Όραση Υπολογιστών

1η Εργασία

Ονοματεπώνυμο: Ιωάννα Σταγωνά

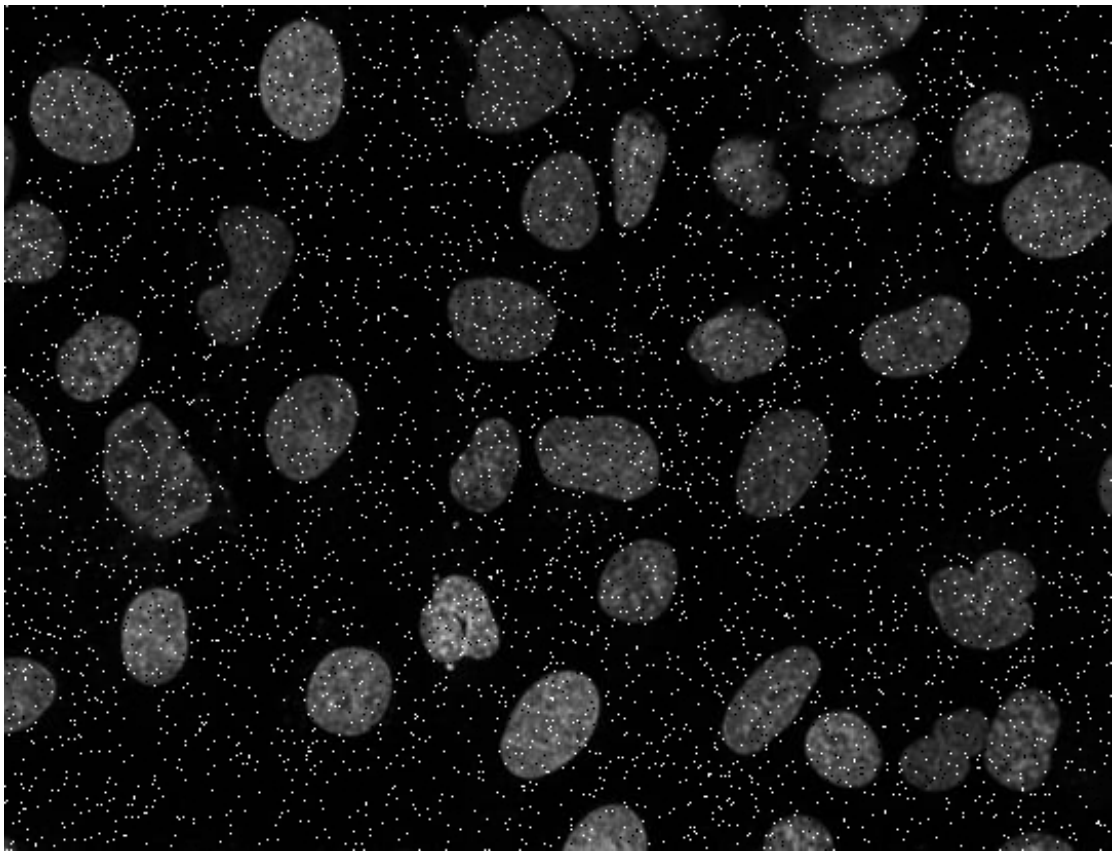
Εξάμηνο φοίτησης:7ο

Αριθμός Μητρώου:58125

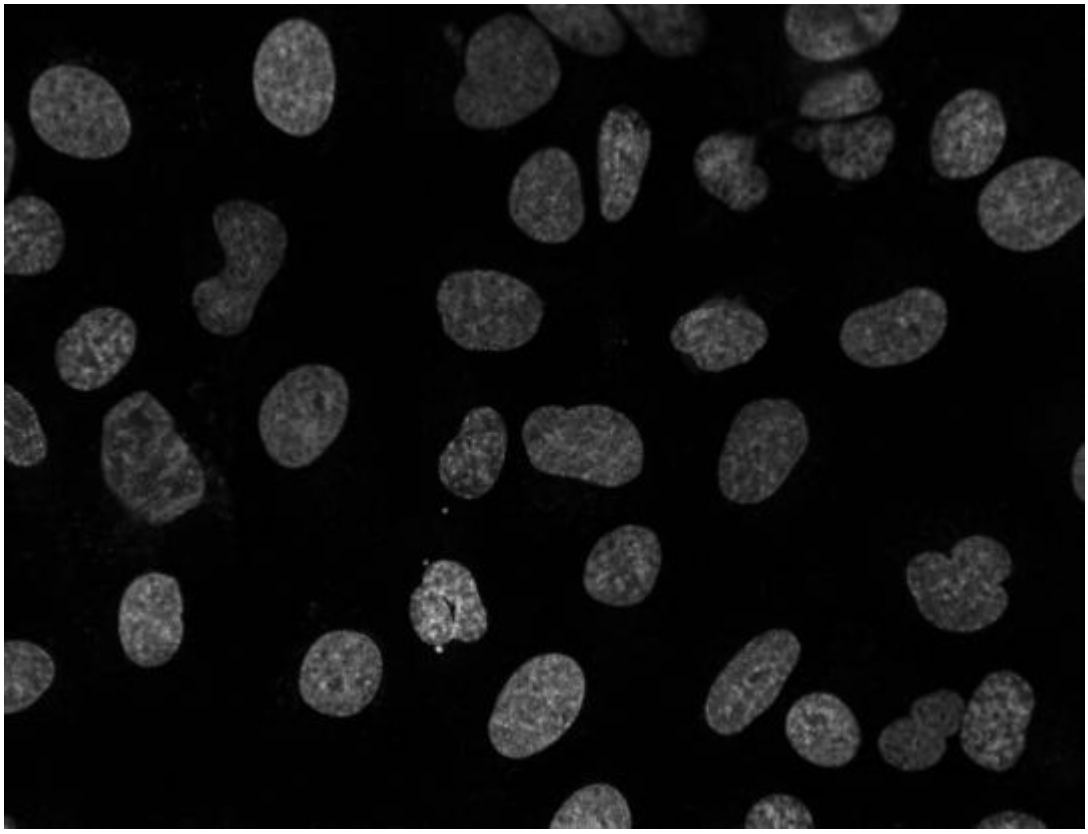
Εισαγωγή

Στην συγκεκριμένη εργασία εργάστηκα πάνω στην δημιουργία και εφαρμογή μη γραμμικού φίλτρου απόρριψης θορύβου και στην ανάλυση της δομής του περιεχομένου εικόνων μικροσκοπίου(ανίχνευση των κυττάρων, μέτρηση επιφάνειας του κυττάρου, σχεδιασμός και μέτρηση επιφάνειας του περιβάλλοντος κουτιού και τέλος υπολογισμός μέσης τιμής διαβάθμισης του γκρι των εικονοστοιχείων που περιέχονται στο περιβάλλον κουτί).

Οι εικόνες που αντιστοιχούν στον αριθμό μητρώου μου είναι αυτές με το νούμερο 5.

