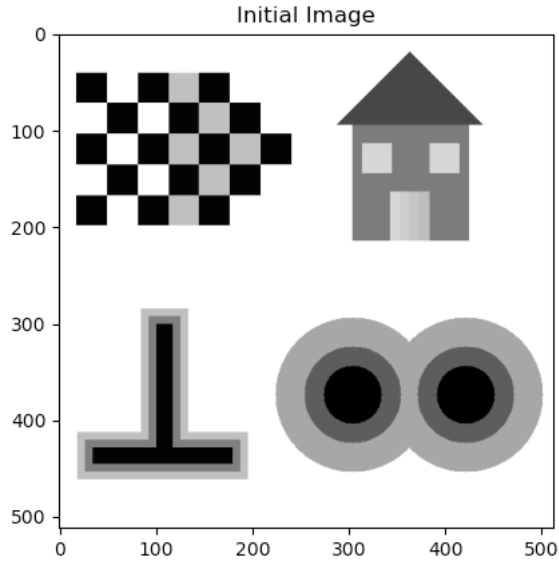


1 Ανίχνευση Ακμών σε Γκριζες Εικόνες

1.1 Δημιουργία Εικόνων Εισόδου

Η αρχική εικόνα "edgetest_10.png" που διαβάστηκε φαίνεται στο παρακάτω σχήμα:



Σχήμα 1: Αρχική εικόνα

Δημιουργήθηκαν δύο θορυβώδεις εικόνες με:

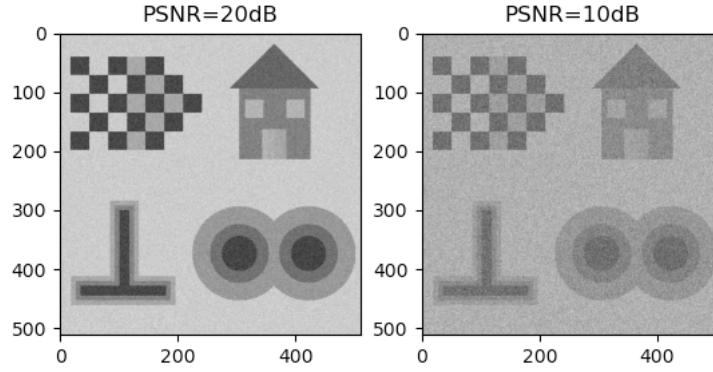
- i) $\text{PSNR} = 20\text{dB} \Rightarrow$

$$20 \log_{10} \left(\frac{I_{\max} - I_{\min}}{\sigma_n} \right) \text{dB} = 10\text{dB} \Rightarrow \sigma_n = \frac{I_{\max} - I_{\min}}{10}$$

- ii) $\text{PSNR} = 10\text{dB} \Rightarrow$

$$20 \log_{10} \left(\frac{I_{\max} - I_{\min}}{\sigma_n} \right) \text{dB} = 20\text{dB} \Rightarrow \sigma_n = \frac{I_{\max} - I_{\min}}{\sqrt{10}}$$

οι οποίες φαίνονται παρακάτω:



Σχήμα 2: Θορυβώδεις εικόνες

1.2 Υλοποίηση Αλγορίθμων Ανίχνευσης Ακμών

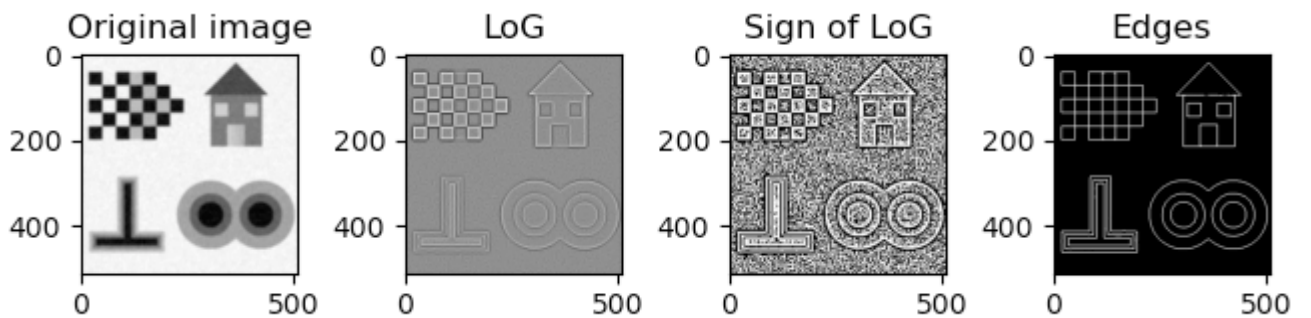
Για την υλοποίηση του φίλτρου που προσεγγίζει τον τελεστή Laplacian-of-Gaussian (LoG) $h(x, y) = \nabla^2 G_\sigma(x, y)$ δημιουργήθηκε διδιάστατο array μεγέθους $(n \times n)$ με $n = 2 \times \lceil 3 \sigma \rceil + 1$, πάνω στο οποίο εφαρμόστηκε η συνάρτηση:

$$\nabla^2 G_\sigma(x, y) = \frac{x^2 + y^2 - 2\sigma^2}{2\pi \sigma^6} \exp \left[\frac{-(x^2 + y^2)}{2\sigma^2} \right]$$

από την οποία προέκυψε η προσέγγιση της κρουστικής $h(x, y)$.

Η προσέγγιση της Laplacian γίνεται είτε με γραμμικό, είτε με μη-γραμμικό τρόπο και εντοπίζονται τα σημεία μηδενισμού της (zero-crossings), στα οποία η κλίση της εικόνας είναι μεγαλύτερη ενός κατωφλίου.

Παρακάτω φαίνονται τα στάδια του αλγορίθμου με είσοδο μια θορυβώδη εικόνα:

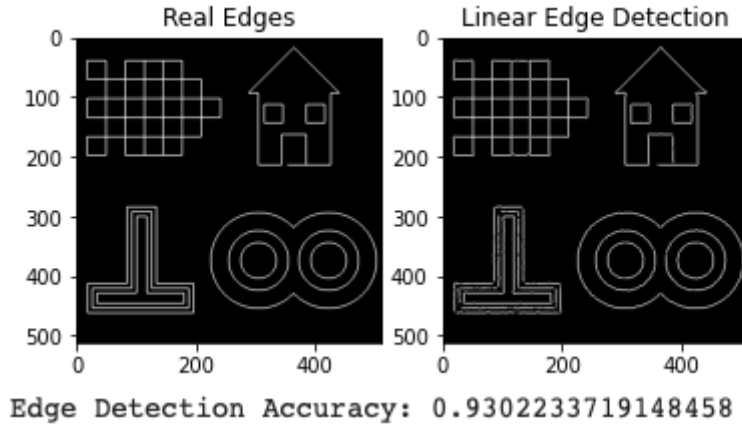


Σχήμα 3: Αλγόριθμος ανίχνευσης ακμών

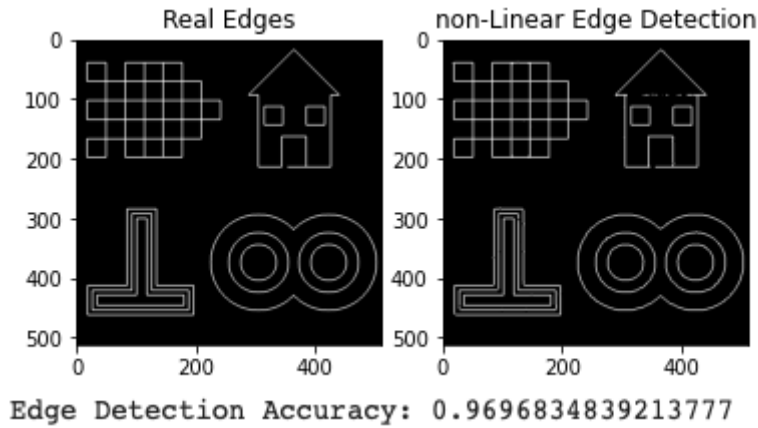
1.3 Αξιολόγηση των Αποτελεσμάτων Ανίχνευσης Ακμών

