«Όραση Υπολογιστών»

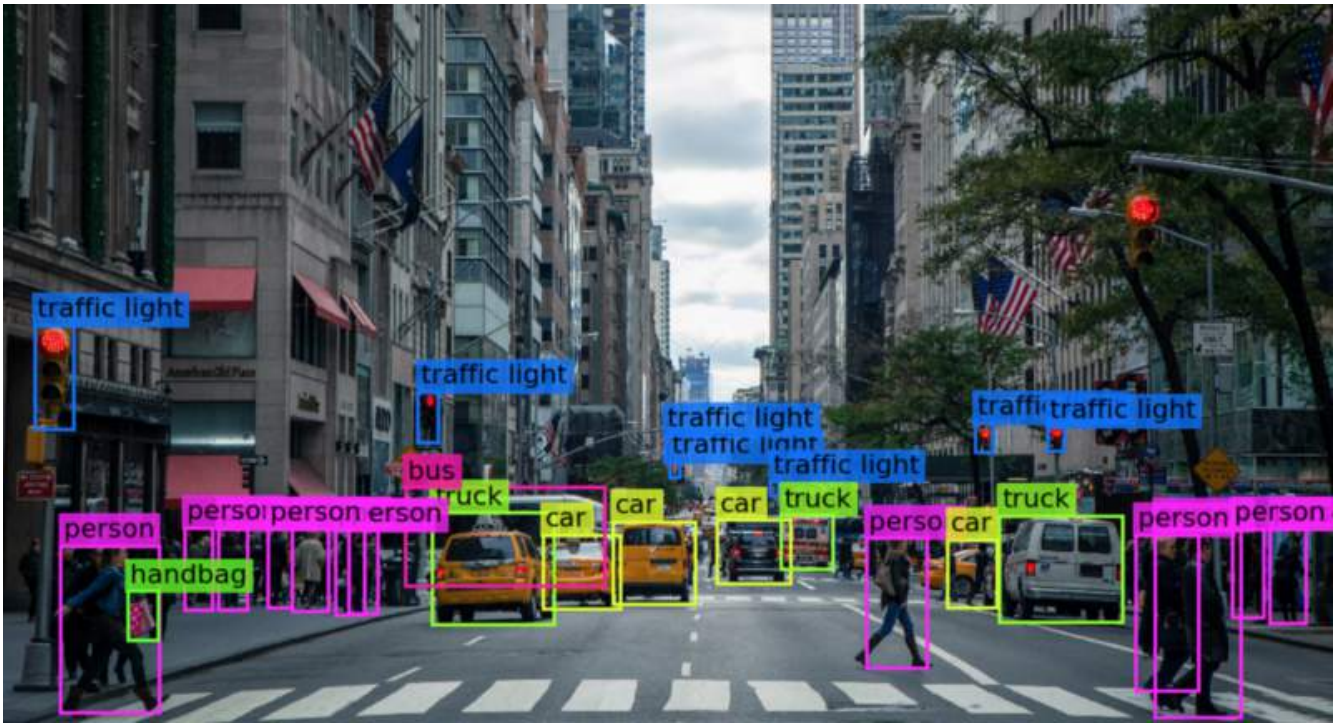
1η Εργαστηριακή Άσκηση: Εντοπισμός Σημείων Ενδιαφέροντος και Εξαγωγή Χαρακτηριστικών σε Εικόνες

Ευάγγελος Γεωργακίλας (03116047)

Μιχαέλα Αικατερίνη Κωνσταντίνου (03117004)

Απρίλιος 2021

Περιεχόμενα



Υλοποίηση κώδικα

Η ανάπτυξη του κώδικα Python έγινε στο cloud Google colaboratory. Για το τρέξιμο του κώδικα αρκεί κανείς να ικανοποιήσει τα dependencies που περιγράφονται από την εκφώνηση καθώς και να ορίσει κατάλληλα τα paths από τα οποια διαβάζονται οι εικόνες. Εμείς τις αποθηκεύσαμε το Drive και κάνοντάς το mount είχαμε πρόσβαση σε αυτές. Επίσης έχει χρησιμοποιηθεί markdown για την καλύτερη οργάνωση του κώδικα, το χωρισμό σε sections σύμφωνα και με τα υποερωτήματα της εκφώνησης και την εύκολη περιήγηση στον κώδικα. Προτείνεται η προβολή του .ipynb να γίνει στο Google colab.

1 Μέρος 1: Ανίχνευση Ακμών σε Γκρίζες Εικόνες

Στόχος του πρώτου μέρους είναι η ανίχνευση ακμών στην εικόνα "edgetest_21.png". Για το σκοπό αυτό χρησιμοποιούνται τόσο γραμμικοί όσο μη-γραμμικοί μέθοδοι, με Gaussian ή Laplacian-of-Gaussian φίλτρα. Στη γραμμική μέθοδο η προσέγγιση της Laplacian L της εξομαλυμένης εικόνας $I_\sigma(x, y)$ γίνεται μέσω συνέλιξης της εικόνας I με την LoG, ενώ για τη μη-γραμμική εκτίμηση της Laplacian της $I_\sigma(x, y)$ χρησιμοποιούνται μορφολογικά φίλτρα. Για να γίνει η διαδικασία της επεξεργασίας πιο ρεαλιστική προστέθηκε στην αρχική εικόνα λευκός gaussian θόρυβος με μέση τιμή 0 και τυπική απόκλιση σn για δύο διαφορετικές τιμές του σηματοθορυβικού λόγου PSNR = 10 dB και 20 dB.

1.1 Δημιουργία Εικόνων Εισόδου

1.1.1 Διάβασμα εικόνων

Για το διάβασμα των εικόνων χρησιμοποιούμε την συναρτηση imread του πακέτου open CV με indicator το IMREAD_GRAYSCALE.
Για το τύπωμα, χρησιμοποιούμε την συνάρτηση του module matplotlib "imshow" και μέσω του cmap ορίζουμε την παλέτα των χρωμάτων "gray".
Σημειώνεται πως είναι καλή πρακτική (και το έχουμε εφαρμόσει) να γίνεται normalization στις εικόνες και για αυτό διαιρούμε την τιμή κάθε pixel με τη μέγιστη τιμή (255).

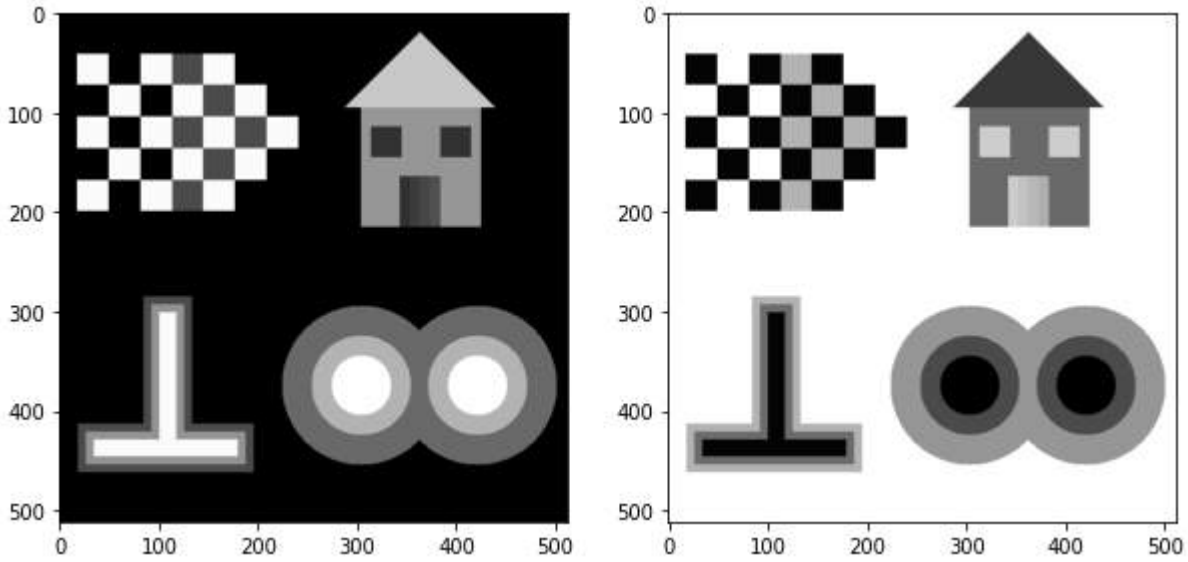


Figure 1: Εικόνα edgetest_10

1.1.2 Δημιουργία θορύβου

Από τον τύπο που μας δίνεται έπειτα από πράξεις παίρνουμε τη σχέση

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

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

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

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

Βάση αυτής για τις δύο τιμές του PSNR βρίσκουμε διαφορετικό σ το οποίο χρησιμοποιούμε για να παράξουμε gaussian θόρυβο τον οποίο προσθέτουμε στις εικόνες.

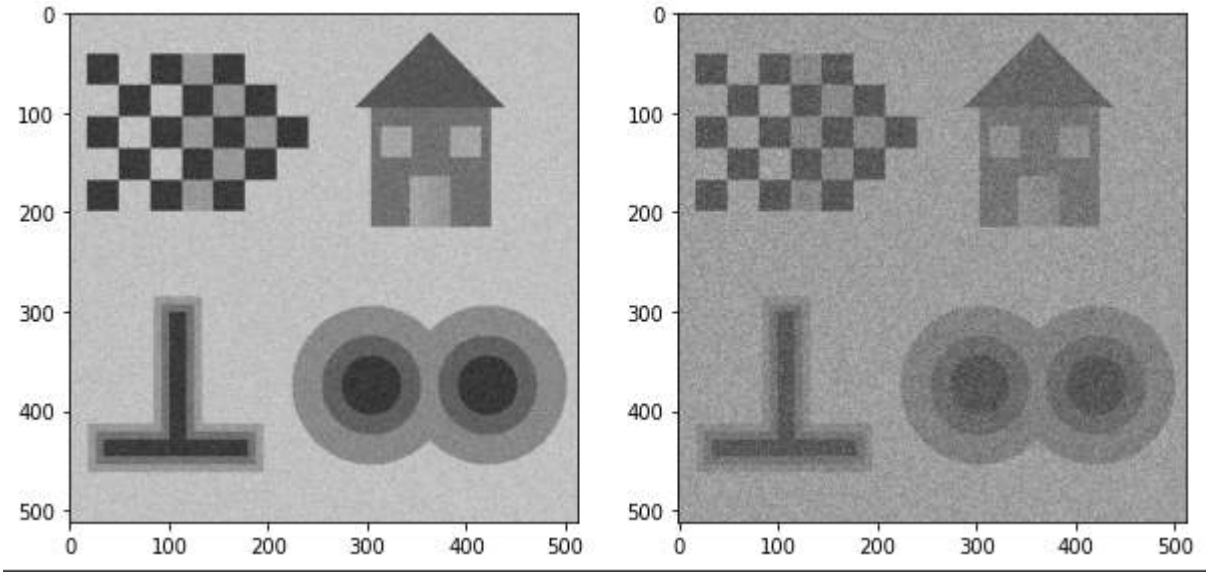


Figure 2: Αποτέλεσμα μετά απο την προσθήκη θορύβου για PSNR 20 και 10 αντίστοιχα

Πράγματι βλέπουμε πως με μείωση του PSNR έχουμε αύξηση του θορύβου *Σημείωση - Clipping* Προκειμένου να μπορούμε να μελετήσουμε τις εικόνες και να μειώσουμε συστηματικά λίγο το θόρυβο εφαρμοζουμε clipping. Εφόσον έχουμε κάνει κανονικοποίηση παίρνουμε για ακραίες τιμές το 0,1.

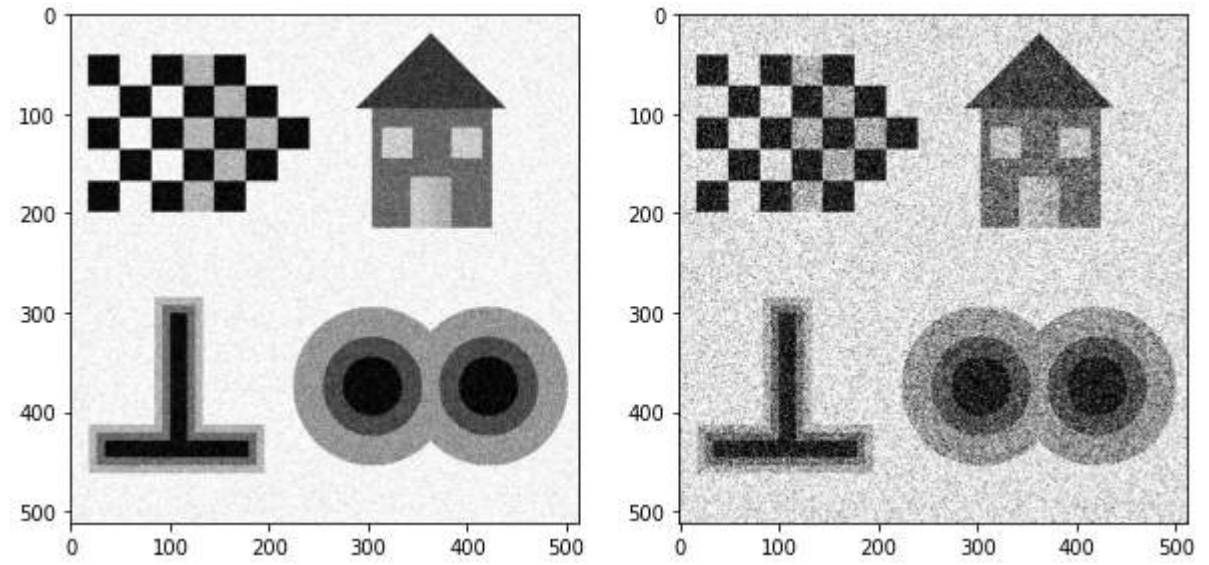


Figure 3: Αποτέλεσμα μετά απο την προσθήκη θορύβου για PSNR 20 και 10 αντίστοιχα και την εφαρμογή clipping

