

# ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ

## Εργασία 2

ΠΑΝΤΕΛΕΗΜΩΝ ΠΡΩΙΟΣ

ice18390023

9ο Εξάμηνο

ice18390023@uniwa.gr

Πέμπτη 16:00-19:00

Ομάδα χρηστών 18



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
UNIVERSITY OF WEST ATTICA

Υπεύθυνοι καθηγητές

ΕΥΦΡΟΣΥΝΗ ΔΟΥΤΣΗ

Τμήμα Μηχανικών και Πληροφορικής Υπολογιστών  
24 Δεκεμβρίου 2022

## Περιεχόμενα

1	Εισαγωγή	1
2	Κατασκευή της συνάρτησης convImage	2
3	Χρήση της conv2	3
4	Χρήση της imfilter	4
5	Χρήση του θεωρήματος Fourier	4
6	Συμπέρασμα	5
7	Appendix A	6

## Κώδικες

2.1	convolutionV2 function . . . . .	3
5.1	Fourier . . . . .	4

## Κατάλογος σχημάτων

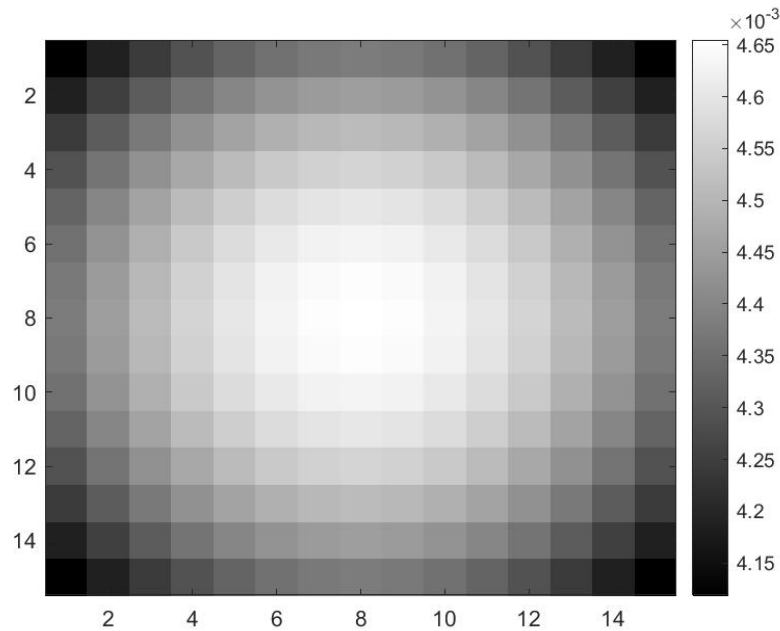
1.1	Original image . . . . .	1
1.2	Kernel image . . . . .	2
7.1	convolutionV2 . . . . .	6
7.2	conv2 sane . . . . .	6
7.3	conv2 valid . . . . .	7
7.4	imfilter . . . . .	7
7.5	fourier . . . . .	8

## 1 Εισαγωγή

Σκοπός της εργασίας, η σύγκριση της συνάρτησης συνέλιξης που κατασκευάσαμε, των δυο build-in συναρτήσεων `conv2` και `imfilter`, όπως και η εξακρίβωση του θεωρήματος Fourier με την χρήση της ίδιας εικόνας (εικ. 1.1) και του ίδιου πυρήνα 1.2.



Σχήμα 1.1: Original image



Σχήμα 1.2: Kernel image

## 2 Κατασκευή της συνάρτησης convImage

Με τις ακόλουθες εντολές, κατασκευάζουμε ένα Γκαουσιανό φίλτρο μεγέθους 15x15 με  $\sigma = 20$ , φορτώνουμε την εικόνα (1.1) και κάνουμε zero padding.

```
%% create gaussian
gaussianFilter = fspecial('gaussian', [15 15], 20);

%% read image
original_img = imread('lena_gray_512.tif');

%% zero padding
padding_len = floor(length(gaussianFilter)/2);
pad_img = padarray(original_img, [padding_len padding_len], 0);

clear padding_len
```

