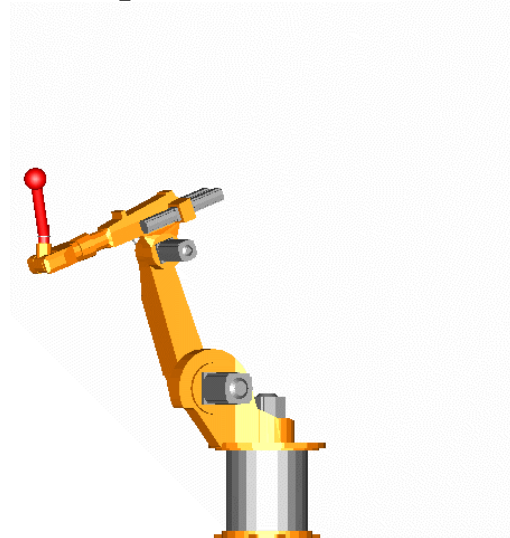
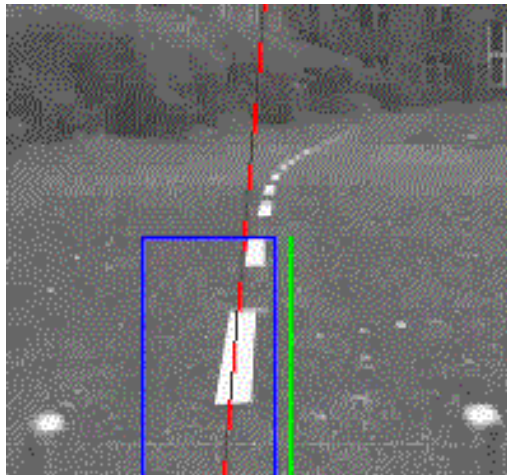
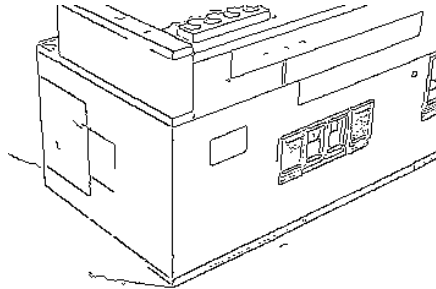
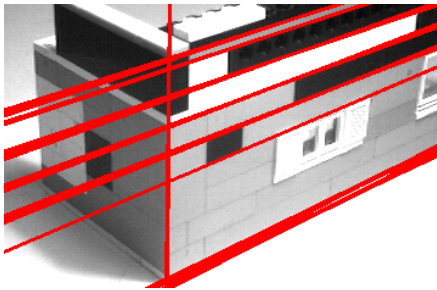
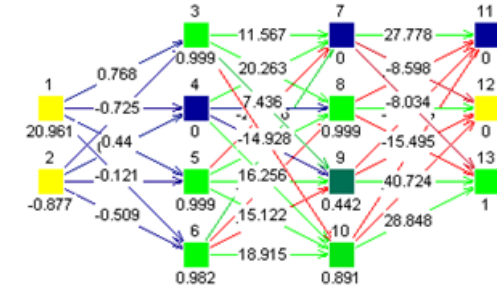


Robot Vision

Prof. Dr.-Ing. Andreas Meisel





Vorgehensweise

Vortragsfolien → Pub oder owncloud

Vertiefungen → Pub oder owncloud

Übungsaufgaben → Pub oder owncloud

frühere Klausuren → Pub oder owncloud

Praktikumsversuche → Pub oder owncloud

Public-Verzeichnis der Informatik

https://users.informatik.haw-hamburg.de/~meisel/pub/Vorl_RobotVision/

Owncloud

<https://owncloud.informatik.haw-hamburg.de/index.php/s/1muXfmiVv541MCB>

PW: MSL_LV



Literatur

Grundlagenwerke

Digital Image Processing, Rafael. C. Gonzalez, Richard E. Woods,
Pearson Education International

Digitale Bildverarbeitung, William Burger, Mark .J. Burge
Springer

Weitergehende Literatur

Digitale Bildverarbeitung, Bernd Jähne,
Springer

Pattern Classification, Richard O. Duda, Peter E. Hart, David G. Stork,
Wiley Interscience

Neural Networks and Learning Machines, Simon Haykin
Pearson Education

Multiple View Geometry in Computer Vision, Richard Hartley, Anrew Zisserman
Cambridge