Για την ποσοτική αξιολόγηση του παραπάνω αλγορίθμου, υπολογίσαμε τις “αληθινές” ακμές με χρήση της εικόνας χωρίς θόρυβο και χρησιμοποιήσαμε το κριτήριο ποιότητας (C) του M.O. των Precision και Recall. Ως θ_{realedge} , εκτιμήσαμε την τιμή 0.2 .

Παρακάτω φαίνονται οι ποσοτικά βέλτιστες προσεγγίσεις για τις δύο θορυβώδεις εικόνες, με το βέλτιστο ζεύγος των παραμέτρων σε κάθε περίπτωση:



Σχήμα 4: Γραμμική ανίχνευση με $\text{PSNR} = 20\text{dB}$, $\sigma = 2.1$, $\theta_{\text{edge}} = 0.23$

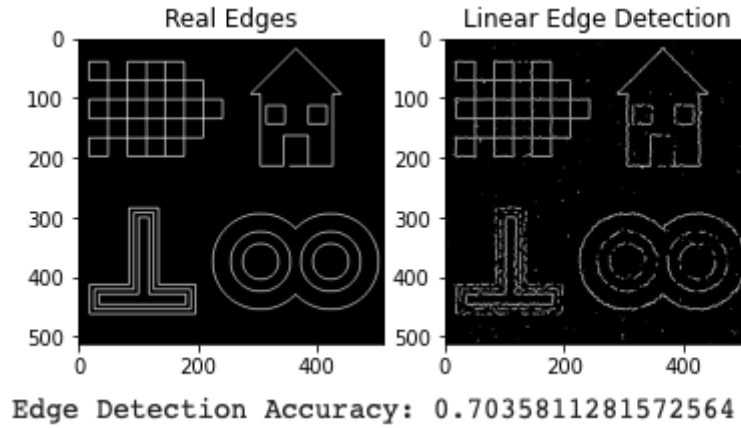


Σχήμα 5: Μη-γραμμική ανίχνευση με $\text{PSNR} = 20\text{dB}$, $\sigma = 1.6$, $\theta_{\text{edge}} = 0.19$

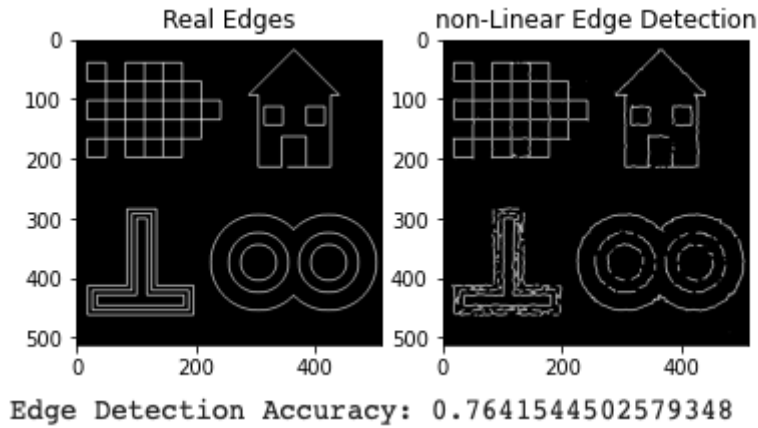
Παρατηρούμε, για την εικόνα με $\text{PSNR} = 20\text{dB}$, ότι πετυχαίνουμε πολύ καλή ακρίβεια, σύμφωνα με το κριτήριο ποιότητας C , και για τις δύο προσεγγίσεις της Laplacian (γραμμική: 93.0%, μη-γραμμική 96.9%).

Σε ό,τι αφορά τις παραμέτρους, μειώνοντας πολύ την τυπική απόκλιση σ ο αλγόριθμος αδυνατεί να αναγνωρίσει πολλές ακμές και ταυτόχρονα υπάρχει αρκετός θόρυβος, ενώ αυξάνοντάς τη πολύ, παρατηρούμε μια διαστρέυλωση των σχημάτων, μετατρέποντας τις κορυφές σε καμπύλες.

Για το κριτήριο κατωφλίου θ_{edge} , επιλέγοντας πολύ μικρές τιμές του (< 0.03), ο αλγόριθμος θεωρεί ως ακμές και τις πολύ ομαλές μεταβολές, οπότε στην τελική εικόνα υπάρχει πολύς θόρυβος και έχουμε πολύ κακή ποιότητα. Για μεγάλες τιμές του θ_{edge} (> 0.5), παρατηρούμε ότι δεν ανιχνεύονται καθόλου οι ακμές των δύο κάτω σχημάτων, ως λιγότερο απότομες μεταβολές, ενώ το πάνω αριστερά σχήμα διατηρεί εξαιρετική ποιότητα μέχρι πολύ μεγάλες τιμές (≈ 0.6).



Σχήμα 6: Γραμμική ανίχνευση με $\text{PSNR} = 10\text{dB}$, $\sigma = 2.0$, $\theta_{\text{edge}} = 0.24$



Σχήμα 7: Μη-γραμμική ανίχνευση με $\text{PSNR} = 10\text{dB}$, $\sigma = 2.7$, $\theta_{\text{edge}} = 0.23$

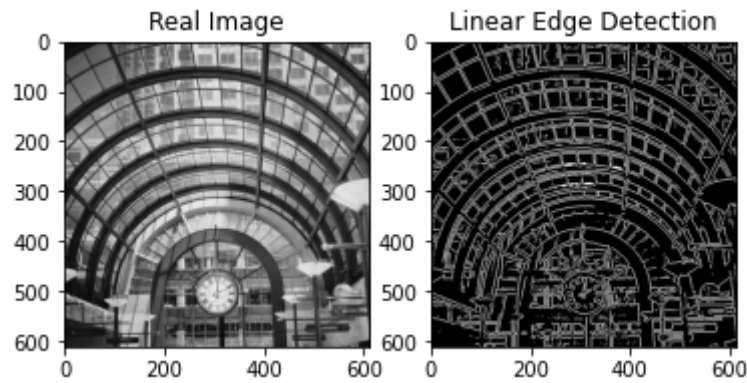
Για την εικόνα με $\text{PSNR} = 10\text{dB}$, πετυχαίνουμε σχετικά καλή ακρίβεια με τη γραμμική προσέγγιση (70.3%) και τη μη-γραμμική: (76.4%). Είναι εμφανώς μειωμένη και στις δύο περιπτώσεις σε σχέση με την προηγούμενη εικόνα, γεγονός που είναι λογικό λόγω της αύξησης του αρχικού θορύβου. Παρόμοια φαινόμενα με προηγούμενως παρατηρούμε με τη μεταβολή των παραμέτρων σ και θ_{edge} .

