Exam

02506
Spring 2022

General information

- The exam consists of 20 questions. All questions have equal weight: the correct answer gives 1 point, incorrect or missing answer gives 0 points. You only need to submit the answers to the questions. You should not upload any notes or calculations.
- There is one correct answer for each question. Some of the numeric results have been rounded, and might slightly deviate from your result. This should not prevent you from being able to pick the correct answer.
- Each page contains one question. If there are illustrations and images, those refer to the question on that page.
- The notation in the questions is the same as in the course note.
- Data and images are provided for some questions. The question text states which data or image should be used.
- To load arrays saved as text files you can use `numpy.loadtxt` in python and `dlmread` in matlab.

Relevant links

- Course note
  `http://www2.imm.dtu.dk/courses/02506/LECTURE_NOTES_2022.pdf`
- Resources are available for a subset of questions in form of data, images, and code. Filenames typeset in `typewriter font` indicate that you can find files in the materials folder
  `http://www2.imm.dtu.dk/courses/02506/EXAM_MATERIAL.zip`

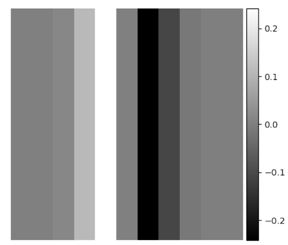
Question 1

Gaussian derivative 2D Gaussian kernels can be used for feature detection in scale-space. Below we show some kernels related to the 2D first order Gaussian derivative with $\sigma = 1$. The kernel values are shown in the spatial range $\pm 2\sigma$ and the image next to the kernel values is shown in the spatial range $\pm 5\sigma$.

Which kernel best represents the 2D first order Gaussian derivative used for scale-space computation?

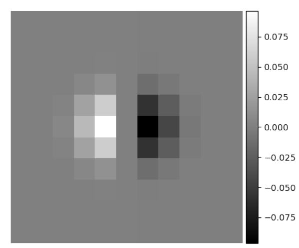
(a)

0.11	0.24	0.00	−0.24	−0.11
0.11	0.24	0.00	−0.24	−0.11
0.11	0.24	0.00	−0.24	−0.11
0.11	0.24	0.00	−0.24	−0.11
0.11	0.24	0.00	−0.24	−0.11



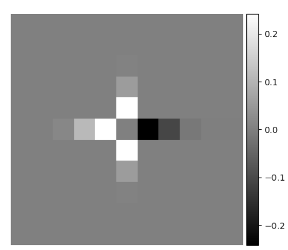
(b)

0.01	0.01	0.00	−0.01	−0.01
0.03	0.06	0.00	−0.06	−0.03
0.04	0.10	0.00	−0.10	−0.04
0.03	0.06	0.00	−0.06	−0.03
0.01	0.01	0.00	−0.01	−0.01



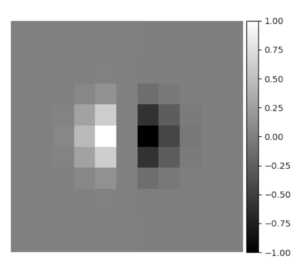
(c)

0.00	0.00	0.05	0.00	0.00
0.00	0.00	0.24	0.00	0.00
0.11	0.24	0.00	−0.24	−0.11
0.00	0.00	0.24	0.00	0.00
0.00	0.00	0.05	0.00	0.00



(d)

0.06	0.14	0.00	−0.14	−0.06
0.27	0.61	0.00	−0.61	−0.27
0.45	1.00	0.00	−1.00	−0.45
0.27	0.61	0.00	−0.61	−0.27
0.06	0.14	0.00	−0.14	−0.06



Question 2

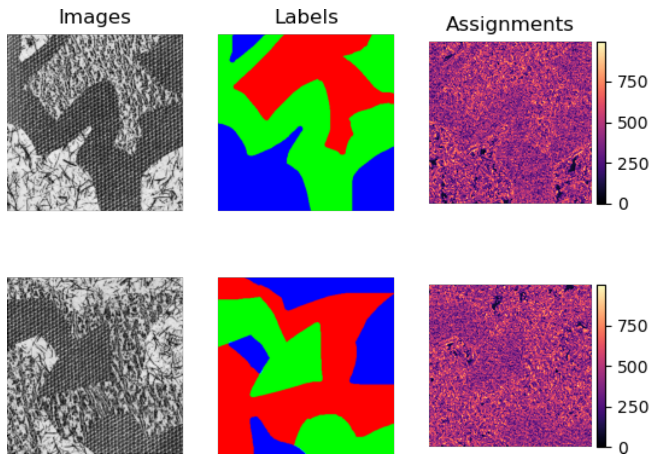
Scale-space blob detection The image `blobs.png` contains a bright and a dark blob. The task is to detect the bright blob using Laplacian scale-space. We have pre-computed a linear Laplacian scale-space and the result is available in the data folder as the 3D image `blobs_Laplace.tif` with `[scale, row, column]` in the three dimensions. You can load the `blobs_Laplace.tif` file using the function `skimage.io.imread` in Python or `tiffreadVolume` in Matlab. The Laplacian scale-space is computed using a kernel with scale $t = 16$ ($\sigma = 4$), and the first layer of the image is at scale $t = 16$, the second layer at $t = 32$, etc.

