

Ψηφιακή Επεξεργασία και Ανάλυση
Εικόνας
1η Εργασία

Βεργίνης Δημήτριος
1066634

Περιεχόμενα

1	Εισαγωγή	2
2	-Άσκηση 1	3
2.1	Οι βιβλιοθήκες της άσκησης	5
3	-Άσκηση 2	6
3.1	Μέθοδος ζώνης	6
3.2	Μέθοδος Κατωφλίου	6
3.3	Οι βιβλιοθήκες της άσκησης	9
4	-Άσκηση 3	9
4.1	Οι βιβλιοθήκες της άσκησης	13
5	-Άσκηση 4	14
5.1	Οι βιβλιοθήκες της άσκησης	17
6	-Άσκηση 5	17
6.1	Μέρος Α	17
6.2	Οι βιβλιοθήκες της άσκησης	18
6.3	Μέρος Β	19
7	-Άσκηση 6	21
7.1	Ιστογράμματα	21
7.2	Μετασχηματισμός Συνημιτόνου	23
7.3	Οι βιβλιοθήκες της άσκησης	25

1 Εισαγωγή

Για την υλοποίηση των ασκήσεων χρησιμοποιήθηκε το περιβάλλον της Python με χρήση διαφόρων βιβλιοθηκών οι οποίες όμως παρουσιάζονται μαζί με την αναφορά για την εκάστοτε άσκηση. Για την υλοποίηση της άσκησης 5β έγινε χρήση του περιβάλλοντος Matlab.

2 - Ασκηση 1

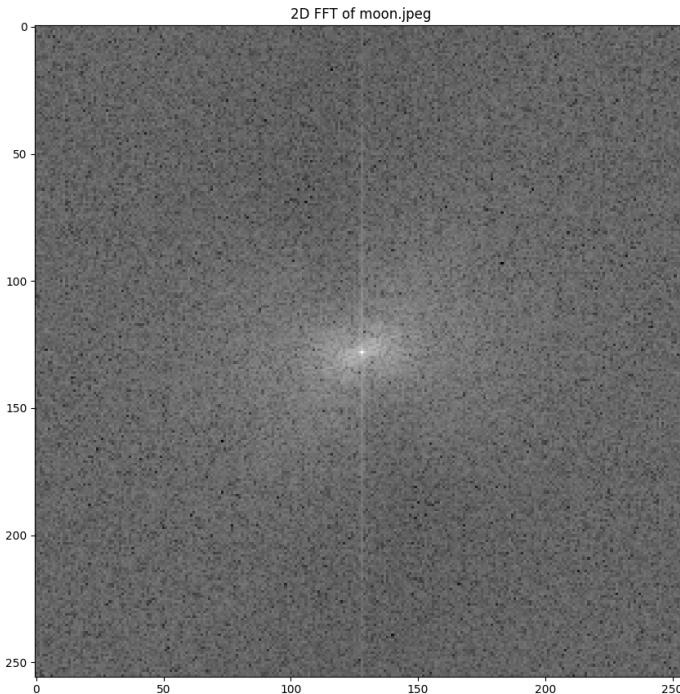
Αρχικά φορτώνουμε την εικόνα και εφαρμόζουμε min-max scaling έτσι ώστε να αλλάξουμε το εύρος των τιμών στο 0-255. Αυτό γίνεται μέσω της διαδικασίας:

$$x_{std} = \left(\frac{x - x_{min}}{x_{max} - x_{min}} \right)$$

$$x_{scaled} = x_{std} * (max - min) + min$$

Στη συνέχεια πολλαπλασιάζουμε την εικόνα με το $(-1)^{(x+y)}$ έτσι ώστε εκμεταλλευόμενοι την ιδιότητα του DFT να μεταφερθούν οι συχνότητες από το (0,0) στο κέντρο της εικόνας.

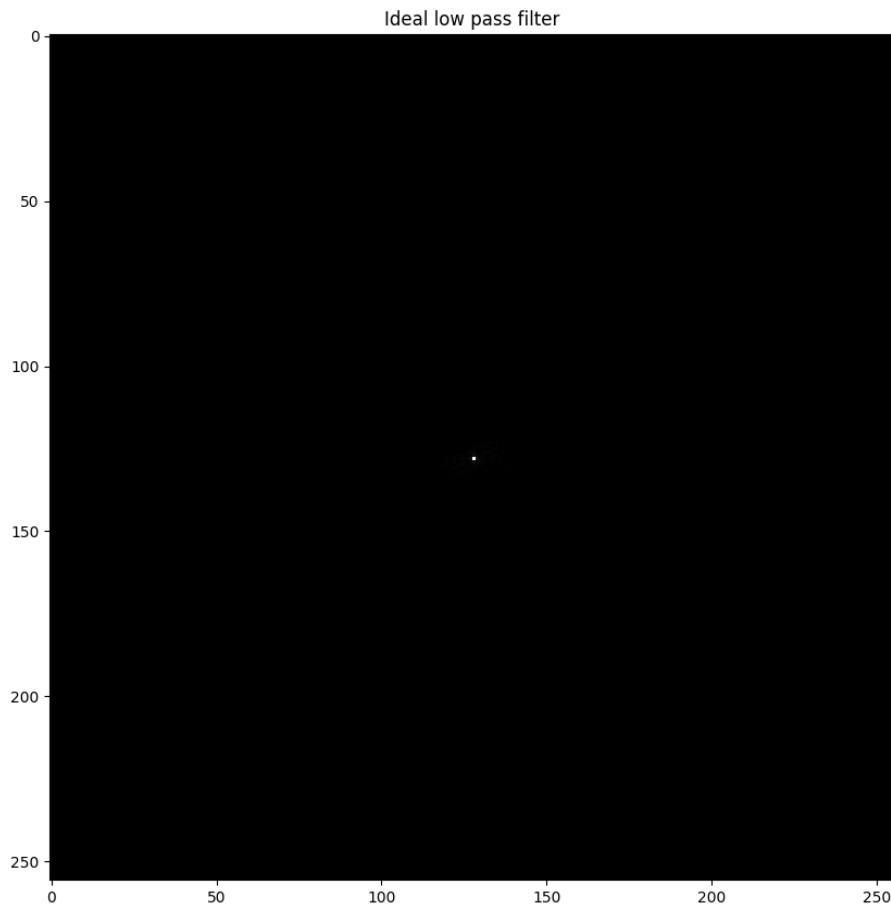
Στη συνέχεια εφαρμόζουμε τον μετασχηματισμό 2DFFT κάνοντας χρήση της μεθόδου γραμμών-στηλών και λαμβάνουμε τα παρακάτω αποτελέσματα:



