CV questionnaire

Simon Stevens

July 2021

Table of Contents

1. Introduction

This chapter is an introduction to computer vision.
be there

1.1 Design Laws

The design laws that were considered in detail:

- Law of proximity ¨
- Law of Similarity
- Law of continuity¨

There are many more design laws.
E.g:

- Law of smooth history
- Law of closure

Law of proximity: ¨

Spatial proximity is an important clue for our analysis of the taken pics.

E.g:

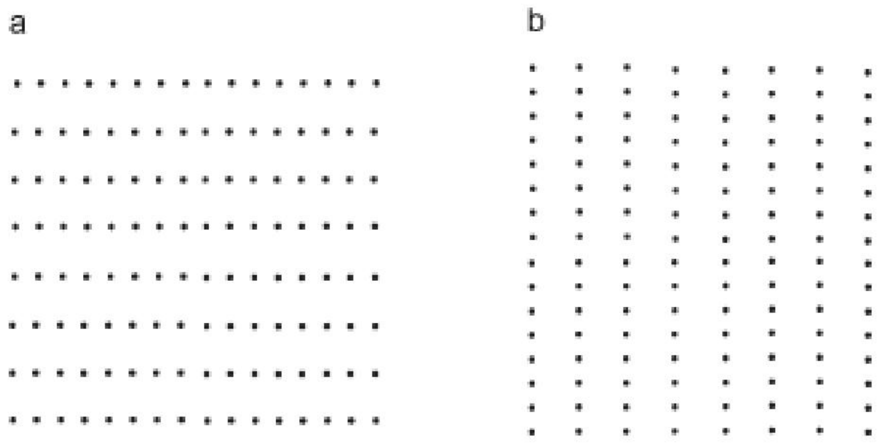

Figure 1: Columns and rows just by nearing the points

In Figure 1 we see impressively as only by the spatial proximity ¨ of the points to each other are perceived as rows and columns will.

Law of Similarity:

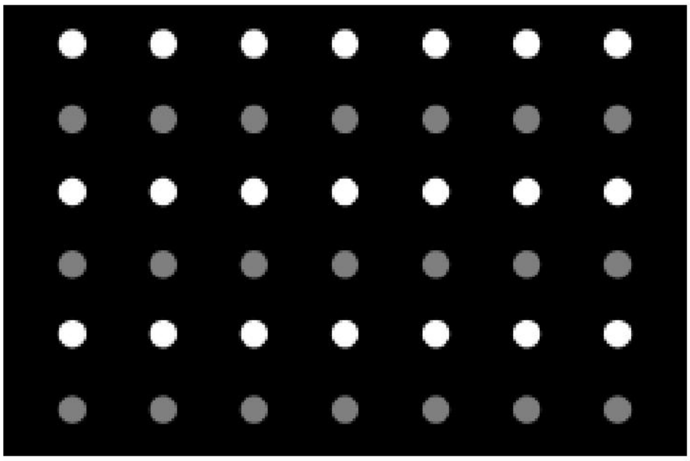

Figure 2: Elements have similar brightnesses

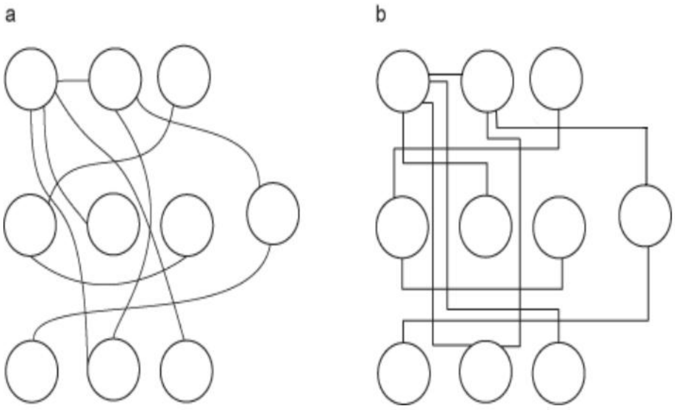

Figure 3: smooth gradient and no smooth gradient

We group objects that are similar to each other. For example in color, brightness or shape. We see an example of this in figure 2.

Law of continuity and good continuation:

That of smooth progression or good continuation states that our perceptual system forms a better unity when the elements are divided by one smooth, jerk-free and continuous connection are connected to each other. Figure 3 is an example of this. The law of continuity is closely connected to this. We instinctively continue the course when we receive a directional impulse. Figure 4 provides an example here.

1.2 optical illusions ¨

Optical illusion

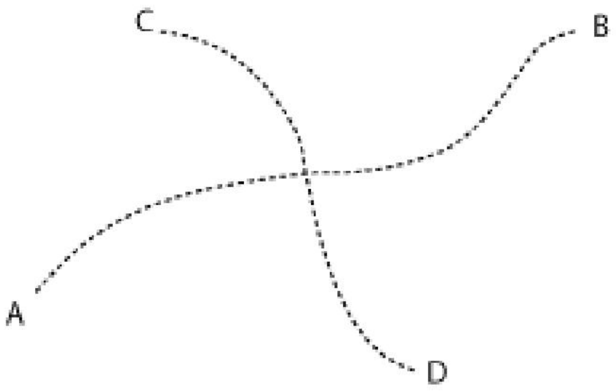

Figure 4: Continuity

- Color perception influenced by backgrounds, for example
- Difference in length although there is none due to attachments

1.3 What can humans do better than computers? know?

The computer can do better:

- Distinguish gray values
- Distinguish colors
- Measure

Humans can do better:

- Interpretation of contours if, for example, parts are missing
- Distinguish background from object
- Interpret objects when shadows and reflections occur

An analytical computer system always needs a definition of background and object. Humans can switch back and forth between different backgrounds based on our focus. Therefore we can recognize the two faces in figure 5 without any problems.

A NN could also recognize these two faces. It had to learn faces once with a black and once with a white background. Besides that, pictures had to be trained in profile.

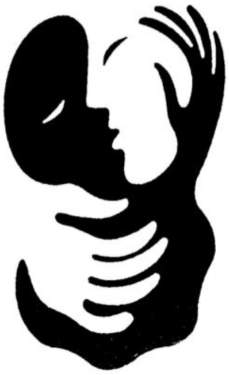

Figure 5: Two faces

A computer system has problems dealing with shadows and reflections. Calculating these effects is crucial to get the objects right
to recognize. Because of this limitation, it is very difficultfor a computer vision system to see the checkerboard pattern in Figure 6.

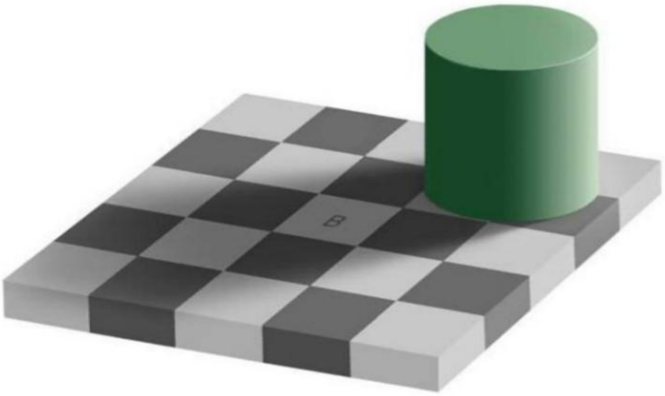

Figure 6: Chess board with shadow

2 Analytical computer vision

Analytical computer vision describes the classic approach to computer vision. Here the analyzes are made by programming models,
that people came up with and thus defined the rules, such as how something must be classified.

2.1 How do you compare two images of the same size?

How do people compare two images of the same size?
Humans scan images in so-called saccades. These are jerky eye movements. It will always refer to Striking points in the
Field of view focused. At such points the eye lands its gaze
very often.

Humans match two images of the same size roughly as follows. search for:

- differently colored surfaces ¨
- or differently patterned surfaces ¨
- geometric shapes that have changed
- if that doesn't help either, I'll go row by row and column by column ¨ through and hope to find something else

In other words, mental differences are formed between the images.
The computer system can calculate differences per pixel and the sum
form the error. There are several ways to do this, which are described below ¨
Subchapters are highlighted.

2.1.1 Different variants for distance calculation

Sum of differences:

To calculate the difference between two images of the same size, you can use simply do as in the following equation.

ÿ I(x, y) ÿ R(x, y) (1)

The disadvantage of this is that positive and negative distances mutually and no can lift. Thus, under umst probably it is present. difference is recognized if

Sum of the differences: ¨

ÿ|I(x, y) ÿ R(x, y)| (2)

The disadvantage of canceling positive and negative differences is eliminated. However, many small differences add up here quickly becomes a big difference.

In addition, use in neural networks is not advisable, since a symmetrical constellation uses the gradient descent method to minimize errors can lead to a dead end. Figure 14 illustrates the problem.