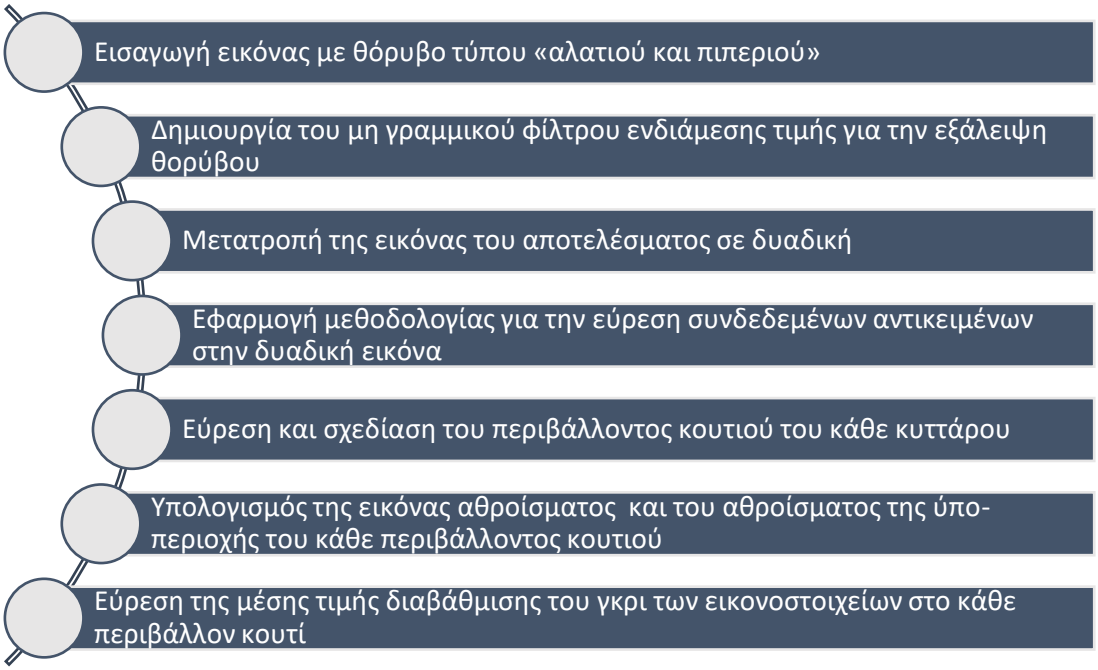


Εικόνα 1: Image 5 with Salt and Pepper noise



Εικόνα 2: Original image 5

Διάγραμμα επίλυσης

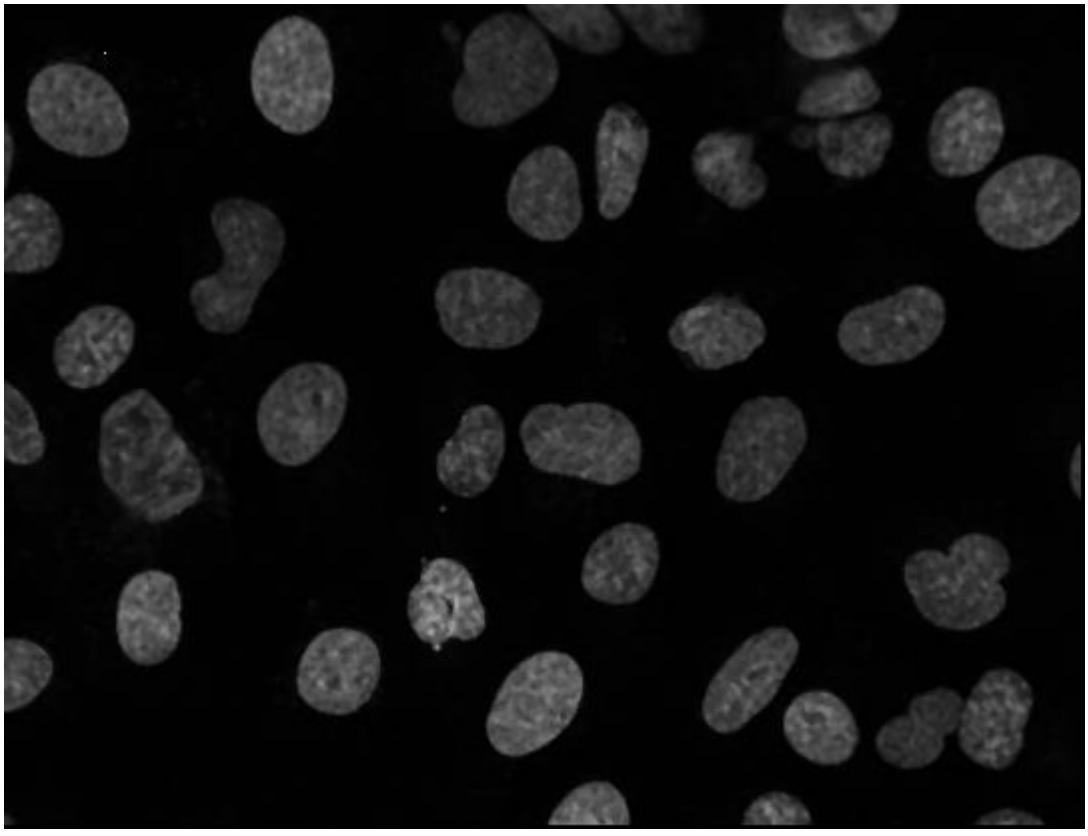


1.Εξάλειψη θορύβου τύπου «αλατιού και πιπεριού»

Για την εξάλειψη του θορύβου τύπου «αλατιού και πιπεριού» υλοποίησα το μη γραμμικό φίλτρο ενδιάμεσης τιμής.Το φίλτρο αυτό έχει την δυνατότητα να απομακρύνει ανεπιθύμητες τιμές και να καθαρίζει τον θόρυβο salt and pepper.

Αρχικά φορτώνω την εικόνα με τον θόρυβο, την μετατρέπω σε εικόνα διαβάθμισης του γκρι και δημιουργώ έναν πίνακα μηδενικών με διαστάσεις ίσες με αυτές της αρχικής εικόνας. Έπειτα ορίζω ένα κυλιόμενο kernel (έχω επιλέξει 3x3) το οποίο αρχικοποιώ με μηδενικές τιμές. Στην