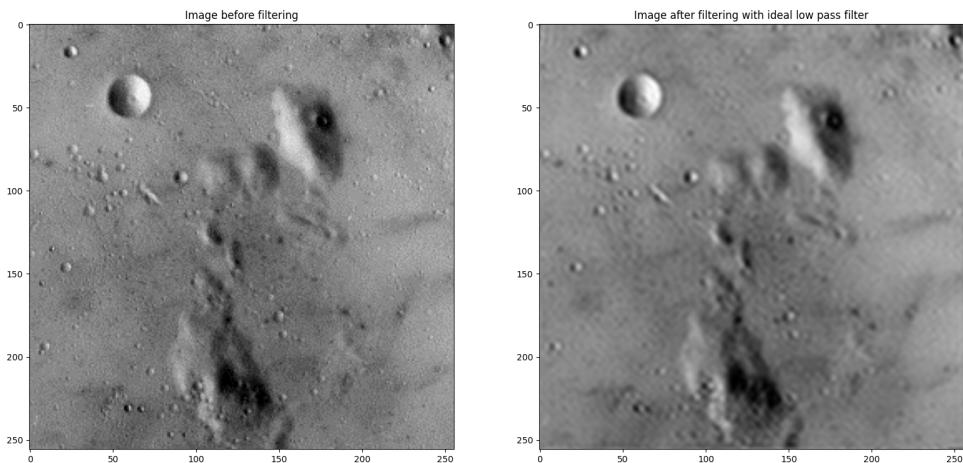
Για το φιλτράρισμα της εικόνας, εφαρμόστηκε ιδανικό κατωπερατό τετραγωνικό φίλτρο

$$H(u, v) = \begin{cases} 1, & D_o < 50 \\ 0, & \alpha \lambda \lambda o v \end{cases}$$

η μορφή του οποίου φαίνεται παραχώτω



Τοπέρα εφαρμόζουμε τον αντίστροφο μετασχηματισμό Fourier για την εικόνα κάνοντας πάλι χρήση τη μέθοδο γραμμών-στηλών και εφαρμόζουμε ξανά την ιδιότητα του DFT πολλαπλασιάζοντας με $(-1)^{(x+y)}$. Παρακάτω βλέπουμε δύο εικόνες, η δεξιά είναι η εικόνα πριν υποστεί το φιλτράρισμα και η αριστερή αφού εφαρμοστεί το ιδανικό κατωπερατό φίλτρο.



Παρατηρούμε ότι έχοντας αφαιρέσει τις υψηλές συχνότητες η φιλτραρισμένη ουδόνη φαίνεται αρκετά θολή.

2.1 Οι βιβλιοθήκες της άσκησης

Οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση αυτής της άσκησης ήταν :

- **Numpy**, για αριθμητικούς υπολογισμούς με πίνακες
- **Matplotlib**, για την απεικόνιση των εικόνων
- **Pillow**, για τη φόρτωση της εικόνας

3 - Άσκηση 2

3.1 Μέθοδος ζώνης

Για τη μέθοδο ζώνης ακολουθήθηκαν τα εξής βήματα για τη συμπίεση της εικόνας. Αρχικά για τη διευκόλυνση της υλοποίησης αρχικά τα τρία κανάλια R, G, B επεξεργάστηκαν ακολουθώντας την ίδια μέθοδο και μετά έγινε η συνένωση στην τελική εικόνα.

1. Η εφαρμογή του δισδιάστατου μετασχηματισμού συνημίτονου 2D-DCT σε κάθε ένα από τα 32x32 μπλόκ που χωρίστηκε η εικόνα
2. Με τη μέθοδο της ζώνης και διασχίζοντας τους πίνακες με τεχνική ζιγκ-ζαγκ διατηρήθηκαν τα γ στοιχεία του πίνακα και τα υπόλοιπα εκμηδενίστηκαν
3. Εφαρμόζοντας τώρα τον αντίστροφο μετασχηματισμό συνημίτονου 2D-DCT σε κάθε ένα από τα μπλόκ επαναφέρθηκαν τα συμπιεσμένα τώρα κανάλια R, G, B στην προηγούμενη τους μορφή
4. Τέλος έγινε συνένωσή των καναλιών για τον σχηματισμό της συμπιεσμένης εικόνας

