
Ψηφιακή Επεξεργασία Εικόνας:
Εργασία 1: Βελτίωση εικόνων μέσω
εξισορρόπησης ιστογράμματος

Αφροδίτη Λάτσκου

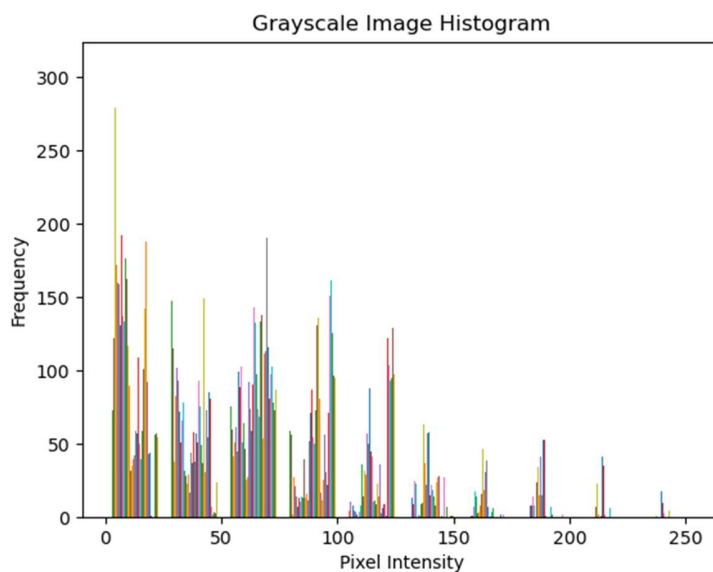
Απρίλιος 2024

Σύνοψη εργασίας:

Ο σκοπός της εργασίας είναι να εφαρμόσουμε εξισορρόπηση ιστογράμματος, συμβατική και προσαρμοστική, σε μια εικόνα που έχουμε ως είσοδο αφού την μετατρέψουμε αρχικά σε grayscale. Έτσι η αρχική μας εικόνα είναι η εξής:



Παρατηρούμε ήδη ότι η εικόνα έχει πολλές σκούρες αποχρώσεις ενώ ελάχιστες πιο κοντά στο λευκό. Στην παρούσα κατάσταση πολλές από τις λεπτομέρειες της εικόνας χάνονται και οι ακμές των αντικειμένων που βρίσκονται σε αυτή είναι σε γενικές γραμμές αδιάκριτες, εκτός και αν το αντικείμενο είναι πιο φωτεινό από αυτά που το περιτριγυρίζουν.



Αυτό επιβεβαιώνεται και από το ιστόγραμμα της εικόνας, όπως φαίνεται δίπλα.

