

# ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ

## Εργασία 1: Βελτίωση εικόνων μέσω εξισορρόπησης ιστογράμματος

Μαυρογεωργιάδη Χριστίνα 10439

Απρίλιος 2024

## Περιεχόμενα

1	Εισαγωγή	1
2	Εξισορρόπηση ιστογράμματος	1
2.1	Συνάρτηση <code>get_equalization_transform_of_img</code>	2
2.1.1	Function definition	2
2.1.2	Code explanation	2
2.2	Συνάρτηση <code>perform_global_hist_equalization</code>	3
2.2.1	Function definition	3
2.2.2	Code explanation	3
2.3	Αποτελέσματα επεξεργασίας	3
3	Προσαρμοστική εξισορρόπηση ιστογράμματος	6
3.1	Συνάρτηση <code>calculate_eq_transformations_of_regions</code>	6
3.1.1	Function definition	6
3.1.2	Code explanation	7
3.2	Συνάρτηση <code>perform_adaptive_hist_equalization</code>	8
3.2.1	Function definition	8
3.2.2	Code explanation	8
3.3	Συνάρτηση <code>perform_no_interpolation_ahc</code>	14
3.3.1	Function definition	14
3.3.2	Code explanation	14
3.4	Παραδοχές	14
3.5	Αποτελέσματα επεξεργασίας	15

## 1 Εισαγωγή

Στην παρούσα εργασία, εφαρμόζονται δύο τεχνικές επεξεργασίας ιστογράμματος, οι οποίες αποσκοπούν στην βελτίωση της αντίθεσης (contrast enhancement) μιας ψηφιακής εικόνας εισόδου. Ειδικότερα, υλοποιούνται σε κώδικα Python η συμβατική μέθοδος Εξισορρόπησης Ιστογράμματος (Histogram Equalization), καθώς και η μέθοδος Προσαρμοστικής Εξισορρόπησης Ιστογράμματος (Adaptive Histogram Equalization). Οι μέθοδοι αυτές εφαρμόζονται σε μια δεδομένη εικόνα εισόδου σε ένα πρόγραμμα `demo.py`, το οποίο επιδεικνύει τα αποτελέσματα της επεξεργασίας. Ο τρόπος υλοποίησης των μεθόδων καθώς και τα αποτελέσματα της εφαρμογής τους στην εικόνα εισόδου επιδεικνύονται και σχολιάζονται στην παρούσα αναφορά. Σημειώνεται πως για την υλοποίηση χρησιμοποιήθηκε η έκδοση Python 3.11.

## 2 Εξισορρόπηση ιστογράμματος

Η μέθοδος της εξισορρόπησης ιστογράμματος συνίσταται στην εφαρμογή ενός διακριτού μετασχηματισμού έντασης σε κάθε ένα από τα pixels της εικόνας εισόδου. Ο μετασχηματισμός αυτός εξάγεται για μια δεδομένη εικόνα εισόδου από την πληροφορία που μας παρέχει το ιστόγραμμά της. Συγκεκριμένα, για μια εικόνα  $f(n_1, n_2)$  με  $L$  στάθμες έντασης  $x_0 \leq x_1 \leq \dots \leq x_{L-1}$ , ο μετασχηματισμός προκύπτει ως:

$$v_k = \sum_{k=0}^{L-1} p(x_k) \quad (1)$$

$$y_k = \text{round}((L - 1) \cdot \frac{v_k - v_0}{1 - v_0}) \quad (2)$$

για  $k = 0, 1, \dots, L - 1$ , όπου  $p(x_k)$  είναι η σχετική συχνότητα της στάθμης  $x_k$  στην εικόνα εισόδου.

Για την υλοποίηση της μεθόδου, κατασκευάζονται δύο συναρτήσεις

```
def get_equalization_transform_of_img(img_array: np.ndarray) -> np.ndarray:
def perform_global_hist_equalization(img_array: np.ndarray) -> np.ndarray:
```

στο αρχείο `global_hist_eq.py`.

Για τους σκοπούς επεξεργασίας 8-bit grayscale εικόνων στην παρούσα εργασία, ορίζουμε μια σταθερά `NUM_LEVELS` με `global scope` στο αρχείο `global_hist_eq.py`, η οποία αναπαριστά το πλήθος των χβαντισμένων σταθμών των τιμών έντασης της εικόνας και της αναθέτουμε την τιμή 256.

## 2.1 Συνάρτηση `get_equalization_transform_of_img`

### 2.1.1 Function definition

```
def get_equalization_transform_of_img(img_array: np.ndarray) -> np.ndarray:
```

Σκοπός αυτής της συνάρτησης είναι να υπολογίσει τον μετασχηματισμό συμβατικής εξισορρόπησης ιστογράμματος μιας εικόνας εισόδου.

- Η εικόνα εισόδου `img_array` εισάγεται στη συνάρτηση με τη μορφή ενός `numpy ndarray` τύπου `dtype= numpy.uint8` 2 διαστάσεων και αναπαριστά μια 8-bit grayscale εικόνα. Αυτό σημαίνει ότι κάθε στοιχείο του πίνακα μπορεί να λάβει μία από τις  $L = 2^8 = 256$  τιμές έντασης  $0, 1, \dots, 255$ .
- Η έξοδος της συνάρτησης είναι ένα μονοδιάστατο διάνυσμα `equalization_transform`, το οποίο απεικονίζει τις τιμές έντασης της εικόνας εισόδου στις τιμές έντασης της εικόνας εξόδου μέσω του μετασχηματισμού εξισορρόπησης. Το διάνυσμα αυτό έχει μήκος ίσο με το πλήθος των χβαντισμένων σταθμών της εικόνας εισόδου, στην περίπτωση μιας 8-bit grayscale εικόνας, ίσο με  $L = 256$ . Αναπαρίσταται από ένα `numpy ndarray` τύπου `dtype= numpy.uint8` 1 διάστασης μήκους  $L$ . Η τιμή έντασης  $i = 0, 1, \dots, 255$  απεικονίζεται στο στοιχείο του πίνακα `equalization_transform` με δείκτη  $i$ , `equalization_transform[i]`.

### 2.1.2 Code explanation

Το πρώτο βήμα για τον καθορισμό της συνάρτησης μετασχηματισμού εξισορρόπησης είναι ο υπολογισμός του ιστογράμματος της εικόνας εισόδου. Για το σκοπό αυτό, μετατρέπουμε τον 2-διάστατο πίνακα αναπαράστασης της εικόνας σε έναν σειριακό μονοδιάστατο πίνακα:

```
img_array_1d = img_array.flatten()
```

Στη συνέχεια ορίζουμε μια μεταβλητή `histogram` της μορφής `numpy ndarray` τύπου `dtype= numpy.uint64`, 1 διάστασης και μήκους  $L$ . Το πλήθος των εμφανίσεων της τιμής έντασης  $i = 0, 1, \dots, 255$  απεικονίζεται στο στοιχείο του πίνακα `histogram` με δείκτη  $i$ , `histogram[i]`. Η μεταβλητή `histogram` αρχικοποιείται με μηδενικά:

```
histogram = np.zeros(NUM_LEVELS, dtype=np.uint64)
```

Ο υπολογισμός του πλήθους εμφανίσεων κάθε τιμής έντασης της εικόνας γίνεται μέσω ενός `for loop`:

```
for pixel in img_array_1d:
    # count how many times an intensity value appears in the image
    histogram[pixel] += 1
```

το οποίο διατρέχει τα `pixels` της εικόνας και προσθέτει μία μονάδα στο στοιχείο `histogram[pixel]` κάθε φορά που συναντά την τιμή έντασης `pixel` στην εικόνα.

Το ιστόγραμμα κανονικοποιείται διαιρώντας με το πλήθος των δειγμάτων της εικόνας μέσω της εντολής:

```
histogram = np.divide(histogram, n)
```

όπου:

```
n = img_array.size
```

είναι το πλήθος των στοιχείων του πίνακα αναπαράστασης της εικόνας.

Με τη βοήθεια του ιστογράμματος υπολογίζεται η αθροιστική συνάρτηση κατανομής της εικόνας εισόδου:

```
cdf = [np.sum(histogram[:(i+1)]) for i in range(NUM_LEVELS)]
```

Τέλος, ο μετασχηματισμός εξισορρόπησης ιστογράμματος υπολογίζεται ως:

```
equalization_transform = np.round((cdf - cdf[0])/(1 - cdf[0])*(NUM_LEVELS-1))
```

όπως εξάγεται από τις Σχέσεις (1) και (2),  
και επιστρέφεται από τη συνάρτηση.

## 2.2 Συνάρτηση `perform_global_hist_equalization`

### 2.2.1 Function definition

```
def perform_global_hist_equalization(img_array: np.ndarray) -> np.ndarray:
```

Σκοπός αυτής της συνάρτησης αποτελεί να εφαρμόσει τον συμβατικό μετασχηματισμό εξισορρόπησης ιστογράμματος στην εικόνα εισόδου `img_array`, αξιοποιώντας την συνάρτηση εύρεσης του πίνακα μετασχηματισμού που αναλύεται στην ενότητα 2.1.

- Η εικόνα εισόδου `img_array` εισάγεται στη συνάρτηση με μορφή πανομοιότυπη με αυτή που αναφέρθηκε στην ενότητα 2.1.1 και αποτελεί μια 8-bit grayscale εικόνα.
- Η έξοδος της συνάρτησης είναι η εικόνα εξόδου του μετασχηματισμού, `equalized_img`, και αναπαρίσταται από έναν 2-διάστατο πίνακα, όπως η εικόνα εισόδου.

