλισης, καθώς και το τελικό I για δική μας ευκολία στα επόμενα ερωτήματα.

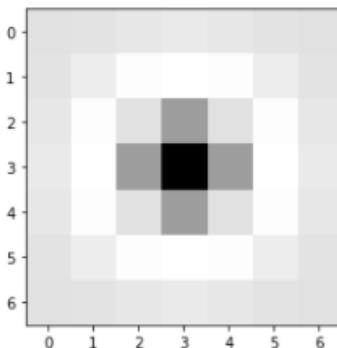
1.2. Υλοποίηση Αλγορίθμων Ανίχνευσης Ακμών

1.2.1) Υλοποιούμε δύο συναρτήσεις, η μία υπολογίζει την γκαουσιανή και άλλη τη λαπλασιανή της γκαουσιανής. Η πρώτη υλοποιείται μέσω της συνάρτησης cv2.getGaussianKernel, αφού πρώτα υπολογιστεί το μέγεθος του πυρήνα μέσω της np.ceil, και μέσω του πολλαπλασιασμού του γκαουσιανού πίνακα με τον transpose γκαουσιανό, ώστε να προκύψει νέος δισδιάστατος πχν γκαουσιανός πίνακας. Τη δεύτερη συνάρτηση την υλοποιήσαμε με τον πρώτο τρόπο, δηλαδή με τη βοήθεια της np.meshgrid:

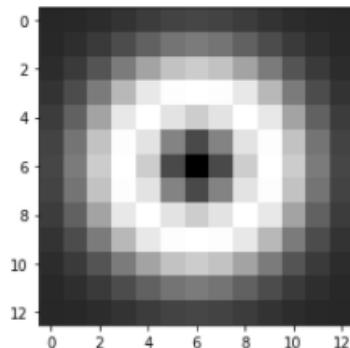
$$Gaussian = \frac{1}{2\pi\sigma^2} * e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$LoG = \frac{1}{2\pi\sigma^2} \left[\frac{x^2-\sigma^2}{\sigma^4} * e^{-\frac{x^2+y^2}{2\sigma^2}} + \frac{y^2-\sigma^2}{\sigma^4} * e^{-\frac{x^2+y^2}{2\sigma^2}} \right]$$

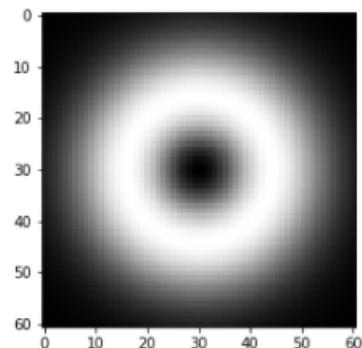
Παρακάτω δοκιμάσαμε διάφορες τιμές της τυπικής απόκλισης:



a) $\sigma_n = 1$



b) $\sigma_n = 2$



c) $\sigma_n = 10$

1.2.2)

i) Γραμμική συνέλιξη: $L_1 = \nabla^2(G_\sigma * I) = (\nabla^2 G_\sigma) * I = h * I$

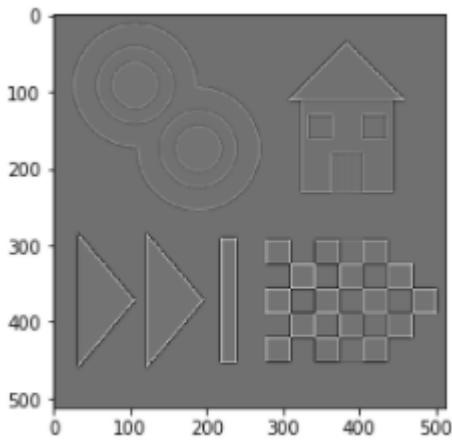
Χρησιμοποιούμε τη συνάρτηση cv2.filter2D, η οποία συνελίσει την αρχική εικόνα με την Laplacian-Of-Gaussian, με παράμετρο depth=-1 ώστε η νέα εικόνα να έχει ίδιο depth με τη νέα μετά την συνέλιξη.

$$B = \begin{array}{|c|c|c|} \hline & \bullet & \\ \hline \bullet & \bullet & \bullet \\ \hline & \bullet & \\ \hline \end{array}$$

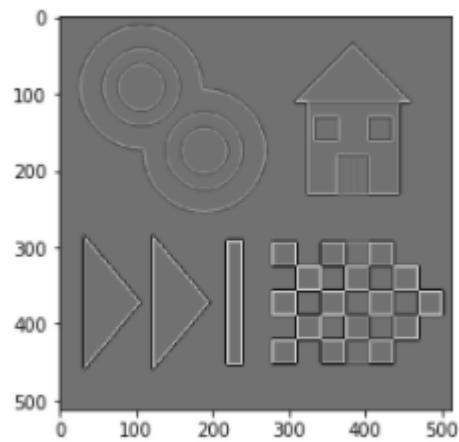
ii) Μη-γραμμική συνέλιξη: $L_2 = I_\sigma \oplus B + I_\sigma \ominus B - 2I_\sigma$,

Γνωρίζουμε ότι $I_\sigma \oplus B = dilation$ και $I_\sigma \ominus B = erosion$, όπου $I_\sigma(x, y) = G_\sigma * I(x, y)$

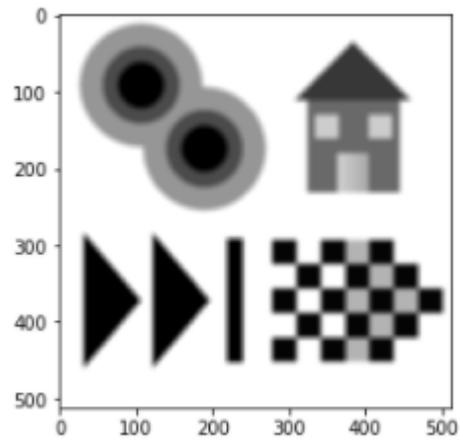
Επομένως, φτιάχνουμε την Iσ, καλώντας την συνάρτηση gaussian και την cv2.filter2D ώστε να συνελίξουμε τη γκαουσιανή με την εικόνα. Το Iσ που βρήκαμε το θέτουμε ως είσοδο στη συνάρτηση non_linear_convolution, προκειμένου να κάνουμε dilation, erosion και την τελική πράξη (dilation+erosion-2*image) με τον πίνακα B να έχει μηδενικά στις κενές θέσεις του σχήματος και άσσους στις υπόλοιπες. Για ση=1 πήραμε τις εξής εικόνες:



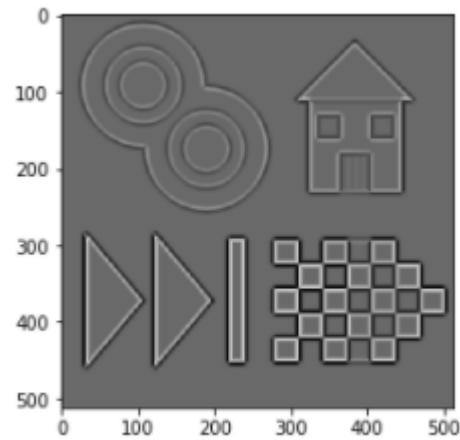
i) Γραμμική συνέλιξη για $\sigma n=1$



ii) Μη-γραμμική συνέλιξη για $\sigma n=1$



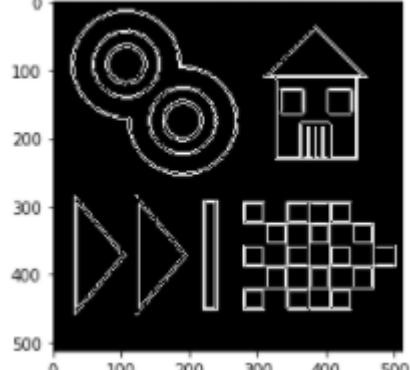
i) Γραμμική συνέλιξη για $\sigma n=2$



ii) Μη-γραμμική συνέλιξη για $\sigma n=2$

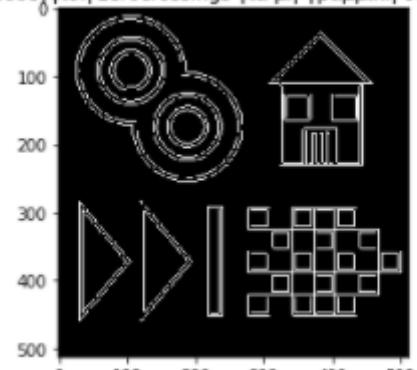
1.2.3) Υλοποιήσαμε μία συνάρτηση algorithm, η οποία καλεί την cv2.threshold, που μετατρέπει μία εικόνα στην αντίστοιχη binary της από 0 έως 1 (δηλαδή ολόκληρη την κανονικοποιημένη μορφή της εικόνας).

Προσέγγιση zerocrossings για γραμμική συνέλιξη



$\sigma n=1$

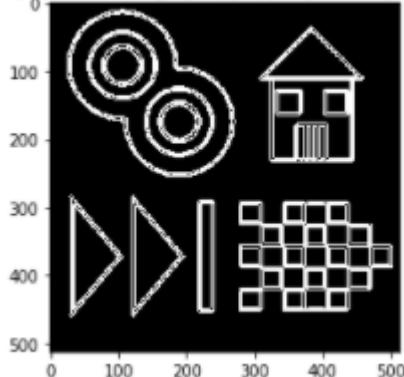
Προσέγγιση zerocrossings για μη-γραμμική συνέλιξη



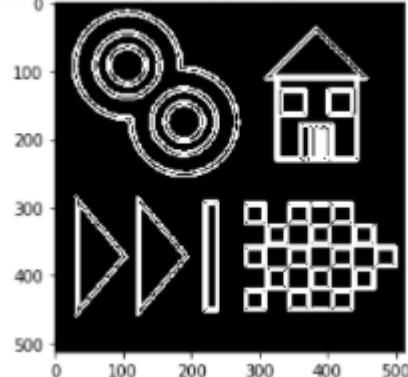
$\sigma n=2$

1.2.4)

Απόρριψη zero-crossings για γραμμική συνέλιξη και $\theta=0.2$

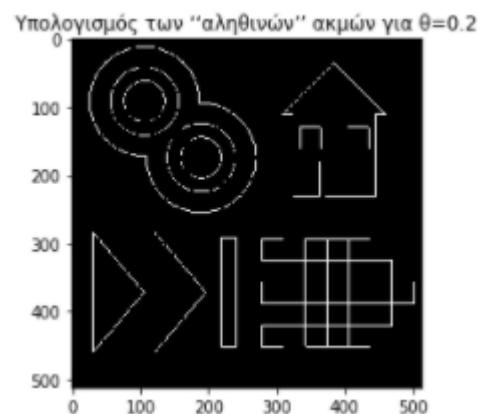
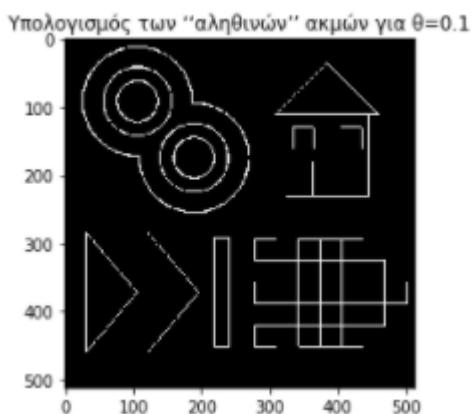


Απόρριψη zero-crossings για μη-γραμμική συνέλιξη και $\theta=0.2$



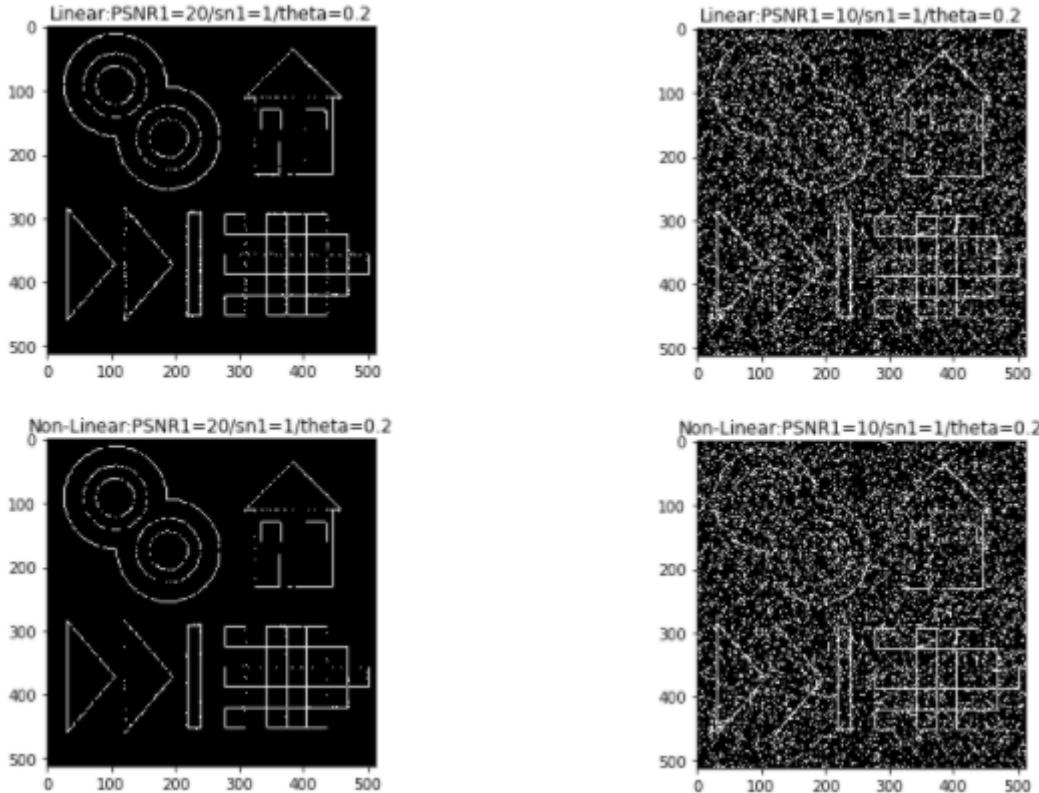
Η συνάρτηση EdgeDetect απαρτίζεται από τις συναρτήσεις των παραπάνω ερωτημάτων. Συγκεκριμένα, παίρνει ως ορίσματα την εικόνα, την τυπική απόκλιση, την θedge και ένα flag το οποίο αν είναι 1, εκτελούμε γραμμική συνέλιξη, αν είναι 0 εκτελούμε μη γραμμική συνέλιξη και για κάθε περίπτωση καλούμε την αντίστοιχη συνάρτηση (linear_convolution ή no_linear_convolution). Στη συνέχεια, προσεγγίζουμε τα zero-crossings με τη συνάρτηση algorithm, ώστε τελικά να τα απορρίψουμε μέσω της συνάρτησης no_zero_crossings. Προκύπτουν οι ίδιες εικόνες με πριν, καθώς ακολουθήσαμε τα ίδια βήματα.

