



ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΚΑΙ ΕΦΑΡΜΟΣΜΕΝΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

ΕΠΛ445: Ψηφιακή Επεξεργασία Εικόνας

Εργαστήριο

4^η Εργαστηριακή Άσκηση

Ομάδα:

Βάκη Νικόλας 1025806
Δημητρίου Καρολίνα 1035986
Ηλιάδη Έλλη 1018368

Ημερομηνία Παράδοσης
11 Σεπτεμβρίου 2022

1. Εισαγωγή - Στόχος

Στόχος της εργασίας είναι η επεξεργασία σε κβαντισμένες εικόνες και ο μετασχηματισμός τους σε εικόνες καλύτερης ποιότητας (με κάποια κριτήρια), εικόνες με ορισμένα χαρακτηριστικά υπερτιμημένα και εικόνες με ορισμένα χαρακτηριστικά με μειωμένη έμφαση. Επιπλέον στόχος μας είναι η αφαίρεση θορύβων όσο το δυνατόν καλύτερα από τις εικόνες με την χρήση των κατάλληλων φίλτρων. Τέλος, στόχος της θεωρητικής άσκησης είναι η ορθή εφαρμογή της γραμμικής συνέλιξης.

2. Μεθοδολογία

1. Να αναζητά (browse) και να φορτώνει μια εικόνα από τον υπολογιστή.

```
def browseButton_handler(self):  
    self.open_dialog_box()
```

2. Να εμφανίζει σε text box το path που είναι αποθηκευμένη η εικόνα στον υπολογιστή.

```
def open_dialog_box(self):  
    global path  
    filename = QFileDialog.getOpenFileName(filter="Images (*.png *.tiff *.jpg)")  
    path = filename[0]  
    self.pathLineEdit.setText(path)
```

3. Να υπάρχει ένα combo box με τις επιλογές θορύβων:

- a. Gaussian για Gaussian white noise with constant mean and variance
- b. Localvar για Zero-mean Gaussian white noise with an intensity-dependent variance
- c. Poisson για Poisson noise
- d. Salt & pepper για On and off pixels
- e. Speckle για Multiplicative noise

```

def noise_select(self, img):
    global noise, titleNoise
    titleNoise=self.comboBox.currentText()

    if titleNoise == 'Gaussian':
        noise = skimage.util.random_noise(img, mode="gaussian")

    elif titleNoise == 'Localvar':
        noise = skimage.util.random_noise(img, mode="localvar")

    elif titleNoise == 'Poisson':
        noise = skimage.util.random_noise(img, mode="poisson")

    elif titleNoise == 'Salt & Pepper':
        noise = skimage.util.random_noise(img, mode="s&p", amount=0.2)

    elif titleNoise == 'Speckle':
        noise = skimage.util.random_noise(img, mode="speckle")

    else:
        print("error")

```

4. Να υπάρχει ένα combo box με τις επιλογές φίλτρων:

- a. Averaging για Averaging filter
- b. Gaussian για Gaussian lowpass filter
- c. Median για Median blurring
- d. Laplacian για Approximates the two-dimensional Laplacian operator
- e. Sobel για Sobel horizontal edge-emphasizing filter

```
def filter_select(self, img):
    global filter, titleFilter
    titleFilter=self.comboBox_2.currentText()

    if titleFilter=='Averaging':
        filter = cv2.blur(img, (5, 5))

    elif titleFilter=='Gaussian':
        filter = cv2.GaussianBlur(img, (5, 5), 0)

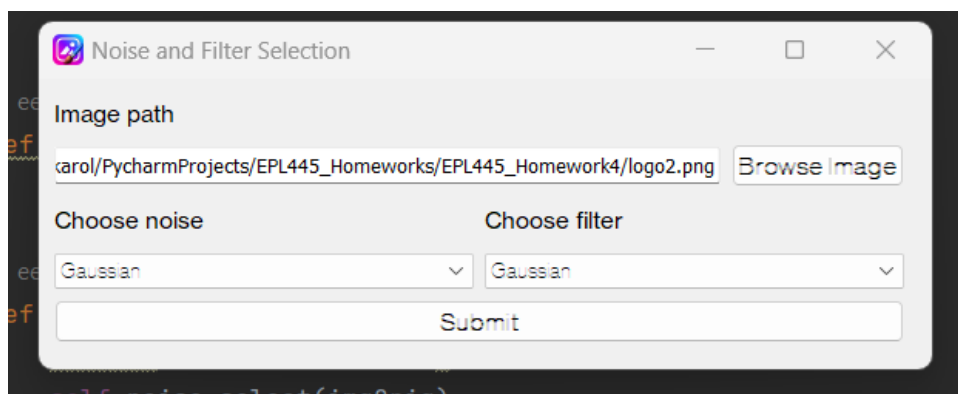
    elif titleFilter=='Median':
        img=np.float32(img)
        filter = cv2.medianBlur(img, 5)

    elif titleFilter=='Laplacian':
        filter = cv2.Laplacian(img, cv2.CV_64F)

    elif titleFilter=='Sobel':
        filter = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=5) #sobelx

    else:
        print("error")
```