Γενικά, με τη μη-γραμμική προσέγγιση της Laplacian έχουμε και ποσοτικά και ποιοτικά καλύτερα αποτελέσματα σε σύγκριση με τη γραμμική.

1.4 Εφαρμογή των Αλγορίθμων Ανίχνευσης Ακμών σε Πραγματικές εικόνες

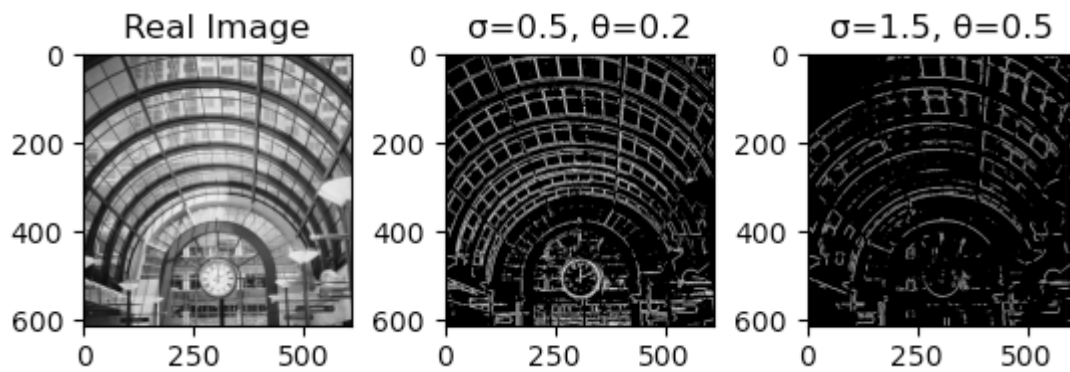
Στη συνέχεια, φορτώθηκε μια νέα καθαρή εικόνα στην οποία εφαρμόσαμε τον αλγόριθμο ανίχνευσης με ποικιλία τιμών του ζεύγους παραμέτρων, με σκοπό να εκτιμήσουμε ποιοτικά τη βέλτιστη προσέγγισή της.

Παρακάτω παραθέτουμε την εκτίμηση με γραμμική προσέγγιση:



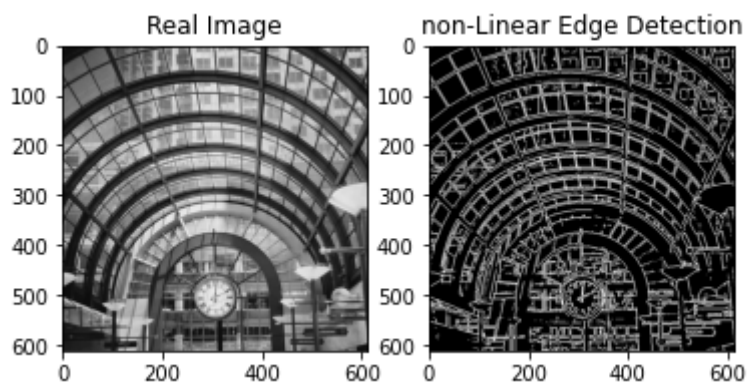
Σχήμα 8: Γραμμική ανίχνευση χωρίς θόρυβο με $\sigma = 1.5$, $\theta_{\text{edge}} = 0.2$

Παραθέτουμε, επίσης, τα αποτελέσματα με διαφορετικές τιμές του ζεύγους παραμέτρων:

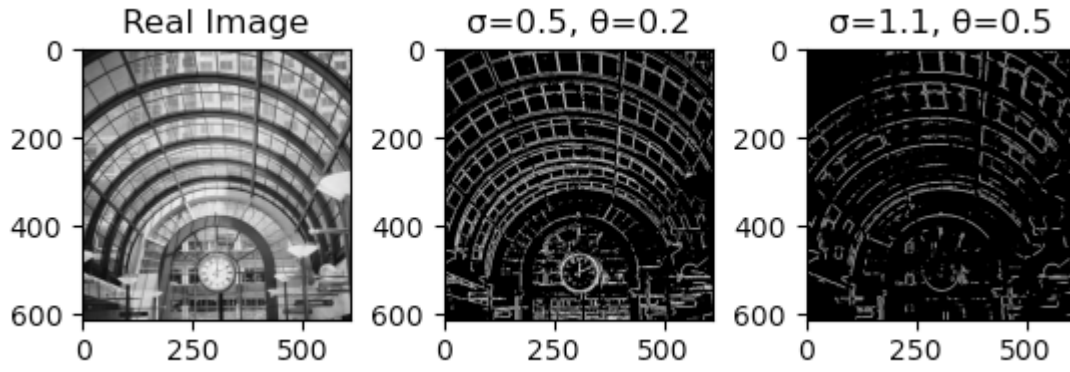


Σχήμα 9: Γραμμική ανίχνευση χωρίς θόρυβο με διαφορετικές τιμές σ , θ_{edge}

Επαναλαμβάνουμε και με μη-γραμμική προσέγγιση:



Σχήμα 10: Μη-γραμμική ανίχνευση χωρίς θόρυβο με $\sigma = 1.1$, $\theta_{\text{edge}} = 0.2$



Σχήμα 11: Μη-γραμμική ανίχνευση χωρίς θόρυβο με διαφορετικές τιμές $\sigma, \theta_{\text{edge}}$

Παρατηρούμε και πάλι και στις δύο προσεγγίσεις, πως το σ επηρεάζει τη λεπτομέρεια των ακμών που ανιχνεύονται και με αύξηση/μείωση του θ_{edge} , ανιχνεύονται οι πιο απότομες/ομαλές ακμές αντίστοιχα.

2 Ανίχνευση Σημείων Ενδιαφέροντος

Στο μέρος αυτό κληθήκαμε να υλοποιήσουμε συναρτήσεις ανίχνευσης γωνιών.

