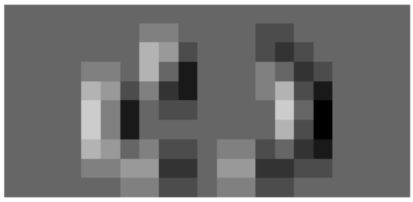
Institut für Mess- und Regelungstechnik
mit Maschinenlaboratorium
Karlsruher Institut für Technologie
(KIT)
Prof. Dr.-Ing. C. Stiller

Solutions for exam “Machine Vision” March 18, 2024

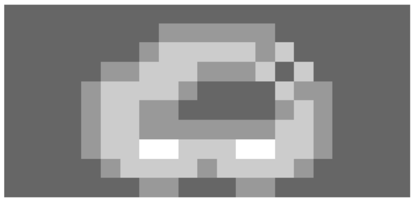
Question 1 (5 points)

Different kernels are convoluted with the graylevel image above. Below you can see the output of the convolution operation. Which kernel belongs to which output image? Draw a line between the matching output image and kernel.

$$\begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 2 \\ 0 & 2 & 0 \end{bmatrix}$$



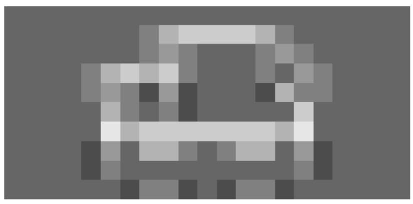
$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Solution

Image 1 = K2
Image 2 = K1
Image 3 = K4
Image 4 = K6
Image 5 = K5

Question 2 (8 points)

Implement a Python function **is_underexposed** that evaluates whether a video is underexposed.

A video is considered underexposed if a certain percentage of all pixels within the video is underexposed.

The function takes three arguments:

- A video of shape $[F, C, H, W]$, with F being the amount of frames, C the channels, H the height, and W the width.
- A threshold defining the percentage of underexposed pixels from which on a video is considered underexposed.
- A threshold defining whether a pixel is underexposed.

```
import numpy as np

def is_underexposed(video, t_vid = 0.3, t_pixel = 0):
  # convert RGB video to grayscale video
  gray_video = to_grayscale(video)

  # @student:  implement function
```

Solution

```
  underexposed_pixels = len(video[video <= t_pixel] == True)
  num_pixels_in_video = np.prod(gray_video.shape)
  percentage = underexposed_pixels / num_pixels_in_video
  return percentage >= t_vid
```

Question 3 (6 points)

The figure below shows a graylevel picture with 5 rows and 5 columns.

(a) Apply the connected components labeling algorithm (CCL) to segment the image. Two gray values should be treated as similar only if they are equal, i.e. gray pixels and white pixels in the figure below are always dissimilar. Visualize the processing of CCL by inserting into each cell in the figure one of the following letters to indicate the processing step:

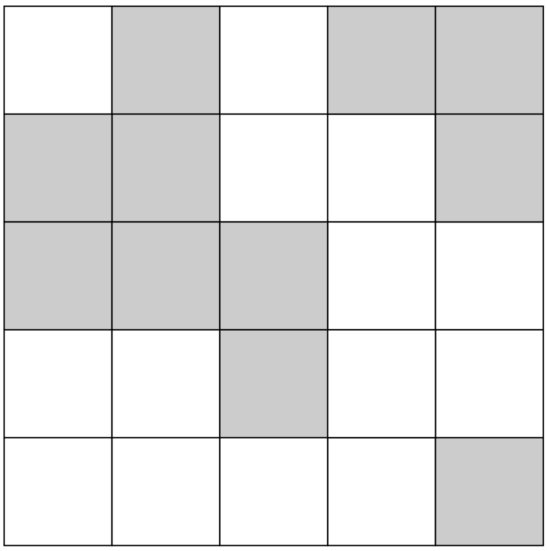
N if CCL initializes a new segment at this pixel

M if CCL merges two segments when passing this pixel

L if CCL assigns the pixel to the segment of its left neighbor, but not to the segment of its upper neighbor

U if CCL assigns the pixel to the segment of its upper neighbor, but not to the segment of its left neighbor

B if CCL assigns the pixel to the segment of its left neighbor and the left and upper neighbor already belong to the same segment



(b) How many segments are finally found?

