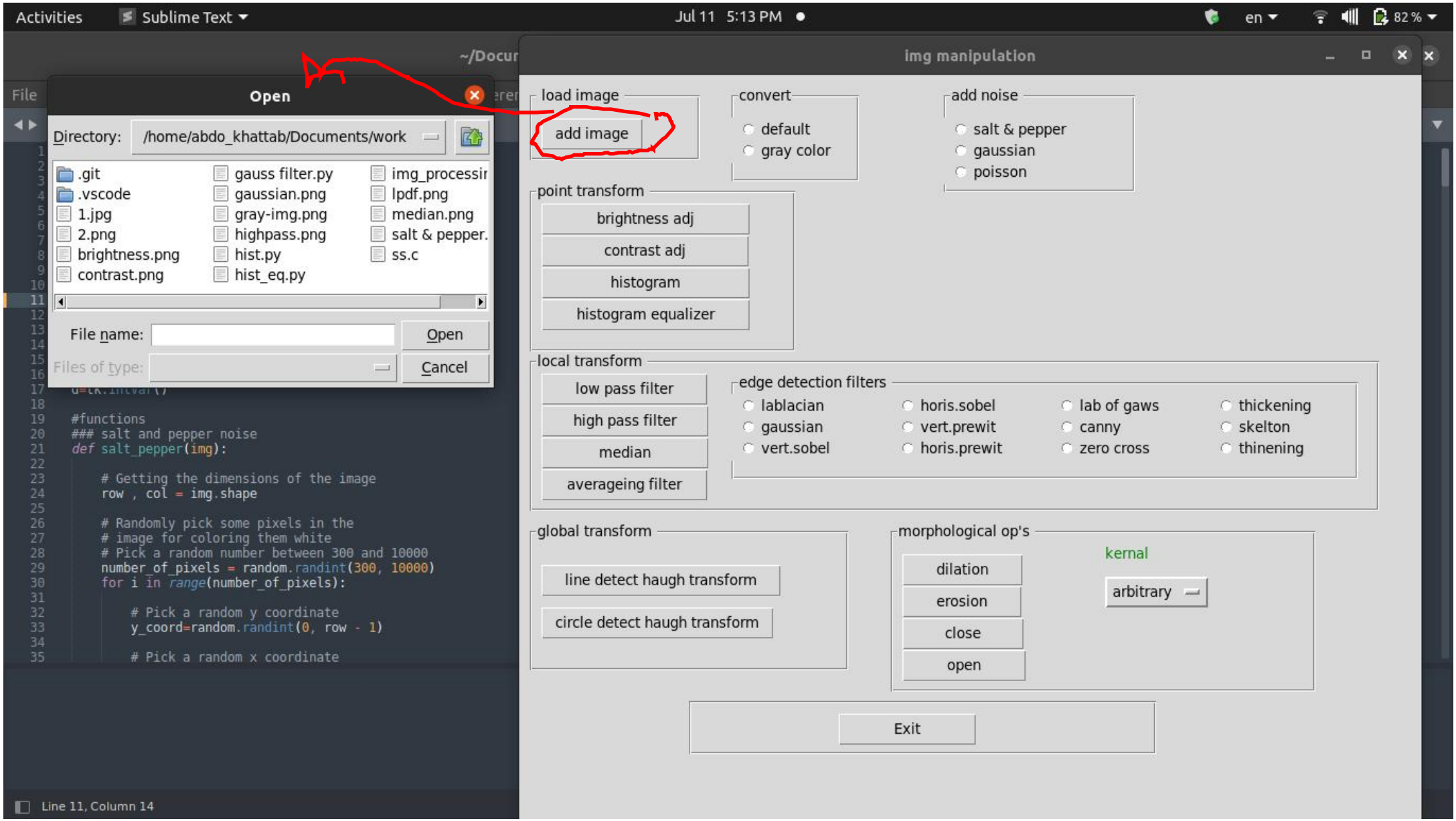
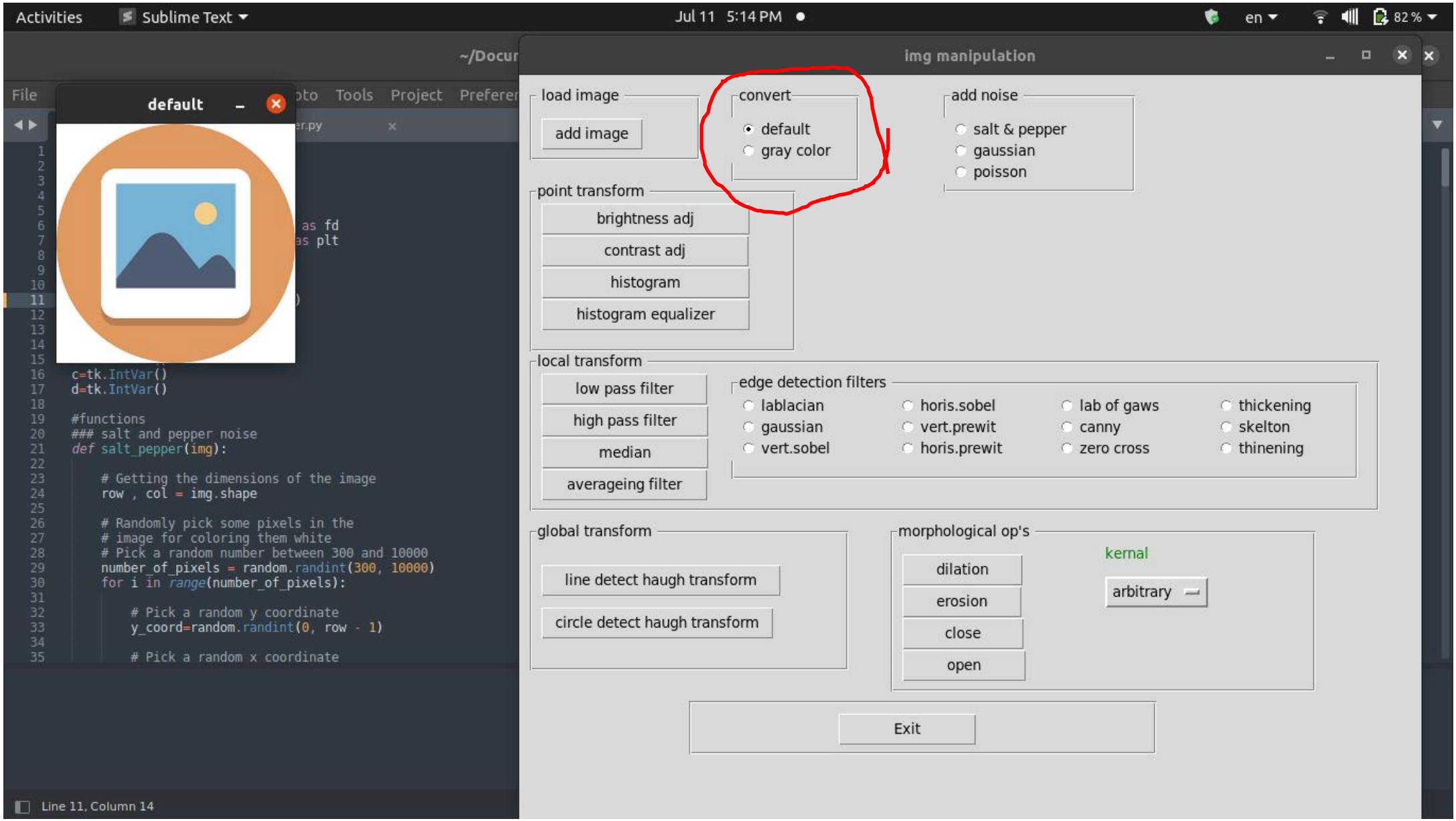
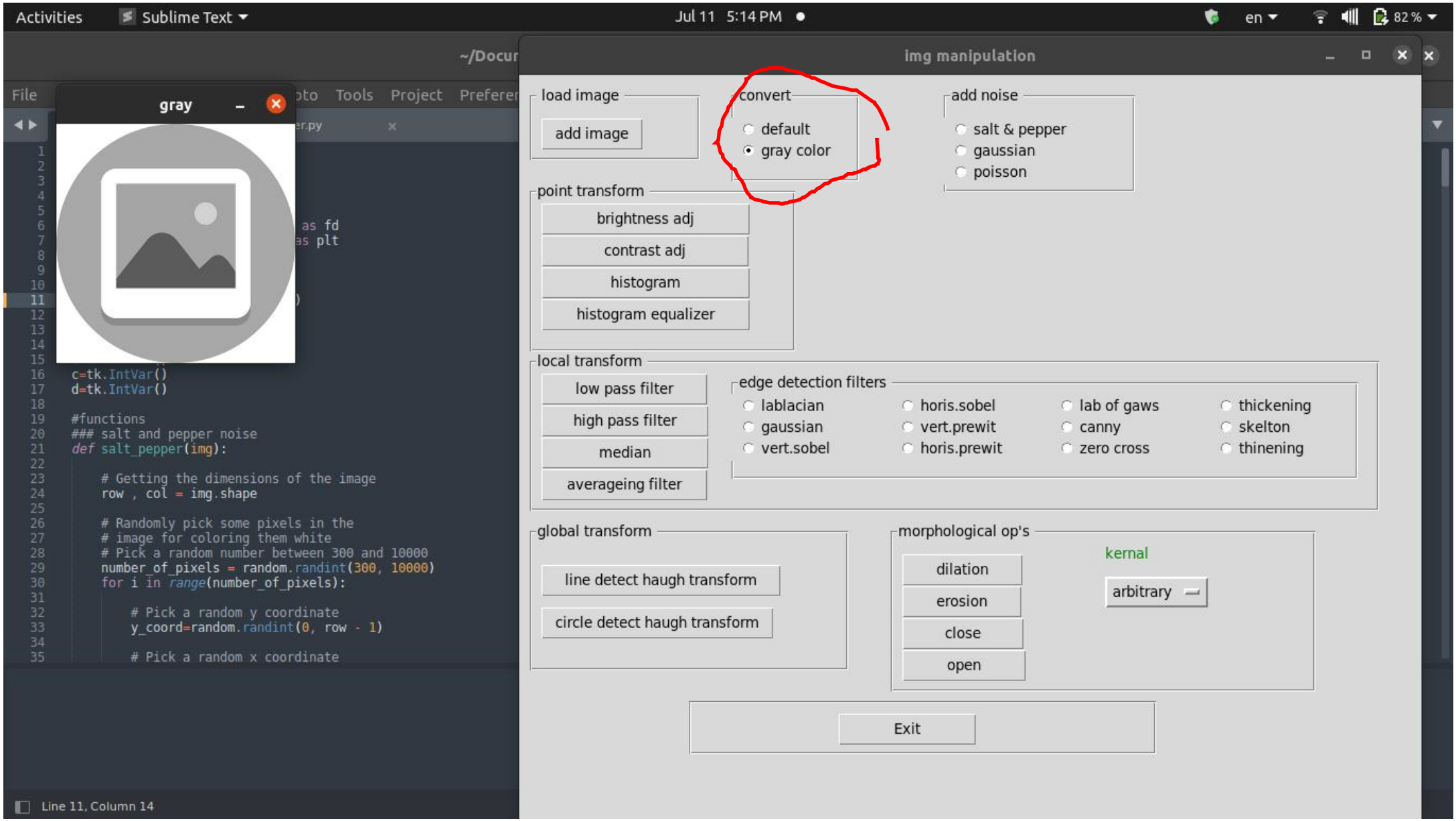
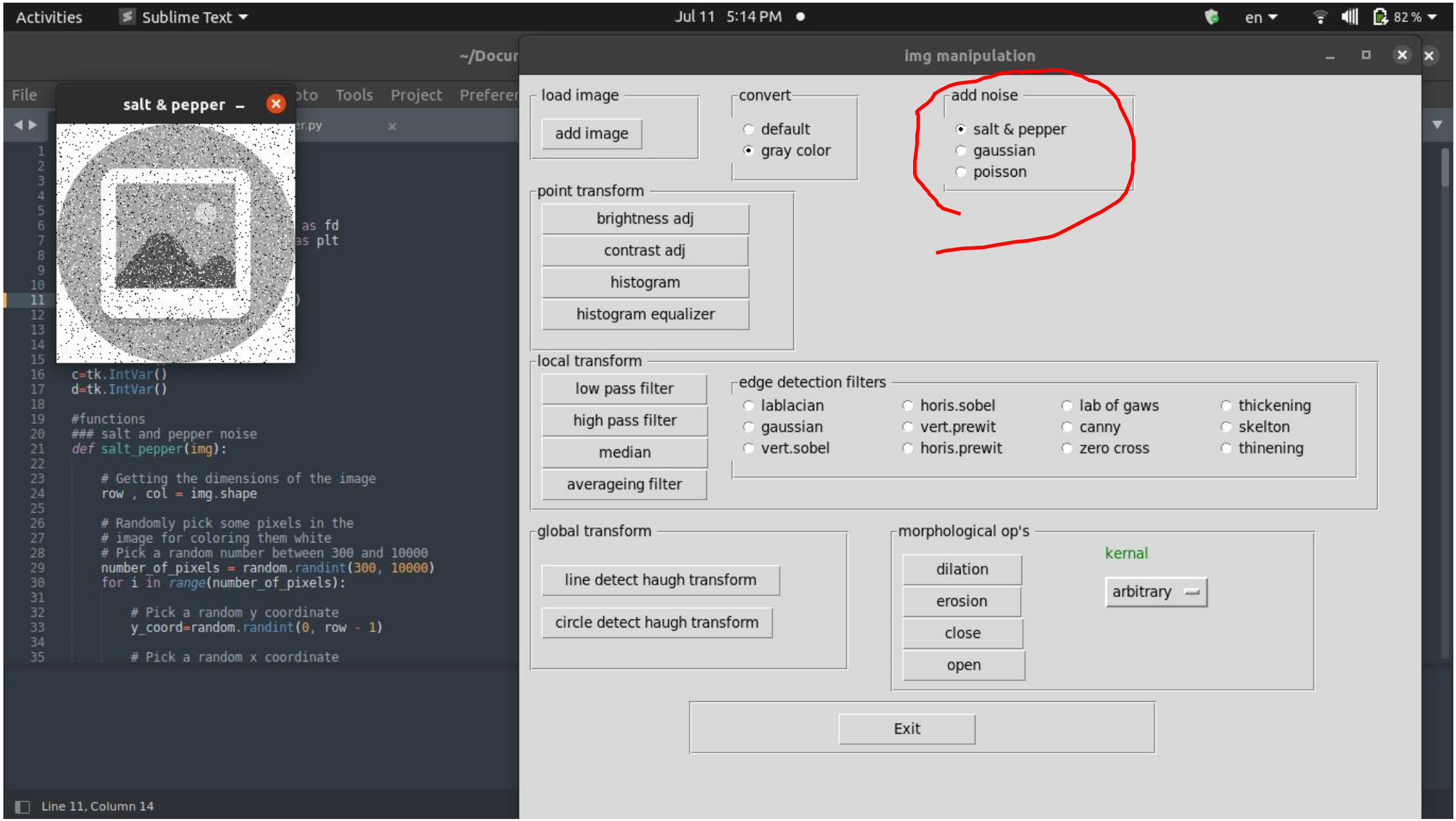
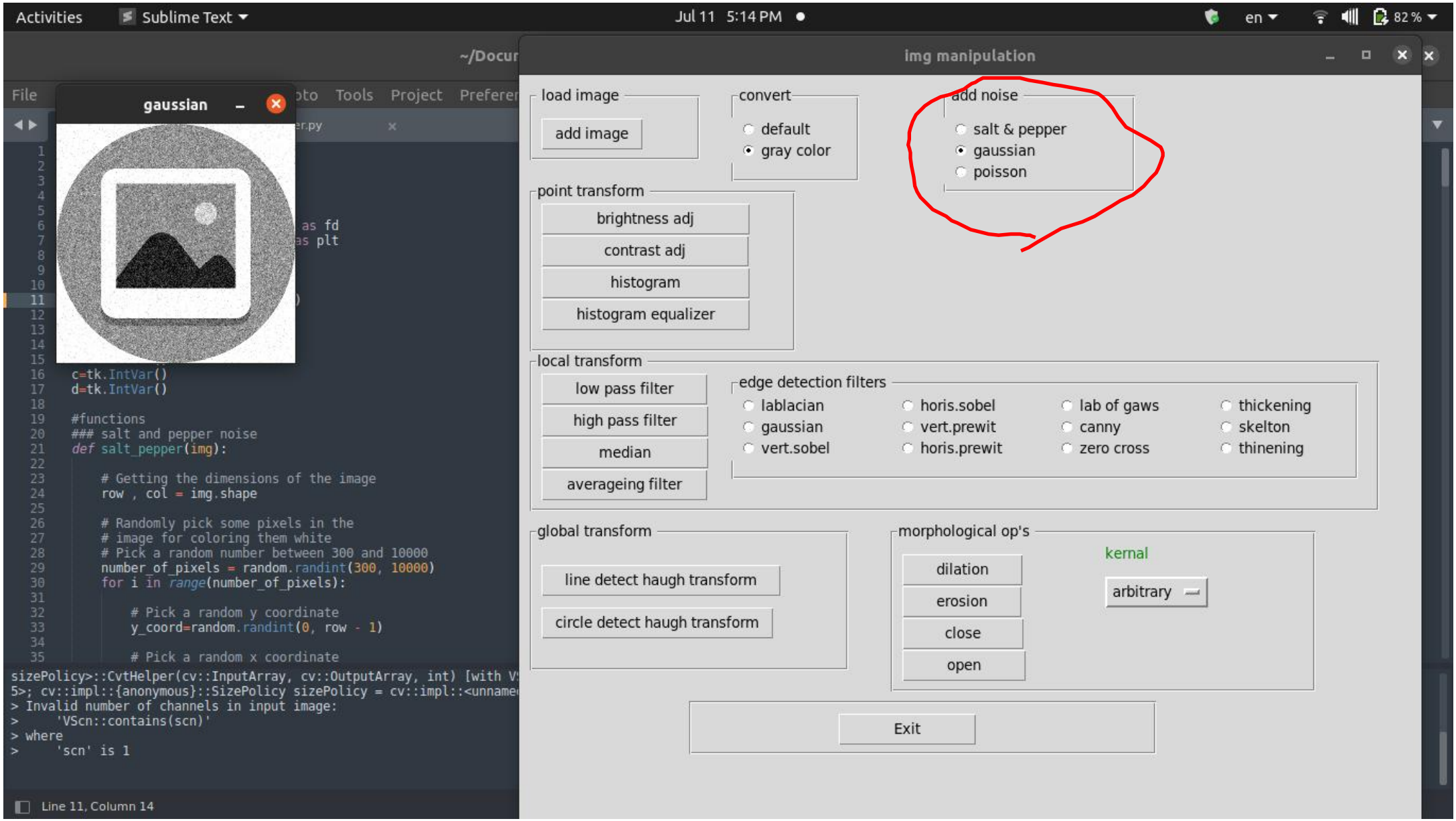

