

Ηλιόπουλος Γεώργιος:

03118815

grgslpls1@gmail.com

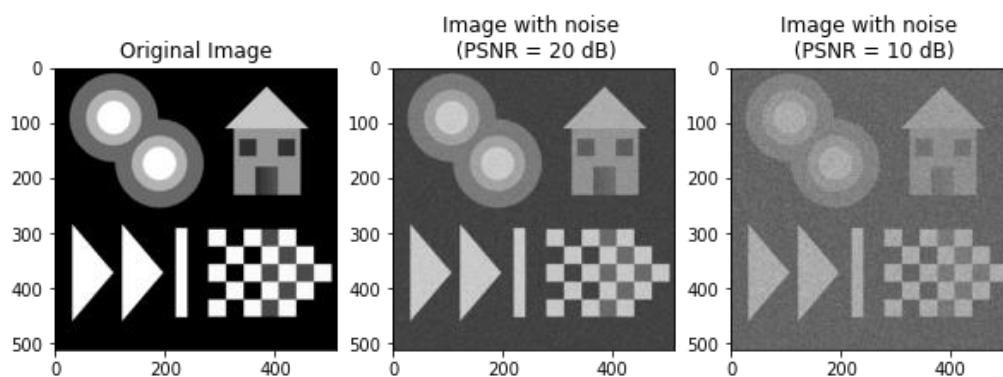
Σερλής Εμμανουήλ Αναστάσιος:

03118125

manosserlis@gmail.com

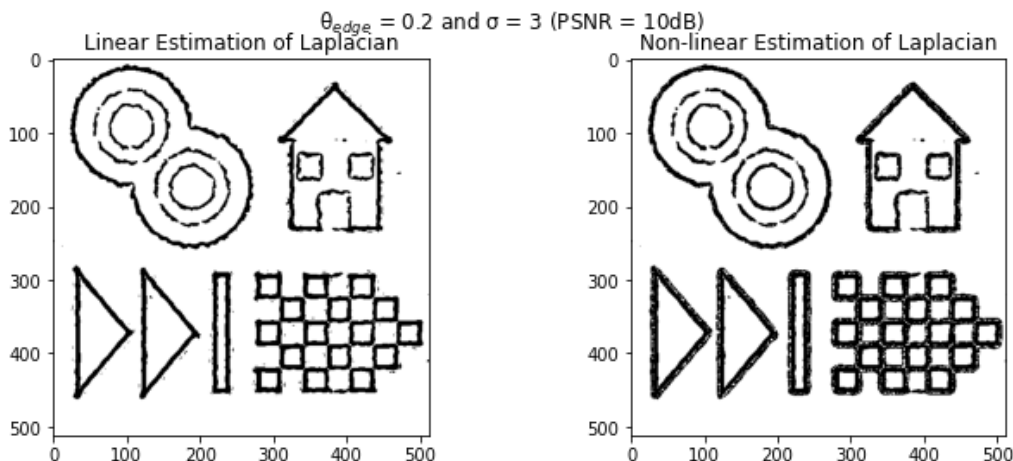
Μέρος 1: Ανίχνευση Ακμών σε Γκριζες Εικόνες

1.1) Για την προσθήκη θορύβου στην αρχική εικόνα χρησιμοποιήθηκε ο τύπος της εκφώνησης στον οποίο το σ_n υπολογίστηκε βάσει των γνωστών δεδομένων ως $\sigma_n = \frac{I_{max} - I_{min}}{10^{PSNR/20}}$. Παρακάτω φαίνονται η αρχική εικόνα και η εικόνα με θόρυβο δύο διαφορετικών τιμών PSNR.



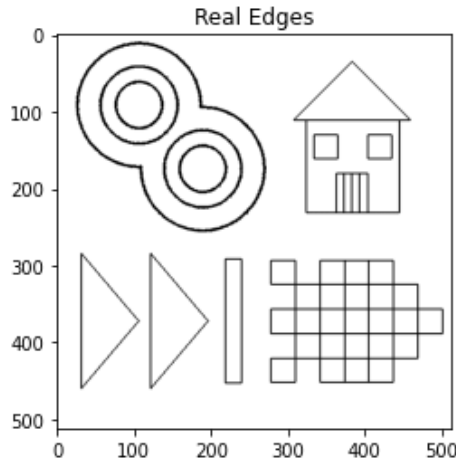
Παρατηρούμε πως για μεγαλύτερες τιμές PSNR ο θόρυβος είναι λιγότερο αισθητός.

1.2) Για την υλοποίηση της συνάρτησης ακολουθήθηκαν τα βήματα της εκφώνησης. Ως παράδειγμα παρακάτω φαίνονται έξοδοι της συνάρτησης για συγκεκριμένες παραμέτρους.



Παρατηρούμε πως με τη μη γραμμική προσέγγιση της Laplace οι ακμές βγαίνουν πολύ πιο παχιές από τις πραγματικές.

1.3) Εφαρμόζοντας τον απλό τελεστή ακμών ($M = (I_0 \oplus B) - (I_0 \ominus B)$) στην αθόρυβη εικόνα μπορούμε να βρούμε τις πραγματικές ακμές της εικόνας. Για την κατωφλιοποίηση και τη μετατροπή σε δυαδική εικόνα χρησιμοποιήθηκε $\theta_{realedge} = 0.01$.



Στην συγκεκριμένη εικόνα που χρησιμοποιούμε ως παράδειγμα στο μέρος 1 γνωρίζοντας τις πραγματικές ακμές οπότε μπορούμε να ποσοτικοποιήσουμε τα αποτελέσματα τους αλγορίθμου μας. Για αυτό το σκοπό κάνουμε χρήση του κριτηρίου ποιότητας της εκφώνησης $C = [\Pr(D|T) + \Pr(T|D)]/2$.

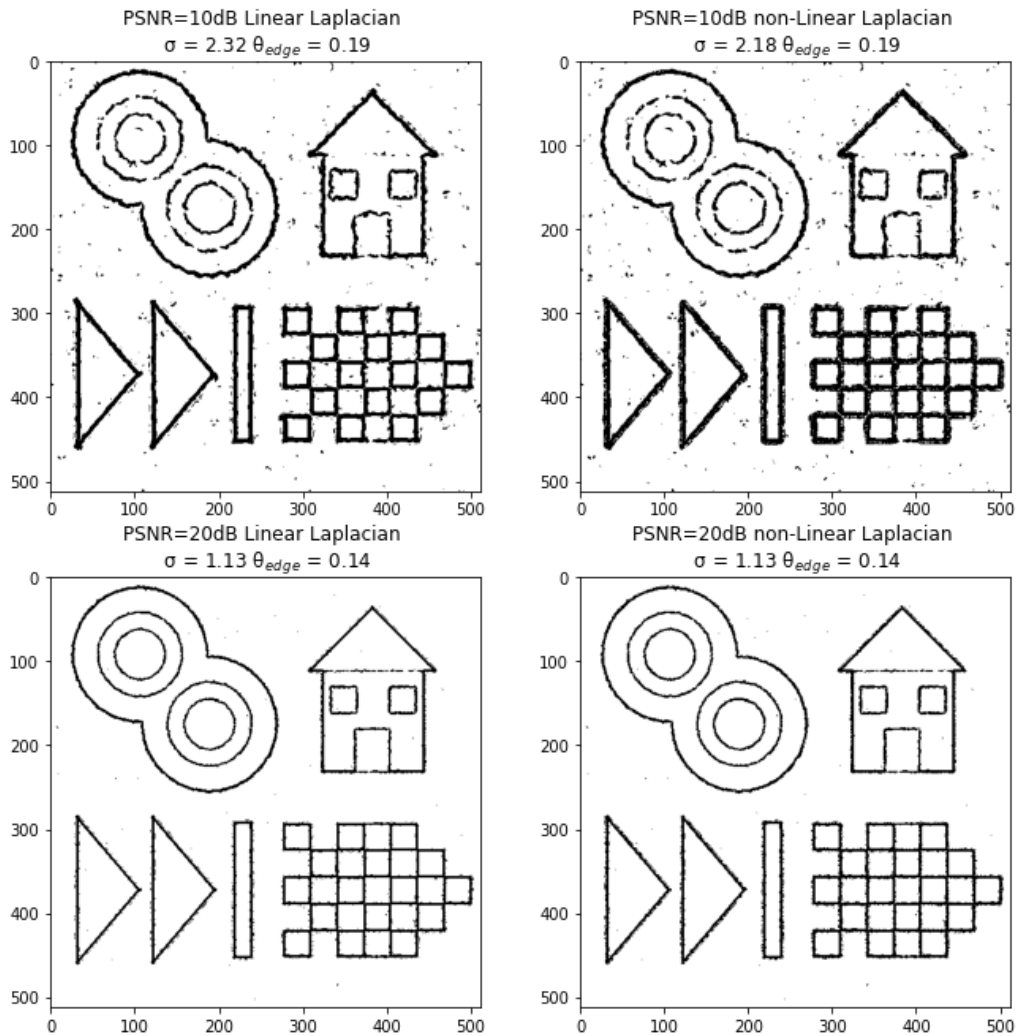
Για παράδειγμα για την εικόνα με θόρυβο 10dB με $\theta_{edge} = 0.2$ και $\sigma = 3$ βρίσκουμε πως $C = 0.639$ για γραμμική προσέγγιση της Laplacian και $C = 0.602$ για μη-γραμμική προσέγγιση της Laplacian.