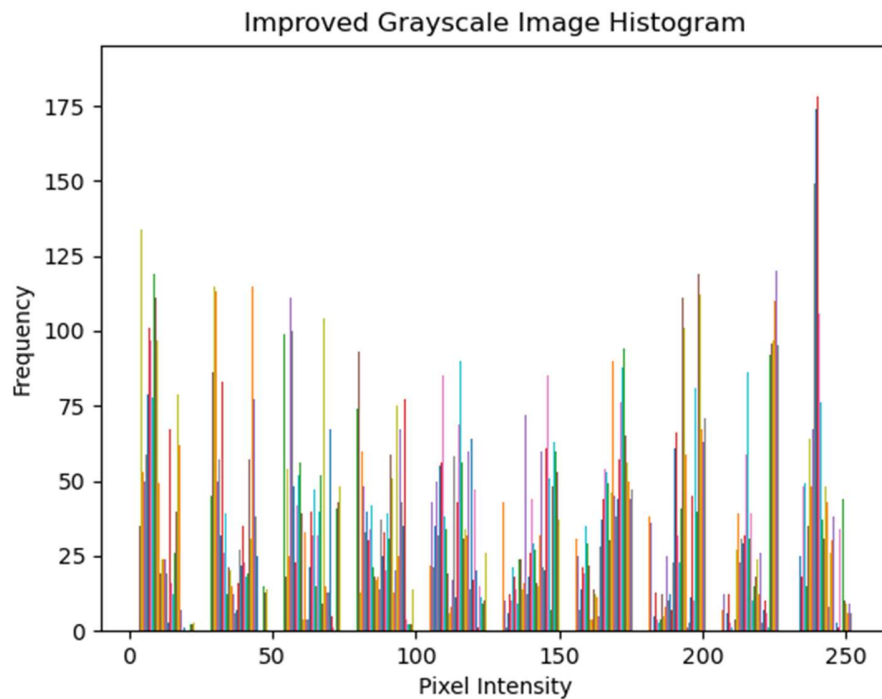
Συμβατική Εξισορρόπηση Ιστογράμματος

Για να επιλυθεί το άνωθεν πρόβλημα, υλοποιούμε την συνάρτηση `get_equalization_transform_of_img()`, η οποία παίρνει σαν είσοδο τον πίνακα με τις τιμές της εικόνας και έχει ως έξοδο έναν μονοδιάστατο πίνακα με τους μετασχηματισμούς για κάθε πιθανή τιμή του luminance. Η συνάρτηση, αφού υπολογίσει το ιστόγραμμα της εικόνας εισόδου, βρίσκει την πιθανότητα εμφάνισης κάθε τιμής και έπειτα την αθροιστική συνάρτηση κατανομής (v_k). Ο τύπος του μετασχηματισμού ορίζεται ως: $y_k = \text{round}(\frac{v_k - v_0}{1 - v_0}(L - 1))$.

Έπειτα δημιουργούμε και την συνάρτηση `perform_global_hist_equalization()`. Αυτή έχει ως είσοδο τον πίνακα που περιγράφει την αρχική grayscale εικόνα και ως έξοδο τον πίνακα που περιγράφει την εξισορροπημένη εικόνα. Η συνάρτηση απλά χρησιμοποιεί την προηγούμενη για να βρει το διάνυσμα μετατροπής και αντιστοιχίζει κάθε pixel στην νέα τιμή του.

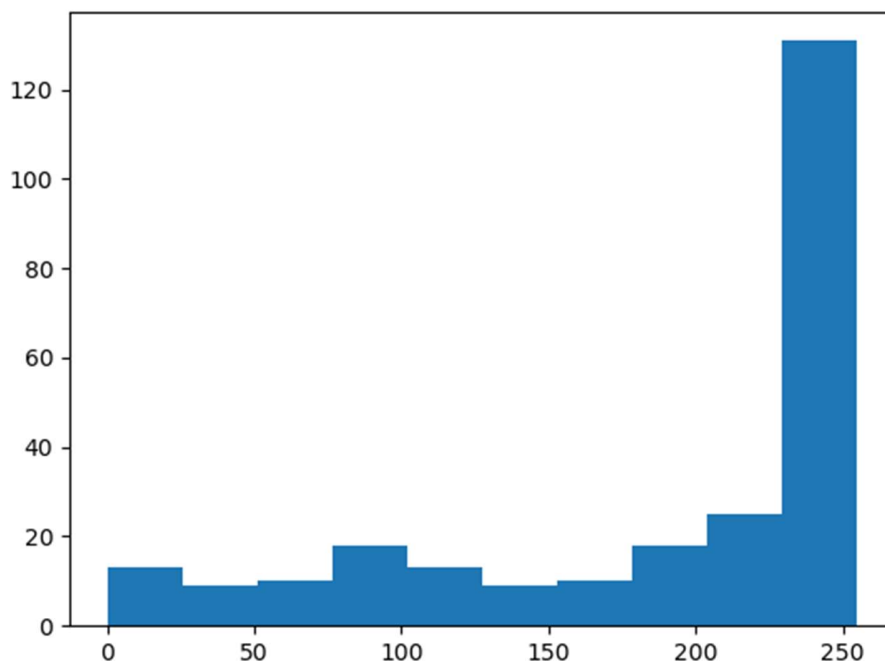
Το αποτέλεσμα που παίρνουμε είναι το εξής.