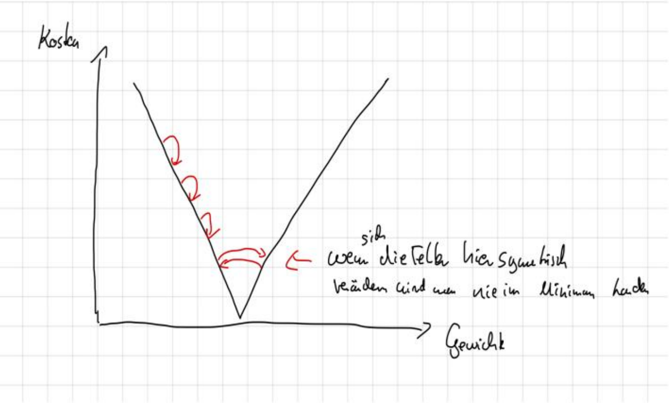
Kosten
sich
wenn die Fehler hier symmetrisch
verändern wird man nie im Minimum landen
Gewicht

Figure 7: Symmetry of error update in absolute value function

Sum of the squared differences:

$$\text{ÿ}\ q\ \overline{(I(x, y)\ \text{ÿ}\ R(x, y))2} \tag{3}$$

Due to the quadratic function, it is much less likely to fall into a symmetrical constellation. Taking the square root and the discretization that entails makes it even less likely. That is ideal for NN. However, for the comparison of 2 images this is very expensive. The function is expensive. In addition, it is also the case with this function that many small errors add up to large differences. If we do the root We didn't let the problem go, because small differences will be even smaller and larger differences are weighted more heavily here.

So the following equation is the most skillful when comparing 2 images:

$$\text{ÿ}(I(x, y)\ \text{ÿ}\ R(x, y))2 \tag{4}$$

2.1.2 Explain correlation k ¨ on

The correlation describes the distance between 2 images. If 2 pictures very are similar, the correlation is very high. In general, the correlation designates a linear statistical connection between two data sets.

So if datasets are independent, they are also uncorrelated. The reverse is not true, however, since there are also, for example, quadratic relationships in addition to linear relationships.

2.2 Comparison of two images that are not the same size: Template Mat explain!

Template matching searches for a template in an image. e.g. a small one Find an image section again in an image. How to go about it You put that Template at the top left of the image and calculates the correlation between the template and the image section on which the template is currently located. this one Value is then stored at the position of the first pixel in the top left of the image. after that you shift the template by one pixel column and calculate again the correlation. At the end of a pixel column, a pixel row is shifted down and the template is shifted from left to right again and the correlation is calculated in each case. The pixel position that shows no deviation, i.e. the correlation is maximum, is the position of the template. In the following two figures, the procedure is shown using an example illustrated.

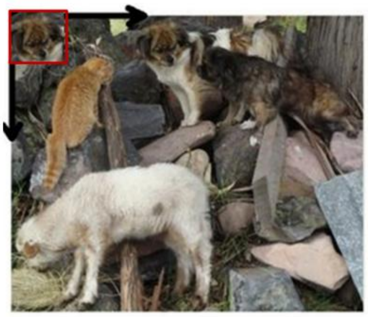

Figure 8: Procedure TM

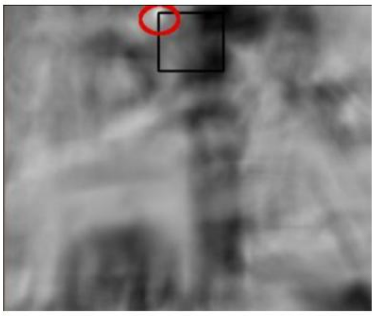

Figure 9: Result TM

2.2.1 How can the HSV color model be useful in this?

Unlike the RGB model, the HSV color model does not save the image information in the three colors RGB, but splits the image information into color (H), saturation (S) and brightness/intensity (V). ¨ ..
For template matching, the brightness components of the HSV system are sufficient to calculate the correlation between two images. Thus saves you have to calculate the effort you only have to compare one channel with each other.

2.2.2 Is the method scaling and rotation invariant?

The method is neither scaling nor rotation invariant. It will be accurate searched for the template in an image. If this template is twice the size like the figure in the picture will calculate the difference, do not recognize that the pictures are the same. Because you actually aren't.
The same applies to rotations. If the template is rotated compared to the figure in the image, the correlation calculation will not find the spot be able. ¨

You could of course run the template over the image in different rotations and scalings. However, this results in so many combinatorial possibilities that this is not practical.

2.2.3 Practical application for template matching.

Examples are AR applications where markers are used to e.g. to place objects in the real world. Here, of course, a perspective distortion must be calculated.

2.3 Which image areas are prominent? Or to put it another way: which image areas can be detected well with a computer vision system?

Unique features are, for example:

- A specific grouping of white pixels on a black background
- Edges and corners in a specific relationship to each other
- Local patterns in the image that create a striking structure with striking have properties and thus have a recognition value

2.4 What is feature tracking?

Feature tracking is the tracking down and tracking of the image areas that match an image template.

2.4.1 Into which three work sections is feature tracking divided? Describe what happens in the individual work steps.

Feature tracking is divided into the following work steps:

- Detector
- Descriptor
- matching

The detector traces the prominent image areas in an image. The descriptor describes these features in compact form. When matching, the corresponding image areas are selected based on the compact representation determined by the descriptor.

2.4.2 What influences can a change in the image features have?

Quality in feature detection means that the method is as usse is robust to change in image features. The following influences can result in such changes:

- perspective distortions
- other distortions when taking pictures and videos develop
- Lighting effects (shades, shadows, specular reflections)
- Affine transformations (rotations, scalings, shears)

2.5 Using selected images/examples to be able to discuss whether feature detection could work or not.

Example Pictures:

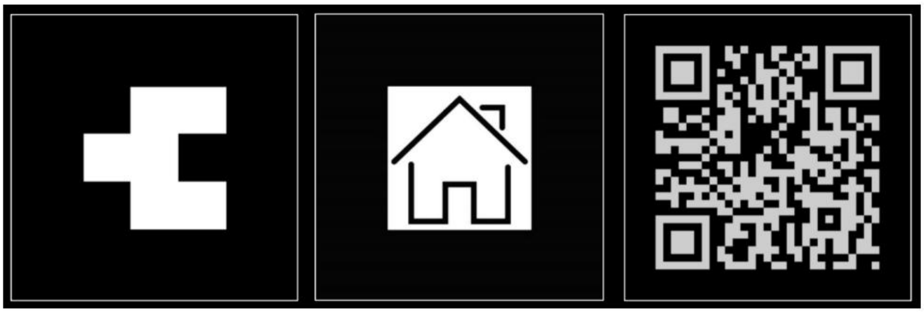

Figure 10: Feature detection examples

In the first picture, feature tracking wasn't working very well. There are very few distinctive areas. Most here are same color
Areas with a few transitions into other colors. As a result, there are only a few prominent areas such as corners or edges that are in specific positions relative to one another.
In the second image there are more prominent areas that can be discerned.
For example, this pictogram of a house has distinctive corners.
The QR code is very suitable for feature detection. Here there are very many corners and edges that stand in striking constellations to each other.
There are few and only small uniform areas.

2.6 Master the SIFT procedure.

First something general about the procedure: SIFT stands for Scale Invariant Feature Transform. It was developed by David G. Lowe in 2004. advantages of the procedure are:

- Scaling invariant! The distinctive features are reliably recognized in images with different resolutions (scaling).
- insensitive to affine transformations (especially rotations)

- Less sensitive to changes in viewing angle
- Less sensitive to changes in lighting Like every feature tracking

method, it is divided into the three areas of detection, description and matching.

The detection:

The detection is divided into the following 5 areas:

1. Detection of extreme points in the scaling spaces
2. Application of the DOG filter in the scaling spaces
3. The maxima and minima of the neighboring images filtered with DOG determine
4. Local extremes are determined: the key points
5. Determine the orientation of the key points

First, several scaling levels of the image are generated. The resolution of the image is halved in each case. Experience has shown that 4-5 scaling levels are ideal, but there is no generally valid measure.

Multiple copies of the scaling levels are then made which the create scaling space. These images will be in a scaling space then different Gaussian filters applied. The images are softened to varying degrees.

Next, the DOG filter is used to always form the difference from 2 different images in a scaling space. comparison Figure 11.

Next, the candidates for extrema in each scaling space are searched. Each pixel is connected to the eight neighboring pixels the same image and compared to the 9 pixels of the image above and below. Only those are extreme value candidates whose gray value is larger or smaller than all compared pixels. Comparison figure 12.

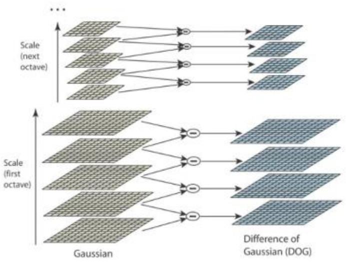

Figure 11: DOG - SIFT

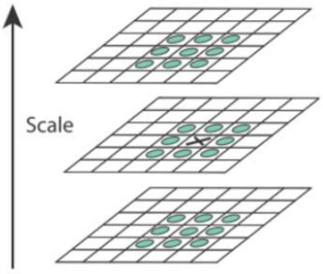

Figure 12: Extreme search - SIFT

The local extrema are now determined. So the candidates are still sorted out who are distinctive locally but differ from neighboring extremes do not differ strongly enough. This is done using Threshold procedure implemented. The intensity (grey value) is looked at. If the difference in the intensities of two adjacent extremes falls below a threshold value, they are sorted out. Finally, the key point candidates are sorted out on an edge, since these pixels are very similar are.