Solution

a)

N	N	N	N	L
N	M	U	L	U
U	B	L	U	L
N	L	U	U	B
U	B	L	M	N

b) 5 segments are found.

Question 4 (8 points)

In a project you need to calibrate a camera. Your colleague already executed two different camera calibration methods resulting in two sets of calibration parameters R_1, t_1, A_1 and R_2, t_2, A_2.

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad R_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$t_1 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix} \qquad t_2 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 20 & 0 & 1000 \\ 0 & 40 & 1000 \\ 0 & 0 & 1 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 20 & 0 & 1000 \\ 0 & 30 & 1000 \\ 0 & 0 & 1 \end{bmatrix}$$

Furthermore, he recorded one world and image point pair:

$$\begin{bmatrix} 1.0 \\ 0.5 \\ 0.5 \end{bmatrix} => \begin{bmatrix} 1060 \\ 1065 \end{bmatrix}$$

Unfortunately, due to his child being sick, he cannot attend work this week and you need to decide which calibration parameters to use. Calculate and explain your reasoning.

Solution

We will reproject the point and calculate the reprojection error to see which calibration is closer to the test point. projection: $z * \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A * R_t * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$

reprojection error: $\frac{1}{N} \sum_i^N \sqrt{(u_i - u_{pi})^2 + (v_i - v_{pi})^2}$

Step 1: project to camera coordinate system (is the same for both methods):

$$Rt * \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.0 \\ 0.5 \\ 1.0 \end{bmatrix}$$

Step 2: Project

$$A_1 * \begin{bmatrix} 1.5 \\ 1.0 \\ 0.5 \\ 1.0 \end{bmatrix} = \begin{bmatrix} 530 \\ 540 \\ 0.5 \\ 1 \end{bmatrix}$$

$$A_2 * \begin{bmatrix} 1.5 \\ 1.0 \\ 0.5 \\ 1.0 \end{bmatrix} = \begin{bmatrix} 530 \\ 530 \\ 0.5 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} 1060 \\ 1080 \end{bmatrix}, \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} 1060 \\ 1060 \end{bmatrix}$$

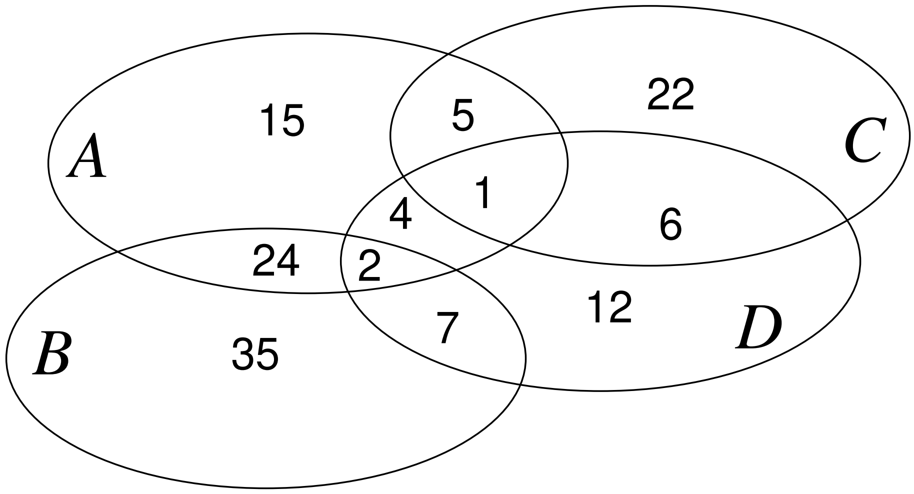
Step 3: Calculate reprojection error
$e1 = \sqrt{(1060 - 1060)^2 + (1065 - 1080)^2} = \sqrt{-15^2} = 15$
$e2 = \sqrt{(1060 - 1060)^2 + (1065 - 1065)^2} = \sqrt{-5^2} = 5$

As the reprojection error of the second calibration e_2 is smaller than e_1 we favor e_2.

