



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Computer Vision

Εαρινό Εξάμηνο 2015-2016

Αναφορά 1ης Εργαστηριακής Άσκησης Μέρος 1 & 2

Μάρτιος 2016

Καλλάς Κωσταντίνος 03112057 - Τζίνης Ευθύμιος
03112007

8^ο Εξάμηνο, Σ.Η.Μ.Μ.Υ., Ε.Μ.Π.

Ο κώδικάς μας περιέχει αρκετές επεξηγήσεις και σχόλια για να γίνει κατανοητό οποιοδήποτε σημείο της υλοποίησής του αλλά και για τον διαχωρισμό μεταξύ των υποερωτημάτων.

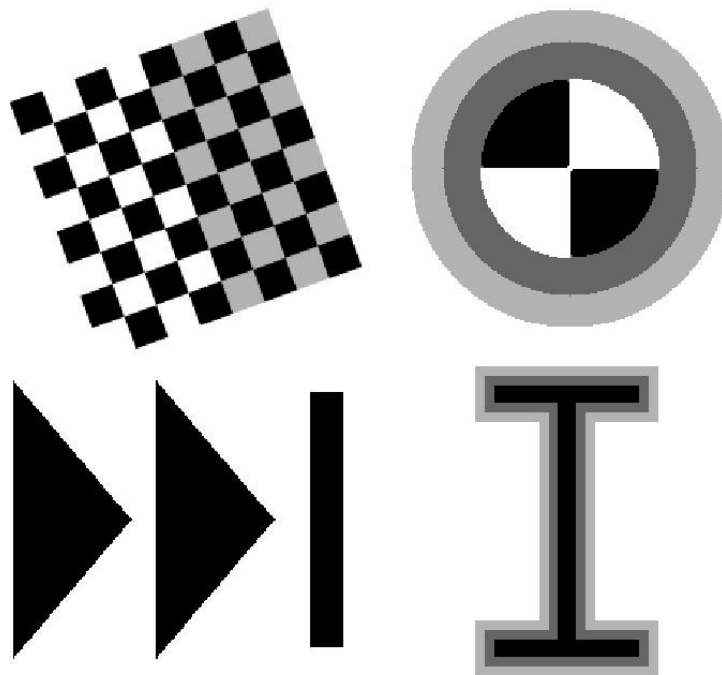
Μέρος 1ο

1.1

1.1.1

Διαβάζουμε την αρχική γκριζα εικόνα:

Original Image

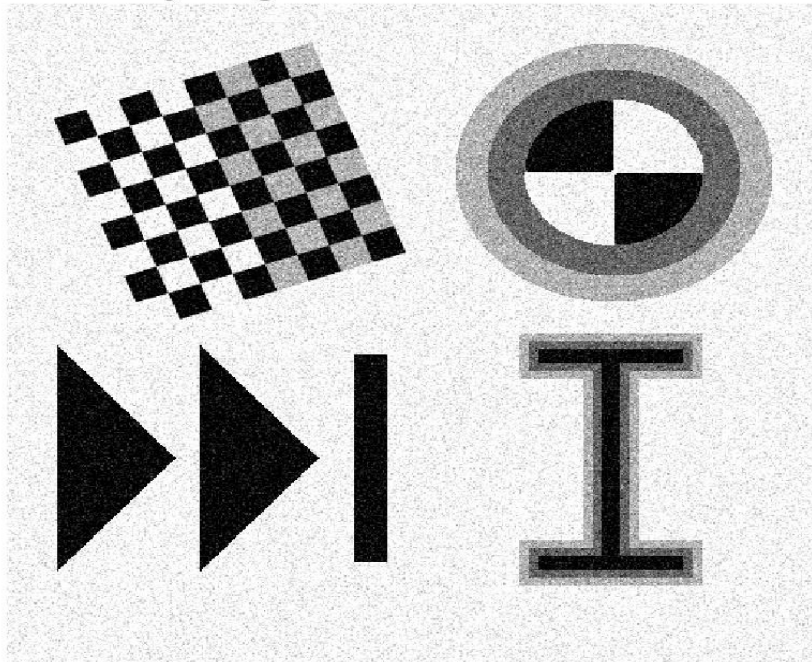


1.1.2

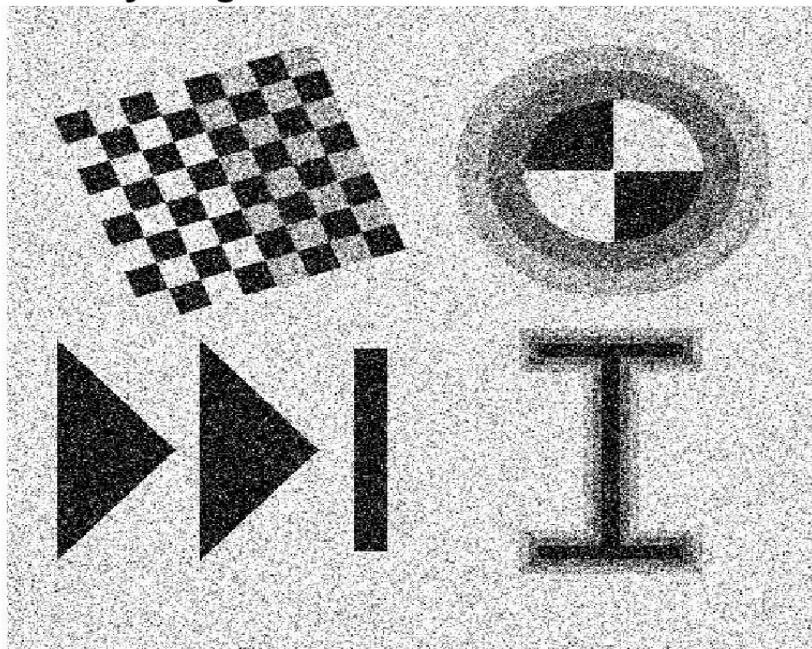
Με βάση την θεωρία βλέπουμε ότι η μείωση του PSNR αυξάνει τα επίπεδα θορύβου και υπολογίζουμε για τις δύο περιπτώσεις το ση που αντιστοιχεί.