1.2 Υλοποίηση Αλγορίθμων Ανίχνευσης Ακμών

1.2.1 Κρουστικές αποκρίσεις διακριτών γραμμικών φίλτρων

- Διδιάστατη Gaussian $G_\sigma(x, y)$
  Όσο το σ αυξάνεται, οι λεπτομέρειες της εικόνας "θολώνουν" κι έτσι δεν λαμβάνονται υπόψιν στην εξαγωγή τον ακμών της εικόνας. Με αυτόν τον τρόπο μπορούμε να επικεντρωθούμε σε αντικείμενα μεγάλης κλίμακας αλλά χάνουμε την λεπτομέρεια της εικόνας. Αυτό οφείλεται στο γεγονός πως η Gaussian είναι επιθυμητό να προσεγγίζει μια δ-Dirac και κάτι τέτοιο είναι εφικτό μέσο επιλογής μικρού σ. Θα δοκιμάσουμε διάφορες τιμές για το σ τα αποτελέσματα των οποίων φαίνονται παρακάτω

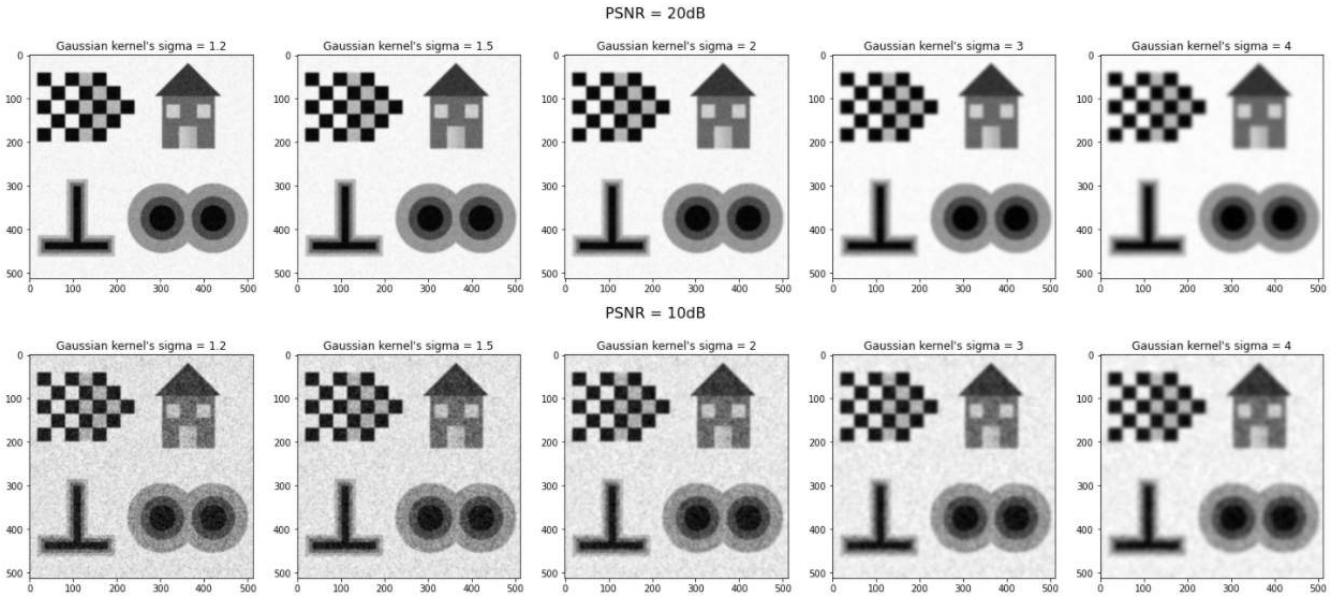


Figure 4: Gauss smoothing για διάφορα σ

- Laplacian-of-Gaussian (LoG) $h(x, y) = \nabla^2 G_\sigma(x, y)$
  Για τη Laplacian of Gaussian έχουμε υλοποιήσει δύο συναρτήσεις. Μία lossy version η οποία υλοποιείται από τη δημιουργία και έπειτα τη συνέλιξη ενός γκαουσιανού και ενός λαπλασιανού φίλτρου και μία optimized μέθοδο που προκύπτει ως μαθηματικός τύπος από τη διαφόριση του τύπου του gaussian φίλτρου. Έτσι δημιουργούμε έναν πυρήνα με το meshgrid και τον συνελίσσουμε μέσω της συνάρτησης του open CV, filter_2D. Τα αποτελέσματα φαίνονται παρακάτω

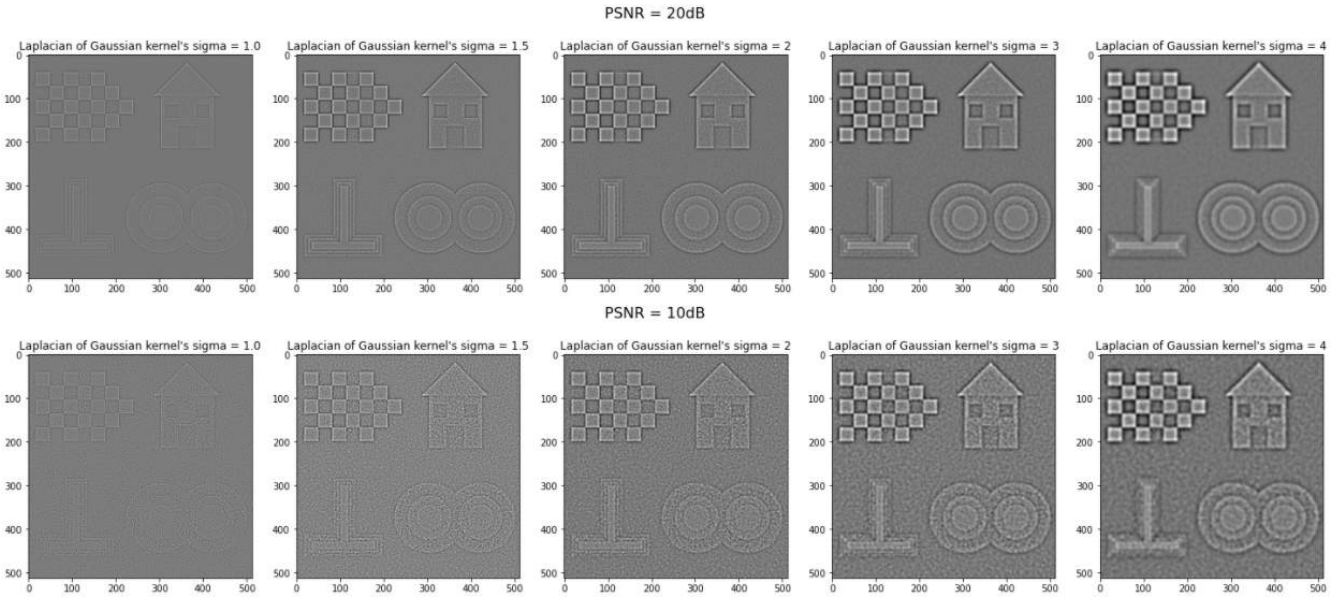


Figure 5: Laplacian of Gaussian (linear)

1.2.2 Non linear approximation of Laplacian with morphology filters

Τέλος, υλοποιούμε την LoG μέσω μη γραμμικής προσέγγισης του Laplacian φίλτρου (και άρα της παραγώγισης) από μορφολοικά φίλτρα.

Απο τα αποτελέσματα συγκρινόμενα με αυτά της απλής LoG παραπάνω ήδη παρατηρούμε την καλύτερη επίδοσή τους αφού οι ακμές εμφανίζονται εντονότερες και με μεγαλύτερη διαφορά (χρώματος) από τα υπόλοιπα pixels, όπως φαίνεται στις εικόνες που ακολουθούν.

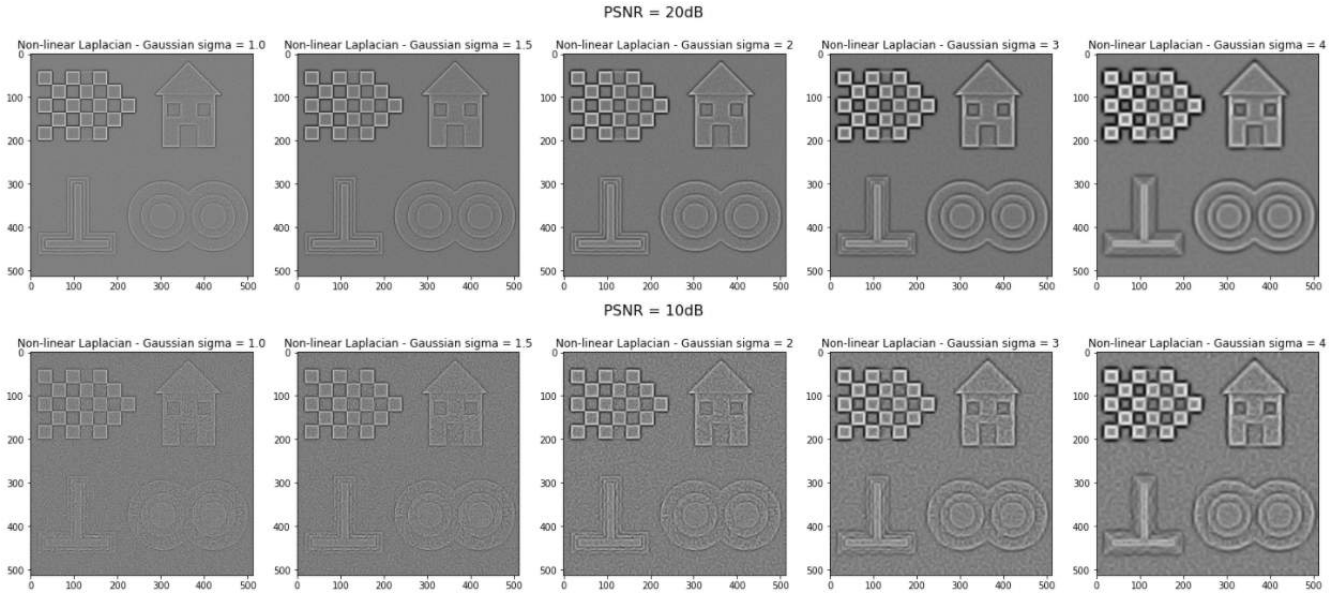


Figure 6: Laplacian of Gaussian - morphology filters(non linear)

1.2.3 Zero-crossing

Τα zero-crossing είναι τα σημεία μηδενισμού της L και για να τα προσεγγίσουμε κατασκευάζουμε τη συνάρτηση zerocrossing. Βρίσκουμε αρχικά το περίγραμμα Y της δυαδικής εικόνας προσήμου X της L μέσω της συνάρτησης image_outline και εντοπίζουμε τα σημεία όπου η δυαδική (πλέον) εικόνα Y έχει την τιμή 1.

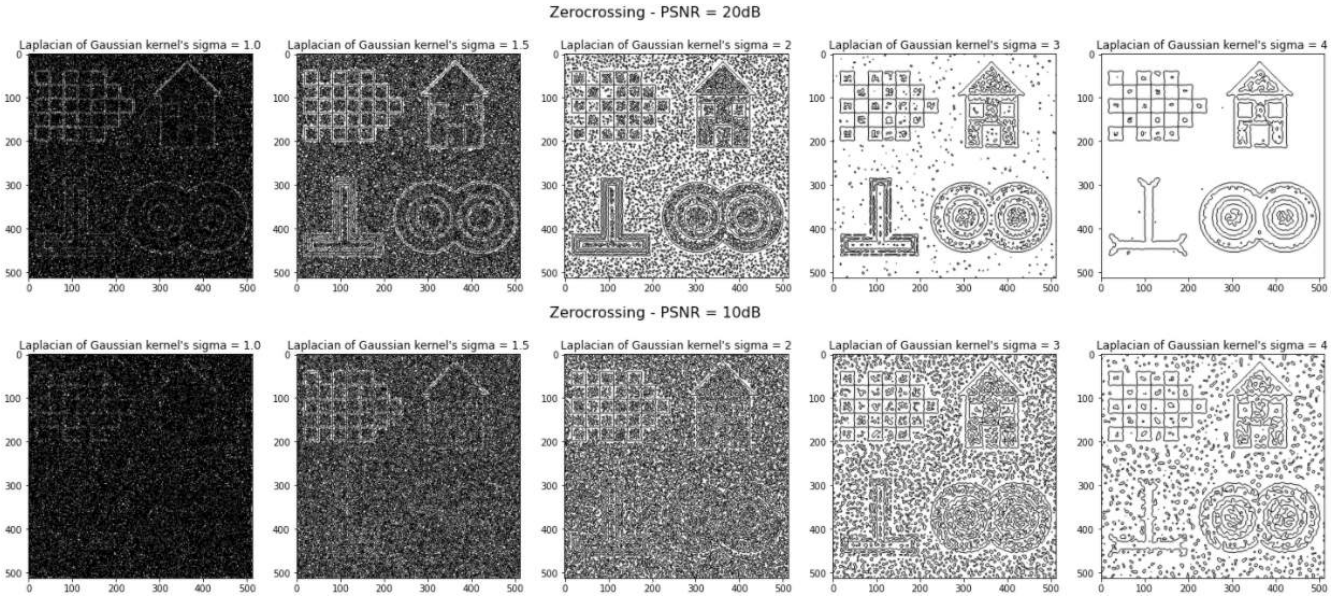


Figure 7: Zerocrossing (linear)

