

## ΕΡΓΑΣΙΑ 1

### ΜΑΘΗΜΑ : ΟΡΑΣΗ ΥΠΟΛΟΓΙΣΤΩΝ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: Ζαχάρη Βάια

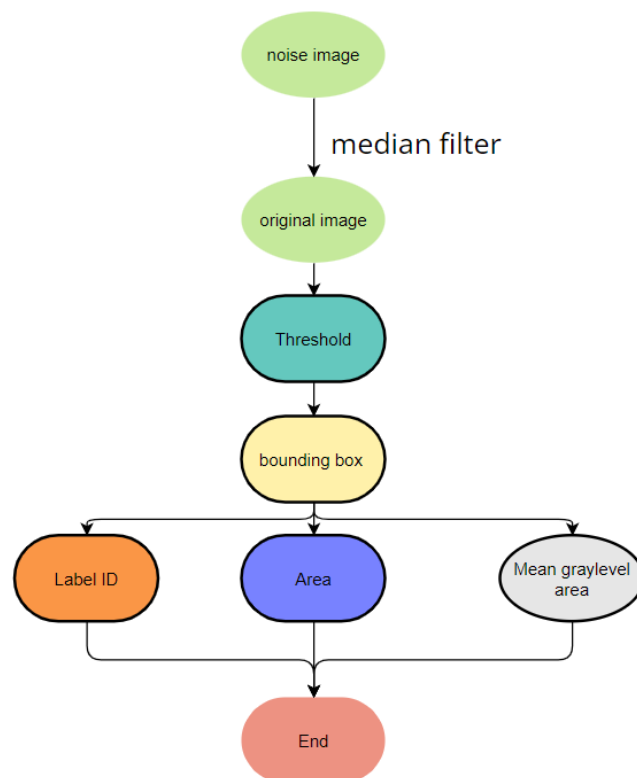
ΑΜ: 58161

#### ΕΙΣΑΓΩΓΗ:

Στη παρούσα εργασία καλούμαστε να αναλύσουμε τη δομή του περιεχομένου εικόνων μικροσκοπίου. Πιο συγκεκριμένα να ανιχνεύσουμε τα κύτταρα των εικόνων ,εφόσον έχει εφαρμοστεί στην εικόνα διαδικασία φιλτραρίσματος απόρριψης θορύβου (salt and pepper noise). Η τελική εικόνα θα πρέπει να πλησιάζει στην original εικόνα χωρίς θόρυβο και επιπλέον να διακριτοποιεί τα κύτταρα μεταξύ τους με τη βοήθεια ενός πλαισίου (bounding box). Για έναν πιο αναλυτικό διαχωρισμό θα εμφανίζεται για κάθε πλαίσιο ένας μοναδικός αριθμός (id), ο αριθμός των pixels που αντιστοιχούν στην περιοχή, καθώς και η μέση τιμή διαβάθμισης του γκρι της περιοχής.

#### ΜΕΘΟΔΟΛΟΓΙΑ:

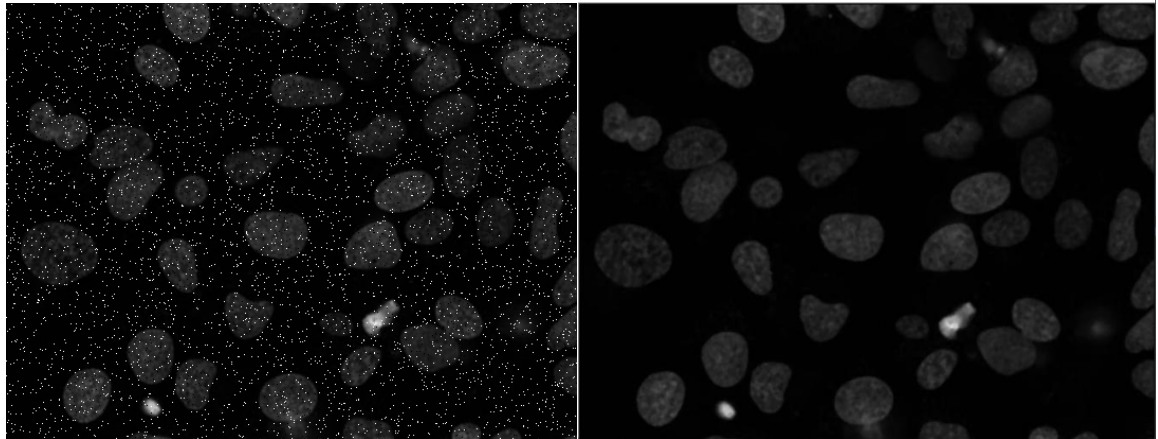
Παρακάτω παρουσιάζεται συνοπτικά η μεθοδολογία που ακολούθησα με τη μορφή block διαγράμματος.