Κάνοντας χρήση for loops εφαρμόζουμε διάφορες τιμές θ_{edge} και σ για τις εικόνες και βρίσκουμε πως οι βέλτιστοι συνδυασμοί είναι:

- για την εικόνα με **θόρυβο 10dB**:
 - γραμμική προσέγγιση της Laplacian $\theta_{edge} = 0.19$ και $\sigma = 2.32$ που βγάζουν $C = 0.649$,
 - μη-γραμμική προσέγγιση της Laplacian $\theta_{edge} = 0.19$ και $\sigma = 2.18$ που βγάζουν $C = 0.617$ και
- για την εικόνα με **θόρυβο 20dB**:
 - γραμμική προσέγγιση της Laplacian $\theta_{edge} = 0.14$ και $\sigma = 1.13$ που βγάζουν $C = 0.787$,
 - μη-γραμμική προσέγγιση της Laplacian $\theta_{edge} = 0.14$ και $\sigma = 1.13$ που βγάζουν $C = 0.763$.

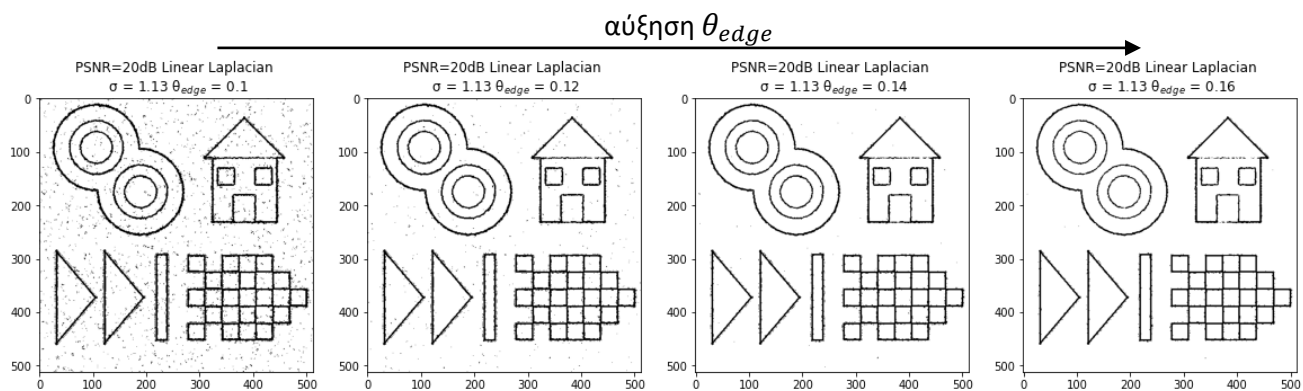
Βλέπουμε πως και για τις δύο εικόνες έχουμε καλύτερο αποτέλεσμα (πιο υψηλό C για την γραμμική προσέγγιση της Laplace.

Παρακάτω φαίνονται τα βέλτιστα αποτελέσματα για κάθε περίπτωση.

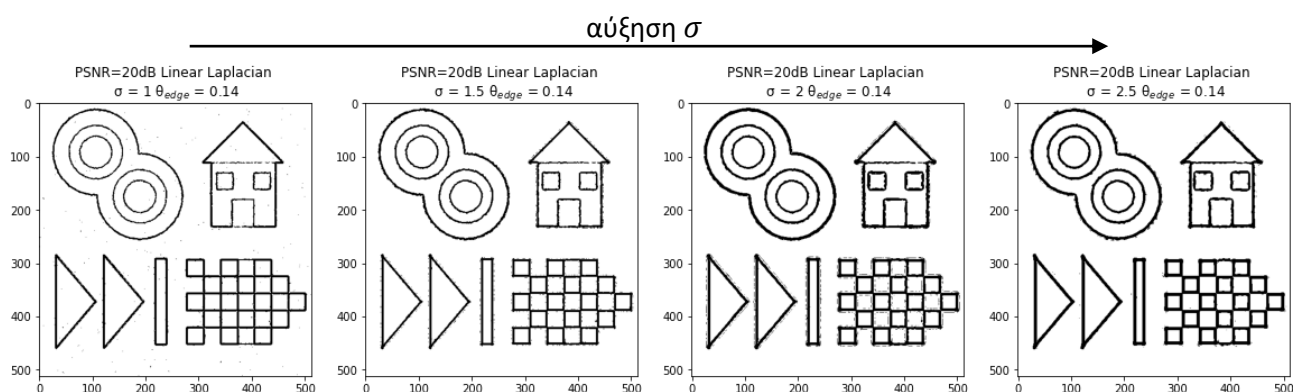


Παρατηρούμε πως:

- για την πιο θορυβώδη εικόνα αρκετά σημεία που δεν είναι ακμές εντοπίζονται από την συνάρτηση ως ακμές και σημεία που θα έπρεπε να εντοπίζονται ως ακμές αγνοούνται,
- στην εικόνα με 20dB θόρυβο οι ακμές εντοπίζονται ικανοποιητικά και δεν υπάρχει ιδιαίτερος θόρυβος στην εικόνα της εξόδου
- οι γραμμές που υπάρχουν στην πόρτα του σπιτιού δεν εντοπίζονται από την συνάρτηση, ίσως λόγω της πολύ μικρής αλλαγής της φωτεινότητας στα σημεία αυτά και
- με την μη γραμμική προσέγγιση της Laplace οι ακμές που εντοπίζονται είναι αισθητά πιο παχιές από τις πραγματικές.



Συγκρίνοντας τις παραπάνω 4 εικόνες στις οποίες μεταβάλλουμε το θ_{edge} διατηρώντας σταθερό το σ παρατηρούμε πως μειώνοντας το θ_{edge} (1η και 2η εικόνα), δηλαδή το κατώφλι της εντάσης που λαμβάνουμε ως ακμή, άσχετα σημεία εμφανίζονται ως ακμές. Αντίθετα αυξάνοντας το θ_{edge} (4η εικόνα), μειώνονται τα θορυβώδη σημεία, ωστόσο η συνάρτηση αγνοεί ορισμένα σημεία των ακμών.

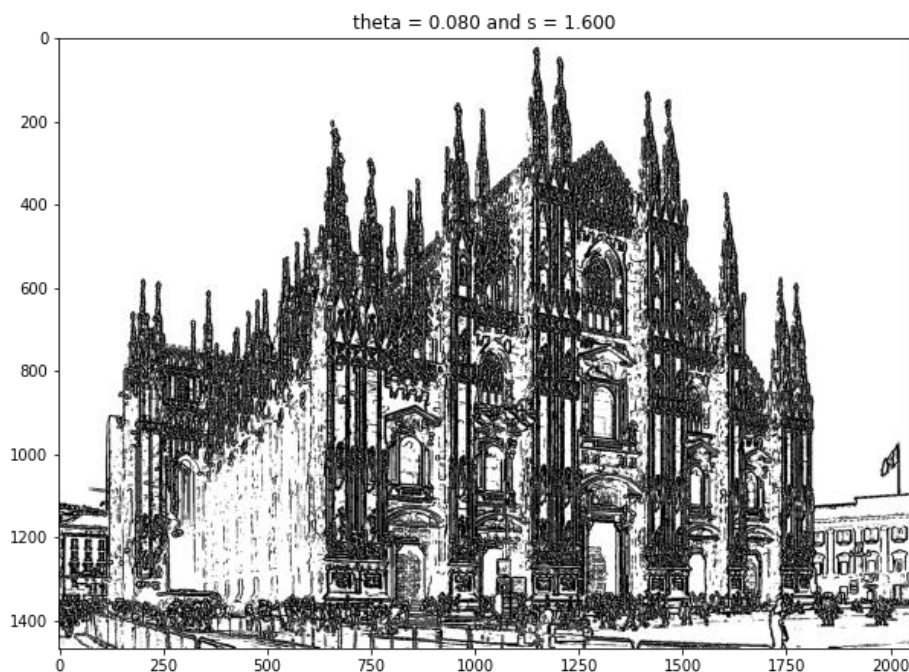


Όπως φαίνεται στις παραπάνω εικόνες αυξάνοντας το σ , δηλαδή την τυπική απόκλιση της Gaussian, εντοπίζονται οι ίδιες ακμές αλλά είναι πιο θορυβώδεις και πιο παχιές.