5. Για κάθε θόρυβο που επιλέγετε να χρησιμοποιήσετε το κατάλληλο φίλτρο για να τον αφαιρέσει όσο το δυνατό καλύτερα.



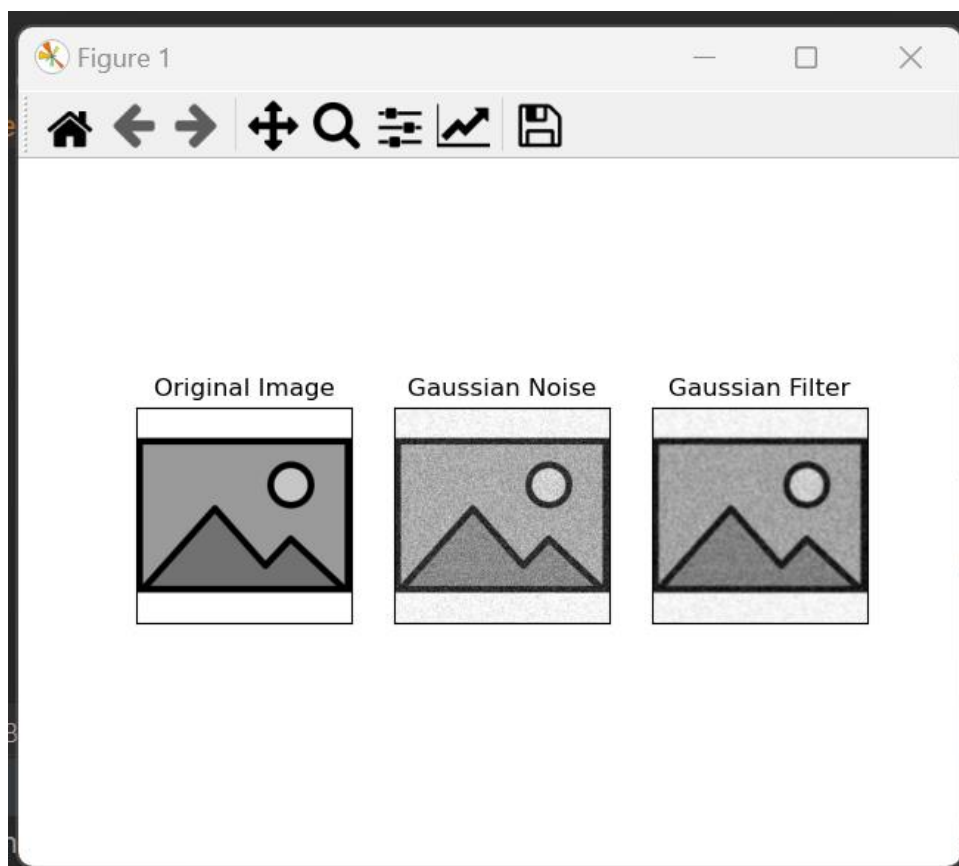
6. Μόλις πατηθεί το κουμπί Submit, για κάθε μια από τις επιλογές σας να τυπώνετε την αρχική εικόνα, την εικόνα με τον θόρυβο και την εικόνα που προκύπτει μετά την εφαρμογή του φίλτρου.

```
def submitButton_handler(self):
    self.submitButton_dialogBox()
```

