Ψηφιακή Επεξεργασία Εικόνας
-Εργασία 1: Παραλλαγές αλγορίθμων εξισορρόπησης και αντιστοίχισης ιστογράμματος

Α. Ντελόπουλος

Άνοιξη 2025

1 Εισαγωγικά

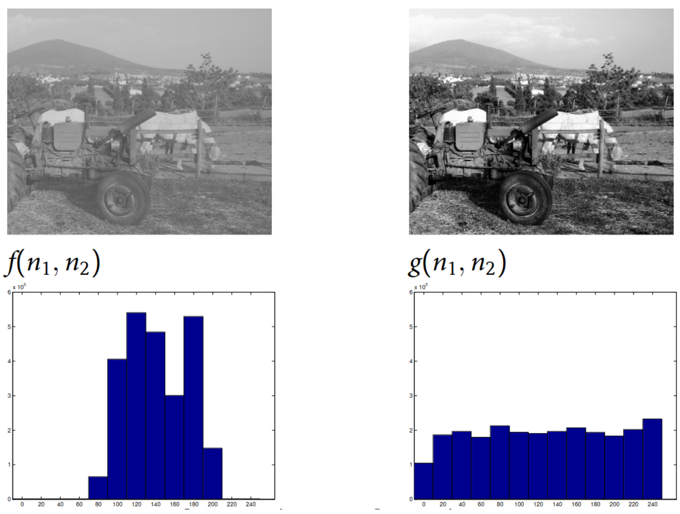
1.1 Εξισορρόπηση ιστογράματος

Θεωρούμε την εικόνα $f(n_1, n_2)$, που παίρνει τιμές στο διάστημα $(f_{\text{min}}, f_{\text{max}})$. Σύμφωνα με τη θεωρία, εάν η f λαμβάνει L στο πλήθος διακριτές τιμές, δηλαδή όταν $(f_{\text{min}}, f_{\text{max}}) = [f_0, f_1, ..., f_{L-1}]$, για το ιστόγραμμά της ισχύει ότι:

$$p_f(f_i) = \frac{\text{num. occurrences of } f_i}{\text{total num. of samples}} = \frac{\text{count}(f_i)}{N} \approx \text{Prob}(f = f_i)$$

Στην περίπτωση που η f είναι συνεχής, και θεωρώντας ότι το διάστημα $(f_{\text{min}}, f_{\text{max}})$ διαμερίζεται στα bins b_i με $b_i = [f_i, f_{i+1})$ (για κατάλληλα επιλεγμένα f_j), έχουμε:

$$p_f(b_i) = \frac{\text{num. of samples with values in } b_i}{\text{total num. of samples}} \approx \text{Prob}(f \in b_i)$$



Σχήμα 1: Παράδειγμα εξισορρόπησης ιστογράμματος: οι στάθμες της παραχθείσας εικόνας g (δεξιά) κατανέμονται κατά προσέγγιση ομοιόμορφα σε σύγκριση με την αρχική εικόνα f (αριστερά)

Γενικότερα, έστω η εικόνα $r = f(n_1, n_2)$ της οποίας οι τιμές κβαντίζονται σε L_r διαφορετικές στάθμες, $x_0 \leq x_1 \leq ... \leq x_{L-1}$, και $p_r(x_i)$ το ιστόγραμμα των τιμών της r, σύμφωνα με τα προηγούμενα. Η *επεξεργασία ιστογράμματος* αποσκοπεί στην παραγωγή μιας εικόνας $s = g(n_1, n_2)$ από την r, εφαρμόζοντας κατάλληλο μετασχηματισμό T, $s = T(r)$, τέτοιον ώστε το ιστόγραμμα των τιμών της s (έστω $y_i,\ i \in 0, 1, .., L_g - 1$ με L_g στο πλήθος στάθμες εξόδου) να παρουσιάζει κάποιες επιθυμητές ιδιότητες. Πρόκειται για ένα σύνολο μεθόδων που εφαρμόζονται στο γενικό πλαίσιο της διαδικασίας βελτίωσης εικόνας. Όταν απαιτείται το ιστόγραμμα εξόδου να προσεγγίζει μια άλλη δοθείσα κατανομή (για παράδειγμα όταν το ιστόγραμμα της εικόνας εξόδου καλείται να προσεγγίζει το ιστόγραμμα μιας άλλης εικόνας αναφοράς), τότε το πρόβλημα καλείται *αντιστοίχιση ιστογράμματος*. Ειδικότερα, όταν ο στόχος είναι το ιστόγραμμα των στάθμεων της εικόνας εξόδου να είναι προσεγγιστικά ομοιόμορφο, γίνεται λόγος για *εξισορρόπηση ιστογράμματος* (Σχήμα 2).