### 2.2.2 Code explanation

Η υλοποίηση της συνάρτησης αυτής συνίσταται σε δύο απλά βήματα. Πρώτα, υπολογίζεται ο πίνακας μετασχηματισμού των τιμών έντασης της εικόνας εισόδου στις τιμές έντασης της εικόνας εξόδου καλώντας τη συνάρτηση `get_equalization_transform_of_img()` με όρισμα την εικόνα εισόδου:

```
equalization_transform = get_equalization_transform_of_img(img_array)
```

Στη συνέχεια, ο μετασχηματισμός αυτός εφαρμόζεται σε κάθε pixel της εικόνας εισόδου:

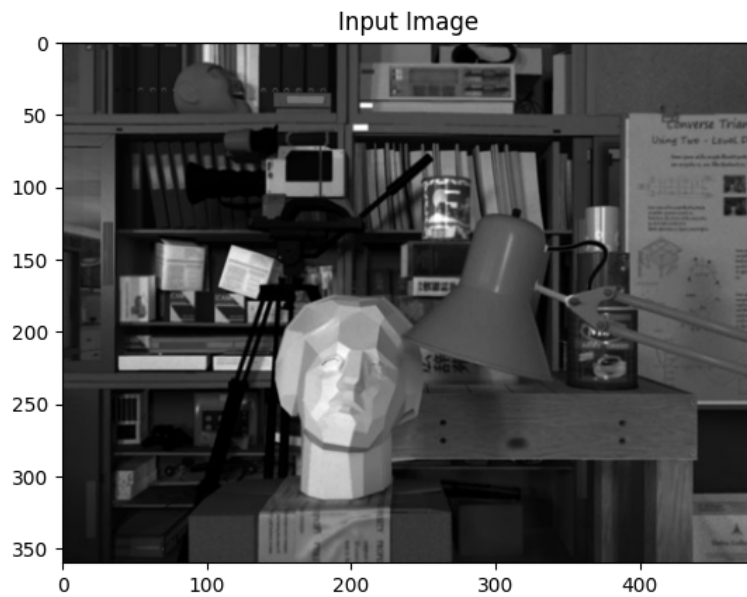
```
equalized_img = np.array([equalization_transform[level] for level in img_array])
```

Τέλος, η συνάρτηση επιστρέφει την εικόνα εξόδου `equalized_img`.

## 2.3 Αποτελέσματα επεξεργασίας

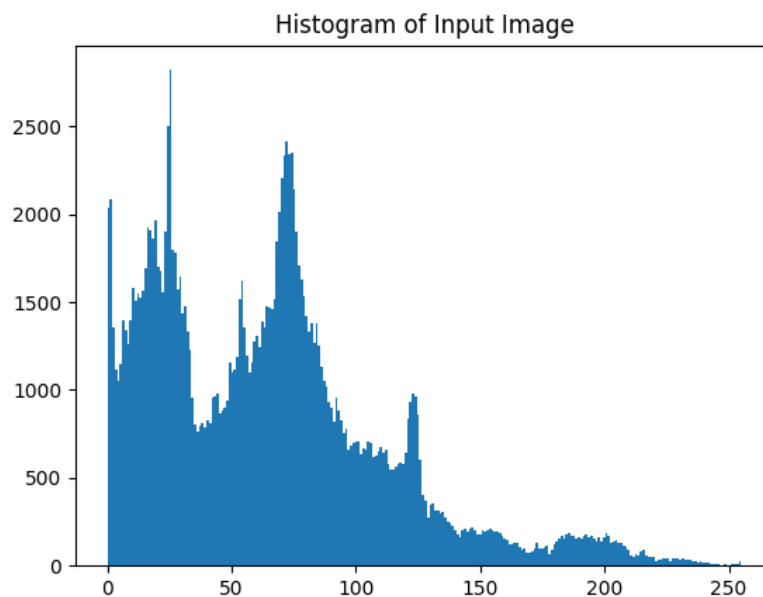
Στην παρούσα υπο-ενότητα, παρουσιάζονται και σχολιάζονται τα αποτελέσματα της εφαρμογής της Εξισορρόπησης ιστογράμματος σε μια δεδομένη εικόνα εισόδου. Τα αποτελέσματα αυτά παράγονται από ένα πρόγραμμα επίδειξης `demo.py`.

Η εικόνα εισόδου του αλγορίθμου επεξεργασίας δίνεται παρακάτω σε κλίμακα Grayscale:



Σχήμα 1: Input Image

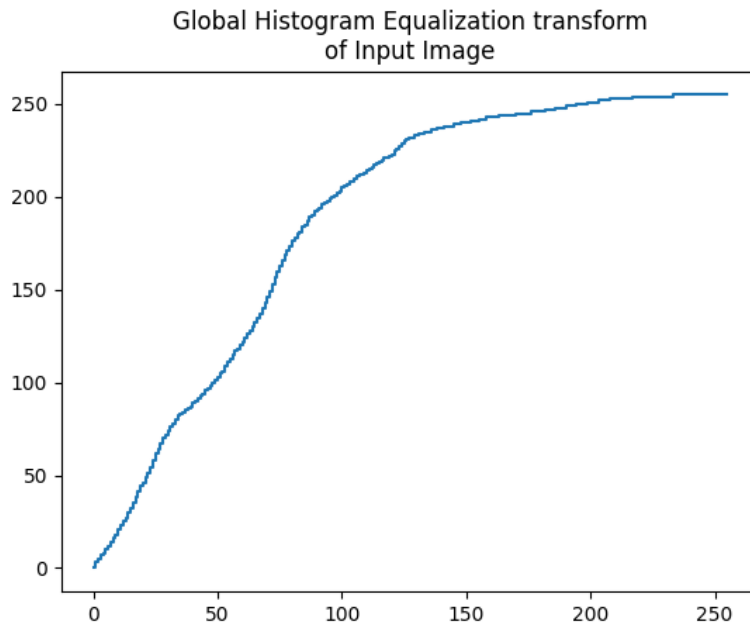
Το ιστόγραμμα της εικόνας εισόδου είναι:



Σχήμα 2: Histogram of Input Image

Παρατηρούμε ότι οι συνιστώσες του ιστογράμματος συγκεντρώνονται στο αριστερό άκρο του άξονα των τιμών έντασης της εικόνας. Το γεγονός αυτό επιδεικνύει πως οι χαμηλές τιμές φωτεινότητας είναι κυρίαρχες στην εικόνα, ενώ λείπουν οι υψηλές τιμές. Με άλλα λόγια, στην εικόνα υπερτερούν το μαύρο χρώμα και οι σκούρες αποχρώσεις του γκρι, ενώ απουσιάζουν το λευκό χρώμα και οι ανοιχτές αποχρώσεις του γκρι. Η παρατήρηση της εικόνας εισόδου καθώς και του ιστογράμμά της μας οδηγούν στο συμπέρασμα ότι πρόκειται για μια σκοτεινή εικόνα με χαμηλή αντίθεση, στην οποία είναι δυσδιάκριτη η λεπτομέρεια.

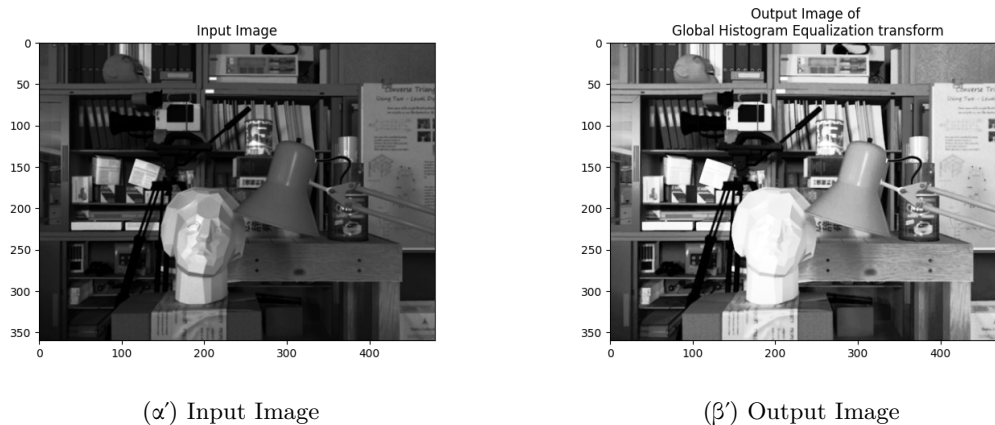
Για την εξισορρόπηση του ιστογράμματος, και κατά συνέπεια τη βελτίωση της αντίθεσης, της εικόνας εισόδου, εφαρμόζουμε στις τιμές έντασης της εικόνας τον μετασχηματισμό που φαίνεται στο παρακάτω διάγραμμα:



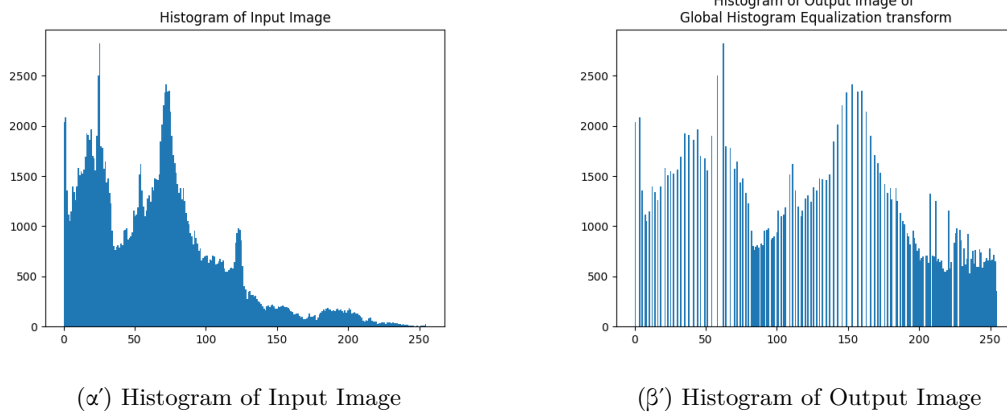
Σχήμα 3: Global Histogram Equalization transform of Input Image

Ο μετασχηματισμός απεικονίζει μια στενότερη ζώνη χαμηλών τιμών έντασης σε μια ευρύτερη περιοχή τιμών εξόδου, διευρύνοντας με αυτόν τον τρόπο τις τιμές των σκοτεινών pixels της εικόνας. Αντίστροφα, οι υψηλότερες τιμές των επιπέδων εισόδου απεικονίζονται σε μια στενότερη περιοχή τιμών στην έξοδο, συμβάλλοντας τελικά στην αύξηση της αντίθεσης της εικόνας εισόδου. Παρατηρούμε ότι πολλές διαδοχικές στάθμες εισόδου απεικονίζονται στην ίδια στάθμη εξόδου, γεγονός που εξηγείται ως απόρροια της πράξης *round()* που εφαρμόζει ο μετασχηματισμός για την χβάντιση των τιμών εξόδου. Ως εκ τούτου, ο μετασχηματισμός εξισορρόπησης δεν είναι αντιστρέψιμος.