Όπως βλέπουμε, τα χρώματα στην εικόνα έχουν γίνει πιο φωτεινά και τα αντικείμενα είναι πιο ευδιάκριτα. Στο ιστόγραμμα φαίνεται πως τα peaks έχουν μικρότερη διαφορά μεταξύ τους και έχουν απλωθεί σε όλη την έκταση των τιμών, δίνοντας ένα πολύ πιο ισορροπημένο αποτέλεσμα, έχουμε δηλαδή μεγαλύτερο πλήθος στάθμεων με μη μηδενικό περιεχόμενο. Ωστόσο τα αντικείμενα που ήταν ήδη φωτεινά (όπως το άγαλμα) τώρα έχουν χάσει πολύ λεπτομέρεια.

Ταυτόχρονα, αν κάνουμε plot το διάνυσμα του transformation παίρνουμε το εξής, από όπου διαπιστώνουμε ότι το πλήθος στάθμεων της τελικής εικόνας θα είναι μικρότερο:



Προσαρμοστική Εξισορρόπηση Ιστογράμματος

Με σκοπό η εξισορρόπηση του ιστογράμματος να λαμβάνει υπόψη την ιδιαίτερη κατανομή της τοπικής ενός σημείου πληροφορίας, ώστε να έχουμε καλύτερη λεπτομέρεια ακόμα και στα αντικείμενα που είναι πολύ φωτεινά στην προκειμένη εικόνα, υλοποιούμε την προσαρμοστική εξισορρόπηση εικόνας. Η μέθοδος λειτουργεί αρχικά χωρίζοντας την εικόνα σε περιοχές και για κάθε περιοχή υπολογίζει το ιστόγραμμα και το transformation vector και έπειτα χτίζει την τελική εικόνα χρησιμοποιώντας μια κατηγοριοποίηση που θα δούμε αργότερα.

Για αρχή υλοποιούμε την συνάρτηση `calculate_eq_transformations_of_regions()`, η οποία έχει ως είσοδο τον πίνακα της αρχικής grayscale εικόνας καθώς και το επιθυμητό ύψος και πλάτος κάθε περιοχής και έχει ως έξοδο ένα dictionary που έχει την έκφραση κάθε περιοχής ως κλειδί και το transformation vector του ως τιμή. Ως έκφραση κάθε περιοχής έχει επιλεχθεί το πρώτο pixel της πρώτης σειράς της περιοχής.

Να πούμε εδώ ότι ο συγκεκριμένος κώδικας, μετά από διευκρίνηση του διδάσκοντα, προϋποθέτει ότι **τα μεγέθη του ύψους και του πλάτους των περιοχών διαιρούν τα αντίστοιχα μεγέθη της εικόνας τέλεια**, καθώς σε αντίθετη περίπτωση αυξάνεται η πολυπλοκότητα του κώδικα και η παρούσα υλοποίηση δε θα λειτουργήσει. Επομένως το ύψος και πλάτος των περιοχών στη συγκεκριμένη εικόνα θα επιλεχθούν ως 36 και 48.

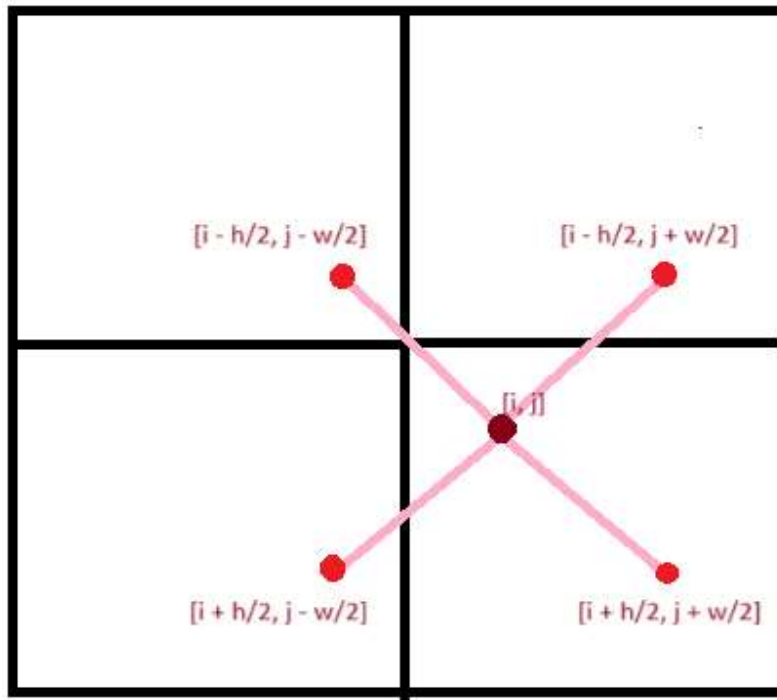
Η συνάρτηση υπολογίζει τον αριθμό των περιοχών σε κάθε σειρά και στήλη και έπειτα για κάθε περιοχή βρίσκει όλα τα pixel που ανήκουν σε αυτή πολλαπλασιάζοντας κάθε φορά τον αριθμό της περιοχής που βρισκόμαστε με τα `region_len_h` και `region_len_w`, βρίσκοντας έτσι το πρώτο pixel της και το τελευταίο πολλαπλασιάζοντας τα `region_len_h` και `region_len_w` με τον αριθμό της επόμενης περιοχής, αφαιρώντας 1. Έπειτα καλεί την `get_equalization_transform_of_img()` και τα αποτελέσματα τα αποθηκεύει στο dictionary.