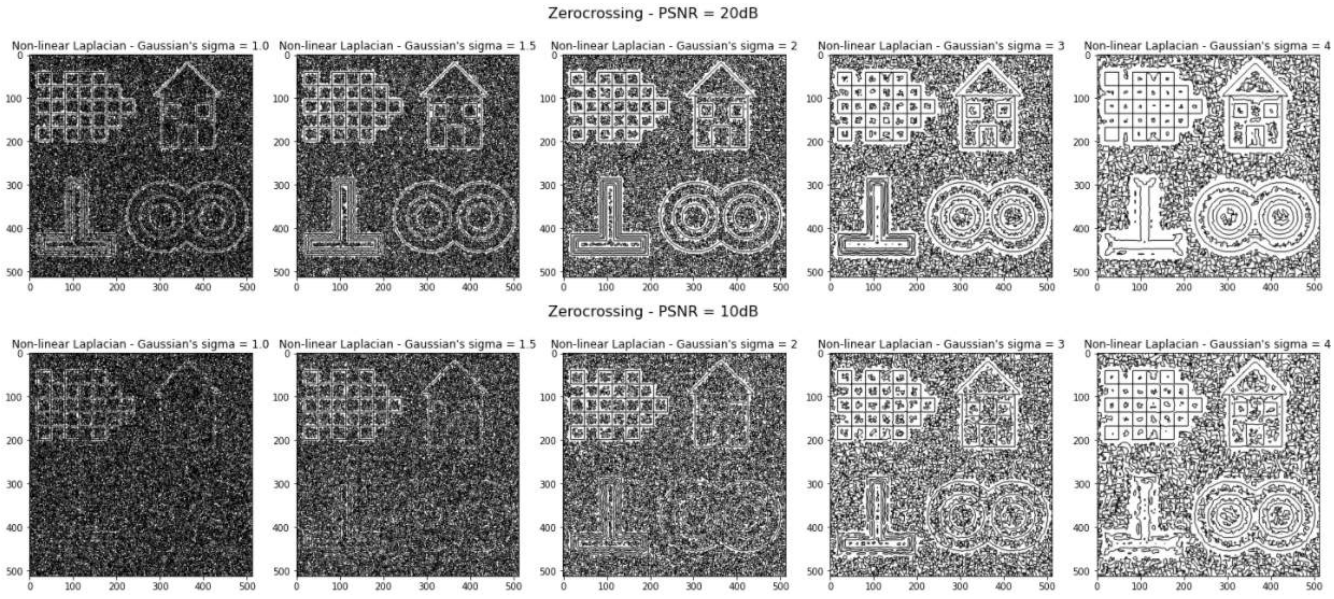


Figure 8: Zerocrossing (non linear)

1.2.4 Threshold - Gradient

Από τα zero-crossing σημεία που έχουμε εντοπίσει χρησιμοποιούμε μία threshold παράμετρο theta με σκοπό τελικά να κρατήσουμε τα zero-crossing σημεία στα οποία η εξομαλυμένη εικόνα Ισ έχει μεγάλη κλίση. Έτσι κατασκευάζουμε τη συνάρτηση threshold η οποία για διάφορα ορίσματα theta γυρνάει τα zero-crossings της Ισ για τα οποία η ευκλείδεια νόρμα της παραγώγου της Ισ στα συγκεκριμένα σημεία είναι μεγαλύτερη του γινομένου theta * της max νορμας στα ίδια σημεία.

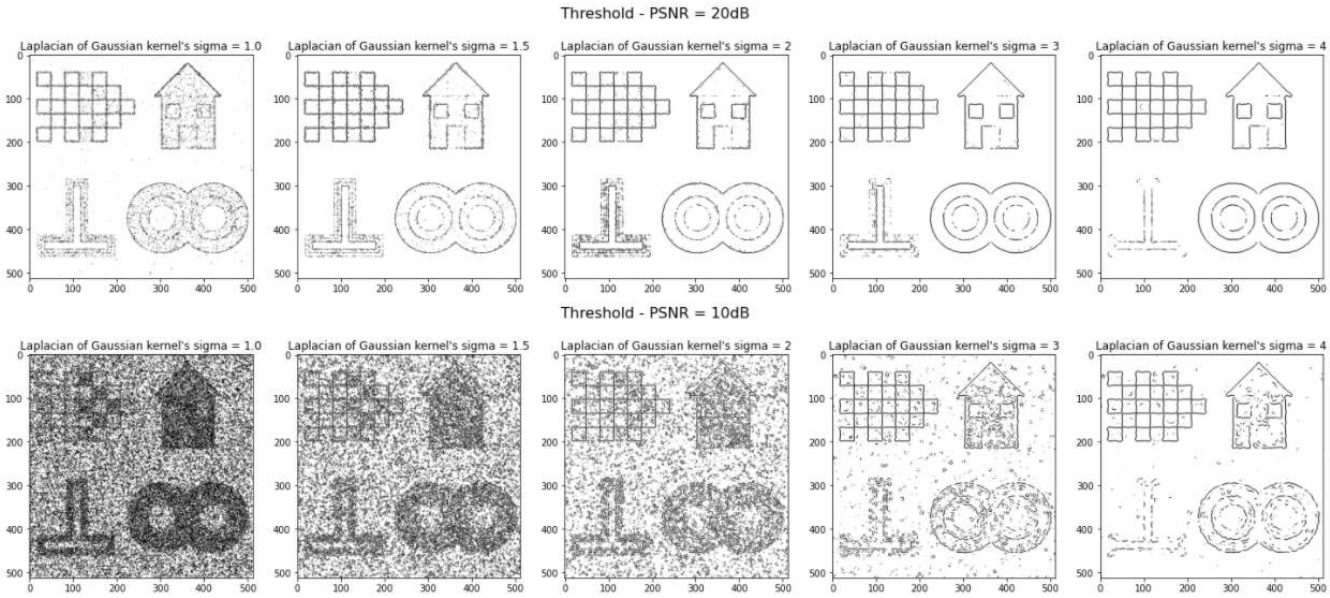


Figure 9: Threshold (linear)

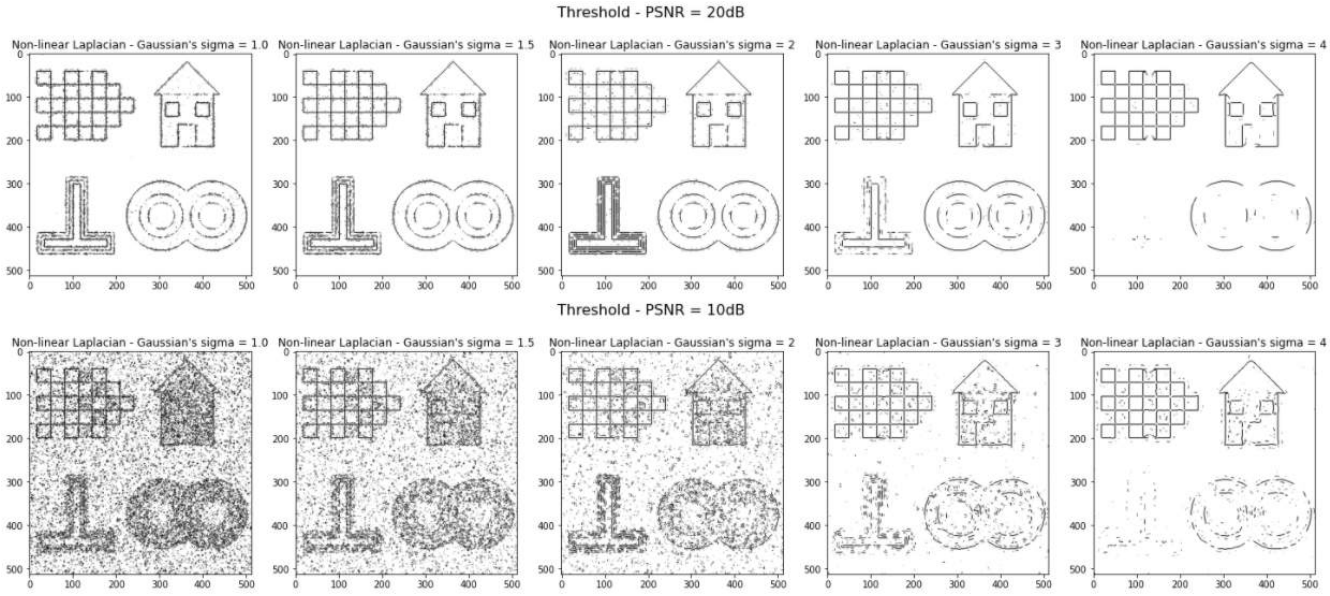


Figure 10: Threshold (non linear)

1.2.5 EdgeDetect()

Συγκεντρώνοντας όλα τα παραπάνω βήματα σε μία συνάρτηση ορίσαμε την EdgeDetect η οποία μπορεί και γυρνάει ένα array το οποίο έχει 1 στα pixel που είναι μέρος ακμών. Ως παραμέτρους αυτή η συνάρτηση δέχεται την τυπική απόκλιση του smoothing gaussian filter σ, το κατώφλι ϑ και τον τύπο της λαπλασιανής ('non-linear', 'linear', 'lossylinear'). Για εύκολη εποπτεία των αποτελεσμάτων συναρτήσει των "υπερπαραμέτρων" ϑ,σ και για εξαντλητική αναζήτηση του βέλτιστου ζεύγους τιμών τους υλοποιήσαμε και ένα intercative widget που μπορεί κανείς να χρησιμοποιήσει σε notebook.

1.3 Αξιολόγηση των Αποτελεσμάτων Ανίχνευσης Ακμών

Πειραματιζόμενοι με διάφορες τιμές των παραμέτρων των αλγορίθμων καταλήξαμε σε αρκετά ενδιαφέροντα συμπεράσματα.
1. για μεγαλύτερη τιμή του PSNR και άρα χαμηλότερες τιμές θορύβου (μικρότερη απόκλιση από την μέση τιμή που είναι το 0) όπως είναι λογικο λαμβάνουμε καλύτερα αποτελέσματα.
2. για μεγάλα ϑ ή\και μεγάλα σ, πολλές ακμές παραλείπονται και στα αντικείμενα μικρής κλίμακας δεν εντοπίζονται τα περιγράμματά τους
3. για μικρά ϑ εκτός από τα εξωτερικά περιγράμματα περιλαμβάνονται και πολλές ακμές που αποτελούν ουσιαστικά αλλαγή χρώματος εσωτερικά των σχημάτων (και λόγω θορύβου) και ιδανικά δεν θα θέλαμε να περιλαμβάνονται.

1.3.1 Υπολογισμός των αληθινών ακμών

Ορίζουμε τη συνάρτηση realEdge με παράμετρο το κατώφλι ϑ η οποία χρησιμοποιεί την ήδη υλοποιημένη (στο βήμα "*1.2.3 Zero-crossing*") συνάρτηση image_outline η οποία επιστρέφει το dilasion μείον το erosion μιας εικόνας που παίρνει ως όρισμα.

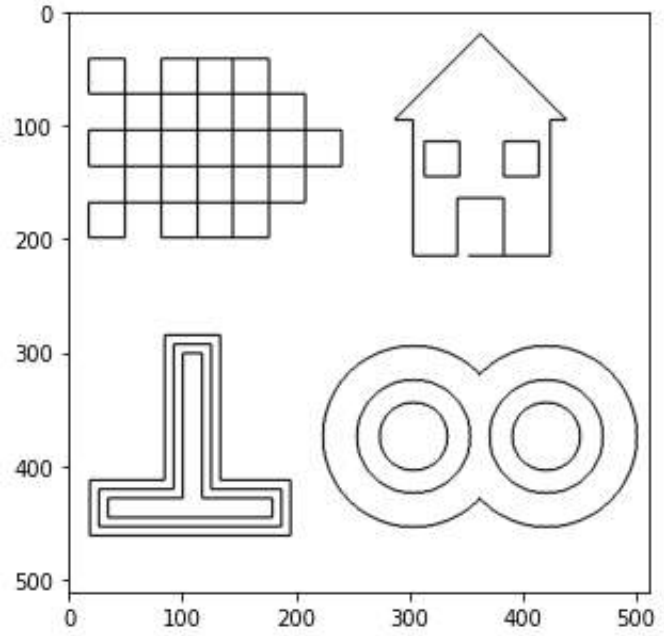


Figure 11: Real edge detection

1.3.2 Evaluation of EdgeDetect()

Για το evaluation υλοποιήσαμε τον πιθανοτικό τύπο που μας δινόταν. Λάβαμε ικανοποιητικά αποτελέσματα για το detection που υλοποιήσαμε, της τάξης του 0.85 για μη γραμμική μέθοδο και της τάξης του 0.7 για τη γραμμική. Όπως είχαμε σχολιάσει και σε προηγούμενα βήματα αυτή η διαφορά μεταξύ γραμμικής και μη γραμμικής μεθόδου ήταν αναμενόμενη.