Παρακάτω, παραθέτουμε την τελική εικόνα σε σύγκριση με την αρχική, καθώς και τα αντίστοιχα ιστογράμμά τους.



Σχήμα 4: Global Histogram Equalization



Σχήμα 5: Global Histogram Equalization

Σε αντίθεση με το συνεχές ανάλογο του μετασχηματισμού εξισορρόπησης, ο διακριτός μετασχηματισμός εξισορρόπησης δεν εξασφαλίζει ότι η εικόνα εξόδου θα χαρακτηρίζεται από ομοιόμορφη κατανομή. Ωστόσο, διακρίνουμε ότι ο μετασχηματισμός διασκορπίζει το ιστογράμμο έτσι ώστε οι τιμές της εικόνας εξόδου να εκτείνονται σε μια ευρύτερη περιοχή της κλίμακας έντασης, παράγοντας μια εικόνα με βελτιωμένη αντίθεση σε σύγκριση με την αρχική.

### 3 Προσαρμοστική εξισορρόπηση ιστογράμματος

Η εξισορρόπηση ιστογράμματος που εφαρμόστηκε στην ενότητα 2 είναι ένας σημειακός μετασχηματισμός που λαμβάνει υπόψη το περιεχόμενο όλης της εικόνας, αλλά όχι την ιδιαίτερη κατανομή της τοπικής ενός σημείου πληροφορίας. Ως εκ τούτου, η εικόνα εξόδου του μετασχηματισμού, παρότι στο σύνολό της λαμβάνει τιμές σε μεγαλύτερο εύρος της κλίμακας έντασης από την αρχική εικόνα, δεν συγκρατεί τοπικά χαρακτηριστικά της κατανομής έντασης μεμονωμένων περιοχών. Για παράδειγμα, η απεικόνιση των χαμηλών-μεσαίων τιμών έντασης της εικόνας εισόδου του Σχήματος (1) (αποχρώσεις του γκρι) σε υψηλότερες τιμές προκαλεί την αύξηση της φωτεινότητας της περιοχής του προσώπου, αλλοιώνει, ωστόσο, την τοπική κατανομή έντασης της περιοχής. Ως αποτέλεσμα, αμβλύνεται η αντίθεση μεταξύ των pixels του προσώπου και χάνεται σημαντική διακριτή λεπτομέρεια των σκιάσεων και των γωνιών αυτής της περιοχής της εικόνας.

Για να αντιμετωπίσουμε το πρόβλημα που θέσαμε παραπάνω, εφαρμόζουμε στην εικόνα εισόδου μια βελτιωμένη εκδοχή της συμβατικής εξισορρόπησης ιστογράμματος, η οποία καλείται Προσαρμοστική Εξισορρόπηση Ιστογράμματος. Η βάση αυτής της μεθόδου συνίσταται στην εφαρμογή του μετασχηματισμού εξισορρόπησης μεμονωμένα σε τοπικές περιοχές της εικόνας εισόδου. Η τιμή απεικόνισης κάθε pixel της εικόνας παράγεται από την διγραμμική παρεμβολή των τιμών απεικόνισης των μετασχηματισμών των περιοχών που περιβάλλουν το εν λόγω pixel.

Για την υλοποίηση της μεθόδου, κατασκευάζονται δύο συναρτήσεις

```
def calculate_eq_transformations_of_regions(img_array: np.ndarray, region_len_h: int,
region_len_w: int) -> Dict[Tuple, np.ndarray]:
```

```
def perform_adaptive_hist_equalization(img_array: np.ndarray, region_len_h: int,
region_len_w: int) -> np.ndarray:
```

στο αρχείο `adaptive_hist_eq.py`.

Επιπλέον, χρησιμοποιείται η εντολή:

```
import global_hist_eq.py
```

έτσι ώστε να γίνει δυνατή η πρόσβαση στις βοηθητικές συναρτήσεις που υλοποιήθηκαν στην προηγούμενη ενότητα.

#### 3.1 Συνάρτηση `calculate_eq_transformations_of_regions`

##### 3.1.1 Function definition

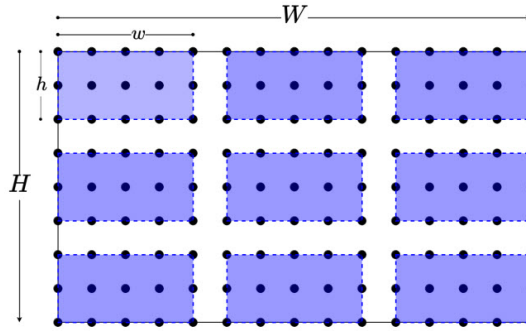
```
def calculate_eq_transformations_of_regions(img_array: np.ndarray, region_len_h: int,
region_len_w: int) -> Dict[Tuple, np.ndarray]:
```

Σκοπός αυτής της συνάρτησης είναι να υπολογίσει τον μετασχηματισμό εξισορρόπησης για κάθε contextual region στα οποία διαχωρίζεται η εικόνα.

- Η εικόνα εισόδου εισάγεται στη συνάρτηση με τη μορφή που αναφέρθηκε στην ενότητα 2.1.1.
- Η συνάρτηση δέχεται επιπλέον ως ορίσματα δύο ακεραίους αριθμούς οι οποίοι αναπαριστούν τις διαστάσεις των contextual regions στα οποία διαχωρίζεται η εικόνα. Η μεταβλητή `region_len_h` τύπου `int` αναπαριστά το ύψος  $h$  της περιοχής (σε πλήθος pixels), ενώ η μεταβλητή `region_len_w` τύπου `int` αναπαριστά το πλάτος  $w$  της περιοχής, όπως φαίνεται στο Σχήμα (6).
- Η έξοδος της συνάρτησης αποτελεί μια αντιστοίχιση της κάθε περιοχής της εικόνας στον μετασχηματισμό εξισορρόπησης της. Στο πλαίσιο αυτής της εργασίας, ένα contextual region ορίζεται μονοσήμαντα από τη θέση της άνω αριστερά κορυφής του. Η θέση του pixel αυτού μπορεί να αναπαρασταθεί από ένα Python tuple που περιλαμβάνει το ζεύγος δεικτών του εν λόγω pixel στην εικόνα. Επομένως, το tuple  $(h_R, w_R)$  αναπαριστά το contextual region:

$$\mathcal{R}_{h_R, w_R} = [h_R, h_R + h) \times [w_R, w_R + w) \quad (3)$$

Η αντιστοίχιση του κάθε contextual region στον μετασχηματισμό εξισορρόπησης του πραγματοποιείται μέσω μιας δομής Python dict, με keys τα tuples που αναπαριστούν τις περιοχές της εικόνας και values τους μετασχηματισμούς αυτών. Οι μετασχηματισμοί αναπαρίστανται όπως έχει αναφερθεί στην ενότητα 2.1.1.



Σχήμα 6: Παράδειγμα δειγμάτων μιας εικόνας και των contextual regions της

### 3.1.2 Code explanation

Πρωταρχικό βήμα της μεθόδου είναι η διαμέριση της εικόνας εισόδου σε contextual regions, με βάση τις επιθυμητές διαστάσεις. Θεωρούμε ότι οι διαστάσεις του contextual region διαιρούν επακριβώς τις αντίστοιχες διαστάσεις της εικόνας εισόδου. Η διαμέριση της εικόνας σε περιοχές ισοδυναμεί με την εύρεση των άνω αριστερά κορυφών των περιοχών αυτών, όπως έχει αναλυθεί παραπάνω. Για το σκοπό αυτό, διατρέχουμε την εικόνα προς την οριζόντια και κατακόρυφη διεύθυνση με βήμα  $h$  και  $w$  αντίστοιχα και αποθηκεύουμε τα ζεύγη δεικτών που προκύπτουν ως tuples σε μια δομή Python list:

```
regions = [(i, j) for i in range(0, m, region_len_h) for j in range(0, n, region_len_w)]
```

όπου  $m$  και  $n$  είναι η κατακόρυφη και η οριζόντια διάσταση της εικόνας εισόδου, αντίστοιχα.