Finally, the orientations of the key points are determined. It will an alignment is determined for each key point. This makes the method less sensitive to rotations.
In order to determine the alignment, a region around the key point is defined depending on the scaling space. In this region, the gradient (slope in the gray value range) of each individual pixel. The gradients will weighted by their lengths, the direction of the gradients is discretized into 36 classes. In other words, at 10° angles there are classes for gradients forgive. The weights of the gradients in a class are summed up and then form the class gradient. The class with the greatest weight dictates the orientation of the keypoint. Should there be another class the relative to the class with the greatest weight also has a weight of at least 80%, two directions are assigned to the key point. So multiple orientation occurs in approximately 15% of cases. The stability of the at process is to be increased significantly in this way.

The description:
The description is made using a 16x16 pixel square around the relevant key point. The key point is in the middle. It will now the gradients of the 16x16 pixels are calculated and in 4 quadrants each the gradients combined. This is done by discretization in 8 directions per quadrant. In the summary play next to the Long the gradient is now also the proximity to the key point a role. So those that are closer to the key point are weighted more heavily. In order to make the features description less sensitive to changes in brightness, the gradients are also standardized to the range 0 and 1.

This ensures that large differences in brightness do not have an undue influence on the description.

The matching:
The matching is based on the following structure:

1. First, the key points are calculated for the entire camera image.
2. The keypoints are then compared with one another on the basis of the properties described by the descriptor.

3. In order to avoid erroneous assignments, ambiguous ones are used assignments discarded.
4. If there are several matches, the one with the highest match is chosen. If the difference between the two matches is very small, both will be discarded.

2.6.1 What are the benefits of building scaling spaces? or image pyramids?

True or false

1. Image pyramids help to gradually identify the structure of an image work.
2. Image pyramids enable the objects depicted to be displayed in different resolutions, so that the image content can be detected from different distances.
3. Image pyramids were specially designed for the SIFT, SURF and ORB methods developed.

1. is true
2. is true
3. is wrong?

2.7 Describe how to use the DOG filter fine and get wide edges.

You get broad edges if you subtract two rather heavily smoothed images from each other. You get fine edges if you subtract two weakly smoothed images from each other.

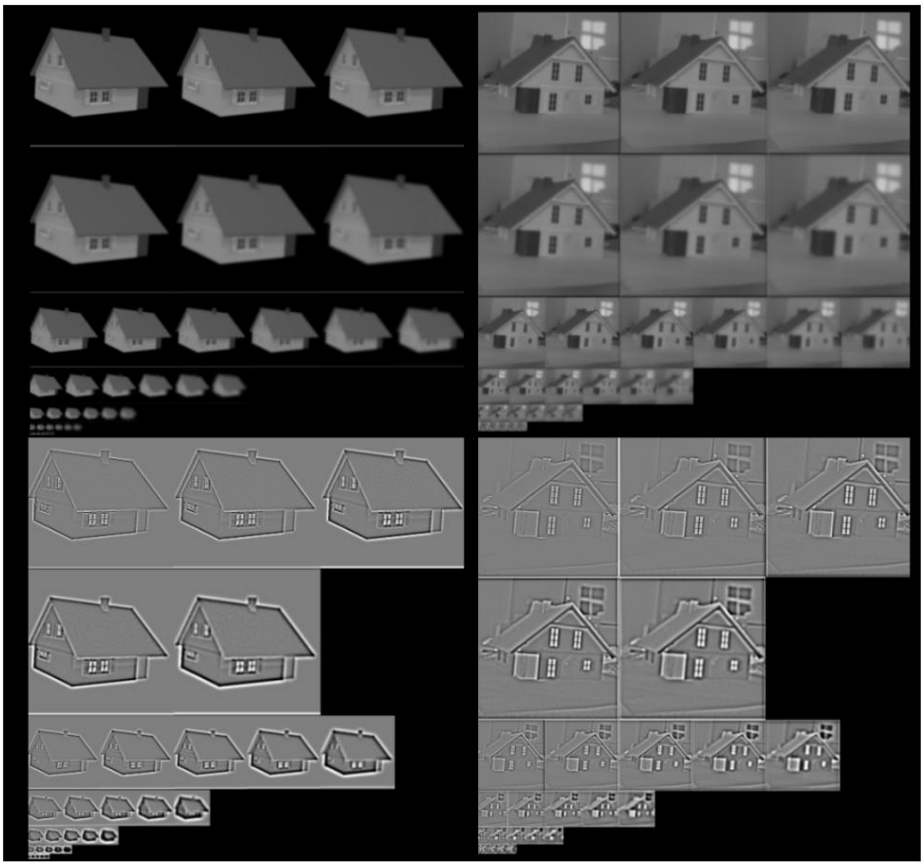

Figure 13: Wide and narrow edges in the DOG filter

2.8 Which pixel candidates do not form local maxima or minimums?

True or false?

1. Local extremes are pixels whose gradient exceeds or falls below a certain threshold.

2. Local extrema are pixels that lie on an edge

3. Local extrema are pixels that lie on a corner

1. is wrong

Reason: The gradient in relation to the gray value progression in an image in a local extrema must be 0 according to the definition of extrema and gradient.

2. is wrong

Pixels on an edge are usually not very different from their neighbors.

Therefore they do not form local extrema.

3. is true

Repetition of what was already mentioned in the general description of the SIFT procedure:

So the candidates that are considered locally are still sorted out

are striking, but do not differ sufficiently from neighboring extrema. This is realized by means of a threshold method. It will become the

Intensity (grey value) viewed. If the difference in the intensity of two adjacent ate extremes falls below a threshold value, they are sorted out. Finally, the keypoint

candidates are sorted out on an edge, that these pixels are very similar.

2.9 How does the description of prominent feature points work te?

Repetition from the overall description of the procedure.

The description is made using a 16x16 pixel square around the relevant key point. The key point is in the middle. It will now the gradients of the 16x16 pixels are calculated and in 4 quadrants each the gradients combined. This is done by discretization in 8 directions per quadrant. In the summary play next to the Long the gradient is now also the proximity to the key point a role. So those that are closer to the key point are weighted more heavily. In order to make the features detection less sensitive to changes in brightness, the gradients are also standardized to the range 0 and 1.

This ensures that large differences in brightness do not have an undue influence on the description.

2.10 Describe the principle of matching in the SIFT procedure.

Repetition from the overall description of the SIFT procedure The matching is based on the following structure:

1. First, the key points are calculated for the entire camera image.
2. The keypoints are then compared with one another on the basis of the properties described by the descriptor.
3. In order to avoid erroneous assignments, ambiguous ones are used assignments discarded.

4. If there are several matches, the one with the highest match is chosen. If the difference between the two matches is very small, both will be discarded.

2.11 Name areas of application for the SIFT method.

Areas of application are, for example, the following:

- Contactless biometric recognition of the venous system
- Iris recognition
- Stitching

Stitching refers to the joining together of a large image from several individual images. In-place merging is realized by detecting overlapping features. For example, creating panorama images In addition to SIFT, other methods can also be used, e.g. Harris corners and differences of Gaussians (DOG) to enable the stitching feature. Only the features must be reliably recognized will.

2.12 What is the significance of Mach's stripes for our perception?

Mach's fringes allow us to perceive transitions between different brightness levels as edges. This happens because receptors that absorb stronger light impulses inhibit neighboring receptors.
As a result, when going from a lighter to a darker area, very black and very light areas appear even in a very small area lines in our perception.
This again clarifies the importance of edges as prominent points in a picture.

2.13 On the basis of which ”more distinctive” pictorial elements do we get the image content?

First, we open up the image content at points with high contrast distinguished. However, not only corners and edges are relevant flat elements in a certain orientation to each other can contain very relevant information.

An example of how flat elements are relevant is when you look up at the sky and suddenly see elephants or other familiar shapes in the clouds.

2.14 Information from position space to frequency space" figuratively". ¨

A clear explanation with example is still missing. Example: Being able to show where, for example, edges of the position space can be found in the frequency space. However, only so far we did that in the lecture. So without phase shift and without calculations.

2.15 What is the meaning of the convolution theorem? Which two mathematical operations can be equated in their mode of action?

Convolutions in image space are equivalent to multiplications in frequency space and vice versa. Thus, convolutions in the frequency domain can do a lot map more easily. Thus, filters are in frequency space and convolutions are in image space equivalent operations. They therefore led to analogous results. They are not equal since there is a discretization in position space.

2.16 What are the two different types of filters and how are they calculated?

There are high pass and low pass filters. High pass filters are edge detectors and low-pass filters are blurrs. All other filters, such as bandpass filters, are derived from these two types.
This can be calculated, for example, by using convolution matrices on the images. The current pixel value is calculated using the surrounding pixels. The extent to which certain pixel values are taken into account is defined in the respective convolution matrix.
A second method is to transform the image into frequency space, for example with a Fourier transformation. In the frequency domain, the convolution can then be expressed as a filter of specific frequencies. With edge detectors, for example, the low frequencies are filtered out, which is why one it calls high-pass filters.