Παρατίθενται οι 2 εικόνες με θόρυβο για τα αντίστοιχα επίπεδα του PSNR:

Noisy Image with PSNR=20 and sn=0.1



Noisy Image with PSNR=10 and sn=0.31623



1.2

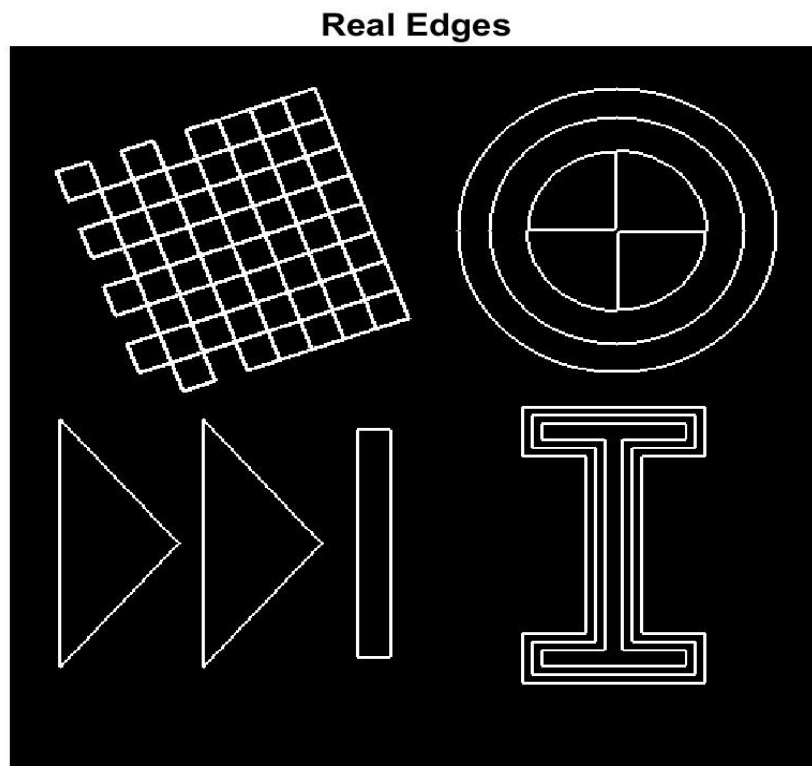
Υλοποιήσαμε με βάση τα βήματα που περιγράφονται στην εκφώνηση της άσκησης την συνάρτηση EdgeDetect που δέχεται σαν ορίσματα: την εικόνα εισόδου I , την τυπική απόκλιση σ της Gaussian, την παράμετρο θ_{edge} και τον τύπο προσέγγισης της Laplacian (1 για Γραμμική ή 2 για Μη-γραμμική). Επίσης, η συνάρτηση αυτή επιστρέφει στην έξοδο μία δυαδική εικόνα ακμών D , με τιμή 1 μόνο στα σημεία που επιλέγονται σαν ακμές.

1.3

Για να μπορέσουμε πράγματι να αξιολογήσουμε τα αποτελέσματα της μεθόδου θα πρέπει να βρούμε τις αληθινές ακμές (με κατωφλιοποίηση στην $T = M > \theta_{realedge}$).

Με βάση λοιπόν αυτές τις ακμές θα βρούμε τα εξής ποσοστά: το ποσοστό των ανιχνευθεισών ακμών που είναι αληθινές (Precision) και το ποσοστό των αληθινών ακμών που ανιχνεύθησαν (Recall).

Παρατίθεται η εικόνα των πραγματικών ακμών με επιλεγμένο κατώφλι $\theta_{realedge} = 0.1$:

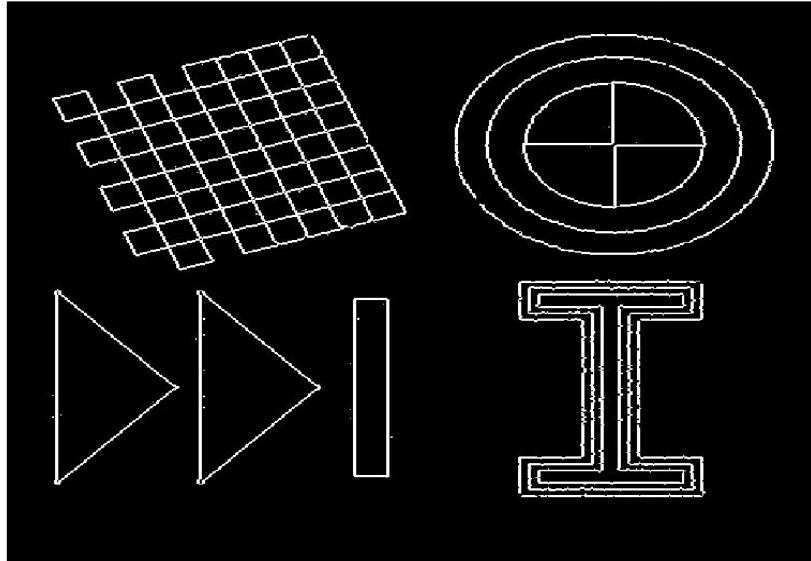


Παρατηρούμε ότι το η συγκεκριμένη τιμή του κατωφλίου μας βγάζει πολύ ικανοποιητικά αποτελέσματα για όλες τις ακμές της εικόνας.