Έπειτα, με την χρήση της συνάρτησης convolutionV2 (2), δίνουμε ως πρώτο το όρισμα της εικόνας με το zero padding και ως δεύτερο όρισμα τον πυρήνα. Η συνάρτηση αυτή θα μας επιστρέψει την συνέλιξη της εικόνας με τον πυρήνα.

```
1 function convImage = convolutionV2(I,K)
2 % h = f * g
3 % h = I * K
4
5 if mod(size(K,1),2)==0 || mod(size(K,2),2) == 0
6     fprintf('Concolution needs to have odd numbers')
7     exit 1;
```

```

8 end
9
10 I = double(I);
11
12 % conv needs to rot kernel 180 deg
13 K = rot90(K,2);
14
15 kernel_rp = floor(size(K,1)/2); %row padding
16 kernel_cp = floor(size(K,2)/2); %column padding
17 k_center = [ceil(size(K,1)/2), ceil(size(K,2)/2)];
18
19 % pad zeros based on kernel
20 % I = double(padarray(I,[kernel_rp kernel_cp],0));
21
22 img_rows = size(I,1);
23 img_cols = size(I,2);
24
25 convImage = zeros(img_rows-2*kernel_rp,img_cols-2*kernel_cp);
26
27 for i=kernel_rp+1:img_rows-kernel_rp
28
29     for j=kernel_cp+1:img_cols-kernel_cp
30
31         convImage(i-kernel_rp,j-kernel_cp) = ....
32             sum(sum(K.*I(i-kernel_rp:i+kernel_rp,j-kernel_cp:j+kernel_cp)));
33     end
34
35 end
36 end

```

Με τις ακόλουθες εντολές, υπολογίζουμε το MSE και PSNR των δύο εικόνων

```

convoluted_img = convolutionV2(pad_img, gaussianFilter);
%% MSE PSNR

err_mse = immse(double(original_img), convoluted_img);
dB_psnr = 10.*log10(255^2/err_mse);
fprintf(' MSE=%0.4f PSNR=%0.4f (convolutionV2)\n',err_mse, dB_psnr);
% MSE=348.5180 PSNR=22.7086 (convolutionV2)

```

Κώδικας 2.1: convolutionV2 function

### 3 Χρήση της conv2

Το ακόλουθο script κάνει συνέλιξη με την χρήση της conv2. Ο πρώτος τρόπος της conv2 είναι στην original image και με την παράμετρο `same`, έτσι ώστε να επιστρέψει το ίδιο μέγεθος. Ενώ, ο δεύτερος τρόπος είναι στην zero padded εικόνα, με την παράμετρο `valid`, έτσι ώστε να επιστρέψει την εικόνα χωρίς τα zero padded edges.

```

convoluted_img_conv2_same = conv2(original_img,gaussianFilter,'same');
convoluted_img_conv2_valid = conv2(pad_img,gaussianFilter,'valid');

%% MSE PSNR for conv2
err_mse = immse(double(original_img), convoluted_img_conv2_same);
dB_psnr = 10.*log10(255^2/err_mse);
fprintf(' MSE=%0.4f PSNR=%0.4f (conv2_same)\n',err_mse, dB_psnr);

```

```
% MSE=348.5180 PSNR=22.7086 (conv2_same)

err_mse = immse(double(original_img), convoluted_img_conv2_valid);
dB_psnr = 10.*log10(255^2/err_mse);
fprintf(' MSE=%0.4f PSNR=%0.4f (conv2_valid)\n',err_mse, dB_psnr);
% MSE=348.5180 PSNR=22.7086 (conv2_valid)
```

## 4 Χρήση της imfilter

Το επόμενο script κάνει συνέλιξη με την χρήση της imfilter.

```
%% build-in imfilter
convoluted_imfilter = imfilter(original_img, gaussianFilter, 'conv', 'same',0);
convoluted_imfilter = double(convoluted_imfilter);

%% MSE PSNR for imfilter
err_mse = immse(double(original_img), convoluted_imfilter);
dB_psnr = 10.*log10(255^2/err_mse);
fprintf(' MSE=%0.4f PSNR=%0.4f (imfilter)\n',err_mse, dB_psnr);
% MSE=348.6201 PSNR=22.7073 (imfilter)
```