1.3.1) Βρίσκουμε τις πραγματικές ακμές, εκτελώντας τη δοθείσα πράξη $M = (I_0 \oplus B) - (I_0 \ominus B)$, δηλαδή dilation-erosion. Κάνουμε δυαδική την εικόνα M και την κατωφλιοποιούμε, ώστε να κρατάμε δεδομένα από θedge έως 1. Παρακάτω απεικονίζουμε την εικόνα που προκύπτει για $\theta=0.1$ και $\theta=0.2$:



1.3.2) Υπολογίζουμε το κριτήριο ποιότητας του αποτελέσματος ανίχνευσης για PSNR=20 και βρίσκουμε ότι κατά την γραμμική συνέλιξη, αυτό ανέρχεται σε 0.8968864221969028, ενώ στην μη-γραμμική συνέλιξη σε 0.9274182633342818.

1.3.3) Ενδεικτικά:



Δοκιμάζουμε διάφορους συνδυασμούς για να βρούμε τον πιο αποτελεσματικό, εκτελώντας εμφωλευμένα for. Το πρώτο for δίνει στο PSNR τις τιμές {10,20}, το δεύτερο δίνει στο θedge τιμές από 0.1 έως και 1 με βήμα 0.1 και το τρίτο δίνει στο ση τιμές από 1 έως και 5 με βήμα 0.5. Σε αυτό το βρόχο επανάληψης κρατάμε μία μεταβλητή cmax, που δίνει το μέγιστο c μέχρι εκείνη τη στιγμή, και τσεκάρουμε αν το συγκεκριμένο cmax προέρχεται από το κριτήριο ποιότητας αποτελέσματος ανίχνευσης γραμμικής ή μη γραμμικής συνέλιξης. Από τους παραπάνω συνδυασμούς προκύπτει ότι:

Maximum C is equal to 0.9790916268947618 for PSNR=20, sn =1.0, theta=0.4 - nonlinear convolution

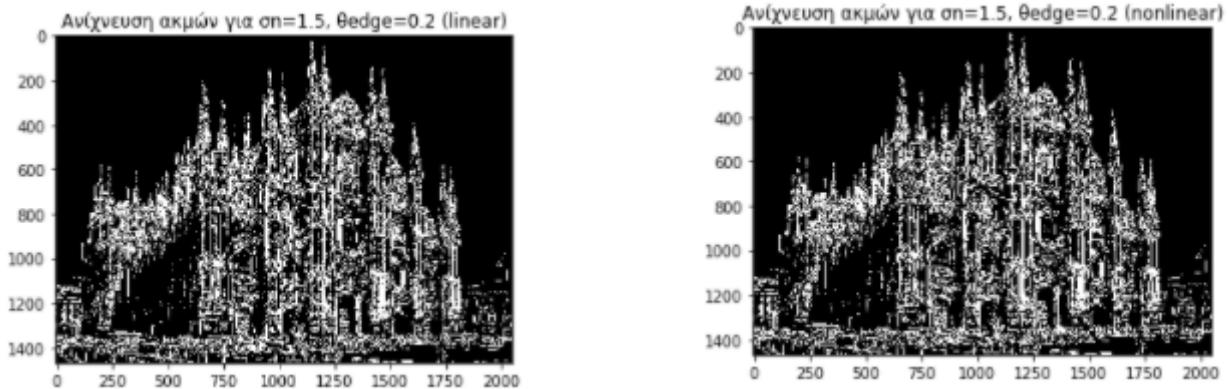
Άρα η καλύτερη περίπτωση είναι **C=0.9790916268947618**, το οποίο προκύπτει για **PSNR=20, sn=1, θedge=0.4-μη γραμμική συνέλιξη**.

Είναι εμφανές αρχικά ότι η μη γραμμική συνέλιξη είναι καλύτερη από την γραμμική. Για μικρές τιμές του σ ο θόρυβος που υπάρχει στις εικόνες δεν εξαφανίζεται και για αυτό εντοπίζονται πολλές ακμές που δεν υπάρχουν στην αρχική μας εικόνα, ενώ για το θedge παρατηρούμε πως για πολύ μικρές τιμές ο αλγόριθμος διατηρεί τα zerocrossings που βρίσκονται στις ομοιόμορφες περιοχές και εντοπίζονται περισσότερες ακμές από όσες υπάρχουν πραγματικά. Αυξάνοντας το σ ο θόρυβος χάνεται, αλλά με μεγάλη αύξηση τού θολώνουν οι ακμές της εικόνας και χάνονται οι λεπτομέρειές τους, ενώ αυξάνοντας την τιμή του θedge εξαλείφονται τα zerocrossings στις ομοιόμορφες περιοχές αλλά για μεγάλες τιμές απορρίπτονται και πραγματικές ακμές της εικόνας.

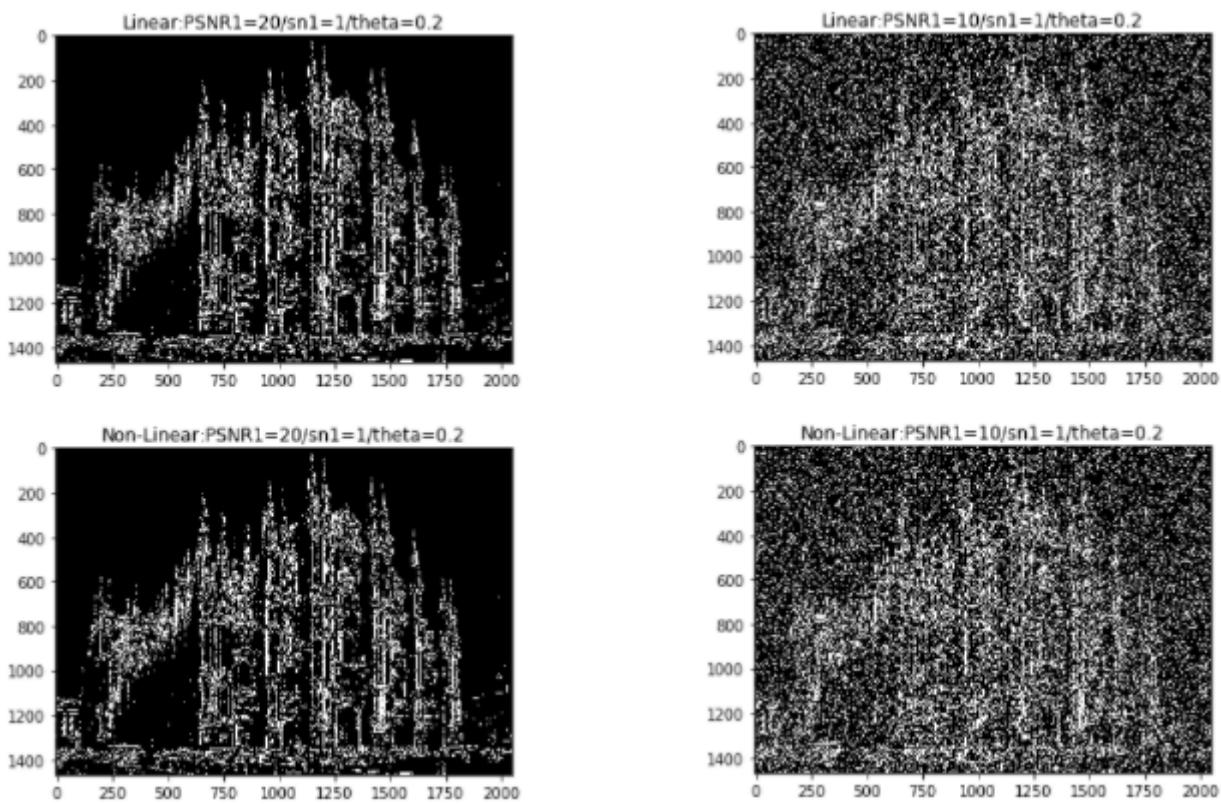
1.4.1)