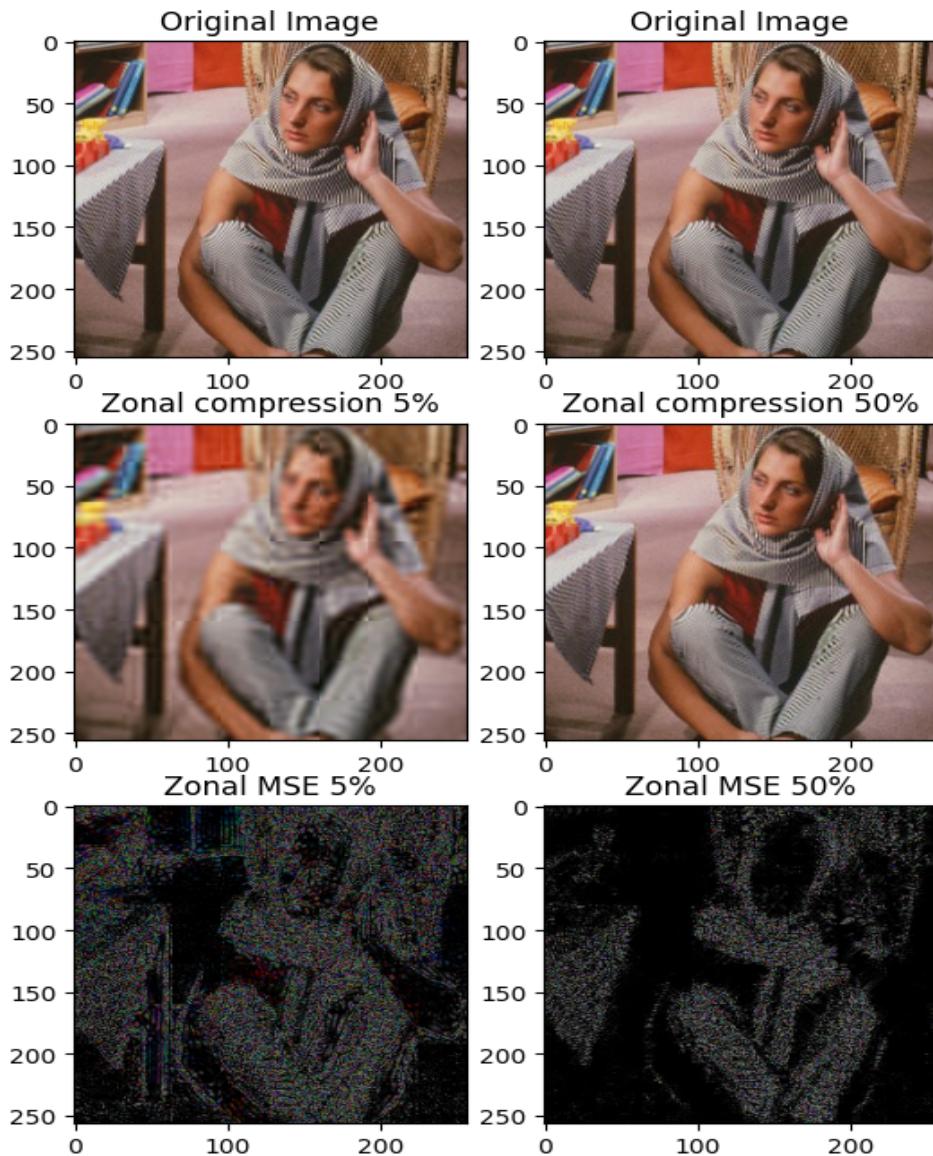
3.2 Μέθοδος Κατωφλίου

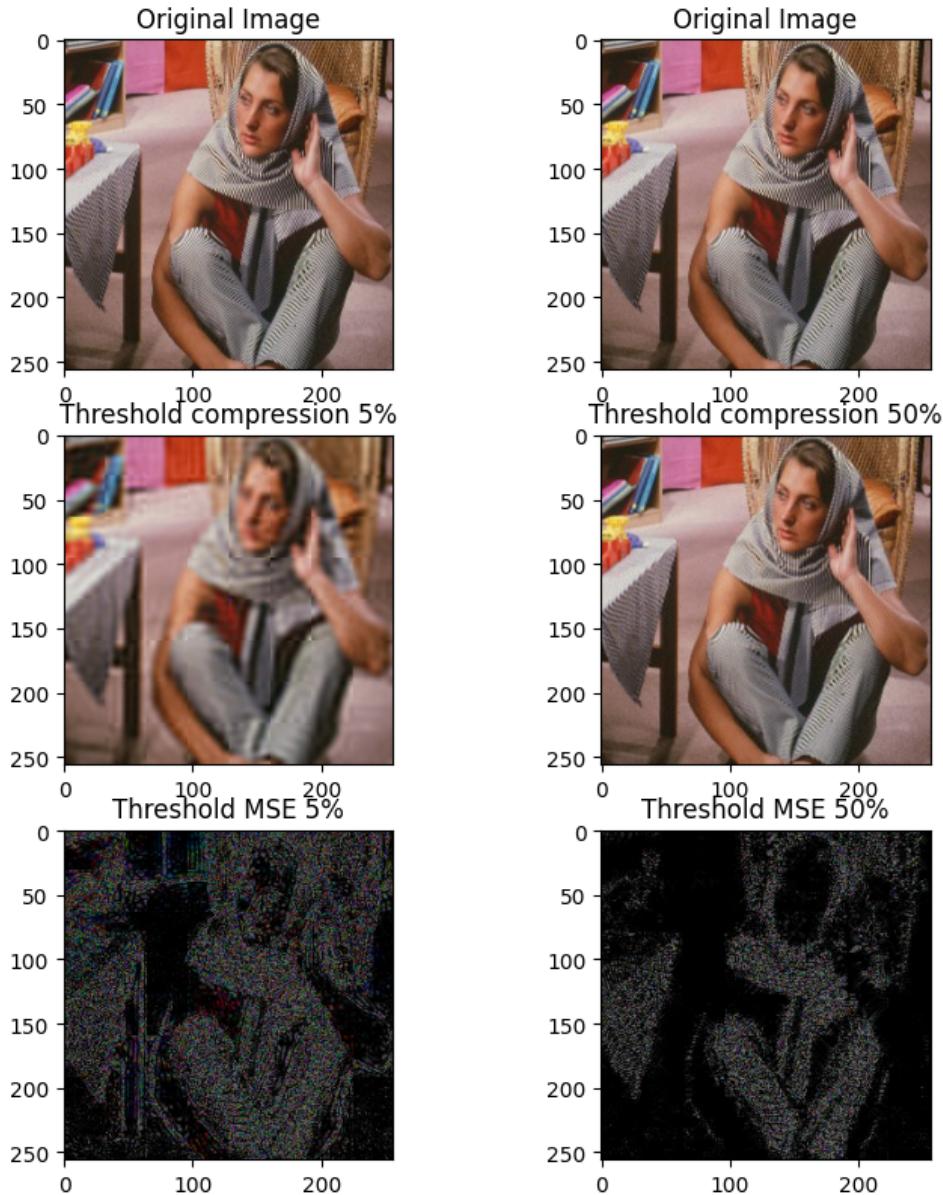
Για τη μέθοδο κατωφλίου ακολουθήθηκαν τα εξής βήματα για τη συμπίεση της εικόνας. Αρχικά για τη διευκόλυνση της υλοποίησης αρχικά τα τρία κανάλια R, G, B επεξεργάστηκαν ακολουθώντας την ίδια μέθοδο και μετά έγινε η συνένωση στην τελική εικόνα.

1. Η εφαρμογή του δισδιάστατου μετασχηματισμού συνημίτονου 2D-DCT σε κάθε ένα από τα 32x32 μπλόκ που χωρίστηκε η εικόνα
2. Εφαρμόζουμε τη μέθοδο κατωφλίου, σύμφωνα με την οποία διατηρήθηκαν τα γ μεγαλύτερα πιξελ από το εκάστοτε μπλοκ και εκμηδενίστηκαν τα υπόλοιπα
3. Εφαρμόζοντας τώρα τον αντίστροφο μετασχηματισμό συνημίτονου 2D-DCT σε κάθε ένα από τα μπλόκ επαναφέρθηκαν τα συμπιεσμένα τώρα κανάλια R, G, B στην προηγούμενη τους μορφή

- Τέλος έγινε συνένωσή των καναλιών για τον σχηματισμό της συμπιεσμένης εικόνας

Παρακάτω φαίνονται τα αποτελέσματα της συμπιεσης για διατήρηση των 5% και 50% των συντελεστών του μετασχηματισμού Fourier και τα αντίστοιχα τετραγωνικά σφάλματα και για τους δύο τρόπους συμπιεσης της εικόνας.





Παρατηρούμε και με τους δύο τρόπους συμπίεσης ότι χρατώντας μόνο το 5% των συντελεστών κατά το μετασχηματισμό του συνημιτόνου λαμβάνουμε μη ικανοποιητικά αποτελέσματα καθώς μπορούμε πλέον να διαχρίνουμε τα σύνορα των 32x32 παραθύρων που χωρίσαμε την εικόνα για να γίνει η συμπίεση και

η εικόνα έχει χάσει καυταρότητα. Στην περίπτωση του 50% έχουμε πολύ καλά αποτελέσματα συμπίεσης, λαμβάνοντας μια καυταρή εικόνα χωρίς διακριτά σύνορα στα παράθυρα και σχεδόν πανομοιότυπη με την αρχική.

3.3 Οι βιβλιοθήκες της άσκησης

Οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση αυτής της άσκησης ήταν :

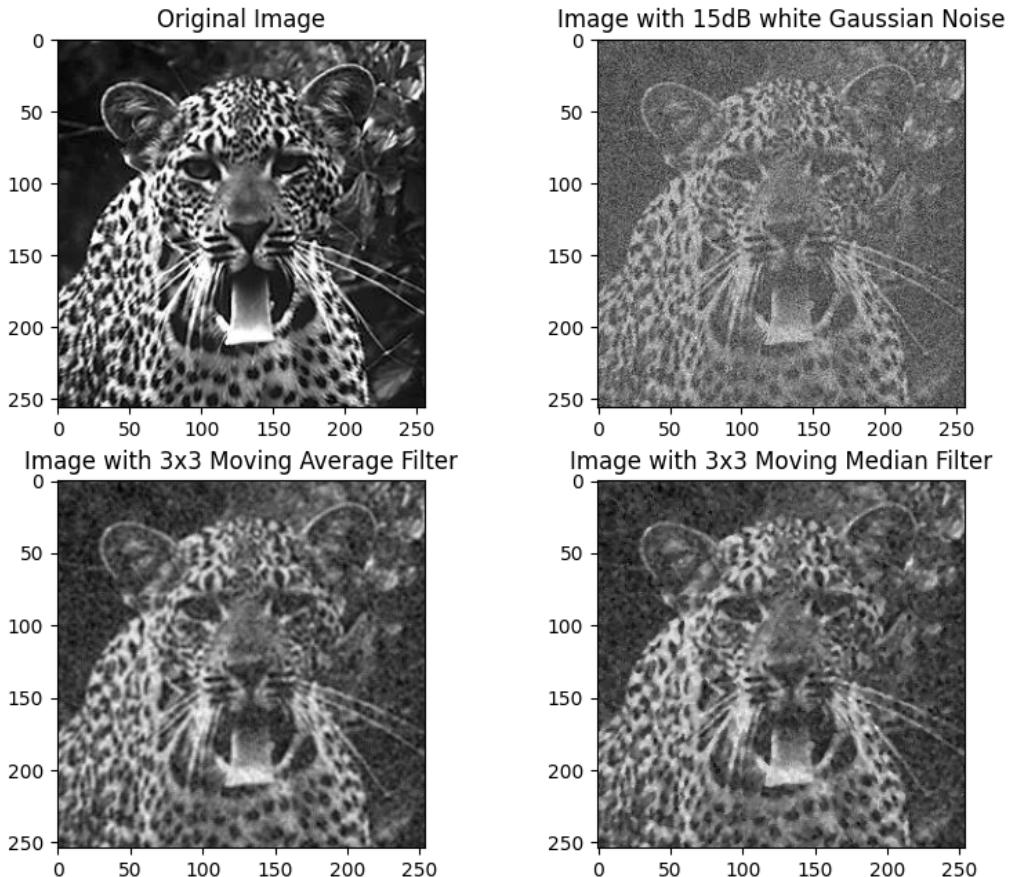
- **Numpy**, για αριθμητικούς υπολογισμούς με πίνακες
- **Matplotlib**, για την απεικόνιση των εικόνων
- **Scipy**, για τη φόρτωση της εικόνας από το Matlab αρχείο, μετασχηματισμοί συνημίτονου

