

Ψηφιακή επεξεργασία εικόνας και εφαρμογές

Εργασία 1

Καλδάνης Θωμάς 2022025

Άσκηση #1, Κβάντιση

Γ) Για $K = 5$ το μέσο τετραγωνικό σφάλμα (595.70) είναι υψηλό αφού η εικόνα χάνει πολύ λεπτομέρεια.

Για $K = 20$ το μέσο τετραγωνικό σφάλμα (158.03) έχει μειωθεί αρκετά, η εικόνα πλησιάζει περισσότερο την αρχική.

Για $K = 200$ το μέσο τετραγωνικό σφάλμα (25.84) είναι αρκετά μικρό με αποτέλεσμα να υπάρχει ελάχιστη διαφορά σε σχέση με την αρχική.

Για $K = 1000$ η εικόνα φαίνεται ίδια με την αρχική αφού το μέσο τετραγωνικό σφάλμα (8.90) είναι πλέον πολύ μικρό.

Αρχική Εικόνα



5 χρώματα
MSE = 595.70



20 χρώματα
MSE = 158.03



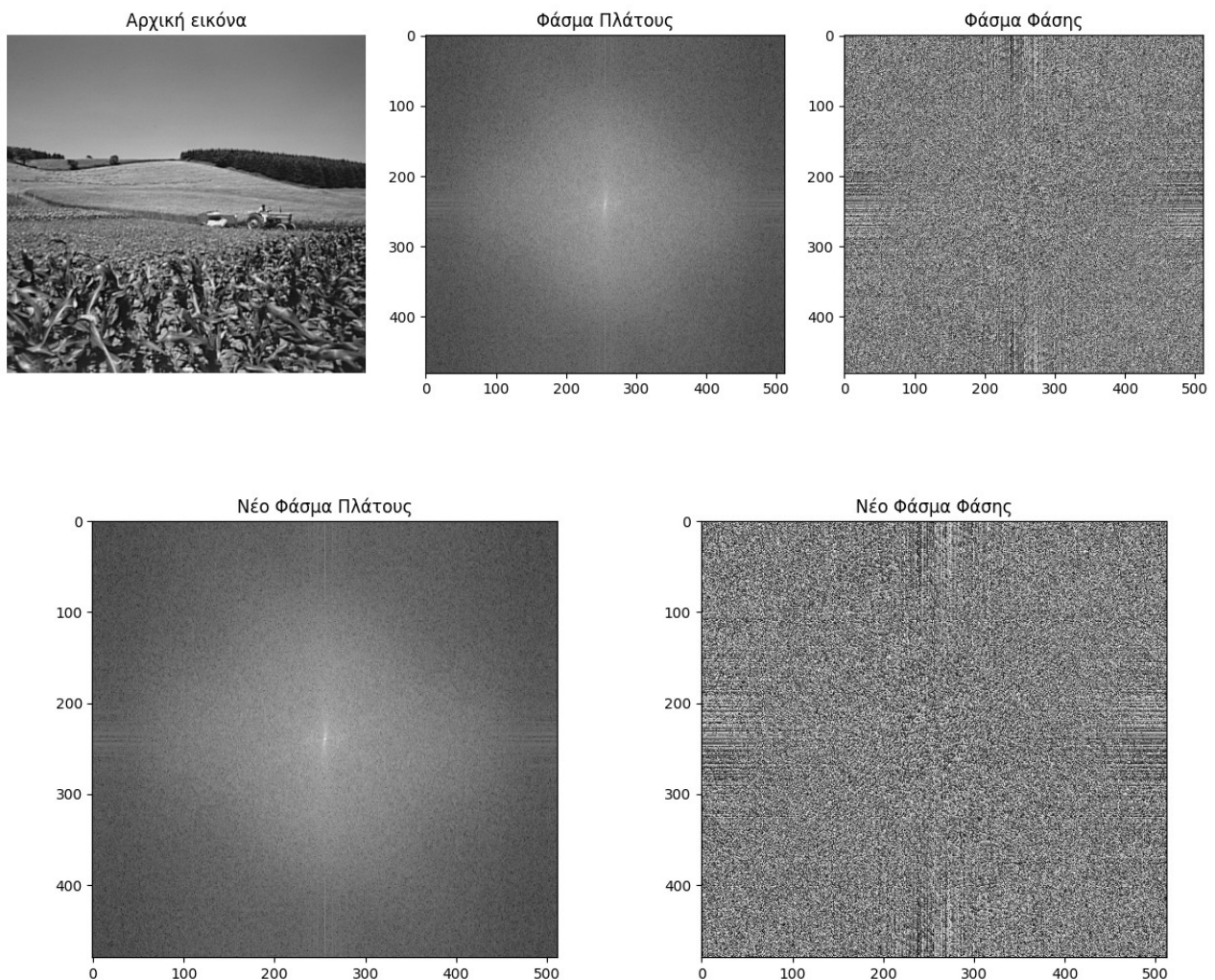
200 χρώματα
MSE = 25.84



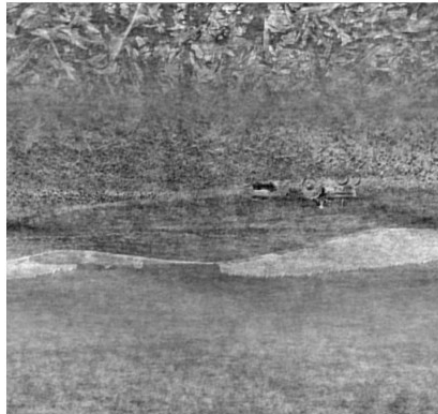


Άσκηση #2, Μετασχηματισμός Fourier:

Γ) Η εικόνα έχει αλλοιωθεί σε μεγάλο βαθμό, έχει χαθεί η λεπτομέρεια και εμφανίζονται παραμορφώσεις (εμφανίζεται μια κατοπτρική εκδοχή της εικόνας). Αυτό συμβαίνει επειδή αλλάξαμε την φάση της εικόνας με την κατακόρυφα συμμετρική της.