1.2 ”Άπληστη” εξισορρόπηση ιστογράματος (greedy histogram equalization)

Ας υποθέσουμε ότι δίνεται η εικόνα εισόδου $f \equiv f(n_1, n_2)$ με N στο πλήθος samples και L_f στάθμες τιμών $f_0, f_1, ..., f_{L_f-1}$, και ζητείται η παραγωγή μιας εικόνας $g \equiv g(n_1, n_2)$ με L_g στάθμες τιμών $g_0, g_1, ..., g_{L_g-1}$, της οποίας το ιστόγραμμα να είναι εξισορροπημένο.

Διευκρινίζεται ότι για διευκόλυνσή σας, στα πλαίσια της εργασίας, εάν η g λαμβάνει τιμές στο διάστημα $(g_{\text{min}}, g_{\text{max}})$, οι στάθμες $g_0, g_1, ..., g_{L_g-1}$ επιλέγονται ώστε να ισαπέχουν μεταξύ τους, εντός αυτού του διαστήματος. Στόχος λοιπόν της εξισορρόπησης ιστογράμματος στο πρόβλημα αυτό, είναι να ανατεθούν N/L_g samples σε κάθε στάθμη g_i.

Μια greedy προσέγγιση επίλυσης, είναι η εξής: ”σαρώνουμε” προοδευτικά τις στάθμες εισόδου $\{f_0, f_1, ...\}$, αναθέτοντας το μικρότερο δυνατό υποσύνολο αυτών στη στάθμη εξόδου g_0, ώστε $\text{count}(g_0) \geq N/L_g$. Ισοδύναμα, για την πρώτη στάθμη εξόδου, αναζητούμε το j τέτοιο ώστε:

$$j = \text{argmin}_i \sum_{k=0}^{i} \text{count}(f_k), \quad \text{s.t.} \sum_{k=0}^{i} \text{count}(f_k) \geq \frac{N}{L_g}$$

Στη συνέχεια επαναλαμβάνεται η ίδια διαδικασία για τις εναπομείνασες διαθέσιμες στάθμες εισόδου $f_j, f_{j+1}, ...$, και την επόμενη στάθμη εξόδου g_1, κ.ο.κ.

1.3 Μη-άπληστη εξισσορρόπηση ιστογράματος (non-greedy histogram equalization)

Είναι εύκολο να αντιληφθεί κανείς ότι η προηγούμενη προσέγγιση μπορεί να παρεκκλίνει αρκετά από την επιθυμητή ομοιόμορφη κατανομή στο εξισορροπημένο ιστόγραμμα. Αυτό συμβαίνει όταν στο ιστόγραμμα της αρχικής εικόνας υπάρχουν στάθμες με αρκετά μεγαλύτερο count σε σχέση με άλλες.

Μια βελτίωση στην παραπάνω λογική, εντοπίζεται στην απόφαση του πότε μια στάθμη εισόδου θα αντιστοιχίζεται πράγματι σε μια δεδομένη στάθμη εξόδου. Ειδικότερα, ακόμα και στην περίπτωση που η απεικόνιση στάθμεων εισόδου σε μια στάθμη εξόδου οδηγεί σε count $< N/L_g$, εάν εντούτοις αυτό το έλειμμα είναι επαρκώς μικρό (εντός κάποιου κατωφλίου), η επόμενη διαθέσιμη στάθμη εισόδου δεν εκχωρείται στην τρέχουσα στάθμη εξόδου (σε αντίθεση με την προηγούμενη περίπτωση).

Για παράδειγμα, εξετάζοντας τη στάθμη εξόδου g_0, οι υποψήφιες στάθμες εισόδου είναι οι $f_0, f_1, f_2...$. Η στάθμη εισόδου f_0 αντιστοιχίζεται στη g_0, ενώ για κάθε άλλη υποψήφια στάθμη εισόδου $j = 1, 2, ...$, υπολογίζουμε την ποσότητα:

$$\text{deficiency}_j = \frac{N}{L_g} - \sum_{i=0}^{j-1} \text{count}(f_i)$$

Εάν για την υποψήφια στάθμη j, είναι $\text{deficiency}_j \geq \text{count}(f_j)/2$, τότε η στάθμη f_j απεικονίζεται και αυτή στη στάθμη εξόδου g_0. Ομοίως συνεχίζουμε με τη στάθμη εξόδου g_1 και τις εναπομείνασες στάθμες εισόδου.