## 5 Χρήση του θεωρήματος Fourier

Με την χρήση του θεωρήματος Fourier, κάνουμε συνέλιξη στο πεδίο των συχνοτήτων πολλαπλασιάζοντας τους πίνακες μεταξύ τους, αφού τους μεταφέρουμε στο πεδίο των συχνοτήτων με την χρήση της συνάρτησης fft2 και αφού τα πολλαπλασιάσουμε, μετατρέπουμε το αποτέλεσμα στο πεδίο του χρόνου με την χρήση της ifft2.

```
%% Fourier

[rI,cI] = size(original_img);
[rK,cK] = size(gaussianFilter);

% output size
row = rI + rK -1;
col = cI + cK -1;

% pad, conv/mul and back to time domain
FT_time = ifft2(fft2(original_img,row,col).*fft2(gaussianFilter,row,col));

% change size output to size original_img
pad_row = ceil((rK-1)./2);
pad_col = ceil((cK-1)./2);

FT_conv_time = FT_time(pad_row+1:rI+pad_row,pad_col+1:cI+pad_col);

err_mse = immse(double(original_img), FT_conv_time);
dB_psnr = 10.*log10(255^2/err_mse);
fprintf(' MSE=%0.4f PSNR=%0.4f (fourier)\n',err_mse, dB_psnr);
```

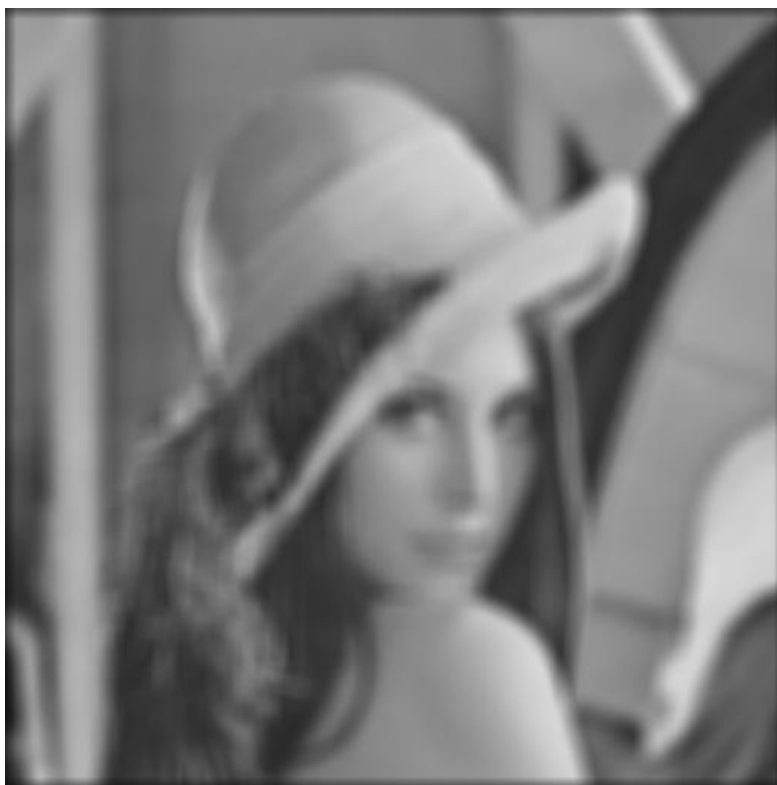
```
% MSE=348.5180 PSNR=22.7086 (fourier)
```