2.17 What does the sigma do in the Gaussian filter and how can it to use it in DOG filter?

The sigma is defined as the standard deviation. It makes the Gaussian curve wider or narrower depending on how large Sigma is. This affects in Gaussian filter the effect of blurring. A large sigma results in a strong blurring.

In the DOG, the procedure for detecting edges is very strong subtract a blurred image from a slightly blurred image. Thus the sigma for the one image must be large for the ¨ other small.
The advantage of the DOG filter is that it already filters out noise. He is a high pass filter.

2.18 What is the relationship between the gradients per pixel and the difference operator?

The gradient maps the slope of a function. In the one-dimensional is the slope a scalar. in several dimensions a vector which over the partial derivatives over the spatial directions is calculated. Only one spatial direction is ever treated as a variable and the others Spatial directions are assumed to be constant. One looks clearly so how the curve changes in each of the spatial directions. This then forms the gradient vector. This therefore always points in the direction of the biggest incline. The length of the vector indicates the amount of slope at the point. Derivatives are computed as limits of differences.

$$G^{0}(x) = \lim_{ÿxÿ0} \frac{g(x + ÿx)\ ÿ\ g(x)}{ÿx} \qquad (5)$$

In discrete image spaces, such as the values of a gray image, for example, in which the smallest jump is an increment of 1, this is simplified the calculation of the gradients to simple differences.

$$\frac{g(x + 1)\ ÿ\ g(x)}{1} = g(x + 1)\ ÿ\ g(x) \qquad (6)$$

These differences can be expressed as linear mappings in difference operators. For example, the following difference operator:

0 0	0	
0-1	1	
0 0	0	

2.19 Which difference operators do you know?

Robert operator:

Table 1: Robert operator ÿx

1 0	
0 -1	

Table 2: Robert operator ÿy

0 1	
-1 0	

Sobel operator:

Table 3: Sobel operator x direction

-1 0	1	
-2 0	2	
-1 0	1	

Table 4: Sobel operator y Direction

-1 -2	-1	
0 0		0
1	2	1

3x3 Prewitt operators:

Table 5: Prewit operator x Direction 3x3

-1 -1 -1		
0 0 0		
1	1	1

Table 6: Prewit operator y Direction 3x3

-1 0 1		
-1 0 1		
-1 0 1		

4x4 Prewitt operator:

Table 7: Prewit operator y Direction 4x4

-3 -1 1 3			
-3 -1 1 3			
-3 -1 1 3			
-3 -1 1 3			

Table 8: Prewit operator x Direction 4x4

3 3 3			3
1	1	1	1
-1 -1 -1 -1			
-3 -3 -3 -3			

2.20 Mastering two variants for implementing the filter operations?

There are two ways to implement filter operations. Variant A: that The filter result is saved in an intermediate image and then copied to the original image. Variant B: The original image is first copied into an intermediate image, and this is filtered. The result of the filtering is shown in the Original image saved.

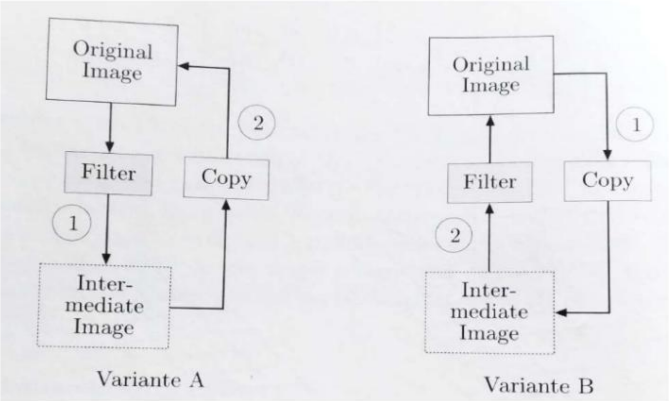

Figure 14: Implementation variants

2.21 How does the Roberts operator work, what does it detect and what advantage does it offer in implementation?

The Roberts operator forms differences of diagonal pixel values. That's why it consists of the following 2x2 convolution matrices:

Table 9: Robert operator ÿx

1 0	
0 -1	

Table 10: Robert operator ÿy

0 1	
-1 0	

It detects edges that can be seen in the image at a 45° angle particularly well.

Advantage in the implementation: The Roberts operator does not need between image. That's because if you take the dot of the first line and column on which 1 pixel is placed and whose operator value is calculated Pixel no longer needs in the further course. So it can go straight to that original image to be written.

2.22 What is a hysteresis?

A hysteresis attempts to piece together disjointed detected edges from the previous edge detection into a continuous line to merge. The hysteresis uses a threshold method. Two threshold values are defined, an upper and a lower one. a pixel is only accepted as an edge if the value of the gradient is above the upper threshold or if the gradient is between the upper and lower threshold and is also directly connected to a pixel whose gradient is above the upper threshold. Pixels below the lower threshold are not accepted as edges. That doesn't always work but small gaps can be closed.

2.23 What is the advantage of the Canny-Edge detector over ¨ the edge detectors based on the 1st and 2nd derivatives?

The Canny Edge detector is not as sensitive to image noise as it initially applies a Gaussian filter to the image.

2.24 Why do computer vision systems mainly use ¨ Corners as distinctive image features and not lines or monochromatic surfaces? ¨

At corners, the picture changes in almost all spatial directions. Therefore, these points are very good to see where you are in the picture. at areas of the same colour, you don't know where you are in the picture. Hardly any information It's better on edges, but at which point the edge is still not clear.

2.25 Is the definition of ”corner" clear? Argue you, why isn't that the case?

There are many different types of corners. for example, corners can include different angles. This becomes clear when you look at a square and looking at a triangle. In addition, corners in space can be oriented differently. They can be parallel to the axis or oblique to it. It is also possible that corners on one side are filled if e.g. the object to which the corner belongs is filled. If the object is only part of the contours then only the edge of the corner is e.g. white and it

goes back to another color, e.g. black.

2.26 What is the definition of ”Corner” in the FAST corner detector?

In the near-corner detector, a circle containing 16 pixels is placed around a point with the intensity Ip. There will be a threshold t matching image chosen. If n of the 16 pixels around the point under consideration lighter than the Ip + t are or darker than the Ip - t the point is marked as a corner. Im almost Corner detector was chosen n = 12. ¨

2.27 Which corners does the FAST corner detector find and which not?

Edges of the corners that are parallel to the coordinate axes are often used not recognized as corners. Due to the fact that the objects are normally not quite parallel to the axis, this hurdle can be overcome with the help of an upstream blur. ¨

The following image shows examples of which edges are recognized and which are not. Note the cross is probably recognized after all. Since probably only 4 points are not darker/lighter than the observed point.

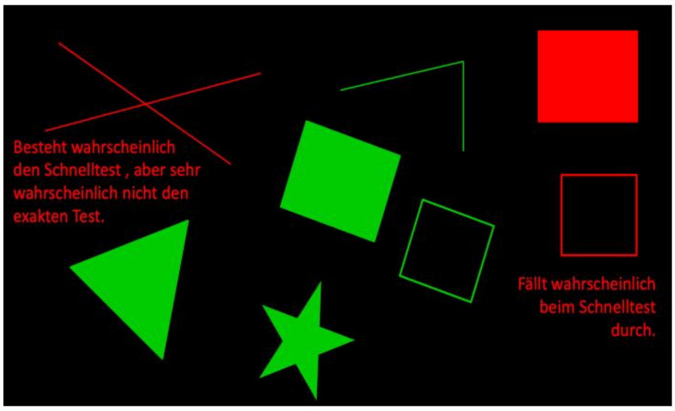
Besteht wahrscheinlich den Schnelltest , aber sehr wahrscheinlich nicht den exakten Test.
Fällt wahrscheinlich beim Schnelltest durch.

Figure 15: What does the FAST detector detect?

2.28 The FAST corner detector algorithm in example task be able to use.

The algorithm:
First it is checked whether the pixel can be a candidate at all. In the first step, only 4 pixels from the 16 pixels in the circle are transferred to the threshold checked. Figure 16 shows that pixels 1,5,9 and 13 to be checked. If at least 3 of the 4 points meet the corners criterion are met, the point is accepted as a keypoint candidate and all 16 pixels checked and seen if 12 pixels meet the criterion. If fewer than 3 of the 4 examined pixels meet the criterion, the pixel is discarded as a corner candidate and the next pixel is examined.