Activities
Sublime Text
Jul 11 5:13 PM
en
82 %
~/Docur
Open
Directory:
/home/abdo_khattab/Documents/work
.git
.vscode
1.jpg
2.png
brightness.png
contrast.png
gauss filter.py
gaussian.png
gray-img.png
highpass.png
hist.py
hist_eq.py
img_processir
lpdf.png
median.png
salt & pepper.
ss.c
File name:
Open
Files of type:
Cancel
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of the image
row , col = img.shape
# Randomly pick some pixels in the
# image for coloring them white
# Pick a random number between 300 and 10000
number_of_pixels = random.randint(300, 10000)
for i in range(number_of_pixels):
# Pick a random y coordinate
y_coord=random.randint(0, row - 1)
# Pick a random x coordinate
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

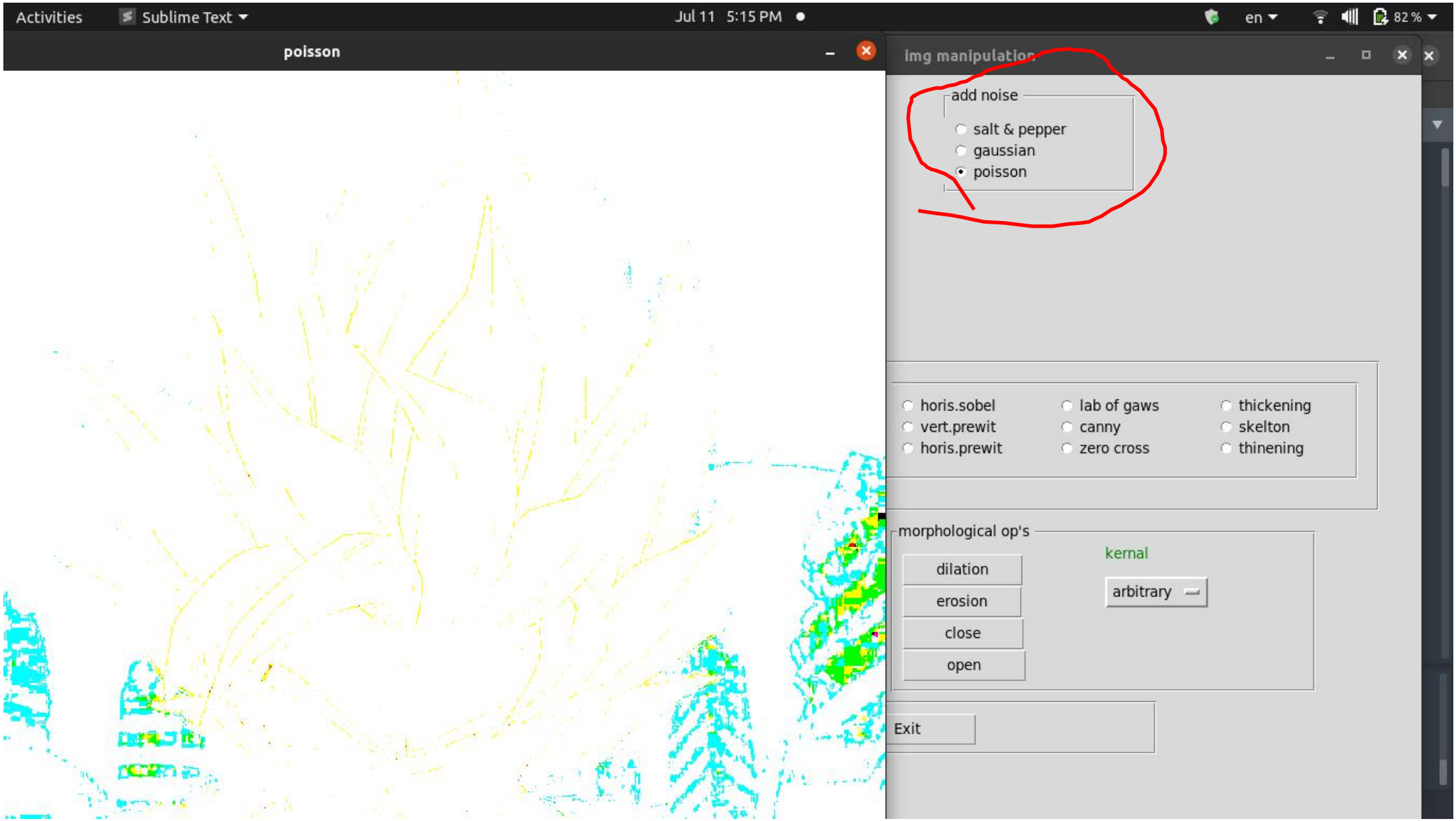

Activities
Sublime Text
Jul 11 5:14 PM
en
82 %
default
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit
Line 11, Column 14

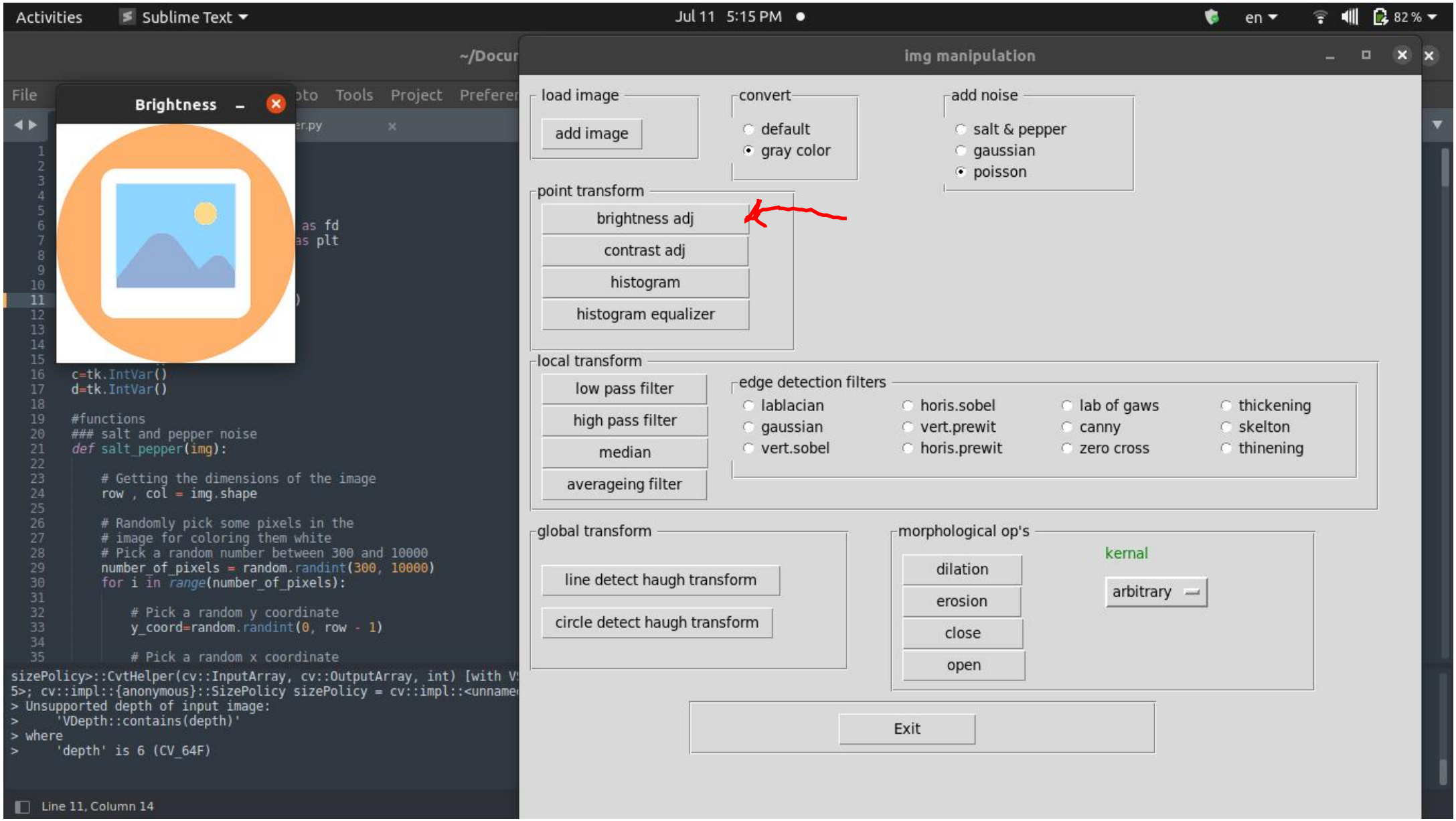

Activities
Sublime Text
Jul 11 5:14 PM
en
82 %
~/Docur
File
Tools
Project
Preferer
gray
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of the image
row , col = img.shape
# Randomly pick some pixels in the
# image for coloring them white
# Pick a random number between 300 and 10000
number_of_pixels = random.randint(300, 10000)
for i in range(number_of_pixels):
# Pick a random y coordinate
y_coord=random.randint(0, row - 1)
# Pick a random x coordinate
Line 11, Column 14