Στη συνέχεια υλοποιούμε την συνάρτηση `perform_adaptive_hist_equalization()`, η οποία έχει ίδιες μεταβλητές εισόδου με την προηγούμενη και ως έξοδο έχει την τελική εικόνα που έχει υποστεί προσαρμοστική εξισορρόπηση ιστογράμματος. Για αρχή καλεί την προηγούμενη συνάρτηση. Τώρα πρέπει να κατηγοριοποιήσει κάθε pixel σε δύο ομάδες: είτε στο εξωτερικό περίβλημα είτε στο εσωτερικό μέρος.

- Το εξωτερικό περίβλημα ορίζεται ως όλα τα pixel που βρίσκονται στην περιοχή $[0, \text{region_len_h}/2] \cup [\text{max_height} - \text{region_len_h}/2, \text{max_height}]$ στον άξονα y και $[0, \text{region_len_w}/2] \cup [\text{max_width} - \text{region_len_w}/2, \text{max_width}]$ στον άξονα x.
- Το εσωτερικό περίβλημα είναι όλα τα υπόλοιπα pixel.

Ο τρόπος με τον οποίο ανιχνεύεται η συγκεκριμένη περιοχή στην οποία ανήκει το pixel είναι ότι ανιχνεύει τον αριθμό της περιοχής σε κάθε κατεύθυνση και τον πολλαπλασιάζει με τα `region_len_h` και `region_len_w`. Συγκεκριμένα, διαιρεί τον δείκτη i και j με τα `region_len_h` και `region_len_w` κάθε φορά και το στρογγυλοποιεί προς τα κάτω, παίρνοντας τον ακέραιο ο οποίος είναι και ο αριθμός της περιοχής στην οποία ανήκει. Πχ. για το pixel [49, 99] όπου $49/36=1.36$ και $99/48=2.06$ η περιοχή που θα ανιχνευτεί είναι η 2^η από πάνω και 3^η από αριστερά και το python tuple που την εκφράζει είναι το [36, 96]

Αν η περιοχή στην οποία ανήκει το pixel είναι στο inner part τότε χρησιμοποιείται ο πάνω τρόπος για να βρεθούν οι περιοχές και τα transformation vectors των κέντρων που περιβάλλουν αυτό το pixel. Αυτό γίνεται προσθέτοντας ή αφαιρώντας τα μισά των μεγεθών των περιοχών από αυτά τα μεγέθη.



Έχουμε δηλαδή 4 νέα pixels των οποίων βρίσκουμε τις περιοχές, τα transformation vectors αυτών αλλά και τα κέντρα αυτών συγκεκριμένα για το πάνω αριστερά και κάτω δεξιά, καθώς αυτά θα χρησιμοποιηθούν στον τύπο αργότερα.

Τελικά, η νέα τιμή για το συγκεκριμένο pixel υπολογίζεται μέσω της εξίσωσης:

$$y = (1 - a)(1 - b)T_{-,-}(x) + (1 - a)bT_{+,-}(x) + a(1 - b)T_{-,+}(x) + abT_{+,+}(x)$$

όπου:

$$a = \frac{w_p - w_-}{w_+ - w_-}$$

$$b = \frac{(h_p - h_-)}{h_+ - h_-}$$

με τα $[h_p, w_p]$, $[h_-, w_-]$, $[h_+, w_+]$ να είναι οι συντεταγμένες του pixel και των κέντρων που προαναφέρθηκαν αντίστοιχα.