Original φωτογραφία



1.4.2)



Linear: $C = 0.7306376741520284$

Non-Linear: $C = 0.7393286201594148$

Linear: $C = 0.4635202590406686$

Non-Linear: $C = 0.47086889603630744$

Καλούμε πάλι τη συνάρτηση που φτιάξαμε στο 1.3.3 και ομοίως βρίσκουμε:

Maximum C is equal to 0.7385760199744862 for PSNR=20, sn =1.0, theta=0.2 - nonlinear convolution

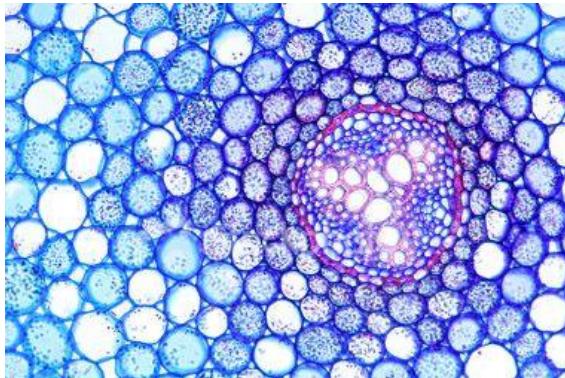
Μέρος 2: Ανίχνευση Σημείων Ενδιαφέροντος

2.1. Ανίχνευση Γωνιών

Μας δίνονται 3 εικόνες εισόδου, στις οποίες καλούμαστε να ανιχνεύσουμε τις γωνίες τους, τις οποίες παραθέτω παρακάτω.



duomo_edges



cells



donuts

2.1.1) Υπολογίζουμε για όλα τα ρίξει της εικόνας (αφού πρώτα την μετατρέψουμε σε grayscale) τα στοιχεία J_1 , J_2 , J_3 , του δομικού τανυστή J με τον τύπο που φαίνεται στη συνέχεια.

$$J_1(x, y) = G_\rho * \left(\frac{\partial I_\sigma}{\partial x} \cdot \frac{\partial I_\sigma}{\partial x} \right) (x, y)$$

$$J_2(x, y) = G_\rho * \left(\frac{\partial I_\sigma}{\partial x} \cdot \frac{\partial I_\sigma}{\partial y} \right) (x, y)$$

$$J_3(x, y) = G_\rho * \left(\frac{\partial I_\sigma}{\partial y} \cdot \frac{\partial I_\sigma}{\partial y} \right) (x, y)$$

2.1.2) Έπειτα υπολογίζουμε τις ιδιοτιμές του J λ+, λ- σύμφωνα με τη σχέση που φαίνεται παρακάτω.

$$\lambda_{\pm}(x, y) = \frac{1}{2} \left(J_1 + J_3 \pm \sqrt{(J_1 - J_3)^2 + 4J_2^2} \right)$$

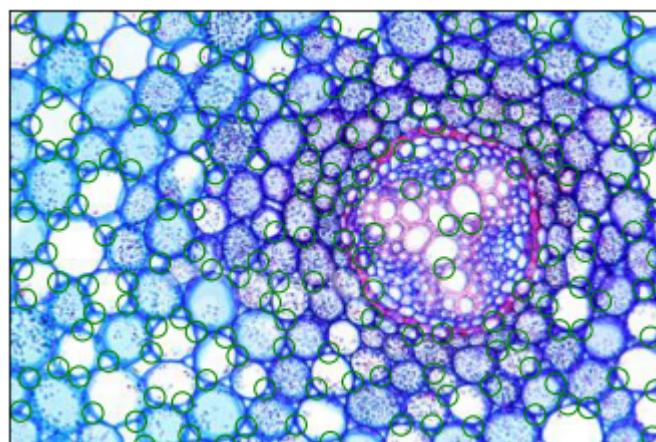
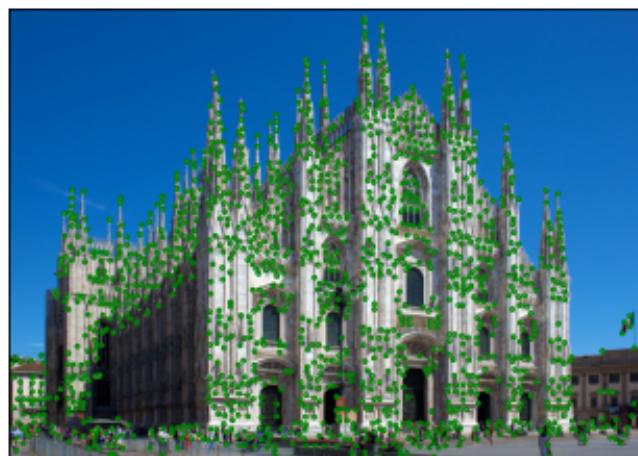
Παρατηρούμε ότι η λ+ έχει υψηλές τιμές στις ακμές και στις γωνίες, ενώ η λ- έχει υψηλή τιμή κυρίως στις γωνίες.

2.1.3) Με βάση τις υπολογισμένες ιδιοτιμές εξάγουμε το κριτήριο γωνιότητας που φαίνεται παρακάτω:

$$R(x, y) = \lambda_- \lambda_+ - k \cdot (\lambda_- + \lambda_+)^2$$

Τελικά επιλέξαμε τα σημεία που είναι τοπικά μέγιστα του R εντός τετραγωνικών παραθύρων και ξεπερνούν ένα κατώφλι $\theta_{corn} * R_{max}$ και αναπαρηστούμε την εικόνα με τις γωνίες με την συνάρτηση που μας δόθηκε. Ενδεικτικές τιμές για τις παραμέτρους που χρησιμοποιήσαμε για το παρακάτω αποτέλεσμα:
 $\sigma = 2$, $\rho = 2$, $k = 0.007$, $\theta_{corn} = 0.005$, $k=0.05$

Οι γωνίες που ανιχνεύτηκαν για τις παραπάνω παραμέτρους:



Στα παρακάτω η ανάλυση θα γίνει μόνο για τις εικόνες cells, donuts.