At which scale and position (coordinates are 0-indexed) is the bright blob detected?

(a) Scale: 8, position (row, col): $(132, 57)$
(b) Scale: 11, position (row, col): $(122, 80)$
(c) Scale: 32, position (row, col): $(132, 57)$
(d) Scale: 44, position (row, col): $(80, 122)$
(e) Scale: 192, position (row, col): $(80, 122)$
(f) Scale: 144, position (row, col): $(132, 57)$

Question 3

Feature-based segmentation The task is feature-based image segmentation. We train a model by clustering features as described in the lecture notes Chapter 3, but here we use two training images. After assigning all pixels from the two training images to one of the feature classes C_i and one of the labels l_1, l_2 or l_3 (red, green, blue), as illustrated below, we obtain the distribution given in the two tables.



Distribution of clusters and labels for pixels from image 1:

	...	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	...
l_1	...	46	49	24	21	4	77	55	21	118	39	63	...
l_2	...	31	15	0	0	350	0	0	187	1	1	12	...
l_3	...	4	16	45	69	1	53	7	2	8	34	32	...

Distribution of clusters and labels for pixels from image 2:

	...	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	...
l_1	...	30	41	26	36	45	61	19	31	43	35	49	...
l_2	...	27	43	21	46	147	56	31	82	51	42	61	...
l_3	...	26	38	13	32	36	60	32	30	55	33	50	...

What is the label probability of the feature cluster C_{16} for all three labels, expressed as a vector $[p_{16}(l_1),\ p_{16}(l_2),\ p_{16}(l_3)]$?

(a) $[0,\ 0,\ 1]$
(b) $[0,\ 1,\ 0]$
(c) $[0.08,\ 0.85,\ 0.06]$
(d) $[0.1,\ 0.9,\ 0]$
(e) $[0.45,\ 0.18,\ 0.37]$
(f) $[49,\ 497,\ 37]$

Question 4

Segmentation with image patches We wish to segment an image using image patches as described in the lecture notes, Section 3.1.1. We have a gray-scale image of size 512×512, we use patches of size 7×7, and we segment to four labels.

What is the size of the resulting probability image?

(a) $256 \times 256 \times 3$
(b) $505 \times 505 \times 1$
(c) $506 \times 506 \times 4$
(d) $510 \times 510 \times 4$
(e) $512 \times 512 \times 1$
(f) $512 \times 512 \times 4$

Question 5

Matching SIFT features SIFT features have been computed for the two images `kidney_1.png` and `kidney_2.png` shown below. The descriptors are stored in the files `SIFT_1_descriptors.txt` and `SIFT_2_descriptors.txt`, available in the data folder. The spatial positions where the SIFT features are detected are in the files `SIFT_1_coordinates.txt` and `SIFT_2_coordinates.txt`, which are also available. In all file names the numbers 1 and 2 refer to the image number. We consider the first feature from Image 1, i.e. the first row of the file `SIFT_1_descriptors.txt`.

Image 1

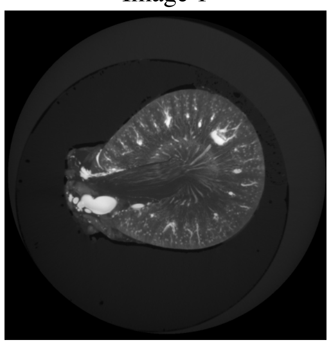
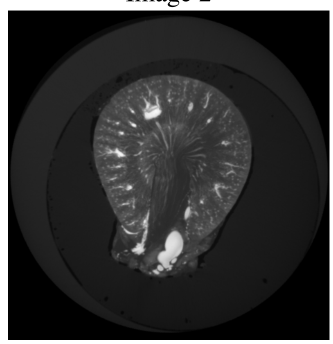


Image 2



What is the spatial (Euclidean) distance between the first SIFT feature from Image 1 and its closest match in Image 2?