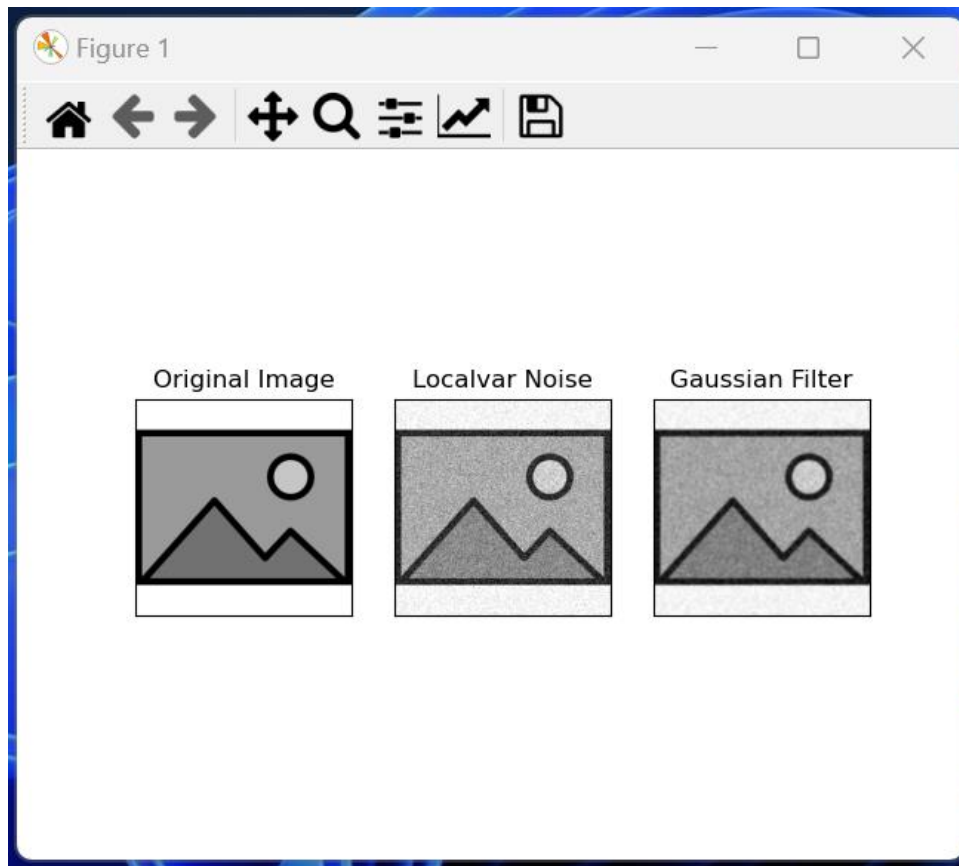
```
def submitButton_dialogBox(self):
    imgOrig = cv2.imread(path_0)
    self.noise_select(imgOrig)
    self.filter_select(noise)
    plt.subplot(131), plt.imshow(imgOrig, cmap='gray')
    plt.title('Original Image'), plt.xticks([], plt.yticks([]))
    plt.subplot(132), plt.imshow(noise, cmap='gray')
    plt.title(titleNoise+' Noise'), plt.xticks([], plt.yticks([]))
    plt.subplot(133), plt.imshow(filter, cmap='gray')
    plt.title(titleFilter+' Filter'), plt.xticks([], plt.yticks([]))

    plt.show()
```