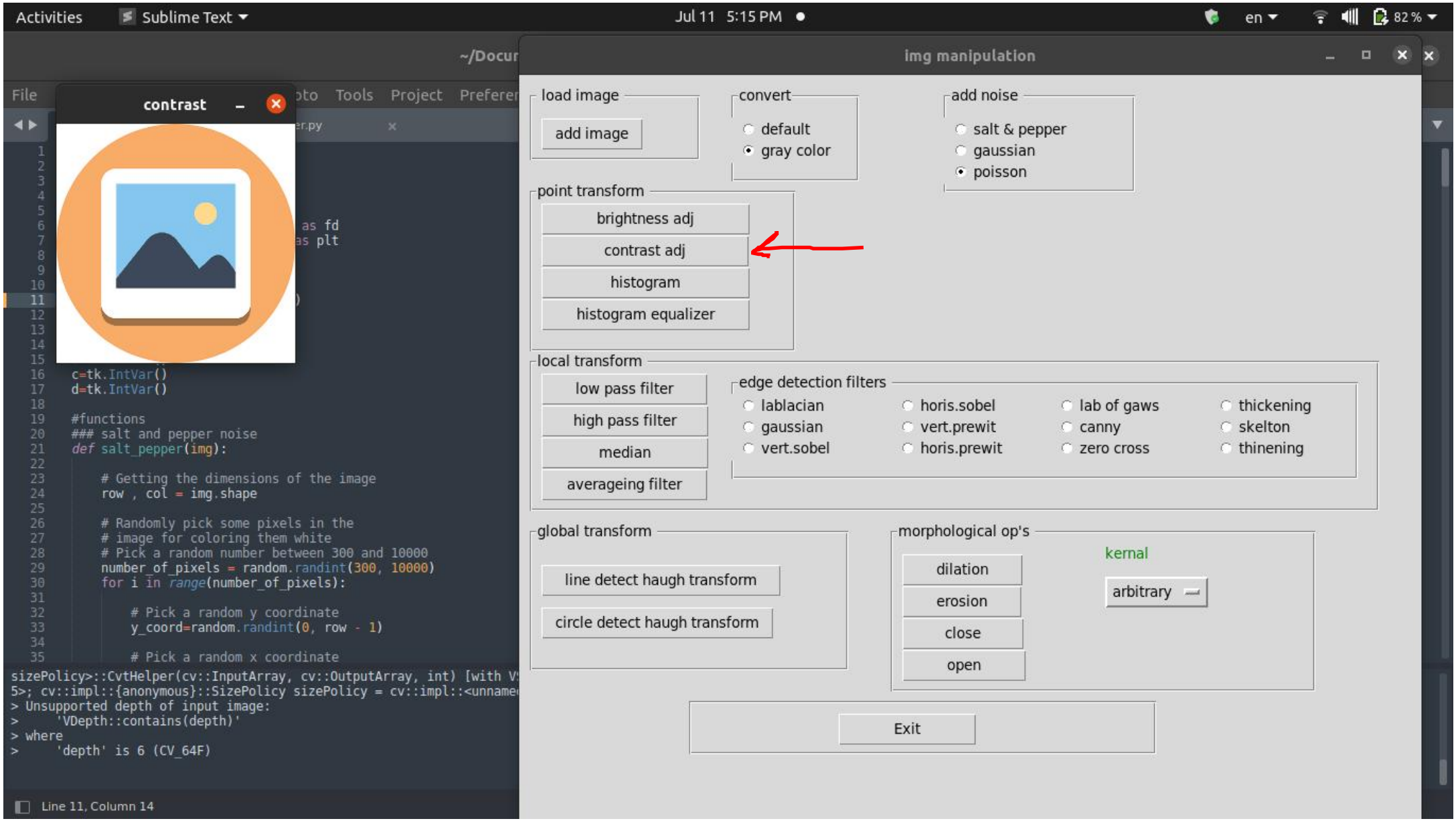


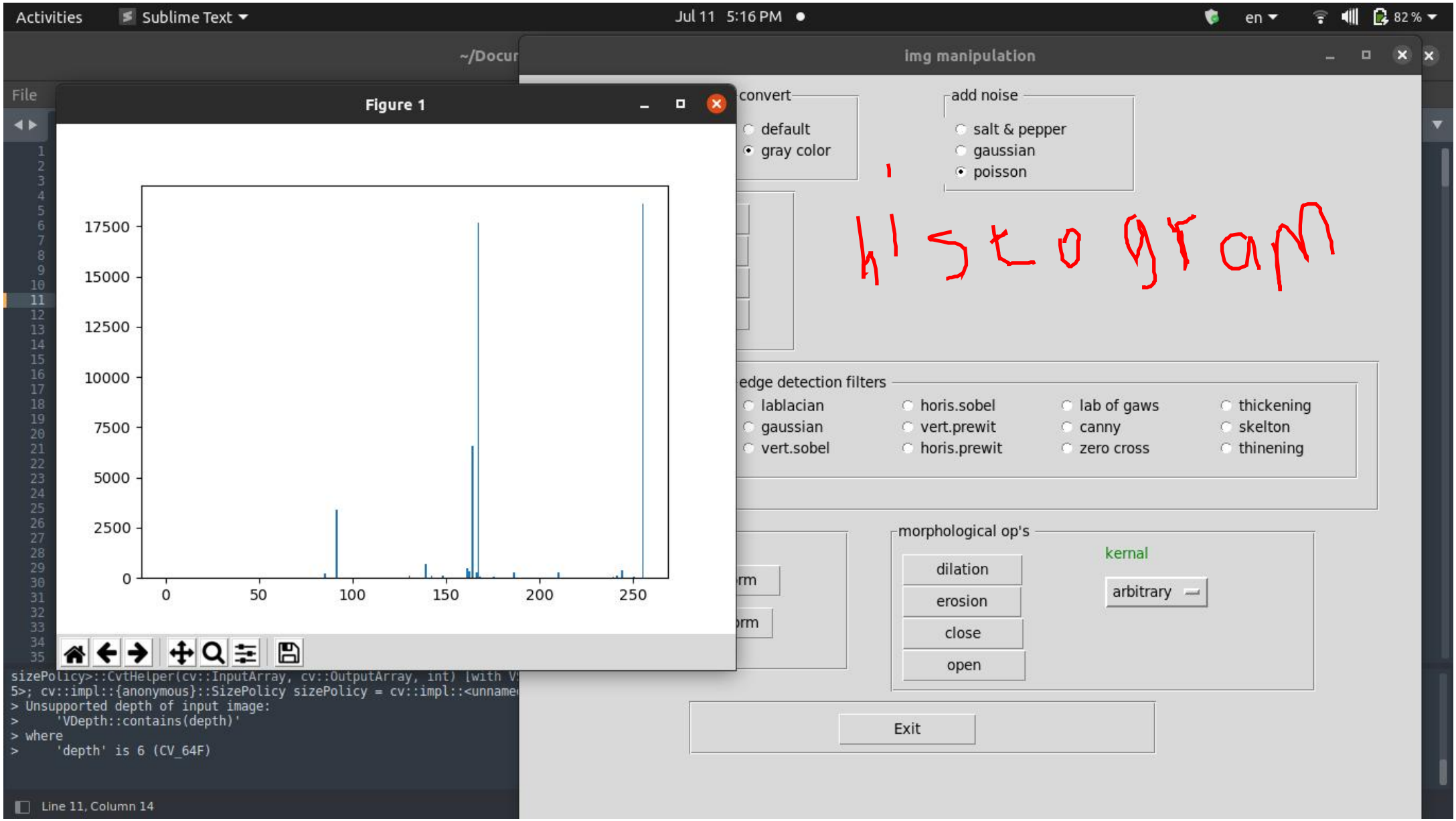

```
c=tk.IntVar()
d=tk.IntVar()

#functions
### salt and pepper noise
def salt_pepper(img):

    # Getting the dimensions of the image
    row , col = img.shape

    # Randomly pick some pixels in the
    # image for coloring them white
    # Pick a random number between 300 and 10000
    number_of_pixels = random.randint(300, 10000)
    for i in range(number_of_pixels):

        # Pick a random y coordinate
        y_coord=random.randint(0, row - 1)

        # Pick a random x coordinate
```

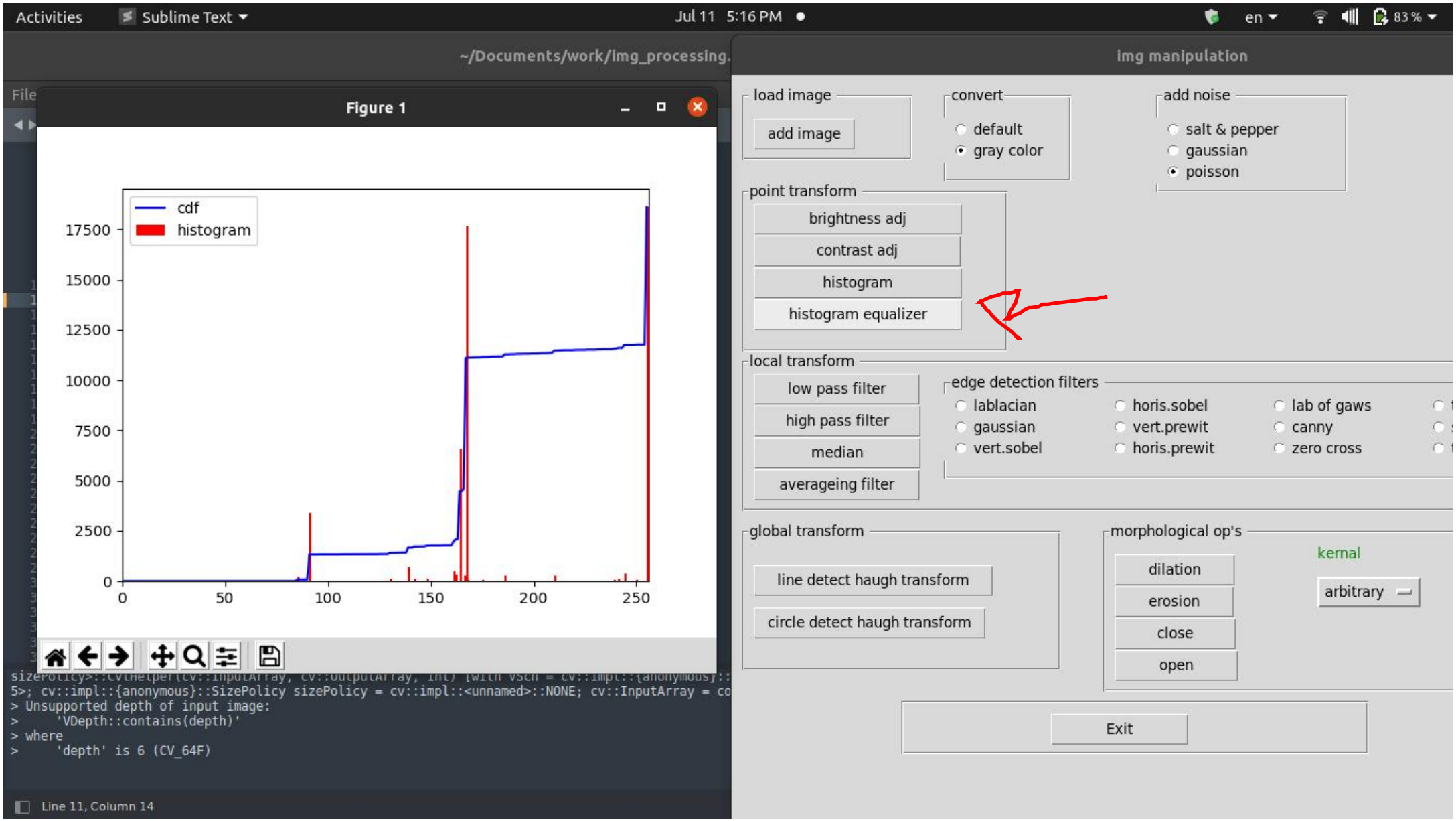

Activities
Sublime Text
Jul 11 5:14 PM
en
82 %
~/Docur
File
Tools
Project
gaussian
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of the image
row , col = img.shape
# Randomly pick some pixels in the
# image for coloring them white
# Pick a random number between 300 and 10000
number_of_pixels = random.randint(300, 10000)
for i in range(number_of_pixels):
# Pick a random y coordinate
y_coord=random.randint(0, row - 1)
# Pick a random x coordinate
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with V
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unname
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14

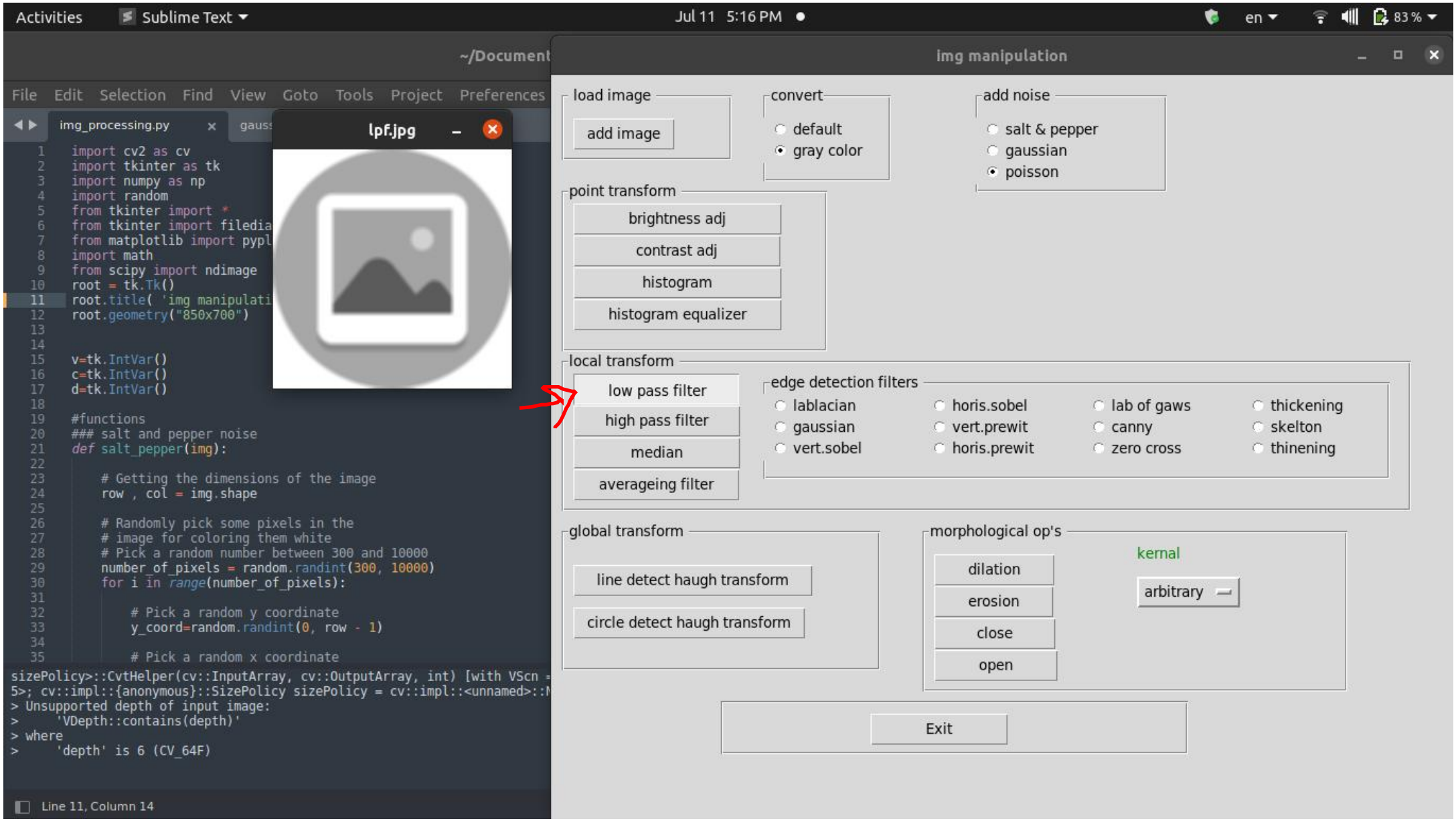

Activities
Sublime Text
Jul 11 5:15 PM
en
82 %
poisson
img manipulation
add noise
salt & pepper
gaussian
poisson
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

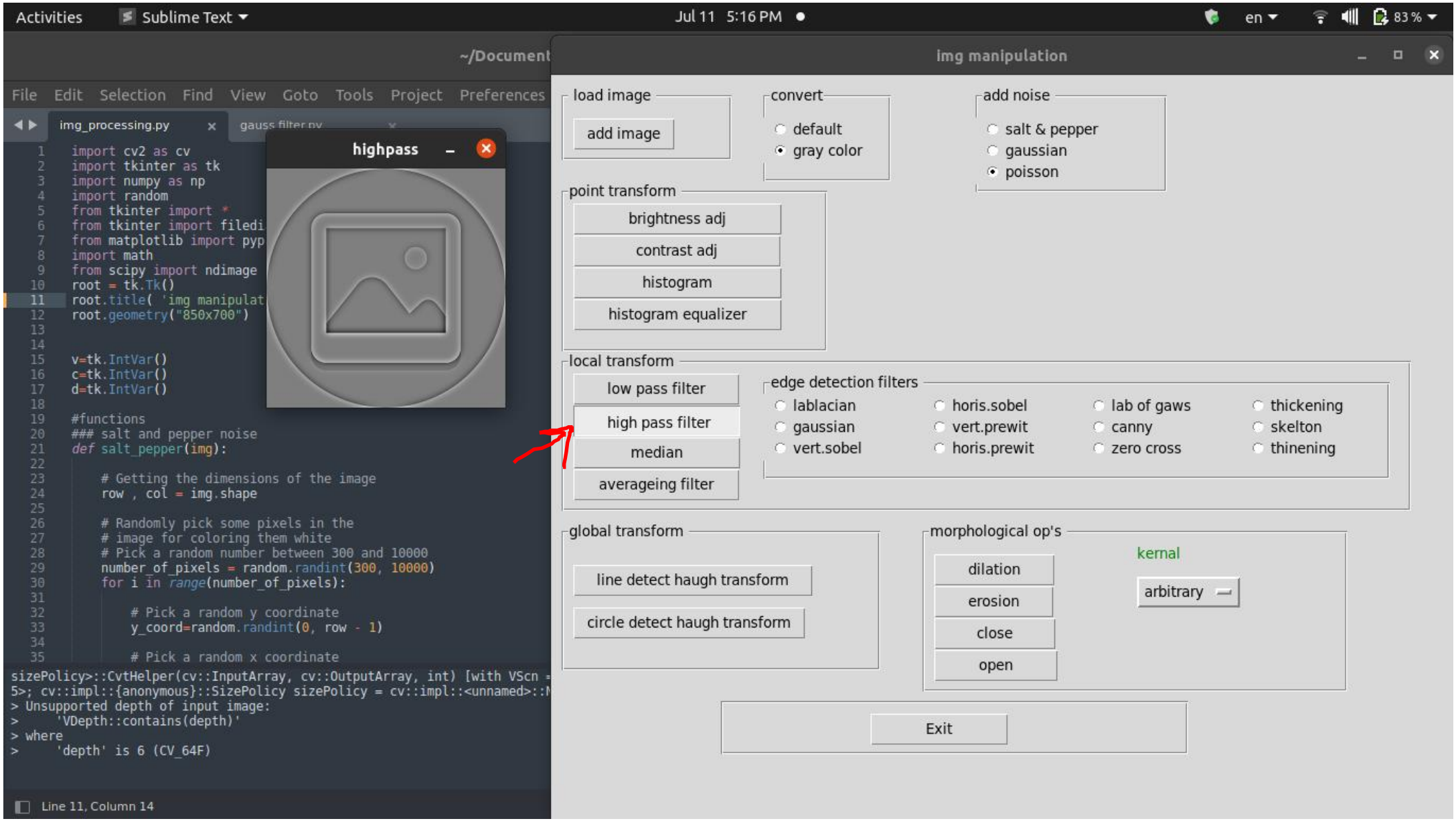

Activities
Sublime Text
Jul 11 5:15 PM
en
82 %
~/Docur
File
Tools
Project
Brightness
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of the image
row , col = img.shape
# Randomly pick some pixels in the
# image for coloring them white
# Pick a random number between 300 and 10000
number_of_pixels = random.randint(300, 10000)
for i in range(number_of_pixels):
# Pick a random y coordinate
y_coord=random.randint(0, row - 1)
# Pick a random x coordinate
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with V
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unname
> Unsupported depth of input image:
> 'VDepth::contains(depth)'
> where
> 'depth' is 6 (CV_64F)
Line 11, Column 14