4 -Άσκηση 3

Στόχος της άσκησης είναι η αποθορυβοποίηση μιας εικόνας σε δύο διαφορετικά είδη θορύβου με χρήση φίλτρων κινούμενο μέσου και φίλτρο διαμέσου.

Αρχικά προστέθηκε λευκός θόρυβος στην εικόνα για να φτάσουμε λόγω σήματος προς θόρυβο ίσο με $15dB$. Ο τρόπος χειρισμού των οριακών γραμμών και στηλών pixel που βρίσκονται εκτός της ισχύος των κινούμενων φίλτρων ήταν η αφαίρεση τους από την τελική εικόνα καθώς αποτελούν ένα πολύ μικρό ποσοστό από την αρχική εικόνα και δεν επηρεάζουν το τελικό αποτέλεσμα.

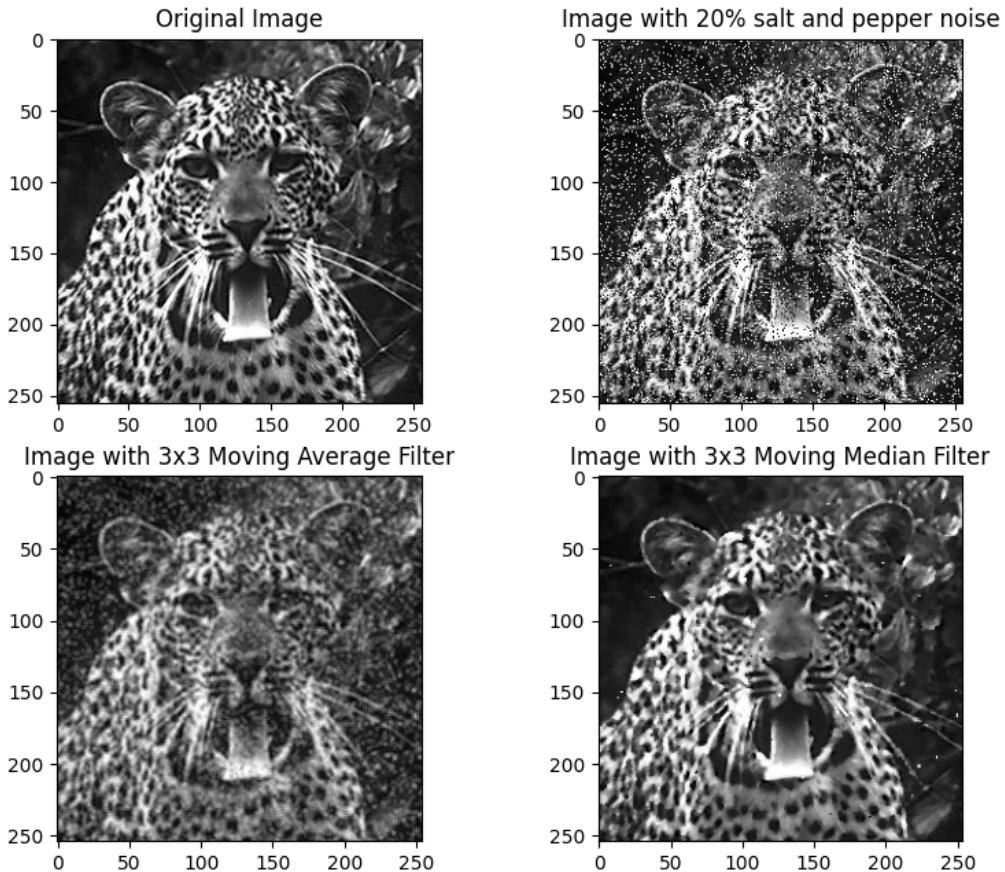
Για την περίπτωση μόνο του λευκού θορύβου αφού εφαρμόσουμε και τα δύο είδη φίλτρων λαμβάνουμε τα παρακάτω αποτελέσματα:



Δεδομένου ότι μια εικόνα με SNR που ανήκει μεταξύ 30dB και 50dB θεωρείται μια καλή εικόνα και μια εικόνα που ανήκει στο εύρος 20-30dB θεωρείται αποδεκτή, τα 15dB καυθιστούν την εικόνα μας αρκετά θορυβώδη. Παρατηρούμε επίσης ότι και τα δύο φίλτρα δίνουν παρόμοια αποτελέσματα.

Στην συνέχεια προστέθηκε θόρυβος τύπου "αλατοπίπερο" σε ποσοστό 20%. Η διαδικασία υλοποίησης ήταν η καταμέτρηση των πίζελ και η προσθήκη με τυχαίο τρόπο ακολουθώντας ομοιόμορφη κατανομή μεταξύ των τιμών "0" για το πιπέρι και "1" για το αλάτι στο ποσοστό που αναφέρθηκε παραπάνω. Η τοποθέτηση των τιμών "0" και "1" έγινε με τη χρήση τυχαίας γεννήτριας αριθμών.