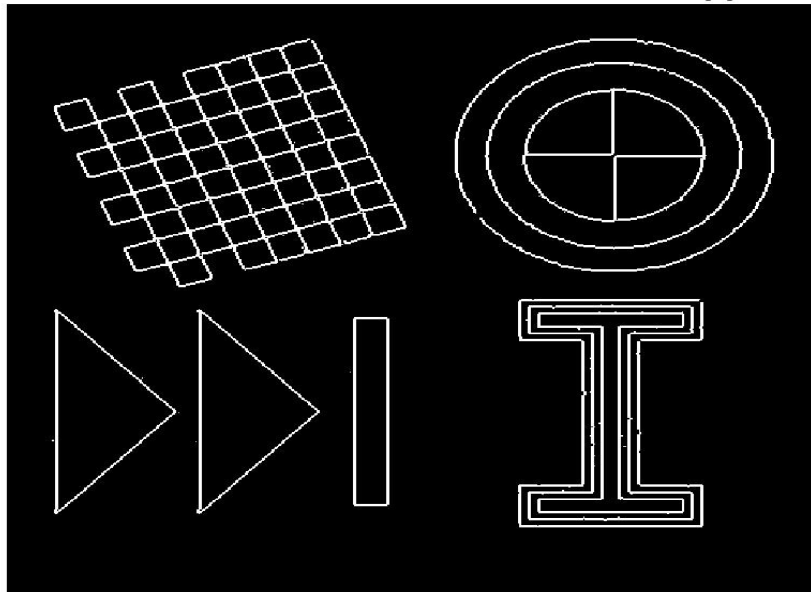
Για την εύρεση των κατάλληλων παραμέτρων σ και θ_{edge} χρησιμοποιήσαμε το script που φτιάξαμε `maximize_params.m` το οποίο βελτιστοποιεί για τα 2 επίπεδα PSNR και τις δύο μεθόδους (Γραμμική και μή-Γραμμική) το Recall και το Precision.

Παρατίθενται οι εικόνες που προέκυψαν από την ανίχνευση ακμών για τις 2 μεθόδους και τα 2 επίπεδα PSNR (τα καλύτερα αποτελέσματα μας):

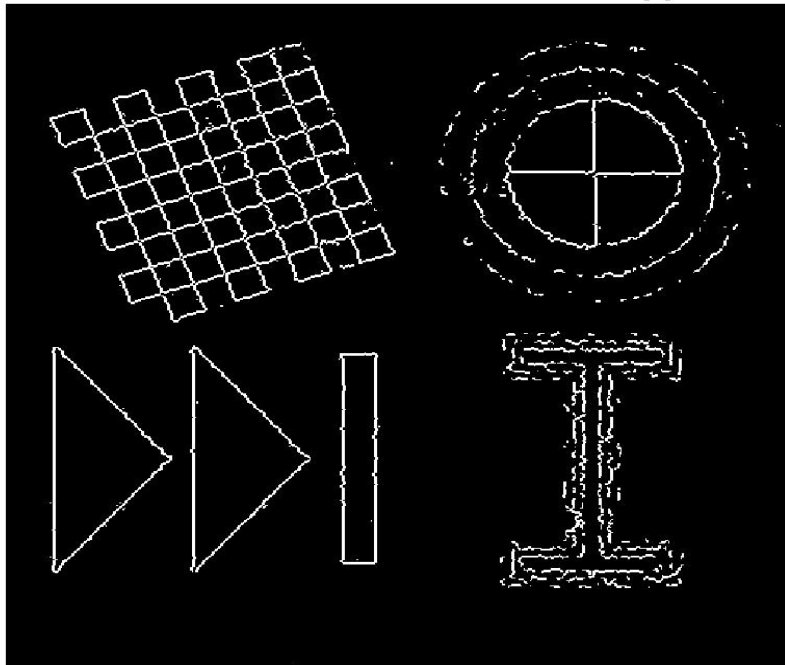
Edge Detection with PSNR=20 and Linear approximation



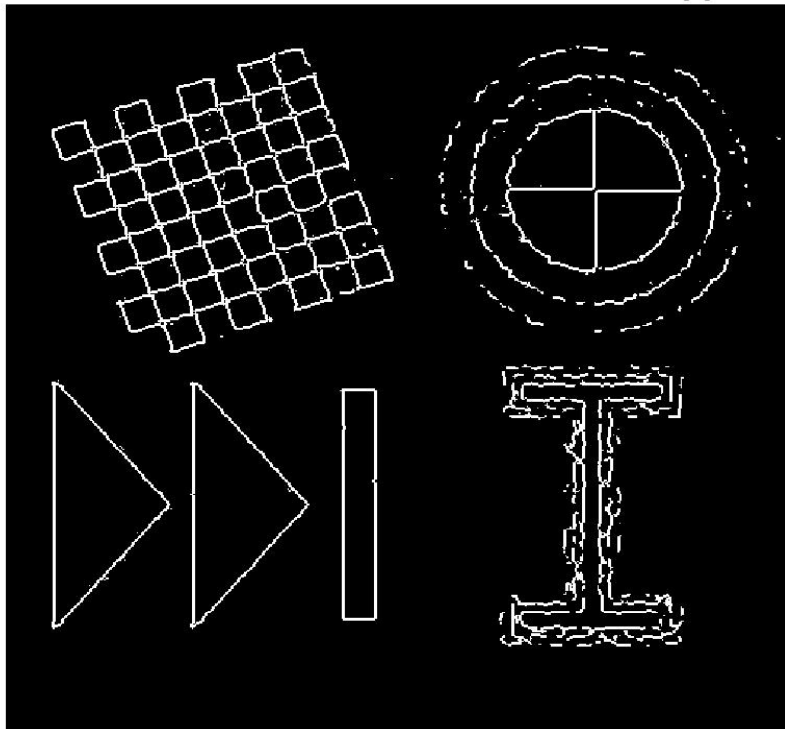
Edge Detection with PSNR=20 and Non-Linear approximation



Edge Detection with PSNR=10 and Linear approximation



Edge Detection with PSNR=10 and Non-Linear approximation



Για τις παραπάνω εικόνες παρατίθεται συγκεντρωτικά ο πίνακας των αποτελεσμάτων μας για άμεση σύγκριση μεταξύ των εικόνων και οι τιμές των παραμέτρων με τις οποίες πετύχαμε τα παρακάτω αποτελέσματα:

| PSNR | Approximation | Precision | Recall | C | σ | θ_{edge} |
|------|---------------|-----------|--------|--------|----------|-----------------|
| 20 | Linear | 97.49% | 89.31% | 93.4% | 1.3 | 0.19 |
| 20 | Non-Linear | 98.4% | 92.2% | 95.3% | 1.3 | 0.18 |
| 10 | Linear | 90.86% | 63.14% | 77% | 1.7 | 0.28 |
| 10 | Non-Linear | 89.81% | 70.17% | 79.99% | 1.9 | 0.26 |