1.4) Θα εφαρμόσουμε τώρα την συνάρτησή μας στη πραγματική φωτογραφία `duomo_edges` που φαίνεται παρακάτω.



Δοκιμάζοντας διάφορους συνδυασμούς καταλήγουμε πως η ανίχνευση ακμών στην πραγματική φωτογραφία επιτυγχάνεται καλύτερα με $\theta_{edge} = 0.08$ και $\sigma = 1.6$.

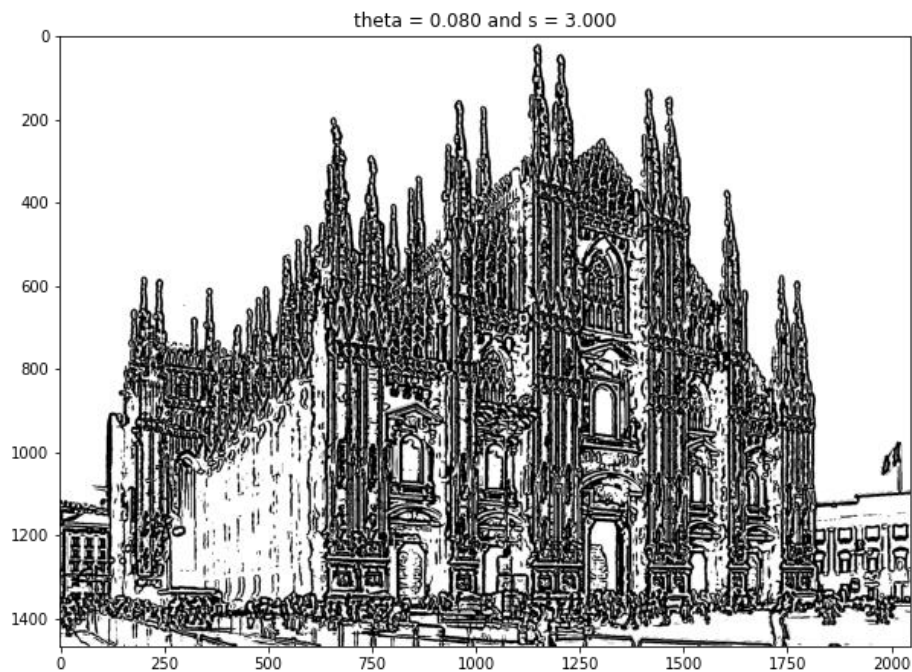


Δυσκολία υπάρχει ωστόσο στην ανίχνευση των ακμών στο αριστερό μέρος της εικόνας όπου το κτίριο είναι σκιασμένο και η φωτεινότητα χαμηλή. Αν θέλαμε να εντοπίσουμε τις ακμές σε αυτή την πλευρά του κτιρίου θα έπρεπε να μειώσουμε αρκετά το θ_{edge} , όπως φαίνεται στην επόμενη εικόνα, θυσιάζοντας όμως έτσι τις υπόλοιπες ακμές όπως φαίνεται

παρακάτω. Στην ουσία στην υπόλοιπη εικόνα όπου η φωτεινότητα είναι υψηλή βγαίνουν στην έξοδο πολλές λεπτομέρειες με αποτέλεσμα να χάνονται στο μάτι οι ακμές.

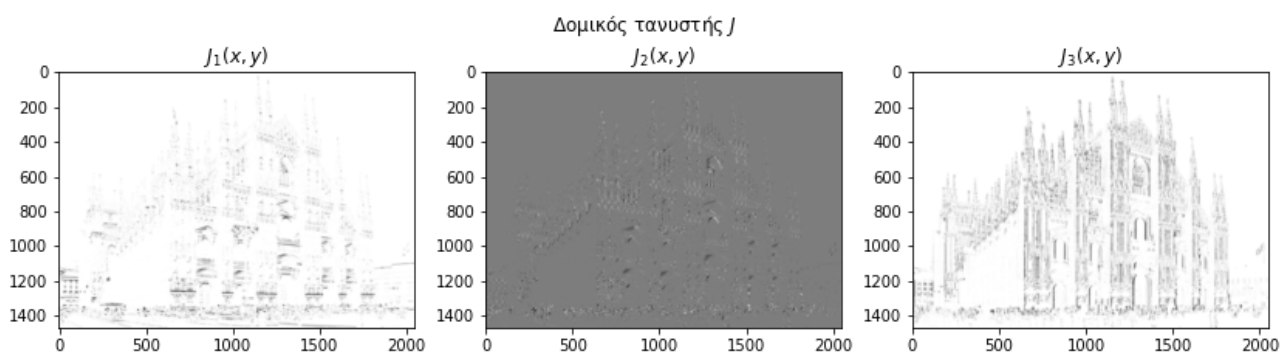


Στη συνέχεια βλέπουμε τις ακμές θέτοντας $\sigma = 3$. Με αυτό το σ βλέπουμε πως οι ακμές στρογγυλοποιούνται και έτσι πάλι δεν έχουμε το επιθυμητό αποτέλεσμα.

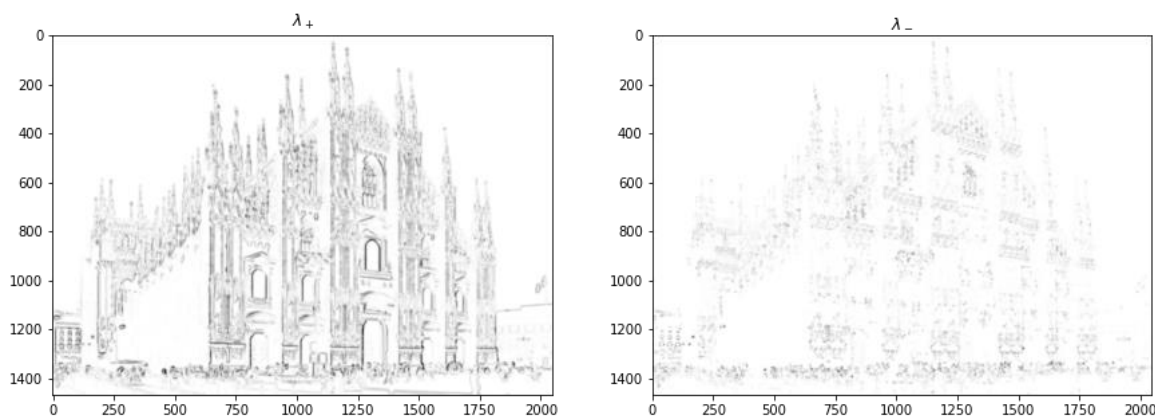


Μέρος 2: Ανίχνευση Σημείων Ενδιαφέροντος

2.1) Σε αυτό το ερώτημα θα υλοποιηθεί μία συνάρτηση ανίχνευσης γωνιών κάνοντας χρήση της κλασικής μεθόδου Harris-Stephens κατά τις οδηγίες της εκφώνησης. Αρχικά υπολογίζονται τα στοιχεία του δυναμικού τανυστή J σε κάθε pixel της εικόνας *duomo_edges*.



Στη συνέχεια εικονίζονται οι ιδιοτιμές λ_- και λ_+ .



Παρατηρούμε πως στην απεικόνιση της ιδιοτιμής λ_+ φαίνονται τα σημεία που αντιστοιχούν στις ακμές της εικόνας μας, ενώ στην απεικόνιση της ιδιοτιμής λ_- φαίνονται σημεία που πιθανώς αντιστοιχούν σε γωνίες. Αν συνδυάσουμε τις δύο παραπάνω απεικονίσεις (όπως γίνεται στην συνέχεια) καταλαβαίνουμε πως θα αναδειχθούν τα σημεία στα οποία αντιστοιχούν οι γωνίες της εικόνας.

Έτσι αφού έχουμε υπολογίσει τις παραπάνω ιδιοτιμές τις συνδυάζουμε με το κριτήριο γωνιότητας το οποίο στην ουσία είναι μαθηματική σχέση που αναλόγως των τιμών της μπορούμε να διαπιστώσουμε αν κάποιο σημείο πρέπει να ληφθεί ως γωνία ή όχι.

Στην συνέχεια φαίνονται τα αποτελέσματα της συνάρτησης για τις τυπικές τιμές των παραμέτρων, δηλαδή $\sigma = 2, \rho = 2.5, \kappa = 0.05, \theta_{corn} = 0.005, s = 1.5$ και $N = 4$ για κάθε μία από τις εικόνες που μας δόθηκαν.