2.1 Ανίχνευση Γωνιών

Ξεκινάμε με την μέθοδο Harris-Laplacian, υπολογίζοντας τον τανυστή J και συγκεκριμένα τα στοιχεία J_1, J_2, J_3 :

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

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

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

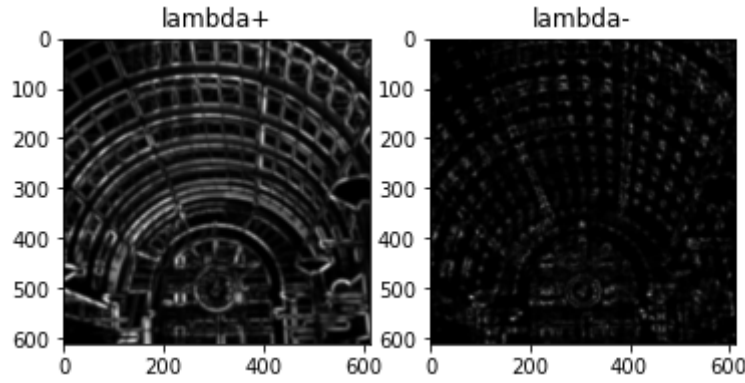
όπου η παραγωγή έγινε με την συνάρτηση **gradient** της **numpy** και ο γκαουσιανός πυρήνας με την **getGaussianKernel**. Το επόμενο βήμα είναι ο υπολογισμός των ιδιοτιμών του τανυστή J μέσω της σχέσης:

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

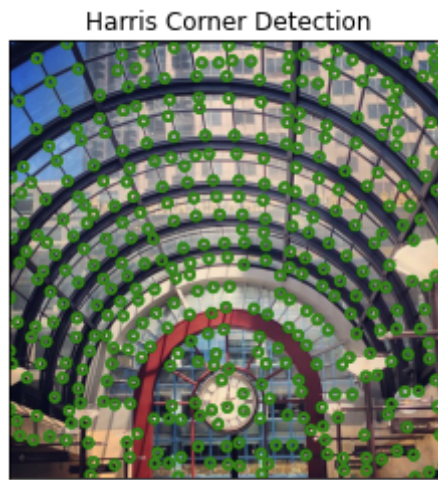
Έπειτα για κάθε σημείο του πίνακα εξάγουμε το ακόλουθο κριτήριο γωνιότητας:

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

Παρακάτω φαίνονται οι εικόνες των ιδιοτιμών καθώς και το αποτέλεσμα του αλγορίθμου πάνω στην εικόνα του προηγούμενου ερωτήματος:



Σχήμα 12: Ιδιοτιμές του τανυστή J



Σχήμα 13: Ανίχνευση γωνιών με τη μέθοδο Harris-Stephens

Σχεδιάζοντας τις γκριζες εικόνες των λ_+ και λ_- παρατηρούμε ότι στην πρώτη αποτυπώνονται με μέτρια ευκρίνεια ακμές της κανονικής εικόνας ενώ στην δεύτερη περιέχεται λιγότερη πληροφορία και αυτή αφορά περισσότερο σημεία παρά ακμές. Από τον αλγόριθμο, επιλέγονται ως γωνίες τα σημεία του κριτηρίου R που έχουν τιμή μεγαλύτερη από ένα επιλεγμένο κατώφλι ($\theta_{corn} \cdot R_{max}$) αλλά και είναι τοπικά μέγιστα για μια μικρή κυκλική περιοχή γύρω από τα x, y . Επί της ουσίας, όσο μεγαλύτερη η τιμή της R τόσο μεγαλύτερη η διαφορά φωτεινότητας του σημείου της αρχικής εικόνας σε σχέση με τα γειτονικά του και ως προς τις δύο κατευθύνσεις (x, y) κάτι που μας οδηγεί στο συμπέρασμα ότι εκεί βρίσκεται γωνία.

2.2 Πολυκλιμακωτή Ανίχνευση Γωνιών

Η παραπάνω ανίχνευση αφορούσε μία μόνο κλίμακα για τις παραμέτρους σ και ρ , που είναι οι κλίμακες διαφόρισης και ολοκλήρωσης αντίστοιχα. Σε αυτό το ερώτημα υλοποιήσαμε την μέθοδο Harris-Laplacian η οποία αποτελείται από δύο βήματα.

Στο πρώτο βήμα υλοποιήσαμε την διαδικασία του προηγούμενου ερωτήματος αλλά για

3 κλίμακες και αποθηκεύσαμε τις συντεταγμένες των στοιχείων σε ένα array με διαστάσεις $N \times 3$ όπου N το πλήθος των σημείων που ανιχνεύθηκαν και η άλλη διάσταση θα είναι (x, y, σ) , δηλαδή οι συντεταγμένες του σημείου και η κλίμακα για την οποία το ανιχνεύσαμε.

Στο δεύτερο βήμα, για κάθε διαφορετική κλίμακα υπολογίζουμε για το σύνολο των σημείων της εικόνας την κανονικοποιημένη Laplacian of Gaussian με βάση την εξίσωση:

$$|\text{LoG}(x, \sigma_i)| = \sigma_i^2 |L_{xx}(x, \sigma_i) + L_{yy}(x, \sigma_i)|$$

Τέλος, κρατάμε τα σημεία τα οποία στην κλίμακα που ανιχνεύθηκαν έχουν μεγαλύτερη τιμή LoG απ' ότι μία κλίμακα παρακάτω και παραπάνω. Αυτό που παρατηρούμε είναι ότι εν τέλει κρατούνται λιγότερα σημεία σε σχέση με την ανίχνευση σε μία κλίμακα, όπως φαίνεται παρακάτω:



Σχήμα 14: Πολυκλιμακωτή ανίχνευση γωνιών με τη μέθοδο Harris-Stephens

2.3 Ανίχνευση Blobs

Στη συνέχεια υλοποιήσαμε τη μέθοδο Harris για ανίχνευση blobs. Στο κομμάτι αυτό χρησιμοποιήσαμε την ορίζουσα του πίνακα Hessian:

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

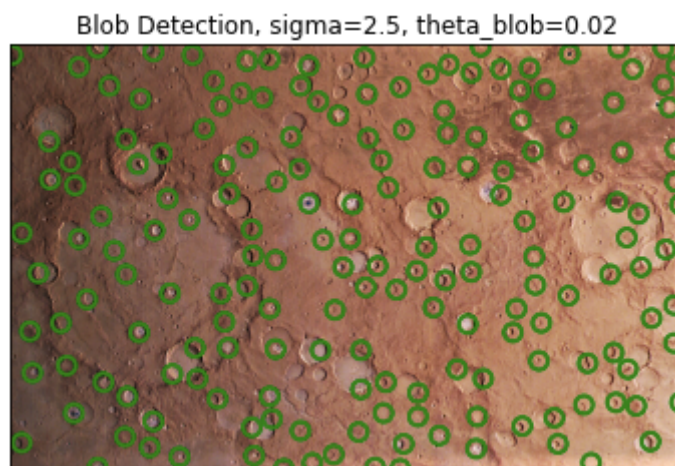
Όπου L_{xx}, L_{yy} και L_{xy} αποτελούν μερικές παραγώγους της κανονικοποιημένης, με γκαουσιανό φίλτρο, αρχικής εικόνας. Τις μερικές παραγώγους τις υπολογίζουμε ξανά με την συνάρτηση **gradient**.

Όπως και στο ερώτημα 2.1 για την ανίχνευση γωνιών, τα κριτήρια για να επιλεγεί ένα σημείο ως σημείο ενδιαφέροντος είναι η ορίζουσα Hessian αυτού του σημείου να είναι μέγιστη σε μία γειτονία και η τιμή της να είναι μεγαλύτερη από ένα προεπιλεγμένο κατώφλι.