Μεθοδολογία σε βήματα:

1) Στην αρχή φορτώνουμε την εικόνα ('1.png') έτσι ώστε να την εφαρμοστεί ένα μη γραμμικό φίλτρο , το median filter. Στη συγκεκριμένη υλοποίηση δεν χρησιμοποιήθηκε η έτοιμη συνάρτηση της OpenCV ('medianBlur()') , αλλά μία σειρά εντολών και συναρτήσεων που εξάγουν το ίδιο αποτέλεσμα με ίδια λογική. Αναφορικά με το φίλτρο ,

αυτό χρησιμοποιεί την μεσαία τιμή ενός πίνακα τιμών kernel μεγέθους 3 επί 3 που παράγεται επαναληπτικά από την αρχική εικόνα ,τοποθετείται στο κέντρο του kernel και στη συνέχεια στο αντίστοιχο σημείο της νέας φιλτραρισμένης εικόνας ίδιου μεγέθους με την αρχική. Έτσι η παραγόμενη εικόνα έχει απομακρύνει τον θόρυβο καθώς αποβάλλονται οι τιμές που αποκλίνουν από τη μεσαία τιμή και επομένως γίνεται πιο smooth, όπως φαίνεται παρακάτω.



2) Στη συνέχεια δουλεύουμε με την φιλτραρισμένη εικόνα. Αφού την φορτώσουμε την μεταφέρουμε σε γκρι κλίμακα, έτσι ώστε να είμαστε πιο ακριβείς στη διακριτοποίηση των κυττάρων. Μάλιστα εφαρμόζουμε ένα threshold με βάση το οποίο η εικόνα θα μετατραπεί σε δυαδική και επομένως θα διαχωρίσουμε το παρασκήνιο από το προσκήνιο, δηλαδή τα αντικείμενα που μας ενδιαφέρουν . Για τη συγκεκριμένη εικόνα επέλεξα ως τιμή threshold το 15 , οπότε αν ένα pixel έχει συνιστώσα του γκρι μεγαλύτερη από 15 γίνεται 255 (binary 1) ,διαφορετικά αν είναι μικρότερη γίνεται 0 (binary 0).

```
img = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
ret, binary = cv2.threshold(img,15,255, cv2.THRESH_BINARY)
```



3) Έπειτα έχουμε ως στόχο να εξάγουμε ορισμένα χαρακτηριστικά της εικόνας ,όπως η επιφάνεια του κυττάρου, που ορίζεται ως ο αριθμός των εικονοστοιχείων που ανήκουν σε αυτό , η επιφάνεια του περιβάλλοντος κουτιού (bounding box) του κυττάρου, ως αριθμός εικονοστοιχείων και η μέση τιμή διαβάθμισης του γκρι των

εικονοστοιχείων που περιέχονται στο περιβάλλον κουτιού. Για να μπορέσουμε να τα υπολογίσουμε χρειαζόμαστε παραμέτρους που θα μας τις δώσει η συνάρτηση της OpenCV `connectedComponentsWithStars()`. Η συνάρτηση επιστρέφει 4 παραμέτρους:

- `total_labels`: πόσα αντικείμενα βρίσκονται στην εικόνα
- `labels_id`: κάθε εικονοστοιχείο έχει το `label` του στοιχείου που ανήκει
- `values`: δίνει 5 νούμερα για κάθε εικονοστοιχείο (όπως φαίνεται στο βήμα 4)
- `centroid`: δίνει τα κέντρα των στοιχείων