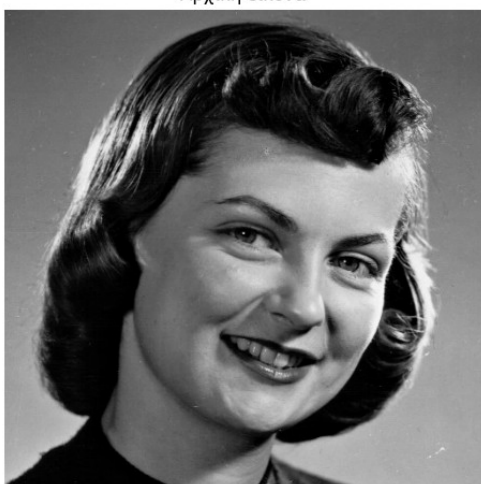
Τροποποιημένη Εικόνα



Άσκηση #3, Φιλτράρισμα:

Ερώτημα Α: Για να εντοπιστούν τα κύρια αντικείμενα στην εικόνα, χρησιμοποιείται η μέθοδος Otsu όπου διαχωρίζει το αντικείμενο με το background. Η αρχική εικόνα ορίζεται στην μεταβλητή `girlface`. Στη συνέχεια γίνεται ο διαχωρισμός αντικειμένου και background (εκεί που παρατηρούνται μεγάλες αλλαγές στην εικόνα) και με βάση τον διαχωρισμό δημιουργείται μία δυαδική εικόνα όπου φαίνονται απομονωμένα τα κύρια στοιχεία του αντικειμένου.

Αρχική εικόνα



Κύρια αντικείμενα εικόνας



Ερώτημα Β: Για τον εντοπισμό των περιγραμμάτων των αντικειμένων, χρησιμοποιείται η μέθοδος Canny. Η αρχική εικόνα ορίζεται στην μεταβλητή `fruits`. Χρησιμοποιείται `Gaussian Blur` ώστε να έχουμε πιο καθαρό αποτέλεσμα και τέλος εφαρμόζεται σε αυτό η μέθοδος Canny για να εντοπιστούν τα περιγράμματα.