Κώδικας 5.1: Fourier

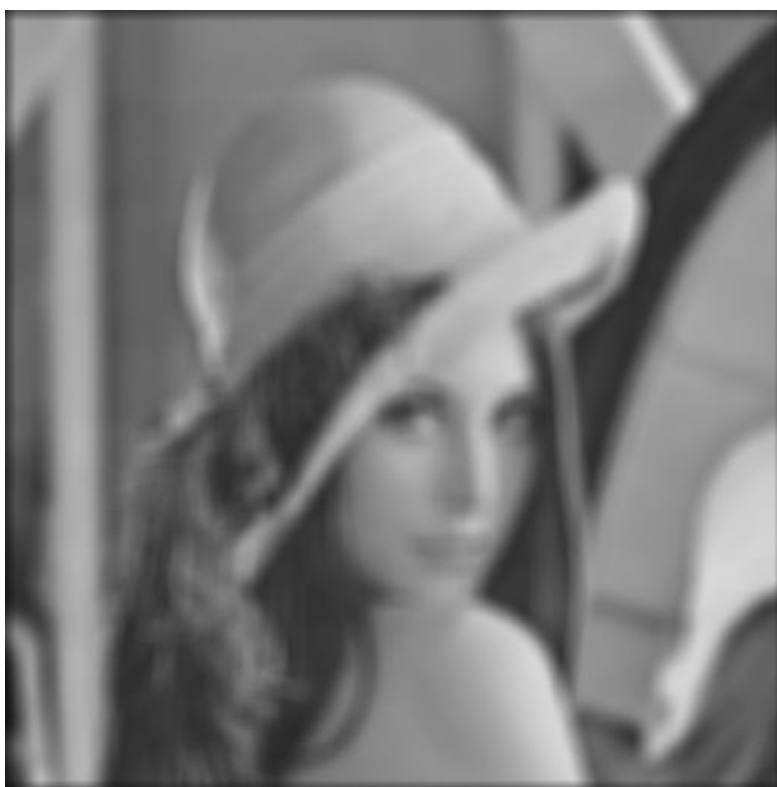
## 6 Συμπέρασμα

Από τα αποτελέσματα των μετρικών MSE και PSNR, είναι φανερό πως με όλες της μεθόδους επιτυγχάνουμε το ίδιο αποτέλεσμα, με μια μηδαμινή διαφορά με την χρήση της `imfilter`.