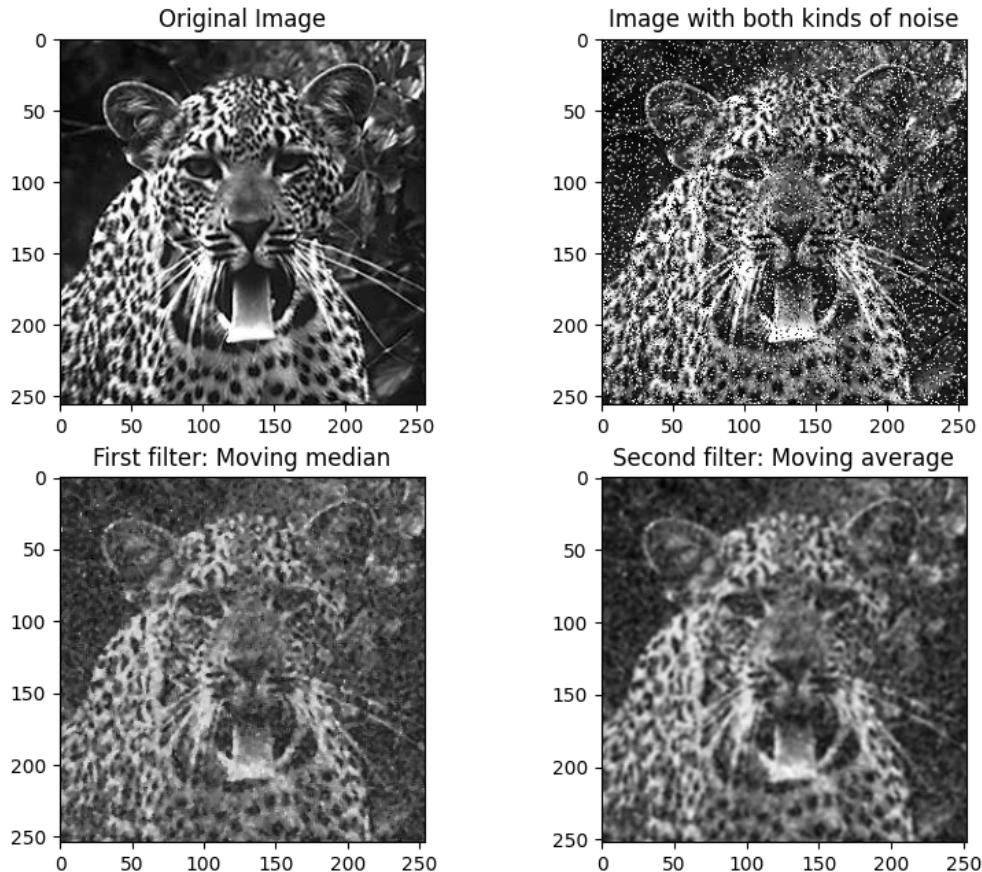
Παρακάτω φαίνονται τα αποτελέσματα της εφαρμογής των δύο φίλτρων στην παραπάνω εικόνα:



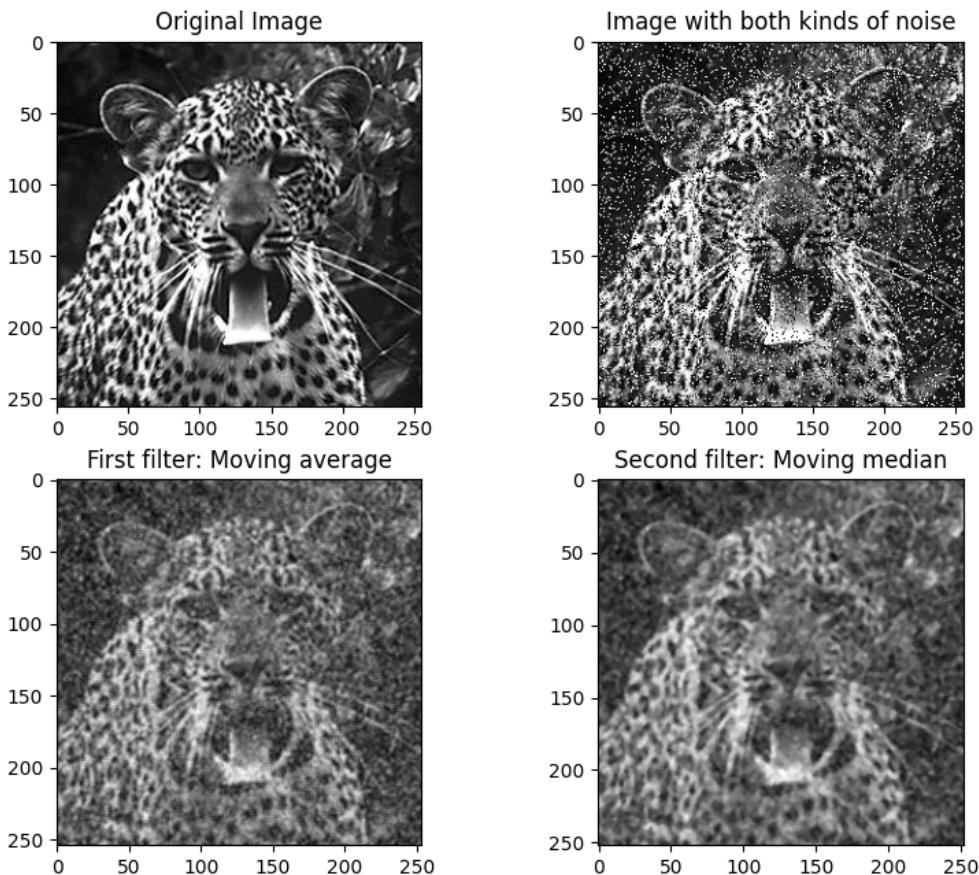
Στην περίπτωση αυτή παρατηρούμε ότι το φίλτρο του κινούμενου μέσου μας δίνει χειρότερα αποτελέσματα, κάτι που είναι λογικό καθώς παίρνοντας τον μέσο όρο της γειτονιάς του εκάστοτε pixel λαμβάνονται υπόψιν και οι ακραίες τιμές του θορύβου μετατοπίζοντας την τιμή του μέσου όρου. Από την άλλη πλευρά το φίλτρο του κινούμενο διαφέσον εφόσον τοποθετεί τα ακραία πίξελ στην άκρη δίνει πολύ καλύτερα αποτελέσματα.

Τέλος έγινε προσθήκη και των δύο ειδών θορύβου και έγινε διαδοχική χρήση των δύο φίλτρων. Τα αποτελέσματα φαίνονται παρακάτω:

Σε αυτή την περίπτωση που πρώτα εφαρμόζουμε το φίλτρο του κινούμενου μέσου λαμβάνουμε χειρότερα αποτελέσματα χάνοντας σημαντική λεπτομέρεια από την αρχική εικόνα.



Στην περίπτωση αυτή που το φίλτρο του κινούμενου διαμέσου εφαρμόστηκε πρώτα, αφαιρείται πρώτα ο ύδρυβος του αλατοπίπερου (δηλαδή οι αχραίες τιμές των πιξελ) και ύστερα με τον κινούμενο μέσο αφαιρείται και ο λευκός ύδρυβος. Με αυτό τον τρόπο λαμβάνουμε καλύτερα αποτελέσματα.



4.1 Οι βιβλιοθήκες της άσκησης

Οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση αυτής της άσκησης ήταν :

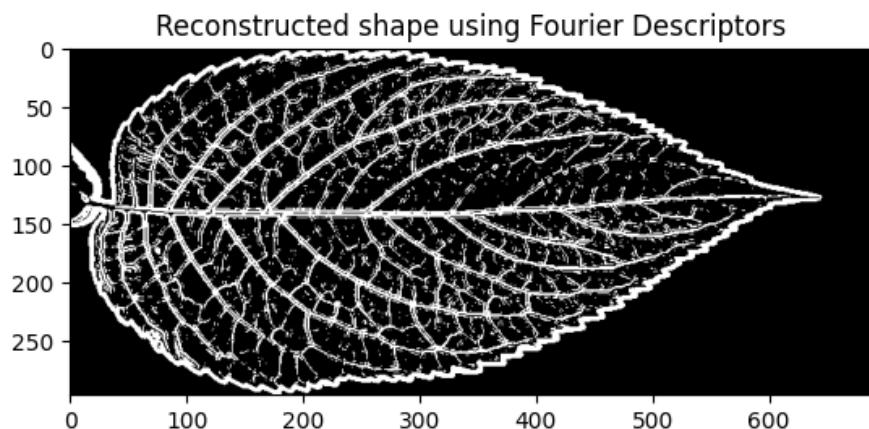
- **Numpy**, για αριθμητικούς υπολογισμούς με πίνακες
- **Matplotlib**, για την απεικόνιση των εικόνων
- **Scipy**, για τη φόρτωση της εικόνας από το Matlab αρχείο