Weiterführende (Online-) Publikationen + Recherche

IEEE Xplore (aus Hochschulnetz)

ACM Digital Library (aus Hochschulnetz)

Springer: International Journal of Computer Vision (aus Hochschulnetz)

Citeseer (frei)

Google Scholar (frei)



Eingesetzte Softwarewerkzeuge / Bibliotheken

Praktikum:

LTILib (s. Sourceforge) entweder mit Linux / Windows (+ VC 2008)

JavaNNS (Neuronale Netze Simulator)

Vorlesung:

MAPLE 14 (Campuslizenz)

Matlab (Campuslizenz) + Image Processing Toolbox + Neural Network Toolbox

OpenCV (live Bildverarbeitungsbeispiele, C/C++)



Was ist Robot Vision ?

= „*Maschinensehen*“ oder auch
„*automatische Informationsgewinnung aus Bildern*“

Aufgabe	Bezeichnung	Ergebnis
<u>Was</u> ist im Bild zu sehen?	<i>Identifikation</i>	Klasse
<u>Wo</u> befindet sich das Objekt?	<i>Lokalisierung</i>	Position
<u>Wie</u> liegt das Objekt?	<i>Lageerkennung</i>	Lagewinkel
<u>Wie bewegt</u> sich das Objekt?	<i>Bewegungs- erkennung</i>	Geschwindigkeit, Winkelgeschwindigkeit
<u>Welche Maße</u> hat das Objekt?	<i>Bildmesstechnik</i>	Abstände, Durchmesser, Eckenwinkel,
Ist das <u>Objekt</u> qualitativ <u>ok</u> ?	<i>Inspektion</i>	gut/schlecht

Erzeugen einer **symbolischen Beschreibung** von dem, was im Bild dargestellt ist („*Bildmustererkennung*“).



1. Stufen der Mustererkennung

1.1 Merkmale, Muster, Klassen ?

1.1.1 Allgemeine Definition

Merkmale erlauben die Unterscheidung zwischen „*Dingen*“ (Objekten).

Ein **Muster** ist eine typische Merkmalsausprägung,

- die charakteristisch für eine bestimmte **Klasse** von Dingen ist und
- eine Unterscheidung von anderen Klassen ermöglicht.

Beispiel: Klasse der lateinischen Buchstaben A

A A A A A A A A A A A A A A ...

Frage: Durch welche Merkmale unterscheidet sich die Klasse der Buchstaben A von der Klasse der Buchstaben B?



1.1.2 Technische Definition

Computer können nur Größen verarbeiten, die sensorisch erfassbar sind.

Beispiele:

Spracherkennung	←	Mikrofon
Gesichtserkennung	←	Kamera
Personenerkennung	←	Fingerdrucksensor

Die für die Mustererkennung verwendeten Merkmale können als Vektor von Meßgrößen aufgefasst werden.

$$\vec{x} = [x_1, x_2, x_3, \dots, x_n]^T$$

Merkmalsvektor
(n-dimensional)

n Merkmale

→ beschreiben Objekteigenschaften

**Beispiel:** Merkmale von Euromünzen (hier als 3-dimensionaler Merkmalsvektor)

					
Gew./g	8.5	7.5	7.8	5.74	4.1
Durchm./mm	25.75	23.25	24.25	22.25	19.75
Dicke/mm	2.2	2.33	2.38	2.14	1.93

Merkmale

Merkmalsvektor
(dreidimensional)



1.1.3 Wahl der Merkmale

1. hohe Unterscheidungsfähigkeit (hoher Informationsgehalt)
2. Unabhängigkeit (geringe Redundanz)
3. möglichst wenig Merkmale (niedrigdimensionaler Merkmalsvektor)

Beispiel: in dieser Form schlecht für die Klassifikation

Die Münzen werden als 1000x1000-Bild aufgenommen.

Die Grauwerte der 10^6 Bildpunkte bilden einen 10^6 -dimensionalen Merkmalsvektor.

1.2 Was ist Mustererkennung?

1.2.1 Arten der Mustererkennung

a) Mustererkennung =
"Wiedererkennen von Dingen"

Zum Wiedererkennen gehört Wissen.
Wissen entsteht durch Lernen.

D.h., man muss die Dinge kennen,
zwischen denen zu unterscheiden ist.