4) -Για την εύρεση της περιοχής `area` για κάθε κύτταρο χρησιμοποιήσα επαναληπτικά την συνάρτηση :

```
area[i] = values[i, cv2.CC_STAT_AREA]
```

-Για την δημιουργία του `bounding box` υπολόγισα τις συντεταγμένες της κάθε περιοχής με τον παρακάτω τρόπο:

```
x1 = values[i, cv2.CC_STAT_LEFT]
y1 = values[i, cv2.CC_STAT_TOP]
w = values[i, cv2.CC_STAT_WIDTH]
h = values[i, cv2.CC_STAT_HEIGHT]
(X, Y) = centroid[i]
```

όπου:

**`cv2.CC_STAT_LEFT`** : Η πιο αριστερή συντεταγμένη (x) που είναι η συμπερίληψη αρχή του πλαισίου οριοθέτησης στην οριζόντια διεύθυνση.

**`cv2.CC_STAT_TOP`**: Η ανώτατη συντεταγμένη (y) που είναι η αρχή που περιλαμβάνει το πλαίσιο οριοθέτησης στην κατακόρυφη διεύθυνση.

**`cv2.CC_STAT_WIDTH`**: Το οριζόντιο μέγεθος του πλαισίου οριοθέτησης.

**`cv2.CC_STAT_HEIGHT`**: Το κατακόρυφο μέγεθος του πλαισίου οριοθέτησης.

**`cv2.CC_STAT_AREA`** : Η συνολική επιφάνεια (σε pixel) του συνδεδεμένου στοιχείου.

Να σημειωθεί ότι έγινε ένας διαχωρισμός των περιοχών με βάση ένα `threshold` που απομακρύνει τις μικρές περιοχές που δεν αποτελούν κύτταρα, αλλά μικρές ομάδες με τιμή 1, για ένα πιο ακριβές αποτέλεσμα.

-Η περιοχή του `bounding box` υπολογίζεται ως εξής:

```
BoundingBoxArea[i] = w*h
```

-Τη μέση τιμή διαβάθμισης του γκρι κάθε στοιχείου θα την βρούμε μέσω της εικόνας σε grayscale και με τη βοήθεια όχι της έτοιμης συνάρτησης `integral` αλλά μέσω μιας άλλης υλοποίησης, όπως έγινε και προηγουμένως. Συγκεκριμένα δημιουργούμε ένα πίνακα αθροισμάτων της αρχικής εικόνας, όπου κάθε σημείο της νέας εικόνας θα περιέχει το άθροισμα των τιμών των εικονοστοιχείων της αρχικής εικόνας που βρίσκονται πάνω και αριστερά από αυτό. Πρακτικά αν υποθέσουμε ότι η αρχική εικόνα είναι η `img` και η εικόνα αθροισμάτων είναι η `img2`, τότε σε ένα σημείο με συντεταγμένες (r,c) το σημείο (r,c) της `img2` υπολογίζεται ως εξής:

```
sumX = img2[r-1,c] + img2[r,c-1] - img2[r-1,c-1] + img[r,c]
img2[r, c] = sumX
```

Επομένως έχοντας κατασκευάσει τον πίνακα αθροισμάτων εμείς μπορούμε να υπολογίσουμε τη μέση τιμή μία περιοχής έχοντας δύο σημεία, το πάνω αριστερά και το κάτω δεξιά, που ορίζουν τη περιοχή.