συνέχεια θα χρειαστώ 4 μετρητές,2 για τις γραμμές και τις στήλες της εικόνας και 2 μετρητές που θα διατρέχουν τις γραμμές και τις στήλες του kernel.Πρέπει να προσέξω ότι το κέντρο του kernel δεν μπορεί να βρεθεί στις ακριανές(πρώτη και τελευταία) στήλες και γραμμές τις εικόνας γι' αυτό και θέτω τους αντίστοιχους περιορισμούς στους μετρητές μου. Στην συνέχεια τοποθετώ το κεντρικό pixel του kernel στο pixel της εικόνας που θέλω να φιλτράρω και ταξινομώ τις τιμές των pixel μέσα στο παράθυρο σε αύξουσα διάταξη. Τέλος επιλέγω τον μεσαίο όρο και τον τοποθετώ στο αντίστοιχο κεντρικό pixel της φιλτραρισμένης εικόνας.

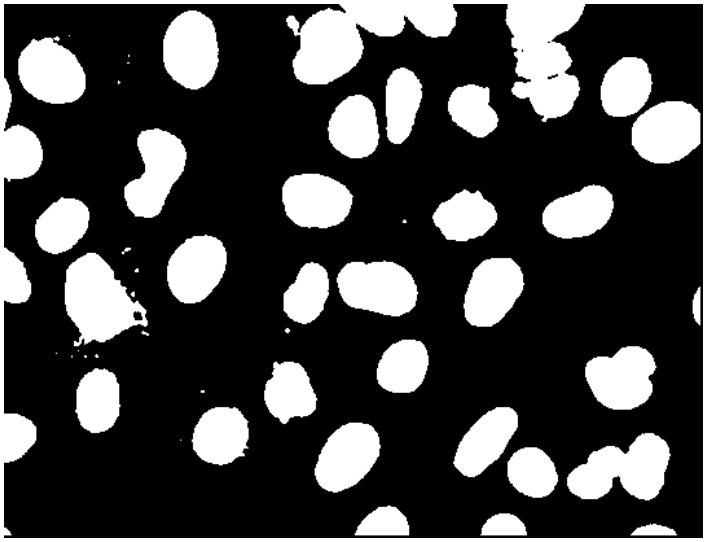
Με την αυτή διαδικασία προκύπτει η παρακάτω εικόνα:



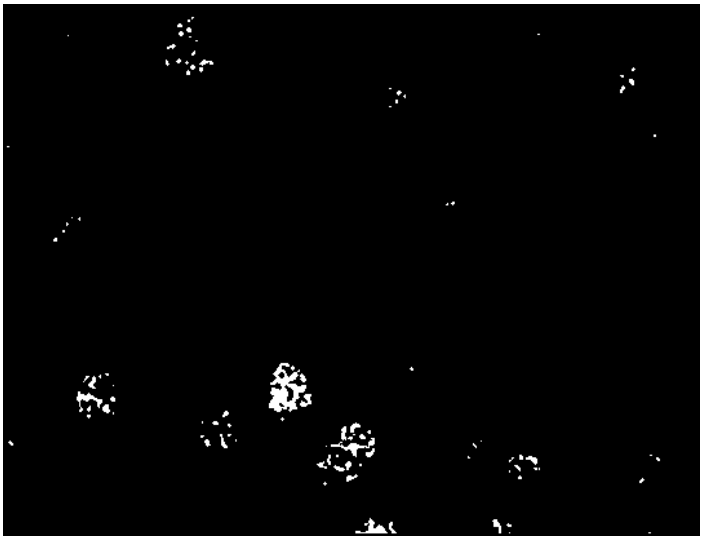
Εικόνα 3:Filtered image 5

2.Μετατροπή της εικόνας σε δυαδική

Για την παρακάτω διεργασία (εύρεση συνδεδεμένων αντικειμένων) θα χρειαστεί να γίνει μετατροπή της Εικόνας 3 σε δυαδική με την χρήση της συνάρτησης ‘cv2.threshold’. Θα πρέπει να επιλέξουμε το κατάλληλο κατώφλι έτσι ώστε να γίνει σωστά η μετατροπή. Παρατηρούμε ότι η εικόνα έχει μαύρο background οπότε τα περισσότερα της pixel θα είναι μαύρα. Με την σωστή επιλογή του κατωφλιού θα καταφέρουμε να «ενεργοποιήσουμε», αλλάζοντας την τιμή τους σε 1, τα pixel που μας αφορούν δηλαδή αυτά που ανταποκρίνονται στα διαφορετικά δαχτυλικά αποτυπώματα. Ενώ τα pixel του background θα πάρουν την τιμή 0.Θα πρέπει να προσέξουμε καθώς αν επιλέξουμε πολύ μεγάλη τιμή κατωφλιού, θα χάσουμε πολύ σημαντική πληροφορία καθώς τα pixel που μας ενδιαφέρουν έχουν γκρι διαβαθμίσεις. Αν πάλι επιλέξουμε πολύ μικρή τιμή κατωφλιού, τα αντικείμενα θα ενωθούν λόγο της ύπαρξης θορύβου και θα χάσουν το αρχικό σχήμα τους. Σκοπός μας είναι να δημιουργήσουμε μια δυαδική εικόνα που θα απεικονίζει το δυνατότερο σωστά την αρχική μας.

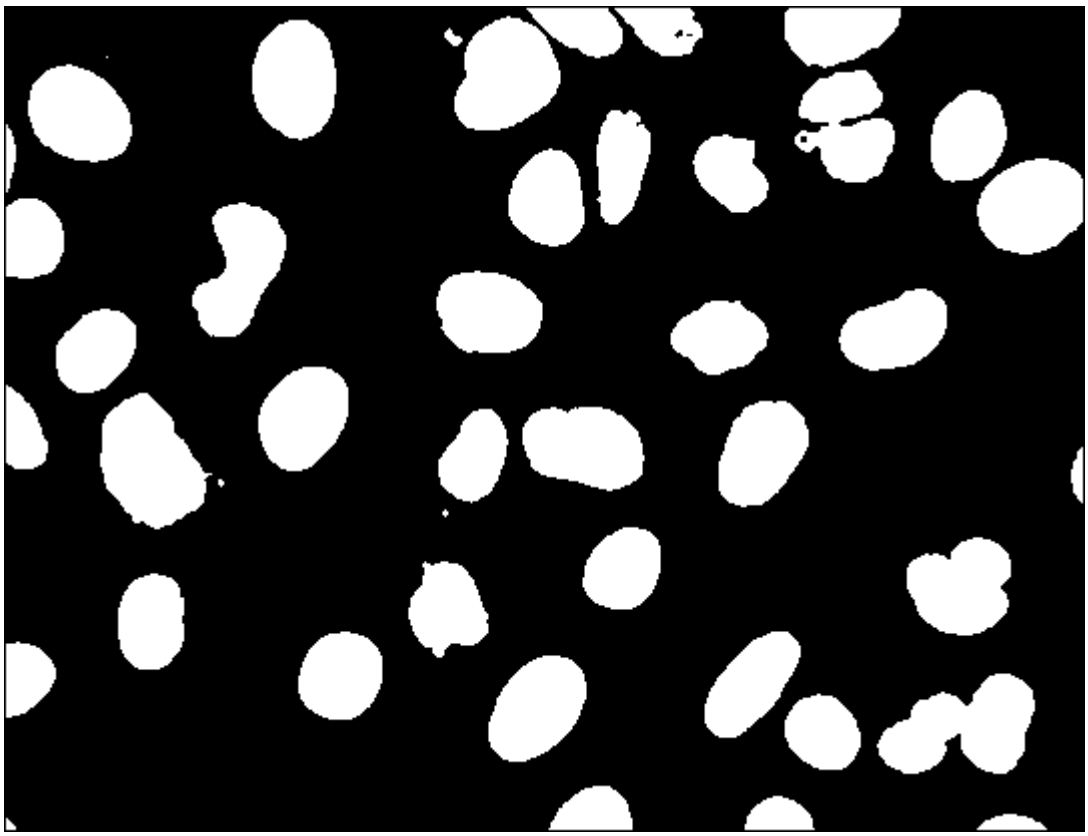


Εικόνα 4: Binary image with small threshold (10)



Εικόνα 5: Binary image with bid threshold (100)

Έπειτα από δοκιμές επιλέχθηκε η τιμή του κατωφλιού ίση με 25 και προέκυψε η παρακάτω εικόνα.