Beispiel:

Welcher der nebenstehenden
Vögel ist eine "Rohrdommel"?



b) Mustererkennung = „*charakteristische Merkmalsausprägungen erkennen*“

Zur Unterscheidung zwischen verschiedenen Mustern gehört kein Vorwissen.



--> Klassenbildung ("*Clustering*")

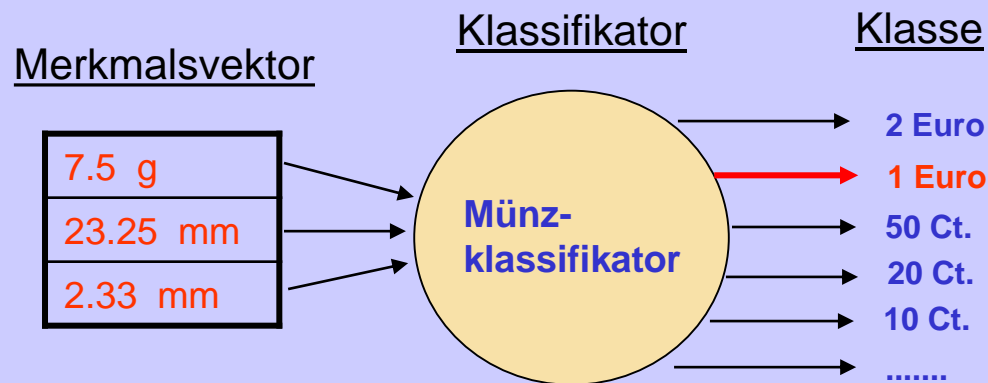


1.2.2 Definition des Begriffs "Mustererkennung":

Mustererkennung ist die Theorie von der bestmöglichen Zuordnung einer unbekannten Merkmalsausprägung zu einer Klasse.

Die Komponente (Softwaremodul), welche die Zuordnungsaufgabe durchführt, wird als **Klassifikator** bezeichnet.

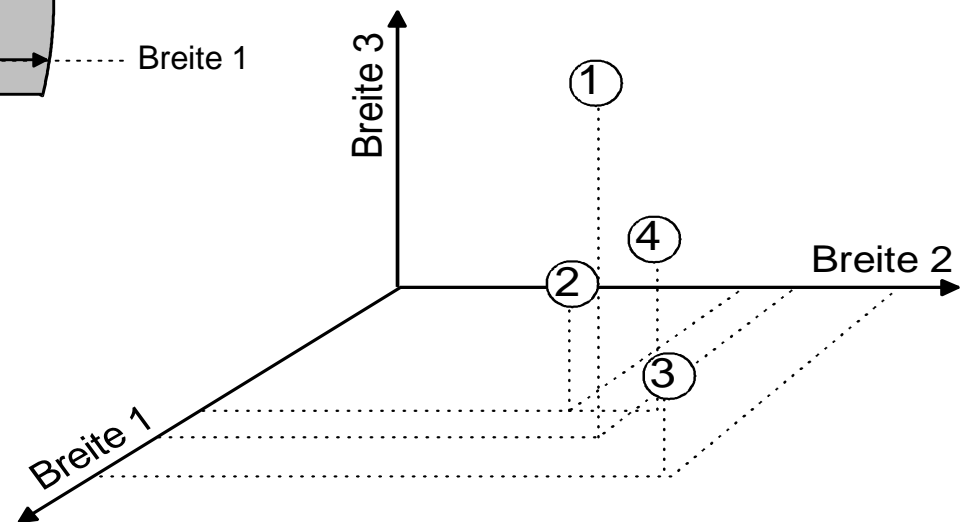
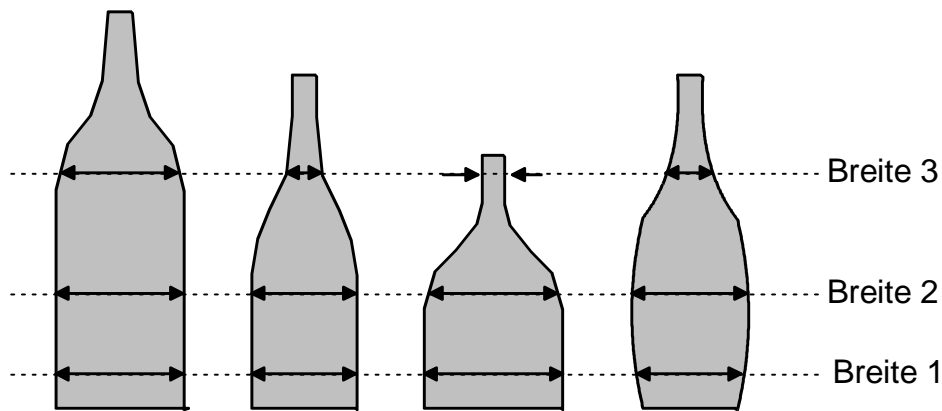
Beispiel:



1.3 Einfache Mustererkennungsaufgaben

1.3.1 Beispiel: Flaschensortierung

Auswertung der Querschnittsbreiten auf 3 Höhen → 3-dimensionaler Merkmalsraum





1.3.2 Was macht eine Mustererkennungsaufgabe einfach ?

1. Die Zuordnungsregeln eines Musters zu einer Klasse sind bekannt.
2. Die Zuordnungsregeln sind leicht algorithmisch formulierbar.
3. Die Zahl der Merkmale ist gering.

Beispiele:

- Gut/Schlecht-Klassifizierung abhängig von Toleranzen
- Sortierung in Größen- und Güteklassen

typ. Lösungsansatz: → regelbasierte Klassifikatoren



1.3.3 Beispiel: Regelbasierter Klassifikator (Boxklassifikator)

```
/* Münzsortierer */  
  
.....  
.....  
  
if (    Gewicht > 8.4  &&  Gewicht < 8.6  
    && Durchms > 25.5 &&  Durchms < 26.0  
    && Dicke   >  2.0 &&  Dicke   <  2.4 )  
    Klasse = EURO_2;  
else if (Gewicht > 7.4  &&  Gewicht < 7.6  
    && Durchms  > 23.0 &&  Durchms < 23.5  
    && Dicke    >  2.1 &&  Dicke   <  2.5 )  
    Klasse = EURO_1;  
else if .....  
    .....
```



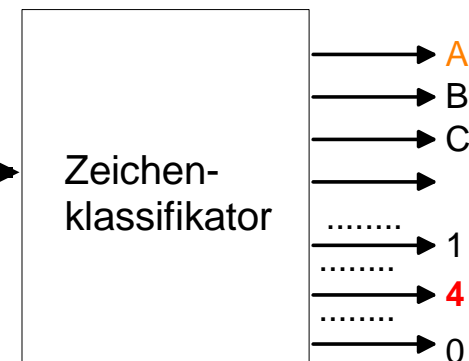

1.4 Komplexe Mustererkennungsaufgaben

1.4.1 Beispiel: Lesen von Autokennzeichen

1. Das Kfz-Kennzeichen wird im Bild lokalisiert.
2. Die einzelnen Zeichen des Kennzeichens werden separiert.
3. Die separierten Zeichen auf die Eingabematrix (z.B. 16 x 24) transformiert.



4. Klassifikation der Einzelzeichen



Bildpunktmatrix
(z.B. 16 x 24) → 384-dimens. Merkmalsvektor

Weitere Beispiele :

- Zuordnung von Fingerabdrücken zu Personen



- Unterschriftprüfung (z.B. Prüfen von Schecks)
- Gesichtserkennung
- Hinderniserkennung für autonome Roboter (z.B. FAUST-Projekt)
- Identifikation von Straßenschildern für Fahrerassistenzsysteme
- visuelle Navigation (z.B. FAUST-Projekt)



1.4.2 Was macht eine Mustererkennungsaufgabe komplex ?

1. Es gibt keine einfach formalisierbaren Regeln für die Zuordnung eines Musters zu einer Klasse.
2. Die Zahl der Merkmale ist hoch.