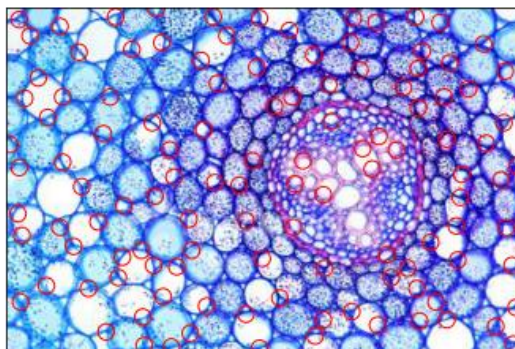
duomo_edges



donuts



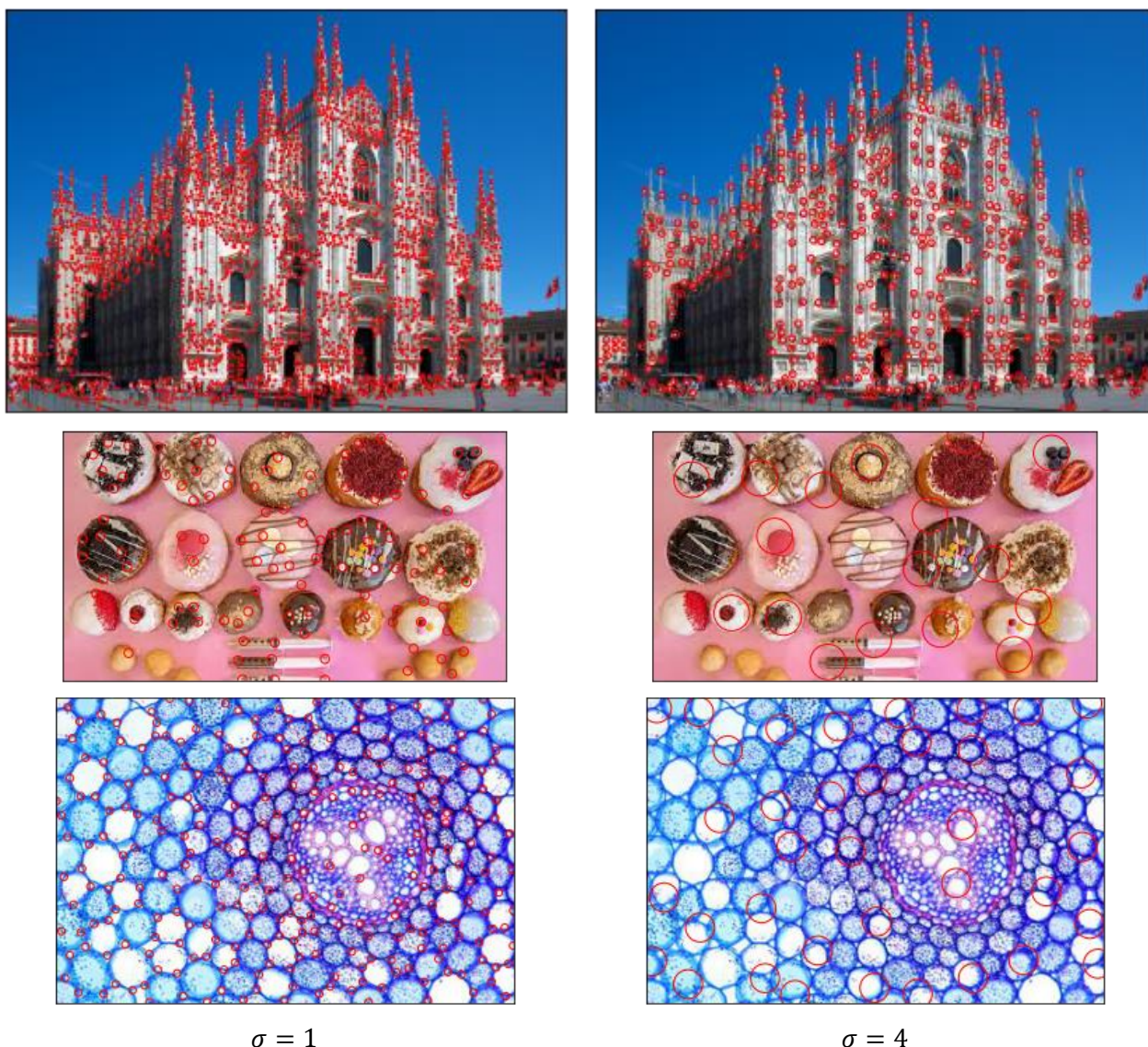
cells



Παρατηρούμε πως ο αλγόριθμός μας εντοπίζει σημεία που ίσως θα μπορούσαν να είναι γωνίες αλλά στην ουσία δεν είναι. Ο αλγόριθμος δηλαδή δεν αντιλαμβάνεται τις γωνίες με τον ίδιο τρόπο όπως το ανθρώπινο μάτι. Έχουμε βασίσει τον εντοπισμό των γωνιών στην απότομη αλλαγή φωτεινότητας προς όλες τις κατευθύνσεις αλλά με αυτό τον τρόπο σημεία στη κυκλική περιφέρεια αντικειμένων λαμβάνονται και αυτά ως γωνίες όπως φαίνεται τόσο στην εικόνα με τα donuts όσο και στην εικόνα των κυττάρων. Παρατηρούμε βέβαια πως στην εικόνα duomo ο εντοπισμός των γωνιών είναι πιο ακριβής καθώς η αρχιτεκτονική του συγκεκριμένου κτιρίου είναι πιο τετραγωνισμένη. Βέβαια στα σημεία στο αριστερό μέρος όπου η φωτεινότητα είναι χαμηλή δεν γίνεται εντοπισμός των γωνιών.

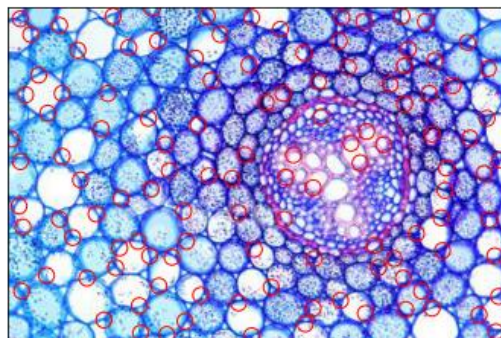
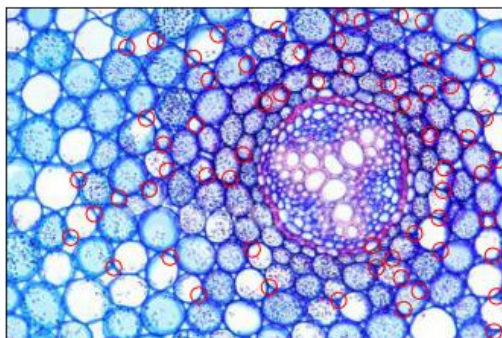
Στην συνέχεια θα πειραματιστούμε με τις τιμές των παραμέτρων ώστε να δούμε τις επηρεάζει η καθεμία από αυτές.

μεταβολή κλίμακας διαφόρισης σ



Παρατηρούμε πως αυξάνοντας το σ αυξάνεται η κλίμακα στην οποία εντοπίζονται οι γωνίες, αφού το σ εκφράζει την κλίμακα διαφόρισης. Για την εικόνα του κτιρίου το $\sigma = 1$ είναι πολύ μικρό και αποτυπώνονται ως γωνίες μικρές αρχιτεκτονικές λεπτομέρειες που πιθανότητα δεν είναι εν τέλει σημεία ενδιαφέροντος. Όμως για την εικόνα των κυττάρων ο εντοπισμών των γωνιών γίνεται πολύ καλύτερα για $\sigma = 1$ αφού οι γωνίες εμφανίζονται σε μικρότερη κλίμακα που δεν καλυπτόταν σωστά για $\sigma = 2$. Αντίστοιχα και για την εικόνα με τα ντόνατ τα αποτελέσματα είναι καλύτερα για μικρότερο σ . Για $\sigma = 4$ τα αποτελέσματα είναι άστοχα για τις εικόνες των κυττάρων και των donuts ενώ για την εικόνα του κτιρίου τα αποτελέσματα είναι καλύτερα.

μεταβολή κατώφλιου θ_{corn}



$\theta_{corn} = 0.2$

$\theta_{corn} = 0.00001$

Με την παραπάνω σύγκριση στα αποτελέσματα με διαφορετικά κατώφλια θ_{corn} παρατηρούμε πως για πολύ μικρό κατώφλι ο αριθμός των γωνιών που εντοπίζονται είναι πολύ μεγάλος ενώ για μικρό κατώφλι είναι μικρός.