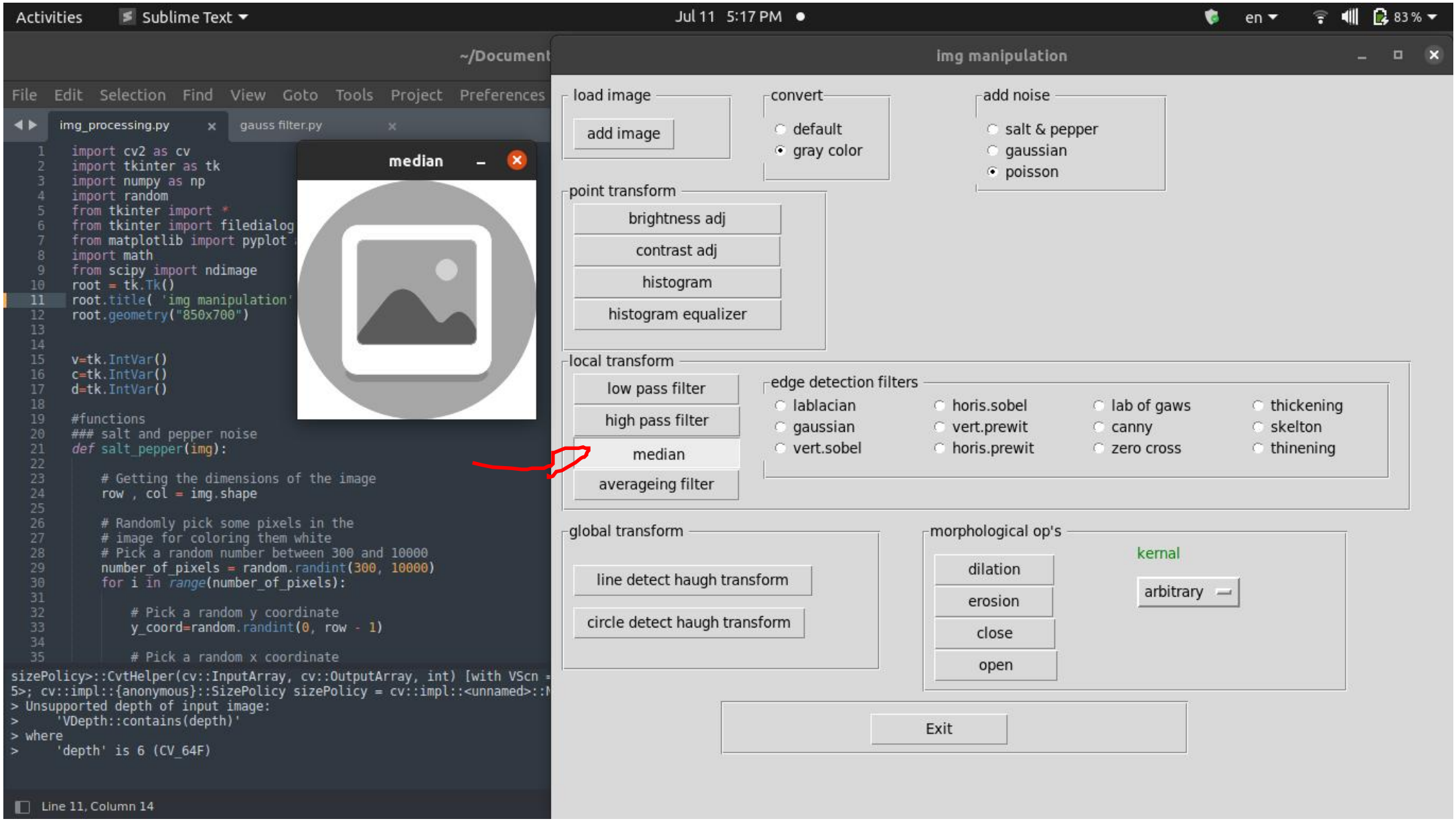

Activities
Sublime Text
Jul 11 5:15 PM
en
82 %
~/Docur
File
Tools
Project
Preferer
contrast
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of the image
row , col = img.shape
# Randomly pick some pixels in the
# image for coloring them white
# Pick a random number between 300 and 10000
number_of_pixels = random.randint(300, 10000)
for i in range(number_of_pixels):
# Pick a random y coordinate
y_coord=random.randint(0, row - 1)
# Pick a random x coordinate
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with V
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unname
> Unsupported depth of input image:
> 'VDepth::contains(depth)'
> where
> 'depth' is 6 (CV_64F)
Line 11, Column 14

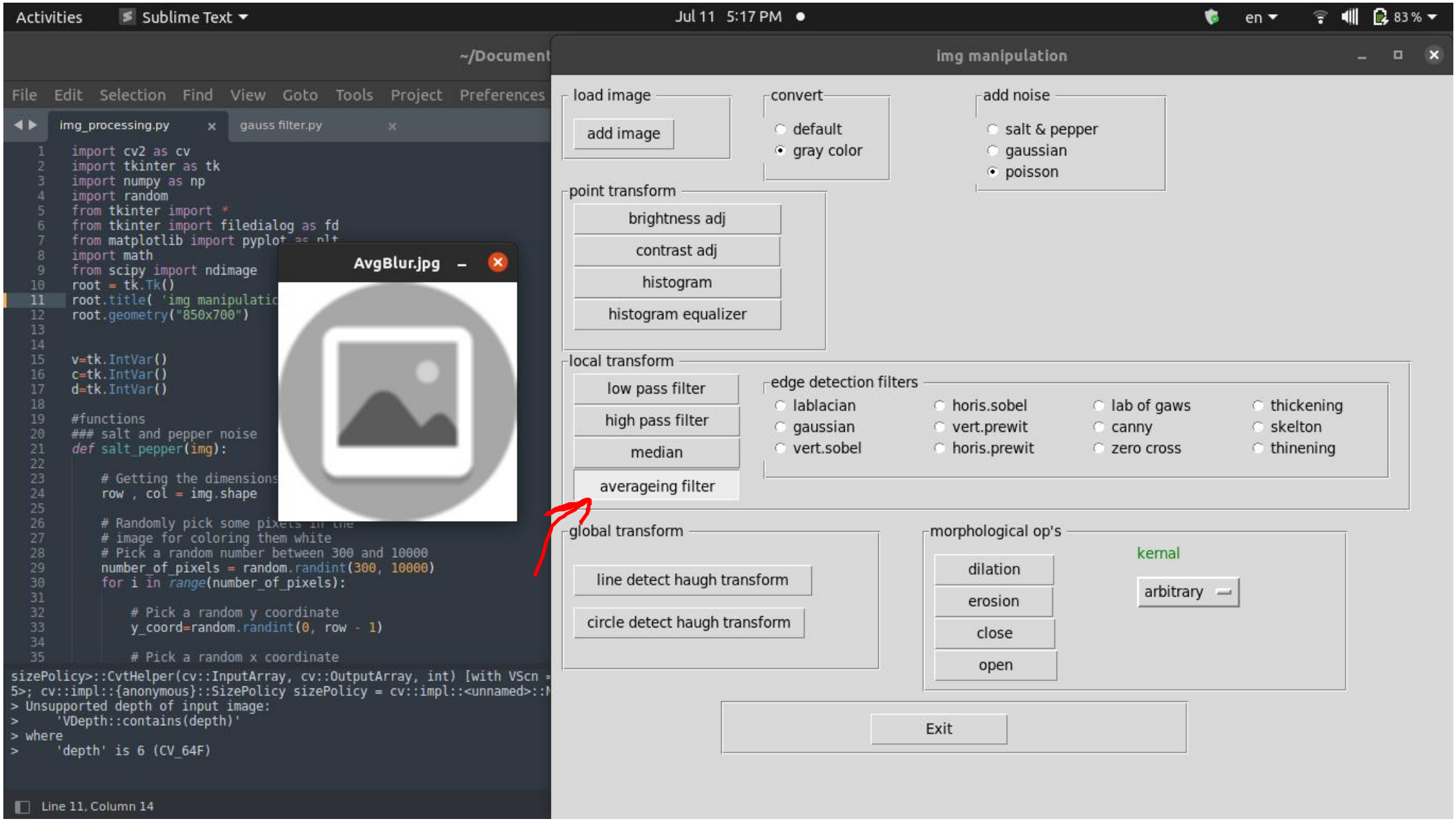

Activities
Sublime Text
Jul 11 5:16 PM
en
82 %
~/Docur
File
img manipulation
Figure 1
17500
15000
12500
10000
7500
5000
2500
0
0
50
100
150
200
250
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
histogram
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with V
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unname
> Unsupported depth of input image:
>   'VDepth::contains(depth)'
> where
>   'depth' is 6 (CV_64F)
Line 11, Column 14

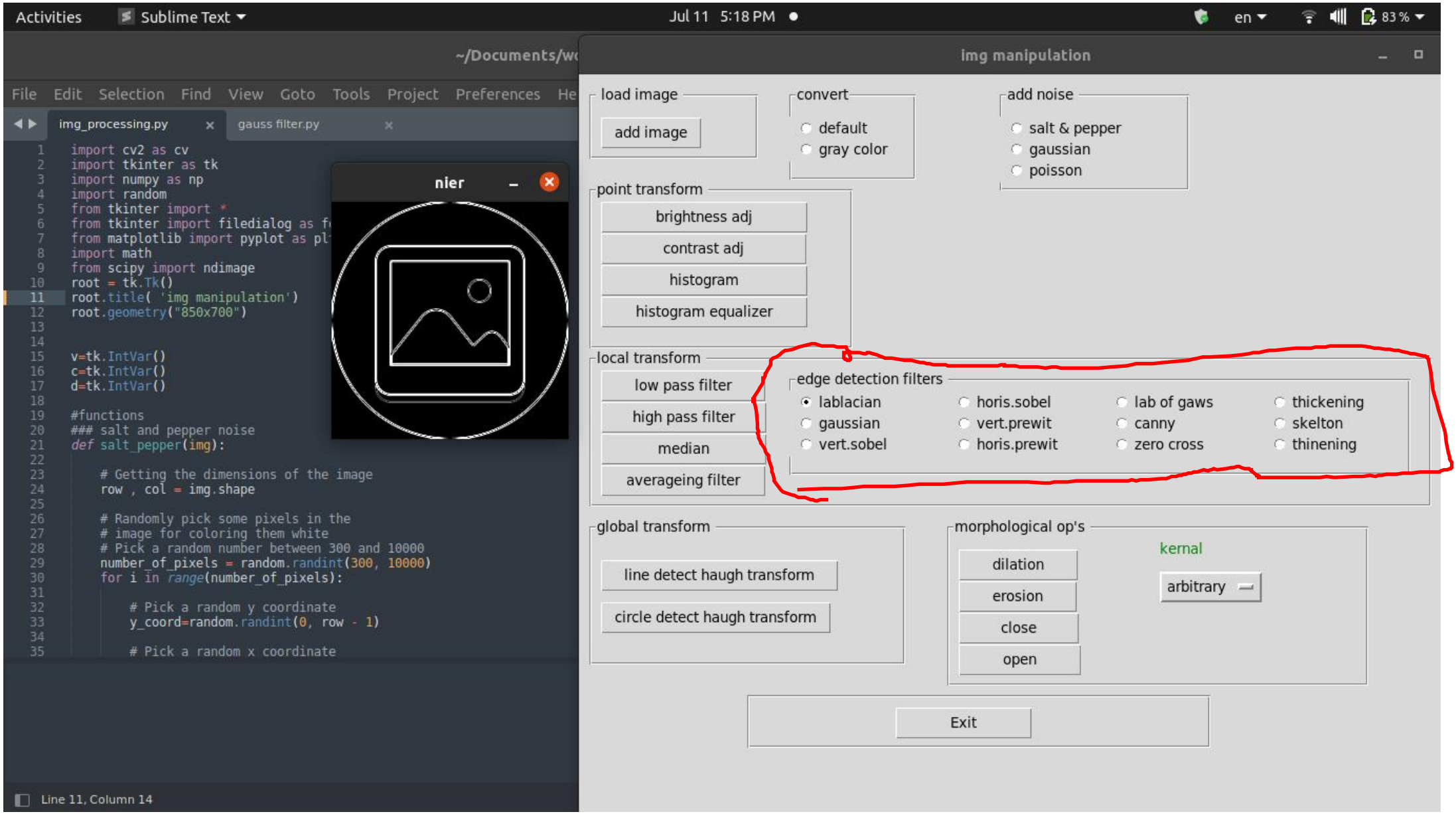

Activities
Sublime Text
Jul 11 5:16 PM
en
83 %
~/Documents/work/img_processing.
img manipulation
Figure 1
cdf
histogram
17500
15000
12500
10000
7500
5000
2500
0
0
50
100
150
200
250
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
kernal
arbitrary
dilation
erosion
close
open
Exit
> Unsupported depth of input image:
> 'VDepth::contains(depth)'
> where
> 'depth' is 6 (CV_64F)
Line 11, Column 14