1.4 Εξισσορρόπηση ιστογράματος με προσθήκη θορύβου (post-disturbance histogram equalization)

Το πρόβλημα και με τις δύο προηγούμενες προσεγγίσεις, είναι ότι μια στάθμη εισόδου εκχωρείται πάντα *ολόκληρη* σε μια στάθμη εξόδου. Όπως έχει αναφερθεί, αυτό σημαίνει ότι μια πολυπληθής στάθμη εισόδου μπορεί να οδηγήσει σε απόκλιση του ιστογράμματος εξόδου από την επιθυμητή κατανομή, δεδομένου του ότι όλα τα samples αυτής θα απεικονιστούν στην ίδια τιμή εξόδου.

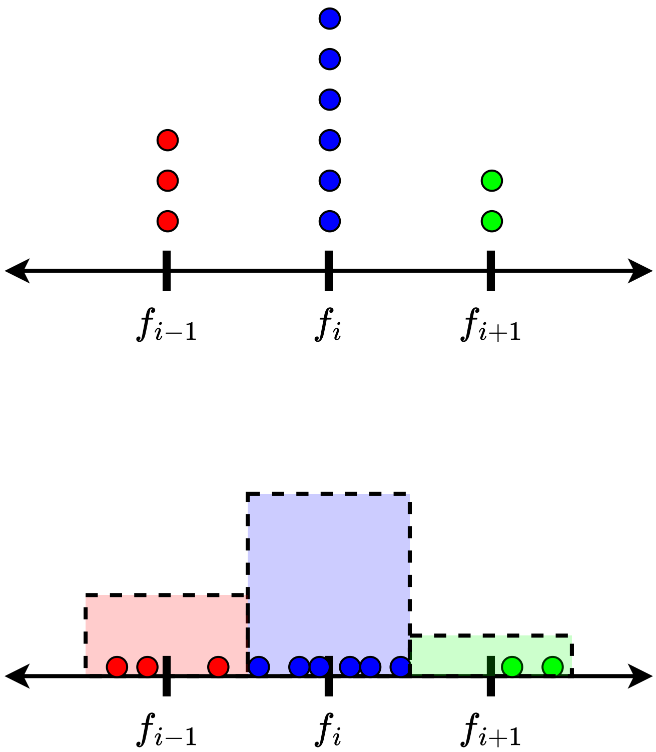
Ας θεωρήσουμε μια εικόνα $f(n_1, n_2)$ που λαμβάνει τις L_f διακριτές τιμές $f_0, f_1, ..., f_{L_f-1}$. Ακόμη, στα πλαίσια της εργασίας θεωρούμε ότι $\Delta f = f_{i+1} - f_i := d$ είναι σταθερό για κάθε i (δηλαδή οι στάθμες εισόδου είναι ισαπέχουσες).

Έστω τώρα η εικόνα $v(n_1, n_2)$, τα δείγματα της οποίας έχουν προέλθει από random uniform sampling στο διάστημα $\left[-\frac{d}{2}, \frac{d}{2}\right]$, όπου d είναι η απόσταση των διαδοχικών τιμών εισόδου.

Σχηματίζουμε την εικόνα:

$$\hat{f}(n_1, n_2) = f(n_1, n_2) + v(n_1, n_2)$$

η οποία λαμβάνει τιμές στο διάστημα $I := \left[f_0 - \frac{d}{2}, f_{L_f+1} + \frac{d}{2}\right]$ Με τον τρόπο αυτό, έχει προκύψει μια εικόνα με το ίδιο πλήθος samples με την αρχική, όμως με μεγαλύτερο πλήθος διαφορετικών τιμών από τις L_f στάθμες της f. Συνεπώς, έχει εξαλειφθεί η αρχική δυσκολία, όπου όλα τα samples μιας στάθμης αντιστοιχίζονται σε μια στάθμη εξόδου. Πλέον, εφαρμόζοντας τη greedy προσέγγιση στην εικόνα $\hat{f}$, μπορεί κανείς να επιλέγει κάθε φορά το επιθυμητό πλήθος τιμών εισόδου που να αντιστοιχίζονται σε μια στάθμη εξόδου, οδηγώντας σε πιο ισορροπημένο ιστόγραμμα.