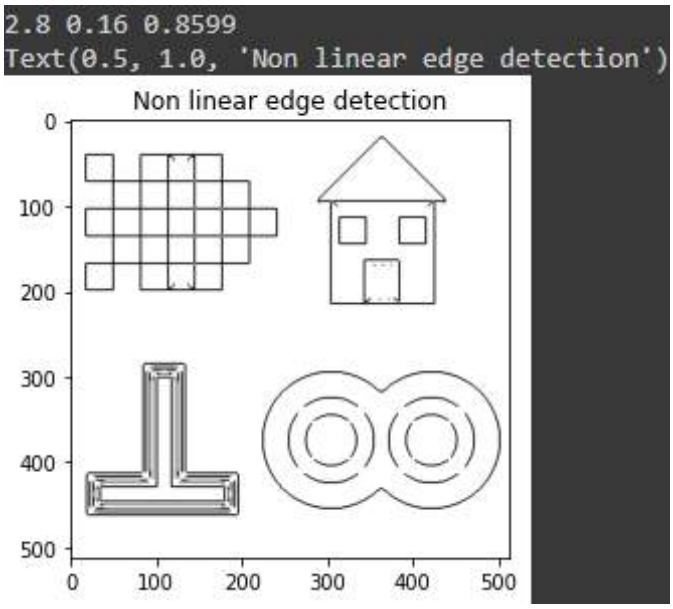


Figure 12: Αξιολόγηση edge detection

1.3.3 Fine tuning of sigma, theta parameters

Ορίζουμε και εφαρμόζουμε μια συνάρτηση "grid search" όπου επιστρέφονται οι βέλτιστες τιμές για τις παραμέτρους. Αξίζει να σημειωθούν δύο πράγματα.
Πρώτο και κύριο, οι βέλτιστες τιμές εμφανίζονται ανά ζεύγη και όχι ανεξάρτητα μεταξύ του και για αυτό κάνουμε εξαντλητική αναζήτηση στο χώρο των παραμέτρων.
Δεύτερον, λόγω της τυχαιότητας του θορύβου που προσθέτουμε υπεισέρχεται τυχαιότητα και στα αποτελέσματά μας. (θα μπορούσε να αντιμετωπιστεί με παράμετρο seed στον ορισμό του λευκού θορύβου αλλά θεωρήσαμε ρεαλιστικότερη αυτήν την εκδοχή). Ωστόσο, οι βέλτιστες τιμές του evaluation δεν αποκλίνουν σημαντικά μεταξύ τους και στον παρακάτω πίνακα φαίνονται ενδεικτικά οι βέλτιστες τιμές για 4 τρεξίματα του αλγορίθμου.

1.4 Εφαρμογή των Αλγορίθμων Ανίχνευσης Ακμών σε Πραγματικές εικόνες

1.4.1 Real image "urban_edges.jpg"

Εφαρμόζουμε τις ήδη υλοποιημένες συναρτήσεις σε φωτογραφία του πραγματικού κόσμου.

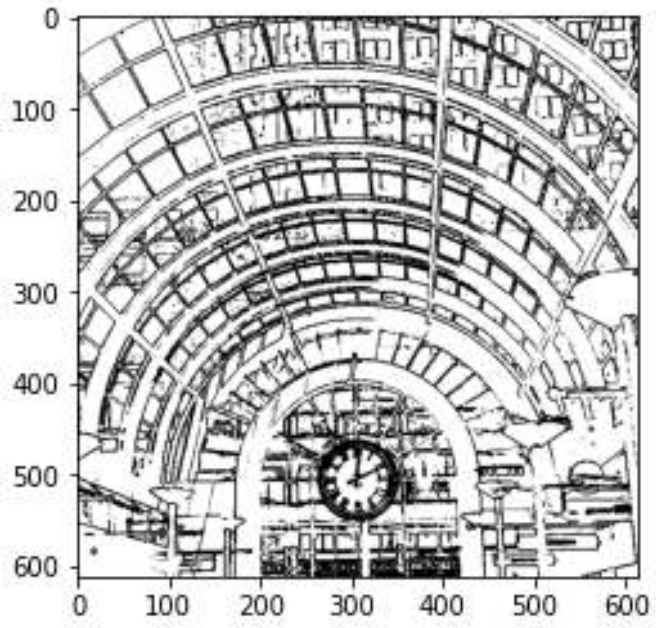


Figure 13: Real image(urban_edges) real edges

1.4.2 Fine tuning of sigma, theta parameters

Αναζητούμε το βέλτιστο ζεύγος τιμών όμοια με το μέρος 1.3.3. Επομένως, για $\sigma = 0.9$, $\vartheta = 0.15$ και μη γραμμική μέθοδο παίρνουμε evaluation ίσο με 0.7552 και ακμές που φαίνονται παρακάτω.

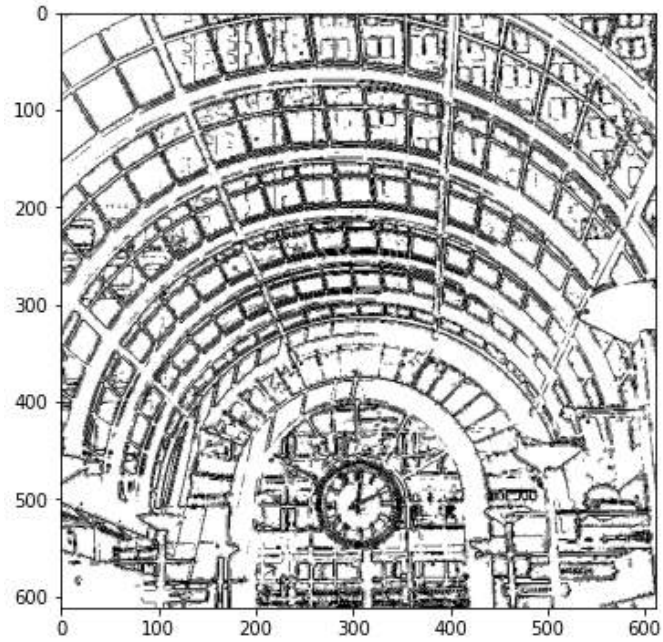


Figure 14: Real image(urban_edges) edge detection fine tuned parameters

Ενώ στη γραμμική μέθοδο παίρνουμε βέλτιστο αποτέλεσμα (evaluation 0.6815) για $\sigma = 1.2$ και $\vartheta = 0.15$ ενώ βέλτιστο evaluation εχουμε για το ίδιο ζεύγος σ, ϑ και για τη lossy μέθοδο η οποία μας δίνει 0.6678.

2 Μέρος 2: Ανίχνευση Σημείων Ενδιαφέροντος (Interest Point Detection)

Στόχος του δεύτερου μέρους είναι η ανίχνευση σημείων ενδιαφέροντος στις εικόνες "blood_smear.png" και "mars" με διάφορες μεθόδους. Η υλοποίηση των αλγορίθμων ανίχνευσης σημείων ενδιαφέροντος γίνεται σε ασπρόμαυρες εικόνες και διακρίνεται σε δύο στάδια. Το πρώτο στάδιο αφορά την εύρεση σημείων της εικόνας τα οποία έχουν έντονη πληροφορία που είναι πολύ χρήσιμη κατά τη διαδικασία κωδικοποίησης της εικόνας, και το δεύτερο στάδιο αφορά την εξαγωγή κάποιων χαρακτηριστικών (feature extraction) για το εκάστοτε σημείο που γίνεται συνήθως με χρήση τοπικών περιγραφητών (local descriptors).Σημεία ενδιαφέροντος θεωρούνται :
Οι γωνίες της εικόνας, γιατί εκεί παρατηρείται έντονη αλλαγή της φωτεινότητας προς όλες τις κατευθύνσεις.
Τα blobs, δηλαδή περιοχές που ξεχωρίζουν έντονα από το background τους, και
σημεία μεγάλης εντροπίας.
Τα σημεία αυτά θέλουμε να είναι ανεξάρτητα τόσο σε περιστροφές όσο σε μικρές αφινικές παραμορφώσεις και αλλαγές στην κλίμακα της εικόνας.

2.1 Ανίχνευση Γωνιών

2.1.1 Στοιχεία J_1, J_2 και J_3

Αν κινηθούμε με ένα τετραγωνικό block προς οποιαδήποτε κατεύθυνση πάνω σε μία εικόνα, περνώντας από μια γωνία θα μας δώσει μεγάλη διαφορά στη φωτεινότητα προς κάθε κατεύθυνση. Σε περιοχή που δεν περιέχει γωνία η εικόνα μας δε θα δούμε καμία αλλαγή στη φωτεινότητα.
Σε περίπτωση ακμής κατά την κατακόρυφη κίνηση πάλι δε εντοπίζονται αλλαγές, ενώ στην οριζόντια μετατόπιση παρατηρούνται πολλές αλλαγές στη φωτεινότητα.
Αν θεωρήσουμε μία μικρή μετατόπιση του τετραγωνικού αυτού block παρατηρούμε μεταβολή στην ενέργειά του E.
Εκφράζουμε τη μεταβολή στην Ενέργεια E(u,v) (με ένα Taylor expansion) μέσω του τελεστή A, ο οποίος είναι ένας 2x2 πίνακας για κάθε pixel που έχει άμεση σχέση με τις παραγώγους της εικόνας στο pixel αυτό.
Τώρα για οποιαδήποτε μικρή μετατόπιση στην ενέργεια E μπορούμε μέσω του τελεστή A να καταλάβουμε αν το σημείο της εικόνας στο οποίο γίνεται η αλλαγή είναι γωνία, ακμή ή τίποτα από αυτά.
Για να το κάνουμε αυτό **βρίσκουμε τις ιδιοτίμες λ1 και λ2 του τελεστή πίνακα A**. Οι ιδιοτιμές αυτές δείχνουν τις κατευθύνσεις μεγαλύτερης αλλαγής στην εικόνα. Οπότε, για μικρή αλλαγή και στις δύο ιδιοτιμές σημαίνει ότι βρισκόμαστε σε μία flat περιοχή. Από την άλλη, μεγάλη αλλαγή προς μία μόνο κατεύθυνση δηλαδή λ2 » λ1 ή λ1 » λ2, σημαίνει ότι στο σημείο αυτό έχουμε ακμή (κατακόρυφη ή οριζόντια).
Τέλος, αν έχουμε μεγάλη αλλαγή και στις δύο κατευθύνσεις συμπεραίνουμε πως πρόκειται για γωνιακό σημείο.
Το ίδιο ακριβώς προσομοιάζουμε με τον τανυστή J του οποίου τα στοιχεία J1, J2 και J3 ισούται με τη συνέλιξη δισδιάστατων Gaussian πυρήνων ομαλοποίησης με παραγώγους της I_σ, εξομαλυμένης εικόνας.