2.2. Πολυκλιμακωτή Ανίχνευση Γωνιών

2.2.1) Αρχικά υπολογίσαμε τις κλίμακες ολοκλήρωσης και διαφόρισης με τον παρακάτω τρόπο.

$$\sigma_0, \sigma_1, \dots, \sigma_{N-1} = s^0 \sigma_0, s^1 \sigma_0, \dots, s^{N-1} \sigma_0$$

$$\rho_0, \rho_1, \dots, \rho_{N-1} = s^0 \rho_0, s^1 \rho_0, \dots, s^{N-1} \rho_0$$

2.2.2) Ύστερα υπολογίσαμε την κανονικοποιημένη LoG για κάθε κλίμακα σι με βάση τον παρακάτω τύπο.

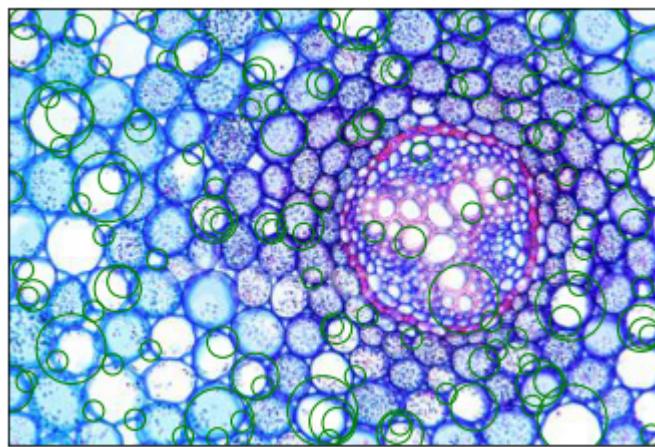
$$|LoG(\mathbf{x}, \sigma_i)| = \sigma_i^2 |L_{xx}(\mathbf{x}, \sigma_i) + L_{yy}(\mathbf{x}, \sigma_i)|, \quad i = 0, \dots, N - 1$$

Παρακάτω αναπαρηστούμε τις γωνίες που ανιχνεύτηκαν για 4 κλίμακες με και μεγιστοποιούν την LoG μετρική σε μια γειτονιά 2 διαδοχικών κλιμάκων.

Οι παράμετροι για το παρακάτω αποτέλεσμα είναι οι εξής:

$N = 4$, $k = 0.05$, $\sigma_0 = 2$, $\rho_0 = 2.5$, $s = 1.5$, $\theta_{corn} = 0.005$.

Οι γωνίες που ανιχνεύτηκαν για τις 4 κλίμακες με τις παραπάνω παραμέτρους:



2.3. Ανίχνευση Blobs

Σε αυτό το ερώτημα ανιχνεύουμε Blobs στην ίδια εικόνα με τρόπο που θα εξηγήσουμε παρακάτω. Αρχικά υπολογίζουμε τον πίνακα Hessian για κάθε pixel της εικόνας όπως φαίνεται παρακάτω.

$$H(x, y) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}$$

$$\text{όπου } L_{xx}(x, y, \sigma) = \frac{\partial^2}{\partial x^2}\{I_\sigma(x, y)\}, L_{yy}(x, y, \sigma) = \frac{\partial^2}{\partial y^2}\{I_\sigma(x, y)\} \text{ και } L_{xy}(x, y, \sigma) = \frac{\partial^2}{\partial x \partial y}\{I_\sigma(x, y)\}.$$

Το κριτήριο που χρησιμοποιήσαμε για την ανίχνευση Blobs είναι το παρακάτω:

$$R(x, y) = \det(H(x, y))$$

Θεωρούμε ως Blobs τα σημεία που είναι τοπικά μέγιστα και έχουν μεγαλύτερη τιμή από ένα ορισμένο κατώφλι. Παρακάτω τα αναπαρηστούμε και οι παράμετροι που χρησιμοποιήσαμε για να πετύχουμε αυτό το

αποτέλεσμα είναι οι εξής: $\sigma = 2$, $\theta_{corn} = 0.005$, $k=0.05$.

Τα Blobs που ανιχνεύτηκαν με την παραπάνω μέθοδο και παραμέτρους:

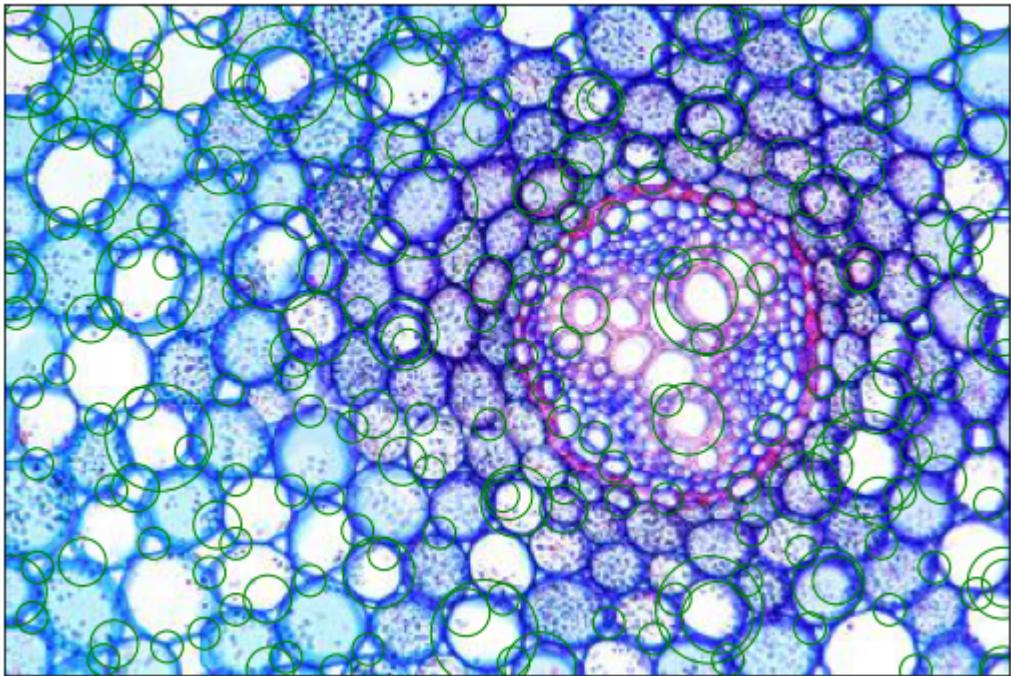


2.4. Πολυκλιμακωτή Ανίχνευση Blob

Υπολογίζουμε τα Blobs για πολλές κλίμακες με τη μέθοδο που περιγράφηκε παραπάνω για την ανίχνευση γωνιών σε πολλές κλίμακες. Οι παράμετροι που χρησιμοποιήσαμε είναι οι εξής:

$N = 4$, $\sigma_0 = 2$, $s = 1.5$, $\theta_{corn} = 0.005$, $k=0.005$.