Αφού έχουμε ορίσει τις περιοχές της εικόνας, διατρέχουμε τη λίστα `regions` και για κάθε στοιχείο της (`region`) υπολογίζουμε τα pixels της αρχικής εικόνας που ανήκουν στην αντίστοιχη περιοχή με βάση την Σχέση (3). Παράγεται, έτσι, μια νέα εικόνα διαστάσεων  $(h \times w)$ , η οποία δίνεται ως είσοδος στη συνάρτηση `get_equalization_transform_of_img` (ορίζεται στο module `global_hist_eq`). Με αυτόν τον τρόπο, υπολογίζεται ο μετασχηματισμός εξισορρόπησης (`eq_transform`) της περιοχής και η μεταβλητή `region_to_eq_transform`, τύπου dictionary, ανανεώνεται σε κάθε επανάληψη του βρόχου με το νέο key-value pair `{region: eq_transform}`.

```
for region in regions:
```

```
    # find image array that corresponds to region of input image
    img_region_array = img_array[region[0]:(region[0] + region_len_h), region[1]:(region[1] + region_len_w)]
    # compute equalization transform on the above image array
    eq_transform = global_hist_eq.get_equalization_transform_of_img(img_region_array)
    # update the dictionary with the key-value pair region: region transform
```



```
region_to_eq_transform.update({region: eq_transform})
```

Τέλος, η συνάρτηση επιστρέφει τη μεταβλητή `region_to_eq_transform`.

## 3.2 Συνάρτηση `perform_adaptive_hist_equalization`

### 3.2.1 Function definition

```
def perform_adaptive_hist_equalization(img_array: np.ndarray, region_len_h: int,
region_len_w: int) -> np.ndarray:
```

Σκοπός της συνάρτησης είναι να εφαρμόσει την Προσαρμοστική Εξισορρόπηση Ιστογράμματος στην εικόνα εισόδου για τις δεδομένες διαστάσεις των contextual regions της εικόνας.

- Η εικόνα εισόδου εισάγεται στη συνάρτηση με τη μορφή που περιγράφεται στην ενότητα 2.1.1.
- Η συνάρτηση δέχεται ως ορίσματα τις επιθυμητές διαστάσεις των contextual regions στις οποίες διαχωρίζεται η εικόνα, όπως περιγράφεται στην ενότητα 3.1.1.
- Η έξοδος της συνάρτησης είναι η εικόνα εξόδου του μετασχηματισμού και έχει μορφή όμοια με αυτή της εικόνας εισόδου, είναι δηλαδή μια 8-bit grayscale εικόνα ιδίου τύπου.

### 3.2.2 Code explanation

Πρώτο βήμα είναι να υπολογίσουμε τους μετασχηματισμούς εξισορρόπησης όλων των πιθανών contextual regions της εικόνας:

```
region_to_eq_transform = calculate_eq_transformations_of_regions(img_array, region_len_h,
region_len_w)
```

Όπως αναφέρεται στην ενότητα 3.1.1, η μεταβλητή `region_to_eq_transform` είναι μια μεταβλητή τύπου `dictionary` που αντιστοιχίζει τις περιοχές της εικόνας στον μετασχηματισμό τους. Μπορούμε, λοιπόν, να εξάγουμε τη λίστα των contextual regions της εικόνας μέσω της εντολής:

```
regions = region_to_eq_transform.keys()
```

Το δεύτερο βήμα της προεργασίας της συνάρτησης είναι να βρει τα contextual κέντρα των περιοχών της εικόνας. Ορίζουμε μια δομή `dict`, `region_to_center`, η οποία αντιστοιχίζει κάθε contextual region στο κέντρο της.

Το κέντρο μιας περιοχής υπολογίζεται με τον εξής τρόπο:

$$(h_C, w_C) = (h_R + \lfloor \frac{h}{2} \rfloor, w_R + \lfloor \frac{w}{2} \rfloor) \quad (4)$$

όπου  $(h \times w)$  είναι οι διαστάσεις της περιοχής και  $(h_R, w_R)$  η άνω αριστερά κορυφή της.

Η αντίστοιχη υλοποίηση σε κώδικα είναι:

```
for region in regions:
    # find center of contextual region
    center = (region[0] + region_len_h // 2, region[1] + region_len_w // 2)
    # update dictionary with the new key-value pair
    region_to_center.update({region: center})
```

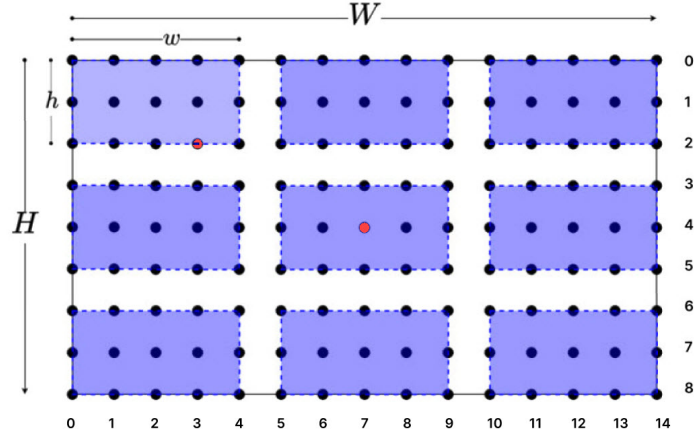
Στη συνέχεια, ακολουθεί το κύριο σώμα της συνάρτησης, το οποίο συνιστά ένα διπλό `for loop` που διατρέχει όλα τα pixels  $(i, j)$  της εικόνας και εφαρμόζει τον κατάλληλο μετασχηματισμό έντασης σε κάθε ένα από αυτά. Σε κάθε επανάληψη του βρόχου εκτελούνται κάποιοι πρωταρχικοί υπολογισμοί σχετικά με το αντίστοιχο pixel της εικόνας. Οι υπολογισμοί αυτοί συνιστούν:

- την εύρεση του contextual region στο οποίο ανήκει το pixel καθώς και του contextual κέντρου αυτού,
- την εύρεση του μετασχηματισμού εξισορρόπησης του παραπάνω contextual region.

Η εύρεση του contextual region  $\mathcal{R}_{h_R, w_R}$  στο οποίο ανήκει ένα pixel  $(i, j) \in [0, H) \times [0, W)$  μιας εικόνας διαστάσεων  $(H \times W)$  πραγματοποιείται με τον εξής τρόπο:

$$(h_R, w_R) = (\lfloor \frac{i}{h} \rfloor \cdot h, \lfloor \frac{j}{w} \rfloor \cdot w) \quad (5)$$





Σχήμα 7: Παράδειγμα εύρεσης contextual region στο οποίο ανήκει ένα pixel

Για παράδειγμα, αν κάθε contextual region έχει διαστάσεις  $3 \times 5$ , το pixel (2,3) αντιστοιχεί στην περιοχή (0,0) (χρησιμοποιώντας την αναπαράσταση που περιγράφεται στην ενότητα 3.1.1), όπως φαίνεται στο Σχήμα (7). Η περιοχή στην οποία ανήκει το pixel βρίσκεται αναλυτικά χρησιμοποιώντας τη Σχέση (5):

$$(h_R, w_R) = (\lfloor \frac{2}{3} \rfloor \cdot 3, \lfloor \frac{3}{5} \rfloor \cdot 5) = (0 \cdot 3, 0 \cdot 5) = (0, 0)$$

Ομοίως, το pixel (4,7) αντιστοιχεί στην περιοχή (3,5):

$$(h_R, w_R) = (\lfloor \frac{4}{3} \rfloor \cdot 3, \lfloor \frac{7}{5} \rfloor \cdot 5) = (1 \cdot 3, 1 \cdot 5) = (3, 5)$$

κ.ο.κ.

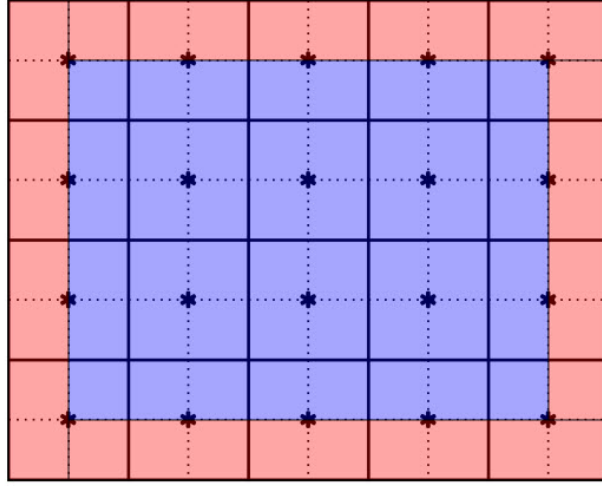
Οι ενέργειες που προαναφέρθηκαν εκτελούνται με τις παρακάτω εντολές σε κώδικα:

```
# find the contextual region each pixel resides in
region = ((i // region_len_h) * region_len_h, (j // region_len_w) * region_len_w)
# find the contextual center of this region
region_center = region_to_center[region]
# find the equalization transform of this contextual region
eq_transform = region_to_eq_transform[region]
```

Αφού εκτελεστούν οι υπολογισμοί αυτοί, το pixel κατηγοριοποιείται σε μια από τις παρακάτω δυνατές κατηγορίες:

- το pixel αποτελεί **contextual center** της περιοχής στην οποία ανήκει. Στην περίπτωση αυτή, αρκεί να εφαρμόσουμε στην τιμή έντασής του τον μετασχηματισμό εξισορρόπησης της περιοχής αυτής.
- το pixel αποτελεί **outer point** της εικόνας. Στην περίπτωση αυτή, ομοίως, αρκεί να εφαρμόσουμε στην τιμή έντασής του τον μετασχηματισμό της περιοχής στην οποία ανήκει.
- το pixel αποτελεί **inner point** της εικόνας, εξαιρουμένων των contextual κέντρων αυτής. Στην περίπτωση αυτή, υπολογίζουμε τους μετασχηματισμούς των 4 γειτονικών contextual regions και εφαρμόζουμε διγραμμική παρεμβολή ανάμεσα σε αυτούς με στόχο την εύρεση της τιμής έντασης στην οποία αντιστοιχίζεται το pixel.