Typ. Lösungsansatz

1. Berechnung aussagekräftiger Merkmale aus den Eingangsdaten d.h. **Dimensionsreduktion** des Merkmalsraumes.
2. Festlegen des Klassifikatorverhaltens anhand von repräsentativen Mustern (Repräsentanten, Trainingsdaten). → **lernende** Klassifikatoren



1.4.3 Beispiel: Dimensionsreduktion des Merkmalsvektors durch Merkmalsextraktion

Beispiel: Identifikation von Fingerabdrücken

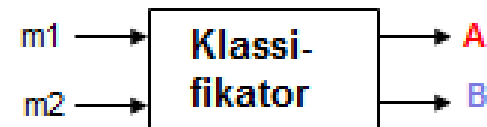
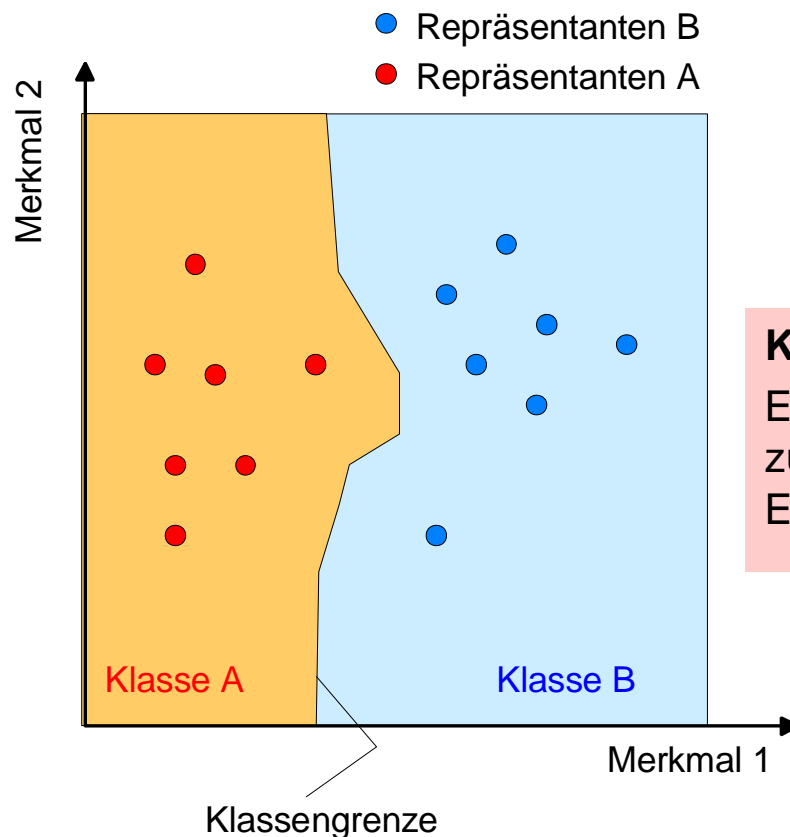


Bei direkter Auswertung des Bildes durch einen Klassifikator, hätte der Merkmalsvektor z.B. die Dimension: 512×256 (Bildgröße) = ca. **131072**

Durch Extraktion aussagekräftiger Merkmale (z.B. End- und Gabelpunkte) reduziert sich die Dimension des Merkmalsvektors auf **20 - 40** (= relative Anordnung der End- und Gabelpunkte zueinander)

1.4.4 Einfaches Beispiel: *Klassifikation anhand von Beispielen mit dem "Nearest-neighbor-Klassifikator"*

Die Repräsentanten (= Lernbeispiele) bestimmen die Klassengrenzen zwischen den Klassen.



Klassifikationsprinzip:

Ein Muster wird derjenigen Klasse zugeordnet, zu deren Repräsentanten es die kürzeste Entfernung hat.



BEISPIEL: Codeskizze eines einfachen Nearest-neighbor-Klassifikators

```
struct Muster {  
    double m1;  
    double m2;  
    char Klasse; };
```

// Aufbau eines Musters (Strukturdeklaration)

```
struct Muster Repraesentant[N] = {  
    0.2, 0.5, `A`,  
    0.3, 0.7, `A`,  
    . . . . usw. };
```

// Liste der Repräsentanten

```
double d_min = SEHR_GROSSE_ZAHL; // Initialisierung  
read(&aktuellesMuster); // Lesen des zu klassifizierenden Musters  
for(int i = 0; i < N; i++){ // Abstand zu allen Repraesent. bestimmen  
    d = Abstand(&aktuellesMuster, &Repraesentant[i] );  
    if(d < d_min) {  
        nearestNeighbor = i;  
        d_min = d;  
    }  
}  
printf("Musterklasse = %c", Repraesentant[nearestNeighbor].Klasse);
```