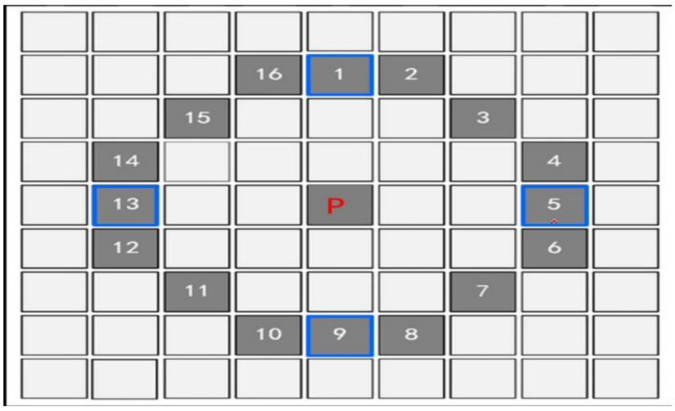

Figure 16: Fast Corner Detector FAST portion

3 Neural Networks

Neural networks are great :)

3.1 Explain the iterative procedure for determining the Discriminant function without learning rate.

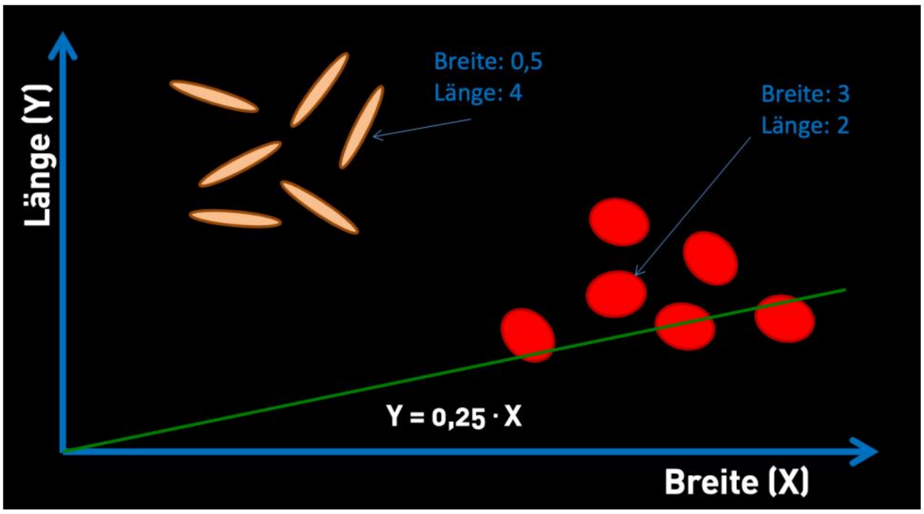

Figure 17: Creating a discriminant function

As an introduction, caterpillars and beetles were classified. We create a classification using the example of this classification. 2 characteristics are considered. The length in the Y axis and the width in the X axis. Assumption: caterpillars are always longer than wide, ladybugs are always wider than long. The discriminant function should become a straight line with a simplified definition of a straight line without an intercept.

$$Y = m \text{ x } X \tag{7}$$

The slope parameter is initialized randomly at the beginning, in this example with 0.25. The first ladybug now comes in with a width of 3 and a length of 2. According to our discriminant function, it should actually have a length of:

$$Y = 0.25 \text{ x } 3 = 0.75 \tag{8th}$$

to have. This is of course not correct. Next, the error of the i-th iteration is calculated.

$$Ei = 2 \text{ ÿ } 0.75 = 1.25 \tag{9}$$

Of course it would be better if you don't adjust m so that the borders are drawn exactly through the ladybug but above the ladybug.

Therefore the setpoint is changed to 2.25. So that the discriminant function runs above the beetle. The change in the i-th iteration ÿmi becomes

now calculated as follows:

$$ÿmi = \frac{egg}{xi} \tag{10}$$

The new slope m then becomes iterative

$$mi = miÿ1 + ÿmi \tag{11}$$

In our example, mi is calculated as mi = 0.25 + $\frac{1.5}{3}$ = 0.75

3.2 Then show the benefit you get from the Leveraging the learning rate. ¨

The problem with the procedure without a learning rate is that individual observations have a very strong influence on the discriminant function. if only such observations occur that support our assumption would be that ¨ no problem. However, there are random processes, including the big ¨ of ladybugs and caterpillars, runaways.

The learning rate helps to reduce the impact of individual observations and so we don't stray too far from the optimum outliers occur.

To stay with the example of our discriminant function, the learning rate ÿ is used in the calculation of ÿm.

$$ÿmi = ÿ \cdot \frac{egg}{xi} \tag{12}$$

3.3 Why is the classification called a statement? ”blurred

A classification is not perfect. It is always flawed. One can only say with a certain probability that a certain object belongs to a certain class. The principle is made clear again if you look at the example of the caterpillars and beetles again ¨

reminds. After observing one beetle, the sharp statement would have been a beetle with width 3 and length ¨ 2 is a ladybug and the discriminant function goes right through the beetle would leave

have to. We deliberately placed the line above the observed beetle ¨

and even limited the influence of the individual beetle by the learning rate.

In addition, we do not only want the objects with a classification describe exactly with which we have trained the classification. That would be an overfitting. Surely you could do something like that, but it would be ¨ Success on unknown dates not very high.

3.4 Based on a layer of a fully connected neural network, create a weight matrix and show that how dependent on the input values the output values are calculated taking into account the activation function.

The path of the input values through the network is called feedforward. the Figure 18 illustrates how the calculation works within an FCNN.

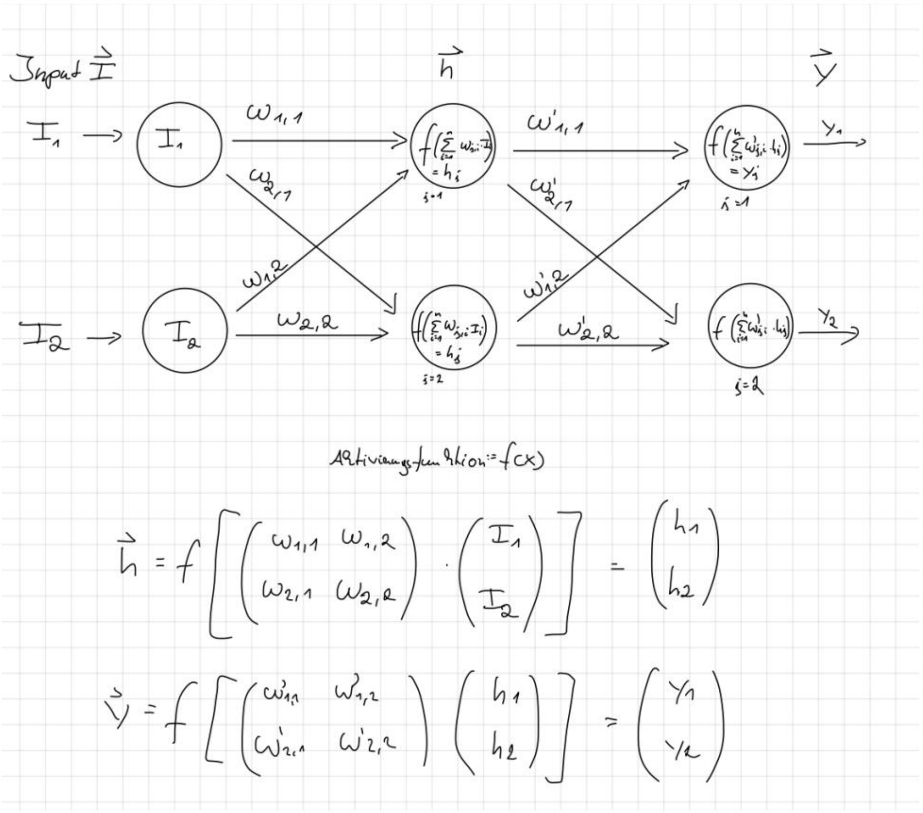
Input $\vec{I}$
$\vec{h}$
$\vec{y}$
I_1
I_2
$\omega_{1,1}$
$\omega_{2,1}$
$\omega_{1,2}$
$\omega_{2,2}$
$f(\sum_{i=1}^{n} \omega_{ji} \cdot I_i) = h_j$
$j=1$
$j=2$
$\omega'_{1,1}$
$\omega'_{2,1}$
$\omega'_{1,2}$
$\omega'_{2,2}$
$f(\sum_{i=1}^{n} \omega'_{ji} \cdot h_i) = y_j$
$j=1$
$j=2$
y_1
y_2
Aktivierungsfunktion := f(x)
$\vec{h} = f\left[\begin{pmatrix} \omega_{1,1} & \omega_{1,2} \\ \omega_{2,1} & \omega_{2,2} \end{pmatrix} \cdot \begin{pmatrix} I_1 \\ I_2 \end{pmatrix}\right] = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}$
$\vec{y} = f\left[\begin{pmatrix} \omega'_{1,1} & \omega'_{1,2} \\ \omega'_{2,1} & \omega'_{2,2} \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}\right] = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$

Figure 18: Calculation of the FCNN output

The values of the nodes in a layer are calculated using the weighted sum of the input values. These can either be the actual input values or the output of a hidden layer. However, the output values at a node are not simply the sum of the weighted inputs

but this sum is still used in the activation function.

This operation can be represented as a matrix multiplication with a vector. The weights form the entries in the weight matrix W of a layer. The vector with which it is multiplied are the input values or the output values of a hidden layer.

3.5 Show how the error/gradient ”correctly” distributed to the individual layers in the network?

The error is distributed to the individual layers in the network using backpropagation.

..

There are different approaches to distribute the errors to the layers. One approach is to make this proportional to the weights. This idea is illustrated in Figure 19.