Αυτα τα δύο σημεία με βάση το `bounding box` είναι:

```
pt1 = (x1,y1)
pt2 = (x1+ w,y1+h)
```

Το άθροισμα που θα εξάγει ο πίνακας αθροισμάτων θα διαιρεθεί με το πλήθος των εικονοστοιχείων της περιοχής για να πάρουμε τη μέση τιμή.

```
Mean_graylevel_value[i] = (CalcAreas(img2,x1,y1,w+x1,h+y1) / bbox)
```

5) Για την δημιουργία του πλαισίου του bounding box χρησιμοποιήσαμε την συνάρτηση της OpenCV cv2.rectangle() με παραμέτρους την εικόνα σε grayscale ,τα σημεία pt1,pt2 και τις συνιστώσες RGB για κόκκινο χρώμα.

```
output = cv2.rectangle(new_image, (x1,y1), (x1+w,y1+h), (0,0,255), 2)
```

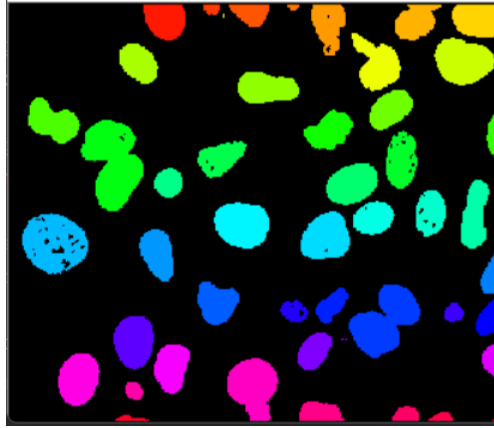
6) Επιπλέον για να εμφανίζεται μαζί με το bounding box ένας μοναδικός αυξάνων αριθμός εκμεταλλευτήκαμε την έτοιμη συνάρτηση της OpenCV cv2.putText() με ορίσματα την εικόνα σε grayscale , την θέση του text (εδώ τα κέντρα των περιοχών) ,το text που θα εμφανίζεται (ο αύξων αριθμός) , τη γραμματοσειρά ,το μέγεθος των γραμμάτων , το χρώμα τους (κόκκινο) και το πάχος τους.

```
new_image =
cv2.putText(img=new_image,text="{}".format(i),org=(int(X),int(Y)),fontFace=cv2.FONT_HERSHEY_SIMPLEX,fontScale=0.5,color=(0, 0, 255),thickness = 2)
```

Συνεπώς το αποτέλεσμα που θα δούμε στην αρχική εικόνα είναι:



7) Σαν επιπρόσθετο βήμα δημιουργούμε μία εικόνα που μας δείχνει τα κύτταρα με διαφορετικά χρώματα μέσω της συνάρτησης imshow\_components(). Η συνάρτηση χρωματίζει κάθε περιοχή με το ίδιο label id με με ένα ξεχωριστό χρώμα.



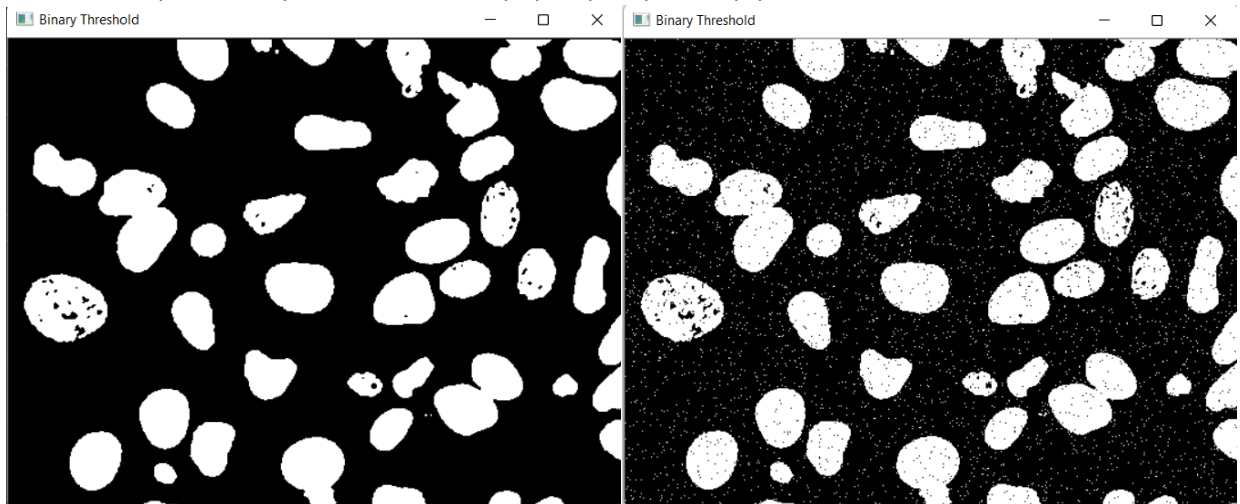
8) Τέλος αποθηκεύουμε την εικόνα με τα bounding boxes.

```
new_image2 = cv2.imwrite(r'RESULT.png',new_image)
```