Question 5 (8 points)

Assume that we train four different classifiers A, B, C, D for a certain classification task and test them on a test set. The figure below shows a Venn diagram of the number of misclassified test examples. Each ellipse represents one of the four classifiers. The numbers provide how many test examples are misclassified by the respective classifiers. E.g., there are 15 examples misclassified by classifier A while all the other classifiers are correct. There are 5 test examples which are misclassified by classifier A and C while the classifiers B and D are correct. There is one test example that is misclassified by the classifiers A, C, and D, while classifier B is correct.

For each possible ensemble of three of these classifiers calculate the number of test examples that are misclassified.



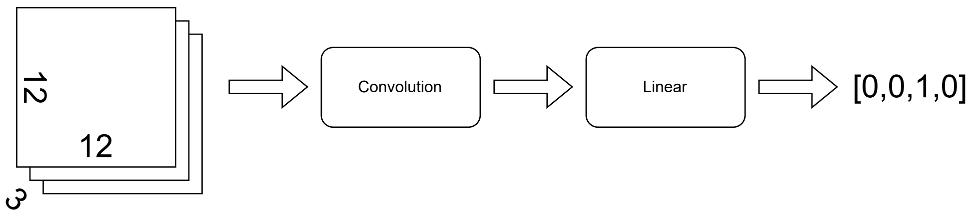
ensemble members	misclassified test examples
A,B,C	
A,B,D	
A,C,D	
B,C,D	

Solution

There are four possible ensembles of size 3. The number of test errors is equal to the number of test examples that are misclassified by at least two ensemble members. Hence, we obtain

ensemble members	misclassified test examples
A,B,C	$24 + 2 + 1 + 5 = 32$
A,B,D	$24 + 2 + 4 + 1 + 7 = 38$
A,C,D	$5 + 1 + 6 + 4 + 2 = 18$
B,C,D	$2 + 7 + 1 + 6 = 16$

Question 6 (3+3+3 points)



Assume the architecture of the neural network in the image above predicts four different classes of images. The input of the architecture is an image of shape $[C, H, W]$ with $C = 3$ (channels), $H = 12$ (height), and $W = 12$ (width).

(a) Assume a filters dimension is $[3, 3, 3]$ and 10 filters are learned in the convolution layer. Calculate how many parameters the convolution layer has.

(b) After the convolution layer is applied to the image, you end up with a feature mask of shape $[10, 10, 10]$. To predict the class probabilities, a linear layer is applied to the flattened feature masks.
Calculate how many parameters the linear layer has.

(c) Calculate how many multiplications are required to execute the convolution layer.

Solution

(a) Formula: $F * (k_h * k_w * C_{in} + 1) => 10 * (3 * 3 * 3 + 1) = 28 * 10 = 280$

(b) Formula: $T * ((F * H * W) + 1) => 4 * ((10 * 10 * 10) + 1) = 4 * (1001) = 4004$

(c) Formula:
$((H - 2) * (W - 2)) * (k_h * k_w * C_{in}) * F = (10 * 10) * (3 * 3 * 3) * 10 = 100 * 27 * 10 = 27000$

Question 7 (1+2+3 points)

Maria wants to train a deep-learning model to classify images of eight different road signs. She collects a data set of 500 images. She uses 400 images for training, 50 images for validation, and 50 images for testing.

To select a model she looks at a leaderboard and chooses the best performing model available. The model is a transformer model and has over one billion parameters. For training, she shuffles the training data and trains for 50 epochs.