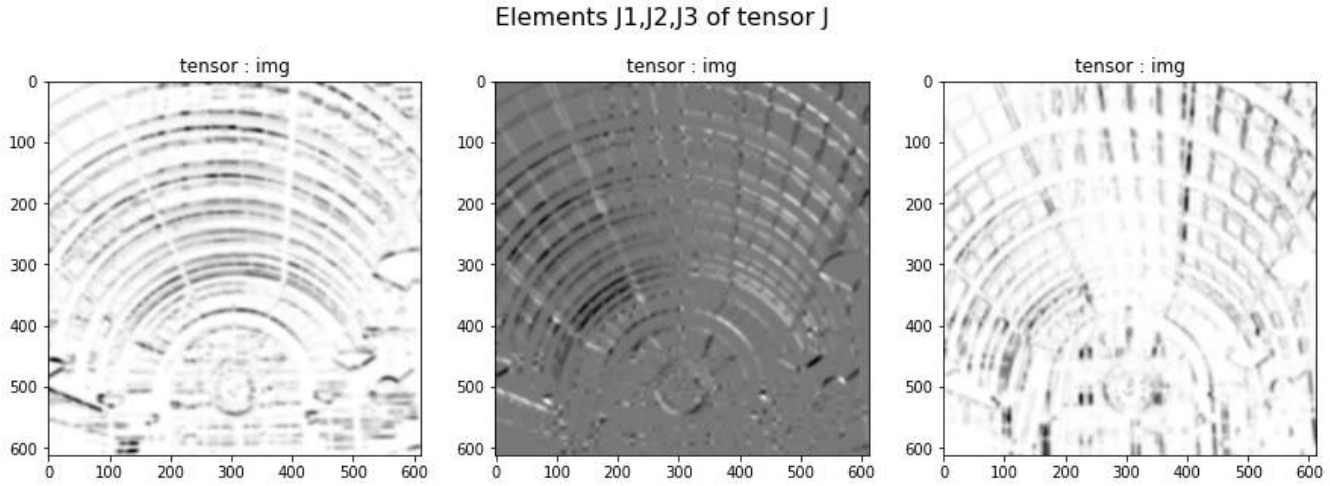


Figure 15: Στοιχεία J

2.1.2 Ιδιοτιμές λ_-, λ_+

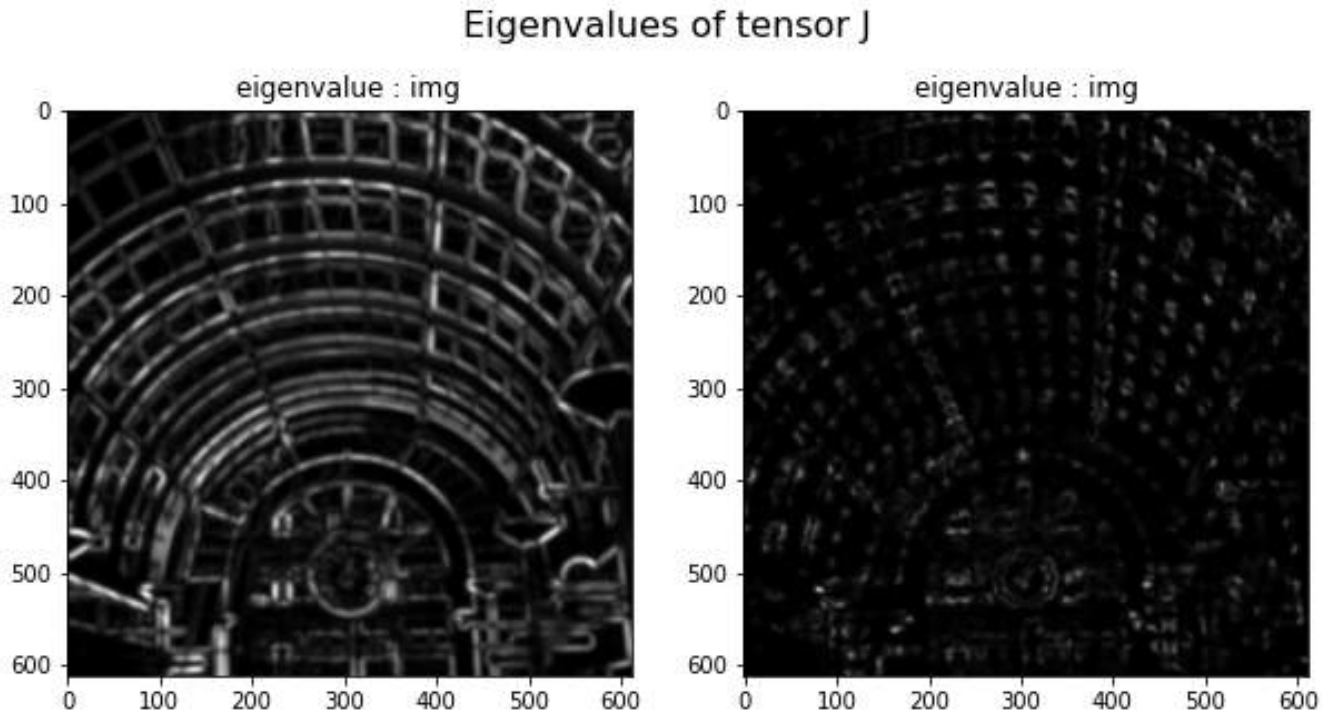


Figure 16: Ιδιοτιμές λ_+, λ_-

2.1.3 Cornerness Criterion

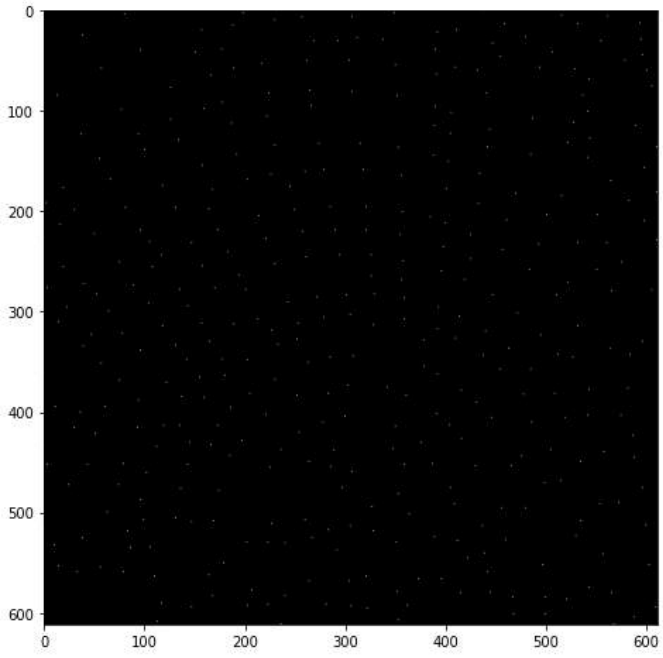


Figure 17: Σημεία - Γωνίες φωτογραφίας urban_edges

2.1.4 Cornerss detection in one scale

Για την τελική επιλογή των σημείων ενδιαφέροντος ταυτόχρονα με το κριτήριο R που περιγράφηκε παραπάνω απαιτούμε να πληροίται και το κριτήριο του μερικού μεγίστου. Πιο συγκεκριμένα, απαιτούμε ένα μερικό μέγιστο να δίνει τιμή μεγαλύτερη από ένα ποσοστό (ϑ) του ολικού μεγίστου.
Συγκεντρώνοντας όλα τα κριτήρια και βήματα του υποερωτήματος υλοποιούμε τη συνάρτηση **cornersDetect** που μας επιστρέφει έναν πίνακα με 1 στα pixels που αντιστοιχούν σε σημεία ενδιάφεροντος.
Παρακάτω με χρήση της συνάρτησης που μας δίνεται *interest_points_visualization* οπτικοποιούμε τα αποτελέσματα της ανίχνευσης γωνιών για κλίμακα = 2.
Σημειώνουμε πως προκειμένου να έχουμε τα δεδομένα μας στην κατάλληλη μορφή που απαιτείται από τη συνάρτηση οπτικοποίησης έχουμε ορίσει βοηθητική συνάρτηση με όνομα ***cornersToVisualize*** η οποία είναι απαραίτητη και για το 3^o μέρος (και η παραλλαγή της ***cornersToVisualize_HOG*** η οποία αγνοεί τα σημεία κοντά στα περιθώρια της εικόνας).

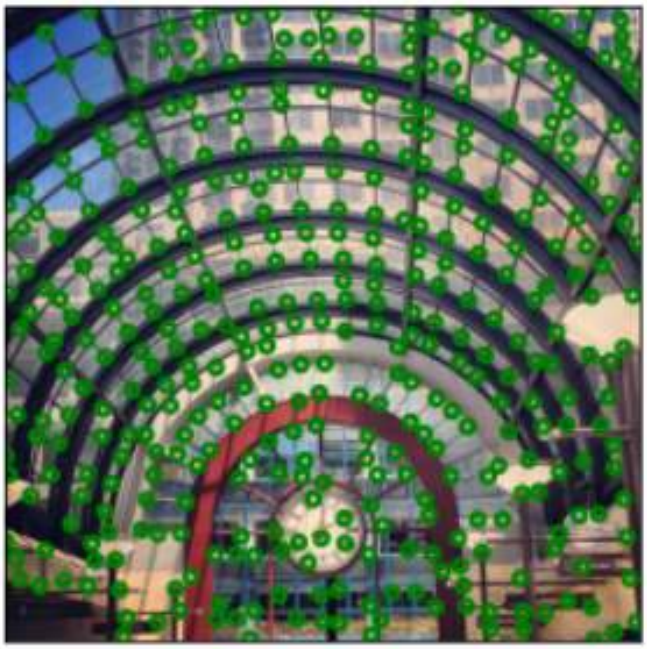


Figure 18: Οπτικοποίηση της εύρεσης γωνιών σε μοναδική κλίμακα

2.2 Πολυκλιμακωτή Ανίχνευση Γωνιών

2.2.1 Αλγόριθμος εύρεσης γωνιών μονής κλίμακας

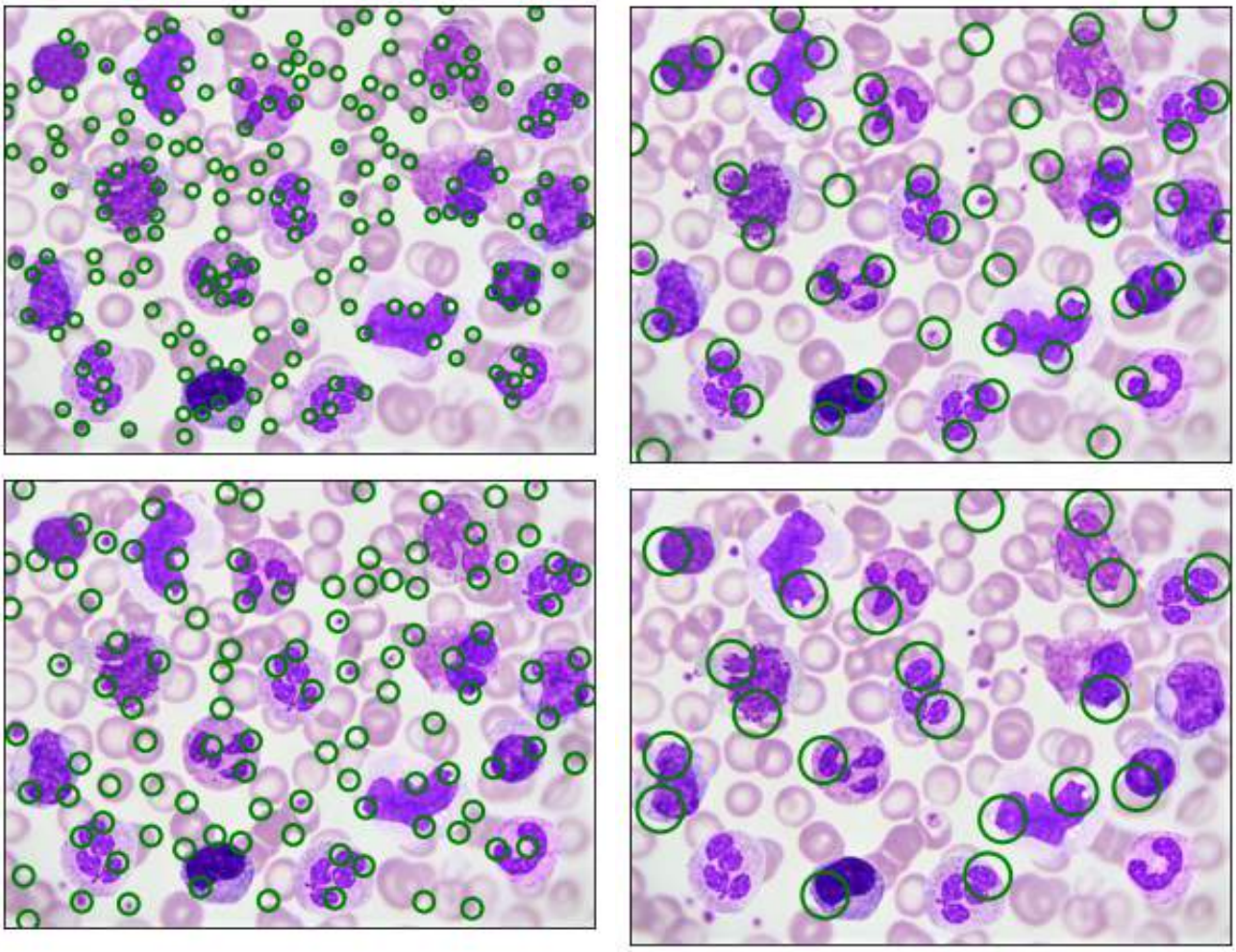


Figure 19: Εύρεση γωνιών σε πολλές κλίμακες, σ = 2, 3, 4.5 και 6.75 και ρ = 2.5, 3.75, 5.625 και 8.4375 αντίστοιχα

2.2.2 Αυτόματη επιλογή χαρακτηριστικής κλίμακας

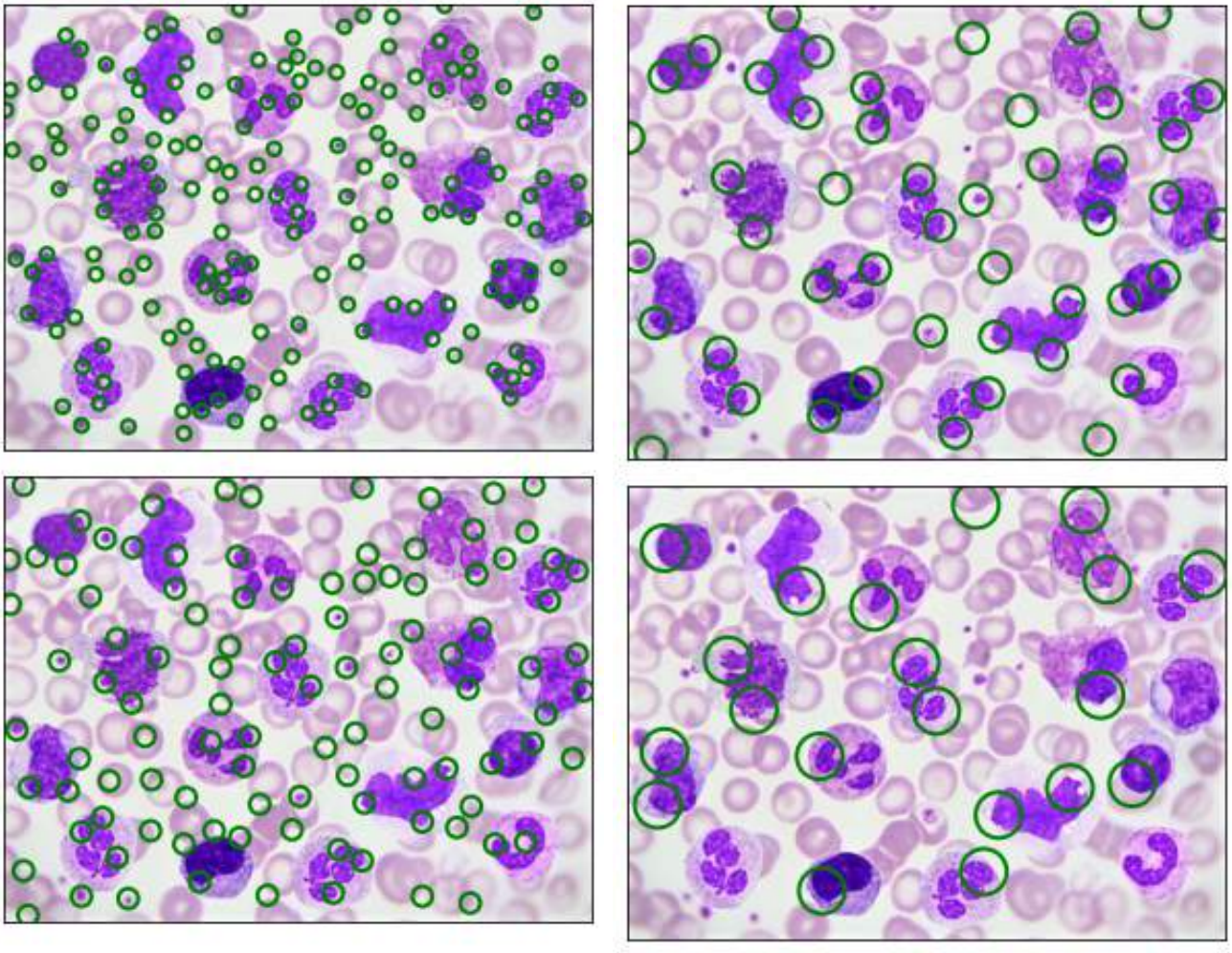
Υλοποιούμε τον τύπο της κανονικοποιημένης Laplacian of Gaussian

$$|LoG(x,\sigma_i)| = \sigma_i^2|L_{xx}(x,\sigma_i) + L_{yy}(x,\sigma_i)|, i = 0, ..., N-1, \text{όπου } L_{ij} = \frac{\partial^2}{\partial i \partial j} I_\sigma(x,y)$$

μέσω της βελτιστοποιημένες εκδοχής του LoG με αντίστοιχο τρόπο με το Μέρος 1. Επίσης έχουμε υλοποιήσει και τη lossy-σειριακή μέθοδο, η οποία έχει απώλειες λόγω ψηφιακής εικόνας (διακριτή) που εισάγει θόρυβο κατά την (διπλή) παραγώγιση. Η παραπάνω συνάρτηση χωρίς απώλειες που χρησιμοποιήθηκε βασίζεται στο μαθηματικό τύπο-προϊόν της συνέλιξης ενός Laplacian και ενός Gaussian

kernel.