Παρατηρούμε ότι οι μεταβολές των παραμέτρων έχουν διαφορετικές επιδράσεις πάνω στην διαδικασία ανίχνευσης των ακμών. Για την παράμετρο σ γνωρίζουμε ότι όσο αυξάνεται, το φίλτρο της αρχικής Gaussian (μεγέθους $n \times n$) γίνεται πιο ομαλό αλλά ταυτόχρονα και πιο μεγάλο λόγω της εξάρτησης του n από το σ , αυτό σημαίνει ότι η εικόνα θολώνεται πολύ περισσότερο και έτσι οι ακμές γίνονται πιο δυσδιάκριτες. Για την παράμετρο θ_{edge}

γνωρίζουμε ότι καθώς αυξάνεται θέτει ένα μεγαλύτερο φράγμα στα zero crossings που βρίσκουμε, ανάλογα με την κλίση τους (είναι πρακτικά το ποσοστό της μέγιστης κλίσης που έχουμε βρει και γι αυτό παίζει το ρόλο του κατωφλίου). Προφανώς οι εικόνες με μεγαλύτερα επίπεδα θορύβου δεν έχουν τόσο καλά αποτελέσματα και αυτό είναι λογικό διότι εμφανίζονται αρκετές περισσότερες ακμές από την αρχική εικόνα εξαιτίας του θορύβου που είναι προφανώς λάθος. Προκειμένου να τις κόψουμε, αναγκάζομαστε να τροποποιήσουμε τις παραμέτρους και να χάσουμε και ακμές που ήταν πραγματικές. Για αυτό τον λόγο, οι εικόνες με μεγαλύτερο θόρυβο βλέπουμε ότι έχουν πτώση κυρίως στο Recall και όχι στο Precision διότι το δεύτερο εξαρτάται κυρίως από την μέθοδο που εφαρμόζουμε για την ανίχνευση ακμών. Επιπλέον, η μη-γραμμική προσέγγιση της Laplacian μας δίνει και στις δύο εικόνες καλύτερα αποτελέσματα στην ανίχνευση ακμών.

Μέρος 2ο

2.1

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

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

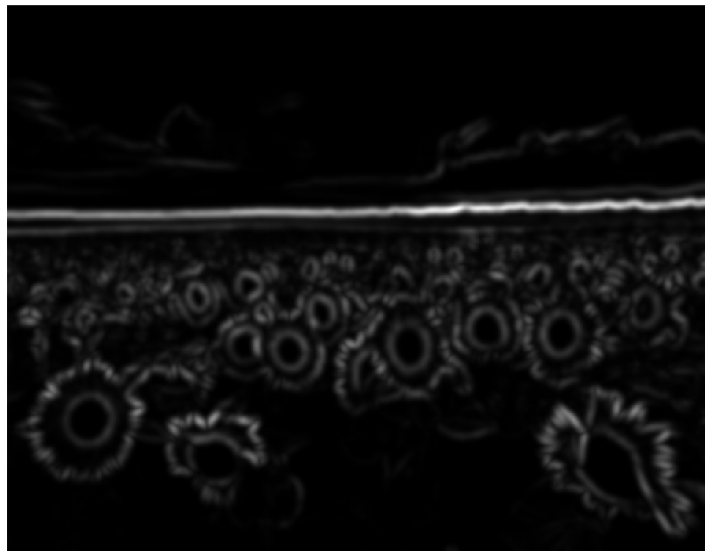
$$J_2(x, y) = G_\rho * \left(\frac{\partial I_\sigma}{\partial x} \cdot \frac{\partial I_\sigma}{\partial y} \right) (x, y)$$

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

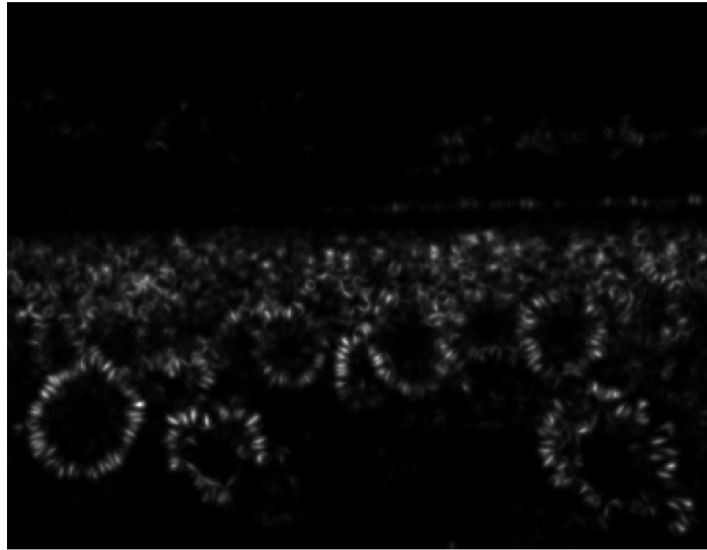
2.1.2) Υπολογίζουμε τις ιδιοτιμές του \mathbf{J} λ_+, λ_- σύμφωνα με τη σχέση που φαίνεται παρακάτω.

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

Ορίστε και οι ιδιοτιμές λ_+, λ_- σαν γκρίζες εικόνες.



λ_+



λ_-

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 $\theta_{corn} \cdot R_{max}$ και αναπαρηστούμε την εικόνα με τις γωνίες με την συνάρτηση που μας δόθηκε.

