



Εργασία 1: Βελτίωση εικόνων μέσω εξισορρόπησης ιστογράμματος

Παλάσκα Αιμιλία 10453 (aimiliarm@ece.auth.gr)

Επιβλέποντες: Αναστάσιος Ντελόπουλος, Αλέτρας Δημήτριος

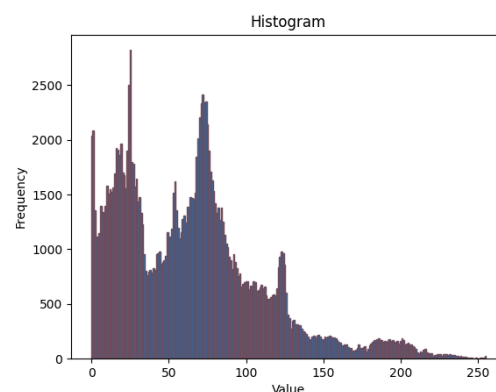
Λειτουργία συναρτήσεων

Τα προγράμματα μπορούν να κληθούν είτε από κάποιο IDE π.χ. VSCode είτε με την εντολή τερματικού `/bin/python3 demo.py`. Να σημειωθεί πως απαιτούνται μερικά δευτερόλεπτα για να ολοκληρωθεί η αποθήκευση όλων των εικόνων και των αντίστοιχων ιστογραμμάτων τους. Επίσης, ο κώδικας μπορεί να βρεθεί και στο [GitHub](#). Στο πρόγραμμα συμπεριλαμβάνονται οι παρακάτω συναρτήσεις με την αντίστοιχη επεξήγηση στα σχόλια της κάθε μιας:

- ❖ `my_hist(data, filename='hist.png')`: υλοποίηση και αποθήκευση ιστογράμματος
- ❖ `plot_vector(vector, filename='vector.png')`: συνάρτηση γραφικής αναπαράστασης και αποθήκευσης διανύσματος
- ❖ `get_equalization_transform_of_img(img_array)`: υπολογισμός του μετασχηματισμού εξισορρόπησης εικόνας
- ❖ `perform_global_hist_equalization(img_array)`: υλοποίηση της συμβατικής μεθόδου εξισορρόπησης ιστογράμματος
- ❖ `calculate_eq_transformations_of_regions(img_array, region_len_h, region_len_w)`: αντιστοίχιση contextual regions με μετασχηματισμούς εξισορρόπησης μέσω λεξικού (dictionary)
- ❖ `perform_adaptive_hist_equalization(img_array, region_len_h, region_len_w)`: υλοποίησης της προσαρμοστικής μεθόδου εξισορρόπησης ιστογράμματος

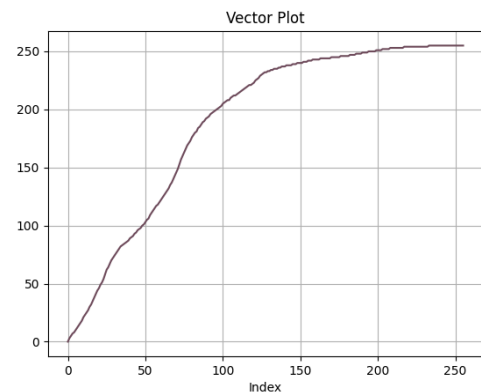
Παρουσίαση και σχολιασμός αποτελεσμάτων

Αρχικά φαίνεται παρακάτω η δοσμένη εικόνα, η οποία έχει μετατραπεί σε μονόχρωμη (grayscale) με τον τρόπο που υποδεικνύει η εκφώνηση, όπως επίσης και το ιστόγραμμα της. Με μια πρώτη ματιά παρατηρούμε ότι η εικόνα είναι αρκετή σκοτεινή, το οποίο επιβεβαιώνεται από την μεγάλη συχνότητα χαμηλών τιμών (< 100) στο ιστόγραμμα. Επομένως, περιμένουμε η εξισορρόπηση να αυξήσει αισθητά την φωτεινότητα.