5 - Άσκηση 4

Στόχος της άσκησης αυτή είναι η αρχικά η λήψη του σχήματος από μια εικόνα, στη συνέχεια η εύρεση των περιγραφών Fourier και τέλος η ανακατασκευή του σχήματος χάνοντας χρήση των περιγραφών. Αρχικά για την εύρεση του σχήματος της εικόνας έγινε εφαρμογή της μάσκας του Sobel με ένα κατώφλι για να πάρουμε μόνο τις ακμές που μας ενδιαφέρουν. Η μάσκα Sobel είναι η ακόλουθη:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 9 & -1 \end{bmatrix} * A \quad , G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

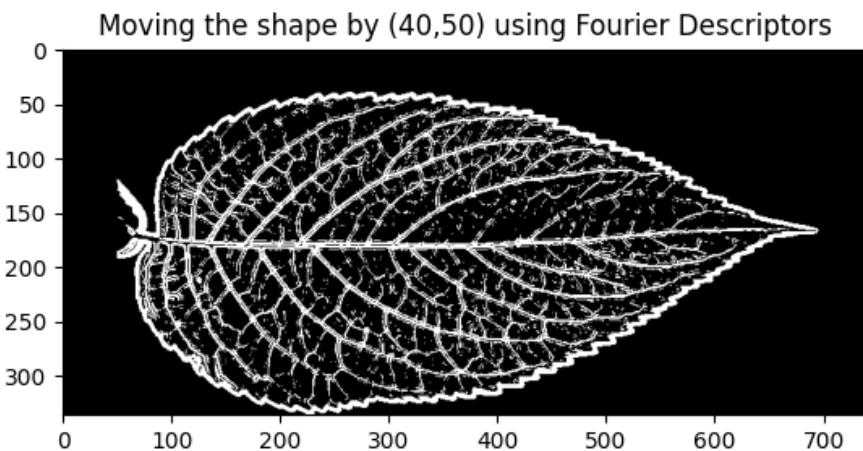
Οι μάσκες αυτές εφαρμόστηκαν με συνέλιξη στο χώρο και στη συνέχεια λάβαμε τις ακμές μέσω του $G = \sqrt{G_x^2 + G_y^2}$. Στη συνέχεια βρέθηκαν οι περιγραφές Fourier και έγινε η ανακατασκευή της εικόνας, η οποία φαίνεται παρακάτω:



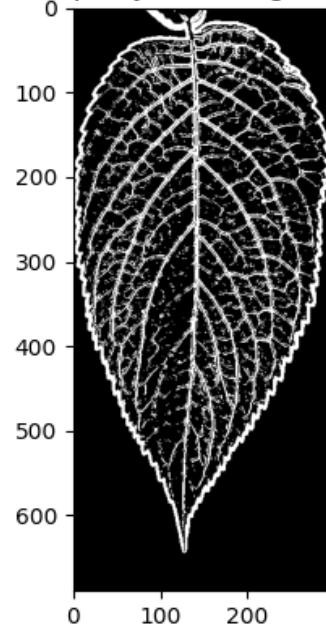
Έπειτα, για τις αλλαγές στο σχήμα εφαρμόστηκαν οι παρακάτω μετασχηματισμοί, όπου α αντιστοιχούν στους περιγραφείς Fourier και s στο περίγραμμα.

Μετατόπιση	$s_t = s(k) + \Delta_{xy}$
Μεγέθυνση	$a_s = \alpha \cdot a(u)$
Περιστροφή	$a_r = a(u) \cdot e^{j\theta}$

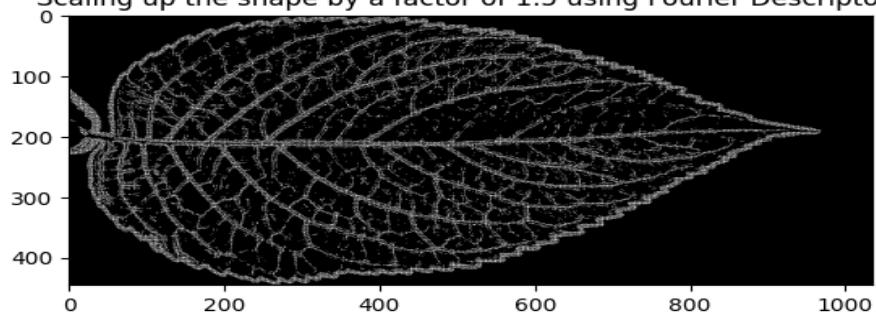
Αφού έγινε η ανακατασκευή των σχημάτων δημιουργήθηκαν οι παρακάτω εικόνες:



Rotating the shape by 90° using Fourier Descriptors



Scaling up the shape by a factor of 1.5 using Fourier Descriptors



Παρατηρούμε ότι τα αποτελέσματα είναι ικανοποιητικά με μια μικρή αισθητή

διαφορά στην μεγέθυνση της εικόνας που δεν φαίνονται τόσο έντονα τα περιγράμματα καθώς επίσης διαχρίνουμε τα λευκά πίξελ του περιγράμματος πιο εύκολα στην τελευταία.

5.1 Οι βιβλιοθήκες της άσκησης

Οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση αυτής της άσκησης ήταν :

- **Numpy**, για αριθμητικούς υπολογισμούς με πίνακες
- **Matplotlib**, για την απεικόνιση των εικόνων
- **Scipy**, για τη συνέλιξη 2 διαστάσεων
- **Pillow**, για τη φόρτωση των εικόνων

6 -'Ασκηση 5

6.1 Μέρος Α

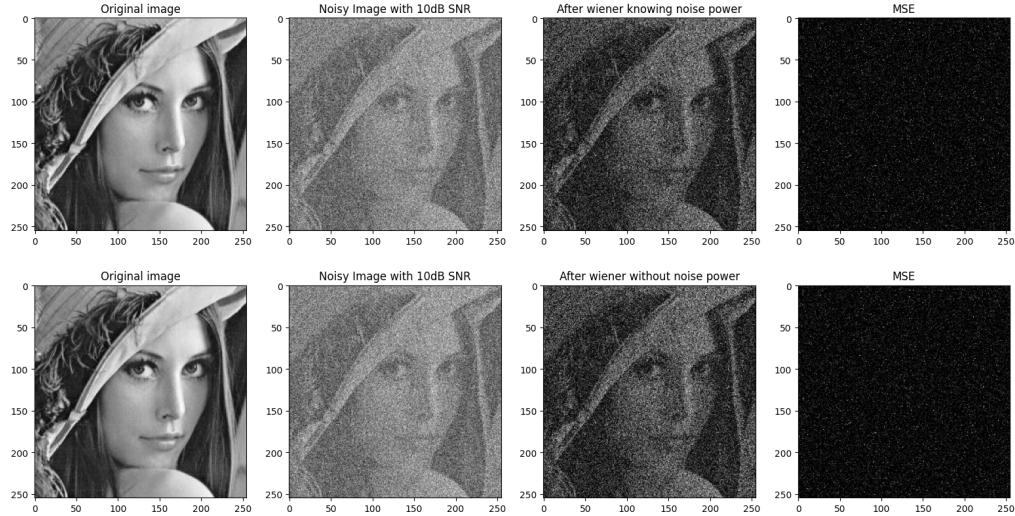
Στο πρώτο μέρος της άσκησης αυτής σκοπός είναι η υλοποίηση δύο φίλτρων Wiener για την αποθρύβοποίηση μιας εικόνας με λόγο SNR = 10 dB. Ο τρόπος δημιουργίας του πρώτου φίλτρου, έχοντας γνώση της ισχύος του θορύβου είναι ο εξής:

1. Zero padding της εικόνας σε διαστάσεις 2Nx2N
2. Αφαίρεση του μέσου όρου της εικόνας
3. Εύρεση του λόγου $H_w = \frac{Pf}{Pf + Pn}$ όπου Pf η ισχύς της εικόνας με το zero padding χωρίς το θόρυβο και Pn η γνωστή ισχύς του θορύβου
4. Ύστερα η εφαρμογή του φίλτρου H_w , η αφαίρεση του zero padding και η προσθήκη ξανά του μεσου όρου της αρχικής εικόνας.