Ενδεικτικές τιμές για τις παραμέτρους που χρησιμοποιήσαμε για το παρακάτω αποτέλεσμα:

- $\sigma = 2$
- $\rho = 2.5$
- $k = 0.007$
- $\theta_{corn} = 0.01$



Οι γωνίες που ανιχνεύτηκαν για τις παραπάνω παραμέτρους

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

$$\begin{aligned}\sigma_0, \sigma_1, \dots, \sigma_{N-1} &= s^0 \sigma_0, s^1 \sigma_0, \dots, s^{N-1} \sigma_0 \\ \rho_0, \rho_1, \dots, \rho_{N-1} &= s^0 \rho_0, s^1 \rho_0, \dots, s^{N-1} \rho_0\end{aligned}$$

2.2.2) Ύστερα υπολογίσαμε την κανονικοποιημένη LoG για κάθε κλίμακα σί 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:

$N = 4$, $k = 0.001$, $\sigma_0 = 2.3$, $\rho_0 = 1$, $s = 1.85$, $\theta_{corn} = 0.01$.



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

2.3.1) Σε αυτό το υποερώτημα ανιχνεύουμε Blobs στην ίδια εικόνα με τρόπο που θα εξηγήσουμε παρακάτω.

Αρχικά υπολογίζουμε τον πίνακα Hessian για κάθε pixel της εικόνας όπως φαίνεται παρακάτω.

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

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

$\sigma = 2$, $\theta_{corn} = 0.12$.



Τα Blobs που ανιχνεύτηκαν με την παραπάνω μέθοδο και παραμέτρους.

2.4.1) Υπολογίζουμε τα Blobs για πολλές κλίμακες με τη μέθοδο που περιγράφηκε παραπάνω για την ανίχνευση γωνιών σε πολλές κλίμακες.

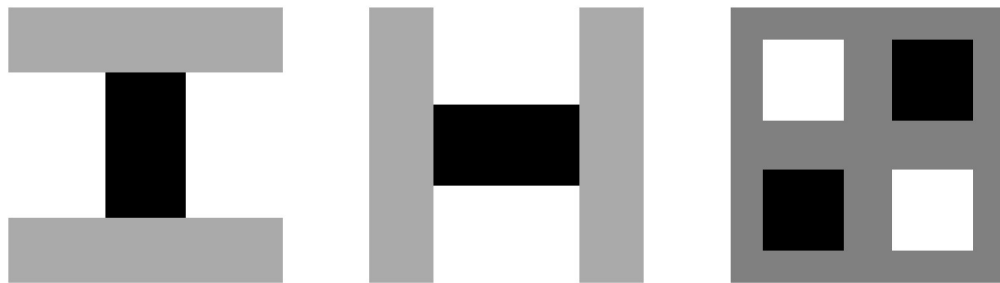
Οι παράμετροι που χρησιμοποιήσαμε είναι οι εξής: **$N = 4$, $\sigma_0 = 2$, $s = 1.85$, $\theta_{corn} = 0.07$**



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

2.5.1) Σε αυτό το υποερώτημα χρησιμοποιούμε Box-Filters και integral images για να βελτιώσουμε την απόδοση της υλοποίησης μας. Χρησιμοποιήσαμε προσεγγίσεις των φίλτρων δεύτερης παραγώγου με box filters. Αρχικά υπολογίσαμε το Integral Image για την εικόνα μας με την χρήση της συνάρτησης `integralImage()` αφού του κάναμε πρώτα το απαραίτητο padding.

2.5.2) Υπολογίζουμε τα Box-Filters με την χρήση της συνάρτησης του Matlab `integralKernel()`. Παρατίθενται ενδεικτικά τα D_{xx} , D_{yy} , D_{xy} για $\sigma = 2.5$.



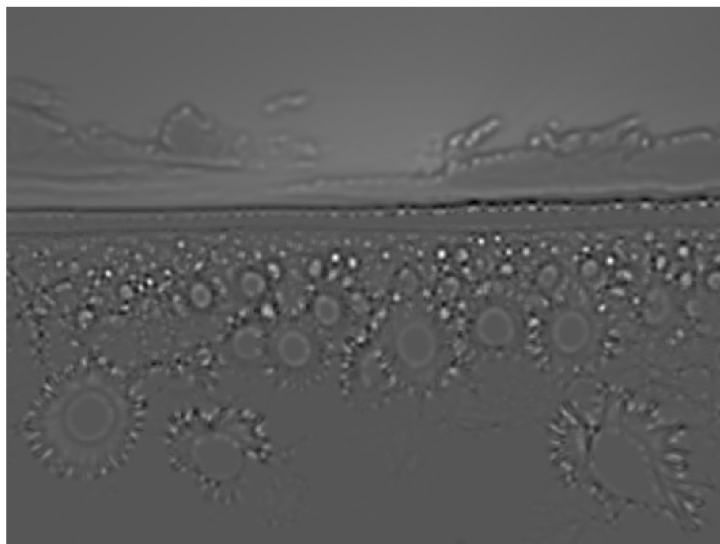
D_{xx} , D_{yy} , D_{xy} για $\sigma = 2.5$

Υστερά υπολογίζουμε τα L_{xx} , L_{yy} , L_{xy} μια συνάρτηση που φτιάξαμε `computesS_integral` που βοηθάει στον υπολογισμό της συνέλιξης με τη χρήση του integral image .