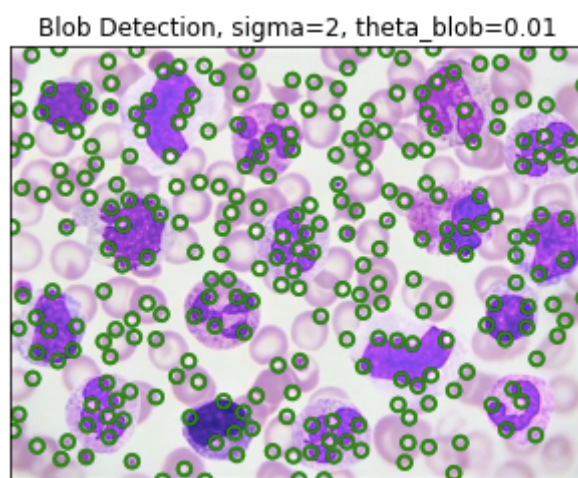
Στην ουσία οι παράγωγοι δεύτερης τάξης μετρούν την καμπυλότητα στο εκάστοτε σημείο και επομένως οι ιδιοτιμές του πίνακα δίνουν το μέγεθος αυτής της καμπυλότητας.

Άρα, όταν ικανοποιούνται τα παραπάνω κριτήρια, αυτό συνεπάγεται μεγάλη μεταβολή της καμπυλότητάς του και άρα σημείο blob.

Παρακάτω φαίνονται τα αποτελέσματα οπτικοποιημένα πάνω στις εικόνες mars και blood_smear:



Σχήμα 15: Ανίχνευση Blob στο mars με τη μέθοδο Harris



Σχήμα 16: Ανίχνευση Blob στο blood_smear με τη μέθοδο Harris

Παρατηρούμε πολύ καλή ανίχνευση και στις δύο εικόνες, αλλά μόνο σε blobs που έχουν μέγεθος της τάξης της κλίμακας σ . Για το λόγο αυτό, στο επόμενο ερώτημα εφαρμόζουμε τη μέθοδο σε πολλαπλές κλίμακες.

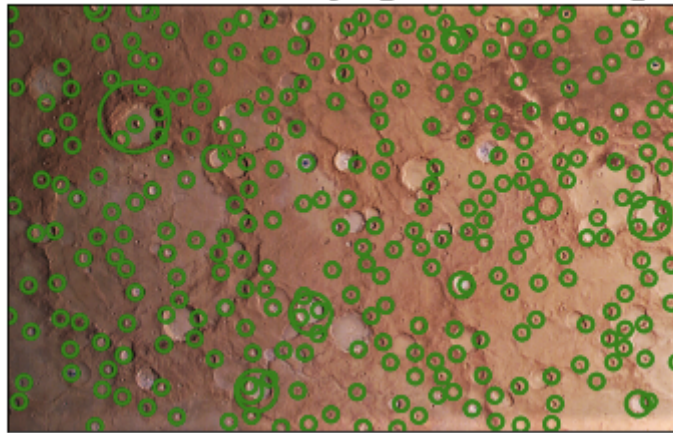
2.4 Πολυκλιμακωτή Ανίχνευση Blobs

Σε αντιστοιχία με το ερώτημα 2.2, για κάθε σημείο ενδιαφέροντος που βρίσκουμε για κάθε κλίμακα, αποθηκεύουμε τις συντεταγμένες του και την κλίμακα στην οποία ανιχνεύθηκε (x, y, σ) και υπολογίζουμε την ορίζουσα Hessian για κάθε σημείο της εικόνας

σε κάθε κλίμακα.

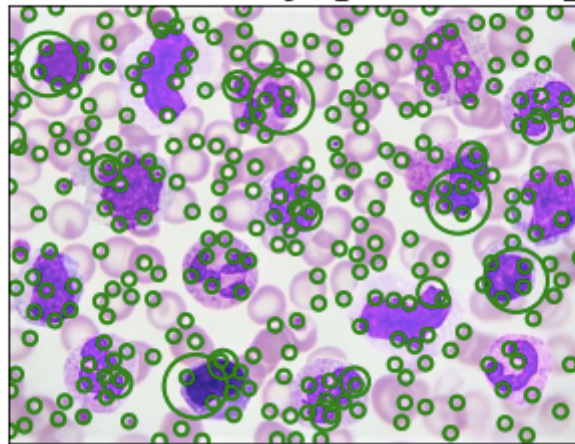
Έπειτα, συγκρίνουμε τις τιμές των σημείων στην κλίμακα που τα έχουμε εντοπίσει, με την τιμή τους στην επόμενη και την προηγούμενη κλίμακα, και αν η τιμή στην κλίμακα που τα εντοπίσαμε είναι μεγαλύτερη απ' ότι στις δύο άλλες, το θεωρούμε σημείο ενδιαφέροντος.

Multiscale Blob Detection, N=4, sigma_0=2.5, s=1.7, theta_blob=0.02



Σχήμα 17: Πολυκλιμακωτή ανίχνευση Blob στο mars με τη μέθοδο Harris

Multiscale Blob Detection, N=3, sigma_0=2, s=2.2, theta_blob=0.005

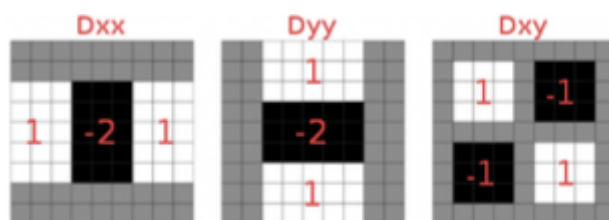


Σχήμα 18: Πολυκλιμακωτή ανίχνευση Blob στο blood_smear με τη μέθοδο Harris

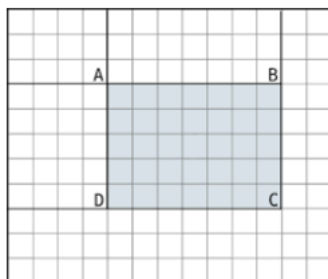
Αυτό που παρατηρούμε είναι ότι, σε αντίθεση με την ανίχνευση σε μία κλίμακα, μπορούμε να ανιχνεύσουμε blobs διαφορετικών μεγεθών, το οποίο πετυχαίνεται ικανοποιητικά στις παραπάνω εικόνες με τις παραμέτρους που φαίνονται.

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

Η παραπάνω διαδικασία απαιτεί συνέλιξη της εικόνας με φίλτρα των οποίων το μέγεθος αυξάνεται με την κλίμακα, γεγονός που την καθιστά υπολογιστικά πολύπλοκη και άρα αργή. Για το λόγο αυτό, χρησιμοποιούμε μια μέθοδο από Speeded Up Robust Features (SURF), η οποία προσεγγίζει τα φίλτρα δεύτερης παραγωγής με φίλτρα όπως τα παρακάτω:



Τα παραπάνω φίλτρα βασίζονται σε αθροίσματα ορθογωνίων περιοχών ίδιων διαστάσεων στο κάθε ένα. Έτσι, εκμεταλλευόμαστε τις Ολοκληρωτικές εικόνες (Integral images), που αποτελούνται από το άθροισμα του κάθε pixel με όλα τα πάνω και αριστερά του pixels. Αυτή υλοποιείται στο πρώτο βήμα για την εικόνα εισόδου με χρήση της **cumsum** πρώτα στον ένα άξονα και έπειτα στον άλλο πάνω στο αποτέλεσμα της πρώτης.