Έπειτα, χρησιμοποιούμε το αποτέλεσμα της παραπάνω συνάρτησης ώστε να επιλέξουμε την κλίμακα η οποία μεγιστοποιεί το LoG στο σημείο ενδιαφέροντος.



(a) Lossy normalized LoG

(b) Normalized LoG

Figure 20: Normalized LoG

2.2.3 Πολυκλιμακωτή Ανίχνευση Γωνίων

Συνθέτοντας τα παραπάνω βήματα λαμβάνουμε την πολυκλιμακωτή ανίχνευση γωνιών που φαίνεται παρακάτω. Στην οποία φυσικά έχουν γίνει discard αρκετά σημεία τα οποία οπτικά, όντως, σε κάποιο βαθμό όφειλαν να παραβλεφθούν.

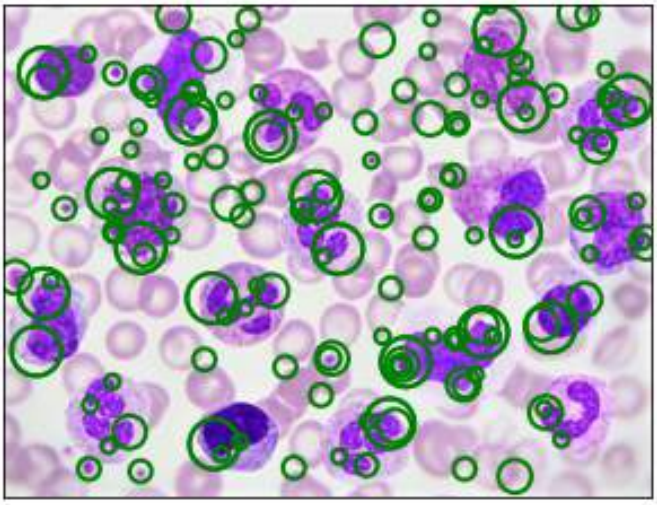


Figure 21: Συνολική ανίχνευση γωνιών και στις 4 κλίμακες ταυτόχρονα

2.3 Ανίχνευση Blobs + Σημεία ενδιαφέροντος - Τοπικά μέγιστα

2.3.1 Μερικές παράγωγοι και κριτήριο R

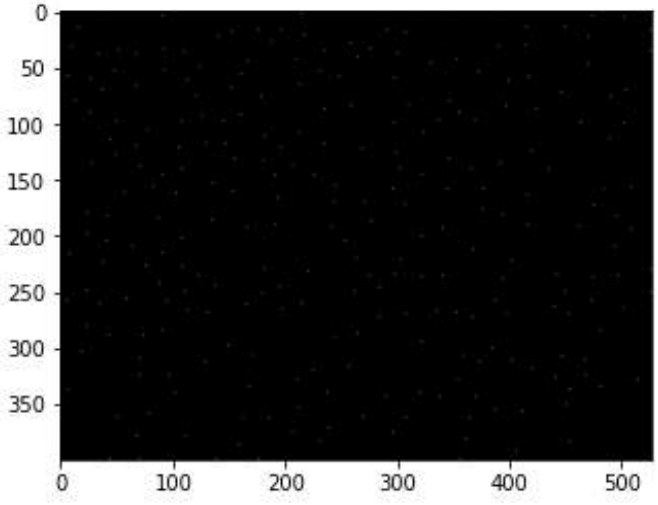


Figure 22: Σημεία - γωνίες στη φωτογραφία blood

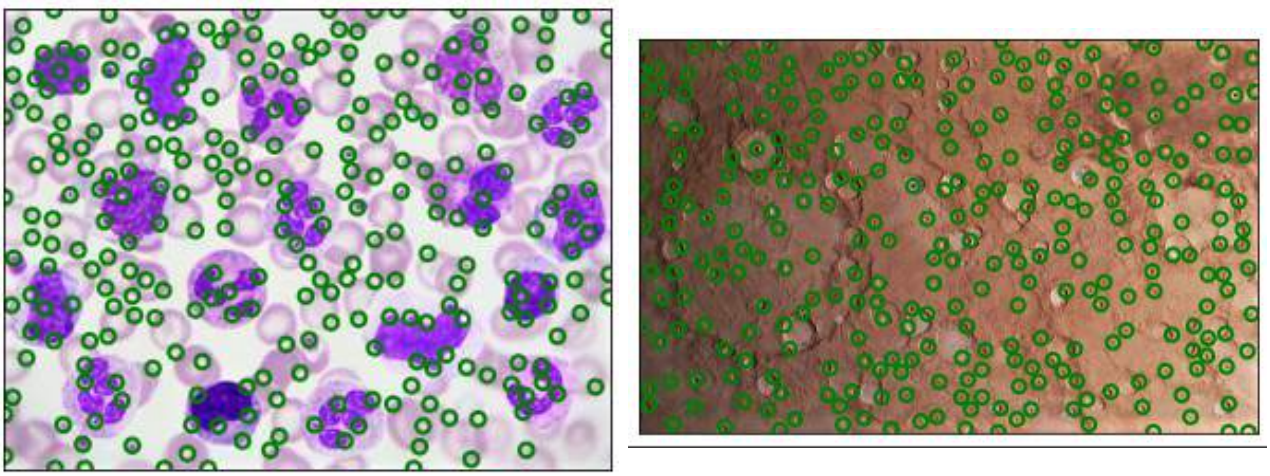


Figure 23: Οπτικοποίηση γωνιών για τις δύο φωτογραφίες με τη μέθοδο Blobs

2.4 Πολυκλιμακωτή Ανίχνευση Blobs

2.4.1 Πολυκλιμακωτή Ανίχνευση + Hessian-Laplace

Σε αυτό το βήμα παρατηρούνται περισσότερα σημεία ενδιαφέροντος (ειδικότερα στις μικρές κλίμακες) από όσα εντοπίστηκαν με την απλή Hessian μέθοδο.

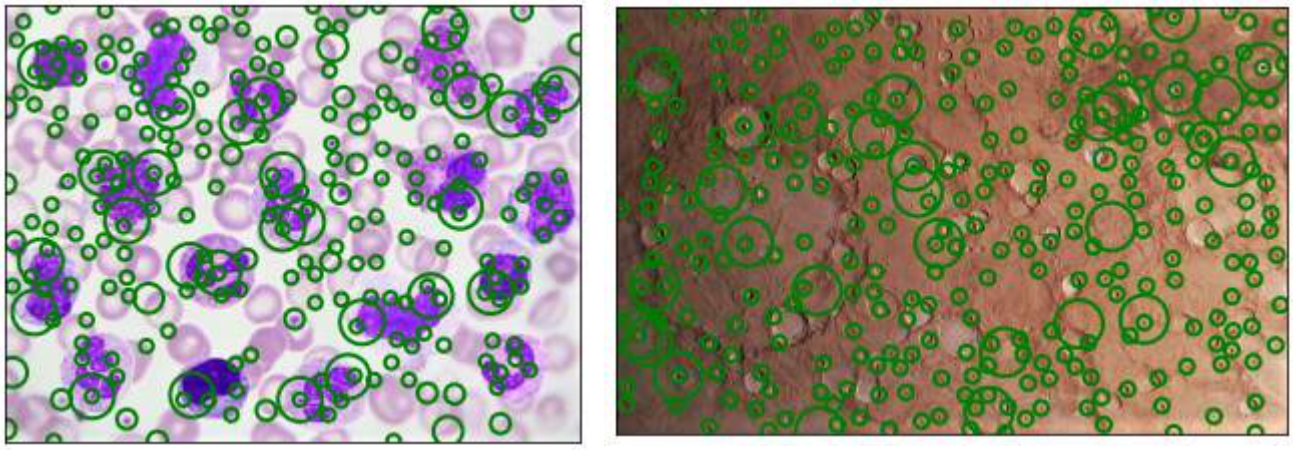


Figure 24: Οπτικοποίηση γωνιών για τις δύο φωτογραφίες με τη μέθοδο Blobs και στις 4 κλίμακες ταυτόχρονα

2.5 Επιτάχυνση με την χρήση Box Filters και Ολοκληρωτικών Εικόνων (Integral Images)

2.5.1 Ολοκληρωτική Εικόνα

Για κάθε pixel υπολογίζεται το άθροισμα των "πάνω" αριστερά pixels μέσω της συνάρτησης cumsum και για τους δύο άξονες του numpy array (εικόνα).

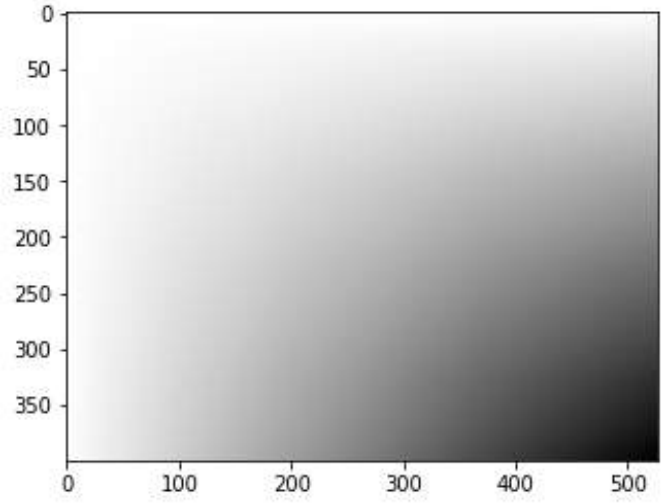


Figure 25: Ολοκληρωτική εικόνα

2.5.2 Υπολογισμός των L_{xx}, L_{xy}, L_{yy} με Box Filters (D_{xx}, D_{xy}, D_{yy})

Υλοποιήσαμε τις ζητούμενες συναρτήσεις και τα αποτελέσματα φαίνονται παρακάτω για κλίμακες 2 και 6.25 αντίστοιχα.

Για την προσέγγιση μέσω Box filters χρησιμοποιήσαμε την integrated εικόνα και zero padding ώστε να έχουμε το επιθυμητό μέγεθος. Έπειτα κάνοντας χρήση των τύπων του παραρτήματος και με κατάλληλο indexing στους numpy πίνακες (εικόνες) υπολογίσαμε τα ζητούμενα αθροίσματα για κάθε pixel με πράξεις πινάκων ώστε να έχουμε ταχύτατα αποτελέσματα.

Η λογική που ακολουθήθηκε ήταν η προσομοίωση του φιλτραρίσματος μέσω συνέλιξης ωστόσο σε όλα τα πιξελς ταυτόχρονα, αυτό ήταν εφικτό καθώς τα αθροίσματα των τιμών κινούμενου παραθύρου (moving sum) είναι προσπελάσιμα σε O(1) λόγω των σταθερών πράξεων πρόσθεσης και αφαίρεσης στα pixels που οριοθετούν το παράθυρο στην αντίστοιχη integral εικόνα.