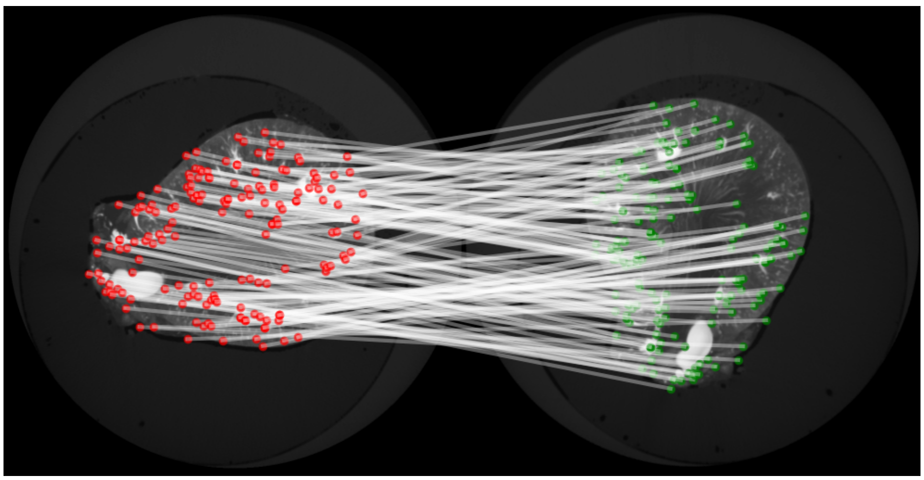
(a) -224.1
(b) 183.6
(c) 201.6
(d) 224.1
(e) 304.8
(f) 259.5
(g) 33708.3
(h) 40662.4

Question 6

Feature-based image registration By matching the SIFT features, as shown in the figure, we have obtained correspondence between feature points. Based on this we want to register the images, as described in the lecture note. We have computed the following covariance matrix

$$C = \begin{bmatrix} 191955 & -937044 \\ 552183 & 379358 \end{bmatrix},$$

that we use for computing the transformation.



What is the rotation angle in degrees?

(a) -96.0
(b) -1.20
(c) 1.20
(d) 21.0
(e) 53.0
(f) 69.0
(g) 72.0
(h) 84.0
(i) 169.0

Question 7

Gaussian convolution Which statement is correct?

(a) To ensure high precision when iteratively applying Gaussian kernels in a Gaussian scale-space, the kernels must be 2D.
(b) To ensure high precision when iteratively applying Gaussian kernels for a Gaussian scale-space, the kernels must have a large spatial support, e.g. $\pm 4\sigma$ to $\pm 5\sigma$.
(c) There is no loss in precision when iteratively applying Gaussian kernels for a Gaussian scale-space, and the kernels must have a small spatial support, e.g. $\pm 1.5\sigma$ to $\pm 2\sigma$ to be computationally efficient.
(d) The loss in precision when iteratively convolving with Gaussian kernels makes the approach unsuited for Gaussian scale-space.

Question 8

MRF prior energy Consider an MRF segmentation with three labels: 1, 2, and 3. As in the lecture note, we use first-order neighborhood. We define 2-clique potentials as

$$V_2(f_i, f_{i'}) = 10|f_i - f_{i'}|$$

which is different than what we used in the lecture note. Consider the configuration `configuration.txt` given in the illustration.

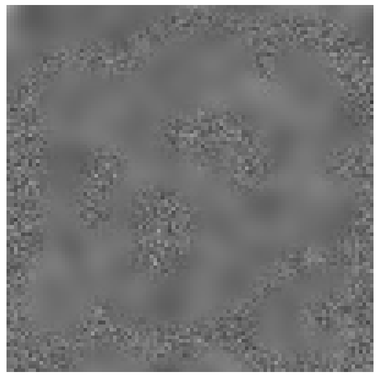
1	1	1	1	2	2	2
1	2	2	1	2	3	2
2	3	2	1	2	3	2
2	2	2	1	1	3	2
1	2	1	1	1	1	1

What is the prior energy of this configuration?

(a) 10
(b) 20
(c) 30
(d) 100
(e) 140
(f) 170
(g) 180
(h) 200
(i) 280
(j) 300
(k) 320
(l) 350

Question 9

Likelihood energy Consider segmentation of the image I, available in the data folder as `blended.png`. Image values should be normalized to be in the range $[0, 1]$ by dividing the pixel values by 255 after loading.