Στη συνέχεια, για να υπολογίσουμε το άθροισμα του κάθε παραθύρου του φίλτρου, αρκεί να εφαρμόσουμε **shifting** με τη συνάρτηση **roll** της **numpy** έτσι ώστε να δημιουργήσουμε τέσσερα νέα arrays τα στοιχεία των οποίων θα είναι τα A, B, C, D του αντίστοιχου, στην αρχική ολοκληρωτική εικόνα, pixel και να υπολογίσουμε το $A + C - B - D$.

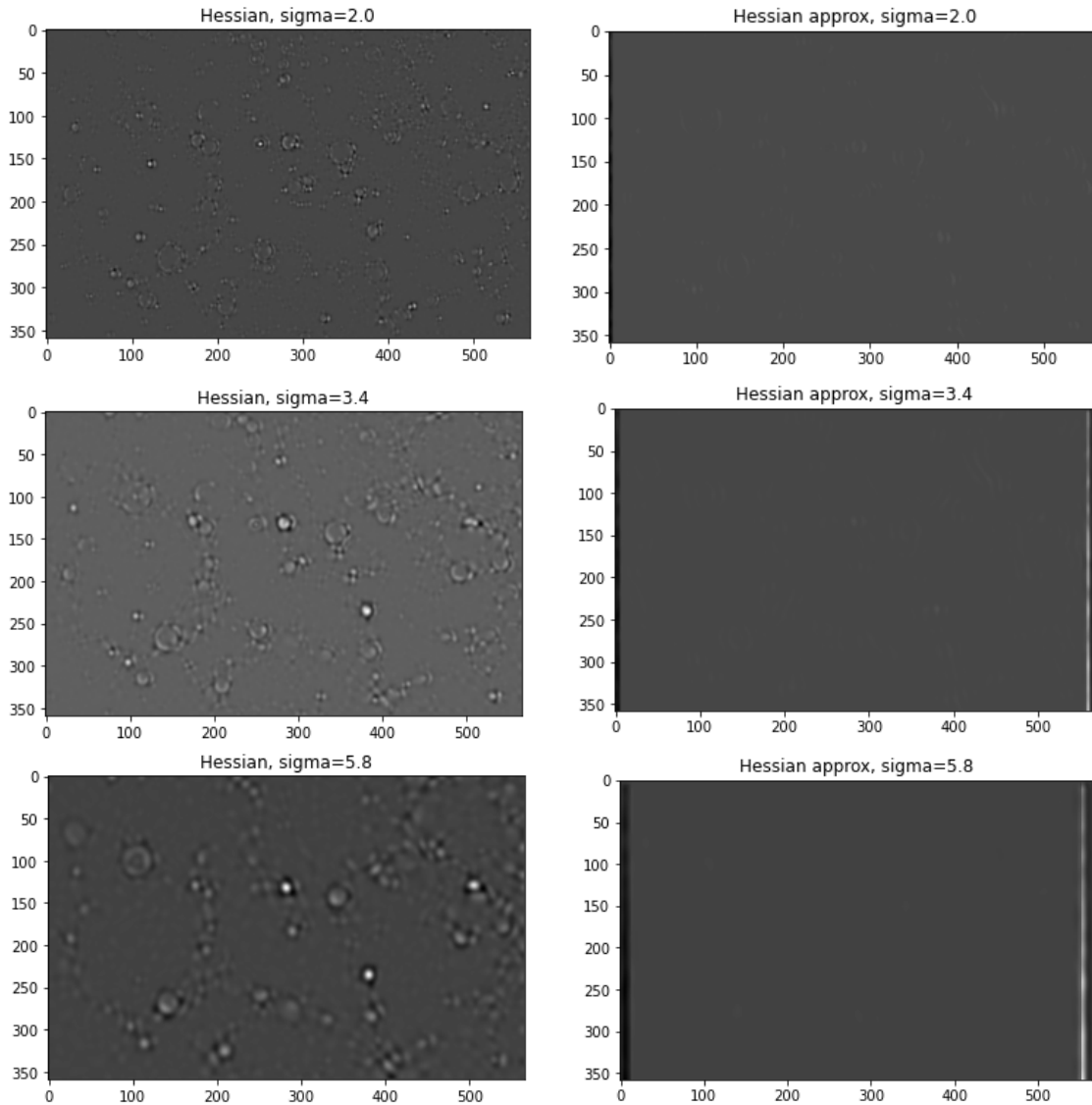
Προτού γίνει αυτό, πρέπει να προηγηθεί ένα **padding** στις γραμμές κατά το μισό του ύψους του παραθύρου στρογγυλοποιημένο προς τα κάτω (i_c) + 1 και στις στήλες κατά το μισό του πλάτους (j_c) + 1. Οι τιμές του **padding** θα είναι μηδενικά προς τα πάνω και προς τα αριστερά, ενώ προς τα κάτω και προς τα δεξιά θα είναι οι τιμές των ακραίων pixels, για να μην υπάρξει πρόβλημα υπολογισμού στις ακραίες περιπτώσεις.

Συγκεκριμένα, με **shift** κατά $i_c + 1$ (κάτω) και $j_c + 1$ (δεξιά) δημιουργούμε ένα array που περιέχει για κάθε pixel το A του παραπάνω σχήματος, που του αντιστοιχεί. Με **shifting** κατά $i_c + 1$ (κάτω) και $-j_c$ (αριστερά) δημιουργούμε το array για το B, με $-i_c$ (πάνω) και $j_c + 1$ (δεξιά) για το D και με $-i_c$ (πάνω) και $-j_c$ (αριστερά) για το C.

Έτσι, προσθέτοντάς τα και αφαιρώντας το padding έχουμε εφαρμόσει ένα μοναδιαίο παράθυρο (με τιμές άσσο) στην εικόνα και επομένως με shifting της εικόνας που προκύπτει, μπορούμε να δημιουργήσουμε arrays στα οποία έχει εφαρμοστεί οποιοδήποτε παράθυρο του φίλτρου με οποιοδήποτε συντελεστή. Συγκεκριμένα, για το D_{xx} αρκεί shift κατά το πλάτος του παραθύρου δεξιά και αριστερά, για το D_{yy} κατά το ύψος πάνω και κάτω και άλλα 4 shifts για το D_{xy} με την ίδια λογική. Προσθέτοντας τις εικόνες για το κάθε φίλτρο, προκύπτει η προσέγγιση της δεύτερης παραγώγου. Ως κριτήριο γωνιότητας θα εφαρμόσουμε και πάλι τον πίνακα Hessian, αλλά αυτή τη φορά με τις προσεγγίσεις των παραγώγων του:

$$R(x, y) = L_{xx}(x, y) L_{yy}(x, y) - (0.9L_{xy}(x, y))^2$$

Παρακάτω φαίνεται η σύγκριση του κριτηρίου R όπως αυτό προέκυψε με τον Hessian πίνακα και με την προσέγγιση του για διάφορες κλίμακες:

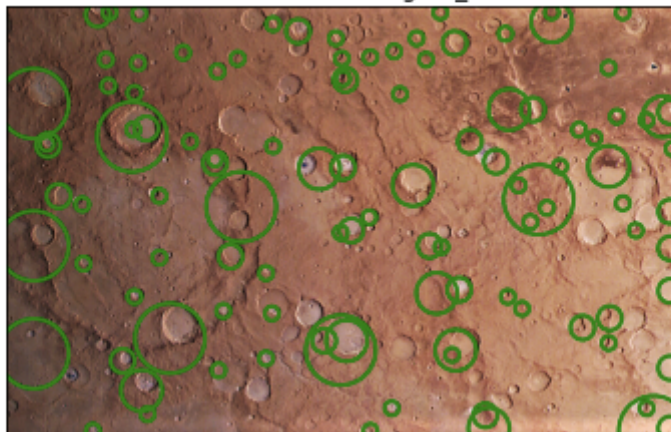


Σχήμα 19: Συγκρίσεις κριτηρίου blob R για 3 κλίμακες

Για μικρές κλίμακες κοντά στο $\sigma = 2$ μπορούμε να διακρίνουμε στην προσέγγιση του κριτηρίου μια ομοιότητα, ενώ όσο οι κλίμακες αυξάνονται η παρατήρηση με το μάτι γίνεται σχεδόν αδύνατη.

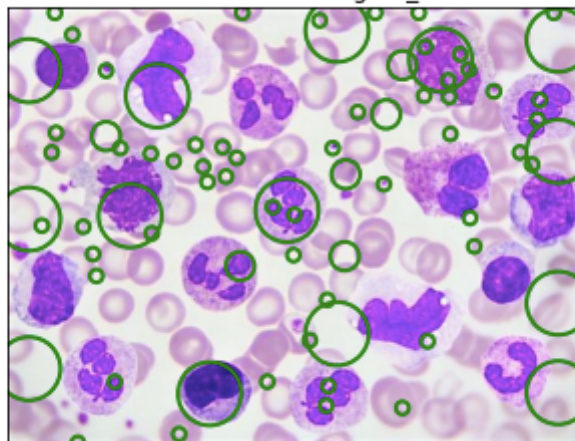
Παρακάτω βλέπουμε τα αποτελέσματα μετά από την επιτάχυνση της μεθόδου και την προσέγγιση του κριτηρίου ανίχνευσης blob:

Speeded up Multiscale Blob Detection N=4, sigma_0=2, s=1.7, theta_blob=0.003



Σχήμα 20: Πολυκλιμακωτή ανίχνευση Blob στο mars με box filters

Speeded up Multiscale Blob Detection N=3, sigma_0=2, s=2.2, theta_blob=0.002



Σχήμα 21: Πολυκλιμακωτή ανίχνευση Blob στο blood_smear με box filters

Παρατηρούμε ικανοποιητική ανίχνευση σε μεγάλες κλίμακες ενώ αρκετά χειρότερη σε μικρές, καθώς και κάποια σφάλματα που παρατηρούνται γύρω από τα κατακόρυφα edges της εικόνας.

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

Στο μέρος αυτό, εφαρμόζουμε τοπικούς περιγραφητές (local descriptors) πάνω στα σημεία ενδιαφέροντος που παράγουν οι ανιχνευτές που υλοποιήσαμε στο προηγούμενο μέρος, με σκοπό την εξαγωγή χαρακτηριστικών, δηλαδή την κωδικοποίηση της πληροφορίας που υπάρχει σε μια περιοχή γύρω από κάθε σημείο ενδιαφέροντος. Τα παραπάνω θα εφαρμοστούν για να πετύχουμε ταίριασμα (matching) και κατηγοριοποίηση (classification) εικόνων.

3.1 Ταίριασμα Εικόνων υπό Περιστροφή και Αλλαγή Κλίμακας

Οι περιγραφητές που θα χρησιμοποιήσουμε είναι οι SURF (Speeded Up Robust Features) και οι HOG (Histogram of Oriented Gradients).

Εν συντομία, οι SURF υπολογίζουν τη γενική κατεύθυνση της εικόνας βρίσκοντας το μεγαλύτερο διάνυσμα που σχηματίζεται για ένα κυκλικό τμήμα της απεικόνισης των αποκρίσεων dx, dy για κάθε pixel, οι οποίες υπολογίζονται με χρήση Haar Wavelets. Στη συνέχεια, η εικόνα στρέφεται στην κατεύθυνση που υπολογίστηκε και χωρίζεται σε blocks. Η τοπική περιγραφή επιτυγχάνεται επιλέγοντας ως διάνυσμα χαρακτηριστικών, το $v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$ για κάθε block.

Σε ό,τι αφορά τους HOG, υπολογίζουν την παράγωγο, δηλαδή μέτρο και γωνία, κάθε pixel της εικόνας με τελεστή κεντρικών διαφορών και δημιουργούν ένα ιστόγραμμα κατεύθυνσης που έχει ως bins εύρη γωνιών και το voting γίνεται διαβαθμισμένο με βάση το μέτρο. Η τοπική περιγραφή εδώ, επιτυγχάνεται χωρίζοντας την εικόνα σε κελιά (επικαλυπτόμενα ή μη) και εξάγοντας ένα ιστόγραμμα για κάθε κελί. Ενοποιώντας τα ιστογράμματα, προκύπτει ένα ενιαίο διάνυσμα χαρακτηριστικών.

Έχοντας τα feature vectors για κάθε εικόνα, θα βασιστούμε στον αλγόριθμο RANSAC (Random sample consensus) για εύρεση μιας αντιστοίχισης που θα μας οδηγήσει σε ένα μετασχηματισμό ομοιότητας, για να επιτύχουμε matching. Δεδομένης, δηλαδή, της περιγραφής μιας περιοχής μιας παραμορφωμένης εικόνας, να μπορέσουμε να βρούμε τα σημεία μιας εικόνας αναφοράς. Ο αλγόριθμος αυτός είναι επαναληπτικός και εκτιμά τις παραμέτρους ενός μοντέλου (στη συγκεκριμένη περίπτωση, του μετασχηματισμού ομοιότητας) από ένα σύνολο δεδομένων με outliers. Εφαρμόζοντας το μετ/σμό, προκύπτουν αντιστοιχίσεις από τις οποίες κρατιέται ένα υποσύνολο (inliers) με βάση κάποιο κριτήριο ανοχής. Τελικά, επιλέγεται ο μετ/σμός που οδηγεί στα περισσότερα inliers.