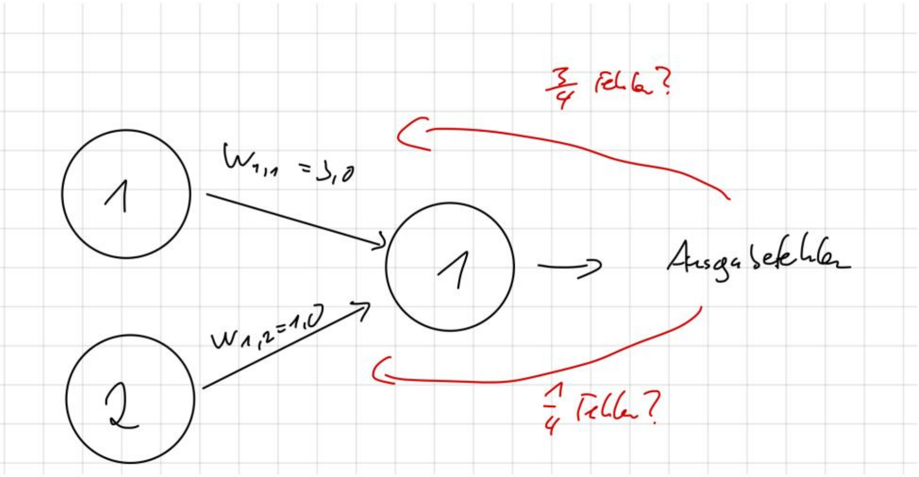

Figure 19: Backpropagation approach

In the following example we now calculate the distributed errors ehj of the hidden layer. According to Figure ¨ 20, the errors eh1 and eh2 are calculated as follows:

$$eh1 = e1 + e2 \frac{0w1.1 \_\_}{w0 + w0\ w0} + w0\ 1.1 \frac{0w2.1 \_\_}{}\quad 1.2\ 2.1\ 2.2 \tag{13}$$

$$eh2 = e1 + e2 \frac{0w1.2 \_\_}{w0 + w0\ w0} + w0\ 1.1 \frac{0w2.2 \_\_}{}\quad 1.2\ 2.1\ 2.2 \tag{14}$$

$$\begin{matrix} eh1 \\ eh2 \end{matrix} = \begin{matrix} ÿ \frac{0\ w1.1}{w0\ 1.1+w0\ 1.2} & \frac{0\ w2.1}{w0\ 2.1+w0\ 2.2} ÿ \\ \frac{0w1.2 \_\_}{ÿ\ w0\ 1.1+w0\ 1.2} & \frac{0w2.2 \_\_}{w0\ 2.1+w0\ 2.2\ ÿ} \end{matrix} \cdot \begin{matrix} e1 \\ e2 \end{matrix} \tag{15}$$

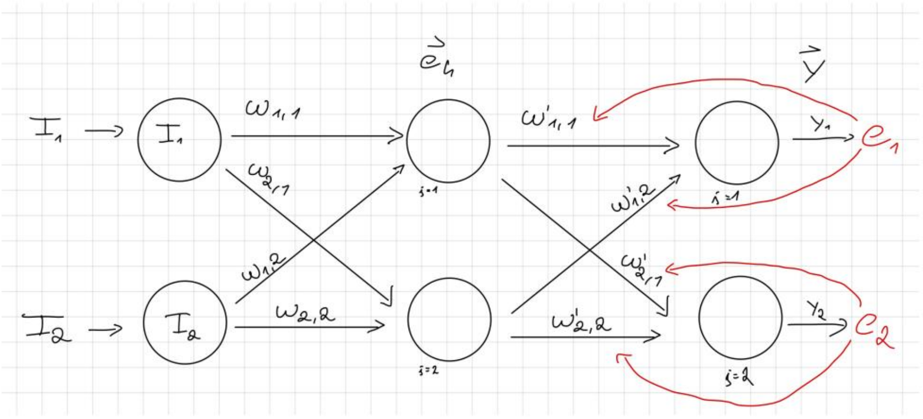

Figure 20: Calculation of the FCNN output

..

If you are not interested in the uniform distribution, you can also omit the denominators in the matrix when backpropagating. Leading to the fact that the matrix of the backbpropagation exactly the transpose of the feed forward matrix is. The problem is that this increases the errors per layer and makes more changes at the nodes that are close to the input.

..

3.6 Deeper question: What leads to disappearance or Exploding the gradients?

Deeper question: In order to be able to use the transposed weight matrix to distribute the error in the network, the denominator in the weight matrix was simply deleted (see slide *11* in parts *4* and *5* of chapter 2). It is only through this procedure that the error or gradient calculation has become unstable and disappears or explodes. Assume that in the numerical solution (derivation of the function F(X) presented in Chapter *4* , these *3* “inaccuracies”, the created by deleting the denominators are not present. What leads ¨ .. but then to the disappearance or explosion of the gradients? Explain your answer.

Answer:

Gradients can disappear or explode due to the influence of the different weights of the layers.

3.7 Can a global minimum of the error can be reached?

Depending on how you initialize the weights at the beginning of the training, you may reach different minimums. The gradient descent method always moves towards a local minimum. Whether this at the same time we don't know what the global minimum is.

In order to get as far as possible to a global minimum, neural networks are trained in many iterations. Different starting values are always used. This increases the chance of landing in a global minimum.
It can also help to choose a certain batch size that is not 1 and also does not equal the number of all data.
The idea is that this gives a vague impression of the error function but does not get the exact error function. Thus, it is more likely to skip local minima and end up in a global minimum.

3.8 What influence does the ”step” in the gradient ab ascent procedure in search of the minimum?

The stride means how far I am going in the opposite direction of the gradient towards a minimum. The increment should not be too be chosen large nor too small. If the increment is too small, it will take forever to get to a minimum. If it's too big step, there is a great danger that you will overtake the minimum. And a Symmetrical constellation can lead to that the minimum is never reached.

It is advantageous to choose a dynamic step size. As one approaches the minimum, the magnitude of the gradient should decrease. Accordingly, the learning rate should be very large if the gradient is large and the step size should be reduced if the gradient is smaller becomes.

3.9 What is a gradient?

A gradient is the generalization of the first derivative in several dimensions. Dimensions in this case are the weights. is derived here the error function after the weights. The gradient describes in each point of my weight readings is the direction of the greatest incline.

4 General information about NN

4.1 What is the difference between the programming paradigms ei ner application in machine learning from a len ”norma application?

In machine learning applications, the programmer sets the rules no longer before. He gives input and desired results to the machine learning application, which is trained accordingly and is iterative independently derives the rules.

4.2 Briefly describe what additional information(s) the data needs so that supervised learning can be implemented?

In addition to the data, you need the corresponding answers you want. So if you want to achieve a classification you have to go to the Data still add the appropriate class. ¨

4.3 Could you briefly explain the principle of reinforcement learning? Is learning monitored here? ¨

In reinforcement learning, results that are perceived by the outside world as good are rewarded and results that are perceived by the outside world as bad are penalized. It represents a weakened form of unsupervised learning. That is, learning is partially supervised by rewarding or punishing certain outcomes.

4.4 How did deep learning get its name?

Deep learning gets its name from the number of layers used and hence the depth of the network. If you have many hidden layers in a network, this is called deep learning. A fixed number of layers that a There is no need to have a deep learning network.

4.5 Describe the different procedures for the Finding solutions to problems with CNNs compared to finding solutions using methods of analytical computer vision?

CNNs are more about training a network appropriately. How do I find the appropriate training and test data sets so that my network is optimally trained. One may also think about how to choose the hyperparameters in order to achieve the best possible results, or you use algorithms that test different hyperparameters and

e.g. make an assessment of the adjustment based on the error.

With analytical computer vision methods you have to think carefully how to reach the goal and map the rules that lead to the goal accordingly in software. So it must be the situation in which the machinelle See work should be analyzed and the appropriate rules be derived and mapped in software.

4.6 Which tasks/applications are better to use Realize CNNs with methods of analytical computer vision?

Give examples and justify them.
An example of an application that can probably be done better with CNNs is the conversion of black and white images into color images.
This is because it is probably very complex to describe analytically all the rules of what should be which color and to map code, so that useful results can be obtained.
E.g. object recognition in moving images. CNNs are better suited for this, since the objects come into the picture from many different angles be able. Describing the relevant features here in such a way that they can be easily recognized by an analytical CV system is probably very difficult.

4.7 Which tasks can be better solved with methods of to realize analytical computer vision than with CNNs?

Analytical computer vision is more suitable for recognizing individual rigid objects, since it can be determined more precisely here and no fuzzy statements (probabilities) are used. If one can say from the outset that the rules, e.g. for object recognition, are fixed and do not change, then analytical computer vision systems are better suited.

Give examples and justify them.

4.8 Are there tasks that deal only with CNNs or only with the analytical computer vision realized?

If yes, which? Justify your answer.
Generating plausible images that did not exist before can only be realized with special CNNs (GAN). For example creating new faces.

5 Convolutional Neural Networks

5.1 Explain the general structure of a CNN ¨

possible important keywords: input data set (RGB images), epochs, batch, convolutional Layer, MaxPooling-Layer, Full Connected-Layer, Flatten, Convolution Kernel, Weights, ...