Η διάκριση των pixels της εικόνας σε **outer** και **inner points** φαίνεται στο παρακάτω σχήμα, όπου η ζώνη των outer points απεικονίζονται με κόκκινο χρώμα, ενώ η ζώνη των inner points με μωβ χρώμα.



Σχήμα 8: Διαχωρισμός της εικόνας σε δύο ζώνες σημείων

Τα **outer points** της εικόνας μοντελοποιούνται ως εξής:

$$\{(i, j) \in \mathcal{R}_{h_R, w_R} : (h_R = 0 \wedge i < h_C) \vee (h_R = H - h_C \wedge i > h_C) \vee (w_R = 0 \wedge j < w_C) \vee (w_R = W - w_C \wedge j > w_C)\} \quad \forall (i, j) \in [0, H) \times [0, W) \quad (6)$$

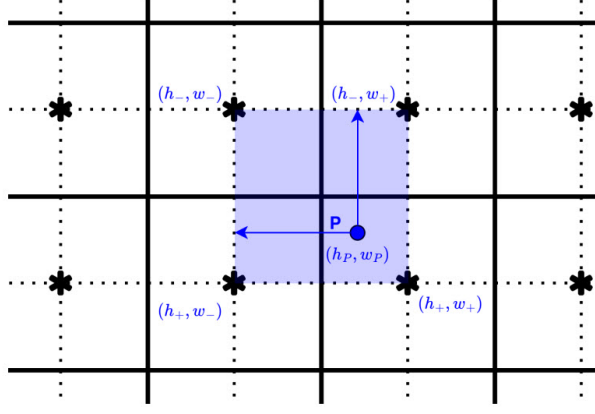
Στις περιπτώσεις των **contextual centers** και **outer points**, το pixel  $(i, j)$  της εικόνας εξόδου `equalized_img` είναι ίσο με το μετασχηματισμό `eq_transform` του αντίστοιχου pixel  $(i, j)$  της εικόνας εισόδου, όπου `eq_transform` είναι ο μετασχηματισμός εξισορρόπησης της περιοχής στην οποία ανήκει το pixel  $(i, j)$ :

```
if (i, j) == region_center:
    # pixel (i, j) is a contextual center

    # use the equalization transform of the contextual region
    # that the pixel resides in
    equalized_img[i, j] = eq_transform[img_array[i, j]]
elif region[0] == 0 and i < region_center[0] or \
    region[0] == m - region_len_h and i > region_center[0] or \
    region[1] == 0 and j < region_center[1] or \
    region[1] == n - region_len_w and j > region_center[1]:
    # pixel (i, j) is an outer point

    # use the equalization transform of the contextual region
    # that the pixel resides in
    equalized_img[i, j] = eq_transform[img_array[i, j]]
```

Στην περίπτωση των **inner points**, εξαφρουμένων των contextual centers, χρειάζεται να εντοπίσουμε τα 4 κοντινότερα contextual κέντρα του pixel  $(i, j)$ . Τα 4 γειτονικά contextual κέντρα αποτελούν τις κορυφές του ορθογωνίου που περικλείει το pixel. Στο παρακάτω σχήμα (Σχήμα (9)), το pixel  $(i, j)$  που μελετάμε αντιστοιχίζεται στο σημείο  $P(h_P, w_P)$ .



Σχήμα 9: Υπολογισμός της μετασχηματισμένης τιμής ενός σημείου  $P$ , με βάση τις γειτονικές του contextual regions

Για να εντοπίσουμε τα γειτονικά κέντρα, ελέγχουμε τη σχετική θέση του σημείου  $P(h_P, w_P)$  με το contextual κέντρο  $(h_C, w_C)$  της περιοχής στην οποία ανήκει. Όπως διακρίνεται στο πλέγμα που ορίζουν τα κέντρα των περιοχών, κάθε contextual region διαχωρίζεται σε 4 τεταρτημόρια (διάστικτες γραμμές), με κέντρο το contextual κέντρο της περιοχής. Ανάλογα με το τεταρτημόριο στο οποίο ανήκει το σημείο  $P$ , μπορούν να υπολογιστούν οι θέσεις των υπόλοιπων 3 γειτονικών κέντρων.

Σημειώνεται ότι εάν το σημείο  $P$  ανήκει στο όριο του ορθογωνίου που σχηματίζουν τα 4 γειτονικά κέντρα, είναι δηλαδή συνενυθιακό με δύο από τις κορυφές του ορθογωνίου, τότε μπορεί να θεωρηθεί ως σημείο δύο συνορευόμενων ορθογωνίων. Σε αυτή την περίπτωση, υιοθετούμε τη σύμβαση τα οριακά σημεία να ανατίθενται πάντοτε στο κάτω ή δεξιά ορθογώνιο, ανάλογα με το αν βρίσκονται στο οριζόντιο ή κατακόρυφο όριο του πλέγματος αντίστοιχα. Η σύμβαση αυτή καταρρίπτεται μόνο στην οριακή περίπτωση όπου δεν υπάρχει συνορευόμενο κάτω ή δεξιά ορθογώνιο. Αυτό συμβαίνει όταν το pixel  $(i, j)$  (σημείο  $P$ ) ικανοποιεί:

$$\{(i, j) \in \mathcal{R}_{h_R, w_R} : (h_R = H - h \quad \wedge \quad i < h_C) \vee (w_R = W - w \quad \wedge \quad j < w_C)\} \quad \forall (i, j) \in [0, H) \times [0, W) \quad (7)$$

Στην περίπτωση αυτή, το pixel αντιστοιχίζεται στο άνω ή αριστερά ορθογώνιο αντίστοιχα.

Επιστρέφουμε, λοιπόν, στην εύρεση των γειτονικών κέντρων του σημείου  $P$  στη γενική περίπτωση. Αρκεί να βρούμε τις τιμές  $\{h_+, h_-, w_+, w_-\}$  που χαρακτηρίζουν τις 4 κορυφές του ορθογωνίου, όπως φαίνεται στο Σχήμα (9). Όταν αναφερόμαστε στο ορθογώνιο, εφεξής θα εννοούμε το ορθογώνιο που σχηματίζουν τα 4 γειτονικά contextual κέντρα του σημείου. Διακρίνουμε τις εξής 4 περιπτώσεις:

(α') Το σημείο  $P$  ανήκει στο άνω αριστερά τεταρτημόριο του contextual region του.

Η περίπτωση αυτή απεικονίζεται στο Σχήμα (9). Τότε, το contextual κέντρο  $(h_C, w_C)$  της περιοχής του αντιστοιχίζεται στην κορυφή  $(h_+, w_+)$  του ορθογωνίου. Μπορούμε να βρούμε τις υπόλοιπες κορυφές του ορθογωνίου λαμβάνοντας υπόψιν ότι όλα τα contextual κέντρα του πλέγματος απέχουν μεταξύ τους κατά την οριζόντια διεύθυνση απόσταση  $w$  και κατά την κατακόρυφη διεύθυνση απόσταση  $h$ . Έχουμε:

$$\begin{cases} h_+ = h_C \\ h_- = h_+ - h \\ w_+ = w_C \\ w_- = w_+ - w \end{cases} \quad (8)$$

(β') Το σημείο  $P$  ανήκει στο άνω δεξιά τεταρτημόριο του contextual region του.

Τότε, το contextual κέντρο  $(h_C, w_C)$  της περιοχής του αντιστοιχίζεται στην κορυφή  $(h_+, w_-)$  του ορθογωνίου. Έχουμε:

$$\begin{cases} h_+ = h_C \\ h_- = h_+ - h \\ w_- = w_C \\ w_+ = w_- + w \end{cases} \quad (9)$$

(γ') Το σημείο  $P$  ανήκει στο κάτω αριστερά τεταρτημόριο του contextual region του.

Τότε, το contextual κέντρο  $(h_C, w_C)$  της περιοχής του αντιστοιχίζεται στην κορυφή  $(h_-, w_+)$  του ορθογωνίου. Έχουμε:

$$\begin{cases} h_- = h_C \\ h_+ = h_- + h \\ w_+ = w_C \\ w_- = w_+ - w \end{cases} \quad (10)$$

(δ') Το σημείο  $P$  ανήκει στο κάτω δεξιά τεταρτημόριο του contextual region του.

Τότε, το contextual κέντρο  $(h_C, w_C)$  της περιοχής του αντιστοιχίζεται στην κορυφή  $(h_-, w_-)$  του ορθογωνίου. Έχουμε:

$$\begin{cases} h_- = h_C \\ h_+ = h_- + h \\ w_- = w_C \\ w_+ = w_- + w \end{cases} \quad (11)$$

Για να εντοπίσουμε το τεταρτημόριο στο οποίο ανήκει το σημείο  $P$ , συγκρίνουμε τις συντεταγμένες του σημείου  $(h_P, w_P)$  με τις συντεταγμένες του contextual κέντρου  $(h_C, w_C)$  της περιοχής στην οποία ανήκει. Αφού αντιστοιχίσουμε το σημείο  $P$  στο κατάλληλο τεταρτημόριο, μπορούμε να υπολογίσουμε τις τιμές  $\{h_+, h_-, w_+, w_-\}$  που αντιστοιχούν στις συντεταγμένες των 4 γειτονικών contextual κέντρων.