Εν τέλει συμπεραίνουμε πως η μεταβολή του σ σχετίζεται με το μέγεθος των αντικειμένων που σχηματίζουν την εκάστοτε γωνία, ενώ το θ_{corn} σχετίζεται με το πλήθος των εικονιζόμενων γωνιών

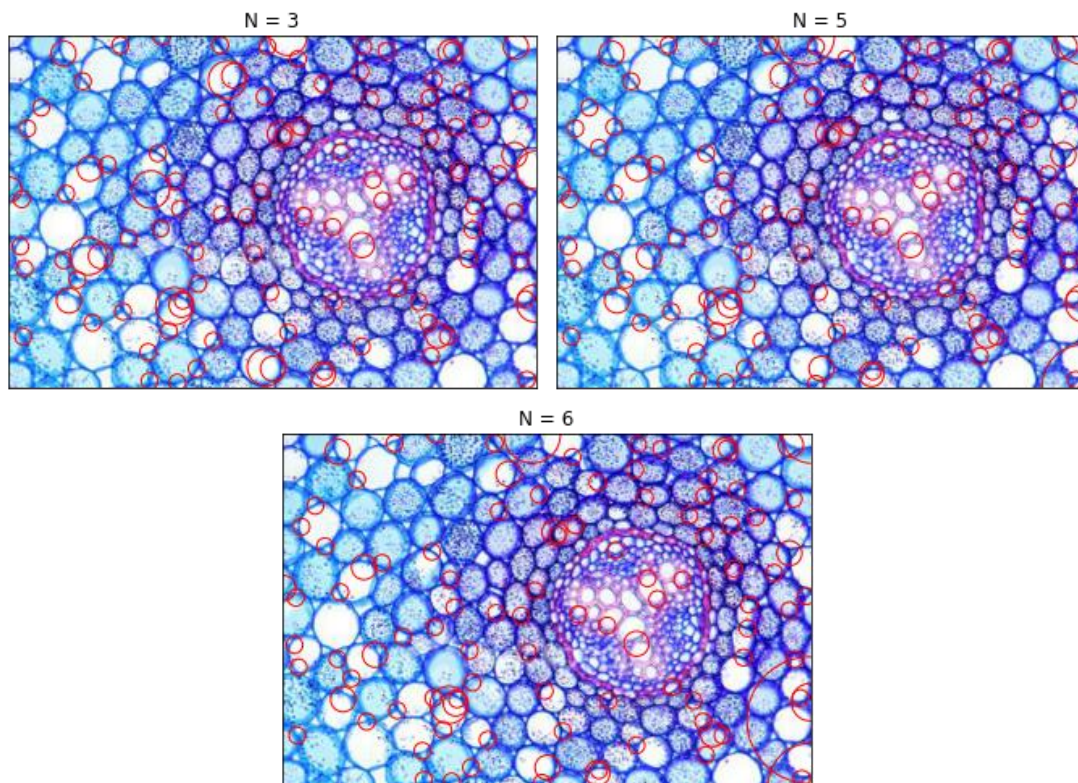
Συνεπώς, οι τιμές των παραμέτρων πρέπει να ορίζονται βάσει της μορφολογίας της εκάστοτε εικόνας ώστε να λειτουργεί σωστά ο αλγόριθμος Harris-Stephans.

2.2) Στην παραπάνω ανίχνευση γωνιών η κλίμακα είναι συγκεκριμένη και καθορισμένη από τις παραμέτρους σ και ρ . Στην συνέχεια θα υλοποιήσουμε πολυκλιμακωτή ανίχνευση γωνιών βασισμένη στον αλγόριθμο Harris-Laplacian. Για αυτή τη μέθοδο αρχικά βρίσκουμε τις γωνίες για διάφορες κλίμακες διαφόρισης και ολοκλήρωσης και στη συνέχεια επιλέγουμε ως γωνίες αυτές οι οποίες μεγιστοποιούν την LoG μετρική, που δίνεται στην εκφώνηση, σε μία γειτονιά τουλάχιστον 2 διαδοχικών κλιμάκων.

Στη συνέχεια φαίνονται τα αποτελέσματα του αλγορίθμου για τις εικόνες εισόδου και για τις τυπικές τιμές των παραμέτρων.

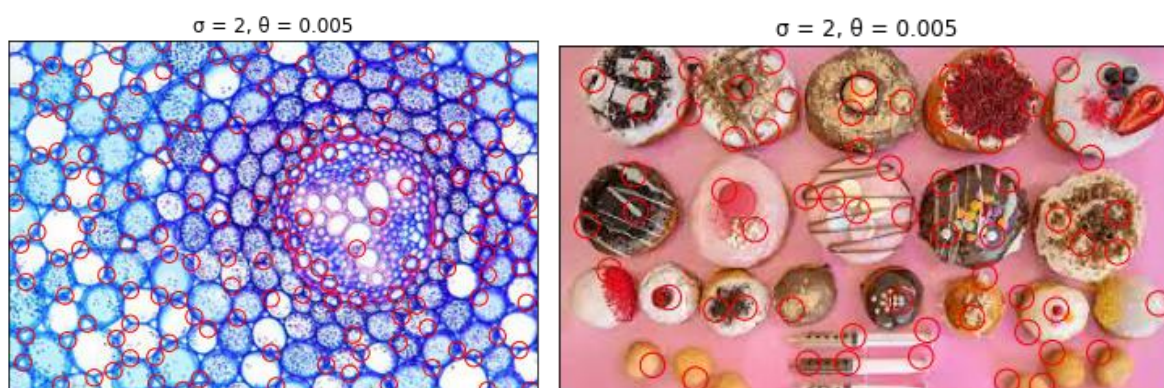


Ενδιαφέρον έχει σε αυτή την περίπτωση να μελετήσουμε την επιρροή της παραμέτρου N , δηλαδή των αριθμό των κλιμάκων. Για αυτό το σκοπό εφαρμόζουμε τον αλγόριθμο στην εικόνα των κυττάρων για διάφορα N και σταθερές τις υπόλοιπες παραμέτρους.



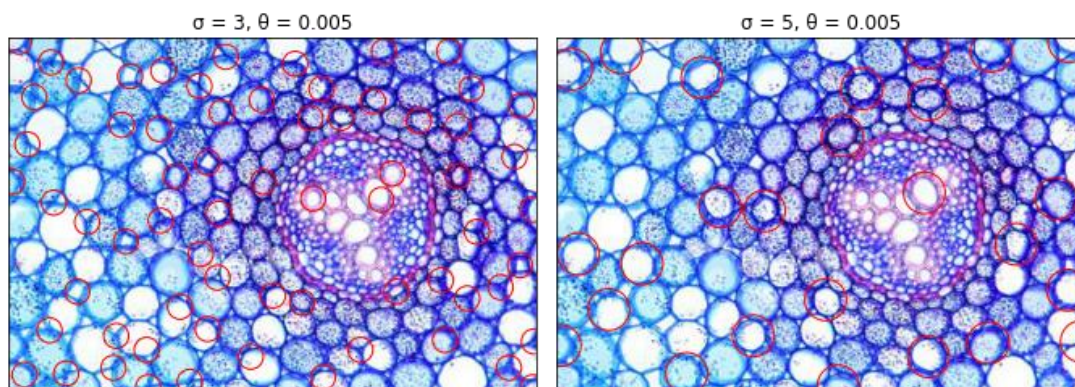
Βλέπουμε πως για μεγαλύτερα N ($N = 6$) εντοπίζονται και γωνίες μεγαλύτερης κλίμακας. Βέβαια συγκρίνοντας και με τα αποτελέσματα του ερωτήματος 2.1 για $\sigma = 4$, οι εντοπιζόμενες γωνίες είναι μεγαλύτερης κλίμακας από αυτές που έχουμε τώρα. Καταλαβαίνουμε λοιπόν πως ο αλγόριθμος Harris Laplacian τείνει να συγκρατεί τις γωνίες μικρότερης κλίμακας με το κριτήριο της LoG μετρικής.

2.3) Ένα άλλο σημείο ενδιαφέροντος αποτελούν τα blobs περιοχές δηλαδή που παρουσιάζουν κάποια ομοιογένεια και διαφέρουν σημαντικά από την γειτονιά τους. Για τον εντοπισμού τους γίνεται χρήση του πίνακα Hessian. Αρχικά υπολογίζουμε τα στοιχεία του πίνακα που είναι οι μερικές παράγωγοι δεύτερης τάξης της εικόνας. Το κριτήριο που χρησιμοποιήθηκε είναι η ορίζουσα του πίνακα Hessian. Βάσει αυτού του κριτηρίου επιλέχθηκαν ως blobs τα σημεία που αποτελούν τοπικά μέγιστα και έχουν τιμή μεγαλύτερη από ένα κατώφλι, δηλαδή επιλέξαμε εκ νέου το set κριτηρίων του ερωτήματος 2.1. Στη συνέχεια φαίνονται τα αποτελέσματα της συνάρτησής μας για τυπικές τιμές των παραμέτρων.