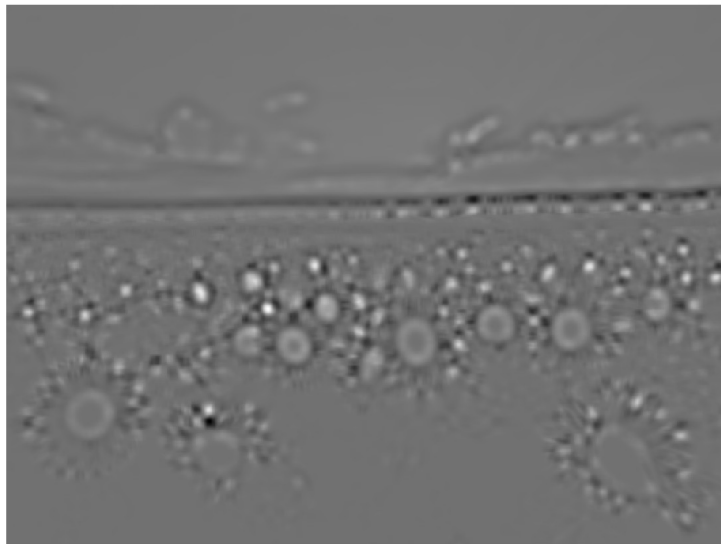
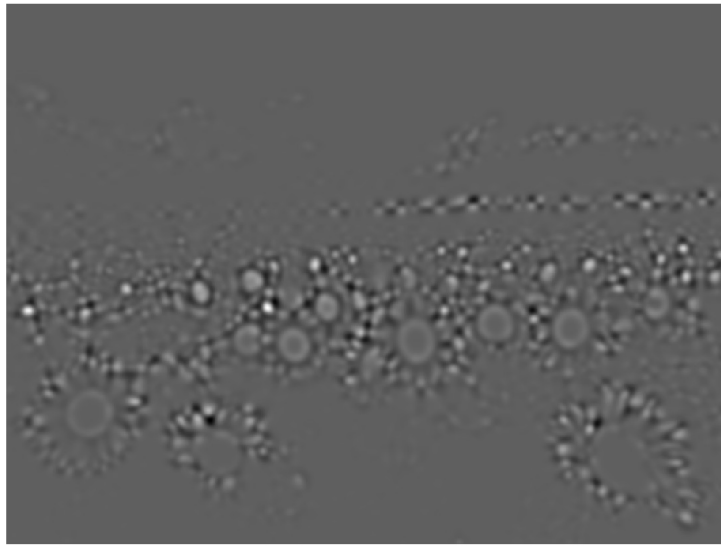
2.5.3) Ύστερα υπολογίζουμε το κριτήριο “Blobness” με τον παρακάτω τρόπο:

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

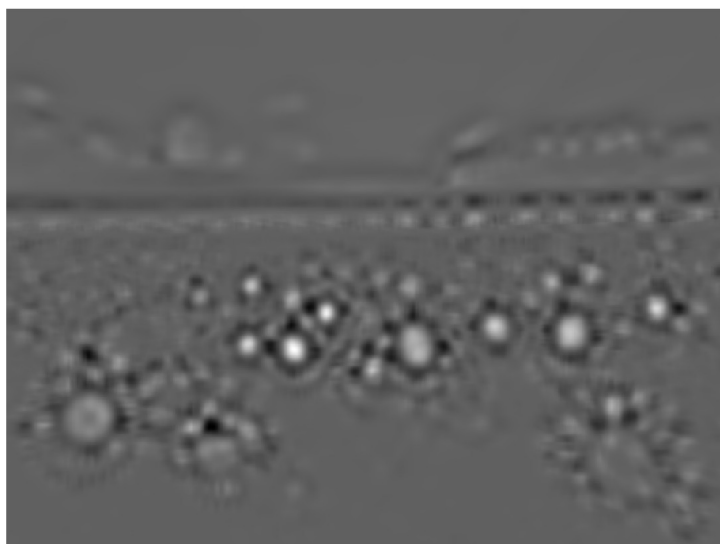
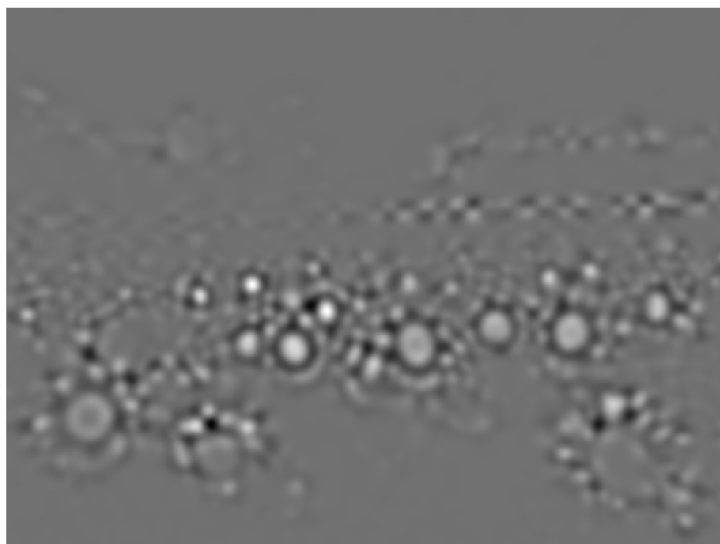
Παρακάτω παρατίθενται τα R που υπολογίζονται από την Hessian μέθοδο και τα R που υπολογίζονται από την μέθοδο με τα Box-Filters για 4 κλίμακες (2.5 ; 4.5 ; 7.5 ; 9.5)



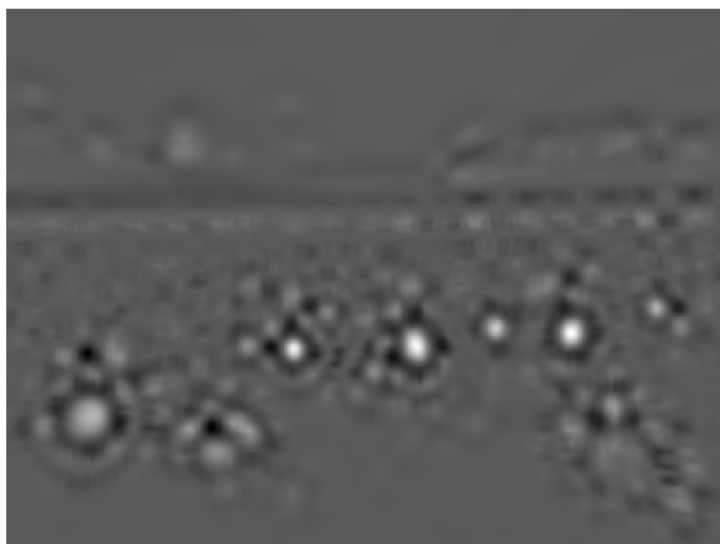
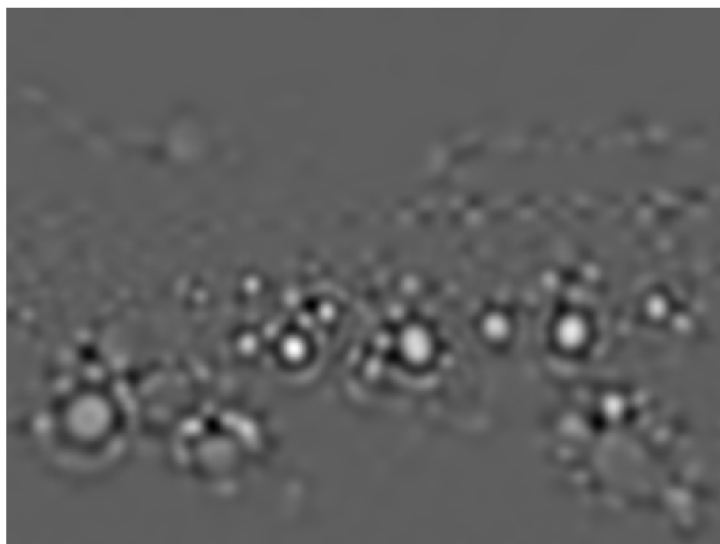
Hessian και Box για κλίμακα $\sigma = 2.5$