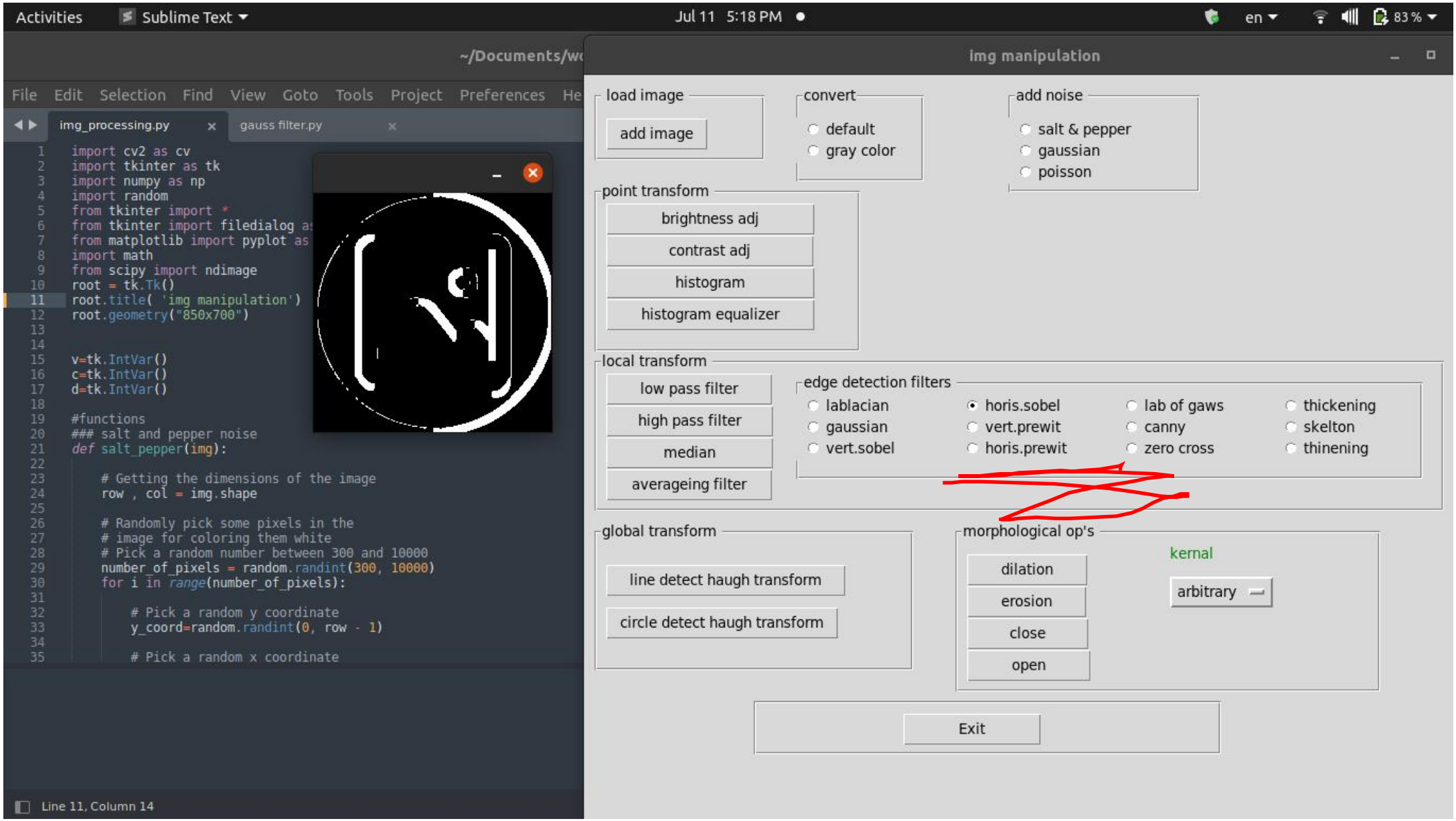

Activities
Sublime Text
Jul 11 5:16 PM
en
83 %
~/Document
File Edit Selection Find View Goto Tools Project Preferences
img_processing.py
lpf.jpg
import cv2 as cv
import tkinter as tk
import numpy as np
import random
from tkinter import *
from tkinter import filedia
from matplotlib import pypl
import math
from scipy import ndimage
root = tk.Tk()
root.title('img manipulati
root.geometry("850x700")
v=tk.IntVar()
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of the image
row , col = img.shape
# Randomly pick some pixels in the
# image for coloring them white
# Pick a random number between 300 and 10000
number_of_pixels = random.randint(300, 10000)
for i in range(number_of_pixels):
# Pick a random y coordinate
y_coord=random.randint(0, row - 1)
# Pick a random x coordinate
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn =
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::N
> Unsupported depth of input image:
> 'VDepth::contains(depth)'
> where
> 'depth' is 6 (CV_64F)
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

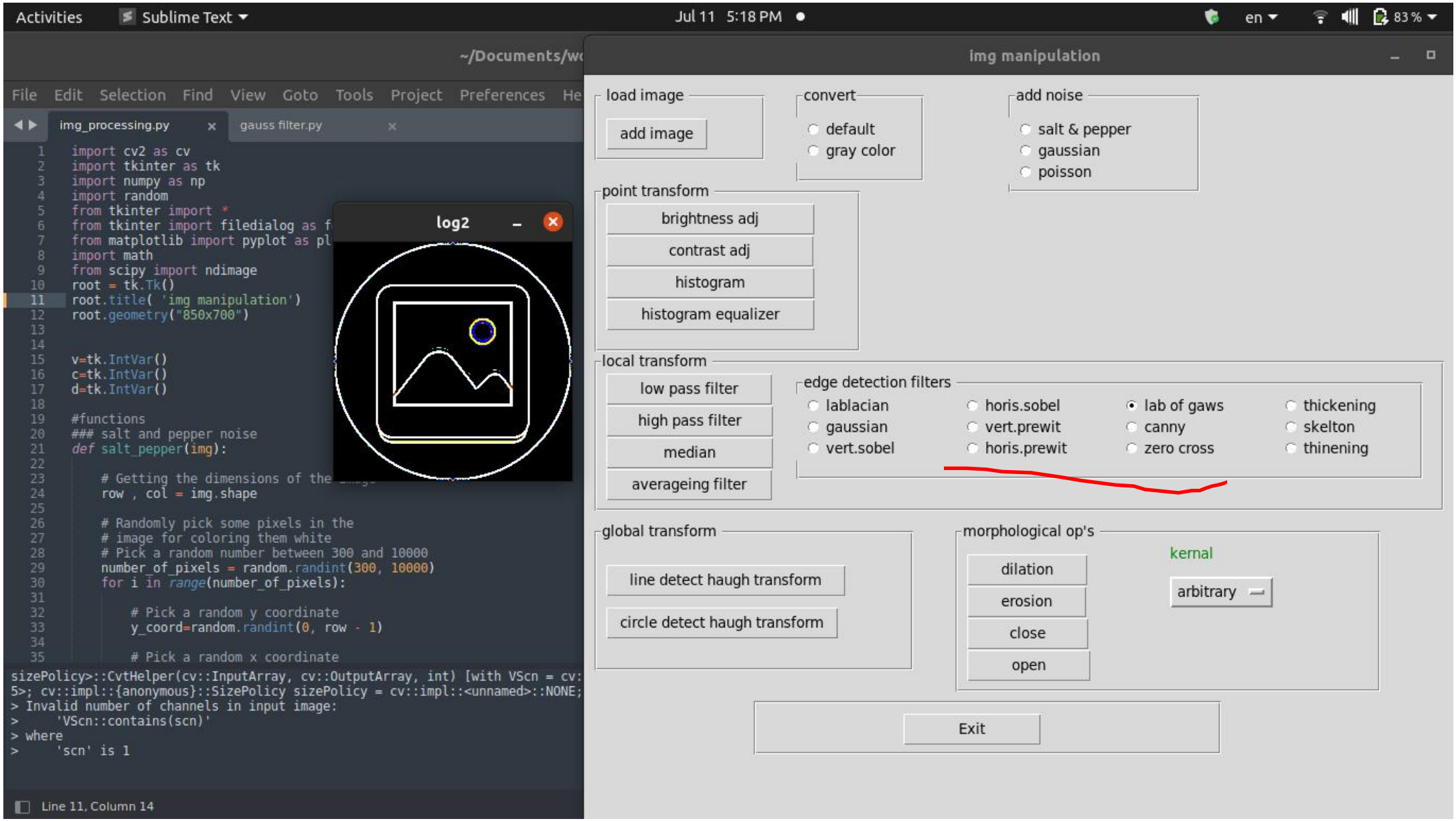

Activities
Sublime Text
Jul 11 5:16 PM
en
83 %
~/Document
File Edit Selection Find View Goto Tools Project Preferences
img_processing.py
gauss filter.py
highpass
import cv2 as cv
import tkinter as tk
import numpy as np
import random
from tkinter import *
from tkinter import filedi
from matplotlib import pyp
import math
from scipy import ndimage
root = tk.Tk()
root.title('img manipulat
root.geometry("850x700")
v=tk.IntVar()
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of the image
row , col = img.shape
# Randomly pick some pixels in the
# image for coloring them white
# Pick a random number between 300 and 10000
number_of_pixels = random.randint(300, 10000)
for i in range(number_of_pixels):
# Pick a random y coordinate
y_coord=random.randint(0, row - 1)
# Pick a random x coordinate
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn =
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::N
> Unsupported depth of input image:
> 'VDepth::contains(depth)'
> where
> 'depth' is 6 (CV_64F)
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

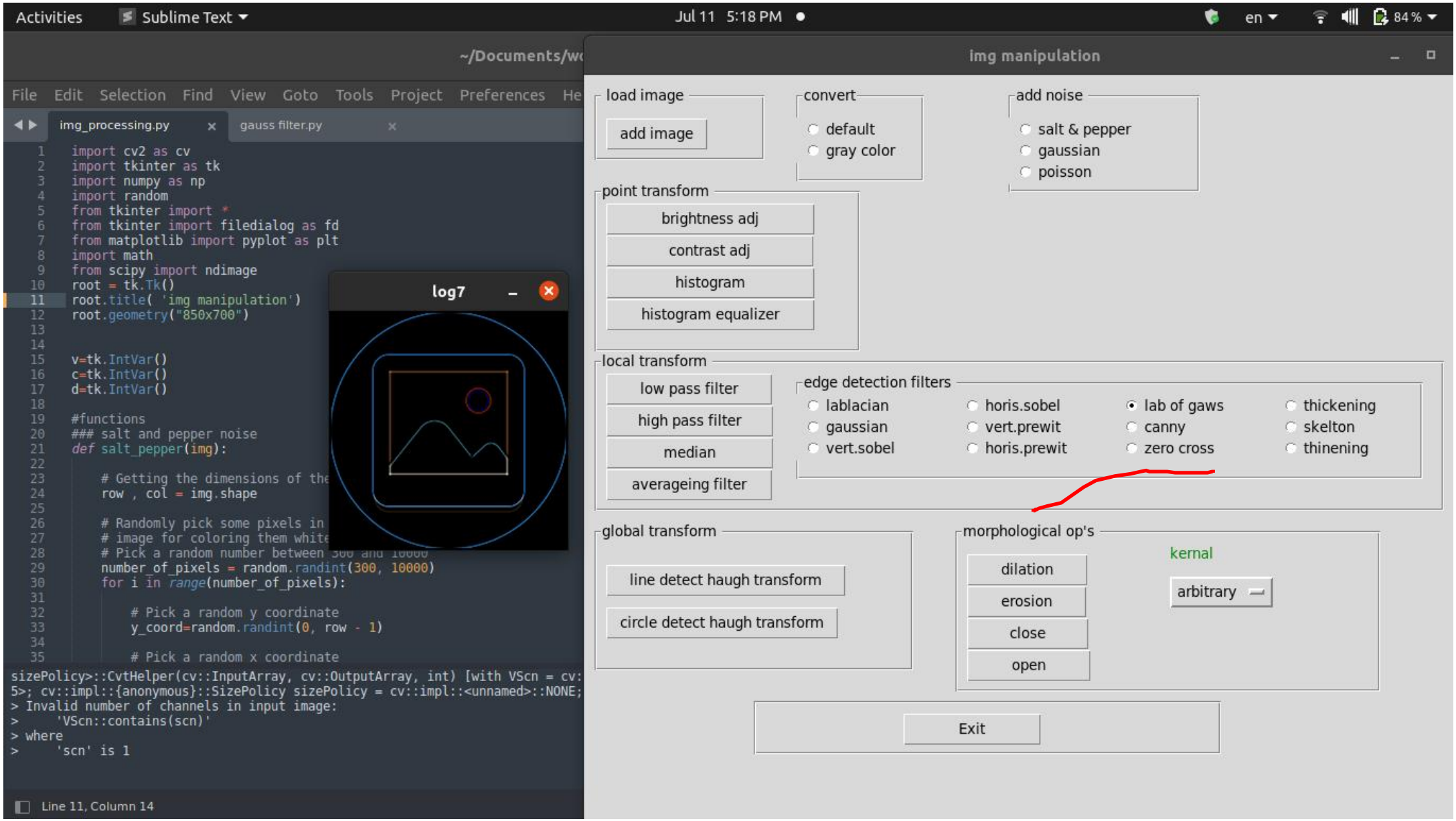

Activities
Sublime Text
Jul 11 5:17 PM
en
83 %
~/Document
File Edit Selection Find View Goto Tools Project Preferences
img_processing.py
gauss filter.py
median
import cv2 as cv
import tkinter as tk
import numpy as np
import random
from tkinter import *
from tkinter import filedialog
from matplotlib import pyplot
import math
from scipy import ndimage
root = tk.Tk()
root.title('img manipulation'
root.geometry("850x700")
v=tk.IntVar()
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of the image
row , col = img.shape
# Randomly pick some pixels in the
# image for coloring them white
# Pick a random number between 300 and 10000
number_of_pixels = random.randint(300, 10000)
for i in range(number_of_pixels):
# Pick a random y coordinate
y_coord=random.randint(0, row - 1)
# Pick a random x coordinate
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn =
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::N
> Unsupported depth of input image:
> 'VDepth::contains(depth)'
> where
> 'depth' is 6 (CV_64F)
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

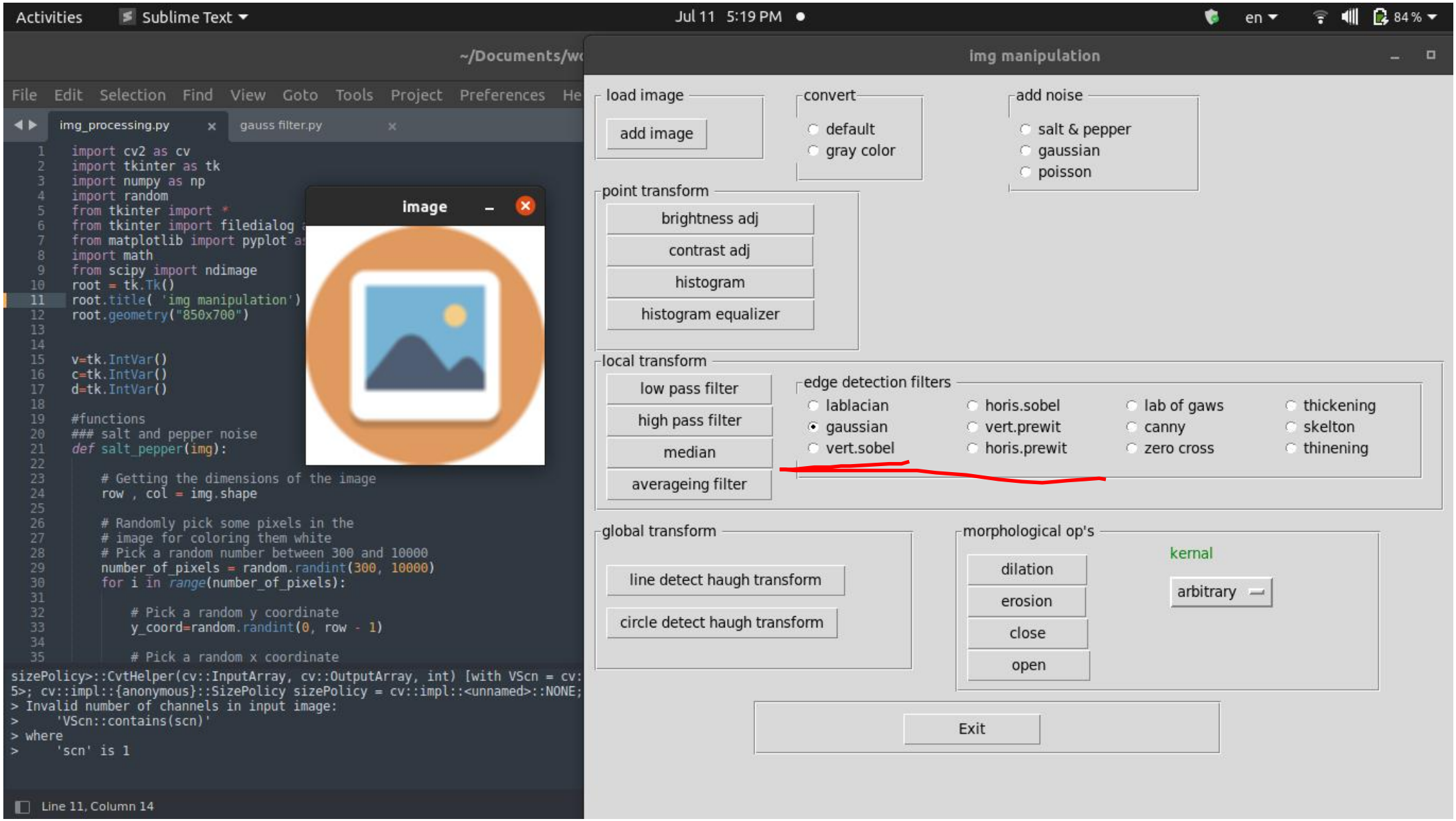

Activities
Sublime Text
Jul 11 5:17 PM
en
83 %
~/Document
File Edit Selection Find View Goto Tools Project Preferences
img_processing.py
gauss filter.py
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedialog as fd
7 from matplotlib import pyplot as plt
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulatio
12 root.geometry("850x700")
13
14
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
18
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
22
23 # Getting the dimensions
24 row , col = img.shape
25
26 # Randomly pick some pixels in the
27 # image for coloring them white
28 # Pick a random number between 300 and 10000
29 number_of_pixels = random.randint(300, 10000)
30 for i in range(number_of_pixels):
31
32 # Pick a random y coordinate
33 y_coord=random.randint(0, row - 1)
34
35 # Pick a random x coordinate
AvgBlur.jpg
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn =
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::N
> Unsupported depth of input image:
> 'VDepth::contains(depth)'
> where
> 'depth' is 6 (CV_64F)
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