Για τη δημιουργία του δεύτερου φίλτρου ακολουθήθηκε ακριβώς η ίδια διαδικασία με τη διαφορά ότι δεν χρησιμοποιήθηκε η γνωστή ισχύς του θορύβου αλλά πήραμε ένα τετραγωνικό πλαίσιο του μετασχηματισμού Fourier της εικόνας

με το θόρυβο στις υψηλές συχνότητες και λάβαμε τον μέσο όρο των pixel αυτών ως ισχύ του θορύβου.

Τα αποτελέσματα και των δύο φίλτρων φαίνονται παρακάτω καθώς και το μέσο τετραγωνικό σφάλμα.



Τα αποτελέσματα που λαμβάνουμε θεωρούνται ικανοποιητικά δεδομένου του πολύ χαμηλού SNR, και τα δύο φίλτρα αφαιρούν μεγάλο μέρος του θορύβου και επαναφέρουν σε έναν βαθμό τις αρχικές αποχρώσεις. Βέβαια τα τελικά αποτελέσματα απέχουν πολύ από την αρχική εικόνα.

6.2 Οι βιβλιοθήκες της άσκησης

Οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση αυτής της άσκησης ήταν :

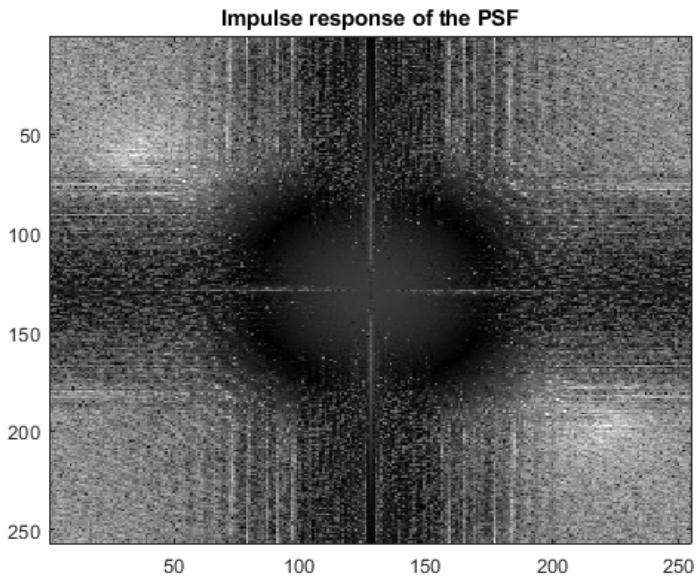
- **Numpy**, για αριθμητικούς υπολογισμούς με πίνακες
- **Matplotlib**, για την απεικόνιση των εικόνων
- **Scipy**, για τους μετασχηματισμούς Fourier δύο διαστάσεων μιας και το αντικείμενο της άσκησης δεν ήταν η υλοποίηση των μετασχηματισμών.
- **Pillow**, για τη φόρτωση των εικόνων

6.3 Μέρος Β

Για το δεύτερο μέρος εφαρμόστηκε στην αρχική εικόνα ο μετασχηματισμός point spread function με την μέθοδο που παρουσιάστηκε από την εκφώνηση $Y = psf(X)$. Στη συνέχεια εξιμεταλλευόμενοι την ιδιότητα του μετασχηματισμού Fourier

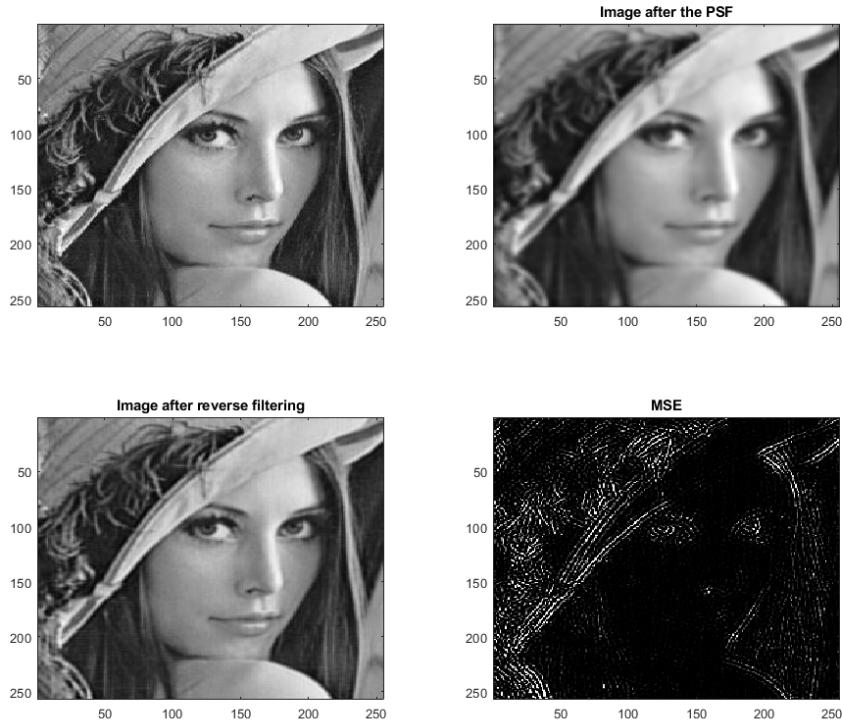
$$Y(u, v) = B(u, v) \cdot X(u, v) \iff B(u, v) = \frac{Y(u, v)}{X(u, v)}$$

παίρνουμε την χρονική απόκρουση του συστήματος υποβάθμισης, όπου η τελευταία φαίνεται παραχάτω:



Κάνοντας χρήση του αντίστροφου φιλτραρίσματος με κατώφλι ίσο με $\gamma=0.9$ και αφού χρησιμοποιήσουμε 2D-IFFT παίρνουμε τα αποτελέσματα που φαίνονται παρακάτω:

$$H(u, v) = \begin{cases} 1/B(u, v) & (1/|B(u, v)|) < \gamma \\ \gamma|B(u, v)|/B(u, v) & (1/|B(u, v)|) \geq \gamma \end{cases}$$



Αφού έχει γίνει χρήση διαφορετικών κατωφλιών για το αντίστροφο φιλτράρισμα επιλέχθηκε η παραπάνω τιμή. Τα αποτελέσματα που λαμβάνουμε είναι παραπάνω από ικανοποιητικά καθώς παρατηρούμε ότι έχει φύγει το θόλωμα και από την απεικόνιση του μέσου τετραγωνικού σφάλματος διακρίνουμε ότι έχουμε χάσει μόνο πληροφορία σε ορισμένες ακμές.