Παρατηρούμε πως πράγματι η συνάρτηση εντοπίζει περιοχές με ομοιογένεια αλλά όχι σε ικανοποιητικό βαθμό. Στη συνέχεια μεταβάλουμε τις παραμέτρους σ και θ_{corn} σε μία προσπάθεια να βελτιώσουμε την απόδοση.

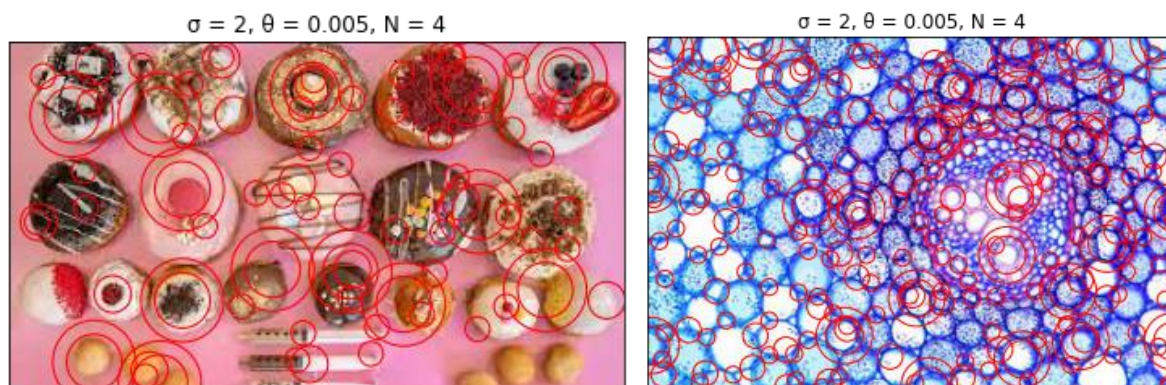




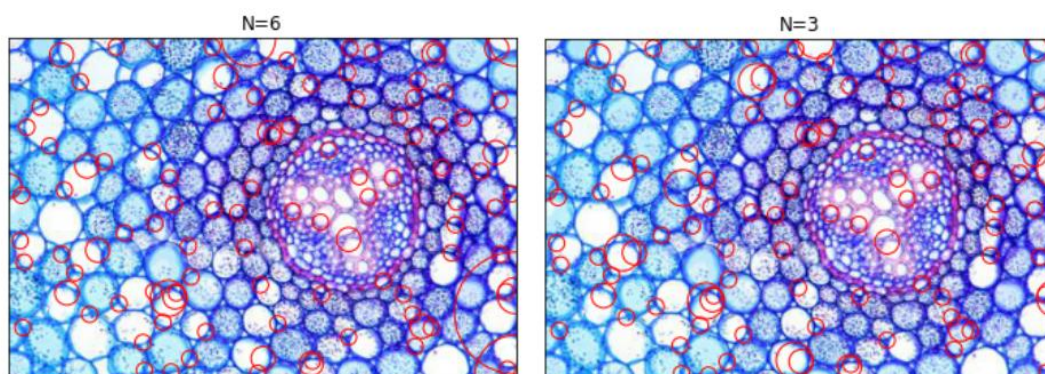
Παρατηρούμε πως για καλύτερο εντοπισμό blobs πρέπει να προσαρμόσουμε κατάλληλα το σ σύμφωνα με το μέγεθος των περιοχών που επιθυμούμε να βρίσκουμε. Για παράδειγμα στην εικόνα των κυττάρων για $\sigma = 5$ τα blobs που εντοπίζονται αντιστοιχούν καλύτερα στο μέγεθος των κυττάρων ενώ για $\sigma = 2$ που είδαμε παραπάνω τα blobs είναι πολύ μικρά.

Όπως παρατηρήσαμε και στην ανίχνευση γωνιών έτσι και τώρα η αλλαγή του κατωφλίου θα επηρέαζε τον αριθμό των εντοπιζόμενων blobs, αφού θα είμαστε είτε πιο ελαστικοί είτε πιο αυστηροί με το πόση διαφορά πρέπει να εντοπίζεται στην ορίζουσα Hessian ώστε να λάβουμε μία περιοχή ως blob.

2.4) Όπως προηγουμένως με την ανίχνευση γωνιών έτσι και τώρα θα δοκιμάσουμε και πολυκλιμακωτή μέθοδο για την ανίχνευση blobs. Τα αποτελέσματα:



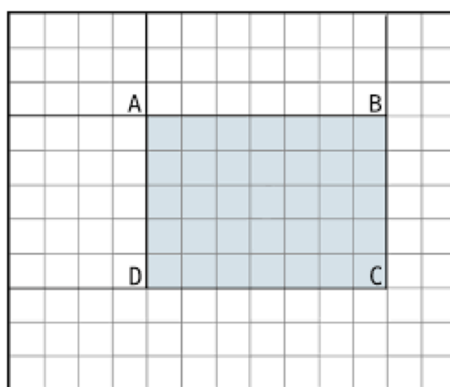
Τα αποτελέσματα για διαφορετικά N:



Καταλήγουμε στο συμπέρασμα ότι η αύξηση του πλήθους των κλιμάκων δεν οδηγεί απαραίτητα σε βελτίωση αποτελεσμάτων εύρεσης των blobs, μιας και κάτι τέτοιο εξαρτάται και από το μέγεθος των περιοχών ενδιαφέροντος που θέλουμε να εντοπίσουμε. Κατ' επέκταση, η παραπάνω υλοποίηση για $N=6$ δεν καθίσταται βελτιωμένη μιας και απουσιάζουν από την `img_cells` μεγάλα blobs.

2.5) Λόγω της υψηλής πολυπλοκότητας της Hessian μεθόδου για το blob detection, στο εν λόγω ερώτημα πραγματοποιείται η μέθοδος των box filters με χρήση ολοκληρωτικής εικόνας. Συγκεκριμένα, τα box filters αποτελούν προσέγγιση των παραγώγων 2ης τάξης και αποτελούνται από επιμέρους αθροίσματα ορθογώνιων περιοχών. Τα box filters εφαρμόζονται πάνω στην ολοκληρωτική εικόνα, της οποίας κάθε pixel αποτελεί το άθροισμα των προηγούμενων-κατά γραμμή και στήλη-στοιχείων της αρχικής εικόνας, ενώ υπολογίζεται πολύ αποδοτικά σε $O(1)$ μέσω της σχέσης:

$$\sum \sum I(i,j) = SA + SC - SB - SD$$



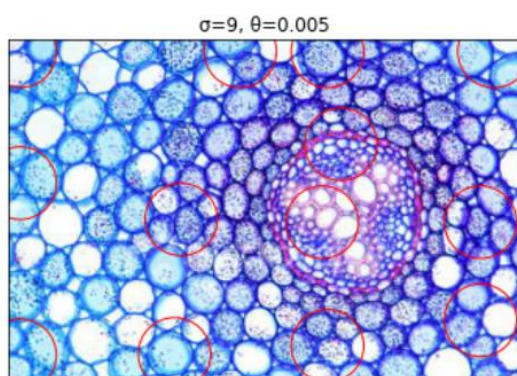
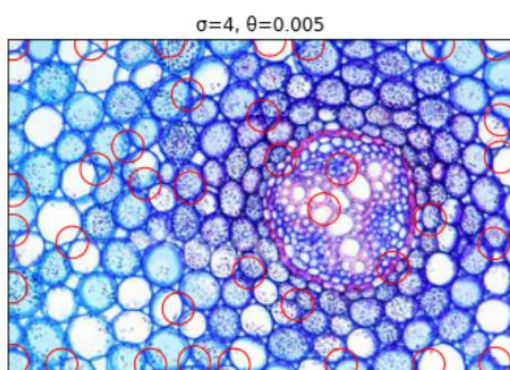
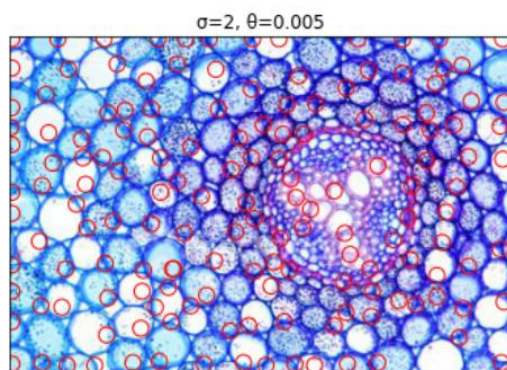
Αυτό έχει ως αποτέλεσμα η εφαρμογή οποιουδήποτε φίλτρου στην ολοκληρωτική εικόνα να αποκτά σταθερή πολυπλοκότητα, η οποία μάλιστα είναι ανεξάρτητη του μεγέθους του παραθύρου προς εφαρμογή, μιας και δεν απαιτείται να περάσουμε από όλα τα pixels του εκάστοτε παραθύρου, όπως γίνεται στην τυπική υλοποίηση της πράξης της συνέλιξης.