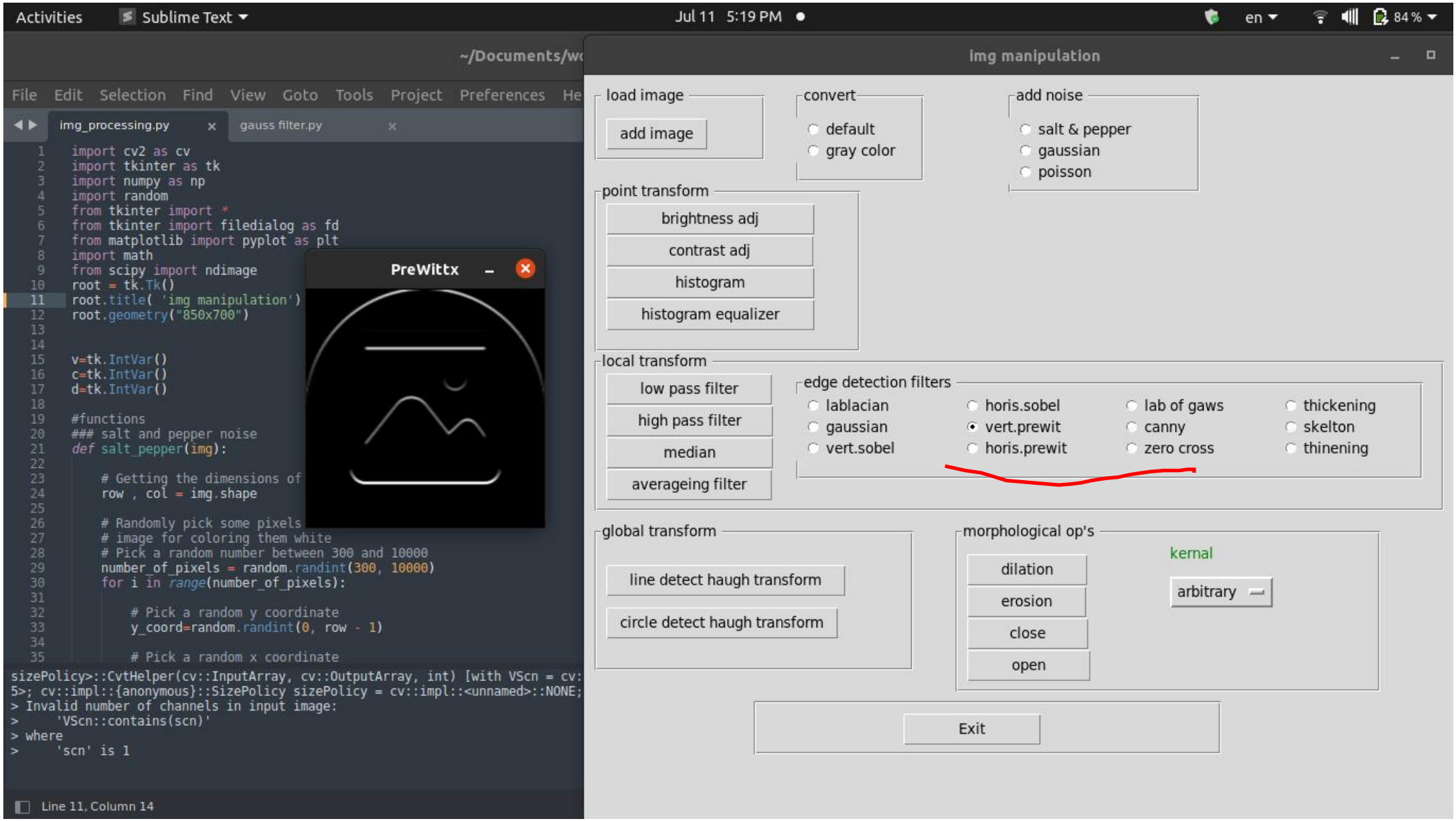

Activities
Sublime Text
Jul 11 5:18 PM
en
83 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedialog as f
7 from matplotlib import pyplot as pl
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulation')
12 root.geometry("850x700")
13
14
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
18
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
22
23 # Getting the dimensions of the image
24 row , col = img.shape
25
26 # Randomly pick some pixels in the
27 # image for coloring them white
28 # Pick a random number between 300 and 10000
29 number_of_pixels = random.randint(300, 10000)
30 for i in range(number_of_pixels):
31
32 # Pick a random y coordinate
33 y_coord=random.randint(0, row - 1)
34
35 # Pick a random x coordinate
nier
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

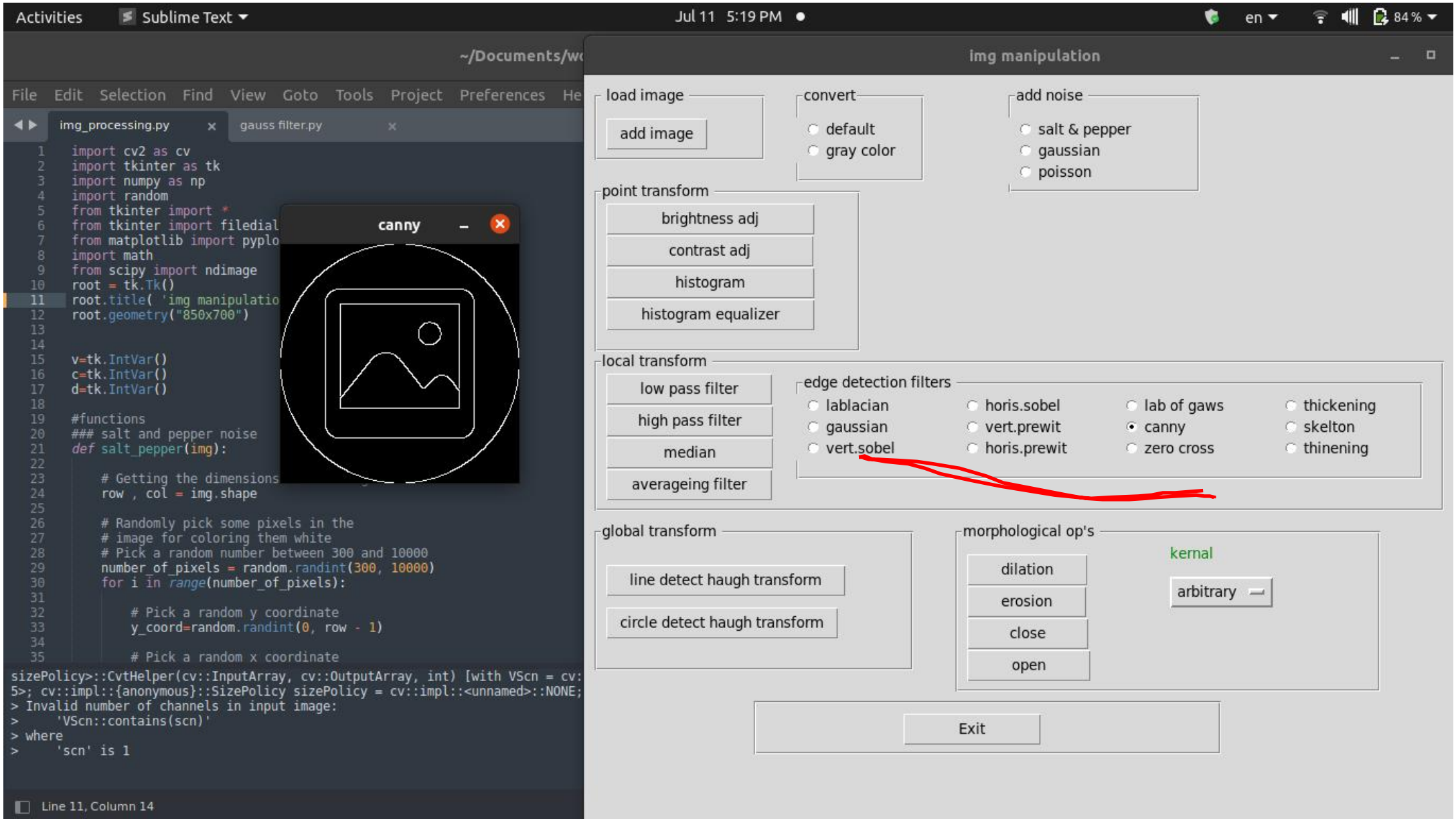

Activities
Sublime Text
Jul 11 5:18 PM
en
83 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedialog a
7 from matplotlib import pyplot as
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulation')
12 root.geometry("850x700")
13
14
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
18
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
22
23 # Getting the dimensions of the image
24 row , col = img.shape
25
26 # Randomly pick some pixels in the
27 # image for coloring them white
28 # Pick a random number between 300 and 10000
29 number_of_pixels = random.randint(300, 10000)
30 for i in range(number_of_pixels):
31
32 # Pick a random y coordinate
33 y_coord=random.randint(0, row - 1)
34
35 # Pick a random x coordinate
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

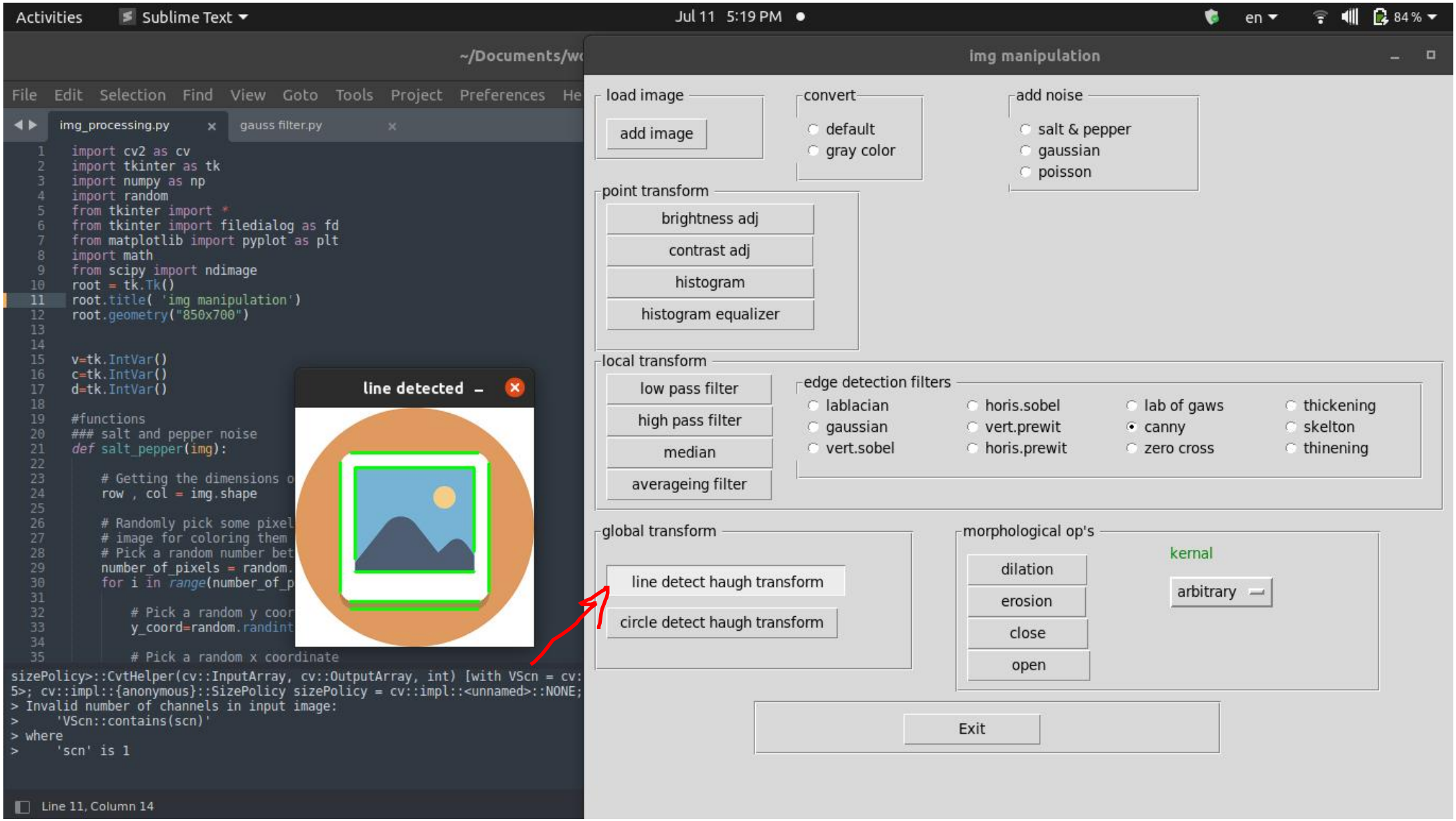

Activities
Sublime Text
Jul 11 5:18 PM
en
83 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedialog as f
7 from matplotlib import pyplot as pl
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulation')
12 root.geometry("850x700")
13
14
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
18
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
22
23 # Getting the dimensions of the
24 row , col = img.shape
25
26 # Randomly pick some pixels in the
27 # image for coloring them white
28 # Pick a random number between 300 and 10000
29 number_of_pixels = random.randint(300, 10000)
30 for i in range(number_of_pixels):
31
32 # Pick a random y coordinate
33 y_coord=random.randint(0, row - 1)
34
35 # Pick a random x coordinate
log2
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn = cv:
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::NONE;
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