Γνωρίζοντας το contextual κέντρο  $(h_C, w_C)$  μιας περιοχής  $\mathcal{R}_{h_R, w_R}$ , η αναπαράσταση  $(h_R, w_R)$  της περιοχής (όπως περιγράφεται στην ενότητα 3.1.1) μπορεί να εξαχθεί αντιστρέφοντας την Σχέση (4) ως εξής:

$$(h_R, w_R) = (h_C - \lfloor \frac{h}{2} \rfloor, w_C - \lfloor \frac{w}{2} \rfloor) \quad (12)$$

Χρησιμοποιώντας την Σχέση (12) μπορούμε να βρούμε την αναπαράσταση των 4 γειτονικών contextual regions  $\{\mathcal{R}_{+,+}, \mathcal{R}_{+,-}, \mathcal{R}_{-,+}, \mathcal{R}_{-,-}\}$  του σημείου  $P$  αντικαθιστώντας όπου  $(h_C, w_C)$  τις κατάλληλες συντεταγμένες των αντίστοιχων contextual κέντρων των περιοχών που έχουν υπολογιστεί με βάση τις Σχέσεις (8) – (11). Αν  $\{\mathcal{T}_{+,+}, \mathcal{T}_{+,-}, \mathcal{T}_{-,+}, \mathcal{T}_{-,-}\}$  είναι οι μετασχηματισμοί εξισορρόπησης των τιμών των αντίστοιχων contextual regions που έχουν υπολογιστεί νωρίτερα, τότε η τιμή  $x$  του σημείου  $P$  μπορεί να αντιστοιχεί στην τιμή  $y$  με τον παρακάτω τρόπο:

$$y = (1 - a) \cdot (1 - b) \cdot \mathcal{T}_{-,-}(x) + (1 - a) \cdot b \mathcal{T}_{+,-}(x) + a \cdot (1 - b) \cdot \mathcal{T}_{-,+}(x) + a \cdot b \cdot \mathcal{T}_{+,+}(x) \quad (13)$$

όπου οι συντελεστές βάρους υπολογίζονται με βάση:

$$\begin{aligned} a &= (w_P - w_-) / (w_+ - w_-) \\ b &= (h_P - h_-) / (h_+ - h_-) \end{aligned} \quad (14)$$

Η τιμή αντιστοίχισης  $y$  πρέπει να στρογγυλοποιηθεί στην κοντινότερη ακέραια τιμή, έτσι ώστε να αποτελεί επιτρεπτή στάθμη χβάντισης της εικόνας.

Τελικά, το pixel  $(i, j)$  της εικόνας εξόδου `equalized_img` τίθεται ίσο με την τιμή `round(y)`, όπου  $y$  είναι η μετασχηματισμένη τιμή του pixel  $(i, j)$  της εικόνας εισόδου (`x = img_array[i, j]`), όπως υπολογίζεται από τη Σχέση (13).

Η οριακή περίπτωση των σημείων  $(i, j)$  που ικανοποιούν την Σχέση (7) καλύπτεται από τον κώδικα με τις παρακάτω συνθήκες:

```
if region[1] == n - region_len_w and j == region_center[1]:
    # pixels in the rightmost contextual regions, that are collinear
    # with their contextual centers
    if i > region_center[0]:
        h_minus = region_center[0]
        w_plus = region_center[1]
        h_plus = h_minus + region_len_h
        w_minus = w_plus - region_len_w
    elif i < region_center[0]:
        h_plus = region_center[0]
        w_plus = region_center[1]
        h_minus = h_plus - region_len_h
        w_minus = w_plus - region_len_w
elif region[0] == m - region_len_h and i == region_center[0]:
```

```

# pixels in the lowermost contextual regions, that are collinear
# with their contextual centers
if j > region_center[1]:
    h_plus = region_center[0]
    w_minus = region_center[1]
    h_minus = h_plus - region_len_h
    w_plus = w_minus + region_len_w
elif j < region_center[1]:
    h_plus = region_center[0]
    w_plus = region_center[1]
    h_minus = h_plus - region_len_h
    w_minus = w_plus - region_len_w

```

όπου ο υπολογισμός των τιμών  $\{h_+, h_-, w_+, w_-\}$  ακολουθεί τους κανόνες που περιγράφηκαν στις παραγράφους (α')-(δ'), θεωρώντας ότι τα οριακά pixels αντιστοιχούν στην άνω ή αριστερά contextual region, ανάλογα με το αν βρίσκονται στο οριζόντιο ή κατακόρυφο όριο του πλέγματος των contextual κέντρων.

Ομοίως, για όλα τα υπόλοιπα inner points (εξαιρουμένων των contextual κέντρων), ο υπολογισμός των  $\{h_+, h_-, w_+, w_-\}$  ακολουθεί τους κανόνες που περιγράφηκαν στις παραγράφους (α')-(δ'), θεωρώντας όμως ότι τα οριακά pixels αντιστοιχούν στην κάτω ή δεξιά contextual region, ανάλογα με το αν βρίσκονται στο οριζόντιο ή κατακόρυφο όριο του πλέγματος των contextual κέντρων. Η υλοποίηση σε κώδικα δίνεται παρακάτω:

```

# all other pixels
if i >= region_center[0] and j >= region_center[1]:
    h_minus = region_center[0]
    w_minus = region_center[1]
    h_plus = h_minus + region_len_h
    w_plus = w_minus + region_len_w

if i >= region_center[0] and j < region_center[1]:
    h_minus = region_center[0]
    w_plus = region_center[1]
    h_plus = h_minus + region_len_h
    w_minus = w_plus - region_len_w

if i < region_center[0] and j >= region_center[1]:
    h_plus = region_center[0]
    w_minus = region_center[1]
    h_minus = h_plus - region_len_h
    w_plus = w_minus + region_len_w

if i < region_center[0] and j < region_center[1]:
    h_plus = region_center[0]
    w_plus = region_center[1]
    h_minus = h_plus - region_len_h
    w_minus = w_plus - region_len_w

```

Τέλος, η εύρεση των 4 γειτονικών contextual regions  $\{\mathcal{R}_{+,+}, \mathcal{R}_{+,-}, \mathcal{R}_{-,+}, \mathcal{R}_{-,-}\}$  και ο υπολογισμός της μετασχηματισμένης τιμής έντασης του pixel  $(i, j)$  εκτελούνται με τον παρακάτω κώδικα:

```

# find corresponding adjacent contextual regions
region_minus_minus = (h_minus - region_len_h // 2, w_minus - region_len_w // 2)
region_minus_plus = (h_minus - region_len_h // 2, w_plus - region_len_w // 2)
region_plus_minus = (h_plus - region_len_h // 2, w_minus - region_len_w // 2)
region_plus_plus = (h_plus - region_len_h // 2, w_plus - region_len_w // 2)

# compute factors for bi-linear interpolation between transforms
a = (j - w_minus) / (w_plus - w_minus)
b = (i - h_minus) / (h_plus - h_minus)

# compute level of pixel (i, j) in the output image

```

```

equalized_img[i, j] = np.round(
    (1 - a) * (1 - b) * region_to_eq_transform[region_minus_minus][img_array[i, j]] +
    (1 - a) * b * region_to_eq_transform[region_plus_minus][img_array[i, j]] +
    a * (1 - b) * region_to_eq_transform[region_minus_plus][img_array[i, j]] +
    a * b * region_to_eq_transform[region_plus_plus][img_array[i, j]]
)

```

όπου το pixel  $(i, j)$  της εικόνας εξόδου `equalized_img` τίθεται ίσο με την στρογγυλοποιημένη τιμή του μετασχηματισμού έντασης που υπολογίζεται μέσω της Σχέσης (13).

Η συνάρτηση τελικά επιστρέφει την μετασχηματισμένη εικόνα `equalized_img`.

### 3.3 Συνάρτηση `perform_no_interpolation_ahe`

#### 3.3.1 Function definition

```

def perform_no_interpolation_ahe(img_array: np.ndarray, region_len_h: int,
region_len_w: int) -> np.ndarray:

```

Σκοπός της συνάρτησης είναι να εφαρμόσει την Προσαρμοστική Εξισορρόπηση Ιστογράμματος στην εικόνα εισόδου για τις δεδομένες διαστάσεις των contextual regions της εικόνας, χωρίς τη χρήση διγραμμικής παρεμβολής μεταξύ των μετασχηματισμών έντασης γειτονικών contextual regions. Δηλαδή, η συνάρτηση υπολογίζει για το κάθε contextual region τον δικό του μετασχηματισμό έντασης και η μετασχηματισμένη τιμή έντασης ενός pixel προκύπτει αποκλειστικά από τον μετασχηματισμό έντασης της περιοχής στην οποία ανήκει.

- Η εικόνα εισόδου εισάγεται στη συνάρτηση με τη μορφή που περιγράφεται στην ενότητα 2.1.1.
- Η συνάρτηση δέχεται ως ορίσματα τις επιθυμητές διαστάσεις των contextual regions στις οποίες διαχωρίζεται η εικόνα, όπως περιγράφεται στην ενότητα 3.1.1.
- Η έξοδος της συνάρτησης είναι η εικόνα εξόδου του μετασχηματισμού και έχει μορφή όμοια με αυτή της εικόνας εισόδου, είναι δηλαδή μια 8-bit grayscale εικόνα ιδίου τύπου.