1.4.5 Einige Klassifikationskonzepte

Regelbasierte Klassifikatoren

- binäre Regeln (Boxklassifikator)
- Fuzzy-logic

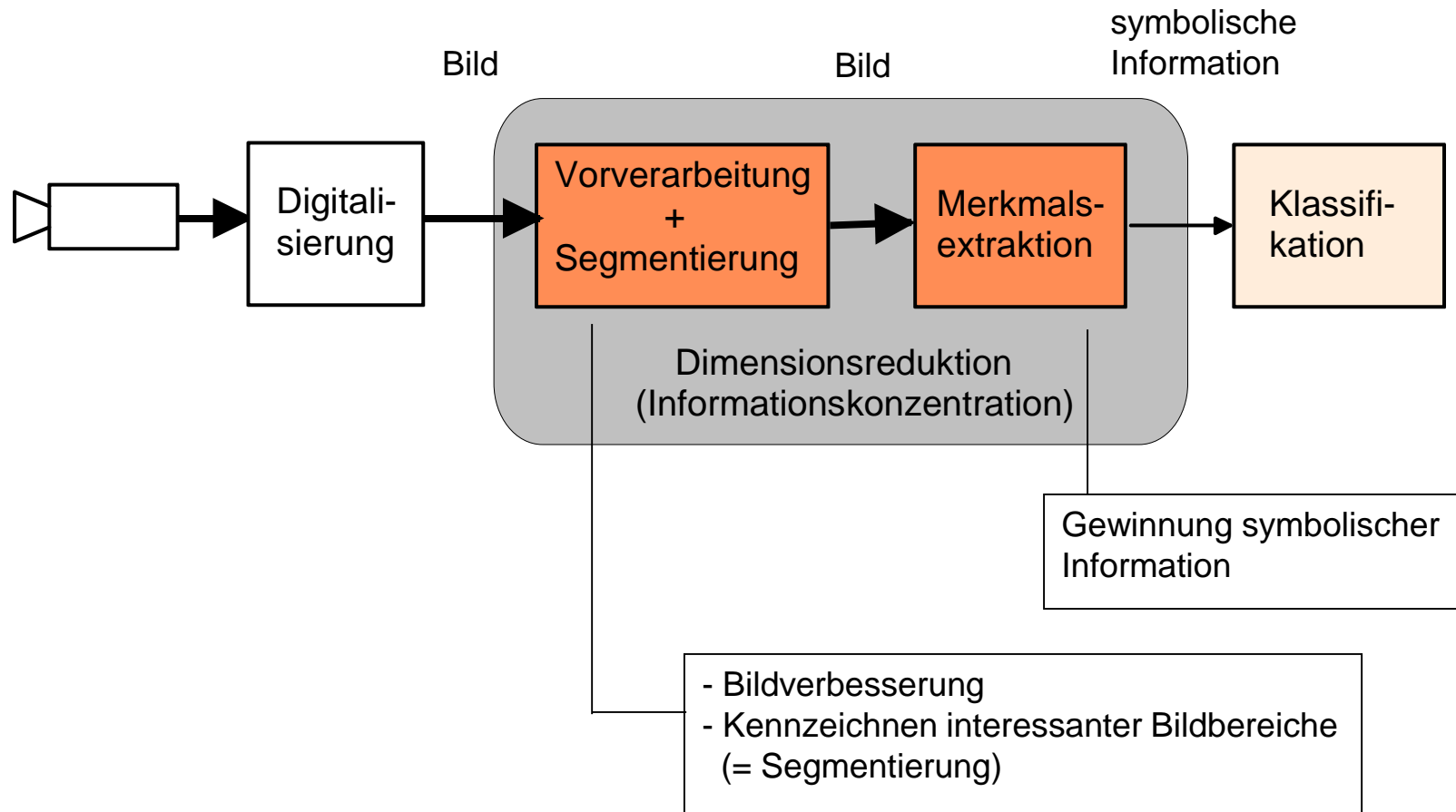
Lernende Klassifikatoren

- Nearest-Neighbor-Klassifikator
- Statistische Klassifikatoren (Maximum-Likelihood, Bayes-Klassifikator)
- Neuronale Netzwerke
- Radiale-Basisfunktionen-Netzwerke (RBF)
- Support-Vector-Machines
-

→ wird später vertieft



1.5 Bildverarbeitungs-basierte Mustererkennung





1.6 Interdisziplinarität der Bildverarbeitung