The image is to be segmented in two segments, one characterized by the smooth appearance, and another one textured. Therefore, for each pixel i we consider a label $f_i \in \{\mathrm{textured}, \mathrm{smooth}\}$. We characterize the segments not by their mean value, but by their standard deviation, estimated to be $\sigma(\mathrm{textured}) = 0.05$ and $\sigma(\mathrm{smooth}) = 0.01$. We define one-clique potential as

$$V_1(f_i) = (\sigma(f_i) - s_i)^2 \,,$$

where s_i is a standard deviation of a 5-by-5 window around a pixel i. Values s_i can be computed as an image

$$S = \sqrt{K * (I^2) - (K * I)^2}$$

where $K*$ denotes a convolution with a 5-by-5 averaging window. The function for computing S from I is available in the data folder as `window_std.py` and `window_std.m`.

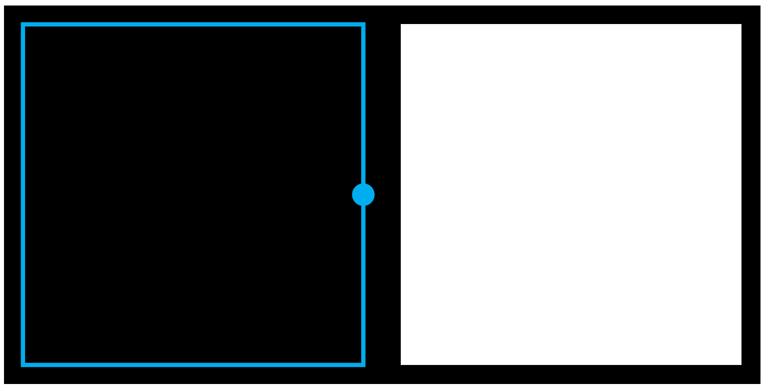
What is the likelihood energy of the max-likelihood segmentation?

(a) -15.015
(b) 0.135
(c) 0.135
(d) 0.410
(e) 0.590
(f) 0.805
(g) 0.914
(h) 14.04
(i) 14.95
(j) 1161
(k) 4096
(l) 5904

Question 10

External force We consider image segmentation using a deformable curve with the external force as in the lecture notes, Section 6.1.

The image is of size 100-by-200 pixels and depicts a white 90-by-90 pixels square on the black background. Pixel values are 0 for black and 1 for white. A curve delineating another 90-by-90 square is placed as shown in the illustration.

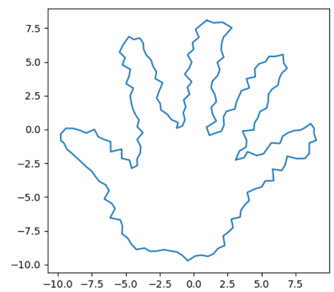


Which of the following best represents the scalar component of the external force acting on the highlighted curve point?

(a) -1
(b) -0.54
(c) 0.16
(d) 0.46
(e) 1
(f) 2

Question 11

Curve smoothing Consider a noisy curve in the illustration, available in the data folder as `hand_noisy.txt`.



We want to smooth the curve using an approach covered in lecture notes Section 6.2 and in the Exercise 1.1.3. We know that high values of α shrink the curve more than high values of β. We want to test this claim experimentally.

We produce two smoothed curves where for $\mathbf{X}_{\text{alpha}}$ we used parameters $\alpha = 100$ and $\beta = 0$, while for $\mathbf{X}_{\text{beta}}$ we used $\alpha = 0$ and $\beta = 100$. Now we compute the length of the two curves and their difference $d = \ell(\mathbf{X}_{\text{alpha}}) - \ell(\mathbf{X}_{\text{beta}})$.

What is the value of d?

(a) -38.4
(b) -24.5
(c) -2.7
(d) 0.46
(e) 16.4
(f) 30.6
(g) 68.8
(h) 69.0

Question 12

Two surface cost We consider two-surface detection with on-surface cost, as in the lecture notes Section 7.1. The image I, available as `layers.txt`, has pixel intensities as given in the illustration. The cost images are defined as

$$c_1 = (I - 90)^2 \text{ and } c_2 = (I - 30)^2 \,.$$

Two surfaces, s_1 (blue) and s_2 (red), are highlighted in the illustration.