The structure of a CNN is shown in Figure 21.

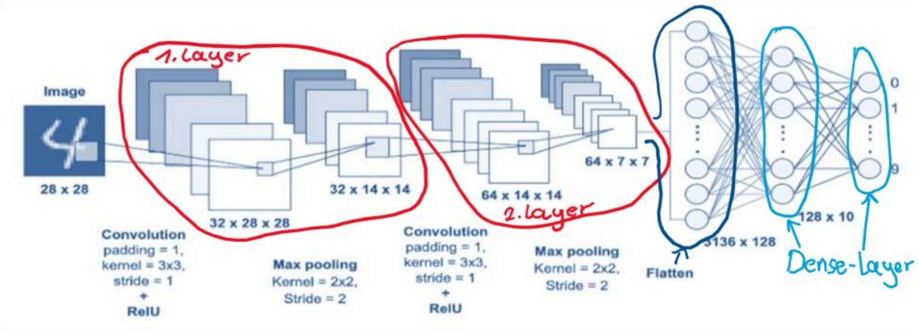

Figure 21: Structure of a CNN

It always starts with an entry in the·CV, usually an image that is in several convolutional layers are passed. A convolutional layer always consists of one or more convolution matrices with random ¨ Weighting is initialized and is followed by a pooling layer which implements a downsampling function. e.g. maxpooling. At the end of A fully connected layer follows the convolutional layer. The input for the fully connected layer is realized by using the result of the last MaxPoolings flattens. So every pixel value of every output image of the last con volutional layer becomes an input value of the FC layer. That determines them too Number of neurons in the first layer of the FC layer.
A CNN can be trained in so-called batches. That is a lot of images given through the net without following each one update the weights of the layers for each image. A backpropagation the error and an update using the gradient descent method therefore only takes place when a batch has been run through. An epoch is when all the images in the training data set have passed through the network once have run and the weights have been updated accordingly.

5.2 Explain and understand how CNNs work

The functional principle of a CNN is always the same. First, the details are worked out from the images in the Convultional Layers. That can, for example, be corner points or edges. Structures are worked out in the next layer. The information is then stored in further layers more and more abstracted and thus the structures and details encoded. In the last

Layer, which is always an FCNN, then becomes the actual classification performed.

The way CNN works is to extract features that .. to filter out the decisive features and to find an abstract description for them. On the basis of the abstract description, the Input record classified.

The features are worked out with convolution kernels and the abstraction and reduction of the information to a few pieces of information realized via Max-Pooling. This depicts the idea of the image pyramid in which the resolution of the image has been halved in each case and can thus describe the prominent areas independently of their scaling and position.

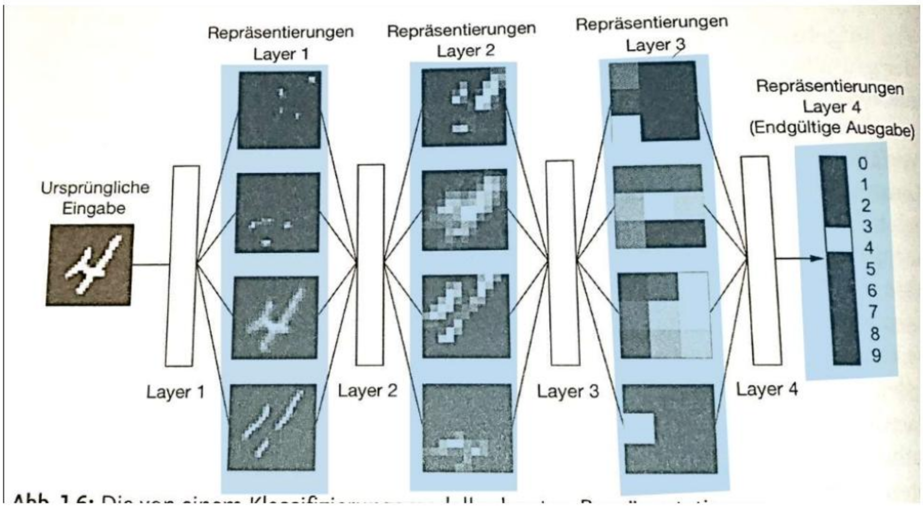

Figure 22: Layer outputs CNN (functional principle)

5.3 Demonstrate how the functionality of the CNNs is implemented with the components of the CNN.

The elaboration of the details and also the structures in further layers and further abstraction is done using convolution matrices. E.g. edges and corners can be worked out with appropriate folding matrices. What we have already learned in analytical computer vision to have. Retaining the information across multiple layers and scaling levels is realized with MaxPooling. The MaxPooling forms the mode of action of the image pyramid used in the SIFT process will, off. In the image pyramid, different scaling levels of the images generated. In the SIFT process, the resolution of the images was halved in each step. Maxpooling does something similar. The idea is to keep only the pixel value of the highest ¨ has value. The assumption is that such pixels are the most important information wear and therefore be saved in the next layer m outside.

..

5.4 Based on the components and structure of the CNN, can you clarify whether the classification of the image information is rotation-invariant, scaling-invariant or translation-invariant or not?

Unlike FCNN, which processes the image information completely, that is an image is evaluated as a whole, the convolutional layers in connection with the pooling layers are able to store the relevant image information independent of their position and scale in the image. The whole picture is not evaluated at once, but rather the relevant details are first worked out in an abstract manner. The adjacent FC layer can thus classify a cat no matter where, for example, its characteristic ears are.

However, a CNN is not rotationally invariant. Unfortunately, for example, it still does not recognize a twisted coffee mug as a coffee mug because the lid is in the compound of the cup is in the wrong place.

5.4.1 Can the mesh be improved so that it becomes more rotation-invariant, scaling-invariant, or translation-invariant? If yes how?

One can make the mesh more rotationally invariant by adding ent to the mesh provides speaking training images. For example, the input images can be expanded by certain degrees by rotation.

5.4.2 Deeper Question: Is a Full Connected Neural Network, like us used it in chapter 2, rotation invariant, scaling invariant or translation invariant?

No, since it evaluates the image as a whole, it does not matter in which scaling, rotation or position in the image an object to be classified is. You could ¨ Here, too, the network can be made a bit more robust with the corresponding test data, but here you quickly get so many combinatorial possibilities that it is not very practicable. You have to train a lot of pictures.

Reasoning is important here!

5.5 What are parameters and hyperparameters?

Parameters of a CNN are all the weights in the convolution matrices and the FC layers which need to be trained. Hyperparameters are those that affect training. The hyperparameters include, for example, ¨

- The choice of activation function
- The batch size
- The learning rate
- Number of layers

- Pooling function
- Kernel size
- Increment
- Loss function
- Optimization function
- Bias

5.6 What happens in the feed forward and what happens in the backpropa ration?

Build the concepts of initialization of the weights, activation function, error and Calculate gradient, batch on.

Feed forward and back propagation are part of the training process. Of the Feed forward designates the path of the input data through the network from receipt to the output of the results. In the case of CNNs in the field of computer vision, these are images. At the beginning the weights of all layers are included relatively randomly initialized. However, one should choose the weights in a small range around 0, depending on which activation function is used. The error calculated at the end is distributed to the individual weights via backpropagation. The error is always calculated for one batch. This does not mean that there is a back propagation after each individual picture carried out but only when a batch has run through. Following the backpropagation, the weights are updated using the gradient descent method.

5.7 After which step are the weights in the Mini Batch optimization or in the mini batch gradient descent method adjusted?

After a batch has been run through, the averaged errors are carried out the backpropagation distributed to the weights. After that the weights updated with the gradient descent method.

5.8 How can one decide whether a CNN also works for unknown data sets? Training, validation, testing, correct classification and loss rate are possible

Keywords.

A CNN is trained with datasets. The data sets are usually divided into 3 parts. In a training data set, a validation data set and a test data set. The parameters of the network, i.e. the weights in the convolution matrices and the weights of the FC layer are with the Training record trained. The hyper parameters of the model, e.g. the batch size the learning rate etc. are tested with the validation data.

So that a CNN is not overfitted, the CNN is again trained on a test data set after the training model has been completed never seen has verified. If this also results in a high correct classification and a low loss rate, it can be assumed that that the network also works well with other unknown data sets.
One can look at the correct classification rate over the epochs in a graph. A faster increase in correct classification can can also be a sign of overfitting. This usually happens when the data is too similar. One Overfitting is recognized when the level of correct classification cannot be maintained on a test dataset.

5.9 What could be the reason if your self-implemented CNN not working properly?

Give four reasons. First the more important reasons. ¨

- too little training data
- bad training data (e.g. too similar data)
- bad initialization of the weights (disappearing/exploding gradient)
- too few layers
- wrong activation function

5.10 How can you still train a functioning network with “too” few data sets?

Do you suspect that your network isn't working because you don't have enough data sets (images) for training? What possibilities are there to maintain a functioning network. Throw away your own implementation and trying something new is only an alternative if the new slogan ¨ with considerably less data. (Here is also after transfer learning asked that you should be able to explain.)