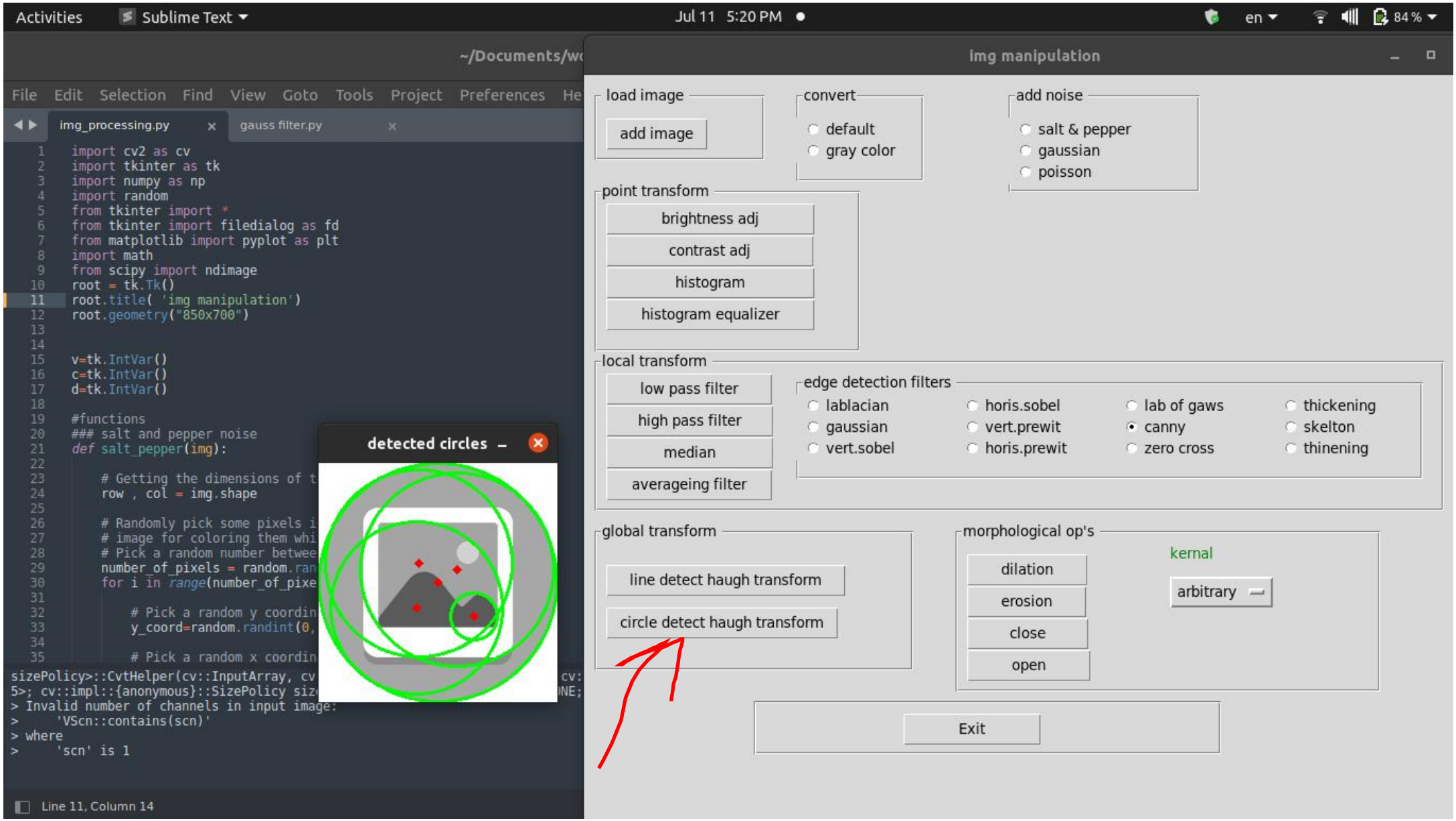

Activities
Sublime Text
Jul 11 5:18 PM
en
84 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedialog as fd
7 from matplotlib import pyplot as plt
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulation')
12 root.geometry("850x700")
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
23 # Getting the dimensions of the
24 row , col = img.shape
26 # Randomly pick some pixels in
27 # image for coloring them white
28 # Pick a random number between 300 and 10000
29 number_of_pixels = random.randint(300, 10000)
30 for i in range(number_of_pixels):
32 # Pick a random y coordinate
33 y_coord=random.randint(0, row - 1)
35 # Pick a random x coordinate
log7
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn = cv:
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::NONE;
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

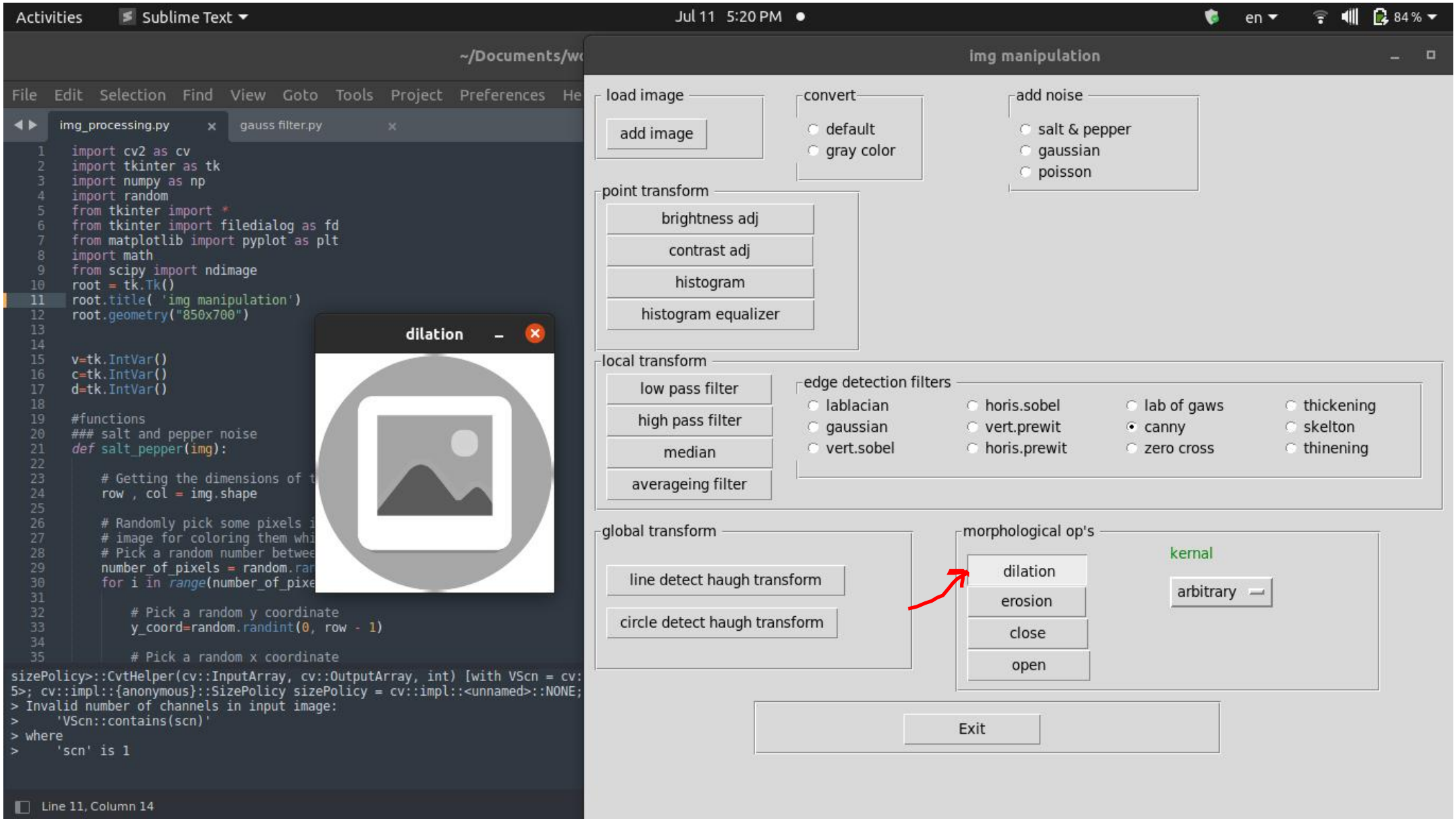

Activities
Sublime Text
Jul 11 5:19 PM
en
84 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedialog
7 from matplotlib import pyplot as
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulation')
12 root.geometry("850x700")
13
14
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
18
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
22
23 # Getting the dimensions of the image
24 row , col = img.shape
25
26 # Randomly pick some pixels in the
27 # image for coloring them white
28 # Pick a random number between 300 and 10000
29 number_of_pixels = random.randint(300, 10000)
30 for i in range(number_of_pixels):
31
32 # Pick a random y coordinate
33 y_coord=random.randint(0, row - 1)
34
35 # Pick a random x coordinate
image
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn = cv:
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::NONE;
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

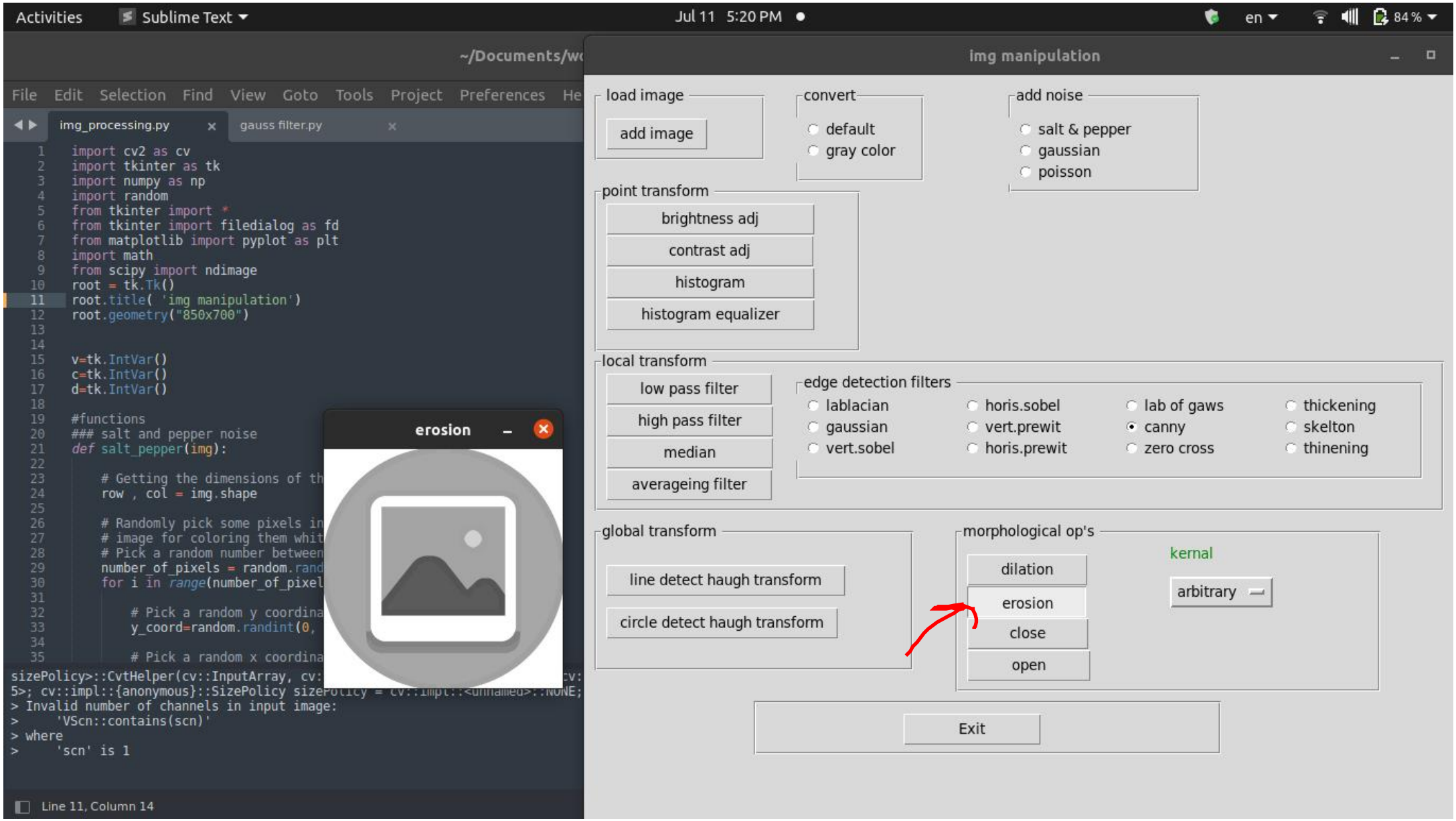

Activities
Sublime Text
Jul 11 5:19 PM
en
84 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedialog as fd
7 from matplotlib import pyplot as plt
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulation')
12 root.geometry("850x700")
13
14
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
18
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
22
23 # Getting the dimensions of
24 row , col = img.shape
25
26 # Randomly pick some pixels
27 # image for coloring them white
28 # Pick a random number between 300 and 10000
29 number_of_pixels = random.randint(300, 10000)
30 for i in range(number_of_pixels):
31
32 # Pick a random y coordinate
33 y_coord=random.randint(0, row - 1)
34
35 # Pick a random x coordinate
PreWittx
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn = cv:
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::NONE;
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

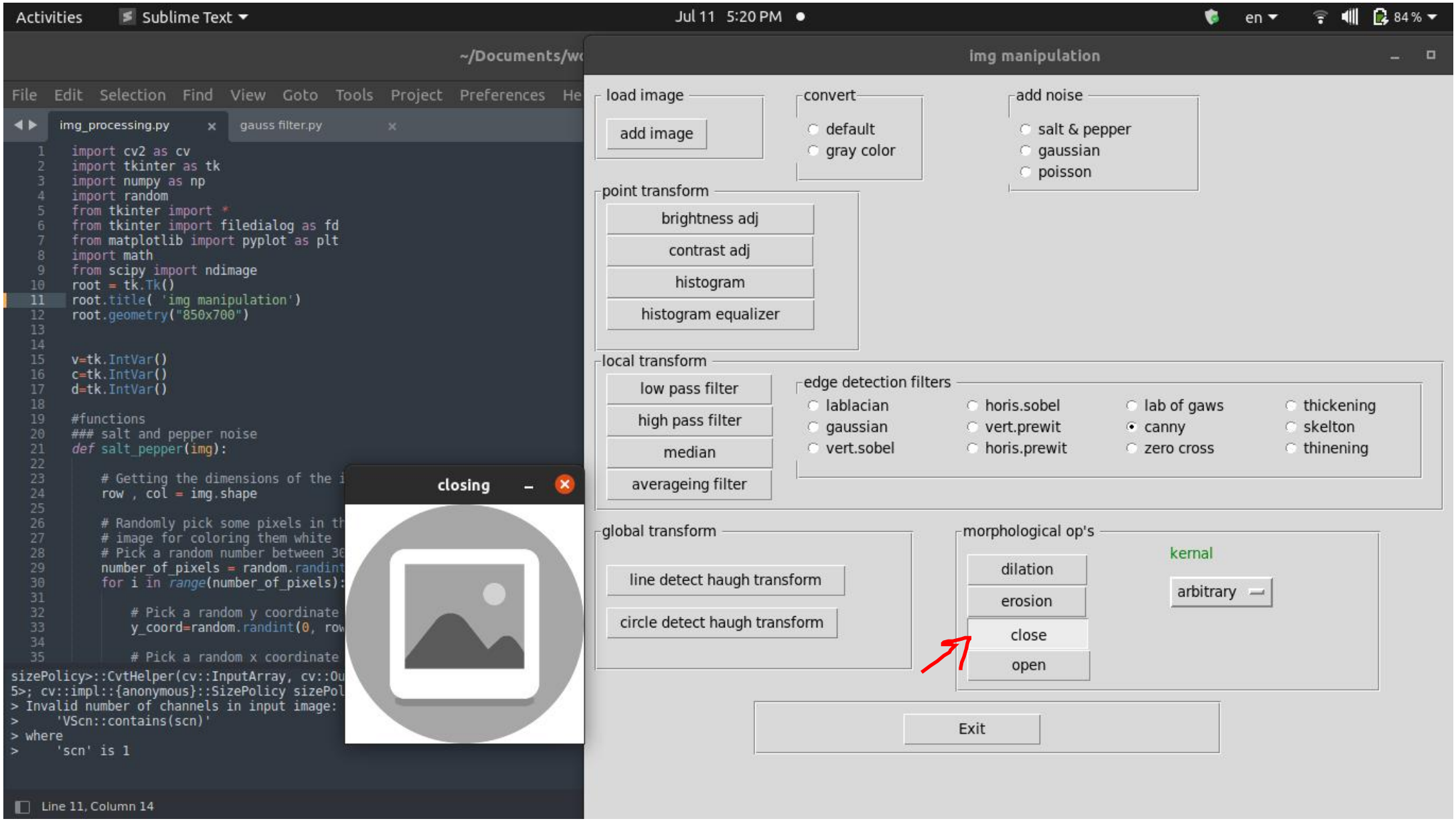

Activities
Sublime Text
Jul 11 5:19 PM
en
84 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedial
7 from matplotlib import pyplo
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulatio
12 root.geometry("850x700")
13
14
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
18
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
22
23 # Getting the dimensions
24 row , col = img.shape
25
26 # Randomly pick some pixels in the
27 # image for coloring them white
28 # Pick a random number between 300 and 10000
29 number_of_pixels = random.randint(300, 10000)
30 for i in range(number_of_pixels):
31
32 # Pick a random y coordinate
33 y_coord=random.randint(0, row - 1)
34
35 # Pick a random x coordinate
canny
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn = cv:
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::NONE;
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