## 7 Appendix A

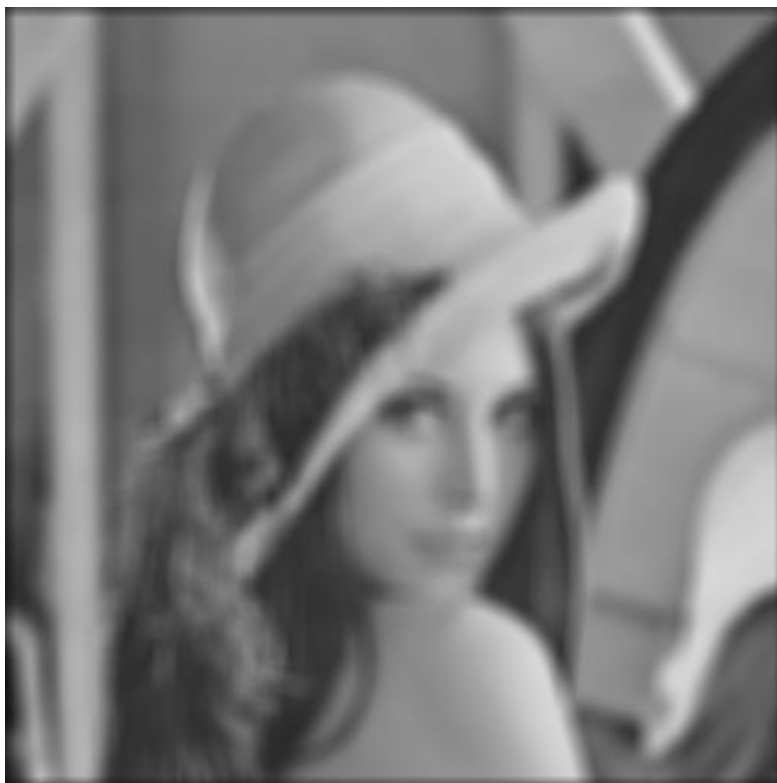


Σχήμα 7.1: convolutionV2

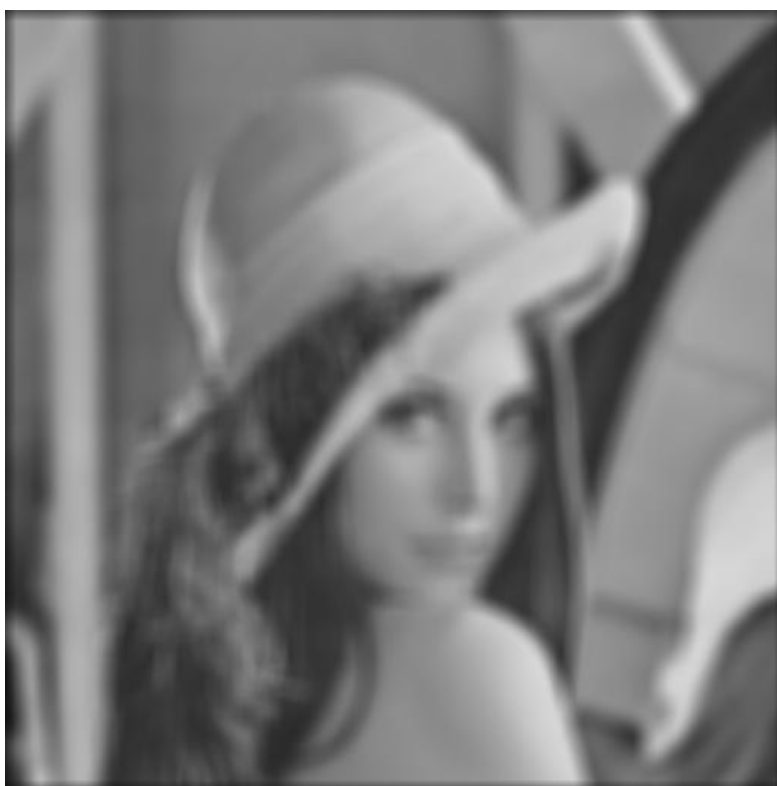


Σχήμα 7.2: conv2 sane

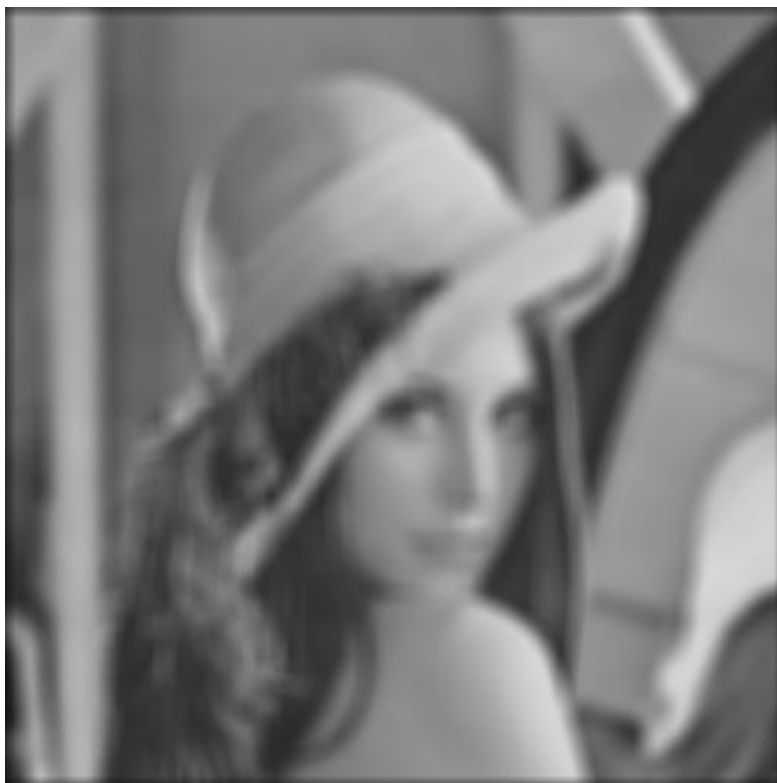




Σχήμα 7.3: conv2 valid



Σχήμα 7.4: imfilter



Σχήμα 7.5: fourier