Answer:

On the one hand, one can try additional ¨ by augmenting the images generate images on which the network is trained. You can, for example Move, scale, rotate, mirror or shear the object. missing Pixel treatment is determined with the Fill mode. for example, pixels could be copied. (leads to streaks) The limits of augmentation can be seen in the non-changeable viewing angles and lighting of the objects. You can also change the lighting or change the image by adding noise. ¨

You can also add virtual objects. This can be done with real textures or with artificial textures. Virtual objects with real textures can be generated, for example, by photogrammetry. You make it

multiple images of an object from different angles. These pictures one joins e.g. by a SURF or ORB procedure.
The quality of the network can also be increased by skilfully mixing the training data and validation data in a K-fold cross-validation even if there is not enough data. It is important here that test data exist that are independent!
Alternatively, you can also try to follow a transfer learning approach. The trick is to take an already trained network and just that FC layer or maybe additionally ¨ 1 or 2 convolutional layers trainable to make and freeze the rest of the parameters. As a result, you need significantly fewer images to achieve good results.

5.11 Deeper question: what is the simulation-to-reality gap? As can you overcome it? ¨

Deeper question: If networks are trained without real images but only with virtual objects, the so-called simulation-to-reality gap has to be overcome ¨ will. Can you explain what is meant by this gap and how to overcome it?

Robotic systems must be trained to work effectively in the real world. However, this is not practicable, since these systems then have a longer ¨ time are useless. Therefore, simulations are used. There is always a certain difference between a simulation and reality.
For example, the colors and textures of objects are not exactly the same or the physics are different in the simulation than in reality. This is called the simulation to reality gap. This difference must be overcome in training ¨ will. It may be that a CV system exploits weaknesses in the simulation to deliver good results there, but not in real use is beneficial. The simulation must therefore be as real as possible. The simulation must behave as realistically as possible. This can be achieved through certain parameters of the simulation. light simulation physics simulation etc.
Answer:

5.12 What options are there for generating additional data (images)?

If the objects that you want to track with neural networks are not contained in one of the known data sets, you have to generate your own images.
Possibilities to generate your own data:

- Take pictures
- Generate images artificially
- Augment images

5.13 The gradients can be found in the network by traversing of the graph or by differentiation of a function F(X).

5.13.1 Explain how the function F(X) arises.

The function F(X) arises as a concatenation of the matrix operations and the application of the activation functions in each layer.

5.13.2 Which function value does the function F(X) deliver and how do you get it I the gradients needed to adjust the weights? ¨

F(X) returns the output value of the network. The gradients are obtained by deriving them from the weights. These are formed by using the chain rule.

5.13.3 Where is the Jacobian matrix used?

The Jakobi matrix describes the derivation of a function involving a vector maps to a vector of the same or different dimensions. This is important for CNN, since the dimensions in the layers can change. It is used when deriving a layer. The derivation after the weighting is then carried out using the chain rule.

5.13.4 What happens after backpropagation?

Following the backpropagation, the weights are updated using an optimizer. This is the gradient descent method here.

5.14 What is meant by vanishing and exploding gradients?

Vanishing gradients are those whose magnitude tends to 0 during backpropagation. Exploding gradients are those whose magnitude is at the forwarding to previous layers in the backpropagation towards infinity walk. Vanishing gradients can arise when the weights on the diagonal of the weight matrix of the individual layers are smaller than 1. Exploding gradients can arise when the weights of the diagonals are greater than 1. This problem becomes more common with deeper meshes greater. ¨

What is the effect of vanishing gradients?

If they occur at the end of the workout, you may not realize you have a problem. The first layers of the mesh will then not more trained as the gradients across the layers become smaller and smaller. The final layers continue to train well.

Result: The correct classification rate may no longer improve. That Web cannot further distinguish images similar to the Web, since the layers that are responsible for working out the details cannot be trained any further.

If this occurs at the beginning of training, then the network is untrainable.

What is the effect of exploding gradients?

With exploding gradients at the beginning, the first layers become very heavily customized. As a result, the network does not learn very well. Since you have a very has a large increment in the direction of the minimum. This may lead to not to the minimum.

What options are there to bring them under control? Name four possibilities and explain them as far as we have done in the lecture.
Answer:

1. Reduce learning rate
2. Skillful initialization of the weights
3. Batch normalization
4. Use of recurrent neural networks

Reduce learning rate to get vanishing or exploding gradients under control

Effects of reducing the learning rate on exploding gradients:
As a result, the layers near the output that are responsible for classification are only trained very slowly. Result: the training takes longer or does not work very well.

Initialization of the weights:

Known methods:

- random
- Xavier / Glorot
- Hey
- ...

The aim of all procedures is the same:
Initialize the weights in a kind of Gaussian distribution around the zero point.

Batch normalization:

With batch normalization layers, the values of the weights of a layer normalized during training. Thus the gradients become uniform ¨ flow through the network. Typically, the batch normalization layer takes place after a convolutional layer or after an FC layer. ..

There are 2 normalization strategies: 1. The truncation of certain undesired values

..

2. Fit gradients linearly in a given area. That's more complicated.
Use of recurrent neural networks:

..

Recurrent networks have memory. The so-called context unit. As a result, outputs are still influenced by inputs that have been processed for a long time. This can make the disappearance of gradients less likely. The gradient is also saved and cannot disappear so easily. The structure of a simple recurrence is illustrated NN in the following figure:

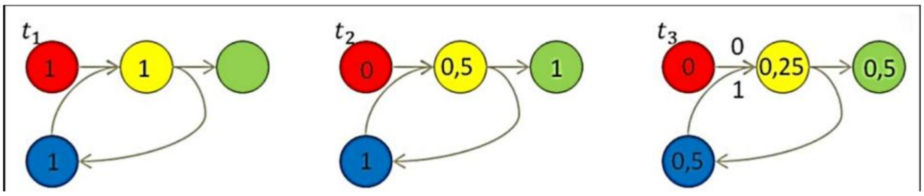

Figure 23: Recurrent neural network

5.15 How does Semantic Segmentation differ from object detection? Give an example of each network?

Semantic segmentation assigns a class to certain parts of an image. So it divides the image into logical areas. It organizes it assigns a specific category to each pixel of an image.
Object detection only aims to find specific objects in an image and does not classify multiple parts of the image. It arranges so not every pixel of an image is assigned a class.
Example nets for semantic segmentation:

- Mask R-CNN & Point Rendering
- Detectron 2

Sample network for object detection: ¨

- YOLO
- Faster R-CNN

5.16 With what kind of network can one make image sequences rate?

Recurrent neural networks can evaluate such image sequences, since the input of several consecutive images due to their

be able to process memories. Several images therefore influence the output of such a network.

5.17 Explain the basic structure of Generative Adversarial Network (GAN).

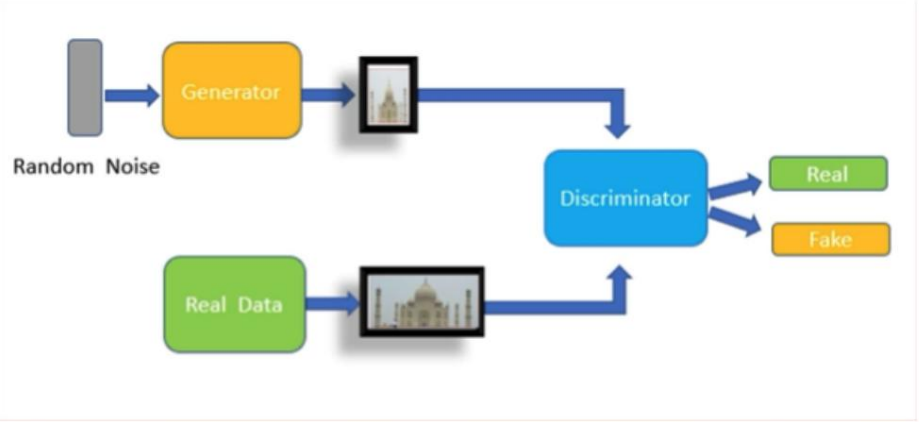

Figure 24: Structure of a GAN

A GAN basically consists of 2 parts. the generator and the discriminator. The aim of this network is for the generator to create such a convincing picture that the discriminator can no longer distinguish whether it is

a real image, e.g. from a database, or a generated object is from the generator.

5.18 How to train a GAN?

When training the GAN, 2 things are trained. The discriminator and the generator. These must be trained together. In addition, both networks alternately freeze again and again. The discriminator and the generator

is trained using supervised learning. The images are labeled according to their source. Errors are then calculated based on the labels that are distributed on the web. And so the two nets become progressive trained. The goal of the discriminator is always to get the images correct

classify as fake or real. The generator has the opposite goal. He wants to change the discriminator and achieve a reverse classification. Accordingly, his error is calculated differently.

A training loop according to Naoki Shibuya was described as follows:

1. Set the discriminator trainable.

2. The discriminator with real MNIST images and that from the generator fake pictures produced.

3. Freeze the discriminator, i.e. set it untrainable.

4. Train the generator as part of the GAN. That means we feed it GAN with fake images from the generator and real images from the MNIST data set. The frozen discriminator classifies the images. With the resulting error, the weights in the generator are corrected.

5. After a certain number of training runs, start again with step 1.