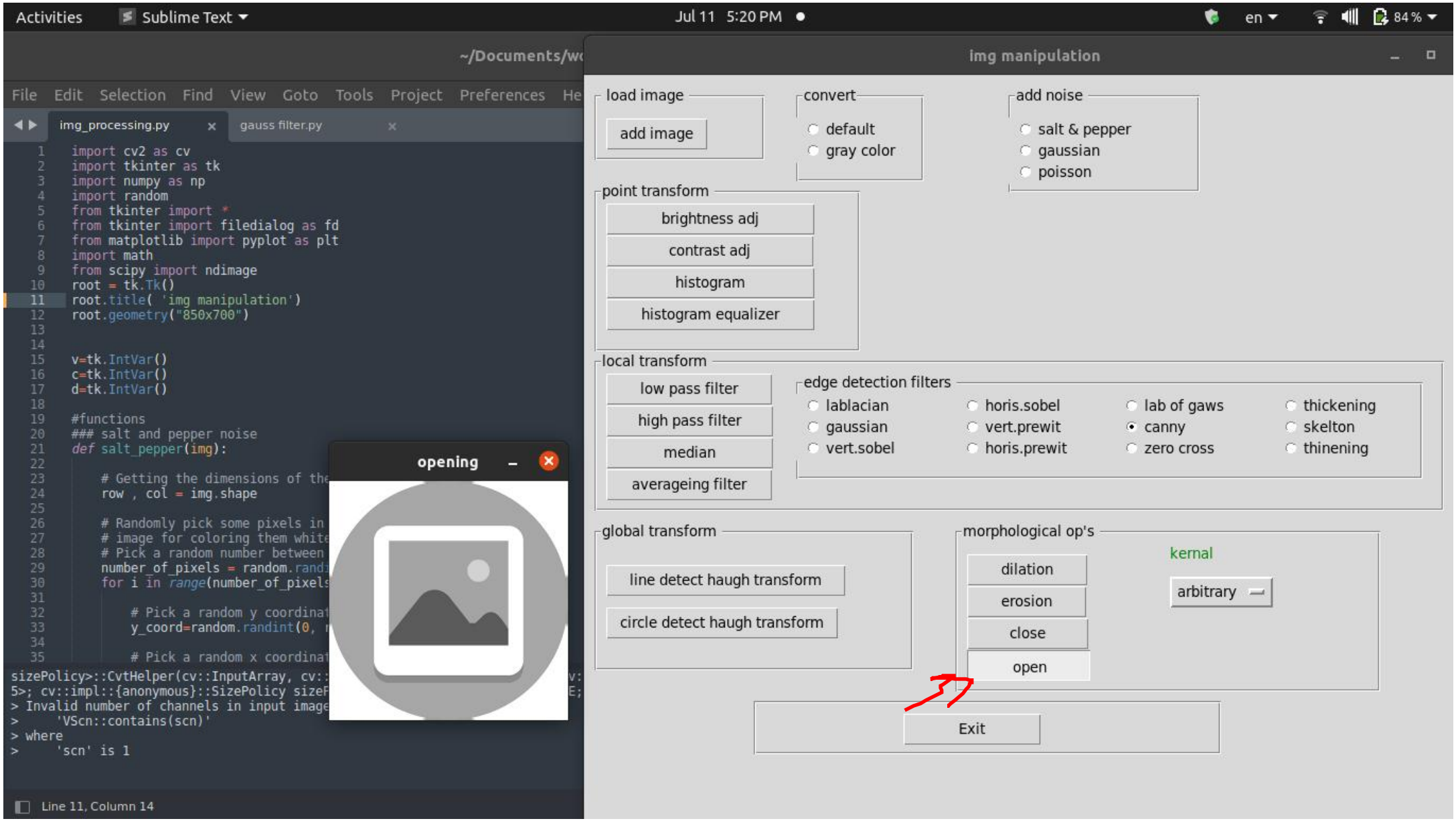

Activities
Sublime Text
Jul 11 5:19 PM
en
84 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
import cv2 as cv
import tkinter as tk
import numpy as np
import random
from tkinter import *
from tkinter import filedialog as fd
from matplotlib import pyplot as plt
import math
from scipy import ndimage
root = tk.Tk()
root.title('img manipulation')
root.geometry("850x700")
v=tk.IntVar()
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions o
row , col = img.shape
# Randomly pick some pixel
# image for coloring them
# Pick a random number bet
number_of_pixels = random.
for i in range(number_of_p
# Pick a random y coor
y_coord=random.randint
# Pick a random x coordinate
line detected
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn = cv:
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::NONE;
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit

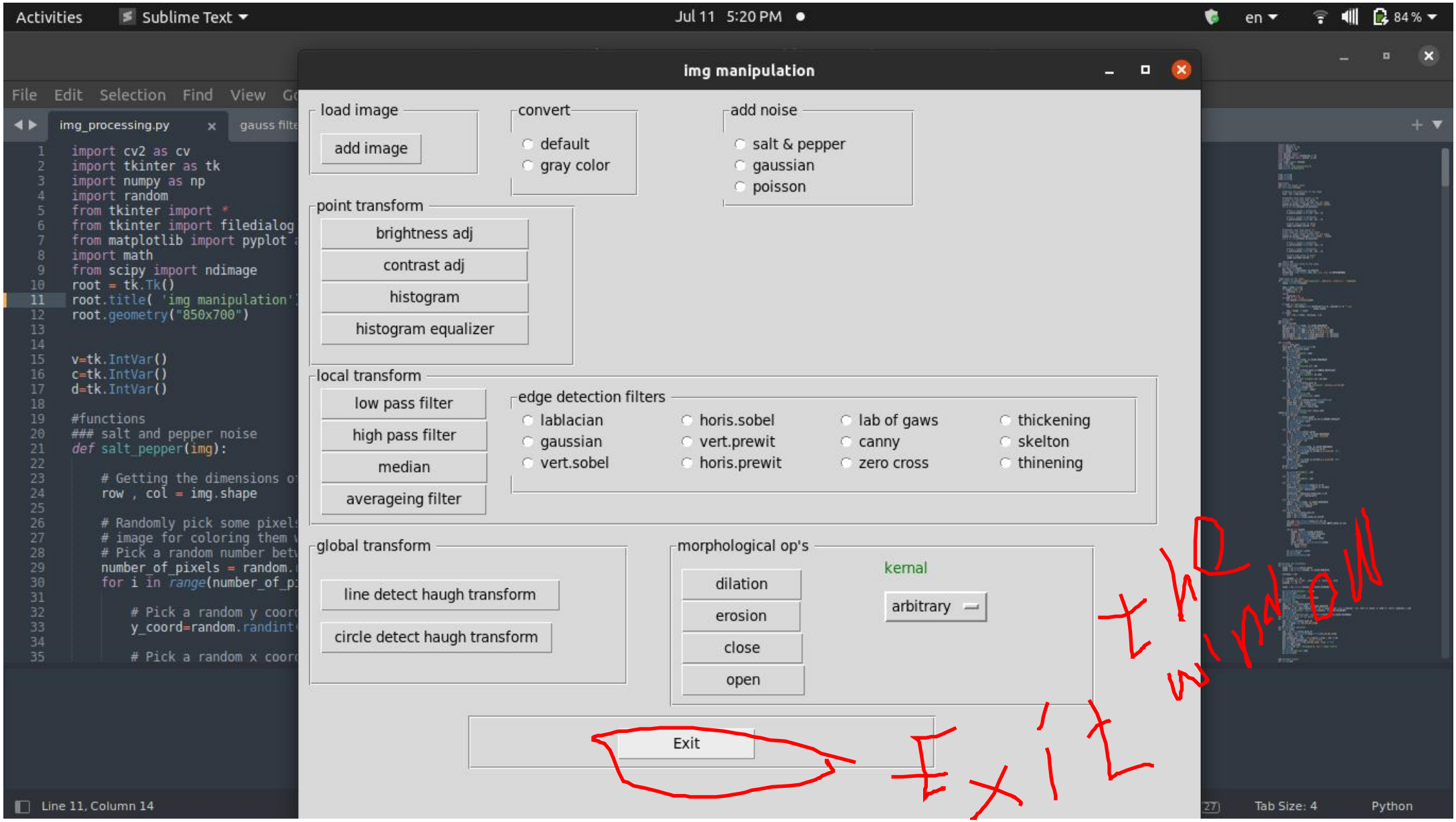

Activities
Sublime Text
Jul 11 5:20 PM
en
84 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
import cv2 as cv
import tkinter as tk
import numpy as np
import random
from tkinter import *
from tkinter import filedialog as fd
from matplotlib import pyplot as plt
import math
from scipy import ndimage
root = tk.Tk()
root.title('img manipulation')
root.geometry("850x700")
v=tk.IntVar()
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of t
row , col = img.shape
# Randomly pick some pixels i
# image for coloring them whi
# Pick a random number betwee
number_of_pixels = random.ran
for i in range(number_of_pixe
# Pick a random y coordin
y_coord=random.randint(0,
# Pick a random x coordin
detected circles
sizePolicy>::CvtHelper(cv::InputArray, cv
5>; cv::impl::{anonymous}::SizePolicy siz
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit


Activities
Sublime Text
Jul 11 5:20 PM
en
84 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedialog as fd
7 from matplotlib import pyplot as plt
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulation')
12 root.geometry("850x700")
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
23 # Getting the dimensions of t
24 row , col = img.shape
26 # Randomly pick some pixels i
27 # image for coloring them whi
28 # Pick a random number betwee
29 number_of_pixels = random.ra
30 for i in range(number_of_pixe
32 # Pick a random y coordinate
33 y_coord=random.randint(0, row - 1)
35 # Pick a random x coordinate
dilation
sizePolicy>::CvtHelper(cv::InputArray, cv::OutputArray, int) [with VScn = cv:
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::NONE;
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit


Activities
Sublime Text
Jul 11 5:20 PM
en
84 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedialog as fd
7 from matplotlib import pyplot as plt
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulation')
12 root.geometry("850x700")
13
14
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
18
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
22
23 # Getting the dimensions of th
24 row , col = img.shape
25
26 # Randomly pick some pixels in
27 # image for coloring them whit
28 # Pick a random number between
29 number_of_pixels = random.rand
30 for i in range(number_of_pixel
31
32 # Pick a random y coordina
33 y_coord=random.randint(0,
34
35 # Pick a random x coordina
sizePolicy>::CvtHelper(cv::InputArray, cv:
5>; cv::impl::{anonymous}::SizePolicy sizePolicy = cv::impl::<unnamed>::NONE;
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
erosion
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit


Activities
Sublime Text
Jul 11 5:20 PM
en
84 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
import cv2 as cv
import tkinter as tk
import numpy as np
import random
from tkinter import *
from tkinter import filedialog as fd
from matplotlib import pyplot as plt
import math
from scipy import ndimage
root = tk.Tk()
root.title('img manipulation')
root.geometry("850x700")
v=tk.IntVar()
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of the i
row , col = img.shape
# Randomly pick some pixels in th
# image for coloring them white
# Pick a random number between 30
number_of_pixels = random.randint
for i in range(number_of_pixels):
# Pick a random y coordinate
y_coord=random.randint(0, row
# Pick a random x coordinate
sizePolicy>::CvtHelper(cv::InputArray, cv::Ou
5>; cv::impl::{anonymous}::SizePolicy sizePol
> Invalid number of channels in input image:
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
closing
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit


Activities
Sublime Text
Jul 11 5:20 PM
en
84 %
~/Documents/w
File Edit Selection Find View Goto Tools Project Preferences He
img_processing.py
gauss filter.py
import cv2 as cv
import tkinter as tk
import numpy as np
import random
from tkinter import *
from tkinter import filedialog as fd
from matplotlib import pyplot as plt
import math
from scipy import ndimage
root = tk.Tk()
root.title('img manipulation')
root.geometry("850x700")
v=tk.IntVar()
c=tk.IntVar()
d=tk.IntVar()
#functions
salt and pepper noise
def salt_pepper(img):
# Getting the dimensions of the
row , col = img.shape
# Randomly pick some pixels in
# image for coloring them white
# Pick a random number between
number_of_pixels = random.randi
for i in range(number_of_pixels
# Pick a random y coordinat
y_coord=random.randint(0,
# Pick a random x coordinat
opening
sizePolicy>::CvtHelper(cv::InputArray, cv::
5>; cv::impl::{anonymous}::SizePolicy sizeP
> Invalid number of channels in input image
> 'VScn::contains(scn)'
> where
> 'scn' is 1
Line 11, Column 14
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit


Activities
Sublime Text
Jul 11 5:20 PM
en
84 %
File Edit Selection Find View
img_processing.py
gauss filte
1 import cv2 as cv
2 import tkinter as tk
3 import numpy as np
4 import random
5 from tkinter import *
6 from tkinter import filedialog
7 from matplotlib import pyplot
8 import math
9 from scipy import ndimage
10 root = tk.Tk()
11 root.title('img manipulation')
12 root.geometry("850x700")
13
14
15 v=tk.IntVar()
16 c=tk.IntVar()
17 d=tk.IntVar()
18
19 #functions
20 ### salt and pepper noise
21 def salt_pepper(img):
22
23 # Getting the dimensions o
24 row , col = img.shape
25
26 # Randomly pick some pixel
27 # image for coloring them
28 # Pick a random number bet
29 number_of_pixels = random.
30 for i in range(number_of_p
31
32 # Pick a random y coor
33 y_coord=random.randint
34
35 # Pick a random x coor
Line 11, Column 14
Tab Size: 4
Python
img manipulation
load image
add image
convert
default
gray color
add noise
salt & pepper
gaussian
poisson
point transform
brightness adj
contrast adj
histogram
histogram equalizer
local transform
low pass filter
high pass filter
median
averageing filter
edge detection filters
lablacian
gaussian
vert.sobel
horis.sobel
vert.prewit
horis.prewit
lab of gaws
canny
zero cross
thickening
skelton
thinening
global transform
line detect haugh transform
circle detect haugh transform
morphological op's
dilation
erosion
close
open
kernal
arbitrary
Exit
to window
Exit