### ΠΕΡΙΠΤΩΣΗ ΜΕ ΘΟΡΥΒΟ

Στην περίπτωση που έχουμε θόρυβο τα αποτελέσματα που παράγονται διαφέρουν αρκετά από αυτά χωρίς θόρυβο.

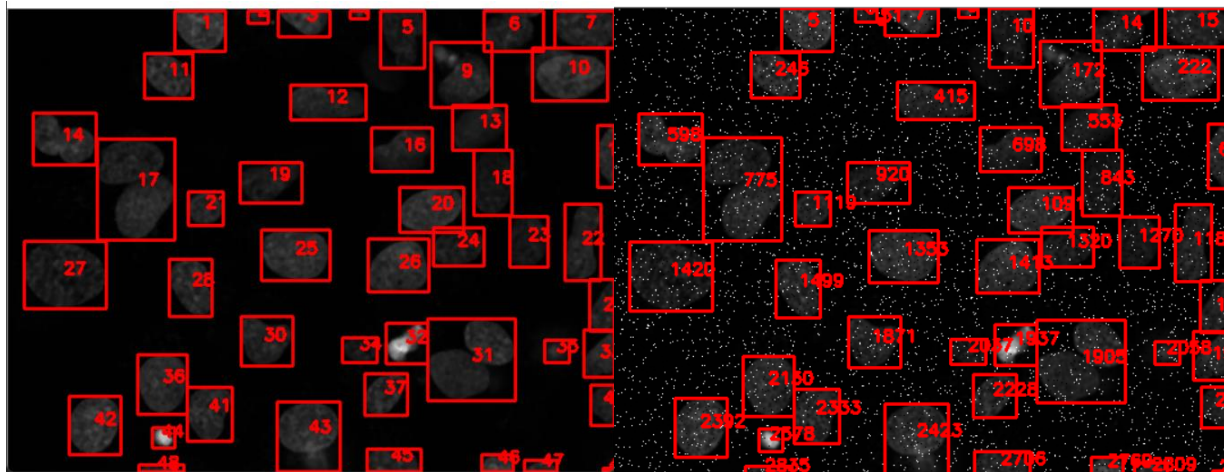
-Αρχικά η δυαδική εικόνα παρουσιάζει οπές στο εσωτερικό των κυττάρων, καθώς ο θόρυβος με τιμή γκρι 0 έχει τιμή κάτω από το threshold και άρα παίρνει τη τιμή 0. Αντίθετα παρατηρούμε τη τιμή 1 σε σημεία που δεν υπάρχει κύτταρο εξαιτίας του θορύβου με τιμή του γκρι 255.