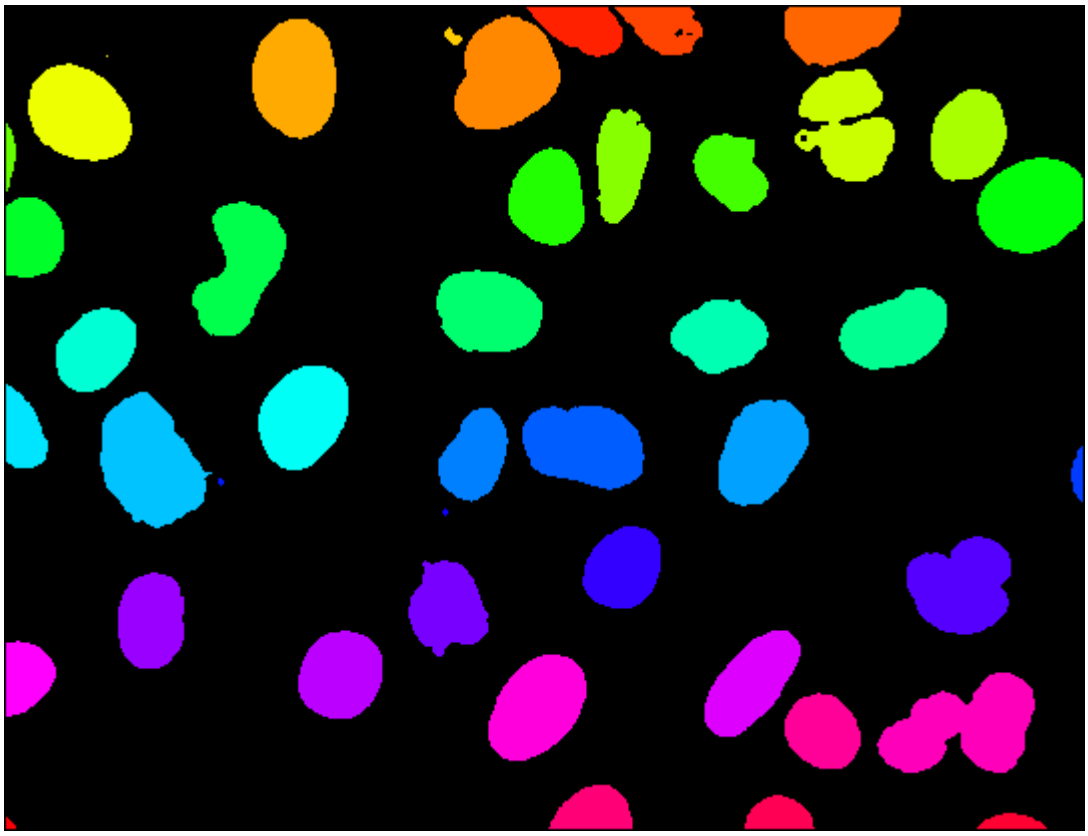
Εικόνα 6: Binary image with threshold=25

3.Εντοπισμός συνδεδεμένων αντικειμένων

Για τον εντοπισμό συνδεδεμένων αντικειμένων έγινε χρήση της συνάρτησης «cv2.connectedComponentsWithStats». Ο εντοπισμός συνδεδεμένων αντικειμένων γίνεται σαρώνοντας την δυαδική εικόνα pixel-pixel (από πάνω προς τα κάτω και από αριστερά προς τα δεξιά) προκειμένου να προσδιοριστούν οι συνδεδεμένες περιοχές pixel, δηλαδή οι περιοχές γειτονικών pixel που μοιράζονται το ίδιο σύνολο τιμών έντασης(στην περίπτωση μας το 1).Ο μετρητής σαρώνει την εικόνα μέχρι να φτάσει σε ένα pixel έστω p που θα έχει την τιμή 1.Όταν αυτό ισχύει, εξετάζει τους τέσσερις γείτονες του που έχουν ήδη συναντηθεί στη σάρωση , δηλαδή πάνω αριστερά ,αριστερά από αυτό και τους δύο άνω διαγώνιους. Με βάση αυτή την σάρωση προκύπτουν 3 περιπτώσεις:

- Εάν και οι 4 γείτονες είναι 0 , αντιστοιχίζει μια νέα ετικέτα στο p
- Εάν μόνο ένας γείτονας έχει 1, αντιστοιχίζει την ετικέτα του στο p
- Εάν περισσότεροι γείτονες έχουν 1,αντιστοιχίζει μια από τις ετικέτες στο p

Μετά την ολοκλήρωση της σάρωσης τα ισοδύναμα ζεύγη ετικετών ταξινομούνται σε ισοδύναμες ομάδες και μια μοναδική ετικέτα δίνεται σε κάθε ομάδα. Συμπληρωματικά, μπορούμε να χρωματίσουμε τις ετικέτες με διαφορετικά χρώματα.



Εικόνα 7:Labels displayed with different colors

Όπως φαίνεται και στην Εικόνα 7 υπάρχουν 3 ζεύγη αποτυπωμάτων τα οποία αναγνωρίζονται ως 3 συνδεδεμένα αντικείμενα αντί για 6 διαφορετικά. Αυτό συμβαίνει διότι τα αποτυπώματα αλληλεπικαλύπτονται στην μία περίπτωση και στις άλλες κάποια σημεία τους είναι ενωμένα ,με αποτέλεσμα να μην είναι ξεκάθαρα τα όρια τους και επομένως η συνάρτηση που υπολογίζει τα συνδεδεμένα αντικείμενα με την διαδικασία που αναφέρθηκε παραπάνω τα συμπεριφέρεται ως ένα, δίνοντας τους την ίδια ετικέτα. Το πρόβλημα αυτό στο ζεύγος της δεύτερης και τρίτης περίπτωσης θα μπορούσε να λυθεί εφαρμόζοντας την διαδικασία του «closing».