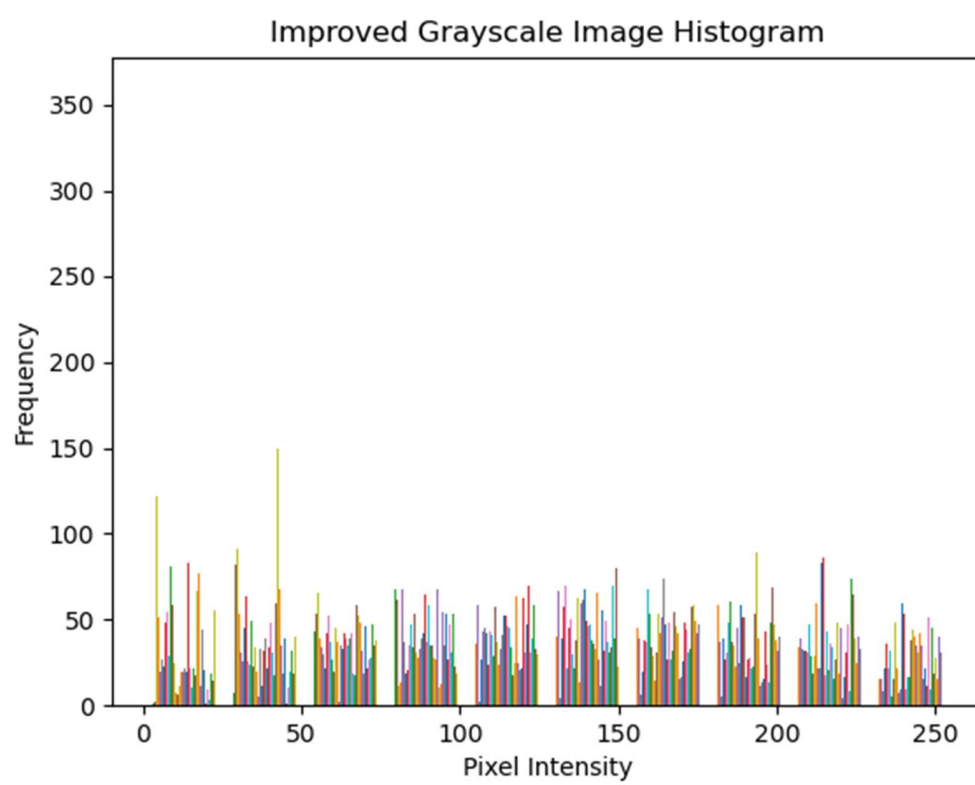
Ο λόγος που κάνουμε την παραπάνω διαδικασία αντί για να λαμβάνουμε υπόψη μόνο το transformation vector της περιοχής του ίδιου του pixel είναι επειδή σε αυτή την περίπτωση, παρότι θα είχαμε και πάλι εξισορρόπηση του ιστογράμματος, θα είχαμε ασυνέχειες στα όρια των περιοχών και η εικόνα θα φαινόταν έτσι:



Ενώ, χρησιμοποιώντας τον τύπο παίρνουμε την εξής εικόνα:



Με το ιστόγραμμα:



Για το ιστόγραμμα παρατηρούμε πως πράγματι, έχει εξισορροπηθεί και οι στάθμες κατανέμονται πολύ πιο ομοιόμορφα.

Στην εικόνα βλέπουμε πως τα αντικείμενα φαίνονται όλα πολύ πιο καθαρά, με όλες τις λεπτομέρειες τους και δεν υπάρχει πλέον το πρόβλημα που προκύπτει από το global equalization. Ωστόσο, ένα βασικό μειονέκτημα της προσαρμοστικής εξισορρόπησης είναι πως στις περιοχές που παρουσιάζουν μεγαλύτερη συνέχεια στις τιμές του luminance (όπως για παράδειγμα στο ντουλάπι πάνω δεξιά) εμφανίζεται έντονος θόρυβος.