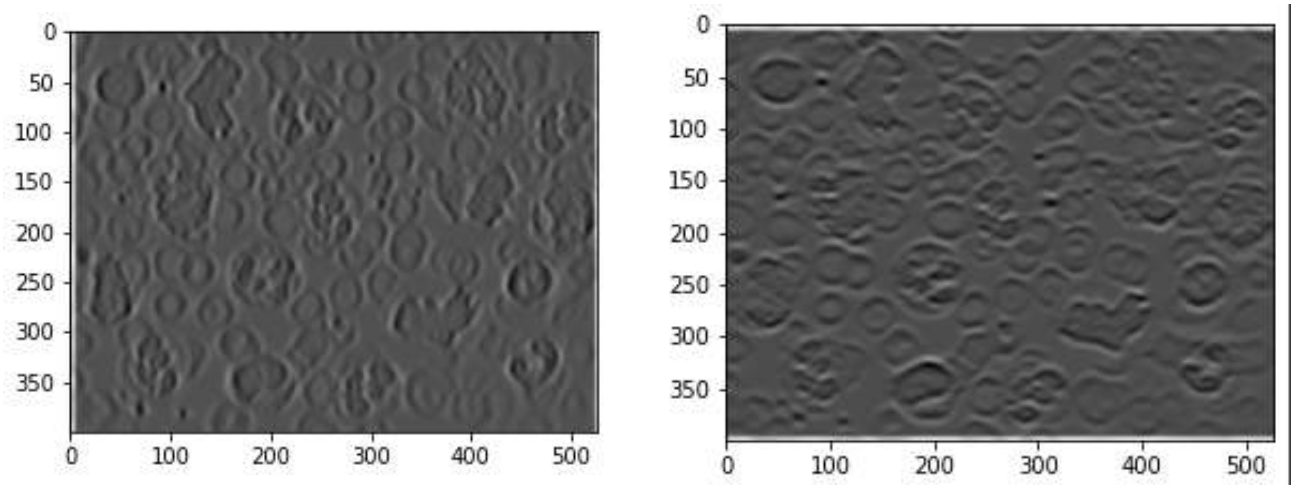


Figure 26: Αποτελέσματα Lxx, Lyy με Box filters Dxx, Dyy

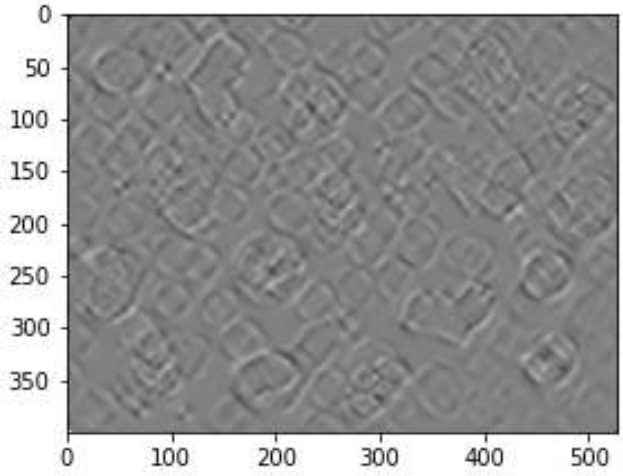


Figure 27: Αποτέλεσμα Lxy με Box filter Dxy

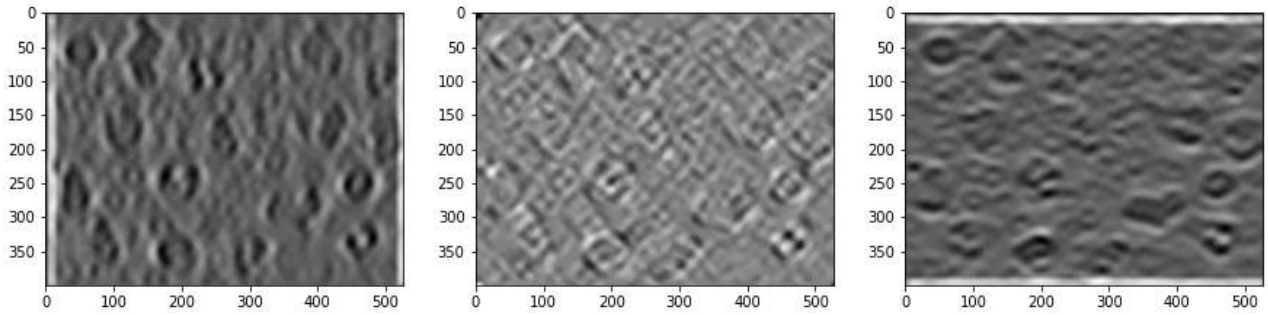


Figure 28: Αποτελέσματα Lxx, Lxy, Lyy με Box filters Dxx, Dxy, Dyy για scale = 6.25

2.5.3 Τοπικά Μέγιστα

Υλοποιούμε το προσεγγιστικό κριτήριο για την ορίζουσα του Hessian και παίρνουμε οπτικοποιημένα το αποτέλεσμα που ακολουθεί. Για σύγκριση, παραθέτουμε και την οπτικοποίηση των κριτηρίων της απλής Hessian μεθόδου και της Blobs (Hessian+Laplacian).

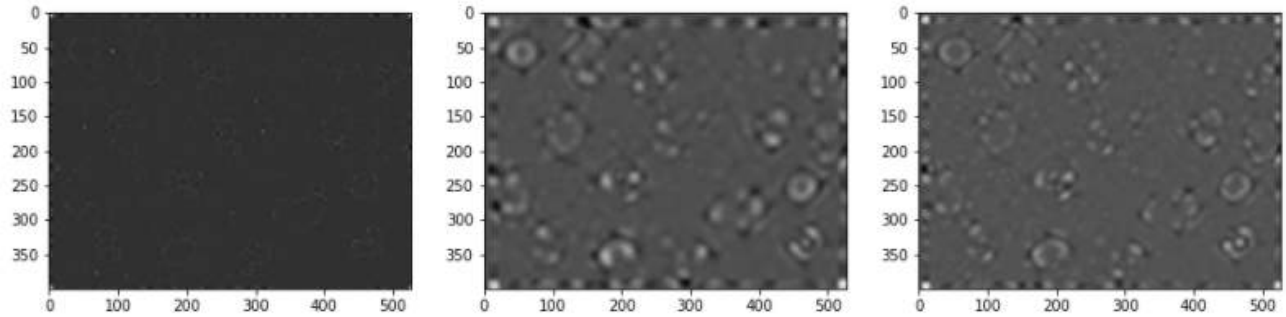


Figure 29: Κριτήριο Box filters για κλίμακες 2, 8 και 6.25

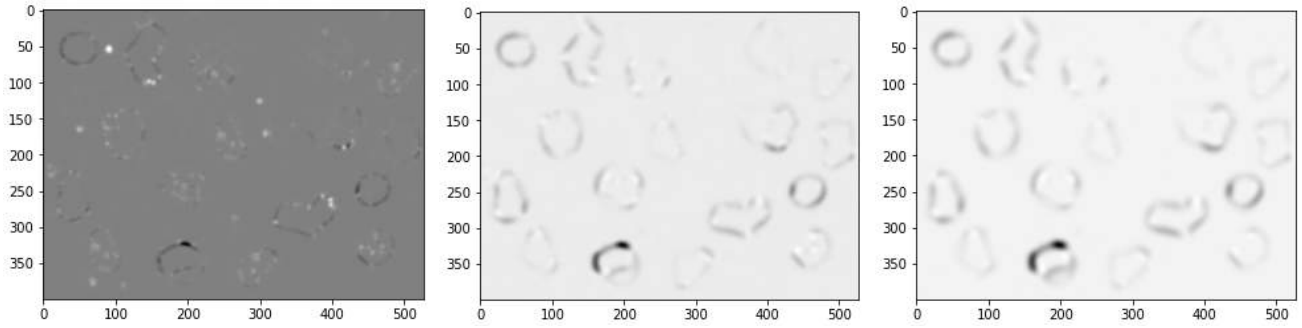


Figure 30: Κριτήριο απλής Hessian για κλίμακες 2, 6.25 και 8

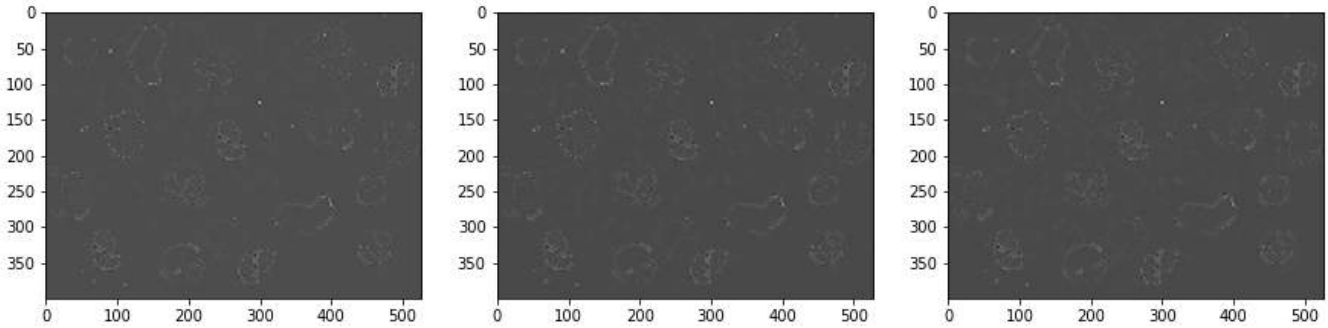


Figure 31: Κριτήριο Blobs για κλίμακες 2, 8 και 6.25

2.5.4 Πολυκλιμακωτή ανίχνευση σημείων ενδιαφέροντος

Ακολουθούμε την ίδια διαδικασία για την επιλογή κλίμακας κάθε σημείου ενδιαφέροντος και συγκεντρώνουμε το σύνολο των σημείων και παίρνουμε τα αποτελέσματα που φαίνονται παρακάτω.
Σημειώνουμε πως παρατηρούμε εντοπισμό λιγότερων σημείων ενδιαφέροντος από το πλήθος που προκύπτει από τον ανιχνευτή Blobs τα οποία όμως οπτικά φαίνονται να είναι πιο accurate ειδικά για αντικείμενα που ξεχωρίζουν σημαντικά από το background τους.

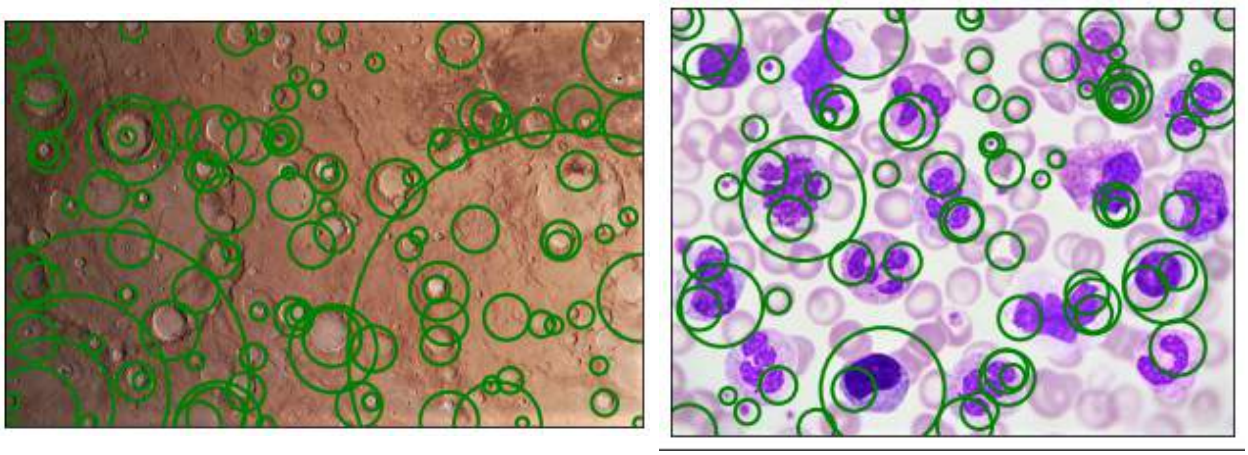


Figure 32: Ανίχνευση σημείων ενδιαφέροντος μέσω Box Filters

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

Στο τελευταίο μέρος της άσκησης συνδυάζουμε δύο τοπικούς περιγραφητές τον SURF και τον HOG για κάθε ένα διαφορετικό ανιχνευτή σημείων ενδιαφέροντος (από το προηγούμενο ερώτημα) αξιολογώντας την αποτελεσματικότητά τους σε ένα πρόβλημα ταιριάσματος εικόνας (matching). Η χρησιμότητα αυτών των τοπικών περιγραφητών έγκειται στη δυνατότητά ταιριάσματος όμοιων σημείων πάνω σε εικόνες που μπορεί να έχουν υποστεί περιστροφή ή και κλιμάκωση. Απώτερος στόχος της επίλυσης του προβλήματος ταιριάσματος των εικόνων, είναι η επίλυση του προβλήματος κατηγοριοποίησης (classification) αυτών ανάλογα με το περιεχόμενό τους σε διάφορες κατηγορίες συγκεκριμένα για την άσκηση μας στις κλάσεις : αυτοκίνητο, άνθρωπος, ποδήλατο.

3.1 Ταίριασμα Εικόνων υπό Περιστροφή και Αλλαγή Κλίμακας

3.1.1 Αποτίμηση matching

Κάνοντας χρήση των 5 ανιχνευτών που υλοποιήσαμε στο 2ο μέρος της εργασίας (παραλείποντας τη μονοκλιμακωτή μέθοδο των BoxFilters) και των δύο τοπικών περιγραφητών SURF και HOG, γίνεται για κάθε συνδυασμό ανιχνευτή-περιγραφητή, μέσω της συνάρτησης matching_evaluation με τις κατάλληλες παραμέτρους η εκτίμηση της κλίμακας και της περιστροφής που έχει εφαρμοσθεί σε κάθε εικόνα.
Κατόπιν, για κάθε περίπτωση υπολογίζεται η απόκλιση της εκτίμησης που βρέθηκε από την πραγματική τιμή της περιστροφής/κλιμακοποίησης. Έτσι, γίνεται τελικά δυνατή η εκτίμηση της απόδοσης του κάθε συνδυασμού και η σύγκριση μεταξύ αυτών. Παρακάτω παρατίθενται τα αποτελέσματα που λάβαμε.

```
SURF descriptor: Avg. Scale Error for Image 1: 0.007
SURF descriptor: Avg. Theta Error for Image 1: 2.992
SURF descriptor: Avg. Scale Error for Image 2: 0.002
SURF descriptor: Avg. Theta Error for Image 2: 0.178
SURF descriptor: Avg. Scale Error for Image 3: 0.065
SURF descriptor: Avg. Theta Error for Image 3: 9.314
HOG descriptor: Avg. Scale Error for Image 1: 0.186
HOG descriptor: Avg. Theta Error for Image 1: 22.619
HOG descriptor: Avg. Scale Error for Image 2: 0.351
HOG descriptor: Avg. Theta Error for Image 2: 19.199
HOG descriptor: Avg. Scale Error for Image 3: 0.285
HOG descriptor: Avg. Theta Error for Image 3: 23.699
```

Figure 33: Αποτελέσματα για απλή Harris μέθοδο (μονής κλίμακας) εκτίμησης σημείων ενδιαφέροντος. Απόκλιση κλίμακας και γωνίας (ϑ) περιστροφής των 3 εικόνων

```
SURF descriptor: Avg. Scale Error for Image 1: 0.002
SURF descriptor: Avg. Theta Error for Image 1: 0.119
SURF descriptor: Avg. Scale Error for Image 2: 0.004
SURF descriptor: Avg. Theta Error for Image 2: 0.402
SURF descriptor: Avg. Scale Error for Image 3: 0.002
SURF descriptor: Avg. Theta Error for Image 3: 0.147
HOG descriptor: Avg. Scale Error for Image 1: 0.153
HOG descriptor: Avg. Theta Error for Image 1: 18.688
HOG descriptor: Avg. Scale Error for Image 2: 0.433
HOG descriptor: Avg. Theta Error for Image 2: 26.303
HOG descriptor: Avg. Scale Error for Image 3: 0.197
HOG descriptor: Avg. Theta Error for Image 3: 13.263
```