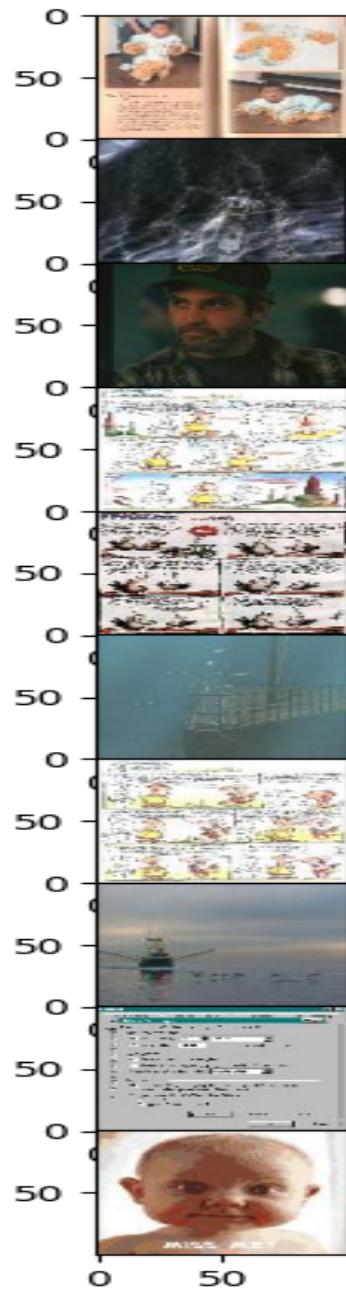
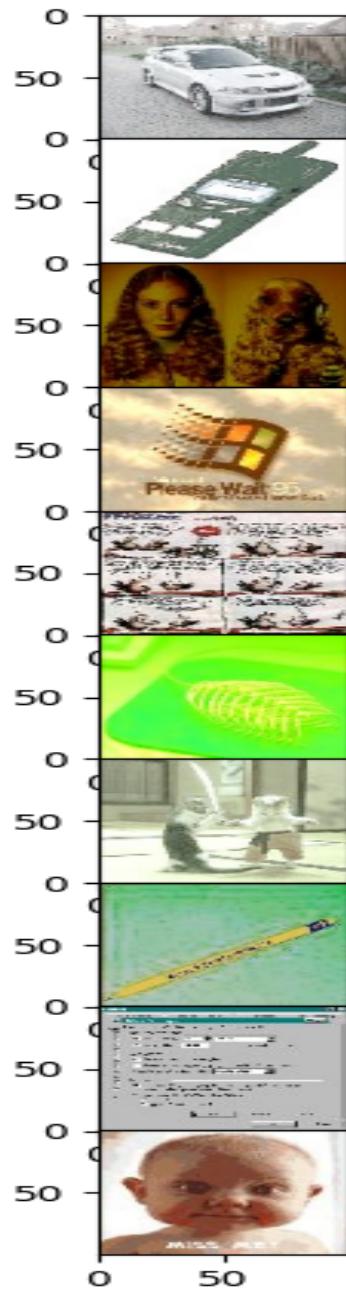
7 -Άσκηση 6

7.1 Ιστογράμματα

Στόχος της άσκησης είναι η αναζήτηση εικόνων από μια Βάση Δεδομένων εικόνων και η εξαγωγή ποσοστών επιτυχίας κάνοντας χρήση δύο διαφορετικές μεθόδους αναζήτησης.

Ο πρώτος τρόπος βασίζεται πάνω στα ιστογράμματα. Αρχικά παίρνουμε όλες τις εικόνες που θέλουμε να αναζητήσουμε στη βάση δεδομένων και εξάγουμε το ιστόγραμμα τους. Στη συνέχεια κάθισε ένα ιστόγραμμα το συγχρίνουμε με το ιστόγραμμα της εκάστοτε εικόνας στη βάση. Υστερα αναζητούμε την εικόνα με το μικρότερο μέσο τετραγωνικό σφάλμα και τις απεικονίζουμε για να δούμε με ποια εικόνα είναι πιο κοντά σύμφωνα με τον τρόπο αναζήτησης μας.

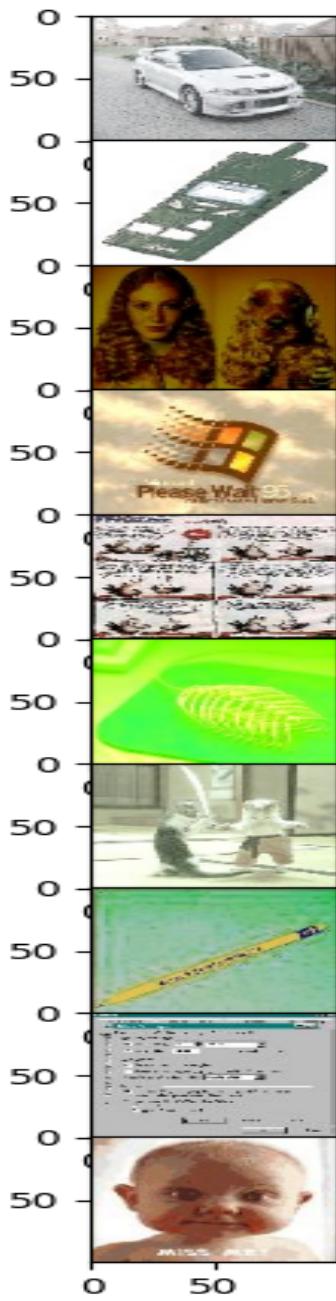
Παρακάτω φαίνονται τα αποτελέσματα της αναζήτησης με βάση το ιστόγραμμα τα οποία μας δίνουν ένα ποσοστό επιτυχίας της τάξης του 30% εντοπίζοντας ορθά μόνο τις 3 από τις 10 εικόνες του φακέλου Test.



7.2 Μετασχηματισμός Συνημιτόνου

Ο δεύτερος τρόπος βασίζεται πάνω στον μετασχηματισμό DCT. Αρχικά παίρνουμε κάθε εικόνα που βρίσκεται στον φάκελο Test βρίσκουμε τον μετασχηματισμό DCT κρατώντας ένα ποσοστό $r\%$ των συντελεστών του. Στη συνέχεια συγχρίνουμε τον μετασχηματισμό αυτό με αυτόν της κάθε εικόνας που βρίσκεται στην βάση δεδομένων και υπολογίζουμε το μεταξύ τους μέσο τετραγωνικό σφάλμα. Τέλος απεικονίζουμε την εικόνα που είχε το μικρότερο σφάλμα ως αυτή που είναι πιο κοντά με την αρχική εικόνα.

Παρακάτω φαίνονται τα αποτελέσματα της αναζήτησης με βάση το ιστόγραμμα τα οποία μας δίνει ένα ποσοστό επιτυχίας της τάξης του 30% ανεξάρτητα από την διαφορετική επιλογή των συντελεστών του μετασχηματισμού DCT, εντοπίζοντας σωστά 3 από τις 10 εικόνες του φακέλου Test.



7.3 Οι βιβλιοθήκες της άσκησης

Οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση αυτής της άσκησης ήταν :

- **Numpy**, για αριθμητικούς υπολογισμούς με πίνακες
- **Matplotlib**, για την απεικόνιση των εικόνων
- **Scipy**, για τον μετασχηματισμό συνημίτονου
- **Os**, για την εύρεση της τοποθεσίας των εικόνων στο σύστημα