Τα Blobs που ανιχνεύτηκαν με την παραπάνω μέθοδο και παραμέτρους για τις 4 κλίμακες:

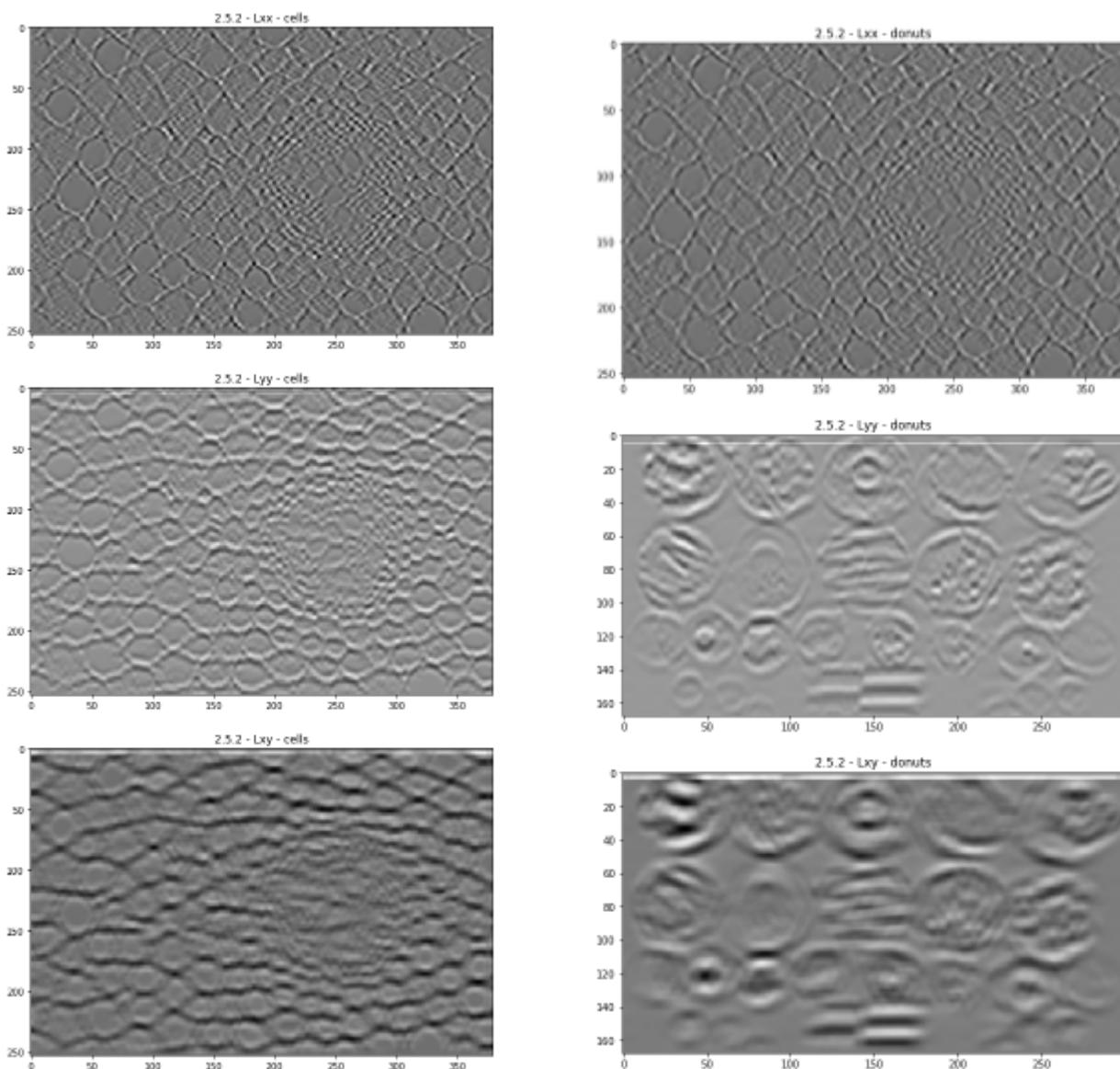


2.5.1 & 2.5.2 & 2.5.3 & 2.5.4)

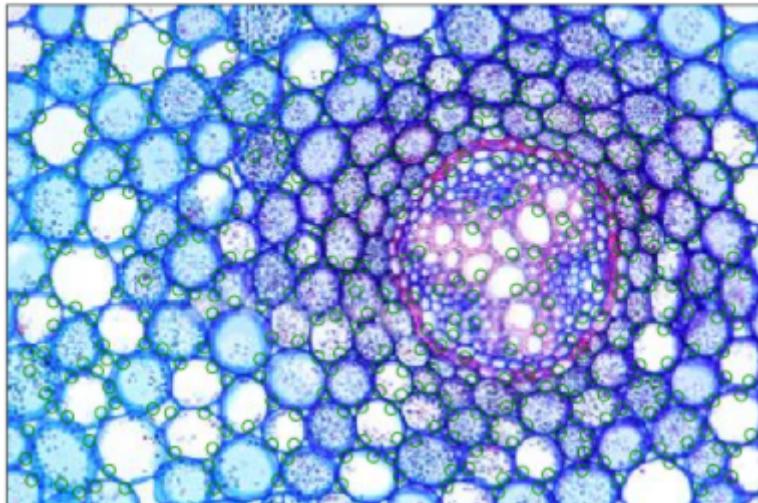
Προκειμένου να εξοικονομηθεί χρόνος, χρησιμοποιούμε μια προσέγγιση των στοιχείων του Hessian πίνακα και θα προτιμήσουμε αντί των τριών συνελίξεων να φιλτράρουμε την εικόνα με ένα γκαουσιανό φίλτρο. Προσεγγίζουμε τα φίλτρα με έτοιμα box filters, που αποτελούνται από ορθογώνια παράθυρα ίδιου μεγέθους για κάθε φίλτρο, και μετά δημιουργούμε την ολοκληρωτική εικόνα. Υπολογίζουμε τη συνέλιξη ενός από τα παράθυρα, αφού όλα έχουν το ίδιο μέγεθος, και τα υπόλοιπα τα μετατοπίζουμε μέσω της `np.roll`. Όπως διευκρινίζεται και στην εκφώνηση, με τη χρήση της ολοκληρωτικής εικόνας οποιοδήποτε άθροισμα εντός ενός δοθέντος παραθύρου πάνω στην αρχική εικόνα μπορεί να υπολογιστεί άμεσα και έτσι χρησιμοποιούμε τον τύπο αθροίσματος $S = A + C - B - D$, όπου A, B, C, D μετατοπισμένα με βάση το κέντρο (xf, yf) του πυρήνα. Έχοντας υπολογίσει τα παραπάνω, προχωράμε στην υλοποίηση της συνάρτησης boxfiltering, η οποία υπολογίζει το κριτήριο R , σύμφωνα με την εξίσωση:

$$R(x, y) = L_{xx}(x, y)L_{yy}(x, y) - (0.9L_{xy}(x, y))^2 \quad \text{.} \quad \text{Τέλος, η multiboxfiltering υλοποιήθηκε με βάση τις προηγούμενες πολυκλιμακωτές συναρτήσεις.}$$

Παρακάτω, παραθέτουμε τα plots που προέκυψαν στα παραπάνω ερωτήματα:



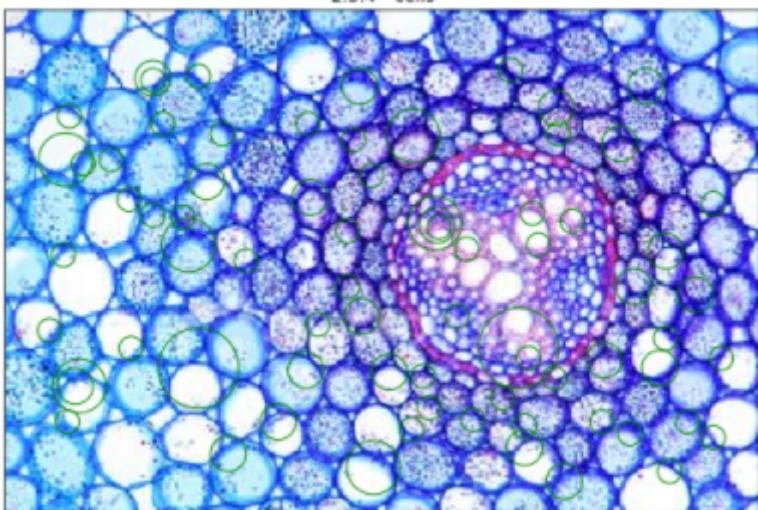
2.5.3 - cells



2.5.3 - donuts



2.5.4 - cells





Μέρος 3: Εφαρμογές σε Ταίριασμα και Κατηγοριοποίηση Εικόνων με Χρήση Τοπικών Περιγραφητών στα Σημεία Ενδιαφέροντος

3.2. Ταίριασμα Εικόνων υπό Περιστροφή και Αλλαγή Κλίμακας

Σε αυτό το ερώτημα υλοποιήσαμε τη συνάρτηση matching_evaluation. Προσαρμόσαμε τις συναρτήσεις των ανιχνευτών που είχαμε υλοποιήσει στο πρώτο μέρος, καθώς η μεταβλητή img_Grey δε χρειαζόταν και ο ορισμός της μας εμπόδιζε στον υπολογισμό της συνάρτησης matching_evaluation.

```

Avg. Scale Error new_anixnefsi_gonion SURF for Image 1: 0.002
Avg. Theta Error new_anixnefsi_gonion SURF for Image 1: 0.152
Avg. Scale Error new_anixnefsi_gonion SURF for Image 2: 0.001
Avg. Theta Error new_anixnefsi_gonion SURF for Image 2: 0.208
Avg. Scale Error new_anixnefsi_gonion SURF for Image 3: 0.015
Avg. Theta Error new_anixnefsi_gonion SURF for Image 3: 2.465
Avg. Scale Error new_HarrisStephens SURF for Image 1: 0.002
Avg. Theta Error new_HarrisStephens SURF for Image 1: 0.204
Avg. Scale Error new_HarrisStephens SURF for Image 2: 0.005
Avg. Theta Error new_HarrisStephens SURF for Image 2: 0.244
Avg. Scale Error new_HarrisStephens SURF for Image 3: 0.002
Avg. Theta Error new_HarrisStephens SURF for Image 3: 0.178
Avg. Scale Error new_blob_detect SURF for Image 1: 0.027
Avg. Theta Error new_blob_detect SURF for Image 1: 7.759
Avg. Scale Error new_blob_detect SURF for Image 2: 0.010
Avg. Theta Error new_blob_detect SURF for Image 2: 0.229
Avg. Scale Error new_blob_detect SURF for Image 3: 0.001
Avg. Theta Error new_blob_detect SURF for Image 3: 0.054
Avg. Scale Error new_multi_blob_detect SURF for Image 1: 0.002
Avg. Theta Error new_multi_blob_detect SURF for Image 1: 0.101
Avg. Scale Error new_multi_blob_detect SURF for Image 2: 0.003
Avg. Theta Error new_multi_blob_detect SURF for Image 2: 0.115
Avg. Scale Error new_multi_blob_detect SURF for Image 3: 0.001
Avg. Theta Error new_multi_blob_detect SURF for Image 3: 0.067
Avg. Scale Error newmultiboxfiltering SURF for Image 1: 0.376
Avg. Theta Error newmultiboxfiltering SURF for Image 1: 23.422
Avg. Scale Error newmultiboxfiltering SURF for Image 2: 0.918
Avg. Theta Error newmultiboxfiltering SURF for Image 2: 31.496
Avg. Scale Error newmultiboxfiltering SURF for Image 3: 0.426
Avg. Theta Error newmultiboxfiltering SURF for Image 3: 16.030

```

```

Avg. Scale Error new_anixnefsi_gonion HOG for Image 1: 0.184
Avg. Theta Error new_anixnefsi_gonion HOG for Image 1: 15.962
Avg. Scale Error new_anixnefsi_gonion HOG for Image 2: 0.493
Avg. Theta Error new_anixnefsi_gonion HOG for Image 2: 19.476
Avg. Scale Error new_anixnefsi_gonion HOG for Image 3: 0.453
Avg. Theta Error new_anixnefsi_gonion HOG for Image 3: 33.304
Avg. Scale Error new_HarrisStephens HOG for Image 1: 0.184
Avg. Theta Error new_HarrisStephens HOG for Image 1: 15.439
Avg. Scale Error new_HarrisStephens HOG for Image 2: 2.300
Avg. Theta Error new_HarrisStephens HOG for Image 2: 28.561
Avg. Scale Error new_HarrisStephens HOG for Image 3: 0.167
Avg. Theta Error new_HarrisStephens HOG for Image 3: 15.452
Avg. Scale Error new_blob_detect HOG for Image 1: 0.186
Avg. Theta Error new_blob_detect HOG for Image 1: 7.231
Avg. Scale Error new_blob_detect HOG for Image 2: 0.100
Avg. Theta Error new_blob_detect HOG for Image 2: 13.674
Avg. Scale Error new_blob_detect HOG for Image 3: 0.154
Avg. Theta Error new_blob_detect HOG for Image 3: 27.219
Avg. Scale Error new_multi_blob_detect HOG for Image 1: 0.141
Avg. Theta Error new_multi_blob_detect HOG for Image 1: 9.503
Avg. Scale Error new_multi_blob_detect HOG for Image 2: 0.279
Avg. Theta Error new_multi_blob_detect HOG for Image 2: 16.918
Avg. Scale Error new_multi_blob_detect HOG for Image 3: 0.195
Avg. Theta Error new_multi_blob_detect HOG for Image 3: 17.321
Avg. Scale Error newmultiboxfiltering HOG for Image 1: 0.390
Avg. Theta Error newmultiboxfiltering HOG for Image 1: 41.429
Avg. Scale Error newmultiboxfiltering HOG for Image 2: 0.917
Avg. Theta Error newmultiboxfiltering HOG for Image 2: 35.223
Avg. Scale Error newmultiboxfiltering HOG for Image 3: 0.382
Avg. Theta Error newmultiboxfiltering HOG for Image 3: 28.070

```

Παρατηρούμε ότι η SURF δίνει μικρότερο avg. scale/theta error σε σχέση με τη HOG, άρα δίνει καλύτερη εκτίμηση περιστροφής και κλίμακας των εικόνων. Επίσης, οι πολυκλιμακώτες μέθοδοι δίνουν καλύτερα αποτελέσματα σε σχέση με τις μονοκλιμακώτες, καθώς εντοπίζουν με μεγάλη ακρίβεια σημεία ενδιαφέροντος και έτσι οι περιγραφητές μπορούν καλύτερα να εκτιμήσουν τα πέριξ των σημειών αυτών.

3.2. Κατηγοριοποίηση Εικόνων

Στο κομμάτι αυτό της εργασίας πραγματοποιήσαμε κατηγοριοποίηση εικόνων σε σύνολα ανάλογα με τα χαρακτηριστικά τους και ύστερα θα αξιολογήσουμε την επίδοση του συστήματος για τους διάφορους ανιχνευτές (Harris-Laplace, Blob Detector, Box Filter) και περιγραφητές (SURF, HOG).