Figure 34: Αποτελέσματα για απλή Harris μέθοδο (πολλαπλής κλίμακας) εκτίμησης σημείων ενδιαφέροντος. Απόκλιση κλίμακας και γωνίας (ϑ) περιστροφής των 3 εικόνων

```
SURF descriptor: Avg. Scale Error for Image 1: 0.032
SURF descriptor: Avg. Theta Error for Image 1: 5.529
SURF descriptor: Avg. Scale Error for Image 2: 0.003
SURF descriptor: Avg. Theta Error for Image 2: 0.123
SURF descriptor: Avg. Scale Error for Image 3: 0.012
SURF descriptor: Avg. Theta Error for Image 3: 3.635
HOG descriptor: Avg. Scale Error for Image 1: 0.076
HOG descriptor: Avg. Theta Error for Image 1: 10.897
HOG descriptor: Avg. Scale Error for Image 2: 0.187
HOG descriptor: Avg. Theta Error for Image 2: 16.254
HOG descriptor: Avg. Scale Error for Image 3: 0.235
HOG descriptor: Avg. Theta Error for Image 3: 26.576
```

Figure 35: Αποτελέσματα για Blobs μέθοδο (μονής κλίμακας) εκτίμησης σημείων ενδιαφέροντος. Απόκλιση κλίμακας και γωνίας (ϑ) περιστροφής των 3 εικόνων

```
SURF descriptor: Avg. Scale Error for Image 1: 0.289
SURF descriptor: Avg. Theta Error for Image 1: 35.738
SURF descriptor: Avg. Scale Error for Image 2: 0.544
SURF descriptor: Avg. Theta Error for Image 2: 48.894
SURF descriptor: Avg. Scale Error for Image 3: 0.383
SURF descriptor: Avg. Theta Error for Image 3: 40.170
```

Figure 36: Αποτελέσματα για Blobs μέθοδο (πολλαπλής κλίμακας) εκτίμησης σημείων ενδιαφέροντος. Απόκλιση κλίμακας και γωνίας (ϑ) περιστροφής των 3 εικόνων

```
SURF descriptor: Avg. Scale Error for Image 1: 0.640
SURF descriptor: Avg. Theta Error for Image 1: 43.809
SURF descriptor: Avg. Scale Error for Image 2: 0.441
SURF descriptor: Avg. Theta Error for Image 2: 31.284
SURF descriptor: Avg. Scale Error for Image 3: 0.561
SURF descriptor: Avg. Theta Error for Image 3: 54.506
HOG descriptor: Avg. Scale Error for Image 1: 0.235
HOG descriptor: Avg. Theta Error for Image 1: 17.136
HOG descriptor: Avg. Scale Error for Image 2: 0.740
HOG descriptor: Avg. Theta Error for Image 2: 18.379
HOG descriptor: Avg. Scale Error for Image 3: 0.176
HOG descriptor: Avg. Theta Error for Image 3: 24.384
```

Figure 37: Αποτελέσματα για Box filter μέθοδο (πολλαπλής κλίμακας) εκτίμησης σημείων ενδιαφέροντος. Απόκλιση κλίμακας και γωνίας (ϑ) περιστροφής των 3 εικόνων

3.1.2 Σχολιασμός της ικανότητας εκτίμησης

Παρατηρούμε ότι ανεξαρτήτως ανιχνευτή - παρατηρητή η προσέγγιση της κλίμακας είναι καλύτερη από αυτή της περιστροφής. Πιο συγκεκριμένα παρατηρούμε ότι με **περιγραφητή HOG** λαμβάνουμε **καλύτερες εκτιμήσεις** όσον αφορά την κλίμακα και την γωνία σε σχέση με τις εκτιμήσεις στις ίδιες μετατροπές με περιγραφητή SURF για τον ανιχνευτή με τα **Box filters**. Ενώ γενικότερα για τον προσδιορισμό της κλίμακας οι δύο περιγραφητές παρουσιάζουν παρόμοια αποτελέσματα με αποκλίσεις που δεν παρουσιάζουν μοτίβο και δεν επιτρέπουν την εξαγωγή ασφαλούς συμπεράσματος. Ενδιαφέρον είναι το γεγονός πως παρόλο που γενικά ο local περιγραφητής SURF φαίνεται να εκτιμά καλύτερα τη γωνία ϑ, στα Box Filters ο HOG υπερτερεί σημαντικά. Συμπεραίνουμε δηλαδή ότι ο local descriptor HOG υλοποιεί πιο αποδοτικά το ταίριασμα των εικόνων σε σχέση με τον SURF. Όσο αφορά τους ανιχνευτές, παρατηρούμε ότι οι μονοκλιμακωτές μέθοδοι δίνουν χειρότερες εκτιμήσεις σε σχέση με τις αντίστοιχες πολυκλιμακωτές, πράγμα λογικό και αναμενόμενο, αφού στην πολυκλιμακωτή μέθοδο έχουμε μεγαλύτερη ακρίβεια στην ανίχνευση των σημείων ενδιαφέροντος. Τέλος, παρατηρούμε ότι τον καλύτερο δυνατό συνδυασμό εκτίμησης κλίμακας και περιστροφής των εικόνων τον πετυχαίνουμε χρησιμοποιώντας τον αντίστοιχο συνδυασμό πολυκλιμακωτής μεθόδου ανίχνευσης μέσω Box Filters με περιγραφητή HOG.

3.2 Κατηγοριοποίηση Εικόνων

Στόχος του ερωτήματος αυτού είναι η χρήση περιγραφητών και ανιχνευτών για την κατηγοριοποίηση ενός συνόλου εικόνων από τη βάση Pascal VOC2005 στις κλάσεις : αυτοκίνητο, άνθρωπος, ποδήλατο. Χρησιμοποιούμε μόνο τις πολυκλιμακωτές μεθόδους μιας και όπως διαπιστώθηκε στο προηγούμενο ερώτημα οι μονοκλιμακωτές μέθοδοι δίνουν αρκετά μεγάλες αποκλίσεις στα αποτελέσματά τους.

3.2.1 Εξαγωγή χαρακτηριστικών

Αρχικά, χρησιμοποιούμε τη συνάρτηση FeatureExtraction που μας δίνεται, με κάθε συνδυασμό περιγραφητή-ανιχνευτή και εξάγουμε από κάθε εικόνα της βάσης τα χαρακτηριστικά βάσει των οποίων θα συνεχιστεί η κατηγοριοποίησή τους.

3.2.2 Split dataset and label encode

Έπειτα, χρησιμοποιούμε τη συνάρτηση createTrainTest που πάλι μας δίνεται, με όρισμα τα χαρακτηριστικά αυτά που εξάγαμε από τη FeutureExtraction και διαχωρίζουμε τις εικόνες μας στις κατηγορίες-κλάσεις train και test, δίνοντας τους την αντίστοιχη ετικέτα-label.
Τώρα για κάθε εικόνα έχουμε ένα σύνολο από descriptos και σημείων ενδιαφέροντος χωρίς να υπάρχει κάποιος ενιαίος συγκριτής εικόνων. Για το πρόβλημα της κατηγοριοποίησης-αναγνώρισης θέλουμε κάθε εικόνα να έχει ενιαία και μοναδική αναπαράσταση σε ένα global διάνυσμα χαρακτηριστικών με την ίδια προφανώς διάσταση για κάθε εικόνα. Για το σκοπό αυτό, προσπαθούμε να περιγράψουμε όλη την πληροφορία που έχουν οι περιγραφητές μέσα στην εικόνα σαν ένα ιστόγραμμα πιθανών οπτικών περιγραφών.

3.2.3 Αναπαράσταση Bag of Visual Words

Τα τοπικά χαρακτηριστικά των εικόνων είναι πλέον διακριτά και βάσει αυτών φτιάχνουμε ένα οπτικό λεξικό το οποίο ονομάζουμε Bag of Visual Words (BOV). Πιο συγκεκριμένα για τον προσδιορισμό των λέξεων του λεξικού ενώνουμε όλους τους περιγραφητές του train συνόλου σε ένα διάνυσμα χαρακτηριστικών και στη συνέχεια χρησιμοποιούμε αλγόριθμο συσταδοποίησης kmeans για ένα τυχαίο υποσύνολο του παραπάνω διανύσματος προσαρμόζοντας την υπερπαράμετρο αριθμού των κέντρων (από 500-2000 όπως προτείνεται). Για την επιλογή του αριθμού των κέντρων λάβαμε υπόψιν ότι όσο μικρότερος ο αριθμός τους τόσο πιο πιθανό είναι να μην υπάρχουν αρκετά κέντρα για να περιγράψουν όλα τα variatons των εικόνων που στην τελική θα είναι αυτά που θα τις διαχωρίσουν στις ζητούμενες κλάσεις, ενώ όσο μεγαλύτερο το πλήθος των κέντρων τόσο πιο πιθανό να αυξηθεί η πολυπλοκότητα-το complexity του τελικού ιστογράμματος πράγμα που θα συντελέσει στο να μην πάρουμε ξεκάθαρες λέξεις στο τελικό μας λεξικό. Έπειτα, για κάθε εικόνα υπολογίζουμε την ελάχιστη ευκλείδια απόσταση του κάθε τοπικού περιγραφητή από τα κέντρα που υπολογίσαμε και με χρήση της συνάρτησης histc (όπως προτείνεται) κατασκευάζουμε το ιστόγραμμα κάθε εικόνας, το οποίο τέλος κανονικοποιούμε (voting) με βάση την L2 νόρμα.

3.2.4 Κατηγοριοποίηση εικόνων

Για την κατηγοριοποίηση τελικά των εικόνων με βάση την BovW αναπαράσταση χρησιμοποιείται ένας SVM Support Vector Machine ταξινομητής μέσω της συνάρτησης svm. Τα αποτελέσματα που

λαμβάνουμε αναφορικά με το αποτέλεσμα της αναγνώρισης και το συνολικό ποσοστό της επιτυχίας για τους διάφορους ανιχνευτές/περιγραφητές παρατίθενται παρακάτω:

Table 1: Local descriptor Surf (Speed Up Robust Features)

algorithm	feature extraction time	accuracy
Harris	245.140	0.5448275862068965
Blobs	223.129	0.5655172413793104
Box-filters	198.797	0.5793103448275863

Table 2: Local descriptor HOG (High Obviously Grindered)

algorithm	feature extraction time	accuracy
Harris	317.382	0.6482758620689655
Blobs	-	-
Box-filters	277.126	0.6482758620689655

Αρχικά, παρατηρούμε ότι χρησιμοποιώντας τον περιγραφητή HOG για την κατηγοριοποίηση των εικόνων έχουμε μεγαλύτερο ποσοστό επιτυχίας σε σχέση με τον SURF.
Πιο συγκεκριμένα, βέλτιστο ποσοστό επιτυχίας επιτυγχάνεται με συνδυασμό περιγραφητή HOG και ανιχνευτή την πολυκλιμακωτή μέθοδο BoxFilters, ενώ λίγο χειρότερος αναμενόταν να είναι ο συνδυασμός HOG με πολυκλιμακωτή μέθοδο ανίχνευσης blobs.
Κάτι τέτοιο είναι αναμενόμενο με βάση το γεγονός πως πρόκειται για δύο παρεμφερείς μεθόδους ανίχνευσης των σημείων ενδιαφέροντος που υλοποιούνται απλώς με ορισμένες μικρές παραλλαγές.
Η πολυκλιμακωτή μέθοδος ανίχνευσης γωνιών σε συνδυασμό με περιγραφητή HOG εξίσου ικανοποιητική απόδοση, ωστόσο χρειάζεται το μεγαλύτερο χρόνο για την εξαγωγή χαρακτηριστικών. [1]. Η ακρίβεια που λαμβάνουμε είναι περίπου στο 65% τόσο για την απλή Hessian μέθοδο όσο και για τη μέθοδο των Box Filters. Παρόμοια αποτελέσματα για την αποδοτικότητα των ανιχνευτών βγάζουμε παρατηρώντας τα αποτελέσματα της κατηγοριοποίησης με τον περιγραφητή SURF.
Έτσι λοιπόν όταν χρησιμοποιείται ο περιγραφητής SURF η καλύτερη μέθοδος ανίχνευσης είναι η πολυκλιμακωτή ανίχνευση BoxFilters, αμέσως μετά η πολυκλιμακωτή μέθοδος ανίχνευσης blobs και η χειρότερη μέθοδος η πολυκλιμακτώ μέθοδος ανίχνευσης γωνιών με ποσοστό επιτυχίας περίπου 54.5%.

Οπότε συνολικά, θα προτείναμε περιγραφητή HOG με πολυκλιμακωτή ανίχνευση με Box Filters.

[1]Σημειώνουμε πως για τον τοπικό περιγραφητή HOG και τον ανιχνευτή Blobs (Hessian-Laplace) πολλαπλής κλίμακας δεν ήταν δυνατή η εξαγωγή αποτελεσμάτων αφού λαμβάναμε error κατά την εκτέλεση του κώδικα *"ValueError: Shape of array too small to calculate a numerical gradient, at least (edge_ order + 1) elements are required."* γεγονός που πιθανότατα οφείλεται στα σημεία ενδιαφέροντος κοντά στα περιθώρια των εικόνων τα οποία βέβαια έχουμε αφαιρέσει - το ιδιο προβλημα παρουσιαζόταν αρχικά και για τους υπόλοιπους ανιχνευτές. Το πρόβλημα αυτό σχετίζεται με την εσωτερική υλοποίηση της συνάρτησης HOG η οποία δεν μας είναι γνωστή και δεν μπορούμε να τροποποιήσουμε

Επίσης, σημαντικό να αναφερθεί το γεγονός πως η εκτίμηση σημείων ενδιαφέροντος αλλά και τελικά η κατηγοριοποίηση των εικόνων με βάση τα χαρακτηριστικά από τις μεθόδους και με τεχνικές deep learning (Support Vector Machines) είναι **σημαντικά πιο αργή** για τον **ανιχνευτή Blobs** (Hessian+Laplacian) προς επίρρωσιν την επιλογής της μεθόδου των Box filters.