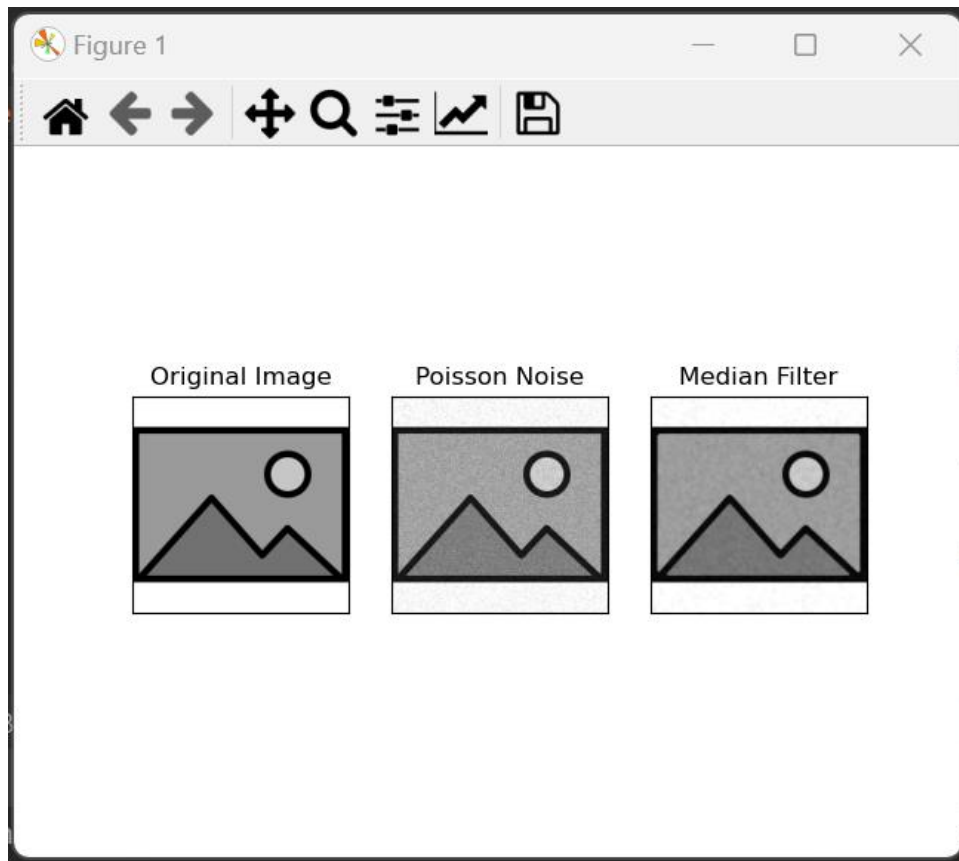
3. Αποτελέσματα



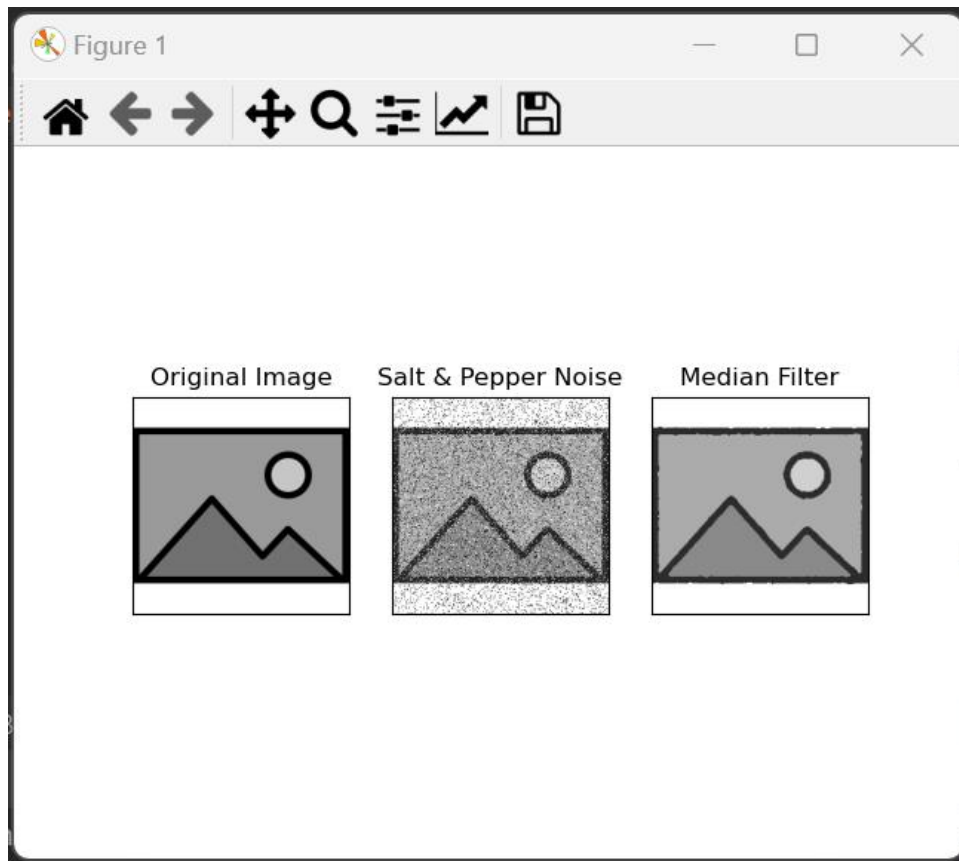
Χρησιμοποιήθηκε Gaussian θόρυβος και επιλέχθηκε Gaussian φίλτρο.