Σχήμα 2: Επίδραση της προσθήκης θορύβου στην εικόνα εισόδου: οι διακριτές τιμές-στάθμες (επάνω) διασκορπίζονται εντός ενός διαστήματος με κέντρο την κάθε τιμή-σταθμη (κάτω)

1.5 Αντιστοίχιση ιστογράμματος (histogram matching)

Στην περίπτωση του προβλήματος histogram matching, δίνεται μια κατανομή αναφοράς $p_{\text{ref}}(g_i),\ i = 0, 1, ..., L_g$ για την εικόνα εξόδου, g. Οι αλγόριθμοι που περιγράφηκαν παραπάνω τροποιούνται εύκολα, λαμβάνοντας υπόψη ότι το επιθυμητό πλήθος δειγμάτων της στάθμης g_i πλέον θα ισούται με $N \cdot p_{\text{ref}}(g_i)$ (αντί του προηγούμενου N/L_g).

2 Παραδοτέα

Για τις ανάγκες της εργασίας θα πρέπει να κατασκευάσετε

1. Ένα σύνολο συναρτήσεων σε Python που θα υλοποιούν τις διαδικασίες της εξισορρόπησης και αντιστοίχισης ιστογράμματος
2. Ένα πρόγραμμα επίδειξης με το όνομα `demo.py`
3. Ένα report που θα εξηγείτε τη λειτουργία των συναρτήσεων, θα επιδεικνύετε και θα σχολιάζετε τα αποτελέσματα και θα περιγράφετε πιθανές παραδοχές που κάνατε κατά την υλοποίηση

Διευκρινίζεται ότι μπορείτε να χρησιμοποιήσετε κάθε supported έκδοση της Python (3.8 - 3.12), καθώς και να εισαγάγετε δικές σας επιμέρους βοηθητικές μεθόδους στην υλοποίηση, αρκεί όλα αυτά να καταγραάφονται στο report σας.

2.1 `hist_utils.py`

Να κατασκευάσετε συνάρτηση

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

Ορίσματα:

- `img_array`: ένας δισδιάστατος `numpy array` τύπου `dtype=float`, που αναπαριστά μια grayscale εικόνα με τιμές στο διάστημα $[0, 1]$

Έξοδοι:

- `equalization_transform`: ένα dictionary, με κάθε key να είναι μια στάθμη εισόδου f_i, και value να είναι η στάθμη εξόδου g_j στην οποία εκείνη αντιστοιχίζεται.

Λειτουργία: η συνάρτηση δέχεται την εικόνα εισόδου, υπολογίζει το ιστόγραμμα των τιμών αυτής, και στη συνέχεια το μετασχηματισμό (συμβατικής) εξισορρόπησης ως ένα python dictionary `equalization_transform`. Επομένως, η στάθμη εισόδου f_i, όπου $i = 0, 1, ..., L-1$, θα απεικονίζεται στην τιμή του στοιχείου `equalization_transform[f_i]`.

Σημειώνεται, ότι δεν επιτρέπεται η απευθείας χρήση έτοιμων συναρτήσεων υπολογισμού ιστογράμματος (όπως για παράδειγμα η `numpy.histogram(.)` κλπ). Ακόμη, θεωρήστε ότι οι στάθμες εισόδου είναι πάντοτε οι συνολικές διαφορετικές (unique) τιμές της εικόνας εισόδου.

Επιπλέον, να υλοποιηθεί η συνάρτηση:

```python
def perform_hist_equalization(
    img_array: np.ndarray,
) -> np.ndarray:
```

Ορίσματα:

- `img_array`: ένας δισδιάστατος `numpy array` τύπου `dtype=float`, που αναπαριστά μια grayscale εικόνα

Έξοδοι:

- `equalized_img`: ένας δισδιάστατος `numpy array` τύπου `dtype=float`, που αναπαριστά την εικόνα εξόδου

Λειτουργία: η συνάρτηση δέχεται την εικόνα εισόδου, χρησιμοποιεί κατάλληλα την προηγούμενη μέθοδο, κι εφαρμόζει τον μετασχηματισμό εξισορρόπησης.