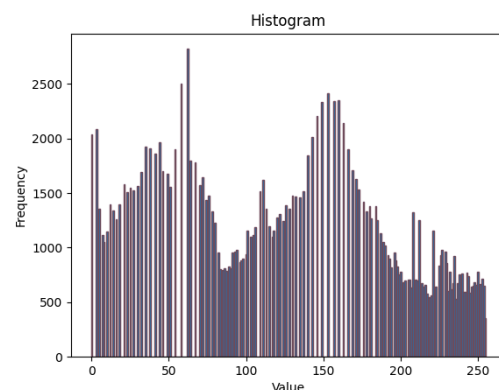




Στην συνέχεια παρατηρούμε τα αποτελέσματα της συμβατικής μεθόδου εξισορρόπησης ιστογράμματος, η οποία υπολογίζει τον μετασχηματισμό εξισορρόπησης σε όλη την εικόνα και έπειτα τον εφαρμόζει. Το διάνυσμα του μετασχηματισμού φαίνεται δίπλα, το οποίο έχει 256 στάθμες τιμών, όσες και οι διαφορετικές τιμές που μπορεί να πάρει ο τύπος δεδομένων uint8. ($2^8=256$)



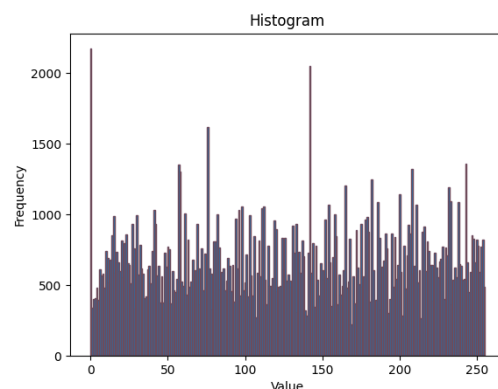
Πρόκειται για μια βελτιωμένη εκδοχή της εικόνας, όπως συμπεραίνουμε τόσο από την απεικόνιση όσο και από το ιστόγραμμα της. Ωστόσο, η φωτεινότητα έχει αυξηθεί υπερβολικά σε ορισμένα σημεία (κέντρο) και άρα δεν είναι ιδανική αισθητικά. Ως προς το ιστόγραμμα, παρατηρούμε μια μετατόπιση και ομαλοποίηση της αρχικής μορφής του, διατηρώντας ωστόσο κάποιες «ανωμαλίες». Αυτό θα μπορούσαμε να το αποδώσουμε στην στρογγυλοποίηση του μετασχηματισμού εξισορρόπησης σε συνδυασμό με το μεγάλο πλήθος σκοτεινών σημείων.



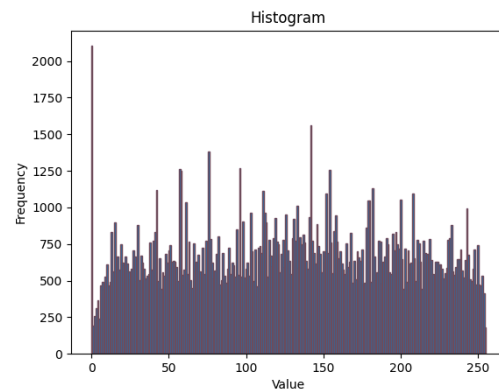
Η τελευταία τεχνική που εφαρμόστηκε είναι η Προσαρμοστική Εξισορρόπηση Ιστογράμματος, κατά την οποία η εικόνα χωρίζεται σε κομμάτια (contextual regions) δοσμένου μήκους, υπολογίζεται ο μετασχηματισμός εξισορρόπησης κάθε κομματιού και εφαρμόζεται παρεμβολή των τεσσάρων κοντινότερων μετασχηματισμών κάθε pixel. Οι αντίστοιχες μαθηματικές σχέσεις δίνονται στο pdf της εκφώνησης.