Χωρίς θόρυβο

Με θόρυβο

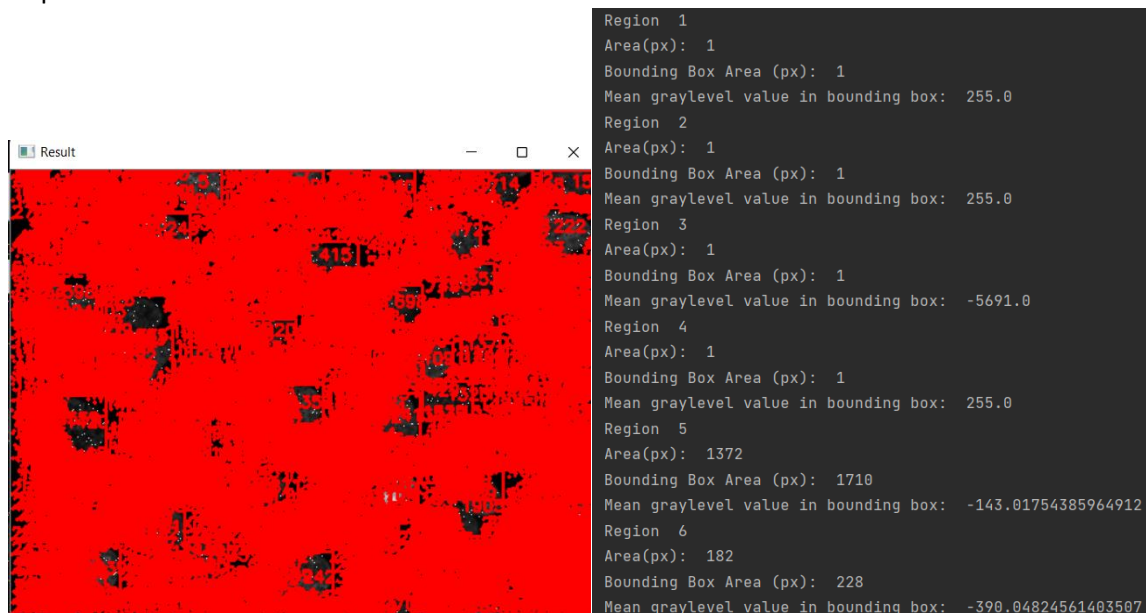
-Επιπλέον η εικόνα που παράγεται που ανιχνεύει τα κύτταρα περιέχει τόσο σφάλματα στην ανίχνευση και στην πλαίσισή τους , όσο και στην αρίθμηση.



Χωρίς θόρυβο

Με θόρυβο

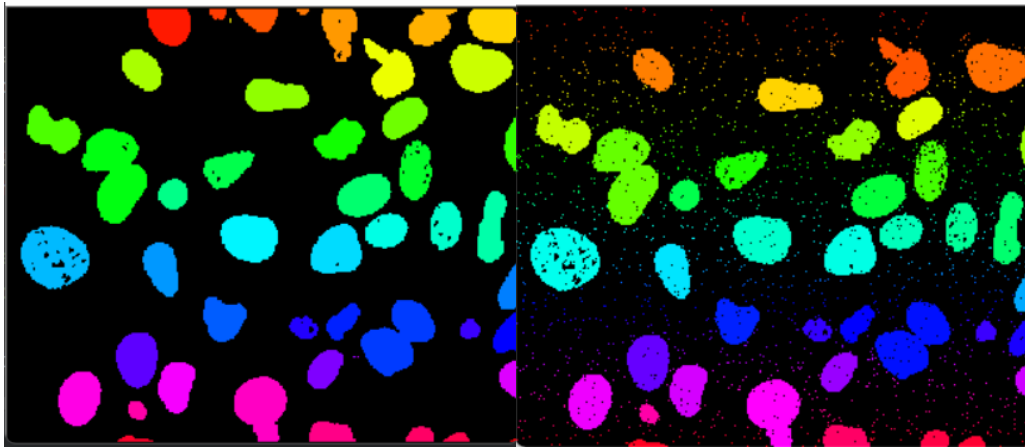
Αυτό συμβαίνει διότι πλέον ως κύτταρα ανιχνεύονται οι κόκκοι από τον θόρυβο. Αν αφαιρούσαμε το threshold που ορίζει για ποιες περιοχές θα δημιουργούνται πλαίσια το αποτέλεσμα θα ήταν το παρακάτω.



Βλέπουμε ότι περιοχές αποτελούνται από ένα μόνο εικονοστοιχείο με μέση διαβάθμιση του γκρι 255, στοιχεία που δηλώνουν ότι αυτό το πλαίσιο προήλθε από ένα κόκκο θορύβου λευκού χρώματος.