Για να διαβάσετε την εικόνα εισόδου, μπορείτε να χρησιμοποιήσετε το πακέτο `pillow` ως εξής:

```python
from PIL import Image
import numpy as np

// set the filepath to the image file
filename = "img1.jpg"

// read the image into a PIL entity
img = Image.open(fp=filename)

// keep only the Luminance component of the image
bw_img = img.convert("L")
```

```
// obtain the underlying np array
img_array = np.array(bw_img).astype(float) / 255.0
```

2.2 hist_modif.py

Η συνάρτηση που θα αποτελέσει τον κορμό των υπόλοιπων ζητούμενων είναι η εξής:

```
def perform_hist_modification(
    img_array: np.ndarray,
    hist_ref: Dict,
    mode: str
) -> np.ndarray
```

Ορίσματα:

- img_array: ένας δισδιάστατος numpy array τύπου dtype=float που αναπαριστά μια grayscale εικόνα με τιμές στο διάστημα $[0, 1]$
- hist_ref: ένα Python dictionary που αναπαριστά την επιδιωκώμενη κατανομή ιστογράμματος της εικόνας εξόδου. Ως key είναι κάθε στάθμη εξόδου, και ως value η σχετική συχνότητα εμφάνισης της στάθμης αυτής στην έξοδο.
- mode: η επιλογή αλγορίθμου επεξεργασίας για την παραγωγή της εικόνας εξόδου. Δυνατές τιμές: "greedy", "non-greedy", "post-disturbance"

Έξοδοι:

- modified_img: ένας δισδιάστατος numpy array τύπου dtype=float, που αναπαριστά την εικόνα εξόδου, της οποίας το ιστόγραμμα προσεγγίζει το δοθέν hist_ref

Λειτουργία: η συνάρτηση δέχεται την εικόνα εισόδου, και υπολογίζει το μετασχηματισμό εξισορρόπησης αυτής, equalization_transform με τη χρήση των βοηθητικών συναρτήσεων της προηγούμενης ενότητας. Το σύνολο των στάθμεων εισόδου είναι το σύνολο των διαφορετικών τιμών που λαμβάνει εκείνη. Στη συνέχεια, και με βάση την παρατήρηση της υποενότητας 1.5, υλοποιείται histogram matching με target κατανομη το δοθέν hist_ref. Η υλοποίηση στο σώμα της συνάρτησης θα ακολουθεί καθεμία από τις παραπάνω προσεγγίσεις, αναλόγως την τιμή του ορίσματος mode.

Σε επόμενο βήμα, ζητείται να υλοποιηθεί η συνάρτηση:

```
def perform_hist_eq(
    img_array: np.ndarray,
    mode: str
) -> np.ndarray
```

Ορίσματα:

- img_array: ένας δισδιάστατος numpy array τύπου dtype=float, διαστάσεων $[H, W]$ που αναπαριστά μια grayscale εικόνα με τιμές στο διάστημα $[0, 1]$.
- mode: η προσέγγιση που θα πρέπει να χρησιμοποιηθεί για την εξισορρόπηση ιστογράμματος της εικόνας img_array. Δυνατές τιμές: "greedy", "non-greedy", "post-disturbance"

Έξοδοι:

- `equalized_img`: ένας δισδιάστατος `numpy array` τύπου `dtype=float`, που αναπαριστά την εικόνα εξόδου

Λειτουργία: η συνάρτηση δέχεται την εικόνα εισόδου, και χρησιμοποιεί τη συνάρτηση `perform_hist_modification(.)` αφού πρώτα ορίσει κατάλληλα το όρισμα-επιθυμητό ιστόγραμμα της τελευταίας.

Επιπλέον, ζητείται να υλοποιηθεί η συνάρτηση:

```
def perform_hist_matching(
    img_array: np.ndarray,
    img_array_ref: np.ndarray,
    mode: str
) -> np.ndarray
```

Ορίσματα:

- `img_array`: ένας δισδιάστατος `numpy array` τύπου `dtype=float`, που αναπαριστά μια grayscale εικόνα εισόδου
- `img_array_ref`: ένας δισδιάστατος `numpy array` τύπου `dtype=float`, που αναπαριστά μια grayscale εικόνα της οποίας το ιστόγραμμα καλείται να προσεγγίσει η τελική επεξεργασμένη εικόνα
- `mode`: η προσέγγιση που θα πρέπει να χρησιμοποιηθεί για τη διαδικασία αντιστοίχισης ιστογράμματος. Δυνατές τιμές: `"greedy"`, `"non-greedy"`, `"post-disturbance"`

Έξοδοι:

- `processed_img`: ένας δισδιάστατος `numpy array` τύπου `dtype=float`, που αναπαριστά την εικόνα εξόδου

Λειτουργία: η συνάρτηση δέχεται την εικόνα εισόδου f, και την εικόνα αναφοράς f_{ref} (της οποίας το ιστόγραμμα καλείται να προσεγγίσει η εικόνα εξόδου), και καλεί κατάλληλα τη συνάρτηση `perform_hist_modification(.)`.