Αρχική εικόνα



Περιγράμματα αντικειμένων εικόνας

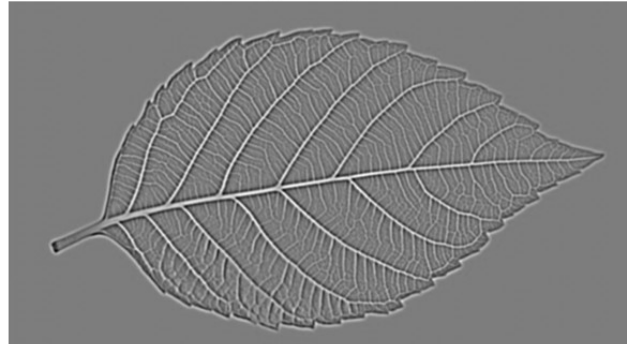


Ερώτημα Γ: Για τον εντοπισμό των λεπτομερειών χρησιμοποιείται η μέθοδος `Laplacian`. Η αρχική εικόνα ορίζεται στην μεταβλητή `leaf`. Χρησιμοποιείται η μέθοδος `Laplacian` στο συγκεκριμένο ερώτημα επειδή είναι η πιο κατάλληλη μέθοδος για να εντοπίσει τις λεπτομέρειες του φύλλου, οι οποίες με κάποιο άλλο φίλτρο δεν θα εντοπίζονταν εύκολα. Πρίνα την εφαρμογή του φίλτρου χρησιμοποιούμε `Gaussian blur` ώστε να έχουμε πιο καθαρό αποτέλεσμα.

Αρχική εικόνα



Λεπτομέρειες αντικειμένων εικόνας

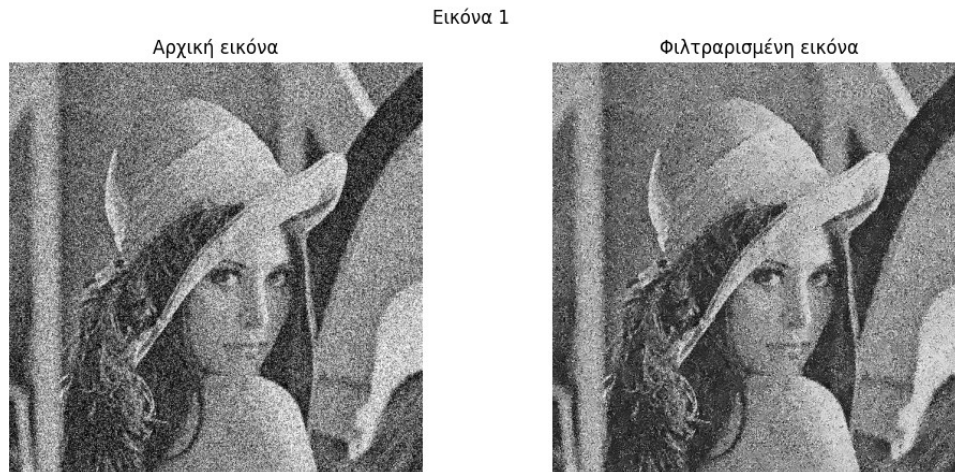


Άσκηση #4, Βελτιστοποίηση:

Ερώτημα Α:

Εικόνα 1: Για την βελτιστοποίηση της εικόνας 1 χρησιμοποιείται Non-local means denoising

(https://scikit-image.org/docs/0.25.x/auto_examples/filters/plot_nonlocal_means.html)



Εικόνα 2: Για την βελτιστοποίηση της εικόνας 2 χρησιμοποιείται Median φίλτρο. Αυτό γιατί το Median είναι κατάλληλο για salt & pepper θόρυβο. Το μόνο που κάνουμε, είναι να χρησιμοποιήσουμε την median() ώστε να απομακρύνει τον θόρυβο απο την εικόνα.