Όσον αφορά το κομμάτι της προγραμματιστικής υλοποίησης, αυτή έγινε με 2 τρόπους (τόσο για σταθερό σ όσο και για την πολυκλιμακωτή ανάλυση με box filters).

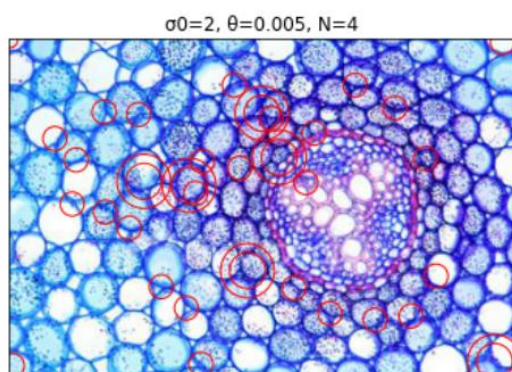
1. Συναρτήσεις “box_filters” και “multi_box_filter”, οι οποίες πραγματοποιούν για κάθε pixel manual υπολογισμό των θέσεων ενδιαφέροντος σε κάθε παράθυρο, για κάθε έναν υπολογισμό των L_{xx} , L_{xy} , L_{yy}

Λάβαμε τα κάτωθι αποτελέσματα:

Για μονοκλιμακωτή ανάλυση:



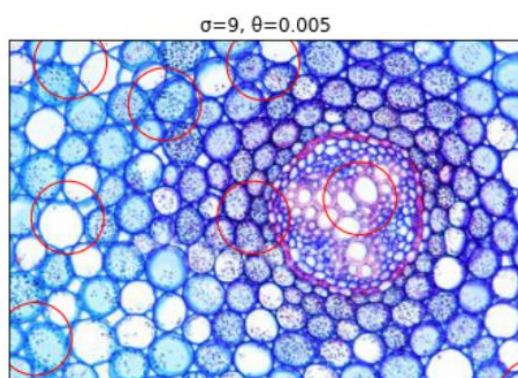
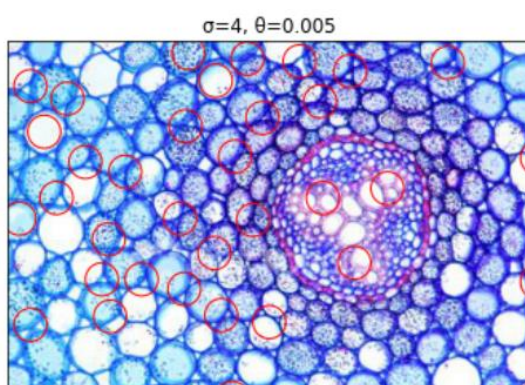
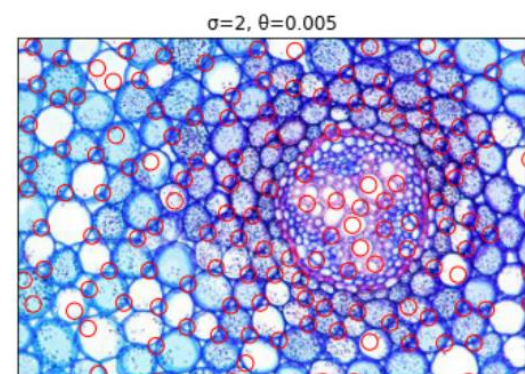
Για πολυκλιμακωτή ανάλυση:



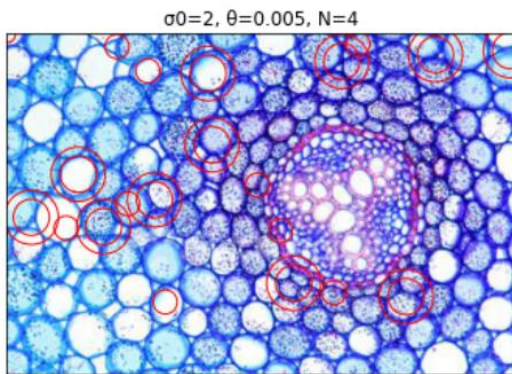
2. Συναρτήσεις “**box_filters_2**” και “**multi_box_filter_2**”, οι οποίες κάνουν χρήση των συναρτήσεων `np.pad` και `np.roll` για την ομαλότερη υλοποίηση της εν λόγω διαδικασίας.

Λάβαμε τα κάτωθι αποτελέσματα:

Για μονοκλιμακωτή ανάλυση:



Για πολυκλιμακωτή ανάλυση:



Από συμπεράσματα, παρατηρούμε πως ο manual αλγόριθμος εμφανίζει πληθώρα false positive τιμών κοντά στα άκρα της εκάστοτε εικόνας, με την ακρίβεια να βελτιώνεται σε κεντρικά σημεία της, κάτι το οποίο φαίνεται να αντιμετωπίζει καλύτερα η δεύτερη υλοποίηση.

Στην συνέχεια, σε σύγκριση με την Hessian μέθοδο του ερωτήματος 2.3, παρατηρούμε μικρή μείωση στην ακρίβεια των box filters, η οποία αντισταθμίζεται όμως από αντίστοιχη μείωση στην υπολογιστική πολυπλοκότητα, που σχετίζεται με τον διαφορετικό προσδιορισμό των δεύτερων παραγώγων των εν λόγω μεθόδων.

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

3.1) Σε αυτό το μέρος θα εξετάσουμε την ικανότητα εύρεσης της περιστροφής και της κλίμακας εικόνων κάνοντας χρήση των ανιχνευτών που κατασκευάσαμε παραπάνω σε συνδυασμό με δύο περιγραφητές: τον HOG (Histogram of Oriented Gradients) και τον SURF (Speed Up Robust Features). Στο πείραμα έγινε χρήση 60 συνολικά εικόνων εικόνων που έχουν προέλθει από 3 αρχικές εικόνες που υπέστησαν 4 παραμορφώσεις κλίμακας και 5 περιστροφές. Έτσι γνωρίζοντας την αρχική εικόνα μπορεί να γίνει εκτίμηση σφάλματος του ταιριάσματος (matching) κάθε συνδυασμό περιγραφητή και ανιχνευτή.

Στον πίνακα της επόμενης σελίδας παρουσιάζονται τα μέσα σφάλματα για κάθε συνδυασμό καθώς και ορισμένα χρήσιμα ποσοστά για να δούμε την γενικότερη απόδοση του ταιριάσματος.

| descriptor | detector | image 1 | | image 2 | | image 3 | | Total Avg. Error for all images |
|---------------------|--------------------------|------------------|------------------|------------------|------------------|------------------|------------------|---------------------------------|
| | | Avg. Scale Error | Avg. Theta Error | Avg. Scale Error | Avg. Theta Error | Avg. Scale Error | Avg. Theta Error | |
| SURF | corner detect | 0.003 | 1.968 | 0.002 | 0.318 | 0.097 | 12.909 | 2.550 |
| | multiscale corner detect | 0.002 | 0.134 | 0.140 | 18.328 | 0.082 | 10.015 | 4.784 |
| | blob detect | 0.027 | 7.759 | 0.010 | 0.229 | 0.001 | 0.054 | 1.347 |
| | multiscale blob detect | 0.002 | 0.100 | 0.002 | 0.058 | 0.001 | 0.061 | 0.037 |
| | box filters | 0.093 | 14.810 | 0.161 | 12.751 | 0.036 | 2.854 | 5.118 |
| | multiscale box filters | 0.084 | 5.783 | 0.176 | 9.379 | 0.031 | 1.670 | 2.854 |
| HOG | corner detect | 0.186 | 22.619 | 0.351 | 19.199 | 0.285 | 23.699 | 11.057 |
| | multiscale corner detect | 0.194 | 16.500 | 0.501 | 22.945 | 0.177 | 27.065 | 11.230 |
| | blob detect | 0.186 | 7.231 | 0.100 | 13.674 | 0.154 | 27.219 | 8.094 |
| | multiscale blob detect | 0.131 | 14.430 | 0.210 | 15.466 | 0.094 | 13.725 | 7.343 |
| | box filters | 0.208 | 20.570 | 0.419 | 33.358 | 0.380 | 41.050 | 15.998 |
| | multiscale box filters | 0.193 | 22.351 | 0.249 | 28.480 | 0.473 | 23.333 | 12.513 |
| Avg. Error for SURF | | 0.035 | 5.092 | 0.082 | 6.844 | 0.041 | 4.594 | 2.781 |
| Avg. Error for HOG | | 0.183 | 17.284 | 0.305 | 22.187 | 0.261 | 26.015 | 11.039 |
| Total Avg | | 0.109 | 11.188 | 0.193 | 14.515 | 0.151 | 15.305 | |