Η παραπάνω συνάρτηση επιλέχθηκε έναντι της απλής «cv2.connectedComponents» καθώς μου δίνει επιπλέον στοιχεία τα οποία θα χρειαστούν στην συνέχεια. Συγκεκριμένα, η συνάρτηση μου δίνει 4 εξόδους.

```
num_cc, labeled, stats, centroids =
cv2.connectedComponentsWithStats(binary)
```

num_cc: Ο αριθμός των ετικετών

labeled: Ο πίνακας των ετικετών, στο μέγεθος της εικόνας εισόδου όπου κάθε στοιχείο έχει μια τιμή ίση με την ετικέτα του.

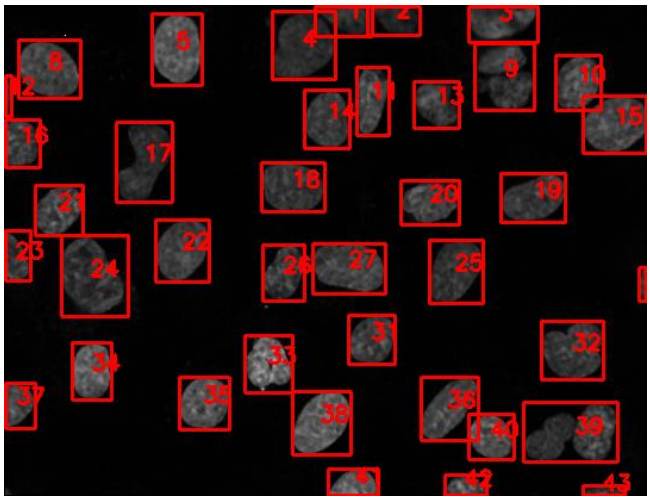
stats: Ο πίνακας των στατιστικών που υπολογίζει η συνάρτηση. Έχει μήκος ίσο με τον αριθμό των ετικετών και πλάτος ίσο με τον αριθμό των στατιστικών. Συγκεκριμένα τα διαθέσιμα στατιστικά για κάθε ετικέτα είναι τα εξής:

- **cv2.CC_STAT_LEFT:** Η πιο αριστερή συντεταγμένη(x) που αποτελεί την αρχή του bounding box στην οριζόντια κατεύθυνση.
- **cv2.CC_STAT_TOP:** Η ανώτατη συντεταγμένη (y) που συμπεριλαμβάνεται στην αρχή του bounding box στην κατακόρυφη κατεύθυνση.
- **cv2.CC_STAT_WIDTH:** Το οριζόντιο μέγεθος του bounding box.
- **cv2.CC_STAT_HEIGHT:** Το κατακόρυφο μέγεθος του bounding box.
- **cv2.CC_STAT_AREA:** Η συνολική περιοχή (σε pixel) του κυττάρου.

centroids: Ο πίνακας με τα τις συντεταγμένες του κέντρου για κάθε συνδεδεμένο αντικείμενο.

4.Σχεδίαση του περιβάλλοντος κουτιού

Για την σχεδίαση του περιβάλλοντος κουτιού θα χρησιμοποιήσω τον πίνακα στατιστικών που υπολογίστηκε με βάση την παραπάνω συνάρτηση.Το πρώτο συνδεδεμένο αντικείμενο στην θέση [0] είναι πάντα το background οπότε θα χρησιμοποιήσω έναν μετρητή ο οποίος θα διατρέχει τις θέσεις των πινάκων των στατιστικών από τη θέση [1] και μετά. Έτσι ορίζω ως x0 και ως y0 τις συντεταγμένες της έναρξης του κάθε περιβάλλοντος κουτιού ενώ ως x1=x0+w και y1=y0+h τις συντεταγμένες του τέλους του περιβάλλοντος κουτιού και με την χρήση της συνάρτησης «cv2.rectangle» γίνεται η σχεδίαση τους. Σημειώνεται ότι έχει προηγηθεί ο ορισμός ενός κατωφλιού στα pixel των επιφανειών που θέλω να σχεδιάσω περιβάλλον κουτί με την βοήθεια του στατιστικού **cv2.CC_STAT_AREA** ,με σκοπό να αποφευχθεί η σχεδίαση παραλληλογράμμου γύρω από πιθανό θόρυβο. Με την χρήση της συνάρτησης «cv2.putText» πραγματοποιώ την αρίθμηση τους χρησιμοποιώντας τον μετρητή που έχω δημιουργήσει. Έτσι τελικά παράγεται η παρακάτω εικόνα:



Εικόνα 8: All connected components with bounding box and labeled

5.Υπολογισμός επιφάνειας κυττάρων και περιβάλλοντος κουτιού

Για τον υπολογισμό της επιφάνειας των κυττάρων χρησιμοποίησα το στατιστικό **cv2.CC_STAT_AREA** ενώ η συνολική επιφάνεια του περιβάλλοντος κουτιού υπολογίστηκε χρησιμοποιώντας τα στατιστικά **cv2.CC_STAT_HEIGHT** και **cv2.CC_STAT_WIDTH** πολλαπλασιάζοντας το οριζόντιο μέγεθος επί το κατακόρυφο μέγεθος του.