Για να το επιτύχουμε αυτό αρχικά θα χρειαστήκαμε τα σημεία ενδιαφέροντος της κάθε εικόνας και με χρήση τους στον εκάστοτε περιγραφητή εξάγαμε τα χαρακτηριστικά της. Χρησιμοποιώντας τους ανιχνευτές που υλοποιήσαμε στο προηγούμενο μέρος της εργασίας μαζί με τη συνάρτηση FeatureExtraction που μας δίνεται καταφέραμε να εξάγουμε τα χαρακτηριστικά της κάθε εικόνας που μας δίνεται στη βάση.

Μετά χωρίσαμε τις εικόνες που μας δόθηκαν σε δύο σύνολα, ένα για να εκπαιδεύσουμε το σύστημα μας και ένα για να πραγματοποιήσουμε την αξιολόγησή του, το οποίο κάναμε με την createTrainTest, η οποία δημιούργησε και ετικέτες της κατηγορίας στην οποία ανήκει η κάθε εικόνα.

Κατηγοριοποιήσαμε τις εικόνες με την τεχνική Bag of Visual Words καθώς βρήκαμε ένα συνολικό διάνυσμα χαρακτηριστικών με την ίδια διάσταση για κάθε εικόνα υπολογίζοντας τα ιστογράμματα εμφάνισης των τοπικών χαρακτηριστικών στο σύνολο της εικόνας.

Αρχικά έχουμε τα χαρακτηριστικά της κάθε εικόνας όπως αυτά προέκυψαν από τον ανιχνευτή που χρησιμοποιήσαμε . Συλλέξαμε τα εξαγόμενα χαρακτηριστικά των εικόνων των συνόλου εκπαίδευσης και αξιολόγησης και τα τοποθετήσαμε σε έναν ενιαίο πίνακα. Μετά με την συναρτηση KMeans, η οποία ομαδοποιεί τα χαρακτηριστικά που εντοπίστηκαν, δημιουργήσαμε τα Visual Words που θα χρησιμοποιηθούν, επιλέγοντας 500 'λέξεις'. Στη συνέχεια, για κάθε εικόνα των συνόλων δημιουργήσαμε το ιστόγραμμα συχνότητας εμφάνισης κάθε στοιχείου του λεξικού μας, αντιστοιχώντας κάθε χαρακτηριστικό της εικόνας στο Visual Word που βρίσκεται στην ελάχιστη ευκλείδεια απόσταση από αυτό. Τέλος, κανονικοποιήσαμε τα αποτελέσματα που προέκυψαν διαιρώντας με την L2 νόρμα των ιστογραμμάτων.

Στη συνέχεια τα ιστογράμματα των εικόνων του συνόλου εκπαίδευσης τροφοδοτούνται σε ένα Support Vector Machine (SVM), το οποίο και εκπαιδεύεται με αυτά. Στο SVM αυτό δίνονται ύστερα και τα ιστογράμματα των εικόνων του συνόλου αξιολόγησης για να τα κατηγοριοποιήσει κατάλληλα σύμφωνα με την εγγύτητά τους στα κέντρα των τριών κλάσεων όπως αυτά προέκυψαν από την εκπαίδευση. Από την κατηγοριοποίηση αυτή προκύπτουν και τα ποσοστά αναγνώρισης τα οποία θα χρησιμοποιήσουμε ως μετρική σύγκρισης παρακάτω.

Το πείραμα πραγματοποιήθηκε για:

Harris-Laplace: s=1.5, N=4, σ=2, ρ=2.5, θ=0.005, k=0.05

Blob Detector: s=2, N=4, σ=2, θ=0.005, k= 0.05

Box Filter: s=1.5, N=4, σ=2, θ=0.005

Οι ανιχνευτές αυτοί χρησιμοποιήθηκαν και για τους 2 περιγραφητές SURF και HOG και τα αποτελέσματα που προέκυψαν είναι τα εξής:

Time for feature extraction: 176.613

Mean accuracy for **new_harrislap** with SURF descriptors: 56.552%

Time for feature extraction: 185.085

Mean accuracy for **new_harrislap** with HOG descriptors: 60.276%

Time for feature extraction: 536.616

Mean accuracy for **new_multi_blob_detect** with SURF descriptors: 57.379%

Time for feature extraction: 559.660

Mean accuracy for **new_multi_blob_detect** with HOG descriptors: 64.690%

Time for feature extraction: 253.058

Mean accuracy for **newmultiboxfiltering** with SURF descriptors: 56.414%

Time for feature extraction: 263.578

Mean accuracy for **newmultiboxfiltering** with HOG descriptors: 64.138%

Τα αποτελέσματα αυτά μας δίνουν μια εικόνα για τον χρόνο που χρειάστηκε η εξαγωγή χαρακτηριστικών για τους διάφορους ανιχνευτές και περιγραφητές που χρησιμοποιήσαμε όπως και τα ποσοστά επιτυχίας της αναγνώρισης για τον κάθε συνδυασμό.

Από τα αποτελέσματα που προέκυψαν βλέπουμε ότι δεν υπάρχει σημαντική διαφορά στον χρόνο εντοπισμού χαρακτηριστικών μεταξύ των διαφόρων ανιχνευτών και περιγραφητών.

Αναφορικά με τα αποτελέσματα της κατηγοριοποίησης παρατηρούμε πως με τη χρήση του HOG descriptor έχουμε εμφανώς καλύτερα αποτελέσματα συγκριτικά με τον SURF (60- 70% για HOG έναντι 50-60% για SURF), το οποίο μπορεί να οφείλεται τόσο στις μικρές περιστροφικές μεταβολές των χαρακτηριστικών των εικόνων αλλά και στην ανθεκτικότητα του περιγραφητή HOG στις μεταβολές φωτεινότητας σε σχέση με τον SURF τις οποίες και παρατηρούμε στις εικόνες μας. Επιπλέον, βλέποντας και τους χρόνους εκτέλεσης

αντιλαμβανόμαστε πως ο περιγραφητής HOG είναι η καλύτερη επιλογή για το δεδομένο σύνολο εικόνων.

Σχετικά με τους ανιχνευτές παρατηρούμε πως ο harrislap υστερεί σε σχέση με τους άλλους δύο (περίπου 60% για HOG έναντι των περίπου 65% για τους άλλους δύο ανιχνευτές) χωρίς να παρατηρούμε κάποια αξιοσημείωτη επιτάχυνση στη διαδικασία. Από τα προηγούμενα εργαστηριακά μέρη ωστόσο γνωρίζουμε πως σε μεγαλύτερες τιμές του σ και πολυπληθέστερα σύνολα ίσως να βλέπαμε μεγαλύτερη διαφορά στο χρόνο εκτέλεσης και το αποτέλεσμα να δικαιολογούσε τη χρήση του. Ο καλύτερος ανιχνευτής για το σύνολο εικόνων που μας δόθηκε φαίνεται να είναι ο blobs, στον οποίο βλέπουμε για τον περιγραφητή HOG να αγγίζει τιμές επιτυχίας κοντά στο 70%. Συνεπώς στο παρόν πρόβλημα προτείνεται ο συνδυασμός του ανιχνευτή blobs με τον περιγραφητή HOG.