Συνολικά, χρησιμοποιήθηκαν 3 εικόνες ως αναφορά και η κάθε μία παραμορφώθηκε σε 4 γωνίες και 3 κλίμακες. Παρακάτω φαίνεται ο συγκεντρωτικός πίνακας με τα μέσα σφάλματα κλίμακας και γωνίας για τον κάθε συνδυασμό ανιχνευτή-περιγραφητή που εφαρμόστηκε και στις 3 εικόνες:

Detectors	Descriptors	Average scale error	Average theta error
Harris-Stephens	SURF	0.002	0.362
	HOG	0.360	16.351
Multiscale Harris	SURF	0.034	7.017
	HOG	0.266	27.935
Blob Detection	SURF	0.008	0.996
	HOG	0.161	14.731
Multiscale Blob	SURF	0.008	1.404
	HOG	0.164	15.331
Multiscale Box Filters	SURF	0.003	0.263
	HOG	0.181	15.714

Πίνακας 1: Σφάλματα ταιριάσματος για τους συνδυασμούς ανιχνευτών-περιγραφητών

Ο περιγραφητής SURF είναι ανεξάρτητος, όχι μόνο από την κλίμακα, αλλά και τη γωνία παραμόρφωσης, το οποίο προκύπτει από τον τρόπο υπολογισμού του, που περιγράψαμε παραπάνω. Αυτό το συμπέρασμα απεικονίζεται και στα αποτελέσματα της αξιολόγησης του ταιριάσματος, όπου τα σφάλματα είναι πολύ μικρά σε όλους τους ανιχνευτές (με εξαίρεση τη γωνία με την πολυκλιμακωτή Harris).

Σε ό,τι αφορά τον περιγραφητή HOG, φαίνεται να είναι πολύ ευαίσθητος στην παραμόρφωση γωνίας, με βάση τα αποτελέσματα που παίρνουμε ($> 15^\circ$ σφάλμα). Θεωρητικά, αυτό εξηγείται από το γεγονός ότι υπολογίζει την κατευθυντική παράγωγο, οπότε μία εικόνα θα έχει αρκετά διαφορετικό ιστόγραμμα από την εικόνα που προκύπτει από την περιστροφή της πρώτης κατά μία γωνία (π.χ. 20°).

Συγκριτικά, ο περιγραφητής SURF είναι ξεκάθαρα αποτελεσματικότερος με κάθε ανιχνευτή, γεγονός που αναμένουμε με βάση τα πλεονεκτήματά του σε σχέση με τον HOG.

Για τον HOG συγκεκριμένα, παρατηρούμε καλύτερη επίδοση στην ανίχνευση blob απ' ό,τι στην ανίχνευση γωνιών, που μας δείχνει πως είναι καταλληλότερος σε τέτοιες εφαρμογές.

Συγκρίνοντας, τώρα, τις δύο εκδοχές πολυκλιμακωτής ανίχνευσης Blobs, παρατηρούμε μια σχετική βελτίωση στην περίπτωση με τα Box Filters, το οποίο αποτελεί ένα επιπλέον πλεονέκτημα πέραν της επιτάχυνσης της μεθόδου.

Γενικά, παρατηρούμε κατά τη διάρκεια του fine tuning, ένα trade-off μεταξύ σφάλματος κλίμακας και γωνίας, στην αλλαγή της παραμέτρου σ . Με μείωση, έχουμε μια μικρή ελάττωση του theta error και μια μικρή αύξηση του scale error.

3.2 Κατηγοριοποίηση Εικόνων

Στο τελευταίο μέρος, θα εφαρμόσουμε τους ανιχνευτές και τους περιγραφητές που έχουν αναλυθεί για κατηγοριοποίηση εικόνων που ανήκουν στις ακόλουθες κλάσεις: αυτοκίνητο, άνθρωπος, ποδήλατο. Τα στάδια του αλγορίθμου αποτελούνται από την εξαγωγή χαρακτηριστικών για όλη τη βάση, τη δημιουργία συνόλου δεδομένων για εκπαίδευση (train set) και για αξιολόγηση (test set) του classifier, την κατασκευή ενός Bag of Visual Words και τέλος την εφαρμογή ενός Support Vector Machine για την κατηγοριοποίηση.

Bag of Visual Words

Το ερώτημα 3.2.3 αφορά στην κατασκευή αναπαράστασης ενός Bag of Visual Words για το οποίο υλοποιήσαμε τις απαραίτητες συναρτήσεις στα αρχεία **bagOfVisualWords.py** και **histogramFunction.py**. Αρχικά, δημιουργήσαμε ένα ενιαίο διάνυσμα χαρακτηριστικών για τους περιγραφητές που εξήχθησαν από όλες τις εικόνες της βάσης, επιλογή ενός τυχαίου υποσυνόλου χαρακτηριστικών (50%) και εφαρμογή του αλγορίθμου kmeans με 500 κέντρα. Αφού δημιουργήθηκαν οι συστάδες, υπολογίστηκε το ιστόγραμμα για κάθε εικόνα, με υπολογισμό της ελάχιστης ευκλείδειας απόστασης μεταξύ κάθε περιγραφητή της εικόνας με κάθε κέντρο και δημιουργία κανονικοποιημένου ιστογράμματος με χρήση της **histogram** της numpy.

Συγκρίνοντας με την έτοιμη BagOfWords παρατηρούμε μια μικρή βελτίωση των ποσοτών επιτυχίας της κατηγοριοποίησης (1-3%), όμως με πολύ μεγαλύτερο runtime.

Detectors	Descriptors	Extraction time (sec)	Mean Accuracy (%)
Multiscale Harris	SURF	193.124	54.759
	HOG	188.316	56.690
Multiscale Blob	SURF	187.242	56.138
	HOG	238.213	66.207
Multiscale Box Filters	SURF	181.782	60.966
	HOG	211.929	66.897

Πίνακας 2: Ποσοστά επιτυχίας της κατηγοριοποίησης για τους συνδυασμούς ανιχνευτών-περιγραφητών

Παρατηρούμε ότι ο περιγραφητής HOG έχει μεγαλύτερα ποσοστά επιτυχίας από τον περιγραφητή SURF, παρ' όλο που όπως είδαμε στο προηγούμενο ερώτημα είχε μεγαλύτερο σφάλμα στην εύρεση της περιστροφής και της κλίμακας της παραμορφωμένης εικόνας.

Συγκριτικά, για τους ανιχνευτές γωνίας και blob παρατηρούμε ότι και για τους δύο τύπους περιγραφητών, οι Blob detectors δίνουν μεγαλύτερα ποσοστά επιτυχίας, που μας οδηγεί στο συμπέρασμα ότι οι εικόνες με τις οποίες έγινε το training περιγράφονται καλύτερα από visual words που “βρίσκονται” γύρω από blobs αντί γωνίες.

Από το συγκεκριμένο dataset, φαίνεται να είναι καταλληλότερος ο συνδυασμός πολυκλιμακωτής ανίχνευσης blob με box filters και περιγραφητή HOG. Αυτό μπορεί να διαφέρει ανάλογα με το μέγεθος και το είδος του dataset, οπότε δεν είναι πολύ ασφαλής επιλογή, ειδικά αν αναλογιστούμε ότι στο προηγούμενο ερώτημα παρατηρήσαμε καλύτερη επίδοση στον περιγραφητή SURF.

Αναφορές

- [1] P. Maragos, *Computer Vision: Chap. 10: Feature Detection & Descriptors*, NTUA June 2018
- [2] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., 2021
- [3] Y Zhang, R Jin, ZH Zhou, *Understanding bag-of-words model: a statistical framework*, 2010