Hessian και Box για κλίμακα $\sigma = 4.5$



Hessian και Box με κλίμακα $\sigma = 7.5$



Hessian και Box-Filters με κλίμακα $\sigma = 9.5$

Γενικά παρατηρούμε ότι δεν υπάρχει μεγάλη διαφορά ανάμεσα στα δύο φίλτρα ακόμη και σε μεγαλύτερες κλίμακες.

2.5.4) Την πολυκλιματική ανίχνευση σημείων ενδιαφέροντος την υλοποιούμε όπως στα προηγούμενα υποερωτήματα με τη χρήση της κανονικοποιημένης LoG.

Παρακάτω παρατίθενται τα αποτελέσματα τα οποία τα πετύχαμε με τις εξής παραμέτρους:

$N = 4$, $\sigma = 2.5$, $s = 1.78$, $\theta_{\text{corn}} = 0.02$



Σημεία ενδιαφέροντος για τις 4 κλίμακες που φαίνονται παραπάνω

Μέρος 3ο

3.1.1-2 Σε αυτό το μέρος της άσκησης χρησιμοποιήσαμε τους τοπικούς περιγραφητές SURF και HOG προκειμένου να κωδικοποιήσουμε μια γειτονιά από Interest points (τοπικοί περιγραφητές) και σε συνδυασμό με τους ανιχνευτές που έχουμε ήδη υλοποιήσει καλούμε την evaluation.p προκειμένου να μας δει για κάθε συνδυασμό περιγραφητή και ανιχνευτή την αντίστοιχη αποτίμηση. Αυτή η αποτίμηση αφορά την ανάθεση τιμών στις μεταβλητές Scale Error και Theta Error που αφορούν το λάθος σε κλίμακα και μοίρες αντίστοιχα του αντίστοιχου συνδυασμού ανιχνευτή-περιγραφητή. Παραθέτουμε τις βέλτιστες παραμέτρους που έχουμε βρει απο το προηγούμενο μέρος της άσκησης (Μέρος 2) και αφορούν τις 5 μεθόδους ανίχνευσης σημείων ενδιαφέροντος.

Corner Detection onyscale: **s=2, r=2.5, k=0.007, thetacorn=0.01,**

Corner Detection multiscale: **N=4,k=0.001,s0=2.3,**
r0=1,scale=1.85,thetacorn=0.01

Blob Detection onyscale: **threshold=0.12, s=2**

Blob Detection multiscale: **N=4, s0=2, scale=1.85, threshold=0.07**

BoxFilters multiscale: **N=4, s0=2.5, scale=1.78**

Με βάση λοιπόν τις παραπάνω παραμέτρους προκύπτουν τα παρακάτω αποτελέσματα για όλους τους πιθανούς συνδυασμούς ανιχνευτών - περιγραφητών.

Συγκεντρωτικά Αποτελέσματα:

| Descriptor | Detector | Scale Error | | | Theta Error | | |
|------------|-----------------------------|-------------|--------|--------|-------------|---------|---------|
| | | | | | | | |
| SURF | Corner Detection One Scale | 0.0589 | 0.0127 | 0.0377 | 3.2528 | 1.1001 | 1.4318 |
| HOG | | 0.2160 | 0.2415 | 0.2660 | 23.2993 | 21.6066 | 30.4521 |
| SURF | Corner Detection Multiscale | 0.0034 | 0.0054 | 0.0045 | 0.2582 | 0.6315 | 0.1910 |
| HOG | | 0.1645 | 0.2862 | 0.2596 | 16.5521 | 21.4931 | 21.5235 |
| SURF | Blobs OneScale | 0.2281 | 0.1220 | 0.0753 | 12.3447 | 8.3837 | 9.3220 |
| HOG | | 0.3748 | 0.3801 | 0.3556 | 23.7663 | 39.7516 | 33.9944 |
| SURF | Blobs Multiscale | 0.0014 | 0.0047 | 0.0071 | 0.1556 | 0.4206 | 0.2338 |
| HOG | | 0.1354 | 0.2657 | 0.3445 | 28.4926 | 21.0514 | 21.7270 |
| SURF | Box Filters Multiscale | 0.0041 | 0.0175 | 0.0095 | 0.3945 | 0.6756 | 0.4734 |
| HOG | | 0.2953 | 0.3533 | 0.2298 | 30.0825 | 30.5860 | 25.2955 |