After training, she achieves a perfect accuracy of 100% on the training data. However, on the validation data she only achieves an accuracy of 12.4%.

(a) Name the phenomenon that Maria is faced with.

(b) Explain why this occurs.

(c) Name three countermeasures to avoid this phenomenon.

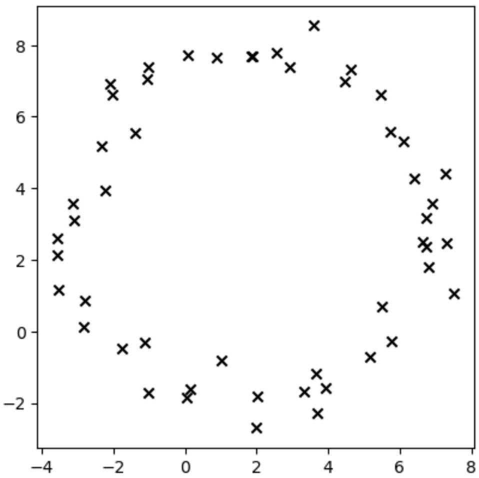
Solution

(a) Overfitting

(b) The model is too big for the task. It solely learns the training dataset

(c) 1) use a smaller model. 2) get more data. 3) augment data.

Question 8 (10 points)



In the image above you can see 2D points extracted by a machine vision algorithm which represent a circle. However, to further process this circle your application requires it in a parametric form.

Extend the RANSAC algorithm to detect circles in an unordered set of 2D points. Based on the RANSAC algorithm, describe your extension step by step.
Explain your steps mathematically. You do not need to fit a circle to the given sample data.

Hint: the equation of a circle is $(x - x_c)^2 + (y - y_c)^2 = r^2$

Solution

Step 1: Sample 3 random points from the set

Step 2: Fit a circle between the three points (x_c, y_c, r).
We create three equations from the three points.
$(x_1 - x_c)^2 + (y_1 - y_c)^2 = r^2$
$(x_2 - x_c)^2 + (y_2 - y_c)^2 = r^2$
$(x_3 - x_c)^2 + (y_3 - y_c)^2 = r^2$
Next solve for x_c, y_c, and r. There are multiple ways to do this. For example, we could subtract the first from the second, and the first from the third equation. Then we solve for x_c, y_c using the two equations and substitution.
$(x_1 - x_c)^2 + (y_1 - y_c)^2 - r^2 - ((x_2 - x_c)^2 + (y_2 - y_c)^2 - r^2)$
$(x_1 - x_c)^2 + (y_1 - y_c)^2 - r^2 - ((x_3 - x_c)^2 + (y_3 - y_c)^2 - r^2)$
Expand and simpify:
$x_1^2 + y_1^2 - x_2^2 - y_2^2 + 2(x_2 x_c - x_1 x_c + y_2 y_c - y_1 y_c)$
$x_1^2 + y_1^2 - x_3^2 - y_3^2 + 2(x_3 x_c - x_1 x_c + y_3 y_c - y_1 y_c)$
Reorder the first one to get x_c:
$x_c = \frac{2(y_1 y_c - y_2 y_c) - (x_1^2 + y_1^2 - x_2^2 - y_2^2)}{2(x_2 - x_1)}$
substitute x_c (a) in the second equation and solve for y_c.
$y_c = \frac{x_1^2 - x_3^2 + y_1^2 - y_3^2 + 2x_3 a - 2x_1 a}{2(y_3 - y_1)}$

Now we have a y_c that does not depent on y_c or x_c and back-substitute in the first equation to get x_c.
Using x_c, y_c and one point we can calculate the radius r

Step 3: Calculate the number of inliers and outliers.
For each point we calculate the distance to the center of the circle
$d = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ and compare it to the radius of the circle r. If the distance is within the radius and some threshold the point is an inlier, otherwiese its an outlier.

Step 4: Iterate

Gesamtpunkte: 60