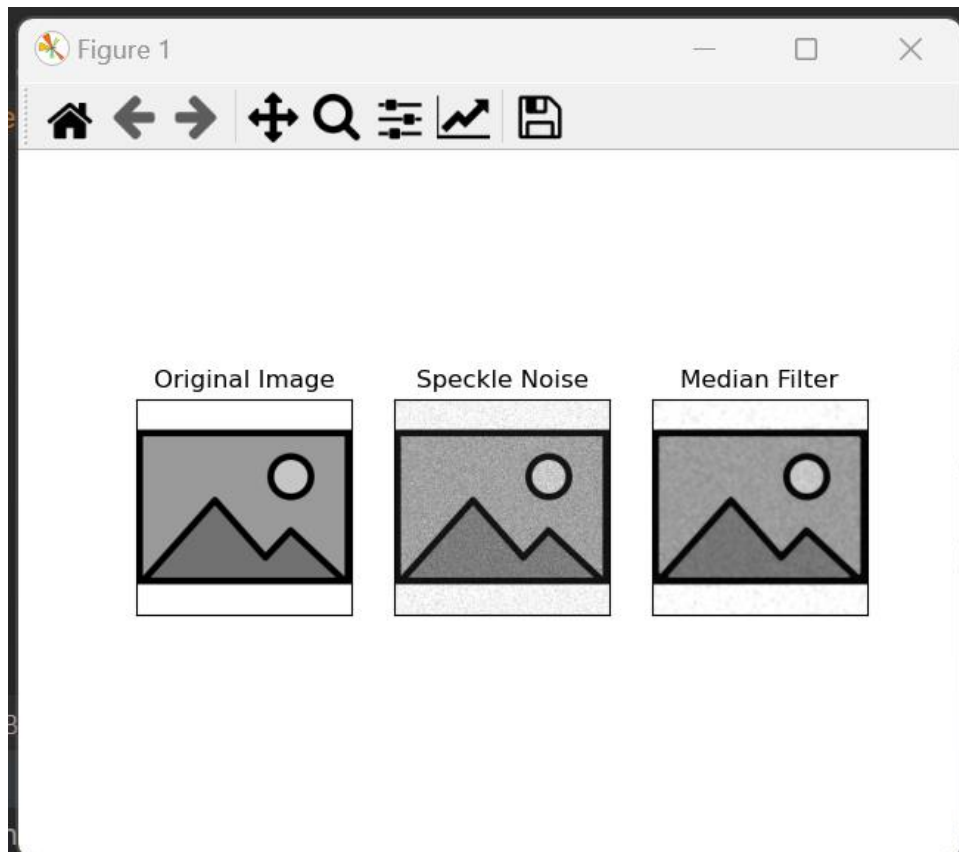
Χρησιμοποιήθηκε Localvar θόρυβος και επιλέχθηκε Guassian φίλτρο.



Χρησιμοποιήθηκε Poisson θόρυβος και επιλέχθηκε Median φίλτρο.



Χρησιμοποιήθηκε Salt & pepper θόρυβος και επιλέχθηκε Median φίλτρο.