Παρατηρούμε γενικότερα ότι ο περιγραφητής HOG έχει γενικότερα χειρότερη συμπεριφορά από τον SURF κυρίως στην παράμετρο λάθους της γωνίας γιατί η παράμετρος του λάθους της κλίμακας είναι περίπου η ίδια και στους δύο περιγραφητές. Αυτό οφείλεται στην υλοποίηση του SURF που αρχικά βρίσκει τον Hessian σε κάθε σημείο, μετά υπολογίζει τα σημεία ενδιαφέροντος και τελικά κανονικοποιεί τα ιστογράμματα με βάση την περιστροφή τους. Από την άλλη πλευρά ο περιγραφητής HOG πρώτα υπολογίζει τα gradients, έπειτα φτιάχνει τα ιστογράμματα με βάση την περιστροφή τους και στο τέλος κάνει merge αρκετά μπλοκς και τα κανονικοποιεί. Αυτό λοιπόν φαίνεται και στα πειραματικά αποτελέσματα που φαίνεται ότι ο περιγραφητής SURF είναι scale και rotation invariant ενώ ο HOG έχει αρκετά μεγάλο λάθος στην εκτίμησή του για την γωνία. Προφανώς βλέπουμε ότι οι πολυκλιμακωτές αντίστοιχες μέθοδοι ανίχνευσης υπερτερούν κατά πολύ των αντίστοιχων μονοκλιματικών τόσο στην κλίμακα όσο και στην περιστροφή. Αυτό είναι και το λογικό αφού και στα προηγούμενα μέρη της άσκησης φαίνεται ξεκάθαρα ότι η ανίχνευση των Interest points γίνεται πολύ καλύτερα με τις πολυκλιμακωτές μεθόδους.

3.2

Σε αυτό το μέρος προσπαθούμε να κατηγοριοποιήσουμε κάποιες εικόνες με βάση τους περιγραφητές και ανιχνευτές που έχουμε υλοποιήσει στα προηγούμενα ερωτήματα.

Για την υλοποίηση της κατηγοριοποίησης δημιουργήσαμε την συνάρτηση `Le_Bag_of_Words` και χρησιμοποιήσαμε το script classification που μας δώθηκε. Παρακάτω αναλύεται συνοπτικά η υλοποίηση του `Le_Bag_of_Words`.

Χρησιμοποιώντας την `kmeans` χωρίσαμε σε `k` clusters ένα τυχαίο υποσύνολο των features των εικόνων και ύστερα βρήκαμε σε ποιο cluster “ανήκει” κάθε feature κάθε εικόνας. Έτσι για κάθε εικόνα δημιουργήσαμε ένα ιστογράμμο που περιέχει την πληροφορία για το ποια clusters την “περιγράφουν καλύτερα”, το οποίο στο τέλος κανονικοποιήσαμε.

Ύστερα πειραματιστήκαμε κατηγοριοποιώντας τις εικόνες με όλους τους συνδυασμούς περιγραφητών και ανιχνευτών, για ένα συγκεκριμένο αριθμό clusters και ένα συγκεκριμένο ποσοστό των features πάνω στο οποίο κάνουμε το clustering. (Στην περίπτωση μας 1000 και 50% αντίστοιχα). Όλοι οι detectors είναι πολυκλιμακωτοί, ο αριθμός clusters είναι πάντα 1000 και το ποσοστό features που χρησιμοποιήθηκε για την `kmeans` είναι πάντα 50%.

Σημαντική παρατήρηση: Όλα τα αποτελέσματα περιλαμβάνουν κάποιο ποσοστό τυχαιότητας λόγω του τυχαίου υποσυνόλου features για την `kmeans`.

Παρακάτω παρουσιάζονται τα αποτελέσματα της κατηγοριοποίησης.

Detector: **Blob Detector**

Descriptor: **SURF**

Classification Accuracy: 67.586207 %

Classification Accuracy: 69.655172 %

Classification Accuracy: 64.827586 %

Classification Accuracy: 66.206897 %

Classification Accuracy: 67.586207 %

Average Classification Accuracy: **67.172414 %**

Detector: **Blob Detector με την χρήση Box-Filters**

Descriptor: **SURF**

Classification Accuracy: 64.827586 %

Classification Accuracy: 61.379310 %

Classification Accuracy: 68.275862 %

Classification Accuracy: 62.068966 %

Classification Accuracy: 64.137931 %

Average Classification Accuracy: **64.137931 %**

Detector: **Corner Detector**

Descriptor: **SURF**

Classification Accuracy: 64.137931 %

Classification Accuracy: 65.517241 %

Classification Accuracy: 71.724138 %

Classification Accuracy: 60.000000 %

Classification Accuracy: 58.620690 %

Average Classification Accuracy: **64.000000 %**

Detector: **Blob Detector**

Descriptor: **HOG**

Classification Accuracy: 73.103448 %

Classification Accuracy: 64.827586 %

Classification Accuracy: 71.034483 %

Classification Accuracy: 67.586207 %

Classification Accuracy: 62.068966 %

Average Classification Accuracy: **67.724138 %**

Detector: **Blob Detector με την χρήση Box-Filters**

Descriptor: **HOG**

Classification Accuracy: 66.206897 %

Classification Accuracy: 64.137931 %

Classification Accuracy: 66.896552 %

Classification Accuracy: 64.137931 %

Classification Accuracy: 66.206897 %

Average Classification Accuracy: **65.517241 %**

Detector: **Corner Detector**

Descriptor: **HOG**

Classification Accuracy: 68.275862 %

Classification Accuracy: 63.448276 %

Classification Accuracy: 72.413793 %

Classification Accuracy: 74.482759 %

Classification Accuracy: 60.689655 %

Average Classification Accuracy: **67.862069 %**

Συμπεράσματα:

Γενικά παρατηρούμε ότι δεν υπάρχει μεγάλη διαφορά ανάμεσα στους διάφορους συνδιασμούς περιγραφητή - ανιχνευτή. Συνολικά όμως με τη μέθοδο HOG έχουμε καλύτερα αποτελέσματα (βελτίωση της τάξης του 2%) και η καλύτερη μέθοδος ανίχνευσης είναι η μέθοδος ανίχνευσης Blobs.

Ας σημειώσουμε ξανά εδώ ότι λόγω τυχαιότητας δεν μπορούμε να αποφανθούμε ότι ένας συνδιασμός μεθόδων υπερέχει ξεκάθαρα των άλλων, αν όμως επιλέγαμε θα επιλέγαμε τον Blob Detector με τον περιγραφητή HOG.