-Τέλος η εικόνα που δείχνει τα κύτταρα με διαφορετικά χρώματα ανάλογα το label id τους , παρουσιάζει περισσότερες διαβαθμίσεις των χρωμάτων στα εικονοστοιχεία που αποτελούν το θόρυβο, αλλά και σε αυτά των κυττάρων. Επίσης παρατηρούμε αστοχίες ανίχνευσης των κυττάρων, γεγονός που δεν συνέβαινε στην binary εικόνα.





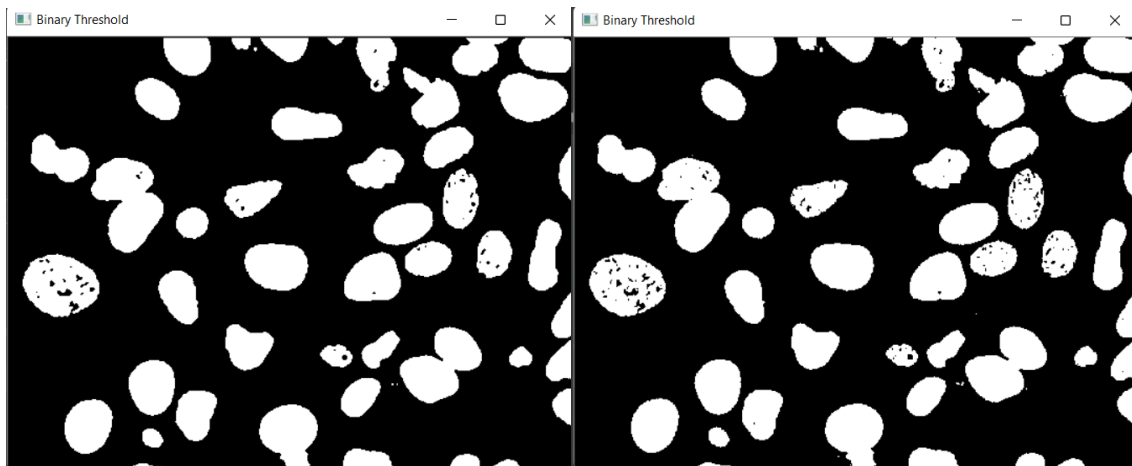
Χωρίς θόρυβο

Με θόρυβο

### Στην περίπτωση της original εικόνας

Αν δοκιμάσουμε να εφαρμόσουμε την original εικόνα στην παραπάνω υλοποίηση και συγκρίνουμε με τα αποτελέσματα με της αποθορυβοποιημένης εικόνας τότε θα παρατηρήσουμε τις εξής διαφορές:

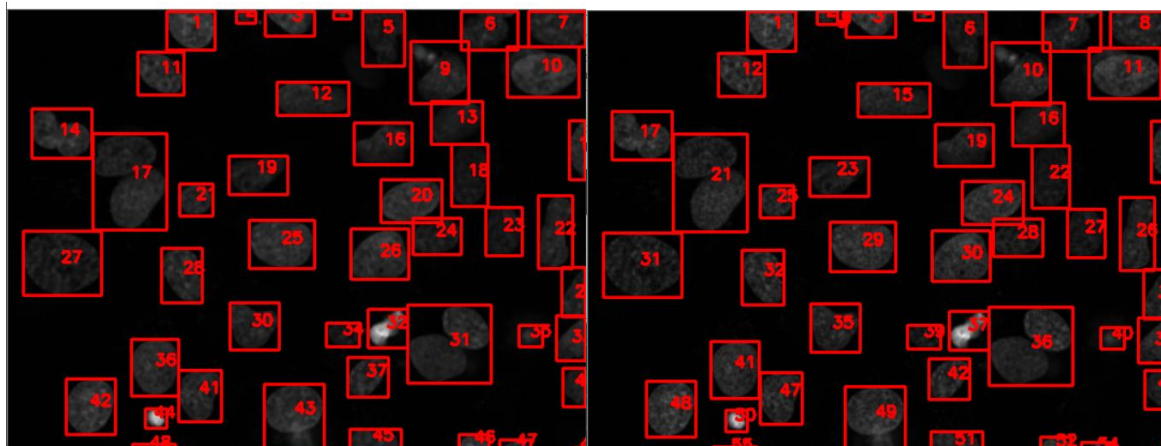
-Στη δυαδική εικόνα:



Αποθορυβοποιημένη εικόνα

Original εικόνα

-Στην εικόνα με τα bounding boxes:



Αποθορυβοποιημένη εικόνα

Original εικόνα

Στην κονσόλα θα παρατηρήσουμε ότι ανιχνεύθηκαν παραπάνω κύτταρα στην original εικόνα.