Χρησιμοποιήθηκε Speckle θόρυβος και επιλέχθηκε Median φίλτρο.

4. Συμπεράσματα - Σχολιασμός Αποτελεσμάτων

- Noise: Gaussian Filter: Gaussian. - Δοκιμάζοντας όλα τα φίλτρα διαπιστώσαμε πως το φίλτρο Gaussian αφαιρεί ικανοποιητικά τον θόρυβο, όλα τα άλλα φίλτρα παρουσίαζαν μια θολή ή πλήρως αλλοιωμένη εικόνα.
- Noise: Localvar Filter: Gaussian. - Δοκιμάζοντας όλα τα φίλτρα διαπιστώσαμε πως το φίλτρο Gaussian αφαιρεί ικανοποιητικά τον θόρυβο, όλα τα άλλα φίλτρα παρουσίαζαν μια θολή ή πλήρως αλλοιωμένη εικόνα.
- Noise: Poisson Filter: Median. - Δοκιμάζοντας όλα τα φίλτρα διαπιστώσαμε πως το φίλτρο Median αφαιρεί ικανοποιητικά τον θόρυβο, όλα τα άλλα φίλτρα παρουσίαζαν μια θολή ή πλήρως αλλοιωμένη εικόνα.
- Noise: Salt&Pepper Filter: Median. - Δοκιμάζοντας όλα τα φίλτρα διαπιστώσαμε πως το φίλτρο Median αφαιρεί ικανοποιητικά τον θόρυβο, παρουσιάζει μια ξεκάθαρα πιο καθαρή εικόνα με έμφαση στις σκούρες

λεπτομέρειες όλα τα άλλα φίλτρα δεν παρουσίαζαν κάποια αλλαγή ή παρουσίαζαν μια πλήρως αλλοιωμένη εικόνα.

- Noise: Speckle Filter: Median. - Δοκιμάζοντας όλα τα φίλτρα διαπιστώσαμε πως το φίλτρο Median αφαιρεί ικανοποιητικά τον θόρυβο, όλα τα άλλα φίλτρα παρουσίαζαν μια θολή ή πλήρως αλλοιωμένη εικόνα.
- Ο Gaussian θόρυβος προσθέτει στην εικόνα λευκό θόρυβο με σταθερό μέσο όρο και διακύμανση.
- Ο Localvar θόρυβος προσθέτει στην εικόνα μηδενικό μέσο Gaussian θόρυβο όπου εξαρτάται από την ένταση.
- Ο Poisson θόρυβος δημιουργεί poisson noise.
- Ο S&P θόρυβος αντικαθιστά τυχαία pixels με 1 ή low_var, όπου το low_var είναι 0 για unsigned εικόνες ή -1 για signed εικόνες.
- Ο Speckle θόρυβος είναι πολλαπλασιαστικός θόρυβος και κάνει χρήση του τύπου $out = image + n * image$, όπου n είναι Gaussian θόρυβος.
- Σύμφωνα με τα αποτελέσματα των εικόνων πιο πάνω, χρησιμοποιήθηκαν τα φίλτρα Gaussian και Median για καθαρισμό των θορύβων. Τα φίλτρα Laplacian και Sobel αλλοίωναν περισσότερο τις φωτογραφίες.
- Το Gaussian είναι γραμμικό φίλτρο, το οποίο δεν διατηρεί τις άκρες στην εικόνα εισόδου.
- Το Median φίλτρο είναι μη γραμμικό και αντικαθιστά τις τιμές των εικονοστοιχείων με τη διάμεση τιμή που είναι διαθέσιμη.
- Το φίλτρο Averaging κάνει μια μέτρια δουλειά ως προς το απομάκρυσμα των θορύβων από τις εικόνες.
- Γενικό συμπέρασμα, είναι πως τα φίλτρα Gaussian και Median αποδεικνύονται πως είναι τα καλύτερα για αφαίρεση κάθε είδους θορύβου, αφού μετά από πολλές δοκιμές όλων των φίλτρων, τα 2 αυτά φίλτρα καθάρισαν την εικόνα παρουσιάζοντας πολύ παρόμοια αποτελέσματα.