Παρατηρούμε πως το μικρότερο μέσο σφάλμα παρουσιάζεται από τον συνδυασμό **SURF Descriptor+ Multiscale Blob Detector**. Γενικότερα, τα σφάλματα είναι μικρότερα για τον περιγραφητή SURF. Ο περιγραφητής SURF σε γενικές γραμμές εμφανίζει πολύ μικρά σφάλματα με εξαίρεση λίγες περιπτώσεις. Αντίθετα, ο περιγραφητής HOG παρουσιάζει αρκετά μεγαλύτερα σφάλματα ειδικά στην περίπτωση της ανίχνευσης γωνίας περιστροφής, όπου κρίνοντας από τα μεγάλα σφάλματα η ανίχνευση έγινε λανθασμένα. Υπάρχει βέβαια γενικότερη δυσκολία στον εντοπισμό της γωνίας περιστροφής των εικόνων και στον περιγραφητή SURF. Αξίζει να σημειωθεί πως και στους δύο περιγραφητές ο ανιχνευτής blob detect και ο πολυκλιμακωτός ανιχνευτής blob detect εμφανίζουν τα μικρότερα σφάλματα, ενώ ο περιγραφητής με τα box filters τα μεγαλύτερα.

Πιο αναλυτικά, η υπεροχή του περιγραφητή SURF στον εντοπισμό της γωνίας περιστροφής ήταν αναμενόμενη αφού, σε αντίθεση με τον HOG, η μέθοδος των SURF υπολογίζει πρώτα της γενική κατεύθυνση της γειτονιάς ώστε να εξαχθούν περιστροφικά ανεξάρτητοι περιγραφητές. Η μέθοδος των HOG αδυνατεί πλήρως να εντοπίσει τις περιστροφές αφού

χρησιμοποιεί κατά κύριο λόγο τη κλίση της εικόνας για την κωδικοποίηση με αποτέλεσμα να μη επιτυγχάνει ικανοποιητικό ταίριασμα όταν η εικόνα είναι περιστραμμένη.

3.2) Σε αυτό το μέρος θα αξιολογηθεί η επίδοση και η καταλληλότητα των διαφόρων ανιχνευτών και περιγραφητών σε ένα τυπικό πρόβλημα κατηγοριοποίησης εικόνων. Συγκεκριμένα θα κατηγοριοποιηθεί ένα σύνολο εικόνων από αυτοκίνητα, ανθρώπους και ζώα στις 3 κατηγορίες.

Αρχικά κάνοντας χρήση της συνάρτησης FeatureExtraction εξάγουμε χαρακτηριστικά-με έναν ανιχνευτή τη φορά-από τις εικόνες όλης της βάσης. Στη συνέχεια με τη συνάρτηση createTrainTest οι εικόνες διαχωρίζονται σε test set και σε train set με τα κατάλληλα labels που θα δείχνουν την κλάση (αυτοκίνητο, άνθρωπος, ζώο) στην οποία ανήκει η εικόνα. Για το πρόβλημα αναγνώρισης εικόνων δεν αρκούν οι τοπικοί περιγραφητές που χρησιμοποιήσαμε προηγουμένως αλλά απαιτείται ένα συνολικό διάνυσμα χαρακτηριστικών με την ίδια διάσταση για κάθε εικόνα. Έτσι χρησιμοποιώντας την συνάρτηση BagOfWords (παρακάτω αναπτύσσουμε και την δικιά μας συνάρτηση BoVW) κατασκευάζεται το ιστόγραμμα εμφάνισης των χαρακτηριστικών, που ανιχνεύθηκαν, σε όλη την εικόνα. Τέλος η κατηγοριοποίηση των εικόνων γίνεται με βάση την BoVW αναπαράσταση κάνοντας χρήση ενός SVM (Support Vector Machine) ταξινομητή, ο οποίος είναι κατάλληλα προσαρμοσμένος για πολλαπλές κλάσεις.

Στον παρακάτω πίνακα φαίνονται τα ποσοστά επιτυχίας κάθε συνδυασμού.

| descriptor | detector | BagOfWords |
|------------|--------------------------|----------------|
| SURF | multiscale corner detect | 63.034% |
| | multiscale blob detect | 62.069% |
| | multiscale box filters | 63.074% |
| | SURF Avg: | 62.726% |
| HOG | multiscale corner detect | 63.172% |
| | multiscale blob detect | 61.793% |
| | multiscale box filters | 64.276% |
| | HOG Avg: | 63.080% |

Παρατηρούμε πως γενικώς για όλου τους συνδυασμούς το ποσοστό επιτυχίας κυμαίνεται γύρω από το 63%. Η καλύτερη κατηγοριοποίηση επιτυγχάνεται με τον συνδυασμό **HOG Descriptor + Multiscale Box Filters**. Αυτό είναι παράδοξο αφού θεωρητικά στις υλοποιήσεις με box filters θυσιάζουμε την ευστοχία ώστε να έχουμε πιο αποδοτικό

αλγόριθμο. Συγκρίνοντας τους δύο περιγραφητές βλέπουμε πως σε αντίθεση με το προηγούμενο πείραμα matching τώρα ο περιγραφητής HOG έχει ελαφρώς καλύτερα αποτελέσματα. Η καλύτερη απόδοση του μπορεί να αποδοθεί στο γεγονός πως τα HOGs παρέχουν από μόνα τους μια πυκνή επικαλυπτόμενη περιγραφή των περιοχών μίας εικόνας και υπολογίζονται σε ένα πυκνό πλέγμα ομοιόμορφα κατανεμημένων κελιών. Έτσι τα HOGs περιγράφουν ολόκληρη την εικόνα το οποίο είναι βοηθητικό για το πείραμα της κατηγοριοποίησης.

Bonus Υλοποίηση συνάρτησης BoVW

Όσον αφορά την υλοποίηση της BoVW, αυτή περιλάμβανε την ενοποίηση όλων των τοπικών περιγραφητών σε ένα ενιαίο σύνολο, ανεξάρτητα από την κλάση στην οποία ανήκουν. Στην συνέχεια, έγινε εφαρμογή του αλγόριθμου kmeans σε ένα τυχαίο υποσύνολο του παραπάνω συνόλου και για αριθμό κέντρων ίσο με 500. Μετέπειτα, υπολογίστηκε η ελάχιστη ευκλείδεια απόσταση κάθε τοπικού περιγραφητή από τα παραπάνω κέντρα, ενώ με βάση αυτό δημιουργήθηκαν τα ιστογράμματα συχνότητας εμφάνισης των οπτικών λέξεων. Τέλος, πραγματοποιήθηκε κανονικοποίηση με βάση την L2 νόρμα.

Στον επόμενο πίνακα παρουσιάζονται τα ποσοστά επιτυχίας τόσο για την έτοιμη συνάρτηση BagOfWords όσο και για την δικιά μας BoVW.

| descriptor | detector | BagOfWords | BoVW |
|------------|--------------------------|------------|---------------|
| SURF | multiscale corner detect | 63.034% | 63.45% |
| | multiscale blob detect | 62.069% | 63.17% |
| | multiscale box filters | 63.074% | 63.59% |
| | SURF Avg: | 62.726% | 63.402% |
| HOG | multiscale corner detect | 63.172% | 60.41% |
| | multiscale blob detect | 61.793% | 63.59% |
| | multiscale box filters | 64.276% | 62.35% |
| | HOG Avg: | 63.080% | 62.12% |
| Total Avg: | | 62.903% | 62.759% |

Παρατηρούμε πως με τη δική μας υλοποίηση ο περιγραφητής SURF βγάζει ελαφρώς καλύτερα αποτελέσματα ενώ ο HOG χειρότερα. Με την συνάρτηση BoVW καλύτερα αποτελέσματα έχουμε για τους συνδυασμούς **SURF Descriptor + Multiscale Box Filters** και **HOG Descriptror + Multisclae Blob Detector**.