#### 3.3.2 Code explanation

Υπολογίζουμε τους μετασχηματισμούς έντασης κάθε contextual region της εικόνας μέσω της κλήσης της συνάρτησης `calculate_eq_transformations_of_regions` (περιγράφεται στην ενότητα 3.1.1):

```

region_to_eq_transform = calculate_eq_transformations_of_regions(img_array,
region_len_h, region_len_w)

```

Στη συνέχεια, διατρέχουμε κάθε pixel  $(i, j)$  της εικόνας εισόδου και εκτελούμε τα παρακάτω βήματα:

- Υπολογίζουμε την περιοχή στην οποία ανήκει το pixel όπως περιγράφηκε στην ενότητα 3.2.2:

```

region = ((i // region_len_h) * region_len_h, (j // region_len_w) * region_len_w)

```

- Εντοπίζουμε τον μετασχηματισμό έντασης της περιοχής αυτής:

```

eq_transform = region_to_eq_transform[region]

```

- Αντιστοιχίζουμε το pixel  $(i, j)$  της εικόνας εισόδου στην μετασχηματισμένη τιμή έντασής του μέσω του μετασχηματισμού `eq_transform` που υπολογίστηκε παραπάνω:

```

equalized_img[i, j] = eq_transform[img_array[i, j]]

```

Τέλος, επιστρέφεται η μεταβλητή `equalized_img`.

### 3.4 Παραδοχές

Για να εφαρμόσουμε την Προσαρμοστική Εξισορρόπηση Ιστογράμματος, τόσο με χρήση διγραμμικής παρεμβολής (στη συνάρτηση `perform_adaptive_hist_equalization`), όσο και χωρίς (στη συνάρτηση `perform_no_interpolation_ahe`), χρειάζεται να διαχωρίσουμε την εικόνα εισόδου σε περιοχές (contextual regions) σταθερών διαστάσεων. Στην παρούσα εργασία, θεωρούμε ότι οι διαστάσεις των περιοχών αυτών διαιρούν επακριβώς τις αντίστοιχες διαστάσεις της εικόνας εισόδου που δίνεται στις άνω συναρτήσεις. Για να εξασφαλίσουμε

αυτή τη θεώρηση, στο αρχείο επίδειξης των αποτελεσμάτων `demo.py` εισάγουμε ένα τμήμα κώδικα το οποίο ελέγχει για δεδομένες διαστάσεις ενός contextual region και διαστάσεις εικόνας εισόδου αν η θεώρησή μας είναι έγκυρη πριν καλέσουμε οποιαδήποτε συνάρτηση. Σε αντίθετη περίπτωση, φροντίζει έτσι ώστε οι διαστάσεις της εικόνας να είναι διαιρέσιμες από αυτές της περιοχής περικυπώντας το απαιτούμενο πλήθος στηλών ή γραμμών από την περιφέρεια της εικόνας.

### Code explanation

Ορίζουμε, αρχικά, την εικόνα εισόδου των συναρτήσεων `input_img_array` ίση με την εικόνα που επιθυμούμε να επεξεργαστούμε:

```
input_img_array = img_array
```

Αν καμία από τις παρακάτω συνθήκες ελέγχου δεν βρεθεί αληθής, τότε η εικόνα εισόδου των συναρτήσεων θα είναι η πρωτογενής εικόνα.

Οι συνθήκες που απαιτείται να ελεγχθούν είναι δύο:

- αν διαιρείται το ύψος της εικόνας από το ύψος των contextual regions
- αν διαιρείται το πλάτος της εικόνας από το πλάτος των contextual regions

Σε κάθε περίπτωση προβαίνουμε στην ανάλογη ενέργεια, όπως παρατίθεται παρακάτω.

```
# if image height is not divisible by contextual region height
h_res = m % region_len_h
if h_res != 0:
    # crop image height
    input_img_array = input_img_array[floor(h_res/2): m - ceil(h_res/2), :]
```

Αν το υπόλοιπο ( $h_{res}$ ) της διαίρεσης του ύψους της εικόνας προς επεξεργασία ( $m$ ) προς το ύψος του contextual region (`region_len_h`) είναι μη μηδενικό, τότε περικυπουμε το κατάλληλο πλήθος γραμμών από το άνω και κάτω άκρο της εικόνας, ενώ οι στήλες της εικόνας παραμένουν ανέπαφες. Συγκεκριμένα, αποκόπτουμε τις πρώτες  $\lfloor (h_{res}/2) \rfloor$  γραμμές και τις τελευταίες  $\lceil (h_{res}/2) \rceil$  γραμμές από την εικόνα. Επομένως, αποκόπτουμε γραμμές ίσες με το υπόλοιπο  $h_{res}$  στο πλήθος<sup>1</sup> και το πλήθος των γραμμών της εικόνας (ύψος  $m$ ) διαιρείται πλέον με το πλήθος γραμμών του contextual region (ύψος `region_len_h`).

```
# if image width is not divisible by contextual region width
w_res = n % region_len_w
if w_res != 0:
    # crop image width
    input_img_array = input_img_array[:, floor(w_res/2): n - ceil(w_res/2)]
```

Ομοίως, αν το υπόλοιπο ( $w_{res}$ ) της διαίρεσης του πλάτους της εικόνας προς επεξεργασία ( $n$ ) προς το πλάτος του contextual region (`region_len_w`) είναι μη μηδενικό, τότε περικυπουμε το κατάλληλο πλήθος στηλών από το αριστερό και δεξί άκρο της εικόνας, ενώ οι γραμμές της εικόνας παραμένουν ανέπαφες. Συγκεκριμένα, αποκόπτουμε τις πρώτες  $\lfloor (w_{res}/2) \rfloor$  στήλες και τις τελευταίες  $\lceil (w_{res}/2) \rceil$  στήλες από την εικόνα. Με αυτό τον τρόπο, αποκόπτουμε στήλες ίσες με το υπόλοιπο  $w_{res}$  στο πλήθος<sup>1</sup> και το πλήθος των στηλών της εικόνας (πλάτος  $n$ ) διαιρείται πλέον με το πλήθος στηλών του contextual region (πλάτος `region_len_w`).

### 3.5 Αποτελέσματα επεξεργασίας

Στην παρούσα υπο-ενότητα, παρουσιάζονται και σχολιάζονται τα αποτελέσματα της εφαρμογής την Προσαρμοστικής Εξισορρόπησης Ιστογράμματος στην εικόνα εισόδου που απεικονίζεται στο Σχήμα (1) και συγκρίνονται με αυτά της συμβατικής Εξισορρόπησης Ιστογράμματος. Τα αποτελέσματα αυτά παράγονται από το πρόγραμμα επίδειξης `demo.py`.

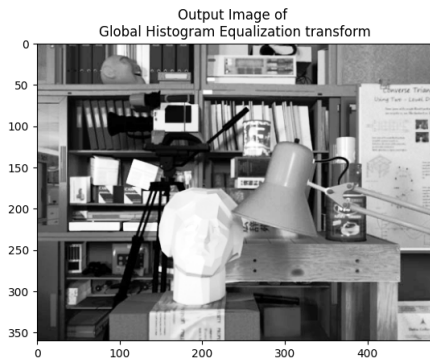
Παρακάτω παρατίθενται η εικόνα εισόδου σε σύγκριση με τις εικόνες εξόδου των δύο μεθόδων. Για την εφαρμογή της μεθόδου της Προσαρμοστικής Εξισορρόπησης Ιστογράμματος, επιλέχθηκαν διαστάσεις contextual regions ίσες

<sup>1</sup> Σημειώνεται ότι για έναν ακέραιο αριθμό  $x \in \mathbb{Z}$  ισχύει η ιδιότητα:

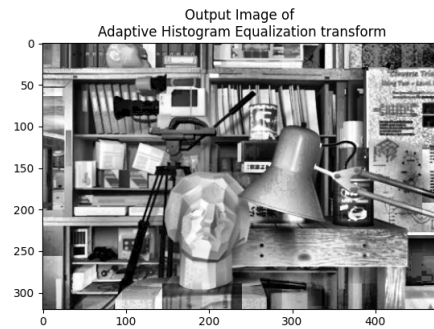
$$\lfloor \frac{x}{2} \rfloor + \lceil \frac{x}{2} \rceil = x$$



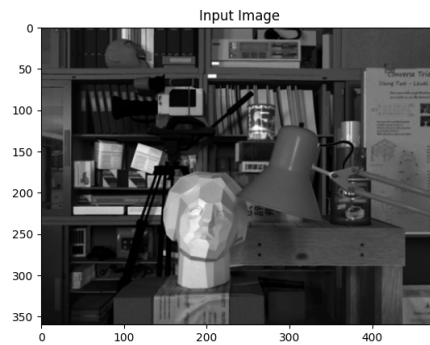
με  $64 \times 48$  για εικόνα εισόδου διαστάσεων  $360 \times 480$ . Παρατηρούμε ότι το ύψος της εικόνας εισόδου δεν διαιρείται επακριβώς με το ύψος των contextual regions. Αφού περικόψουμε την εικόνα, έτσι ώστε αυτή να έχει διαστάσεις συμβατές με αυτές των contextual regions (όπως περιγράφεται στην ενότητα 3.4), η εικόνα εισόδου έχει διαστάσεις  $320 \times 480$ .