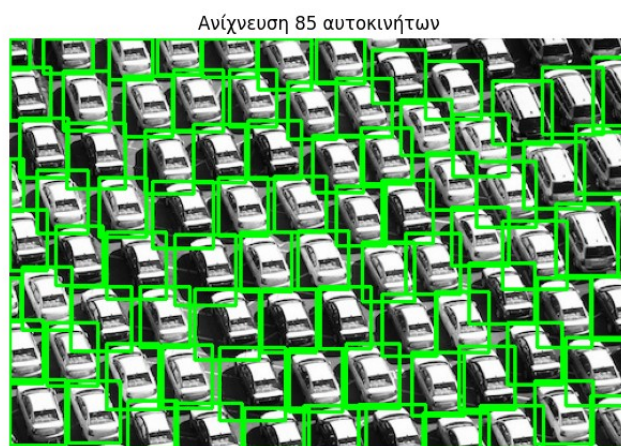
Ερώτημα Β:

Εικόνα 1: Το SSIM για την εικόνα 1 είναι 0.005, που σημαίνει ότι το φίλτρο που χρησιμοποιείται δεν είναι αποτελεσματικό ώστε να αφαιρέσει τον θόρυβο. (Χρησιμοποίησα και άλλα φίλτρα, όμως το αποτέλεσμα παρέμενε ίδιο, οπότε επέλεξα τυχαία αυτό)

Εικόνα 2: Το SSIM για την εικόνα 2 είναι 0.8446 που σημαίνει ότι έχει γίνει πολύ καλή αποκατάσταση της εικόνας.

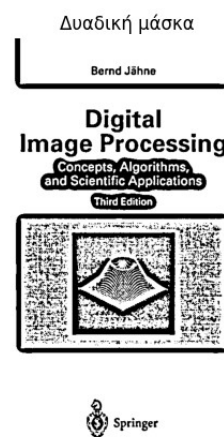
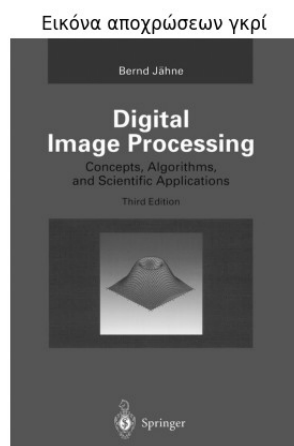
Άσκηση #5, Τμηματοποίηση:

Για την τμηματοποίηση χρησιμοποιούμε το μοντέλο YOLO το οποίο βασίζει την εκπαίδευσή του σε αντικείμενα που βρίσκονται στην εικόνα. Αρχικά πρέπει να φορτώσουμε το μοντέλο και την εικόνα που θα αναλύσουμε και χρησιμοποιούμε πάνω της το φίλτρο όπου κάνει και την τμηματοποίηση. Το results επιστρέφει τις περιοχές που έχει εντοπίσει. Στη συνέχεια ψάχνουμε μέσα στο results για αντικείμενα τύπου car και όταν βρούμε ένα τέτοιο αντικείμενο αυξάνουμε την μεταβλητή που μετράει τα αυτοκίνητα και σχεδιάζουμε το bounding box γύρω από το αντικείμενο car που εντοπίσαμε. Τέλος εμφανίζεται το τελικό αποτέλεσμα, αφού η εικόνα έχει μετατραπεί πρώτα από BGR σε RGB (αυτό γιατί η cv2 επεξεργάζεται εικόνες BGR και για να εμφανίσουμε με matplotlib χρειαζόμαστε RGB). Στο τελικό αποτέλεσμα, επειδή η εικόνα έχει κάποια αμάξια που δεν φαίνονται ολόκληρα, το μοντέλο δεν τα αναγνωρίζει με αποτέλεσμα να τα προσπερνάει.



Άσκηση #6, Τμηματοποίηση 2:

Η μετατροπή της εικόνας σε αποχρώσεις γκρι γίνεται κατευθείαν με το που την διαβάζουμε και κάνουμε Gaussian blur ώστε να είναι πιο καθαρά τα αποτελέσματα. Με την adaptiveThreshold δημιουργούμε την δυαδική εικόνα όπου διαχωρίζει με λευκό και μαύρο τον φόνο με τα γράμματα. Για τον εντοπισμό των γραμμάτων χρησιμοποιούμε την pytesseract.image_to_string(), η οποία χρησιμοποιεί την δυαδική μάσκα. Για να εμφανιστούν τα σωστά αποτελέσματα σχετικά με τον εντοπισμό χαρακτήρων έγιναν πολλές δοκιμές στις παραμέτρους blockSize: 31 και C: 0, στην δημιουργία της μάσκας.



Αποτελέσματα:

Bernd Jähne

Digital
Image Processing

Concepts, Algorithms,
and Scientific Applications

Third Edition

& Springer