Αρχικά η μέθοδος εφαρμόζεται χωρίς παρεμβολή και όπως φαίνεται η εικόνα έχει πολύ έντονα διακριτές περιοχές, που επιβεβαιώνουν τις παρατηρήσεις της θεωρίας. Για την παραγωγή της εικόνας χωρίς παρεμβολή των μετασχηματισμών, η συνάρτηση `perform_adaptive_hist_equalization` ματαβλήθηκε με τον εξής τρόπο:

```
def perform_adaptive_hist_equalization(img_array, region_len_h,
region_len_w):
    equalized_img = img_array.copy()
    m, n = img_array.shape
    h_range = m // region_len_h
    w_range = n // region_len_w
    region_to_eq_transform =
calculate_eq_transformations_of_regions(img_array, region_len_h,
region_len_w)
    for h in range(h_range):
        h1 = h * region_len_h
        h2 = m - 1 if h == h_range - 1 else (h + 1) * region_len_h
        for w in range(w_range):
            w1 = w * region_len_w
            w2 = n - 1 if w == w_range - 1 else (w + 1) * region_len_w
            T = region_to_eq_transform[(h, w)]
            equalized_img[h1:h2, w1:w2] = T[img_array[h1:h2, w1:w2]]
    return equalized_img
```



Στην συνέχεια εφαρμόζεται η τεχνική με παρεμβολή και το αποτέλεσμα είναι σαφώς πιο ομαλό. Πρόκειται για την πιο αποτελεσματική τεχνική, εφόσον το ιστόγραμμα της προσεγγίζει περισσότερο την ομοιόμορφη κατανομή. Ως προς την μεγάλη συχνότητα του 0 (spike) πρόκειται για συσσώρευση σκοτεινών σημείων στην πάνω δεξιά γωνία της εικόνας. Παρόλο πολύ αποτελεσματική, η προσαρμοστική εξισορρόπηση ιστογράμματος με παρεμβολή παρουσιάζει ασυνέχειες μεταξύ των ορίων και μεταξύ των κομματιών της εικόνας με πολύ μεγάλη ή μικρή φωτεινότητα. Επίσης, εισάγει εμφανώς αρκετό θόρυβο αλλά αναμφίβολα παράγει την πιο ευκρινή και εξισορροπημένη εικόνα από τις τρεις περιπτώσεις.



Παραδοχές που χρησιμοποιήθηκαν

- ❖ Για την λειτουργία των scripts είναι απαραίτητο όλα τα αρχεία να βρίσκονται στον ίδιο φάκελο, όπως επίσης και η δοσμένη εικόνα *input_img.png*.
- ❖ Όπως υποδεικνύει το pdf της εκφώνησης, αντί για τον διαχωρισμό σε corner points, outer points και inner points επιλέχθηκε η απλουστευμένη προσέγγιση του διαχωρισμού σε outer points και inner points.
- ❖ Στην συνάρτηση `def calculate_eq_transformations_of_regions()` του αρχείου *adaptive_hist_eq.py*, σε περίπτωση που τα region lengths δεν διαιρούν τέλεια τις διαστάσεις της εικόνας, τα pixel που περισσεύουν από την διαίρεση αποδίδονται στα contextual regions που εφάπτονται στο δεξιά και κάτω άκρο της εικόνας. Η ίδια παραδοχή γίνεται επίσης όταν εφαρμόζουμε τον μετασχηματισμό εξισορρόπησης στα outer regions.
- ❖ Στην συνάρτηση `get_equalization_transform_of_img` αρχικά το μέγεθος του μετασχηματισμού επιλέχθηκε να καθορίζεται από την μέγιστη τιμή του πίνακα, αλλά κατά την προσαρμοστική εξισορρόπηση ιστογράμματος κάποια regions δεν περιείχαν όλες τις τιμές, άρα προκλήθηκαν σφάλματα. Αντ' αυτού, το μέγεθος του διανύσματος μετασχηματισμού θεωρήθηκε σταθερά 256, εξαιτίας του αριθμού διαφορετικών τιμών του τύπου δεδομένων `uint8`.
- ❖ Οι τεχνικές αυτές εφαρμόστηκαν και σε άλλες εικόνες τοπικά και διαπιστώθηκε πως, όσο μεγαλώνει ο λόγος του μεγέθους της εικόνας προς το μέγεθος contextual regions, το αποτέλεσμα αλλοιώνεται σημαντικά. Μία καλή προσέγγιση επιτυγχάνεται όταν η εικόνα χωρίζεται σε περίπου 50 regions.

Πηγές

- ❖ Γλωσσικό μοντέλο [ChatGPT 3.5](#)
- ❖ Επέκταση κειμενογράφου [Code Blocks](#)
- ❖ [Visual Studio Code](#) - Community Edition
- ❖ [Python](#) - Version 3.10.12