## ΠΑΡΑΤΗΡΗΣΕΙΣ-ΣΥΜΠΕΡΑΣΜΑΤΑ

- 1) Για το φίλτρο median επιλέγουμε τη μεσαία τιμή κι όχι τη μέση τιμή , διότι η μέση τιμή επηρεάζεται αρνητικά τόσο από τις πολύ μεγάλες τιμές όσο και από τις πολύ μικρές.
- 2) Στην περίπτωση που δεν εφαρμόζουμε φίλτρο τα αποτελέσματα είναι ανακριβή και αποκλίνουν κατα πολύ από την πραγματικότητα.
- 3) Παρατηρούμε ότι κατά την διακριτοποίηση των κυττάρων υπάρχουν σφάλματα. Δηλαδή σε ένα bounding box υπάρχουν δύο κύτταρα κι όχι ένα. Αυτό συμβαίνει λόγω της συνάρτησης των συνδεδεμένων αντικειμένων όπου τα δύο αντικείμενα αλληλοκαλύπτονται. Το πρόβλημα αυτό μπορεί να αντιμετωπιστεί με τη μέθοδο μορφολογίας opening . Πρώτα δηλαδή θα εφαρμοστεί διάβρωση και ύστερα διαστολή του αντικειμένου στην δυαδική εικόνα. Με τη μέθοδο αυτή ωστόσο έχουμε ένα μειονέκτημα , θα χαθεί το πραγματικό μέγεθος του κάθε κυττάρου και άρα θα επηρεάσει τα χαρακτηριστικά κάθε περιοχής.
- 4) Ιδιαίτερη σημασία έχει το threshold που θα ορίσει την δυαδική εικόνα , το οποίο καθορίζεται τυχαία μετά από ένα εύρος τιμών των περιοχών της συγκεκριμένης εικόνας κυττάρων.
- 5) Όταν εφαρμόζουμε την υλόπιση σε μία εικόνα που έχουμε χρησιμοποιήσει το φίλτρο και σε στην original της παρατηρούμε ότι οι διαφορές εμφανίζονται κυρίως στη μέση τιμή της διαβάθμισης του γκρι και στην αρίθμηση των κυττάρων. Αυτό οφείλεται στην επιλογή του threshold καθώς η original είναι πιο καθαρή και άρα οι τιμές της πιο έντονες .

Συμπερασματικά η διαδικασία ανίχνευσης των κυττάρων και γενικότερα αντικειμένων της ίδιας ομάδος φαίνεται ιδιαίτερη χρήσιμη τόσο στην ιατρική όσο και σε άλλους τομείς έρευνας , καθώς το ανθρώπινο μάτι είναι πιο επιρρεπές σε σφάλματα και σε ιδιαίτερες συνθήκες η ανίχνευση γίνεται δύσκολη από τη φύση της. Παρομοίως τα εξαγόμενα χαρακτηριστικά είναι χρήσιμα για πιο λεπτομερή αποτελέσματα σε έρευνες. Στη συγκεκριμένη εργασία, για παράδειγμα, παρατηρούμε ότι υπάρχουν πολλών ειδών κύτταρα που διαφέρουν στο σχήμα. Με βάση όμως την βιολογία γνωρίζουμε ότι το μέγεθος των κυττάρων είναι σημαντικό, αφού αντιπροσωπεύει την ικανότητα τους για εξελικτική προσαρμογή σε διαφορετικά περιβάλλοντα. Συνεπώς μπορούμε εύκολα να καταλήξουμε πως η διαδικασία αυτή αυτοματοποιεί διεργασίες όπως οι παραπάνω και εξάγει πιο ακριβή αποτελέσματα.