6.Υπολογισμός της μέσης τιμής διαβάθμισης του γκρι των εικονοστοιχείων που περιέχονται στο περιβάλλον κουτί

Για τον συγκεκριμένο υπολογισμό , με τέτοιο τρόπο ώστε η ταχύτητα εκτέλεσης του να είναι ανεξάρτητη του μεγέθους της υπο-περιοχής δημιούργησα μία συνάρτηση που αρχικά παράγει τη συνολική εικόνα του αθροίσματος της Εικόνας 3 και στην συνέχεια το άθροισμα της υπο-περιοχής του περιβάλλοντος κουτιού .

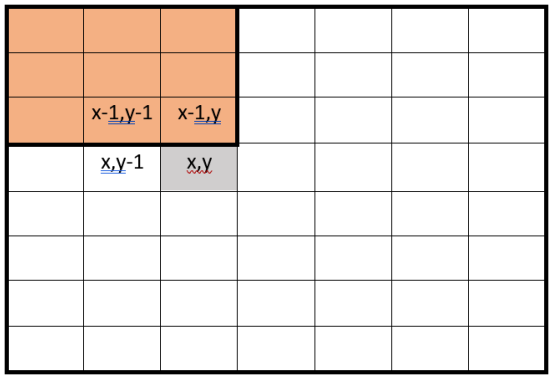
Μέθοδος υλοποίησης της συνάρτησης «integralimg»:

Αρχικά υπολογίζουμε τις διαστάσεις τις Εικόνας 3 και δημιουργούμε τον πίνακα «Summed Area Table» όπου τον ορίζω κατά μία γραμμή και στήλη μεγαλύτερος από την αρχική εικόνα. Αρχικοποιούμε την μεταβλητή sum και στην συνέχεια δημιουργούμε 2 μετρητές. Κάθε στοιχείο της εικόνας αθροίσματος θα περιέχει το άθροισμα του με τα στοιχεία που βρίσκονται πάνω και αριστερά από αυτό. Θα υπολογίσουμε τα αθροίσματα του υπόλοιπου πίνακα χρησιμοποιώντας κάθε φορά τους 3 γείτονες του στοιχείου. Για να μην χρειαστεί να υπολογίσω την πρώτη γραμμή και στήλη ξεχωριστά, θα αφήσω για ευκολία την πρώτη γραμμή και στήλη του πίνακα αθροίσματος με μηδενικά. Έτσι ο αλγόριθμος που υλοποίησα μπορεί να λειτουργήσει για ολόκληρο τον πίνακα.

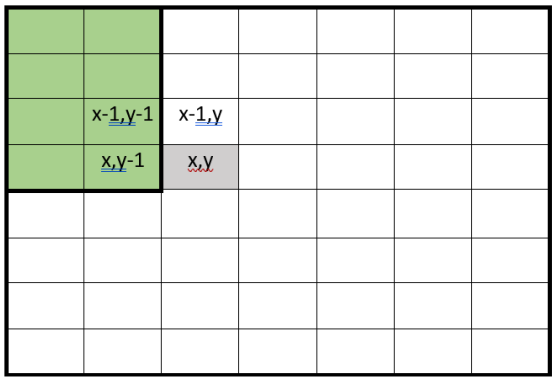
Έστω ότι θέλω να υπολογίσω την τιμή στον πίνακα αθροισμάτων(Summed Area Table) του στοιχείου που βρίσκεται στην θέση [x,y]:

- Το στοιχείο που βρίσκεται μία γραμμή πάνω από αυτό στην ίδια στήλη, δηλαδή στη θέση [x-1,y] περιέχει το άθροισμα όλων των στοιχείων που βρίσκονται πάνω αριστερά από αυτό, με τον εαυτό του(πορτοκαλί περιοχή).
- Το στοιχείο που βρίσκεται μία στήλη πριν από το [x,y] και στην ίδια γραμμή, δηλαδή στη θέση [x,y-1] περιέχει επίσης το άθροισμα όλων των στοιχείων που βρίσκονται πάνω και αριστερά από αυτό, με τον εαυτό του(πράσινη περιοχή).

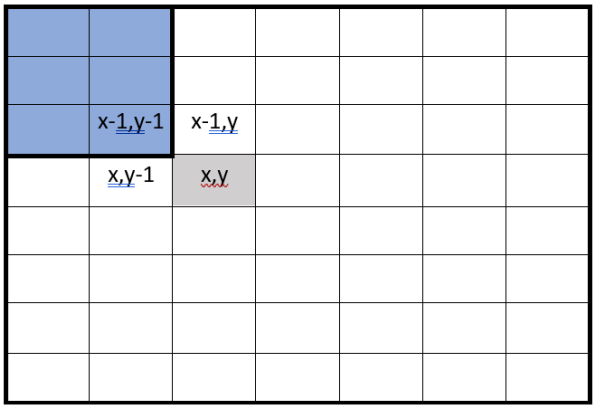
Πίνακες για την κατανόηση της επίλυσης:



Πίνακας 1:Summed Area Table



Πίνακας 2:Summed Area Table



Πίνακας 3: Summed Area Table

Για να υπολογίσουμε την τιμή του αθροίσματος στην θέση [x,y] θα πρέπει αφού προσθέσουμε τις τιμές των 2 γειτόνων που προαναφέρθηκαν να αφαιρέσουμε την τιμή του στοιχείου στη θέση [x-1,y-1] καθώς όπως φαίνεται η μπλε περιοχή αποτελεί κομμάτι και της πορτοκαλί αλλά και της πράσινης περιοχής .Τέλος προσθέτω και το στοιχείο που βρίσκεται στη θέση [x-1,y-1] της εικόνας μου. Τονίζω ότι έχω βάλει το -1 στις συντεταγμένες του στοιχείου της εικόνας καθώς έχω βάλει τους μετρητές του πίνακα αθροίσματος να ξεκινάνε μετά την πρώτη στήλη και την πρώτη γραμμή του. Πρακτικά αναφερόμαστε στο ίδιο στοιχείο απλά στον πίνακα αθροίσματος είναι μετατοπισμένο .

```
sum = Summed_Area_Table[x - 1, y] + Summed_Area_Table[x, y - 1] -
Summed_Area_Table[x - 1, y - 1] + image[x-1, y-1]
Summed_Area_Table[x, y] = sum
```