(α') Output Image of Global Histogram Equalization transform



(β') Output Image of Adaptive Histogram Equalization transform

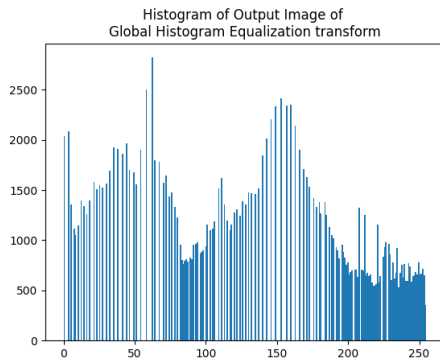


(γ') Input Image

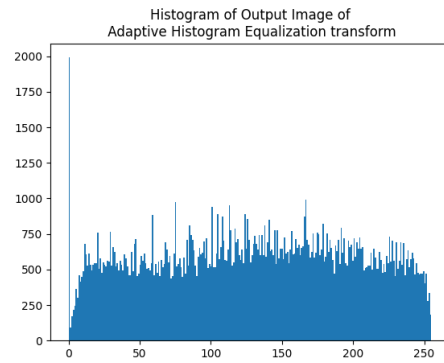
Σχήμα 10: Input and Output Images

Παρατηρούμε ότι η εικόνα εξόδου που παράγεται από τον μετασχηματισμό Προσαρμοστικής Εξισορρόπησης Ιστογράμματος έχει βελτιωμένη αντίθεση τόσο σε σύγκριση με την αρχική εικόνα, όσο και σε σύγκριση με την εικόνα εξόδου του μετασχηματισμού συμβατικής Εξισορρόπησης. Αυτό εξηγείται μέσω του γεγονότος ότι εκτελούμε τοπική επεξεργασία ιστογράμματος στην εικόνα και λαμβάνουμε υπόψη τις τοπικές κατανομές έντασης των pixels. Ως αποτέλεσμα, η αντίθεση μεταξύ των pixels του προσώπου, η οποία ήταν χαμηλή στην περίπτωση της συμβατικής εξισορρόπησης, οξύνεται και οδηγεί σε διακριτότερη λεπτομέρεια των σκιάσεων και των γωνιών του προσώπου. Ωστόσο, είναι εμφανή στην εικόνα εξόδου κάποια προβλήματα της εφαρμογής της μεθόδου αυτής. Το πρώτο πρόβλημα που διακρίνουμε είναι ότι τα pixels που ανήκουν στην κατηγορία των outer points (χόκκινη ζώνη στο Σχήμα (8)) εμφανίζουν ασυνέχειες στις τιμές έντασής τους στα όρια των γειτονικών contextual regions. Αυτό αποτελεί απόρροια του γεγονότος ότι δεν έχουμε εφαρμόσει γραμμική παρεμβολή ανάμεσα στους τοπικούς μετασχηματισμούς των γειτονικών περιοχών για τον υπολογισμό της τιμής έντασης των outer points για λόγους απλότητας της υλοποίησης. Ένα δεύτερο πρόβλημα που εντοπίζεται είναι η ανεπιθύμητη αύξηση της αντίθεσης σε ορισμένες περιοχές της εικόνας όπου παρατηρούνται χρωματικές διαβαθμίσεις. Για παράδειγμα, στο άνω δεξιό άκρο της εικόνας, λόγω της ιδιαίτερης υφής του ξύλου, οι τιμές έντασης της περιοχής του ντουλαπιού παρουσιάζουν μια ομαλή διαβάθμιση των τιμών του γκριζού. Ωστόσο, στην εικόνα εξόδου της Προσαρμοστικής Εξισορρόπησης λόγω της απεικόνισης των επιπέδων της τοπικής περιοχής με σκοπό την εξισορρόπηση, παρατηρούμε ότι αυτές οι διαβαθμισμένες τιμές έντασης της περιοχής απεικονίζονται σε τιμές έντασης με πολύ μεγαλύτερη αντίθεση μεταξύ τους. Ως εκ τούτου, χάνεται η ομαλότητα της διαβάθμισης και εμφανίζονται μεμονωμένες σκούρες κουκίδες στην περιοχή, οι οποίες αλλοιώνουν την αναπαράσταση της υφής της εικόνας. Επιπλέον, μια περιοχή της εικόνας που παρουσιάζει σημαντική αλλοίωση για τους προαναφερθέντες λόγους αποτελεί η περιοχή της αφίσας, όπου δεν είναι πλέον διακριτά τα γράμματα του κειμένου που απεικονίζεται πάνω σε αυτή, ενώ εμφανίζονται μεμονωμένες σκουρότερες κουκίδες πάνω στο λευκό υπόβαθρο του χαρτιού.

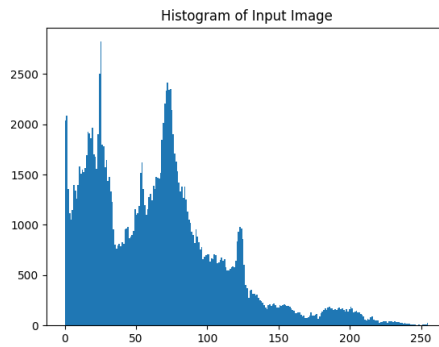
Παραθέτουμε, στη συνέχεια, τα ιστογράμματα των εικόνων εξόδου των δύο μεθόδων σε σύγκριση με την αρχική εικόνα.



(α') Histogram of Output Image of Global Histogram Equalization transform



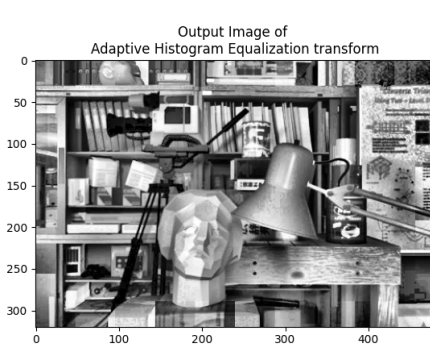
(β') Histogram of Output Image of Adaptive Histogram Equalization transform



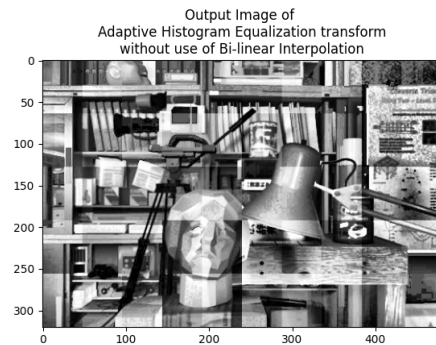
(γ') Histogram of Input Image

Σχήμα 11: Input and Output Images Histograms

Διαπιστώνουμε ότι το ιστογράμμο της εικόνας εξόδου που παράγεται από την Προσαρμοστική Εξισορρόπηση Ιστογράμματος φαίνεται να προσομοιάζει την ομοιόμορφη κατανομή περισσότερο από ότι το ιστογράμμο που παράγεται από την συμβατική Εξισορρόπηση. Οι συνιστώσες του ιστογράμματος παρουσιάζονται περισσότερο διασκορπισμένες και με μικρότερες αποκλίσεις στο ύψος τους. Αξιοσημείωτη είναι η αιχμή που εμφανίζεται στην μηδενική τιμή έντασης της εικόνας, η οποία έχει πλήθος εμφάνισης σημαντικά μεγαλύτερο από τις υπόλοιπες τιμές έντασης. Με σκοπό να αναδείξουμε την επίδραση της χρήσης της διγραμμικής παρεμβολής μεταξύ των τοπικών μετασχηματισμών εξισορρόπησης των contextual regions της εικόνας, αντιπαραθέτουμε παρακάτω τις εικόνες εξόδου που παράγονται από την Προσαρμοστική Εξισορρόπηση Ιστογράμματος με και χωρίς τη χρήση παρεμβολής, καθώς και τα ιστογράμμά τους.

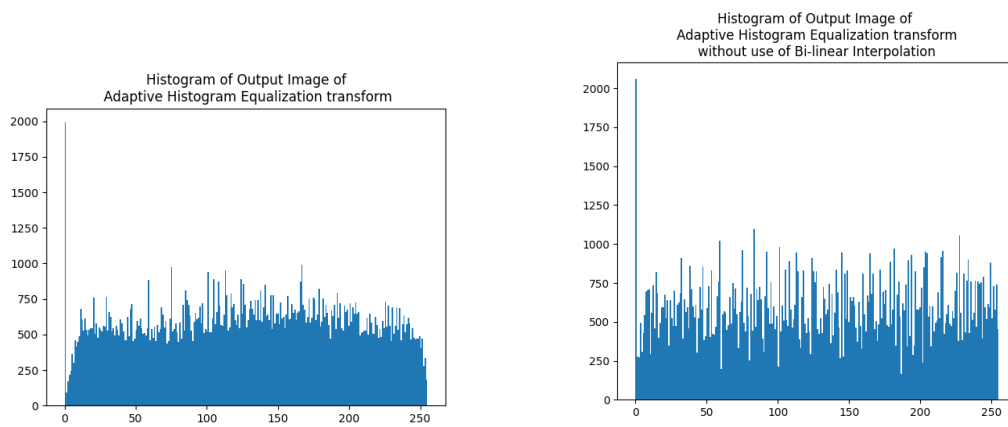


(α') Output Image of Adaptive Histogram Equalization transform



(β') Output Image of Adaptive Histogram Equalization transform without use of Bi-linear Interpolation

Σχήμα 12: Output Images of Adaptive Histogram Equalization transform



(α') Histogram of Output Image of Adaptive Histogram Equalization transform

(β') Histogram of Output Image of Adaptive Histogram Equalization transform without use of Bi-linear Interpolation

Σχήμα 13: Output Images Histograms

Διαπιστώνουμε ότι η εικόνα εξόδου που παράγεται χωρίς χρήση παρεμβολής παρουσιάζει ασυνέχειες στα όρια των contextual regions με αποτέλεσμα να φαίνεται διαχωρισμένη σε επιμέρους ορθογώνια τμήματα και όχι ως μια ενιαία εικόνα. Η ύπαρξη ασυνεχειών στις τιμές έντασης επιβεβαιώνεται και μέσω της παρατήρησης των ιστογραμμάτων των δύο εικόνων. Συγκεκριμένα, διαδοχικές συνιστώσες του ιστογράμματος της δεύτερης εικόνας παρουσιάζουν μεγαλύτερες αλλαγές του ύψους τους σε σύγκριση με τις συνιστώσες του ιστογράμματος της πρώτης εικόνας.