62	58	58	67	62	68	62	58	66	58	
63	25	56	66	21	21	32	21	39	53	
24	64	39	32	54	52	50	52	66	30	s_2
53	57	68	50	67	59	67	55	69	68	
53	54	54	58	69	63	87	67	68	67	
64	93	65	82	92	86	58	88	85	82	s_1
80	52	91	51	67	64	59	66	65	59	
61	67	69	60	56	61	69	52	62	60	

What is the cost associated with a solution yielding surfaces s_1 and s_2?

(a) 14.12
(b) 37
(c) 770
(d) 1030
(e) 1150
(f) 1412
(g) 70610
(h) 666692

Question 13

Surface distance We consider the optimal surface detection with on-surface cost as in the lecture notes Section 7.1. The cost image c_{on} has values as given in the illustration, and also available in the data folder as `cost.txt`.

40	43	14	30	90	12	80	70	83	95
43	25	56	66	21	21	32	21	39	53
26	13	68	50	67	59	67	55	69	68
78	11	65	52	32	46	58	38	45	22
21	24	91	51	67	64	59	66	65	59
90	13	69	60	56	61	69	52	62	60

Denote by s_0 the solution obtained using constraint $\Delta_x = 0$, and by s_5 the solution obtained using constraint $\Delta_x = 5$.

What is the mean absolute distance

$$d = \frac{1}{X} \sum_{x=1}^{X} |s_0(x) - s_5(x)|$$

between s_0 and s_5?

(a) 0
(b) 0.1
(c) 0.25
(d) 1
(e) 1.25
(f) 2
(g) 2.25
(h) 5

Question 14

Classification loss We consider point classification as in the course note Chapter 8. The network takes in 3D points and assigns them to one of the two classes, labeled 0 and 1.

A part of the data is a batch of 6 points (here given as matrix rows)

$$P = \begin{bmatrix} 0.2 & 0.5 & 0.1 \\ -0.8 & -0.1 & 0.1 \\ 0.5 & 0.4 & -0.3 \\ -0.5 & 0.7 & 0.1 \\ 0.3 & -0.3 & 0.2 \\ 0.2 & 0.1 & -0.6 \end{bmatrix} \text{ and associated targets } T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

The network output for this batch, before the last activation is

$$\hat{Y} = \begin{bmatrix} -0.097 & 0.537 \\ 0.185 & 0.261 \\ -0.021 & 0.532 \\ 0.215 & 0.244 \\ 0.191 & 0.132 \\ 0.221 & 0.549 \end{bmatrix}$$

where the first (left) column corresponds to class 0 and the second to class 1. Elements of P, T and $\hat{Y}$, placed together in a matrix $[P, T, \hat{Y}]$ are available in the data folder as `in_t_out.txt`.

The network uses softmax activation in the last layer, and cross entropy loss.

What is the total loss for this batch?

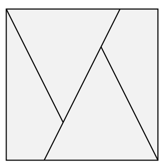
(a) -0.80
(b) 0.80
(c) 2.15
(d) 2.77
(e) 4.93
(f) 8.52

Question 15

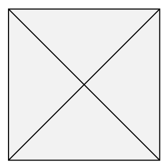
MLP classification An MLP has been trained for point classification as in the course note Chapter 8. The network takes in a 2D point from $[-1, 1] \times [-1, 1]$ and assigns it to one of the 4 classes. There are two hidden layers with 6 neurons (5 + bias) each The network uses ReLU activation in the hidden layers and softmax in the last layer.

The weights of the trained network are available in the files `W1.txt`, `W2.txt` and `W3.txt`. In all matrices the first (leftmost) column contains weights of the bias neuron.

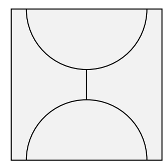
Which of the following patterns has the network been trained on?



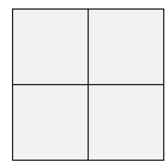
(a)



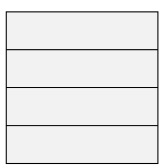
(b)



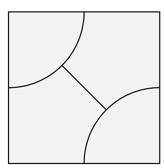
(c)



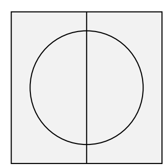
(d)



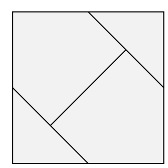
(e)



(f)



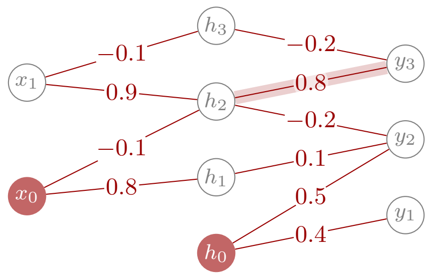
(g)



(h)

Question 16

MLP backpropagation Consider a classification network



with ReLU activation in the hidden layer, softmax in the last layer, and cross entropy loss. We know that the input point $x_1 = 0.5$ should belong to the class corresponding to then neuron y_2. We back-propagate the loss L for this point and we focus on the edge $w^{(2)}_{32}$ (highlighted).