2.3 `demo.py`

Με το script `demo.py` να επιδείξετε τη λειτουργία των ζητούμενων συναρτήσεων, χρησιμοποιώντας την εικόνα εισόδου και αναφοράς, input_img.png και ref_img.png, αντίστοιχα, που σας δίνεται. Στο πρόγραμμά σας εκείνη θα διαβάζεται ως grayscale εικόνα, με τον τρόπο που παρουσιάστηκε σε προηγούμενη ενότητα. Εστιάστε στις εξής πτυχές της μελέτης:

- Παρουσιάστε την τελική (κατά τη διαδικασία εξισορρόπησης ιστογράμματος) εικόνα σε σύγκριση με την αρχική, μαζί με τα αντίστοιχα ιστογράμματά τους, για κάθε δυνατή προσέγγιση που έχει αναλυθεί. Σχολιάστε τα αποτελέσματα.
- Παρουσιάστε την τελική (κατά τη διαδικασία αντιστοίχισης ιστογράμματος) εικόνα σε σύγκριση με την εικόνα εισόδου, μαζί με τα αντίστοιχα ιστογράμματά τους, για κάθε δυνατή προσέγγιση που έχει αναλυθεί. Σχολιάστε τα αποτελέσματα, με βάση και την ίδια την εικόνα αναφοράς που σας έχει δοθεί.

Οι εικόνες που θα περιλάβετε στην αναφορά της εργασίας θα πρέπει να παραχθούν από το script αυτό. Για κάθε εικόνα εξόδου, το `demo.py` πρέπει να παράγει και τα αντίστοιχα ιστογράμματα.

3 Για την υποβολή της εργασίας

Παραδώστε μία αναφορά με τις περιγραφές και τα συμπεράσματα που σας ζητούνται στην εκφώνηση. Η αναφορά θα πρέπει να επιδεικνύει την ορθή λειτουργία του κώδικά σας στην εικόνα που σας δίνεται και να παρουσιάζει και σχολιάζει τις εικόνες και τα ιστογράμματα που παράγονται από το πρόγραμμα επίδειξης.

Ο κώδικας θα πρέπει να είναι σχολιασμένος ώστε να είναι κατανοητό τι ακριβώς λειτουργία επιτελεί (σε θεωρητικό επίπεδο, όχι σε επίπεδο κλίσης συναρτήσεων). Επίσης, ο κώδικας θα πρέπει να εκτελείται και να υπολογίζει τα σωστά αποτελέσματα για *οποιαδήποτε* είσοδο πληροί τις υποθέσεις της εκφώνησης, και όχι μόνο για την εικόνα που σας δίνεται.

Απαραίτητες προϋποθέσεις για την βαθμολόγηση της εργασίας σας είναι ο κώδικας να εκτελείται χωρίς σφάλμα (μόνο demos που εκτελούνται επιτυχώς θα βαθμολογηθούν), καθώς και να τηρούνται τα ακόλουθα:

- Υποβάλετε ένα και μόνο αρχείο, τύπου zip.
- Το όνομα του αρχείου πρέπει να είναι `AEM.zip`, όπου `AEM` είναι τα τέσσερα ψηφία του Α.Ε.Μ. του φοιτητή της ομάδας.
- Το προς υποβολή αρχείο πρέπει να περιέχει τα αρχεία κώδικα Python και το αρχείο `report.pdf` το οποίο θα είναι η αναφορά της εργασίας.
- Η αναφορά πρέπει να είναι ένα αρχείο τύπου PDF, και να έχει όνομα `report.pdf`.
- Όλα τα αρχεία κώδικα πρέπει να είναι αρχεία κειμένου τύπου UTF-8, και να έχουν κατάληξη `m`.
- Το αρχείο τύπου zip που θα υποβάλετε δεν πρέπει να περιέχει κανέναν φάκελο.
- Μην υποβάλετε τις εικόνες που σας δίνονται για πειραματισμό.
- Μην υποβάλετε αρχεία που δεν χρειάζονται για την λειτουργία του κώδικά σας, ή φακέλους/αρχεία που δημιουργεί το λειτουργικό σας, πχ “Thumbs.db”, “.DS_Store”, “.directory”.
- Για την ονομασία των αρχείων που περιέχονται στο προς υποβολή αρχείο, χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ “#”, “$”, “%” κλπ.