Με την παραπάνω διαδικασία έχω καταφέρει να βρω την τιμή του αθροίσματος του στοιχείου στη θέση [x,y] με τέτοιο τρόπο έτσι ώστε ο υπολογισμός μου να είναι γρήγορος και ανεξάρτητος του μεγέθους της υπο-περιοχής και της εικόνας.

7.Ολοκλήρωση της επίλυσης

Έχοντας υλοποιήσει όλες τις παραπάνω διεργασίες έχουμε φτάσει στην ολοκλήρωση της επίλυσης της εργασίας. Παρακάτω παρουσιάζεται η παραγόμενη έξοδος για τα πρώτα 7 παραδειγματικά. Παρατηρούμε ότι η αρίθμηση των περιοχών μπορεί να πάει από το 5-9 ,αυτό σημαίνει ότι οι περιοχές που δεν αναγράφονται αναφέρονται σε θόρυβο σύμφωνα με το κατώφλι που έχω ορίσει προηγουμένως.

```
--Region 1 --
Area(px): 803
Bounding Box Area: 1200
Mean gray level in bounding box: 35.87083333333333

--Region 2 --
Area(px): 710
Bounding Box Area: 1056
Mean gray level in bounding box: 27.255681818181817

--Region 3 --
Area(px): 1329
Bounding Box Area: 1740
Mean gray level in bounding box: 42.87068965517241

--Region 4 --
Area(px): 2214
Bounding Box Area: 3021
Mean gray level in bounding box: 33.55875537901357

--Region 5 --
Area(px): 1986
Bounding Box Area: 2520
Mean gray level in bounding box: 64.2297619047619

--Region 8 --
Area(px): 1905
Bounding Box Area: 2548
Mean gray level in bounding box: 48.777864992150704

--Region 9 --
Area(px): 1863
Bounding Box Area: 2800
Mean gray level in bounding box: 34.707857142857144
```

Εικόνα 9:Τελική έξοδος με είσοδο την Εικόνα 1

8.Σύγκριση αποτελεσμάτων στην περίπτωση που χρησιμοποιούμε την αρχική εικόνα χωρίς θόρυβο

Τρέχοντας τον κώδικα μου με είσοδο την αρχική εικόνα χωρίς θόρυβο, παρακάμπτοντας την διαδικασία του φιλτραρίσματος παίρνω αποτελέσματα κοντινά με αυτά της προηγούμενης υλοποίησης. Παρακάτω φαίνεται παραδειγματικά η έξοδος για τα πρώτα 6 κύτταρα.

```
--Region 1 --
Area(px): 842
Bounding Box Area: 1248
Mean gray level in bounding box: 36.38782051282051

--Region 2 --
Area(px): 736
Bounding Box Area: 1100
Mean gray level in bounding box: 27.694545454545455

--Region 3 --
Area(px): 1379
Bounding Box Area: 1740
Mean gray level in bounding box: 44.27528735632184

--Region 4 --
Area(px): 2213
Bounding Box Area: 3021
Mean gray level in bounding box: 33.62528963919232

--Region 5 --
Area(px): 1984
Bounding Box Area: 2520
Mean gray level in bounding box: 64.49285714285715

--Region 8 --
Area(px): 1909
Bounding Box Area: 2600
Mean gray level in bounding box: 48.27346153846154
```

Εικόνα 10:Τελική έξοδος με είσοδο την Εικόνα 2

Οι διαφορές είναι υπαρκτές. Οι περιοχές αντικειμένων που μετρήθηκαν άρα και οι ετικέτες που δημιουργήθηκαν δεν είναι οι ίδιες σε αριθμό, επίσης διαφέρουν ως προς τα pixel και μας δίνουν διαφορετικά αποτελέσματα. Αυτό συμβαίνει διότι η αρχική εικόνα είναι πιο καθαρή και επίσης είναι πιο έντονη η διαβάθμιση του γκρι ανάμεσα στα κύτταρα και το background. Παρατηρούμε ότι στην πρώτη περίπτωση συμπεριλαμβάνονται περισσότερα pixel στην περιοχή του κυττάρου και συνεπώς και του περιβάλλοντος κουτιού .Αυτό θεωρώ πως συμβαίνει καθώς παρόλο που η εικόνα έχει φιλτραριστεί και συνεπώς έχει απομακρυνθεί ο (περισσότερος) θόρυβος τύπου «αλατιού και πιπεριού», είναι πιο θολή σε σχέση με την εικόνα στην περίπτωση 2 και συνεπώς υπάρχει μικρότερη ακρίβεια στα όρια (περιγράμματα)των κυττάρων. Επίσης στην 2η περίπτωση εμφανίζονται περισσότερες περιοχές(συνολικά 50 έναντι 43 της 1ης περίπτωσης) οι οποίες αντιστοιχούν σε pixel της τελευταίας στήλης και γραμμής που στη φιλτραρισμένη εικόνα έχουν μηδενική τιμή.