What is the partial derivative

$$\frac{\partial L}{\partial w^{(2)}_{32}}\,?$$

(a) -0.252
(b) -0.247
(c) 0.098
(d) 0.103
(e) 0.252
(f) 0.247

Question 17

Stochastic gradient decent Opimization of neural networks are often based on stochastic gradient decent (SGD). Why is this algorithm called *stochastic* gradient decent?

(a) The SGD algorithm computes the gradients of the network parameters based on the input and the target training data. Then it iteratively updates the parameters by small steps in the negative gradient direction. The algorithm is called stochastic because the network is randomly initialized and the optimization will therefore vary randomly.

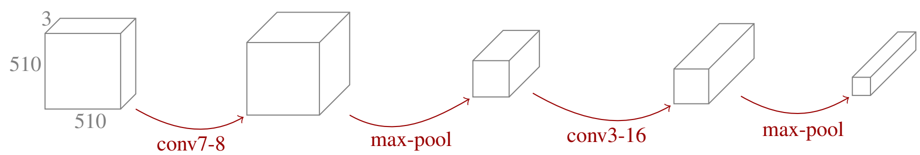
(b) The SGD algorithm computes the gradients of the network parameters based on the input and the target training data. Then it iteratively updates the parameters by small steps in the negative gradient direction. However, not all training data is used in each iteration, but only a random subset of the data. Therefore, the algorithm is termed stochastic.

(c) The SGD algorithm computes the gradients of the network parameters based on the input and the target training data. Then it adds a small random number to each gradient before it iteratively updates the parameters by small steps in the negative gradient direction. Therefore, it is called stochastic.

(d) The SGD algorithm draws random values for gradients and iteratively updates the parameters by small steps in the negative gradient direction. If the update results in a decrease of the loss, the updates are kept, and if not, the parameters are not updated. Therefore, the algorithm is termed stochastic.

Question 18

CNN receptive field We have a convolutional neural network as sketched in the drawing below.



We start with an RGB image of size $510 \times 510 \times 3$. The image is convolved with 8 filter kernels of spatial dimensions 7×7 and no padding. Then we have a max pooling in a 2×2 neighborhood with down scaling. Following this, we convolve with 16 filter kernels of spatial dimensions 3×3, again with no padding and followed by max pooling in a 2×2 neighborhood with down scaling. Each convolutional filter also has a bias neuron.

The receptive field is the area of pixels in the input layer that is connected to one neuron in the output layer. What is the receptive field of this convolutional neural network?

(a) 1×1
(b) 2×2
(c) 3×3
(d) 4×4
(e) 5×5
(f) 6×6
(g) 8×8
(h) 10×10
(i) 12×12
(j) 14×14
(k) 16×16
(l) 18×18
(m) 20×20
(n) 21×21

Question 19

CNN parameters We use the same convolutional neural network as in the previous question. I.e. an RGB image of $510 \times 510 \times 3$ as input. First, 8 convolutional kernels of 7×7 with no padding followed by 2×2 max pooling with down scaling. After that, 16 convolutional kernels of 3×3 with no padding followed by 2×2 max pooling and down-scaling. Each convolutional filter also has a bias neuron.

How many learnable parameters are there in this convolutional neural network?

(a) 0
(b) 60
(c) 560
(d) 576
(e) 1168
(f) 1184
(g) 2328
(h) 2352
(i) 2360
(j) 250000

Question 20

Data augmentation Which statement is correct?

(a) Data augmentation is widely used in deep learning for image analysis, and is used to obtain higher precision on test data. It is however important to consider your deep learning problem and avoid unsuitable augmentation. An example is flipping or rotating images of digits so they change meaning.

(b) Data augmentation is guaranteed to improve performance of deep learning models and the more augmentation that is applied the better.

(c) Data augmentation is not suited for images since images are recording of physical objects and by distorting or changing intensity, the meaning of the images will be changed.

(d) Data augmentation is widely used in deep learning for image analysis, but only steps of 90° rotations, flips, and intensity change can be recommended. If the image is rotated with another angle, there will be black areas that will decrease performance of the trained neural network.